FreeBSD Handboek

Unresolved directive in book.adoc - include::introduction.adoc[]

Inhoudsopgave

Voorwoord . . . 10
Bedoeld publiek . . . 10
Wijzigingen ten opzichte van de derde editie . . . 10
Wijzigingen ten opzichte van de tweede editie (2004) . . . 10
Veranderingen ten opzichte van de eerste editie (2001) . . . 11
De opbouw van dit boek . . . 12
Overeenkomsten in dit boek . . . 15
Dankwoorden . . . 17
I: Beginnen . . . 18
1. Introductie . . . 19
1.1. Overzicht . . . 19
1.2. Welkom bij FreeBSD! . . . 19
1.3. Over het FreeBSD Project . . . 22
2. FreeBSD 9.X en nieuwer installeren . . . 28
2.1. Overzicht . . . 28
3. UNIX® beginselen . . . 29
3.1. Overzicht . . . 29
3.2. Virtuele consoles en terminals . . . 29
3.3. Rechten . . . 33
3.4. Mappenstructuur . . . 38
3.5. Organisatie van schijven . . . 40
3.6. Het koppelen en ontkoppelen van bestandssystemen . . . 47
3.7. Processen . . . 50
3.8. Daemons, signalen en het stoppen van processen . . . 52
3.9. Shells . . . 54
3.10. Teksteditors . . . 56
3.11. Apparaten en apparaatnodes . . . 57
3.12. Binaire formaten . . . 57
3.13. Meer informatie . . . 59
4. Applicaties installeren: pakketten en ports . . . 62
4.1. Overzicht . . . 62
4.2. Overzicht van softwareinstallatie . . . 62
4.3. Applicaties zoeken . . . 64
4.4. Het pakkettensysteem gebruiken . . . 66
4.5. De Portscollectie gebruiken . . . 69
4.6. Activiteiten na het installeren . . . 81
4.7. Omgaan met kapotte ports . . . 82
5. Het X Window systeem . . . 84

5.1. Overzicht . . . 84
5.2. X begrijpen . . . 84
5.3. X11 installeren . . . 87
5.4. X11 instellen . . . 88
5.5. Lettertypen gebruiken in X11 . . . 94
5.6. De X beeldschermmanager . . . 98
5.7. Bureaubladomgevingen . . . 101
II: Algemene taken . . . 106
6. Bureaubladapplicaties . . . 107
6.1. Overzicht . . . 107
6.2. Browsers . . . 107
6.3. Productiviteit . . . 113
6.4. Documentviewers . . . 117
6.5. Financiën . . . 119
6.6. Samenvatting . . . 121
7. Multimedia . . . 122
7.1. Overzicht . . . 122
7.2. Geluidskaart installeren . . . 123
7.3. MP3 audio . . . 128
7.4. Video afspelen . . . 130
7.5. TV-kaarten installeren . . . 140
7.6. MythTV . . . 141
7.7. Scanners . . . 142
8. De FreeBSD-kernel instellen . . . 148
8.1. Samenvatting . . . 148
8.2. Redenen om een aangepaste kernel te bouwen . . . 148
8.3. De systeemhardware vinden . . . 149
8.4. Kernel stuurprogramma's, subsystemen, en modules . . . 150
8.5. Bouwen en installeren van een aangepaste kernel . . . 151
8.6. Het instellingenbestand . . . 154
8.7. Problemen oplossen . . . 171
9. Afdrukken . . . 173
9.1. Overzicht . . . 173
9.2. Inleiding . . . 173
9.3. Standaardinstallatie . . . 174
9.4. Geavanceerde printerinstallatie . . . 189
9.5. Printers gebruiken . . . 220
9.6. Alternatieven voor het standaard wachtrijsysteem . . . 229
9.7. Problemen oplossen . . . 229
10. Linux® binaire compatibiliteit . . . 234
10.1. Overzicht . . . 234

10.2. Installatie . . . 234
10.3. Mathematica® installeren . . . 238
10.4. Maple™ installeren . . . 240
10.5. MATLAB® installeren . . . 243
10.6. Oracle® installeren . . . 246
10.7. Gevorderde onderwerpen . . . 249
III: Systeembeheer . . . 252
11. Instellingen en optimalisatie . . . 253
11.1. Overzicht . . . 253
11.2. Initiële instellingen . . . 253
11.3. Hoofdinstellingen . . . 255
11.4. Toepassingen instellen . . . 256
11.5. Diensten starten . . . 256
11.6. `cron` instellen . . . 258
11.7. Gebruik van rc met FreeBSD . . . 260
11.8. Netwerkkaarten instellen . . . 262
11.9. Virtuele hosts . . . 268
11.10. De systeemlogger syslogd configureren . . . 269
11.11. Instellingenbestanden . . . 273
11.12. Optimaliseren met sysctl . . . 275
11.13. Harde schijven optimaliseren . . . 276
11.14. Fijnafstemming van kernellimieten . . . 281
11.15. Wisselbestandruimte toevoegen . . . 284
11.16. Energie- en bronnenbeheer . . . 286
11.17. FreeBSD ACPI gebruiken en debuggen . . . 288
12. Het FreeBSD opstartproces . . . 296
12.1. Overzicht . . . 296
12.2. Het bootprobleem . . . 296
12.3. De bootmanager en opstartstadia . . . 297
12.4. Interactie met de kernel tijdens opstarten . . . 304
12.5. Device hints . . . 304
12.6. Init: start van procesbesturing . . . 305
12.7. Afsluitvolgorde . . . 306
13. Beveiliging . . . 308
13.1. Overzicht . . . 308
13.2. Introductie . . . 308
13.3. FreeBSD beveiligen . . . 310
13.4. DES, Blowfish, MD5, SHA256, SHA512 en crypt . . . 319
13.5. Eenmalige wachtwoorden . . . 320
13.6. TCP Wrappers . . . 324
13.7. Kerberos5 . . . 327

13.8. OpenSSL . . . 335
13.9. VPN via IPsec . . . 338
13.10. OpenSSH . . . 345
13.11. Bestandssysteem toegangscontrolelijsten (ACLs) . . . 351
13.12. Monitoren van beveiligingsproblemen met andere software . . . 353
13.13. FreeBSD beveiligingswaarschuwingen . . . 355
13.14. Procesaccounting . . . 357
14. Jails . . . 359
14.1. Overzicht . . . 359
14.2. Termen en begrippen van jails . . . 359
14.3. Introductie . . . 360
14.4. Creeëren en controleren van jails . . . 361
14.5. Optimaliseren en administratie . . . 363
14.6. Toepassing van jails . . . 365
15. Verplichte Toegangscontrole (MAC) . . . 372
15.1. Overzicht . . . 372
15.2. Sleuteltermen in dit hoofdstuk . . . 373
15.3. Uitleg over MAC . . . 375
15.4. MAC-labels begrijpen . . . 376
15.5. De beveiligingsconfiguratie plannen . . . 381
15.6. Module-instellingen . . . 382
15.7. MAC-module seeotheruids . . . 383
15.8. MAC-module bsdextended . . . 383
15.9. MAC-module ifoff . . . 384
15.10. MAC-module portacl . . . 385
15.11. MAC-module partition . . . 387
15.12. MAC-module Multi-Level Security . . . 388
15.13. MAC-module Biba . . . 390
15.14. MAC-module LOMAC . . . 391
15.15. Nagios in een MAC-jail . . . 392
15.16. Gebruikers afsluiten . . . 396
15.17. Problemen oplossen met het MAC-raamwerk . . . 397
16. Security Event Auditing . . . 399
16.1. Overzicht . . . 399
16.2. Sleutelwoorden in dit hoofdstuk . . . 400
16.3. Installeren van audit ondersteuning. . . . 400
16.4. Audit Configuratie . . . 401
16.5. Het audit subsysteem beheren. . . . 404
17. Opslag . . . 408
17.1. Overzicht . . . 408
17.2. Apparaatnamen . . . 408

17.3. Schijven toevoegen . . . 409
17.4. RAID . . . 412
17.5. USB-opslagapparaten . . . 417
17.6. Optische media (CD's) aanmaken en gebruiken . . . 420
17.7. Optische media (DVD's) aanmaken en gebruiken . . . 427
17.8. Diskettes aanmaken en gebruiken . . . 433
17.9. Gegevensbanden aanmaken en gebruiken . . . 435
17.10. Naar diskettes back-uppen . . . 436
17.11. Back-up strategieën . . . 437
17.12. Back-upbeginselen . . . 438
17.13. Netwerk-, geheugen-, en bestandsgebaseerde bestandssystemen . . . 443
17.14. Snapshots van bestandssystemen . . . 446
17.15. Bestandssysteemquota . . . 447
17.16. Schijfpartities versleutelen . . . 451
17.17. Het versleutelen van de wisselbestand ruimte . . . 459
17.18. Highly Available Storage (HAST) . . . 460
18. GEOM. Modulair schijftransformatie raamwerk . . . 470
18.1. Overzicht . . . 470
18.2. GEOM inleiding . . . 470
18.3. RAID0 - aaneengeschakeld . . . 470
18.4. RAID1 - spiegelen . . . 472
18.5. RAID3 - Striping op byte-niveau met toegewijde pariteit . . . 482
18.6. GEOM Gate netwerk apparaten . . . 484
18.7. Het labelen van schijven . . . 485
18.8. UFS logboeken door middel van GEOM . . . 488
19. Ondersteuning van bestandssystemen . . . 490
19.1. Overzicht . . . 490
19.2. Het Z File System (ZFS) . . . 490
19.3. Linux® bestandssystemen . . . 501
20. Virtualisatie . . . 503
20.1. Overzicht . . . 503
20.2. FreeBSD als een gast-besturingssysteem . . . 503
20.3. FreeBSD als een gastheer-besturingssysteem . . . 535
21. Lokalisatie - I18N/L10N gebruiken en instellen . . . 538
21.1. Overzicht . . . 538
21.2. Beginselen . . . 538
21.3. Lokalisatie gebruiken . . . 539
21.4. I18N-programma's compileren . . . 546
21.5. FreeBSD lokaliseren naar talen . . . 546
22. FreeBSD updaten en upgraden . . . 550
22.1. Overzicht . . . 550

22.2. FreeBSD Update . . . 550
22.3. Portsnap: een updategereedschap voor de Portscollectie . . . 558
22.4. De documentatie bijwerken . . . 559
22.5. Een ontwikkelingstak volgen . . . 564
22.6. Broncode synchroniseren . . . 568
22.7. De "wereld" opnieuw bouwen . . . 569
22.8. Het verwijderen van overbodige bestanden, directories en bibliotheken . . . 586
22.9. Meerdere machines bijwerken . . . 588
23. DTrace . . . 590
23.1. Overzicht . . . 590
23.2. Implementatieverschillen . . . 590
23.3. Ondersteuning voor DTrace aanzetten . . . 591
23.4. DTrace gebruiken . . . 592
23.5. De taal D . . . 595
IV: Netwerkcommunicatie . . . 596
24. Seriële communicatie . . . 597
24.1. Overzicht . . . 597
24.2. Inleiding . . . 597
24.3. Terminals . . . 602
24.4. Inbeldienst . . . 607
24.5. Uitbeldienst . . . 616
24.6. Seriële console opzetten . . . 620
25. PPP en SLIP . . . 630
25.1. Overzicht . . . 630
25.2. Gebruikmaken van gebruiker-PPP . . . 630
25.3. Kernel-PPP gebruiken . . . 644
25.4. Het problemen oplossen van PPP-verbindingen . . . 652
25.5. PPP gebruiken over Ethernet (PPPoE) . . . 656
25.6. Gebruik maken van PPP over ATM (PPPoA) . . . 658
25.7. Gebruik maken van SLIP . . . 662
26. Elektronische mail . . . 672
26.1. Overzicht . . . 672
26.2. Gebruik maken van elektronische mail . . . 672
26.3. sendmail instellen . . . 675
26.4. De Mail Transfer Agent vervangen . . . 678
26.5. Problemen oplossen . . . 680
26.6. Geavanceerde onderwerpen . . . 683
26.7. SMTP met UUCP . . . 686
26.8. Instellen om alleen te versturen . . . 688
26.9. Mail gebruiken met een inbelverbinding . . . 689
26.10. SMTP-authenticatie . . . 690

26.11. Mail User Agents . . . 692
26.12. fetchmail gebruiken . . . 700
26.13. procmail gebruiken . . . 701
27. Netwerkdiensten . . . 703
27.1. Overzicht . . . 703
27.2. De inetd"Super-Server" . . . 703
27.3. Netwerkbestandssysteem (NFS) . . . 708
27.4. Netwerkinformatiesysteem (NIS/YP) . . . 715
27.5. Automatisch netwerk instellen (DHCP) . . . 733
27.6. Domeinnaamsysteem (DNS) . . . 738
27.7. Apache HTTP server . . . 757
27.8. File Transfer Protocol (FTP) . . . 762
27.9. Bestands- en printdiensten voor Microsoft® Windows® cliënten (Samba) . . . 764
27.10. Tijd synchroniseren met NTP . . . 767
27.11. Hosts op afstand loggen met `syslogd` . . . 770
28. Firewalls . . . 775
28.1. Inleiding . . . 775
28.2. Firewallconcepten . . . 775
28.3. Firewallsoftware . . . 776
28.4. De OpenBSD Packet Filter (PF) en ALTQ . . . 776
28.5. De IPFILTER (IPF) firewall . . . 780
28.6. IPFW . . . 801
29. Geavanceerde netwerken . . . 822
29.1. Samenvatting . . . 822
29.2. Gateways en routes . . . 822
29.3. Draadloze netwerken . . . 829
29.4. Bluetooth . . . 851
29.5. Bridging . . . 861
29.6. Verbindingsaggregatie en failover . . . 867
29.7. Schijfloos werken . . . 873
29.8. Met PXE en een NFS-root-bestandssysteem opstarten . . . 880
29.9. ISDN . . . 884
29.10. Network Address Translation . . . 889
29.11. IPv6 . . . 893
29.12. Asynchronous Transfer Mode (ATM) . . . 898
29.13. Common Address Redundancy Protocol (CARP) . . . 900
V: Bijlagen . . . 903
Bijlage A: FreeBSD verkrijgen . . . 904
A.1. CD-ROM en DVD uitgevers . . . 904
A.2. FTP sites . . . 906
A.3. Subversion-sites . . . 912

A.4. Anonieme CVS . . . 913
A.5. CTM gebruiken . . . 916
A.6. CVSup gebruiken . . . 920
A.7. CVS labels . . . 933
A.8. rsync sites . . . 940
Bijlage B: Bibliography . . . 942
B.1. Boeken & tijdschriften over FreeBSD . . . 942
B.2. Voor gebruikers . . . 943
B.3. Voor beheerders . . . 943
B.4. Voor programmeurs . . . 943
B.5. Dieper in het besturingssysteem . . . 944
B.6. Over beveiliging . . . 944
B.7. Over hardware . . . 945
B.8. UNIX® geschiedenis . . . 945
B.9. Tijdschriften en periodieken . . . 946
Bijlage C: Bronnen op Internet . . . 947
C.1. Mailinglijsten . . . 947
C.2. Usenet-nieuwsgroepen . . . 966
C.3. World wide webservers . . . 967
C.4. Email-adressen . . . 970
Bijlage D: PGP sleutels . . . 971
D.1. Beambten . . . 971
Colofon . . . 980

Voorwoord

Bedoeld publiek

De nieuwkomers bij FreeBSD zullen zien dat de eerste sectie van dit boek ze begeleidt door de FreeBSD installatieprocedure en de geleidelijke introductie in de concepten van UNIX®. Om deze sectie goed te kunnen doorlopen is meer nodig dan de wens om te ontdekken en de mogelijkheid om nieuwe concepten op te nemen wanneer ze geïntroduceerd worden.

De tweede, veel grotere, sectie van het handboek is een uitvoerige referentie naar alle mogelijke (relevante) onderwerpen die interessant zijn voor FreeBSD systeembeheerders. Sommige van deze hoofdstukken adviseren mogelijk om eerdere documentatie te lezen. Dit wordt aangegeven in de samenvatting aan het begin van elk hoofdstuk.

Voor een lijst van extra bronnen van informatie zie Bibliografie.

Wijzigingen ten opzichte van de derde editie

De huidige online versie van het Handboek representeert de gezamenlijke inspanning van vele honderden bijdragende vrijwilligers van de laatste 10 jaar. Hieronder staan enkele van de belangrijke wijzigingen sinds de tweedelige derde editie in 2004 werd uitgegeven:

- DTrace, DTrace, is toegevoegd met informatie over het krachtige prestatie-analysegereedschap DTrace.
- Ondersteuning van bestandssystemen, Ondersteuning voor bestandssystemen, is toegevoegd met informatie over vreemde bestandssystemen in FreeBSD, zoals ZFS van Sun™.
- Security Event Auditing, Beveiligingsgebeurtenissen auditen, is toegevoegd om de nieuwe auditing-mogelijkheden van FreeBSD te bespreken en het gebruik ervan uit te leggen.
- Virtualisatie, Virtualisatie, is toegevoegd met informatie over het installeren van FreeBSD op virtualisatiesoftware.
- FreeBSD 9.X en nieuwer installeren, FreeBSD 9.*X* en nieuwer installeren, is toegevoegd om het installeren van FreeBSD met het nieuwe installatiegereedschap, bsdinstall te behandelen.

Wijzigingen ten opzichte van de tweede editie (2004)

De derde editie was het resultaat van meer dan twee jaar werk van de toegewijde leden van het FreeBSD Documentation Project. De gedrukte editie werd zo groot dat het noodzakelijk was om het was om het als twee afzonderlijke delen te publiceren. Hieronder staan de grootste veranderingen in deze nieuwe editie:

- Configuration et optimisation, Instellingen en optimalisatie, is uitgebreid met nieuwe informatie over ACPI power en resource management, het systeemhulpprogramma `cron` en er staan meer opties voor het optimaliseren van de kernel beschreven.
- Sécurité, Beveiliging, is uitgebreid met meer informatie over virtuele private netwerken (VPN's), toegangscontrolelijsten voor het bestandssysteem (ACL's) en beveiligingswaarschuwingen.

- Mandatory Access Control, Verplichte toegangscontrole (MAC), is een nieuw hoofdstuk in deze editie. Er wordt in uitgelegd wat MAC is en hoe het gebruikt kan worden om FreeBSD te beveiligen.
- Stockage des données, Opslag, is uitgebreid met informatie over USB opslagapparaten, snapshots van bestandssystemen, bestandssystemen op basis van bestanden en het netwerk en versleutelde partities op schijven.
- Aan PPP et SLIP, PPP en SLIP, is een paragraaf toegevoegd over problemen oplossen.
- Courrier électronique, E-mail, is uitgebreid met informatie over alternatieve transport programma's, SMTP authenticatie, UUCP, fetchmail, procmail en een aantal andere gevorderde onderwerpen.
- Serveurs réseau, Netwerkdiensten, is nieuw in deze editie. Dit hoofdstuk bevat informatie over het opzetten van een Apache HTTP Server, ftpd en het opzetten van een server voor Microsoft® Windows® clients met Samba. Een aantal paragrafen uit Geavanceerd netwerken, Geavanceerde Netwerken, zijn om reden van presentatie naar dit hoofdstuk verplaatst.
- Geavanceerd netwerken, Netwerken voor gevorderden, is uitgebreid met informatie over het gebruik van Bluetooth® apparaten met FreeBSD, het opzetten van draadloze netwerken en Asynchronous Transfer Mode (ATM) netwerken.
- Er is een termenoverzicht toegevoegd als centrale locatie voor definities van technische termen die in dit boek gebruikt worden.
- Tenslotte zijn er nog veel esthetische wijzigingen doorgevoerd aan tabellen en figuren in het boek.

Veranderingen ten opzichte van de eerste editie (2001)

Deze tweede editie is een optelsom van meer dan twee jaar werk door vaste leden van het FreeBSD Documentation Project. Het volgende zijn de grote wijzigingen in deze editie:

- Er is een complete INDEX toegevoegd.
- Alle ASCII-figuren zijn vervangen door grafische diagrammen.
- Aan elk hoofdstuk is een standaardsamenvatting toegevoegd om een snel overzicht te geven welke informatie zich in het hoofdstuk bevindt en wat de lezer geacht wordt te weten.
- De inhoud is logisch ingedeeld in drie delen: "Starten", "Systeembeheer" en "Appendix".
- FreeBSD installeren op FreeBSD 8.X en eerder ("FreeBSD installeren") is compleet herschreven met veel schermafdrukken erbij om het makkelijker te maken voor nieuwe gebruikers om greep te krijgen op de tekst.
- UNIX® beginselen ("UNIX® beginselen") is uitgebreid met extra informatie over processen, daemons en signalen.
- Applicaties installeren. pakketten en ports ("Applicaties installeren") is uitgebreid met extra informatie over binair package-beheer.
- Het X Window systeem ("Het X Window systeem") is compleet herschreven met de nadruk op het gebruik van moderne bureaubladtechnologiën zoals KDE en GNOME op XFree86™ 4.X.
- Het FreeBSD Opstartproces ("Het FreeBSD Opstartproces") is uitgebreid.

- Opslag ("Opslag") is herschreven uit wat eens twee aparte hoofdstukken waren over "schijven" en "back-ups". We vinden dat de onderwerpen beter begrijpbaar zijn wanneer ze in één hoofdstuk zijn ondergebracht. Er is ook een sectie over RAID (zowel hardware- als softwarematig) toegevoegd.
- Seriële communicatie ("Seriële communicatie") is compleet gereorganiseerd en bijgewerkt voor FreeBSD 4.X/5.X.
- PPP en SLIP ("PPP en SLIP") is aanzienlijk bijgewerkt.
- Veel nieuwe secties zijn toegevoegd aan Geavanceerd netwerken ("Geavanceerd netwerken").
- Elektronische mail ("E-mail") is uitgebreid met meer informatie over het instellen van sendmail.
- Linux® binaire compatibiliteit ("Linux® binaire compatibiliteit") is uitgebreid met informatie over het installeren van Oracle®.
- De volgende nieuwe onderwerpen worden behandeld in de tweede editie:
  - Instellingen en optimalisatie (Instellingen en optimalisatie).
  - Multimedia (Multimedia)

De opbouw van dit boek

Dit boek is opgedeeld in vijf logische secties. De eerste sectie, *Beginnen*, behandelt de installatie en het basisgebruik van FreeBSD. Er wordt verwacht dat lezers deze hoofdstukken volgt, en mogelijk hoofdstukken overslaat met bekende onderwerpen. De tweede sectie, *Algemene Taken*, behandelt veelgebruikte functies van FreeBSD. Deze sectie en alle volgende kunnen in een willekeurige volgorde gelezen worden. Iedere sectie begint met een beknopte samenvatting die beschrijft wat het hoofdstuk inhoudt en wat de lezer al moet weten. Dit is bedoeld om de lezer de kans te geven alleen dat te lezen wat voor hem van belang is. In de derde sectie, *Systeembeheer*, wordt het beheer behandeld. De vierde sectie, *Netwerkcommunicatie*, gaat over netwerken en servers. De vijfde sectie bevat appendices met referentiemateriaal.

***Introductie*, Introductie**

Introduceert FreeBSD aan een nieuwe gebruiker. Het beschrijft de geschiedenis van het FreeBSD project, de doelen en het ontwikkelmodel.

***FreeBSD installeren op FreeBSD 8.X en eerder*, Installatie van FreeBSD 8._X en eerder_**

Begeleidt de gebruiker door het gehele installatieproces van FreeBSD 8.*X* en eerder door middel van sysinstall. Sommige geavanceerde onderwerpen over installeren, zoals installeren via een seriële console, worden ook behandeld.

***FreeBSD 9.X en nieuwer installeren*, Installatie van FreeBSD 9._X en nieuwer_**

Begeleidt een gebruiker door het gehele installatieproces van FreeBSD 9.*X* en nieuwer door middel van bsdinstall.

***UNIX® beginselen*, UNIX® beginselen**

Behandelt de basiscommando's en functionaliteit van het FreeBSD besturingssysteem. Als de lezer bekend is met Linux® of een andere UNIX® variant, kan dit hoofdstuk waarschijnlijk overgeslagen worden.

Applicaties installeren. pakketten en ports, ***Applicaties installeren***

Behandelt de installatie van software van derden, met zowel FreeBSD's innovatieve "Portscollectie" als de standaard binaire packages.

Het X Window systeem, ***Het X Window systeem***

Beschrijft het X Window systeem in het algemeen en het gebruik van X11 op FreeBSD in het bijzonder. Het beschrijft ook standaard bureaubladomgevingen zoals KDE en GNOME.

Bureaubladapplicaties, **Bureaubladapplicaties_**

Levert standaard bureaubladapplicaties in een lijst, zoals webbrowsers en productiviteitspakketten, en beschrijft hoe ze te installeren op FreeBSD.

Multimedia, ***Multimedia***

Laat zien hoe geluid- en video-ondersteuning te installeren voor een systeem. Het beschrijft ook een aantal voorbeeld audio- en video- applicaties.

De FreeBSD-kernel instellen, ***Instellen van de FreeBSD kernel***

Beschrijft waarom misschien een nieuwe kernel ingesteld moet worden en levert gedetailleerde instructies voor het instellen, bouwen en installeren van een eigen kernel.

Afdrukken, ***Afdrukken***

Beschrijft hoe printers beheerd worden onder FreeBSD, met informatie over bannerpagina's, afdruk-accounting en initiële installatie.

Linux® binaire compatibiliteit, ***Linux® binaire compatibiliteit***

Beschrijft de mogelijkheden van FreeBSD voor binaire compatibiliteit met Linux®. Het biedt ook gedetailleerde installatie-instructies voor vele populaire Linux® applicaties zoals Oracle®, SAP® R/3®, en Mathematica®.

_Instellingen en optimalisatie, **Instellingen en optimalisatie _**

Beschrijft de parameters beschikbaar voor systeembeheerders om een FreeBSD te optimaliseren voor de beste prestaties. Het beschrijft ook diverse instellingenbestanden die gebruikt worden in FreeBSD en waar die te vinden zijn.

Het FreeBSD opstartproces, ***Het FreeBSD opstartproces***

Beschrijft de FreeBSD opstartprocedure en legt uit hoe deze aan te passen met instellingen.

Gebruikers- en basisaccountbeheer, ***Gebruikers en basis accountbeheer***

Beschrijft hoe gebruikersaccounts aan te maken en te wijzigen. Het beschrijft ook welke resourcebeperkingen er gezet kunnen worden op gebruikers en andere account-beheerstaken.

Beveiliging, ***Beveiliging***

Beschrijft vele verschillende hulpapplicaties die beschikbaar zijn die helpen om een FreeBSD systeem veilig te houden, met oa: Kerberos, IPsec en OpenSSH.

Jails, ***Jails***

Beschrijft het jail-raamwerk, en de verbeteringen van jails (gevangenissen) ten opzichte van de traditionele ondersteuning voor chroot van FreeBSD.

Verplichte Toegangscontrole (MAC), ***Verplichte Toegangscontrole (MAC)***

Legt uit was Verplichte Toegangscontrole (MAC) is en hoe het gebruikt kan worden om een FreeBSD te beveiligen.

Security Event Auditing, ***Security Event Auditing***

Beschrijft wat FreeBSD Event Auditing is, hoe het geïnstalleerd kan worden, en hoe audit trails geïnspecteerd en gemonitord kunnen worden.

Opslag, ***Opslag***

Beschrijft hoe opslagmedia en bestandssystemen beheerd worden onder FreeBSD. Dit omvat fysieke schijven, RAID arrays, optische en tape media, geheugenschijven en netwerkbestandssystemen.

Overzicht, ***GEOM***

Beschrijft wat het GEOM raamwerk in FreeBSD is en hoe de verschillende ondersteunde RAID-niveau's in te stellen.

Ondersteuning van bestandssystemen, ***Ondersteuning van bestandssystemen***

Gaat de ondersteuning voor vreemde bestandssystemen in FreeBSD na, zoals het Z File System van Sun™.

Virtualisatie, ***Virtualisatie***

Beschrijft wat virtualisatiesystemen bieden, en hoe ze met FreeBSD gebruikt kunnen worden.

Lokalisatie - I18N/L10N gebruiken en instellen, ***Lokalisatie - I18N/L10N gebruiken en instellen***

Beschrijft hoe FreeBSD met andere talen dan Engels te gebruiken is. Behandelt zowel het systeem- als applicatieniveau van localisatie.

FreeBSD updaten en upgraden, ***FreeBSD updaten en upgraden***

Geeft uitleg over de verschillen tussen FreeBSD-STABLE, FreeBSD-CURRENT en FreeBSD uitgaven. Beschrijft welke gebruikers voordeel hebben van het bijhouden van een ontwikkelsysteem en legt dat proces uit. Beschrijft de manier waarop gebruikers hun systeem naar de laatste beveiligingsuitgave kunnen bijwerken.

DTrace, **DTrace_**

Beschrijft hoe het gereedschap DTrace van Sun™ te configureren en gebruiken in FreeBSD. Dynamisch tracen kan helpen bij het lokaliseren van prestatieproblemen, door real-time systeemanalyse uit te voeren.

Seriële communicatie, ***Seriële communicatie***

Legt uit hoe een verbinding te maken met terminals en modems op een FreeBSD systeem voor zowel dial-in als dial-out verbindingen.

PPP en SLIP, ***PPP en SLIP***

Beschrijft hoe PPP, SLIP en PPP over Ethernet te gebruiken om verbinding te maken met remote systemen met FreeBSD.

Elektronische mail, ***E-mail***

Legt verschillende componenten uit van een mailserver en gaat dieper in op simpele instellingen voor de populairste mailserver software: sendmail.

Netwerkdiensten, ***Netwerkdiensten***

Geeft gedetailleerde instructies en voorbeeldinstellingen om een FreeBSD machine als een netwerk bestandssysteem server, DNS server, netwerk informatiesysteem server of tijdserver in te stellen.

Firewalls, ***Firewalls***

Licht de filosofie achter op software gebaseerde firewalls toe en beschrijf in detail hoe de verschillende firewalls die in FreeBSD beschikbaar zijn ingesteld kunnen worden.

Netwerken voor gevorderden, ***Netwerken voor gevorderden***

Beschrijft meerdere netwerk onderwerpen, inclusief het delen van een Internetverbinding met andere computers in een LAN, routeren voor gevorderden, draadloze netwerken, Bluetooth®, ATM, IPv6 en nog veel meer.

FreeBSD verkrijgen, ***FreeBSD verkrijgen***

Geeft verschillende bronnen aan voor het verkrijgen van FreeBSD media op CD-ROM of DVD evenals verschillende sites op het Internet die gebruikers in staat stellen FreeBSD te downloaden en te installeren.

Bibliografie, ***Bibliografie***

Dit boek behandelt veel verschillende onderwerpen die de lezer misschien hongerig maken naar een gedetailleerdere uitleg. De bibliografie bevat verwijzingen naar een aantal uitstekende boeken.

Bronnen op Internet, ***Bronnen op Internet***

Beschrijft de vele forums die beschikbaar zijn voor FreeBSD gebruikers om vragen te stellen, en om deel te nemen aan technische conversaties over FreeBSD.

PGP sleutels, ***PGP sleutels***

Geeft de PGP-vingerafdrukken van verschillende FreeBSD ontwikkelaars.

Overeenkomsten in dit boek

Om consistentie en leesbaarheid te behouden en de leesbaarheid te behouden worden er een aantal overeenkomsten nageleefd in dit boek.

Typografische overeenkomsten

Italic

Een *italic* lettertype wordt gebruikt voor bestandsnamen, URL's, benadrukte tekst, en het eerste gebruik van technische termen.

`Monospace`

Een `monospaced` lettertype wordt gebruikt voor foutmeldingen, commando's, omgevingsvariabelen, namen van ports, hostnamen, gebruikersnamen, groepsnamen, apparaatnamen, variabelen en stukjes code.

Vet

Een vet lettertype wordt gebruikt voor applicaties, commando's en toetsen.

Gebruikersinvoer

Toetsen worden weergegeven in **bold** om op te vallen tussen andere tekst. Toetscombinaties die bedoeld zijn om tegelijkertijd getypt te worden, worden weergeven met + tussen de toetsen zoals

`Ctrl` + `Alt` + `Del`

Betekent dat de gebruiker de volgende toetsen op hetzelfde moment moet indrukken: `Ctrl`, `Alt` en `Del`.

Toetsen die bedoeld zijn om achter elkaar te typen worden gescheiden door komma's, bijvoorbeeld

`Ctrl` + `X`, `Ctrl` + `S`

zou betekenen dat de gebruiker de `Ctrl` en `X` toetsen tegelijk moet indrukken en erna `Ctrl` en `S` tegelijkertijd moet indrukken.

Voorbeelden

Voorbeelden die beginnen met E:\ geven aan dat het een MS-DOS® commando betreft. Tenzij anders vermeld, kunnen deze commando's in een "Command prompt"scherm in een moderne Microsoft® Windows® omgeving worden gebruikt.

```
E:\ tools\fdimage floppies\kern.flp A:
```

Voorbeelden die starten met een # geven aan dat een commando ingegeven moet worden als de superuser in FreeBSD. Er kan aangemeld worden met `root` om het commando in te typen, of er kan na als gewone gebruiker aangemeld te hebben gebruikt gemaakt worden van su(1) om superuser-rechten te verkrijgen.

```
# dd if=kern.flp of=/dev/fd0
```

Voorbeelden die starten met % geven aan dat een commando opgegeven moet worden vanuit een normale gebruikersaccount. Tenzij anders vermeld, wordt de C-shell syntaxis gebruikt voor het instellen van omgevingsvariabelen en andere shellcommando's.

```
% top
```

Dankwoorden

Het boek dat nu voorligt representeert de inspanningen van honderden mensen over de hele wereld. Of ze nu foutjes verbeteren of complete hoofdstukken inleveren, ze hebben allemaal nuttig bijgedragen.

Verschillende bedrijven hebben bijgedragen aan het maken van dit document door de schrijvers te betalen om hier voltijds aan te werken, door te betalen voor de publicatie, etc. In het bijzonder heeft BSDi (Overgenomen door Wind River Systems) leden van het FreeBSD Documentation Project betaald om voltijds te werken aan het verbeteren van dit boek, wat leidde tot de publicatie van de eerste editie in maart 2000 (ISBN 1-57176-241-8). Wind River Systems heeft daarna verschillende schrijvers betaald om een aantal verbeteringen uit te voeren voor de printuitvoer-infrastructuur en om extra hoofdstukken toe te voegen aan de tekst. Dit werk leverde de publicatie van de tweede gedrukte editie in november 2001 (ISBN 1-57176-303-1). In 2003-2004 heeft FreeBSD Mall, Inc een aantal mensen die bijdragen hebben geleverd betaald om het handboek te verbeteren voor een derde gedrukte editie.

Deel I: Beginnen

Dit deel van het FreeBSD handboek is voor gebruikers en beheerders die net beginnen met FreeBSD. Deze hoofdstukken:

- Geven een inleiding in FreeBSD;
- Lichten het installatieproces toe;
- Bespreken de UNIX® basisbegrippen en grondslag;
- Tonen hoe de vele aanvullende applicaties voor FreeBSD geïnstalleerd kunnen worden;
- Introduceren X, het venstersysteem van UNIX® en gaan uitvoerig in op hoe een bureaubladomgeving wordt ingesteld die een gebruiker helpt productiever te zijn.

Er is geprobeerd het aantal vooruitwijzingen tot een minimum te beperken zodat het handboek van begin tot einde gelezen kan worden zonder bladeren.

Hoofdstuk 1. Introductie

1.1. Overzicht

Welkom bij FreeBSD! Dit hoofdstuk beschrijft de verschillende aspecten van het FreeBSD Project: geschiedenis, doelen, ontwikkelmodel en meer.

Na het lezen van dit hoofdstuk weet de lezer:

- Hoe FreeBSD gerelateerd is aan andere besturingssystemen;
- De geschiedenis van het FreeBSD Project;
- De doelen van het FreeBSD Project;
- De fundering van het FreeBSD open-source ontwikkelmodel;
- En natuurlijk: waar de naam "FreeBSD" vandaan komt.

1.2. Welkom bij FreeBSD!

FreeBSD is een op 4.4BSD-Lite gebaseerd besturingssysteem voor Intel (x86 en Itanium®), AMD64 en Sun UltraSPARC® computers. Er zijn ook ports naar andere architecturen in voorbereiding. Er is nog meer informatie over de geschiedenis van FreeBSD of over de huidige uitgave. Als de lezer wil bijdragen aan het project (code, hardware, geld) wordt aangeraden het artikel Bijdragen aan FreeBSD te lezen.

1.2.1. Wat kan FreeBSD?

FreeBSD heeft veel mogelijkheden die het bespreken waard zijn. Hier zijn er enkele op een rij gezet:

- *Preemptive multitasking* zorgt ervoor dat meerdere programma's en gebruikers op dezelfde computer kunnen werken, zonder dat de systeemrespons of stabiliteit beïnvloed wordt.
- Ondersteuning voor *meerdere gebruikers* maakt het mogelijk dat verschillende mensen een FreeBSD systeem tegelijkertijd kunnen gebruiken voor een groot aantal taken. Dit betekent bijvoorbeeld dat randapparaten als printers en tapedrives gedeeld kunnen worden door alle gebruikers van het systeem en dat individuele beperkingen ingesteld kunnen worden voor gebruikers of voor groepen gebruikers, zodat kritieke systeembronnen beschermd kunnen worden tegen onrechtmatig of overmatig gebruik.
- Krachtige mogelijkheden voor *TCP/IP netwerken* met ondersteuning voor industriestandaarden als SCTP, DHCP, NFS, NIS, PPP, SLIP, IPsec en IPv6. Dit betekent dat een FreeBSD-systeem makkelijk kan samenwerken met andere systemen en dat het kan functioneren als bedrijfsserver, waarbij het belangrijke functies als NFS (bestandsdeling over het netwerk), email, webdiensten, FTP, routing en firewall-diensten kan aanbieden.
- *Geheugenbeveiliging* garandeert dat applicaties (of gebruikers) elkaar niet kunnen storen. Een crashende applicatie heeft totaal geen effect op andere applicaties.
- FreeBSD is een *32-bits* besturingssysteem (*64-bits* op de Itanium®, AMD64, en UltraSPARC®) en is van de grond af aan zo ontworpen.

- Het *X Window systeem* (X11R7), een industriële standaard, biedt een grafische gebruikersinterface (GUI) met als enige benodigdheden een VGA-kaart en een beeldscherm.
- Door *binaire compatibiliteit* met veel programma's voor Linux®, SCO, SVR4, BSDI en NetBSD is het mogelijk om deze programma's zonder snelheidsverlies op FreeBSD te draaien.
- Er zijn duizenden applicaties beschikbaar in de FreeBSD *ports* en *pakketten* collectie. Waarom zoeken op het Internet als het allemaal al klaarstaat?
- Duizenden andere en *makkelijk over te zetten* applicaties zijn beschikbaar op het Internet. FreeBSD is broncode-compatibel met de meeste populaire commerciële UNIX® systemen, wat betekent dat veel applicaties nagenoeg geen wijzigingen vereisen om te compileren op FreeBSD.
- Het demand-paged *virtueel geheugen* en de "gecombineerde VM/buffer cache" van FreeBSD zorgen ervoor dat applicaties met grote geheugenbehoeften niets te kort komen, terwijl de systeemrespons niet achteruit gaat.
- *SMP*-ondersteuning voor computers met meerdere processoren.
- Een volledige *C* en *C++* ontwikkelomgeving. Vele andere programmeertalen, te gebruiken voor onderzoek of geavanceerde ontwikkeling, zijn ook beschikbaar in de ports- en pakketcollectie.
- De *broncode* van het hele systeem is beschikbaar, zodat gebruikers de volledige controle over het systeem in handen hebben. Waarom genoegen nemen met alleen het erewoord van de softwarefabrikant, als een compleet open systeem ook tot de mogelijkheden behoort?
- Uitgebreide *online documentatie*.
- *En nog veel meer!*

FreeBSD is gebaseerd op de 4.4BSD-Lite uitgave van de Computer Systems Research Group (CSRG) aan de University of California in Berkeley en borduurt voort op een lange traditie van ontwikkeling van BSD-systemen. Het FreeBSD Project heeft duizenden uren gestoken in het afstellen van het systeem voor maximale prestaties en betrouwbaarheid in realistische en veel voorkomende situaties. Terwijl veel commerciële bedrijven blijven worstelen met het uitbrengen van besturingssystemen met dergelijke mogelijkheden, prestaties en betrouwbaarheid, kan FreeBSD deze *nu* bieden!

De toepassingen voor FreeBSD worden alleen beperkt door eigen fantasie. Van software-ontwikkeling tot fabrieksautomatisering, van voorraadbeheersing tot de azimuth-correctie van een satellietantenne: als het kan met een commercieel UNIX®product, dan kan het ook met FreeBSD! FreeBSD vaart ook wel bij de letterlijk duizenden open-source programma's, vaak van bijzonder hoge kwaliteit, die ontwikkeld zijn in onderzoekscentra, universiteiten over de hele wereld en open-source gemeenschappen, en die beschikbaar zijn voor weinig of geen geld. Ook steeds meer commerciële applicaties vinden hun weg naar FreeBSD.

Omdat ook de broncode van FreeBSD zelf vrij beschikbaar is, kan het systeem aangepast worden voor speciale toepassingen of projecten, op manieren die meestal niet mogelijk zijn met besturingssystemen van vooraanstaande commerciële softwarehuizen. Hier zijn een aantal voorbeelden van toepassingen waar FreeBSD voor gebruikt wordt:

- *Internetdiensten:* de robuuste TCP/IP netwerkarchitectuur die in FreeBSD zit, maakt het een ideaal platform voor uiteenlopende Internetdiensten als:
    - FTP servers;

- World Wide Webservers (standaard of beveiligd [SSL]);
- IPv4 en IPv6 routering
- Firewalls en NAT ("IP-maskering") gateways;
- E-mail servers;
- USENET nieuws of Bulletin Board (BBS) systemen;
- En meer...

  FreeBSD kan eenvoudig geleerd worden op een goedkope standaard-PC, om later verder te groeien naar een professioneel Xeon-systeem met 4 processoren (of meer!) en RAID opslagsystemen als een bedrijf groeit.

- *Onderwijs:* is de lezer informaticastudent of werkzaam in een ander vakgebied dat hier mee te maken heeft? Er is geen betere manier om besturingssystemen, computerarchitecturen en netwerken te bestuderen dan de hands-on open-source ervaring die FreeBSD kan bieden. Gratis beschikbare programma's voor CAD, wiskundige toepassingen en grafisch ontwerp maken FreeBSD ook heel handig voor mensen wiens primaire interesse voor de computer ligt bij het voltooien van *ander* werk!
- *Onderzoek:* omdat de broncode van het volledige systeem beschikbaar is, vormt FreeBSD een uitstekende basis voor het onderzoeken van besturingssystemen of andere takken in de informatica. De open natuur van FreeBSD maakt het ook mogelijk voor groepen mensen over de hele wereld om met elkaar samen te werken, zonder dat men zich zorgen hoeft te maken over speciale licentieovereenkomsten of beperkingen op wat er besproken kan worden in open fora.
- *Netwerken:* nieuwe router nodig? Of een nameserver (DNS)? Een firewall om een intern netwerk te beschermen? FreeBSD kan die ongebruikte 486 of Pentium PC die nog ergens in een hoekje ligt gemakkelijk omtoveren tot een geavanceerde router met uitgebreide pakketfilter mogelijkheden.
- *X Window werkstation:* FreeBSD is een prima keuze als goedkope X terminal oplossing, door gebruik te maken van de gratis beschikbare X11 server. In tegenstelling tot een pure X terminal kan FreeBSD ook applicaties lokaal draaien, wat een verlichting van de centrale server tot gevolg kan hebben. FreeBSD heeft zelfs de mogelijkheid om "schijfloos" op te starten, zodat individuele werkstations nog goedkoper en makkelijker te beheren zijn.
- *Bureaublad:* de beschikbaarheid van geavanceerde bureaubladomgevingen als KDE en GNOME en kantoortoepassingen als tekstverwerkers en spreadsheet-programma's in de ports- en pakketcollectie maken van FreeBSD een uitgebreid desktop-platform. Thuis en op het werk zorgt FreeBSD ervoor dat er snel, efficiënt en veilig gewerkt kan worden!
- *Software Ontwikkeling:* bij het standaard FreeBSD-systeem zit al een volledige verzameling van ontwikkelgereedschappen, inclusief de bekende GNU C/C++ compiler en debugger.

FreeBSD is beschikbaar in zowel broncode als binaire vorm op CD-ROM, DVD en via FTP. In FreeBSD verkrijgen staat meer informatie over het verkrijgen van FreeBSD.

1.2.2. Wie gebruiken FreeBSD?

FreeBSD wordt gebruikt als platform voor apparaten en producten van vele van 's werelds grootste IT-bedrijven, waaronder:

- Apple
- Cisco
- Juniper
- NetApp

FreeBSD wordt ook gebruikt om sommige van de grootste sites op het Internet te draaien, waaronder:

- Yahoo!
- Yandex
- Apache
- Rambler
- Sina
- Pair Networks
- Sony Japan
- Netcraft
- NetEase
- Weathernews
- TELEHOUSE America
- Experts Exchange

en nog veel meer sites.

1.3. Over het FreeBSD Project

Deze paragraaf geeft wat meer achtergrondinformatie over het project, inclusief een korte geschiedenis, projectdoelen, en het ontwikkelmodel van het project.

1.3.1. Een korte geschiedenis van FreeBSD

Het FreeBSD Project zag het licht in het begin van 1993, gedeeltelijk als een voortzetting van de "Unofficial 386BSD Patchkit" door de 3 laatste coördinatoren van de patchkit: Nate Williams, Rod Grimes en ikzelf.

Het oorspronkelijke doel was om een zogenaamde 'snapshot'-uitgave te maken van 386BSD, om zo een aantal problemen op te lossen die niet op te lossen waren met het patchkit-mechanisme dat eerder gebruikt was. Sommigen kunnen zich misschien nog herinneren dat de werktitel van het project in het begin nog "386BSD 0.5" of "386BSD Interim" was, refererend aan het oorspronkelijke

doel.

386BSD was het besturingssysteem van Bill Jolitz en had tot op dat moment geleden onder het feit dat er al bijna een jaar niet naar omgekeken was. Terwijl de patchkit steeds groter en onhandiger werd, was een groep mensen het er over eens dat er iets moest gebeuren en beslisten om Bill te assisteren bij het maken van een tussentijdse "cleanup"-snapshot. Deze plannen kwamen echter tot een plotseling einde toen Bill Jolitz besliste om zijn toestemming voor het project in te trekken, zonder dat er een alternatief werd geboden.

Het duurde niet lang om te beslissen dat het doel nog steeds belangrijk was, zelfs zonder de ondersteuning van Bill, dus werd de naam "FreeBSD" aangenomen, naar een idee van David Greenman. De oorspronkelijke doelen werden opgesteld na het raadplegen van de gebruikers van het systeem. Toen het erop begon te lijken dat dit project misschien wel snel realiteit kon worden, werd contact opgenomen met Walnut Creek CD-ROM vanuit het oogpunt om de distributiekanalen van FreeBSD te verbeteren voor diegenen die geen toegang hadden tot Internet. Walnut Creek CD-ROM ondersteunde niet alleen het idee om FreeBSD op CD-ROM te distribueren, maar bood het project ook een systeem en een snelle Internetverbinding om mee te werken. Zonder Walnut Creek CD-ROM's bijna onbeperkte vertrouwen in wat op dat moment nog een compleet onbekend project was, is het onwaarschijnlijk dat FreeBSD zo ver gekomen zou zijn, en zo snel, als het vandaag de dag is.

De eerste CD-ROM (en algemene op het net beschikbare) distributie was FreeBSD 1.0, uitgebracht in december 1993. Deze versie was gebaseerd op de 4.3BSD-Lite ("Net/2") tape van U.C. Berkeley, met veel toevoegingen van 386BSD en de Free Software Foundation. Het werd een redelijk succes voor een eerste aanbod, en werd opgevolgd door de zeer succesvolle FreeBSD 1.1 uitgave in mei 1994.

Rond deze tijd vormde zich nogal onverwacht een stormachtige lucht aan de horizon toen Novell en U.C. Berkeley hun langlopende rechtszaak over de legale status van de Berkeley Net/2 tape oplosten met een schikking. Een voorwaarde van deze schikking was dat U.C. Berkeley toegaf dat grote delen van Net/2 "beladen" code was en het eigendom van Novell, die deze code op haar beurt overgenomen had van AT&T enige tijd hiervoor. Wat Berkeley hiervoor terugkreeg was Novell's "zegen" over de 4.4BSD-Lite uitgave; wanneer deze uitkwam zou Novell verklaren dat geen van de code hierin eigendom van Novell was, en bestaande Net/2 gebruikers zou sterk aanbevolen worden om over te stappen naar deze nieuwe versie. Dit gold ook voor FreeBSD en het project werd de tijd gegeven tot juli 1994 om te stoppen met het distribueren van het eigen op Net/2-gebaseerde product. De schikking liet wel toe dat nog een laatste uitgave werd uitgebracht voor de deadline en dat was FreeBSD 1.1.5.1.

FreeBSD nam toen de enorme taak op zich om zichzelf letterlijk opnieuw uit te vinden, met als basis een volledig nieuwe en nogal incomplete verzameling van delen van 4.4BSD-Lite. De "Lite" uitgaven werden zo genoemd omdat Berkeley's CSRG grote delen code die nodig waren om een werkend systeem te construeren had weggelaten (om allerlei legale redenen) en omdat de Intel port van 4.4 grotendeels incompleet was. Het kostte het project tot november 1994 om deze overstap te maken. Op dat moment werd FreeBSD 2.0 op het net en op CD-ROM (aan het einde van december) uitgebracht. Ondanks het feit dat deze uitgave nog wat ruige kanten had, werd het een groot succes en werd het gevolgd door de robuustere en makkelijker te installeren FreeBSD 2.0.5 in juni 1995.

In augustus 1996 is FreeBSD 2.1.5 uitgebracht en deze bleek populair genoeg bij Internet service

providers (ISP's) en andere commerciële gebruikers van FreeBSD om nog een uitgave van de 2.1-STABLE tak te rechtvaardigen. Dit was FreeBSD 2.1.7.1, uitgebracht in februari 1997. Deze uitgave markeerde het einde van de hoofdstroomontwikkeling op 2.1-STABLE; alleen beveilingsupdates en andere kritieke bugfixes werden nog op deze tak uitgevoerd (RELENG_2_1_0).

FreeBSD 2.2 werd afgesplitst van de ontwikkelingstak ("-CURRENT") in november 1996 als RELENG_2_2 en de eerste volledige uitgave (2.2.1) werd uitgebracht in april 1997. Andere uitgaven van de 2.2 tak werden uitgebracht in de zomer en herfst van '97. De laatste (2.2.8) verscheen in november 1998. De eerste officiële 3.0 uitgave verscheen in oktober 1998 en was het begin van het einde voor de 2.2 tak.

Er was opnieuw een afsplitsing op 20 januari 1999, wat leidde tot de 4.0-CURRENT en 3.X-STABLE takken. Vanuit 3.X-STABLE werd versie 3.1 uitgebracht op 15 februari 1999, 3.2 op 15 mei 1999, 3.3 op 16 september 1999, 3.4 op 20 december 1999 en 3.5 op 24 juni 2000. De laatste werd enkele dagen later gevolgd door een puntuitgave-update naar 3.5.1, om enkele net-ontdekte beveiligingsfouten in Kerberos te corrigeren. Dit was de laatste uitgave van de 3.X tak.

Een nieuwe tak werd gemaakt op 13 maart 2000, de 4.X-STABLE tak. Er zijn verschillende uitgaven van deze tak gemaakt: 4.0-RELEASE werd geïntroduceerd in maart 2000, en de laatste 4.11-RELEASE verscheen in januari 2005.

De langverwachte 5.0-RELEASE werd aangekondigd op 19 januari 2003. Dit resultaat van bijna drie jaar werk zette FreeBSD stevig neer op de weg naar geavanceerde multiprocessor- en threading-ondersteuning en introduceerde nieuwe FreeBSD ports voor de UltraSPARC® en `ia64` architecturen. Deze uitgave werd gevolgd door 5.1 in juni 2003. De laatste 5.X uitgave uit de -CURRENT-tak was 5.2.1-RELEASE uit februari 2004.

De RELENG_5 tak is gemaakt in augustus 2004 en werd gevolgd door 5.3-RELEASE, die het begin van de 5-STABLE tak markeert. De meest recente 5.5-RELEASE is uitgekomen in mei 2006. Er staan geen nieuwe versies gepland voor de RELENG_5 tak.

De RELENG_6 tak is gemaakt in juli 2005, de eerste uitgave van de 6.X tak werd vrijgegeven in november 2005. De meest recente 6.4-RELEASE kwam uit in november 2008. Er zullen geen verdere uitgaven komen van de RELENG_6 tak. Deze tak is de laatste tak waarin ondersteuning zit voor de Alpha architectuur.

De RELENG_7 tak is gemaakt in oktober 2007. De eerste uitgave van deze tak is 7.0-RELEASE, welke is uitgekomen in februari 2008. De meest recente 7.4-RELEASE kwam uit in februari 2011. Er zullen geen andere uitgaven van de RELENG_7 tak uitkomen.

De RELENG_8 tak is gemaakt in augustus 2009. De eerste uitgave van de 8.X tak is 8.0-RELEASE, vrijgegeven in november 2009. De meest recente uitgave 11.2-RELEASE kwam uit in June 28, 2018. Er zullen nog andere uitgaven van de RELENG_8 tak uitkomen.

De RELENG_9 tak is gemaakt in september 2011. De eerste uitgave van deze tak was 12.0-RELEASE, vrijgegeven in December 11, 2018. Er zullen nog andere uitgaven van de RELENG_9 tak uitkomen.

Op dit moment vinden lange-termijn ontwikkelprojecten plaats in de 10.X-CURRENT tak, en snapshot uitgaven van 10.X op CD-ROM (en natuurlijk op het Net) worden continu beschikbaar gemaakt op de snapshot server.

1.3.2. Doelen van het FreeBSD Project

Het doel van het FreeBSD Project is om software aan te bieden die gebruikt kan worden voor iedere mogelijke toepassing, zonder beperkingen. Vele ontwikkelaars hebben een belangrijke investering in de code (en het project) zitten en vinden het niet erg om af en toe een financiële compensatie te ontvangen, maar dat is zeker geen voorwaarde. De ontwikkelaars van FreeBSD geloven dat de eerste en belangrijkste "missie" het aanbieden van code is, aan iedereen die het wil hebben, voor wat voor doel dan ook, zodat de code zo breed mogelijk gebruikt kan worden tot voordeel van zoveel mogelijk mensen. Dit is een van de meest fundamentele doelen van Vrije Software dat FreeBSD enthousiast ondersteunt.

Sommige code in FreeBSD valt onder de GNU General Public License (GPL) of Library General Public License (LGPL). Deze code heeft iets meer beperkingen, maar in ieder geval aan de kant waarbij vrije toegang tot de code geforceerd wordt, in plaats van het gebruikelijke tegenovergestelde hiervan. Door de toegevoegde moeilijkheden die kunnen voortkomen uit het commerciële gebruik van GPL software geeft het FreeBSD Project echter de voorkeur aan het meer vrije BSD copyright, wanneer er een redelijk alternatief voor handen is.

1.3.3. Het FreeBSD ontwikkelmodel

De ontwikkeling van FreeBSD is een erg open en flexibel proces en wordt gevormd door de bijdragen van letterlijk honderden mensen over de hele wereld, zoals te zien is in de lijst van medewerkers. De infrastructuur die wordt gebruikt voor de ontwikkeling van FreeBSD zorgt ervoor dat deze honderden ontwikkelaars kunnen samenwerken over het Internet. Het FreeBSD Project is continu op zoek naar nieuwe ontwikkelaars en ideeën. Om bij te dragen aan de ontwikkeling van FreeBSD is een mail naar FreeBSD technische discussie mailinglijst voldoende. De FreeBSD aankondigingen mailinglijst is beschikbaar om mededelingen te doen aan andere FreeBSD-gebruikers over grote veranderingen.

Een aantal dingen over het FreeBSD Project en haar ontwikkelingsproces zijn handig om te weten, of een bijdrage nu onafhankelijk of in samenwerking met anderen komt:

Het CVS-archief

Gedurende een aantal jaren werd de centrale broncode voor FreeBSD bijgehouden door CVS (Concurrent Versions System), een vrij verkrijgbaar pakket voor het onderhouden van broncode dat bij FreeBSD zit. In juni 2008 is het Project SVN (Subversion) gaan gebruiken. Deze overgang werd nodig geacht omdat de technische beperkingen die door CVS worden opgelegd duidelijk werden wegens de snelle uitbreiding van de broncode en de hoeveelheid geschiedenis die reeds is opgeslagen. De reservoirs van het Documentatieproject en de Portscollectie zijn ook omgezet van CVS naar SVN, respectievelijk in mei 2012 en juli 2012.

Hoewel de reservoirs voor `src/` en `ports/` nu SVN gebruiken, blijven cliëntgereedschappen zoals csup die van de oudere CVS-infrastructuur afhankelijk zijn normaal werken - veranderingen in het SVN-archief worden voor dit doel teruggeplaatst naar CVS. In tegenstelling tot `src/` en `ports/` wordt het SVN-reservoir voor de documentatie niet teruggeplaatst naar CVS.

Het primaire CVS archief staat op een systeem in Santa Clara, Californië, in de VS, waar het wordt gesynchroniseerd met verschillende "mirrors" over de hele wereld. De boomstructuur van SVN , waarin de broncode voor -CURRENT en -STABLE is te vinden, kan ook makkelijk met

die op een eigen systeem gesynchroniseerd worden. Synchroniseren van broncode bevat meer informatie over dit onderwerp.

Committers

De zogenaamde *committers* zijn alle mensen die *schrijf*-rechten hebben in het Subversion-archief van FreeBSD. Deze mensen mogen veranderingen maken aan de broncode van FreeBSD (de term "committer" is afkomstig van het `commit` commando van versiebeheersystemen, wat gebruikt wordt om veranderingen door te voeren in het archief). De beste manier om eigen bijdragen te laten keuren door een van de committers is door gebruik te maken van send-pr(1). Als het erop lijkt dat een bijdrage ergens in het systeem blijft hangen, dan is het ook mogelijk om mail te sturen naar de FreeBSD committer's mailinglijst.

Het FreeBSD Core Team

Het *FreeBSD core team* zou het equivalent zijn van een raad van bestuur als het FreeBSD Project een bedrijf zou zijn. De primaire taak van het core team is ervoor zorg te dragen dat het project, in zijn geheel, in goede vorm verkeert en de goede richting opgaat. Toegewijde en verantwoordelijke ontwikkelaars uitnodigen om deel te worden van de committers is één van de taken van het core team, net als het rekruteren van nieuwe leden van het core team. Het huidige core team is gekozen door de committers uit een groep van kandidaten (ook allen committers) in juli 2012. Elke twee jaar worden verkiezingen gehouden.

Sommige leden van het core team hebben een bijzondere verantwoordelijkheid, wat wil zeggen dat zij er speciaal op toezien dat een bepaald deel van het systeem werkt zoals het hoort. In de lijst van medewerkers staat een complete lijst van ontwikkelaars en hun verantwoordelijkheden.



De meeste leden van het core team zijn vrijwilligers. "Toewijding" betekent dus niet "gegarandeerde ondersteuning". De "raad van bestuur"-analogie hierboven klopt niet helemaal en het is misschien beter om te zeggen dat dit de mensen zijn die hun leven opgaven voor FreeBSD, tegen beter weten in!

Externe Bijdragen

De grootste groep ontwikkelaars zijn de gebruikers zelf, die FreeBSD continu voorzien van constructief commentaar en oplossingen voor fouten. De handigste manier om contact te houden met het niet-gecentraliseerde deel van de ontwikkeling van FreeBSD is een abonnement nemen op de FreeBSD technische discussie mailinglijst, waar allerlei bijdragen, patches en nieuwe ideeën worden bediscussieerd. In Bronnen op Internet is meer informatie te vinden over de verschillende FreeBSD mailinglijsten.

De lijst van medewerkers is lang en groeit iedere dag, dus wat let de lezer om zelf een bijdrage te doen aan FreeBSD?

Programmeren is niet de enige manier om een bijdrage te leveren aan het project. Een meer volledige lijst van dingen die gedaan moeten worden staat op de FreeBSD website.

Samengevat is het FreeBSD ontwikkelmodel georganiseerd als een onsamenhangende verzameling van concentrische cirkels. Het gecentraliseerde model is ontworpen voor het gemak van de *gebruikers* van FreeBSD, die op deze manier makkelijk de wijzigingen in het project kunnen volgen.

Niet om potentiële medewerkers buiten de deur te houden! Het is wenselijk om een stabiel besturingssysteem te maken, met een grote verzameling samenhangende applicaties. Dit model heeft zijn waarde op dat gebied bewezen.

Om bij te dragen en samen FreeBSD verder te ontwikkelen, is het enige wat het FreeBSD Project vraagt dat te doen met dezelfde toewijding als de huidige ontwikkelaars: succes gegarandeerd!

1.3.4. Huidige FreeBSD uitgave

FreeBSD is een open source, op 4.4BSD-Lite gebaseerd besturingssysteem voor Intel (x86 en Itanium®), AMD64, n Sun UltraSPARC® computers. Het is grotendeels gebaseerd op software van de Computer Systems Research Group (CSRG) van de University of California in Berkeley (U.C. Berkeley), met verbeteringen overgenomen van NetBSD, OpenBSD, 386BSD en de Free Software Foundation.

Sinds het uitbrengen van FreeBSD 2.0 tegen het einde van 1994, zijn de prestaties, mogelijkheden en stabiliteit van FreeBSD dramatisch verbeterd. FreeBSD heeft namelijk de beschikking over een compleet nieuw subsysteem voor virtueel geheugen, dat niet alleen de prestaties ten goede komt, maar er ook voor zorgt dat het systeem minder geheugen gebruikt dan ooit tevoren. Andere belangrijke verbeteringen zijn de ondersteuning van veel nieuwe hardware, een compleet nieuw systeem voor de ondersteuning van machines met meerdere processoren (SMP) en een nieuwe bibliotheek voor de ondersteuning van multithreading in applicaties.

Behalve de basisdistributie van het besturingssysteem, biedt FreeBSD ook een enorme softwarecollectie met duizenden veelgebruikte programma's, de zogenaamde ports. Op het moment van schrijven zijn er al meer dan 37000 ports! In de ports zitten alle mogelijke klassen van software die te bedenken zijn, van HTTP-servers tot spellen, van kantoorapplicaties tot multimedia en alles wat er tussenin zit. De complete Portscollectie beslaat zo'n 3 GB aan schijfruimte. Meer informatie over de ports en over de pakketten is te vinden in Applicaties installeren. pakketten en ports.

Alle recente versies van FreeBSD bieden een optie aan in de installer (ofwel sysinstall(8) ofwel bsdinstall(8)) om aanvullende documentatie te installeren onder /usr/local/shared/doc/freebsd tijdens de eerste installatie van het systeem. De documentatie kan ook op elk later tijdstip worden geïnstalleerd door pakketten te gebruiken zoals beschreven in Documentatiepakketten gebruiken. De lokaal geïnstalleerde documentatie kan in een browser bekeken worden door de volgende URLs te gebruiken:

Het FreeBSD handboek

/usr/local/shared/doc/freebsd/handbook/index.html

De FreeBSD FAQ

/usr/local/shared/doc/freebsd/faq/index.html

De nieuwste versies van deze documenten zijn altijd te vinden op http://www.FreeBSD.org/.

Hoofdstuk 2. FreeBSD 9.X en nieuwer installeren

2.1. Overzicht

Wordt nog vertaald.

Hoofdstuk 3. UNIX® beginselen

3.1. Overzicht

Het volgende hoofdstuk behandelt de basiscommando's en functionaliteit van het FreeBSD besturingssysteem. Veel van dit materiaal is relevant voor elk UNIX® achtig besturingssysteem. Als de lezer reeds bekend is met het materiaal, hoeft dit hoofdstuk niet gelezen te worden. Lezer die nog niet eerder met FreeBSD te maken hebben gehad wordt aangeraden door te lezen.

Na het lezen van dit hoofdstuk weet de lezer:

- Hoe "virtuele consoles" in FreeBSD gebruikt kunnen worden;
- Hoe UNIX® bestandspermissies werken en hoe bestandsvlaggen in FreeBSD werken;
- Hoe het standaard FreeBSD bestandssysteem eruit ziet;
- Hoe een FreeBSD harde schijf is ingedeeld;
- Hoe bestandssystemen gekoppeld en ontkoppeld worden;
- Wat processen, daemons en signalen zijn;
- Wat een shell is en hoe de standaard omgevingsvariabelen veranderd kunnen worden;
- Hoe elementaire tekstverwerkers te gebruiken;
- Wat apparaten en apparaatkoppelpunten zijn;
- Welk binair formaat FreeBSD gebruikt;
- Hoe handleidingen te gebruiken meer informatie.

3.2. Virtuele consoles en terminals

FreeBSD kan op diverse manieren gebruikt worden. Één van deze manieren is het typen van commando's in een teksterminal. Veel van de flexibiliteit en kracht van een UNIX® besturingssysteem is gemakkelijk beschikbaar als je FreeBSD op deze manier gebruikt. Dit onderdeel beschrijft wat "terminals " en "consoles" zijn en hoe je deze kan gebruiken in FreeBSD.

3.2.1. De console

Als FreeBSD niet is ingesteld om automatisch een grafische omgeving te starten tijdens het opstarten, geeft het systeem een login prompt als het gestart is. Dit gebeurt direct nadat de startscripts klaar zijn. Er wordt iets als het volgende getoond:

```
Additional ABI support:.
Local package initialization:.
Additional TCP options:.

Fri Sep 20 13:01:06 EEST 2002

FreeBSD/i386 (pc3.example.org) (ttyv0)
```

```
login:
```

De meldingen op het scherm kunnen wellicht iets anders zijn op een systeem, maar het zal iets soortgelijks zijn. De laatste twee regels zijn de regels waar het nu over gaat. De voorlaatste regel toont:

```
FreeBSD/i386 (pc3.example.org) (ttyv0)
```

Deze regel bevat enkele informatie over het systeem dat net gestart is: dit is een "FreeBSD" console, draaiend op een Intel of soortgelijke processor op de x86 architectuur. De naam van de machine (elke UNIX® machine heeft een naam) is `pc3.example.org` en dit is de console van het systeem, de ttyv0 terminal.

De laatste regel is altijd:

```
login:
```

Dit is het deel waar een "gebruikersnaam" ingevuld moet worden om aan te melden op FreeBSD. Het volgende deel beschrijft hoe dat werkt.

3.2.2. Aanmelden op FreeBSD

FreeBSD is een multi-user en multi-processing systeem. Dit is de formele beschrijving die meestal gegeven wordt aan een systeem dat gebruikt wordt door meerdere personen die gelijktijdig verschillende programma's draaien op één enkele machine.

Elk multi-user systeem heeft een manier nodig om een "gebruiker" van alle andere gebruikers te kunnen onderscheiden. In FreeBSD (en alle andere UNIX® achtige besturingssystemen), wordt dit bereikt door te eisen dat elke gebruiker moet "aanmelden" op het systeem voordat hij/zij programma's kan draaien. Elke gebruiker heeft een unieke naam (de "gebruikersnaam") en een persoonlijke, geheime sleutel (het "wachtwoord"). FreeBSD vraagt om deze twee gegevens voordat het een gebruiker toegestaat om programma's te draaien.

Direct nadat FreeBSD is opgestart en de opstartscripts afgerond zijn, wordt een prompt getoond dat vraagt om een geldige aanmeldnaam op te geven.

```
login:
```

In dit voorbeeld wordt aangenomen de gebruikersnaam `john` is. Als na deze prompt `john` wordt getype en op `Enter` wordt gedrukt, verschijnt hierna een prompt om het " wachtwoord" in te voeren:

```
login: john
Password:
```

Nu kan `john's wachtwoord ingevoerd worden en op `Enter` gedrukt worden. Het wachtwoord wordt *niet getoond*! Daarover hoeft geen zorg te bestaan. Het is voldoende om te zeggen dat dit om veiligheidsredenen gedaan wordt.

Als het juiste wachtwoord is ingegeven, is er aangemeld bij op FreeBSD en in het systeem klaar om alle beschikbare commando's uit te voeren.

Na het aanmelden is de MOTD of het bericht van de dag zichtbaar, gevolgd door een commandoprompt (een `#`, `$` of een `%` karakter). Dit geeft aan dat er succesvol is aangemeld op FreeBSD.

3.2.3. Meerdere consoles

UNIX® programma's draaien in één console is prima, maar FreeBSD kan veel programma's tegelijk draaien. Om maar één console te hebben waar commando's ingetypt kunnen worden zou zonde zijn van een besturingssysteem als FreeBSD waar meerdere programma's tegelijkertijd op kunnen draaien. Hier kunnen "virtuele consoles" van pas komen.

FreeBSD kan ingesteld worden om verschillende virtuele consoles te tonen. Met toetscombinaties kan van de ene console naar de gewisseld worden. Elke console heeft zijn eigen uitvoerkanaal, en FreeBSD zorgt ervoor dat alle toetsenbordinvoer en monitoruitvoer goed wordt gezet als er van de ene console naar de volgende wordt gewisseld.

In FreeBSD kunnen speciale toetscombinaties gebruikt worden om te wisselen naar een ander virtueel console. In FreeBSD kan `Alt` + `F1`, `Alt` + `F2` tot en met `Alt` + `F8` gebruikt worden om te wisselen naar een ander virtueel console.

Als wordt gewisseld van de ene naar de andere console zorgt FreeBSD dat de uitvoer bewaard blijft. Het resultaat is een "illusie" van het hebben van meerdere schermen en toetsenborden die gebruikt kunnen worden om commando's in te voeren om FreeBSD te laten draaien. De programma's die in de ene virtuele console draaien, stoppen niet als de console niet zichtbaar is. Ze blijven doordraaien als naar een andere virtuele console wordt gewisseld.

3.2.4. Het bestand /etc/ttys

De standaardinstelling van FreeBSD start op met acht virtuele consoles. Dit is echter geen vaste waarde en een installatie kan eenvoudig aangepast worden, zodat het systeem gestart wordt met meer of minder virtuele consoles. De hoeveelheid en instellingen van de virtuele consoles worden ingesteld in /etc/ttys.

/etc/ttys kan gebruikt worden om virtuele consoles in te stellen. Elke niet-commentaar regel in dit bestand (regels die niet beginnen met een `#` karakter) bevat instellingen voor een terminal of virtuele console. De standaardversie van dit bestand die meegeleverd wordt met FreeBSD stelt negen virtuele consoles in en activeert er acht. Dit zijn de regels die beginnen met `ttyv`:

```
# naam  getty                            type    status          commentaar
#
ttyv0   "/usr/libexec/getty Pc"         cons25  on  secure
# Virtual terminals
```

```
ttyv1   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv2   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv3   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv4   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv5   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv6   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv7   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv8   "/usr/X11R6/bin/xdm -nodaemon"  xterm   off secure
```

Een uitgebreide beschrijving van elke kolom in dit bestand en alle mogelijke opties voor virtuele consoles staan in de ttys(5) hulppagina gebruiken.

3.2.5. Single-user console

In Single-user modus staat een gedetailleerde beschrijving van de "single-user modus". Het is belanrijk te melden dat er in single-user modus maar één console is. Er zijn geen virtuele consoles beschikbaar. De instellingen van de single-user modus console staan ook in /etc/ttys. De regel begint met `console`:

```
# name  getty                           type    status          commentaar
#
# Als een console gemarkeerd is als "insecure", zal het init script om het root-
wachtwoord
# vragen wanneer het in single-user mode komt.
console none                            unknown off secure
```



Zoals het commentaar boven de `console` regel aangeeft, kan in deze regel het woord `secure` gewijzigd worden in `insecure`. In dat geval vraagt FreeBSD bij het opstarten in single-user modus nog steeds om een `root`-wachtwoord.

Pas op als dit wordt veranderd in `_insecure`\`_. Als het wachtwoord van de gebruiker \`root zoek is, wordt het opstarten in single-user modus lastig. Het is nog steeds mogelijk, maar het kan vrij moeilijk zijn voor iemand die FreeBSD niet zo goed kent met betrekking tot het opstarten en de programma's die daarbij gebruikt worden.

3.2.6. Het wijzigen van de console video mode

De FreeBSD standaard video mode kan worden gewijzigd in 1024x768, 1280x1024, of een van de vele andere formaten die ondersteund worden door de grafische kaart en monitor. Laad de module `VESA` om gebruik te maken van de verschillende video modes:

```
# kldload vesa
```

Kijk daarna welke video modes er ondersteund worden door de hardware door gebruik te maken van de vidcontrol(1) applicatie. Om een overzicht te krijgen van de ondersteunde video modes

moet het volgende ingevoerd worden:

```
# vidcontrol -i mode
```

Het resultaat van dit commando is een lijst van video modes welke ondersteund worden door de hardware. Hierna kan de nieuwe video mode gekozen worden door dit aan te geven aan vidcontrol(1):

```
# vidcontrol MODE_279
```

Als de nieuwe video mode acceptabel is, kan dit permanent ingesteld worden door het volgende in /etc/rc.conf te zetten:

```
allscreens_flags="MODE_279"
```

3.3. Rechten

FreeBSD, direct afgeleid van BSD UNIX®, is gebaseerd op verschillende belangrijke UNIX® concepten. Het meest bekende is dat FreeBSD een multi-user systeem is. Het systeem kan meerdere gebruikers behandelen die tegelijkertijd totaal verschillende dingen doen. Het systeem is verantwoordelijk voor het netjes delen en beheren voor aanvragen voor hardware, randapparatuur, geheugen en cpu tijd tussen elke gebruiker.

Omdat het systeem in staat is om meerdere gebruikers te ondersteunen, heeft alles wat door het systeem beheerd wordt een set van rechten die aangeeft wie mag lezen, schrijven en de bron mag uitvoeren. Deze rechten zijn opgeslagen in drie octetten, die weer in drie stukjes onderverdeeld zijn: één voor de eigenaar van het bestand, één voor de groep waar het bestand toe behoort en één voor de overigen. De numerieke weergave werkt als volgt:

Waarde	Recht	Maprecht
0	Niet lezen, niet schrijven, niet uitvoeren	`---`
1	Niet lezen, niet schrijven, uitvoeren	`--x`
2	Niet lezen, schrijven, niet uitvoeren	`-w-`
3	Niet lezen, schrijven, uitvoeren	`-wx`
4	Lezen, niet schrijven, niet uitvoeren	`r--`
5	Lezen, niet schrijven, uitvoeren	`r-x`
6	Lezen, schrijven, niet uitvoeren	`rw-`

Waarde	Recht	Maprecht
7	Lezen, schrijven, uitvoeren	`rwx`

De `-l` optie kan gebruikt worden met ls(1) om een lange lijst met de inhoud van een map te zien die een kolom heeft met informatie over bestandsrechten voor de eigenaar, groep en de rest. `ls -l` in een willekeurige map kan het volgende laten zien:

```
% ls -l
total 530
-rw-r--r--  1 root  wheel     512 Sep  5 12:31 myfile
-rw-r--r--  1 root  wheel     512 Sep  5 12:31 otherfile
-rw-r--r--  1 root  wheel    7680 Sep  5 12:31 email.txt
...
```

Zo ziet de eerste kolom van `ls -l` eruit:

```
-rw-r--r--
```

Het eerste (meest linkse) karakter geeft aan of dit een reguliere bestand is, een map, een speciaal karakter component(!), een socket of een andere pseudo-file component(!). In dit geval betekent de `-` dat het een regulier bestand is. De volgende drie karakters, `rw-` in dit voorbeeld, geven de rechten voor de eigenaar van het bestand. De drie karakters `r--` erna geven de rechten van voor de groep van het bestand. De overige drie karakters `r--` tonen de rechten voor alle overige gebruikers. Een streepje betekent dat de rechten uitgeschakeld zijn. In het geval van dit bestand zijn de rechten zo ingesteld dat de eigenaar kan lezen en schrijven naar het bestand, de groep het bestand kan lezen, en alle overige gebruikers kunnen ook het bestand lezen. Volgens de tabel hierboven worden de rechten `644`, waar de cijfers de drie stukjes van de rechten aangeven.

Dit is allemaal leuk en aardig, maar hoe controleert het systeem dan rechten voor apparaten? FreeBSD behandelt de meeste hardware apparaten als bestanden die door programma's kunnen worden geopend en gelezen, en waar data naar toe kan worden geschreven, net zoals elk ander bestand. Deze speciale apparaat bestanden worden bewaard in de map /dev.

Mappen worden ook behandeld als bestanden. Ze hebben lees, schrijf en uitvoerbare rechten. De uitvoerbare vlag voor een map heeft een klein verschil qua betekenis dan die voor gewone bestanden. Als een map als uitvoerbaar gemarkeerd is, betekent het dat erin gekeken mag worden. Het is dus mogelijk om te wisselen naar de map met `cd` (wissel van map). Dit betekent ook dat in de map bestanden benaderd kunnen worden waarvan de naam bekend is. Dit is natuurlijk afhankelijk van de rechten op het bestand zelf.

In het bijzonder, om een lijst van de map te kunnen maken, moet een gebruiker leesrechten op de map hebben. Om een bestand te verwijderen zijn de naam van het bestand en schrijf *en* uitvoerrechten op de map nodig waarin het bestand zich bevindt.

Er zijn meer rechtenvlaggen, maar die worden slechts gebruikt in speciale gevallen, zoals bij setuid binaries en sticky mappen. Meer informatie over bestandsrechten en hoe die aangepast kunnen worden staat in chmod(1).

3.3.1. Symbolische rechten

Symbolische rechten, soms ook wel symbolische expressies, gebruiken karakters in plaats van octale getallen om rechten aan bestanden en mappen te geven. Symbolische expressies gebruiken de volgende opbouw: (wie) (actie) (permissies), waar de volgende waardes beschikbaar zijn:

Optie	Letter	Vertegenwoordigt
(wie)	u	Gebruiker
(wie)	g	Groepseigenaar
(wie)	o	Overigen
(wie)	a	Iedereen ("wereld")
(actie)	+	Rechten toevoegen
(actie)	-	Rechten verwijderen
(actie)	=	Stel deze rechten in
(recht)	r	Lezen
(recht)	w	Schrijven
(recht)	x	Uitvoeren
(recht)	t	Sticky bit
(recht)	s	Verander UID of GID

Deze waardes worden gebruikt met chmod(1), net zoals eerder, alleen nu met letters. Het volgende commando kan gebruikt worden om de overige gebruikers toegang tot *BESTAND* te ontzeggen:

```
% chmod go= BESTAND
```

Er kan een door komma's gescheiden lijst geleverd worden als meer dan één wijziging aan een bestand moet worden uitgevoerd. Het volgende commando past de rechten voor de groep en de "wereld" aan door de schrijfrechten te ontnemen om daarna iedereen uitvoerrechten te geven:

```
% chmod go-w,a+x BESTAND
```

3.3.2. FreeBSD bestandsvlaggen

Naast de bestandsrechten die hiervoor zijn besproken, biedt FreeBSD ondersteuning voor "bestandsvlaggen." Deze vlaggen bieden een aanvullend beveiligingsniveau en controle over bestanden, maar niet over mappen.

Bestandsvlaggen voegen een extra niveau van controle over bestanden, waardoor verzekerd kan worden dat in sommige gevallen zelfs `root` een bestand niet kan verwijderen of wijzigen.

Bestandsvlaggen worden gewijzigd met het hulpprogramma chflags(1), dat een eenvoudige interface heeft. Om bijvoorbeeld de systeemvlag niet verwijderdbaar in te stellen op het bestand

file1:

```
# chflags sunlink file1
```

Om de vlag niet verwijderbaar weer te verwijderen kan het voorgaande commando met "no" voor `sunlink` worden uitgevoerd:

```
# chflags nosunlink file1
```

Om de vlaggen op een bestand te bekijken, kan het ls(1) commando met de vlaggen `-lo` gebruikt worden:

```
# ls -lo file1
```

De uitvoer hoort er ongeveer als volgt uit te zien:

```
-rw-r--r--  1 trhodes  trhodes  sunlnk 0 Mar  1 05:54 file1
```

Een aantal vlaggen kan alleen ingesteld of verwijderd worden door de gebruiker `root`. In andere gevallen kan de eigenaar van een bestand vlaggen instellen. Meer informatie voor beheerders staat in chflags(1) en chflags(2).

3.3.3. De `setuid`-, `setgid`-, en `klevende` toestemmingen

Buiten de toestemmingen die reeds besproken zijn, zijn er nog drie specifieke instellingen waarvan alle beheerders kennis dienen te hebben. Dit zijn de `setuid`-, `setgid`-, en `sticky` toestemmingen.

Deze instellingen zijn belangrijk voor sommige UNIX®-bewerkingen omdat ze functionaliteit bieden die normaliter niet aan normale gebruikers wordt gegeven. Om ze te begrijpen, dient ook het verschil tussen de echte gebruikers-ID en de effectieve gebruikers-ID opgemerkt te worden.

De echte gebruikers-ID is de UID die het proces start of bezit. De effectieve UID is de gebruikers-ID waaronder het proces draait. Bijvoorbeeld, het gereedschap passwd(1) draait met de echte gebruikers-ID van de gebruiker die het wachtwoord verandert; echter, om de database met wachtwoorden te manipuleren, draait het met de effectieve ID van de gebruiker `root`. Dit is wat normale gebruikers in staat stelt om hun wachtwoorden te veranderen zonder een fout `Permission Denied` te zien.



De mount(8)-optie `nosuid` zorgt ervoor dat deze binairen zwijgend falen. Dit houdt in dat ze niet worden uitgevoerd zonder ooit de gebruiker op de hoogte te stellen. Deze optie is ook niet geheel betrouwbaar aangezien een `nosuid`-wrapper dit volgens de handleidingpagina mount(8) kan omzeilen.

De setuid-toestemming kan aangezet worden door het cijfer vier (4) voor een toestemmingenverzameling te plaatsen zoals te zien is in het volgende voorbeeld:

```
# chmod 4755 suidvoorbeeld.sh
```

De toestemmingen op het bestand suidvoorbeeld.sh dienen er nu als volgt uit te zien:

```
-rwsr-xr-x   1 trhodes  trhodes    63 Aug 29 06:36 suidvoorbeeld.sh
```

Het zou in dit voorbeeld te zien moeten zijn dat een `s` nu deel is van de toestemmingenverzameling bestemd voor de bestandseigenaar, en de uitvoerbare bit vervangt. Dit staat gereedschappen toe die verhoogde toestemmingen nodig hebben, zoals `passwd`.

Open twee terminals om dit in real-time te zien. Start op het ene het proces `passwd` als een normale gebruiker. Controleer de procestabel terwijl het op een nieuw wachtwoord wacht en kijk naar de gebruikersinformatie van het commando `passwd`.

In terminal A:

```
Changing local password for trhodes
Old Password:
```

In terminal B:

```
# ps aux | grep passwd
```

```
trhodes  5232  0.0  0.2  3420  1608   0  R+    2:10AM   0:00.00 grep passwd
root     5211  0.0  0.2  3620  1724   2  I+    2:09AM   0:00.01
```

Zoals boven vermeld, wordt `passwd` door een normale gebruiker gedraaid, maar gebruikt het de effectieve UID van `root`.

De `setgid`-toestemming voert dezelfde functie uit als de `setuid`-toestemming; behalve dat het de groepsinstellingen verandert. Wanneer een applicatie of gereedschap met deze instelling wordt gedraaid, krijgt het de toestemmingen gebaseerd op de groep die het bestand bezit, niet op de gebruiker die het proces startte.

Om de `setgid`-toestemming op een bestand aan te zetten, dient een voorlopende twee (2) aan het commando `chmod` gegeven te worden zoals in het volgende voorbeeld:

```
# chmod 2755 sgidvoorbeeld.sh
```

De nieuwe instelling kan zoals hierboven bekeken worden, merk op dat de `s` nu in het veld bestemd voor de instellingen van de groepstoestemmingen staat:

```
-rwxr-sr-x   1 trhodes  trhodes    44 Aug 31 01:49 sgidvoorbeeld.sh
```



In deze voorbeelden zal het shellscript niet met een andere EUID of effectief gebruikers-ID draaien, zelfs al is het shellscript uitvoerbaar. Dit is omdat shellscripts geen toegang hebben tot de setuid(2)-systeemaanroepen.

De eerste twee speciale toestemmingsbits die we besproken hebben (de toestemmingsbits `setuid` en `setgid`) kunnen de systeemveiligheid verlagen, door verhoogde toestemmingen toe te staan. Er is een derde bit voor speciale toestemmingen die de veiligheid van een systeem kan verhogen: de `klevende bit`.

De `klevende bit`, wanneer deze op een map is ingesteld, staat alleen het verwijderen van bestanden toe door de eigenaar van die bestanden. Deze toestemmingenverzameling is nuttig om het verwijderen van bestanden in publieke mappen, zoals /tmp, door gebruikers die het bestand niet bezitten te voorkomen. Zet een één (1) voor de toestemming om deze toestemming te gebruiken. Bijvoorbeeld:

```
# chmod 1777 /tmp
```

Het effect kan nu met het commando `ls` bekeken worden:

```
# ls -al / | grep tmp
```

```
drwxrwxrwt  10 root  wheel         512 Aug 31 01:49 tmp
```

De toestemming `klevende bit` is te onderscheiden met de `t` aan het einde van de verzameling.

3.4. Mappenstructuur

De FreeBSD mappenstructuur is erg belangrijk om het systeem goed te leren kennen. Het belangrijkste concept om greep op te krijgen is die van de rootmap, "/". Deze map is de eerste die gekoppeld wordt tijdens het opstarten en bevat het basissysteem dat nodig is om het besturingssysteem gereed te maken voor multi-user taken. De rootmap bevat ook koppelpunten voor elk ander bestandssysteem dat misschien gekoppeld wordt.

Een koppelpunt is een map waar extra bestandssystemen aan het een bestandssysteem gekoppeld kunnen worden (meestal het root bestandssysteem). Dit wordt beschreven in Organisatie van schijven. Standaard koppelpunten zijn /usr, /var, /tmp, /mnt en /cdrom. Naar deze mappen wordt meestal verwezen in /etc/fstab, een tabel met bestandssystemen en koppelpunten ter referentie voor het systeem. De meeste bestandssystemen in /etc/fstab worden automatisch gekoppeld tijdens het opstarten door het script rc(8), behalve als de optie `noauto` gedefinieerd is. Details staan beschreven in Het bestand fstab.

Een complete beschrijving over het bestandssysteem staat in hier(7). Hier wordt volstaan met een

overzicht van de voorkomende mappen.

Map	Omschrijving
/	Rootmap van het bestandssysteem.
/bin/	Gebruikersapplicaties, belangrijk voor zowel single user als multi-user omgevingen.
/boot/	Programma's en instellingenbestanden die gebruikt worden tijdens het opstarten van het besturingssysteem.
/boot/defaults/	Bestanden met standaardinstellingen voor opstarten;, zie loader.conf(5).
/dev/	Apparaatnodes;, zie intro(4).
/etc/	Bestanden met systeeminstellingen en scripts.
/etc/defaults/	Bestanden met standaard systeeminstellingen;, zie rc(8).
/etc/mail/	Instellingenbestanden voor mail transport programma's zoals sendmail(8).
/etc/namedb/	Instellingenbestanden voor `named`, zie named(8).
/etc/periodic/	Scripts die dagelijks, wekelijks en maandelijks via cron(8) worden uitgevoerd, zie periodic(8).
/etc/ppp/	Instellingenbestanden voor `ppp`, zie ppp(8).
/mnt/	Lege map, veel gebruikt door systeembeheerders als tijdelijk koppelpunt voor opslagruimtes.
/proc/	Process bestandssysteem;, zie procfs(5) en mount_procfs(8).
/rescue/	Statisch gelinkte programma's voor noodherstel, zie rescue(8).
/root/	Thuismap van de gebruiker `root`.
/sbin/	Systeemprogramma's en administratieprogramma's belangrijk voor zowel single-user en multi-user omgevingen.
/tmp/	Tijdelijke bestanden. De inhoud van /tmp blijft meestal NIET bewaard na een herstart. Er wordt vaak een geheugengebaseerd bestandssysteem gekoppeld op /tmp. Dit kan geautomatiseerd worden met de tmpmfs-gerelateerde variabelen van rc.conf(5) (of met een regel in /etc/fstab). Zie mdmfs(8).
/usr/	Hier bevindt zich het leeuwendeel van alle hulpprogramma's en gewone programma's.

Map	Omschrijving
/usr/bin/	Standaard programma's, programmeertools.
/usr/include/	Standaard C invoegbestanden.
/usr/lib/	Functiebibliotheken.
/usr/libdata/	Diverse databestanden voor hulpprogramma's.
/usr/libexec/	Systeemdaemons en systeemhulpprogramma's (uitgevoerd door andere programma's).
/usr/local/	Lokale programma's, bibliotheken, etc. Wordt ook gebruikt als standaard locatie voor de FreeBSD ports. Binnen /usr/local, wordt de algemene layout bepaald door hier(7), dat ook voor /usr wordt gebruikt. Uitzonderingen is de map man, die direct onder /usr/local ligt in plaats van onder /usr/local/share, en de documentatie voor ports is te vinden in share/doc/port.
/usr/obj/	Architectuur afhankelijke doelstructuur voor resultaten van de bouw van /usr/src.
/usr/ports/	De FreeBSD Portscollectie (optioneel).
/usr/sbin/	Systeemdaemons en systeemhulpprogramma's (uitgevoerd door gebruikers).
/usr/shared/	Architectuur onafhankelijke bestanden.
/usr/src/	BSD en/of lokale broncodebestanden.
/usr/X11R6/	Uitvoerbare bestanden en bibliotheken, etc, voor de X11R6 distributie (optioneel).
/var/	Multifunctionele logboek-, tijdelijke, transparante en spool bestanden.
/var/log/	Diverse logboekbestanden van het systeem.
/var/mail/	Postbusbestanden van gebruikers.
/var/spool/	Diverse printer- en mailsysteemspoolingmappen.
/var/tmp/	Tijdelijke bestanden die bewaard worden bij een herstart van het systeem.
/var/yp/	NIS maps.

3.5. Organisatie van schijven

De kleinste vorm van organisatie die FreeBSD gebruikt om bestanden te vinden is de bestandsnaam. Bestandsnamen zijn hoofdlettergevoelig, wat betekent dat readme.txt en README.TXT twee verschillende bestanden zijn. FreeBSD gebruikt de extensie niet (.txt) van een

bestand om te bepalen of het bestand een programma, een document of een vorm van data is.

Bestanden worden bewaard in mappen. Een map kan leeg zijn of honderden bestanden bevatten. Een map kan ook andere mappen bevatten, wat het mogelijk maakt om een hiërarchie van mappen te maken. Dit maakt het veel makkelijker om data te organiseren.

Bestanden en mappen worden aangegeven door het bestand of de map aan te geven, gevolgd door een voorwaardse slash, `/`, gevolgd door andere mapnamen die nodig zijn. Als map foo de map bar bevat, die op zijn beurt het bestand readme.txt bevat, dan wordt de volledige naam of *pad* naar het bestand foo/bar/readme.txt.

Mappen en bestanden worden bewaard op een bestandssysteem. Elk bestandssysteem bevat precies één map op het hoogste niveau die *de rootmap* van het bestandssysteem heet. Deze rootmap kan op zijn beurt andere mappen bevatten.

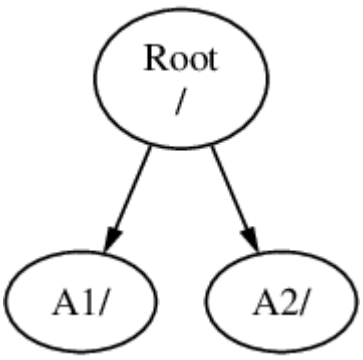
Tot zover is dit waarschijnlijk hetzelfde als voor elk ander besturingssysteem. Er zijn een paar verschillen. MS-DOS® gebruikt bijvoorbeeld een `\` om bestanden en mappen te scheiden, terwijl Mac OS® gebruik maakt van `:`.

FreeBSD gebruikt geen schijfletters, of andere schijfnamen in het pad. FreeBSD gebruikt geen c:/foo/bar/readme.txt.

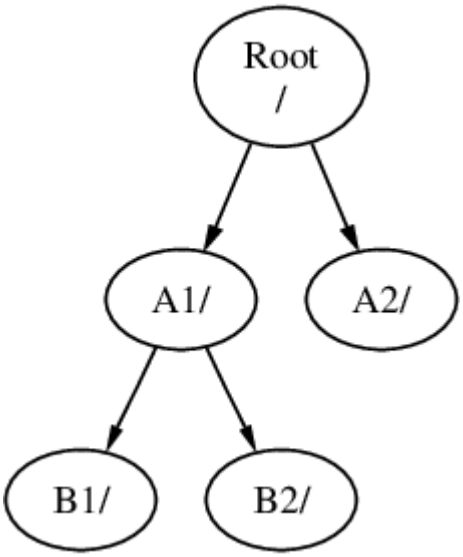
Eén bestandssysteem wordt aangewezen als *root* bestandssysteem, waar naar wordt verwezen met `/`. Elk ander bestandssysteem wordt daarna *gekoppeld* onder het root bestandssysteem. Hoeveel schijven er ook aan een FreeBSD systeem hangen, het lijkt alsof elke map zich op dezelfde schijf bevindt.

Stel er zijn drie bestandssystemen met de namen `A`,`B` en `C`. Elk bestandssysteem heeft één root map die twee andere mappen bevat, `A1` en `A2` (zo ook voor de andere twee: `B1`, `B2`, `C1` en `C2`).

`A` wordt het root besturingsysteem. Met `ls`, dat de inhoud van de map kan tonen, zijn de twee mappen `A1` en `A2` te zien. De mappenstructuur ziet er als volgend uit:

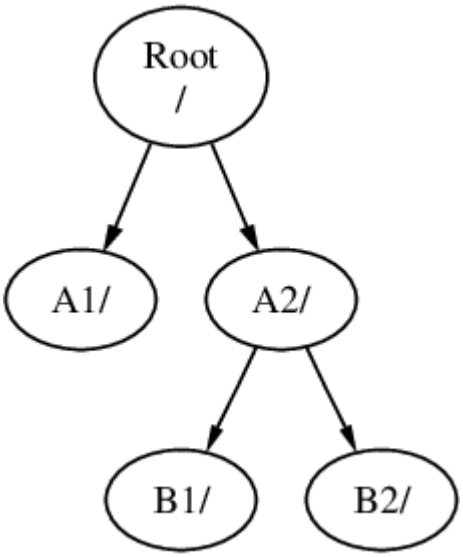


Een bestandssysteem moet gekoppeld worden in een map op een ander bestandssysteem. Als nu bestandssysteem `B` wordt gekoppeld onder de map `A1` vervangt `B``A1` en zien de koppelingen in `B` er als volgt uit:



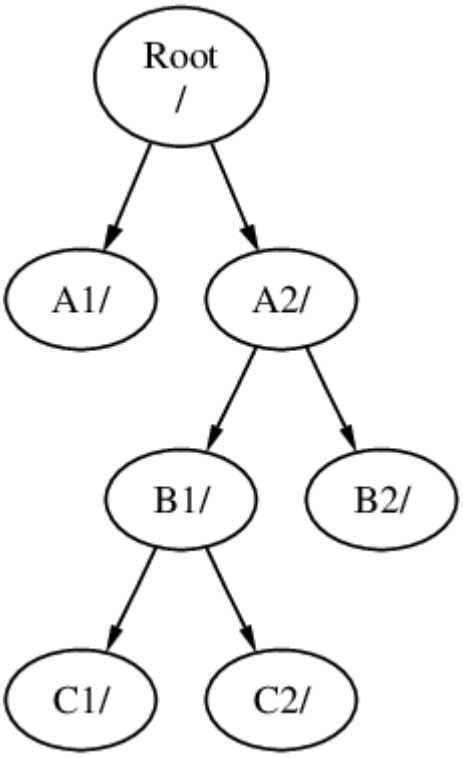
Elk bestand dat in de mappen `B1` en `B2` aanwezig is, kan benaderd worden met het pad /A1/B1 of /A1/B2. Elk bestand dat in /A1 stond is tijdelijk verborgen en komt tevoorschijn als `B`is _ontkoppeld_van A.

Als `B` gekoppeld is onder `A2` ziet de diagram er als volgt uit:

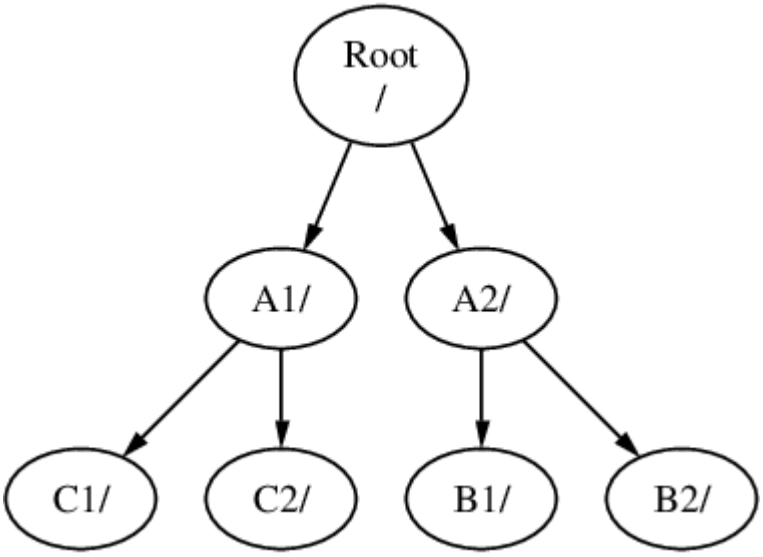


en de paden zouden dan respectievelijk /A2/B1 en /A2/B2 zijn.

Bestandssystemen kunnen op elkaar worden gekoppeld. Doorgaand op het vorige voorbeeld kan het bestandssysteem `C` gekoppeld worden bovenop de map `B1` in het bestandssysteem `B`. Dit resulteert in:



Of `C` kan direct onder het bestandssysteem `A` gekoppeld worden, onder de map `A1`:



Hoewel het niet gelijk is, lijkt het op het gebruik van `join` in MS-DOS®.

Beginnende gebruikers hoeven zich hier gewoonlijk niet mee bezig te houden. Normaal gesproken worden bestandssystemen gemaakt als FreeBSD wordt geïnstalleerd en er wordt besloten waar ze gekoppeld worden. Meestal worden ze ook niet gewijzigd tot er een nieuwe schijf aan een systeem wordt toegevoegd.

Het is mogelijk om één groot root bestandssysteem te hebben en geen andere. Deze benadering heeft voordelen en nadelen.

Voordelen van meerdere bestandssystemen

- Verschillende bestandssystemen kunnen verschillende *mount opties* hebben. Met een goede

voorbereiding kan het root bestandssysteem bijvoorbeeld als alleen-lezen gekoppeld worden, waardoor het onmogelijk wordt om per ongeluk kritische bestanden te verwijderen of te bewerken. Het scheiden van andere bestandssystemen die beschrijfbaar zijn door gebruikers, zoals /home van andere bestandssystemen stelt de beheerder in staat om ze *nosuid* te koppelen. Deze optie voorkomt dat *suid*/*guid* bits op uitvoerbare bestanden effectief gebruikt kunnen worden, waardoor de beveiliging mogelijk beter wordt.

- FreeBSD optimaliseert automatisch de layout van bestanden op een bestandssysteem, afhankelijk van hoe het bestandssysteem wordt gebruikt. Een bestandsysteem dat veel bestanden bevat waar regelmatig naar geschreven wordt, wordt anders geoptimaliseerd dan een bestandssysteem dat minder maar grotere bestanden bevat. Door het gebruik van één groot bestandssysteem werkt deze optimalisatie niet.
- FreeBSD's bestandssystemen zijn erg robuust als er bijvoorbeeld een stroomstoring is, hoewel een stroomstoring op een kritiek moment nog steeds kan leiden tot schade aan de structuur van het bestandssysteem. Door het verdelen van data over meerdere bestandssystemen, is de kans groter dat het systeem nog opstart, wat terugzetten van een back-up makkelijker maakt als dat nodig is.

Voordeel van één bestandssysteem

- Bestandssystemen hebben een vaste grootte. Als bij de installatie van FreeBSD een bestandssysteem wordt gemaakt, is het later mogelijk dat de partitie groter gemaakt moet worden. Dit is niet zo makkelijk zonder een back-up, het opnieuw maken van het bestandssysteem met gewijzigde grootte en het terugzetten van de geback-upte gegevens.



FreeBSD heeft growfs(8) waarmee de grootte van het bestandssysteem is aan te passen terwijl het draait.

Bestandssystemen worden opgeslagen in partities. Dit betekent niet hetzelfde als de algemene betekenis van de term partitie (bijvoorbeeld, MS-DOS® partitie), vanwege FreeBSD's UNIX® achtergrond. Elke partitie wordt geïdentificeerd door een letter van a tot en met h. Elke partitie kan slechts één bestandssysteem hebben, wat betekent dat bestandssystem vaak omschreven worden aan de hand van hun koppelpunt in de bestandssysteem hiërarchie of de letter van de partitie waar ze in opgeslagen zijn.

FreeBSD gebruikt ook schijfruimte voor *wisselbestanden*. Wisselbestanden geven FreeBSD *virtueel geheugen*. Dit geeft de computer de mogelijkheid om net te doen alsof er veel meer geheugen in de machine aanwezig is dan werkelijk het geval is. Als FreeBSD geen geheugen meer heeft, verplaatst het data die op dat moment niet gebruikt wordt naar de wisselbestanden en plaatst het terug als het wel nodig is (en zet iets anders in ruil daarvoor terug).

Aan sommige partities zijn bepaalde conventies gekoppeld.

Partitie	Conventie
a	Bevat meestal het root bestandssysteem
b	Bevat meestal de swapruimte

Partitie	Conventie
c	Heeft meestal dezelfde grootte als de hele harde schijf. Dit geeft hulpprogramma's de mogelijkheid om op een complete schijf te werken (voor bijvoorbeeld een bad block scanner) om te werken op de c partitie. Meest wordt hierop dan ook geen bestandssysteem gecreeërd.
d	Partitie d had vroeger een speciale betekenis, maar die is verdwenen. d zou nu kunnen werken als een normale partitie.

Elke partitie die een bestandssysteem bevat is opgeslagen in wat FreeBSD noemt een *slice*. Slice is FreeBSD's term voor wat meeste mensen partities noemen. Dit komt wederom door FreeBSD's UNIX® achtergrond. Slices zijn genummerd van 1 tot en met 4.

Slicenummers volgen de apparaatnamen, voorafgegaan door een s die begint bij 1. Dus "da0_s1_" is de eerste slice op de eerste SCSI drive. Er kunnen maximaal vier fysieke slices op een schijf staan, maar er kunnen logische slices in fysieke slices van het correcte type staan. Deze uitgebreide slices zijn genummerd vanaf 5. Dus "ad0_s5_" is de eerste uitgebreide slice op de eerste IDE schijf. Deze apparaten worden gebruikt door bestandssystemen waarvan verwacht wordt dat ze een slice in beslag nemen.

Slices, "gevaarlijk toegewijde" (dangerously dedicated) fysieke drivers en andere drives bevatten *partities*, die worden weergegeven door letters vanaf a tot h. Deze letter wordt achter de apparaatnaam geplakt. Dus "da0_a_" is de a partitie op de eerste da drive, die "gevaarlijk toegewijd" is. "ad1s3_e_" is de vijfde partitie op de derde slice van de tweede IDE schijf.

Elke schijf op het systeem wordt geïdentificeerd. Een schijfnaam start met een code die het type aangeeft en dan een nummer dat aangeeft welke schijf het is. In tegenstelling tot bij slices, start het nummeren van schijven bij 0. Standaardcodes staan beschreven in Schijf apparaatcodes.

Bij een referentie aan een partitie verwacht FreeBSD ook dat de slice en schijf refereert naar die partitie en als naar een slice wordt verwezen moet ook de schijfnaam genoemd worden. Dit kan door de schijfnaam, s, het slice nummer en de partitieletter aan te geven. Voorbeelden staan in Voorbeeld schijf-, slice- en partitienamen.

In Conceptmodel van een schijf staat een conceptmodel van een schijflayout die een en ander verduidelijkt.

Voordat FreeBSD geïnstalleerd kan worden moeten eerst de schijfslices gemaakt worden en daarna moeten de partities op de slices voor FreeBSD gemaakt worden. Daarna wordt op elke partitie het bestandssysteem (of wisselbestand) gemaakt en als laatste wordt besloten waar het filesysteem gekoppeld wordt.

Tabel 1. Schijf apparaatcodes

Code	Betekenis
ad	ATAPI (IDE) schijf
da	SCSI directe toegang schijf
acd	ATAPI (IDE) CDROM
cd	SCSI CDROM
fd	Floppydisk

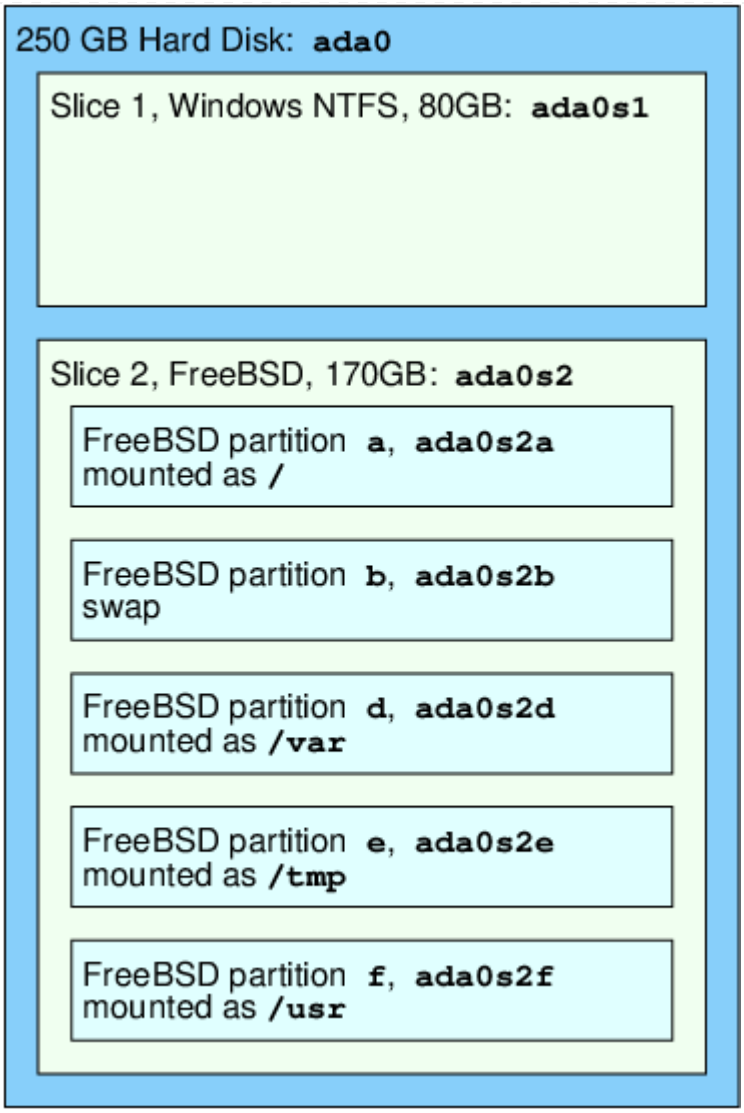
Voorbeeld 1. Voorbeeld schijf-, slice- en partitienamen

Name	Betekenis
`ad0s1a`	De eerste partitie (`a`) op de eerste slice (`s1`) op de eerste IDE schijf (`ad0`).
`da1s2e`	De vijfde partitie (`e`) op de tweede slice (`s1`) op de tweede SCSI schijf (`da1`).

Voorbeeld 2. Conceptmodel van een schijf

Het onderstaande diagram geeft aan hoe FreeBSD de eerste IDE schijf in het systeem ziet. Stel dat de schijf 4 GB groot is en dat deze twee 2 GB slices (MS-DOS® partities) bevat. De eerste slice bevat een MS-DOS® schijf, C: en de tweede slice bevat een FreeBSD installatie. Deze FreeBSD installatie heeft drie partities en een partitie met een wisselbestand.

De drie partities hebben elk een bestandssysteem. Partitie `a` wordt gebruikt voor het root bestandssysteem, `e` voor de map /var en `f` voor de map /usr.



3.6. Het koppelen en ontkoppelen van bestandssystemen

Het bestandssysteem wordt het best weergegeven als een boom, met de stam als /. /dev, /usr en de andere map in root zijn takken die weer hun eigen takken kunnen hebben, zoals /usr/local, etc.

Er zijn verschillende redenen om sommige van deze mappen op aparte bestandssystemen te plaatsen. /var bevat de mappen log/, spool/ en verschillende types tijdelijke bestanden en kan volraken. Het laten vollopen van het root bestandssysteem is geen goed idee, dus het splitsen van /var van / is vaak de favoriet.

Een andere vaak voorkomende reden om bepaalde mapbomen op aparte bestandssystemen te plaatsen, is om ze op verschillende fysieke schrijven te zetten of gescheiden virtuele schijven zoals gemounte Netwerk bestandssystemen of cd-rom drives.

3.6.1. Het bestand fstab

Tijdens het opstartproces, worden bestandssystemen die vermeld staan in /etc/fstab automatisch gekoppeld (tenzij ze vermeld staan met `noauto`).

/etc/fstab bevat een lijst van regels die aan het volgende formaat voldoen:

```
apparaat        /koppelpunt fstype     opties      dumpfreq     passno
```

`apparaat`
: Een apparaatnaam (die moet bestaan) zoals uitgelegd in Apparaatnamen.

`koppelpunt`
: Een map (die moet bestaan) waarop het bestandssysteem gekoppeld moet worden.

`fstype`
: Het bestandssysteem type dat aan mount(8) gegeven wordt. Het standaard FreeBSD bestandssysteem is `ufs`.

`opties`
: Dit is of `rw` voor lezen en schrijven bestandssytemen, of `ro` voor alleen lezen, gevolgd door elke andere optie die mogelijk nodig is. Een standaard optie is `noauto` voor bestandssystemen die niet automatisch gekoppeld worden tijdens het opstarten. Andere opties staan in mount(8).

`dumpfreq`
: Dit wordt gebruikt door dump(8) om te bepalen welke bestandssystemen gedumpt moeten worden. Als het veld niet is ingevuld, wordt aangenomen dat er een nul staat.

`passno`
: Dit bepaalt in welke volgorde bestandssystemen gecontroleerd moeten worden. Bestandssystemen die overgeslagen moeten worden moeten hun `passno` waarde op nul hebben staan. Voor het root bestandssysteem (dat voor alle andere gecontroleerd moet worden) moet `passno` op één staan en `passno` waarden voor andere bestandssystemen moeten een waarde hebben groter dan één. Als bestandssysteem dezelfde `passno` waarde hebben probeert fsck(8) deze bestandssystemen tegelijkertijd te controleren.

In fstab(5) staat meer informatie over de opmaak van /etc/fstab en de mogelijke opties.

3.6.2. Het commando `mount`

mount(8) wordt gebruikt om bestandsystemen te koppelen.

De meest eenvoudige vorm is:

```
# mount apparaat koppelpunt
```

Alle opties voor het commando staat in mount(8), maar de meest voorkomende zijn:

Mountopties

`-a`
: Mount alle bestandssystemen die in /etc/fstab staan, behalve die gemarkeerd staan als "noauto",

uitgesloten zijn door de optie `-t` of die al gekoppeld zijn.

`-d`

Doe alles behalve het echt aanroepen van de systeemopdracht `mount`. Deze optie is handig in samen met de optie `-v` om te bepalen wat mount(8) eigenlijk probeert te doen.

`-f`

Forceert het koppelen van een niet schoon bestandssysteem (gevaarlijk) of forceert het innemen van schrijftoegang als de koppelstatus van een bestandssysteem wijzigt van lezen en schrijven naar alleen lezen.

`-r`

Mount het bestandssysteem alleen lezen. Dit is identiek aan de optie `ro` voor de optie `-o`.

`-t` ***fstype***

Mount het opgegeven bestandssysteem als het opgegeven type bestandssysteem of koppelt alleen bestandssystemen van het aangegeven type als ook de optie `-a` is opgegeven.

"ufs" is het standaard bestandssysteem.

`-u`

Werk koppel opties van het bestandssysteem bij.

`-v`

Geef uitgebreide informatie (verbose).

`-w`

Mount het bestandssysteem lezen en schrijven.

De optie `-o` accepteert een door komma's gescheiden lijst van opties, waaronder de volgende:

noexec

Sta geen uitvoerbare bestanden toe op dit bestandssysteem. Ook dit is een nuttige veiligheidsoptie.

nosuid

Interpreteer geen setuid of setgid opties op het bestandssysteem. Ook dit is een nuttige veiligheidsoptie.

3.6.3. Het commando `umount`

umount(8) heeft een koppelpunt, een apparaatnaam, `-a` of `-A` als parameter.

Alle vormen kunnen de optie `-f` hebben om een bestandsysteem te forceren te ontkoppelen en de optie `-v` voor uitgebreide informatie. De optie `-f` is meestal geen goed idee. Forceren dat een bestandssysteem ontkoppeld wordt kan de computer laten crashen of data op het bestandssysteem beschadigen.

De opties `-a` en `-A` worden gebruikt om alle bestandssystemen te unmounten, mogelijk nader

gespecificeerd door de optie `-t` met daarachter op welke typen bestandssystemen het betrekking heeft. Voor de optie `-A` geldt dat deze niet probeert het root bestandssysteem te ontkoppelen.

3.7. Processen

FreeBSD is een multi-tasking besturingssysteem. Dit betekent dat het lijkt alsof er meer dan één proces tegelijkertijd draait. Elk programma dat draait wordt een *proces* genoemd. Elk commando dat wordt uitgevoerd start op zijn minst één nieuw proces en er zijn systeemprocessen die continu draaien om het systeem functioneel te houden.

Elk proces wordt geïdentificeerd door een nummer dat *process ID* of *PID* heet, en net zoals bij bestanden heeft elk proces één eigenaar en groep. De eigenaars- en groepsinformatie wordt gebruikt om te bepalen welke bestanden en apparaten het proces mag openen, waarbij gebruik wordt gemaakt van de bestandsrechten die eerder zijn behandeld. Veel processen hebben ook een ouderproces (parent process). Een ouderproces is een proces dat het nieuwe proces heeft gestart. Als commando's in een shell worden ingevoerd, start de shell een proces en elk commando dat draait is ook een proces. De uitzondering hierop is het speciale proces init(8). `init` is altijd het eerste proces, dus het PID is altijd 1. `init` wordt automatisch gestart door de kernel als FreeBSD opstart.

Twee commando's die erg handig zijn om te zien welke processen er draaien zijn ps(1) en top(1). `ps` wordt gebruikt om een statische lijst op te vragen van de processen die op het moment van uitvoeren draaien en kan hun PID, geheugengebruik, de startende commandoregel, enzovoort, tonen. `top` geeft alle draaiende processen weer en werkt de status elke paar seconden bij zodat interactief wordt weergegeven wat een computer aan het doen is.

Standaard laat `ps` alleen zien welke commando's draaien waarvan de gebruiker die het uitvoert de eigenaar is:

```
% ps
  PID  TT  STAT      TIME COMMAND
  298  p0  Ss     0:01.10 tcsh
 7078  p0  S      2:40.88 xemacs mdoc.xsl (xemacs-21.1.14)
37393  p0  I      0:03.11 xemacs freebsd.dsl (xemacs-21.1.14)
48630  p0  S      2:50.89 /usr/local/lib/netscape-linux/navigator-linux-4.77.bi
48730  p0  IW     0:00.00 (dns helper) (navigator-linux-)
72210  p0  R+     0:00.00 ps
  390  p1  Is     0:01.14 tcsh
 7059  p2  Is+    1:36.18 /usr/local/bin/mutt -y
 6688  p3  IWs    0:00.00 tcsh
10735  p4  IWs    0:00.00 tcsh
20256  p5  IWs    0:00.00 tcsh
  262  v0  IWs    0:00.00 -tcsh (tcsh)
  270  v0  IW+    0:00.00 /bin/sh /usr/X11R6/bin/startx -- -bpp 16
  280  v0  IW+    0:00.00 xinit /home/nik/.xinitrc -- -bpp 16
  284  v0  IW     0:00.00 /bin/sh /home/nik/.xinitrc
  285  v0  S      0:38.45 /usr/X11R6/bin/sawfish
```

In het bovenstaande voorbeeld is de uitvoer van ps(1) georganiseerd in een aantal kolommen. `PID` is

het proces ID. PIDs worden toegekend vanaf 1 en lopen op tot 99999. Als ze allemaal zijn gebruikt, worden ze hergebruikt. (een PID wordt niet hergebruikt als deze reeds in gebruik is). De `TT` kolom toont de tty vanwaar het programma draait en wordt nu buiten beschouwing gelaten. `STAT` toont de huidige staat van het programma en ook deze kolom wordt buiten beschouwing gelaten. `TIME` is de hoeveelheid tijd die het programma gedraaid heeft op de CPU. Dit is meestal niet de verstreken tijd vanaf het moment dat het programma is gestart. Veel programma's wachten omdat er alleen gebruik wordt gemaakt van de CPU als er iets voor het programma te doen is. Als laatste is `COMMAND` de commandoregel die gebruikt is om het programma te starten.

ps(1) ondersteunt een aantal opties die de informatie wijzigen die wordt weergegeven. Één van de meest nuttige combinaties is `auxww`. De optie `a` toont informatie over alle draaiende processen, niet alleen die van de gebruiker die is aangemeld. De optie `u` toont de gebruikersnaam van de proceseigenaar, evenals geheugengebruik. De optie `x` toont informatie over daemonprocessen en met de optie `ww` laat ps(1) de volledige commandoregel zien voor elk proces, in plaats van een mogelijk afgekorte regel omdat die te lang is om op het scherm te passsen.

De uitvoer van top(1) is hetzelfde:

```
% top
last pid: 72257;  load averages:  0.13,  0.09,  0.03    up 0+13:38:33  22:39:10
47 processes:  1 running, 46 sleeping
CPU states: 12.6% user,  0.0% nice,  7.8% system,  0.0% interrupt, 79.7% idle
Mem: 36M Active, 5256K Inact, 13M Wired, 6312K Cache, 15M Buf, 408K Free
Swap: 256M Total, 38M Used, 217M Free, 15% Inuse

  PID USERNAME PRI NICE  SIZE    RES STATE    TIME   WCPU    CPU COMMAND
72257 nik   28   0  1960K  1044K RUN      0:00 14.86%  1.42% top
 7078 nik   2   0 15280K 10960K select   2:54  0.88%  0.88% xemacs-21.1.14
  281 nik   2   0 18636K  7112K select   5:36  0.73%  0.73% XF86_SVGA
  296 nik   2   0  3240K  1644K select   0:12  0.05%  0.05% xterm
48630 nik   2   0 29816K  9148K select   3:18  0.00%  0.00% navigator-linu
  175 root  2   0   924K   252K select   1:41  0.00%  0.00% syslogd
 7059 nik   2   0  7260K  4644K poll     1:38  0.00%  0.00% mutt
...
```

De uitvoer is gesplitst in twee secties. De kop (de eerste vijf regels) toont het laatst uitgegeven PID, de gemiddelde systeembelasting (hoe druk is een systeem), de uptime van het systeem (tijd verstreken sinds laatste reboot) en de huidige tijd. De andere cijfers in de kop tonen hoeveel processen er draaien (in dit geval 47) , hoeveel geheugen en swap er gebruikt wordt en hoeveel processortijd het systeem besteed aan verschillende taakgroepen.

Daaronder staat een serie van kolommen die soortgelijke informatie bevatten als de uitvoer van ps(1). Zo zijn het PID, de gebruikersnaam, de hoeveelheid processortijd en het commando dat gebruikt is om het proces te starten te zien. top(1) laat standaard ook zien hoeveel geheugen er gebruikt wordt door een proces. Dit staat in twee kolommen waarbij in de eerste kolom het maximale geheugengebruik wordt getoond en in de tweede kolom het huidige geheugengebruik. Maximale gebruik is de hoeveelheid geheugen die het proces nodig had in de tijd dat het bestaat en het residente gebruik is hoeveel er op het moment van weergeven gebruikt wordt. In dit voorbeeld

is zichtbaar dat getenv(3) bijna 30 MB RAM nodig had, maar op het moment van uitvoeren 9 MB verbruikt.

top(1) werkt het beeld automatisch iedere twee seconden bij. Dat kan gewijzigd worden met de optie `s`.

3.8. Daemons, signalen en het stoppen van processen

Als een gebruiker een editor draait is het makkelijk om de editor te besturen, te vertellen om bestanden te openen, etc. Dit kan omdat de editor de mogelijkheden geeft om dat te doen en omdat de editor gekoppeld is aan een *terminal*. Sommige programma's zijn niet ontworpen om te draaien met continue gebruikersinvoer, dus als zij de kans krijgen ontkoppelen zij zich van de terminal. Een webserver reageert bijvoorbeeld de hele dag op webaanvragen en heeft eigenlijk geen input van een lokale gebruiker nodig. Programma's die email van locatie naar locatie transporteren zijn een ander voorbeeld.

Deze programma's heten *daemons*. Daemons waren karakters in de Griekste mythologie, goed noch slecht, ze waren dienende geesten die op grote schaal nuttige dingen deden voor de mensheid. Net zoals de huidige webservers en mailservers nuttige dingen doen. Dit is waarom de mascotte voor BSD al lang een vrolijk kijkende daemon met puntoren en een drietand is.

Er is een overeenkomst om programma's die meestal draaien als daemon te voorzien van het achtervoegsel "d". BIND is de Berkeley Internet Name Domain (het echte programma heet `named`), de Apache webserver heet `httpd`, de printerspooldriver heet `lpd`, etc. Deze overeenkomst geldt niet altijd. De hoofd maildaemon voor Sendmail heet bijvoorbeeld `sendmail` en niet `maild`.

Soms is communicatie met een daemon nodig. Een manier om dit te doen is het versturen van een signaal (*signals*). Er zijn een verschillende signalen. Sommige hebben een specifieke bedoeling, andere worden geïntrepeteerd door de applicatie. In de documentatie van de applicatie staat hoe de applicatie signalen intrepeteert. Er kan alleen een signaal naar een proces gezonden worden waar de uitvoerende gebruiker eigenaar van is. Als met kill(1) of kill(2) een signaal naar een proces van een andere gebruiker wordt gestuurd, wordt de toegang geweigerd. De enige uitzondering hierop is de `root` gebruiker, die signalen naar processen van alle gebruikers kan sturen.

FreeBSD stuurt soms ook signalen naar applicaties. Als een applicatie slecht geschreven is en hij probeert geheugen te benaderen waar hij niet naartoe mag, stuurt FreeBSD het proces een *Segmentation Violation* signaal (`SIGSEGV`). Als een applicatie de systeemaanroep alarm(3) heeft gebruikt om na een bepaalde periode een alarm te ontvangen, wordt er een Alarm signaal heen gestuurd (`SIGALRM`), etc.

Twee signalen kunnen gebruikt worden om een proces te stoppen: `SIGTERM` en `SIGKILL`. `SIGTERM` is de nette manier om een proces te killen. Het proces kan het signaal *afvangen*, begrijpen dat de eigenaar wil dat het wordt afgesloten, wellicht logboekbestanden sluiten die geopend zijn en alle onderhanden activiteiten afhandelen. In een aantal gevallen kan een proces `SIGTERM` negeren: als het midden in een taak zit die niet beëindigd kan worden.

`SIGKILL` mag niet worden genegeerd door een proces. Dit is het "Wat je ook aan het doen bent, stop er nu mee" signaal. Na een `SIGKILL` stopt FreeBSD het proces meteen.

Andere veelgebruikte signalen zijn SIGHUP, SIGUSR1 en SIGUSR2. Dit zijn algemeen bruikbare signalen en verschillende applicaties zullen verschillend reageren als ze verstuurd worden.

Stel dat het bestand met instellingen voor de webserver is aangepast. Dan moet aan de webserver verteld worden dat die de instellingen opnieuw moet lezen. Hiervoor zou httpd gestopt en gestart kunnen worden, maar dit resulteert in een korte onderbreking van de webserverdienst, wat ongewenst kan zijn. De meeste daemons zijn geschreven om te reageren op het SIGHUP signaal door het opnieuw inlezen van het instellingenbestand. Dus in plaats van het stoppen en herstarten van httpd kan het SIGHUP signaal gezonden worden. Omdat er geen standaard manier is om op deze signalen te reageren, reageren verschillende daemons anders. Het is verstandig eerst de documentatie van de daemon in kwestie te lezen.

Zoals onderstaand voorbeeld laat zien, worden signalen door kill(1) verzonden.

Procedure: Het versturen van een signaal naar een proces

Dit voorbeeld toont hoe een signaal naar inetd(8) wordt verstuurd. Het bestand met instellingen voor inetd is /etc/inetd.conf en inetd leest dit bestand opnieuw in als er een SIGHUP wordt verstuurd.

1. Eerst moet het proces ID worden opgezocht van het proces waar een signaal naar verzonden moeten worden. Dit kan door pgrep(1) te gebruiken.

   ```
   % pgrep -l inetd
   198  inetd -wW
   ```

   Dus het PID van inetd(8) is 198.

2. Met kill(1) kan het signaal verzonden worden. Omdat inetd(8) wordt gedraaid door root moet su(1) gebruikt worden om root te worden.

   ```
   % su
   Password:
   # /bin/kill -s HUP 198
   ```

   Zoals zovaak met UNIX® commando's, geeft kill(1) geen uitvoer als het succesvol uitgevoerd is. Als een signaal wordt verzonden naar een proces waarvan de gebruiker niet zelf de eigenaar is, dan is de melding: kill: PID: Operation not permitted. Als het PID verkeerd wordt ingevuld, wordt het signaal naar het verkeerde proces verzonden, wat slecht kan zijn, of, als de gebruiker geluk heeft, wordt het verzonden naar een PID dat momenteel niet in gebruik is, waarop de foutmelding kill: PID: No such process verschijnt.



Waarom /bin/kill gebruiken?

Veel shells leveren kill als ingebouwd commando. Dat betekent dat de shell het signaal direct verstuurt in plaats van door het starten van /bin/kill. Dit kan erg nuttig zijn, maar verschillende shells hebben een

verschillende opdrachtregel voor het specificeren van de naam van het signaal dat verstuurd moet worden. In plaats van ze allemaal te leren, is het eenvoudiger om gewoon `/bin/kill PID` te gebruiken.

Andere signalen versturen werkt bijna hetzelfde door `TERM` of `KILL` op de commandoregel te vervangen door wat nodig is.



Het stoppen van willekeurige processen op een systeem is meestal een slecht idee. In het bijzonder bij init(8) met proces ID 1. Het draaien van `/bin/kill -s KILL 1` is een snelle manier om een systeem uit te zetten. Argumenten die aan kill(1) worden meegegeven moeten *altijd* twee keer gecontroleerd worden *voordat* op `Enter` gedrukt wordt.

3.9. Shells

In FreeBSD wordt een groot deel van het alledaagse werk gedaan vanuit een omgeving met een commandoregel die shell heet. De grootste taak van een shell is om commando's van het invoerkanaal op te vangen en deze uit te voeren. Veel shells hebben ook functies ingebouwd om mee te helpen om alledaagse taken zoals bestandsbeheer, bestandsglobbing, bestanden wijzigen vanaf de commandoregel, commandomacro's schrijven en uitvoeren en omgevingsvariabelen instellen en wijzigen. FreeBSD heeft een aantal shells bijgeleverd zoals `sh`, de Bourne Shell en `tcsh`, de verbeterde C-shell. Er zijn veel andere shells beschikbaar in de FreeBSD Portscollectie zoals `zsh` en `bash`.

Welke shell gebruiken? Dit is een kwestie van smaak. Een C-programmeur voelt zich misschien prettiger bij een C-achtige shell, zoals `tcsh`. Een voormalig Linux® gebruiker of iemand die niet veel ervaring heeft met een UNIX® commandoregel interface wil misschien `bash` proberen. Elke shell heeft zijn eigen unieke eigenschappen die wel of niet werken voor een bepaalde gebruiker.

Een standaard optie in een shell is bestandsnaam completie. Door het intikken van de eerste paar letters van een commando of bestandsnaam, kan de shell opdracht gegeven worden om automatisch de rest het commando of bestandsnaam toe te voegen met de `Tab` toets op het toetsenbord. Stel dat er twee bestanden zijn met de namen foobar en foo.bar en foo.bar moet verwijderd worden. Dan kan op het toetsenbord `rm fo[Tab].[Tab]` ingevoerd worden.

De shell geeft `rm foo[BEEP].bar` weer.

De [BEEP] geeft aan dat de shell in staat was om de bestandsnaam te completeren omdat er meer dan één soortgelijk bestand was. foobar en foo.bar beginnen met `fo`, maar het was in staat om het af te maken tot `foo`. Na het invoeren van een `.` en daarna `Tab`, is de shell in staat om de rest van de bestandsnaam aan te vullen.

Een andere optie van de shell is het gebruik van omgevingsvariabelen. Omgevingsvariabelen zijn variabele sleutelparen die opgeslagen zijn in de omgevingsruimte van een shell. Deze ruimte kan uitgelezen worden door elk programma dat door de shell wordt uitgevoerd en bevat dus veel programmainstellingen. Hieronder staat een lijst van standaard omgevingsvariabelen en wat ze betekenen:

Variabele	Omschrijving
USER	Gebruikersnaam van de gebruiker die is aangemeld.
PATH	Een lijst van mappen, gescheiden door een : voor het zoeken naar binaire bestanden.
DISPLAY	Netwerknaam van het X11 scherm om verbinding mee te maken, indien beschikbaar.
SHELL	De huidige shell.
TERM	De naam van de huidige gebruikersterminal. Gebruikt om de mogelijkheden van de terminal te bepalen.
TERMCAP	Databaseregel met terminal escape codes voor het uitvoeren van diverse terminalfuncties.
OSTYPE	Type besturingssysteem, bijvoorbeeld FreeBSD.
MACHTYPE	De CPU architectuur waar het systeem op draait.
EDITOR	De teksteditor waar de gebruiker de voorkeur aan geeft.
PAGER	De tekstpager waar de gebruiker de voorkeur aan geeft.
MANPATH	Lijst van mappen gescheiden door een : voor het zoeken naar handleidingen.

Het instellen van omgevingsvariabelen verschilt van shell tot shell. In de C-achtige shells zoals tcsh en csh moet setenv gebruikt worden om omgevingsvariabelen in te stellen. In Bourne-shells zoals sh en bash moet export gebruikt worden om de omgevingsvariabelen in te stellen. Om bijvoorbeeld de omgevingsvariabele EDITOR te wijzigen naar /usr/local/bin/emacs onder csh of tcsh moet het volgende gedaan worden:

```
% setenv EDITOR /usr/local/bin/emacs
```

In Bourne shells is dat:

```
% export EDITOR="/usr/local/bin/emacs"
```

Met de meeste shells kunnen de omgevingsvariabelen ook weergegeven worden door een $ karakter voor de variabelenaam te plaatsen op de commandoregel. echo $TERM zou weergeven wat er in $TERM gezet is, omdat de shell $TERM uitbreid en het resultaat doorgeeft aan echo.

Shells kennen veel speciale karakters, die meta-karakters heten, als speciale weergaves van data. De meest voorkomende is het karakter * karakter, dat elk karakter in een bestandsnaam voorstelt. Deze speciale meta-karakters kunnen gebruikt worden om bestandsnaamglobbing te doen. Door bijvoorbeeld echo * in te voeren, is het resultaat bijna hetzelfde als door het uitvoeren van ls,

omdat de shell alle bestanden die van toepassing zijn aan echo geeft om ze daarna te tonen.

Om te voorkomen dat de shell deze speciale tekens verwerkt, kunnen ze uitgeschakeld worden door er het backslash karakter (\) voor te plaatsen. `echo $TERM` print de inhoud van TERM naar het scherm. `echo \$TERM` print $TERM zoals het geschreven is.

3.9.1. Shell wijzigen

De makkelijkste manier om de shell te wijzigen is door het `chsh` commando te gebruiken. Door `chsh` te starten wordt de editor gestart die in de `EDITOR` omgevingsvariable staat. Als deze niet is ingesteld, wordt vi gestart. In de editor kan de regel waarop "Shell:" staat gewijzigd worden.

Aan `chsh` kan ook de optie `-s` meegegeven worden. Dit stelt de shell in, zonder dat een editor gebruikt hoeft te worden. Als de shell bijvoorbeeld gewijzigd moet worden in `bash`, kan dat als volgt:

```
% chsh -s /usr/local/bin/bash
```



De te gebruiken shell *moet* geregistreerd zijn in /etc/shells. Als een shell uit de Portscollectie is geïnstalleerd, is dit meestal automatisch gebeurd. Als de shell met de hand is geïnstalleerd moet het onderstaande gedaan worden.

Als bijvoorbeeld `bash` met de hand geïnstalleerd is in /usr/local/bin, dient het onderstaande te gebeuren:

```
# echo "/usr/local/bin/bash" >> /etc/shells
```

Hierna kan `chsh` weer gedraaid worden.

3.10. Teksteditors

Een groot deel van de instellingen in FreeBSD wordt gemaakt door het bewerken van tekstbestanden. Hierdoor is het een goed idee om bekend te zijn met een tekstverwerker. FreeBSD heeft er een paar in het basissysteem en veel anderen zijn beschikbaar via de Portscollectie.

De makkelijkste en simpelste editor om te leren is de editor ee, wat "easy editor" betekent. Om ee te starten, moet op de commandoregel `ee bestandsnaam` ingevoerd worden, waar *bestandsnaam* de naam is van het bestand dat bewerkt moet worden. Om bijvoorbeeld /etc/rc.conf te bewerken, wordt `ee /etc/rc.conf` ingegeven. Eenmaal in `ee` worden alle manipulatie commando's die de editor heeft weergegeven aan de bovenkant van het scherm. Het karakter dakje `^` staat voor de toets `CTRL` op het toetsenbord, dus `^e` vormt de toetscombinatie `Ctrl` + `e`. Om uit ee te komen wordt op de toets `Esc` gedrukt en daar kan gekozen worden om de editor te verlaten. De editor vraagt dan of de wijzigingen bewaard moeten worden als het bestand veranderd is.

FreeBSD heeft ook uitgebreidere tekstverwerkers, zoals vi, in het basissysteem en andere editors als Emacs en vim maken onderdeel uit van de FreeBSD Portscollectie (editors/emacs en editors/vim).

Deze editors leveren veel meer functionaliteit en kracht maar zijn lastiger om te leren. Als echter veel met tekstverwerking gedaan wordt, is het leren van een krachtige editor als vim of Emacs verstandig omdat deze uiteindelijk veel tijd kan besparen.

Veel applicaties die bestanden wijzigen of getypte invoer nodig hebben zullen automatisch een tekstverwerker openen. Om de tekstverwerker te wijzigen die standaard wordt gebruikt, stelt u de omgevingsvariabele `EDITOR` in. Zie de sectie shells voor meer details.

3.11. Apparaten en apparaatnodes

Apparaat is een term die meestal wordt gebruikt voor hardwareonderdelen in een systeem, zoals schijven, printers grafische kaarten en toetsenborden. Als FreeBSD opstart laat het vooral zien welke apparaten gedetecteerd worden. Deze opstartmeldingen kunnen nagekeken worden door het bestand /var/run/dmesg.boot te bekijken.

acd0 is bijvoorbeeld de eerste IDE cd-rom drive, terwijl kbd0 staat voor het toetsenbord.

Veel van deze apparaten moeten in een UNIX® besturingssysteem benaderd worden via speciale bestanden die apparaatnodes heten en te vinden zijn in de map /dev.

3.11.1. Apparaatnodes maken

Als een nieuw apparaat wordt toegevoegd aan een systeem of als ondersteuning voor extra apparaten wordt gecompileerd, dan moeten er misschien nieuwe apparaat nodes aangemaakt worden.

`DEVFS` (apparaatbestandssysteem - DEVice File System)

Het apparaatbestandssysteem of `DEVFS`, levert toegang tot de apparaatruimte van de kernel in het globale bestandssysteem. In plaats van dat het nodig is om apparaatnodes te maken en te wijzigen, doet `DEVFS` dit.

In devfs(5) staat meer informatie.

3.12. Binaire formaten

Om te kunnen begrijpen waarom FreeBSD gebruik maakt van het elf(5) formaat, is het belangrijk op de hoogte zijn van de drie "dominante" uitvoerbare formaten voor UNIX®:

- a.out(5)

  Het oudste en "klassieke" UNIX® object formaat. Het gebruikt een korte en compacte kop met een magisch nummer aan het begin dat veel gebruikt wordt om het formaat aan te geven (a.out(5) geeft meer details). Het bevat drie laadbare segmenten: .tekst, .data en .bss, een symbolentabel en een stringtabel.

- COFF

  Het SVR3 object formaat. De kop bestaat uit een sectietabel, dus er kunnen meer dan alleen

.tekst, .data, en .bss secties zijn.

- elf(5)

  De opvolger van COFF, heeft meerdere secties en 32-bit of 64-bit als mogelijke waarden. Één nadeel: ELF was ook ontworpen met de aanname dat er maar één ABI per systeemarchitectuur zou zijn. Deze aanname is eigenlijk redelijk incorrect, zelfs niet in de commerciële SYSV wereld (die op zijn minst drie ABIs heeft: SRV4, Solaris en SCO).

  FreeBSD probeert om dit probleem heen te werken door een hulpprogramma te leveren voor het *brandmerken* van een bekend ELF uitvoerbaar bestand met informatie over de ABI waar hij mee kan werken. In brandelf(1) staat meer informatie.

FreeBSD komt uit het "klassieke" kamp en gebruikt het a.out(5) formaat, een technologie die zich bewezen heeft door meerdere generaties van BSD versies heen, tot het begin van de 3.X versies. Alhoewel het al mogelijk was om ELF programma's en kernels te bouwen en te draaien op een FreeBSD systeem , verzette FreeBSD zich eerst tegen de druk om over te schakelen naar ELF als standaard formaat. Waarom? Toen het Linux® kamp hun pijnlijke wissel maakte naar ELF, was dat niet zozeer om van het a.out formaat af te komen, maar meer omdat van het op de inflexibele jump-tabel gebaseerde gedeelde bibliotheekmechanisme af te komen, die het maken van gedeelde bibliotheken erg moeilijk maakte voor bedrijven en ontwikkelaars. Omdat de ELF hulprogramma's een oplossing voor het gedeelde bibliotheek probleem waren en algemeen gezien werden als een "stap vooruit", werd de migratie geaccepteerd als noodzakelijk kwaad en werd de wissel uitgevoerd. Het gedeelde bibliotheek mechanisme van FreeBSD is meer gebaseerd op het gedeelde bibliotheek mechanisme van Sun's SunOS™ en daardoor erg makkelijk te gebruiken.

Waarom zijn er zoveel verschillende formaten?

In het duistere donkere verleden was er simpele hardware. Deze simpele hardware ondersteunde een simpel klein systeem. a.out was volledig adequaat voor de taak om binaire bestanden op dat simpele systeem te vertegenwoordigen (een PDP-11). Toen mensen UNIX® van deze machine gingen porten, behielden ze het a.out formaat omdat het voldeed voor de vroege ports van UNIX® naar architecturen als Motorola 68k, VAXen, enzovoort.

Toen besloot een slimme hardware engineer dat als hij de software kon forceren om wat simpele truckjes te doen, hij in staat was om een paar onderdelen van het ontwerp af te schaven, waardoor zijn processorcore sneller kon draaien. Terwijl men probeerde om het met deze nieuwe vorm van hardware te laten werken (vandaag de dag beter bekend als RISC), was a.out te beperkt voor deze hardware. Dus werden er vele formaten ontworpen om betere prestaties te krijgen uit deze hardware dan het simpele formaat a.out kon leveren. Toen werden COFF, ECOFF en een paar andere duistere formaten uitgevonden en werden de limieten verkend, waarna men besloot om zich te richten op ELF.

Daarnaast werden programma's groter en bleven schijven (en fysiek geheugen) relatief klein, zodat het concept van een gedeelde bibliotheek werd geboren. Het VM systeem werd ook meer verfijnd. Terwijl al deze verbeteringen bereikt werden door het a.out formaat, werd het nut met elke nieuwe eigenschap verder uitgerekt. Daarnaast wilde men dingen dynamisch laden tijdens het starten of delen weggooien nadat het programma zijn intiële code had gedraaid om te blijven hangen in het hoofdgeheugen en in de wisselbestanden. Talen werden verder verfijnd en men wilde dat code

automatisch werd aangeroepen voor main. Er werden veel hacks gedaan in het a.out formaat om alles mogelijk te maken en dit werkte ook enige tijd. Na verloop van tijd was a.out niet meer in staat om alle problemen te adresseren zonder toenemende overhead in code en complexibiliteit. Hoewel ELF veel van deze problemem verhielp, was het moeilijk om te wisselen naar een systeem dat compleet anders werkte. Dus moest ELF wachten totdat het pijnlijker was om a.out te behouden dan het te migreren naar ELF.

Met het verstrijken van de tijd, werden de bouwprogramma's die FreeBSD heeft afgeleid van hun bouwprogramma's (vooral de assembler en de loader) ontwikkeld in twee parallel lopende takken. De FreeBSD tree voegde gedeelde bibliotheken toe en heeft wat bugs opgelost. De mensen van GNU die deze programma's hebben geschreven, hebben ze herschreven en simpelere ondersteuning toegevoegd voor het bouwen van cross-compilers, waarbij verschillende formaten zo nodig ingevoegd konden worden, enzovoort. Omdat veel mensen cross-compilers wilden bouwen die gericht waren op FreeBSD, hadden die pech, omdat de oudere broncode van FreeBSD voor as en ld niet opgewassen was tegen deze taak. De nieuwe GNU programmaketen (binutils) ondersteunt cross-compiling, ELF, gedeelde bibliotheken, C++ extensies, enzovoort. Daarnaast leveren veel leverancierds ELF binaire bestanden en is het goed voor FreeBSD om het te draaien.

ELF heeft meer expressiemogelijkheden dan a.out en geeft meer uitbreidingsmogelijkheden aan het basissysteem. De ELF hulpprogramma's worden beter onderhouden en geven de mogelijkheid tot ondersteuning voor cross compilatie, wat voor veel mensen belangrijk is. ELF is misschien iets trager dan a.out, maar het meten daarvan kan vrij lastig zijn. Er zijn ook ontelbare verschillen tussen de twee in hoe ze pages opslaan, initiële code verwerken, enzovoort. Geen van allen zijn ze erg belangrijk, maar er zijn verschillen. Na verloop van tijd verdwijnt de ondersteuning voor a.out uit de GENERIC kernel en uiteindelijk ook helemaal uit de kernel als de noodzaak voor a.out gebaseerde programma's voorbij is.

3.13. Meer informatie

3.13.1. Handleidingen

De meest uitvoerige documentatie van FreeBSD is geschreven in de vorm van handleidingen. Bijna elk programma op het systeem heeft een kleine handleiding die uitlegt wat de basisopties en verschillende argumenten doen. Deze handleidingen bekeken worden met `man`. Het gebruik van `man` gaat als volgt:

```
% man commando
```

`commando` is de naam van het commando waar meer informatie over getoond moet worden. Om bijvoorbeeld meer informatie weer te geven over `ls` kan het volgende uitgevoerd worden:

```
% man ls
```

De handleidingen zijn opgedeeld in genummerde onderdelen:

1. Gebruikerscommando's.

2. Systeemaanroepen en foutnummernummers.
3. Functies in de C bibliotheken.
4. Apparaatdrivers.
5. Bestandsindelingen.
6. Spelletjes en andere afleidingen.
7. Diverse informatie.
8. Systeemonderhoud en commando's
9. Kernelontwikkelaars.

In sommige gevallen kan een bepaald onderwerp vaker voorkomen in een onderdeel van de handleidingen. Er is bijvoorbeeld een gebruikerscommando `chmod` en een systeemaanroep `chmod()`. In deze gevallen kan `man` aangegeven worden welke documentatie weer te geven door het specificeren van het onderdeel:

```
% man 1 chmod
```

Dit geeft de handleiding van het gebruikerscommando `chmod` weer. Verwijzingen naar een bepaald onderdeel van de handleiding worden traditioneel tussen haakjes geplaatst: chmod(1) verwijst naar het commando `chmod` en chmod(2) verwijst naar de systeemaanroep.

Dit werkt prima als de naam van het commando bekend is en alleen informatie nodig is over hoe het commando gebruikt kan worden, maar wat als de naam van het commando niet bekend is? Dan kan `man` gebruikt worden om naar trefwoorden te zoeken in de commandobeschrijvingen door de optie `-k` te gebruiken:

```
% man -k mail
```

Met dit commando wordt een overzicht getoond met commando's die het trefwoord "mail" in hun omschrijving hebben. Dit is gelijk aan het commando `apropos`.

Dus om meer informatie over spannende commando's met een onbekende functie in /usr/bin te krijgen is het volgende commando voldoende:

```
% cd /usr/bin
% man -f *
```

Het onderstaande commando resulteert in hetzelfde:

```
% cd /usr/bin
% whatis *
```

3.13.2. Gnu infobestanden

FreeBSD heeft veel applicaties en hulpmiddelen die gemaakt zijn door de Free Software Foundation (FSF). Als extraatje voor de documentatie hebben deze programma's uitgebreidere html bestanden die `info`bestanden heten, die uitgelezen kunnen worden met `info` of, als emacs is geïnstalleerd, de infomodus van emacs.

info(1) wordt als volgt gebruikt:

```
% info
```

`h` geeft een korte beschrijving en `?` toont een kort commando-overzicht.

Hoofdstuk 4. Applicaties installeren: pakketten en ports

4.1. Overzicht

FreeBSD bevat een grote collectie aan systeemgereedschappen als onderdeel van het basissysteem. De mogelijkheden reiken echter niet heel ver en daarom is er snel een applicatie van een andere partij nodig. FreeBSD bevat twee complementaire technologieën om andere applicaties te installeren: de FreeBSD Portscollectie (voor het installeren vanuit broncode) en pakketten (voor het installeren vanuit voorgecompileerde binaire bestanden). Beide systemen kunnen gebruikt worden om de nieuwste versies van een gewenste applicatie te installeren van lokale media of rechtstreeks van het netwerk.

Na het lezen van dit hoofdstuk weet de lezer:

- Hoe binaire softwarepakketten van derden te installeren;
- Hoe software van derden vanuit de Portscollectie vanuit broncode te installeren;
- Hoe eerder geïnstalleerde pakketten of ports te verwijderen;
- Hoe standaardwaarden die door de ports worden gebruikt te wijzigen;
- Hoe het juiste softwarepakket te vinden;
- Hoe applicaties bij te werken.

4.2. Overzicht van softwareinstallatie

Als u eerder gebruik heeft gemaakt van een UNIX®-systeem dan is het bekend dat de standaardprocedure voor het installeren van software van derden ongeveer als volgt is:

1. Download de software als broncode of als binair bestand;
2. Pak de software uit vanuit zijn originele distributietype (meestal een tar-bestand gecomprimeerd met compress(1), gzip(1), of bzip2(1));
3. Zoek de documentatie (meestal een INSTALL of README bestand of enkele bestanden in een submap doc/) en lees zorgvuldig hoe de software geïnstalleerd moet worden;
4. Als de software als broncode is gedistribueerd, moet de broncode gecompileerd worden. Dit kan wijzigingen in een Makefile vereisen of het draaien van een `configure` script en andere werkzaamheden;
5. De software installeren en testen.

En dat geldt alleen als alles goed gaat. Als er een softwarepakket geïnstalleerd wordt dat niet specifiek gemaakt is voor FreeBSD moet mogelijkerwijs zelfs de code aangepast worden om alles goed te laten werken.

Als de gebruiker het wenst, kan hij in FreeBSD doorgaan met het installeren van software op de "traditionele" manier. FreeBSD levert echter twee technologieën die veel moeite kunnen besparen: pakketten en ports. Op dit moment zijn zo meer dan 37000 applicaties beschikbaar.

Voor iedere gewenste applicatie is het FreeBSD pakket voor die applicatie één te downloaden bestand. Het pakket bevat voorgecompileerde kopiën met alle commando's voor de applicatie en alle instellingenbestanden of documentatie. Een gedownload pakketbestand kan gemanipuleerd worden met FreeBSD pakketbeheercommando's zoals pkg_add(1), pkg_delete(1), pkg_info(1), enzovoort. Het installeren van een nieuwe applicatie kan met één commando.

Een FreeBSD port van een applicatie is een groep bestanden ontworpen om het proces van compileren van een applicatie vanuit broncode te automatiseren.

Het is te vergelijken met de stappen die normaal gevolgd worden om een programma te compileren (downloaden, uitpakken, aanpassen, compileren en installeren). De bestanden die samen een port vormen bevatten alle noodzakelijke informatie om het systeem dit te laten doen. Met een aantal eenvoudige commando's wordt de broncode voor de applicatie automatisch gedownload, uitgepakt, aangepast, gecompileerd en geïnstalleerd.

Het portssysteem kan zelfs gebruikt worden om pakketten te maken die later weer gemanipuleerd kunnen worden met `pkg_add` en andere pakketbeheercommando's, waarover later meer uitleg wordt gegeven.

Zowel pakketten als ports kennen afhankelijkheden (*dependencies*). Stel dat er een applicatie geïnstalleerd gaat worden die er vanuit gaat dat een specifieke bibliotheek wordt geïnstalleerd. Zowel de applicatie als de bibliotheek zijn beschikbaar als FreeBSD ports en pakketten. Als het commando `pkg_add` of het portssysteem wordt gebruikt om de applicatie toe te voegen, dan zien beiden dat de bibliotheek niet geïnstalleerd is en wordt deze automatisch eerst geïnstalleerd.

Gezien het feit dat beide technologieën vrijwel identiek zijn, kan de vraag rijzen waarom FreeBSD de moeite neemt om beide te faciliteren. Pakketten en ports hebben ieder hun eigen kracht. Welke gebruikt wordt hangt af van voorkeuren en omstandigheden.

Voordelen van pakketten

- Een gecomprimeerd pakket tar-bestand is meestal kleiner dan het gecomprimeerde tar-bestand met de broncode van de applicatie;
- Pakketten vereisen geen additionele compilatie. Voor grote applicaties als Mozilla, KDE of GNOME kan dit belangrijk zijn, vooral als een systeem wat trager is;
- Pakketten vereisen geen begrip van het proces van het compileren van software op FreeBSD.

Voordelen van ports

- Pakketten worden meestal gecompileerd met conservatieve opties, omdat ze moeten draaien op een maximaal aantal systemen. Bij het installeren vanuit de port kunnen de compilatie-instellingen aangepast worden om zo bijvoorbeeld code te maken die specifiek voor een Pentium 4 of een Athlon processor is;
- Sommige applicaties hebben compilatie-instellingen gerelateerd aan wat ze wel of niet kunnen doen. Apache kan bijvoorbeeld ingesteld worden met een uitgebreide hoeveelheid verschillende ingebouwde instellingen. Door vanuit de port te werken hoeven niet alle

standaardinstellingen geaccepteerd te worden en kunnen ze ingesteld worden;

In sommige gevallen zijn er meerdere pakketten voor dezelfde applicatie om specifieke instellingen aan te geven. Ghostscript is bijvoorbeeld beschikbaar als een ghostscript pakket en ghostscript-nox11 pakket, afhankelijk van het al dan niet geïnstalleerd hebben van een X11 server. Deze ruwe vorm van tweaking is mogelijk met pakketten, maar dit wordt snel onmogelijk als een applicatie meer dan één of twee verschillende compilatie-instellingen heeft;

- De licentievoorwaarden van sommige softwaredistributies verbieden binaire distributie. Ze moeten dus gedistribueerd worden als broncode;
- Sommige mensen vertrouwen binaire distributies niet. Broncode kan tenminste (in theorie) zelf doorgelezen en gecontroleerd worden op potentiële problemen;
- Als er lokale modificaties zijn, is de broncode nodig om ze toe te passen;
- Sommige mensen hebben graag de broncode zodat ze die kunnen lezen als ze zich vervelen, erin kunnen hacken, code kunnen overnemen (indien de licentie dit toestaat natuurlijk), enzovoort.

Om vernieuwingen van ports bij te houden kan een abonnement genomen worden op de FreeBSD ports mailinglijst en/of de FreeBSD ports bugs mailinglijst.



Voordat een applicatie wordt geïnstalleerd is het aan te raden op http://vuxml.freebsd.org/ na kijken of er geen beveiligingsproblemen voor de gewenste applicatie bekend zijn.

Het is ook mogelijk om ports-mgmt/portaudit te installeren, dat automatisch alle geïnstalleerde applicaties controleert op bekende fouten. Deze controle wordt ook uitgevoerd voordat een port wordt geïnstalleerd. Met het commando `portaudit -F -a` kunnen de pakketten die al geïnstalleerd zijn worden gecontroleerd.

In de rest van dit hoofdstuk wordt uitgelegd hoe pakketten en ports gebruikt kunnen worden om software in FreeBSD te installeren en te beheren.

4.3. Applicaties zoeken

Voordat een applicatie geïnstalleerd kan worden, moeten de doelen bekend zijn en hoe de applicatie heet.

De lijst met voor FreeBSD beschikbare applicaties groeit continu. Gelukkig zijn er een aantal manieren om te zoeken:

- Op de FreeBSD website staat een recente doorzoekbare lijst met alle beschikbare applicaties: http://www.FreeBSD.org/ports/. De ports zijn onderverdeeld in categorieën. Er kan naar een applicatie gezocht worden op naam (als die bekend is) of alle applicaties in een categorie kunnen bekeken worden.
- Dan Langille onderhoudt FreshPorts op http://www.FreshPorts.org/. FreshPorts volgt veranderingen in applicaties in de ports en biedt de mogelijkheid om of meer ports te volgen. Er wordt dan een email gestuurd als de port is bijgewerkt.

- Als de naam van de gewenst applicatie niet bekend is, is het wellicht mogelijk deze te achterhalen via een website als Freecode (http://www.freecode.com/) en kan daarna op de FreeBSD site gecontroleerd worden of de applicatie al geschikt gemaakt is voor gebruik met FreeBSD.
- Als de precieze naam van de port bekend is, maar niet bekend is in welke categorie deze staat, kan dit achterhaald worden met whereis(1). Door simpelweg `whereis bestand` in te geven, waar *bestand* het te installeren programma is. Als het op het systeem staat, wordt dat als volgt aangegeven:

  ```
  # whereis lsof
  lsof: /usr/ports/sysutils/lsof
  ```

  Dit geeft aan dat `lsof` (een systeemhulpprogramma) in de map /usr/ports/sysutils/lsof staat.

- U kunt ook een eenvoudig echo(1)-statement gebruiken om uit te zoeken waar een port zich in te ports tree bevindt. Bijvoorbeeld:

  ```
  # echo /usr/ports/*/*lsof*
  /usr/ports/sysutils/lsof
  ```

  Merk op dat dit alle overeenkomstige bestanden die gedownload zijn in de map /usr/ports/distfiles terruggeeft.

- Nog een andere manier om een port op te sporen is door het ingebouwde zoekmechanisme van de Portscollectie te gebruiken. Hiervoor moet het huidige pad de map /usr/ports zijn. Vanuit die map kan `make search name=programmanaam` uitgevoerd worden, waar *programmanaam* de naam is van het programma dat wordt gezocht. Als bijvoorbeeld `lsof` wordt gezocht:

  ```
  # cd /usr/ports
  # make search name=lsof
  Port:   lsof-4.56.4
  Path:   /usr/ports/sysutils/lsof
  Info:   Lists information about open files (similar to fstat(1))
  Maint:  obrien@FreeBSD.org
  Index:  sysutils
  B-deps:
  R-deps:
  ```

  Het belangrijkste onderdeel van de uitvoer is in dit geval de regel waarop "Path:" staat, omdat die aangeeft waar de port staat. De andere informatie is niet nodig voor de installatie van de port en wordt hier niet behandeld.

  Voor nog dieper zoeken kan ook `make search key=string` gebruikt worden waar *string* tekst is waarnaar gezocht moet worden. Hiermee wordt naar namen van ports, commentaar, beschrijvingen en afhankelijkheden gezocht en dit kan gebruikt worden om ports te vinden die te maken hebben met een bepaald onderwerp als onbekend is hoe het gezochte programma

heet.

In beide gevallen is de zoekstring niet hoofdlettergevoelig. Zoeken naar "LSOF" geeft hetzelfde resultaat als zoeken naar "lsof".

4.4. Het pakkettensysteem gebruiken

Er zijn verschillende gereedschappen die gebruikt worden om pakketten op FreeBSD te beheren:

- Het gereedschap `sysinstall` kan op een draaiend systeem worden gebruikt om beschikbare en geïnstalleerde pakketten te installeren, te verwijderen, en weer te geven. Zie voor meer informatie Pakketten installeren.
- De opdrachtregelgereedschappen om pakketten te beheren, welke het onderwerp van de rest van deze sectie zijn.

4.4.1. Pakketten installeren

Met pkg_add(1) kan een FreeBSD softwarepakket geïnstalleerd worden vanaf een lokaal bestand of vanaf een server op het netwerk.

Voorbeeld 3. Handmatig pakketten downloaden en lokaal installeren

```
# ftp -a ftp2.FreeBSD.org
Connected to ftp2.FreeBSD.org.
220 ftp3.FreeBSD.org FTP server (Version 6.00LS) ready.
331 Guest login ok, send your email address as password.
230-
230-     This machine is in Vienna, VA, USA, hosted by Verio.
230-         Questions? E-mail freebsd@vienna.verio.net.
230-
230-
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /pub/FreeBSD/ports/packages/sysutils/
250 CWD command successful.
ftp> get lsof-4.56.4.tgz
local: lsof-4.56.4.tgz remote: lsof-4.56.4.tgz
200 PORT command successful.
150 Opening BINARY mode data connection for 'lsof-4.56.4.tgz' (92375 bytes).
100% |***********************************************| 92375       00:00 ETA
226 Transfer complete.
92375 bytes received in 5.60 seconds (16.11 KB/s)
ftp> exit
# pkg_add lsof-4.56.4.tgz
```

Als er lokaal geen bron is voor pakketten (zoals de FreeBSD CD-ROM-verzameling) dan is het

waarschijnlijk makkelijker om de `-r` optie te gebruiken met pkg_add(1). Deze optie zorgt er voor dat het hulpprogramma automatisch het correcte formaat en de juiste versie bepaalt en die daarna binnenhaalt en installeert vanaf een FTP site.

```
# pkg_add -r lsof
```

Het voorbeeld hierboven haalt het correcte pakket binnen en installeert het zonder dat de gebruiker iets hoeft te doen. Als u een alternatieve FreeBSD Pakkettenmirror wilt specificeren, in plaats van de hoofddistributiesite, dan moet u de omgevingsvariabele `PACKAGESITE` overeenkomstig instellen om de standaardinstellingen aan te passen. pkg_add(1) gebruikt fetch(3) om de bestanden binnen te halen, dat gebruik maakt van diverse omgevingsvariabelen zoals `FTP_PASSIVE_MODE`, `FTP_PROXY`, en `FTP_PASSWORD`. Mogelijk moeten ook één of meer van deze variabelen gebruikt worden als een machine achter een firewall staat of als gebruik gemaakt moet worden van een FTP/HTTP proxy. In fetch(3) staat de complete lijst. In het voorbeeld hierboven is gebruik gemaakt van `lsof` in plaats van `lsof-4.56.4`. Als het pakket wordt binnengehaald met behulp van de bovenstaande instellingen, dan moet het versienummer van het pakket niet gebruikt worden. pkg_add(1) haalt automatisch de laatste versie van de applicatie binnen.



pkg_add(1) downloadt de meest recente versie van een applicatie als FreeBSD-CURRENT of FreeBSD-STABLE. Als een -RELEASE versie wordt gebruikt, wordt het pakket dat bij die release hoort gebruikt. Het is mogelijk dit gedrag te veranderen door `PACKAGESITE` te wijzigen. Als u bijvoorbeeld FreeBSD 8.1-RELEASE draait, dan haalt pkg_add(1) standaard de pakketten uit `ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-8.1-release/Latest/`. Om pkg_add(1) de FreeBSD 8-STABLE pakketten te laten downloaden kan `PACKAGESITE` ingesteld worden op `ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-8-stable/Latest/`.

Pakketbestanden worden gedistribueerd in de formaten .tgz en .tbz. Ze zijn te vinden op ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/packages/ of op de FreeBSD CD-ROM-distributie. Iedere CD-ROM in de FreeBSD 4-CD-ROM-verzameling (en de PowerPak, enzovoort) bevat pakketten in de map /packages. De opbouw van de pakketten is ongeveer gelijk aan die van /usr/ports. Iedere categorie heeft zijn eigen map en ieder pakket staat ook in de map All.

De mappenstructuur van het pakkettensysteem is gelijk aan die van het portssysteem. Samen vormen ze het pakket/portssysteem.

4.4.2. Pakketten beheren

pkg_info(1) is een hulpprogramma dat de diverse geïnstalleerde pakketten toont en beschrijft.

```
# pkg_info
cvsup-16.1          A general network file distribution system optimized for CV
docbook-1.2         Meta-port for the different versions of the DocBook DTD
...
```

pkg_version(1) is een hulpprogramma dat een samenvatting van de versie van alle geïnstalleerde

pakketten geeft. Het vergelijkt de versie van het pakket met de huidige versie in de Portscollectie.

```
# pkg_version
cvsup                         =
docbook                       =
...
```

De symbolen in de tweede kolom geven aan hoe de geïnstalleerde versie staat ten opzichte van de versie die beschikbaar is in de lokale Portscollectie.

Symbool	Betekenis
=	De versie van het geïnstalleerde pakket komt overeen met die in de lokale Portscollectie.
<	De geïnstalleerde versie is ouder dan die beschikbaar is in de ports.
>	De geïnstalleerde versie is nieuwer dan die in de lokale Portscollectie. De lokale Portscollectie is waarschijnlijk verouderd.
?	Het geïnstalleerde pakket kan niet gevonden worden in index van de Portscollectie. Dit kan bijvoorbeeld gebeuren als een geïnstalleerde port uit de Portscollectie wordt verwijderd of hernoemd.
*	Er zijn meerdere versies van het pakket.
!	Het geïnstalleerde pakket bestaat in de index maar om de een of andere reden was `pkg_version` niet in staat om het versienummer van het geïnstalleerde pakket met de overeenkomstige ingang in de index te vergelijken.

4.4.3. Pakketten verwijderen

Voor het verwijderen van een geïnstalleerd pakket wordt het hulpprogramma pkg_delete(1) gebruikt.

```
# pkg_delete xchat-1.7.1
```

Merk op dat pkg_delete(1) de volledige naam en het volledige nummer van het pakket nodig heeft; het bovenstaande commando zou niet werken als *xchat* in plaats van *xchat-1.7.1* was gegeven. Het is echter eenvoudig om pkg_version(1) te gebruiken om de versie van het geïnstalleerde pakket te achterhalen. U zou ook eenvoudigweg een wildcard kunnen gebruiken:

```
# pkg_delete xchat\*
```

In dit geval zullen alle pakketten waarvan de naam met `xchat` begint worden verwijderd.

4.4.4. Diversen

Alle informatie over pakketten wordt opgeslagen in de map /var/db/pkg. De lijst met geïnstalleerde bestanden en beschrijvingen van ieder pakket staat in de bestanden in deze map.

4.5. De Portscollectie gebruiken

In de volgende paragrafen worden basisinstructies gegeven over het gebruik van de Portscollectie om programma's op een systeem te installeren of ervan te verwijderen. Een gedetailleerde beschrijving van de `make`-doelen en omgevingsvariabelen staat in ports(7).



Sinds eind 2012 is het FreeBSD Ports Project bezig om het versiebeheersysteem te migreren van CVS naar Subversion. Als gevolg hiervan zijn deze instructies aan verandering onderhevig. Het aanbevolen mechanisme voor algemeen gebruik van de ports is Portsnap. Gebruikers die lokale aanpassingen van ports nodig hebben (dus aanvullende lokale patches beheren) zullen er waarschijnlijk de voorkeur aan geven om rechtstreeks Subversion te gebruiken. De dienst CVSup wordt per 28 februari 2013 uitgefaseerd en verder gebruik wordt ontmoedigd.

4.5.1. De Portscollectie verkrijgen

De Portscollectie is een verzameling van Makefiles, patches en bestanden met beschrijvingen in /usr/ports. Deze verzameling bestanden wordt gebruikt om applicaties op FreeBSD te bouwen en te installeren. De onderstaande instructies laten verschillende methodes zien om de Portscollectie te verkrijgen als dit niet tijdens de initiële installatie van FreeBSD is gebeurd.

Procedure: Met Portsnap

Portsnap is een snel en gebruiksvriendelijk gereedschap om de Portscollectie te verkrijgen en de aanbevolen manier voor de meeste gebruikers. Zie Portsnap gebruiken voor een gedetailleerde beschrijving van Portsnap.

1. Download een gecomprimeerde momentopname van de Portscollectie naar /var/db/portsnap.

   ```
   # portsnap fetch
   ```

2. Pak de momentopname bij het eerste gebruik van Portsnap uit naar /usr/ports:

   ```
   # portsnap extract
   ```

Nadat het eerste gebruik van Portsnap is voltooid zoals hierboven is aangegeven, kan /usr/ports worden bijgewerkt met:

```
# portsnap update
```

Procedure: Met Subversion

Als meer controle over de ports-boom nodig is (om bijvoorbeeld lokale veranderingen te beheren) kan Subversion worden gebruikt om de Portscollectie te verkrijgen. Zie de Subversion Primer voor een gedetailleerde beschrijving van Subversion.

1. Subversion moet geïnstalleerd zijn voordat het gebruikt kan worden om de ports-boom uit te checken. Als er reeds een kopie van de ports-boom aanwezig is, installeer dan Subversion als volgt:

   ```
   # cd /usr/ports/devel/subversion
   # make install clean
   ```

   Als de ports-boom niet beschikbaar is, kan Subversion worden geïnstalleerd als een pakket:

   ```
   # pkg_add -r subversion
   ```

   Als pkgng wordt gebruikt om pakketten te beheren, kan Subversion in plaats daarvan worden geïnstalleerd met:

   ```
   # pkg install subversion
   ```

2. Check een kopie van de ports-boom uit. Gebruik voor een betere prestatie een specifieke Subversion mirror dichtbij u in plaats van *svn.FreeBSD.org* in onderstaand commando. Committers dienen eerst de Subversion Primer te lezen om er zeker van te zijn dat het juiste protocol is gekozen.

   ```
   # svn checkout svn://svn.FreeBSD.org/ports/head /usr/ports
   ```

3. Om /usr/ports na de initiële checkout met Subversion bij te werken:

   ```
   # svn update /usr/ports
   ```

Procedure: Met CVSup



Het gebruik van CVsup om de Portscollectie te verkrijgen en te synchroniseren wordt ontmoedigd als onderdeel van een migratie naar Subversion. Hoewel het ondersteund blijft zal de dienst niet meer worden geleverd na 28 februari 2013.

Dit is een snelle methode voor het verkrijgen en bijhouden van een kopie van Portscollectie met behulp van het CVSup-protocol. Meer informatie over CVSup staat in CVSup gebruiken.



De implementatie van het CVSup-protocol dat met FreeBSD wordt geleverd heet csup.

Zorg ervoor dat /usr/ports leeg is voordat csup voor het eerst gebruikt wordt! Als er reeds een Ports Collectie aanwezig is die via een andere bron is opgehaald, zal csup verwijderde patchbestanden niet verwijderen.

1. Draai `csup`:

   ```
   # csup -L 2 -h cvsup.FreeBSD.org /usr/shared/examples/cvsup/ports-supfile
   ```

   Wijzig *cvsup.FreeBSD.org* in een CVSup server in de buurt. In CVSup Mirrors (CVSup sites) staat een complete lijst van mirrorsites;

   Het kan wenselijk zijn een aangepaste ports-supfile te gebruiken, bijvoorbeeld om een CVSup server niet mee te hoeven geven op de commandoregel.

   a. Kopieer in dit geval, als `root`, /usr/shared/examples/cvsup/ports-supfile naar een nieuwe locatie, zoals /root of een thuismap.

   b. Wijzig ports-supfile.

   c. Wijzig *CHANGE_THIS.FreeBSD.org* in een CVSup server in de buurt. In CVSup Mirrors (CVSup sites) staat een volledige lijst met mirrorsites.

   d. Roep nu als volgt `csup` aan:

      ```
      # csup -L 2 /root/ports-supfile
      ```

2. Het later draaien van csup(1) zal alle recente veranderingen aan uw Portscollectie downloaden en toepassen, behalve het eigenlijke herbouwen van ports voor uw eigen systeem.

Procedure: Met sysinstall

Bij deze methode wordt sysinstall gebruikt om de Portscollectie van installatiemedia te installeren. Hier wordt wel de Portscollectie op het moment dat de release gemaakt is geïnstalleerd. Bij toegang tot Internet is het advies altijd een andere methode te gebruiken.

1. Draai als `root sysinstall` zoals hieronder aangegeven:

```
# sysinstall
```

2. Scroll naar beneden en selecteer Configure, druk op `Enter`.
3. Scroll naar beneden en selecteer Distributions, druk op `Enter`.
4. Scroll naar ports, druk op `Space`.
5. Scroll naar boven naar Exit, druk op `Enter`.
6. Selecteer de gewenste installatiemedia, zoals CD-ROM, FTP, enzovoort.
7. Scroll omhoog naar Exit en druk op `Enter`.
8. Druk op `X` om sysinstall af te sluiten.

4.5.2. Migreren van CVSup/csup naar portsnap



Per 28 februari 2013 zal de ports-boom niet langer naar CVS worden en daarom geëxporteerd zullen CVSup en csup niet langer updates voor de Portscollectie bieden.

Procedure: Migreren naar Portsnap

De migratie zal ongeveer 1 GB aan schijfruimte op /usr nodig hebben, en Portsnap zal ongeveer 150 MB aan schijfruimte op /var nodig hebben.

1. Schakel alle automatische updates aan ports die u gebruikt, zoals een cron(8)-taak die CVSup of csup uit.
2. Verplaats de bestaande ports-boom naar een tijdelijke lokatie:

```
# mv /usr/ports /usr/ports.old
```

3. Haal de nieuwe ports-boom met Portsnap op en pak deze uit in /usr/ports:

```
# portsnap fetch extract
```

4. Verplaats distfiles en bewaarde pakketten naar de nieuwe ports-boom:

```
# mv /usr/ports.old/distfiles /usr/ports
```

```
# mv /usr/ports.old/packages /usr/ports
```

5. Verwijder de oude ports-boom:

```
# rm -rf /usr/ports.old
```

6. Indien voorheen CVSup gebruikt werd, kan het nu worden gedeïnstalleerd:

```
# pkg_delete -r -v cvsup-without-gui-\*
```

   Gebruikers van pkgng kunnen het volgende commando gebruiken:

```
# pkg remove cvsup-without-gui
```

Zie Portsnap gebruiken voor een gedetailleerde beschrijving van Portsnap en hoe de ports-boom met Portsnap bij te werken.

4.5.3. Ports installeren

Het eerste wat uitleg behoeft als het over de Portscollectie gaat is de term "skelet" ("skeleton"). In een notendop is een portskelet een minimaal aantal bestanden dat FreeBSD aangeeft hoe een programma gecompileerd en geïnstalleerd kan worden. Ieder portskelet bevat:

- Een Makefile. De Makefile bevat verschillende definities die aangeven hoe de applicatie gecompileerd moet worden en waar die op een systeem geïnstalleerd moet worden;
- Een bestand distinfo. Dit bestand bevat informatie over de bestanden die gedownload moeten worden om de port te bouwen, en hun checksums (door gebruik te maken van sha256(1)), om vast te stellen dat de bestanden niet corrupt zijn geraakt tijdens de download;
- Een map files. Deze map bevat patches om het programma op een FreeBSD systeem te laten compileren en installeren. Patches zijn in essentie kleine bestanden waarin kleine veranderingen aan andere, specifieke, bestanden staan aangegeven. Ze zijn opgesteld in platte tekst en er staan dingen in als "Verwijder regel 10" of "Wijzig regel 26 in ...". Patches staan ook wel bekend als "diffs" omdat ze gemaakt worden met het programma diff(1).

  Deze map kan ook andere bestanden bevatten die gebruikt worden om de port te bouwen;

- Een bestand pkg-descr. Dit is een meer gedetailleerde beschrijving van het programma, vaak in één regel;
- Een bestand pkg-plist. Dit is een lijst met alle bestanden die door de port geïnstalleerd worden. Het geeft het portssysteem ook aan welke bestanden bij het verwijderen van de port weer verwijderd kunnen worden.

Sommige ports bevatten nog andere bestanden, zoals pkg-message. Het portssysteem gebruikt die bestanden voor het afhandelen van bijzondere situaties. Meer details over die bestanden en over

ports in het algemeen zijn na te lezen in het FreeBSD Handboek voor Porters.

De port bevat instructies over hoe de broncode gebouwd moet worden, maar de broncode zelf is er geen onderdeel van. De broncode staat op een CD-ROM of op Internet. De broncode wordt verspreid op de wijze waarop de auteur dat wenst. Vaak is dat als een tar of gzip bestand, maar het kan ook ingepakt zijn met een ander programma of helemaal niet ingepakt zijn. De broncode van een programma, in welke vorm dan ook, heet een "distributiebestand". De twee methoden om een FreeBSD port te installeren worden hieronder beschreven.



Ports installeren dient als `root` te gebeuren.

Voordat een port wordt geïnstalleerd is het aan te raden op http://vuxml.freebsd.org/ na kijken of er geen beveiligingsproblemen voor de gewenste port bekend zijn.



Er kan automatisch een controle op beveiligingsproblemen door portaudit gedaan worden voordat er een nieuwe applicatie wordt geïnstalleerd. Dit gereedschap kan in de Portscollectie gevonden worden (ports-mgmt/portaudit). Overweeg om `portaudit -F` te draaien voordat er een nieuwe port wordt geïnstalleerd, om de huidige database met beveiligingsproblemen op te halen. Tijdens de dagelijkse beveiligingscontrole van het systeem zal er een beveiligingsaudit en een update van de database plaatsvinden. Lees voor meer informatie de hulppagina's portaudit(1) en periodic(8).

De Portscollectie neemt aan dat er een werkende Internetverbinding is. Als die niet aanwezig is, zet dan handmatig een kopie van het benodigde distributiebestand in /usr/ports/distfiles.

Ga om te beginnen naar de juiste map voor een port:

```
# cd /usr/ports/sysutils/lsof
```

Eenmaal in de map lsof is het skelet van de port te zien. In de volgende stap wordt de broncode voor de port gecompileerd of "gebouwd". Dit wordt gedaan door op het prompt `make` in te voeren. Dat levert iets als het volgende op:

```
# make
>> lsof_4.57D.freebsd.tar.gz doesn't seem to exist in /usr/ports/distfiles/.
>> Attempting to fetch from ftp://lsof.itap.purdue.edu/pub/tools/unix/lsof/.
===>  Extracting for lsof-4.57
...
[uitvoer van uitpakken verwijderd]
...
>> Checksum OK for lsof_4.57D.freebsd.tar.gz.
===>  Patching for lsof-4.57
===>  Applying FreeBSD patches for lsof-4.57
===>  Configuring for lsof-4.57
...
[uitvoer van configure verwijderd]
```

```
...
===>  Building for lsof-4.57
...
[uitvoer van compileren verwijderd]
...
#
```

Als het compileren is afgerond is het prompt weer zichtbaar. In de volgende stap wordt de port geïnstalleerd. Om dat te bewerkstelligen wordt het woord `install` aan `make` toegevoegd:

```
# make install
===>  Installing for lsof-4.57
...
[uitvoer installatie verwijderd]
...
===>   Generating temporary packing list
===>   Compressing manual pages for lsof-4.57
===>   Registering installation for lsof-4.57
===>  SECURITY NOTE:
      This port has installed the following binaries which execute with
      increased privileges.
#
```

Als de prompt weer beschikbaar is, is de applicatie klaar voor gebruik. Omdat `lsof` met verhoogde rechten wordt uitgevoerd, wordt er een waarschuwing getoond. Tijdens het bouwen en installeren van ports zijn de getoonde waarschuwingen van belang.

Het is verstandig om de submap die als werkmap wordt gebruikt te verwijderen. Hierin staan alle tijdelijke bestanden die tijdens het compileren worden gebruikt. Die bestanden gebruiken niet alleen waardevolle schijfruimte, maar ze kunnen later ook problemen veroorzaken als de port wordt bijgewerkt.

```
# make clean
===>  Cleaning for lsof-4.57
#
```



Het is mogelijk twee stappen minder te gebruiken door `make install clean` uit te voeren in plaats van `make`, `make install` en `make clean` als drie afzonderlijke stappen.



Wanneer een port alleen met `make install` wordt geïnstalleerd, betekent dit dat er in het begin mogelijk veel gewacht moet worden tussen interacties van de gebruiker aangezien het standaardgedrag is om de gebruiker te vragen om keuzes voor opties. Wanneer er veel afhankelijkheden zijn, kan dit voor het bouwen van een enkele port soms een hele opgave zijn. Om dit te voorkomen, kan `make config-recursive` gedraaid worden om de configuratie in één keer te doen. Draai daarna

`make install [clean]`.



Wanneer `config-recursive` wordt gebruikt, wordt de lijst met ports om te configureren opgesteld door het doel `all-depends-list` van make(1). Het wordt vaak aangeraden om `make config-recursive` totdat de opties van alle afhankelijke ports zijn gedefinieerd en er geen schermen van dialog(1) voor opties van ports meer verschijnen, om er zeker van te zijn dat de opties van alle ports zijn geconfigureerd zoals bedoeld.



Sommige shells houden een cache bij van de commando's die in de mappen uit de omgevingsvariabele `PATH` staan om het opzoeken van een uitvoerbaar bestand te versnellen. Als zo'n shell wordt gebruikt, moet er na de installatie van een port het commando `rehash` worden uitgevoerd voordat zojuist geïnstalleerde commando's kunnen worden gebruikt. Dit commando werkt voor shells zoals `tcsh`. Gebruik voor shells als `sh hash -r`. In de documentatie van een shell staat meer informatie.

Sommige DVD-ROM-producten van andere partijen, zoals de FreeBSD Toolkit van de FreeBSD Mall bevatten distributiebestanden. Die kunnen met de Portscollectie gebruikt worden. Koppel de DVD-ROM aan op /cdrom. Stel bij gebruik van een ander aankoppelpunt de make variabele `CD_MOUNTPTS` in. De benodigde distributiebestanden worden automatisch gebruikt als ze op de schijf aanwezig zijn.



Licenties van sommige ports staan niet toe dat de code wordt opgenomen in een CD-ROM. Dit kan komen doordat er een formulier ingevuld moet worden voor een download of doordat herdistributie niet is toegestaan of om een andere reden. Om een port te installeren die niet op de CD-ROM staat moet de computer waarop de port geïnstalleerd wordt een Internetverbinding hebben.

Het portssysteem gebruikt fetch(1) om bestanden te downloaden. Dat programma maakt gebruik van een aantal omgevingsvariabelen, waaronder `FTP_PASSIVE_MODE`, `FTP_PROXY`, en `FTP_PASSWORD`. Als een systeem achter een firewall staat, is het wellicht noodzakelijk om een of meer van deze omgevingsvariabelen in te stellen of om gebruik te maken van een FTP/HTTP proxy. In fetch(3) staat een complete lijst.

Als er geen continue Internetverbinding is, kan gebruik gemaakt worden van `make fetch`. Door dit commando in de map /usr/ports uit te voeren worden *alle* benodigde bestanden gedownload. Dit commando werkt ook op een lager niveau als /usr/ports/net of /usr/ports/net/xmule. Als een port afhankelijk is van bibliotheken of andere ports dan worden de distributiebestanden van die ports *niet* opgehaald. Om dat de bereiken dient `fetch` vervangen te worden door `fetch-recursive`.



Het is mogelijk alle ports in een categorie te bouwen door `make` in een hogere map uit te voeren, naar analogie van het voorbeeld voor `make fetch`. Dit is wel gevaarlijk, omdat sommige ports niet tegelijk met andere geïnstalleerd kunnen zijn. In andere gevallen installeren twee ports hetzelfde bestand met een andere inhoud.

In zeldzame gevallen willen of moeten gebruikers de tar-bestanden van een andere site dan de

MASTER_SITES halen (de locatie waar de bestanden vandaan komen). Dat is mogelijk met de optie MASTER_SITES met een volgend commando:

```
# cd /usr/ports/directory
# make MASTER_SITE_OVERRIDE= \
ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/distfiles/ fetch
```

In het voorgaande voorbeeld is de optie MASTER_SITES gewijzigd naar ftp.FreeBSD.org/pub/FreeBSD/ports/distfiles/.



Sommige ports staan toe (of schrijven zelfs voor) dat er een aantal instellingen worden meegegeven die bepaalde onderdelen (niet gebruikt, beveiligingsinstellingen en andere aanpassingen) van de applicatie in- of uitschakelen. Voorbeelden van ports waarbij dat het geval is zijn www/firefox, security/gpgme en mail/sylpheed-claws. Er wordt een bericht getoond als dit soort instellingen beschikbaar zijn.

Standaardmappen voor ports wijzigen

Soms is het handig (of verplicht) om een andere map voor werk of ports te gebruiken. Met de variabelen WRKDIRPREFIX en PREFIX kunnen de standaardmappen veranderd worden:

```
# make WRKDIRPREFIX=/usr/home/example/ports install
```

Het voorbeeld hierboven compileert de port in /usr/home/example/ports en installeert alles in /usr/local.

```
# make PREFIX=/usr/home/example/local install
```

Het voorbeeld hierboven compileert in /usr/ports en installeert in /usr/home/example/local.

```
# make WRKDIRPREFIX=../ports PREFIX=../local install
```

Het voorbeeld hierboven combineert de twee instellingen. Het gaat te ver om dit volledig in het handboek te beschrijven, maar hier krijgt de lezer een idee van de mogelijkheden.

Het is ook mogelijk de bovenstaande variabelen als deel van de omgeving in te stellen. In de hulppagina's van de gebruikte shell staat hoe dat mogelijk is.

Omgaan met imake

Er zijn ports die imake gebruiken (een onderdeel van het X Window systeem) die niet goed werken met PREFIX en erop staan te installeren in /usr/X11R6. Er zijn ook een aantal Perl ports die PREFIX negeren en in de Perl hiërarchie installeren. Deze ports op de PREFIX locatie laten installeren is meestal erg moeilijk of onmogelijk.

Ports herconfigureren

Tijdens het bouwen van bepaalde ports kan er een menu dat op ncurses is gebaseerd verschijnen waaruit u bepaalde bouwopties kunt selecteren. Het is niet ongebruikelijk dat gebruikers dit menu opnieuw willen bezoeken om deze opties toe te voegen, te verwijderen, of te veranderen nadat een port is gebouwd. Er zijn vele manieren om dit te doen. Eén optie is om naar de map waarin de port staat te gaan en `make config` te typen, wat eenvoudigweg het menu opnieuw toont met daarin de zelfde opties geselecteerd. Een andere optie is om `make showconfig` te gebruiken, wat alle instelopties voor de port aan u laat zien. Nog een andere optie is om `make rmconfig` uit te voeren wat alle geselecteerde opties zal verwijderen en u toestaat opnieuw te beginnen. Al deze opties, en anderen, worden zeer gedetailleerd uitgelegd in de hulppagina voor ports(7).

4.5.4. Geïnstalleerde ports verwijderen

Nu u weet hoe ports te installeren, zult u zich waarschijnlijk afvragen hoe ze te verwijderen, in het geval dat u er een installeert en later besluit dat u de verkeerde port heeft geïnstalleerd. We zullen ons vorige voorbeeld (`lsof`) verwijderen. Ports worden op precies dezelfde manier verwijderd als pakketten met het commando pkg_delete(1) (zoals beschreven in het onderdeel Pakketten):

```
# pkg_delete lsof-4.57
```

4.5.5. Ports bijwerken

Stel als eerste een lijst samen met ports waarvoor een nieuwere versie beschikbaar is in de Portscollectie met het commando pkg_version(1):

```
# pkg_version -v
```

/usr/ports/UPDATING

Als de Portscollectie eenmaal is bijgewerkt vóór het bijwerken van ports, is het verstandig het bestand /usr/ports/UPDATING te raadplegen. In dat bestand staan aanwijzingen en wijzigingen voor gebruikers die van belang zijn bij het bijwerken van ports, zoals het veranderen van bestandsformaten, veranderen van de locatie van configuratie bestanden, en andere incompatibiliteiten met voorgaande versies.

Als UPDATING tegenstrijdig is met wat hier beschreven is, moet men UPDATING als waar beschouwen.

Ports bijwerken met portupgrade

Het hulpprogramma portupgrade is ontworpen om geïnstalleerde ports eenvoudig bij te werken. Het is beschikbaar via de port ports-mgmt/portupgrade. Installeer het net als iedere andere port met het commando `make install clean`:

```
# cd /usr/ports/ports-mgmt/portupgrade
```

```
# make install clean
```

Scan de lijst met geïnstalleerde ports met het commando `pkgdb -F` en corrigeer alle gerapporteerde inconsistenties. Het is verstandig dit regelmatig te doen, voor iedere keer bijwerken.

Door het draaien van `portupgrade -a` zal portupgrade beginnen met het bijwerken van alle geïnstalleerde ports op een systeem waarvoor een nieuwere versie beschikbaar is. Met de vlag `-i` is het mogelijk in te stellen dat voor iedere bij te werken port om bevestiging wordt gevraagd.

```
# portupgrade -ai
```

Gebruik om alleen een specifieke applicatie bij te werken en niet alle beschikbare ports `portupgrade pkgname`. Gebruik de vlag `-R` om portupgrade eerst alle ports bij te laten werken die voor een bij te werken toepassing benodigd zijn.

```
# portupgrade -R firefox
```

Gebruik de vlag `-P` om bij installatie van pakketten in plaats van ports gebruik te maken. Met deze optie zoekt portupgrade in de lokale mappen uit `PKG_PATH` of haalt de pakketten via het netwerk op als ze lokaal niet worden aangetroffen. Als een pakket niet lokaal en niet via het netwerk wordt gevonden, dan gebruikt portupgrade ports. Om het gebruik van ports te voorkomen kan gebruik gemaakt worden van de optie `-PP`:

```
# portupgrade -PP gnome2
```

Om alleen de distributiebestanden op te halen (of pakketten als `-P` is opgegeven), zonder bouwen of installeren, is `-F` beschikbaar. Meer informatie staat in portupgrade(1).

Ports bijwerken met portmaster

portmaster is nog een gereedschap voor het bijwerken van geïnstalleerde ports. portmaster was ontworpen om gebruik te maken van de gereedschappen die in het "basis" systeem te vinden zijn (het hangt niet af andere ports) en het gebruikt de informatie in /var/db/pkg om te bepalen welke ports bij te werken. Het is beschikbaar via de port ports-mgmt/portmaster:

```
# cd /usr/ports/ports-mgmt/portmaster
# make install clean
```

portmaster verdeelt ports in vier categoriën:

- Wortelpoorten (geen afhankelijkheden, wordt niet van afgehangen)
- Stampoorten (geen afhankelijkheden, wordt van afgehangen)
- Takpoorten (hebben afhankelijkheden, wordt van afgehangen)
- Bladpoorten (hebben afhankelijkheden, wordt niet van afgehangen)

U kunt de optie -L gebruiken om alle geïnstalleerde ports tonen en naar updates te zoeken:

```
# portmaster -L
===>>> Root ports (No dependencies, not depended on)
===>>> ispell-3.2.06_18
===>>> screen-4.0.3
        ===>>> New version available: screen-4.0.3_1
===>>> tcpflow-0.21_1
===>>> 7 root ports
...
===>>> Branch ports (Have dependencies, are depended on)
===>>> apache-2.2.3
        ===>>> New version available: apache-2.2.8
...
===>>> Leaf ports (Have dependencies, not depended on)
===>>> automake-1.9.6_2
===>>> bash-3.1.17
        ===>>> New version available: bash-3.2.33
...
===>>> 32 leaf ports

===>>> 137 total installed ports
        ===>>> 83 have new versions available
```

Alle geïnstalleerde ports kunnen met dit eenvoudige commando worden bijgewerkt:

```
# portmaster -a
```



Standaard maakt portmaster een back-up-pakket aan voordat het een bestaande port verwijderd. Als de installatie van de nieuwe versie succesvol is, zal portmaster de reservekopie verwijderen. Het gebruik van -b zal portmaster instrueren om de reservekopie niet automatisch te verwijderen. Het toevoegen van de optie -i zal portmaster in interactieve modus opstarten, en u vragen voordat het elke port bijwerkt.

Als u fouten tegenkomt tijdens het bijwerkproces, kunt u de optie -f gebruiken om alle ports bij te werken/te herbouwen:

```
# portmaster -af
```

U kunt portmaster ook gebruiken om nieuwe ports op het systeem te installeren, en alle afhankelijkheden bijwerken voordat de nieuwe port gebouwd en geïnstalleerd wordt:

```
# portmaster shells/bash
```

Bekijk portmaster(8) voor meer informatie.

4.5.6. Ports en schijfruimte

Werken met de Portscollectie kan in de loop der tijd veel schijfruimte gebruiken. Na het bouwen en installeren van software uit de ports, is het van belang altijd de tijdelijke mappen work op te ruimen met het commando `make clean`. De complete Portscollectie kan geschoond worden met het volgende commando:

```
# portsclean -C
```

In de loop der tijd komen ook veel oude bestanden met broncode in de map distfiles te staan. Die kunnen handmatig verwijderd worden of met het volgende commando dat alle distributiebestanden waarnaar in de huidige ports geen verwijzingen meer staan verwijdert:

```
# portsclean -D
```

Of om alle distributiebestanden te verwijderen waardoor momenteel door geen één geïnstalleerde port op uw systeem wordt verwezen:

```
# portsclean -DD
```



Het hulpprogramma `portsclean` is onderdeel van de suite portupgrade.

Vergeet niet ports die niet langer gebruikt worden te verwijderen. Een handig hulpmiddel hiervoor kan de port ports-mgmt/pkg_cutleaves zijn.

4.6. Activiteiten na het installeren

Na het installeren van een nieuwe applicatie is het meestal verstandig om de documentatie te lezen die bij een applicatie zit, bestanden met instellingen die vereist zijn aan te passen, ervoor te zorgen dat de applicatie start na het opstarten (als het een daemon is), enzovoort.

De exacte stappen om een applicatie in te stellen zijn natuurlijk voor iedere applicatie anders. Maar als er net een nieuwe applicatie is geïnstalleerd en het is niet vanzelfsprekend hoe verder te gaan, dan kunnen de volgende tips helpen:

- Met pkg_info(1) kan uitgevonden worden welke bestanden geïnstalleerd zijn en waar. Om bijvoorbeeld uit te vinden welke bestanden door FooPackage versie 1.0.0 zijn geïnstalleerd:

  ```
  # pkg_info -L foopackage-1.0.0 | less
  ```

  Bestanden in mapnamen met man/ zijn hulppagina's, etc/ bevat bestanden met instellingen en doc/ bevat uitgebreidere documentatie.

Als niet helemaal duidelijk is welke versie van het programma is geïnstalleerd, kan een commando als volgt gebruikt worden:

```
# pkg_info | grep -i foopackage
```

Hiermee worden alle pakketten getoond waar *foopackage* in de pakketnaam voorkomt.

- Als de hulppagina's zijn gevonden, kunnen die bekeken worden met man(1). Zo kan er ook in de bestanden met voorbeeldinstellingen gekeken worden en naar aanvullende documentatie, als die is bijgeleverd.
- Als er een website is voor de applicatie staat daar vaak ook aanvullende documentatie, veelgestelde vragen, enzovoort. Als het webadres niet bekend is, kan dat nog staan in de uitvoer van het volgende commando:

```
# pkg_info foopackage-1.0.0
```

Als er een regel met `WWW:` in staat, is dat de URL naar de website voor de applicatie.

- Ports die na het opstarten moeten starten (zoals Internet diensten) hebben meestal een voorbeeldscript in /usr/local/etc/rc.d. Dit script kan bekeken, aangepast en hernoemd worden waar nodig. Meer informatie staat in Diensten Starten.

4.7. Omgaan met kapotte ports

Als een port niet werkt, zijn er een aantal mogelijke manieren om verder te komen:

1. Zoek uit of er een oplossing voor de port staat te wachten in de Problem Report database. Als dat zo is kan wellicht de voorgestelde reparatie gebruikt worden.
2. Vraag de beheerder van de port om hulp. Voor het emailadres van de beheerder kan `make maintainer` getypt worden of het kan in de Makefile staan. Zet in de mail in ieder geval de naam en versie van de port (de regel met `$FreeBSD:` in de Makefile) en de uitvoer tot en met de foutmelding.



Sommige ports worden niet beheerd door een individu maar in plaats daarvan door een mailinglijst. Veel, maar niet alle, van deze adressen zien eruit als freebsd-lijstnaam@FreeBSD.org. Houd hier alstublieft rekening mee bij het formuleren van vragen.

In het bijzonder worden ports die geregistreerd staan als onderhouden door ports@FreeBSD.org helemaal niet onderhouden. Reparaties en ondersteuning, als dat al beschikbaar is, komt vanuit de gemeenschap die is geabonneerd op die mailinglijst. Meer vrijwilligers zijn altijd nodig!

Als er geen antwoord komt, stuur dan met send-pr(1) een foutrapport in. Zie Writing FreeBSD Problem Reports).

3. Repareren! In het Handboek voor de Porter is gedetailleerde informatie te vinden over de infrastructuur van de "Ports", zodat een kapotte port gemaakt kan worden of er zelfs een nieuwe port ingestuurd kan worden.
4. Zoek een pakket van een FTP site in de buurt. De "master" pakketcollectie staat op `ftp.FreeBSD.org` in de map pakketten, maar het is van belang dat er *eerst* in de buurt wordt gekeken! Dat het pakket werkt is waarschijnlijker dan wanneer uit de broncode wordt gecompileerd en het is nog sneller ook. Een pakket kan met pkg_add(1) geïnstalleerd worden.

Hoofdstuk 5. Het X Window systeem

5.1. Overzicht

FreeBSD gebruikt X11 om gebruikers een krachtige grafische gebruikersschil te bieden. X11 is een vrij beschikbare versie van het X Window System dat geïmplementeerd is in XorgXFree86™ (en andere softwarepakketten die hier niet worden besproken). De standaard en officiele smaak van X11 in FreeBSD is Xorg, de X11-server die is ontwikkeld door de X.Org Foundation onder een licentie die veel lijkt op degene die door FreeBSD wordt gebruikt.

Meer informatie over de videohardware die X11 ondersteunt kan gevonden worden op de Xorg website.

Na het lezen van dit hoofdstuk weet de lezer:

- Wat de componenten van het X Window systeem zijn en hoe zij samenwerken.
- Hoe X11 geïnstalleerd en ingesteld kan worden.
- Hoe verschillende window managers geïnstalleerd en gebruikt kunnen worden.
- Hoe TrueType® lettertypen in X11 te gebruiken.
- Hoe het systeem ingesteld moet worden voor grafisch aanmelden (XDM).

Aangeraden voorkennis:

- Hoe extra software van derden te installeren (Applicaties installeren. pakketten en ports).

5.2. X begrijpen

X voor de eerste keer gebruiken kan een hele schok zijn voor mensen die gewend zijn aan andere grafische omgevingen, zoals Microsoft® Windows® of Mac OS®.

Het is niet noodzakelijk om alle details te kennen over de X componenten en hoe zij samenwerken, maar enige basiskennis draagt wel bij aan krachtiger gebruik kunnen maken van X.

5.2.1. Waarom X?

X is niet het eerste windows systeem dat geschreven is voor UNIX®, maar wel het meest populaire. Het oorspronkelijke X ontwikkelteam werkte eerst aan een ander window systeem. De naam van dat systeem was "W" (van "Window"). X was gewoon de volgende letter in het alfabet.

X kan gewoon "X", "X Window systeem", "X11" of nog anders genoemd worden. X11 "X Windows" noemen kan door sommigen als een belediging opgevat worden. X(7) kan hierover wat licht laten schijnen.

5.2.2. Het X client/server model

X is vanaf het begin aan ontworpen om netwerk-centraal te zijn en gebruikt een "client-server" model.

In het X model draait de "X server" op de computer waar het toetsenbord, beeldscherm en muis aan vast zit. De server is verantwoordelijk voor het regelen van beeldinformatie, verwerken van invoer van toetsenbord en muis, en andere invoer- of uitvoerapparaten (i.e., een "tablet" kan als invoerapparaat worden gebruikt, en een videoprojector kan een alternatief uitvoerapparaat zijn). Iedere X applicatie (zoals XTerm of Firefox) is een "cliënt". Een cliënt stuurt berichten naar de server zoals "teken een venster op deze coördinaten" en de server stuurt berichten terug zoals "de gebruiker heeft op de OK knop gedrukt".

Thuis of in kleine bedrijven draaien zowel de X server als de X clients op dezelfde machine. Het is heel goed mogelijk dat de X server op een minder krachtige desktop computer draait en de X applicaties (de clients) op een, zeg maar, dure krachtige machine van het bedrijf. Hier vindt de communicatie tussen de X client en server plaats over het netwerk.

Dit verwart sommige mensen, omdat de X terminologie geheel omgekeerd is aan wat ze verwachten. Dat is namelijk dat de "X server" de grote krachtige machine aan het eind van de gang is en de "X client" de machine op hun bureau is.



De X server is de machine met het beeldscherm en het toetsenbord en de X clients zijn de programma's die de vensters tonen.

Het protocol vereist niet dat de clients en servers hetzelfde besturingssysteem moeten draaien of hetzelfde soort computer moeten zijn. Het is heel goed mogelijk om X server op een Microsoft® Windows® of Apple's Mac OS® te draaien en er zijn verschillende gratis en commerciële applicaties die dat doen.

5.2.3. De window manager

De filosofie van het X ontwerp lijkt veel op die van UNIX®: "gereedschappen, geen beleid". Dit houdt in dat X niet bepaalt hoe een taak volbracht moet worden. In plaats daarvan worden gereedschappen geleverd aan de gebruiker die verantwoordelijk is voor het juiste gebruik hiervan.

Deze filosofie verbreedt zich door X niet te laten bepalen hoe vensters er moeten uitzien op het scherm, hoe ze verplaatst moeten worden met de muis, welke toetsaanslagen gebruikt moeten worden om te schakelen tussen vensters (bijvoorbeeld `Alt` + `Tab` in het geval van Microsoft® Windows®), hoe de titelbalken eruit moeten zien, of ze wel of niet sluitknoppen moeten hebben, enzovoort.

In plaats daarvan delegeert X deze verantwoordelijkheid aan een applicatie die "Window Manager" heet. Er zijn tientallen window managers beschikbaar voor X. Elk van deze window managers heeft een eigen voorkomen en werking. Er zijn window managers met "virtual desktops" of met eigen toetscombinaties om de desktop te beheren; of hebben een "Start" knop of iets gelijksoortig. Sommige gebruiken "thema's" die uiterlijk en beleving compleet veranderen door een nieuw thema te kiezen. Window managers zijn te vinden in de categorie x11-wm van de Portscollectie.

De KDE en GNOME desktop omgevingen hebben hun eigen window managers die in het bureaublad zijn geïntegreerd.

Iedere windows manager heeft zijn eigen manier van instellen. Sommige werken met handgetypte bestanden, anderen beschikken over grafische gereedschappen voor de meeste instellingen. Er is er

minstens één (Sawfish) waarvan het instellingenbestand is geschreven in een dialect van de taal Lisp.

Focusbeleid

De window manager is ook verantwoordelijk voor het "focusbeleid" van de muis. Ieder window geörienteerd systeem heeft een manier nodig om te bepalen welk venster actief is, toetsaanslagen ontvangt en daarbij zichtbaar aangeeft welk venster actief is.

Een bekend focus beleid heet "click-to-focus". Dit model wordt gebruikt door Microsoft® Windows®, waarbij een venster actief wordt door er met de muis op te klikken.

X ondersteunt geen specifiek focusbeleid. In plaats daarvan bepaalt de window manager op welk venster, op welk moment, de focus ligt. Een aantal window managers ondersteunen verschillende focusmethoden. Ze ondersteunen allemaal "click to focus" en de meerderheid ondersteunt ook nog andere.

De meest populaire zijn:



focus-volgt-muis (focus-follows-mouse)

Het venster dat onder de muis zit is het venster waarop de focus ligt. Dit hoeft niet het venster te zijn dat bovenop alle andere vensters ligt. De focus verandert door te wijzen naar een ander venster. Het is niet nodig om er ook nog eens op te klikken.

slordige-focus (sloppy-focus)

Dit beleid is een kleine uitbreiding op focus-follows-mouse. Indien bij focus-follows-mouse de muis over het root venster (of de achtergrond) gaat, ligt op geen enkel venster de focus en gaan alle toetsaanslagen verloren. Bij sloppy-focus, verandert de focus alleen als de muis in een nieuw venster komt en niet als het huidige venster wordt verlaten.

klik-voor-focus (click-to-focus)

Het actieve venster wordt geselecteerd door erop te klikken. Het venster wordt dan "opgetild" en verschijnt dan voor alle andere vensters. Alle toetsaanslagen worden nu naar dit venster gestuurd, zelfs als de cursor naar een ander scherm wordt verplaatst.

Veel window managers ondersteunen andere soorten of variaties op de bovenstaande typen muisbeleid. Hierover staat meestal meer in de documentatie van de betreffende window manager.

5.2.4. Widgets

De X aanpak door gereedschappen te leveren en niets af te dwingen breidt zich uit naar de widgets die in elk applicatievenster te zien zijn.

"Widget" is een term voor alle dingen van de gebruikersinterface waarop geklikt kan worden of een

andere actie mee uitgevoerd kan worden: knoppen, vinkvakjes, iconen, lijsten en ga zo maar door. Microsoft® Windows® noemt ze "controls".

Microsoft® Windows® en Apple's Mac OS® hebben beide een erg strikt widgetbeleid. Van de applicatieontwikkelaars wordt verwacht dat hun applicaties eenduidig zijn wat betreft uiterlijk en beleving. Bij X is ervoor gekozen geen grafische stijl of widgets te verplichten.

X applicaties hebben dus niet allemaal hetzelfde uiterlijk. Er zijn populaire widgetsets en variaties, inclusief Qt, gebruikt door KDE, of GTK+ van het GNOME project. Vanuit dit oogpunt lijkt het enigszins op de UNIX® desktop, wat het makkelijker maakt voor de beginnende gebruiker.

5.3. X11 installeren

Xorg is de X11-implementatie voor FreeBSD. Xorg is de X11 server van de open source implementatie die is uitgebracht door de X.Org Foundation. Xorg is gebaseerd op de code van XFree86™ 4.4RC2 en X11R6.6. De versie van Xorg die momenteel beschikbaar is in de FreeBSD Portscollectie is 7.7.

Om Xorg vanuit de Portscollectie te bouwen en te installeren:

```
# cd /usr/ports/x11/xorg
# make install clean
```



Om Xorg compleet te bouwen is tenminste 4 GB vrije schijfruimte nodig.

X11 kan ook als pakket geïnstalleerd worden doordat er binaire pakketten beschikbaar zijn voor pkg_add(1). Als hiervoor de optie "remote fetching" van pkg_add(1) wordt gebruikt, dan moet het versienummer verwijderd worden. pkg_add(1) haalt automatisch de laatste versie van het programma op.

Om het pakket voor Xorg op te halen en te installeren:

```
# pkg_add -r xorg
```



Het voorbeeld hierboven installeert de complete X11 distributie inclusief de servers, clients, lettertypen enz. Er zijn ook afzonderlijke pakketten en ports beschikbaar voor verschillende delen van X11.

Om een minimale X11-distributie te installeren kunt u als alternatief x11/xorg-minimal installeren.

De rest van dit hoofdstuk licht toe hoe X11 wordt ingesteld en hoe een productieve desktopomgeving gebouwd kan worden.

5.4. X11 instellen

5.4.1. Voorbereiding

In de meeste gevallen configureert X11 zichzelf. Voor degenen met oudere of ongebruikelijke apparatuur kan het nuttig zijn om informatie over de hardware te verzamelen voordat er met de configuratie wordt begonnen.

- Monitor synchronisatiefrequenties
- Chipset van de videokaart
- Geheugen van de videokaart

De schermresolutie en ververssnelheid worden bepaald door de horizontale en verticale synchronisatiefrequenties. Bijna alle monitoren ondersteunen het automatisch elektronisch detecteren van deze waardes. Sommige monitoren geven deze waardes niet, dus moeten de specificaties worden bepaald uit de geprinte handleiding of van de website van de fabrikant.

De chipset van de videokaart wordt ook automatisch gedetecteerd en gebruikt om het juiste videostuurprogramma te selecteren. Het kan handig voor de gebruiker zijn om te weten welke chipset is geïnstalleerd wanneer de automatische detectie niet het gewenste resultaat geeft.

Het geheugen van de videokaart bepaalt de maximale resolutie en de kleurdiepte die afgebeeld kunnen worden.

5.4.2. X11 instellen

Xorg gebruikt HAL om toetsenborden en muizen automatisch te detecteren. De ports sysutils/hal en devel/dbus worden als afhankelijkheden van x11/xorg geïnstalleerd, maar moeten met de volgende regels in het bestand /etc/rc.conf worden aangezet:

```
hald_enable="YES"
dbus_enable="YES"
```

Deze diensten dienen gestart te worden (ofwel handmatig of door opnieuw op te starten) voordat er verder wordt gegaan met de configuratie of gebruik van Xorg.

Xorg werkt vaak zonder enige verdere configuratie door het volgende op de prompt te typen:

```
% startx
```

De automatische configuratie kan met sommige hardware mislukken, of het kan dingen anders instellen dan gewenst is. In deze gevallen is handmatige configuratie nodig.



Bureaubladomgevingen als GNOME, KDE, of Xfce hebben gereedschappen waarmee de gebruiker eenvoudig de schermparameters zoals de resolutie kan instellen. Dus als de standaardconfiguratie niet acceptabel is en u van plan bent

om een bureaubladomgeving te installeren kunt u gewoon doorgaan met de installatie van de bureaubladomgeving en het juiste scherminstelgereedschap gebruiken.

Het instellen van X11 bestaat uit meerdere stappen. De eerste stap is het bouwen van een instellingenbestand. Dit kan als de supergebruiker met:

```
# Xorg -configure
```

Dit genereert een kaal X11-instellingenbestand in de map /root met de naam xorg.conf.new. Feitelijk wordt bepaald waar de map staat door hoe er superuser rechten zijn verkregen. $HOME is anders bij gebruik van su(1) of bij direct aanmelden. Het X11 programma probeert dan de grafische hardware te detecteren en schrijft een instellingenbestand dat de juiste stuurprogramma's laadt voor de gevonden hardware van het systeem.

De volgende stap is het testen van de bestaande instellingen om te controleren of Xorg met de grafische kaart van het doelsysteem kan werken. Typ:

```
# Xorg -config xorg.conf.new -retro
```

Als er een zwart/grijs rooster en een X muis cursor verschijnen was de instelling succesvol. Om de test te stoppen dient naar de virtuele console waarmee de test werd gestart overgeschakeld te worden door op `Ctrl` + `Alt` + `Fn` (`F1` voor de eerste virtuele console) en `Ctrl` + `C` te drukken.

De toetsencombinatie `Ctrl` + `Alt` + `Backspace` kan ook gebruikt worden om uit Xorg te breken. Om het aan te zetten, kunt u òfwel het volgende commando uitvoeren vanaf elke X-terminal-emulator:

```
% setxkbmap -option terminate:ctrl_alt_bksp
```

òf een instellingenbestand voor het toetsenbord genaamd x11-input.fdi voor hald aanmaken en het in de map /usr/local/etc/hal/fdi/policy opslaan. Dit bestand dient het volgende te bevatten:



```
<?xml version="1.0" encoding="utf-8"?>
<deviceinfo version="0.2">
  <device>
    <match key="info.capabilities" contains="input.keyboard">
      <merge key="input.x11_options.XkbOptions"
type="string">terminate:ctrl_alt_bksp</merge>
    </match>
  </device>
</deviceinfo>
```

U moet uw machine opnieuw opstarten om hald te forceren om dit bestand te

lezen.

De volgende regel dient ook aan de sectie `ServerLayout` of `ServerFlags` van xorg.conf.new te worden toegevoegd:

```
Option "DontZap"    "off"
```

Als de muis niet werkt, dan moet deze eerst ingesteld worden. Zie Muisinstellingen in het FreeBSD installatiehoofdstuk. In recente versies van Xorg worden de secties `InputDevice` in xorg.conf genegeerd ten voorkeur van de automatisch gedetecteerde apparaten. Voeg de volgende regel aan de sectie `ServerLayout` of `ServerFlags` van dit bestand toe om het oude gedrag te herstellen:

```
Option "AutoAddDevices" "false"
```

Invoerapparaten kunnen dan zoals in vorige versies worden geconfigureerd, tezamen met eventuele andere benodigde opties (bijvoorbeeld omschakelen van toetsenbordindeling).

Zoals al eerder is uitgelegd zal de daemon hald standaard automatisch uw toetsenbord detecteren. Het kan zijn dat de indeling of het model van uw toetsenbord niet juist zijn. Bureaubladomgevingen zoals GNOME, KDE of Xfce bieden gereedschappen om het toetsenbord in te stellen. Het is echter mogelijk om de eigenschappen direct in te stellen met behulp van het gereedschap setxkbmap(1) of met een configuratieregel van hald.

Als men bijvoorbeeld een PC-toetsenbord met 102 toetsen met een Franse indeling wilt gebruiken, dienen we een instellingenbestand voor het toestenbord voor hald aan te maken genaamd x11-input.fdi en het op te slaan in de map /usr/local/etc/hal/fdi/policy. Het dient de volgende regels te bevatten:

```
<?xml version="1.0" encoding="utf-8"?>
<deviceinfo version="0.2">
  <device>
    <match key="info.capabilities" contains="input.keyboard">
      <merge key="input.x11_options.XkbModel"
type="string">pc102</merge>
      <merge key="input.x11_options.XkbLayout" type="string">fr</merge>
    </match>
  </device>
<deviceinfo>
```

Als dit bestand al bestaat, kunt u de regels betreffende de configuratie van het toetsenbord kopiëren en aan uw bestand toevoegen.

U dient uw machine opnieuw op te starten om hald te forceren om dit bestand te lezen.

> Het is mogelijk om hetzelfde te bereiken vanaf een X-terminal of een script met dit commando:
>
> ```
> % setxkbmap -model pc102 -layout fr
> ```
>
> Het bestand /usr/local/shared/X11/xkb/rules/base.lst noemt de beschikbare toetsenborden, indelingen en opties.

Het bestand xorg.conf.new kan nu naar wens worden aangepast. Open het bestand in een tesktverwerker zoals emacs(1) of ee(1). Indien de monitor een ouder of ongebruikelijk model is dat geen automatische detectie van de synchronisatiefrequenties ondersteunt, dan kunnen deze instellingen worden toegevoegd aan xorg.conf.new in de sectie "Monitor":

```
Section "Monitor"
        Identifier   "Monitor0"
        VendorName   "Monitor Vendor"
        ModelName    "Monitor Model"
        HorizSync    30-107
        VertRefresh  48-120
EndSection
```

De meeste monitoren ondersteunen de automatische detectie van de synchronisatiefrequentie, wat het handmatig invoeren van deze waardes overbodig maakt. Voor de enkele monitoren die geen automatische detectie ondersteunen, dienen om mogelijke schade te voorkomen alleen waardes die door de fabrikant zijn opgegeven te worden ingevoerd.

X kan DPMS (Energy Star) eigenschappen gebruiken bij monitoren die dit ondersteunen. xset(1) regelt de timeouts en kan de statussen standby, suspend of uit forceren. Om DPMS eigenschappen voor een monitor te activeren, moet de volgende regel toegevoegd worden aan de monitor sectie:

```
Option       "DPMS"
```

Als het instellingenbestand xorg.conf.new toch open staat in de editor dan kan ook meteen de gewenste standaardresolutie en kleurdiepte gekozen worden. Dit staat in het onderdeel "Screen":

```
Section "Screen"
        Identifier "Screen0"
        Device     "Card0"
        Monitor    "Monitor0"
        DefaultDepth 24
        SubSection "Display"
                Viewport  0 0
                Depth     24
                Modes     "1024x768"
        EndSubSection
```

```
EndSection
```

Het sleutelwoord `DefaultDepth` beschrijft de kleurdiepte die standaard wordt gebruikt. Met de commandoregeloptie `-depth` van Xorg(1) kan dit overschreven worden. Het sleutelwoord `Modes` beschrijft de resolutie waarmee gewerkt wordt bij de opgegeven kleurdiepte. Alleen VESA standaarden die door de grafische kaart van het systeem worden gedefinieerd worden ondersteund. In het voorbeeld hierboven is de standaard kleurdiepte 24 bits per pixel. Bij deze kleurdiepte is de toegestane resolutie 1024 bij 768 pixels.



Bij het oplossen van problemen zijn de logboekbestanden van X11 vaak een goede hulp. Ze bevatten informatie voor ieder apparaat waar de X11 server verbinding mee maakt. Namen van Xorg logboekbestanden hebben de vorm /var/log/Xorg.0.log. De precieze naam van een logboekbestand van variëren van Xorg.0.log tot Xorg.8.log enzovoort.

Als alles is ingesteld, moet het instellingenbestand op een plaats gezet worden waar Xorg(1) het kan vinden. Dit is meestal /etc/X11/xorg.conf of /usr/local/etc/X11/xorg.conf:

```
# cp xorg.conf.new /etc/X11/xorg.conf
```

Het instellen van X11 is nu gereed. Xorg gestart worden met startx(1). De X11-server kan ook gestart worden met behulp van xdm(1).

5.4.3. Bijzondere instellingen

Instellen met de Intel® `i810` grafische chipset

Instellen met Intel® i810 geïntegreerde chipsets vereist de agpgart AGP programmeerinterface voor X11 om de kaart aan te sturen. Zie de agp(4) handleiding voor meer informatie.

Hierdoor wordt het instellen van de hardware net als ieder andere grafische kaart. Bij systemen die zonder agp(4) stuurprogramma gecompileerd zijn slaagt het laden van module met kldload(8) niet. Het stuurprogramma moet in de kernel geladen zijn tijdens het opstarten door te compileren of door /boot/loader.conf te gebruiken.

Een Breedbeeld Flatpanel toevoegen

Deze sectie gaat uit van wat diepere configuratiekennis. Als pogingen om de bovenstaande standaard instelgereedschappen niet tot een werkende configuratie leidden, dan is er genoeg informatie in de logbestanden om de opstelling aan de praat te krijgen. Het gebruik van een tekstverwerker zal nodig zijn.

Huidige breedbeeldformaten (zoals WSXGA, WSXGA+, WUXGA, WXGA en WXGA+) ondersteunen 16:10 en 10:9 formaten of aspectverhoudingen die problematisch kunnen zijn. Voorbeelden van enkele veelvoorkomende schermresoluties voor 16:10 aspectverhoudingen zijn:

- 2560x1600

- 1920x1200
- 1680x1050
- 1440x900
- 1280x800

Op een gegeven moment zal het toevoegen van een van deze resoluties net zo eenvoudig zijn als een mogelijke `Mode` in het `Section "Screen"`:

```
Section "Screen"
Identifier "Screen 0"
Device     "Card0"
Monitor    "Monitor0"
DefaultDepth 24
SubSection "Display"
    Viewport 0 0
    Depth    24
    Modes    "1680x1050"
EndSubSection
EndSection
```

Xorg is slim genoeg om de resolutie-informatie via I2C/DDC-informtie uit het flatpanel te onttrekken zodat het weet wat de monitor aan kan wat betreft frequenties en resoluties.

Als die `ModeLines` niet bestaan in de stuurprogramma's, dient men Xorg een kleine hint te geven. Met behulp van /var/log/Xorg.0.log kan men genoeg informatie onttrekken om handmatig een werkende `ModeLine` aan te maken. Kijk naar informatie die op deze lijkt:

```
(II) MGA(0): Supported additional Video Mode:
(II) MGA(0): clock: 146.2 MHz   Image Size:  433 x 271 mm
(II) MGA(0): h_active: 1680  h_sync: 1784  h_sync_end 1960 h_blank_end 2240 h_border:
0
(II) MGA(0): v_active: 1050  v_sync: 1053  v_sync_end 1059 v_blanking: 1089 v_border:
0
(II) MGA(0): Ranges: V min: 48  V max: 85 Hz, H min: 30  H max: 94 kHz, PixClock max
170 MHz
```

Deze informatie wordt EDID-informatie genoemd. Hiervan een `ModeLine` maken is gewoon een kwestie van de nummers in de juiste volgorde zetten:

```
ModeLine <name> <clock> <4 horiz. timings> <4 vert. timings>
```

Dus de `ModeLine` in `Section "Monitor"` zou er voor dit voorbeeld uitzien als:

```
Section "Monitor"
Identifier       "Monitor1"
```

```
VendorName      "GroteNaam"
ModelName       "BesteModel"
ModeLine        "1680x1050" 146.2 1680 1784 1960 2240 1050 1053 1059 1089
Option          "DPMS"
EndSection
```

Na het voltooien van deze eenvoudige stappen, zou X moeten starten op uw nieuwe breedbeeldmonitor.

5.5. Lettertypen gebruiken in X11

5.5.1. Type1 lettertypen

De standaard lettertypen van X11 zijn allerminst ideaal voor het typische bureaubladprogramma. Grote presentatielettertypen zien er hoekig en onprofessioneel uit en kleine lettertypen zijn bijna onleesbaar. Er zijn diverse gratis, kwalitatief goede Type1 (PostScript®) lettertypen die meteen gebruikt kunnen worden met X11. De URW lettertypecollectie (x11-fonts/urwfonts) heeft bijvoorbeeld hoge kwaliteit versies van standaard Type1 lettertypen (Times Roman™, Helvetica™, Palatino™ en anderen). De Freefonts collectie (x11-fonts/freefonts) heeft nog meer lettertypen, maar de meesten ervan zijn bedoeld om in grafische software als Gimp gebruikt te worden en zijn niet compleet genoeg om als schermlettertypen te gebruiken. Daarbij kan X11 zonder veel moeite ingesteld worden om TrueType® lettertypen te gebruiken. Meer informatie staat in X(7) of de paragraaf over TrueType® Lettertypen.

Om de bovenstaande Type1 lettertypecollectie van de Portscollectie te installeren:

```
# cd /usr/ports/x11-fonts/urwfonts
# make install clean
```

Dat geldt ook voor de freefont en andere collecties. Om de X server te vertellen dat deze lettertypen bestaan, dient de volgende regel toegevoegd te worden aan het instellingenbestand van de X server (/etc/X11/xorg.conf):

```
FontPath "/usr/local/lib/X11/fonts/URW/"
```

Ook kan op de commando regel in de X sessie het volgende gestart worden:

```
% xset fp+ /usr/local/lib/X11/fonts/URW
% xset fp rehash
```

Dit werkt wel, maar zodra de X sessie wordt afgesloten is het weer verdwenen tenzij het is toegevoegd aan het opstartbestand (~/.xinitrc voor een normale `startx` sessie of ~/.xsession als er wordt aangemeld met een grafische aanmeldmanager als XDM). Een derde manier is het gebruik van het nieuwe bestand /usr/local/etc/fonts/local.conf: zie hiervoor de paragraaf over Anti-aliasing.

5.5.2. TrueType® lettertypen

Xorg heeft ingebouwde ondersteuning voor het renderen van TrueType® lettertypen. Er zijn twee verschillende modules die deze functionaliteit activeren. In dit voorbeeld wordt de freetype module gebruikt omdat deze beter werkt met de andere lettertypen die back-ends renderen. Om de freetype module te activeren dient de volgende regel toegevoegd te worden aan het onderdeel `"Module"` van /etc/X11/xorg.conf.

```
Load  "freetype"
```

Hierna dient een map voor de TrueType® lettertypen gemaakt te worden (bijvoorbeeld /usr/local/lib/X11/fonts/TrueType) en alle TrueType® lettertypen moeten naar deze map gekopieerd worden. TrueType® lettertypen kunnen niet direct van een Macintosh® gehaald worden. Ze moeten in een UNIX®/MS-DOS®/Windows® formaat zijn voor X11. Zodra de bestanden naar deze map zijn gekopieerd, kan ttmkfdir gestart worden om een fonts.dir bestand te maken zodat de X lettertyperenderer weet waar deze nieuwe bestanden zijn geïnstalleerd. `ttmkfdir` zit in de FreeBSD Portscollectie als x11-fonts/ttmkfdir.

```
# cd /usr/local/lib/X11/fonts/TrueType
# ttmkfdir -o fonts.dir
```

Nu moet de TrueType® map toe aan het lettertypepad toegevoegd worden. Dit gebeurt op dezelfde wijze als boven is beschreven voor Type1 lettertypen:

```
% xset fp+ /usr/local/lib/X11/fonts/TrueType
% xset fp rehash
```

of door een `FontPath` regel toe te voegen aan xorg.conf.

Dat is alles. Nu herkennen Gimp, Apache OpenOffice en alle andere X applicaties de geïnstalleerde TrueType® lettertypen. Extreem kleine lettertypen (zoals hoge resolutie tekst op een webpagina) en extreme grote lettertypen (in StarOffice™) zien er nu veel beter uit.

5.5.3. Antialias lettertypen

Alle lettertypen die X11 in de mappen /usr/local/lib/X11/fonts/ en ~/.fonts/ staan zijn automatisch beschikbaar voor anti-aliasing in applicaties die Xft ondersteunen. De meeste recente applicaties ondersteunen Xft, inclusief KDE, GNOME, en Firefox.

Om te kunnen regelen welke lettertypen gebruik maken van anti-alias of om de eigenschappen van anti-aliasing in te stellen kan /usr/local/etc/fonts/local.conf gemaakt of gewijzigd worden. In dit bestand kunnen speciale eigenschappen van het Xft lettertypesysteem aangepast worden. Deze paragraaf beschrijft wat eenvoudige mogelijkheden. Meer details staan in fonts-conf(5).

Dit bestand moet in het XML formaat opgemaakt worden. Hoofdletters en kleine letters worden onderscheiden en alle tags moeten netjes worden afgesloten. Het bestand begint met de gewone

XML header gevolgd door een DOCTYPE definitie en daarna de `<fontconfig>` tag:

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">
<fontconfig>
```

Zoals al eerder is vermeld zijn alle lettertypen in /usr/local/lib/X11/fonts/ en in ~/.fonts/ al geschikt gemaakt voor Xft applicaties. Als naast deze twee mappen nog een andere lettertypen moeten kunnen bevatten, dan dient een soortgelijke regel als de onderstaande aan /usr/local/etc/fonts/local.conf toegevoegd te worden:

```
<dir>/pad/naar/mijn/fonts</dir>
```

Na het toevoegen van nieuwe lettertypen en zeker nieuwe lettertypemappen dienen de lettertypecaches opnieuw opgebouwd worden met:

```
# fc-cache -f
```

Anti-aliasing maakt randen een beetje wazig wat kleine teksten beter leesbaar maakt en voorkomt "trapvorming" van grote letters. Maar het kan oogkramp veroorzaken als het op normale tekst wordt toegepast. Om lettertypen kleiner dan 14 punten uit te sluiten van anti-aliasing moeten de volgende regels toegevoegd worden:

```
<match target="font">
    <test name="size" compare="less">
        <double>14</double>
    </test>
    <edit name="antialias" mode="assign">
        <bool>false</bool>
    </edit>
</match>
<match target="font">
    <test name="pixelsize" compare="less" qual="any">
        <double>14</double>
    </test>
    <edit mode="assign" name="antialias">
        <bool>false</bool>
    </edit>
</match>
```

Spatiëring voor sommige enkel gespatieerde lettertypen kan ook ongepast zijn bij anti-aliasing. Dit lijkt vooral een probleem te zijn bij KDE. Een mogelijke oplossing hiervoor is het vergroten van de spatiëring van die lettertypen naar 100:

```
<match target="pattern" name="family">
```

```
   <test qual="any" name="family">
       <string>fixed</string>
   </test>
   <edit name="family" mode="assign">
       <string>mono</string>
   </edit>
</match>
<match target="pattern" name="family">
    <test qual="any" name="family">
        <string>console</string>
    </test>
    <edit name="family" mode="assign">
        <string>mono</string>
    </edit>
</match>
```

Het bovenstaande hernoemt de standaardnamen van lettertypen naar "mono"). Voeg daarna het volgende toe:

```
<match target="pattern" name="family">
    <test qual="any" name="family">
        <string>mono</string>
    </test>
    <edit name="spacing" mode="assign">
        <int>100</int>
    </edit>
</match>
```

Bepaalde lettertypen, zoals Helvetica, kunnen problemen hebben met anti-aliasing. Dit uit zich meestal in een lettertype dat verticaal door midden lijkt gesneden. Op zijn ergst kan het applicaties laten crashen. Om dit te voorkomen kan overwogen worden om ook de volgende regels toe te voegen aan local.conf:

```
<match target="pattern" name="family">
    <test qual="any" name="family">
        <string>Helvetica</string>
    </test>
    <edit name="family" mode="assign">
        <string>sans-serif</string>
    </edit>
</match>
```

Als de wijzigingen in local.conf zijn gemaakt dient niet vergeten te worden het bestand te eindigen met de tag `</fontconfig>` tag. Als dit niet gedaan wordt, dan worden de wijzigingen niet gezien.

Als laatste kunnen gebruikers hun eigen instellingen aan een persoonlijk .fonts.conf bestand toevoegen. Om dit te doen moet iedere gebruiker het bestand ~/.fonts.conf maken. Ook dit bestand moet in het XML formaat zijn.

Nog een laatste punt: bij een LCD scherm kan sub-pixel sampling prettig zijn. Eigenlijk zorgt dit er voor dat de (horizontaal gesplitste) rode, groene en blauwe componenten gewijzigd worden om de horizontale resolutie te verbeteren. Het resultaat is geweldig. Voeg hiervoor de volgende regels ergens aan local.conf toe:

```
<match target="font">
    <test qual="all" name="rgba">
        <const>unknown</const>
    </test>
    <edit name="rgba" mode="assign">
        <const>rgb</const>
    </edit>
</match>
```



Afhankelijk van het soort beeldscherm kan `rgb` veranderd moeten worden in `bgr`, `vrgb` of `vbgr`. Experimenteren levert de beste instelling op.

5.6. De X beeldschermmanager

5.6.1. Overzicht

De X beeldschermmanager (XDM) is een optioneel onderdeel van het X Window systeem dat gebruikt wordt voor beheer van aanmeldsessies. Dit is vaak erg handig bij bijvoorbeeld "X Terminals", desktops en grote netwerk beeldschermservers. Omdat het X Window systeem netwerk- en protocolonafhankelijk is, zijn er veel mogelijkheden om X clients en servers op verschillende machines in een netwerk te verbinden. XDM levert een grafische interface waarmee er gekozen kan worden welke beeldschermserver gebruikt moet worden en handelt autorisatie informatie (gebruikersnaam en wachtwoord) af.

XDM levert de gebruiker dezelfde functionaliteit levert als getty(8) (zie Instellen). Dus het regelt de systeemaanmeldingen voor de schermen waaraan verbonden moet worden en start dan een sessie manager namens de gebruiker (meestal een X window manager). XDM wacht dan tot het programma stopt en geeft aan dat de gebruiker klaar is en afgemeld kan worden. Hierna kan XDM het aanmeldscherm weer tonen zodat de volgende gebruiker kan aanmelden.

5.6.2. XDM gebruiken

Om XDM te gebruiken moet de port x11/xdm geïnstalleerd worden (het wordt in recente versies van Xorg niet standaard geïnstalleerd). Het daemon-programma XDM is daarna beschikbaar in /usr/local/bin/xdm. Dit programma kan als `root` altijd gestart worden en regelt dan het X weergavegedeelte van de lokale machine. Als XDM iedere keer bij het opstarten moet starten is het handig om een regel toe te voegen aan /etc/ttys. Meer informatie over het gebruik van dit bestand staat in Een regel aan /etc/ttys toevoegen. In de standaardversie van /etc/ttys staat een regel om de applicatie daemon XDM op een virtuele terminal te draaien:

```
ttyv8   "/usr/local/bin/xdm -nodaemon"  xterm   off secure
```

Standaard staat deze regel uit. Om hem aan te zetten moet veld 5 van `off` naar `on` gewijzigd worden en moet met init(8) herstart worden met gebruikmaking van de aanwijzingen in init forceren om /etc/ttys opnieuw te lezen. Het eerste veld, de naam van de terminal die het programma aanstuurt, is `ttyv8`. Dit houdt in dat XDM op de negende virtuele terminal begint te draaien.

5.6.3. XDM instellen

De map met instellingen voor XDM is /usr/local/lib/X11/xdm. In deze map staan diverse bestanden die gebruikt kunnen worden om het gedrag en uiterlijk van XDM te veranderen. Meestal zijn dit de volgende bestanden:

Bestand	Omschrijving
Xaccess	Regels voor client authorisatie.
Xresources	Standaard waarden voor X bronnen.
Xservers	Lijst met op afstand en lokaal te beheren schermen.
Xsession	Standaard sessie script voor logins.
Xsetup_*	Script die applicaties start voordat de login interface start.
xdm-config	Algehele instellingen voor alle schermen op deze machine.
xdm-errors	Fouten die gegenereerd zijn door het serverprogramma.
xdm-pid	Het proces ID van de draaiende XDM.

Tevens staan in deze map een aantal scripts en programma's om het bureaublad in te stellen als XDM draait. Het doel van elk van deze bestanden wordt kort omschreven. De juiste syntaxis en het gebruik van deze bestanden staat in xdm(1).

De standaardinstelling regelt een eenvoudig rechthoekig aanmeldvenster met bovenin de hostnaam van de machine in een groot lettertype met een "Login:" en "Password:" prompt eronder. Dit is een goed beginpunt om het uiterlijk en werking van het XDM venster te veranderen.

Xaccess

Om een verbinding te maken met XDM-gestuurde schermen wordt het protocol X Display Manager Connection Protocol (XDMCP) gebruikt. Het bestand is een set regels die XDMCP verbindingen met andere machines bestuurt. Het wordt genegeerd, tenzij xdm-config is gewijzigd zodat er wordt geluisterd naar inkomende verbindingen. Standaard wordt het clients niet toegestaan te verbinden.

Xresources

Dit is een bestand met standaarden voor de schermkiezer en de aanmeldschermen. Hier kan het uiterlijk van het aanmeldprogramma gewijzigd worden. De indeling is hetzelfde als bij het app-defaults bestand en is beschreven in de X11 documentatie.

Xservers

Dit is een lijst met netwerkschermen waaruit gekozen kan worden.

Xsession

Dit is het standaard sessiescript voor XDM dat start nadat de gebruiker is aangemeld. Normaal heeft iedere gebruiker een eigen sessiescript in ~/.xsession dat dit script overheerst.

Xsetup_*

Deze starten automatisch voordat de kiezers of aanmeldschermen getoond worden. Er is een script voor ieder gebruikt scherm met de naam Xsetup_ gevolgd door het lokale schermnummer (bijvoorbeeld Xsetup_0). Normaal draaien deze scripts éé of twee programma's in de achtergrond zoals `xconsole`.

xdm-config

Dit bevat de instellingen die toegepast worden op ieder scherm die deze installatie aanstuurt. De indeling is hetzelfde als van app-defaults.

xdm-errors

Hierin staan de meldingen die de X servers geven als XDM ze probeert te starten. Als een scherm dat gestart is door XDM om onduidelijke reden hangt, is dit een goede plaats om te zoeken naar foutmeldingen. Deze meldingen worden ook per sessie naar het ~/.xsession-errors van de gebruiker gestuurd.

5.6.4. Een netwerk beeldschermserver gebruiken

Om gebruikers een verbinding te laten maken met een X server moeten de toegangsregels gewijzigd worden en de connectielistener moet aangezet worden. Deze hebben standaard wat terughoudende waarden. Om XDM te laten luisteren naar verbindingen moet als eerste een regel uitgecommentarieerd worden in xdm-config:

```
! SECURITY: do not listen for XDMCP or Chooser requests
! Comment out this line if you want to manage X terminals with XDM
DisplayManager.requestPort:     0
```

Hierna moet XDM herstart worden. Afwijkend in dit bestand is dat commentaar in app-defaults bestanden begint met het karakter "!" en niet met het karakter "#". Het kan wenselijk zijn om de toegangscontrole aan te scherpen - hiervoor staan voorbeeldregels in Xaccess en lees de hulppagina xdm(1) voor meer informatie.

5.6.5. Alternatieven voor XDM

Er bestaan diverse alternatieven voor het XDM programma. KDM (wordt geleverd bij KDE) wordt later in dit hoofdstuk behandeld. De beeldschermmanager KDM biedt vele grafische verbeteringen en cosmetische franje en de mogelijkheid om de gebruiker de kans te geven een window manager

te laten kiezen bij het aanmelden.

5.7. Bureaubladomgevingen

Deze sectie beschrijft de verschillende bureaubladomgevingen voor X op FreeBSD. Een "bureaubladomgeving" kan van alles inhouden: van een simpele window manager tot een complete suite van bureaubladapplicaties zoals KDE of GNOME.

5.7.1. GNOME

Over GNOME

GNOME is een gebruikersvriendelijke bureaubladomgeving die de gebruiker de mogelijkheid geeft om gemakkelijk de computer te gebruiken en in te stellen. GNOME heeft een paneel (voor het starten en tonen van statusinformatie van applicaties), een bureaublad (waar data en applicaties geplaatst kunnen worden), een set standaard bureaubladapplicaties en een regels die het makkelijker maakt voor applicaties om eenduidig met elkaar samen te werken. Gebruikers van andere besturingssystemen of omgevingen voelen zich meestal meteen thuis bij het gebruik van de krachtige grafisch gestuurde omgeving die GNOME biedt. Meer informatie over GNOME op FreeBSD staat op de FreeBSD GNOME Project website. De website bevat ook redelijk complete FAQ's over het installeren, instellen en beheren van GNOME.

GNOME installeren

De software kan eenvoudig worden geïnstalleerd vanuit een pakket of de Portscollectie:

Om het GNOME pakket te installeren:

```
# pkg_add -r gnome2
```

Om GNOME vanuit de Portscollectie te installeren:

```
# cd /usr/ports/x11/gnome2
# make install clean
```

Voor een correcte werking, vereist GNOME dat het /proc bestandssysteem gekoppeld is. Voeg

```
proc           /proc       procfs  rw  0   0
```

toe aan /etc/fstab om procfs(5) automatisch te koppelen tijdens het opstarten.

Zodra GNOME geïnstalleerd is, moet de X server verteld worden dat in plaats van de standaard window manager GNOME gebruikt moet worden.

De meest eenvoudige manier om GNOME te starten is via GDM, de GNOME Display Manager. GDM wordt meegeïnstalleerd met de GNOME bureaubladomgeving, maar staat standaard uitgeschakeld.

Dit programma kan ingeschakeld worden door het volgende toe te voegen aan /etc/rc.conf:

```
gdm_enable="YES"
```

Na een herstart zal GDM automatisch gestart worden.

Meestal is het gewenst om alle GNOME applicaties tegelijkertijd met GDM te starten. Om dit te bereiken moet de volgende regel worden toegevoegd aan /etc/rc.conf:

```
gnome_enable="YES"
```

GNOME kan ook gestart worden vanaf de commandoregel door het bestand .xinitrc juist in te stellen. Als er al een .xinitrc is, dan hoeft alleen de regel die de huidige window manager start veranderd te worden in een regel die /usr/local/bin/gnome-session start. Als er niets speciaals met dit instellingenbestand is gedaan:

```
% echo "/usr/local/bin/gnome-session" > ~/.xinitrc
```

Nu kan met `startx` de GNOME bureaubladomgeving gestart worden.



Als een beeldschermmanager als XDM gebruikt wordt werkt het bovenstaande niet. In plaats daarvan moet een uitvoerbaar .xsession gemaakt worden met hetzelfde commando erin. Hiervoor moet het bestand aangepast worden door het bestaande window manager commando te vervangen door /usr/local/bin/gnome-session:

```
% echo "#!/bin/sh" > ~/.xsession
% echo "/usr/local/bin/gnome-session" >> ~/.xsession
% chmod +x ~/.xsession
```

Het is ook mogelijk de beeldschermmanager zo in te stellen dat de window manager gekozen kan worden tijdens het aanmelden. In de paragraaf Meer KDE Details wordt uitgelegd hoe dit gedaan moet worden voor de KDM beeldschermmanager van KDE.

5.7.2. KDE

Over KDE

KDE is een bureaubladomgeving die eigentijds is en makkelijk in gebruik. KDE biedt de gebruiker:

- Een schitterende eigentijdse desktop;
- Een desktop die volledig netwerktransparant is;
- Een geïntegreerd hulpsysteem dat eenvoudig bruikbare informatie geeft over het gebruik van het KDE bureaublad en de applicaties;

- Alle KDE applicaties werken op dezelfde manier en zien er hetzelfde uit;
- Gestandaardiseerde menu's en werkbalken, keybindings, kleurschema's, enzovoort;
- Internationalisatie: KDE is beschikbaar in meer dan 55 talen;
- Gecentraliseerde, consistente, dialooggedreven bureaubladinstelling;
- Een grote hoeveelheid bruikbare KDE applicaties;

KDE wordt geleverd met een webbrowser genaamd Konqueror die niet onder doet voor de andere bestaande webbrowsers op UNIX® systemen. Meer informatie over KDE staat op de KDE website. Voor FreeBSD specifieke informatie en bronnen over KDE is er de website KDE/FreeBSD initiatief.

Er zijn twee versies van KDE beschikbaar op FreeBSD. Versie 3 is sinds lange tijd aanwezig en is nog steeds beschikbaar in de Portscollectie alhoewel het nu onbeheerd en gedeeltelijk kapot is. Versie 4 wordt punctueel bijgewerkt en is de standaardkeuze voor gebruikers van KDE. Ze kunnen zelfs naast elkaar worden geïnstalleerd.

KDE installeren

Net als bij GNOME of iedere andere bureaubladomgeving kan de software eenvoudig geïnstalleerd met een pakket of uit de Portscollectie:

Om het KDE 3 pakket van het netwerk te installeren:

```
# pkg_add -r kde
```

Om het KDE 4 pakket van het netwerk te installeren:

```
# pkg_add -r kde4
```

pkg_add(1) haalt automatisch de laatste versie van de applicatie op.

Om KDE 3 vanuit de Portscollectie te bouwen en te installeren:

```
# cd /usr/ports/x11/kde3
# make install clean
```

Gebruik de Portscollectie om KDE 4 vanuit de broncode te bouwen:

```
# cd /usr/ports/x11/kde4
# make install clean
```

Nadat KDE geïnstalleerd is, moet de X server verteld worden dat déze applicatie gestart moet worden in plaats van de standaard window manager. Hiervoor kan .xinitrc aangepast worden:

Voor KDE 3:

```
% echo "exec startkde" > ~/.xinitrc
```

Voor KDE 4:

```
% echo "exec /usr/local/kde4/bin/startkde" > ~/.xinitrc
```

Als het X Window System wordt gestart met `startx` is KDE het bureaublad.

Als er een beeldschermmanager als XDM gebruikt wordt, is de instelling anders. Dan moet .xsession gewijzigd worden. Instructies voor KDM worden later in dit hoofdstuk beschreven.

5.7.3. Meer KDE details

Nadat KDE geïnstalleerd is op een systeem, kunnen de meeste dingen uitgezocht worden via de hulppagina's of door de verschillende menu's aan te wijzen en erop te klikken. Windows® en Mac® gebruikers voelen zich meestal helemaal thuis.

Het beste naslagwerk voor KDE is de on-line documentatie. KDE heeft zijn eigen web browser, Konqueror, tientallen handige applicaties en uitgebreide documentatie. De volgende paragrafen beschrijven de technische zaken die moeilijk proefondervindelijk te achterhalen zijn.

De KDE beeldschermmanager

Een beheerder van een multi-user systeem die een grafisch aanmeldscherm willen hebben voor zijn gebruikers kan hiervoor XDM gebruiken, zoals eerder beschreven. KDE biedt KDM als alternatief. Dat is ontworpen met een beter uiterlijk en heeft meer aanmeldopties. Gebruikers kunnen via een menu kiezen welke bureaubladomgeving (KDE, GNOME of een andere) zij na het aanmelden willen gebruiken.

Om KDM te starten, moeten verschillende bestanden gewijzigd worden, afhankelijk van de versie van KDE.

Voor KDE 3 dient de regel met `ttyv8` als volgt aangepast te worden:

```
ttyv8 "/usr/local/bin/kdm -nodaemon" xterm on secure
```

Voor KDE 4 dient procfs(5) te worden aangekoppeld en de volgende regel aan /etc/rc.conf te worden toegevoegd:

```
kdm4_enable="YES"
```

5.7.4. Xfce

Over Xfce

Xfce is een bureaubladomgeving die gebaseerd is op de GTK+ toolkit die gebruikt wordt bij GNOME, maar is eenvoudiger en bedoeld voor gebruikers die een simpel en efficiënt bureaublad willen dat toch eenvoudig en makkelijk in te stellen is. Het ziet er bijna hetzelfde uit als CDE dat bij commerciële UNIX® systemen zit. Een aantal Xfce functies zijn:

- Een eenvoudige, makkelijk te bedienen desktop;
- Geheel in te stellen met de muis, met klikken en slepen, enzovoort;
- Hoofdpaneel hetzelfde als CDE met menu's, applets en applicaties
- Geïntegreerde window manager, bestandsmanager, geluidsmanager, GNOME compliance module en meer zaken;
- Thema's (sinds het gebruik van GTK+);
- Snel, licht en efficiënt: ideaal voor de oudere of langzamere machines of machines met beperkte hoeveelheid geheugen;

Meer informatie over Xfce staat op de Xfce website.

Installeren van Xfce

Xfce is met een pakket te installeren:

```
# pkg_add -r xfce4
```

Of vanuit de Portscollectie:

```
# cd /usr/ports/x11-wm/xfce4
# make install clean
```

Nu moet de X server weten dat Xfce gestart moet worden als X de volgende keer start:

```
% echo "/usr/local/bin/startxfce4" > ~/.xinitrc
```

De volgende keer dat X start is Xfce het bureaublad. Wederom: als een beeldschermmanager als XDM gebruikt wordt, moet .xsession gemaakt worden zoals beschreven in de paragraaf over GNOME. Nu moet echter het command /usr/local/bin/startxfce4 gebruikt. Het is ook mogelijk de beeldschermmanager in te stellen om bureaublad te kiezen bij het aanmelden, zoals is uitgelegd in de paragraaf over kdm.

Unresolved directive in book.adoc - include::wayland/_index.adoc[leveloffset=+1]

Unresolved directive in book.adoc - include::network/_index.adoc[leveloffset=+1]

Deel II: Algemene taken

Na de inleiding gaat dit deel van het FreeBSD handboek over een aantal vaak gebruikte mogelijkheden van FreeBSD. De volgende hoofdstukken:

- Geven een inleiding in populaire en handige desktop toepassingen: browsers, productieviteitsgereedschappen, documentviewers, etc;
- Geven een inleiding in een aantal multimediatoepassingen die in FreeBSD beschikbaar zijn;
- Geven uitleg over het proces waarmee een aangepaste kernel voor FreeBSD kan worden gemaakt om extra functionaliteit aan een systeen toe te voegen;
- Beschrijven gedetailleerd het afdruksysteem, zowel voor met een desktop verbonden als met het netwerk verbinden printers;
- Beschrijven hoe applicaties voor Linux® op FreeBSD kunnen draaien.

In een aantal van de hoofdstukken wordt voorkennis aangeraden. Dit staat vermeld in de inleiding van ieder hoofdstuk.

Hoofdstuk 6. Bureaubladapplicaties

6.1. Overzicht

FreeBSD kan een groot aantal bureaubladapplicaties draaien, zoals browsers en tekstverwerkers. De meeste hiervan zijn beschikbaar als pakketten of kunnen automatisch vanuit de Portscollectie gebouwd worden. Veel nieuwe gebruikers verwachten dit soort applicaties op hun bureaublad. Dit hoofdstuk laat zien hoe populaire bureaubladapplicaties moeiteloos geïnstalleerd kunnen worden vanuit een pakket of vanuit de Portscollectie.

Als programma's vanuit ports geïnstalleerd worden, wordt hun broncode gecompileerd. Dit kan erg lang duren, afhankelijk van wat er gecompileerd wordt en de rekenkracht van een machine. Als compileren vanuit broncode te veel tijd kost, kunnen de meeste programma's van de Portscollectie als een voorgebouwd pakket geïnstalleerd worden.

Omdat FreeBSD compatibel is met Linux®, zijn veel applicaties die voor Linux® zijn ontwikkeld beschikbaar een FreeBSD bureaublad. Het wordt sterk aanbevolen om Linux® binaire compatibiliteit te lezen voordat Linux® applicaties geïnstalleerd worden. Veel ports die gebruik maken van Linux® compatibiliteit beginnen met "linux-". Dit is handig om te onthouden wanneer er naar een port gezocht wordt met bijvoorbeeld whereis(1). In dit hoofdstuk wordt aangenomen dat Linux® binaire compatibiliteit is ingeschakeld voordat Linux® applicaties worden geïnstalleerd.

In dit hoofdstuk worden de volgende categoriën behandeld:

- Browsers (zoals Firefox, Opera, Konqueror, Chromium)
- Productiviteit (zoals KOffice, AbiWord, The GIMP, Apache OpenOffice, LibreOffice)
- Documentviewers (zoals Acrobat Reader®, gv, Xpdf, GQview)
- Financieel (zoals GnuCash, Gnumeric, Abacus)

Er wordt aangenomen dat de lezer van dit hoofdstuk:

- Weet hoe aanvullende software van derde partijen geïnstalleerd wordt (Applicaties installeren. pakketten en ports).
- Weet hoe aanvullende Linux® software geïnstalleerd wordt (Linux® binaire compatibiliteit).

Meer informatie over een multimedia-omgeving staat in Multimedia. Installatie van email staat beschreven in Elektronische mail.

6.2. Browsers

FreeBSD wordt zonder een voorgeïnstalleerde browser geleverd. In plaats hiervan bevat de www map van de Portscollectie browsers om te installeren. Het is ook mogelijk voor de meeste ports een pakket te installeren als compileren niet gewenst is. Compileren kan soms lang duren.

KDE en GNOME bevatten reeds HTML-browsers. In Bureaubladomgevingen staat meer informatie over de installatie van deze complete bureaubladen.

Lichtgewicht browsers uit de Portscollectie zijn onder andere www/dillo2, www/links of www/w3m.

Dit gedeelte behandelt deze applicaties:

Applicatie	Bronnen	Ports	Grote afhankelijkheden
Firefox	gemiddeld	zwaar	Gtk+
Opera	weinig	licht	FreeBSD en Linux® versies beschikbaar. De Linux® versie is afhankelijk van de Linux® binaire compatibiliteit en linux-openmotif.
Konqueror	gemiddeld	zwaar	KDE bibliotheken
Chromium	gemiddeld	gemiddeld	Gtk+

6.2.1. Firefox

Firefox is een moderne, gratis, stabiele open-source browser die volledig geporteerd is naar FreeBSD: het heeft een motor voor HTML-weergave die zich zeer strikt aan de standaarden houdt, browsen met tabbladen, blokkeren van pop-ups, uitbreidingen, verbeterde veiligheid, en meer. Firefox is gebaseerd op de codebase van Mozilla.

Installeer het pakket door het volgende te typen:

```
# pkg_add -r firefox
```

Dit zal de laatste uitgave van Firefox installeren, als u in plaats hiervan de Extended Support Release (ESR) van Firefox wilt draaien, gebruik dan:

```
# pkg_add -r firefox-esr
```

De Portscollectie kan ook gebruikt worden als u liever vanuit de broncode installeert.

```
# cd /usr/ports/www/firefox
# make install clean
```

Voor Firefox ESR dient `firefox` in het vorige commando vervangen te worden door `firefox-esr`.

6.2.2. Firefox en Java™ plugin



In deze en de volgende twee secties wordt er vanuit gegaan dat Firefox reeds

> geïnstalleerd is.

Installeer OpenJDK 6 vanuit de Ports Collectie door het volgende typen:

```
# cd /usr/ports/java/openjdk6
# make install clean
```

Installeer daarna de port java/icedtea-web:

```
# cd /usr/ports/java/icedtea-web
# make install clean
```

Zorg ervoor dat de standaard configuratieopties voor beide ports zijn geselecteerd.

Start de browser en voer `about:plugins` in de locatie balk en druk op `Enter`. Er zal een pagina gepresenteerd worden die de geïnstalleerde plugins toont; de Java™ plugin zal nu getoond moeten worden.

Als de browser de plugin niet kan vinden, dient elke gebruiker het volgende commando uit te voeren en de browser opnieuw te starten:

```
% ln -s /usr/local/lib/IcedTeaPlugin.so \
  $HOME/.mozilla/plugins/
```

6.2.3. Firefox en Adobe® Flash™ plugin

De Adobe® Flash™ plugin is niet beschikbaar voor FreeBSD. Er is echter wel een softwarelaag (wrapper) om de Linux®-versie van de plugin te draaien. Deze wrapper ondersteunt ook Adobe® Acrobat® plugin, RealPlayer® plugin en meer.

Afhankelijk van de versie van FreeBSD die u draait zijn er verschillende stappen nodig:

1. Op FreeBSD 7.X

   Installeer de port www/nspluginwrapper. Deze port heeft emulators/linux_base-fc4 nodig, wat een grote port is.

   De volgende stap is om de port www/linux-flashplugin9 te installeren. Dit zal Flash™ 9.X installeren, van deze versie is bekend dat die correct werkt op FreeBSD 7.X.

2. Op FreeBSD 8.X of nieuwer

   Installeer de port www/nspluginwrapper. Deze port heeft emulators/linux_base-f10 nodig, wat een grote port is.

   De volgende stap is om de Flash™ 11.X vanuit de port www/linux-f10-flashplugin11 te

installeren.

Voor deze versie is het nodig om de volgende koppeling aan te maken:

```
# ln -s /usr/local/lib/npapi/linux-f10-flashplugin/libflashplayer.so \
  /usr/local/lib/browser_plugins/
```

De /usr/local/lib/browser_plugins directory moet handmatig aangemaakt worden als deze nog niet op het systeem bestaat.

Wanneer de juiste Flash™ port, afhankelijk van de versie van FreeBSD die u draait, is geïnstalleerd, moet de plugin door elke gebruiker worden geïnstalleerd met `nspluginwrapper`:

```
% nspluginwrapper -v -a -i
```

Start dan de browser en voer op de adresbalk `about:plugins` in en druk op `Enter`. Een pagina met alle geïnstalleerde plugins wordt nu getoond.

6.2.4. Firefox en Swfdec Flash™ plugin

Swfdec is de bibliotheek om Flash™-animaties te decoderen en af te beelden. Swfdec-Mozilla is een plugin voor Firefox-browsers dat de Swfdec-bibliotheek gebruikt om SWF-bestanden af te spelen. Er wordt nog steeds veel aan ontwikkeld.

Als u het niet kunt of wilt compileren, kan het pakket vanaf het netwerk worden geïnstalleerd:

```
# pkg_add -r swfdec-plugin
```

Als het pakket niet beschikbaar is, kunt u het vanuit de Portscollectie compileren en installeren:

```
# cd /usr/ports/www/swfdec-plugin
# make install clean
```

Herstart hierna uw browser om deze plugin effectief te maken.

6.2.5. Opera

Opera is een volledige en een standaard volgende browser. Hij wordt standaard geleverd met een ingebouwde email-client, een nieuwslezer, een IRC client, een RSS/ATOM feed lezer en nog veel meer. Ondanks dat is Opera relatief gezien niet zwaar en erg snel. Hij komt in twee smaken: een FreeBSD versie en een versie die draait onder Linux® emulatie.

De FreeBSD pakketversie van Opera wordt zo geïnstalleerd:

```
# pkg_add -r opera
```

Sommige FTP-sites hebben niet alle pakketten, maar Operakan worden nog altijd via de Portscollectie worden verkregen door te typen:

```
# cd /usr/ports/www/opera
# make install clean
```

De Linux® versie van Opera kan geïnstalleerd worden door bij de bovenstaande voorbeelden `linux-opera` te gebruiken in plaats van `opera`.

De Adobe® Flash™ plugin is niet beschikbaar voor FreeBSD. Er bestaat echter een Linux® versie van de plugin. Om deze versie te installeren moet de port www/linux-f10-flashplugin11 geïnstalleerd zijn, installeer daarna de port www/opera-linuxplugins:

```
# cd /usr/ports/www/linux-f10-flashplugin11
# make install clean
# cd /usr/ports/www/opera-linuxplugins
# make install clean
```

U kunt controleren of de plugin aanwezig is: start uw browser, geef `opera:plugins` in op de adresbalk en druk op `Enter`. Er zou een lijst moeten verschijnen met alle huidig beschikbare plugins.

Volg de instructies voor Firefox om de Java™ plugin te installeren.

6.2.6. Konqueror

Konqueror is deel van KDE, maar kan ook buiten KDE gebruikt worden door x11/kdebase3 te installeren. Konqueror is meer dan een browser, het is ook een bestandsbeheerder en multimedia-viewer.

Er is ook een verzameling plugins beschikbaar voor Konqueror, beschikbaar in misc/konq-plugins.

Konqueror ondersteunt WebKit naast het eigen KHTML. WebKit wordt gebruikt door vele moderne browsers waaronder Chromium. Om WebKit met Konqueror op FreeBSD te gebruiken:

```
# cd /usr/ports/www/kwebkitpart
# make install clean
```

Klik vervolgens in Konqueror op "Settings", "Configure Konqueror", en "Change KHTML to WebKit".

Konqueror ondersteunt ook Flash™; een "How To" gids om ondersteuning voor Flash™ in Konqueror te krijgen is beschikbaar op http://freebsd.kde.org/howtos/konqueror-flash.php.

6.2.7. Chromium

Chromium is een open-source browserproject dat er op gericht is om een veiligere, snellere en stabielere surfervaring op te bouwen. Chromium biedt surfen met tabbladen, het blokkeren van pop-ups, uitbreidingen en nog veel meer. Chromium is het open-source project waar de browser Google Chrome op is gebaseerd.

Chromium kan als volgt als een pakket worden geïnstalleerd:

```
# pkg_add -r chromium
```

Als alternatief kan Chromium worden gecompileerd vanuit de broncode door de Portscollectie te gebruiken:

```
# cd /usr/ports/www/chromium
# make install clean
```



Chromium wordt geïnstalleerd als /usr/local/bin/chrome, niet als /usr/local/bin/chromium.

6.2.8. Chromium en Java™ plugin



Deze sectie neemt aan dat Chromium al is geïnstalleerd.

Installeer OpenJDK 6 vanuit de Portscollectie:

```
# cd /usr/ports/java/openjdk6
# make install clean
```

Installeer vervolgens java/icedtea-web vanuit de Portscollectie:

```
# cd /usr/ports/java/icedtea-web
# make install clean
```

Start Chromium en geef `about:plugins` op in de adresbalk. IcedTea-Web zou genoemd moeten worden als één van de geïnstalleerde plugins.

Als Chromium de plugin IcedTea-Web niet vermeldt, voer dan de volgende commando's uit en herstart de browser:

```
# mkdir -p /usr/local/shared/chromium/plugins
# ln -s /usr/local/lib/IcedTeaPlugin.so \
  /usr/local/shared/chromium/plugins/
```

6.2.9. Chromium en Adobe® Flash™ plugin



Deze sectie neemt aan dat Chromium al is geïnstalleerd.

Het configureren van Chromium en Adobe® Flash™ lijkt op de instructies voor Firefox. Raadpleeg die sectie voor gedetailleerdere instructies en het installeren van Adobe® Flash™ op FreeBSD. Er zou geen verdere configuratie nodig moeten zijn, aangezien Chromium sommige plugins van andere browsers kan gebruiken.

6.3. Productiviteit

Als het op productiviteit aankomt, zoeken nieuwe gebruikers vaak een goed kantoorpakket of een vriendelijke tekstverwerker. Hoewel sommige bureaubladomgevingen zoals KDE reeds een kantoorpakket verschaffen, is er geen standaard produktiviteitspakket. FreeBSD kan alles verschaffen wat nodig is, ongeacht de bureaubladomgeving.

In dit gedeelte worden de onderstaande applicaties beschreven:

Applicatie	Bronnen	Ports	Afhankelijkheden
KOffice	weinig	zwaar	KDE
AbiWord	weinig	licht	Gtk+ of GNOME
The GIMP	weinig	licht	Gtk+
Apache OpenOffice	veel	erg zwaar	JDK™, Mozilla
LibreOffice	enigszins veel	zwaar	Gtk+, of KDE / GNOME, of JDK™

6.3.1. KOffice

De KDE-gemeenschap heeft zijn bureaubladomgeving met een kantoorpakket geleverd dat buiten KDE gebruikt kan worden. Het bevat de vier standaardcomponenten uit andere kantoorpakketten. KWord is de tekstverwerker, KSpread is het spreadsheetprogramma, KPresenter beheert diapresentaties en Kontour voorziet in grafische mogelijkheden.

Voordat de nieuwste KOffice wordt geïnstalleert, moet er een recente versie van KDE geïnstalleerd zijn.

KOffice voor KDE als pakket installeren gaat met het volgende commando:

```
# pkg_add -r koffice-kde4
```

Als het pakket niet beschikbaar is, kan de Portscollectie gebruiken worden. Om KOffice voor KDE4 te installeren:

```
# cd /usr/ports/editors/koffice-kde4
```

```
# make install clean
```

6.3.2. AbiWord

AbiWord is een vrij tekstverwerkingsprogramma, ongeveer gelijk aandoet als Microsoft® Word. Het is geschikt om verslagen, brieven, rapporten, memo's, enzovoort mee te typen. Het programma is snel, bevat veel mogelijkheden en is gebruikersvriendelijk.

AbiWord kan veel bestandsformaten importeren en exporteren, waaronder enkele gesloten formaten, zoals Microsoft®'s .doc.

AbiWord is beschikbaar als pakket en te installeren met:

```
# pkg_add -r abiword
```

Als het pakket niet beschikbaar is, kan het worden gecompileerd vanuit de Portscollectie. De Portscollectie is meer recent. Dat kan als volgt:

```
# cd /usr/ports/editors/abiword
# make install clean
```

6.3.3. The GIMP

Voor het bewerken of retoucheren van afbeeldingen is The GIMP een zeer geavanceerd afbeeldingenmanipulatieprogramma. Het kan als eenvoudig tekenprogramma worden gebruikt of als kwalititeitspakket voor het retoucheren van foto's. Het ondersteunt een groot aantal plugins en bevat een scripting interface. The GIMP kan een groot aantal bestandsformaten lezen en schrijven. Het ondersteunt interfaces met scanners en tabletten.

Het pakket is te installeren met:

```
# pkg_add -r gimp
```

Als een FTP-site dit pakket niet heeft, kan de Portscollectie gebruikt worden. De graphics map van de Portscollectie bevat ook The GIMP Manual. Die kan zo geïnstalleerd worden:

```
# cd /usr/ports/graphics/gimp
# make install clean
# cd /usr/ports/graphics/gimp-manual-pdf
# make install clean
```



De graphics map van de Portscollectie bevat de ontwikkelversie van The GIMP in graphics/gimp-devel. Een HTML-versie van The GIMP Manual staan in graphics/gimp-manual-html.

6.3.4. Apache OpenOffice

Op 1 juni 2011 doneerde Oracle Corporation de codebasis van OpenOffice.org aan de Apache Software Foundation. OpenOffice.org staat nu bekend als Apache OpenOffice en wordt ontwikkeld onder de vleugels van de Incubator van de Apache Software Foundation.

Apache OpenOffice bevat alle noodzakelijke applicaties in een compleet kantoorproductiviteitspakket: een tekstverwerker, een spreadsheet, een presentatiebeheerder en een tekenprogramma. De gebruikersinterface is vrijwel gelijk aan die van andere kantoorpakketten en het kan veel populaire bestandsformaten in- en uitvoeren. Het is beschikbaar in een aantal verschillende talen - internationalisatie is uitgebreid tot interfaces, spellingcontrole, en woordenboeken.

De tekstverwerker van Apache OpenOffice gebruikt een eigen XML-bestandsformaat voor overdraagbaarheid en flexibiliteit. Het spreadsheetprogramma bevat een macrotaal en kan gekoppeld worden aan externe databases. Apache OpenOffice is stabiel en draait zonder aanpassingen op Windows®, Solaris™, Linux, FreeBSD en Mac OS® X. Meer informatie over Apache OpenOffice staat op de Apache OpenOffice website. Voor specifieke FreeBSD informatie en om direct pakketten te downloaden is er de website van het FreeBSD Apache OpenOffice Porting Team.

Om Apache OpenOffice te installeren:

```
# pkg_add -r apache-openoffice
```



Dit hoort te werken als er een -RELEASE versie van FreeBSD wordt gedraaid. In andere gevallen is het verstandig om te kijken op de website van het FreeBSD Apache OpenOffice Porting Team en het juiste pakket met pkg_add(1) te downloaden en te installeren. Zowel de huidige release als de ontwikkelversie kunnen op die locatie gedownload worden.

Als het pakket geïnstalleerd is, start dan met het volgende commando Apache OpenOffice:

```
% openoffice-X.Y.Z
```

waarbij *X.Y.Z* het versienummer van de geïnstalleerde Apache OpenOffice is, bijvoorbeeld *3.4.0*.



Tijdens de eerste keer starten worden er een aantal vragen gesteld en wordt de map .openoffice.org in de thuismap van de aangemelde gebruiker gemaakt.

Als de Apache OpenOffice pakketten niet beschikbaar zijn, kan het uit de ports gecompileerd worden. Hiervoor is veel schijfruimte en tijd nodig:

```
# cd /usr/ports/editors/openoffice-3
# make install clean
```



Vervang om een gelokaliseerde versie te bouwen de voorgaande commandoregel door de volgende:

```
# make LOCALIZED_LANG=uw_taal install clean
```

Vervang *taal* door de juiste ISO-taalcode. Een lijst met ondersteunde taalcodes is beschikbaar in het bestand files/Makefile.localized in de map van de port.

Start hierna Apache OpenOffice met:

```
% openoffice-X.Y.Z
```

waarbij *X.Y.Z* het versienummer van de geïnstalleerde Apache OpenOffice is, bijvoorbeeld *3.4.0*.

6.3.5. LibreOffice

LibreOffice is een gratis kantoorpakket ontwikkeld door The Document Foundation en is compatibel met andere grote kantoorpakketten en is beschikbaar op meerdere platforms. Het is een afsplitsing van OpenOffice.org onder een nieuw merk en bevat alle verwachte toepassingen van een compleet kantoorpakket: een tekstverwerker, een spreadsheet, een presentatiebeheerder, een tekenprogramma, een databasebeheerprogramma, en een programma om wiskundige formules te bewerken. Het is beschikbaar in een aantal verschillende talen - internationalisatie heeft zich uitgebreid naar interfaces, spellingcheckers en woordenboeken.

De tekstverwerker van LibreOffice gebruikt een eigen XML-bestandsformaat voor verhoogde portabiliteit en flexibiliteit. Het spreadsheetprogramma bevat een macrotaal en kan met externe databases gebruikt worden. LibreOffice is reeds stabiel en draait op Windows®, Linux, FreeBSD, en Mac OS® X. Meer informatie over LibreOffice is te vinden op de website van LibreOffice.

Om LibreOffice als een pakket te installeren:

```
# pkg_add -r libreoffice
```



Dit zou moeten werken met een -RELEASE-versie van FreeBSD.

Als het pakket is geïnstalleerd, dient de volgende opdracht gebruikt te worden om LibreOffice te draaien:

```
% libreoffice
```



Tijdens de eerste keer draaien worden u wat vragen gesteld en wordt er een map .libreoffice aangemaakt in uw thuismap.

Als er geen pakket voor LibreOffice beschikbaar is, heeft u nog altijd de optie om de port te

compileren. Denk er echter aan dat dit veel schijfruimte en redelijk veel tijd kost.

```
# cd /usr/ports/editors/libreoffice
# make install clean
```



Als u een gelokaliseerde versie wilt bouwen, dient u de vorige opdracht door het volgende te vervangen:

```
# make LOCALIZED_LANG=uw_taal install clean
```

U dient *uw_taal* te vervangen door de juiste ISO-taalcode. Een lijst met ondersteunde talen is beschikbaar in het doel `pre-fetch` van de Makefile van de port.

Wanneer dit is gedaan, kan LibreOffice gestart worden met deze opdracht:

```
% libreoffice
```

6.4. Documentviewers

Sommige nieuwe documentformaten hebben aan populariteit gewonnen sinds de komst van UNIX®; het kan zijn dat de standaardviewers die ze vereisen niet in het basissysteem zitten. In dit gedeelte wordt aangegeven hoe zulke viewers geïnstalleerd kunnen worden.

Dit gedeelte behandelt de onderstaande applicaties:

Applicatie	Bronnen	Ports	Afhankelijkheden
Acrobat Reader®	weinig	licht	Linux® binaire compatibiliteit
gv	weinig	licht	Xaw3d
Xpdf	weinig	licht	FreeType
GQview	weinig	licht	Gtk+ of GNOME

6.4.1. Acrobat Reader®

Documenten worden vaak als PDF-bestanden, "Portable Document Format", verspreid. Een van de aanbevolen viewers voor dit bestandstype is Acrobat Reader® dat Adobe voor Linux® heeft uitgegeven. Omdat FreeBSD Linux® binaries kan draaien, is het ook beschikbaar voor FreeBSD.

Om Acrobat Reader® 8 te installeren uit de Portscollectie:

```
# cd /usr/ports/print/acroread8
```

```
# make install clean
```

Vanwege de licentie is een pakket niet beschikbaar.

6.4.2. gv

gv is een PostScript® en PDF viewer. Het is gebaseerd op ghostview maar heeft een vriendelijker uiterlijk dankzij de Xaw3d bibliotheek. Het is snel en heeft mogelijkheden, zoals oriëntatie, papiergrootte, schalen en anti-aliassen. Bijna elke bewerking kan met het toetsenbord of de muis worden gedaan.

gv is als pakket te installeren:

```
# pkg_add -r gv
```

Of uit de Portscollectie:

```
# cd /usr/ports/print/gv
# make install clean
```

6.4.3. Xpdf

Xpdf een efficiënte lichtgewicht PDF-viewer voor FreeBSD. Het heeft erg weinig bronnen nodig en is zeer stabiel. Het gebruikt de standaard X-fonts en is niet afhankelijk van Motif® of andere X-toolkits.

Xpdf is als pakket te installeren:

```
# pkg_add -r xpdf
```

Of uit de Portscollectie:

```
# cd /usr/ports/graphics/xpdf
# make install clean
```

Als de installatie voltooid is, kan Xpdf gestart worden en het menu kan met de rechtermuisknop geactiveerd worden.

6.4.4. GQview

GQview is een afbeeldingenbeheerder. Een bestand kan met één klik bekeken worden, er kan een externe editor opgestart worden er kunnen thumbnail-voorbeelden gemaakt worden en nog veel meer. Het bevat ook een diapresentatie-modus en enkele standaard bestandsoperaties. Er kunnen afbeeldingsverzamelingen beheerd worden en eenvoudig duplicaten gevonden worden. GQview kan het complete scherm gebruiken en ondersteunt meerdere talen.

GQview is als pakket te installeren:

```
# pkg_add -r gqview
```

Of uit de Portscollectie:

```
# cd /usr/ports/graphics/gqview
# make install clean
```

6.5. Financiën

Om financiën via het FreeBSD bureaublad te beheren zijn er krachtige en gemakkelijk te gebruiken applicaties om te installeren. Sommige zijn compatibel met wijdverbreide bestandsformaten, zoals de formaten gebruikt door Quicken en Excel om documenten op te slaan.

Dit gedeelte behandelt deze programma's:

Applicatie	Bronnen	Ports	Afhankelijkheden
GnuCash	weinig	zwaar	GNOME
Gnumeric	weinig	zwaar	GNOME
Abacus	weinig	licht	Tcl/Tk
KMyMoney	weinig	zwaar	KDE

6.5.1. GnuCash

GnuCash is onderdeel van GNOME dat gebruikersvriendelijke en krachtige applicaties aan eindgebruikers wil leveren. Met GnuCash kunnen inkomsten en uitgaven, bankrekeningen en voorraden bijgehouden worden. Het bevat een intuïtieve interface terwijl het erg professioneel blijft.

GnuCash levert een slim kasboek, een hiërarchisch systeem van rekeningen, en veel toetsenbordversnellers en auto-invul mogelijkheden. Het kan een transactie splitsen in meer gedetailleerde stukken. GnuCash kan Quicken QIF-bestanden invoeren en samenvoegen. Het kan ook met de meeste internationale datum- en valutaformaten omgaan.

GnuCash is als pakket te installeren:

```
# pkg_add -r gnucash
```

Of uit de Portscollectie:

```
# cd /usr/ports/finance/gnucash
# make install clean
```

6.5.2. Gnumeric

Gnumeric is een spreadsheetprogramma uit de GNOME bureaubladomgeving. Het maakt gebruikt van "auto-invullen" afhankelijk van het celformaat. Het kan bestanden in een aantal populaire formaten zoals Excel, Lotus 1-2-3 en Quattro Pro inlezen. Gnumeric ondersteunt grafieken door middel van het grafiekprogramma math/guppi. Het heeft een groot aantal ingebouwde functies en kent gebruikelijke celformaten als nummer, valuta, datum, tijd en veel meer.

Gnumeric is als pakket te installeren:

```
# pkg_add -r gnumeric
```

Of uit de Portscollectie:

```
# cd /usr/ports/math/gnumeric
# make install clean
```

6.5.3. Abacus

Abacus is een kleine en gemakkelijk te gebruiken spreadsheetprogramma. Het bevat veel ingebouwde functies die nuttig zijn in verschillende domeinen zoals statistiek, financiën, en wiskunde. Het kan Excel-bestanden lezen en schrijven. Abacus kan PostScript® uitvoer produceren.

Abacus is als pakket te installeren:

```
# pkg_add -r abacus
```

Of uit de Portscollectie:

```
# cd /usr/ports/deskutils/abacus
# make install clean
```

6.5.4. KMyMoney

KMyMoney is een persoonlijke financiële beheerder gebouwd voor KDE. KMyMoney poogt om alle belangrijke eigenschappen die in commerciële persoonlijke financiële beheerders zitten te bieden en te integreren. Gebruiksgemak en degelijke dubbele accounting zijn eigenschappen die worden benadrukt. KMyMoney importeert vanuit standaard Quicken Interchange Format (QIF) bestanden, houdt investeringen bij, kan met meerdere munteenheden overweg, en biedt een waaier aan rapporten. Mogelijkheden om OFX te importeren zijn via een aparte plugin beschikbaar.

Om KMyMoney als een pakket te installeren:

```
# pkg_add -r kmymoney2
```

Als het pakket niet beschikbaar is, kan de Portscollectie gebruikt worden:

```
# cd /usr/ports/finance/kmymoney2
# make install clean
```

6.6. Samenvatting

Hoewel FreeBSD populair is bij ISP's om zijn prestaties en stabiliteit, is het behoorlijk klaar voor dagelijks gebruik als een bureaublad. Met enkele duizenden applicaties als pakketten of ports, is een perfect bureaublad te bouwen dat aan alle noden voldoet.

Nu volgt nog een overzicht van alle bureaubladapplicaties die in dit hoofdstuk zijn behandeld:

Applicatie	Package	Port
Opera	linux-opera	www/linux-opera
Firefox	firefox	www/firefox
Chromium	chromium	www/chromium
KOffice	koffice-kde4	editors/koffice-kde4
AbiWord	abiword	editors/abiword
The GIMP	gimp	graphics/gimp
Apache OpenOffice	openoffice	editors/openoffice-3
LibreOffice	libreoffice	editors/libreoffice
Acrobat Reader®	acroread	print/acroread8
gv	gv	print/gv
Xpdf	xpdf	graphics/xpdf
GQview	gqview	graphics/gqview
GnuCash	gnucash	finance/gnucash
Gnumeric	gnumeric	math/gnumeric
Abacus	abacus	deskutils/abacus
KMyMoney	kmymoney2	finance/kmymoney2

Hoofdstuk 7. Multimedia

7.1. Overzicht

FreeBSD ondersteunt een breed bereik aan geluidskaarten, waardoor het mogelijk is van geluid van hoge kwaliteit op een computer te genieten. Hieronder vallen mogelijkheden om geluid op te nemen en af te spelen in de MPEG Audio Layer 3 (MP3), WAV en Ogg Vorbis formaten en vele andere formaten. De FreeBSD Portscollectie bevat ook programma's waarmee opgenomen audio bewerkt kan worden, waarmee geluidseffecten toegevoegd kunnen worden en aangesloten MIDI apparaten bestuurd kunnen worden.

Met wat experimenteren kunnen met FreeBSD videobestanden en DVD's afgespeeld worden. Er zijn minder programma's om video te encoderen, te converteren en af te spelen dan er zijn voor audio. Op het moment van schrijven is er bijvoorbeeld geen goed hercoderingsprogramma in de FreeBSD Portscollectie beschikbaar wat gebruikt kan worden om tussen formaten onderling te converteren, zoals mogelijk is met audio/sox. De software in dit landschap is echter sterk aan verandering onderhevig.

In dit hoofdstuk worden de stappen beschreven die uitgevoerd moeten worden om een geluidskaart in te stellen. Bij de installatie en instelling van X11 (Het X Window systeem) is al beschreven hoe videokaarten ingesteld kunnen worden, hoewel er nog wel een aantal mogelijkheden zijn om het afspelen te verbeteren.

Na het lezen van dit hoofdstuk weet de lezer:

- Hoe een systeem zo in te stellen dat een geluidskaart wordt herkend;
- Hoe getest kan worden of een kaart werkt;
- Hoe problemen op te lossen met betrekking tot geluidsinstellingen;
- Hoe MP3's en andere audio af te spelen en te maken;
- Hoe video wordt ondersteund door de X server;
- Welke video speler/encoderports goede resultaten geven;
- Hoe DVD's, .mpg en .avi bestanden af te spelen;
- Hoe de inhoud van CD's en DVD's naar bestanden geript kan worden;
- Hoe een TV-kaart in te stellen;
- Hoe een scanner in te stellen.

Er wordt aangenomen dat de lezer van dit hoofdstuk:

- Weet hoe een nieuwe kernel in te stellen en te installeren (De FreeBSD-kernel instellen).



Het proberen aan te koppelen van audio-CD's met mount(8) resulteert in ieder geval in een foutmelding en in het ergste geval tot een *kernel panic*. Dat type media heeft een formaat dat afwijkt van het gebruikelijke ISO-bestandssysteem.

7.2. Geluidskaart installeren

7.2.1. Systeem instellen

Alvorens te beginnen is het van belang te weten welk model een geluidskaart is, welke chip erop wordt gebruikt en of het een PCI of ISA kaart is. FreeBSD ondersteunt vele PCI en ISA kaarten. De ondersteunde audio-apparaten staan in een lijst in de Hardware Notes. In de Hardware Notes staat ook beschreven welk stuurprogramma uw kaart ondersteunt.

Om een geluidsapparaat te gebruiken dient het juiste apparaatstuurprogramma geladen te worden. Dit kan op twee manieren. De meest eenvoudige manier is simpelweg een kernelmodule te laden voor de gewenste geluidskaart met kldload(8). Dit kan vanaf de commandoregel:

```
# kldload snd_emu10k1
```

Of door als volgt de juiste regel toe te voegen aan /boot/loader.conf:

```
snd_emu10k1_load="YES"
```

De bovenstaande voorbeelden zijn voor een Creative SoundBlaster® Live! geluidskaart. De overige beschikbare laadbare geluidsmodules staan beschreven in /boot/defaults/loader.conf. Als niet compleet duidelijk is welk stuurprogramma gebruikt dient te worden, dan kan het met de module snd_driver geprobeerd worden:

```
# kldload snd_driver
```

Dit is een metastuurprogramma, dat in één keer de meest voorkomende apparaatstuurprogramma's laadt. Hiermee kan het zoeken naar het juiste stuurprogramma versneld worden. Het is ook mogelijk om alle geluidsstuurprogramma's te laden via de optie /boot/loader.conf.

Om uit te vinden welk stuurprogramma na het laden van het metastuurprogramma snd_driver wordt geladen kan de inhoud van het bestand /dev/sndstat nagekeken worden met `cat /dev/sndstat`.

Een tweede mogelijkheid is ondersteuning voor een geluidskaart statisch in de kernel te compileren. In de onderstaande paragrafen staat meer informatie over hoe op die manier ondersteuning voor hardware toegevoegd kan worden. Meer informatie over het hercompileren van een kernel staat in De FreeBSD-kernel instellen.

Aangepaste kernel maken met geluidsondersteuning

Eerst moet het stuurprogramma voor het audioraamwerk sound(4) aan de kernel toegevoegd worden. Daarvoor dient het volgende te worden opgenomen in het bestand met kernelinstellingen:

```
device sound
```

Daarna kan ondersteuning voor de specifieke geluidskaart toegevoegd worden. Daarvoor moet bekend zijn welk stuurprogramma de kaart ondersteunt. Dit kan opgezocht worden in de lijst met ondersteunde audio-apparaten in de Hardware Notes, waar de correcte stuurprogramma's voor geluidskaarten beschreven staan. Zo wordt een Creative SoundBlaster® Live! geluidskaart bijvoorbeeld ondersteund door het stuurprogramma snd_emu10k1(4). Ondersteuning voor deze kaart kan als volgt worden toegevoegd:

```
device snd_emu10k1
```

In de hulppagina voor een stuurprogramma staat welke syntaxis gebruikt kan worden. De expliciete syntaxis voor de kernelinstellingen voor elk ondersteund geluidsstuurprogramma staat ook in /usr/src/sys/conf/NOTES.

Voor niet-PnP ISA-geluidskaarten kan het nodig zijn dat de kernel informatie gegeven moet worden over de instellingen van de kaart (IRQ, I/O poort, enzovoort), zoals dat geldt voor alle niet-PnP ISA-kaarten. Dit kan via het bestand /boot/device.hints. Bij het starten van een systeem leest de loader(8) dat bestand uit en geeft de instellingen door aan de kernel. Zo gebruikt een oude Creative SoundBlaster® 16 ISA niet-PnP-kaart het stuurprogramma snd_sbc(4) samen met `snd_sb16` en dient de volgende regel toegevoegd te worden aan het kernelinstellingenbestand:

```
device snd_sbc
device snd_sb16
```

Daarnaast moet het volgende worden toegevoegd aan /boot/device.hints:

```
hint.sbc.0.at="isa"
hint.sbc.0.port="0x220"
hint.sbc.0.irq="5"
hint.sbc.0.drq="1"
hint.sbc.0.flags="0x15"
```

In dit geval gebruikt de kaart I/O poort `0x220` en IRQ `5`.

De gebruikte syntaxis voor /boot/device.hints staat beschreven in de hulppagina sound(4) en de hulppagina voor het gevraagde stuurprogramma.

De bovenstaande instellingen zijn de standaardinstellingen. In sommige gevallen moeten IRQ of andere instellingen gewijzigd worden om een apparaat juist te laten werken. In snd_sbc(4) staat meer informatie over deze kaart.

7.2.2. Geluidskaart testen

Na het herstarten met de aangepaste kernel of na het laden van de benodigde module, hoort de

geluidskaart ongeveer als volgt te verschijnen in de systeemberichtbuffer (dmesg(8)):

```
pcm0: <Intel ICH3 (82801CA)> port 0xdc80-0xdcbf,0xd800-0xd8ff irq 5 at device 31.5 on
pci0
pcm0: [GIANT-LOCKED]
pcm0: <Cirrus Logic CS4205 AC97 Codec>
```

De status van de geluidskaart kan gecontroleerd worden via het bestand /dev/sndstat:

```
# cat /dev/sndstat
FreeBSD Audio Driver (newpcm)
Installed devices:
pcm0: <Intel ICH3 (82801CA)> at io 0xd800, 0xdc80 irq 5 bufsz 16384
kld snd_ich (1p/2r/0v channels duplex default)
```

De uitvoer kan per systeem wat verschillen. Als er geen apparaten pcm genoemd worden, dienen eerdere stappen herzien te worden. Bekijk nogmaals de instellingen van de kernel en bevestig dat het juiste apparaatstuurprogramma was gekozen. Veel voorkomende problemen staan beschreven in Bekende problemen.

Als het goed is werkt de geluidskaart nu. Als pinnen voor audio-out van de CD-ROM- of DVD-ROM-drive juist zijn aangesloten op de geluidskaart, dan kan er een CD in de drive gestopt worden en kan deze met cdcontrol(1) afgespeeld worden:

```
% cdcontrol -f /dev/acd0 play 1
```

Applicaties als audio/workman kunnen een vriendelijker interface bieden. Wellicht is het handig om een applicatie als audio/mpg123 te installeren om naar MP3 audiobestanden te luisteren.

Een snelle manier om de kaart te testen is het als volgt sturen van gegevens naar /dev/dsp:

```
% cat bestandsnaam > /dev/dsp
```

bestandsnaam kan ieder bestand zijn. Deze commandoregel hoort wat ruis te maken, waardoor wordt bevestigd dat de geluidskaart echt werkt.



De appparaat nodes /dev/dsp* worden automatisch aangemaakt wanneer dat nodig is. Als deze niet worden gebruikt, bestaan ze niet en zullen ze niet terugkomen in de terugkoppeling van ls(1).

Niveaus voor de geluidskaartmixer kunnen aangepast worden met het commando mixer(8). Er staan meer details in mixer(8).

Bekende problemen

Fout	Oplossing
`sb_dspwr(XX) timed out`	De I/O poort is niet correct ingesteld.
`bad irq XX`	Het IRQ is niet correct ingesteld. Zorg dat het ingestelde IRQ en het IRQ voor het geluid hetzelfde zijn.
`xxx: gus pcm not attached, out of memory`	Er is niet genoeg geheugen beschikbaar om het apparaat te gebruiken.
`xxx: can't open /dev/dsp!`	Controleer `fstat | grep dsp` of een ander programma het apparaat geopend heeft. Bekende probleemgevallen zijn esound en KDE's geluidsondersteuning.

Een ander euvel is dat moderne grafische kaarten voor het gebruik van HDMI en dergelijken vaak zijn uitgerust met hun eigen geluidsstuurprogramma. Dit geluidsapparaat wordt soms opgesomd voor het eigenlijke geluidskaart en daardoor wordt deze niet gebruikt als het standaard afspeelapparaat. Om te zien of dit het geval is, kan dmesg worden gedraaid en gezocht worden naar `pcm`. De uitvoer ziet er ongeveer als volgt uit:

```
...
hdac0: HDA Driver Revision: 20100226_0142
hdac1: HDA Driver Revision: 20100226_0142
hdac0: HDA Codec #0: NVidia (Unknown)
hdac0: HDA Codec #1: NVidia (Unknown)
hdac0: HDA Codec #2: NVidia (Unknown)
hdac0: HDA Codec #3: NVidia (Unknown)
pcm0: <HDA NVidia (Unknown) PCM #0 DisplayPort> at cad 0 nid 1 on hdac0
pcm1: <HDA NVidia (Unknown) PCM #0 DisplayPort> at cad 1 nid 1 on hdac0
pcm2: <HDA NVidia (Unknown) PCM #0 DisplayPort> at cad 2 nid 1 on hdac0
pcm3: <HDA NVidia (Unknown) PCM #0 DisplayPort> at cad 3 nid 1 on hdac0
hdac1: HDA Codec #2: Realtek ALC889
pcm4: <HDA Realtek ALC889 PCM #0 Analog> at cad 2 nid 1 on hdac1
pcm5: <HDA Realtek ALC889 PCM #1 Analog> at cad 2 nid 1 on hdac1
pcm6: <HDA Realtek ALC889 PCM #2 Digital> at cad 2 nid 1 on hdac1
pcm7: <HDA Realtek ALC889 PCM #3 Digital> at cad 2 nid 1 on hdac1
...
```

Hier is de grafische kaart (`NVidia`) opgesomd voor de geluidskaart (`Realtek ALC889`). Om de geluidskaart als standaard afspeelapparaat te gebruiken, dient `hw.snd.default_unit` veranderd te worden in de eenheid dat voor afspelen gebruikt moet worden:

```
# sysctl hw.snd.default_unit=n
```

Hier is `n` het nummer van het geluidsapparaat wat gebruikt dient te worden, in dit voorbeeld `4`. U kunt deze verandering permanent maken door de volgende regel aan /etc/sysctl.conf toe te voegen:

```
hw.snd.default_unit=4
```

7.2.3. Meerdere geluidsbronnen gebruiken

Het is vaak wenselijk om meerdere geluidsbronnen tegelijkertijd af te kunnen spelen, zoals wanneer esound of artsd het delen van een geluidsapparaat met een andere applicatie niet ondersteunen.

Met FreeBSD kan dit met *Virtuele Geluidskanalen*, die aangezet kunnen worden met de faciliteit sysctl(8). Met virtuele kanalen kunnen het afspelen van een geluidskaart gemultiplext worden door het geluid in de kernel te mixen.

Het aantal virtuele kanalen kan met drie sysctl knoppen als `root` als volgt ingesteld worden:

```
# sysctl dev.pcm.0.play.vchans=4
# sysctl dev.pcm.0.rec.vchans=4
# sysctl hw.snd.maxautovchans=4
```

In het bovenstaande voorbeeld worden vier virtuele kanalen toegewezen, wat in het dagelijks gebruik voldoende is. Zowel `dev.pcm.0.play.vchans=4` als `dev.pcm.0.rec.vchans=4` zijn het aantal virtuele kanalen dat pcm0 heeft voor afspelen en opnemen, en zijn instelbaar als een apparaat is aangesloten. In `hw.snd.maxautovchans` staat het aantal virtuele kanalen dat aan een nieuw audio-apparaat wordt gegeven als het wordt aangesloten met kldload(8). Omdat de module pcm onafhankelijk van de hardware stuurprogramma's geladen kan worden, kan in `hw.snd.maxautovchans` opgeslagen worden hoeveel virtuele kanalen apparaten die later worden aangesloten krijgen. Voor meer informatie wordt naar pcm(4) verwezen.



Het aantal virtuele kanalen voor een apparaat kan niet gewijzigd worden als het in gebruik is. Sluit eerst alle programma's die het apparaat gebruiken, zoals muziekspelers of geluidsdaemons.

Het juiste pcm apparaat zal automatisch en transparant gealloceerd worden voor programma's die /dev/dsp0 aanroepen.

7.2.4. Standaardwaarden voor mixerkanalen instellen

De standaardwaarden voor de mixerkanalen zijn ingesteld in de broncode van het stuurprogramma pcm(4). Er zijn vele applicaties en daemons waarmee waarden voor de mixer ingesteld en onthouden kunnen worden en iedere keer bij het starten weer kunnen worden ingesteld, maar dit is geen nette oplossing. Het is mogelijk om de standaardwaarden in te stellen op het niveau van het stuurprogramma - dit wordt bereikt door de gewenste waarden in te stellen in /boot/device.hints, bijvoorbeeld:

```
hint.pcm.0.vol="50"
```

Met de bovenstaande instelling wordt het volume van een kanaal standaard op 50 ingesteld bij het laden van de module pcm(4).

7.3. MP3 audio

Met MP3 (MPEG Layer 3 Audio) kan geluid bijna in CD-kwaliteit weergegeven worden en dus is er een goede reden om dit vooral niet na te laten op een FreeBSD werkstation.

7.3.1. MP3 spelers

Verreweg de meest populaire X11 MP3 speler is XMMS (X Multimedia Systeem). In XMMS kunnen Winamp skins gebruikt worden, omdat de GUI vrijwel gelijk is aan die van Nullsoft's Winamp. XMMS heeft ook een eigen plug-in ondersteuning.

XMMS kan geïnstalleerd worden via de multimedia/xmms port of pakket.

De interface van XMMS is intuïtief met een afspeellijst, grafische equalizer en meer. Gebruikers die bekend zijn met Winamp vinden XMMS vast eenvoudig te gebruiken.

De port audio/mpg123 is een alternatieve MP3-speler die gebruik maakt van de commandoregel.

mpg123 werkt door het geluidsapparaat en het MP3-bestand aan te geven op de commandoregel. Aangenomen dat uw audio-apparaat /dev/dsp1.0 is en u het MP3-bestand *Foobar-GreatestHits.mp3* wilt afspelen, zou u het volgende opgeven:

```
# mpg123 -a /dev/dsp1.0 Foobar-GreatestHits.mp3
High Performance MPEG 1.0/2.0/2.5 Audio Player for Layer 1, 2 and 3.
Version 0.59r (1999/Jun/15). Written and copyrights by Michael Hipp.
Uses code from various people. See 'README' for more!
THIS SOFTWARE COMES WITH ABSOLUTELY NO WARRANTY! USE AT YOUR OWN RISK!

Playing MPEG stream from Foobar-GreatestHits.mp3 ...
MPEG 1.0 layer III, 128 kbit/s, 44100 Hz joint-stereo
```

7.3.2. CD audio tracks rippen

Voordat een CD of een CD track naar MP3 ge-encodeerd kan worden moeten de audiogegevens naar de harde schijf geript worden. Dit gaat door de ruwe CDDA (CD Digital Audio) gegevens naar WAV-bestanden te kopiëren.

Het hulpprogramma `cdda2wav`, dat onderdeel is van de suite sysutils/cdrtools, kan gebruikt worden om audio-informatie en de daarbij behorende informatie van CD's te rippen.

Als de audio CD in de drive zit, kan het volgende commando als `root` uitgevoerd worden om een hele CD naar individuele (per track) WAV-bestanden te rippen:

```
# cdda2wav -D 0,1,0 -B
```

cdda2wav ondersteunt ATAPI (IDE) CD-ROM-drives. Om van een IDE drive te rippen, dient de apparaatnaam aangegeven te worden in plaats van de SCSI eenheidsnummers. Om bijvoorbeeld track 7 van een IDE drive te rippen:

```
# cdda2wav -D /dev/acd0 -t 7
```

De optie `-D 0,1,0` geeft het SCSI apparaat 0,1,0 aan, dat overeenkomt met de uitvoer van `cdrecord -scanbus`.

Om individuele tracks te rippen kan gebruik gemaakt worden van de optie `-t`:

```
# cdda2wav -D 0,1,0 -t 7
```

In het bovenstaande voorbeeld wordt track 7 van de audio CD geript. Om een reeks tracks te rippen, bijvoorbeeld van 1 tot 7, kan een reeks opgegeven worden:

```
# cdda2wav -D 0,1,0 -t 1+7
```

Ook het hulpprogramma dd(1) kan gebruikt worden om audio tracks van ATAPI drives af te halen. Deze mogelijkheid wordt beschreven in Audio-CD's dupliceren.

7.3.3. MP3's encoderen

Tegenwoordig is *de* MP3 encoder lame. Lame staat in audio/lame in de portsstructuur.

Met de geripte WAV-bestanden converteert het volgende commando audio01.wav naar audio01.mp3:

```
# lame -h -b 192 \
--tt "Foo Titel" \
--ta "FooBar Artiest" \
--tl "FooBar Album" \
--ty "2005" \
--tc "Geript en encoded door Foo" \
--tg "Genre" \
audio01.wav audio01.mp3
```

192 kbits lijkt de standaard bitrate voor MP3 te zijn. Het is ook mogelijk 128 of 160 of andere bitrates te gebruiken. Hoe hoger de bitrate, hoe meer schijfruimte de uiteindelijke MP3-bestanden gebruiken, maar ook de kwaliteit wordt dan hoger. Met de optie `-h` wordt de modus "hogere kwaliteit, maar iets langzamer" ingeschakeld. Met de opties vanaf `--t` worden de ID3 tags ingegeven, die meestal informatie over een nummer bevatten en onderdeel uitmaken van het MP3-bestand. In de hulppagina voor lame staan nog meer opties die gebruikt kunnen worden bij het encoderen beschreven.

7.3.4. MP3's decoderen

Om een CD te kunnen branden van MP3's, moeten ze omgezet worden naar een niet gecomprimeerd WAV-formaat. Zowel XMMS als mpg123 ondersteunen de uitvoer van MP3 naar een niet gecomprimeerd bestandsformaat.

Naar schijf schrijven met XMMS:

1. Start XMMS;
2. Klik rechts op het venster om het XMMS menu te zien;
3. Selecteer `Preference` onder `Options`;
4. Wijzig de Output Plugin naar "Disk Writer Plugin";
5. Klik `Configure`;
6. Voer een map in (of kies "browse") waar de ongecomprimeerde bestanden naar toe geschreven moeten worden;
7. Laad de MP3-bestanden zoals gewoonlijk in XMMS, met het volume op 100% en de EQ instellingen uitgeschakeld;
8. Klik `Play`. XMMS lijkt nu de MP3 af te spelen, maar er is geen muziek te horen. Nu wordt feitelijk de MP3 afgespeeld naar een bestand;
9. Zorg ervoor dat de standaard Output Plugin wordt teruggezet naar hoe de instellingen waren om weer naar MP3's te kunnen luisteren.

Schrijven naar stdout vanuit mpg123:

1. Voer `mpg123 -s audio01.mp3 > audio01.pcm` uit.

XMMS schrijft een bestand in het WAV-formaat, terwijl mpg123 de MP3 converteert naar ruwe PCM audio data. Beide formaten kunnen gebruikt worden met cdrecord om audio CD's te maken. Met burncd(8) moeten ruwe PCM-bestanden gebruikt worden. Als er WAV-bestanden worden gebruikt, is er een tikgeluid te horen bij het begin van iedere track. Dit is het geluid van de kop van ieder WAV-bestand. Met het hulpprogramma SoX kan de kop van WAV-bestanden verwijderd worden. Dit programma kan geïnstalleerd worden met de port of pakket audio/sox

```
% sox -t wav -r 44100 -s -w -c 2 track.wav track.raw
```

In Optische media (CD's) aanmaken en gebruiken staat meer informatie over het gebruiken van een CD-brander in FreeBSD.

7.4. Video afspelen

Video afspelen is een relatief nieuwe en zich snel ontwikkelende richting voor applicaties. In tegenstelling tot voor audio werkt alles hier niet zo soepel.

Voor er wordt begonnen is het van belang te weten welk model videokaart zich in een systeem bevindt en welke chip die gebruikt. Hoewel Xorg vele videokaarten ondersteunt, zijn er veel minder geschikt om goed video mee af te spelen. Er kan een lijst met ondersteunde extensies getoond worden voor X server met de gebruikte videokaart door het commando xdpyinfo(1) uit te voeren terwijl X11 draait.

Het is verstandig een kort MPEG-bestand beschikbaar te hebben dat gebruikt kan worden als testbestand voor het evalueren van de spelers en hun opties. Omdat sommige DVD-spelers standaard zoeken naar DVD media in /dev/dvd of deze apparaatnaam standaard in de broncode hebben staan, is het wellicht verstandig om een symbolische link te maken naar de juiste apparaten:

```
# ln -sf /dev/acd0 /dev/dvd
# ln -sf /dev/acd0 /dev/rdvd
```

Vanwege de werking van devfs(5), blijven handmatig aangemaakte links niet bestaan als een systeem wordt herstart. Om automatisch symbolische links aan te laten maken als een systeem start, kunnen de volgende regels toegevoegd worden aan /etc/devfs.conf:

```
link acd0 dvd
link acd0 rdvd
```

Daarnaast zijn voor het decoderen van DVD, waarvoor bijzondere DVD-ROM functies aangeroepen worden, schrijfrechten op de DVD-apparaten nodig.

Om de gedeeld-geheugeninterface van X11 te verbeteren, wordt aangeraden dat een aantal variabelen van sysctl(8) worden verhoogd:

```
kern.ipc.shmmax=67108864
kern.ipc.shmall=32768
```

7.4.1. Videomogelijkheden vaststellen

Er zijn een aantal methoden om video weer te geven onder X11. Welke echt werkt, is voornamelijk afhankelijk van de gebruikte hardware. Iedere hieronder beschreven methode geeft andere resultaten op andere hardware. De laatste tijd krijgt het renderen van video in X11 veel aandacht en bij iedere versie van Xorg kan er een aanzienlijke verbetering zijn.

Een lijst van veel gebruikte video-interfaces:

1. X11: normale X11 uitvoer met gebruikmaking van gedeeld geheugen;
2. XVideo: een uitbreiding op de X11 interface die video in een door X11 getekend object ondersteunt;
3. SDL: de Simple Directmedia Layer;
4. DGA: de Direct Graphics Access;

5. SVGAlib: low level console grafische laag.

XVideo

Xorg kent een uitbreiding *XVideo*, ook bekend als Xvideo, Xv of xv, waarmee video direct weergegeven kan worden in getekende objecten door een speciale versneller. Deze uitbreiding geeft een goede afspeelkwaliteit, zelfs op machines met mindere specificaties.

Of de uitbreiding actief is, kan gecontroleerd worden met het commando `xvinfo`:

```
% xvinfo
```

XVideo wordt ondersteund als de uitvoer er ongeveer als volgt uitziet:

```
X-Video Extension version 2.2
screen #0
  Adaptor #0: "Savage Streams Engine"
    number of ports: 1
    port base: 43
    operations supported: PutImage
    supported visuals:
      depth 16, visualID 0x22
      depth 16, visualID 0x23
    number of attributes: 5
      "XV_COLORKEY" (range 0 to 16777215)
              client settable attribute
              client gettable attribute (current value is 2110)
      "XV_BRIGHTNESS" (range -128 to 127)
              client settable attribute
              client gettable attribute (current value is 0)
      "XV_CONTRAST" (range 0 to 255)
              client settable attribute
              client gettable attribute (current value is 128)
      "XV_SATURATION" (range 0 to 255)
              client settable attribute
              client gettable attribute (current value is 128)
      "XV_HUE" (range -180 to 180)
              client settable attribute
              client gettable attribute (current value is 0)
    maximum XvImage size: 1024 x 1024
    Number of image formats: 7
      id: 0x32595559 (YUY2)
        guid: 59555932-0000-0010-8000-00aa00389b71
        bits per pixel: 16
        number of planes: 1
        type: YUV (packed)
      id: 0x32315659 (YV12)
        guid: 59563132-0000-0010-8000-00aa00389b71
        bits per pixel: 12
```

```
        number of planes: 3
        type: YUV (planar)
      id: 0x30323449 (I420)
        guid: 49343230-0000-0010-8000-00aa00389b71
        bits per pixel: 12
        number of planes: 3
        type: YUV (planar)
      id: 0x36315652 (RV16)
        guid: 52563135-0000-0000-0000-000000000000
        bits per pixel: 16
        number of planes: 1
        type: RGB (packed)
        depth: 0
        red, green, blue masks: 0x1f, 0x3e0, 0x7c00
      id: 0x35315652 (RV15)
        guid: 52563136-0000-0000-0000-000000000000
        bits per pixel: 16
        number of planes: 1
        type: RGB (packed)
        depth: 0
        red, green, blue masks: 0x1f, 0x7e0, 0xf800
      id: 0x31313259 (Y211)
        guid: 59323131-0000-0010-8000-00aa00389b71
        bits per pixel: 6
        number of planes: 3
        type: YUV (packed)
      id: 0x0
        guid: 00000000-0000-0000-0000-000000000000
        bits per pixel: 0
        number of planes: 0
        type: RGB (packed)
        depth: 1
        red, green, blue masks: 0x0, 0x0, 0x0
```



Sommige van de weergegeven formaten (YUV2, YUV12, enzovoort) zijn niet in iedere implementaties van XVideo beschikbaar en hun afwezigheid kan sommige spelers hinderen.

Als het resultaat er als hieronder uitziet, is er geen ondersteuning voor XVideo aanwezig op de videokaart in een systeem:

```
X-Video Extension version 2.2
screen #0
no adaptors present
```

Als XVideo voor een kaart niet wordt ondersteund, dan betekent dat alleen dat het lastiger wordt om op een beeldscherm aan de vereisten voor het renderen van video te voldoen. Afhankelijk van de videokaart en de processor kan het toch nog mogelijk zijn om acceptabele prestaties neer te

zetten. In Verder lezen staan verwijzingen naar leesvoer over mogelijkheden voor het verbeteren van prestaties.

Eenvoudige Directmedia Laag

De Eenvoudige Directmedia Laag (Simple Directmedia Layer), SDL, is een porting-laag voor vele besturingssystemen waardoor cross-platform toepassingen kunnen worden ontwikkeld die efficiënt gebruik maken van geluid en beelden. De SDL laag biedt een abstractie op laag niveau naar de hardware die soms efficiënter kan zijn dan de X11 interface.

De SDL staat in devel/sdl12.

Directe Grafische Toegang

Directe Grafische Toegang (Direct Graphics Access) is een X11 uitbreiding die een programma in staat stelt voorbij te gaan aan de X server en de framebuffer direct kan wijzigen. Omdat hij afhankelijk is van geheugenmapping op een laag niveau om dit delen uit te voeren, moeten programma's die er gebruik van maken als `root` draaien.

De DGA uitbreiding kan getest en gebenchmarkt worden met dga(1). Als `dga` draait, verandert het de kleuren op een scherm als er een toets wordt ingedrukt. Om te stoppen kan de toets `q` gebruikt worden.

7.4.2. Ports en pakketten met video

In dit onderdeel wordt de software die vanuit de FreeBSD Portscollectie beschikbaar is voor het afspelen van video beschreven. Het afspelen van video is een tak van softwareontwikkeling die erg in beweging is en de mogelijkheden van de verschillende applicaties verschillen zeer waarschijnlijk van wat hier is beschreven.

Als eerste is het belangrijk om te weten dat veel applicaties die met video te maken hebben en op FreeBSD draaien ontwikkeld zijn als Linux® applicaties. Veel van die applicaties zijn op het moment van schrijven van beta-kwaliteit. Problemen die te verwachten zijn bij het gebruik van de beschreven videopakketten op FreeBSD zijn:

1. Een applicatie kan geen bestanden afspelen die zijn gemaakt met een andere applicatie;
2. Een applicatie kan geen bestanden afspelen die met de applicatie zelf zijn gemaakt;
3. Dezelfde applicatie, op twee verschillende machines gebouwd, speelt hetzelfde bestand op twee machines anders af;
4. Een ogenschijnlijk triviale filter, zoals het herschalen van beeldgrootte, kan resulteren in vreselijk vervelende artefacten door fouten in de routine voor het herschalen;
5. Een applicatie dumpt zijn core regelmatig;
6. Documentatie wordt niet geïnstalleerd bij de port en staat op het web of in de map work van de port.

Veel van deze applicaties kunnen ook "Linux-ismes" vertonen. Zo kunnen er bijvoorbeeld problemen ontstaan door de wijze waarop standaard bibliotheken zijn geïmplementeerd in de Linux® distributies of een aantal van de mogelijkheden van de Linux®-kernel, waarvan door de

makers van de applicatie wordt aangenomen dat ze aanwezig zijn. Dit soort problemen zijn niet altijd zichtbaar en er wordt ook omheen gewerkt door de beheerders van ports, wat tot de volgende mogelijke problemen kan leiden:

1. Het gebruik van /proc/cpuinfo om processorkarakteristieken uit te lezen;
2. Het verkeerd gebruiken van threads, waardoor een programma hangt als het klaar is, in plaats van dat het echt eindigt;
3. Software die nog niet in de FreeBSD Portscollectie zit en vaak gebruikt wordt samen met een applicatie die daar wel onderdeel van uitmaakt.

Tot nu toe is gebleken dat de ontwikkelaars van applicaties wel coöperatief waren met de beheerders van ports om zo het aantal work-arounds dat nodig was voor het overzetten tot een minimum te beperken.

MPlayer

MPlayer is een zich snel ontwikkelende videospeler. De doelen van het MPlayer-team zijn snelheid en flexibiliteit onder Linux® en andere Unices. Het project is gestart toen de oprichter van het team genoeg had van de slechte afspeelprestaties van de destijds beschikbare spelers. Er zijn mensen die zeggen dat het grafische ontwerp is opgeofferd voor het stroomlijnen van het ontwerp, maar het blijkt dat, als een gebruiker gewend is aan de commandoregelopties en de toetsencommando's, de applicatie erg goed werkt.

MPlayer bouwen

MPlayer staat in multimedia/mplayer. MPlayer voert een aantal hardwarecontroles uit tijdens het bouwen, wat resulteert in een binair bestand dat niet van het ene naar het andere systeem verplaatst kan worden. Daarom is het van belang dat het uit de ports wordt gebouwd en niet als binair pakket wordt geïnstalleerd. Daarnaast staan er ook nog opties die vanaf de `make` commandoregel meegegeven kunnen worden beschreven in de Makefile en aan het begin van de build:

```
# cd /usr/ports/multimedia/mplayer
# make
N - O - T - E

Take a careful look into the Makefile in order
to learn how to tune mplayer towards you personal preferences!
For example,
make WITH_GTK1
builds MPlayer with GTK1-GUI support.
If you want to use the GUI, you can either install
/usr/ports/multimedia/mplayer-skins
or download official skin collections from
http://www.mplayerhq.hu/homepage/dload.html
```

De standaard portopties zijn voor de meeste gebruikers voldoende. Maar als bijvoorbeeld de XviD codec nodig is, dan moet de optie `WITH_XVID` op de commandoregel meegegeven worden. Het

standaard DVD-apparaat kan ook gedefinieerd worden met de optie `WITH_DVD_DEVICE`, waarbij standaard /dev/acd0 wordt gebruikt.

Op het moment van schrijven wordt de MPlayer port gebouwd met de HTML documentatie en twee uitvoerbare bestanden, `mplayer` en `mencoder`, wat een hulpmiddel is voor het opnieuw encoderen van video.

De HTML documentatie voor MPlayer is erg informatief. Als de lezer vindt dat er informatie over videohardware en interfaces in dit hoofdstuk mist, dan is de documentatie van MPlayer een zeer grondige aanvulling. Het is de moeite waard de tijd te nemen om de documentatie van MPlayer te lezen, als meer informatie over de ondersteuning van video in UNIX® welkom is.

MPlayer gebruiken

Iedere gebruiker van MPlayer dient een submap .mplayer in zijn thuismap te hebben. Die kan als volgt gemaakt worden:

```
% cd /usr/ports/multimedia/mplayer
% make install-user
```

De commando-opties voor `mplayer` staan in de hulppagina. Nog meer details staan in de HTML documentatie. In dit onderdeel worden slechts een aantal gebruiksmogelijkheden beschreven.

Om een bestand als testbestand.avi af te spelen met een van de beschikbare video-interfaces, kan de optie `-vo` gebruikt worden:

```
% mplayer -vo xv testbestand.avi
```

```
% mplayer -vo sdl testbestand.avi
```

```
% mplayer -vo x11 testbestand.avi
```

```
# mplayer -vo dga testbestand.avi
```

```
# mplayer -vo 'sdl:dga' testbestand.avi
```

Het is de moeite waard alle bovenstaande opties uit te proberen omdat hun relatieve prestatie afhangt van vele factoren die aanzienlijk verschillen tussen hardware.

Om een DVD af te spelen dient testbestand.avi vervangen te worden door `dvd://N -dvd-device APPARAAT` waar *N* het titelnummer is dat afgespeeld moeten worden en APPARAAT het apparaatknooppunt is voor de DVD-ROM. Om bijvoorbeeld titel 3 van /dev/dvd af te spelen:

```
# mplayer -vo xv dvd://3 -dvd-device /dev/dvd
```



Het standaard DVD-apparaat kan ingesteld worden bij het bouwen van de MPlayer port met de optie `WITH_DVD_DEVICE`. Standaard is dit apparaat /dev/acd0. Meer details staan in de Makefile van de port.

Om te stoppen, pauzeren, verder te spoelen, enzovoort, kunnen de toetsendefinities gebruikt worden, die in te zien zijn door `mplayer -h` uit te voeren of de hulppagina te lezen.

Overige belangrijke opties voor het afspelen zijn: `-fs -zoom`, waarmee het volledige scherm wordt gebruikt, en `-framedrop`, die prestatieverhogend werkt.

Om ervoor te zorgen dat de commandoregels niet te lang worden, kan het bestand .mplayer/config met voorkeursinstellingen gemaakt worden:

```
vo=xv
fs=yes
zoom=yes
```

Tenslotte kan `mplayer` gebruikt worden om een DVD naar een bestand van het type .vob te rippen. Om de tweede titel van een DVD de dumpen kan het volgende commando gebruikt worden:

```
# mplayer -dumpstream -dumpfile out.vob dvd://2 -dvd-device /dev/dvd
```

Het uitvoerbestand out.vob, is van het type MPEG en kan bewerkt worden met andere in dit onderdeel besproken programma's.

`mencoder`

Voordat `mencoder` wordt gebruikt, is het verstandig de opties uit de HTML-documentatie te bekijken. Er is een hulppagina, maar die is niet echt bruikbaar zonder de HTML-documentatie. Er zijn ontelbare mogelijkheden om de kwaliteit te verhogen, de bitrate te verlagen en formaten te wijzigen en een aantal van die truuks maken het verschil tussen goede en slechte prestaties. Hieronder staan een aantal voorbeelden beschreven. Eerst een eenvoudige kopie:

```
% mencoder invoer.avi -oac copy -ovc copy -o uitvoer.avi
```

Verkeerde combinaties van commandoregelopties kunnen resulteren in uitvoerbestanden die zelfs niet af te spelen zijn door `mplayer`. Daarom wordt aangeraden om het bij de optie `-dumpfile` in `mplayer` te houden als het alleen maar nodig is een bestand te rippen.

Om invoer.avi te converteren naar de MPEG4-codec met MPEG3-audio encodering (audio/lame is verplicht):

```
% mencoder invoer.avi -oac mp3lame -lameopts br=192 \
```

```
-ovc lavc -lavcopts vcodec=mpeg4:vhq -o utvoer.avi
```

Hiermee wordt uitvoer gemaakt die af te spelen is met `mplayer` en `xine`.

invoer.avi kan worden vervangen door `dvd://1 -dvd-device /dev/dvd` en als `root` gedraaid worden om een DVD-titel direct te hercoderen. Omdat het waarschijnlijk is dat de eerste experimenten niet direct tevredenstellend zijn, wordt aangeraden een titel eerst naar een bestand te dumpen en dat als werkbestand te gebruiken.

xine videospeler

De xine videospeler is een project met een brede scope, dat niet alleen tracht een allesomvattende video-oplossing te bieden, maar ook probeert een herbruikbare basisbibliotheek en een modulair uitvoerbaar bestand te maken dat uitgebreid kan worden met plug-ins. Het kan als pakket en port geïnstalleerd worden uit multimedia/xine.

De xine speler heeft nog wat ruwe randjes, maar is zeker goed van start gegaan. In de praktijk heeft xine een snelle CPU met een snelle videokaart of ondersteuning voor de XVideo extensie nodig. De GUI is bruikbaar, maar wat onhandig.

Op het moment van schrijven wordt er geen invoermodule bij xine geleverd waarmee CSS gecodeerde DVD's afgespeeld kunnen worden. Er zijn er die door andere partijen zijn gebouwd die dat type modules wel hebben, maar die zijn niet beschikbaar in de FreeBSD Portscollectie.

Vergeleken met MPlayer, doet xine meer voor de gebruiker, maar tegelijkertijd neemt het wat van de fijnafstellingsmogelijkheden weg. De videospeler xine werkt het beste op XVideo-interfaces.

Standaard start de xine speler op in een grafische gebruikersinterface. Via het menu kan een specifiek bestand geopend worden:

```
% xine
```

Het is ook mogelijk om zonder de GUI direct een bestand af te laten spelen:

```
% xine -g -p mijnfilm.avi
```

transcode hulpprogramma's

De software transcode is geen speler, maar een verzameling hulpprogramma's voor het hercoderen van video- en audiobestanden. Met transcode wordt het mogelijk om videobestanden samen te voegen, kapotte bestanden te repareren en commandoregelprogramma's te gebruiken met stdin/stdout stream interfaces.

Tijdens het bouwen van de port multimedia/transcode kan een groot aantal opties opgegeven worden en de volgende commandoregel wordt geadviseerd om transcode te bouwen:

```
# make WITH_OPTIMIZED_CFLAGS=yes WITH_LIBA52=yes WITH_LAME=yes WITH_OGG=yes \
```

```
WITH_MJPEG=yes -DWITH_XVID=yes
```

De geadviseerde instellingen zijn toereikend voor de meeste gebruikers.

Om de mogelijkheden van `transcode` te illustreren volgt nu een voorbeeld van hoe een DivX-bestand om te zetten in een PAL MPEG-1-bestand (PAL VCD):

```
% transcode -i invoer.avi -V --export_prof vcd-pal -o uitvoer_vcd
% mplex -f 1 -o uitvoer_vcd.mpg uitvoer_vcd.m1v uitvoer_vcd.mpa
```

Het resulterende MPEG-bestand, uitvoer_vcd.mpg, is klaar om afgespeeld te worden met MPlayer. Het kan ook op een CD-R gebrand worden om er een Video-CD mee te maken. In dat geval is het nodig om de programma's multimedia/vcdimager en sysutils/cdrdao te installeren.

Er is een hulppagina voor `transcode`, maar kijk ook op transcode wiki voor meer informatie en voorbeelden.

Als de twee vergeleken worden, draait `transcode` aanzienlijk langzamer dan `mencoder`, maar is de kans wel groter dat er een bestand uit komt dat op de meeste spelers afgespeeld kan worden. MPEG-bestanden die met `transcode` zijn gemaakt, zijn bijvoorbeeld al afgespeeld op Windows Media® Player en Apple's Quicktime®.

7.4.3. Verder lezen

De beschikbare videosoftware pakketten voor FreeBSD zijn fors in ontwikkeling. Het is goed mogelijk dat in de nabije toekomst de meeste problemen die hier aan de kaak zijn gesteld, zijn opgelost. Intussen kunnen zij die het hoogst haalbare uit de A/V mogelijkheden voor FreeBSD willen halen, dat het beste doen door wat beschikbaar is bij elkaar te scharrelen uit de beschikbare FAQ's and tutorials en meerdere programma's gebruiken. Het doel van deze paragraaf is de lezer wat richting te geven op dat vlak.

De MPlayer documentatie is technisch erg informatief. Deze documenten kunnen het beste bekeken worden door iemand die veel kennis wil opdoen over video in UNIX®. Op de MPlayer mailinglijst wordt het niet op prijs gesteld als iemand de documentatie niet heeft gelezen, dus het is verstandig RTFM in gedachten te houden alvorens bug rapportages naar ze te mailen.

De xine HOWTO bevat een hoofdstuk over het verbeteren van prestaties, dat op alle spelers van toepassing is.

Tenslotte zijn er nog een aantal veelbelovende applicaties die het proberen waard zijn:

- Avifile bestaat ook als port: multimedia/avifile;
- Ogle is er ook als port: multimedia/ogle;
- Xtheater;
- multimedia/dvdauthor, een open source pakket voor authoring van DVD content.

7.5. TV-kaarten installeren

7.5.1. Inleiding

Met TV-kaarten is het mogelijk om naar (kabel)uitzendingen te kijken op een computer. Op de meeste kaarten kan composiet video aangeleverd worden via een RCA of S-video input en sommige kaarten hebben ook een FM tuner.

FreeBSD biedt ondersteuning voor PCI-gebaseerde TV-kaarten met een Brooktree Bt848/849/878/879 of een Conexant CN-878/Fusion 878a Video Capture Chip met het stuurprogramma bktr(4). Het is van belang dat er op de kaart ook een ondersteunde tuner zit. Hiervoor kan bktr(4) geraadpleegd worden, waarin een lijst met ondersteunde tuners staat.

7.5.2. Stuurprogramma toevoegen

Voordat de kaart gebruikt kan worden, dient het stuurprogramma bktr(4) geladen te worden. Dit kan door de volgende regel aan /boot/loader.conf toe te voegen:

```
bktr_load="YES"
```

Daarnaast is het ook mogelijk om statisch ondersteuning voor de TV-kaart in de kernel te compileren. Dan dient de volgende regel toegevoegd te worden aan de kernelinstellingen:

```
device   bktr
device  iicbus
device  iicbb
device  smbus
```

De extra stuurprogramma's zijn nodig omdat de kaartcomponenten verbonden zijn via een I2C bus. Met deze instellingen kan een nieuwe kernel gebouwd en geïnstalleerd worden.

Als een systeem eenmaal ondersteuning biedt, hoort de TV-kaart ongeveer als volgt bij een herstart getoond te worden:

```
bktr0: <BrookTree 848A> mem 0xd7000000-0xd7000fff irq 10 at device 10.0 on pci0
iicbb0: <I2C bit-banging driver> on bti2c0
iicbus0: <Philips I2C bus> on iicbb0 master-only
iicbus1: <Philips I2C bus> on iicbb0 master-only
smbus0: <System Management Bus> on bti2c0
bktr0: Pinnacle/Miro TV, Philips SECAM tuner.
```

Deze berichten kunnen afwijken, afhankelijk van de gebruikte hardware. Het is van belang te controleren of de tuner juist herkend wordt; er kunnen nog een aantal instellingen gemaakt worden voor parameters met sysctl(8) MIB's en in het kernelinstellingenbestand. Om bijvoorbeeld het gebruik van een Philips SECAM tuner te forceren, kan de volgende regel aan het bestand met kernelinstellingen worden toegevoegd:

```
options OVERRIDE_TUNER=6
```

Dit kan ook via een instelling van sysctl(8):

```
# sysctl hw.bt848.tuner=6
```

In de hulppagina voor bktr(4) en /usr/src/sys/conf/NOTES staan meer details over de beschikbare opties.

7.5.3. Handige programma's

Om een TV-kaart te gebruiken, dient een van de volgende applicaties geïnstalleerd te worden:

- multimedia/fxtv biedt TV-in-een-window en beeld/audio/videocapture mogelijkheden;
- multimedia/xawtv is ook een TV applicatie met dezelfde mogelijkheden als fxtv;
- misc/alevt decodeert Videotext/Teletext en kan deze weergeven;
- audio/xmradio, een applicatie om de FM-tuner die bij sommige TV-kaarten zit te gebruiken;
- audio/wmtune, een handige bureaubladapplicatie voor radiotuners.

Er zijn nog meer applicaties beschikbaar in de Portscollectie.

7.5.4. Problemen oplossen

Bij problemen met een TV-kaart dient eerst gecontroleerd te worden of de videocapture chip en de tuner echt ondersteund worden door het stuurprogramma bktr(4) en of de juiste instellingen worden gebruikt. Voor meer ondersteuning en vragen over een specifieke TV-kaart is het aan te raden de archieven van de FreeBSD multimedia mailinglijst mailinglijst te raadplegen of er contact mee op te nemen.

7.6. MythTV

MythTV is een open-source PVR software project.

Het staat in de Linux®-wereld bekend als een complexe toepassing met veel afhankelijkheden, en daarom moeilijk om te installeren. De Portscollectie van FreeBSD versimpelt veel van het proces, maar sommige componenten moeten handmatig worden geïnstalleerd. Deze sectie is bedoeld om te helpen en te begeleiden in het installeren van MythTV.

7.6.1. Hardware

MythTV is ontworpen om V4L te gebruiken om invoerapparatuur voor video zoals encoders en tuners te benaderen. Momenteel werkt MythTV het beste met USB DVB-S/C/T kaarten die ondersteund worden door multimedia/webcamd omdat webcamd een gebruikerstoepassing levert voor V4L. Elke DVB-kaart die ondersteund wordt door webcamd zou met MythTV moeten werken, een lijst van kaarten waarvan hun werking bekend is kan hier gevonden worden. Er zijn ook

stuurprogramma's bekend voor Hauppauge-kaarten in de pakketten multimedia/pvr250 en multimedia/pvrxxx, maar deze leveren een niet-standaard interface met hun stuurprogramma dat niet werkt met versies van MythTV nieuwer dan 0.23.

HTPC bevat een lijst van alle beschikbare stuurprogramma's voor DVB.

7.6.2. Afhankelijkheden

Doordat MythTV flexibel en modulair is, staat het de gebruiker toe om de voorkant en de achterkant op verschillende machines te hebben.

Voor de voorkant is multimedia/mythtv-frontend nodig, alsook een X-server welke in x11/xorg beschikbaar is. Idealiter beschikt de voorkant-computer ook over een videokaart die XvMC ondersteunt en optioneel over een afstandsbediening die compatibel is met LIRC.

Voor de achterkant is multimedia/mythtv nodig, alsook een MySQL™ database en optioneel een tuner en opslag voor opnames. Het MySQL™ pakket zou automatisch als een afhankelijkheid geïnstalleerd moeten worden tijdens de installatie van multimedia/mythtv.

7.6.3. MythTV installeren

Gebruik de volgende stappen om MythTV te installeren. Installeer als eerste MythTV van de FreeBSD Portscollectie:

```
# cd /usr/ports/multimedia/mythtv
# make install
```

Installeer de database voor MythTV:

```
# mysql -uroot -p < /usr/local/shared/mythtv/database/mc.sql
```

Configureer de achterkant:

```
# mythtv-setup
```

Start de achterkant:

```
# echo 'mythbackend_enable="YES"' >> /etc/rc.conf
# service mythbackend start
```

7.7. Scanners

7.7.1. Inleiding

In FreeBSD is toegang tot scanners mogelijk met SANE (Scanner Access Now Easy) API uit de FreeBSD Portscollectie. SANE gebruikt ook een aantal FreeBSD apparaatstuurprogramma's om toegang te krijgen tot de hardware van de scanner.

FreeBSD ondersteunt SCSI en USB scanners. Het is van belang te controleren of een scanner door SANE wordt ondersteund voordat er instellingen worden gemaakt. SANE heeft een lijst met ondersteunde apparaten waarin gekeken kan worden of een scanner wordt ondersteund en wat de status voor ondersteuning is.

7.7.2. Kernel instellen

Zoals hierboven al is aangegeven, worden zowel SCSI als USB-scanners ondersteund. Afhankelijk van de gebruikte scannerinterface zijn verschillende apparaatstuurprogramma's nodig.

USB-interface

In de GENERIC kernel zitten standaard de apparaatstuurprogramma's die nodig zijn voor ondersteuning van USB-scanners. In het geval wordt besloten tot het maken van een aangepaste kernel, dan dienen de volgende regels in het kernelinstellingenbestand te worden opgenomen:

```
device usb
device uhci
device ohci
device uscanner
device ehci
```

Na een herstart met de juiste kernel kan de USB-scanner aangesloten worden. Een regel die de detectie van uw scanner aangeeft zou in de berichtenbuffer van het systeem (dmesg(8)) moeten verschijnen:

```
ugen0.2: <EPSON> at usbus0
```

Deze berichten geven aan dat de scanner /dev/ugen0.2 als apparaatknooppunt gebruikt. Voor dit voorbeeld was een EPSON Perfection® 1650 USB-scanner gebruikt.

SCSI interface

Als een scanner een SCSI interface heeft, is het belangrijk te weten welk SCSI controllerbord gebruikt gaat worden. Afhankelijk van de gebruikte SCSI chipset, dient het bestand met kernelinstellingen aangepast te worden. De GENERIC kernel ondersteunt de meest voorkomende SCSI controllers. In het bestand NOTES is de juiste instelling te vinden die toegevoegd moet worden aan het bestand met kernelinstellingen. Naast het toevoegen van het juiste SCSI-adapter stuurprogramma, dienen ook de volgende regels opgenomen te worden in het kernelinstellingenbestand:

```
device scbus
device pass
```

Als de kernel juist gecompileerd en geïnstalleerd is, horen de apparaten tijdens het opstarten zichtbaar te zijn in de systeemberichtbuffer:

```
pass2 at aic0 bus 0 target 2 lun 0
pass2: <AGFA SNAPSCAN 600 1.10> Fixed Scanner SCSI-2 device
pass2: 3.300MB/s transfers
```

Als een scanner niet aan staat tijdens het opstarten, is het nog mogelijk handmatig detectie te forceren door de SCSI-bus te laten scannen met camcontrol(8):

```
# camcontrol rescan all
Re-scan of bus 0 was successful
Re-scan of bus 1 was successful
Re-scan of bus 2 was successful
Re-scan of bus 3 was successful
```

In het bovenstaande geval zal de scanner ongeveer als volgt verschijnen in de lijst met SCSI-apparaten:

```
# camcontrol devlist
<IBM DDRS-34560 S97B>              at scbus0 target 5 lun 0 (pass0,da0)
<IBM DDRS-34560 S97B>              at scbus0 target 6 lun 0 (pass1,da1)
<AGFA SNAPSCAN 600 1.10>           at scbus1 target 2 lun 0 (pass3)
<PHILIPS CDD3610 CD-R/RW 1.00>     at scbus2 target 0 lun 0 (pass2,cd0)
```

Meer details over SCSI-apparaten staan in de hulppagina's voor scsi(4) en camcontrol(8).

7.7.3. SANE instellen

Het SANE systeem is opgesplitst in twee delen: de backends (graphics/sane-backends) en de frontends (graphics/sane-frontends). Het deel met de backends zorgt voor de toegang tot de scanner zelf. In de lijst met door SANE ondersteunde apparaten staat welk backend welke scanner(s) ondersteunt. Het is echt nodig het juiste backend vast te stellen, omdat het anders bijzonder lastig wordt een scanner aan de praat te krijgen. Het deel met frontends levert een grafische scaninterface (xscanimage).

De eerste stap is om de port of het pakket graphics/sane-backends te installeren. Daarna kan met het commando `sane-find-scanner` gecontroleerd worden welke scanner er door het SANE systeem is gedetecteerd:

```
# sane-find-scanner -q
```

```
found SCSI scanner "AGFA SNAPSCAN 600 1.10" at /dev/pass3
```

In de uitvoer is te lezen welk type interface en welk apparaatknooppunt worden gebruikt om de scanner met een systeem te verbinden. Het merk en het model worden wellicht niet getoond, maar dat is ook niet echt van belang.



Sommige USB-scanners verlangen dat er firmware wordt geladen. Dit wordt uitgelegd in de hulppagina van het backend. Het is ook van belang sane-find-scanner(1) en linprocfs(7) te lezen.

Hierna kan gecontroleerd worden of de scanner ook te zien is voor een scanner-frontend. Er zit bij de SANE backends een standaard hulpprogramma sane(1). Met dit commando kunnen de apparaten zichtbaar gemaakt worden en kan vanaf de commandoregel gescand worden. Met de optie `-L` kunnen de scannerapparaten getoond worden:

```
# scanimage -L
device `snapscan:/dev/pass3' is a AGFA SNAPSCAN 600 flatbed scanner
```

Of, met bijvoorbeeld de USB-scanner die in USB-interface wordt gebruikt:

```
# scanimage -L
device 'epson2:libusb:/dev/usb:/dev/ugen0.2' is a Epson GT-8200 flatbed scanner
```

Deze uitvoer komt van een FreeBSD 8.X systeem, het item `'epson2:libusb:/dev/usb:/dev/ugen0.2'` geeft de naam van het backend (`epson2`) en het apparaatknooppunt (`/dev/ugen0.2`) dat door onze scanner wordt gebruikt.



De afwezigheid van uitvoer of een bericht dat aangeeft dat er geen scanners zijn aangetroffen, betekent dat sane(1) niet in staat is een scanner te identificeren. Als dit gebeurt, dient het instellingenbestand voor het backend aangepast te worden en dient daar de juiste instelling gemaakt te worden. De map /usr/local/etc/sane.d/ bevat alle bestanden met instellingen voor de backends. Het is bekend dat dit identificatieprobleem optreedt bij bepaalde USB-scanners.

De USB-scanner die in USB-interface wordt gebruikt, wordt in FreeBSD 8.X prima gedetecteerd en werkt daar, maar in eerdere versies van FreeBSD (waar uscanner(4) wordt gebruikt) toont het de volgende informatie met `sane-find-scanner`:

```
# sane-find-scanner -q
found USB scanner (UNKNOWN vendor and product) at device /dev/uscanner0
```

De bovenstaande uitvoer geeft aan dat de scanner juist is gedetecteerd, dat het de USB-interface gebruikt en is aangesloten op het apparaatknooppunt /dev/uscanner0. Nu kan gecontroleerd worden of de scanner juist wordt

geïdentificeerd:

```
# scanimage -L

No scanners were identified. If you were expecting something different,
check that the scanner is plugged in, turned on and detected by the
sane-find-scanner tool (if appropriate). Please read the documentation
which came with this software (README, FAQ, manpages).
```

Omdat in het bovenstaande voorbeeld de scanner niet wordt geïdentificeerd, dient het bestand /usr/local/etc/sane.d/epson2.conf gewijzigd te worden. De gebruikte scanner is een EPSON Perfection® 1650, dus in dit geval dient voor de scanner het backend `epson2` gebruikt te worden. Het is van belang om het commentaar in de instellingenbestanden van de backends te lezen. Het aanpassen van regels is eenvoudig: plaats een commentaarkarakter voor alle regels voor andere interfaces dan die nodig zijn weg (in dit geval worden alle regels die beginnen met het woord `scsi` uitgeschakeld, omdat er een USB-interface wordt gebruiken), en dan kan onderaan het bestand een regel met de gebruikte interface en apparaatknooppunt geplaatst worden:

```
usb /dev/uscanner0
```

Het is aan te raden de opmerkingen te lezen in het bestand met instellingen voor het backend en ook de hulppagina, omdat daarin meer details en de correcte syntaxis te vinden zijn. Nu kan gecontroleerd worden of de scanner wèl juist wordt geïdentificeerd:

```
# scanimage -L
device `epson:/dev/uscanner0' is a Epson GT-8200 flatbed scanner
```

De USB-scanner is geïdentificeerd. Het is niet belangrijk dat het merk en model niet overeenkomen met de scanner. Het belangrijkste is het veld `` `epson:/dev/uscanner0' ``, dat de juiste benamingen voor het backend en het apparaatknooppunt aangeeft.

Als `scanimage -L` in staat is een scanner goed te zien, dan zijn de instellingen compleet. Er kan nu met het apparaat gescand worden.

Hoewel sane(1) in staat is om vanaf de commandoregel te scannen, is het aan te raden beelden te scannen vanuit de grafische gebruikersinterface. SANE heeft een eenvoudige, maar efficiënte grafische interface: xscanimage (graphics/sane-frontends).

Xsane (graphics/xsane) is een ander populair grafisch scanfrontend, dat geavanceerde mogelijkheden biedt, zoals meerdere scanmodi (fotokopie, fax, enzovoort), kleurcorrectie, batchscannen, enzovoort. Beide applicaties zijn als plug-in voor GIMP te gebruiken.

7.7.4. Andere gebruikers toegang tot de scanner geven

Alle voorgaande taken zijn uitgevoerd met `root` rechten, maar het is wellicht ook nodig dat andere gebruikers de scanner kunnen gebruiken. Dan heeft een gebruiker lees- en schrijfrechten nodig op de apparaatknooppunt voor een scanner. Onze USB-scanner gebruikt bijvoorbeeld apparaatknooppunt /dev/ugen0.2 wat in feite slechts een symbolische koppeling is naar het echte apparaatknooppunt genaamd /dev/usb/0.2.0 (een blik op de inhoud van de map /dev bevestigt dit). Zowel de symbolische koppeling als het apparaatknooppunt zijn van respectievelijk de groepen `wheel` en `operator`. Door de gebruiker `joe` aan deze groepen toe te voegen kan hij de scanner zien, maar vanwege duidelijke veiligheidsredenen dient het toevoegen van een gebruiker aan elke groep met zorg te gebeuren, vooral aan de groep `wheel`. Een betere oplossing is om een specifieke groep aan te maken voor het gebruik van USB-apparaten en de scanner toegankelijk te maken voor leden van deze groep.

We zullen dus bijvoorbeeld een groep genaamd `usb` gebruiken. De eerste stap is het aanmaken van deze groep met behulp van het commando pw(8):

```
# pw groupadd usb
```

Hierna moeten we de symbolische koppeling /dev/ugen0.2 aanmaken en het apparaatknooppunt /dev/usb/0.2.0 met de juiste schrijfpermissies toegankelijk maken voor de groep `usb` (`0660` of `0664`), omdat standaard alleen de eigenaar van deze bestanden (`root`) ernaar kan schrijven. Dit alles wordt gedaan door de volgende regels aan /etc/devfs.rules toe te voegen:

```
[system=5]
add path ugen0.2 mode 0660 group usb
add path usb/0.2.0 mode 0666 group usb
```

Nu dienen er alleen nog gebruikers aan de groep `usb` toegevoegd te worden om toegang tot de scanner toe te staan:

```
# pw groupmod usb -m joe
```

Lees voor meer details de handleidingpagina van pw(8).

Hoofdstuk 8. De FreeBSD-kernel instellen

8.1. Samenvatting

De kernel is de kern van het FreeBSD-besturingssysteem en is verantwoordelijk voor het geheugenbeheer, het opleggen van beveiligingsregels, het aansturen van het netwerk, de toegang tot schijven en nog veel meer. Hoewel steeds meer in FreeBSD dynamisch instelbaar wordt, is het af en toe nodig om de kernel opnieuw in te stellen en te compileren.

Na het lezen van dit hoofdstuk weet de lezer:

- Waarom het nodig is om een aangepaste kernel te bouwen;
- Hoe een nieuw kernelinstellingenbestand te schrijven of een bestaand kernelinstellingenbestand aan te passen;
- Hoe het kernelinstellingenbestand te gebruiken om een nieuwe kernel aan te maken en te bouwen;
- Hoe een nieuwe kernel te installeren;
- Hoe problemen op te lossen als er iets verkeerd gaat.

Alle opdrachten die in dit hoofdstuk als voorbeeld zijn gegeven moeten als `root` uitgevoerd worden om te slagen.

8.2. Redenen om een aangepaste kernel te bouwen

Traditioneel heeft FreeBSD zoals dat heet een "monolitische" kernel gehad. Dit betekent dat de kernel één groot programma was, een vaste lijst van apparaten ondersteunde en als het gewenst was om het gedrag van de kernel te veranderen, moest er een nieuwe kernel gecompileerd worden en moest daarna de computer opnieuw gestart worden met de nieuwe kernel.

Vandaag de dag beweegt FreeBSD zich snel naar een model waar veel van de functionaliteit van de kernel in modules zit die dynamisch in en uit de kernel kunnen worden geladen, naargelang dat noodzakelijk is. Dit stelt de kernel in staat om zich aan nieuwe hardware aan te passen die plotseling beschikbaar komt (zoals PCMCIA-kaarten in een laptop) of om nieuwe functionaliteit in zich op te nemen die niet noodzakelijk was toen de kernel oorspronkelijk werd gecompileerd. Dit staat bekend als een modulaire kernel.

Desondanks is het nog steeds nodig om enkele dingen van de kernel statisch in te stellen. In sommige gevallen komt dit doordat de functionaliteit zo diep geworteld zit in de kernel dat het niet dynamisch laadbaar gemaakt kan worden. In andere gevallen kan het simpelweg komen doordat nog niemand de tijd heeft genomen om een dynamisch laadbare kernelmodule voor die functionaliteit te schrijven.

Het bouwen van een aangepaste kernel is een van de meest belangrijke beproevingen die geavanceerde BSD-gebruikers moet doorstaan. Hoewel dit proces veel tijd in beslag neemt, levert het veel voordelen op voor een FreeBSD systeem. In tegenstelling tot de GENERIC-kernel, die vele typen hardware moet ondersteunen, ondersteunt een aangepaste kernel alleen de hardware van de

computer waar hij voor gemaakt is. Dit biedt een aantal voordelen, zoals:

- Een snellere opstarttijd. Aangezien de kernel alleen de hardware zoekt die zich in het systeem bevindt, kan de tijd die het systeem nodig heeft om op te starten aanzienlijk korter worden;
- Minder geheugengebruik. Een aangepaste kernel gebruikt vaak minder geheugen dan de GENERIC-kernel door ongebruikte mogelijkheden en apparaatstuurprogramma's weg te laten. Dit is van belang aangezien de kernelcode altijd in het fysieke geheugen aanwezig blijft, waardoor dit geheugen niet door applicaties gebruikt kan worden. Om deze reden is een aangepaste kernel geknipt voor een systeem met een kleine hoeveelheid RAM;
- Aanvullende hardware-ondersteuning. Een aangepaste kernel kan ingebouwde ondersteuning bieden voor apparaten die zich niet in de GENERIC-kernel bevinden, zoals geluidskaarten.

8.3. De systeemhardware vinden

Alvorens in de kernelconfiguratie te duiken, zou het verstandig zijn om een inventarisatie van de hardware van de machine te maken. In het geval dat FreeBSD niet het primaire besturingssysteem is, kan de inventarisatielijst eenvoudig worden gemaakt door de configuratie van het huidige besturingssysteem te bekijken. De Device Manager van Microsoft® bijvoorbeeld bevat normaliter belangrijke informatie over geïnstalleerde apparaten. De Device Manager bevindt zich in het controlepaneel.



Sommige versies van Microsoft® Windows® hebben een icoon System dat een scherm weer zal geven waarmee Device Manager kan worden benaderd.

Als er geen ander besturingssysteem op de machine staat, moet de beheerder deze informatie handmatig vinden. Eén manier is om de gereedschappen dmesg(8) en man(1) te gebruiken. De meeste apparaatstuurprogramma's van FreeBSD hebben een handleiding, die de ondersteunde hardware noemen, en tijdens het opstarten wordt gevonden hardware getoond. De volgende regels geven bijvoorbeeld aan dat het stuurprogramma voor psm een muis heeft gevonden:

```
psm: <PS/2 Mouse> irq 12 on atkbdc0
psm0: [GIANT-LOCKED]
psm0: [ITHREAD]
psm0: model Generic PS/2 mouse, device ID 0
```

Dit stuurprogramma zal in het eigen kernelinstellingenbestand opgenomen moeten worden of worden geladen met loader.conf(5).

Soms geven de gegevens van `dmesg` alleen de systeemboodschappen weer in plaats van de uitvoer van de opstartonderzoeken. In deze gevallen kan de uitvoer worden verkregen door het bestand /var/run/dmesg.boot te bekijken.

Een andere methode om hardware te vinden is door pciconf(8) te gebruiken welke meer gedetailleerde uitvoer geeft. Bijvoorbeeld:

```
ath0@pci0:3:0:0:        class=0x20000 card=0x058a1014 chip=0x1014168c rev=0x01
```

```
hdr=0x00
    vendor     = 'Atheros Communications Inc.'
    device     = 'AR5212 Atheros AR5212 802.11abg wireless'
    class      = network
    subclass   = ethernet
```

Dit beetje uitvoer, verkregen met `pciconf -lv` geeft aan dat het stuurprogramma ath een draadloos Ethernetapparaat heeft gevonden. Het gebruik van `man ath` zal de handleiding voor ath(4) teruggeven.

Wanneer de vlag `-k` aan man(1) wordt gegeven kan deze nuttige informatie geven. Met het bovenstaande kan dit gedaan worden:

```
# man -k Atheros
```

om een lijst handleidingen te krijgen die dat ene woord bevatten:

```
ath(4)                  - Atheros IEEE 802.11 wireless network driver
ath_hal(4)              - Atheros Hardware Access Layer (HAL)
```

Gewapend met een inventarisatielijst van de hardware zou het proces van het bouwen van een eigen kernel minder angstaanjagend moeten lijken.

8.4. Kernel stuurprogramma's, subsystemen, en modules

Bekijk, voordat er een eigen kernel gebouwd wordt, de redenen om dit te doen. Als er de noodzaak is voor specifieke hardwareondersteuning, kan dit reeds beschikbaar zijn als een module.

Kernelmodules staan in de map /boot/kernel en kunnen dynamisch in de draaiende kernel worden geladen met kldload(8). De meeste, als niet alle, kernelstuurprogramma's hebben een specifieke module en een handleiding. De laatste sectie merkte bijvoorbeeld het draadloze Ethernetstuurprogramma ath op. Van dit stuurprogramma staat de volgende informatie in de handleiding:

```
Plaats de volgende regel in man:loader.conf[5] om
      het stuurprogramma tijdens het opstarten als een module te laden:

           if_ath_load="YES"
```

Zoals aangegeven, zal het toevoegen van de regel `if_ath_load="YES"` aan /boot/loader.conf deze module dynamisch laden tijdens het opstarten.

In sommige gevallen is er geen geassocieerde module. Dit geldt het vaakst voor bepaalde subsystemen en zeer belangrijke stuurprogramma's, het fast file system (FFS) bijvoorbeeld is een

verplichte optie in de kernel, net zoals netwerkondersteuning (INET). Helaas is de enige manier om te zien of een stuurprogramma nodig is naar de module zelf zoeken.



Het is eenvoudig om ondersteuning voor een apparaat of optie te verwijderen en met een kapotte kernel opgezadeld te zitten. Als bijvoorbeeld het stuurprogramma ata(4) uit het kernelinstellingenbestand gehaald wordt, zal een systeem dat ATA schijfstuurprogramma's gebruikt niet opstarten zonder de module aan loader.conf toe te voegen. Kijk bij twijfel of de module aanwezig is en laat ondersteuning dan gewoon in de kernel.

8.5. Bouwen en installeren van een aangepaste kernel



Het is noodzakelijk om de volledige broncode van FreeBSD geïnstalleerd te hebben om de kernel te bouwen.

Eerst wordt er een overzicht gegeven van de mappen waarin de kernel gebouwd wordt. Alle genoemde mappen staan onder de map /usr/src/sys, die ook toegankelijk is via de padnaam /sys. Er zijn hier een aantal mappen aanwezig die de verschillende delen van de kernel representeren, maar de meest belangrijke hiervan zijn arch/conf, waarin de kernelinstellingen bewerkt worden en compile, waarin de aangepaste kernel gebouwd wordt. *arch* representeert hier één van i386, amd64, ia64, powerpc, sparc64 of pc98 (een alternatieve ontwikkelingstak van PC-hardware die populair is in Japan). Alles binnen de map van een bepaalde architectuur is er alleen voor die architectuur. De rest van de code is machine-onafhankelijk en hetzelfde op alle platformen waarnaar FreeBSD eventueel overgezet kan worden. De indeling van de mapstructuur is logisch: alle ondersteunde apparaten, bestandssystemen en opties staan in een eigen submap.

Dit voorbeelden in dit hoofdstuk veronderstellen dat de i386-architectuur gebruikt wordt. Als dit voor de lezer anders is, moeten de bijhorende aanpassingen aan de padnamen worden gemaakt.



Als de map /usr/src/ *niet* aanwezig is op een systeem (of als het leeg is), dan is de broncode niet geïnstalleerd. De eenvoudigste manier om de volledige broncode te installeren is csup(1) te gebruiken zoals beschreven in Broncode synchroniseren. U dient tevens een symbolische link naar /usr/src/sys/ aan te maken:

```
# ln -s /usr/src/sys /sys
```

Daarna kan vanuit de map arch/conf het instellingenbestand GENERIC naar de naam voor de aangepaste kernel gekopieerd worden. Bijvoorbeeld:

```
# cd /usr/src/sys/i386/conf
# cp GENERIC MIJNKERNEL
```

Traditioneel bestaat deze naam geheel uit hoofdletters en als er meerdere FreeBSD-machines worden beheerd met verschillende hardware is het een goed idee om het te vernoemen naar de hostnaam van de machine. Omwille van dit voorbeeld wordt het MIJNKERNEL genoemd.



Het kernelinstellingenbestand direct onder /usr/src opslaan kan een slecht idee zijn. In geval van problemen kan het verleidelijk zijn om /usr/src te verwijderen en opnieuw te beginnen. Nadat dit gedaan is kost het vaak maar enkele seconden om te realiseren dat het instellingenbestand voor de aangepaste kernel verwijderd is. Ook moet GENERIC niet gewijzigd worden, omdat het tijdens de volgende keer dat de broncodeboom bijgewerkt wordt, overschreven kan worden waarbij de wijzigingen in de kernelinstellingen verloren gaan.

Het kan gewenst zijn om het kernelinstellingenbestand ergens anders op te slaan en een symbolische link naar het bestand in de map i386 aan te maken:

```
# cd /usr/src/sys/i386/conf
# mkdir /root/kernels
# cp GENERIC /root/kernels/MIJNKERNEL
# ln -s /root/kernels/MIJNKERNEL
```

Nu moet MIJNKERNEL met de favoriete tekstverwerker bewerkt worden. Voor beginners is waarschijnlijk alleen de tekstverwerker vi beschikbaar, die te ingewikkeld is om hier te beschrijven, maar goed is beschreven in vele boeken in de bibliografie. FreeBSD biedt ook de eenvoudigere tekstverwerker ee, die voor een beginner de keuze bij uitstek is. De commentaarregels in het begin kunnen gewijzigd worden om de persoonlijke instellingen of de veranderingen die gemaakt zijn ten opzichte van GENERIC weer te geven.

Voor degenen die een kernel op SunOS™ of een andere BSD hebben gebouwd zal veel van dit bestand bekend voorkomen. Echter, voor degenen die van een ander besturingssysteem zoals DOS komen, kan het instellingenbestand GENERIC overdonderend overkomen, dus moeten de beschrijvingen in de sectie Het Instellingenbestand zorgvuldig opgevolgd worden.



Als de broncodeboom gesynchroniseerd is met de nieuwste broncode van het FreeBSD-project, moet altijd /usr/src/UPDATING gelezen worden voordat enige bijwerkstappen worden genomen. Dit bestand beschrijft alle belangrijke zaken en gebieden binnen de broncodestructuur die speciale aandacht nodig hebben. /usr/src/UPDATING komt altijd overeen met de lokale versie van de FreeBSD-broncode en is daarom meer bijgewerkt met nieuwe informatie dan dit handboek.

Nu moet de broncode voor de kernel gecompileerd worden.

Procedure: Een kernel bouwen



Het is noodzakelijk om de volledige broncode van FreeBSD geïnstalleerd te hebben om te kernel te bouwen.

1. Ga naar de map /usr/src:

```
# cd /usr/src
```

2. Compileer de kernel:

```
# make buildkernel KERNCONF=MIJNKERNEL
```

3. Installeer de nieuwe kernel:

```
% make installkernel KERNCONF=MIJNKERNEL
```

Bij het bouwen van een aangepaste kernel worden standaard *alle* kernelmodules ook herbouwd. Om de kernel sneller bij te werken en alleen de aangepaste modules te bouwen kan /etc/make.conf aangepast worden voordat de kernel wordt gebouwd:

```
MODULES_OVERRIDE = linux acpi sound/sound sound/driver/ds1 ntfs
```



Met deze variabele wordt een lijst van te bouwen modules ingesteld die gebouwd moeten worden in plaats van allen.

```
WITHOUT_MODULES = linux acpi sound ntfs
```

Deze variabele stelt een lijst in van modules op het topniveau die moeten worden uitgesloten van het bouwproces. Andere variabelen die mogelijk ook nuttig zijn in het proces van het bouwen van een kernel staan beschreven in de handleiding voor make.conf(5).

De nieuwe kernel wordt naar de map /boot/kernel gekopieerd als /boot/kernel/kernel en de oude kernel wordt verplaatst naar /boot/kernel.old/kernel. Nu moet het systeem afgesloten worden en opnieuw worden opgestart om gebruik te maken van de nieuwe kernel. Er zijn wat instructies voor problemen oplossen aan het einde van dit hoofdstuk, die erg nuttig kunnen zijn als er iets misgaat. Vergeet niet om het gedeelte te lezen waarin staat uitgelegd hoe te herstellen als de nieuwe kernel niet opstart.



Andere bestanden die te maken hebben met het opstartproces, zoals de boot loader(8) en instellingen worden opgeslagen in /boot. Modules van derde partijen of eigen modules kunnen in /boot/kernel opgeslagen worden, alhoewel gebruikers erop bedacht moeten zijn dat het erg belangrijk is dat de modules synchroon worden gehouden met de gecompileerde kernel. Modules die niet bedoeld zijn om met de gecompileerde kernel te draaien kunnen voor instabiliteit of onjuistheden zorgen.

8.6. Het instellingenbestand

Het algemene formaat van een instellingenbestand is vrij eenvoudig. Elke regel bevat een sleutelwoord en één of meer argumenten. Omwille van de eenvoud bevatten de meeste regels maar één argument. Alles wat na een `#` komt, wordt als commentaar beschouwd en genegeerd. De volgende gedeelten beschrijven elk sleutelwoord, in het algemeen in dezelfde volgorde als GENERIC, alhoewel sommige samenhangende sleutelwoorden gegroepeerd zijn in een enkel gedeelte (zoals Netwerken) zelfs al staan ze verspreid in het bestand GENERIC. Een uitputtende lijst van architectuurafhankelijke opties en apparaten staat in het bestand NOTES, dat in dezelfde map staat als het bestand GENERIC. Architectuuronafhankelijke opties staan in /usr/src/sys/conf/NOTES.

Een nieuwe directief `include` is beschikbaar om te gebruiken in instellingenbestanden. Hiermee kan een ander instellingenbestand logisch in het huidige worden opgenomen, waardoor het eenvoudig wordt om kleine veranderingen relatief aan een bestaand bestand te onderhouden. Als u bijvoorbeeld een GENERIC kernel nodig heeft met slechts een klein aantal aanvullende opties of stuurprogramma's, hoeft u hiermee slechts een delta ten opzichte van GENERIC te onderhouden:

```
include GENERIC
ident MIJNKERNEL

options     IPFIREWALL
options     DUMMYNET
options     IPFIREWALL_DEFAULT_TO_ACCEPT
options     IPDIVERT
```

Veel beheerders zullen aanzienlijke voordelen in dit model zien vergeleken met de vroegere gewoonte om instellingenbestanden vanuit het niets te schrijven: het lokale instellingenbestand zal alleen lokale verschillen uitdrukken ten opzichte van een GENERIC kernel en wanneer upgrades worden uitgevoerd zullen nieuwe mogelijkheden die aan GENERIC zijn toegevoegd ook aan de lokale kernel worden toegevoegd tenzij dit expliciet verhinderd wordt met `nooptions` of `nodevice`. De rest van dit hoofdstuk behandelt de inhoud van een typisch instellingenbestand en de verschillende rollen die opties en apparaten spelen.



Draai het volgende commando als `root` om een bestand te bouwen dat alle beschikbare opties bevat, wat normaliter voor testdoeleinden gedaan wordt:

```
# cd /usr/src/sys/i386/conf && make LINT
```

Het volgende is een voorbeeld van het kernelinstellingenbestand GENERIC met aanvullend commentaar omwille van de helderheid. Dit voorbeeld is redelijk gelijk aan de versie in /usr/src/sys/i386/conf/GENERIC.

```
machine   i386
```

Dit is de architectuur van de machine. Het moet één van `amd64`, `i386`, `ia64`, `pc98`, `powerpc` of `sparc64`

zijn.

```
cpu   I486_CPU
cpu   I586_CPU
cpu   I686_CPU
```

Bovenstaande optie geeft het type CPU aan dat in een systeem zit. De CPU-regel kan meerdere keren voorkomen (als bijvoorbeeld onbekend is of `I586_CPU` of `I686_CPU` gebruikt moet worden), maar voor een aangepaste kernel is het beter om alleen de aanwezige CPU aan te geven. Als er twijfel bestaat over het type CPU, kan het bestand /var/run/dmesg.boot worden bekeken voor de opstartberichten.

```
ident   GENERIC
```

Dit is de identificatie van de kernel. Dit moet veranderd worden in de naam van de kernel, dus `MIJNKERNEL` als de instructies van de voorgaande voorbeelden gevolgd zijn. De waarde in de string `ident` wordt afgebeeld wanneer de kernel opstart, dus is het handig om de nieuwe kernel een andere naam te geven als deze apart moet worden gehouden van de gebruikelijke kernel (als er bijvoorbeeld een experimentele kernel gebouwd wordt).

```
#Om apparaatbindingen statisch in te compileren in plaats van via /boot/device.hints.
#hints    "GENERIC.hints"     # Standaardlocatie voor devices.
```

device.hints(5) wordt gebruikt om opties van de programma's die de apparaten aansturen in te stellen. De standaardplaats die loader(8) controleert tijdens het opstarten is /boot/device.hints. Met de optie `hints` is het mogelijk om deze aanwijzingen statisch in de kernel te compileren, waardoor er geen noodzaak is om een bestand device.hints in /boot aan te maken.

```
makeoptions        DEBUG=-g  # Bouw kernel met gdb(1) debugsymbolen.
```

Het normale bouwproces van FreeBSD voegt debuginformatie toe wanneer de kernel met de optie `-g` gebouwd wordt, wat debuginformatie doorgeeft aan gcc(1).

```
options   SCHED_ULE    # ULE taakplanner
```

De standaard taakplanner voor FreeBSD. Laat dit staan.

```
options   PREEMPTION # Zet kernelthreadpreëmptie aan
```

Sta toe dat threads in de kernel worden gepreëmpt door threads met een hogere prioriteit. Het help bij interactiviteit en staat toe dat interruptthreads eerder draaien in plaats van te moeten wachten.

```
options   INET      # internetwerken
```

Netwerkondersteuning. Laat dit aanstaan, zelfs als een verbinding met een netwerk niet gepland is. De meeste programma's hebben tenminste een teruglusnetwerk nodig (dat wil zeggen het maken van netwerkverbindingen binnen de PC), dus dit is eigenlijk verplicht.

```
options   INET6     # IPv6 communicatieprotocollen
```

Dit zet de IPv6-communicatieprotocollen aan.

```
options   FFS       # Berkeley Fast Bestandssysteem
```

Dit is het basisbestandssysteem voor de harde schijf. Laat dit erin staan als er vanaf de harde schijf wordt opgestart.

```
options   SOFTUPDATES   # Schakel FFS Softupdates ondersteuning in
```

Deze optie zet softupdates in de kernel aan en helpt om de schijftoegang voor schrijven te verhogen. Zelfs als deze functionaliteit door de kernel geleverd wordt, moet die voor specifieke schijven worden aangezet. Bekijk de uitvoer van mount(8) om te zien of softupdates aanstaat voor de systeemschijven. Als de optie `soft-updates` niet zichtbaar is, dient deze geactiveerd te worden met behulp van tunefs(8) voor bestaande bestandssystemen of newfs(8) voor nieuwe bestandssystemen.

```
options   UFS_ACL   # Ondersteuning voor toegangscontrolelijsten
```

Met deze optie wordt de ondersteuning voor toegangscontrolelijsten aangezet. Hiervoor zijn uitgebreide attributen en UFS2 nodig. Een en ander wordt in detail beschreven in Bestandssysteem toegangscontrolelijsten (ACLs). ACL's staan standaard aan en moeten niet uitgezet worden in de kernel als ze al eerder op een bestandssysteem zijn gebruikt, omdat dit de toegangscontrolelijsten verwijdert en hierdoor de manier waarop bestanden beschermd worden op onvoorspelbare wijze verandert.

```
options   UFS_DIRHASH   # Verbeter prestaties in grote mappen
```

Deze optie bevat functionaliteit om schijfoperaties op grote mappen te versnellen, ten koste van extra geheugen. Deze staat normaalgesproken, zoals voor een grote server of interactief werkstation, aan en wordt uitgezet als FreeBSD op een kleiner systeem wordt gebruikt waar geheugen het belangrijkste en schijfsnelheid minder belangrijk is, zoals voor een firewall.

```
options   MD_ROOT   # MD is een potentieel rootapparaat
```

Deze optie zet ondersteuning aan voor een virtuële schijf die in het geheugen wordt geïmplementeerd en als rootapparaat wordt gebruikt.

```
options   NFSCLIENT    # Netwerk Bestandssysteem Client
options   NFSSERVER    # Netwerk Bestandssysteem Server
options   NFS_ROOT     # NFS bruikbaar als /, NFSCLIENT nodig
```

Het netwerkbestandssysteem. Dit kan weggelaten worden tenzij er gepland is om partities te aan te koppelen van een UNIX® bestandsserver over TCP/IP.

```
options   MSDOSFS   # MSDOS Bestandssysteem
```

Het MS-DOS® bestandssysteem. Dit kan veilig weggelaten worden, tenzij er gepland is om een DOS-geformatteerde partitie van de harde schijf tijdens het opstarten aan te koppelen. Het wordt automatisch geladen als er voor de eerste keer een DOS-partitie wordt aangekoppeld, zoals boven beschreven. Bovendien geeft de uitstekende software emulators/mtools toegang tot DOS-floppies zonder dat ze aangekoppeld en afgekoppeld moeten worden en heeft het `MSDOSFS` helemaal niet nodig.

```
options   CD9660    # ISO 9660 Bestandssysteem
```

Het ISO 9960-bestandssysteem voor CD-ROMs. Commentarieer dit uit als er geen CD-ROM drive aanwezig is of als er slechts af en toe gegevens-CD-ROMs aangekoppeld worden (aangezien het dynamisch geladen wordt als er voor de eerste keer een gegevens-CD-ROM aangekoppeld wordt). Audio-CD's hebben dit bestandssysteem niet nodig.

```
options   PROCFS    # Procesbestandssysteem (vereist PSEUDOFS)
```

Het procesbestandssysteem. Dit is een "als-of" bestandssysteem, aangekoppeld op /proc, dat programma's als ps(1) in staat stelt om meer informatie over de draaiende processen te geven. Het is in de meeste omstandigheden niet nodig om `PROCFS` te gebruiken, omdat de meeste debug- en monitorgereedschappen zijn aangepast om zonder `PROCFS` te draaien: installaties koppelen dit bestandssysteem standaard niet aan.

```
options   PSEUDOFS  # Pseudo-bestandssysteem raamwerk
```

Kernels die `PROCFS` gebruiken moeten ook ondersteuning voor `PSEUDOFS` opnemen.

```
options   GEOM_PART_GPT  # GUID Partitietabellen.
```

Voegt ondersteuning voor GUID Partitietabellen toe. GPT biedt de mogelijkheid om een groot aantal partities per schijf te hebben, 128 is de standaardconfiguratie.

```
options   COMPAT_43 # Compatibel met BSD 4.3 [ERIN HOUDEN!]
```

Compatibiliteit met 4.3BSD. Laat dit aanstaan. Sommige programma's gedragen zich vreemd als dit uitgecommentarieerd wordt.

```
options   COMPAT_FREEBSD4     # Compatibel met FreeBSD 4
```

Deze optie is nodig om ondersteuning te bieden aan applicaties die gecompileerd zijn op oudere versies van FreeBSD en gebruik maken van oudere systeemaanroep-interfaces. Het is aanbevolen dat deze optie gebruikt wordt op alle i386™ systemen die mogelijk oudere applicaties draaien. Voor platformen die pas in 5.X ondersteuning verwierven, zoals ia64 en sparc64, is deze optie niet nodig.

```
options   COMPAT_FREEBSD5     # Compatibel met FreeBSD5
```

Deze optie is vereist om ondersteuning te geven aan applicaties die gecompileerd zijn op FreeBSD 5.X die gebruik maken van de systeemaanroepinterfaces van FreeBSD 5.X.

```
options   COMPAT_FREEBSD6     # Compatibel met FreeBSD5
```

Deze optie is vereist om ondersteuning te geven aan applicaties die gecompileerd zijn op FreeBSD 6.X die gebruik maken van de systeemaanroepinterfaces van FreeBSD 6.X.

```
options   COMPAT_FREEBSD7     # Compatibel met FreeBSD5
```

Deze optie is vereist om ondersteuning te geven aan applicaties die gecompileerd zijn op FreeBSD 7.X die gebruik maken van de systeemaanroepinterfaces van FreeBSD 7.X.

```
options   SCSI_DELAY=5000    # Vertraging (in ms) voordat SCSI wordt ondergezocht.
```

Dit zorgt ervoor dat de kernel vijf seconden wacht voordat die elk SCSI-apparaat in het systeem onderzoekt. Als er alleen IDE-harde schijven zijn, kan deze optie genegeerd worden, anders kan geprobeerd worden dit getal te verlagen, om het opstarten te versnellen. Uiteraard moet deze waarde weer verhoogd worden als FreeBSD problemen heeft om de SCSI-apparaten te herkennen.

```
options   KTRACE    # ktrace(1) ondersteuning
```

Dit schakelt kernelondersteuning voor het volgen processen in, wat handig is tijdens debuggen.

```
options   SYSVSHM   # SYSV-stijl gedeeld geheugen
```

Deze optie biedt System V gedeeld geheugen. Meestal wordt dit wegens de XSHM-uitbreiding in X gebruikt, waar door vele grafische programma's automatisch gebruik van wordt gemaakt voor extra snelheid. Als X gebruik wordt, is het raadzaam om dit op te nemen.

```
options   SYSVMSG   # SYSV-stijl berichtwachtrijen
```

Dit biedt ondersteuning voor System V berichten. Ook deze optie voegt slechts een paar honderd bytes aan de kernel toe.

```
options   SYSVSEM   # SYSV-stijl semaforen
```

Dit biedt ondersteuning voor System V semaforen. Het wordt minder vaak gebruikt, maar voegt slechts een paar honderd bytes aan de kernel toe.



De optie `-p` van het commando ipcs(1) geeft een lijst van alle processen die een van deze System V faciliteiten gebruikt.

```
options   _KPOSIX_PRIORITY_SCHEDULING   # POSIX P1003_1B real-time extensies
```

Dit biedt real-time-uitbreidingen die in de 1993 POSIX® zijn toegevoegd. Bepaalde applicaties in de Portscollectie gebruiken deze (zoals StarOffice™).

```
options   KBD_INSTALL_CDEV    # installeer een CDEV-ingang in /dev
```

Deze optie is nodig om apparaatknooppunten voor het toetsenbord aan te maken in /dev.

```
options   ADAPTIVE_GIANT      # Giant mutex is adaptief.
```

Giant is de naam van een wederzijds uitsluitingsmechanisme (een sleep mutex) dat een grote verzameling kernelbronnen beschermt. Vandaag de dag is dit een onacceptabele prestatie-bottleneck die actief door sloten wordt vervangen die individuele bronnen beschermen. De optie `ADAPTIVE_GIANT` zorgt ervoor dat Giant in de verzamelingen van mutexen wordt opgenomen waar actief wordt opgespind. Dit betekent dat wanneer een thread de Giant-mutex wil nemen, maar die reeds door een thread op een andere CPU genomen is, de eerste thread blijft draaien en wacht tot er een slot vrijkomt. Normaalgesproken zou de thread weer gaan slapen en wachten op de volgende kans om te draaien. Laat dit er in geval van twijfel instaan.



Merk op dat in FreeBSD 8.0-RELEASE en later alle mutexen standaard adaptief zijn, tenzij ze expliciet op niet-adaptief zijn gezet door met de optie `NO_ADAPTIVE_MUTEXES` te compileren. Een gevolg is dat Giant nu standaard adaptief is, en dat de optie `ADAPTIVE_GIANT` uit de kernelinstellingen is verwijderd.

```
device    apic      # I/O APIC
```

Het apic-apparaat zet de ondersteuning voor I/O-APIC voor het afleveren van interrupts aan. Het apic-apparaat kan zowel in UP- als in SMP-kernels gebruikt worden, maar is noodzakelijk voor

SMP-kernels. Voeg `options SMP` toe om ondersteuning voor meerdere processoren op te nemen.



Het apic-apparaat bestaat alleen in de i386-architectuur, deze instelregel dient niet op andere architecturen gebruikt te worden.

```
device    eisa
```

Neem dit op voor een EISA-moederbord. Dit zet ondersteuning voor zelfdetectie en -instelling aan voor alle apparaten op de EISA-bus.

```
device    pci
```

Neem dit op voor een PCI-moederbord. Dit zet ondersteuning voor zelfdetectie van PCI-kaarten en gatewaying van PCI-naar-ISA-bus aan.

```
# Floppy drives
device    fdc
```

Dit is de controller voor de floppydrive.

```
# ATA- en ATAPI-apparaten
device    ata
```

Dit stuurprogramma biedt ondersteuning aan alle ATA- en ATAPI-apparaten. Er is slechts één `device ata`-regel nodig om de kernel alle PCI ATA/ATAPI-apparaten te laten ontdekken op moderne machines.

```
device    atadisk   # ATA schijven
```

Dit is samen met `device ata` nodig voor ATA schijven.

```
device    ataraid   # ATA RAID schijven
```

Dit is samen met `device ata` nodig voor ATA RAID-schijven.

```
device    atapicd   # ATAPI CD-ROM drives
```

Dit is samen met `device ata` nodig voor ATAPI CD-ROM drives.

```
device   atapifd    # ATAPI floppy drives
```

Dit is samen met `device ata` nodig voor ATAPI floppydrives.

```
device    atapist   # ATAPI tape drives
```

Dit is samen met `device ata` nodig voor ATAPI tapedrives.

```
options   ATA_STATIC_ID   # Statische apparaatnummering
```

Dit zorgt ervoor dat de controller statisch nummert. Zonder deze optie worden nummers dynamisch toegewezen.

```
# SCSI Controllers
device          ahb        # EISA AHA1742 familie
device          ahc        # AHA2940 en onboard AIC7xxx apparaten
options         AHC_REG_PRETTY_PRINT    # Print registerbitvelden in
                                        # debuguitvoer.  Voegt ~128k
                                        # aan stuurprogramma toe.
device          ahd        # AHA39320/29320 en onboard AIC79xx apparaten
options         AHD_REG_PRETTY_PRINT    # Print registerbitvelden in
                                        # debuguitvoer.  Voegt ~215k
                                        # aan stuurprogramma toe.
device          amd        # AMD 53C974 (Teckram DC-390(T))
device          isp        # Qlogic familie
#device         ispfw      # Firmware voor QLogic HBAs- normaliter een module
device          mpt        # LSI-Logic MPT-Fusion
#device         ncr        # NCR/Symbios Logic
device          sym        # NCR/Symbios Logic (nieuwere chipsets + die van `ncr')
device          trm        # Tekram DC395U/UW/F DC315U adapters

device          adv        # Advansys SCSI adapters
device          adw        # Advansys wide SCSI adapters
device          aha        # Adaptec 154x SCSI adapters
device          aic        # Adaptec 15[012]x SCSI adapters, AIC-6[23]60.
device          bt         # Buslogic/Mylex MultiMaster SCSI adapters

device          ncv        # NCR 53C500
device          nsp        # Workbit Ninja SCSI-3
device          stg        # TMC 18C30/18C50
```

SCSI controllers. Commentarieer de regels uit voor apparaten die niet in het systeem aanwezig zijn. Als het een systeem met alleen IDE apparaten betreft, kunnen ze allemaal weggelaten worden. De regels met `*_REG_PRETTY_PRINT` zijn debugopties voor hun respectievelijke stuurprogramma's.

```
# SCSI randapparaten
device          scbus      # SCSI bus (nodig voor SCSI)
device          ch         # SCSI media changers
device          da         # Direct Access (schijven)
```

```
device          sa          # Sequential Access (tape, enzovoort)
device          cd          # CD
device          pass        # Passthrough apparaat (directe SCSI-toegang)
device          ses         # SCSI Omgevingsdiensten (en SAF-TE)
```

SCSI-aanhangsels. Ook hier geldt dat apparaten die niet aanwezig zijn uitgecommentarieerd kunnen worden, of als alleen IDE-hardware aanwezig is, ze allemaal weggelaten kunnen worden.



Het USB-stuurprogramma umass(4) en enkele andere stuurprogramma's gebruiken het SCSI-subsysteem, alhoewel ze geen echte SCSI-apparaten zijn. Daarom mag SCSI-ondersteuning niet verwijderd worden als dit soort stuurprogramma's in de kernelinstellingen worden opgenomen.

```
# RAID controllers met interfaces naar het SCSI subsysteem
device          amr         # AMI MegaRAID
device          arcmsr      # Areca SATA II RAID
device          asr         # DPT SmartRAID V, VI en Adaptec SCSI RAID
device          ciss        # Compaq Smart RAID 5*
device          dpt         # DPT Smartcache III, IV - Zie NOTES voor opties
device          hptmv       # Highpoint RocketRAID 182x
device          hptrr       # Highpoint RocketRAID 17xx, 22xx, 23xx, 25xx
device          iir         # Intel Integrated RAID
device          ips         # IBM (Adaptec) ServeRAID
device          mly         # Mylex AcceleRAID/eXtremeRAID
device          twa         # 3ware 9000 series PATA/SATA RAID

# RAID controllers
device          aac         # Adaptec FSA RAID
device          aacp        # SCSI passthrough voor aac (heeft CAM nodig)
device          ida         # Compaq Smart RAID
device          mfi         # LSI MegaRAID SAS
device          mlx         # Mylex DAC960 famile
device          pst         # Promise Supertrak SX6000
device          twe         # 3ware ATA RAID
```

Ondersteunde RAID-controllers. Als een van deze niet aanwezig is, kan deze uitgecommentarieerd of verwijderd worden.

```
# atkbdc0 bestuurt het toetsenbord en de PS/2 muis
device    atkbdc    # AT toetsenbordcontroller
```

De toetsenbordcontroller (atkbdc) biedt I/O-diensten aan voor het AT-toetsenbord en het PS/2-type van aanwijsapparaten. Deze controller is noodzakelijk voor het toetsenbordstuurprogramma (atkbd) en het PS/2-aanwijsapparaatstuurprogramma (psm).

```
device    atkbd     # AT toetsenbord
```

Het stuurprogramma atkbd biedt samen met de controller atkbdc toegang tot het AT84-toetsenbord of het uitgebreide AT-toetsenbord dat verbonden is met de controller voor het AT-toetsenbord.

```
device    psm       # PS/2 muis
```

Dit apparaat kan gebruikt worden als de muis in de PS/2-muispoort wordt geplugd.

```
device    kbdmux    # toetsenbordmultiplexer
```

Basisondersteuning voor multiplexing van toetsenborden. Als u niet van plan bent om meerdere toetsenborden op het systeem te gebruiken, kunt u deze regel veilig verwijderen.

```
device    vga       # VGA videokaart stuurprogramma
```

Het stuurprogramma voor de videokaart.

```
device    splash    # Splash screen en screensaver ondersteuning
```

Een splash-scherm tijdens het opstarten! Screensavers hebben deze optie ook nodig.

```
# syscons is het standaard consolestuurprogramma, lijkt op een SCO console
device    sc
```

sc is het standaard consolestuurprogramma en lijkt op een SCO-console. Aangezien de meeste programma's die met een volledig scherm werken de console via een terminaldatabase zoals termcap benaderen, moet het niet uitmaken of dit of vt, het VT220-compatibele consolestuurprogramma, gebruikt wordt. Wanneer er aangemeld wordt, dient de variabele TERM op scoansi gezet worden indien programma's die met een volledig scherm werken problemen hebben om met dit console te draaien.

```
# Schakel dit in voor het pcvt (VT220 compatibele) consolestuurprogramma
#device     vt
#options    XSERVER     # ondersteuning voor X server op een vt console
#options    FAT_CURSOR  # begin met een blokcursor
```

Dit is een VT220-compatibel consolestuurprogramma, achterwaarts compatibel met de VT100/102. Het werkt goed op enkele laptops die hardware-incompatibiliteiten hebben met sc. Ook dient de variabele TERM op vt100 of vt220 gezet te worden bij het aanmelden. Dit stuurprogramma kan ook nuttig zijn wanneer er verbinding wordt gemaakt met een groot aantal verschillende machines in een netwerk, waarbij de ingangen termcap of terminfo voor het apparaat sc vaak niet beschikbaar zijn. vt100 is op bijna elk platform beschikbaar.

```
device    agp
```

Neem dit op als er een AGP-kaart in het systeem aanwezig is. Dit zet ondersteuning voor AGP aan, en ondersteuning voor AGP GART voor borden die deze mogelijkheden hebben.

```
# Ondersteuning voor energiebeheer (zie NOTES voor meer opties)
#device   apm
```

Ondersteuning voor geavanceerd energiebeheer (Advanced Power Management). Dit is nuttig voor laptops, alhoewel dit standaard uitgeschakeld is in GENERIC.

```
# Schakel suspend/resume ondersteuning voor de i8254 in.
device    pmtimer
```

Het stuurprogramma voor het timerapparaat voor energiebeheergebeurtenissen, zoals APM en ACPI.

```
# PCCARD (PCMCIA) ondersteuning.
# PCMCIA en cardbus bridge ondersteuning.
device    cbb       # cardbus (yenta) bridge
device    pccard    # PC Card (16-bit) bus
device    cardbus   # CardBus (32-bit) bus
```

Ondersteuning voor PCMCIA. Dit is wenselijk voor laptopgebruikers.

```
# Serial (COM) poorten
device    sio       # 8250, 16[45]50-gebaseerde seriële poorten
```

Dit zijn de seriële poorten waarnaar in de wereld van MS-DOS®/Windows® verwezen wordt als COM-poorten.



Als er een intern modem op COM4 en een seriële poort op COM2 aanwezig is, moet het IRQ van het modem in 2 worden veranderd (om duistere technische redenen geldt dat IRQ2 = IRQ9) om er vanuit FreeBSD toegang toe te krijgen. Als er een multipoort seriële kaart aanwezig is, staat in sio(4) meer informatie over de juiste waarden die aan /boot/device.hints toegevoegd moeten worden. Sommige videokaarten (vaak gebaseerd op S3 chips) gebruiken IO-adressen van de vorm `0x*2e8`, en omdat vele goedkope serieële kaarten de 16-bits IO-adresruimte niet volledig decoderen, botsen ze met deze kaarten waardoor de COM4-poort praktisch onbruikbaar is.

Elke serieële poort moet een uniek IRQ hebben (tenzij er gebruik wordt gemaakt van een van de multipoortkaarten waarbij gedeelde interrupts ondersteund worden), dus kunnen de standaard IRQ's voor COM3 en COM4 niet gebruikt

worden.

```
# Parallelle poort
device    ppc
```

Dit is de interface voor de parallelle poort op de ISA-bus.

```
device    ppbus     # Parallelle poortbus (verplicht)
```

Biedt ondersteuning voor de parallelle poortbus.

```
device    lpt       # Printer
```

Ondersteuning voor parallelle poort-printers.



Alle van de bovenstaande drie zijn noodzakelijk om ondersteuning voor parallelle printers aan te zetten.

```
device    ppi       # Parallelle poort interface apparaat
```

De algemene I/O ("geek-poort") + IEEE1284 I/O.

```
#device   vpo       # scbus en da verplicht
```

Dit is voor een Iomega Zipdrive. Hiervoor is ondersteuning voor `scbus` en `da` nodig. De beste prestaties worden gehaald met poorten in EPP 1.9-modus.

```
#device   puc
```

Dit dient uitgecommentarieerd te worden indien er een "domme" seriële of parallelle PCI-kaart aanwezig is die ondersteund wordt door het puc(4) verbindingsstuurprogramma.

```
# PCI Ethernet NIC's.
device    de        # DEC/Intel DC21x4x (Tulip)
device    em        # Intel PRO/1000 adapter Gigabit Ethernet Card
device    ixgb      # Intel PRO/10GbE Ethernet Card
device    txp       # 3Com 3cR990 (Typhoon)
device    vx        # 3Com 3c590, 3c595 (Vortex)
```

Verscheidene PCI-netwerkkaartstuurprogramma's. Degenen die niet in het systeem aanwezig zijn kunnen uitgecommentarieerd of verwijderd worden.

```
# PCI Ethernet NIC's die de MII bus controller code gebruiken.
# NB: 'device miibus' moet behouden blijven om deze NIC's te kunnen gebruiken!
device    miibus    # MII bus ondersteuning
```

Ondersteuning voor MII-bus is noodzakelijk voor sommige PCI 10/100 Ethernet-NICs, namelijk voor diegenen die MII-geldige transceivers gebruiken of interfaces voor transceiverbesturing implementeren die als een MII werken. Door `device miibus` aan de kernelinstellingen toe te voegen wordt de ondersteuning voor de generieke miibus-API en voor alle PHY-stuurprogramma's opgenomen, waaronder een generieke voor PHYs die niet specifiek door een individueel stuurprogramma worden behandeld.

```
device    bce       # Broadcom BCM5706/BCM5708 Gigabit Ethernet
device    bfe       # Broadcom BCM440x 10/100 Ethernet
device    bge       # Broadcom BCM570xx Gigabit Ethernet
device    dc        # DEC/Intel 21143 en verschillende gelijkwerkenden
device    fxp       # Intel EtherExpress PRO/100B (82557, 82558)
device    lge       # Level 1 LXT1001 gigabit Ethernet
device    msk       # Marvell/SysKonnect Yukon II Gigabit Ethernet
device    nge       # NatSemi DP83820 gigabit Ethernet
device    nve       # nVidia MCP on-board Ethernet Networking
device    pcn       # AMD Am79C97x PCI 10/100 (voorrang op 'lnc')
device    re        # RealTek 8139C+/8169/8169S/8110S
device    rl        # RealTek 8129/8139
device    sf        # Adaptec AIC-6915 (Starfire)
device    sis       # Silicon Integrated Systems SiS 900/SiS 7016
device    sk        # SysKonnect SK-984x & SK-982x gigabit Ethernet
device    ste       # Sundance ST201 (D-Link DFE-550TX)
device    stge      # Sundance/Tamarack TC9021 gigabit Ethernet
device    ti        # Alteon Networks Tigon I/II gigabit Ethernet
device    tl        # Texas Instruments ThunderLAN
device    tx        # SMC EtherPower II (83c170 EPIC)
device    ge        # VIA VT612x gigabit Ethernet
device    vr        # VIA Rhine, Rhine II
device    wb        # Winbond W89C840F
device    xl        # 3Com 3c90x (Boomerang, Cyclone)
```

Stuurprogramma's die gebruik maken van de MII bus-controllercode.

```
# ISA Ethernet NIC's.  Inclusief pccard NIC's.
device    cs        # Crystal Semiconductor CS89x0 NIC
# 'device ed' heeft 'device miibus' nodig
device    ed        # NE[12]000, SMC Ultra, 3c503, DS8390 kaarten
device    ex        # Intel EtherExpress Pro/10 en Pro/10+
device    ep        # Etherlink III-gebaseerde kaarten
device    fe        # Fujitsu MB8696x-gebaseerde kaarten
device    ie        # EtherExpress 8/16, 3C507, StarLAN 10, etc.
device    lnc       # NE2100, NE32-VL Lance Ethernet kaarten
device    sn        # SMC's 9000 serie Ethernet chips
```

```
device    xe        # Xircom pccard Ethernet

# ISA apparaten die de oude ISA shims gebruiken
#device   le
```

ISA Ethernetstuurprogramma's. In /usr/src/sys/i386/conf/NOTES staan details over welke kaarten door welk stuurprogramma ondersteund worden.

```
# Draadloze NIC kaarten
device    wlan      # 802.11 ondersteuning
```

Generieke 802.11 ondersteuning. Deze regel is vereist voor draadloos netwerken.

```
device          wlan_wep  # 802.11 WEP-ondersteuning
device          wlan_ccmp   # 802.11 CCMP-ondersteuning
device          wlan_tkip   # 802.11 TKIP-ondersteuning
```

Crypto-ondersteuning voor 802.11-apparaten. Deze regels zijn nodig als u van plan bent om versleuteling en 802.11i-beveiligingsprotocollen te gebruiken.

```
device          an              # Aironet 4500/4800 802.11 draadloze NIC's.
device          ath             # Atheros PCI/CardBus NICs
device          ath_hal         # Atheros HAL (Hardware Access Layer)
device          ath_rate_sample # SampleRate verzendsnelheidbeheer voor ath
device          awi        # BayStack 660 en anderen
device          ral        # Ralink Technologies RT2500 draadloze NICs.
device          wi         # WaveLAN/Intersil/Symbol 802.11 draadloze NIC's.
#device         wl         # Oudere niet-802.11 Wavelan draadloze NIC.
```

Ondersteuning voor verscheidene draadloze kaarten.

```
# Pseudo-apparaten
device    loop      # Netwerk teruglussen
```

Dit is het generieke teruglusapparaat voor TCP/IP. Als telnet of FTP op `localhost` (ook bekend als `127.0.0.1`) gebruikt wordt, loopt dat via dit apparaat. Dit is *verplicht*.

```
device    random    # Entropy apparaat
```

Cryptografisch veilige willekeurige getallengenerator.

```
device    ether     # Ethernet ondersteuning
```

`ether` is allen noodzakelijk als er een Ethernetkaart aanwezig is. Het bevat code voor het generieke Ethernetprotocol.

```
device    sl        # Kernel SLIP
```

`sl` dient voor SLIP-ondersteuning. Dit is bijna geheel overgenomen door PPP, wat eenvoudiger is op te zetten, beter geschikt is voor modem-naar-modem-verbindingen en krachtiger is.

```
device    ppp       # Kernel PPP
```

Dit dient voor PPP-ondersteuning van inbelverbindingen door de kernel. Er is ook een versie van PPP als gebruikersapplicatie geïmplementeerd die `tun` gebruikt en meer flexibiliteit en mogelijkheden biedt zoals demand-bellen.

```
device    tun       # Packet tunnel.
```

Dit wordt gebruikt door de gebruikers-PPP-software. In PPP staat meer informatie.

```
device    pty       # Pseudo-ttys (telnet, etc.)
```

Dit is een "pseudo-terminal" of gesimuleerde aanmeldpoort. Die wordt gebruikt door binnenkomende sessies van `telnet` en `rlogin`, door xterm en voor sommige andere applicaties zoals Emacs.

```
device    md        # Geheugenschijven
```

Pseudo-apparaten die een schijf in het geheugen implementeren.

```
device    gif       # IPv6 en IPv4 tunnelen
```

Dit implementeert IPv6-over-IPv4-tunneling, IPv4-over-IPv6-tunneling, IPv4-over-IPv4-tunneling en IPv6-over-IPv6-tunneling. Het apparaat `gif` is "zelfklonend" en zal naar behoefte apparaatknooppunten aanmaken.

```
device    faith     # IPv6-naar-IPv4-relay (vertaling)
```

Dit pseudo-apparaat onderschept pakketten die ernaar verzonden worden en leidt ze om naar het IPv4/IPv6-vertaaldaemon.

```
# Het `bpf' apparaat schakelt de Berkeley Pakketfilter in.
# Wees bewust van de administratieve consequenties die dit heeft!
```

```
# 'bpf' is nodig bij gebruik van DHCP.
device    bpf       # Berkeley pakketfilter
```

Dit is het Berkeley Pakketfilter. Dit pseudo-apparaat staat netwerkinterfaces toe om in luistermodus gezet te worden, zodat elk pakket op een uitzendnetwerk (bijvoorbeeld een Ethernet) onderschept wordt. Deze pakketten kunnen naar schijf onderschept en/of onderzocht worden met het programma tcpdump(1).



Het apparaat bpf(4) wordt ook gebruikt door dhclient(8) om het IP-adres van de standaardrouter (gateway) te verkrijgen, enzovoorts. Als DHCP gebruikt wordt, dient dit ingeschakeld te blijven.

```
# USB-ondersteuning
device          uhci          # UHCI PCI->USB interface
device          ohci          # OHCI PCI->USB interface
device          ehci          # EHCI PCI->USB interface (USB 2.0)
device          usb           # USB Bus (verplicht)
#device         udbp          # USB Double Bulk Pipe apparaten
device          ugen          # Generic
device          uhid          # Human Interface Devices
device          ukbd          # Toetsenbord
device          ulpt          # Printer
device          umass         # Schijven/Massaopslag - heeft scbus en da nodig
device          ums           # Muis
device          ural          # Ralink Technology RT2500USB draadloze NICs
device          urio          # Diamond Rio 500 MP3 speler
device          uscanner      # Scanners
# USB Ethernet, heeft mii nodig
device          aue           # ADMtek USB Ethernet
device          axe           # ASIX Electronics USB Ethernet
device          cdce          # Generic USB over Ethernet
device          cue           # CATC USB Ethernet
device          kue           # Kawasaki LSI USB Ethernet
device          rue           # RealTek RTL8150 USB Ethernet
```

Ondersteuning voor verscheidene USB-apparaten.

```
# FireWire ondersteuning
device    firewire  # FireWire bus code
device    sbp       # SCSI over FireWire (scbus en da nodig)
device    fwe       # Ethernet over FireWire (niet-standaard!)
```

Ondersteuning voor verscheidene Firewire-apparaten.

Meer informatie en aanvullende apparaten die door FreeBSD ondersteund worden staan in /usr/src/sys/i386/conf/NOTES.

8.6.1. Instellingen bij veel geheugen (PAE)

Sommige machines (PAE) hebben meer geheugen nodig dan limiet van 4 gigabyte op User+Kernel Virtual Adress (KVA) ruimte. Vanwege deze limiet voegde Intel ondersteuning toe voor toegang tot 36-bits fysieke adresruimte in de Pentium® Pro en nieuwere lijn van CPU's.

De Physical Address Extension (PAE) mogelijkheden van de Intel® Pentium® Pro en nieuwere CPU's staan geheugenhoeveelheden toe tot 64 gigabyte. FreeBSD biedt ondersteuning voor deze mogelijkheid via de kernelinsteloptie `PAE`, die beschikbaar is in alle recent uitgegeven versies van FreeBSD. Vanwege de beperkingen van de geheugenarchitectuur van Intel wordt er geen onderscheid gemaakt tussen geheugen boven of beneden 4 gigabytes. Geheugen dat boven de 4 gigabytes is toegewezen wordt gewoon bij het beschikbare gevoegd.

Om ondersteuning voor PAE in de kernel aan te zetten, dient de volgende regel aan het kernelinstellingenbestand te worden toegevoegd:

```
options   PAE
```



De ondersteuning voor PAE in FreeBSD is alleen beschikbaar voor Intel® IA-32-processoren. Ook dient opgemerkt te worden dat ondersteuning voor PAE nog niet wijdverbreid getest is en als betakwaliteit beschouwd dient te worden vergeleken met andere stabiele kenmerken van FreeBSD.

Ondersteuning voor PAE in FreeBSD heeft enige beperkingen:

- Een proces kan niet meer dan 4 gigabyte VM-ruimte krijgen;
- Apparaatstuurprogramma's die geen gebruik maken van de bus_dma(9)-interface zullen gegevenscorruptie veroorzaken in een kernel die PAE aan heeft staan en hun gebruik wordt afgeraden. Om deze reden wordt er de kernelinstellingenbestand voor de PAE-kernel geleverd met FreeBSD, dat alle stuurprogramma's uitsluit waarvan niet bekend is dat ze werken in een kernel die PAE aan heeft staan;
- Sommige systeeminstellingen bepalen het geheugenbronverbruik aan de hand van de hoeveelheid beschikbaar fysiek geheugen. Zulke instellingen kunnen onnodig veel toewijzen vanwege de grote hoeveelheid geheugen in een PAE systeem. Een voorbeeld hiervan is de sysctl `kern.maxvnodes`, die het maximum aantal vnodes dat in de kernel aanwezig mag zijn beheert. Het is aan te raden om deze en andere van dit soort instellingen aan te passen aan een redelijke waarde;
- Het kan nodig zijn om de virtuele kerneladresruimte (KVA) te vergroten of om het aantal kernelbronnen dat veel gebruikt wordt (zie boven) te verminderen om zo uitputting van KVA te voorkomen. De kerneloptie `KVA_PAGES` kan gebruikt worden om de KVA-ruimte te vergroten.

Om prestatie- en stabiliteitsredenen is het aan te raden om tuning(7) te raadplegen. pae(4) bevat bijgewerkte informatie over de ondersteuning voor PAE in FreeBSD.

8.7. Problemen oplossen

Er zijn vier probleemcategoriën die op kunnen treden tijdens het bouwen van een aangepaste kernel:

`config` faalt

Als het commando config(8) faalt bij het verwerken van de kernelbeschrijving, is er waarschijnlijk ergens een eenvoudige fout gemaakt. Gelukkig geeft config(8) het nummer van de regel weer waarmee het problemen had, dus kan snel de regel gevonden worden waarin de fout zit. In het onderstaande voorbeeld dient gecontroleerd te worden of het sleutelwoord juist is ingevoerd door het met de kernel GENERIC of een andere referentie te vergelijken:

```
config: line 17: syntax error
```

`make` faalt

Als `make` faalt, duidt dit meestal op een fout in de kernelbeschrijving die niet erg genoeg is om door config(8) opgemerkt te worden. De instellingen dienen nogmaals nagekeken te worden. Als het probleem nog steeds niet is op te lossen, stuur dan een mail naar de FreeBSD algemene vragen mailinglijst met de kernelinstellingen. Dat leidt meestal snel tot een diagnose.

De kernel start niet op

Als de nieuwe kernel niet opstart of de apparaten niet herkent is kalmte geboden. FreeBSD heeft een uitstekend mechanisme om van niet-compatibele kernels te herstellen. De gewenste kernel om mee op te starten kan vanuit de FreeBSD boot loader gekozen worden. Als het systeemopstartmenu verschijnt, kan deze gekozen worden. Selecteer de optie "Escape to a loader prompt", nummer zes. Typ op de prompt `boot kernel.old` of de naam van een andere kernel die correct opstart. Als de kernelinstellingen gewijzigd worden, is het altijd aan te raden om een kernel bij de hand te houden waarvan bekend is dat die juist werkt.

Nadat er met een goede kernel is opgestart, kan het instellingenbestand gecontroleerd worden en geprobeerd worden om de kernel nogmaals te bouwen. Een behulpzame bron is het bestand /var/log/messages, dat onder andere alle kernelberichten van alle keren dat er succesvol is opgestart vastlegt. Ook geeft dmesg(8) alle kernelberichten weer van de huidige opstartprocedure.



Als er problemen zijn met het bouwen van een kernel, dient een GENERIC, of een andere kernel waarvan bekend is dat die werkt, bewaard te worden onder een andere naam die niet verwijderd wordt als de volgende kernel gebouwd wordt. Er kan niet op kernel.old vertrouwd worden omdat bij de installatie van een nieuwe kernel kernel.old overschreven wordt met de laatst geïnstalleerde kernel, die niet hoeft te werken. Ook dient de werkende kernel zo snel mogelijk naar de juiste plaats /boot/kernel verplaatst te worden, omdat anders commando's als ps(1) eventueel onjuist werken. Hiervoor dient simpelweg de map met de goede kernel hernoemd te worden:

```
# mv /boot/kernel /boot/kernel.slecht
```

```
# mv /boot/kernel.goed /boot/kernel
```

De kernel werkt, maar ps(1) werkt niet meer

Als er een andere versie van de kernel is geïnstalleerd dan degene waarmee de systeemgereedschappen gebouwd zijn, bijvoorbeeld een kernel voor -CURRENT op een -RELEASE-systeem, werken vele systeemstatuscommando's als ps(1) en vmstat(8) niet langer. De wereld moet opnieuw gecompileerd en geïnstalleerd worden en met dezelfde broncodestructuur als de kernel zijn gebouwd. Dit is een van de redenen waarom het normaliter geen goed idee is om een afwijkende versie van de kernel ten opzichte van de rest van de wereld te gebruiken.

Hoofdstuk 9. Afdrukken

9.1. Overzicht

FreeBSD kan gebruikt worden om op een scala aan printers af te drukken, van de oudste matrixprinter tot de nieuwste laserprinters en alles er tussenin, waardoor op hoge kwaliteit afgedrukt kan worden vanuit de gebruikte programma's.

FreeBSD kan ook ingesteld worden als printserver in een netwerk. Dan kan FreeBSD afdrukopdrachten ontvangen van uiteenlopende computers, waaronder FreeBSD computers, Windows® en Mac OS® hosts. FreeBSD zorgt ervoor dat er één afdrukopdracht per keer wordt afgedrukt, houdt statistieken bij van gebruikers en computers die de meeste afdrukken maken, drukt "voorbladen" af, zodat duidelijk is van wie de afdruk is en nog veel meer.

Na het lezen van dit hoofdstuk weet de lezer:

- Hoe het FreeBSD afdrukwachtrijsysteem moet worden ingesteld;
- Hoe afdrukfilters kunnen worden geïnstalleerd, om bepaalde afdrukopdrachten op een andere manier af te handelen, zoals het omzetten van documenten naar formaten die een printer begrijpt;
- Hoe voorbladen kunnen worden afgedrukt;
- Hoe er op printers die op andere computers zijn aangesloten kan worden afgedrukt;
- Hoe er op printers die direct op het netwerk zijn aangesloten kan worden afgedrukt;
- Hoe afdrukbeperkingen kunnen worden opgelegd, zoals het beperken van de grootte van de afdrukopdracht, en bepaalde gebruikers verbieden af te drukken;
- Hoe afdrukstatistieken kunnen worden bijgehouden en het printergebruik in de gaten kan worden gehouden;
- Hoe problemen met afdrukken kunnen worden opgelost.

Aangeraden voorkennis:

- Hoe een nieuwe kernel wordt ingesteld, gebouwd en geïnstalleerd (De FreeBSD-kernel instellen).

9.2. Inleiding

Om printers onder FreeBSD te kunnen gebruiken moeten ze kunnen werken met het Berkeley lijnafdrukwachtrijsysteem, ook wel bekend als het wachtrijsysteem LPD of simpelweg LPD. Dit is het standaard afdruksysteem onder FreeBSD. Dit hoofdstuk introduceert LPD en begeleidt bij het instellen.

Als de gebruiker al bekend is met LPD of een ander afdrukwachtrijsysteem, dan kan verder worden lezen vanaf Standaardinstallatie.

LPD regelt alles met betrekking tot de printer van een host. Het is verantwoordelijk voor een aantal

zaken:

- Het regelt de toegang tot aangesloten printers en printers die op andere hosts op het netwerk zijn aangesloten;
- Het geeft gebruikers de mogelijkheid bestanden aan te bieden die afgedrukt moeten worden; deze aangeboden bestanden worden *afdrukopdrachten* genoemd;
- Het voorkomt dat gebruikers tegelijkertijd een printer benaderen door een *wachtrij* bij te houden voor elke printer;
- Het kan *voorbladen* afdrukken (in het Engels ook wel bekend als *banner*, of *burst* pagina's) zodat gebruikers hun afdruk tussen andere afdrukken makkelijk terug kunnen vinden;
- Het handelt de communicatie af voor printers die op een seriële poort zijn aangesloten;
- Het kan afdrukopdrachten over een netwerk versturen naar een LPD wachtrij op een andere host;
- Het kan speciale filters aanroepen die afdrukopdrachten converteren voor verschillende printertalen of afdrukmogelijkheden;
- Het houdt statistieken bij van het printergebruik.

Middels een instellingenbestand (/etc/printcap) en door speciale filters beschikbaar te stellen, kan het LPD systeem alle, of enkele van bovenstaande taken uitvoeren op een grote verscheidenheid aan afdrukhardware.

9.2.1. Waarom het wachtrijsysteem gebruikt zou moeten worden

Het wachtrijsysteem biedt nog steeds voordelen op een systeem met een enkele gebruiker en dient gebruikt te worden omdat:

- LPD afdrukopdrachten in de achtergrond afhandelt. Dan is het niet nodig te wachten tot de gegevens naar de printer zijn verzonden;
- LPD op eenvoudige wijze een afdrukopdracht door een filter kan afdrukken om kopteksten met datum/tijd toe te voegen of een speciaal bestandsformaat (zoals een TeX DVI-bestand) om te zetten naar een formaat dat de printer begrijpt. Deze handelingen hoeven dan niet handmatig uitgevoerd te worden;
- Veel gratis en commerciële software met een afdrukfunctie verwacht dat er een wachtrijsysteem aanwezig is op een systeem om afdrukopdrachten naar te sturen. Door een wachtrijsysteem op te zetten, wordt toekomstig te installeren of reeds geïstalleerde software op eenvoudige wijze ondersteund.

9.3. Standaardinstallatie

Om printers met het LPD wachtrijsysteem te kunnen gebruiken, dienen zowel de printerhardware als de LPD software geïnstalleerd te worden. Dit document beschrijft deze installatie in twee stappen:

- In het onderdeel Eenvoudige printerinstallatie staat hoe een printer moet worden aangesloten, hoe LPD er mee kan communiceren en hoe tekstbestanden afgedrukt kunnen worden.

- In Geavanceerde printerinstallatie staat beschreven hoe een scala aan bestandsformaten afgedrukt kan worden, hoe voorbladen kunnen worden afgedrukt en hoe statistieken van de printer kunnen worden bijgehouden.

9.3.1. Eenvoudige printerinstallatie

Nu wordt toegelicht hoe de printerhardware en de LPD software ingesteld moeten worden om een printer te kunnen gebruiken. Het behandelt de basis:

- Hardware-instellingen geeft een aantal aanwijzingen voor het aansluiten van een printer op een poort van een computer.
- Software-instellingen toont hoe het instellingenbestand (/etc/printcap) voor het LPD-systeem moet worden opgezet.

Hoe een printer geïnstalleerd moet worden die via een netwerkprotocol gegevens ontvangt, in plaats van een seriële of parallelle poort, staat in Printers met netwerkinterfaces.

Hoewel dit onderdeel "Eenvoudige printerinstallatie" heet, is het redelijk complex. De printer met de computer en het LPD-systeem laten samenwerken is het moeilijkste. De geavanceerde opties, zoals voorbladen en statistieken, zijn relatief makkelijk als de printer eenmaal werkt.

Hardware-instellingen

Hier worden de verschillende manieren waarop een printer op een computer kan worden aangesloten beschreven. Het bespreekt de soorten poorten en kabels en de kernelinstellingen die nodig kunnen zijn om FreeBSD met een printer te laten communiceren.

Als een printer al is aangesloten en succesvol is gebruikt onder een ander besturingssysteem, dan kan waarschijnlijk verder gelezen worden in Software-instellingen.

Poorten en kabels

De printers die tegenwoordig voor PC’s verkocht worden hebben eigenlijk altijd een van de volgende drie poorten:

- *Seriële* poort, ook bekend als RS-232- of COM-poorten, gebruiken een seriële poort op een computer om gegevens naar een printer te sturen. Seriële poorten zijn heel gebruikelijk in de computerindustrie en kabels zijn eenvoudig verkrijgbaar en makkelijk te maken. Seriële poorten hebben soms speciale kabels nodig en vereisen soms het instellen van ingewikkelde communicatieparameters. De meeste seriële poorten hebben een maximale doorvoersnelheid van 115.200 bps waardoor het afdrukken van grote grafische afdrukopdrachten erg onpraktisch wordt.
- *Parallelle* poorten gebruiken een parallelle poort op een computer om gegevens naar een printer te sturen. Parallelle poorten zijn gebruikelijk in de PC-markt en zijn sneller dan RS-232 serieel. Kabels zijn goed verkrijgbaar, maar moeilijker handmatig te vervaardigen. Meestal zijn er geen communicatieparameters bij parallelle poorten, wat het instellen erg eenvoudig maakt.

  Parallelle poorten staan ook wel bekend als "Centronics" poorten, genoemd naar het soort aansluiting op de printer.

- USB poorten, genoemd naar de Universal Serial Bus, kunnen zelfs op nog hogere snelheid werken dan parallelle of RS-232 seriële poorten. De kabels zijn eenvoudig en goedkoop. USB is voor afdrukken superieur aan RS-232 Serieel en Parallel, maar wordt op UNIX®-systemen niet altijd goed ondersteund. Een van de manieren om dit te omzeilen is de aanschaf van een printer met zowel een USB als een parallelle poort, zoals veel printers die hebben.

Over het algemeen kunnen parallelle poorten meestal in één richting communiceren (van computer naar printer), terwijl seriële en USB poorten in twee richtingen kunnen communiceren. Nieuwere parallelle poorten (EPP en ECP) en printers kunnen onder FreeBSD in beide richtingen communiceren, mits een IEEE-1284 gekeurde kabel wordt gebruikt.

Tweewegcommunicatie met een printer over een parallelle poort verloopt meestal op een van de volgende twee manieren. De eerste manier is door gebruik te maken van een op maat gemaakt stuurprogramma voor FreeBSD dat de taal spreekt die door de printer wordt gebruikt. Dit geldt meestal voor inkjet printers en er kan dan gebruikt gemaakt worden van rapportagemogelijkheden over bijvoorbeeld inktniveaus en andere statusinformatie. De tweede methode wordt gebruikt als een printer PostScript® ondersteunt.

PostScript®-taken zijn eigenlijk programma's die naar de printer worden gestuurd. Het hoeft zelfs niet in een afdruk te resulteren; het resultaat van de opdracht kan direct weer naar de computer worden gestuurd. PostScript® gebruikt ook tweewegcommunicatie om een computer op de hoogte te stellen van opgetreden fouten, zoals fouten in het PostScript®-programma of vastgelopen papier. Gebruikers kunnen dit soort informatie handig vinden. De beste manier om bij een PostScript®-printer effectief bij te houden wat het printergebruik is, vraagt om tweewegcommunicatie: de printer wordt gevraagd om het totaal aantal afgedrukt pagina's, de afdrukopdracht wordt verzonden en vervolgens wordt nogmaals om het totaal aantal afgedrukte pagina's gevraagd. Het verschil van deze getallen geeft het aantal afgedrukte pagina's van de afdrukopdracht van de betreffende gebruiker.

Parallelle poorten

Om een printer met een parallelle poort aan te sluiten, moet een Centronics kabel de printer met de computer verbinden. De instructies die geleverd zijn bij de printer, de computer of beide, moeten voldoende zijn om dit te verduidelijken.

Onthoud op welke parallelle poort de printer is aangesloten. De eerste parallelle poort heet onder FreeBSD ppc0, de tweede ppc1, enzovoort. De benaming voor de printer gaat analoog: /dev/lpt0 voor de printer op de eerste parallelle poort enzovoort.

Seriële poorten

Gebruik de juiste seriële kabel om een printer met een seriële poort op een computer aan te sluiten. De instructies die geleverd zijn bij de printer, de computer of beide, moeten voldoende zijn om dit te verduidelijken.

Als onduidelijk is wat de "juiste seriële kabel" is, kan een van onderstaande opties geprobeerd worden:

- Een *modem*kabel verbindt elke pin van de stekker aan het ene eind direct met de corresponderende pin van de stekker aan het andere eind. Dit type kabel heet ook wel een

"DTE-naar-DCE"-kabel.

- Een *null-modem* kabel verbindt enkele pinnetjes direct, verwisselt andere (bijvoorbeeld van verstuur gegevens naar ontvang gegevens) en sluit sommige draden kort in de stekker. Dit type kabel heet ook wel een "DTE-to-DTE"-kabel.
- Een *seriële printer*kabel, nodig bij sommige ongebruikelijke printers, is als een null-modem kabel, maar stuurt sommige signalen naar hun tegenhangers in plaats van ze intern kort te sluiten.

Het is ook nodig de communicatieparameters voor de printer in te stellen, meestal via het bedieningspaneel of middels DIP-schakelaars op de printer. Selecteer de hoogste `bps` (bits per seconde, soms *baud*) die zowel door de computer als de printer wordt ondersteund. Kies 7 of 8 data bits. Geen, even of oneven pariteit en 1 of 2 stop bits. Selecteer ook het flow-control protocol: ofwel geen, ofwel XON/XOFF (ook bekend als "in-band" of "software") flow-control. Onthoud deze instellingen voor de hier op volgende software-instellingen.

Software-instellingen

Nu wordt beschreven welke software-instellingen nodig zijn om onder FreeBSD af te drukken met behulp van het wachtrijsysteem LPD.

Een overzicht van de te doorlopen stappen:

1. Maak, indien nodig, de kernel geschikt voor de poort die door de printer wordt gebruikt. In Kernelinstellingen is te lezen hoe dit gedaan kan worden.
2. Stel de communicatievorm voor de parallelle poort in, als gebruik wordt gemaakt van een parallelle printer. In Communicatietype instellen voor een parallelle poort staan de details.
3. Test of het besturingssysteem gegevens naar de printer kan sturen. In Printercommunicatie controleren staat een aantal suggesties.
4. Stel LPD in voor de printer door /etc/printcap aan te passen. Dat wordt later in het hoofdstuk beschreven.

Kernelinstellingen

Het besturingssysteem is gecompileerd om met een beperkte verzameling apparaten te kunnen werken. De seriële en parallelle poorten zijn onderdeel van deze verzameling. Daarom kan het nodig zijn om ondersteuning voor een extra seriële of parallelle poort toe te voegen als een kernel hier nog niet voor is ingesteld.

Om te achterhalen of de huidige kernel een seriële poort ondersteunt:

```
# grep sioN /var/run/dmesg.boot
```

Hier is *N* het aantal seriële poorten, beginnende bij nul. Als de uitvoer op het volgende lijkt, dan wordt de poort door de kernel ondersteund:

```
sio2 at port 0x3e8-0x3ef irq 5 on isa
sio2: type 16550A
```

Om te achterhalen of de kernel een parallelle poort ondersteunt:

```
# grep ppcN /var/run/dmesg.boot
```

Hier is *N* het aantal parallelle poorten beginnende bij nul. Als de uitvoer er ongeveer als volgt uit ziet, dan wordt de poort door de kernel ondersteund:

```
ppc0: <Parallel port> at port 0x378-0x37f irq 7 on isa0
ppc0: SMC-like chipset (ECP/EPP/PS2/NIBBLE) in COMPATIBLE mode
ppc0: FIFO with 16/16/8 bytes threshold
```

Het kan nodig zijn een kernel aan te passen om het besturingssysteem in staat te stellen een parallelle of seriële poort die voor een printer wordt gebruikt te herkennen en te gebruiken.

In het onderdeel over kernelinstellingen staat meer informatie om ondersteuning voor een seriële poort toe te voegen. Lees de betreffende *en* de volgende sectie om ondersteuning voor een parallelle poort toe te voegen.

Communicatietype instellen voor een parallelle poort

Wanneer een parallelle poort wordt gebruikt, kan worden aangegeven of FreeBSD voor de printer interrupt-gestuurde of "polled" communicatie moet gebruiken. Het generieke printerapparaatstuurprogramma (lpt(4)) onder FreeBSD gebruikt het systeem ppbus(4). Dit bestuurt de chipset van de poort met het stuurprogramma ppc(4).

- De *interrupt-gestuurde* methode is standaard in de GENERIC kernel. In dit geval gebruikt het besturingssysteem een IRQ om te bepalen of de printer klaar is om gegevens te ontvangen.
- Bij de *polled* methode vraagt het besturingssysteem met vaste intervallen aan de printer of deze klaar is om gegevens te ontvangen. Als de printer antwoordt met "klaar", stuurt de kernel meer gegevens.

De interrupt-gestuurde methode is meestal iets sneller, maar gebruikt een kostbaar IRQ-nummer. Van sommige HP printers wordt beweerd dat ze niet goed werken in interruptmodus, schijnbaar door een (nog niet begrepen) timing-probleem. Deze printers moeten gebruik maken van de polled methode. Gebruik de methode die werkt. Sommige printers werken met beide methodes, maar zijn tergend langzaam in de interrupt modus.

Het communicatietype kan op twee manieren worden ingesteld: door de kernel in te stellen of door gebruik te maken van lptcontrol(8).

Het communicatietype instellen door de kernel aan te passen:

1. Pas het kernelinstellingenbestand aan. Zoek naar een `ppc0` ingang. Gebruik `ppc1` voor het

opzetten van een tweede parallelle poort. Gebruik `ppc2` voor de derde poort, enzovoort.

- Als u gebruik wilt maken van de interrupt gestuurde modus, bewerk dan de regel hieronder:

  ```
  hint.ppc.0.irq="N"
  ```

  Het kernelinstellingenbestand moet ook het stuurprogramma ppc(4) bevatten:

  ```
  device ppc
  ```

- Om gebruik te maken van polled modus verwijder dan het volgende regel uit /boot/device.hints:

  ```
  hint.ppc.0.irq="N"
  ```

  In sommige gevallen is het onder FreeBSD niet voldoende om een poort in polled modus te zetten. In veel gevallen komt dat door het stuurprogramma acpi(4). Dit is in staat om apparaten te testen en aan te sluiten en kan zodoende het communicatietype van de printer wijzigen. Raadpleeg de instellingen voor acpi(4) om dit probleem te verhelpen.

2. Sla het bestand op. Maak en installeer de nieuwe kernel en herstart de computer. In De FreeBSD-kernel instellen staan meer details.

Communicatietype instellen met lptcontrol(8):

1. Typ:

   ```
   # lptcontrol -i -d /dev/lptN
   ```

   om `lptN` op interrupt-gestuurde modus in te stellen.

2. Typ:

   ```
   # lptcontrol -p -d /dev/lptN
   ```

   om `lptN` op polled modus in te stellen.

   ```
   # lptcontrol -p -d /dev/lptN
   ```

Zet deze commando's in het bestand /etc/rc.local zodat het communicatietype juist wordt ingesteld

bij het opstarten. In lptcontrol(8) staat meer informatie.

Printercommunicatie controleren

Voor het instellen van het wachtrijsysteem, is het verstandig te controleren of het besturingssysteem gegevens naar een printer kan versturen. Het is een stuk makkelijker om problemen met printercommunicatie en het wachtrijsysteem apart op te lossen.

De printer wordt getest door er tekst naar toe te sturen. Voor printers die direct tekens kunnen afdrukken is het programma lptest(1) handig: het genereert alle 96 afdrukbare ASCII-tekens op 96 regels.

Voor PostScript® (of andere op taal gebaseerde) printers, is een meer geavanceerde test nodig. Een eenvoudig PostScript®-programma zoals het volgende volstaat:

```
%!PS
100 100 moveto 300 300 lineto stroke
310 310 moveto /Helvetica findfont 12 scalefont setfont
(Werkt dit?) show
showpage
```

Bovenstaande PostScript®-code kan in een bestand worden opgeslagen en in de voorbeelden in de volgende paragrafen gebruikt worden.



Als in dit document wordt gesproken over een printertaal, wordt uitgegaan van een taal als PostScript® en niet PCL van HP. Hoewel PCL zeer functioneel is, kan het direct platte tekst afdrukken door gebruik te maken van escape-tekens. PostScript® kan niet direct platte tekst afdrukken. Voor dat soort printertalen zijn speciale aanpassingen nodig.

Parallelle printer controleren

In deze sectie wordt beschreven hoe te controleren of FreeBSD kan communiceren met een printer die op een parallelle poort is aangesloten.

Voer de volgende stappen uit om een printer op een parallelle poort te testen:

1. su(1) naar `root`.
2. Stuur gegevens naar de printer.
   - Gebruik lptest(1) als de printer platte tekst af kan drukken:

     ```
     # lptest > /dev/lptN
     ```

     Hier is *N* het nummer van de parallelle poort, beginnende bij nul.
   - Als de printer PostScript® of een andere printertaal begrijpt, stuur dan een klein

programma naar de printer:

```
# cat > /dev/lptN
```

Geef het programma regel voor regel *heel nauwkeurig* in. Een regel kan niet worden gewijzigd als er op `RETURN` of `ENTER` is gedrukt. Geef na het afronden van de invoer voor het programma het einde-van-invoer-teken. Dit is meestal `CONTROL+D`.

Het programma kan ook in een bestand worden opgeslagen:

```
# cat bestand > /dev/lptN
```

Hier is *bestand* de naam van het bestand waarin het programma is opgeslagen dat naar een printer gestuurd kan worden.

Nu moet er iets worden afgedrukt. Tekst die er niet goed uitziet is geen probleem. Dit wordt later gerepareerd.

Seriële printer controleren

In deze sectie wordt beschreven hoe te controleren of FreeBSD kan communiceren met een printer die op een seriële poort is aangesloten.

Voer de volgende stappen uit om een printer op de seriële poort te testen:

1. su(1) naar `root`.
2. Voeg de volgende regel toe aan /etc/remote:

   ```
   printer:dv=/dev/poort:br#bps-snelheid:pa=pariteit
   ```

   Hier is *poort* de apparaatingave voor de seriële poort (`ttyu0`, `ttyu1`, enzovoort), *bps-snelheid* is het aantal bits per seconde waarop de printer communiceert en *pariteit* is de pariteit die door de printer wordt vereist (`even`, `odd`, `none` of `zero`).

   Hier volgt een voorbeeldregel voor een printer verbonden met een seriële lijn op de derde seriële poort op 19200 bps, zonder pariteit:

   ```
   printer:dv=/dev/ttyu2:br#19200:pa=none
   ```

3. Maak verbinding met de printer met tip(1):

   ```
   # tip printer
   ```

Als dit niet werkt, pas dan /etc/remote opnieuw aan en probeer gebruik te maken van /dev/cuaaN in plaats van /dev/ttyuN.

4. Stuur gegevens naar de printer.
   - Gebruik lptest(1) als de printer platte tekst af kan drukken:

     ```
     % $lptest
     ```

   - Als de printer PostScript® of een andere printertaal begrijpt, stuur dan een klein programma naar de printer. Geef het programma regel voor regel *heel nauwkeurig* in. Backspace of andere speciale toetsen kunnen een speciale betekenis hebben voor de printer. Het kan ook nodig zijn een speciaal einde-van-invoer-teken te geven zodat de printer weet dat het gehele programma ontvangen is. Druk voor PostScript®-printers CONTROL+D.

     Het programma kan ook in een bestand worden opgeslagen:

     ```
     % >bestand
     ```

     Hier is *bestand* de naam van het bestand waarin het programma is opgeslagen. Nadat tip(1) het bestand heeft verstuurd kan het juiste einde-van-invoer-teken ingegeven worden.

Nu moet er iets worden afgedrukt. Tekst die er niet goed uitziet is geen probleem. Dit wordt later gerepareerd.

De wachtrij aanzetten: /etc/printcap

Op dit punt moet de printer zijn aangesloten, de kernel ingesteld zijn om met de printer te communiceren (indien nodig) en is het mogelijk eenvoudige gegevens naar de printer te sturen. Nu kan LPD ingesteld worden zodat de toegang tot de printer wordt geregeld.

LPD wordt ingesteld door het bestand /etc/printcap aan te passen. Het wachtrijsysteem LPD leest dit bestand iedere keer dat het systeem wordt aangeroepen zodat wijzigingen direct van toepassing zijn.

De opmaak van het bestand printcap(5) is voor de hand liggend. Met een willekeurige tekstverwerker kunnen wijzigen in /etc/printcap aangebracht worden. De opmaak is identiek aan die van andere bestanden die voor dergelijke instellingen worden gebruik, zoals /usr/shared/misc/termcap en /etc/remote. In cgetent(3) staat een uitgebreid overzicht van dit formaat.

De vereenvoudigde instellingen bestaan uit de volgende stappen:

1. Kies een naam (en een paar handige aliassen) voor de printer en voeg ze toe aan /etc/printcap. In Printernaamgeving staat meer informatie over het toekennen van een

naam aan een printer.

2. Het afdrukken van voorbladen (standaard) kan uitgezet worden met de optie `sh`. In Voorbladen onderdrukken staat meer informatie.
3. Maak een wachtrijmap aan en specificeer de locatie door middel van de optie `sd`. In Wachtrijmap aanmaken staat meer informatie.
4. Bepaal welke ingave in /dev voor de printer wordt gebruikt en geef dit in /etc/printcap aan door gebruik te maken van de optie `lp`. In Printerapparaat identificeren staat meer informatie. Als de printer is aangesloten op een seriële poort moeten de communicatieparameters worden ingesteld met de optie `ms#`. Dit wordt beschreven in Communicatieparameters voor het wachtrijsysteem instellen.
5. Installeer een filter voor platte tekst. In Tekstfilter installeren staan details.
6. Test de instellingen door iets met lpr(1) af te drukken. Details staan in Printer uitproberen en Problemen oplossen.



Op taal gebaseerde printers, zoals PostScript®-printers, kunnen niet direct platte tekst afdrukken. De vereenvoudigde instellingen, zoals hierboven beschreven en hieronder verder beschreven, gaan er van uit dat alleen bestanden naar een printer worden gestuurd die de printer begrijpt.

Gebruikers verwachten vaak dat ze platte tekst naar printers op een systeem kunnen sturen. Programma's die LPD gebruiken om af te drukken gaan hier ook vaak van uit. Als een dergelijke printer wordt geïnstalleerd en het moet mogelijk zijn zowel afdrukopdrachten in de printertaal als in platte tekst naar een printer te sturen, dan is het zeer aan te raden een extra stap in deze vereenvoudigde opzet in te voegen: installeer een conversieprogramma dat automatisch platte tekst omzet in PostScript® (of een andere printertaal). In Platte tekst op PostScript®-printers afdrukken staat hoe dit in zijn werk gaat.

Printernaamgeving

De eerste (makkelijke) stap is het kiezen van een naam voor een printer. Het maakt niet uit of een naam functioneel of grappig is, aangezien ook een aantal aliassen aan een printer toegekend kunnen worden.

Ten minste één van de printers die in /etc/printcap worden genoemd moet het alias `lp` hebben. Dit is de standaardnaam voor de printer. Als gebruikers de omgevingsvariabele `PRINTER` niet ingesteld hebben en ook geen printernaam specificeren als ze LPD gebruiken, dan wordt standaard de printer `lp` gebruikt.

Het is verder gebruikelijk om het laatste alias zo te kiezen dat het een volledige beschrijving van de printer is, inclusief merk en model.

Als een naam en een aantal aliassen zijn gekozen, kunnen ze aan /etc/printcap worden toegevoegd. De naam van een printer wordt in de meest linker kolom geplaatst. Scheid ieder alias met een verticale streep en plaats een dubbele punt achter het laatste alias.

In het volgende voorbeeld is de beginsituatie een uitgekleed /etc/printcap waarin twee printers

worden gedefinieerd (een Diablo 630 lijnprinter en een Panasonic KX-P4455 PostScript®-laserprinter):

```
#
#  /etc/printcap voor host rose
#
rattan|line|diablo|lp|Diablo 630 Line Printer:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:
```

In dit voorbeeld heet de eerste printer `rattan` en heeft de volgende aliassen: `line`, `diablo`, `lp` en `Diablo 630 Line Printer`. Omdat deze printer het alias `lp` heeft, is het de standaard printer. De tweede printer heet `bamboo` en heeft de aliassen `ps`, `PS`, `S`, `panasonic` en `Panasonic KX-P4455 PostScript v51.4`.

Voorbladen onderdrukken

Het wachtrijsysteem LPD drukt standaard een *voorblad* af voor elke afdrukopdracht. Het voorblad bevat de gebruikersnaam van de gebruiker die de afdrukopdracht gaf, de computer waar de opdracht is gegeven en, in mooie grote letters, de naam van de afdrukopdracht. Het nadeel hiervan is dat al deze extra tekst het debuggen van de eenvoudige printerinstallatie bemoeilijkt. Daarom wordt het afdrukken van voorbladen onderdrukt.

Om voorbladen te onderdrukken, wordt de optie `sh` toegevoegd voor de relevante printer in /etc/printcap. Hieronder staat een voorbeeld van /etc/printcap met de optie `sh`:

```
#
#  /etc/printcap voor host rose - nergens worden voorbladen afgedrukt
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:
```

Het juiste formaat is gebruikt: de eerste regel begint in de meest linker kolom, volgende regels springen in. Elke regel eindigt met een backslash, behalve de laatste.

Wachtrijmap aanmaken

De volgende stap in deze eenvoudige opzet is het aanmaken van een *wachtrijmap*. Dit is een map waar afdrukopdrachten geplaatst worden totdat ze worden afgedrukt. Ook wordt er een aantal bestanden geplaatst die nodig zijn voor het functioneren van het wachtrijsysteem.

Vanwege het veranderlijke karakter van wachtrijmappen is het gebruikelijk om deze mappen onder /var/spool te plaatsen. Het is niet nodig om een reservekopie van de inhoud van deze mappen te maken. Ze kunnen eenvoudigweg opnieuw worden aangemaakt met mkdir(1).

Het is ook gebruikelijk om de naam van de map overeen te laten komen met die van de printer, zoals onder is weergegeven:

```
# mkdir /var/spool/printernaam
```

Als er veel printers zijn aangesloten op een netwerk, is het beter de wachtrijmappen aan te maken in een enkele map die speciaal wordt gebruikt voor afdrukken met LPD. In dit voorbeeld wordt dat gedaan voor de printers `rattan` en `bamboo`:

```
# mkdir /var/spool/lpd
# mkdir /var/spool/lpd/rattan
# mkdir /var/spool/lpd/bamboo
```



Als de afdrukopdrachten privé moeten blijven, dan is het belangrijk de wachtrijmap niet algemeen toegankelijk te maken. Wachtrijmappen moeten eigendom zijn van gebruiker `daemon` en groep `daemon`. Uitsluitend deze gebruiker en groep moeten de map kunnen lezen, schrijven en doorzoeken. We doen dit voor onze voorbeeldprinters:

```
# chown daemon:daemon /var/spool/lpd/rattan
# chown daemon:daemon /var/spool/lpd/bamboo
# chmod 770 /var/spool/lpd/rattan
# chmod 770 /var/spool/lpd/bamboo
```

Tenslotte moet LPD verteld worden dat deze mappen bestaan. Dit kan met het bestand /etc/printcap. De locatie van de wachtrijmap wordt opgegeven met de optie `sd`:

```
#
#  /etc/printcap voor host rose - wachtrijmappen toegevoegd
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:
```

De naam van de printer staat in de eerste kolom, maar alle andere regels die de printer beschrijven worden ingesprongen en elke regel eindigt met een backslash.

Als geen wachtrijmap wordt opgegeven met `sd`, dan wordt standaard /var/spool/lpd gebruikt.

Printerapparaat identificeren

In de sectie Hardware-instellingen is bepaald welke poort en ingang in de map /dev door FreeBSD worden gebruikt om met een printer te communiceren. Nu moet LPD dit ook weten. Als het

wachtrijsysteem een afdrukopdracht krijgt, opent het het relevante apparaat namens het filterprogramma (dat verantwoordelijk is voor het sturen van gegevens naar een printer).

Geef de locatie van de ingang in /dev op in /etc/printcap door gebruik te maken van de optie `lp`.

In het huidige voorbeeld wordt aangenomen dat `rattan` op de eerste parallelle poort is aangesloten en `bamboo` op de zesde seriële poort. Hier volgen de toevoegingen voor /etc/printcap:

```
#
#  /etc/printcap voor host rose - bepaald welke apparaten te gebruiken
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:\
        :lp=/dev/lpt0:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:\
        :lp=/dev/ttyu5:
```

Als voor een printer de optie `lp` niet wordt gebruikt in /etc/printcap, dan gebruikt LPD standaard /dev/lp. Momenteel bestaat /dev/lp niet in FreeBSD.

Als de te installeren printer is aangesloten op een parallelle poort, dan staan verdere instructies in Tekstfilter installeren. In andere gevallen kunnen de instructies in de volgende paragraaf gevold worden.

Communicatieparameters voor het wachtrijsysteem instellen

Voor printers die zijn aangesloten op een seriële poort kan LPD de bps-snelheid, pariteit en andere seriële communicatie parameters instellen voor het filterprogramma dat gegevens naar een printer stuurt. Dit is gunstig omdat:

- De verschillende communicatieparameters uitgeprobeerd kunnen worden door /etc/printcap aan te passen. Het is niet nodig het filterprogramma opnieuw te compileren;
- Het wachtrijsysteem kan hetzelfde filter gebruiken voor verschillende printers die mogelijk verschillende seriële communicatie-instellingen hebben.

Met de volgende opties in /etc/printcap kunnen seriële communicatieparameters worden ingesteld voor het apparaat waar `lp` naar verwijst:

`br#bps-snelheid`

Stelt de communicatiesnelheid van het apparaat in op *bps-snelheid*, waarbij *bps-snelheid* de waarde 50, 75, 110, 134, 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600 of 115200 bits-per-seconde kan aannemen.

`ms#stty-modus`

Bepaalt de opties voor het geval het printerapparaat een terminal is. In stty(1) staat uitleg over de beschikbare opties.

Als LPD het apparaat opent dat met `lp` is opgegeven, worden de eigenschappen van het apparaat bepaald door de optie `ms#`. Met name van belang zijn de modi `parenb`, `parodd`, `cs5`, `cs6`, `cs7`, `cs8`, `cstopb`, `crtscts` en `ixon`. Deze worden uitgelegd in stty(1).

Nu wordt de voorbeeldprinter op de zesde seriële poort aangepast. De bps-snelheid wordt ingesteld op 38400. Als modus wordt gekozen: geen pariteit met `-parenb`, 8-bit tekens met `cs8`, geen modemcontrole met `clocal` en hardware flow-control met `crtscts`:

```
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:\
        :lp=/dev/ttyu5:ms#-parenb cs8 clocal crtscts:
```

Tekstfilter installeren

Nu kan LPD verteld worden welke tekstfilters gebruikt moeten worden bij het versturen van afdrukopdrachten. Een *tekstfilter* is een programma dat LPD aanroept als het een afdrukopdracht krijgt. Wanneer LPD het tekstfilter aanroept, wordt de standaard invoer van het filter gekoppeld aan de afdrukopdracht en de standaard uitvoer aan het printerapparaat dat door de optie `lp` is opgegeven. Er wordt aangenomen dat het filter van standaard invoer leest, vervolgens de nodige handelingen uitvoert en het resultaat naar de standaard uitvoer schrijft, zodat het afgedrukt wordt. In Filters staat meer informatie over het tekstfilter.

Voor deze eenvoudige printerinstallatie kan het tekstfilter een klein shellscript zijn dat `/bin/cat` aanroept om de afdrukopdracht naar de printer te sturen. FreeBSD wordt geleverd met een ander filter, lpf, dat backspaces en onderlijnde tekst afhandelt voor printers die hier niet mee overweg kunnen. Natuurlijk kan elk filter gebruikt worden dat gewenst is. Het filter `lpf` wordt uitgebreid beschreven in lpf: een tekstfilter.

Nu wordt eerst het shellscript /usr/local/libexec/if-simple gemaakt dat als simpel tekstfilter dient. Plaats de volgende tekst in het bestand met een tekstverwerker naar keuze:

```
#!/bin/sh
#
# if-simple - Eenvoudig tekstfilter voor lpd
# Geïnstalleerd in /usr/local/libexec/if-simple
#
# Kopieert eenvoudigweg stdin naar stdout.
# Filterargumenten worden genegeerd.

/bin/cat && exit 0
exit 2
```

Zorg dat het bestand uitvoerbaar is:

```
# chmod 555 /usr/local/libexec/if-simple
```

Zorg dat LPD het filter gebruikt door dit aan te geven met de optie `if` in /etc/printcap. Nu volgt hoe

dit te doen voor de twee printers uit het voorbeeld:

```
#
#  /etc/printcap voor host rose - met tekstfilter
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:\ :lp=/dev/lpt0:\
        :if=/usr/local/libexec/if-simple:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:\
        :lp=/dev/ttyu5:ms#-parenb cs8 clocal crtscts:\
        :if=/usr/local/libexec/if-simple:
```



Een kopie van het script if-simple staat in de map /usr/shared/examples/printing.

LPD aanzetten

lpd(8) wordt gestart vanuit /etc/rc door de variabele `lpd_enable`. Standaard staat deze variabele op `NO`. Als dit nog niet is gedaan, voeg dan de volgende regel toe aan /etc/rc.conf en herstart de computer:

```
lpd_enable="YES"
```

Of voer het commando lpd(8) uit:

```
# lpd
```

Printer uitproberen

Nu volgt het laatste onderdeel van de eenvoudige LPD installatie. Helaas zijn felicitaties nog niet gepast. De printer moet worden getest en eventuele problemen moeten worden opgelost. Om de installatie te testen kan iets afgedrukt worden. Afdrukken gaat met het commando lpr(1). Dit stuurt een opdracht naar een printer.

Het programma lpr(1) is te combineren met het programma lptest(1) uit Printercommunicatie controleren om tekst te genereren.

Om de eenvoudige installatie van LPD te testen:

```
# lptest 20 5 | lpr -Pprinternaam
```

Hier is *printernaam* de naam van een printer (of een alias) die in /etc/printcap wordt genoemd. De standaard printer kan worden getest door bij het aanroepen van lpr(1) de optie `-P` weg te laten. Nogmaals: test een PostScript®-printer door een PostScript®-programma naar een printer te sturen en maak geen gebruik van lptest(1). Dit kan door het programma in een bestand op te slaan en de

volgende commandoregel uit te voeren: `lpr bestand`.

Voor een PostScript®-printer moet het resultaat van het programma verschijnen. Als gebruik wordt gemaakt van lptest(1) ziet het resultaat er ongeveer zo uit:

```
!"#$%&'()*+,-./01234
"#$%&'()*+,-./012345
#$%&'()*+,-./0123456
$%&'()*+,-./01234567
%&'()*+,-./012345678
```

Om de printer uitvoeriger te testen kunnen grotere programma's geprobeerd worden (voor taalgebaseerde printers) of kan lptest(1) aangeroepen worden met andere argumenten. Bijvoorbeeld: `lptest 80 60`, drukt 60 regels af met elk 80 karakters.

Als de printer niet werkt, lees dan verder in Problemen oplossen.

9.4. Geavanceerde printerinstallatie

Deze sectie behandelt het gebruik van filters om speciaal opgemaakte tekst en voorbladen af te drukken, via het netwerk af te drukken en printergebruik te beperken en statistieken bij te houden.

9.4.1. Filters

Hoewel LPD veel van het afdrukwerk afhandelt (netwerkverkeer, wachtrijafhandeling, toegangscontrole, enzovoort), wordt het *echte* werk door de filters gedaan. Filters zijn programma's die met een printer communiceren en inspelen op printerspecifieke eigenschappen. In de eenvoudige printeropzet is een filter geïnstalleerd voor platte tekst, een zeer eenvoudig filter dat met de meeste printers zou moeten werken (Tekstfilter installeren).

Om echter gebruik te maken van formaatomzetting, printeradministratie, printerspecifieke aanpassingen, enzovoort, is het nodig te weten hoe filters werken. Uiteindelijk is het de verantwoordelijkheid van het filter om deze zaken af te handelen. Het slechte nieuws is dat *de beheerder* in het merendeel van de gevallen het filter moet aanleveren. Het goede nieuws is dat veel filters algemeen beschikbaar zijn en als ze dat niet zijn, zijn ze vaak makkelijk te schrijven.

FreeBSD heeft een ingebouwd filter, /usr/libexec/lpr/lpf, die met veel printers werkt die platte tekst kunnen afdrukken. Het filter regelt backspace en tabs in bestanden en administreert printergebruik, maar dat is zo'n beetje alles wat dit filter doet. Er zijn ook diverse filters en filtercomponenten in de FreeBSD Portscollectie.

Hieronder wordt het volgende beschreven:

- In Hoe filters werken staat een overzicht van de rol die een filter speelt in het afdrukproces. Lees dat onderdeel om een indruk te krijgen wat er "onder de motorkap" gebeurt als LPD filters gebruikt. Deze kennis helpt mogelijke problemen te voorkomen of op te lossen als meerdere filters worden geïnstalleerd voor printers.
- LPD gaat er van uit dat elke printer standaard platte tekst af kan drukken. Dit geeft problemen

voor PostScript® (of andere op taal gebaseerde) printers die niet in staat zijn direct platte tekst af te drukken. In Platte tekst op PostScript®-printers afdrukken staat wat er kan worden gedaan om dit probleem te verhelpen. Lees verder in dit onderdeel als het om PostScript®-printers gaat.

- Voor veel programma's is PostScript® een populair uitvoerformaat. Sommige mensen schrijven PostScript® code zelfs direct. PostScript®-printers zijn echter kostbaar. In PostScript® simuleren op niet-PostScript®-printers staat hoe de tekstfilter van een printer aangepast moet worden zodat die PostScript® accepteert en afdrukt op een *niet-PostScript*®-printer. Dit onderdeel is van toepassing voor niet-PostScript®-printers.
- In Conversiefilters wordt een methode beschreven om de conversie van bepaalde bestandsformaten te automatiseren, zoals van grafische of tekstopmaakprogramma's, naar formaten die een printer kan begrijpen. Na het lezen van dit onderdeel is een beheerder in staat om een printer zodanig in te stellen dat gebruikers `lpr -t` kunnen invoeren om troff-gegevens af te drukken, `lpr -d` om TeX DVI-gegevens af te drukken of `lpr -v` om rasterplaatjes af te drukken, enzovoorts. Het wordt aangeraden deze sectie te lezen.
- In Uitvoerfilters wordt een niet vaak gebruikte functionaliteit van LPD behandeld: uitvoerfilters. Tenzij voorbladen worden afgedrukt (Voorbladen), kan deze sectie waarschijnlijk overgeslagen worden.
- lpf: een tekstfilter beschrijft `lpf`, een redelijk complete, eenvoudige tekstfilter voor lijnprinters (en laserprinters die zich als lijnprinters voordoen) dat wordt geleverd bij FreeBSD. Voor een snelle manier om printeradministratie aan de praat te krijgen voor platte tekst of voor printers waar rook uit komt bij het zien van backspace karakters, is het serieus te overwegen gebruik te maken van `lpf`.



Een kopie van de scripts die hieronder worden beschreven, staan in de map /usr/shared/examples/printing.

Hoe filters werken

Zoals eerder genoemd, is een filter een programma dat wordt uitgevoerd door LPD voor het afhandelen van het apparaatafhankelijke deel van de communicatie met een printer.

Als LPD een bestand wil afdrukken uit een afdrukopdracht, start het een filterprogramma. Het koppelt de standaard invoer van de filter aan het af te drukken bestand, de standaard uitvoer aan de printer en de standaard foutmelding aan het logboekbestand voor foutmeldingen (zoals opgegeven via de optie `lf` in /etc/printcap of standaard /dev/console).

Welk filter LPD start en de argumenten van het filter hangen af van wat er in het bestand /etc/printcap wordt opgegeven en de argumenten die de gebruiker geeft op de commandoregel van lpr(1). Als een gebruiker bijvoorbeeld `lpr -t` ingeeft, start LPD het filter troff, zoals wordt opgegeven via de optie `tf` voor de betreffende printer. Als een gebruiker platte tekst wilt afdrukken, dan wordt het filter `if` gestart (dit klopt bijna: zie Uitvoerfilters voor de details).

Er zijn drie soorten filters die in /etc/printcap kunnen worden opgegeven:

- Het tekstfilter, dat in de LPD documentatie verwarrend genoeg *input filter* wordt genoemd, verwerkt het afdrukken van gewone tekst. Beschouw het als het standaardfilter. LPD verwacht

dat elke printer standaard platte tekst kan afdrukken en het is de taak van het tekstfilter om er voor te zorgen dat backspaces, tabs en andere speciale karakters de printer niet in de war sturen. In een omgeving waar moet worden bijgehouden hoeveel er wordt afgedrukt, moet het tekstfilter ook administreren hoeveel pagina's er zijn afgedrukt. Dit gaat meestal door het aantal afgedrukte regels te tellen en dit te vergelijken met het aantal regels per pagina dat door de printer wordt ondersteund. Het tekstfilter wordt aangeroepen met de volgende lijst argumenten:

`filter-name` [-c] -w *width* -l *length* -i *indent* -n *login* -h *host acct-file*

met

`-c`
: wordt gebruikt als de afdrukopdracht is gegeven met `lpr -l`

width
: is de waarde van de optie `pw` (*page width*: paginabreedte), zoals opgegeven in /etc/printcap, standaard 132

length
: is de waarde van de optie `pl` (*page length*: paginalengte), standaard 66

indent
: geeft aan hoeveel wordt ingesprongen door `lpr -i`, standaard 0

login
: de gebruikersnaam van de gebruiker die de afdrukopdracht gaf

host
: de hostnaam waar de afdrukopdracht gegeven is

acct-file
: de naam van het administratiebestand zoals opgegeven via de optie `af`.

- Een *conversiefilter* converteert een specifiek bestandsformaat naar een formaat dat een printer begrijpt. Bijvoorbeeld: ditroff typesettinggegevens kunnen niet direct worden afgedrukt, maar er bestaat wel een conversiefilter om ditroff-gegevens te converteren naar een formaat dat een printer kan verteren en afdrukken. Dit wordt in Conversiefilters beschreven. Conversiefilters zijn ook nodig om printergebruik te administreren, mocht dat nodig zijn. Conversiefilters worden met de volgende argumenten aangeroepen:

  `filter-name` -x *pixel-width* -y *pixel-height* -n *login* -h *host acct-file*

  Hier is *pixel-width* de waarde van de optie `px` (standaard 0) en *pixel-height* is de waarde van de optie `py` (standaard 0).

- Het *uitvoerfilter* wordt alleen gebruikt als er geen tekstfilter is of als er voorbladen worden afgedrukt. De ervaring leert dat uitvoerfilters zelden worden gebruikt. In sectie Uitvoerfilters worden ze beschreven. Er zijn slechts twee argumenten die aan een uitvoerfilter worden

meegegeven:

`filter-name` -w *width* -l *length*

Deze zijn identiek aan de argumenten `-w` en `-l` van het tekstfilter.

Filters moeten *afsluiten* met de volgende waarde:

exit 0

Als het filter een bestand succesvol heeft afgedrukt.

exit 1

Als het filter niet geslaagd is om een bestand af te drukken, maar wil dat LPD het nogmaals probeert. LPD herstart het filter als die afsluit met deze status.

exit 2

Als het filter niet geslaagd is om een bestand af te drukken, maar niet wil dat LPD het nogmaals probeert. LPD verwijdert het bestand uit de wachtrij.

Het tekstfilter dat bij FreeBSD wordt geleverd, /usr/libexec/lpr/lpf, benut de argumenten voor paginabreedte en -lengte om te bepalen wanneer een nieuwe pagina moet worden begonnen en om het printergebruik bij te houden. Het gebruikt de argumenten voor login, host en administratiebestand om accountingregels aan te maken.

Controleer bij het zoeken naar filters of ze LPD-compatibel zijn. Zo ja, dan ondersteunen ze de argumenten zoals hierboven beschreven. Zorg bij het zelf schrijven van filters voor algemeen gebruik dat ze dezelfde argumenten en exitcodes ondersteunen.

Platte tekst op PostScript®-printers afdrukken

Als een computer en PostScript® (of andere op taal gebaseerde) printer maar één gebruiker hebben die belooft nooit platte tekst naar de printer te sturen of programma's te gebruiken die dat doen, dan is dit onderdeel overbodig.

Als gebruikers zowel PostScript® als platte tekst naar een printer willen sturen, dan is het aan te raden de printerinstellingen hierop aan te passen. Hiervoor moet het tekstfilter bij elke nieuwe opdracht bepalen of het om platte tekst of PostScript® gaat. Alle PostScript®-opdrachten beginnen met `%!` (raadpleeg de printerhandleiding voor andere printertalen). Als dit de eerste twee karakters zijn van een opdracht is het PostScript® en kan de rest van een opdracht direct doorgestuurd worden. Is dit niet het geval, dan moet de filter de tekst omzetten in PostScript® en het resultaat afdrukken.

Hoe gaat dat werken?

Voor seriële printers kan het meest eenvoudig `lprps` geïnstalleerd worden. `lprps` is een PostScript®-afdrukfilter die tweewegcommunicatie met een printer heeft. Het werkt het statusbestand van een printer bij met uitgebreide informatie afkomstig van een printer, zodat gebruikers en beheerders precies kunnen zien wat de status van een printer is (zoals: `toner bijna op` of `papier vastgelopen`). Maar belangrijker, het omvat het programma `psif` dat bepaalt of een binnenkomende opdracht platte tekst is en `textps` (dat ook geleverd wordt met `lprps`) om opdrachten om te zetten naar

PostScript®. Vervolgens wordt een opdracht met `lprps` naar een printer gestuurd.

`lprps` is onderdeel van de FreeBSD Portscollectie (zie De Portscollectie). U kunt één van de ports print/lprps-a4 of print-lprps-letter installeren afhankelijk van de gebruikte papiermaat. Nadat `lprps` is geïnstalleerd moet de installatielocatie ervan aan `psif` worden doorgeven dat onderdeel is van `lprps`. Als `lprps` is geïnstalleerd via de Portscollectie, gebruik dan het volgende voor de seriële PostScript®-printer in /etc/printcap:

```
:if=/usr/local/libexec/psif:
```

Ook moet de optie `rw` worden opgeven, die LPD vertelt om een printer in lezen/schrijvenmodus te openen.

Als een parallelle PostScript®-printer wordt ingesteld (en dus geen tweewegcommunicatie toegepast kan worden met de printer, zoals vereist door `lprps`), dan kan het volgende shellscript gebruikt worden als tekstfilter:

```
#!/bin/sh
#
#  psif - Druk PostScript(R) of platte tekst af op een PostScript(R)
#  printer. Script versie; NIET de versie die wordt geleverd bij lprps
#  Geïnstalleerd in /usr/local/libexec/psif
#

IFS="" read -r first_line
first_two_chars=`expr "$first_line" : '\(..\)'`

if [ "$first_two_chars" = "%!" ]; then
    #
    #  PostScript(R) opdracht, afdrukken.
    #
    echo "$first_line" && cat && printf "\004" && exit 0
    exit 2
else
    #
    #  Platte tekst, converteren en dan afdrukken.
    #
    ( echo "$first_line"; cat ) | /usr/local/bin/textps && printf "\004" && exit 0
    exit 2
fi
```

In bovenstaand script is `textps` een programma dat geïnstalleerd is om platte tekst om te zetten naar PostScript®. Elk tekst-naar-PostScript® programma volstaat. De FreeBSD Portscollectie (zie De Portscollectie) bevat een uitgebreid tekst-naar-PostScript®-programma, `a2ps`, dat wellicht handig is om te gebruiken.

PostScript® simuleren op niet-PostScript®-printers

PostScript® is *de facto* de standaard voor op hoge kwaliteit typesetten en afdrukken. PostScript® is echter een *dure* standaard. Gelukkig heeft Aladdin Enterprises een gratis PostScript®-kloon, Ghostscript, die werkt onder FreeBSD. Ghostscript kan de meeste PostScript®-bestanden lezen en de pagina's op verschillende soorten apparaten weergeven, waaronder veel niet-PostScript®-printers. Door Ghostscript te installeren en een printer gebruik te laten maken van een speciaal tekstfilter voor uw printer, kan uw niet-PostScript®-printer zich gedragen als een echte PostScript®-printer.

Ghostscript is beschikbaar via de FreeBSD Portscollectie, vele versies zijn beschikbaar, de meest gebruikte versie is print/ghostscript-gpl.

Om PostScript® te simuleren moet een tekstfilter detecteren of het een PostScript®-bestand aan het afdrukken is. Zo niet, dan stuurt het filter het bestand direct naar een printer, anders gebruikt het filter Ghostscript om het bestand om te zetten naar een formaat dat door een printer wordt begrepen.

Een voorbeeld: het volgende script is een tekstfilter voor Hewlett Packard DeskJet 500 printers. Voor andere printers moet het argument `-sDEVICE` voor het commando `gs` (Ghostscript) vervangen worden. (Met `gs -h` wordt een lijst met apparaten getoond worden die de huidige installatie van Ghostscript ondersteunt.)

```
#!/bin/sh
#
#  ifhp - Druk Ghostscript-gesimuleerd PostScript(R) af op een DeskJet
#  500.  Geïnstalleerd in /usr/local/libexec/ifhp

#
#  Behandel LF als CR+LF (om een "trapeffect" op HP/PCL
#  printer te voorkomen):
#
printf "\033&k2G" || exit 2

#
#  Lees de eerste twee karakters van het bestand
#
IFS="" read -r first_line
first_two_chars=`expr "$first_line" : '\(..\)'`

if [ "$first_two_chars" = "%!" ]; then
    #
    #  Het is PostScript(R). Gebruik Ghostscript om te converteren
    #  en druk het af.
    #
    /usr/local/bin/gs -dSAFER -dNOPAUSE -q -sDEVICE=djet500 \
        -sOutputFile=- - && exit 0
else
    #
    #  Platte tekst of HP/PCL, dus direct afdrukken;  druk een
```

```
    #  pagina-einde af om de laatste pagina te ejecteren.
    #
    echo "$first_line" && cat && printf "\033&l0H" &&
exit 0
fi

exit 2
```

Tot slot moet LPD op de hoogte gebracht worden van het filter via de optie `if`:

```
:if=/usr/local/libexec/ifhp:
```

Dat is alles. Nu kan `lpr platte.tekst` en `lpr watdanook.ps` ingevoerd worden en beiden worden juist afgedrukt.

Conversiefilters

Na de eenvoudige installatie, zoals beschreven in Eenvoudige printerinstallatie, te hebben voltooid, is het waarschijnlijk wenselijk om conversiefilters te installeren voor favoriete bestandsformaten (naast platte ASCII-tekst).

Waarom conversiefilters installeren?

Conversiefilters maken het afdrukken van verschillende bestanden eenvoudig. Stel dat veel gebruik gemaakt wordt van het tekstverwerkingsprogramma TeX en een PostScript® printer. Elke keer als door TeX een DVI-bestand wordt gegenereerd, kan dat niet direct afgedrukt worden. Het DVI-bestand moet omgezet worden naar PostScript®. De te geven opdrachten zijn de volgende:

```
% dvips zeewieranalyse.dvi
% lpr zeewieranalyse.ps
```

Na installatie van een conversiefilter voor DVI-bestanden kan deze handmatige conversie overgeslagen worden door LPD de conversie te laten uitvoeren. Elke keer als een DVI-bestand wordt afgedrukt, hoeft alleen de volgende opdracht gegeven te worden:

```
% lpr -d zeewieranalyse.dvi
```

LPD voert de DVI-bestandsconversie uit door `-d` te geven. In Opties voor opmaak en conversie staat een lijst van conversie-opties.

Voor elke conversie-optie moet een *conversiefilter* geïnstalleerd worden en moet in /etc/printcap de locatie worden opgegeven. Een conversiefilter is als het tekstfilter voor de eenvoudige printerinstallatie (Tekstfilter installeren), behalve dat in plaats van platte tekst af te drukken, het conversiefilter het bestand converteert naar een formaat dat een printer begrijpt.

Welke conversiefilters installeren?

Installeer de conversiefilters die nodig zijn. Als veel DVI-bestanden worden afgedrukt, dan is het handig een DVI-filter te installeren. Als veel troff wordt afgedrukt, dan is het waarschijnlijk handig een troff-filter te installeren.

De volgende tabel geeft een samenvatting van filters waarmee LPD kan werken, hoe ze in /etc/printcap kunnen worden aangeroepen en hoe ze met `lpr` kunnen worden aangeroepen:

Bestandsformaat	/etc/printcap optie	lpr optie
cifplot	`cf`	`-c`
DVI	`df`	`-d`
plot	`gf`	`-g`
ditroff	`nf`	`-n`
FORTRAN-tekst	`rf`	`-f`
troff	`tf`	`-f`
raster	`vf`	`-v`
platte tekst	`if`	geen, `-p`, of `-l`

In het voorbeeld waarbij `lpr -d` wordt gebruikt, moet voor de printer een optie `df` gedefinieerd staan in /etc/printcap.

Ondanks wat anderen mogelijk beweren, zijn formaten als FORTRAN-tekst en plot waarschijnlijk verouderd. Dit biedt de mogelijkheid een nieuwe betekenis te geven aan deze opties door zelf een filter te installeren. Stel dat direct Printerleaf-bestanden afgedrukt moeten worden (bestanden van het bureaubladpublicatieprogramma Interleaf), maar nooit plotbestanden worden afgedrukt. Dan kan een Printerleaf-conversiefilter geïnstalleerd worden onder de optie `gf` en gebruikers kunnen geïnstrueerd worden om `lpr -g` te gebruiken om Printerleaf-bestanden af te drukken.

Conversiefilters installeren

Aangezien conversiefilters programma's zijn die niet onder de FreeBSD-basisinstallatie vallen, kunnen ze het best onder /usr/local geplaatst worden. De map /usr/local/libexec is een veelgebruikte locatie, omdat hier programma's te vinden zijn die alleen door LPD gebruikt worden. Gewone gebruikers hoeven ze nooit te gebruiken.

Om een conversiefilter te activeren, moet de bestandslocatie onder de juiste optie voor de betreffende printer in /etc/printcap opgegeven worden.

In het onderstaande voorbeeld wordt het DVI-conversiefilter toegevoegd onder de sectie van de printer `bamboo`. Hieronder staat opnieuw het voorbeeldbestand /etc/printcap, nu met de nieuwe optie `df` voor de printer `bamboo`:

```
#
#  /etc/printcap voor host rose - df-filter voor bamboo toegevoegd
#
```

```
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:\
        :lp=/dev/lpt0:\
        :if=/usr/local/libexec/if-simple:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:\
        :lp=/dev/ttyu5:ms#-parenb cs8 clocal crtscts:rw:\
        :if=/usr/local/libexec/psif:\
        :df=/usr/local/libexec/psdf:
```

Het DVI-filter is een shellscript met de naam /usr/local/libexec/psdf. Het script ziet er als volgt uit:

```
#!/bin/sh
#
#  psdf - DVI naar PostScript afdrukfilter
#  Geïnstalleerd in /usr/local/libexec/psdf
#
# Aangeroepen door lpd wanneer een gebruiker lpr -d uitvoert
#
exec /usr/local/bin/dvips -f | /usr/local/libexec/lprps "$@"
```

Dit script roept `dvips` in filtermodus aan (het `-f` argument) op de standaard uitvoer, de af te drukken opdracht. Vervolgens start het PostScript® afdrukfilter `lprps` (zie Platte tekst op PostScript®-printers afdrukken) met de argumenten die LPD aan het script doorgeeft. `lprps` gebruikt deze argumenten om de afgedrukte pagina's te administreren.

Meer voorbeelden van conversiefilters

Er is geen vaste procedure om conversiefilters te installeren, er worden in deze sectie wat werkende voorbeelden gegeven. Gebruik deze als hulp bij het zelf maken van filters. Gebruik ze zonder aanpassingen indien mogelijk.

Dit voorbeeldscript is een raster (eigenlijk een GIF-bestand) conversiefilter voor een HP LaserJet III-Si printer:

```
#!/bin/sh
#
#  hpvf - Converteer GIF-bestanden naar HP/PCL, druk vervolgens af
#  Geïnstalleerd in /usr/local/libexec/hpvf

PATH=/usr/X11R6/bin:$PATH; export PATH
giftopnm | ppmtopgm | pgmtopbm | pbmtolj -resolution 300 \
    && exit 0 \
    || exit 2
```

Het script converteert achtereenvolgens het GIF-bestand naar een PNM-bestand (portable anymap), een PGM-bestand (portable graymap), een PBM-bestand (portable bitmap) en tenslotte naar

LaserJet/PCL formaat.

Een /etc/printcap bestand dat bovenstaand filter gebruikt ziet er als volgt uit:

```
#
#  /etc/printcap voor host orchid
#
teak|hp|laserjet|HP LaserJet 3Si:\
        :lp=/dev/lpt0:sh:sd=/var/spool/lpd/teak:mx#0:\
        :if=/usr/local/libexec/hpif:\
        :vf=/usr/local/libexec/hpvf:
```

Het volgende script is een conversiefilter voor troff-gegevens afkomstig van het groff-typesettingsysteem voor de PostScript®-printer `bamboo`:

```
#!/bin/sh
#
#  pstf - Converteert groff's troffgegevens naar PS, drukt vervolgens af.
#  Geïnstalleerd in /usr/local/libexec/pstf
#
exec grops | /usr/local/libexec/lprps "$@"
```

Bovenstaande script maakt eveneens gebruik van `lprps` om de communicatie met een printer af te handelen. Als een printer op een parallelle poort is aangesloten, ziet het er als volgt uit:

```
#!/bin/sh
#
#  pstf - Converteert groff's troff naar PS, drukt vervolgens af.
#  Geïnstalleerd in /usr/local/libexec/pstf
#
exec grops
```

Dat is alles. In /etc/printcap moet het volgende toegevoegd worden om het filter beschikbaar te maken:

```
:tf=/usr/local/libexec/pstf:
```

Hieronder een voorbeeld waarvan FORTRAN-programmeurs waarschijnlijk tranen in hun ogen krijgen: een FORTRAN-tekstfilter voor een willekeurige printer die in staat is platte tekst af te drukken. Het filter wordt actief gemaakt voor `teak`:

```
#!/bin/sh
#
# hprf - FORTRAN tekstfilter voor LaserJet 3si:
# Geïnstalleerd in /usr/local/libexec/hprf
```

```
#

printf "\033&k2G" && fpr && printf "\033&l0H" &&
 exit 0
exit 2
```

De onderstaande regel wordt toegevoegd aan /etc/printcap voor de printer `teak` om het filter beschikbaar te maken:

```
:rf=/usr/local/libexec/hprf:
```

Het laatste voorbeeld is wellicht complexer. Er wordt een DVI-filter toegevoegd voor de eerder genoemde LaserJet printer `teak`. Eerst het makkelijke gedeelte: in /etc/printcap wordt de locatie van het DVI-filter opgegeven:

```
:df=/usr/local/libexec/hpdf:
```

Nu het moeilijke gedeelte: het schrijven van het filter. Daarvoor is een DVI-naar-LaserJet/PCL conversieprogramma nodig. De FreeBSD Portscollectie (zie Portscollectie) heeft er een: print/dvi2xx. Door deze port te installeren komt het programma dat nodig is beschikbaar, `dvilj2p`, waarmee DVI geconverteerd kan worden naar LaserJet IIp-, LaserJet III- en LaserJet 2000-formaten.

Het hulpprogramma `dvilj2p` maakt het filter `hpdf` redelijk complex, omdat `dvilj2p` niet van de standaard invoer kan lezen. Het wil werken met een bestandsnaam. Nog lastiger is dat de bestandsnaam moet eindigen op .dvi, zodat moeilijk gebruik gemaakt kan worden van /dev/fd/0 als standaard. Dit probleem kan omzeild worden door een (symbolische) koppeling aan te maken van een tijdelijk bestand (eindigend op .dvi) naar /dev/fd/0. Hiermee wordt `dvilj2p` gedwongen van de standaard invoer te lezen.

De enige andere hobbel die genomen moet worden, is dat /tmp niet gebruikt kan worden als tijdelijke koppeling. Symbolische koppelingen zijn eigendom van de gebruiker en groep `bin`. Het filter wordt uitgevoerd door de gebruiker `daemon`. De map /tmp heeft het sticky-bit aan staan. Het filter kan de koppeling wel aanmaken, maar het is niet mogelijk de koppeling te verwijderen als de opdracht is uitgevoerd, omdat de koppeling eigendom is van een andere gebruiker.

In plaats hiervan maakt het filter een symbolische koppeling aan in de huidige werkmap, de wachtrijmap (zoals opgegeven in de optie `sd` in /etc/printcap). Dit is een perfecte plaats voor filters om hun werk te doen, zeker gezien er (soms) meer vrije schijfruimte is in de wachtrijmap dan onder /tmp.

Dit is het uiteindelijke filter:

```
#!/bin/sh
#
#  hpdf - Druk DVI-gegevens af op een HP/PCL printer
#  Geïnstalleerd in /usr/local/libexec/hpdf
```

```
PATH=/usr/local/bin:$PATH; export PATH

#
#  Definieer een functie om tijdelijke bestanden op te ruimen. Deze
#  staan in de huidige map; de wachtrijmap voor de printer.
#
cleanup() {
   rm -f hpdf$$.dvi
}

#
#  Definieer een functie om fatale fouten te verwerken: geef de
#  opgegeven boodschap weer en sluit af met 2. Afsluiten met 2 vertelt
#  LPD niet nog eens te proberen de afdrukopdracht af te drukken.
#
fatal() {
    echo "$@" 1>&2
    cleanup
    exit 2
}

#
#  Als de gebruiker de opdracht annuleert, stuurt LPD een SIGINT, dus
#  ondervang SIGINT (en enkele andere signalen) om onze rommel op te
#  ruimen.
#
trap cleanup 1 2 15

#
#  Voor de zekerheid bestaande tijdelijke bestanden opruimen
#
cleanup

#
#  Koppel het DVI-invoerbestand aan de standaard invoer (het af te
#  drukken bestand).
#
ln -s /dev/fd/0 hpdf$$.dvi || fatal "Cannot symlink /dev/fd/0"

#
#  Maak LF = CR+LF
#
printf "\033&k2G" || fatal "Cannot initialize printer"

#
#  Converteer en druk af.  De retourneerwaarde van dvilj2p lijkt niet
#  betrouwbaar, dus negeren we het.
#
dvilj2p -M1 -q -e- dfhp$$.dvi

#
```

```
#  Opruimen en afsluiten
#
cleanup
exit 0
```

Automatische conversie: een alternatief voor conversiefilters

Al deze conversiefilters bieden vele mogelijkheden voor afdrukomgevingen, maar dwingen de gebruiker aan te geven (op de lpr(1) commandoregel) welk filter gebruikt moet worden. Als gebruikers niet zo vaardig zijn in het gebruik van computers, wordt het al snel vervelend steeds aan te moeten geven welk filter gebruikt moet worden. Vervelender is echter wanneer een gebruiker een verkeerd filter gebruikt voor een bepaald bestandsformaat. Het resultaat kan zijn dat een printer honderden pagina's papier uitspuugt.

In plaats van het installeren van conversiefilters, is het te proberen om het (standaard) tekstfilter het bestandstype van het af te drukken bestand te laten detecteren en dan automatisch het juiste conversiefilter aan te laten roepen. Programma's als `file` kunnen hierbij handig zijn. Voor *sommige* bestandsformaten kan het moeilijk zijn de verschillen te ontdekken en voor deze bestanden kan alsnog een conversiefilter beschikbaar worden gesteld.

De FreeBSD Portscollectie heeft een tekstfilter dat automatisch converteert genaamd `apsfilter` (print/apsfilter). Het detecteert platte tekst, PostScript® en DVI-bestanden, voert de juiste conversie uit en druk de bestanden af.

Uitvoerfilters

Het wachtrijsysteem LPD ondersteunt een ander type filter waar nog geen aandacht aan is besteed: een uitvoerfilter. Een uitvoerfilter is bedoeld om alleen platte tekst af te drukken, net als een tekstfilter, maar met veel vereenvoudigingen. Wanneer een uitvoerfilter wordt gebruikt, maar geen tekstfilter, dan:

- start LPD een uitvoerfilter voor de gehele opdracht, in plaats van voor elk bestand in de opdracht;
- biedt LPD het uitvoerfilter niet de voorziening van het identificeren van het begin of eind van de bestanden in de afdrukopdracht;
- stuurt LPD de gebruikersnaam en de hostnaam niet door aan het filter. Het is dus niet bedoeld om een afdrukadministratie bij te houden. In feite zijn er maar twee argumenten:

  `filter-name` -w_width_ -l_length_

  Hierbij is *width* afkomstig van de optie `pw` en *length* afkomstig van de optie `pl` voor de betreffende printer.

De eenvoud van een uitvoerfilter is verleidelijk. Als elk bestand in een afdrukopdracht op een nieuwe pagina moet beginnen, is een uitvoerfilter *niet geschikt*. In dat geval dient een tekstfilter (ook wel invoerfilter) gebruikt te worden (zie Tekstfilter installeren. Verder is een uitvoerfilter eigenlijk *veel ingewikkelder*, omdat de te verwerken bytestroom gecontroleerd moet worden op speciale tekens en steeds signalen naar zichzelf moet sturen in opdracht van LPD.

Een uitvoerfilter is *noodzakelijk* als voorbladen gewenst zijn en het nodig is om escape-reeksen of andere initialisatietekens te sturen voor het afdrukken van het voorblad. Maar het is tevens *nutteloos* als het voorblad voor rekening van de afkomstige gebruiker moet komen, aangezien LPD geen gebruiker of hostinformatie naar het uitvoerfilter stuurt.

Op een enkele printer staat LPD het gebruik van zowel een uitvoerfilter als van een tekst of andere filter toe. In deze gevallen start LPD het uitvoerfilter alleen voor het afdrukken van het voorblad (zie Voorbladen). LPD verwacht vervolgens van het uitvoerfilter dat deze *zichzelf stopt* door twee bytes naar het filter te sturen: ASCII 031 gevolgd door ASCII 001. Als een uitvoerfilter deze twee bytes ziet (031, 001), moet die stoppen door een `SIGSTOP` naar zichzelf te sturen. Als LPD klaar is met het uitvoeren van alle andere filters, dan herstart deze het uitvoerfilter door er een `SIGCONT` naar toe te sturen.

Als er wel een uitvoerfilter, maar *geen* tekstfilter is en LPD is niet bezig met het verwerken van een opdracht met platte tekst, dan gebruikt LPD het uitvoerfilter voor het afdrukken van de opdracht. Zoals eerder vermeld, drukt het uitvoerfilter elk bestand van de opdracht achter elkaar af zonder pagina-einden of andere signalen voor paginavoortgang. Dit is waarschijnlijk *niet* gewenst. In bijna alle gevallen is een tekstfilter nodig.

Het programma `lpf`, dat eerder geïntoduceerd is als tekstfilter, kan ook worden uitgevoerd als uitvoerfilter. Als een ad-hoc uitvoerfilter nodig is, maar het schrijven van de bytedetectie en signaalverzending code niet wenselijk is, dan is `lpf` het proberen waard. `lpf` kan ook opgenomen worden in een shellscript om initialisatiecode af te handelen die eventueel nodig is voor een printer.

`lpf`: een tekstfilter

Het programma /usr/libexec/lpr/lpf uit de gecompileerde FreeBSD-distributie is een tekstfilter (invoerfilter) die uitvoer kan inspringen (een opdracht gegeven met `lpr -i`), karakters onveranderd kan doorlaten (een opdracht gegeven met `lpr -l`), de printpositie voor backspaces en tabs in de opdracht kan aanpassen en afgedrukte pagina's kan administreren. Het kan ook functioneren als uitvoerfilter.

Het filter `lpf` is geschikt voor vele afdrukomgevingen. Hoewel het zelf niet in staat is initialisatiesequenties naar een printer te sturen, is het vrij eenvoudig om een shellscript te schrijven dat de initialisatie doet en vervolgens `lpf` aanroept.

Als `lpf` afgedrukte pagina's moet administreren, is het nodig om de juiste waarden in te vullen voor de opties `pw` en `pl` in het bestand /etc/printcap. Deze waarden worden gebruikt om te bepalen hoeveel tekst er op een pagina past en hoeveel pagina's er in een afdrukopdracht zijn afgedrukt. Zie Printergebruik administreren voor meer informatie.

9.4.2. Voorbladen

Als er *veel* gebruikers zijn die allemaal verschillende printers gebruiken, dan is het te overwegen gebruik te maken van *voorbladen* als noodzakelijk kwaad.

Voorbladen, in het Engels ook wel bekend als *banner* of *burst* pagina's, identificeren wie een bepaalde opdracht heeft afgedrukt. Ze worden meestal bedrukt met grote, dikgedrukte letters,

eventueel met een decoratieve rand, zodat ze in een stapel afdrukken opvallen tussen de afgedrukte documenten. Ze maken het gebruikers mogelijk hun afdrukopdracht snel te vinden. Het nadeel van het gebruik van voorbladen is dat er een extra blad moet worden afgedrukt voor elke opdracht, waarmee hun nut niet langer duurt dan een paar minuten. Uiteindelijk belanden ze in een papierbak of afvalberg. Voorbladen gaan vooraf aan elke opdracht, niet aan elk bestand in een opdracht, waardoor de verspilling beperkt blijft.

Het LPD-systeem kan automatisch voorbladen afdrukken *als* een printer direct platte tekst kan afdrukken. In geval van een PostScript®-printer, is het nodig een extern programma aan te roepen om een voorblad te genereren (zie Voorbladen op PostScript®-printers.

Voorbladen afdrukken

In de sectie Eenvoudige printerinstallatie is het afdrukken van voorbladen uitgeschakeld door de optie `sh` ("suppress header") in het bestand /etc/printcap op te geven. Om wel voorbladen af te drukken, hoeft alleen de optie `sh` verwijderd te worden.

Dit klinkt wat al te makkelijk, of niet?

Dat klopt. Het *kan* nodig zijn een uitvoerfilter op te geven die initialisatiestrings naar een printer stuurt. Hier is een voorbeeld uitvoerfilter voor HP PCL-compatible printers:

```
#!/bin/sh
#
#  hpof - Uitvoerfilter voor HP PCL-compatible printers
#  Geïnstalleerd in /usr/local/libexec/hpof

printf "\033&k2G" || exit 2
exec /usr/libexec/lpr/lpf
```

Geef de locatie van het uitvoerfilter op met de optie `of`. Zie Uitvoerfilters voor meer informatie.

Hier is een voorbeeldbestand /etc/printcap voor de printer `teak` die eerder is geïntroduceerd;. Het afdrukken van voorbladen is geactiveerd en bovenstaande uitvoerfilter is toegevoegd:

```
#
#  /etc/printcap voor host orchid
#
teak|hp|laserjet|HP LaserJet 3Si:\
        :lp=/dev/lpt0:sd=/var/spool/lpd/teak:mx#0:\
        :if=/usr/local/libexec/hpif:\
        :vf=/usr/local/libexec/hpvf:\
        :of=/usr/local/libexec/hpof:
```

Als gebruikers nu een opdracht sturen naar `teak`, wordt er bij elke opdracht een voorblad afgedrukt. Als gebruikers liever willen zoeken naar hun afdrukken, dan kunnen ze de voorbladen onderdrukken door de opdracht te geven met het commando `lpr -h`. Zie Voorbladopties voor meer opties voor lpr(1).



LPD drukt een karakter voor pagina-einde af na elk voorblad. Als een printer een ander teken of sequentie gebruikt voor het beëindigen van een pagina, dan kan dit opgeven worden met de optie `ff` in /etc/printcap.

Voorbladen beheren

Door het afdrukken van voorbladen aan te zetten, produceert LPD een *lang voorblad* waarop in grote letters de gebruiker, host en opdracht te lezen zijn. Hier volgt een voorbeeld (`kelly` heeft de opdracht "outline" afgedrukt vanaf host `rose`):

```
k                 ll       ll
k                  l        l
k                  l        l
k   k     eeee     l        l     y    y
k  k     e    e    l        l     y    y
k k      eeeeee    l        l     y    y
kk k     e         l        l     y    y
k   k    e    e    l        l     y   yy
k    k    eeee    lll      lll     yyy y
                                       y
                                  y    y
                                   yyyy

                          ll
                  t        l        i
                  t        l
 oooo    u    u   ttttt    l       ii     n nnn     eeee
o    o   u    u     t      l        i     nn   n   e    e
o    o   u    u     t      l        i     n    n   eeeeee
o    o   u    u     t      l        i     n    n   e
o    o   u   uu     t  t    l        i     n    n   e    e
 oooo     uuu u      tt     lll     iii    n    n    eeee

r rrr     oooo     ssss     eeee
rr   r   o    o   s    s   e    e
r        o    o    ss      eeeeee
r        o    o      ss    e
r        o    o   s    s   e    e
r         oooo     ssss     eeee

                                    Job:  outline
                                    Date: Sun Sep 17 11:04:58 1995
```

LPD geeft een paginabegin na deze tekst, zodat de opdracht op een nieuwe pagina begint (tenzij de optie `sf` (*suppress form feeds*, "onderdruk paginabegin") is toegevoegd bij de desbetreffende printer in /etc/printcap).

Als dit wenselijk is, kan LPD ook een *korte tekst* op het voorblad afdrukken; geef hiervoor de optie `sb` (*short banner*, "kort voorblad") op in het bestand /etc/printcap. Het voorblad ziet er dan als volgt

uit:

```
rose:kelly  Job: outline  Date: Sun Sep 17 11:07:51 1995
```

Standaard drukt LPD het voorblad als eerste af en vervolgens de opdracht. Om dat om te keren, moet de optie `hl` (*header last*, "voorblad laatst") in /etc/printcap worden opgeven.

Voorbladen administreren

Het gebruik van LPD's ingebouwde voorbladen dwingt een bepaald paradigma af wat betreft het administreren van printergebruik: voorbladen moeten *gratis* zijn.

Waarom?

Omdat het uitvoerfilter het enige externe programma is dat controle heeft als het voorblad afgedrukt wordt dat het gebruik zou kunnen administreren. Het heeft echter geen beschikking over informatie over *gebruiker of host* of een administratiebestand. Het heeft dus geen idee wie voor het gebruik moet worden belast. Het volstaat ook niet om gewoon "het aantal pagina's met één op te hogen" door het tekstfilter of een van de conversiefilters (dat wel beschikt over gebruiker- en hostinformatie) te veranderen, omdat gebruikers het afdrukken van een voorblad kunnen onderdrukken met `lpr -h`. Ze worden dan aangeslagen voor voorbladen die niet zijn afgedrukt. Milieubewuste gebruikers gebruiken vast `lpr -h`, maar dit kan niet worden afgedwongen.

Het is *ook niet voldoende* om elk filter zijn eigen voorblad te laten genereren (om zo het gebruik te kunnen administreren). Als gebruikers het afdrukken van voorbladen willen onderdrukken met `lpr -h`, krijgen ze toch een voorblad en worden er ook voor belast, aangezien LPD geen kennis over de optie `-h` doorgeeft aan de filters.

Wat zijn dan de mogelijkheden?

- Accepteer het paradigma van LPD en maak voorbladen gratis;
- Installeer een alternatief voor LPD, zoals LPRng. In Alternatieven voor het standaard wachtrijsysteem staat meer over andere afdruksoftware die in plaats van LPD geïnstalleerd kan worden;
- Schrijf een *slim* uitvoerfilter. Gewoonlijk is een uitvoerfilter bedoeld om niet meer te doen dan het initialiseren van een printer of wat eenvoudige karakterconversie. Het is geschikt voor voorbladen en opdrachten met platte tekst (als er een tekstfilter is). Maar als er een tekstfilter is voor opdrachten met platte tekst, dan start LPD het uitvoerfilter alleen voor voorbladen. Het uitvoerfilter kan dan het voorblad dat LPD genereert analyseren om te bepalen welke gebruiker en host belast moeten worden voor het afdrukken van het voorblad. Het enige probleem is dat het uitvoerfilter nog steeds niet weet in welk bestand het gebruik moet worden bijgehouden (de naam van het bestand opgegeven in de `af` wordt niet meegegeven), maar als een bekend bestand gebruikt wordt, kan dit in het uitvoerfilter worden opgeven. Om het parsen af te handelen kan gebruik gemaakt worden van de optie `sh` (*short header*, "kort voorblad") in /etc/printcap. Dit kan echter wat omslachtig zijn en gebruikers waarderen zeker de meer gulle systeembeheerder die voorbladen gratis maakt.

Voorbladen op PostScript®-printers

Zoals hierboven beschreven, kan LPD een voorblad in platte tekst genereren, dat geschikt is voor de meeste printers. Natuurlijk kan PostScript® platte tekst niet direct afdrukken, zodat de voorbladfunctie van LPD nutteloos is.

Een voor de hand liggende manier om voorbladen te krijgen, is elk conversiefilter en tekstfilter zijn eigen voorblad te laten genereren. De filters moeten gebruik maken van de argumenten gebruiker en host om een geschikt voorblad te genereren. Het nadeel van deze methode is dat gebruikers altijd een voorblad krijgen, ook wanneer zij een opdracht geven met `lpr -h`.

Deze methode wordt nader beschreven. Het volgende script heeft drie argumenten (gebruikersnaam, hostnaam en de naam van de opdracht) en maakt een eenvoudig PostScript®-voorblad:

```
#!/bin/sh
#
#  make-ps-header - genereer een PostScript-voorblad op stdout
#  Geïnstalleerd in /usr/local/libexec/make-ps-header
#

#
#  Dit zijn PostScript-eenheden (72 in een inch). Pas dit aan voor A4
#  of het gebruikte formaat:
#
page_width=612
page_height=792
border=72

#
#  Controleer argumenten
#
if [ $# -ne 3 ]; then
    echo "Usage: `basename $0` <user> <host> <job>" 1>&2
    exit 1
fi

#
#  Bewaar deze, voornamelijk voor de leesbaarheid in de PostScript(R)-code.
#
user=$1
host=$2
job=$3
date=`date`

#
#  Stuur de PostScript(R)-code naar stdout.
#
exec cat <<EOF
%!PS
```

```
%
%  Vermijd conflicten met de opdracht van de gebruiker die volgt.
%
save

%
%  Maak een dikke, onaangename border in de marge van het papier.
%
$border $border moveto
$page_width $border 2 mul sub 0 rlineto
0 $page_height $border 2 mul sub rlineto
currentscreen 3 -1 roll pop 100 3 1 roll setscreen
$border 2 mul $page_width sub 0 rlineto closepath
0.8 setgray 10 setlinewidth stroke 0 setgray

%
%  Toon de gebruikersnaam duidelijk, groot en prominent
%
/Helvetica-Bold findfont 64 scalefont setfont
$page_width ($user) stringwidth pop sub 2 div $page_height 200 sub moveto
($user) show

%
%  Nu volgen de saaie bijzonderheden
%
/Helvetica findfont 14 scalefont setfont
/y 200 def
[ (Job:) (Host:) (Date:) ] {
200 y moveto show /y y 18 sub def }
forall

/Helvetica-Bold findfont 14 scalefont setfont
/y 200 def
[ ($job) ($host) ($date) ] {
        270 y moveto show /y y 18 sub def
} forall

%
% Dat is alles
%
restore
showpage
EOF
```

Nu kan zowel het conversiefilter als het tekstfilter dit script aanroepen om eerst een voorblad te genereren en vervolgens de opdracht van de gebruiker af te drukken. Hier volgt het eerder gebruikte DVI-conversieprogramma, aangepast om een voorblad te maken:

```
#!/bin/sh
```

```
#
#  psdf - DVI naar PostScript printfilter
#  Geïnstalleerd in /usr/local/libexec/psdf
#
#  Aangeroepen door lpd, wanneer de gebruiker lpr -d uitvoert
#

orig_args="$@"

fail() {
    echo "$@" 1>&2
    exit 2
}

while getopts "x:y:n:h:" option; do
    case $option in
        x|y)  ;; # Ignore
        n)    login=$OPTARG ;;
        h)    host=$OPTARG ;;
        *)    echo "LPD started `basename $0` wrong." 1>&2
              exit 2
              ;;
    esac
done

[ "$login" ] || fail "No login name"
[ "$host" ] || fail "No host name"

( /usr/local/libexec/make-ps-header $login $host "DVI File"
  /usr/local/bin/dvips -f ) | eval /usr/local/libexec/lprps $orig_args
```

Merk op hoe het filter eerst de argumentenlijst moet nagaan om te bepalen wat de gebruikers- en hostnaam zijn. Dit is gelijk voor de andere conversiefilters. Het tekstfilter heeft echter een andere verzameling argumenten (zie Hoe filters werken).

Zoals eerder is beschreven, is het in bovenstaande opzet, hoewel deze simpel is, niet mogelijk "voorbladen te onderdrukken" (de optie `-h` in `lpr`). Als gebruikers een boom willen sparen (of een paar centen bij betaalde voorbladen) dan is dit dus niet mogelijk, aangezien elk filter een voorblad afdrukt voor iedere opdracht.

Om gebruikers in staat te stellen per opdracht voorbladen te onderdrukken, moet gebruik gemaakt worden van de truc uit Voorbladen administreren: schrijf een uitvoerfilter dat het door LPD gegenereerde voorblad inleest en een PostScript®-versie genereert. Als de gebruiker de opdracht geeft met `lpr -h`, dan genereert LPD geen voorblad en het uitvoerfilter ook niet. Anders leest het uitvoerfilter de tekst van LPD in en stuurt een geschikt voorblad in PostScript® naar de printer.

Voor een PostScript®-printer op een seriële lijn kan gebruik gemaakt worden van `lprps`, dat met een uitvoerfilter wordt geleverd en het bovenstaande kan doen. Voorbladen worden door `psof` niet geteld.

9.4.3. Afdrukken via het netwerk

FreeBSD ondersteunt afdrukken via het netwerk: het sturen van opdrachten naar printers op afstand. Afdrukken via een netwerk betekent over het algemeen twee verschillende dingen:

- Het benaderen van een printer aangesloten op een andere computer. Een printer met een conventionele seriële of parallelle verbinding wordt op een bepaalde computer geïnstalleerd. Vervolgens wordt LPD zodanig ingesteld dat afdrukken vanaf andere computers in het netwerk mogelijk is. In Printers geïnstalleerd op andere hosts staat hoe dit te doen.
- Het benaderen van een printer die direct is aangesloten op een netwerk. Een printer heeft een netwerkinterface naast (of in plaats van) een gewone seriële of parallelle poort. Zo een printer kan als volgt werken:
  - Het begrijpt het LPD protocol en kan zelfs opdrachten van andere hosts in de wachtrij plaatsen. In dit geval werkt een printer als een gewone host die LPD heeft draaien. Volg de procedure in Printers geïnstalleerd op andere hosts om een dergelijke printer te installeren
  - Het kan zijn dat een printer een netwerkverbinding ondersteunt. In dit geval kan een printer worden "aangesloten" op een bepaalde host op het netwerk door deze host verantwoordelijk te maken voor het plaatsen van opdrachten in een wachtrij en het versturen van opdrachten naar de printer. In Printers met netwerkinterfaces staan enkele suggesties om zulke printers te installeren.

Printers geïnstalleerd op andere hosts

Het wachtrijsysteem LPD heeft een ingebouwde mogelijkheid om opdrachten naar andere hosts te sturen die ook LPD draaien (of een systeem dat compatibel is met LPD). Deze eigenschap maakt het mogelijk om een printer op een host te installeren en deze toegankelijk te maken voor andere hosts. Het werkt ook met printers die over een netwerkinterface beschikken en het LPD-protocol begrijpen.

Om dit soort afdrukken op afstand mogelijk te maken, moet een printer eerst op een host geïnstalleerd worden, de *printerhost*, door de printerinstallatie te volgen als beschreven in Eenvoudige printerinstallatie. Stel desgewenst de printer in voor geavanceerde taken volgens Geavanceerde printerinstallatie. Test de printer en controleer of deze werkt met eventueel speciaal ingestelde opties voor LPD. De *lokale host* moet geautoriseerd zijn om de LPD-dienst op de *verre host* te gebruiken (zie Opdrachten van hosts op afstand beperken).

Als een printer een netwerkinterface heeft die compatibel is met LPD, dan is de *printerhost* in onderstaande beschrijving de printer zelf en de *printernaam* is de naam die voor de printer is ingesteld. Meer informatie staat in de documentatie bij de printer en/of de printernetwerkinterface.



Bij een HP LaserJet voert de printernaam `text` automatisch de CRLF-conversie uit. Het is dan niet nodig het script hpif te gebruiken.

Op hosts die toegang moeten krijgen tot de printer, moet in /etc/printcap een regel worden toegevoegd met het volgende:

1. Geef de regel een willekeurige naam. Om het eenvoudig te houden kunnen wellicht het beste dezelfde namen en aliassen worden gebruikt als op de printerhost;

2. Laat de optie `lp` expliciet leeg (`:lp=:`);
3. Maak een wachtrijmap aan en geef de locatie op met de optie `sd`. LPD slaat hier afdrukopdrachten op alvorens ze naar de printerhost te sturen;
4. Geef de naam van de printerhost op met de optie `rm`;
5. Geef de naam van de printer op de *printerhost* op met de optie `rp`.

Dit is het. Conversiefilters, paginadimensies, enzovoort, hoeven niet in /etc/printcap opgegeven te worden.

Hier volgt een voorbeeld. De host `rose` heeft twee printers: `bamboo` en `rattan`. Gebruikers op de host `orchid` krijgen toegang tot deze printers. Hier volgt /etc/printcap voor `orchid` (uit Voorbladen afdrukken). Er stond in het bestand al een regel voor de printer `teak`. Voor de twee printers op de host `rose` zijn twee regels toegevoegd:

```
#
#  /etc/printcap voor host orchid - printers (op afstand) op rose toegevoegd
#

#
#  teak is lokaal; het is direct aangesloten op orchid:
#
teak|hp|laserjet|HP LaserJet 3Si:\
        :lp=/dev/lpt0:sd=/var/spool/lpd/teak:mx#0:\
        :if=/usr/local/libexec/ifhp:\
        :vf=/usr/local/libexec/vfhp:\
        :of=/usr/local/libexec/ofhp:

#
#  rattan is aangesloten op rose; stuur opdrachten voor rattan naar rose:
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :lp=:rm=rose:rp=rattan:sd=/var/spool/lpd/rattan:

#
#  bamboo is ook aangesloten op rose:
#
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :lp=:rm=rose:rp=bamboo:sd=/var/spool/lpd/bamboo:
```

Op `orchid` moeten wachtrijmappen worden aangemaakt:

```
# mkdir -p /var/spool/lpd/rattan /var/spool/lpd/bamboo
# chmod 770 /var/spool/lpd/rattan /var/spool/lpd/bamboo
# chown daemon:daemon /var/spool/lpd/rattan /var/spool/lpd/bamboo
```

Nu kunnen gebruikers op `orchid` afdrukken op `rattan` en `bamboo`. Een gebruiker op `orchid` geeft bijvoorbeeld de volgende invoer:

```
% lpr -P bamboo -d sushi-review.dvi
```

Dan kopieert LPD op `orchid` de opdracht naar de wachtrijmap /var/spool/lpd/bamboo en ziet dat het een DVI-opdracht is. Zodra de host `rose` ruimte heeft in zijn wachtrijmap `bamboo`, sturen de twee LPD's het bestand naar `rose`. Het bestand wacht in de wachtrij van `rose` totdat het succesvol is afgedrukt. Het wordt geconverteerd naar PostScript® (aangezien `bamboo` een PostScript®-printer is) op `rose`.

Printers met netwerkinterfaces

Netwerkkaarten voor printers zijn er in twee versies: een versie die een wachtrij nabootst (de duurdere versies), of versies die alleen de mogelijkheid geven om er informatie naar te sturen alsof het een seriële of parallelle poort is (de goedkopere versies). In Printers geïnstalleerd op andere hosts wordt het voor de duurdere beschreven.

Het formaat van /etc/printcap maakt het mogelijk om op te geven welke seriële, of parallelle poort gebruikt moet worden en (in geval van een seriële poort) de baud-snelheid, of er communicatie moet worden toegepast, vertragingen voor tabs, conversies voor nieuwe regelkarakters en meer. Er is geen mogelijkheid om een verbinding met een printer op te geven die op een TCP/IP of andere netwerkpoort luistert.

Om informatie naar een netwerkprinter te sturen, is het nodig een programma te ontwikkelen dat door tekst- en conversiefilters kan worden aangeroepen. Hier volgt een voorbeeld: het script `netprint` stuurt alle informatie van de standaard invoer naar een netwerkprinter. Als eerste argument wordt de hostnaam van de printer opgegeven en als tweede argument het poortnummer waarmee de verbinding moet worden opgezet. Er wordt alleen eenrichtingcommunicatie ondersteund (FreeBSD naar printer). Veel netwerkprinters ondersteunen tweewegcommunicatie. Het kan wenselijk zijn hiervan gebruik te maken (om printerstatus op te vragen, statistieken bij te houden, enzovoort).

```
#!/usr/bin/perl
#
#  netprint - Tekstfilter voor printer aangesloten op het netwerk
#  Geïnstalleerd in /usr/local/libexec/netprint
#
$#ARGV eq 1 || die "Usage: $0 <printer-hostname> <port-number>";

$printer_host = $ARGV[0];
$printer_port = $ARGV[1];

require 'sys/socket.ph';

($ignore, $ignore, $protocol) = getprotobyname('tcp');
($ignore, $ignore, $ignore, $ignore, $address)
    = gethostbyname($printer_host);

$sockaddr = pack('S n a4 x8', &AF_INET, $printer_port, $address);
```

```
socket(PRINTER, &PF_INET, &SOCK_STREAM, $protocol)
    || die "Can't create TCP/IP stream socket: $!";
connect(PRINTER, $sockaddr) || die "Can't contact $printer_host: $!";
while (<STDIN>) { print PRINTER; }
exit 0;
```

Dit script kan vervolgens in verschillende filters gebruikt worden. Stel dat een Diablo 750-N matrixprinter op het netwerk is aangesloten. Op poort 5100 accepteert de printer informatie om af te drukken. De hostnaam van de printer is `scrivener`. Hier volgt het tekstfilter voor de printer:

```
#!/bin/sh
#
#  diablo-if-net - Tekstfilter voor Diablo printer 'scrivener' luistert
#  op poort 5100. Geïnstalleerd in /usr/local/libexec/diablo-if-net
#
exec /usr/libexec/lpr/lpf "$@" | /usr/local/libexec/netprint scrivener 5100
```

9.4.4. Printergebruik beperken

Nu volgt informatie over het beperken van printergebruik. Het LPD-systeem maakt het mogelijk te bepalen wie er toegang heeft tot een printer, zowel lokaal als op afstand, of meerdere kopieën afgedrukt mogen worden, hoe lang opdrachten mogen zijn en hoe lang wachtrijen mogen worden.

Meerdere kopieën beperken

Het LPD systeem maakt het heel makkelijk voor gebruikers om meerdere afdrukken van een bestand te maken. Gebruikers kunnen opdrachten afdrukken met bijvoorbeeld `lpr -#5` en krijgen dan vijf kopieën van elk bestand in de opdracht. De systeembeheerder kan beslissen of dit wenselijk is.

Wanneer meerdere kopieën onwenselijk zijn, kan de optie `- van lpr(1) worden uitgeschakeld door de optie sc in /etc/printcap op te nemen. Als gebruikers opdrachten versturen met de optie -`, zien ze het volgende:

```
lpr: multiple copies are not allowed
```

Als het mogelijk is van andere hosts af te drukken (zie Printers geïnstalleerd op andere hosts), moet de optie `sc` ook in /etc/printcap van de andere hosts aanwezig zijn. Anders kunnen gebruikers nog steeds multi-kopie opdrachten van andere hosts sturen.

Hier volgt een voorbeeld. Hieronder staat /etc/printcap voor de host `rose`. De printer `rattan` is redelijk krachtig, dus meerdere kopieën zijn toegestaan. De laserprinter `bamboo` is wat gevoeliger, dus meerdere kopieë zijn uitgeschakeld door de optie `sc` toe te voegen:

```
#
#  /etc/printcap voor host rose - beperk meerdere kopieën op bamboo
#
```

```
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:\
        :lp=/dev/lpt0:\
        :if=/usr/local/libexec/if-simple:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:sc:\
        :lp=/dev/ttyu5:ms#-parenb cs8 clocal crtscts:rw:\
        :if=/usr/local/libexec/psif:\
        :df=/usr/local/libexec/psdf:
```

Nu moet ook de optie `sc` worden toegevoegd in /etc/printcap van host `orchid` (tegelijk worden meerdere kopieën voor de printer `teak` uitgeschakeld):

```
#
#  /etc/printcap voor host orchid - geen meerdere kopieën voor lokale
#  printer teak of printer op afstand bamboo
teak|hp|laserjet|HP LaserJet 3Si:\
        :lp=/dev/lpt0:sd=/var/spool/lpd/teak:mx#0:sc:\
        :if=/usr/local/libexec/ifhp:\
        :vf=/usr/local/libexec/vfhp:\
        :of=/usr/local/libexec/ofhp:

rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :lp=:rm=rose:rp=rattan:sd=/var/spool/lpd/rattan:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :lp=:rm=rose:rp=bamboo:sd=/var/spool/lpd/bamboo:sc:
```

Door de optie `sc` te gebruiken, wordt het gebruik van `lpr -#i` voorkomen, maar dat weerhoudt gebruikers er nog steeds niet van om lpr(1) meerdere keren te aanroepen of meerdere keren hetzelfde bestand te versturen in een opdracht:

```
% lpr voorverkoop.teken voorverkoop.teken voorverkoop.teken voorverkoop.teken
voorverkoop.teken
```

Er zijn vele manieren om dit misbruik te voorkomen (onder andere door het te negeren), welke vrij zijn om te verkennen.

Printertoegang beperken

Door gebruik te maken van het UNIX® groepmechanisme en de optie `rg` in /etc/printcap kan geregeld worden wie er op welke printer kan afdrukken. De gebruikers die toegang hebben tot een printer moeten in een groep worden geplaatst en deze groep moet in de optie `rg` worden genoemd.

Als gebruikers buiten de groep (inclusief `root`) naar de beheerde printer proberen te printen, worden ze begroet met het volgende bericht:

```
lpr: Not a member of the restricted group
```

Net als met de optie `sc` (*suppress multiple copies*: onderdruk meerdere kopieën) moet `rg`, indien wenselijk, ook op andere hosts worden opgegeven die ook toegang hebben tot printers (zie Printers geïnstalleerd op andere hosts).

In het volgende voorbeeld heeft iedereen toegang tot de printer `rattan`, maar alleen gebruikers in de groep `artists` kunnen gebruik maken van `bamboo`. Hier volgt het bekende /etc/printcap voor de host `rose`:

```
#
#  /etc/printcap voor host rose - beperkte toegang voor groep bamboo
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:\
        :lp=/dev/lpt0:\
        :if=/usr/local/libexec/if-simple:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:sc:rg=artists:\
        :lp=/dev/ttyu5:ms#-parenb cs8 clocal crtscts:rw:\
        :if=/usr/local/libexec/psif:\
        :df=/usr/local/libexec/psdf:
```

De andere voorbeeldbestanden /etc/printcap (voor de host `orchid`) worden niet aangepast. Natuurlijk kan iedereen op `orchid` afdrukken op `bamboo`. Het kan zijn dat er sowieso alleen bepaalde gebruikers op `orchid` zijn toegestaan en dat deze gebruikers toegang mogen hebben tot de printer. Of wellicht niet.



Er kan per printer slechts één groep worden opgegeven.

Grootte van afdrukopdrachten bepalen

Als veel gebruikers toegang hebben tot printers kan het nodig zijn een limiet op te geven voor de grootte van de bestanden die gebruikers naar een printer kunnen sturen. Er is immers slechts beperkte ruimte op het bestandssysteem en er moet ook voldoende ruimte zijn voor opdrachten van andere gebruikers.

LPD heeft de mogelijkheid om met de optie `mx` een limiet op te geven voor het maximum aantal bytes van een bestand in een afdrukopdracht. De eenheden worden opgegeven in `BUFSIZ` blokken, die 1024 bytes groot zijn. Een nul voor deze optie betekent geen limiet aan de bestandsgrootte. Als de optie wordt weggelaten, wordt een standaardlimiet van 1000 blokken gebruikt.



De limiet heeft betrekking op de *bestanden* in een opdracht, *niet* op de totale grootte van een opdracht.

LPD weigert een bestand dat groter is dan de opgegeven limiet niet. In plaats daarvan plaatst het zo

veel mogelijk van het bestand op de wachtrij, om dit vervolgens af te drukken. De rest wordt genegeerd. Of dit gedrag wenselijk is, is onderwerp van debat.

Nu worden limieten voor de voorbeeldprinters `rattan` en `bamboo` opgegeven. Aangezien de PostScript®-bestanden van die `artists` nogal groot kunnen worden, krijgen ze een limiet van vijf megabyte opgelegd. Er wordt geen limiet opgelegd voor de platte tekst printer:

```
#
#  /etc/printcap voor host rose
#

#
#  Geen limiet op opdrachtgrootte:
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:mx#0:sd=/var/spool/lpd/rattan:\
        :lp=/dev/lpt0:\
        :if=/usr/local/libexec/if-simple:

#
#  Limiet van vijf megabyte:
#
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:sc:rg=artists:mx#5000:\
        :lp=/dev/ttyu5:ms#-parenb cs8 clocal crtscts:rw:\
        :if=/usr/local/libexec/psif:\
        :df=/usr/local/libexec/psdf:
```

Ook hier zijn de limieten alleen van toepassing op lokale gebruikers. Als toegang tot deze printers van andere hosts mogelijk is, worden deze gebruikers niet beperkt. Het is daarom nodig de optie `mx` ook in de /etc/printcap van de betreffende hosts op te geven. In Printers geïnstalleerd op andere hosts staat meer informatie over afdrukken op andere hosts.

Er is een andere gespecialiseerde manier om opdrachtgrootte voor printers op afstand te beperken (zie Opdrachten van hosts op afstand beperken.

Opdrachten van hosts op afstand beperken

Het wachtrijsysteem LPD beschikt over verschillende methoden om afdrukopdrachten van hosts op afstand te beperken:

Hostbeperkingen

Met de bestanden /etc/hosts.equiv en /etc/hosts.lpd kan worden ingesteld van welke hosts op afstand een lokale LPD-opdracht wordt geaccepteerd. LPD controleert of een inkomend verzoek afkomstig is van een host die wordt genoemd in een van deze bestanden. Zo niet, dan weigert LPD het verzoek.

Het formaat van deze bestanden is eenvoudig: één host per regel. /etc/hosts.equiv wordt ook gebruikt door het protocol ruserok(3) en heeft invloed op programma's als rsh(1) en rcp(1).

Voorzichtigheid is dus geboden.

Als voorbeeld volgt hier /etc/hosts.lpd voor de host `rose`:

```
orchid
violet
madrigal.fishbaum.de
```

Dit betekent dat `rose` verzoeken accepteert van de hosts `orchid`, `violet` en `madrigal.fishbaum.de`. Voor iedere andere host die verbinding probeert te maken met LPD op `rose`, wordt de opdracht geweigerd.

Omvangbeperkingen

De hoeveelheid vrije ruimte die over moet blijven op een bestandssysteem waar een wachtrij zich bevindt kan ook worden ingesteld. Hiervoor moet een bestand met de naam minfree in de wachtrijmap worden aangemaakt. In dit bestand kan een getal worden gezet dat het aantal schijfblokken (512 bytes) aan vrije ruimte aangeeft dat beschikbaar moet blijven wil een opdracht worden geaccepteerd.

Hiermee kan worden gegarandeerd dat gebruikers op afstand een bestandssysteem niet vol kunnen schrijven. Ook kan hierdoor een soort voorrang worden gegeven aan lokale gebruikers: zij kunnen nog opdrachten plaatsen als de vrije schijfruimte al lang beneden de opgegeven limiet uit minfree is gekomen.

Als voorbeeld wordt een bestand minfree voor de printer `bamboo` toegevoegd. In /etc/printcap staat de juiste wachtrijmap:

```
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:sc:rg=artists:mx#5000:\
        :lp=/dev/ttyu5:ms#-parenb cs8 clocal crtscts:rw:mx#5000:\
        :if=/usr/local/libexec/psif:\
        :df=/usr/local/libexec/psdf:
```

De wachtrijmap wordt opgegeven met de optie `sd`. Er wordt een limiet van drie megabyte ingesteld (wat gelijk staat aan 6144 schijfblokken) voor de hoeveelheid vrije schijfruimte die op het bestandssysteem beschikbaar moet zijn voordat LPD een opdracht op afstand accepteert:

```
# echo 6144 > /var/spool/lpd/bamboo/minfree
```

Gebruikersbeperkingen

Met de optie `rs` in /etc/printcap kan worden geregeld welke gebruikers op afstand kunnen afdrukken op lokale printers. Als `rs` voorkomt voor een lokale printer accepteert LPD opdrachten van hosts op afstand *als* de gebruiker die de opdracht wil plaatsen ook een account heeft met dezelfde gebruikersnaam op de lokale host. Anders weigert LPD de opdracht.

Deze optie is met name nuttig in een omgeving waar (bijvoorbeeld) verschillende afdelingen een

netwerk delen en gebruikers de grenzen van de afdeling overschrijden. Door ze een account te geven op een systeem kunnen ze de aangesloten printers gebruiken vanaf het systeem van hun eigen afdeling. Wanneer ze *alleen* gebruik mogen maken van de printers en niet van overige diensten op de computer, kunnen "tokenaccounts" worden aangemaakt, zonder thuismap en met een nutteloze shell als /usr/bin/false.

9.4.5. Printergebruik administreren

Het kan nodig zijn om afdrukken te doorbelasten. Inkt en papier kosten geld en er zijn onderhoudskosten. Printers zitten vol met bewegende delen en hebben de neiging kapot te gaan. Nu is er gekeken naar de printers, het gebruikerspatroon en de onderhoudskosten en op basis hiervan is een prijs vastgesteld per pagina (of per centimeter, per meter, of per wat dan ook). Hoe wordt nu een administratie bijgehouden van gemaakte afdrukken?

Het slechte nieuws is dat het wachtrijsysteem LPD hierbij niet echt helpt. Het administreren van afdrukken is erg afhankelijk van het type printer, het afdrukformaat en de wensen die een systeembeheerder heeft ten aanzien van het doorbelasten van printergebruik.

Om het administreren te implementeren, is het nodig om aanpassingen te maken in de tekstfilter (om platte tekst opdrachten te belasten) en de conversiefilters (om opdrachten in andere bestandsformaten te belasten), om pagina's te tellen, of de printer te vragen hoeveel pagina's er zijn afgedrukt. Het volstaat niet om het eenvoudige uitvoerfilter te gebruiken, aangezien dit niet in staat is het gebruik te administreren. Zie Filters.

In het algemeen zijn er twee manieren om gebruik te administreren:

- *Periodiek administreren* is de meer gebruikelijke manier, omdat het waarschijnlijk makkelijker is. Als iemand een opdracht afdrukt, schrijft het filter de gebruiker, host en het aantal pagina's in een administratiebestand. Elke maand, semester, jaar, of een andere gewenste periode kunnen de administratiebestanden verzameld worden om het aantal afgedrukte pagina's op te tellen en het gebruik in rekening te brengen. De logboekbestanden kunnen vervolgens geschoond worden, zodat met een schone lei de volgende periode begonnen kan worden.
- *Directe administratie* is minder gebruikelijk, waarschijnlijk omdat het moeilijker is. Met deze methode zorgen de filters ervoor dat gebruikers voor hun printergebruik worden afgerekend op het moment dat ze er gebruik van maken. Net als schijfquota is de administratie onmiddellijk. Hiermee wordt voorkomen dat gebruikers kunnen afdrukken wanneer ze over hun limiet zijn gegaan. Ook biedt dit de mogelijkheid voor gebruikers om hun afdrukquotum te controleren, of aan te passen. Deze methode vereist databasecode om gebruikers en hun quota bij te houden.

Het wachtrijsysteem LPD ondersteunt beide methoden op eenvoudige wijze. Aangezien de filters (meestal) moeten worden aangeleverd, moet ook de code voor de administratie worden geleverd. Er is echter een voordeel: er is grote flexibiliteit in de administratiemethode. Zo kan bijvoorbeeld gekozen worden tussen periodieke of directe administratie. Er kan gekozen worden welke informatie opgeslagen wordt: gebruikersnamen, hostnamen, type opdracht, aantal afgedrukte pagina's, hoe lang het afdrukken duurde, enzovoort. Dit alles kan worden gedaan door de filters aan te passen.

Kort door de bocht printeradministratie

FreeBSD wordt met twee programma's geleverd waarmee periodieke administratie direct kan worden opgezet. Het zijn het tekstfilter `lpf`, beschreven in lpf: een tekstfilter en pac(8), een programma dat posten uit administratiebestanden verzamelt en optelt.

Zoals beschreven in de sectie over filters (Filters), roept LPD de tekst- en conversiefilters aan met de naam van het administratiebestand als argument. De filters kunnen dit argument gebruiken om te bepalen in welk bestand de gegevens voor de administratie moeten worden weggeschreven. De naam van dit bestand is afkomstig van de optie `af` uit /etc/printcap. Als er geen absoluut pad wordt opgegeven, dan is de locatie relatief aan de wachtrijmap.

LPD start `lpf` met paginabreedte en -lengte argumenten (afkomstig uit de opties `pw` en `pl`). Het filter `lpf` gebruikt deze argumenten om te bepalen hoeveel papier er gebruikt zal worden. Nadat het bestand naar de printer is gestuurd, schrijft het een post in het administratiebestand. De posten zien er als volgt uit:

```
2.00 rose:andy
3.00 rose:kelly
3.00 orchid:mary
5.00 orchid:mary
2.00 orchid:zhang
```

Aangezien `lpf` geen ingebouwde logica voor bestandslocking kent, moet voor elke printer een apart administratiebestand gebruikt worden. Twee `lpf`s kunnen elkaars posten corrumperen als ze tegelijk in hetzelfde bestand schrijven. De optie `af=acct` in /etc/printcap biedt een makkelijke manier om er zeker van te zijn dat aparte bestanden worden gebruikt. Dan bevindt elk administratiebestand zich in de wachtrijmap van de betreffende printer en krijgt de naam acct krijgen

Wanneer het tijd is om met gebruikers af te rekenen voor hun afdrukken, kan het programma pac(8) gedraaid worden. Ga naar de wachtrijmap van de printer waarvoor betaald moet worden en typ `pac`. Er verschijnt een dollar-centrische samenvatting zoals het volgende:

```
  Login               pages/feet   runs    price
orchid:kelly               5.00     1   $  0.10
orchid:mary               31.00     3   $  0.62
orchid:zhang               9.00     1   $  0.18
rose:andy                  2.00     1   $  0.04
rose:kelly               177.00   104   $  3.54
rose:mary                 87.00    32   $  1.74
rose:root                 26.00    12   $  0.52

total                    337.00   154   $  6.74
```

Dit zijn de argumenten die pac(8) verwacht:

-P printer

De *printer* waarvoor een samenvatting moet worden gegenereerd. Deze optie werkt alleen als er een absoluut pad is gegeven in de optie `af` in /etc/printcap.

-c

Sorteer de uitvoer op kosten, in plaats van alfabetisch op gebruikersnaam.

-m

Negeer de hostnamen in het administratiebestand. Met deze optie is de gebruiker `smith` op host `alpha` dezelfde gebruiker als `smith` op host `gamma`. Zonder deze optie zijn het verschillende gebruikers.

-p prijs

Bereken de prijs met *prijs* dollar per pagina of per voet, in plaats van de prijs uit de optie `pc` in /etc/printcap of twee cent (de standaard). De *prijs* kan worden opgegeven als een decimaal getal.

-r

Keer de sorteervolgorde om.

-s

Maak een bestand met een samenvatting van de administratie en leeg het administratiebestand.

namen...

Druk de administratiegegevens alleen af voor gebruikersnamen *namen*.

In de standaard samenvatting die pac(8) genereert, is het aantal pagina's te zien dat iedere gebruiker vanaf een bepaalde host heeft afgedrukt. Wanneer de hostnaam niet van belang is (bijvoorbeeld omdat gebruikers iedere host kunnen gebruiken), gebruik dan `pac -m` om de volgende samenvatting te genereren:

```
  Login               pages/feet   runs    price
andy                        2.00     1   $  0.04
kelly                     182.00   105   $  3.64
mary                      118.00    35   $  2.36
root                       26.00    12   $  0.52
zhang                       9.00     1   $  0.18

total                     337.00   154   $  6.74
```

Om het verschuldigde bedrag te berekenen gebruikt pac(8) de optie `pc` uit /etc/printcap (standaard aantal van 200 of 2 cent per pagina). Specificeer, in honderden centen, de prijs per pagina of per voet die berekent moet worden. Deze waarde kan worden aangepast door pac(8) aan te roepen met de optie `-p`. De eenheden van de optie `-p` zijn echter in dollars, niet in honderden centen. Bijvoorbeeld,

```
# pac -p1.50
```

zorgt ervoor dat elke pagina 1,50 dollar kost. U kunt echt grote winsten maken met deze optie.

Tenslotte kan met `pac -s` de samenvatting worden opgeslagen in een bestand dat dezelfde naam krijgt als het administratiebestand van de printer, maar dan met `_sum` toegevoegd aan de naam. Vervolgens wordt het administratiebestand geleegd. Als pac(8) opnieuw wordt aangeroepen, herleest pac(8) het samenvattingsbestand om de startwaarden te bepalen en telt daar de informatie bij op van het standaard administratiebestand.

Hoe kan het aantal afgedrukte pagina's worden geteld?

Om ook maar de minste nauwkeurigheid bij het administreren te verkrijgen, is het nodig te weten hoeveel papier een afdrukopdracht gebruikt. Dit is het centrale probleem van het bijhouden van printerstatistieken.

Voor opdrachten met platte tekst is het probleem niet zo moeilijk op te lossen: het aantal regels in een opdracht wordt geteld en vergeleken met het aantal regels per pagina dat door een printer wordt ondersteund. Hierbij moet niet worden vergeten dat backspaces in het bestand regels overschrijven en dat lange logische regels worden afgedrukt als meerdere fysieke regels.

Het tekstfilter `lpf` (geïntroduceerd in lpf: een tekstfilter) houdt met deze zaken rekening bij het administreren. Als het nodig is een tekstfilter te schrijven dat ook het printergebruik moet bijhouden, dan is het nuttig de broncode van `lpf` te bestuderen.

Hoe worden andere bestandsformaten dan verwerkt?

Voor een DVI-naar-LaserJet, of DVI-naar-PostScript® conversie kan het filter de diagnostische uitvoer van `dvilj` of `dvips` bekijken om te bepalen hoeveel pagina's er zijn geconverteerd. Voor andere formaten kan hetzelfde worden gedaan met behulp van de betreffende conversieprogramma's.

Deze methoden hebben echter als nadeel dat een printer eventueel niet alle pagina's ook daadwerkelijk afdrukt. Zo kan het papier vast komen te zitten, de toner opraken of de printer ontploffen, terwijl de gebruiker toch moet betalen.

Dus, wat kan hieraan worden gedaan?

Er is slechts één *betrouwbare* manier om *nauwkeurig* te administreren. Dat is met behulp van een printer die kan vertellen hoeveel papier er is gebruikt. Deze moet vervolgens worden aangesloten met een seriële lijn, of een netwerkverbinding. Bijna alle PostScript®-printers hebben deze mogelijkheid, andere modellen en merken mogelijk ook (bijvoorbeeld Imagen netwerklaserprinters). De filters dienen voor deze printers aangepast te worden om het papierverbruik na elke opdracht te achterhalen en de administratieve informatie *alleen* op deze waarde te baseren. Er is geen noodzaak om foutgevoelig regels te tellen of bestanden te analyseren.

Natuurlijk kan een beheerder ook vrijgevig zijn en alle afdrukken gratis maken.

9.5. Printers gebruiken

Hieronder wordt beschreven hoe printers die onder FreeBSD geïnstalleerd zijn gebruikt moeten worden. Nu volgt een overzicht van de commando's op gebruikersniveau:

lpr(1)
: Druk opdrachten af

lpq(1)
: Controleer printerwachtrijen

lprm(1)
: Verwijder opdrachten uit de wachtrij van een printer

Er is ook een administratief commando, lpc(8), beschreven in Printers beheren, dat gebruikt wordt om printers en hun wachtrijen in te stellen.

Alledrie de commando's lpr(1), lprm(1) en lpq(1) accepteren een optie `-Pprinternaam` om aan te geven op welke printer uit /etc/printcap een opdracht van toepassing is. Dit biedt de mogelijkheid opdrachten te versturen, verwijderen en controleren voor verschillende printers. Als `-P` niet wordt gebruikt, werken deze commando's op de printer gedefinieerd in de omgevingsvariabele `PRINTER`. Tot slot, wanneer de omgevingsvariabele `PRINTER` niet is gedefinieerd, wordt standaard verwezen naar de printer met de naam `lp`.

9.5.1. Opdrachten afdrukken

Om bestanden af te drukken:

```
% lpr bestandsnaam ...
```

Dit drukt elk van de opgegeven bestanden af op de standaard printer. Als geen bestanden worden opgegeven, drukt lpr(1) de standaard invoer af. De volgende opdracht drukt bijvoorbeeld een paar belangrijke systeembestanden af:

```
% lpr /etc/host.conf /etc/hosts.equiv
```

Om een specifieke printer te selecteren:

```
% lpr -P printernaam bestandsnaam ...
```

Dit voorbeeld drukt een lange opgave van de huidige map af op de printer `rattan`:

```
% ls -l | lpr -P rattan
```

Omdat er geen bestanden worden meegegeven aan het commando lpr(1), drukt `lpr` de gegevens af die het van de standaard invoer leest: de uitvoer van het commando `ls -l`.

lpr(1) accepteert ook een breed scala aan opties om de vorm aan te passen, bestandsconversies toe te passen, meerdere kopieën af te drukken, enzovoort. Meer informatie staat in Afdrukopties.

9.5.2. Opdrachten controleren

Als lpr(1) wordt gebruikt om af te drukken, dan worden de gegevens die afdrukt moet worden in een pakketje samengevoegd dat een "afdrukopdracht" wordt genoemd en naar het wachtrijsysteem LPD gestuurd. Elke printer heeft een wachtrij met opdrachten van alle gebruikers. Een printer drukt deze opdrachten op volgorde van binnenkomst af.

De wachtrij voor de standaardprinter kan worden weergegeven met lpq(1). Voor een specifieke printer moet de optie `-P` meegegeven worden. Het volgende commando toont de wachtrij van printer `bamboo`:

```
% lpq -P bamboo
```

Hieronder volgt een voorbeeld van de uitvoer van het commando `lpq`:

```
bamboo is ready and printing
Rank   Owner    Job  Files                              Total Size
active kelly    9    /etc/host.conf, /etc/hosts.equiv   88 bytes
2nd    kelly    10   (standard input)                   1635 bytes
3rd    mary     11   ...                                78519 bytes
```

Dit laat drie opdrachten zien in de wachtrij voor `bamboo`. De eerste opdracht, gegeven door gebruiker `kelly`, heeft opdrachtnummer 9 gekregen. Elke opdracht voor een printer krijgt een uniek opdrachtnummer. Dit nummer kan in de meeste gevallen genegeerd worden, maar is nodig om een opdracht te annuleren. In Opdrachten verwijderen staan meer details.

Opdrachtnummer negen bestaat uit twee bestanden; meerdere bestanden opgegeven naar lpr(1), worden als één enkele opdracht behandeld. Het is de actieve opdracht (`active` onder de kolom "Rank"), wat betekent dat de printer deze opdracht momenteel aan het afdrukken is. De tweede opdracht bestaat uit gegevens doorgegeven aan lpr(1) als standaard invoer. De derde opdracht is afkomstig van gebruiker `mary`. Het is een veel grotere opdracht. De bestandsnaam van het bestand dat ze probeert af te drukken is te lang voor het overzicht, daarom toont lpq(1) drie puntjes.

De allereerste regel uitvoer van lpq(1) is ook handig: die vertelt wat de printer momenteel aan het doen is; dat wil zeggen, wat LPD denkt dat de printer aan het doen is.

Het commando lpq(1) ondersteunt ook een optie `-l` om een gedetailleerd, lang overzicht te geven. Hieronder volgt voorbeelduitvoer van `lpq -l`:

```
waiting for bamboo to become ready (offline ?)
kelly: 1st              [job 009rose]
       /etc/host.conf                   73 bytes
       /etc/hosts.equiv                 15 bytes

kelly: 2nd              [job 010rose]
       (standard input)                 1635 bytes
```

```
mary: 3rd                                [job 011rose]
      /home/orchid/mary/research/venus/alpha-regio/mapping 78519 bytes
```

9.5.3. Opdrachten verwijderen

Een gebruiker die van gedachten verandert over een af te drukken opdracht, kan een opdracht uit een wachtrij halen met het commando lprm(1). Vaak kan met lprm(1) zelfs een actieve opdracht worden verwijderd, maar een deel of alles van de opdracht kan desondanks toch worden afgedrukt.

Om een opdracht van de standaardprinter te verwijderen dient eerst met lpq(1) het opdrachtnummer gevonden te worden. Typ vervolgens:

```
% lprm opdrachtnummer
```

Om een opdracht van een specifieke printer te verwijderen, moet de optie `-P` worden toegevoegd. Het volgende commando verwijdert opdrachtnummer 10 uit de wachtrij van printer `bamboo`:

```
% lprm -P bamboo 10
```

Het commando lprm(1) heeft een aantal snelkoppelingen:

lprm -
: Verwijder alle opdrachten (voor de standaardprinter) van de huidige gebruiker.

lprm *gebruiker*
: Verwijder alle opdrachten (voor de standaardprinter) die van *gebruiker* zijn. De supergebruiker kan opdrachten van andere gebruikers verwijderen. Andere gebruikers kunnen alleen hun eigen opdrachten verwijderen.

lprm
: Zonder een opdrachtnummer, gebruikersnaam of `-` op de opdrachtregel, verwijdert lprm(1) de huidige actieve opdracht van de huidige gebruiker op de standaard printer. Alleen de supergebruiker kan iedere actieve opdracht verwijderen.

Gebruik de optie `-P` met bovenstaande snelkoppelingen om een specifieke printer in plaats van de standaard printer te selecteren. Het volgende voorbeeld verwijdert alle opdrachten van de huidige gebruiker uit de wachtrij van printer `rattan`:

```
% lprm -P rattan -
```



Als in een netwerkomgeving wordt gewerkt, staat lprm(1) alleen toe opdrachten te verwijderen vanaf hosts waarvan de afdrukopdrachten zijn gegeven, ook als dezelfde printer vanaf andere hosts bereikbaar is. Het volgende voorbeeld demonstreert dit:

```
% lpr -P rattan mijnbestand
% rlogin orchid
% lpq -P rattan
Rank   Owner      Job  Files                         Total Size
active seeyan     12    ...                          49123 bytes
2nd    kelly      13   myfile                        12 bytes
% lprm -P rattan 13
rose: Permission denied
% logout
% lprm -P rattan 13
dfA013rose dequeued
cfA013rose dequeued
```

9.5.4. Meer dan platte tekst: afdrukopties

Het commando lpr(1) ondersteunt een aantal opties voor de opmaak van platte tekst, het converteren van grafische en andere bestandsformaten, het afdrukken van meerdere kopieën, afwikkeling van een opdracht en meer. In deze sectie worden die opties beschreven.

Opties voor opmaak en conversie

De volgende opties voor lpr(1) zorgen voor de opmaak van de bestanden in de opdracht. Gebruik deze opties als de opdracht geen platte tekst bevat of als platte tekst opgemaakt dient te worden met behulp van pr(1).

Het volgende commando drukt bijvoorbeeld een DVI-bestand af (van het TeX typesettingsysteem) met de naam visrapport.dvi op de printer `bamboo`:

```
% lpr -P bamboo -d visrapport.dvi
```

Deze opties zijn van toepassing op alle bestanden in de opdracht. Het is dus niet mogelijk om bijvoorbeeld DVI- en ditroff-bestanden in een opdracht samen te voegen. In plaats hiervan moeten deze bestanden als aparte opdrachten worden gegeven, elk met een andere conversie-optie.



Al deze opties, behalve `-p` en `-T`, vereisen dat er conversiefilters zijn geïnstalleerd voor een printer. De optie `-d` vereist bijvoorbeeld het DVI-conversiefilter. In Conversiefilters staan de details beschreven.

`-c`
: Afdrukken van cifplot-bestanden.

`-d`
: Afdrukken van DVI-bestanden.

`-f`
: Afdrukken van FORTRAN tekstbestanden.

`-g`

Afdrukken van plotgegevens.

`-i aantal`

De uitvoer wordt *aantal* kolommen ingesprongen. Als *nummer* wordt weggelaten, wordt acht kolommen ingesprongen. Deze optie werkt alleen met bepaalde conversiefilters.



Plaats geen spatie tussen de `-i` en het nummer.

`-l`

Drukt letterlijke tekstgegevens af, inclusief controlekarakters.

`-n`

Afdrukken van ditroff (apparaatonafhankelijke troff) gegevens.

-p

Opmaak van platte tekst met pr(1) alvorens af te drukken. Zie pr(1) voor meer informatie.

`-T titel`

Gebruik *titel* op de pr(1) koptekst in plaats van de bestandsnaam. Deze optie heeft alleen effect in combinatie met de optie `-p`.

`-t`

Afdrukken van troffgegevens.

`-v`

Afdrukken van rastergegevens.

In het volgende voorbeeld wordt een mooi opgemaakte versie van de handleiding ls(1) afgedrukt op de standaardprinter:

```
% zcat /usr/shared/man/man1/ls.1.gz | troff -t -man | lpr -t
```

Het commando zcat(1) pakt de broncode van de ls(1) handleiding uit en geeft het door aan het commando troff(1), dat de broncode opmaakt, er GNU troff van maakt en dit doorstuurt naar lpr(1), dat de opdracht naar de LPD wachtrij stuurt. Omdat de optie `-t` meegeven wordt aan lpr(1), converteert het wachtrijsysteem de GNU troff uitvoer naar een formaat dat de standaardprinter begrijpt als de opdracht wordt afgedrukt.

Opties voor opdrachtafhandeling

De volgende opties voor lpr(1) geven LPD aan de opdracht speciaal te behandelen:

-# kopieën

Produceer een aantal van *kopieën* kopieën van elk bestand in de opdracht, in plaats van één kopie. Een beheerder kan deze optie uitschakelen om slijtage van de printer te voorkomen en gebruik van een kopieerapparaat aan te moedigen. Zie Meerdere kopieën beperken.

Dit voorbeeld drukt drie kopieën af van parser.c gevolgd door drie kopieën van parser.h op de standaardprinter:

```
% lpr -#3 parser.c parser.h
```

-m

Stuur een email na voltooiing van de afdrukopdracht. Met deze optie stuurt het LPD-systeem een email als een opdracht is afgehandeld. In dit bericht vertelt het of de opdracht succesvol is uitgevoerd of dat er een fout was met (vaak) de aard van de fout.

-s

Kopieer de bestanden niet naar de wachtrijmap, maar maak in plaats hiervan een symbolische link.

Bij het afdrukken van een grote opdracht is het handig van deze optie gebruik te maken. Het spaart ruimte in de wachtrijmap (het kan zijn dat de opdracht de vrije ruimte verbruikt in het bestandssysteem waarin de wachtrijmap zich bevindt). Het bespaart ook tijd, omdat LPD niet elke byte van de opdracht naar de wachtrijmap hoeft te kopieëren.

Er is echter een nadeel: aangezien LPD het originele bestand nodig heeft, is het niet mogelijk dit te wijzigen, of te verwijderen totdat het is afgedrukt.



Bij het afdrukken op een printer in een netwerk, moet LPD een bestand uiteindelijk toch kopieëren van een lokale host naar een netwerkhost. De optie `-s` bespaart dus ruimte in een lokale wachtrijmap, niet in die van een host in een netwerk. Het blijft echter nuttig.

-r

Verwijder bestanden in een opdracht na ze naar een wachtrij gekopieerd te hebben of na ze te hebben afgedrukt als de optie `-s` is gebruikt. Wees voorzichtig met deze optie!

Voorbladopties

Deze opties voor lpr(1) passen de tekst aan die gewoonlijk op het voorblad van een opdracht verschijnt. Deze opties hebben geen effect als het afdrukken van voorbladen wordt onderdrukt op een gebruikte printer. Zie Voorbladen voor meer informatie over het opzetten van voorbladen.

-C *tekst*

Vervang de hostnaam op het voorblad door *tekst*. De hostnaam is gewoonlijk de naam van de host waarvan de opdracht is verstuurd.

-J *tekst*

Vervang de naam van de opdracht op het voorblad door *tekst*. De naam van de opdracht is standaard de naam van het eerste bestand in de opdracht of stdin als de standaard uitvoer wordt afgedrukt.

-h

Druk geen voorblad af.



Bij sommige installaties kan het zijn dat deze optie geen effect heeft door de manier waarop de voorbladen worden gegenereerd. Zie Voorbladen voor de details.

9.5.5. Printers beheren

De beheerder van de printers in een netwerk heeft deze moeten installeren, opzetten en testen. Met het commando lpc(8) kan een beheerder op nog meer manieren communiceren met printers. Met lpc(8) is het mogelijk om:

- Printers te starten en te stoppen;
- Wachtrijen aan en uit te zetten;
- De volgorde van opdrachten in elke wachtrij aan te passen.

Ten eerste een opmerking over terminologie: als een printer is *gestopt*, drukt die niets uit een wachtrij af. Gebruikers kunnen nog steeds opdrachten geven, maar opdrachten wachten in een wachtrij totdat de bijbehorende printer is *gestart* of als de wachtrij vrij is.

Als een wachtrij is *uitgeschakeld*, kan geen enkele gebruiker (behalve `root`) opdrachten naar een printer versturen. Een *ingeschakelde* wachtrij accepteert opdrachten. Een printer met een uitgeschakelde wachtrij kan worden *gestart* en drukt dan alle afdrukopdrachten in de wachtrij af tot deze leeg is.

In het algemeen is het nodig `root`-rechten te hebben om het commando lpc(8) te gebruiken. Gewone gebruikers kunnen het commando lpc(8) gebruiken om een printerstatus op te vragen en om een vastgelopen printer te herstarten.

Nu volgt een samenvatting van de lpc(8) commando's. De meeste commando's accepteren een argument *printernaam*, om aan te geven op welke printer te werken. Om op alle printers te werken die in /etc/printcap genoemd worden, kan `all` worden gebruikt als *printernaam*.

`abort printernaam`

Annuleer de huidige opdracht en stop de printer. Gebruikers kunnen nog steeds opdrachten versturen als de wachtrij is ingeschakeld.

`clean printernaam`

Verwijder oude bestanden uit de wachtrijmap van de betreffende printer. Het kan wel eens gebeuren dat de bestanden waaruit een opdracht bestaat niet juist worden verwijderd door LPD. Dit gebeurt bijvoorbeeld wanneer er fouten zijn opgetreden tijdens het afdrukken of tijdens grote administratieve activiteit. Dit commando vindt en verwijdert bestanden die niet in de wachtrijmap thuishoren.

`disable printernaam`

Nieuwe opdrachten kunnen niet meer in de wachtrij worden geplaatst. Als de printer nog draait, drukt die de opdrachten die zich nog in de wachtrij bevinden af. De supergebruiker (`root`) kan

altijd opdrachten versturen, ook naar een uitgeschakelde wachtrij.

Dit commando is handig bij het testen van een nieuwe printer of een filterinstallatie: schakel de wachtrij uit en verstuur als `root` opdrachten. Andere gebruikers kunnen geen opdrachten versturen totdat het testen is voltooid en de wachtrij weer is ingeschakeld met het commando `enable`.

`down printernaam boodschap`

Schakel een printer uit. Equivalent aan `disable` gevolgd door `stop`. De *boodschap* verschijnt als de status van de printer als een gebruiker de wachtrij van de printer controleert met lpq(1) of de status met `lpc status`.

`enable printernaam`

Schakel de wachtrij van een printer in. Gebruikers kunnen opdrachten versturen, maar de printer drukt ze pas af als deze is gestart.

`help commandonaam`

Geef hulp over het commando *commandonaam*. Zonder *commandonaam*, wordt een samenvatting van de beschikbare commando's getoond.

`restart printernaam`

Start de printer. Gewone gebruikers kunnen dit commando gebruiken als door een uitzonderlijke omstandigheid LPD hangt, maar ze kunnen een printer niet starten die gestopt is met een van de commando's `stop` of `down`. Het commando `restart` is equivalent aan `abort` gevolgd door `start`.

`start printernaam`

Start de printer. De printer drukt opdrachten in zijn wachtrij af.

`stop printernaam`

Stop de printer. De printer maakt de huidige opdracht af en drukt opdrachten in de wachtrij niet af. Gebruikers kunnen nog steeds opdrachten versturen naar een ingeschakelde wachtrij, ook al is de printer gestopt.

`topq printernaam opdracht-of-gebruikersnaam`

Herschik de wachtrij voor *printernaam* door de opdrachten met de opgegeven *opdracht* nummers of opdrachten van *gebruikersnaam* bovenaan de wachtrij te plaatsen. Voor dit commando is het niet mogelijk `all` te gebruiken als *printernaam*.

`up printernaam`

Schakel een printer in. Het omgekeerde van het commando `down`. Equivalent aan `start` gevolgd door `enable`.

lpc(8) accepteert bovenstaande commando's op de opdrachtregel. Als er geen commando's worden gegeven, schakelt lpc(8) over op een interactieve modus, waar opdrachten gegeven kunnen worden totdat het commando `exit`, `quit` of einde-van-bestand wordt gegeven.

9.6. Alternatieven voor het standaard wachtrijsysteem

Na het lezen van deze handleiding, heeft de lezer zo'n beetje alles gelezen wat er te leren valt over het wachtrijsysteem LPD zoals het te vinden is in FreeBSD. Er zijn veel tekortkomingen te onderkennen, wat vanzelf leidt tot de vraag: "Welke andere wachtrijsystemen zijn er beschikbaar (en werken onder FreeBSD)?"

LPRng

LPRng, dat "LPR: the Next Generation" betekent, is een compleet herschreven PLP. Patrick Powell en Justin Mason (de voornaamste beheerder van PLP) hebben samengewerkt om LPRng te maken. De thuispagina voor LPRng is http://www.lprng.org/.

CUPS

CUPS, het Common UNIX Printing System, voorziet in een overzetbare printlaag voor UNIX®-achtige besturingssystemen. Het is ontwikkeld door Easy Software Product, om een standaard afdrukoplossing voor alle UNIX®-producenten en gebruikers te promoten.

CUPS gebruikt het Internet Printing Protocol (IPP) als basis voor het beheren van afdrukopdrachten en wachtrijen. De protocollen Line Printer Daemon (LPD), Server Message Block (SMB) en AppSocket (ook bekend als JetDirect) worden ook ondersteund met minder functionaliteit. CUPS biedt bladeren naar netwerkprinters en PostScript® Printer Description (PPD) gebaseerde afdrukopties om echt printen onder UNIX® te ondersteunen.

De thuispagina voor CUPS is http://www.cups.org/.

HPLIP

HPLIP, het HP Linux® Imaging and Printing systeem, is een suite van programma's ontwikkeld door HP dat printen, scannen en faxen voor toepassingen van HP ondersteunt. Deze suite van programma's maakt gebruikt van het printsysteem CUPS als een backend voor sommige van de printmogelijkheden.

De thuispagina voor HPLIP is http://hplipopensource.com/hplip-web/index.html.

9.7. Problemen oplossen

Na het uitvoeren van een simpele test met lptest(1) is mogelijk een van onderstaande resultaten verkregen, in plaats van de juiste uitvoer:

Het werkte na enige tijd of er kwam geen volle pagina.

De printer drukte bovenstaande af, maar wachtte enige tijd zonder iets te doen. Het was zelfs nodig om een PRINT REMAINING, of FORM FEED-knop op te printer in te drukken om enig resultaat te krijgen.

Als dit het geval is, dan stond de printer waarschijnlijk te wachten of er nog meer gegevens van de opdracht zouden komen, alvorens iets af te drukken. Om dit probleem op te lossen, kan het tekstfilter worden aangepast zodat deze een FORM FEED-karakter (of wat er ook nodig is) naar de printer stuurt. Dit is meestal voldoende om een printer zover te krijgen om tekst af te drukken die zich nog in de interne buffer bevindt. Het is ook nuttig om er zeker van te zijn dat

elke afdrukopdracht eindigt op een hele pagina, zodat de volgende opdracht niet ergens midden op de laatste pagina van de vorige opdracht begint.

De volgende vervanging voor het shellscript /usr/local/libexec/if-simple drukt een form feed af nadat de opdracht naar een printer is gestuurd:

```
#!/bin/sh
#
# if-simple - Eenvoudige tekst invoerfilter voor lpd
# Geïnstalleerd in /usr/local/libexec/if-simple
#
# Kopieert eenvoudig stdin naar stdout.  Negeer alle filterargumenten.
# Schrijft een form feed karakter (\f) na het afdrukken van de opdracht.

/bin/cat && printf "\f" && exit 0
exit 2
```

De opdracht produceerde een getrapt effect.

Het resultaat ziet er als volgt uit:

```
!"#$%&'()*+,-./01234
               "#$%&'()*+,-./012345
                              #$%&'()*+,-./0123456
```

Dit krijgen slachtoffers van het *trapeffect* te zien. Het wordt veroorzaakt door conflicterende interpretaties van de karakters die een regeleinde aangeven. UNIX®-achtige besturingssystemen gebruiken een enkel karakter: ASCII-code 10, de line feed (LF). MS-DOS®, OS/2® en andere besturingssystemen gebruiken twee karakters: ASCII-code 10 *en* ASCII-code 13 (de carriage return, CR). Veel printers gebruiken de MS-DOS®-conventie voor het representeren van regeleinden.

Als onder FreeBSD wordt afgedrukt, bevat de tekst alleen het line feed-karakter. Na het zien van een line feed-karakter vervolgt de printer zijn werk op de volgende regel, maar behoudt dezelfde horizontale positie op de pagina voor het afdrukken van het volgende teken. Hier is de carriage return voor bedoeld: om het volgende karakter af te drukken aan de linkerkant van de pagina.

Dit is wat FreeBSD wil dat de printer doet:

Printer ontvangt CR	Printer drukt CR af
Printer ontvangt LF	Printer drukt CR + LF af

Hier volgen een aantal manieren om dit te bereiken:

- Gebruik de instellingentoetsen of het bedieningspaneel van de printer om de interpretatie van deze karakters aan te passen. Controleer de handleiding van de printer om uit te vinden hoe dit moet.



Als een systeem in een ander besturingssysteem dan FreeBSD wordt opgestart, kan het nodig zijn een printer *opnieuw* in te stellen, zodat die een interpretatie voor CR- en LF-karakters gebruikt die bij dat andere besturingssysteem horen. Het kan de voorkeur genieten een van onderstaande oplossingen te gebruiken.

- Zorg dat het seriële lijnstuurprogramma van FreeBSD automatisch LF naar CR+LF converteert. Dit werkt natuurlijk *alleen* voor printers op een seriële poort. Gebruik de optie `ms#` en zet de modus `onlcr` in het bestand /etc/printcap voor de printer om deze functionaliteit in te schakelen.
- Stuur een *escape-code* naar een printer om tijdelijk LF-karakters anders te behandelen. Raadpleeg hiervoor de handleiding van de printer om escape-codes te achterhalen die de printer ondersteunt. Als de juiste escape-code is gevonden, moet de tekstfilter worden aangepast zodat deze eerst de code stuurt en vervolgens de afdrukopdracht.

  Hier volgt een eenvoudig tekstfilter voor printers die HP PCL-escape-codes begrijpen. Dit filter zorgt dat een printer LF-karakters behandelt als LF en CR, vervolgens verstuurt het de opdracht en tot slot een form feed om de laatste pagina in de opdracht uit te voeren. Het zou met alle HP printers moeten werken.

  ```
  #!/bin/sh
  #
  # hpif - Eenvoudig tekst invoerfilter voor lpd voor HP PCL-printers
  # Geïnstalleerd in /usr/local/libexec/hpif
  #
  # Kopieert eenvoudig stdin naar stdout.  Negeert alle filterargumenten.
  # Vertelt de printer om LF te zien als CR+LF.
  # Werpt de pagina uit na voltooiing.

  printf "\033&k2G" && cat && printf "\033&l0H" && exit 0
  exit 2
  ```

  Nu volgt een voorbeeldbestand /etc/printcap voor host `orchid`. Er is een printer aangesloten op de eerste parallelle poort; een HP LaserJet 3Si, genaamd `teak`. Die gebruikt bovenstaand script als tekstfilter:

  ```
  #
  #  /etc/printcap voor host orchid
  #
  teak|hp|laserjet|HP LaserJet 3Si:\
          :lp=/dev/lpt0:sh:sd=/var/spool/lpd/teak:mx#0:\
          :if=/usr/local/libexec/hpif:
  ```

De regels zijn over elkaar afgedrukt.

De printer is nooit een regel opgeschoven. Alle regels tekst lopen over elkaar en zijn op dezelfde regel afgedrukt.

Dit probleem is het "omgekeerde" van het trapeffect, zoals boven beschreven, en is veel zeldzamer. Ergens worden de LF-karakters die FreeBSD gebruikt om een regel te eindigen gezien als CR-karakters om de afdruklocatie te verplaatsen naar de linkerkant van het papier, zonder óók een regel naar beneden te gaan.

Gebruik de instellingentoetsen, of het bedieningspaneel van de printer om de volgende interpretatie van LF en CR af te dwingen:

Printer ontvangt	Printer drukt af
CR	CR
LF	CR + LF

De printer is karakters kwijt.

Tijdens het afdrukken heeft de printer een paar karakters per regel niet afgedrukt. Het kan zijn dat het probleem erger werd naarmate de printer zijn werk deed, steeds meer karakters verliezend.

Het probleem is dat de printer de snelheid waarmee de computer gegevens over een seriële lijn stuurt niet bij kan houden (dit probleem zou zich niet voor moeten doen met printers op een parallelle poort). Er zijn twee manieren om dit probleem te verhelpen:

- Als de printer XON/XOFF flow-control ondersteunt, zorg dan dat FreeBSD dit gebruikt door de modus `ixon` in de optie `ms#` te specificeren.
- Als de printer de Request to Send / Clear to Send hardware-handshake ondersteunt, (ook bekend als `RTS/CTS`), specificeer dan de modus `crtscts` in de optie `ms#`. Zorg dat de bedrading van de kabel die printer met de computer verbindt juist is voor hardware flow-control.

Er werd onzin afgedrukt.

Het lijkt alsof de printer willekeurige onzin afdrukte en niet de gewenste tekst.

Dit is meestal een ander symptoom van verkeerde communicatieparameters voor een seriële printer. Controleer de bps-snelheid in de optie `br` en de instelling voor pariteit in de optie `ms#`. Wees er zeker van dat de printer dezelfde instellingen gebruikt als in het bestand /etc/printcap worden opgegeven.

Er gebeurde niets.

Als er niets gebeurde, ligt het probleem waarschijnlijk bij FreeBSD en niet bij de hardware. Voeg de optie logboekbestand (`lf`) toe in /etc/printcap voor de betreffende printer. Hier is bijvoorbeeld de definitie voor `rattan` met de optie `lf`:

```
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:\
        :lp=/dev/lpt0:\
        :if=/usr/local/libexec/if-simple:\
        :lf=/var/log/rattan.log
```

Probeer vervolgens nogmaals af te drukken. Controleer het logboekbestand (in dit voorbeeld

/var/log/rattan.log) op mogelijke foutmeldingen. Probeer op basis van deze melding het probleem te verhelpen.

Als er geen optie `lf` is opgegeven, gebruikt LPD /dev/console als standaard.

Hoofdstuk 10. Linux® binaire compatibiliteit

10.1. Overzicht

FreeBSD levert binaire compatibiliteit met verscheidene andere UNIX® achtige besturingssystemen, waaronder Linux®. Op dit moment kan de vraag gesteld worden waarom FreeBSD nu precies Linux®-binairen moet kunnen draaien. Het antwoord is dat veel bedrijven en ontwikkelaars alleen ontwikkelen voor Linux®, omdat dat het nieuwste "hebbeding" is in de wereld van computers. Dat laat FreeBSD gebruikers al zeurend achter bij diezelfde bedrijven en ontwikkelaars om originele FreeBSD versies van hun applicaties. Het probleem is dat veel van deze bedrijven zich niet goed realiseren hoeveel mensen hun product zouden gebruiken als er ook FreeBSD versies van waren en de meesten blijven alleen voor Linux® ontwikkelen. Dus wat moet een FreeBSD gebruiker doen? Hier komt de Linux® binaire compatibiliteit van FreeBSD om de hoek kijken.

In een notendop stelt de compatibiliteit FreeBSD in staat om rond de 90% van alle Linux® applicaties zonder wijzigingen te draaien. Dit omvat applicaties zoals StarOffice™, de Linux® versie van getenv(3), Adobe® Acrobat®, RealPlayer®, Oracle®, WordPerfect®, Doom, Quake en meer. Er wordt zelfs gemeld dat in sommige gevallen Linux®-binairen beter presteren op FreeBSD dan op Linux®.

Er zijn echter enkele Linux®-specifieke besturingssysteemeigenschappen die niet door FreeBSD ondersteund worden. Linux®-binairen werken niet op FreeBSD als ze overvloedig gebruik maken van i386™ specifieke aanroepen, zoals het aanzetten van de virtuele 8086 modus.

Na het lezen van dit hoofdstuk weet de lezer:

- Hoe Linux® binaire compatibiliteit op een systeem aan te zetten;
- Hoe aanvullende Linux® gedeelde bibliotheken te installeren;
- Hoe Linux® applicaties op een FreeBSD systeem te installeren;
- De implementatiedetails van Linux® compatibiliteit in FreeBSD.

Aangeraden voorkennis:

- Hoe extra software van derden te installeren (Applicaties installeren. pakketten en ports).

10.2. Installatie

Linux® binaire compatibiliteit staat standaard niet aan. De gemakkelijkste manier om deze functionaliteit aan te zetten is door het `linux` KLD object ("Kernel LoaDable object") te laden. Deze module kan geladen worden door het volgende commando als `root` uit te voeren:

```
# kldload linux
```

Als Linux® compatibiliteit altijd aan moet staan, dan moet de volgende regel aan /etc/rc.conf toegevoegd worden:

```
linux_enable="YES"
```

Met kldstat(8) kan gecontroleerd worden of de KLD geladen is:

```
% kldstat
Id Refs Address    Size     Name
 1    2 0xc0100000 16bd8    kernel
 7    1 0xc24db000 d000     linux.ko
```

Als het om enige reden ongewenst of onmogelijk is de KLD te laden, dan kan de Linux® binaire compatibiliteit statisch in de kernel gecompileerd worden door `options COMPAT_LINUX` aan het kernelinstellingenbestand toe te voegen. Daarna kan de nieuwe kernel zoals beschreven in De FreeBSD-kernel instellen geïnstalleerd worden.

10.2.1. Linux® runtime bibliotheken installeren

Dit kan op twee manieren gedaan worden: door de linux_base port te gebruiken of door ze handmatig te installeren.

Installeren uit de linux_base port

Dit is verreweg de gemakkelijkste weg om te bewandelen om de runtime bibliotheken te installeren. Het is net als het installeren van andere ports uit de Portscollectie. Dit kan met het volgende commando:

```
# cd /usr/ports/emulators/linux_base-f10
# make install distclean
```



Op FreeBSD-systemen vóór FreeBSD 8.0 dient u de port emulators/linux_base-fc4 in plaats van emulators/linux_base-f10 te gebruiken.

Nu is er werkende Linux® binaire compatibiliteit. Sommige programma's kunnen klagen over onjuiste kleine versies van de systeembibliotheken. Over het algemeen schijnt dit echter geen probleem te zijn.



Er kunnen verschillende versies van de emulators/linux_base port beschikbaar zijn, overeenkomend met verschillende versies van verscheidene Linux® distributies. Het is verstandig de port te installeren die het meest voldoet aan de eisen van de Linux® applicaties die geïnstalleerd gaan worden.

Bibliotheken handmatig installeren

Als de Portscollectie niet is geïnstalleerd, kunnen de bibliotheken met de hand geïnstalleerd worden. Om alles te laten werken moeten de Linux® gedeelde bibliotheken waarvan het programma afhankelijk is en de runtime linker geïnstalleerd worden. Ook moet een "shadow root" map aangemaakt worden, /compat/linux, voor Linux® bibliotheken op een FreeBSD systeem. Elke gedeelde bibliotheek die wordt geopend door Linux® programma's die op FreeBSD draaien, kijken eerst in deze boomstructuur. Dus als een Linux® programma bijvoorbeeld /lib/libc.so laadt, probeert FreeBSD eerst /compat/linux/lib/libc.so te openen, en als die niet bestaat, probeert het /lib/libc.so proberen. Gedeelde bibliotheken moeten in de schaduwmapstructuur geïnstalleerd worden in plaats van in de paden die het Linux® `ld.so` rapporteert.

In het algemeen geldt dat alleen de eerste paar keer dat een Linux® binary wordt geïnstalleerd op een FreeBSD systeem naar de gedeelde bibliotheken gezocht wordt waar Linux®-binairen van afhankelijk zijn. Na een tijd is de verzameling van Linux® gedeelde bibliotheken op een systeem voldoende groot om nieuw geïmporteerde Linux®-binairen te kunnen draaien zonder enig extra werk.

Extra gedeelde bibliotheken installeren

Wat als de linux_base port is geïnstalleerd en een applicatie nog steeds klaagt over ontbrekende gedeelde bibliotheken? Op zich zijn er twee mogelijkheden (voor het opvolgen van deze instructies zijn `root` rechten op een FreeBSD systeem vereist).

Als er toegang is tot een Linux® systeem kan gekeken worden welke gedeelde bibliotheken de applicatie nodig heeft en kunnen ze gekopieerd worden naar het FreeBSD systeem. Dit wordt toegelicht in het volgende voorbeeld:

Stel dat FTP gebruikt is om de Linux® binary van Doom op te halen en die op een Linux® systeem staat waar toegang tot is. Dan kan met `ldd linuxdoom` gecontroleerd worden welke gedeelde bibliotheken er nodig zijn:

```
% ldd linuxdoom
libXt.so.3 (DLL Jump 3.1) => /usr/X11/lib/libXt.so.3.1.0
libX11.so.3 (DLL Jump 3.1) => /usr/X11/lib/libX11.so.3.1.0
libc.so.4 (DLL Jump 4.5pl26) => /lib/libc.so.4.6.29
```

Alle bestanden uit de laatste kolom zijn nodig en moeten onder /compat/linux komen te staat en de namen uit de eerste kolom moeten er als symbolische links naar verwijzen. Dit betekent dat uiteindelijk deze bestanden op een FreeBSD systeem staan:

```
/compat/linux/usr/X11/lib/libXt.so.3.1.0
/compat/linux/usr/X11/lib/libXt.so.3 -> libXt.so.3.1.0
/compat/linux/usr/X11/lib/libX11.so.3.1.0
/compat/linux/usr/X11/lib/libX11.so.3 -> libX11.so.3.1.0
/compat/linux/lib/libc.so.4.6.29
/compat/linux/lib/libc.so.4 -> libc.so.4.6.29
```

Als er al een Linux® gedeelde bibliotheek met een groot revisienummer overeenstemmend met de eerste kolom van de `ldd` uitvoer is, dan hoeft het bestand uit de laatste kolom niet naar een systeem gekopieerd te worden. Het bestand dat er al staat moet werken. Het is aan te raden om de gedeelde bibliotheek sowieso te kopiëren als het een nieuwere versie is. De oude kan verwijderd worden, zolang de symbolische link maar naar de nieuwe wijst. Dus als deze bibliotheken op een systeem staan:

```
/compat/linux/lib/libc.so.4.6.27
/compat/linux/lib/libc.so.4 -> libc.so.4.6.27
```



en een nieuwe binary zegt een latere versie nodig te hebben volgens de uitvoer van `ldd`:

```
libc.so.4 (DLL Jump 4.5pl26) -> libc.so.4.6.29
```

Als slechts één of twee versies verouderd zijn in het laatste cijfer, dan hoeft /lib/libc.so.4.6.29 niet gekopieerd te worden, omdat het programma goed moet werken met de ietwat oudere versie. Als er echter behoefte aan is, kan besloten worden om libc.so sowieso te verplaatsen, en dat resulteert in:

```
/compat/linux/lib/libc.so.4.6.29
/compat/linux/libc.so.4 -> lbic.so.4.6.29
```



Het symbolische linkmechanisme is *alleen* nodig voor Linux®-binairen. De FreeBSD runtime linker zorgt zelf voor het kijken naar passende grote revisienummers en daar hoeft geen zorg over te bestaan.

10.2.2. Linux® ELF-binairen installeren

ELF-binairen hebben soms een extra stap van "branding" nodig. Als er ongemerkte ELF-binairen worden gedraaid, ontstaat er een foutmelding zoals de volgende:

```
% ./mijn-linux-elf-binary
ELF binary type not known
Abort
```

Om de FreeBSD kernel te helpen FreeBSD ELF-binairen en Linux® binairen uit elkaar te houden, kan brandelf(1) gebruikt worden.

```
% brandelf -t Linux mijn-linux-elf-binary
```

De GNU gereedschapskist plaatst nu automatisch de juiste merkinformatie in ELF-binairen , dus

deze stap zou steeds overbodiger moeten worden in de toekomst.

10.2.3. Een willekeurige toepassing gebaseerd op Linux RPM installeren

FreeBSD heeft zijn eigen pakketdatabase die wordt gebruikt om alle ports te volgen (ook Linux® ports). De Linux® RPM-database wordt dus niet gebruikt (noch ondersteund).

Als u echter een willekeurige toepassing die op Linux® RPM is gebaseerd moet installeren kan dit bereikt worden met:

```
# cd /compat/linux
# rpm2cpio -q < /pad/naar/linux.archief.rpm | cpio -id
```

Draai daarna brandelf op de geïnstalleerde ELF-binairen (niet de bibliotheken!). Een schone deïnstallatie is niet mogelijk, maar het kan helpen met testen.

10.2.4. De hostnaamresolver instellen

```
resolv+: "bind" is an invalid keyword resolv+:
"hosts" is an invalid keyword
```

Als DNS niet werkt of de bovenstaande melding ontstaat, dan moet /compat/linux/etc/host.conf ingesteld worden met daarin:

```
order hosts, bind
multi on
```

De volgorde geeft aan dat /etc/hosts als eerste doorzocht wordt en DNS als tweede. Als /compat/linux/etc/host.conf niet geïnstalleerd is, vinden Linux® applicaties /etc/host.conf van FreeBSD en klagen ze over de incompatibele FreeBSD syntaxis. `bind` moet verwijderd worden als er geen naamserver is ingesteld die gebruik maakt van /etc/resolv.conf.

10.3. Mathematica® installeren

Dit document beschrijft het installatieproces van de Linux® versie van Mathematica® 5.X op een FreeBSD systeem.

De Linux® versie van Mathematica® of Mathematica® for Students kan direct bij Wolfram besteld worden op http://www.wolfram.com/.

10.3.1. De Mathematica® Installer draaien

Ten eerste dient FreeBSD te weten dat de Linux-binairen van Mathematica® de Linux ABI gebruiken. De gemakkelijkste manier om dit te doen is om het standaard ELF-merk op Linux te zetten voor alle ongemerkte binairen met het commando:

```
# sysctl kern.fallback_elf_brand=3
```

Dit laat FreeBSD aannemen dat alle ongemerkte ELF-binairen de Linux ABI gebruiken en dus zou de installer rechtstreeks van de CD-ROM moeten kunnen draaien.

Kopieer nu het bestand MathInstaller naar de harde schijf:

```
# mount /cdrom
# cp /cdrom/Unix/Installers/Linux/MathInstaller /localdir/
```

Vervang binnen dit bestand `/bin/sh` op de eerste regel door `/compat/linux/bin/sh`. Dit zorgt ervoor dat de installer door de Linux-versie van sh(1) wordt uitgevoerd. Vervang vervolgens met een tekstverwerker of het onderstaande script in de volgende sectie alle voorkomens van `Linux)` door `FreeBSD)`. Dit zorgt ervoor dat de Mathematica® installer, dat `uname -s` gebruikt om het besturingssysteem te bepalen, om FreeBSD als een Linux-achtig besturingssysteem te behandelen. Het aanroepen van `MathInstaller` zal nu Mathematica® installeren.

10.3.2. De Mathematica®-executables wijzigen

De shellscripts die Mathematica® aanmaakte tijdens de installatie moeten gewijzigd worden voordat u ze kunt gebruiken. Als u /usr/local/bin kiest als de map om Mathematica®-executables in te plaatsen, zult u in deze map symbolische links naar bestanden genaamd math, mathematica, Mathematica, en MathKernel aantreffen. Vervang met een tekstverwerker of het volgende shellscript in elk van deze `Linux)` door `FreeBSD`:

```
#!/bin/sh
cd /usr/local/bin
for i in math mathematica Mathematica MathKernel
  do sed 's/Linux)/FreeBSD)/g' $i > $i.tmp
  sed 's/\/bin\/sh/\/compat\/linux\/bin\/sh/g' $i.tmp > $i
  rm $i.tmp
  chmod a+x $i
done
```

10.3.3. Mathematica® wachtwoord opvragen

Wanneer u Mathematica® voor de eerste keer start, zal u om een wachtwoord gevraagd worden. Als u nog geen wachtwoord van Wolfram heeft verkregen, draait u het programma `mathinfo` in de installatiemap om uw "machine-ID" te verkrijgen. Dit machine-ID is alleen op het MAC-adres van uw eerste Ethernetkaart gebaseerd, zodat u uw kopie van Mathematica® niet op andere machines kunt draaien.

Bij een registratie bij Wolfram, per email, telefoon of fax, wordt het "machine ID" opgegeven en zij reageren met een overeenkomstig wachtwoord dat uit groepen getallen bestaat.

10.3.4. Het Mathematica® frontend over een netwerk draaien

Mathematica® gebruikt enkele speciale lettertypen om tekens af te beelden die niet aanwezig zijn in een standaard lettertypeverzameling (integralen, sommen, Griekse letters, enzovoort). Het X-protocol vereist dat deze lettertypen *lokaal* worden geïnstalleerd. Dit betekent dat deze lettertypen gekopieerd moeten worden vanaf de CD-ROM of vanaf een host met Mathematica® erop naar de lokale machine. Deze lettertypen worden meestal opgeslagen in /cdrom/Unix/Files/SystemFiles/Fonts op de CD-ROM of in /usr/local/mathematica/SystemFiles/Fonts op de harde schijf. De eigenlijke lettertypen staan in de submap Type1 en X. Er zijn verschillende manieren om ze te installeren, zoals hieronder staat beschreven.

De eerste manier is om ze te kopiëren in één van de bestaande lettertypenmappen in /usr/X11R6/lib/X11/fonts. Hiertoe dient fonts.dir bewerkt te worden door de namen van de lettertypen eraan toe te voegen het aantal lettertypen op de eerste regel te veranderen. Als alternatief kan ook eenvoudig mkfontdir(1) in de map gedraaid worden waar de lettertypen heen zijn gekopieerd.

De tweede manier om dit te doen is door de mappen naar /usr/X11R6/lib/X11/fonts te kopiëren:

```
# cd /usr/X11R6/lib/X11/fonts
# mkdir X
# mkdir MathType1
# cd /cdrom/Unix/Files/SystemFiles/Fonts
# cp X/* /usr/X11R6/lib/X11/fonts/X
# cp Type1/* /usr/X11R6/lib/X11/fonts/MathType1
# cd /usr/X11R6/lib/X11/fonts/X
# mkfontdir
# cd ../MathType1
# mkfontdir
```

Voeg nu de nieuwe lettertypenmappen toe aan het lettertypenpad:

```
# xset fp+ /usr/X11R6/lib/X11/fonts/X
# xset fp+ /usr/X11R6/lib/X11/fonts/MathType1
# xset fp rehash
```

Als de Xorg server gebruikt wordt, kunnen deze lettertypenmappen automatisch geladen worden door ze aan xorg.conf toe te voegen.

Als er nog *geen* map /usr/X11R6/lib/X11/fonts/Type1 bestaat, kan de naam van de map MathType1 in het bovenstaande voorbeeld veranderd worden naar Type1.

10.4. Maple™ installeren

Maple™ is een commercieel wiskundeprogramma vergelijkbaar met Mathematica®. De software is te koop op http://www.maplesoft.com/ en kan daar ook geregistreerd worden voor een licentiebestand. Om deze software op FreeBSD te installeren kunnen de volgende eenvoudige

stappen gevolgd worden:

1. Voer het INSTALL> shellscript uit van de productdistributie. Kies de "RedHat" optie als daarom wordt gevraagd door het installatieprogramma. Een typische installatiemap zou /usr/local/maple zijn.
2. Bestel, als dat nog niet gedaan is, een licentie voor Maple™ van Maple Waterloo Software (http://register.maplesoft.com/) en kopieer deze naar /usr/local/maple/license/license.dat.
3. Installeer de FLEXlm licentiebeheerder met het installatieshellscript INSTALL_LIC, dat geleverd wordt bij Maple™. Stel de primaire hostnaam voor de machine in voor de licentieserver.
4. Patch het bestand /usr/local/maple/bin/maple.system.type met het volgende:

   ```
      ----- knip ------------------
   *** maple.system.type.orig      Sun Jul  8 16:35:33 2001
   --- maple.system.type   Sun Jul  8 16:35:51 2001
   ***************
   *** 72,77 ****
   --- 72,78 ----
             # the IBM RS/6000 AIX case
             MAPLE_BIN="bin.IBM_RISC_UNIX"
             ;;
   +     "FreeBSD"|\
         "Linux")
             # the Linux/x86 case
           # We have two Linux implementations, one for Red Hat and
      ----- knip einde van patch -----
   ```

   Achter `"FreeBSD"|` mogen geen verdere witvelden staan.

   Deze patch instrueert Maple™ om "FreeBSD" als een Linux® systeem te herkennen. Het shellscript bin/maple roept het shellscript bin/maple.system.type aan, dat op zijn beurt `uname -a` aanroept om achter de naam van het besturingssysteem te komen. Afhankelijk van de naam van het besturingssysteem zoekt het uit welke binairen het moet gebruiken.

5. Start de licentieserver.

   Het volgende script, geïnstalleerd als /usr/local/etc/rc.d/lmgrd, is een gemakkelijke manier om `lmgrd` op te starten:

   ```
      ----- knip ------------

   #! /bin/sh
   PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bi
   n
   PATH=${PATH}:/usr/local/maple/bin:/usr/local/maple/FLEXlm/UNIX/LINUX
   export PATH
   ```

```
LICENSE_FILE=/usr/local/maple/license/license.dat
LOG=/var/log/lmgrd.log

case "$1" in
start)
    lmgrd -c ${LICENSE_FILE} 2>> ${LOG} 1>&2
    echo -n " lmgrd"
    ;;
stop)
    lmgrd -c ${LICENSE_FILE} -x lmdown 2>> ${LOG} 1>&2
    ;;
*)
    echo "Usage: `basename $0` {start|stop}" 1>&2
    exit 64
    ;;
esac

exit 0
   ----- knip ------------
```

6. Maple™ testen:

```
% cd /usr/local/maple/bin
% ./xmaple
```

Nu hoort het programma te draaien. Het is belangrijk om Maplesoft te schrijven om ze te laten weten dat een echte FreeBSD versie gewenst is!

10.4.1. Gemeenschappelijke verborgen gevaren

- De FLEXlm licentiebeheerder kan een lastig programma zijn om mee te werken. Aanvullende documentatie staat op http://www.globetrotter.com/.
- `lmgrd` staat er bekend om erg kieskeurig over het licentiebestand te zijn en core te dumpen als er een probleem is. Een correct licentiebestand ziet er zo uit:

```
# ====================================================
# License File for UNIX Installations ("Pointer File")
# ====================================================
SERVER chillig ANY
#USE_SERVER
VENDOR maplelmg

FEATURE Maple maplelmg 2000.0831 permanent 1 XXXXXXXXXXXX \
         PLATFORMS=i86_r ISSUER="Waterloo Maple Inc." \
         ISSUED=11-may-2000 NOTICE=" Technische Universitat Wien" \
         SN=XXXXXXXXX
```



Het serienummer en de sleutel zijn vervangen door "X"'en. `chillig` is de hostnaam.

Het bewerken van het licentiebestand lukt zolang de regel "FEATURE" niet verandert (die beschermd is door de licentiesleutel).

10.5. MATLAB® installeren

Dit document beschrijft het installatieproces van de Linux® versie van MATLAB® 6.5 op een FreeBSD systeem. Het werkt best goed, met uitzondering van de Java Virtual Machine™ (zie De Java™ runtime-omgeving linken).

De Linux® versie van MATLAB® kan besteld worden bij The MathWorks op http://www.mathworks.com. Er dient ook een licentiebestand of instructies hoe dat te maken te zijn. Het is belangrijk om Maplesoft te schrijven om ze te laten weten dat een echte FreeBSD versie gewenst is!

10.5.1. MATLAB® installeren

Om MATLAB® te installeren:

1. Laad de installatie-CD-ROM en koppel die aan. Start het installatiescript als `root`:

```
# /compat/linux/bin/sh /cdrom/install
```



Het is een grafisch installatieprogramma. Als er foutmeldingen verschijnen dat het programma geen scherm kan openen, kan `setenv HOME ~GEBRUIKER` uitgevoerd worden, waar *GEBRUIKER* de gebruiker is waarmee su(1) is gedaan.

2. Als om de MATLAB® rootmap wordt gevraagd, dient `/compat/linux/usr/local/matlab` opgegeven te worden.



Voer op de commandoregel het volgende uit om de rest van het installatieproces gemakkelijk te houden: `set MATLAB=/compat/linux/usr/local/matlab`.

3. Wijzig het licentiebestand zoals aangegeven tijdens het verkrijgen van de licentie voor MATLAB®.



Dit bestand kan van tevoren gemaakt worden met een tekstverwerker en door het te kopiëren naar $MATLAB/license.dat voordat het installatieprogramma vraagt om het te bewerken.

4. Maak het installatieproces af.

Nu is de installatie van MATLAB® compleet. De volgende stappen "lijmen" het aan het FreeBSD systeem.

10.5.2. Licentiebeheerder starten

1. Maak symbolische links voor de scriptbestanden van de licentiebeheerder:

```
# ln -s $MATLAB/etc/lmboot /usr/local/etc/lmboot_TMW
# ln -s $MATLAB/etc/lmdown /usr/local/etc/lmdown_TMW
```

2. Maak een opstartbestand in /usr/local/etc/rc.d/flexlm. Onderstaand voorbeeld is een gewijzigde versie van het meegeleverde $MATLAB/etc/rc.lm.glnx86. De wijzigingen omvatten bestandslocaties en het starten van de licentiebeheerder onder Linux®-emulatie.

```
#!/bin/sh
case "$1" in
  start)
        if [ -f /usr/local/etc/lmboot_TMW ]; then
              /compat/linux/bin/sh /usr/local/etc/lmboot_TMW -u gebruikersnaam
&& echo 'MATLAB_lmgrd'
        fi
        ;;
  stop)
    if [ -f /usr/local/etc/lmdown_TMW ]; then
            /compat/linux/bin/sh /usr/local/etc/lmdown_TMW  > /dev/null 2>&1
    fi
        ;;
  *)
    echo "Usage: $0 {start|stop}"
    exit 1
    ;;
esac

exit 0
```



Het bestand moet uitvoerbaar zijn:

```
# chmod +x /usr/local/etc/rc.d/flexlm
```

Ook moet bovenstaande *gebruikersnaam* vervangen worden door een geldige gebruikersnaam op het systeem (maar niet door `root`).

3. Start de licentiebeheerder op met het commando:

```
# service flexlm start
```

10.5.3. De Java™ runtime-omgeving linken

Verander de Java™ Runtime Environment Link naar een die werkt op FreeBSD:

```
# cd $MATLAB/sys/java/jre/glnx86
# unlink jre; ln -s ./jre1.1.8 ./jre
```

10.5.4. MATLAB® opstartscript maken

1. Plaats het volgende startscript in /usr/local/bin/matlab:

```
#!/bin/sh
/compat/linux/bin/sh /compat/linux/usr/local/matlab/bin/matlab "$@"
```

2. Geef vervolgens het commando `chmod +x /usr/local/bin/matlab`.



Afhankelijk van de versie van emulators/linux_base, kunnen er fouten optreden als dit script draait. Om dat te voorkomen, dient in /compat/linux/usr/local/matlab/bin/matlab de regel:

```
if [ `expr "$lscmd" : '.*->.*'` -ne 0 ]; then
```

(in versie 13.0.1 staat dit op regel 410) veranderd te worden in:

```
if test -L $newbase; then
```

10.5.5. MATLAB® afsluitscript maken

Het volgende is nodig om een probleem op te lossen dat samenhangt met het onjuist afsluiten van MATLAB®.

1. Maak het bestand $MATLAB/toolbox/local/finish.m dat alleen de volgende regel bevat:

```
! $MATLAB/bin/finish.sh
```



`$MATLAB$` is hier letterlijk bedoeld.



In dezelfde map staan de bestanden finishsav.m en finishdlg.m, die de mogelijkheid geven om de werkomgeving te bewaren vóór het afsluiten. Als één van deze scripts gebruikt wordt, dient de bovenstaande regel direct na het commando `save` ingevoegd te worden.

2. Maak het bestand $MATLAB/bin/finish.sh, dat het volgende bevat:

```
#!/compat/linux/bin/sh
(sleep 5; killall -1 matlab_helper) &
exit 0
```

3. Maak het bestand uitvoerbaar:

```
# chmod +x $MATLAB/bin/finish.sh
```

10.5.6. MATLAB® gebruiken

Nu kan met `matlab` het programma gestart worden.

10.6. Oracle® installeren

10.6.1. Voorwoord

Hieronder wordt het installatieproces van Oracle® 8.0.5 en Oracle® 8.0.5.1 Enterprise Edition voor Linux® op een FreeBSD-machine beschreven.

10.6.2. De Linux®-omgeving installeren

Uit de Portscollectie dienen emulators/linux_base en devel/linux_devtools geïnstalleerd te zijn. Als er problemen zijn met deze ports, kan het zijn dat de pakketten of oudere versies uit de Portscollectie gebruikt moeten worden.

Om de intelligente agent te draaien, moet ook het Red Hat Tcl package geïnstalleerd worden: tcl-8.0.3-20.i386.rpm. Het algemene commando om pakketten te installeren met de officiële RPM port (archivers/rpm) is:

```
# rpm -i --ignoreos --root /compat/linux --dbpath /var/lib/rpm package
```

De installatie van het *package* hoort foutloos te verlopen.

10.6.3. De Oracle®-omgeving creëren

Voordat Oracle® geïnstalleerd kan worden, moet een juiste omgeving opgezet worden. Dit document beschrijft alleen welke *speciale* dingen gedaan moeten worden om Oracle® voor Linux®

op FreeBSD te draaien, en niet wat beschreven staat in de Oracle® installatiehandleiding.

Kerneloptimalisatie

Zoals beschreven staat in de Oracle® installatiehandleiding moet de maximale grootte van het gedeelde geheugen ingesteld worden. Op FreeBSD moet `SHMMAX` niet gebruikt worden. `SHMMAX` wordt slechts uit `SHMMAXPGS` en `PGSIZE` berekend. Daarom dient `SHMMAXPGS` gedefinieerd te worden. Alle andere opties kunnen gebruikt worden zoals in de handleiding staat beschreven. Bijvoorbeeld:

```
options SHMMAXPGS=10000
options SHMMNI=100
options SHMSEG=10
options SEMMNS=200
options SEMMNI=70
options SEMMSL=61
```

Deze opties kunnen naargelang het gebruik van Oracle® ingesteld worden.

Ook de volgende opties dienen in het kernelinstellingenbestand te staan:

```
options SYSVSHM #SysV gedeeld geheugen
options SYSVSEM #SysV semaforen
options SYSVMSG #SysV interprocescommunicatie
```

Oracle® account

Creeër een `oracle` account op dezelfde manier als elk ander account. Het `oracle` account is alleen bijzonder in het opzicht dat het een Linux® shell moet hebben. Dat kan door `/compat/linux/bin/bash` toe te voegen aan /etc/shells en de shell voor het `oracle` account in te stellen op /compat/linux/bin/bash.

Omgeving

Naast de normale Oracle® variabelen als `ORACLE_HOME` en `ORACLE_SID` moeten de volgende omgevingsvariabelen ingesteld worden:

Variabele	Waarde
`LD_LIBRARY_PATH`	`$ORACLE_HOME/lib`
`CLASSPATH`	`$ORACLE_HOME/jdbc/lib/classes111.zip`
`PATH`	`/compat/linux/bin; /compat/linux/sbin; /compat/linux/usr/bin; /compat/linux/usr/sbin; /bin; /sbin; /usr/bin; /usr/sbin; /usr/local/bin; $ORACLE_HOME/bin`

Het is aan te raden om alle omgevingsvariabelen in .profile in te stellen. Een volledig voorbeeld is:

```
ORACLE_BASE=/oracle; export ORACLE_BASE
```

```
ORACLE_HOME=/oracle; export ORACLE_HOME
LD_LIBRARY_PATH=$ORACLE_HOME/lib
export LD_LIBRARY_PATH
ORACLE_SID=ORCL; export ORACLE_SID
ORACLE_TERM=386x; export ORACLE_TERM
CLASSPATH=$ORACLE_HOME/jdbc/lib/classes111.zip
export CLASSPATH
PATH=/compat/linux/bin:/compat/linux/sbin:/compat/linux/usr/bin
PATH=$PATH:/compat/linux/usr/sbin:/bin:/sbin:/usr/bin:/usr/sbin
PATH=$PATH:/usr/local/bin:$ORACLE_HOME/bin
export PATH
```

10.6.4. Oracle® installeren

Wegens een kleine inconsistentie in de Linux® emulator moet de map .oracle aangemaakt worden in /var/tmp voordat het installatieprogramma wordt gestart. De gebruiker `oracle` moet de eigenaar van deze map zijn. Nu hoort Oracle® zonder problemen te installeren. Bij problemen dienen eerst de Oracle® distributie en/of de instellingen gecontroleerd te worden! Nadat Oracle® is geïnstalleerd, moeten de patches uit de volgende twee secties geïnstalleerd worden.

Een veelvoorkomend probleem is dat de adapter voor het TCP-protocol niet goed is geïnstalleerd. De consequentie daarvan is dat er geen TCP-listeners gestart kunnen worden. De volgende acties helpen om dit probleem op te lossen:

```
# cd $ORACLE_HOME/network/lib
# make -f ins_network.mk ntcontab.o
# cd $ORACLE_HOME/lib
# ar r libnetwork.a ntcontab.o
# cd $ORACLE_HOME/network/lib
# make -f ins_network.mk install
```

Hierna dient root.sh nogmaals te draaien!

root.sh patchen

Als Oracle® geïnstalleerd wordt, worden sommige acties die als `root` moeten worden uitgevoerd geregistreerd in een shellscript met de naam root.sh. Dit script komt in de map orainst te staan. De volgende patch dient uitgevoerd te worden op root.sh om het de juiste locatie van `chown` te laten gebruiken of als alternatief kan het script onder een originele Linux® shell gedraaid worden

```
*** orainst/root.sh.orig Tue Oct 6 21:57:33 1998
--- orainst/root.sh Mon Dec 28 15:58:53 1998
***************
*** 31,37 ****
# This is the default value for CHOWN
# It will redefined later in this script for those ports
# which have it conditionally defined in ss_install.h
! CHOWN=/bin/chown
```

```
#
# Define variables to be used in this script
--- 31,37 ----
# This is the default value for CHOWN
# It will redefined later in this script for those ports
# which have it conditionally defined in ss_install.h
! CHOWN=/usr/sbin/chown
#
# Define variables to be used in this script
```

Als Oracle® niet vanaf een CD-ROM wordt geïnstalleerd, kan de broncode van root.sh aangepast worden. Die heet rthd.sh en staat in de map orainst in de broncodestructuur.

genclntsh patchen

Het script `genclntsh` wordt gebruikt om één enkele gedeelde bibliotheek voor de cliënt aan te maken. Het wordt gebruikt tijdens het maken van de demonstraties. Met de volgende patch wordt de definitie van `PATH` uitgecommentarieerd:

```
*** bin/genclntsh.orig Wed Sep 30 07:37:19 1998
--- bin/genclntsh Tue Dec 22 15:36:49 1998
***************
*** 32,38 ****
#
# Explicit path to ensure that we're using the correct commands
#PATH=/usr/bin:/usr/ccs/bin export PATH
! PATH=/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin export PATH
#
# each product MUST provide a $PRODUCT/admin/shrept.lst
--- 32,38 ----
#
# Explicit path to ensure that we're using the correct commands
#PATH=/usr/bin:/usr/ccs/bin export PATH
! #PATH=/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin export PATH
#
# each product MUST provide a $PRODUCT/admin/shrept.lst
```

10.6.5. Oracle® draaien

Als de instructies worden gevolgd, draait Oracle® als op Linux® zelf.

10.7. Gevorderde onderwerpen

Hier wordt beschreven hoe de Linux® binaire compatibiliteit werkt. Het meeste van wat nu volgt is sterk gebaseerd op een e-mailbericht van Terry Lambert tlambert@primenet.com aan FreeBSD babbel mailinglijst (Message ID: <199906020108.SAA07001@usr09.primenet.com>).

10.7.1. Hoe werkt het?

FreeBSD heeft een abstractie met de naam "execution class loader". Dit is een wig in de systeemaanroep execve(2).

Wat er gebeurt is dat FreeBSD een lijst van loaders heeft, in plaats van een enkele loader die terugvalt op de `#!` loader voor het draaien van elke shellinterpreter of shellscript.

Vroeger onderzocht de enige loader op het UNIX® platform het magische getal (in het algemeen de eerste 4 of 8 bytes van het bestand) om te zien of het een binary was die het systeem kende en als dat het geval was laadde het de binaire loader.

Als het niet het binaire type voor het systeem was, faalde de aanroep naar execve(2) en probeerde de shell het als shellopdrachten uit te voeren.

Deze aanname was een standaard voor "wat de huidige shell ook is."

Later werd er een hack gemaakt voor sh(1) om de eerste twee tekens te onderzoeken en als die bestonden uit `:\n` voerde het in plaats hiervan de csh(1) shell uit (het idee is dat SCO de hack als eerste maakte).

Wat FreeBSD nu doet is door een lijst van loaders gaan met een generieke `#!` loader die kennis heeft van interpreters in de zin van de karakters die volgen op de volgende witruimte tot de laatste, met uiteindelijk een terugval op /bin/sh.

Voor Linux® ABI-ondersteuning ziet FreeBSD het magische getal als een ELF-binary (het maakt op dit punt geen onderscheid tussen FreeBSD, Solaris™, Linux® of elk ander besturingssysteem dat een ELF-beeldtype heeft).

De ELF loader zoekt naar een gespecialiseerd *merk*, dat een commentaargedeelte in het ELF-beeld is en dat niet aanwezig is in SVR4/Solaris™ ELF-binairen.

Om Linux®-binairen werkend te krijgen, moeten ze *gemerkt* worden als het type `Linux` met brandelf(1):

```
# brandelf -t Linux bestand
```

Als dit gedaan is, ziet de ELF loader het `Linux`-merk in het bestand.

Als de ELF loader het `Linux`-merk tegenkomt, verplaatst de loader een pointer in de `proc`-structuur. Alle systeemaanroepen worden met deze pointer geïndexeerd (in een traditioneel UNIX® systeem is dit de `sysent[]`-structuurarray, die de systeemaanroepen bevat). Ook wordt het proces gemerkt voor speciale behandeling door de valstrikvector van de signaal-trampolinecode samen met nog meer (kleine) aanpassingen die door de Linux® kernelmodule worden afgehandeld.

De Linux® kernelmodule bevat naast andere dingen een lijst van `sysent[]`-ingangen waarvan de adressen in de kernelmodule staan.

Als een systeemaanroep door de Linux®-binary wordt aangeroepen, verwijdert de valstrikcode de referentie aan de functiepointer van de systeemaanroep en geeft die de ingangspunten van de

systeemaanroep van Linux® en niet van FreeBSD.

Verder *reroot* de Linux®-modus dynamisch lookups. Dit is wat de optie `union` (*niet* het `unionfs` bestandssysteemtype!) voor het aankoppelen van bestandssystemen effectief doet. Eerst wordt een poging gedaan om het bestand in de map /compat/linux/origineel-pad op te zoeken en *vervolgens* alleen als dat mislukt, wordt het bestand in /origineel-pad opgezocht. Dit zorgt ervoor dat binairen die andere binairen nodig hebben kunnen draaien (zo kan bijvoorbeeld de Linux®-gereedschapskist geheel onder Linux® ABI-ondersteuning draaien). Dit betekent ook dat Linux®-binairen FreeBSD-binairen kunnen laden en draaien als er geen overeenkomende Linux®-binairen zijn en dat er een uname(1)-opdracht in de mappenstructuur /compat/linux gezet kan worden om er zeker van te zijn dat Linux®-binairen niet kunnen weten dat ze niet op Linux® draaien.

Effectief bevindt er zich een Linux®-kernel in de FreeBSD-kernel. De verschillende onderliggende functies die alle functies implementeren die de kernel aanbiedt, zijn dezelfde tabelingangen voor de systeemaanroepen van FreeBSD als van Linux®: bestandssysteembewerkingen, bewerkingen op het virtuële geheugen, signaalaflevering, System V IPC, enzovoort. Het enige verschil is dat FreeBSD-binairen de *lijm* functies voor FreeBSD krijgen en dat de Linux®-binairen de *lijm*-functies voor Linux® krijgen (de meeste oudere besturingssystemen hadden alleen hun eigen *lijm*-functies: adressen van functies die in een statische globale `sysent[]` structuurarray werden opgeslagen, in plaats van adressen van functies waarvan dynamisch een geïnitialiseerde pointer wordt verwijderd in de `proc`-structuur van het proces dat de aanroep doet).

Welke is de eigenlijke FreeBSD ABI? Dat maakt niet uit. Eigenlijk is het enige verschil dat (op dit moment; dit kan eenvoudig veranderen in een toekomstige uitgave, en dat gebeurt waarschijnlijk na deze uitgave) de *lijm*-functies van FreeBSD statisch gelinkt zijn in de kernel en dat de *lijm* -functies van Linux® zowel statisch gelinkt kunnen worden als dat ze door een kernelmodule worden benaderd.

Maar is dit nu echt emulatie? Nee. Het is een ABI-implementatie, geen emulatie. Er is geen emulator (of simulator, om de volgende vraag voor te zijn) bij betrokken.

Dus waarom wordt het dan soms "Linux®-emulatie" genoemd? Om het moeilijk te maken om FreeBSD te verkopen! Serieus, het is zo omdat de historische implementatie in een tijd werd gedaan toen er echt geen ander woord was om te beschrijven wat er aan de hand was, om te zeggen dat FreeBSD Linux®-binairen draaide was niet waar als de code niet in de kernel gecompileerd werd of als een module geladen werd en er moest een woord zijn voor hetgeen geladen werd. Vandaar "de Linux®-emulator".

Unresolved directive in book.adoc - include::wine/_index.adoc[leveloffset=+1]

Deel III: Systeembeheer

De resterende hoofdstukken van het handboek beslaan alle aspecten van het FreeBSD systeembeheer. Elk hoofdstuk begint met de leerdoelen van het hoofdstuk en geeft ook een beschrijving van de verwachte voorkennis.

Deze hoofdstukken kunnen worden gelezen als naslagwerk. Ze kunnen in willekeurige volgorde worden gelezen en zijn niet vereist om te kunnen beginnen met FreeBSD.

Hoofdstuk 11. Instellingen en optimalisatie

11.1. Overzicht

Systeeminstellingen zijn een belangrijk aspect van FreeBSD. Correcte instellingen helpen moeilijkheden bij toekomstige upgrades te voorkomen. In dit hoofdstuk wordt het instellen van FreeBSD beschreven, alsmede een aantal prestatiebevorderende maatregelen waarmee een FreeBSD systeem geoptimaliseerd kan worden.

Na het lezen van dit hoofdstuk weet de lezer:

- Hoe efficiënt om te gaan met bestandssystemen en wisselpartities;
- De grondbeginselen van het rc.conf instellingensysteem en van het opstarten van toepassingen (diensten) met /usr/local/etc/rc.d;
- Hoe een netwerkkaart ingesteld en getest wordt;
- Hoe virtuele hosts op netwerkapparatuur ingesteld worden;
- Hoe de instellingenbestanden in /etc gebruikt worden;
- Hoe FreeBSD geoptimaliseerd kan worden met `sysctl`-variabelen;
- Hoe schijfprestaties te verbeteren en hoe kernelbeperkingen gewijzigd kunnen worden.

Veronderstelde voorkennis:

- De grondbeginselen van UNIX® en FreeBSD (UNIX® beginselen) begrijpen;
- Bekend zijn met de grondbeginselen van kernelinstellingen en compilatie (De FreeBSD-kernel instellen).

11.2. Initiële instellingen

11.2.1. Partitioneren

Basispartities

Bij het aanmaken van bestandssystemen met bsdlabel(8) of sysinstall(8) is het van belang dat op een harde schijf de gegevensoverdracht het snelst is aan de buitenste sporen en het langzaamst aan de binnenste. Kleinere en veelgebruikte bestandssystemen kunnen daarom het beste aan de buitenkant van de schijf geplaatst worden, terwijl grotere partities als /usr meer naar de binnenkant van de schijf geplaatst kunnen worden. Het is een goed idee om partities aan te maken in deze of gelijksoortige volgorde: root, swap, /var, /usr.

De grootte van de partitie /var hangt af van de wijze waarop de machine gebruikt gaat worden. Het bestandssysteem /var wordt gebruikt voor onder meer postbussen, logbestanden en printergegevens en -wachtrijen. Postbussen en logbestanden kunnen onverwacht groot worden, afhankelijk van het aantal systeemgebruikers en de bewaarduur van logbestanden. De meeste gebruikers zullen zelden meer dan ongeveer een gigabyte aan vrije schijfruimte op /var nodig hebben.



Er zijn een aantal gevallen waar een grote hoeveelheid ruimte in /var/tmp nodig is. Wanneer er nieuwe software wordt geïnstalleerd met pkg_add(1) pakken de pakketprogramma's een tijdelijke kopie van de pakketten uit in /var/tmp. Grote softwarepakketten, zoals Firefox, OpenOffice of LibreOffice kunnen lastig zijn om te installeren wanneer er onvoldoende vrije schijfruimte beschikbaar is onder /var/tmp.

De partitie /usr bevat veel van de benodigde systeembestanden, waaronder de ports(7) collectie (aanbevolen) en de broncode (optioneel). Beide zijn optioneel tijdens de installatie, maar we raden voor deze partitie tenminste 2 gigabyte aan.

Het is verstandig rekening te houden met de vereiste schijfruimte bij het kiezen van partitiegroottes. Als in een partitie onvoldoende vrije schijfruimte is, terwijl een andere vrijwel niet gebruikt wordt, is dat een vervelend en niet optimaal oplosbaar probleem.



sysinstall(8)'s `Auto-defaults` partitiekeuze kan in de ervaring van sommige gebruikers mogelijk te kleine /var en / partities opleveren. Partitioneren moet verstandig en niet te zuinig gebeuren.

Wisselpartities (swap)

De vuistregel is dat het wisselbestand ongeveer het dubbele van de grootte van het systeemgeheugen (RAM) moet zijn. Als de machine bijvoorbeeld 128 megabytes geheugen heeft, kan het beste een wisselbestand van (tenminste) 256 megabytes gebruikt worden. Minder dan 256 megabytes swap is in dit geval af te raden. Systemen met weinig geheugen kunnen overigens beter functioneren met meer swap. Ook is het verstandig rekening te houden met eventuele geheugenuitbreiding in de toekomst. Bovendien zijn de VM paging-algoritmen van de kernel zo afgestemd dat ze het beste presteren bij een wisselbestand van tenminste tweemaal de grootte van het geheugen. Een te kleine swap kan dus inefficiënties in de VM-code tot gevolg hebben en mogelijk problemen veroorzaken als het systeemgeheugen uitgebreid wordt.

Op grotere systemen met meerdere SCSI-schijven (of meerdere IDE-schijven op verschillende controllers) is het aan te raden om op elke schijf een wisselpartitie in te stellen (dit kan tot en met vier schijven), elk met ongeveer dezelfde grootte. De kernel kan met arbitraire groottes werken, maar interne datastructuren schalen tot viermaal de grootste swappartitie. De kernel kan de beschikbare ruimte voor het wisselbestand het meest optimaal indelen als de partities ongeveer even groot zijn. Een grote swap is prima, ook als ze zelden gebruikt wordt. Zo kan het gemakkelijker zijn om een (uit de hand gelopen) proces dat het systeem grotendeels bezet houdt te beëindigen, voordat er opnieuw opgestart moet worden.

Waarom partitioneren?

Waarom niet één enkele grote partitie gebruiken? Er zijn verscheidene redenen waarom dit niet zo'n goed idee is. De verschillende partities hebben hun eigen karakteristieke operationele gedrag en vereisten. Door ze te scheiden zijn er betere mogelijkheden om het systeem te optimaliseren. Vanaf de / en /usr partities wordt bijvoorbeeld vooral gelezen en er wordt weinig naar geschreven, terwijl er in /var en /var/tmp zowel veel gelezen als geschreven wordt.

Door een systeem goed te partitioneren wordt vermeden dat fragmentatie die optreedt in de kleinere partities met veel schrijfactiviteit doorsijpelt naar partities die vooral lees-intensief zijn. Door schrijf-intensieve partities aan het begin van de schijf te plaatsen, zijn de prestaties wat betreft invoer/uitvoer het beste daar waar het het meest nodig is. Ofschoon er natuurlijk ook de best mogelijke in/uit prestaties wenselijk zijn in de grotere partities, weegt het plaatsen van deze bestandssystemen aan het begin van de schijf niet tegen de voordelen van het plaatsen van /var aan het begin van de schijf (na root en swap) voor de totale snelheid van het systeem. Tenslotte zijn er veiligheidsoverwegingen. Een compacte en nette rootpartitie die vrijwel alleen-lezen is, heeft een betere kans om een nare crash te overleven.

11.3. Hoofdinstellingen

De voornaamste lokatie voor systeeminstellingen is /etc/rc.conf. Dit bestand bevat een scala aan instellingen, die gebruikt wordt om het systeem in te stellen bij het opstarten. De naam impliceert dit al. Het is informatie voor de rc* bestanden (rc staat voor "resource configuration" of broninstellingen).

De systeembeheerder wordt geacht regels toe te voegen aan rc.conf om de standaardinstellingen uit /etc/defaults/rc.conf aan te passen. Het standaardbestand moet niet letterlijk gekopiëerd worden naar /etc. Het bevat standaardwaardes en is niet bedoeld als voorbeeld. Alle wijzigingen die specifiek zijn voor een systeem horen in /etc/rc.conf thuis.

In een clusterscenario is het nuttig om systeemspecifieke instellingen te scheiden van algemene instellingen die voor het hele cluster gelden. Hiervoor kunnen een aantal strategieën worden gebruikt. De aanbevolen benadering is om systeem-specifieke instellingen in /etc/rc.conf.local te plaatsen. Een voorbeeld:

- /etc/rc.conf:

```
sshd_enable="YES"
keyrate="fast"
defaultrouter="10.1.1.254"
```

- /etc/rc.conf.local:

```
hostname="node1.example.org"
ifconfig_fxp0="inet 10.1.1.1/8"
```

rc.conf kan vervolgens naar elk systeem gedistribueerd worden met `rsync` of een gelijksoortig programma, terwijl rc.conf.local uniek blijft.

Het actualiseren van het systeem met sysinstall(8) of `make world` overschrijft rc.conf niet, zodat de bestaande systeeminstellingen niet verloren gaan.



Het instellingenbestand /etc/rc.conf wordt gelezen door sh(1). Dit stelt systeembeheerders in staat om een zekere hoeveelheid logica aan dit bestand toe te voegen, dat kan helpen in het creëren van zeer ingewikkelde

configuratiescenario's. Bekijk rc.conf(5) voor meer informatie over dit onderwerp.

11.4. Toepassingen instellen

Geïnstalleerde toepassingen hebben meestal hun eigen instellingenbestanden, met hun eigen syntaxis, etc. Het is van belang deze bestanden apart te houden van het basissysteem, zodat ze makkelijk gelokaliseerd kunnen worden en beheerd kunnen worden met de hulpmiddelen voor pakketbeheer.

Deze bestanden worden meestal geïnstalleerd in /usr/local/etc. Als een toepassing een uitgebreide verzameling bestanden voor instellingen heeft, wordt er een submap voor aangemaakt.

Bij de installatie van een port of pakket, worden normaliter ook voorbeeldbestanden met instellingen geïnstalleerd. Deze zijn doorgaans te herkennen aan een toevoegsel .default. Als er geen bestaande instellingenbestanden voor de toepassing zijn, kunnen ze gemaakt worden door de .default-bestanden te kopiëren.

Een voorbeeld is de map /usr/local/etc/apache:

```
-rw-r--r--  1 root  wheel   2184 May 20  1998 access.conf
-rw-r--r--  1 root  wheel   2184 May 20  1998 access.conf.default
-rw-r--r--  1 root  wheel   9555 May 20  1998 httpd.conf
-rw-r--r--  1 root  wheel   9555 May 20  1998 httpd.conf.default
-rw-r--r--  1 root  wheel  12205 May 20  1998 magic
-rw-r--r--  1 root  wheel  12205 May 20  1998 magic.default
-rw-r--r--  1 root  wheel   2700 May 20  1998 mime.types
-rw-r--r--  1 root  wheel   2700 May 20  1998 mime.types.default
-rw-r--r--  1 root  wheel   7980 May 20  1998 srm.conf
-rw-r--r--  1 root  wheel   7933 May 20  1998 srm.conf.default
```

Aan de grootte van de bestanden is te zien dat alleen srm.conf gewijzigd is. Als later de port Apache wordt vernieuwd, wordt dit bestand niet overschreven.

11.5. Diensten starten

Veel gebruikers kiezen ervoor om software van derden te installeren op FreeBSD vanuit de Portscollectie. In veel gevallen is het noodzakelijk om de software dusdanig in te stellen dat het opstart tijdens het opstarten van de computer. Diensten zoals mail/postfix of www/apache22 zijn slechts twee voorbeelden van softwarepakketten die gestart kunnen worden tijdens de systeemstart. In deze paragraaf wordt toegelicht hoe software van derde partijen kan worden gestart.

In FreeBSD worden de meeste diensten, zoals cron(8), door de opstartscripts van het systeem gestart. Deze scripts kunnen verschillen tussen FreeBSD en leverancierversies, echter het meest belangrijke aspect om in gedachten te houden is dat hun opstartinstellingen verwerkt kunnen worden door simpele opstartscripts.

11.5.1. Uitgebreide applicatieinstellingen

Nu FreeBSD rc.d heeft, zijn de instellingen van applicaties die mee moeten opstarten versimpeld en rijker aan mogelijkheden. Door gebruik te maken van de sleutelwoorden die in de paragraaf rc.d behandeld worden, kunnen applicaties nu starten na andere diensten. DNS kan bijvoorbeeld extra opties meekrijgen van /etc/rc.conf in plaats van hard ingestelde opties in het opstartscript. Een basisscript ziet er ongeveer als volgt uit:

```
#!/bin/sh
#
# PROVIDE: utility
# REQUIRE: DAEMON
# KEYWORD: shutdown

. /etc/rc.subr

name=utility
rcvar=utility_enable

command="/usr/local/sbin/utility"

load_rc_config $name

#
# VERANDER DE STANDAARDWAARDEN HIER NIET
# STEL ZE IN HET BESTAND /etc/rc.conf IN
#
utility_enable=${utility_enable-"NO"}
pidfile=${utility_pidfile-"/var/run/utility.pid"}

run_rc_command "$1"
```

Dit script zorgt ervoor dat utility wordt gestart na de pseudodienst DAEMON. Het biedt ook de mogelijkheid voor het instellingen en volgen van het PID of het proces-ID bestand.

Voor deze applicatie kan dan de volgende regel in /etc/rc.conf geplaatst worden:

```
utility_enable="YES"
```

Deze methode maakt het volgende mogelijk: makkelijker commandoregelopties manipuleren, importeren van standaardfuncties uit /etc/rc.subr, compatibiliteit met het gereedschap rcorder(8) en het levert makkelijkere configuratie via rc.conf.

11.5.2. Diensten met diensten starten

Andere diensten, zoals POP3-server daemons, IMAP, enzovoort, kunnen gestart worden door gebruik te maken van inetd(8). Daaraan is voorafgegaan dat die dienst uit de Portscollectie is geïstalleerd en dat er een regel met instellingen is toegevoegd aan /etc/inetd.conf of één van de

bestaande niet-actieve regels is geactiveerd. Werken met inetd en zijn instellingen wordt uitgebreid toegelicht in de paragraaf over inetd.

In sommige gevallen is het handiger om cron(8) te gebruiken om diensten te starten. Deze aanpak heeft een aantal voordelen omdat `cron` start als de eigenaar van crontab. Dit stelt reguliere gebruikers in staat om sommige applicaties te starten en te onderhouden.

`cron` levert een unieke optie: in plaats van een tijdsspecificatie kan `@reboot` gebruikt worden. Dit zorgt ervoor dat de taak gestart wordt als cron(8) gestart wordt, meestal tijdens een systeemstart.

11.6. `cron` instellen

Een zeer nuttig hulpprogramma in FreeBSD is cron(8). De daemon `cron` draait op de achtergrond en controleert voortdurend /etc/crontab. Ook controleert `cron` de map /var/cron/tabs, op zoek naar nieuwe crontab bestanden. Deze crontab bestanden bevatten informatie over specifieke taken die `cron` moet verrichten op gezette tijden.

`cron` gebruikt twee verschillende soorten instellingenbestanden: de systeemcrontab en gebruikerscrontabs. Deze formaten verschillen alleen in het zesde en verdere velden. In de systeemcrontab zal `cron` het commando draaien als de gebruiker die in het zesde veld is opgegeven. In een gebruikerscrontab draaien alle commando's onder de gebruiker die de crontab heeft aangemaakt, dus is het zesde veld het laatste veld; dit is een belangrijk beveiligingsaspect. Het laatste veld is altijd het commando dat gedraaid wordt.



Gebruikerscrontabs geven individuele gebruikers de mogelijkheid om bepaalde terugkerende taken automatisch te laten uitvoeren zonder dat `root`-rechten noodig zijn. Commando's in de crontab van een gebruiker worden uitgevoerd met de rechten van de eigenaar.

`root` kan ook een gebruikerscrontab aanleggen net als elke andere gebruiker. Dit is niet dezelfde als /etc/crontab, de systeemcrontab. Omdat de systeemcrontab in de praktijk de commando's als root uitvoert, is het doorgaans niet nodig om een gebruikerscrontab voor `root` te maken.

/etc/crontab (de systeemcrontab) ziet er uit als volgt:

```
# /etc/crontab - root's crontab for FreeBSD
#
# $FreeBSD: src/etc/crontab,v 1.32 2002/11/22 16:13:39 tom Exp $
#①
#
SHELL=/bin/sh
PATH=/etc:/bin:/sbin:/usr/bin:/usr/sbin ②
HOME=/var/log
#
#
#minuut uur     mdag    maand   wdag    wie     commando ③
#
#
```

```
*/5     *       *       *       *       root    /usr/libexec/atrun ④
```

① Zoals in de meeste instellingenbestanden van FreeBSD zijn regels die met het karakter `#` beginnen commentaar. Commentaar wordt gebruikt als uitleg en geheugensteun. Commentaar dient niet vermengd te worden met commando's, anders wordt het commentaar opgevat als deel van het commando. Blanco regels worden genegeerd.

② Eerst worden omgevingsvariabelen gedefiniëerd. Hoervoor wordt het is-gelijk karakter (`=`) gebruikt. In het bovenstaande voorbeeld wordt het gebruikt voor de variabelen `SHELL`, `PATH` en `HOME`. Als de regel `SHELL` ontbreekt, gebruikt `cron` standaard `sh` als shell. Voor de omgevingsvariabele `PATH` bestaat geen standaardwaarde. Als `PATH` ontbreekt moeten absolute paden gebruikt worden. Als `HOME` ontbreekt, gebruikt `cron` de thuismap van de gebruiker die `cron` aanroept.

③ In deze commentaarregel staan de zeven velden van een crontabdefinitie. Dit zijn `minuut`, `uur`, `mdag`, `maand`, `wdag`, `wie` en `commando`. De betekenissen liggen voor de hand: `minuut` is het aantal minuten van het tijdstip waarop het commando moet worden uitgevoerd; `uur` geeft het uur aan; `mdag` staat voor de dag van de maand; `maand` staat voor het maandnummer en `wdag` geeft de dag van de week aan. Het veld `wie` is bijzonder en bestaat alleen in /etc/crontab. Het geeft aan als welke gebruiker het commando uitgevoerd moet worden. Het laatste veld bevat het uit te voeren commando.

④ In deze regel worden aan de hierboven besproken opties waarden toegekend. Er wordt gebruik gemaakt van `/5 en` karakters. Deze betekenen "eerst-laatst" en kunnen gezien worden als *telkens*. In deze regel staat dus dat `atrun` elke vijf minuten moet worden uitgevoerd door `root`, ongeacht welke dag of maand het is. Meer informatie over `atrun` staat in atrun(8).Commando's kunnen een willekeurig aantal opties of argumenten meekrijgen. Als commando's echter meerdere regels nodig hebben moeten deze regels afgebroken worden met een backslash "\" karakter, om aan te geven dat ze op de volgende regel vervolgd worden.

Dit is de basisopzet voor elk crontab bestand. De enige uitzondering is de aanwezigheid van veld zes, waar de gebruikersnaam wordt aangegeven. Dit veld bestaat alleen in de systeemversie van /etc/crontab. Voor crontab-bestanden van individuele gebruikers moet dit veld worden weggelaten.

11.6.1. Een crontab installeren



De onderstaande procedure moet niet gebruikt worden om de systeemcrontab /etc/crontab te wijzigen of te installeren. Er kan een gewone editor gebruikt worden. `cron` ziet dat het bestand veranderd is en begint direct met het gebruiken van de nieuwe versie. Deze FAQ vraag geeft verdere uitleg.

Om een nieuwe crontab te installeren moet eerst een bestand in het juiste formaat gemaakt worden en daarna moet het geiuml;nstalleerd worden met commando `crontab`:

```
# crontab crontabbestand
```

In dit voorbeeld is crontabbestand de naam van een eerder gemaakt crontab-bestand.

Er bestaat ook een optie om een lijst van geïnstalleerde crontab-bestanden op te vragen, namelijk

de optie `-l` van `crontab`.

Gebruikers die hun eigen crontabbestand willen schrijven zonder het gebruik van een sjabloon, kunnen gebruik maken van `crontab -e`. Dit opent de `EDITOR` met een leeg bestand. Als het bestand wordt opgeslagen en de editor wordt afgesloten, wordt het bestand automatisch als crontab geïnstalleerd.

Een gebruikers crontab kan verwijderd worden door de met `crontab` de optie `-r` te gebruiken.

11.7. Gebruik van rc met FreeBSD

Sinds 2002 gebruikt FreeBSD het NetBSD rc.d systeem bij het opstarten van het systeem. Veel van de bestanden in /etc/rc.d zijn scripts voor basisdiensten die werken met de opties `start`, `stop` en `restart`, analoog aan hoe diensten die via een port of pakket zijn geïnstalleerd gestart worden met de scripts in /usr/local/etc/rc.d. sshd(8) kan bijvoorbeeld als volgt herstart worden:

```
# service restart
```

Deze procedure is vrijwel gelijk voor andere diensten. Uiteraard worden diensten meestal automatisch tijdens het opstarten van de computer gestart zoals in rc.conf(5) staat. Om de Network Address Translation daemon bij het opstarten te laten starten is de volgende regel in /etc/rc.conf bijvoorbeeld voldoende:

```
natd_enable="YES"
```

Als er reeds een `natd_enable="NO"` regel is, kan `NO` gewoon in `YES` veranderd worden. De rc scripts starten, voor zover nodig, automatisch andere afhankelijke diensten.

Omdat het rc.d systeem in eerste instantie bedoeld is om diensten te starten en stoppen bij het opstarten en afsluiten van het systeem, werken de standaardopties `start`, `stop` en `restart` alleen als de juiste variabelen in /etc/rc.conf zijn ingesteld. Het commando `sshd restart` alleen dan als `sshd_enable` de waarde `YES` heeft in /etc/rc.conf. Als er een dienst gestart, gestopt of herstart moet worden, ongeacht de definities in /etc/rc.conf, moet het commando voorafgegaan worden door "one". Dus om `sshd` te herstarten ongeacht de instellingen in /etc/rc.conf, voldoet het volgende commando:

```
# service sshd onerestart
```

Het is eenvoudig te controleren of een dienst is ingeschakeld is in /etc/rc.conf door het bijpassende rc.d-script uit te voeren met de optie `rcvar`. Voor `sshd`:

```
# service sshd rcvar
# sshd
$sshd_enable=YES
```



De tweede regel (`# sshd`) is de uitvoer van `sshd`, geen `root`-console.

De optie `status` wordt gebruikt om vast te stellen of een dienst gestart is. Om bijvoorbeeld te controleren of `sshd` gestart is:

```
# service sshd status
sshd is running as pid 433.
```

In sommige gevallen is het ook mogelijk om een dienst te herstarten met de optie `reload`. Dan wordt er getracht een signaal te sturen aan een individuele dienst, waarbij de dienst de bestanden met instellingen opnieuw in moet lezen. Meestal komt dit neer op het verzenden van het signaal `SIGHUP`. Deze optie wordt niet door alle diensten ondersteund.

Het rc.d-systeem wordt niet alleen gebruikt voor netwerkdiensten, maar ook voor het merendeel van de systeemstart. In dit kader is bijvoorbeeld het bestand bgfsck interessant. Als dit script wordt uitgevoerd, wordt de volgende boodschap getoond:

```
Starting background file system checks in 60 seconds.
```

Dit script wordt dus gebruikt voor bestandssysteemcontrole in de achtergrond, hetgeen alleen tijdens de systeemstart gebeurt.

Veel systeemdiensten zijn afhankelijk van andere diensten om correct te kunnen functioneren. Zo starten NIS en andere RPC-gebaseerde diensten niet als de dienst `rpcbind` (portmapper) nog niet draait. Om dit te stroomlijnen wordt informatie over afhankelijkheden en andere metagegevens ingevoegd in het commentaar bovenaan het opstartscript. Deze commentaarregels worden vervolgens tijdens de systeemstart met rcorder(8) verwerkt om zo vast te stellen in welke volgorde de systeemdiensten gestart moeten worden.

De volgende woorden moeten in alle opstartscripts staan (ze zijn benodigd door rc.subr(8) om het opstartscript te activeren):

- `PROVIDE`: geeft aan in welke diensten dit bestand voorziet.
- `REQUIRE`: geeft aan welke andere diensten vereist zijn voor deze dienst. Dit script wordt uitgevoerd *na* de aangegeven diensten.
- `BEFORE`: geeft diensten aan die afhankelijk zijn van deze dienst. Dit bestand wordt uitgevoerd *vóór* de aangegeven diensten.

Met deze methode kan een systeembeheerder gemakkelijk systeemdiensten besturen, zonder gedoe met "runlevels" zoals bij sommige andere UNIX® systemen.

Meer informatie over het rc.d-systeem staat in rc(8) en rc.subr(8). Als u geïnteresseerd bent in het schrijven van uw eigen rc.d-script of om de huidige scripts te verbeteren is wellicht dit artikel interessant.

11.8. Netwerkkaarten instellen

Het is tegenwoordig nauwelijks voorstelbaar dat een computer geen netwerkverbinding heeft. Het toevoegen en instellen van een netwerkkaart is een gebruikelijke taak voor een FreeBSD-beheerder.

11.8.1. Het juiste stuurprogramma vinden

Voor het zoeken begint, moet duidelijk zijn om welke kaart het gaat, welke chip erop zit en of het een PCI- of ISA-kaart is. FreeBSD ondersteunt vele kaarten. Op de Hardware Compatibiliteitslijst voor de betreffende uitgave staan de kaarten die ondersteund worden.

Als duidelijk is dat een kaart ondersteund wordt, moet vastgesteld worden wat het geschikte stuurprogramma is. In het bestand /usr/src/sys/conf/NOTES staat een lijst van stuurprogramma's voor netwerkinterfaces met wat informatie over de ondersteunde chipsets of kaarten. In geval van twijfel biedt de hulppagina voor het stuurprogramma (`man`) vaak uitkomst. In het algemeen bevat deze meer informatie over de ondersteunde hardware en mogelijke problemen die kunnen optreden.

Als een veelgebruikte kaart gebruikt wordt, hoeft meestal niet ver gezocht te worden. Stuurprogramma's voor veelvoorkomende netwerkinterfaces zijn al aanwezig in de algemene kernel GENERIC. In dat geval wordt zo'n kaart al gevonden bij het opstarten, bijvoorbeeld met het volgende bericht:

```
dc0: <82c169 PNIC 10/100BaseTX> port 0xa000-0xa0ff mem 0xd3800000-0xd38
000ff irq 15 at device 11.0 on pci0
miibus0: <MII bus> on dc0
bmtphy0: <BCM5201 10/100baseTX PHY> PHY 1 on miibus0
bmtphy0:  10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto
dc0: Ethernet address: 00:a0:cc:da:da:da
dc0: [ITHREAD]
dc1: <82c169 PNIC 10/100BaseTX> port 0x9800-0x98ff mem 0xd3000000-0xd30
000ff irq 11 at device 12.0 on pci0
miibus1: <MII bus> on dc1
bmtphy1: <BCM5201 10/100baseTX PHY> PHY 1 on miibus1
bmtphy1:  10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto
dc1: Ethernet address: 00:a0:cc:da:da:db
dc1: [ITHREAD]
```

In dit voorbeeld zitten er twee kaarten in het systeem die het stuurprogramma dc(4) gebruiken.

Als het stuurprogramma voor een NIC geen onderdeel is van de kernel GENERIC, dan dient het juiste stuurprogramma voor die NIC geladen te worden. Dit kan op twee manieren:

- De meest eenvoudige manier is het laden van een kernelmodule voor een netwerkkaart met kldload(8) of automatisch tijdens het opstarten van het systeem door de benodigde regel toe te voegen aan /boot/loader.conf. Niet alle NIC-stuurprogramma's zijn als module beschikbaar. Zo zijn er bijvoorbeeld geen modules beschikbaar voor ISA-kaarten.

- Ondersteuning voor een kaart kan ook in de kernel gecompileerd worden. In /usr/src/sys/conf/NOTES, /usr/src/sys/arch/conf/NOTES en de hulppagina van het stuurprogramma is na te lezen wat er in het kernelinstellingenbestand moet staan. In De FreeBSD-kernel instellen staat meer informatie over het compileren van een eigen kernel. Als een netwerkkaart al bij het opstarten wordt herkend door de kernel GENERIC, is er geen reden om een andere kernel te bouwen.

Gebruik maken van Windows® NDIS-stuurprogramma's

Helaas zijn er nog steeds veel leveranciers die geen schema's leveren voor stuurprogramma's aan de open-source gemeenschap, omdat ze deze informatie beschouwen als handelsgeheimen. Als gevolg daarvan hebben de ontwikkelaars van FreeBSD en andere projecten twee keuzes: zelf de stuurprogramma's ontwikkelen door een langdurig en pijnlijk proces van de huidige stuurprogramma's te ontcijferen, of door gebruik te maken van de huidige binaire bestanden voor het Microsoft® Windows® platform. De meeste ontwikkelaars, inclusief diegeen die gekoppeld zijn aan FreeBSD, hebben voor het laatste gekozen.

Dankzij de bijdragen van Bill Paul (wpaul) is er "native" ondersteuning voor de Network Driver Interface Specification (NDIS). De FreeBSD NDISulator (ook wel bekend als Project Evil) neemt een binair Windows® stuurprogramma en doet net alsof deze in een Windows® systeem draait. Omdat het stuurprogramma ndis(4) een Windows® binary gebruikt; draait het alleen op i386™- en amd64-systemen. PCI, CardBus, PCMCIA (PC-Card) en USB-apparaten worden ondersteund.

Om de NDISulator te gebruiken zijn drie dingen nodig:

1. De bronbestanden van de kernel
2. Een Windows® XP stuurprogramma (met de extensie .SYS)
3. Een instellingenbestand van het Windows® XP stuurprogramma (met de extensie .INF)

Lokaliseer de bestanden voor uw specifieke kaart. Over het algemeen kunnen deze gevonden worden op de bijgeleverde CD's of op de website van de leverancier. In de volgende voorbeelden maken we gebruik van W32DRIVER.SYS en W32DRIVER.INF.

De bit-breedte van het stuurprogramma moet overeenkomen met die van het stuurprogramma. Gebruik voor FreeBSD/i386 een 32-bits Windows® stuurprogramma. Voor FreeBSD/amd64 is een 64-bits Windows® stuurprogramma nodig.

De volgende stap is het compileren van het binaire stuurprogramma in een laadbare kernelmodule. Gebruik ndisgen(8) als `root`:

```
# ndisgen /pad/naar/W32DRIVER.INF
/pad/naar/W32DRIVER.SYS
```

ndisgen(8) is interactief en vraagt om extra informatie als het dat nodig heeft. Een nieuwe kernel-module wordt in de huidige map geschreven. Gebruik kldload(8) om de nieuwe module te laden:

```
# kldload ./W32DRIVER_SYS.ko
```

Naast de gegenereerde kernelmodule, moeten ook de modules ndis.ko en if_ndis.ko geladen worden. Dit zou automatisch moeten gebeuren als er een module geladen wordt dit afhankelijk is van ndis(4). Als ze handmatig ingeladen moeten worden gebruik dan de volgende commando's:

```
# kldload ndis
# kldload if_ndis
```

Het eerste commando laadt de stuurprogrammawrapper voor de NDIS miniport, de tweede laadt de daadwerkelijke netwerkinterface.

Controleer nu dmesg(8) om te zien of er ergens fouten voorkomen. Als alles goed gegaan is ziet u ongeveer het volgende:

```
ndis0: <Wireless-G PCI Adapter> mem 0xf4100000-0xf4101fff irq 3 at device 8.0 on pci1
ndis0: NDIS API version: 5.0
ndis0: Ethernet address: 0a:b1:2c:d3:4e:f5
ndis0: 11b rates: 1Mbps 2Mbps 5.5Mbps 11Mbps
ndis0: 11g rates: 6Mbps 9Mbps 12Mbps 18Mbps 36Mbps 48Mbps 54Mbps
```

Vanaf dit moment kan de ndis0 net zo gebruikt worden als elke andere netwerkkaart (bv. dc0).

Het systeem kan geconfigureerd worden zodat de NDIS-modules automatisch gestart worden tijdens het opstarten van het systeem, net zoals bij andere modules. Kopieer eerst de gegenereerde module W32DRIVER_SYS.ko naar de map /boot/modules. Voeg daarna de volgende regel toe aan /boot/loader.conf:

```
W32DRIVER_SYS_load="YES"
```

11.8.2. De netwerkkaart instellen

Nadat een geschikt stuurprogramma geladen is, moet de kaart nog ingesteld worden. Mogelijk is dit al gebeurd door sysinstall tijdens de installatie.

Om de instellen van de netwerkkaarten weer te geven:

```
% ifconfig
dc0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80008<VLAN_MTU,LINKSTATE>
        ether 00:a0:cc:da:da:da
        inet 192.168.1.3 netmask 0xffffff00 broadcast 192.168.1.255
        media: Ethernet autoselect (100baseTX <full-duplex>)
        status: active
dc1: flags=8802<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80008<VLAN_MTU,LINKSTATE>
        ether 00:a0:cc:da:da:db
        inet 10.0.0.1 netmask 0xffffff00 broadcast 10.0.0.255
```

```
        media: Ethernet 10baseT/UTP
        status: no carrier
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
        options=3<RXCSUM,TXCSUM>
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x4
        inet6 ::1 prefixlen 128
        nd6 options=3<PERFORMNUD,ACCEPT_RTADV>
```

In dit voorbeeld werden de volgende apparaten weergegeven:

- dc0: de eerste Ethernet-interface;
- dc1: de tweede Ethernet-interface;
- lo0: het loopback-apparaat;

FreeBSD gebruikt de naam van het stuurprogramma gevolgd door een nummer voor de volgorde waarop de kaarten gedetecteerd zijn bij het opstarten. sis2 is de derde netwerkkaart in het systeem die het stuurprogramma sis(4) gebruikt.

In het vorige voorbeeld is het apparaat dc0 volledig operationeel. Dit blijkt uit de volgende indicatoren:

1. `UP` betekent dat de kaart ingesteld is en klaar is voor gebruik;
2. De kaart heeft een Internet (`inet`) adres (in dit geval `192.168.1.3`);
3. Het heeft een geldig subnetmasker (`netmask`; `0xffffff00` is hetzelfde als `255.255.255.0`);
4. Het heeft een geldig broadcastadres (in dit geval `192.168.1.255`);
5. Het MAC-adres van de kaart (`ether`) is `00:a0:cc:da:da:da`;
6. De fysieke mediaselectie staat in autoselectiemodus (`media: Ethernet autoselect (100baseTX full-duplex)`). dc1 is ingesteld om met `10baseT/UTP`-media te werken. Meer informatie over de mogelijke mediatypes staan in de hulppagina's voor het betreffende stuurprogramma.
7. De status van de verbinding (`status`) is `active`, dat wil zeggen dat de drager is gevonden. Bij dc1 staat echter `status: no carrier`. Dit is normaal als er geen Ethernetkabel in de kaart gestoken is.

Als de uitvoer ifconfig(8) er ongeveer zoals hieronder uitziet, dan is de netwerkkaart nog niet ingesteld:

```
dc0: flags=8843<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80008<VLAN_MTU,LINKSTATE>
        ether 00:a0:cc:da:da:da
        media: Ethernet autoselect (100baseTX <full-duplex>)
        status: active
```

Om de kaart in te stellen zijn `root`-rechten nodig. De netwerkkaart kan vanaf de console worden ingesteld met ifconfig(8), maar dan moet dat na elke herstart herhaald worden. Daarom wordt het vrijwel altijd in /etc/rc.conf gezet.

In /etc/rc.conf moet voor elke netwerkkaart in een systeem een regel toegevoegd worden. In het huidige voorbeeld zou dat het volgende kunnen zijn:

```
ifconfig_dc0="inet 192.168.1.3 netmask 255.255.255.0"
ifconfig_dc1="inet 10.0.0.1 netmask 255.255.255.0 media 10baseT/UTP"
```

dc0, dc1, enzovoort, moeten vervangen worden door de correcte stuurprogramma's voor de netwerkkaarten, zo ook de IP-adressen. In de handleiding van het stuurprogramma en van ifconfig(8) staan meer details over de mogelijke opties en in rc.conf(5) staat meer informatie over /etc/rc.conf.

Als het netwerk al is ingesteld tijdens het installeren van FreeBSD staan er al enkele regels met betrekking tot de netwerkkaart(en) in /etc/rc.conf. Het is dus handig /etc/rc.conf te controleren voordat er regels toegevoegd worden.

Ook /etc/hosts moet worden gewijzigd om de namen en IP adressen van verschillende machines op het lokale netwerk, als ze er nog niet in staan. Meer informatie staat in hosts(5) en /usr/shared/examples/etc/hosts.



Als internettoegang nodig is met dit apparaat, kan het zijn dat de default gateway en de naamserver handmatig moeten worden ingesteld:

```
# echo 'defaultrouter="your_default_router"' >> /etc/rc.conf
# echo 'nameserver your_DNS_server' >> /etc/resolv.conf
```

11.8.3. Testen en problemen oplossen

Als de veranderingen in /etc/rc.conf zijn gemaakt, moet het systeem opnieuw gestarten worden (of moeten nauwkeurig alle daemons gestart of herstart worden). Veranderingen aan de interface(s) worden dan toegepast en dan kan er controleerd worden of herstarten goed werkt zonder foutmeldingen. Als alternatief kan ook het netwerk systeem herstart worden:

```
# service netif restart
```



Als er ook een default gateway ingesteld is in het /etc/rc.conf bestand, moet ook onderstaand command worden gegeven:

```
# service routing restart
```

Zodra het netwerk systeem is herstart, moeten de netwerk interfaces opnieuw getest worden.

Testen van de netwerkkaart

Om te controleren of een ethernet kaart goed geconfigureerd is, moeten er twee dingen gedaan

worden. Allereerst, ping de interface zelf, en daarna een andere machine op het LAN.

Test eerst de lokale interface:

```
% ping -c5 192.168.1.3
PING 192.168.1.3 (192.168.1.3): 56 data bytes
64 bytes from 192.168.1.3: icmp_seq=0 ttl=64 time=0.082 ms
64 bytes from 192.168.1.3: icmp_seq=1 ttl=64 time=0.074 ms
64 bytes from 192.168.1.3: icmp_seq=2 ttl=64 time=0.076 ms
64 bytes from 192.168.1.3: icmp_seq=3 ttl=64 time=0.108 ms
64 bytes from 192.168.1.3: icmp_seq=4 ttl=64 time=0.076 ms

--- 192.168.1.3 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.074/0.083/0.108/0.013 ms
```

Nu kan er een andere machine op het LAN gepinged worden:

```
% ping -c5 192.168.1.2
PING 192.168.1.2 (192.168.1.2): 56 data bytes
64 bytes from 192.168.1.2: icmp_seq=0 ttl=64 time=0.726 ms
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=0.766 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.700 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=0.747 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=0.704 ms

--- 192.168.1.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.700/0.729/0.766/0.025 ms
```

Dit kan ook worden geprobeerd met de machine naam in plaats van met `192.168.1.2` als dit geconfigureerd is in /etc/hosts.

Problemen oplossen

Het testen en zoeken van problemen is altijd een pijnpunt, welke verminderd kan worden door een aantal simpele dingen eerst te controleren. Is de netwerkkabel ingestoken? Zijn de netwerk instellingen correct opgegeven? Is de firewall goed geconfigureerd? Is de netwerkkaart ondersteund door FreeBSD? Controleer altijd de hardware notities voordat er een probleem rapport wordt verstuurd. Update naar de laatste -STABLE versie, en controleer de mailing lijsten en misschien zelfs het internet.

Als de kaart werkt, maar de prestaties zijn slecht, dan kan het de moeite waard zijn om tuning(7) door te nemen. Incorrecte netwerkinstellingen kunnen ook tot langzame verbindingen leiden.

Soms kunnen enkele `device timeouts` optreden. Met sommige kaarten is dit normaal gedrag. Maar als dit continu gebeurt of storend is, is het verstandig uit te zoeken of er geen sprake is van een hardwareconfict tussen de netwerkkaart en een ander apparaat. Ook dient nogmaals de bekabeling

gecontroleerd te worden. Misschien zit er niets anders op dan een andere netwerkkaart te gebruiken.

Het is ook mogelijk dat er `watchdog timeout` foutmeldingen optreden. Als eerste moet dan de netwerkkabel gecontroleerd worden. Veel kaarten hebben een PCI-slot nodig dat Bus Mastering ondersteunt. Sommige oudere moederborden hebben maar één PCI-slot waarmee dit kan (meestal slot 0). In de documentatie van de netwerkkaart en het moederbord is na te gaan of dit het probleem is.

`No route to host` meldingen treden op als het systeem niet in staat is om een pakket naar de eindbestemming te routeren. Dit kan gebeuren als er geen standaardroute aangegeven is of als er een kabel niet verbonden is. De uitvoer van `netstat -rn` moet gecontroleerd worden of er een geldige route is naar de bestemming. Mocht dit niet het geval zijn, dan staat er meer informatie in Netwerken voor gevorderden.

`ping: sendto: Permission denied` foutmeldingen worden vaak veroorzaakt door een verkeerd ingestelde firewall. Als de kernel `ipfw` activeert bij het opstarten zonder dat er firewallregels zijn gedefiniëerd, is het standaardbeleid om alle verkeer te weigeren, zelfs pings! In Firewalls staat meer informatie.

Er kan ook sprake zijn van onvoldoende prestaties doordat de instelling van de mediaselectie niet optimaal is. In dergelijke gevallen is het mogelijk om de mediaselectie niet als `autoselect` in te stellen, maar expliciet aan te geven wat de mediaselectie moet zijn, bijvoorbeeld 10baseT/UTP voor twisted pair. Hoewel dit voor de meeste hardware helpt, kan het zijn dat de problemen blijven. Dan moeten nogmaals de netwerkinstellingen gecontroleerd worden en geeft de tuning(7) handleiding wellicht meer informatie.

11.9. Virtuele hosts

FreeBSD wordt veel gebruikt voor virtuele sitehosting, waarbij één fysieke server er op het netwerk uitziet alsof het meerdere servers zijn. Dit kan bereikt worden door meerdere IP-adressen toe te kennen aan dezelfde interface.

Een bepaalde netwerkinterface heeft een "echt" adres en kan daarnaast een willekeurig aantal "alias"-adressen hebben. Normaliter worden dergelijke aliassen toegevoegd door aliasregels toe te voegen aan /etc/rc.conf.

Een aliasregel voor de interface fxp0 ziet er zo uit:

```
ifconfig_fxp0_alias0="inet xxx.xxx.xxx.xxx netmask xxx.xxx.xxx.xxx"
```

De aliasregels moeten beginnen met `alias0` en moete elkaar dan opvolgen (bijvoorbeeld `_alias1`,, `_alias2`, enzovoort). Het instelproces stopt als er een nummer ontbreekt.

Het is belangrijk dat aliassen het juiste netmasker hebben. Dit is eenvoudig: Een bepaalde interface moet altijd één adres hebben dat het netmasker van het netwerk correct representeert. Elk ander adres binnen dit netwerk op deze interface (alias) moet een netmasker van allemaal `1`'en (bits) hebben (getoond als `255.255.255.255` of `0xffffffff`).

Een voorbeeld. Stel de interface fxp0 is verbonden met twee netwerken, het netwerk `10.1.1.0` met masker `255.255.255.0` en het netwerk `202.0.75.16` met netmasker `255.255.255.240`. Het systeem moet ook de adressen `10.1.1.1` tot en met `10.1.1.5` en `202.0.75.17` tot en met `202.0.75.20` krijgen. Zoals hierboven vermeld, heeft alleen het eerste adres in een netwerkreeks (in dit geval `10.0.1.1` en `202.0.75.17`) een geldig netmasker. Alle overige (`10.1.1.2` tot en met `10.1.1.5` en `202.0.75.18` tot en met `202.0.75.20`) moeten ingesteld worden met het netmasker `255.255.255.255`.

De volgende regels voor /etc/rc.conf stellen een adapter in voor het bovenstaande scenario:

```
ifconfig_fxp0="inet 10.1.1.1 netmask 255.255.255.0"
ifconfig_fxp0_alias0="inet 10.1.1.2 netmask 255.255.255.255"
ifconfig_fxp0_alias1="inet 10.1.1.3 netmask 255.255.255.255"
ifconfig_fxp0_alias2="inet 10.1.1.4 netmask 255.255.255.255"
ifconfig_fxp0_alias3="inet 10.1.1.5 netmask 255.255.255.255"
ifconfig_fxp0_alias4="inet 202.0.75.17 netmask 255.255.255.240"
ifconfig_fxp0_alias5="inet 202.0.75.18 netmask 255.255.255.255"
ifconfig_fxp0_alias6="inet 202.0.75.19 netmask 255.255.255.255"
ifconfig_fxp0_alias7="inet 202.0.75.20 netmask 255.255.255.255"
```

11.10. De systeemlogger syslogd configureren

Systeemlogging is een belangrijk aspect van systeembeheer. Het wordt zowel gebruikt voor het opsporen van hardware-problemen als voor software-problemen in het systeem. Het speelt ook zeer belangrijke rol bij het controleren van de beveiliging en het reageren op incidenten. Systeem-daemons die niet in een terminal beheerd worden, loggen gewoonlijk informatie naar een systeemlogfaciliteit of een ander logbestand.

Deze sectie beschrijft hoe de FreeBSD systeemlogger, syslogd(8), te configureren en te gebruiken, en behandelt logrotatie en logbeheer met newsyslog(8). De focus ligt bij het opzetten en gebruiken van `syslogd` op een lokale machine. Meer geavanceerdere opstellingen die een aparte loghost gebruiken staan in Hosts op afstand loggen met syslogd.

11.10.1. syslogd gebruiken

In de standaardconfiguratie van FreeBSD wordt syslogd(8) gestart tijdens het opstarten. Dit wordt bepaald door de variabele `syslogd_enable` in /etc/rc.conf. Er zijn vele toepassingsargumenten die het gedrag van syslogd(8) beïnvloeden. Gebruik `syslogd_flags` in /etc/rc.conf om ze te veranderen. Bekijk syslogd(8) voor meer informatie over de argumenten, en rc.conf(5), Hoofdinstellingen en Gebruik van rc met FreeBSD voor meer informatie over /etc/rc.conf en het deelsysteem rc(8).

11.10.2. syslogd configureren

Het configuratiebestand, standaard /etc/syslog.conf, bepaalt wat syslogd(8) doet met de logregels nadat ze eenmaal ontvangen zijn. Er zijn verschillende parameters om de afhandeling van binnenkomende gebeurtenissen te beheren, waarvan de twee basaalste *faciliteit* en *niveau* zijn. De faciliteit beschrijft welk deelsysteem het bericht genereerde, zoals de kernel of een daemon, het niveau beschrijft de ernst van de opgetreden gebeurtenis. Dit maakt het mogelijk om het bericht

naar verschillende logbestanden te loggen, of het weg te gooien, afhankelijk van de faciliteit en het niveau. Het is ook mogelijk om actie te nemen afhankelijk van de toepassing dat het bericht verstuurde, en in het geval van loggen op afstand, ook de hostnaam van de machine dat het logbericht genereerde.

Het configureren van syslogd(8) is vrij rechttoe-rechtaan. Het configuratiebestand bevat één regel per actie, de syntaxis van elke regel is een selecteerderveld gevolgd door een actieveld. De syntaxis van het selecteerderveld is *faciliteit.niveau* dat overeenkomt met logberichten van *faciliteit* op niveau *niveau* of hoger. Het is ook mogelijk om een optionele vergelijkingsvlag voor het niveau toe te voegen om meer precies te specificeren wat er gelogd wordt. Er kunnen meerdere selecteerdervelden worden gebruikt voor dezelfde actie, ze worden gescheiden door een puntkomma (;). Het gebruik van * zal met alles overeenkomen. Het actieveld bepaalt waar het logbericht naar toe wordt gezonden, zoals een bestand of een loghost op afstand. Als voorbeeld is hier de standaard syslog.conf van FreeBSD:

```
# $FreeBSD$
#
#       Spaces ARE valid field separators in this file. However,
#       other *nix-like systems still insist on using tabs as field
#       separators. If you are sharing this file between systems, you
#       may want to use only tabs as field separators here.
#       Consult the man:syslog.conf[5] manpage.
*.err;kern.warning;auth.notice;mail.crit                /dev/console ①
*.notice;authpriv.none;kern.debug;lpr.info;mail.crit;news.err   /var/log/messages
security.*                                      /var/log/security
auth.info;authpriv.info                         /var/log/auth.log
mail.info                                       /var/log/maillog ②
lpr.info                                        /var/log/lpd-errs
ftp.info                                        /var/log/xferlog
cron.*                                          /var/log/cron
*.=debug                                        /var/log/debug.log ③
*.emerg                                         *
# uncomment this to log all writes to /dev/console to /var/log/console.log
#console.info                                   /var/log/console.log
# uncomment this to enable logging of all log messages to /var/log/all.log
# touch /var/log/all.log and chmod it to mode 600 before it will work
#*.*                                            /var/log/all.log
# uncomment this to enable logging to a remote loghost named loghost
#*.*                                            @loghost
# uncomment these if you're running inn
# news.crit                                     /var/log/news/news.crit
# news.err                                      /var/log/news/news.err
# news.notice                                   /var/log/news/news.notice
!ppp ④
*.*                                             /var/log/ppp.log
!*
```

① Komt overeen met alle berichten met een `err` of hoger, alsook met `kern.warning`, `auth.notice` en `mail.crit`, en stuur deze logberichten naar de console (/dev/console).

② Komt overeen met alle berichten van de faciliteit `mail` op niveau `info` of hoger, en logt de berichten in /var/log/maillog.

③ Deze regel gebruikt een vergelijkingsvlag, `=` om alleen met de berichten op niveau `debug` overeen te komen en ze op te slaan in /var/log/debug.log.

④ Hier volgt een gebruiksvoorbeeld van een *programmaspecificatie*. Dit zorgt ervoor dat de regels alleen geldig zijn voor het programma in de programmaspecificatie. In dit geval zorgen deze en de volgende regel ervoor dat alle berichten van `ppp`, maar niet van andere programma's, in /var/log/ppp.log terechtkomen.

Dit voorbeeld toont dat er vele niveaus en deelsystemen zijn. De niveaus zijn, in volgorde van meest naar minst kritisch: `emerg`, `alert`, `crit`, `err`, `warning`, `notice`, `info` en `debug`.

De faciliteiten zijn, in geen specifieke volgorde: `auth`, `authpriv`, `console`, `cron`, `daemon`, `ftp`, `kern`, `lpr`, `mail`, `mark`, `news`, `security`, `syslog`, `user`, `uucp` en `local0` tot en met `local7`. Let erop dat andere besturingssystemen andere faciliteiten kunnen hebben.

Met deze kennis is het eenvoudig om een nieuwe regel aan /etc/syslog.conf toe te voegen om alles van de verschillende daemons op niveau `notice` en hoger naar /var/log/daemon.log te loggen:

```
daemon.notice                                   /var/log/daemon.log
```

Bekijk syslog(3) en syslogd(8) voor meer informatie over de verschillende niveaus en faciliteiten. Zie syslog.conf(5) en Hosts op afstand loggen met syslogd voor meer informatie over syslog.conf, de syntaxis, en geavanceerdere gebruiksvoorbeelden.

11.10.3. Logbeheer en -rotatie met newsyslog

Logbestanden hebben de neiging om snel te groeien en gestadig opgehoopt te raken. Dit leidt tot bestanden die vol zitten met minder direct bruikbare informatie en de harde schijf volmaken. Logbeheer kan gebruikt worden om dit te beheersen. In FreeBSD wordt newsyslog(8) gebruikt om logbestanden te beheren. Dit programma wordt gebruikt om periodiek logbestanden te roteren en te comprimeren en om optioneel ontbrekende logbestanden aan te maken en programma's te signaleren dat logbestanden zijn verplaatst. De logbestanden hoeven niet per sé van syslog afkomstig te zijn; newsyslog(8) werkt met elke log van elk programma. Het is belangrijk om op te merken dat `newsyslog` normaliter vanuit cron(8) wordt gedraaid en niet een systeem-daemon is. In de standaardconfiguratie wordt het elk uur gedraaid.

newsyslog configureren

Om te weten wat het moet doen leest newsyslog(8) zijn configuratiebestand, standaard is dit /etc/newsyslog.conf. Dit configuratiebestand bevat één regel voor elk bestand dat newsyslog(8) beheert. Elke regel noemt de eigenaar van het bestand, rechten, en wanneer dat bestand te roteren, alsook optionele vlaggen die de logrotatie beïnvloeden (zoals compressie) en naar welke programma's een signaal te sturen wanner de log is geroteerd. Als voorbeeld is hier de standaard configuratie in FreeBSD:

```
# configuration file for newsyslog
```

```
# $FreeBSD$
#
# Entries which do not specify the '/pid_file' field will cause the
# syslogd process to be signalled when that log file is rotated.  This
# action is only appropriate for log files which are written to by the
# syslogd process (ie, files listed in /etc/syslog.conf).  If there
# is no process which needs to be signalled when a given log file is
# rotated, then the entry for that file should include the 'N' flag.
#
# The 'flags' field is one or more of the letters: BCDGJNUXZ or a '-'.
#
# Note: some sites will want to select more restrictive protections than the
# defaults.  In particular, it may be desirable to switch many of the 644
# entries to 640 or 600.  For example, some sites will consider the
# contents of maillog, messages, and lpd-errs to be confidential.  In the
# future, these defaults may change to more conservative ones.
#
# logfilename          [owner:group]    mode count size when  flags [/pid_file]
[sig_num]
/var/log/all.log                        600  7     *    @T00  J
/var/log/amd.log                        644  7     100  *     J
/var/log/auth.log                       600  7     100  @0101T JC
/var/log/console.log                    600  5     100  *     J
/var/log/cron                           600  3     100  *     JC
/var/log/daily.log                      640  7     *    @T00  JN
/var/log/debug.log                      600  7     100  *     JC
/var/log/init.log                       644  3     100  *     J
/var/log/kerberos.log                   600  7     100  *     J
/var/log/lpd-errs                       644  7     100  *     JC
/var/log/maillog                        640  7     *    @T00  JC
/var/log/messages                       644  5     100  @0101T JC
/var/log/monthly.log                    640  12    *    $M1D0 JN
/var/log/pflog                          600  3     100  *     JB
/var/run/pflogd.pid
/var/log/ppp.log        root:network    640  3     100  *     JC
/var/log/security                       600  10    100  *     JC
/var/log/sendmail.st                    640  10    *    168   B
/var/log/utx.log                        644  3     *    @01T05 B
/var/log/weekly.log                     640  5     1    $W6D0 JN
/var/log/xferlog                        600  7     100  *     JC
```

Elke regel begint met de naam van het bestand dat geroteerd moet worden, optioneel gevolgd door een eigenaar en groep voor zowel de geroteerde als nieuw aangemaakte bestanden. Het volgende veld, `mode` is de modus van de bestanden en `count` geeft aan hoeveel geroteerde logbestanden bewaard moeten worden. De velden `size` en `when` vertellen `newyslog` wanneer het bestand geroteerd moet worden. Een logbestand wordt geroteerd wanneer òfwel de grootte meer is dan de waarde in het veld `size`, òfwel wanneer de tijd in het veld `when` is verstreken. `*` geeft aan dat dit veld genegeerd wordt. Het veld *flags* geeft newsyslog(8) verdere instructies, zoals hoe het geroteerde bestand te comprimeren of om het logbestand aan te maken als het ontbreekt. De laatste twee velden zijn optioneel en specificeren het PID-bestand van een proces en een naar dat proces te verzenden

signaalnummer wanneer het bestand wordt geroteerd. Raadpleeg newsyslog.conf(5) voor meer informatie over alle velden, geldige vlaggen en hoe de rotatietijd te specificeren. Herinner dat `newsyslog` wordt gedraaid vanuit `cron` en niet vaker bestanden kan roteren dan dat het gedraaid wordt vanuit cron(8).

11.11. Instellingenbestanden

11.11.1. /etc layout

Instellingengegevens wordt in een aantal mappen bewaard. Daar zijn onder andere:

/etc	Generieke systeeminstellingenbestanden, specifiek voor het systeem.
/etc/defaults	De standaardversies van systeeminstellingenbestanden.
/etc/mail	Extra sendmail(8) instellingenbestanden of instellingenbestanden voor andere MTAs.
/etc/ppp	Instellingen voor zowel gebruiker- als kernel-ppp programma's.
/etc/namedb	Standaardlocatie voor named(8) gegevens. Normaal gesproken bevinden zich hier named.conf en zonebestanden.
/usr/local/etc	Instellingenbestanden voor geïnstalleerde software. Kan submappen hebben waarin bij elkaar horende instellingengegevens van een applicatie gegroepeerd zijn.
/usr/local/etc/rc.d	Start- en stopscripts voor geïnstalleerde diensten.
/var/db	Automatisch gemaakte systeemspecifieke databasebestanden, zoals de pakketdatabase, de locate(1) database, enzovoort.

11.11.2. Hostnamen

/etc/resolv.conf

In /etc/resolv.conf wordt voorgeschreven op welke wijze FreeBSD het Domain Name System (DNS) moet gebruiken.

De meest voorkomende termen in resolv.conf zijn:

`nameserver`	Het IP-adres van een naamserver die ondervraagd moet worden voor naam/IP-conversie. De servers worden in volgorde geprobeerd en het maximale aantal is drie.

`search`	Zoeklijst voor het opzoeken van hostnamen. Meestal wordt deze bepaald door het domein waarop de lokale hostnaam zich bevindt.
`domain`	De lokale domeinnaam.

Een typisch resolv.conf bestand:

```
search example.com
nameserver 147.11.1.11
nameserver 147.11.100.30
```



`search` en `domain` dienen niet tegelijk gebruikt te worden.

Als DHCP wordt gebruikt: dhclient(8) overschrijft meestal resolv.conf met informatie ontvangen van de DHCP-server.

/etc/hosts

/etc/hosts is een eenvoudige tekstdatabase uit de dagen van het oude Internet. Het werkt samen met DNS en NIS om namen en IP adressen over en weer te vertalen. Lokale computers, verbonden via een LAN, kunnen hier het beste in opgenomen worden om zo op simpele wijze naam/IP conversie voor een LAN te hebben, zonder noodzaak voor een named(8) server. Ook kunnen naamaliassen toegekend worden (vergelijkbaar met CNAMES bij DNS). Op soortgelijke wijze kan /etc/hosts gebruikt worden als een (zeer beperkte) lokale DNS cache.

```
# $FreeBSD$
#
# Host Database
# Dit bestand hoort de adressen en aliassen te bevatten
# voor de lokale hosts die dit bestand gebruiken.
# Bij gebruik van DNS of NIS hoeft dit bestand helemaal niet gebruikt
# te worden.  Zie /etc/nsswitch.conf voor de volgorde van resolutie.
#
#
::1                     localhost localhost.my.domain myname.my.domain
127.0.0.1               localhost localhost.my.domain myname.my.domain

#
# Verzonnen netwerk.
#10.0.0.2               myname.my.domain myname
#10.0.0.3               myfriend.my.domain myfriend
#
# Volgens RFC 1918 mogen de volgende IP netwerken gebruikt worden
# als private netwerken die niet met Internet verbonden zijn:
#
#        10.0.0.0        -   10.255.255.255
#        172.16.0.0      -   172.31.255.255
```

```
#        192.168.0.0     -   192.168.255.255
#
# Als er toch verbinding moet zijn met Internet, zijn echte
# officieel toegewezen nummers nodig.  Probeer ECHT GEEN eigen
# netwerknummers te verzinnen, maar vraag ze op bij de provider
# (als die er is) of bij de Internet Registry (ftp naar
# rs.internic.net, map `/templates').
#
```

/etc/hosts heeft als formaat:

```
[Internet address] [official hostname] [alias1] [alias2] ...
```

Bijvoorbeeld:

```
10.0.0.1 myRealHostname.example.com myRealHostname foobar1 foobar2
```

In hosts(5) staat meer informatie.

11.11.3. sysctl.conf

sysctl.conf lijkt veel op rc.conf. Waardetoekenning heeft weer de vorm `variable=value`. De ingestelde sysctl(8)-waarden worden doorgevoerd op het moment dat het systeem naar multi-user modus gaat. Niet alle variabelen kunnen in deze modus gewijzigd worden.

Om te voorkomen dat er logregels geplaatst worden als processen crashen en om te voorkomen dat andere gebruikers kunnen zien welke processen er gestart zijn door een andere gebruiker, kunnen de volgende instellingen worden gezet in sysctl.conf:

```
#Log exits met fatale signalen niet (bv. sig 11)
kern.logsigexit=0

# Voorkom dat gebruikers informatie zien over processen die
# worden gedraaid onder een ander UID.
security.bsd.see_other_uids=0
```

11.12. Optimaliseren met sysctl

sysctl(8) is een interface waarmee veranderingen gemaakt kunnen worden aan een draaiend FreeBSD-systeem. Er zijn onder meer vele geavanceerde opties voor de TCP/IP-stack en het virtuele geheugensysteem, waarmee een ervaren systeembeheerder de systeemprestaties drastisch kan verbeteren. Met sysctl(8) kunnen meer dan vijfhonderd systeemvariabelen opgevraagd en ingesteld worden.

In essentie heeft sysctl(8) twee functies: het lezen en wijzigen van systeeminstellingen.

Om alle leesbare variabelen te tonen:

```
% sysctl -a
```

Om een bepaalde variabele op te vragen, bijvoorbeeld `kern.maxproc`:

```
% sysctl kern.maxproc
kern.maxproc: 1044
```

Om een bepaalde variabele toe te kennen (te wijzigen), is de syntaxis *variable=value*:

```
# sysctl kern.maxfiles=5000
kern.maxfiles: 2088 -> 5000
```

Waarden van sysctl-variabelen zijn doorgaans strings (tekst), getallen of booleans (`1` als waar, `0` als onwaar).

Om automatisch variabelen in te stellen als de machine start, kunnen ze toegevoegd worden aan /etc/sysctl.conf. Meer informatie staat in sysctl.conf(5) en sysctl.conf.

11.12.1. sysctl(8) alleen-lezen

In sommige gevallen is het wenselijk om sysctl(8)-waarden die alleen-lezen zijn toch te wijzigen. Hoewel dit soms onontkoombaar is, kan het alleen bij een (her)start gedaan worden.

Op sommige laptops is bijvoorbeeld het apparaat cardbus(4) niet in staat om geheugenregio's af te tasten, met als gevolg foutmeldingen als:

```
cbb0: Could not map register memory
device_probe_and_attach: cbb0 attach returned 12
```

In dergelijke gevallen moeten er meestal enkele sysctl(8)-instellingen gewijzigd worden die alleen-lezen zijn en een standaardwaarde hebben. Dit kan bereikt worden door sysctl(8) "OIDs" in de lokale /boot/loader.conf te zetten. Standaardinstellingen staan in /boot/defaults/loader.conf.

Om het bovenstaande probleem op te lossen moet in /boot/loader.conf`hw.pci.allow_unsupported_io_range=1` ingesteld worden. Dan werkt cardbus(4) wel goed.

11.13. Harde schijven optimaliseren

11.13.1. Sysctl-variabelen

vfs.vmiodirenable

De sysctl-variabele vfs.vmiodirenable kan de waarde 0 (uit) of 1 (aan) hebben. De standaardwaarde is 1. Deze variabele bepaalt hoe mappen door het systeem in een cache bewaard worden. De meeste mappen zijn klein en gebruiken slechts een klein fragment (typisch 1 K) in het bestandssysteem en nog minder (typisch 512 bytes) in de buffercache. Als deze variabele uit staat (op 0) bewaart de buffercache slechts een bepaald aantal mappen in de cache, ook al is er een overvloed aan geheugen beschikbaar. Wanneer deze aan staat (op 1), wordt de VM paginacache gebruikt, waardoor voor het cachen van mappen al het geheugen kan worden gebruikt. Het is echter wel zo dat het minimale in-core geheugen dat gebruikt wordt om een map te cachen in dat geval de fysieke paginagrootte is (typisch 4 K) in plaats van 512 bytes. Het is aan te raden deze optie aan te laten staan als gebruik gemaakt wordt van diensten die met grote aantallen bestanden werken, zoals webcaches, grote mailsystemen en newsservers. Als deze optie aan blijft staan, verlaagt die de prestaties niet, ook al kost het meer geheugen. Door experimenteren is dit voor een systeem na te gaan.

vfs.write_behind

De sysctl-variabele vfs.write_behind staat standaard aan (1). Dit betekent dat het bestandssysteem gegevens naar het medium gaat schrijven op het moment dat er een volledig cluster aan gegevens verzameld is. Dit is meestal het geval bij het schrijven van grote sequentiële bestanden. Het idee is om te voorkomen dat de buffercache verzadigd raakt met vuile buffers zonder dat dit bijdraagt aan de I/O-prestaties. Dit kan echter processen ophouden en onder sommige omstandigheden is het wellicht beter deze sysctl uit te zetten.

vfs.hirunningspace

De sysctl-variabele vfs.hirunningspace bepaalt hoeveel nog te schrijven gegevens er in het complete systeem op elk moment in de wachtrij naar schijfcontrollers mag staan. De standaardwaarde is meestal voldoende, maar op machines met veel schijven, is het beter deze te verhogen naar vier of vijf *megabyte*. Het instellen van een te hoge waarde (groter dan de schrijfdrempel van de buffercache) kan leiden tot zeer slechte prestaties bij clustering. Stel deze waarde niet arbitrair hoog in! Hogere schrijfwaarden kunnen vertraging veroorzaken in het lezen, als dit tegelijk plaatsvindt.

Er zijn verscheidene andere sysctl's voor buffercache en VM-pagecache. Het wordt afgeraden deze te wijzigen. Het VM-systeem is zeer goed in staat zichzelf automatisch te optimaliseren.

vm.swap_idle_enabled

De sysctl-variabele vm.swap_idle_enabled is nuttig in grote meergebruikersystemen met veel gebruikers die af- en aanmelden en veel onbenutte processen. Dergelijke systemen hebben de neiging om voortdurend de vrije geheugenreserves onder druk te zetten. Het is mogelijk om de prioriteit van geheugenpagina's die verband houden met onbenutte processen sneller te laten dalen dan met het normale pageout-algoritme, door deze sysctl aan te zetten en via vm.swap_idle_threshold1 en vm.swap_idle_threshold2 de swapout hysterese (in seconden onbenut) af te stemmen. Deze optie dient alleen gebruikt te worden als ze echt nodig is, want de andere kant van de medaille is dat dit eerder pre-page geheugen inhoudt in plaats van later, waardoor het meer wisselbestand- en schijfbandbreedte kost. In een klein systeem heeft deze optie een voorspelbaar effect, maar in grote systemen waar al sprake is van een matige paging kan deze optie het mogelijk

maken voor het VM-systeem om hele processen gemakkelijk in en uit het geheugen te halen.

hw.ata.wc

Ten tijde van FreeBSD 4.3 is er geflirt met het uitzetten van IDE-schrijfcaching. Hierdoor neemt de bandbraadte naar IDE-schijven af, maar het werd als noodzakelijk beschouwd vanwege ernstige problemen met gegevensinconsistentie die door harde schijfproducenten geëintroduceerd waren. Het probleem is dat IDE-schijven niet de waarheid vertellen over wanneer een schrijfactie klaar is. Door IDE-schrijfcaching wordt data niet alleen ongeordend geschreven, maar soms kan zelfs het schrijven van sommige blokken voortdurend uitgesteld worden als er sprake is van een hoge schijfbelasting. Een crash of stroomstoring kan dan ernstige corruptie aan het bestandssysteem veroorzaken. Daarom werd de standaardinstelling van FreeBSD voor alle zekerheid gewijzigd. Helaas was het resultaat een groot verlies aan prestaties en na die uitgave is de standaardwaarde weer terug veranderd. Met de sysctl-variabele `hw.ata.wc` kan gecontroleerd worden of schrijfcaching aan of uit staat. Als schrijfcaching uit staat, kan het die weer aangezet worden door `hw.ata.wc` op 1 te zetten. Aangezien dit een kernelvariabele is, moet deze ingesteld worden vanuit de bootloader tijdens het opstarten. Nadat de kernel eenmaal opgestart is, heeft het wijzigen van deze sysctl geen effect.

Meer informatie staat in ata(4).

SCSI_DELAY (kern.cam.scsi_delay)

De kernelinstelling `SCSI_DELAY` kan gebruikt worden om de opstarttijd te versnellen. De standaardwaarde is nogal hoog en kan `15` seconden vertraging veroorzaken. Met modernere SCSI-systemen is `5` seconden al voldoende (zeker met moderne schijven). De `kern.cam.scsi_delay` opstart variabele moet hier gebruikt worden. De variabele en kernelconfiguratie-optie accepteren waarden uitgedrukt in *milliseconden* en *niet* in *seconden*.

11.13.2. Softupdates

tunefs(8) kan gebruikt worden om een bestandsysteem nauwkeurig af te stellen. Het heeft veel opties, maar nu wordt alleen het aan- en uitzetten van softupdates besproken. Dat gaat als volgt:

```
# tunefs -n enable /filesystem
# tunefs -n disable /filesystem
```

Een bestandssysteem kan niet met tunefs(8) gewijzigd worden als het aangekoppeld is. Softupdates aanzetten wordt dus in het algemeen gedaan vanuit enkelegebruikermodus, voordat partities aangekoppeld zijn.

Softupdates zorgen voor een drastische verbetering van de prestaties met betrekking tot metagegevens, met name het aanmaken en verwijderen van bestanden, door gebruik van een geheugencache. Het wordt dan ook aangeraden om op alle bestandssystemen softupdates te gebruiken. Er zijn twee nadelen aan softupdates: softupdates garanderen een consistent bestandssysteem in geval van een crash, maar het kan makkelijk enkele seconden (zelfs een minuut) achter liggen met het daadwerkelijk bijwerken op de fysieke harde schijf. Als een systeem crasht gaat wellicht meer werk verloren dan anders het geval zou zijn. Daarnaast vertragen

softupdates het vrijgeven van bestandssysteemblokken. Als een bestandssysteem (zoals de rootpartitie) bijna vol is, dan kan het verrichten van een grote update, zoals `make installworld`, ertoe leiden dat het bestandssysteem ruimtegebrek krijgt en dat daardoor de operatie mislukt.

Meer over softupdates

Er zijn traditioneel twee methodes om de metagegevens van een bestandssysteem terug naar de schijf te schrijven. Het bijwerken van metagegevens houdt het bijwerken van van niet-inhoudelijke gegevens zoals inodes of mappen in.

Historisch gezien was het gebruikelijk om updates aan metagegevens synchroon weg te schrijven. Als een map bijvoorbeeld gewijzigd was, wachtte het systeem totdat de verandering daadwerkelijk naar de schijf geschreven was. De gegevensbuffers (de inhoud van een bestand) werden doorgeschoven naar de buffercache en op een later moment asynchroon op de schijf opgeslagen. Het voordeel van deze benadering is dat ze altijd veilig is. Als het systeem faalt tijdens het bijwerken, zijn de metagegevens nog altijd consistent. Een bestand kan volledig gecreëerd zijn of helemaal niet. Als de gegevensblokken van een bestand nog niet van de buffercache naar de schijf geschreven zijn ten tijde van de crash, is fsck(8) in staat om dit te herkennen en het bestandssysteem te repareren door de lengte van het bestand nul te maken. Deze implementatie is ook helder en eenvoudig. Het nadeel is echter dat het wijzigen van metagegevens een traag proces is. Een `rm -r` benadert bijvoorbeeld alle bestanden in een map sequentiëel, maar elke mapverandering (verwijderen van een bestand) wordt synchroon naar de schijf geschreven. Dit omvat ook het bijwerken van de map zelf, van de inodetabel en mogelijk ook van indirecte blokken die voor het bestand in kwestie zijn gealloceerd. Gelijksoortige processen spelen zich af bij een commando als `tar -x`, waarbij een grote bestandshiëearchie wordt uitgepakt.

De tweede mogelijkheid is om het bijwerken van metagegevens asynchroon weg te schrijven. Dit is standaard in Linux®/ext2fs en als een *BSD UFS-bestandssysteem met `mount -o async` aangekoppeld is, is de werking hetzelfde. Alle bijwerkingen aan metagegevens worden eenvoudigweg doorgegeven aan de buffercache en vermengd met inhoudelijke updates van de bestandsgegevens. Het voordeel is een grote winst aan snelheid, omdat er niet telkens gewacht hoeft te worden op het bijwerken van metagegevens tot deze daadwerkelijk naar de schijf geschreven zijn. De implementatie is ook in dit geval helder en eenvoudig. Het grote nadeel is uiteraard dat er geen enkele garantie is voor de consistentie van het bestandssysteem. Als het systeem faalt tijdens een operatie waarbij veel metagegevens worden bijgewerkt (bijvoorbeeld door een stroomstoring of iemand drukt op de resetknop), blijft het bestandssysteem in een onvoorspelbare toestand achter. Er is geen mogelijkheid om de toestand van het bestandssysteem te onderzoeken als het systeem weer opstart, want de gegevensblokken van een bestand kunnen al weggeschreven zijn geweest terwijl het wegschrijven van bijwerkingen aan de inodetabel of de bijhorende map nog niet plaats heeft gevonden. Het is zelfs onmogelijk om een `fsck` te implementeren die de overgebleven chaos kan opruimen: de benodigde informatie is gewoon niet volledig aanwezig op de schijf. Als een bestandssysteem op deze manier onherstelbaar beschadigd is, is de enige optie newfs(8) te gebruiken en vervolgens te herstellen van een back-up.

De gebruikelijke oplossing voor dit probleem is het implementeren van *dirty region logging*, ook wel *journaling* genoemd, hoewel deze term niet consistent gebruikt wordt en soms ook wordt gebruikt voor andere vormen van transactielogging. Het bijwerken van metagegevens wordt nog steeds synchroon geschreven, maar slechts naar een klein gebied van de schijf. Later worden ze dan naar de juiste locatie verplaatst. Omdat het loggebied klein is, hoeven de koppen van de schijf

zelfs tijdens schrijfintensieve operaties nog maar over een kleine fysieke afstand te bewegen en door deze snellere respons zijn dit soort operaties sneller dan op de traditionele manier. De extra complexiteit van de implementatie is nogal beperkt, dus het risico van introductie van extra bugs valt mee. Een nadeel is dat alle metagegevens tweemaal geschreven worden (eerst naar het loggebied en later nog eens naar de definitieve locatie). Dus bij normaal gebruik kan er sprake zijn van wat men wel noemt een "performance pessimization". Anderzijds kunnen in geval van een crash alle nog uitstaande metagegevensoperaties snel worden teruggedraaid of vanuit het loggebied alsnog worden afgemaakt wanneer de machine weer opstart. Het bestandssysteem start dan snel op.

Kirk McKusick, de vader van het Berkeley FFS, loste dit probleem op met softupdates, wat betekent dat alle uitstaande acties voor het bijwerken van metagegevens in het geheugen bewaard worden en dan geordend naar de schijf geschreven worden. Dit heeft het gevolg dat in geval van intensieve operaties met betrekking tot metagegevens, latere bijwerkingen aan een item eerdere bewerkingen opvangen ("catch") als deze nog in het geheugen zitten en nog niet weggeschreven waren. Dus alle operaties, op bijvoorbeeld een map, worden in het algemeen eerst in het geheugen uitgevoerd voordat er wordt bijgewerkt naar schijf. De gegevensblokken worden geordend conform hun positie, zodat ze nooit weggeschreven worden voordat hun metagegevens geschreven zijn. Als het systeem een crash ondervindt, veroorzaakt dat impliciet het terugdraaien van uitstaande operaties ("log rewind"): alle operaties die nog niet weggeschreven waren lijken nooit gebeurd te zijn. Zo wordt een consistent bestandssysteem in stand gehouden dat eruit ziet alsof het 30 tot 60 seconden eerder was. Het gebruikte algoritme garandeert dat alle bronnen die in gebruik zijn als zodanig gemarkeerd worden in hun daarvoor geschikte bitmaps: blokken en inodes. Na een crash is de enige allocatiefout die kan optreden dat bronnen gemarkeerd kunnen zijn als in gebruik ("used"), terwijl ze feitelijk alweer beschikbaar ("free") zijn. fsck(8) herkent deze situatie en stelt dergelijke vrij te maken bronnen opnieuw beschikbaar. Het is volkomen veilig om na een crash te negeren dat het bestandssysteem niet schoon is en het tot aankoppelen te dwingen met `mount -f`. Om niet langer gebruikte bronnen vrij te maken moet later fsck(8) uitgevoerd worden. Dit is dan ook het idee achter *background fsck*: op het moment dat het systeem aan het opstarten is, wordt er alleen een *snapshot* van het systeem bewaard. `fsck` kan later uitgevoerd worden. Alle bestandssystemen kunnen "dirty" aangekoppeld worden en het systeem kan gewoon verder opstarten naar meergebruikermodus. Vervolgens zijn er `fsck`s gepland die in de achtergrond draaien voor elk bestandssysteem dat niet schoon is en waarmee bezette bronnen vrijgegeven worden. Bestandssystemen die geen gebruik maken van softupdates moeten echter nog steeds gebruik maken van de normale `fsck` in de voorgrond.

Het voordeel van softupdates is dat operaties op metagegevens bijna net zo snel zijn als asynchrone updates (dat wil zeggen sneller dan met *logging*, waarbij de metagegevens keer op keer geschreven worden). Nadelen zijn de complexiteit van de code (wat een groter risico op bugs impliceert in een gebied dat bijzonder gevoelig is voor verlies van gebruikersgegevens) en een groter geheugenverbruik. Tevens moet de gebruiker wennen aan enkele eigenaardigheden. Na een crash lijkt de toestand van het bestandssysteem wat "ouder". In situaties waar de standaard synchrone benadering een aantal lege bestanden zou hebben achtergelaten na `fsck`, is het met softupdates juist zo dat dergelijke bestanden er helemaal niet zijn, omdat de metagegevens of de bestandsinhoud nooit naar de schijf zijn geschreven. Schijfruimte wordt pas vrijgegeven als de bijwerkingen aan metagegevens en inhoudelijke bestandsgegevens weggeschreven zijn, wat mogelijk pas enige tijd na het uitvoeren van `rm` plaatsvindt. Dit kan problemen veroorzaken als er grote hoeveelheden gegevens naar een bestandssysteem geschreven worden dat onvoldoende vrije

ruimte heeft om alle bestanden twee keer te kunnen bevatten (bijvoorbeeld in /tmp).

11.14. Fijnafstemming van kernellimieten

11.14.1. Bestandsproceslimieten

`kern.maxfiles`

`kern.maxfiles` kan worden verhoogd of verlaagd, afhankelijk van de systeembehoeften. Deze variabele geeft het maximale aantal bestandsdescriptors op een systeem. Als de bestandsdescriptortabel vol is, toont de systeembuffer meerdere malen `file: table is full`, hetgeen achteraf te zien is met `dmesg`.

Elk geopend bestand, socket of fifo heeft een bestandsdescriptor. Een grote produktieserver kan makkelijk enige duizenden bestandsdescriptors nodig hebben, afhankelijk van het soort en aantal diensten die tegelijk draaien.

In oudere versies van FreeBSD werd de standaard waarde van `kern.maxfiles` afgeleid van de optie `maxusers` in het kernelconfiguratiebestand. `kern.maxfiles` groeit evenredig met de waarde van `maxusers`. Als een aangepaste kernel wordt gebouwd, is het een goed idee om deze kerneloptie in te stellen afhankelijk van het gebruikt van een systeem (maar niet te laag). Hoewel een produktieserver misschien niet 256 gelijktijdige gebruikers heeft, kunnen de benodigde systeembronnen het beste vergeleken worden met een grootschalige webserver.

De optie `maxusers` stelt de grootte van een aantal belangrijke systeemtabellen in. Dit aantal moet ruwweg gelijk zijn aan het aantal gebruikers dat verwacht wordt gelijktijdig van de machine gebruik te maken.

Vanaf FreeBSD 4.5 wordt `kern.maxusers` automatisch ingesteld tijdens het opstarten gebaseerd op de hoeveelheid beschikbare geheugen in het systeem en kan worden vastgesteld tijdens het draaien door te kijken naar de alleen-lezen sysctl `kern.maxusers`. Sommige configuraties hebben grotere of kleinere waarden nodig van `kern.maxusers`, deze kunnen worden gezet als een opstartvariabele. Waardes van 64, 128 en 256 zijn daarin niet ongewoon. We raden aan om niet boven de 256 te gaan tenzij er heel veel bestandsdescriptors benodigd zijn; veel van de aanpasbaare waarden die standaard worden bepaald door `kern.maxusers` kunnen individueel worden overschreven tijdens het opstarten en/of tijdens het draaien van het systeem in /boot/loader.conf (zie de handleiding loader.conf(5) of /boot/defaults/loader.conf voor een paar aanwijzingen) of zoals elders beschreven in dit document.

Voor oudere versies stelt het systeem deze waarde zelf in als deze uitdrukkelijk op `0` is gezet. Als het gewenst is om deze waarde zelf aan te geven, wordt aangeraden om `maxusers` minstens op 4 te zetten, met name als het X Window systeem in gebruik is of als er software gecompileerd wordt. De reden hiervoor is dat de belangrijkste tabel die door `maxusers` ingesteld wordt, het maximum aantal processen is, dat ingesteld wordt op `20 + 16 * maxusers`, dus als `maxusers` op 1 ingesteld wordt, zijn er maar 36 gelijktijdige processen mogelijk, inclusief de ongeveer achttien processen die door het systeem tijdens het opstarten start en de ongeveer vijftien processen die waarschijnlijk aangemaakt worden door het opstarten van het X Window systeem. Zelfs een eenvoudige taak als het afbeelden van een hulppagina start negen processen op om de pagina te filteren, te decomprimeren en af te beelden. Als `maxusers` op 64 ingesteld wordt, zijn er 1044 gelijktijdige processen mogelijk, wat

genoeg moet zijn voor bijna alle soorten gebruik. Als echter de gevreesde fout verschijnt als er geprobeerd wordt om een programma op te starten of als er een server gedraaid wordt met een groot aantal gelijktijdige gebruikers, zoals `ftp.FreeBSD.org`, kan het getal altijd verhoogd worden en kan de kernel opnieuw gebouwd worden.



> `maxusers` stelt *geen* grens aan het aantal gebruikers dat zich op de machine kan aanmelden. Het stelt gewoon verschillende tabelgroottes in op redelijke waardes, uitgaande van het maximum aantal gebruikers dat waarschijnlijk de machine gebruikt en van het aantal processen dat elk van deze gebruikers zal draaien. Een sleutelwoord dat *wel* het aantal gelijktijdige aanmeldingen op afstand en X-terminalvensters begrensd is `pseudo-device pty 16`. In FreeBSD 5.X kan dit getal genegeerd worden omdat daar het stuurprogramma pty(4) "auto-cloning" is. Er kan eenvoudig gebruik worden gemaakt van de regel `device pty` in het instellingenbestand.

`kern.ipc.somaxconn`

De sysctl-variabele `kern.ipc.somaxconn` beparkt de grootte van de luisterwachtrij voor het accepteren van nieuwe TCP-verbindingen. De standaardwaarde van `128` is meestal te laag voor robuuste behandeling van nieuwe verbindingen in een zwaarbeladen webserveromgeving. Voor zulke omgevingen wordt aangeraden deze waarde te verhogen tot `1024` of hoger. De dienstdaemon beperkt misschien zelf de luisterwachtrij (bijvoorbeeld sendmail(8) of Apache), maar heeft vaak een mogelijkheid in een configuratiebestand de wachtrijgrootte aan te passen. Grote luisterwachtrijen zijn ook beter in het ontwijken van Ontzegging van Dienst () aanvallen.

11.14.2. Netwerkbeperkingen

De kerneloptie `NMBCLUSTERS` bepaalt het aantal netwerk-Mbufs dat beschikbaar is voor een systeem. Een veel bezochte server met een laag aantal Mbufs beperkt de mogelijkheden van FreeBSD. Elk cluster staat voor ongeveer 2 K geheugen, dus een waarde van 1024 stelt 2 megabyte aan kernelgeheugen voor, dat is gereserveerd voor netwerkbuffers. Een simpele berekening geeft aan hoeveel er nodig is. Stel dat een webserver met een maximum van 1000 simultane verbindingen voor elke verbinding 16 K aan ontvangstnetwerkbuffers en 16 K aan zendbuffers kost, dan is ongeveer 32 MB aan netbuffers nodig voor de webserver. Een goede vuistregel is te vermeniguldigen met twee, dus 2x32 MB / 2 KB = 64 MB / 2 kB = 32768. Voor machines met veel geheugen wordt 4096 tot 32768 aangeraden. Er moet in geen geval een arbitrair hoge waarde voor deze sysctl opgegeven worden, want dat kan leiden tot een crash tijdens het opstarten. Met de optie `-m` van netstat(1) kan het clustergebruik van het netwerk bekeken worden.

De loaderparameter `kern.ipc.nmbclusters` moet gebruikt worden om dit tijdens het opstarten toe te passen. Alleen voor oudere versies van FreeBSD is het nodig om de kerneloptie `NMBCLUSTERS` te gebruiken.

Voor drukke servers die extensief gebruik maken van de systeemaanroep sendfile(2), kan het nodig zijn het aantal sendfile(2)-buffers te verhogen via de kerneloptie `NSFBUFS` of door de waarde in te stellen in /boot/loader.conf (in loader(8) staan details). Als er in de procestabel processen staan met een status `sfbufa` is dat een algemene indicator dat deze parameter aangepast moet worden. De sysctl-variabele `kern.ipc.nsfbufs` is alleen-lezen en laat zien op welke waarde deze kernelvariabele

is ingesteld. Deze parameter schaalt engiszins met de variabele `kern.maxusers`, maar het kan nodig zijn om deze bij te stellen.



Zelfs als een socket als non-blocking gemarkeerd is, dan nog kan het aanroepen van sendfile(2) op de non-blocking socket ertoe leiden dat er toch blokkade optreedt totdat er voldoende `struct sf_buf`'s vrijgemaakt zijn.

`net.inet.ip.portrange.*`

De sysctl-variabelen `net.inet.ip.portrange.*` bepalen welke reeks poortnummers automatisch gebonden wordt aan TCP- en UDP-sockets. Er zijn drie gebieden: een laag gebied, een (standaard) middengebied en een hoog gebied. De meeste netwerkprogramma's gebruiken het standaardbereik, wat begrensd wordt door `net.inet.ip.portrange.first` en `net.inet.ip.portrange.last` met standaardwaarden van respectievelijk 1024 en 5000. Gebonden poortreeksen worden gebruikt voor uitgaande verbindingen en het is onder bepaalde omstandigheden mogelijk dat poorten op raken. Dit gebeurt meestal in het geval van een zwaar belaste webproxy. Poortbereik is niet van belang als vooral diensten draaien die zich bezighouden met inkomende verbindingen, zoals een normale webserver, of als het aantal uitgaande verbindingen beperkt is, zoals bij een mailrelay. Voor situaties waarin een tekort aan poorten dreigt, wordt aangeraden om `net.inet.ip.portrange.last` bescheiden op te hogen. Een waarde van `10000`, `20000` of `30000` is redelijk. Er moet ook rekening met effecten op firewalls gehouden worden als de poortreeks gewijzigd wordt. Sommige firewalls kunnen grote poortreeksen blokkeren, meestal de lagere poorten, en verwachten dat andere systemen hogere poorten gebruiken voor uitgaande verbindingen. Om deze reden wordt het niet aanbevolen om `net.inet.ip.portrange.first` te verlagen.

TCP Bandbreedtevertragingsproduct (TCP Bandwidth Delay Product)

De TCP-bandbreedtevertragingsproductlimitatie lijkt op TCP/Vegas in NetBSD. Het kan aangezet worden door de sysctl-variabele `net.inet.tcp.inflight.enable` de waarde `1` te geven. Het systeem tracht dan het bandbreedtevertragingssprodukt te berekenen voor elke verbinding en beperkt dan de hoeveelheid gegevens in de wachtrij naar het netwerk tot de hoeveelheid die vereist is om maximale doorvoer te kunnen handhaven.

Dit is nuttig bij gebruik van modems, Gigabit Ethernet of zelfs bij WAN-verbindingen met hoge snelheid (of elke andere verbinding met een groot bandbreedtevertragingsprodukt), in het bijzonder als ook windowschaling of een groot verzendwindow gebruikt wordt. Als deze optie aangezet wordt, dient ook `net.inet.tcp.inflight.debug` de waarde `0` te krijgen (geen debugging) en voor produktiegebruik kan het instellen van `net.inet.tcp.inflight.min` naar minstens `6144` voordeel opleveren. Het instellen van hoge minima kan effectief het beperken van bandbreedte ondermijnen, afhankelijk van de verbinding. De mogelijkheid tot limitering zorgt ervoor dat de hoeveelheid gegevens die opgebouwd wordt, in tussentijdse route- en switchwachtrijen verlaagd kan worden en tevens kan de hoeveelheid gegevens die opgebouwd wordt in de interfacewachtrij van de lokale host verlaagd worden. Met minder pakketten in wachtrijen kunnen interactieve verbindingen opereren met lagere *Round Trip* tijden, met name over langzame modems. Deze optie gaat alleen over datatransmissie (upload / serverkant) en heeft geen effect gegevensontvangst (download / cliëntkant).

Aanpassen van `net.inet.tcp.inflight.stab` wordt *niet* aangeraden. Deze parameter krijgt

standaard een waarde van 20, wat 2 maximale pakketten opgeteld bij de bandbreedtevensterberekening representeert. Het extra venster is nodig om het algoritme stabiel te houden en om de reactietijd bij veranderende omstandigheden te verbeteren, maar het kan ook leiden tot langere pingtijden over langzame verbindingen (zonder het inflight-algoritme kan dit echter nog erger zijn). In dergelijke gevallen kan deze parameter misschien verlaagd worden naar 15, 10 of 5 en misschien moet voor het gewenste effect ook `net.inet.tcp.inflight.min` verlaagd worden (bijvoorbeeld naar 3500). Het verlagen van deze parameters moet pas in laatste instantie overwogen worden.

11.14.3. Virtueel Geheugen

`kern.maxvnodes`

Een vnode is de interne representatie van een bestand of een map. Het verlagen van het aantal beschikbare vnodes voor het besturingssysteem leidt dus tot een daling van schijf-I/O. Normaliter wordt dit door het besturingssysteem afgehandeld en hoeft de instelling niet gewijzigd te worden. Im sommige gevallen kan schijf-I/O de beperkende factor zijn en kan het systeem alle beschikbare vnodes in gebruik hebben. Dan dient deze instelling gewijzigd te worden. De hoeveelheid inactief en beschikbaar RAM dient meegenomen te worden in de beslissing.

Het huidige aantal gebruikte vnodes kan als volgt bekeken worden:

```
# sysctl vfs.numvnodes
vfs.numvnodes: 91349
```

Om het maximale aantal vnodes weer te geven:

```
# sysctl kern.maxvnodes
kern.maxvnodes: 100000
```

Als het huidige aantal gebruikte vnodes dicht bij het maximale aantal ligt, is het verstandig om `kern.maxvnodes` op te hogen met 1.000. Ook `vfs.numvnodes` dient in de gaten gehouden te worden. Als de waarde weer tot aan het maximum stijgt, dan moet `kern.maxvnodes` verder opgehoogd worden. Er dient een verschuiving op te treden in het door top(1) gerapporteerde geheugengebruik. Er hoort meer geheugen actief te zijn.

11.15. Wisselbestandruimte toevoegen

Hoe goed er ook gepland wordt, soms draait een systeem gewoon niet zoals verwacht. Een oorzaak hiervoor kan een tekort aan wisselbestandruimte zijn. Als blijkt dat er meer wisselbestandruimte nodig is, kan dat eenvoudig. Er zijn drie manieren om de totale ruimte beschikbaar als wisselbestand te vergroten: een nieuwe harde schijf toevoegen, swappen over NFS of een wisselbestand maken op een bestaande (UFS of andere) partitie.

Kijk voor informatie over het beveiligen van het wisselbestand, welke opties hiervoor bestaan, en waarom dit gedaan zou moeten worden in Het versleutelen van de wisselbestand ruimte van het handboek.

11.15.1. Swap op een nieuwe of bestaande harde schijf

Een nieuwe harde schijf voor swap toevoegen geeft betere prestaties dan een partitie aan een bestaande schijf toevoegen. Het aanmaken van partities en harde schijven wordt uitgelegd in Schijven toevoegen. Initiële instellingen bespreekt de overwegingen van partitie-indelingen en de grootte van swap-partities.

Gebruik swapon(8) om een swap-partitie aan het systeem toe te voegen, bijvoorbeeld:

```
# swapon /dev/ada1s1b
```



Het is mogelijk om elke partitie te gebruiken die momenteel niet aangekoppeld is, zelfs als deze al gegevens bevat. Het gebruik van swapon(8) op een partitie die gegevens bevat zal deze gegevens overschrijven en vernietigen. Zorg ervoor dat de partitie die als swap toegevoegd wordt echt de bedoelde partitie is voordat swapon(8) gebruikt wordt.

Voeg een regel toe aan /etc/fstab voor de partitie om deze swap-partitie automatisch toe te voegen tijdens het opstarten:

```
/dev/ada1s1b    none    swap    sw  0   0
```

Raadpleeg fstab(5) voor een uitleg over de regels in /etc/fstab.

11.15.2. Swappen over NFS

In het algemeen wordt swappen over NFS niet aangeraden behalve als het onmogelijk is om naar een lokale schijf te swappen. NFS-swappen wordt gelimiteerd door de hoeveelheid beschikbare bandbreedte en belast het de NFS-server.

11.15.3. Wisselbestanden

Het is mogelijk om een bestand aan te maken van een bepaalde grootte en dit als swap te gebruiken. In dit voorbeeld wordt een bestand van 64 MB gebruikt, /usr/swap0. Uiteraard kan een willekeurige naam gebruikt worden.

Voorbeeld 4. Een wisselbestand aanmaken op FreeBSD

1. De kernel GENERIC bevat reeds het stuurprogramma voor geheugenschijven (md(4)) dat nodig is voor deze bewerking. Zorg ervoor dat tijdens het bouwen van een eigen kernel de volgende regel in uw configuratiebestand zit:

   ```
   device md
   ```

   Kijk voor meer informatie over het bouwen van een eigen kernel in De FreeBSD-kernel instellen.

2. Het wisselbestand /usr/swap0 aanmaken:

```
# dd if=/dev/zero of=/usr/swap0 bs=1024k count=64
```

3. De correcte rechten op /usr/swap0 instellen:

```
# chmod 0600 /usr/swap0
```

4. Het wisselbestand opnemen in /etc/rc.conf:

```
swapfile="/usr/swap0"   # Instellen op naam van wisselbestand als
hulpwisselbestand gewenst is
```

5. De machine moet herstart worden of om het wisselbestand direct in te schakelen:

```
# mdconfig -a -t vnode -f /usr/swap0 -u 0  swapon /dev/md0
```

11.16. Energie- en bronnenbeheer

Het is belangrijk om hardwarebronnen op een efficiënte wijze te benutten. Voordat ACPI geïntroduceerd werd was het lastig en onflexibel om het energieverbruik en de thermische eigenschappen van een systeem te beheersen. De hardware werd beheerst de BIOS en dus had de gebruiker minder controle en zichtbaarheid in de energiebeheerinstellingen. Enige gelimiteerde configuratie was mogelijk via *Advanced Power Management (APM)*. Energie- en bronnenbeheer is een belangrijk onderdeel van moderne machines. Het besturingssysteem moet bijvoorbeeld systeemlimieten in de gaten houdt (en mogelijk een SMS sturen of iets dergelijks) als de systeemtemperatuur onverwacht toeneemt.

In dit deel van het FreeBSD handboek wordt uitgebreide informatie verschaft over ACPI. Aan het einde worden referenties geleverd naar meer leesmateriaal.

11.16.1. Wat is ACPI?

Advanced Configuration and Power Interface (ACPI) is een standaard die door een alliantie van producenten geschreven is, met als doel te voorzien in een standaardinterface voor hardwarebronnen- en energiebeheer. Een belangrijk element is dat het meer flexibiliteit en beheersmogelijkheden biedt aan het besturingssysteem (OS). Moderne systemen hebben de limieten van de huidige PNP-interfaces verder opgerekt dan wenselijk en misschien wel mogelijk was. ACPI is de directe opvolger van APM (Advanced Power Management). Centraal is het verleggen van hardwarebeheer en -monitoring naar de OS-laag in plaats van de zeer beperkte BIOS-laag.

11.16.2. Tekortkomingen van APM

Met de *Advanced Power Management (APM)* faciliteit kan het energieverbruik van een systeem geregeld worden op basis van de systeemactiviteit. Het APM-BIOS wordt geleverd door de systeemproducent of -verkoper en het is specifiek voor dat betreffende hardwareplatform. Een APM-stuurprogramma in het besturingssysteem regelt vervolgens de toegang tot de *APM Software Interface*, die het besturen van vermogensniveau mogelijk maakt. APM dient nog steeds gebruikt te worden met systemen die gefabriceerd zijn voor het jaar 2000.

Er zijn vier hoofdproblemen met APM te onderscheiden: ten eerste wordt het energiebeheer verricht door een BIOS (afhankelijk van producent) en het besturingssysteem heeft daar geen kennis van. De gebruiker die idle-time waarden instelt voor een harde schijf in het APM-BIOS is hier een voorbeeld van. Dan zal het BIOS de harde schijf langzamer kunnen laten draaien zonder dat het besturingssysteem de noodzaak ziet of het goedkeurt. Ten tweede: de APM-logica is ingebed in de BIOS, waardoor het buiten het besturingssysteem om opereert. Dit houdt in dat gebruikers problemen met hun APM-BIOS alleen kunnen verhelpen door een nieuw BIOS in het ROM te flashen, wat een gevaarlijke en mogelijk onherstelbare operatie is. Ten derde is APM een producent-specifieke technologie, in de zin dat er altijd een hoge mate van duplicatie zal zijn van al dan niet geslaagde pogingen om het wiel opnieuw uit te vinden en uiteraard ook van bugs. Er is geen enkele garantie dat het wegnemen van een bug door een producent ook een zelfde bug wegneemt bij een concurrent. Tenslotte is het van belang te weten dat de APM-BIOS in het algemeen gewoon te weing geheugen kon gebruiken om een ingewikkeld energiebeheer te kunnen implementeren. Laat staan dat deze goed aanpasbaar was aan veranderlijke doelstellingen voor de betreffende machine.

Plug-n-play BIOS (PNPBIOS) was in veel situaties onbetrouwbaar. PNPBIOS is 16-bitstechnologie, dus het besturingssysteem moet 16-bit emulatie gebruiken om met PNPBIOS-methoden te kunnen samenwerken.

Het FreeBSD-stuurprogramma APM is gedocumenteerd in apm(4).

11.16.3. ACPI instellen

Het stuurprogramma acpi.ko wordt standaard geladen bij het opstarten door de loader(8) en hoeft *niet* gecompileerd te worden. De redenatie is dat er met modules gemakkelijker gewerkt kan worden, bijvoorbeeld een andere acpi.ko gebruiken zonder dat er een nieuwe kernel gebouwd moet worden. Dit heeft het voordeel dat testen eenvoudiger is. Een andere reden is dat het opstarten van ACPI nadat een systeem eenmaal volledig opgestart is meestal niet goed werkt. Mocht er hinder ondervonden worden, dan kan ACPI beter uitgeschakeld worden. Dit stuurprogramma kan niet gestopt worden als het eenmaal geladen is, omdat de systeembus het gebruikt voor allerlei interacties met hardware. ACPI kan uitgezet worden door het instellen van `hint.acpi.0.disabled="1"` in /boot/loader.conf of in de loader(8) prompt.



ACPI en APM kunnen niet samenleven en moeten afzonderlijk en exclusief gebruikt worden. De laatste die gestart wordt bepaalt of het stuurprogramma de ander wel of niet ziet.

In haar eenvoudigste vorm kan ACPI gebruikt worden om het systeem in slaapmodus te zetten met acpiconf(8) met de vlag `-s` en een optie `1-5`. De meeste gebruikers hebben alleen `1` of `3` nodig. De

optie 5 verricht een "soft-off", wat hetzelfde is als:

```
# halt -p
```

Andere opties zijn mogelijk via sysctl(8). Zie de handleidingen van acpi(4) en acpiconf(8) voor meer informatie.

11.17. FreeBSD ACPI gebruiken en debuggen

ACPI is een totaal nieuwe manier om apparaten te ontdekken, om energieverbruik te beheren en om een gestandaardiseerde toegang te bieden tot allerlei apparaten die eerder via het BIOS beheerd werden. Er wordt voortdurend vooruitgang geboekt om ACPI op alle systemen te laten werken, maar bugs in de *ACPIMachine Language* (AML) bytecode van sommige moederborden, onvolledigheden in de subsystemen van de kernel van FreeBSD en bugs in de Intel® ACPI-CA interpreter blijven opduiken.

Deze tekst is bedoeld om u te helpen met het bijstaan van de FreeBSD ACPI beheerders met het vinden van de hoofdoorzaken van problemen die u opmerkt en met het debuggen en het vinden van een oplossing.

11.17.1. Debuginformatie aanleveren



Voordat een probleem wordt gemeld, moet het zeker zijn dat de laatste BIOS versie draait en indien beschikbaar de geïntregeerde controller firmware versie.

Diegenen die meteen een probleem willen indienen, sturen de volgende informatie naar freebsd-acpi@FreeBSD.org:

- Omschrijving van het foutieve gedrag, inclusief systeemtype en -model en alles wat de fout kan veroorzaken. Als het een nieuw fenomeen is, dan dient ook zo accuraat mogelijk aangegeven te worden wanneer de fout het eerst optrad.
- De uitvoer van dmesg(8) van `boot -v`, inclusief foutmeldingen die gegenereerd worden als de fout optreedt.
- De uitvoer van dmesg(8) van `boot -v` met ACPI uitgeschakeld, indien het uitzetten van ACPI het probleem oplost.
- Uitvoer van `sysctl hw.acpi`. Dit is tevens een goede manier om uit te vinden welke ACPI-mogelijkheden een systeem heeft.
- Een URL waar de *ACPISource Language* (ASL) gevonden kan worden. De ASL dient *niet* rechtstreeks naar de lijst gezonden te worden, omdat deze nogal groot kan zijn. Een kopie van een ASL kan gemaakt worden met het volgende commando:

  ```
  # acpidump -dt > naam-systeem.asl
  ```

  (Vervang uw aanmeldnaam door $NAME en producent/model door $SYSTEM. Bijvoorbeeld: njl-

FooCo6000.asl)

De meeste FreeBSD-programmeurs lezen de FreeBSD-CURRENT mailinglijst, maar problemen gaan bij voorkeur ook naar FreeBSD ACPI mailinglijst zodat ze zeker gezien worden. Het kan enige tijd duren voordat er antwoord komt, omdat deze mensen elders ook nog volledige banen hebben. Als de bug niet meteen duidelijk is, komt er waarschijnlijk en verzoek om een PR in te dienen via send-pr(1). Als er een PR moet worden opgesteld, dan dient alle hierboven gevraagde informatie vermeld te worden. Dit helpt om het probleem te kunnen volgen en oplossen. Het sturen van een PR zonder eerst FreeBSD ACPI mailinglijst te mailen is niet wenselijk, aangezien men PRs gebruikt als herinnering van bestaande problemen, niet als rapportagesysteem. Mogelijk is een probleem al eens door iemand anders gemeld.

11.17.2. Achtergrond

ACPI is aanwezig op alle moderne computers die voldoen aan de ia32 (x86), ia64 (Itanium) of amd64 (AMD) architecturen. De volledige standaard heeft vele mogelijkheden zoals CPU-prestatiebeheer, energiebeheer, thermische zones, diverse batterijsystemen, ingebedde controllers en busnummering. De meeste systemen implementeren minder dan de volledige standaard. Een desktopsysteem implementeert bijvoorbeeld meestal alleen busnummering, terwijl laptops mogelijk ook koeling- en batterijbeheer ondersteunen. Laptops hebben ook suspend en resume (slapen en wakker worden) met hun eigen aanverwante comlexiteit.

Een ACPI-compliant systeem heeft verscheidene componenten. Het BIOS- en chipsetverkopers bieden verscheidene vaste tabellen aan zoals FADT in het geheugen die zaken als de APIC-afbeelding (gebruikt voor SMP), configuratieregisters, en eenvoudige configuratiewaarden specificeren. Ook wordt er een tabel van bytecode (de *Differentiated System Description Table* of DSDT) geleverd die een op een boomstructuur lijkende namespace biedt voor apparaten en methoden.

Het stuurprogramma ACPI moet de voorgedefinieerde tabellen verwerken, een interpreter voor de bytecode implementeren en apparaatstuurprogramma's en de kernel aanpassen om informatie van het ACPI-subsysteem te accepteren. Intel® heeft een interpreter beschikbaar gesteld (ACPI-CA) die door FreeBSD en ook door Linux® en NetBSD gebruikt wordt. De ACPI-CA-broncode staat in src/sys/contrib/dev/acpica. De lijmcode die ACPI-CA laat werken met FreeBSD staat in src/sys/dev/acpica/Osd. Stuurprogramma's die verscheidene ACPI-apparaten implementeren staan in src/sys/dev/acpica.

11.17.3. Algemene problemen

Wil ACPI goed werken, dan moeten alle onderdelen goed werken. Hieronder staan enkele algemene problemen in volgorde van hoe vaak ze optreden en enkele mogelijke oplossingen of manieren om de problemen te vermijden.

Muisproblemen

Soms doet een muis het niet bij het opstarten uit de slaapstand. Een bekend lapmiddel is het toevoegen van `hint.psm.0.flags="0x3000"` aan /boot/loader.conf. Als dat niet werkt, dan wordt aangeraden een bugrapport in te sturen, zoals eerder is beschreven.

Suspend/resume

ACPI heeft drie slaapstanden waarbij het geheugen (RAM) wordt ingezet. Dit zijn de STR-toestanden S1-S3, en nog een slaap-met-gebruik-van-harde-schijf toestand (STD) die S4 heet. S5 is "zacht uit" en is de normale status van een systeem als het is aangesloten maar niet is aangezet. S4 kan feitelijk op twee manieren geïmplementeerd worden: S4BIOS is een slaapstand naar schijf met behulp van het BIOS en S4OS wordt volledig door het besturingssysteem geïmplenteerd.

als eerste dienen de `sysctl hw.acpi` items die iets met de slaapstand te maken hebben gecontroleerd te worden. Hieronder staan de resultaten voor een Thinkpad:

```
hw.acpi.supported_sleep_state: S3 S4 S5
hw.acpi.s4bios: 0
```

Dit betekent dat hier `acpiconf -s` gebruikt kan worden om S3, S4OS en S5 te testen. Als `s4bios` gelijk was aan (1), dan zou er S4BIOS ondersteuning zijn in plaats van S4OS.

Als suspend/resume getest moet worden, dient, indien ondersteund, bij S1 begonnen te worden. Deze toestand heeft de grootste kans om te werken, omdat deze niet veel stuurprogrammaondersteuning vereist. Niemand heeft nog S2 geïmplementeerd, maar het is ongeveer hetzelfde als S1. Daarna wordt S3 getest. Dit is het diepste STR-niveau en heeft uitgebreide ondersteuning van stuurprogramma's nodig om hardware goed opnieuw te kunnen starten. Mochten er blokkades optreden, dan kan naar de FreeBSD ACPI mailinglijst lijst gemaild worden. Er kan echter geen snelle oplossing verwacht worden, omdat er nog de nodige stuurprogramma's/hardware liggen om getest en bewerkt te worden.

Een veelvoorkomend probleem met suspend/resume is dat veel apparaatstuurprogramma's hun firmware, registers of apparaatgeheugen niet fatsoenlijk opslaan, herstellen, of herinitialiseren. Een eerste poging om het probleem te vinden omvat:

```
# sysctl debug.bootverbose=1
# sysctl debug.acpi.suspend_bounce=1
# acpiconf -s 3
```

Deze test emuleert de suspend/resume-cyclus van alle apparaten zonder daadwerkelijk naar de toestand S3 te gaan. In sommige gevallen kunt u zo eenvoudig problemen vaststellen (bijvoorbeeld het verliezen van de firmware-toestand, timeout van de apparaatwaakhond, en steeds opnieuw iets proberen). Merk op dat het systeem niet werkelijk naar de toestand S3 gaat, wat inhoudt dat apparaten geen spanning verliezen waardoor velen prima zullen werken zelfs als de suspend/resume-methoden geheel ontbreken, dit in tegenstelling tot de echte toestand S3.

Moeilijkere gevallen vereisen aanvullende hardware, dat is een serieële poort/kabel voor de serieële console of een Firewire poort/kabel voor dcons(4), en vaardigheden in het debuggen van de kernel.

Om een probleem te kunnen isoleren helpt het om zoveel mogelijk stuurprogramma's uit de kernel te halen. Als dit werkt, kan er teruggewerkt worden naar het stuurprogramma dat schuldig is aan het falen. Meestal vertonen binaire stuurprogramma's als nvidia.ko, X11

beeldschermstuurprogramma's en USB de meeste problemen, terwijl bijvoorbeeld Ethernet-interfaces meestal meteen goed werken. Als de stuurprogramma's zonder problemen geladen en verwijderd kunnen worden, dan is dit te automatiseren door de juiste commando's in /etc/rc.suspend en /etc/rc.resume te zetten. Er staat een voorbeeld (achter commentaartekens) voor het laden en verwijderen van een stuurprogramma. Als het beeldscherm er na wakker worden vreemd uitziet, kan geprobeerd worden `hw.acpi.reset_video` op nul te zetten. Met langere of kortere waarden voor `hw.acpi.sleep_delay` kan bekeken worden of dat helpt.

In geval van problemen is het ook een optie om een recente Linux® distibutie met ondersteuning voor ACPI support te starten en daarvan de suspend/resume ondersteuning op dezelfde hardware uit te proberen. Als het werkt met Linux®, dan is het waarschijnlijk een FreeBSD stuurprogrammaprobleem en als het mogelijk is uit te vinden over welk stuurprogramma het gaat, kan dat bijdragen aan het oplossen van het probleem. ACPI houdt zich in het algemeen niet bezig met andere stuurprogramma's zoals geluid, ATA, enzovoort. Als er dus een echt probleem met een stuurprogramma is, dan is waarchijnlijk uiteindelijk ook nodig naar de FreeBSD-CURRENT mailinglijst lijst te posten en naar de beheerder van het stuurprogramma. Voor degenen met moed is het vooral aan te raden een paar printf(3)s in problematische stukken van een stuurprogramma te plaatsen voor debugging om na te gaan waar de resumefunctie precies hangt.

Tot slot kan geprobeerd worden om ACPI uit te zetten en in plaats daarvan APM aan te zetten. Als suspend/resume werkt met APM, is het wellicht verstandig het daarbij te houden, vooral met wat oudere apparatuur (voor 2000). Producenten hebben nogal wat tijd nodig gehad om ACPI ondersteuning goed te krijgen en voor oudere hardware is het waarschijnlijker dat er BIOS-problemen zijn met ACPI.

Systeem hangt (tijdelijk of permanent)

Meestal is het hangen van het systeem het gevolg van verloren interrupts of een interruptstorm. Chipsets kunnen een heleboel problemen hebben, afhankelijk van hoe het BIOS interrupts instelt voor het opstarten, of de APIC (MADT) tabel correct is en de routering van het *System Control Interrupt* (SCI).

Interruptstorms kunnen onderscheiden worden van verloren geraakte interrupts door de uitvoer van `vmstat -i` te controleren en de regel met `acpi0` goed te lezen. Als de teller in toenemende mate hoger staat dan enkele per seconde, dan is sprake van een interruptstorm. Als het systeem lijkt te hangen, is het wellicht nog mogelijk door te dringen tot de DDB (`CTRL` + `ALT` + `ESC`) en `show interrupts` uit te voeren.

De beste hoop in geval van interruptproblemen is om APIC-ondersteuning uit te zetten met `hint.apic.0.disabled="1"` in loader.conf.

Panics

Panics zijn relatief zeldzaam met ACPI en krijgen de hoogste prioriteit bij het oplossen. Eerst moeten de verschillende gebeurtenissen waarmee de panic (als mogelijk) te reproduceren is geïsoleerd worden en moet een backtrace gemaakt worden. `options DDB` dient aangezet te worden en er dient een seriële console (De debugger DDB gebruiken via de seriële verbinding) of een dump(8) partitie te komen. In DDB is een backtrace te maken met `tr`. Als de backtrace handmatig opgeschreven moet worden, is het belangrijk dat in ieder geval de bovenste en onderste vijf (5)

regels van de backtrace genoteerd worden.

Daarna dient getracht te worden het systeem te starten zonder ACPI. Als dat werkt, is het ACPI-subsysteem geïsoleerd en kunnen de verschillende `debug.acpi.disable`-waarden uitgeprobeerd worden. In acpi(4) staan enkele voorbeelden.

Systeem slaat aan na slaapstand of stop

`hw.acpi.disable_on_poweroff="0"` kan uitgezet worden in loader.conf(5). Hierdoor schakelt ACPI bepaalde gebeurtenissen tijdens het afsluitproces niet uit. Om dezelfde redenen moeten sommige systemen deze waarde altijd op `1` (standaard) hebben staan. In het algemeen lost dit een probleem op waarbij een systeem spontaan weer opkomt nadat het in slaapstand is gezet of geheel gestopt is.

Overige problemen

Als er nog andere problemen zijn met ACPI (met een docking station of apparaten niet gedetecteerd, enzovoort), dan kan een mail met beschijving naar de mailinglijst gezonden worden. Sommige zaken kunnen echter gerelateerd zijn aan delen van het ACPI-subsysteem die nog niet af zijn, dus het kan in sommige gevallen een tijd duren. Gebruikers moeten soms geduld en de bereidheid om eventuele patches uit te proberen hebben.

11.17.4. ASL, `acpidump` en IASL

Het grootste probleem is dat BIOS-producenten vaak incorrecte (of gewoon foutieve) bytecode leveren. Dit blijkt doorgaans uit kernelboodschappen als:

```
ACPI-1287: *** Error: Method execution failed [\\_SB_.PCI0.LPC0.FIGD._STA] \\
(Node 0xc3f6d160), AE_NOT_FOUND
```

Vaak kunnen dergelijke problemen geoplost worden door de BIOS bij te werken tot de laatste revisie. De meeste consoleberichten zijn onschuldig, maar als er andere problemen zijn, zoals batterijstatus die niet werkt, dan ligt het voor de hand te zoeken naar problemen in de AML-code. De bytecode die AML genoemd wordt, wordt gecompileerd van een broncodetaal ASL. Deze staat weer in een tabel DSDT. Met acpidump(8) kan een kopie van de ASL gemaakt worden. Dan moeten zowel de opties `-t` (laat inhoud van vaste tabellen zien) als `-d` (disassembleer AML naar ASL) gebruikt worden. In Debuginformatie aanleveren staat een voorbeeld.

De eenvoudigste eerste controle is de ASL-code opnieuw compileren en kijken of er foutmeldingen optreden. Waarschuwingen kunnen doorgaans genegeerd worden, maar fouten zijn bugs die er meestal toe leiden dat ACPI niet correct werkt. Om ASL te hercompileren:

```
# iasl eigen.asl
```

11.17.5. ASL repareren

Op langere termijn is het de bedoeling dat voor vrijwel elke machine ACPI werkt zonder enig ingrijpen van de gebruiker. Op dit moment wordt er echter nog gewerkt aan oplossingen voor veel

voorkomende vergissingen die BIOS-producenten maken. De Microsoft® interpreter (acpi.sys en acpiec.sys) controleert niet strikt of het BIOS volledig aan de standaard voldoet, waardoor het voorkomt dat BIOS-makers die alleen testen onder Windows® bepaalde fouten in hun ASL nooit correct repareren. FreeBSD hoopt door te gaan met de identificatie en documentatie van welk niet-standaard gedrag precies wordt toegelaten door Microsoft®'s interpreter en te dit te repliceren zodat FreeBSD kan werken zonder dat gebruikers zich gedwongen zien om de ASL te repareren. Als een tijdelijke oplossing en om te helpen met het in kaart brengen van bepaald gedrag, kan de ASL handmatig gerepareerd worden. Mocht dit lukken, dan wordt erop aangedrongen een diff(1) van de oude en de nieuwe ASL te mailen, zodat het foutieve gedrag mogelijk in ACPI-CA kan worden verwerkt, waardoor andere gebruikers niet meer handmatig met hun ASL aan de gang hoeven.

Hieronder staat een lijst algemene foutmeldingen, hun oorzaken en hoe ze op te lossen:

_OS afhankelijkheden

Sommige AMLs gaan ervan uit dat de wereld enkel bestaat uit Windows® versies. FreeBSD kan zich voordoen als elk OS om te kijken of dit problemen oplost. Een gemakkelijke manier om dit te doen is `hw.acpi.osname="Windows 2001"` in te stellen in /boot/loader.conf of andere gelijksoortige strings die in een ASL staan.

Ontbrekende return-opdrachten

Sommige methoden hebben geen specifieke returnwaarde, zoals wel vereist wordt door de standaard. Hoewel ACPI-CA hier niets mee doet, heeft FreeBSD de mogelijkheid tot impliciete returns. Er kunnen ook expliciete return-opdrachten toegevoegd worden waar vereist, als het bekend is welke waarden teruggevoerd moeten worden. Om `iasl` te dwingen tot compilatie van ASL kan de schakeloptie `-f` gebruikt worden.

De standaard AML aanpassen

Nadat eigen.asl aangepast is, kan deze als volgt gecompileerd worden:

```
# iasl eigen.asl
```

Met de optie `-f` is af te dwingen dat de AML gemaakt wordt, zelfs als er compileerfouten optreden. Sommige fouten (zoals ontbrekende return-opdrachten) worden automatisch opgelost door de interpreter.

DSDT.aml is de standaardnaam voor het bestand dat door `iasl` wordt geproduceerd. Dit is in plaats van de foutieve versie uit het BIOS (die nog steeds aanwezig is in het flashgeneugen) te laden door /boot/loader.conf als volgt te wijzigen:

```
acpi_dsdt_load="YES"
acpi_dsdt_name="/boot/DSDT.aml"
```

DSDT.aml moet in de map /boot staan.

11.17.6. Debuguitvoer van ACPI verkrijgen

Het stuurprogramma ACPI heeft een zeer flexibele debugfaciliteit. Er kan zowel een verzameling van subsystemen aangegeven worden als het niveau van uitvoerigheid. De te debuggen subsystemen worden aangegeven als lagen ("layers") en zijn opgedeeld in ACPI-CA-componenten (ACPI_ALL_COMPONENTS) en ACPI-hardware-ondersteuning (ACPI_ALL_DRIVERS). De uitvoerigheid van debuguitvoer wordt aangegeven als het niveau ("level") en gaat van CPI_LV_ERROR (alleen fouten rapporteren) tot ACPI_LV_VERBOSE (alles). Het niveau is een bitmasker en dus kunnen er meerdere opties tegelijk ingeschakeld worden (gescheiden door spaties). In de praktijk wordt wellicht een seriële console gebruikt om de uitvoer te loggen als deze zo omvangrijk is dat de console berichtbuffer vol loopt (misschien wel meerdere keren). Een complete lijst van de individuele lagen en niveaus staat in acpi(4).

Debuguitvoer staat standaard niet aan. Door `options ACPI_DEBUG` toe te voegen aan het bestand met kernelinstellingen als ACPI als de kernel is gebouwd, wordt het ingeschakeld. Door `ACPI_DEBUG=1` toe te voegen aan /etc/make.conf wordt het systeembreed ingeschakeld. Als ACPI als module wordt gebruikt (de normale situatie), dan hoeft slechts de module acpi.ko opnieuw gecompileerd te worden:

```
# cd /sys/modules/acpi/acpi
 make clean
make ACPI_DEBUG=1
```

acpi.ko moet in /boot/kernel komen te staan en de gewenste debuglaag en het gewenste niveau van uitvoerigheid dienen toegevoegd te worden aan loader.conf. Hieronder een voorbeeld waarmee debuguitvoer wordt aangezet voor alle ACPI-CA-componenten en alle ACPI-hardware-stuurprogramma's (CPU, LID, enzovoort. Het niveau van uitvoerigheid is het laagst mogelijke. Er worden alleen fouten gemeld.

```
debug.acpi.layer="ACPI_ALL_COMPONENTS ACPI_ALL_DRIVERS"
debug.acpi.level="ACPI_LV_ERROR"
```

Als de gezochte informatie wordt veroorzaakt door een specifieke gebeurtenis (bijvoorbeeld in en uit slaapstand gaan), dan kunnen wijzigingen aan loader.conf achterwege blijven en in plaats daarvan kan `sysctl` gebruikt worden om laag en niveau in te stellen na het opstarten en zo het systeem voor te bereiden op die specifieke gebeurtenis. De `sysctl`s hebben dezelfde namen als de parameters in loader.conf.

11.17.7. Verwijzingen

Meer informatie over ACPI staat op de volgende locaties:

- De FreeBSD ACPI mailinglijst
- De ACPI mailinglijst archieven http://lists.freebsd.org/pipermail/freebsd-acpi/
- De oude ACPI mailinglijst archieven http://home.jp.FreeBSD.org/mail-list/acpi-jp/
- De ACPI specificatie

- FreeBSD Handleidingen: acpi(4), acpi_thermal(4), acpidump(8), iasl(8), acpidb(8)
- DSDT debugging informatie. (Gebruikt Compaq als voorbeeld, maar van algemeen nut).

Hoofdstuk 12. Het FreeBSD opstartproces

12.1. Overzicht

Het proces van het starten van de computer en het laden van het besturingssysteem wordt het "bootstrapproces" of simpelweg "booten" genoemd. Het FreeBSD opstartproces levert een grote mate van flexibiliteit doordat gewijzigd kan worden wat er gebeurt als het systeem start en geeft de mogelijkheid om te kiezen uit verschillende geïnstalleerde besturingssystemen op dezelfde computer of zelfs verschillende versies van hetzelfde besturingssysteem of geïnstalleerde kernel.

Dit hoofdstuk geeft gedetailleerde informatie over instellingen die gebruikt kunnen worden en hoe het FreeBSD opstartproces veranderd kan worden. Dit omvat alles wat er gebeurt totdat de FreeBSD kernel wordt geladen, gezocht heeft naar apparaten en init(8) start. Dit vindt plaats als tijdens het booten de tekstkleur verandert van helder wit naar grijs.

Na het lezen van dit hoofdstuk weet de lezer:

- Wat de onderdelen zijn van het FreeBSD bootstrap-systeem en hoe zij onderling communiceren;
- De opties die meegegeven kunnen worden aan de componenten in de bootstrap om het proces te sturen;
- Meer over device.hints(5);



Alleen voor x86

Dit hoofdstuk beschrijft alleen het opstartproces van FreeBSD dat draait op een Intel x86 systeem.

12.2. Het bootprobleem

Het aanzetten van een computer en het starten van het besturingssysteem zorgt voor een interessant dilemma. Vast staat dat een computer niet weet wat hij moet doen totdat het besturingssysteem gestart is. Daar valt ook het starten van programma's op schijf onder. Dus als een computer geen programma van schijf kan starten zonder besturingssysteem en het besturingssysteem staat op schijf, hoe wordt het besturingssysteem dan gestart?

Dit is een gelijksoortig probleem als dat in het boek De avonturen van Baron von Münchausen. Iemand is in een put gevallen en heeft zichzelf eruit gehaald door zijn laarsriempjes (bootstraps) vast te pakken en zich op te trekken. In het begin van het computertijdperk is de term *bootstrap* gegeven aan het mechanisme dat het besturingssysteem laadt. Later werd dit afgekort tot "booten".

Op x86 machines is het Basis Input/Output Systeem (BIOS) verantwoordelijk voor het laden van het besturingssysteem. Om dit te doen zoekt het BIOS op de harde schijf naar het Master Boot Record (MBR), dat op een vaste plek op de schijf staat. Het BIOS heeft voldoende kennis om het MBR te starten en gaat er vanuit dat de MBR de rest van de taken uitvoert die nodig zijn om het besturingssysteem te kunnen laden, mogelijk met hulp van het BIOS.

Aan de code binnen de MBR wordt meestal gerefereerd als een *bootmanager*, in het bijzonder als

die interactie heeft met een gebruiker. In dit geval heeft de bootmanager meestal meer code in de eerste *track* van een schijf binnen het bestandssysteem van een besturingssysteem. Een bootmanager wordt soms ook *boot loader* genoemd, maar FreeBSD gebruikt die term voor een later stadium van het starten. Populaire bootmanagers zijn onder andere boot0 (ook bekend als Boot Easy, de standaard FreeBSD bootmanager), Grub, GAG en LILO (alleen boot0 past binnen de MBR.)

Als er maar één besturingssysteem en een schijf geïnstalleerd is, voldoet een standaard PC MBR. Dit MBR zoekt naar de eerste opstartbare (alias actieve) slice op schijf en start de code op deze slice om de rest van het besturingssysteem te laden. De MBR die standaard door fdisk(8) wordt geïnstalleerd is zo'n MBR. Die is gebaseerd op /boot/mbr.

Indien er meerdere besturingssystemen op schijven staan, kan er een andere bootmanager geïnstalleerd worden, een die een lijst toont met verschillende besturingssystemen en de mogelijkheid geeft om er één te kiezen dat opgestart moet worden. In de volgende paragrafen worden er twee beschreven.

Het resterende deel van het FreeBSD bootstrap-systeem is verdeeld in drie fases. De eerste fase wordt gestart door het MBR, dat net voldoende informatie heeft om de computer in een bepaalde toestand te zetten en de tweede fase te starten. De tweede fase kan net iets meer doen voordat hij de derde fase start. De derde fase voltooit het laden van het besturingssysteem. Dit proces is verdeeld in drie fases omdat de PC-standaarden grenzen stellen aan de grootte van programma's die gedraaid kunnen worden in de eerste twee fases van dit proces. Door deze taken aan elkaar te koppelen krijgt FreeBSD een flexibeler laadgedeelte.

Daarna wordt de kernel gestart en begint met het zoeken naar en initialiseren van apparaten. Zodra het kernel-opstartproces klaar is, geeft de kernel de controle over aan het gebruikerproces init(8), dat controleert of de schijven een bruikbare status hebben. Dan start init(8) de instellingen op gebruikersniveau die de bestandssystemen mount, de netwerkkaarten instelt voor communicatie met het netwerk en in het algemeen worden de processen gestart die moeten draaien op een FreeBSD systeem bij het opstarten.

12.3. De bootmanager en opstartstadia

12.3.1. De bootmanager

De code in de MBR of bootmanager wordt soms ook wel *stage zero* van het opstartproces genoemd. In dit onderdeel worden twee eerder genoemde bootmanagers beschreven: boot0 en LILO.

De boot0 bootmanager: De MBR die standaard door de FreeBSD installer of boot0cfg(8) wordt geïnstalleerd is gebaseerd op /boot/boot0. Het programma boot0 is erg eenvoudig, omdat maar 446 bytes lang mag zijn vanwege de slicetabel en de `0x55AA` identificatie aan het einde van de MBR. Als de FreeBSD MBR is geïnstalleerd en er staan andere besturingssystemen op een harde schijf, dan is bij het opstarten een scherm te zien dat er ongeveer zo uitziet:

Voorbeeld 5. boot0 schermafbeelding

```
F1 DOS
F2 FreeBSD
```

```
F3 Linux
F4 ??
F5 Drive 1

Default: F2
```

Andere besturingssystemen, Windows® in het bijzonder, staan er om bekend dat zij bestaande MBRs overschrijven met die van zichzelf. Als dit is gebeurd of als het bestaande MBR vervangen moet worden door het FreeBSD MBR:

```
# fdisk -B -b /boot/boot0 apparaat
```

Waar *apparaat* het apparaat is waar de computer van boot, zoals ad0 voor de eerste IDE-schijf ad2 voor de eerste IDE-schijf op de tweede IDE-controller, da0 voor de eerste SCSI-schijf, enzovoort. Als het wenselijk is een aangepaste instelling te gebruiken voor de MBR, dan kan boot0cfg(8) gebruikt worden.

De LILO bootmanager: Start, om deze bootmanager te installeren zodat er ook FreeBSD mee gestart kan worden, eerst Linux® en voeg het volgende toe aan het bestaande instellingenbestand /etc/lilo.conf:

```
other=/dev/hdXY
table=/dev/hdX
loader=/boot/chain.b
label=FreeBSD
```

Geef in de bovenstaande regels de primaire partitie en schijf van FreeBSD op met Linux® instellingen, waarbij *X* vervangen wordt door de Linux® schijfletter en *Y* door het primaire partitienummer van Linux®. Wijzig bij gebruik van een SCSI-schijf */dev/hd* in iets als */dev/sd*. De regel `loader=/boot/chain.b` kan achterwege blijven als de besturingssystemen op dezelfde schijf staan. Voer daarna `/sbin/lilo -v` uit om de wijzigingen vast te leggen. Controleer het vastleggen door controle van de schermberichten.

12.3.2. Fase één /boot/boot1 en fase twee /boot/boot2

Conceptueel zijn de eerste en tweede fase onderdeel van hetzelfde programma op hetzelfde stukje schijf. Door ruimtebeperkingen zijn ze in twee stukken gesplitst. Ze worden echter altijd samen geïnstalleerd. Ze worden gekopieerd uit het gecombineerde bestand /boot/boot door het installatieprogramma of bsdlabel (zie verderop).

Ze staan buiten bestandssystemen in de eerste track van de opstartslice, beginnend bij de eerste sector. Dit is waar boot0 en iedere andere bootmanager een programma verwacht om door te gaan met het opstartproces. Het aantal gebruikte sectoren kan eenvoudig bepaald worden uit de grootte van /boot/boot.

boot1 is erg simpel omdat dit slechts 512 bytes groot kan zijn en net genoeg weet over het FreeBSD

bsdlabel, dat informatie bevat over de slice om boot2 te vinden en te starten.

boot2 is iets verfijnder en begrijpt het FreeBSD bestandssysteem genoeg om er bestanden op te vinden en geeft een simpele interface om de kernel of loader te kiezen die gestart moet worden.

boot2 start meestal de loader, doordat deze veel slimmer is en gebruikersvriendelijke opstartinstellingen heeft. Voorheen was het zijn taak direct de kernel te starten.

Voorbeeld 6. boot2 schermafbeelding

```
>> FreeBSD/i386 BOOT
Default: 0:ad(0,a)/boot/loader
boot:
```

Als ooit eens de geïnstalleerde boot1 en boot2 vervangen moeten worden kan dat met bsdlabel(8):

```
# bsdlabel -B schijfslice
```

In het voorbeeld hierboven is *schijfslice* de schijf en slice waarvan opgestart wordt, zoals ad0s1 voor de eerste slice op de eerste IDE-schijf.



Gevaarlijk toegewijde modus

Als alleen een schijfnaam als ad0 gebruikt wordt in bsdlabel(8) wordt er een gevaarlijk toegewijde schijf zonder slices gemaakt. Dit is niet aan te raden en daarom wordt aangeraden voor het uivoeren van bsdlabel(8) de commandoregel nog een keer te controleren voordat er op `Return` wordt gedrukt.

12.3.3. Fase drie, /boot/loader

De loader is de laatse fase van de drietraps-bootstrap en deze bevindt zich op het bestandssysteem, meestal als /boot/loader.

De loader is bedoeld als een gebruikersvriendelijke manier voor de instelling, door gebruik te maken van een makkelijke commandoverzameling, gesteund door een krachtige vertaler met een wat complexere commandoverzameling.

Loader programmaverloop

Tijdens de start zoekt de loader naar een console en schijven en kijkt van welke schijf er opgestart wordt. Variabelen worden hiernaar gezet en er wordt een vertaler gestart zodat gebruikercommando's interactief of via een script kunnen worden doorgegeven.

Dan leest de loader /boot/loader.rc, die dan standaard /boot/defaults/loader.conf leest. Deze plaatst redelijke standaarden in variabelen en leest /boot/loader.conf voor lokale wijzigingen op deze variabelen. loader.rc reageert op deze variabelen door de geselecteerde modules en kernel te laden.

Als laatste wordt standaard door de loader 10 seconden gewacht op toetsinvoer en als dit niet wordt onderbroken laadt loader de kernel. Als het wel wordt onderbroken krijgt de gebruiker een prompt aangeboden die een eenvoudige commandoverzameling begrijpt. Hier kan de gebruiker variabelen wijzigen, alle modules stoppen en/of starten en uiteindelijk opstarten of herstarten.

Ingebouwde loadercommando's

Hieronder worden de meest gebruikte loadercommando's besproken. Een volledige omschrijving van alle beschikbare commando's staat in loader(8).

autoboot *seconden*

Gaat door met het opstarten van de kernel als deze niet wordt onderbroken binnen de opgegeven tijd in seconden. Er wordt een aftelproces getoond dat standaard op 10 seconden staat.

boot [-opties] [kernelnaam]

Start direct de kernel op met de opgegeven opties en naam, indien meegegeven. Het opgeven van een kernelnaam op de opdrachtregel is alleen van toepassing nadat een *unload*-commando is gegeven, anders wordt de kernel die hiervoor was geladen gebruikt.

boot-conf

Doorloopt hetzelfde automatische instellen van modules gebaseerd op variabelen zoals ook gebeurt bij het opstarten. Dit is alleen zinnig als eerst `unload` is gebruikt en enkele variabelen zijn gewijzigd, meestal `kernel`.

help [onderwerp]

Toont documentatie uit /boot/loader.help. Als het opgegeven onderwerp `index` is, wordt een lijst met beschikbare onderwerpen getoond.

include *bestandsnaam ...*

Verwerkt het bestand met de opgegeven naam. Het bestand wordt ingelezen en regel voor regel vertaald. Iedere foutmelding stopt direct het include-commando.

load [-t type] *bestandsnaam*

Laadt de kernel, kernel-module of bestand van opgegeven type en naam. Ieder argument achter de bestandsnaam wordt doorgegeven aan het bestand.

ls [-l] [padnaam]

Toont de lijst bestanden in het opgegeven pad of van de rootmap als geen pad wordt opgegeven. Als `-l` wordt meegegeven wordt ook de bestandsgrootte weergegeven.

lsdev [-v]

Toont de lijst met alle apparaten waarvan het mogelijk is om modules te kunnen laden. Als `-v` wordt meegegeven worden meer details getoond.

lsmod [-v]

Toont geladen modules. Als `-v` wordt meegegeven worden meer details getoont.

more *bestandsnaam*

Toont de inhoud van het opgegeven bestand met een pauze na iedere `LINES` regels.

reboot

Herstart het systeem onmiddelijk.

set *variabele*

Vult de omgevingsvariabele van de loader.

unload

Verwijdert alle geladen modules.

Loader voorbeelden

Hier zijn wat practische voorbeelden van het gebruik van loader:

- De kernel opstarten in single-user modus:

```
boot -s
```

- De gebruikelijke kernel en modules ontladen om daarna de oude (of een andere) kernel te laden:

```
unload
load kernel.old
```

kernel.GENERIC kan gebruikt worden als de algemene kernel die meegeleverd is bij de installatieschijf of kernel.old om de vorige geïnstalleerde kernel te gebruiken (als bijvoorbeeld de kernel is vervangen).



Zo worden de bekende modules geladen met een andere kernel:

```
unload
set kernel="kernel.old"
boot-conf
```

- Voor het laden van een kernelinstellingenscript (een script dat dingen doet die anders met de hand ingegeven zouden worden):

```
load -t userconfig_script /boot/kernel.conf
```

Splash-schermen tijdens het opstarten

Het splash-scherm creëert een visueel aantrekkelijker scherm in vergelijking met de originele opstartberichten. Dit scherm zal worden afgebeeld totdat een aanmeldprompt op de console

verschijnt of een X-schermbeheerder een aanmeldprompt aanbiedt.

Er zijn twee basisomgevingen beschikbaar in FreeBSD. De eerste is de verouderde standaardomgeving met de opdrachtregel op de virtuële console. Nadat het systeem klaar is met opstarten, wordt er een aanmeldprompt op de console gepresenteerd. De tweede omgeving is de grafische omgeving van het X11 Bureaublad. Nadat X11 en één van de grafische bureaubladomgevingen, zoals GNOME, KDE, of XFce zijn geïnstalleerd, kan het X11-bureaublad worden gestart door `startx` te gebruiken.

Sommige gebruikers verkiezen het grafische aanmeldscherm van X11 boven de traditionele op tekst gebaseerde aanmeldprompt. Schermbeheerders zoals XDM voor Xorg, gdm voor GNOME, en kdm voor KDE (en anderen van de Portscollectie) bieden een grafisch aanmeldscherm in plaats van de aanmeldprompt op het console. Na succesvol aanmelden bieden ze de gebruiker een grafisch bureaublad.

In de opdrachtregelomgeving zou het splash-scherm alle berichten over aftasten tijdens het opstarten en het starten van taken verbergen voordat het de aanmeldprompt laat zien. In een X11-omgeving zouden gebruikers een visueel overzichtelijkere opstartervaring krijgen dat meer lijkt op wat een gebruiker van een (Microsoft® Windows® of niet-Unix-systeem) zou ervaren.

Splash-schermfuncties

De splash-schermfunctie ondersteunt 256-kleuren-bitmaps (.bmp), ZSoft PCX (.pcx) en TheDraw (.bin) bestanden. Verder moeten de splash-afbeeldingsbestanden een resolutie van 320 bij 200 pixels of minder hebben om op de standaard VGA-adapters te werken.

Activeer de VESA-ondersteuning die in FreeBSD zit om grotere afbeeldingen, tot de maximale resolutie van 1024 bij 768 pixels, te gebruiken. Dit kan worden aangezet door de VESA-module tijdens het opstarten van het systeem te laden, of door de kernelconfiguratieoptie `VESA` toe te voegen en een eigen kernel te bouwen (zie De FreeBSD-kernel instellen). De ondersteuning voor VESA geeft gebruikers de mogelijkheid om een splash-schermafbeelding af te beelden dat het hele scherm vult.

Zolang het splash-scherm wordt afgebeeld tijdens het opstartproces, kan het ten alle tijden worden uitgezet door op een toetsenbordtoets te drukken.

Het splash-scherm is standaard ook een schermbeveiliging buiten X11. Na een periode van inactiviteit zal het scherm in het splash-scherm veranderen en herhaald door stappen van het veranderen van de intensiteit van de afbeelding lopen, van helder tot zeer donker. Dit standaardgedrag van het splash-scherm (schermbeveiliging) kan overruled worden door een regel met `saver=` toe te voegen aan /etc/rc.conf. De optie `saver=` heeft verschillende ingebouwde schermbeveiligingen om uit te kiezen, de volledige lijst staat in de handleidingpagina splash(4). De standaard schermbeveiliging heet "warp". Merk op dat de optie `saver=` die in /etc/rc.conf is gespecificeerd alleen betrekking heeft op virtuele consoles. Het heeft geen effect op X11-schermbeheerders.

Enkele meldingen van de bootloader, inclusief de opties van het opstartmenu en een getimede wachtende aftelprompt worden afgebeeld tijdens het opstarten, zelfs als het splash-scherm aanstaat.

Voorbeelden van splash-schermen kunnen gedownload worden van de galerij op

http://artwork.freebsdgr.org/. Door de port sysutils/bsd-splash-changer te installeren, kunnen de splash-afbeeldingen willekeurig elke keer dat er wordt opgestart uit een verzameling worden gekozen.

De splash-schermfunctie aanzetten

Het splash-schermbestand (.bmp, .pcx of .bin) moet op de rootpartitie staan, bijvoorbeeld in de map /boot.

Bewerk voor de standaardresolutie van het opstartscherm (256 kleuren, 320 bij 200 pixels, of minder) /boot/loader.conf zodat het volgende erin staat:

```
splash_bmp_load="YES"
bitmap_load="YES"
bitmap_name="/boot/splash.bmp"
```

Bewerk /boot/loader.conf voor grotere videoresoluties (tot maximaal 1024 bij 768 pixels) zodat dit bestand het volgende bevat:

```
vesa_load="YES"
splash_bmp_load="YES"
bitmap_load="YES"
bitmap_name="/boot/splash.bmp"
```

Het bovenstaande neemt aan dat /boot/splash.bmp voor het splash-scherm wordt gebruikt. Wanneer een PCX-bestand gewenst is, dienen de volgende opdrachten gebruikt te worden, en afhankelijk van de resolutie de regel `vesa_load="YES"`.

```
splash_pcx_load="YES"
bitmap_load="YES"
bitmap_name="/boot/splash.pcx"
```

In versie 8.3 is een andere mogelijkheid het gebruik van ASCII-kunst in TheDraw formaat.

```
splash_txt="YES"
bitmap_load="YES"
bitmap_name="/boot/splash.bin"
```

De bestandsnaam is niet beperkt tot "splash" zoals in het bovenstaande voorbeeld. Het kan van alles zijn zolang het een van de bovenstaande types is, zoals splash_640x400.bmp of bluewave.pcx.

Enkele andere interessante opties voor loader.conf:

`beastie_disable="YES"`

: Dit zal het menu met opstartopties niet weergeven, maar de getimede wachtende aftelprompt zal nog steeds aanwezig zijn. Zelfs zonder dat het menu met opstartopties wordt afgebeeld, zal

het invoeren van een optie in de getimede wachtende aftelprompt de actie van de overeenkomstige opstartoptie uitvoeren.

`loader_logo="beastie"`

Dit zal de standaardwoorden "FreeBSD", welke rechts van het menu met opstartopties worden afgebeeld vervangen door het gekleurde beastie-logo zoals vroegere uitgaven die hadden.

Raadpleeg voor meer informatie de handleidingpagina's splash(4), loader.conf(5), en vga(4).

12.4. Interactie met de kernel tijdens opstarten

Zodra de kernel is geladen door de loader (zoals gewoonlijk) of door boot2 (zonder de loader), wordt er als ze er zijn gekeken naar de opstartvlaggen en wordt het gedrag zo nodig aangepast.

12.4.1. Opstartvlaggen kernel

De meest voorkomende opstartvlaggen:

`-a`

Vraag tijdens de opstart van de kernel om het apparaat dat gemount moet worden als root bestandssysteem.

`-C`

Boot van cd-rom.

`-c`

Start UserConfig om instellingen te maken voor de kernel tijdens het opstarten.

`-s`

Start naar single-user modus.

`-v`

Geef meer tekst en uitleg tijdens het opstarten van de kernel.



In boot(8) staan alle bootvlaggen beschreven.

12.5. Device hints

Tijdens het opstarten van het systeem leest de boot loader(8) het bestand device.hints(5). Dit bestand slaat opstartinformatie voor de kernel op in variabelen, ook wel "device hints". Deze "device hints" worden door stuurprogramma's gebruikt voor instelling van apparaten.

Device hints kunnen ook bij het " Fase drie, /boot/loader" prompt ingevoerd worden. Variabelen kunnen toegevoegd worden met behulp van `set`, verwijderd worden met `unset` en bekeken worden met `show`. Variabelen uit /boot/device.hints kunnen hier ook herroepen worden. Device hints die ingevoerd zijn bij de boot loader zijn niet permanent en zijn bij de volgende boot niet meer aanwezig.

Zodra het systeem opgestart is, kan kenv(1) gebruikt worden om alle variabelen te bekijken.

De schrijfwijze voor /boot/device.hints is één variabele per regel. Het standaard hekje "#" wordt gebruikt voor commentaar. Regels worden als volgt opgebouwd:

```
hint.driver.unit.keyword="waarde"
```

De syntaxis voor de Fase 3 bootloader is:

```
set hint.driver.unit.keyword=waarde
```

`driver` is de naam van het apparaatstuurprogramma, `unit` is het apparaatnummer van het stuurprogramma en `keyword` is het hint-sleutelwoord. Dit sleutelwoord kan uit de volgende opties bestaan:

- `at`: beschrijft de bus waarop het apparaat is aangesloten.
- `port`: beschrijft het startadres van de I/O die gebruikt wordt.
- `irq`: beschrijft het interrupt request nummer dat gebruikt wordt.
- `drq`: beschrijft het DMA kanaalnummer.
- `maddr`: beschrijft het fysieke geheugenadres dat gebruikt wordt door het apparaat.
- `flags`: zet verschillende vlagbits voor het apparaat.
- `disabled`: is `1` als het apparaat is uitgezet.

Apparaatstuurprogramma's kunnen hints die hier niet genoemd zijn accepteren (of eisen). Zie hiervoor de betreffende handleiding: device.hints(5), kenv(1), loader.conf(5) en loader(8).

12.6. Init: start van procesbesturing

Als de kernel klaar is met opstarten geeft die de besturing over aan het gebruikerproces init(8), te vinden in /sbin/init of de padnaam die staat is in de variabele `init_path` in `loader`.

12.6.1. Automatische herstart

De automatische herstart ("Automatic Reboot Sequence") controleert of de beschikbare bestandssystemen betrouwbaar zijn. Als dat niet zo is en fsck(8) kan de fouten niet repareren, dan brengt init(8) het systeem terug naar Single-user modus voor de systeembeheerder, die het probleem dan directer kan aanpakken.

12.6.2. Single-user modus

Deze modus kan bereikt worden vanuit de Automatische herstart of door de gebruiker die opstart met de optie `-s` of door de variabele `boot_single` aan te zetten in de `loader`.

Het kan ook door shutdown(8) te starten zonder de optie reboot (`-r`) of halt (`-h`), vanuit Multi-user modus.

Als het systeem `console` op `insecure` staat in /etc/ttys, dan vraagt het systeem om het `root` wachtwoord voordat de single-user modus wordt gestart.

Voorbeeld 7. Onveilige console in /etc/ttys

```
# name  getty                           type    status          comments
#
# Als de console op "insecure" staat vraagt init om het root wachtwoord
# voor het naar single-user modus gaan.
console none                            unknown off insecure
```



Met een `insecure` console wordt bedoeld dat de fysieke beveiliging van het console niet goed is en dat alleen personen die het `root` wachtwoord kennen naar single-user modus mogen gaan. Het betekent niet dat het console onveilig wordt ingesteld. Als het veilig moet, wordt er dus voor `insecure` gekozen en niet voor `secure`.

12.6.3. Multi-user modus

Als init(8) vindt dat het bestandssysteem in orde is of zodra de gebruiker klaar is in Single-user modus, gaat het systeem over naar multi-user modus, waarin het de resource configuration (broninstellingen) van het systeem start.

Bronconfiguratie (rc)

Het broninstellingensysteem leest de standaard instellingen in vanuit /etc/defaults/rc.conf en specifieke systeemdetails uit /etc/rc.conf en gaat daarna door met het mounten van de bestandssystemen voor het systeem die genoemd worden in /etc/fstab, start netwerkdiensten, start andere systeemdaemons en start als laatste de opstartscripts van lokaal geïnstalleerde packages.

rc(8) is een goede referentie voor het broninstellingensysteem. Dat zijn de scripts zelf natuurlijk ook.

12.7. Afsluitvolgorde

Bij een gecontroleerde shutdown met shutdown(8) probeert init(8) om het script /etc/rc.shutdown te starten en daarna aan alle processen het `TERM` signaal te sturen en eventueel het `KILL` signaal aan alle processen die niet op tijd zijn gestopt.

Om een FreeBSD machine uit te zetten die energiebeheer ondersteund, kan het commando `shutdown -p now` gegeven worden om gelijk de stroom af te schakelen. Als er herstart moet worden dan kan `shutdown -r now` gebruikt worden. De gebruiker die dit uitvoert moet wel `root` zijn of lid van de `operator` groep om shutdown(8) te mogen gebruiken. halt(8) en reboot(8) kunnen ook gebruikt worden. Meer informatie is in de betreffende handleidingpagina's te vinden.



Voor energiebeheer is acpi(4) ondersteuning in de kernel nodig of via een module

> die ingeladen moet worden.

Hoofdstuk 13. Beveiliging

13.1. Overzicht

Dit hoofdstuk biedt een basisinleiding in systeembeveiligingsconcepten, een aantal goede basisregels en een paar gevorderde onderwerpen binnen FreeBSD. Veel van de onderwerpen die worden behandeld kunnen ook worden toegepast op systemen en Internet in het algemeen. Het Internet is niet langer een "vriendelijke" omgeving waar iedereen een goede buur wil zijn. Het beveiligen van een systeem is onontbeerlijk als gegevens, intellectueel eigendom, tijd en wat dan ook uit de handen van hackers en dergelijke gehouden moeten worden.

FreeBSD biedt veel hulpmiddelen en mechanismen om te zorgen voor de integriteit en veiligheid van een systeem en netwerk.

Na het lezen van dit hoofdstuk weet de lezer:

- Van basis systeembeveiligingsconcepten in relatie tot FreeBSD.
- Meer over verschillende versleutelingsmechanismen die beschikbaar zijn in FreeBSD zoals DES en MD5.
- Hoe eenmalige wachtwoordautenticatie opgezet kan worden.
- Hoe TCP Wrappers in te stellen voor gebruik met inetd.
- Hoe Kerberos5 op FreeBSD opgezet kan worden.
- Hoe IPsec wordt ingesteld en hoe een VPN op te zetten tussen FreeBSD en Microsoft® Windows® machines.
- Hoe OpenSSH, FreeBSD's SSH implementatie, in te stellen en te gebruiken.
- Wat bestandssysteem-ACLs zijn en hoe die te gebruiken;
- Hoe het hulpprogramma Portaudit gebruikt kan worden om softwarepakketten uit de Portscollectie te auditen.
- Hoe om te gaan met publicaties van FreeBSD beveiligingswaarschuwingen.
- Iets van procesaccounting en hoe dat is in te schakelen in FreeBSD.

Er wordt aangenomen dat de lezer van dit hoofdstuk:

- Basisbegrip heeft van FreeBSD en Internetconcepten.

In dit boek worden nog meer onderwerpen met betrekking tot beveiliging beschreven. Zo wordt bijvoorbeeld Verplichte Toegangscontrole (Mandatory Access Control) besproken in Mandatory Access Control en Internet Firewalls in Firewalls.

13.2. Introductie

Beveiliging is een taak die begint en eindigt bij de systeembeheerder. Hoewel alle BSD UNIX® meergebruikerssystemen enige inherente beveiliging kennen, is het bouwen en onderhouden van additionele beveiligingsmechanismen om de gebruikers "eerlijk" te houden waarschijnlijk een van

de zwaarste taken voor de systeembeheerder. Machines zijn zo veilig als ze gemaakt worden en beveiligingsoverwegingen staan altijd op gespannen voet met de wens om gebruiksvriendelijkheid. UNIX® systemen zijn in het algemeen in staat tot het tegelijkertijd uitvoeren van een enorm aantal processen en veel van die processen acteren als server - daarmee wordt bedoeld dat externe entiteiten er verbindingen mee kunnen maken en ertegen kunnen praten. Nu de minicomputers en mainframes van gisteren de desktops van vandaag zijn en computers onderdeel zijn van netwerken en internetwerken, wordt beveiliging nog belangrijker.

Systeembeveiliging heeft ook te maken met het omgaan met verschillende vormen van aanvallen, zoals een poging om een systeem te crashen of op een andere manier onstabiel te maken, zonder te proberen de `root` account aan te vallen ("break root"). Aandachtspunten voor beveiliging kunnen opgesplitst worden in categorieën:

1. Ontzeggen van dienst aanvallen ("Denial of Service").
2. Gebruikersaccounts compromitteren.
3. `root` compromitteren via toegankelijke servers.
4. `root` compromitteren via gebruikersaccounts.
5. Achterdeur creëren ("Backdoor").

Een ontzegging van dienst (DoS) aanval is een techniek die de machine middelen ontneemt. In het algemeen zijn DoS aanvallen brute kracht mechanismen die proberen de machine te crashen of op een andere manier onbruikbaar te maken door de machine of de netwerkcode te overvragen. Sommige DoS aanvallen proberen misbruik te maken van bugs in de netwerkcode om een machine met een enkel pakket te crashen. Zoiets kan alleen gerepareerd worden door een aanpassing aan de kernel te maken. Aanvallen op servers kunnen vaak hersteld worden door op de juiste wijze opties in stellen om de belasting van servers te limiteren in ongunstige omstandigheden. Omgaan met brute kracht aanvallen is lastiger. Zo is een aanval met gefingeerde pakketten ("spoofed-packet") vrijwel niet te stoppen, behalve dan door het systeem van Internet los te koppelen. Misschien gaat de machine er niet door plat, maar het kan wel een volledige Internetverbinding verzadigen.

Een gecompromitteerde gebruikersaccount komt nog veel vaker voor dan een DoS aanval. Veel systeembeheerders draaien nog steeds standaard telnetd, rlogind, rshd en ftpd servers op hun machines. Deze servers communiceren standaard niet over beveiligde verbindingen. Het resultaat is dat als er een redelijk grote gebruikersgroep is, er altijd wel van een of meer van de gebruikers die van afstand op dat systeem aanmelden (wat toch de meest normale en makkelijke manier is om op een systeem aan te melden) het wachtwoord is afgeluisterd ("sniffed"). Een oplettende systeembeheerder analyseert zijn logboekbestanden om te zoeken naar verdachte bronadressen, zelfs als het om succesvolle aanmeldpogingen gaat.

Uitgangspunt moet altijd zijn dat als een aanvaller toegang heeft tot een gebruikersaccount, de aanvaller de `root` account kan compromitteren. In werkelijkheid is het wel zo dat voor een systeem dat goed beveiligd is en goed wordt onderhouden, toegang tot een gebruikersaccount niet automatisch betekent dat de aanvaller ook `root` privileges kan krijgen. Het is van belang dit onderscheid te maken, omdat een aanvaller zonder toegang tot `root` in het algemeen zijn sporen niet kan wissen en op z'n best wat kan rommelen met bestanden van de gebruiker of de machine kan crashen. Gecompromitteerde gebruikersaccounts zijn vrij normaal omdat gebruikers normaliter niet de voorzorgsmaatregelen nemen die systeembeheerders nemen.

Systeembeheerders moeten onthouden dat er in potentie heel veel manieren zijn om toegang tot `root` te krijgen. Een aanvaller zou het `root` wachtwoord kunnen kennen, een bug kunnen ontdekken in een dienst die onder `root` draait en daar via een netwerkverbinding op in kunnen breken of een aanvaller zou een probleem kennen met een suid-root programma dat de aanvaller in staat stelt `root` te worden als hij eenmaal toegang heeft tot een gebruikersaccount. Als een aanvaller een manier heeft gevonden om `root` te worden op een machine, dan hoeft hij misschien geen achterdeur ("backdoor") te installeren. Veel bekende manieren die zijn gevonden om `root` te worden, en weer zijn afgesloten, vereisen veel werk van de aanvaller om zijn rommel achter zich op te ruimen, dus de meeste aanvallers installeren een achterdeur. Een achterdeur biedt de aanvaller een manier om makkelijk opnieuw `root` toegang tot het systeem te krijgen, maar dit geeft de slimme systeembeheerder ook een makkelijke manier om de inbraak te ontdekken. Het onmogelijk maken een achterdeur te installeren zou best wel eens nadelig kunnen zijn voor beveiliging, omdat hiermee nog niet het gat gedicht is waardoor er in eerste instantie is ingebroken.

Beveiligingsmaatregelen moeten altijd geïmplementeerd worden in een meerlagenmodel en worden als volgt gecategoriseerd:

1. Beveiligen van `root` en medewerkersaccounts.
2. Beveiligen van `root` - servers onder `root` en suid-/sgid-binaire bestanden.
3. Beveiligen van gebruikersaccounts.
4. Beveiligen van het wachtwoordbestand.
5. Beveiligen van de kern van de kernel, ruwe apparaten en bestandssystemen.
6. Snel detecteren van ongeoorloofde wijzigingen aan het systeem.
7. Paranoia.

In het volgende onderdeel van dit hoofdstuk gaan we dieper in op de bovenstaande punten.

13.3. FreeBSD beveiligen



Commando versus protocol

In dit hele document gebruiken we vette tekst om te verwijzen naar een commando of applicatie en een `monospaced` lettertype om te verwijzen naar specifieke commando's. Protocollen staan vermeld in een normaal lettertype. Dit typografische onderscheid is zinvol omdat bijvoorbeeld ssh zowel een protocol als een commando is.

In de volgende onderdelen behandelen we de methodes uit de vorige paragraaf om een FreeBSD-systeem te beveiligen.

13.3.1. Beveiligen van `root` en medewerkersaccounts.

Om te beginnen: doe geen moeite om medewerkersaccounts te beveiligen als de `root` account niet beveiligd is. Op de meeste systemen heeft de `root` account een wachtwoord. Als eerste moet aangenomen worden dat dit wachtwoord *altijd* gecompromitteerd is. Dit betekent niet dat het wachtwoord verwijderd moet worden. Het wachtwoord is namelijk bijna altijd nodig voor toegang

via het console van de machine. Het betekent wel dat het niet mogelijk gemaakt moet worden om het wachtwoord te gebruiken buiten het console om en mogelijk zelfs niet via het su(1) commando. Pty's moeten bijvoorbeeld gemarkeerd staan als onveilig ("insecure") in het bestand /etc/ttys zodat direct aanmelden met `root` via `telnet` of `rlogin` niet wordt toegestaan. Als andere aanmelddiensten zoals sshd gebruikt worden, dan hoort direct aanmelden via `root` uitgeschakeld staat. Dit kan door het bestand /etc/ssh/sshd_config te bewerken en ervoor te zorgen dat `PermitRootLogin` op `no` staat. Dit moet gebeuren voor iedere methode van toegang - diensten zoals FTP worden vaak over het hoofd gezien. Het direct aanmelden van `root` hoort alleen te mogen via het systeemconsole.

Natuurlijk moet een systeembeheerder de mogelijkheid hebben om `root` te worden. Daarvoor kunnen een paar gaatjes geprikt worden. Maar dan moet ervoor gezorgd worden dat er voor deze gaatjes extra aanmelden met een wachtwoord nodig is. Eén manier om `root` toegankelijk te maken is door het toevoegen van de juiste medewerkersaccounts aan de `wheel` groep (in /etc/group). De medewerkers die lid zijn van de groep `wheel` mogen `su`-en naar `root`. Maak medewerkers nooit "native" lid van de groep `wheel` door ze in de groep `wheel` te plaatsen in /etc/group. Medewerkersaccounts horen lid te zijn van de groep `staff` en horen dan pas toegevoegd te worden aan de groep `wheel` in het bestand /etc/group. Alleen medewerkers die ook echt toegang tot `root` nodig hebben horen in de groep `wheel` geplaatst te worden. Het is ook mogelijk, door een autenticatiemethode als Kerberos te gebruiken, om het bestand .k5login van Kerberos in de `root` account te gebruiken om een ksu(1) naar `root` toe te staan zonder ook maar iemand lid te maken van de groep `wheel`. Dit is misschien wel een betere oplossing, omdat het `wheel`-mechanisme het nog steeds mogelijk maakt voor een inbreker `root` te breken als de inbreker een wachtwoordbestand te pakken heeft gekregen en toegang kan krijgen tot één van de medewerkersaccounts. Hoewel het instellen van het `wheel`-mechanisme beter is dan niets, is het niet per se de meest veilige optie.

Om een account volledig op slot te zetten, dient het commando pw(8) gebruikt te worden:

```
# pw lock staff
```

Dit voorkomt dat de gebruiker zich aanmeldt via enig mechanisme, inclusief ssh(1).

Een andere manier om toegang tot accounts te blokkeren is om het versleutelde wachtwoord door een enkel "*"-karakter te vervangen. Dit karakter zal nooit overeenkomen met het versleutelde wachtwoord en dus gebruikerstoegang blokkeren. Het volgende medewerkersaccount bijvoorbeeld:

```
foobar:R9DT/Fa1/LV9U:1000:1000::0:0:Foo Bar:/home/foobar:/usr/local/bin/tcsh
```

zou veranderd moeten worden in:

```
foobar:*:1000:1000::0:0:Foo Bar:/home/foobar:/usr/local/bin/tcsh
```

Dit voorkomt dat de gebruiker `foobar` zich aanmeldt met conventionele methoden. Deze methode om toegang te beperken werkt niet op sites die Kerberos gebruiken of in situaties waarin de gebruiker met ssh(1) sleutels heeft geïnstalleerd.

Deze beveiligingsmechanismen hebben ook als uitgangspunt dat vanaf een zwaarder beveiligde machine wordt aangemeld op een minder beveiligd systeem. Als een hoofdserver bijvoorbeeld allerlei servers draait, zou het werkstation er geen moeten draaien. Om een werkstation redelijk veilig te laten zijn, dienen er zo min mogelijk servers op te draaien, bij voorkeur zelfs geen en er zou een schermbeveiliging met wachtwoordbeveiliging op moeten draaien. Maar als een aanvaller fysieke toegang heeft tot een werkstation, dan kan hij elke beveiliging die erop is aangebracht omzeilen. Dit probleem dient echt overwogen te worden, net als het feit dat de meeste aanvallen van een afstand plaatsvinden, via het netwerk, door mensen die geen fysieke toegang hebben tot werkstations of servers.

Het gebruik van iets als Kerberos geeft de mogelijkheid om het wachtwoord van de account van een medewerker buiten gebruik te stellen of te wijzigen op één plaats, waarbij het meteen actief is op alle machines waarop die medewerker een account heeft. Als de account van een medewerker gecompromitteerd raakt, moet vooral de mogelijkheid om per direct het wachtwoord voor machines te kunnen aanpassen niet onderschat worden. Met afzonderlijke wachtwoorden kan het veranderen van wachtwoorden op N systemen een puinhoop worden. Met Kerberos kunnen ook wachtwoordrestricties opgelegd worden: het is niet alleen mogelijk om een Kerberos "ticket" na een bepaalde tijd te laten verlopen, maar het Kerberos systeem kan afdwingen dat de gebruiker na een bepaalde tijd een nieuw wachtwoord kiest (na bijvoorbeeld een maand).

13.3.2. Beveiligen van `root` - servers onder `root` en suid-/sgid-binaire bestanden

Een voorzichtige systeembeheerder draait alleen die servers die nodig zijn, niets meer, niets minder. Bedenk dat servers van derde partijen vaak de meeste neiging hebben tot het vertonen van bugs. Zo staat bijvoorbeeld het draaien van een oude versie van imapd of popper gelijk aan het weggeven van de `root` account aan de hele wereld. Draai nooit een server die niet zorgvuldig is onderzocht. Veel servers hoeven niet te draaien als `root`. Zo kunnen de ntalk, comsat en finger daemons bijvoorbeeld draaien in speciale gebruikerszandbakken ("*sandboxes*"). Een zandbak is niet perfect, tenzij er heel veel moeite gedaan wordt, maar de meerlagenbenadering blijft bestaan: als iemand via een server die in een zandbak draait weet in te breken, dan moeten ze eerst nog uit de zandbak komen. Hoe groter het aantal lagen is waar een inbreker doorheen moet, hoe kleiner de kans op succes is. `root` gaten zijn historisch gezien aanwezig geweest in vrijwel iedere server die ooit als `root` gedraaid heeft, inclusief de basisservers van een systeem. Op een machine waarop mensen alleen aanmelden via sshd en nooit via telnetd of rshd of rlogind dienen die servers uitgeschakeld te worden!

FreeBSD draait ntalkd, comsat en finger tegenwoordig standaard in een zandbak. Een ander programma dat misschien beter in een zandbak kan draaien is named(8). In /etc/defaults/rc.conf staat als commentaar welke parameters er nodig zijn om named in een zandbak te draaien. Afhankelijk van of het een nieuwe systeeminstallatie of het bijwerken van een bestaand systeem betreft, worden de speciale gebruikersaccounts die bij die zandbakken horen misschien niet geïnstalleerd. Een voorzichtige systeembeheerder onderzoekt en implementeert zandbakken voor servers waar dat ook maar mogelijk is.

Er zijn een aantal diensten die vooral niet in een zandbak draaien: sendmail, popper, imapd, ftpd en andere. Voor sommige servers zijn alternatieven, maar dat kost misschien meer tijd dan er te besteden is (gemak dient de mens). Het kan voorkomen dat deze servers als `root` moeten draaien en

dat er vertrouwd moet worden op andere mechanismen om een inbraak via die servers te detecteren.

De andere grote mogelijkheid voor `root` gaten in een systeem zijn de suid-root en sgid-binaire bestanden die geïnstalleerd zijn op een systeem. Veel van die bestanden, zoals rlogin, staan in /bin, /sbin, /usr/bin of /usr/sbin. Hoewel het niet 100% veilig is, mag aangenomen worden dat de suid- en sgid-binaire bestanden van een standaardsysteem redelijk veilig zijn. Toch worden er nog wel eens `root` gaten gevonden in deze bestanden. Zo is er in 1998 een `root` gat gevonden in `Xlib` waardoor xterm (die normaliter suid is) kwetsbaar bleek. Een voorzichtige systeembeheerder kiest voor "better to be safe than sorry" door de suid-bestanden die alleen medewerkers hoeven uit te voeren aan een speciale groep toe te wijzen en de suid-bestanden die niemand gebruikt te lozen (`chmod 000`). Een server zonder monitor heeft normaal gezien xterm niet nodig. Sgid-bestanden kunnen bijna net zo gevaarlijk zijn. Als een inbreker een sgid-kmem stuk kan krijgen, dan kan hij wellicht /dev/kmem lezen en dus het gecodeerde wachtwoordbestand, waardoor mogelijk ieder account met een wachtwoord besmet is. Een inbreker toegang tot de groep `kmem` kan krijgen, zou bijvoorbeeld mee kunnen kijken met de toetsaanslagen die ingegeven worden via de pty's, inclusief die pty's die gebruikt worden door gebruikers die via beveiligde methodes aanmelden. Een inbreker die toegang krijgt tot de groep `tty` kan naar bijna alle tty's van gebruikers schrijven. Als een gebruiker een terminalprogramma of een terminalemulator met een toetsenbordsimulatieoptie draait, dan kan de inbreker in potentie een gegevensstroom genereren die ervoor zorgt dat de terminal van de gebruiker een commando echot, dat dan wordt uitgevoerd door die gebruiker.

13.3.3. Beveiligen van gebruikersaccounts

Gebruikersaccounts zijn gewoonlijk het meest lastig om te beveiligen. Hoewel er allerlei draconische maatregelen genomen kunnen worden met betrekking tot de medewerkers en hun wachtwoorden "weggesterd" kunnen worden, gaat dat waarschijnlijk niet lukken met de gewone gebruikersaccounts. Als er toch voldoende vrijheid is, dan prijst de beheerder zich gelukkig en is het misschien toch mogelijk de accounts voldoende te beveiligen. Als die vrijheid er niet is, dan moeten die accounts gewoon netter gemonitord worden. Het gebruik van ssh en Kerberos voor gebruikersaccounts is problematischer vanwege het extra beheer en de ondersteuning, maar nog steeds een prima oplossing in vergelijking met een versleuteld wachtwoordbestand.

13.3.4. Beveiligen van het wachtwoordbestand

De enige echte oplossing is zoveel mogelijk wachtwoorden wegsterren en ssh of Kerberos gebruiken voor toegang tot die accounts. Hoewel een gecodeerd wachtwoordbestand (/etc/spwd.db) alleen gelezen kan worden door `root`, is het wel mogelijk dat een inbreker leestoegang krijgt tot dat bestand zonder dat de aanvaller root-schrijftoegang krijgt.

Beveiligingsscripts moeten altijd controleren op en rapporteren over wijzigingen in het wachtwoordbestand (zie ook Bestandsintegriteit Controleren hieronder).

13.3.5. Beveiligen van de kern van de kernel, ruwe apparaten en bestandssystemen

Als een aanvaller toegang krijgt tot `root` dan kan hij ongeveer alles, maar er zijn een paar slimmigheidjes. Zo hebben bijvoorbeeld de meeste moderne kernels een ingebouwd

pakketsnuffelstuurprogramma ("packet sniffing"). Bij FreeBSD is dat het bpf apparaat. Een inbreker zal in het algemeen proberen een pakketsnuffelaar te draaien op een gecompromitteerde machine. De inbreker hoeft deze mogelijkheid niet te hebben en bij de meeste systemen is het niet verplicht het bpf apparaat mee te compileren.

Maar zelfs als het bpf apparaat is uitgeschakeld, dan zijn er nog /dev/mem en /dev/kmem. De inbreker kan namelijk nog schrijven naar ruwe schrijfapparaten. En er is ook nog een optie in de kernel die modulelader ("module loader") heet, kldload(8). Een ondernemende inbreker kan een KLD-module gebruiken om zijn eigen bpf-apparaat of een ander snuffelapparaat te installeren in een draaiende kernel. Om deze problemen te voorkomen, moet de kernel op een hoger veiligheidsniveau draaien, ten minste securelevel 1.

Het veiligheidsniveau van de kernel kan op een aantal manieren worden ingesteld. De eenvoudigste manier om het veiligheidsniveau van een draaiende kernel te verhogen is met `sysctl` op de kernelvariabele `kern.securelevel`:

```
# sysctl kern.securelevel=1
```

Standaard start de kernel van FreeBSD op met een veiligheidsniveau van -1. Het veiligheidsniveau blijft -1 tenzij het is veranderd, òfwel door de beheerder òfwel door init(8) vanwege een instelling in de opstartscripts. Het veiligheidsniveau kan tijdens het opstarten van het systeem verhoogd worden door de variabele `kern_securelevel_enable` op `YES` te zetten in het bestand /etc/rc.conf, en de waarde van de variabele `kern_securelevel` op het gewenste veiligheidsniveau in te stellen.

Het standaard veiligheidsniveau van een FreeBSD-systeem direct nadat de opstartscripts zijn uitgevoerd is -1. Dit wordt "onveilige modus" genoemd omdat de onveranderlijke bestandsvlag uitgezet kan worden, er van/naar alle apparaten mag worden gelezen en geschreven, enzovoorts.

Als eenmaal het veiligheidsniveau op 1 of een hogere waarde is ingesteld, worden de alleen-toevoegen en onveranderlijke bestanden gehonoreerd, deze kunnen niet worden uitgezet, en wordt toegang tot rauwe apparaten ontzegd. Hogere niveaus beperken nog meer bewerkingen. Lees, voor een volledige beschrijving van het effect van de verschillende veiligheidsniveaus, de handleidingpagina security(7).



Het ophogen van het veiligheidsniveau naar 1 of hoger kan enkele problemen met X11 (toegang tot /dev/io zal worden geblokkeerd), of met de installatie van FreeBSD wanneer die vanaf de broncode is gebouwd (het gedeelte `installword` van het proces moet tijdelijk de alleen-toevoegen en onveranderlijke vlaggen van sommige bestanden uitzetten), en met enkele andere gevallen veroorzaken. Soms, zoals het geval is met X11, is het mogelijk om dit te omzeilen door xdm(1) behoorlijk vroeg in het opstartproces te starten, wanneer het veiligheidsniveau nog laag genoeg is. Omzeilmethoden zoals deze zijn misschien niet voor alle veiligheidsniveaus of voor alle beperkingen die ze opleggen mogelijk. Wat vooruit plannen is een goed idee. Het is belangrijk om de beperkingen die door elk veiligheidsniveau worden opgelegd te begrijpen omdat ze het gebruiksgemak van het systeem sterk verminderen. Het vergemakkelijkt ook het kiezen van eens standaardinstelling en voorkomt allerlei verassingen.

Als het veiligheidsniveau van de kernel naar 1 of hoger wordt verhoogd, kan het nuttig zijn om de vlag `schg` aan te zetten voor kritieke opstartprogramma's, mappen, en scriptbestanden (i.e., alles dat gedraaid wordt tot het punt waar het veiligheidsniveau wordt ingesteld). Dit kan overdreven zijn, en het bijwerken van het systeem is veel moeilijker wanneer het op een hoog veiligheidsniveau werkt. Een minder beperkend compromis is om het systeem op een hoger veiligheidsniveau te draaien maar het aanzetten van de vlag `schg` voor elk systeembestand en -map onder de zon over te slaan. Een andere mogelijkheid is om / en /usr simpelweg als alleen-lezen aan te koppelen. Het dient opgemerkt te worden dat het te draconisch zijn over wat is toegestaan het belangrijke detecteren van een inbraak kan verhinderen.

13.3.6. Bestandsintegriteit controleren: binaire bestanden, instellingenbestanden, enzovoort

Als puntje bij paaltje komt kan de kern van een systeem maar tot een bepaald punt beveiligd worden zonder dat het minder prettig werken wordt. Zo werk het zetten van de `schg` bit met `chflags` op de meeste bestanden in / en /usr waarschijnlijk averechts, omdat, hoewel de bestanden beschermd zijn, ook het venster waarin detectie plaats kan vinden is gesloten. De laatste laag van beveiliging is waarschijnlijk de meest belangrijke: detectie. Alle overige beveiliging is vrijwel waardeloos (of nog erger: geeft een vals gevoel van beveiliging) als een mogelijke inbraak niet gedetecteerd kan worden. Een belangrijk doel van het meerlagenmodel is het vertragen van een aanvaller, nog meer dan hem te stoppen, om hem op heterdaad te kunnen betrappen.

De beste manier om te zoeken naar een inbraak is zoeken naar gewijzigde, ontbrekende of onverwachte bestanden. De beste manier om te zoeken naar gewijzigde bestanden is vanaf een ander (vaak gecentraliseerd) systeem met beperkte toegang. Met zelfgeschreven scripts op dat extra beveiligde systeem met beperkte toegang is een beheerder vrijwel onzichtbaar voor mogelijke aanvallers en dat is belangrijk. Om het nut te maximaliseren moeten in het algemeen dat systeem met beperkte toegang best veel rechten gegeven worden op de andere machines in het netwerk, vaak via een alleen-lezen NFS-export van de andere machines naar het systeem met beperkte toegang of door ssh sleutelparen in te stellen om het systeem met beperkte toegang een ssh verbinding te laten maken met de andere machines. Buiten het netwerkverkeer, is NFS de minst zichtbare methode. Hierdoor kunnen de bestandssystemen op alle cliëntmachines vrijwel ongezien gemonitord worden. Als de server met beperkte toegang verbonden is met de cliëntmachines via een switch, dan is de NFS-methode vaak de beste keus. Als de server met beperkte toegang met de andere machines is verbonden via een hub of door meerdere routers, dan is de NFS-methode wellicht niet veilig genoeg (vanuit een netwerk standpunt) en kan beter ssh gebruikt worden, ondanks de audit-sporen die ssh achterlaat.

Als de machine met beperkte toegang eenmaal minstens leestoegang heeft tot een cliëntsysteem dat het moet gaan monitoren, dan moeten scripts gemaakt worden om dat monitoren ook echt uit te voeren. Uitgaande van een NFS-koppeling, kunnen de scripts gebruik maken van eenvoudige systeem hulpprogramma's als find(1) en md5(1). We adviseren minstens één keer per dag een md5 te maken van alle bestanden op de cliëntmachine en van instellingenbestanden als in /etc en /usr/local/etc zelfs vaker. Als er verschillen worden aangetroffen ten opzichte van de basis md5 informatie op het systeem met beperkte toegang, dan hoort het script te gillen om een beheerder die het moet gaan uitzoeken. Een goed beveiligingsscript controleert ook op onverwachte suid-bestanden en op nieuwe en verwijderde bestanden op systeempartities als / en /usr.

Als ssh in plaats van NFS wordt gebruikt, dan is het schrijven van het script lastiger. Dan moeten de scripts met `scp` naar de cliënt verplaatst worden om ze uit te voeren, waardoor ze zichtbaar worden. Voor de veiligheid dienen ook de binaire bestanden die het script gebruikt, zoals find(1), gekopieerd te worden. De ssh-cliënt op de cliënt zou al gecompromitteerd kunnen zijn. Het is misschien noodzakelijk ssh te gebruiken over onveilige verbindingen, maar dat maakt alles een stuk lastiger.

Een goed beveiligingsscript voert ook controles uit op de instellingenbestanden van gebruikers en medewerkers: .rhosts, .shosts, .ssh/authorized_keys, enzovoort. Dat zijn bestanden die buiten het bereik van de `MD5`-controle vallen.

Als gebruikers veel schijfruimte hebben, dan kan het te lang duren om alle bestanden op deze partitie te controleren. In dat geval is het verstandig de koppelvlaggen zo in te stellen dat suid-binaire bestanden op die partities niet zijn toegestaan. Zie daarvoor de optie `nosuid` (zie mount(8)). Die partities moeten wel toch nog minstens eens per week doorzocht worden, omdat het doel van deze beveiligingslaag het ontdekken van een inbraakpoging is, of die nu succesvol is of niet.

Procesverantwoording (zie accton(8)) kost relatief gezien weinig en kan bijdragen aan een evaluatie mechanisme voor na inbraken. Het is erg handig om uit te zoeken hoe iemand precies heeft ingebroken op het systeem, mits het bestand nog onbeschadigd is na de inbraak.

Tenslotte horen beveiligingsscripts de logboekbestanden te verwerken en de logboekbestanden zelf horen zo veilig mogelijk tot stand te komen. "remote syslog" kan erg zinvol zijn. Een aanvaller zal proberen zijn sporen uit te wissen en logboekbestanden zijn van groot belang voor een systeembeheerder als het gaat om uitzoeken wanneer en hoe er is ingebroken. Een manier om logboekbestanden veilig te stellen is door het systeemconsole via een seriële poort aan te sluiten op een veilige machine en zo informatie te verzamelen.

13.3.7. Paranoia

Een beetje paranoia is niet verkeerd. Eigenlijk kan de systeembeheerder zoveel beveiligingsopties inschakelen als hij wil, als deze maar geen impact hebben op het gebruiksgemak en de beveiligingsopties die *wel* impact hebben op het gebruiksgemak kunnen ingeschakeld worden als daar zorgvuldig mee wordt omgegaan. Nog belangrijker is misschien dat er een juiste combinatie wordt gevonden. Als de aanbevelingen uit dit document woord voor woord worden opgevolgd, dan worden daarmee de methodes aan een toekomstige aanvaller verraden, die ook toegang heeft tot dit document.

13.3.8. Ontzeggen van Dienst aanvallen

In deze paragraaf worden Ontzeggen van Dienst aanvallen ("Denial of Service" of DoS) behandeld. Een DoS-aanval wordt meestal uitgevoerd als pakketaanval. Hoewel er weinig gedaan kan worden tegen de huidige aanvallen met gefingeerde pakketten die een netwerk kunnen verzadigen, kan de schade geminimaliseerd worden door ervoor te zorgen dat servers er niet door plat gaan door:

1. Limiteren van server forks.
2. Limiteren van springplank ("springboard") aanvallen (ICMP response aanvallen, ping broadcast, etc.).

3. De Kernel Route Cache overloaden.

Een veelvoorkomende DoS-aanval is om een server aan te vallen door het zoveel kindprocessen aan te laten maken dat het hostsysteem uiteindelijk geen bestandsdescriptors, geheugen enzovoort meer heeft en het dan opgeeft. inetd (zie inetd(8)) kent een aantal instellingen om dit type aanval af te zwakken. Hoewel het mogelijk is ervoor te zorgen dat een machine niet plat gaat, is het in het algemeen niet mogelijk te voorkomen dat de dienstverlening door de aanval wordt verstoord. Meer is te lezen in de handleiding van inetd en het advies is in het bijzonder aandacht aan de `-c`, `-C` en `-R` opties te besteden. Aanvallen met gefingeerde IP adressen omzeilen de `-C` optie naar inetd, dus in het algemeen moet een combinatie van opties gebruikt worden. Sommige op zichzelf staande servers hebben parameters waarmee het aantal forks gelimiteerd kan worden.

Sendmail heeft de optie `-OMaxDaemonChildren` die veel beter blijkt te werken dan het gebruik van de opties van Sendmail waarmee de werklast gelimiteerd kan worden. De parameter `MaxDaemonChildren` moet zodanig ingesteld worden dat als sendmail start; deze hoog genoeg is om de te verwachten belasting aan te kunnen, maar niet zo hoog is dat de computer het aantal instanties van Sendmails niet aankan zonder plat te gaan. Het is ook verstandig om Sendmail in de wachtrijmodus (`-ODeliveryMode=queued`) te draaien en de daemon (`sendmail -bd`) los te koppelen van de verwerking van de wachtrij (`sendmail -q15m`). Als de verwerking van wachtrij real-time moet, kunnen de tussenpozen voor verwerking verkort worden door deze bijvoorbeeld op `-q1m` in te stellen, maar dan is een redelijke instelling van `MaxDaemonChildren` van belang om *die* Sendmail te beschermen tegen trapsgewijze fouten.

Syslogd kan direct aangevallen worden en het is sterk aan te raden de `-s` optie te gebruiken waar dat ook maar mogelijk is en anders de `-a` optie.

Er dient voorzichtig omgesprongen te worden met diensten die terugverbinden zoals TCP Wrapper's reverse-identd die direct aangevallen kan worden. In het algemeen is het hierom onverstandig gebruik te maken van de reverse-ident optie van TCP Wrapper.

Het is een goed idee om interne diensten af te schermen voor toegang van buitenaf door ze te firewallen op de routers aan de rand van een netwerk ("border routers"). Dit heeft als achtergrond dat verzadigingsaanvallen voorkomen van buiten het LAN voorkomen kunnen worden. Daarmee wordt geen aanval op `root` via het netwerk en die diensten daaraan voorkomen. Er dient altijd een exclusieve firewall te zijn, dat wil zeggen "firewall alles *behalve* poorten A, B, C, D en M-Z". Zo worden alle lage poorten gefirewalled behalve die voor specifieke diensten als named (als er een primary is voor een zone), ntalkd, sendmail en andere diensten die vanaf Internet toegankelijk moeten zijn. Als de firewall andersom wordt ingesteld, als een inclusieve of tolerante firewall, dan is de kans groot dat er wordt vergeten een aantal diensten af te "sluiten" of dat er een nieuwe interne dienst wordt toegevoegd en de firewall niet wordt bijgewerkt. Er kan nog steeds voor gekozen worden de hoge poorten open te zetten, zodat een tolerante situatie ontstaat, zonder de lage poorten open te stellen. FreeBSD biedt ook de mogelijkheid een reeks poortnummers die gebruikt worden voor dynamische verbindingen in te stellen via de verscheidene `net.inet.ip.portrange``sysctl`s (`sysctl -a | fgrep portrange`), waardoor ook de complexiteit van de firewall instellingen kan vereenvoudigen. Zo kan bijvoorbeeld een normaal begin tot eindbereik ingesteld worden van 4000 tot 5000 en een hoog poortbereik van 49152 tot 65535. Daarna kan alles onder 4000 op de firewall geblokkeerd worden (met uitzondering van bepaalde poorten die vanaf Internet bereikbaar moeten zijn natuurlijk).

Een andere veelvoorkomende DoS-aanval is de springplankaanval: een server zo aanvallen dat de respons van die server de server zelf, het lokale netwerk of een andere machine overbelast. De meest voorkomende aanval van dit type is de *ICMP ping broadcast aanval*. De aanvaller fingeert ping-pakketten die naar het broadcast-adres van het LAN worden gezonden met als bron het IP-adres van de machine die hij eigenlijk aan wil vallen. Als de routers aan de rand van het netwerk niet zijn ingesteld om een ping-pakketten aan een broadcast-adres te blokkeren, dan kan het LAN genoeg antwoorden produceren om de verbinding van het slachtoffer (het gefingeerde bronadres) te verzadigen, zeker als de aanvaller hetzelfde doet met tientallen andere netwerken. Broadcastaanvallen met een volume van meer dan 120 megabit zijn al voorgekomen. Een tweede springplankaanval is er een tegen het ICMP-foutmeldingssysteem. Door een pakket te maken waarop een ICMP-foutmelding komt, kan een aanvaller de inkomende verbinding van een server verzadigen en de uitgaande verbinding laten verzadigen met ICMP-foutmeldingen. Dit type aanval kan een server ook laten crashen door te zorgen dat het geheugen ervan vol zit, zeker als de server de ICMP-antwoorden niet zo snel kwijt kan als dat het ze genereert. Gebruik de sysctl-variabele `net.inet.icmp.icmplim` om deze aanvallen te beperken. De laatste belangrijke klasse springplankaanvallen hangt samen met een aantal interne diensten van inetd zoals de UDP-echodienst. Een aanvaller fingeert eenvoudigweg een UDP-pakket met als bronadres de echopoort van Server A en als bestemming de echopoort van Server B, waar Server A en B allebei op een LAN staan. Die twee servers gaan dat pakket dan heen en weer kaatsen. Een aanvaller kan beide servers overbelasten door een aantal van deze pakketten te injecteren. Soortgelijke problemen kunnen ontstaan met de poort chargen. Een competente systeembeheerder zal al deze interne inetd testdiensten uitschakelen.

Gefingeerde pakketten kunnen ook gebruikt worden om de kernel route cache te overbelasten. Raadpleeg daarvoor de `net.inet.ip.rtexpire`, `rtminexpire` en `rtmaxcache``sysctl` parameters. Een aanval met gefingeerde pakketten met een willekeurig bron-IP zorgt ervoor dat de kernel een tijdelijke gecachede route maakt in de routetabel, die uitgelezen kan worden met `netstat -rna | fgrep W3`. Deze routes hebben een levensduur van ongeveer 1600 seconden. Als de kernel merkt dat de gecachede routetabel te groot is geworden, dan wordt `rtexpire` dynamisch verkleind, maar deze waarde wordt nooit lager dan `rtminexpire`. Er zijn twee problemen:

1. De kernel reageert niet snel genoeg als een laag belaste server wordt aangevallen.
2. `rtminexpire` is niet laag genoeg om de kernel de aanval te laten overleven.

Als servers verbonden zijn met het Internet via een E3 of sneller, dan is het verstandig om handmatig `rtexpire` en `rtminexpire` aan te passen via sysctl(8). Als de een van de parameters op nul wordt gezet, dan crasht de machine. Het instellen van beide waarden op 2 seconden is voldoende om de routetabel tegen een aanval te beschermen.

13.3.9. Aandachtspunten voor toegang met Kerberos en SSH

Er zijn een aantal aandachtspunten die in acht genomen moeten worden als Kerberos of ssh gebruikt worden. Kerberos 5 is een prima autenticatieprotocol, maar er zitten bugs in de Kerberos-versies van telnet en rlogin waardoor ze niet geschikt zijn voor binair verkeer. Kerberos codeert standaard de sessie niet, tenzij de optie `-x` wordt gebruikt. ssh codeert standaard wel alles.

Ssh werkt prima, maar het stuurt coderingssleutels standaard door. Dit betekent dat als gegeven een veilig werkstation met sleutels die toegang geven tot de rest van het systeem en ssh wordt

gebruikt om verbinding te maken met een onveilige machine, die sleutels gebruikt kunnen worden. De sleutels zelf zijn niet bekend, maar ssh stelt een doorstuurpoort in zolang als een gebruikers aangemeld blijft. Als de aanvaller `root`toegang heeft op de onveilige machine, dan kan hij die poort gebruiken om toegang te krijgen tot alle machines waar de sleutels van de gebruiker toegang toe geven.

Het advies is ssh in combinatie met Kerberos te gebruiken voor het aanmelden door medewerkers wanneer dat ook maar mogelijk is. Ssh kan gecompileerd worden met Kerberos-ondersteuning. Dit vermindert de kans op blootstelling van ssh-sleutels en beschermt tegelijkertijd de wachtwoorden met Kerberos. Ssh-sleutels zouden alleen gebruikt moeten worden voor geautomatiseerde taken vanaf veilige machines (iets waar Kerberos ongeschikt voor is). Het advies is om het doorsturen van sleutels uit te schakelen in de ssh-instellingen of om de `from=IP/DOMAIN` optie te gebruiken die ssh in staat stelt het bestand authorized_keys te gebruiken om de sleutel alleen bruikbaar te maken voor entiteiten die zich aanmelden vanaf vooraf aangewezen machines.

13.4. DES, Blowfish, MD5, SHA256, SHA512 en crypt

Iedere gebruiker op een UNIX® systeem heeft een wachtwoord bij zijn account. Het lijkt voor de hand liggend dat deze wachtwoorden alleen bekend horen te zijn bij de gebruiker en het eigenlijke besturingssysteem. Om deze wachtwoorden geheim te houden, zijn ze gecodeerd in een "eenweg hash" ("one-way hash"), wat betekent dat ze eenvoudig gecodeerd kunnen worden maar niet gedecodeerd. Met andere woorden, wat net gesteld werd is helemaal niet waar: het besturingssysteem kent het *echte* wachtwoord niet. De enige manier om een wachtwoord in "platte tekst" te verkrijgen, is door er met brute kracht naar te zoeken in alle mogelijke wachtwoorden.

Helaas was DES, de Data Encryption Standard, de enige manier om wachtwoorden veilig te coderen toen UNIX® ontstond. Dit was geen probleem voor gebruikers in de VS, maar omdat de broncode van DES niet geëxporteerd mocht worden moest FreeBSD een manier vinden om zowel te gehoorzamen aan de wetten van de Verenigde Staten als aansluiting te houden bij alle andere varianten van UNIX® die nog steeds DES gebruikten.

De oplossing werd gevonden in het splitsen van de coderingsbibliotheken zodat gebruikers in de Verenigde Staten de DES-bibliotheken konden installeren en gebruiken en internationale gebruikers een coderingsmethode konden gebruiken die geëxporteerd mocht worden. Zo is het gekomen dat FreeBSD MD5 is gaan gebruiken als coderingsmethode. Van MD5 wordt aangenomen dat het veiliger is dan DES, dus de mogelijkheid om DES te installeren is vooral beschikbaar om aansluiting te kunnen houden.

13.4.1. Het crypt-mechanisme herkennen

Op dit moment ondersteunt de bibliotheek DES, MD5, Blowfish, SHA256 en SHA512 hashfuncties. Standaard gebruikt FreeBSD 9.1 en nieuwer SHA512 om wachtwoorden te coderen. Oudere versies gebruiken standaard MD5.

Het is vrij makkelijk om uit te vinden welke coderingsmethode FreeBSD op een bepaald moment gebruikt. De gecodeerde wachtwoorden in /etc/master.passwd bekijken is een manier. Wachtwoorden die gecodeerd zijn met MD5 zijn langer dan wanneer ze gecodeerd zijn met DES-hash. Daarnaast beginnen ze met de karakters `$1$`. Wachtwoorden die beginnen met `$2a$` zijn

gecodeerd met de Blowfish hashfunctie. DES-wachtwoordstrings hebben geen bijzondere kenmerken, maar ze zijn korter dan MD5 wachtwoorden en gecodeerd in een 64-karakter alfabet waar geen `$` karakter in zit. Een relatief korte string die niet begint met een dollar teken is dus waarschijnlijk een DES-wachtwoord. Zowel SHA256 als SHA512 beginnen met de tekens `$6$`.

Het wachtwoordformaat voor nieuwe wachtwoorden wordt ingesteld met de `passwd_format` aanmeldinstelling in /etc/login.conf waar `des`, `md5`, `blf`, `sha256` of `sha512` in mag staan. Zie de login.conf(5) handleiding voor meer informatie over aanmeldinstellingen.

13.5. Eenmalige wachtwoorden

Standaard biedt FreeBSD ondersteuning voor OPIE (Eenmalige Wachtwoorden in Alles - "One-time Passwords In Everything"), wat standaard een MD5-hash gebruikt.

Hier worden drie verschillende soorten wachtwoorden besproken. De eerste is het normale UNIX® of Kerberos wachtwoord. Dit heet het "UNIX® wachtwoord". Het tweede type is een eenmalig wachtwoord dat wordt gemaakt met het OPIE-programma opiekey(1) en dat wordt geaccepteerd door opiepasswd(1) en de aanmeldprocedure. Dit heet het "eenmalige wachtwoord". Het laatste type wachtwoord is het wachtwoord dat wordt opgegeven aan het programma `opiekey` (en soms aan het programma `opiepasswd`) dat gebruikt wordt om eenmalige wachtwoorden te maken. Dit type heet "geheim wachtwoord" of gewoon een "wachtwoord" zonder toevoeging.

Het geheime wachtwoord heeft niets te maken met het UNIX® wachtwoord; ze kunnen hetzelfde zijn, dat wordt afgeraden. OPIE geheime wachtwoorden kennen niet de beperking van 8 karakters zoals de oude UNIX® wachtwoorden. Ze mogen onbeperkt lang zijn. Wachtwoorden van een zes of zeven woorden lange zin zijn niet ongewoon. Voor het overgrote deel werkt het OPIE-systeem volledig onafhankelijk van het UNIX® wachtwoordsysteem.

Buiten het wachtwoord zijn er nog twee stukjes gegevens die van belang zijn voor OPIE. Het eerste wordt "zaad" ("seed") of "sleutel" ("key") genoemd en bestaat uit twee letters en vijf cijfers. Het tweede stukje gegevens heet de "iteratieteller", een nummer tussen 1 en 100. OPIE maakt een eenmalig wachtwoord door het zaad en het geheime wachtwoord aaneen te schakelen en daarop het door de iteratieteller aangegeven keren MD5-hash toe te passen. Daarna wordt het resultaat omgezet in zes korte Engelse woorden. Deze zes woorden zijn een eenmalige wachtwoord. Het autenticatiesysteem (hoofdzakelijk PAM) houdt bij welk eenmalig wachtwoord het laatst is gebruikt en de gebruiker wordt geautenticeerd als de hash van het door de gebruiker ingegeven wachtwoord gelijk is aan het vorige wachtwoord. Omdat er een eenweg hash wordt gebruikt, is het onmogelijk om toekomstige eenmalige wachtwoorden te maken als iemand toch een eenmalig wachtwoord heeft afgevangen. De iteratieteller wordt verlaagd na iedere succesvolle aanmelding om de gebruiker en het aanmeldprogramma synchroon te houden. Als de iteratieteller op 1 staat, moet OPIE opnieuw ingesteld worden.

Er zijn enkele programma's bij ieder systeem betrokken die hieronder worden besproken. Het programma `opiekey` heeft een iteratieteller, zaad en een geheim wachtwoord nodig en maakt dan een eenmalig wachtwoord of een lijst van opeenvolgende eenmalige wachtwoorden. Het programma `opiepasswd` wordt gebruikt om OPIE te initialiseren en om wachtwoorden, iteratietellers en zaad te wijzigen. Het accepteert zowel wachtwoordzinnen als een iteratieteller, zaad en een eenmalig wachtwoord. Het programma `opieinfo` bekijkt de relevante bestanden waarin de

eigenschappen staan (/etc/opiekeys) en toont de huidige iteratieteller en zaad van de gebruiker die het commando uitvoert.

Nu worden vier verschillende acties besproken. Bij de eerste wordt `opiepasswd` gebruikt in een beveiligde verbinding om voor het eerst eenmalige wachtwoorden in te stellen of om een wachtwoord of zaad aan te passen. Bij de tweede wordt `opiepasswd` gebruikt over een onbeveiligde verbinding samen met `opiekey` over een beveiligde verbinding om hetzelfde te bereiken. In een derde scenario wordt `opiekey` gebruikt om aan te melden over een onveilige verbinding. Het vierde scenario behandelt het gebruik van `opiekey` om een aantal sleutels aan te maken die opgeschreven of afgedrukt kunnen worden, zodat ze meegenomen kunnen worden naar een plaats van waar geen enkele veilige verbinding opgezet kan worden.

13.5.1. Veilige verbinding initialiseren

Gebruik het commando `opiepasswd` om OPIE voor de eerste keer te initialiseren:

```
% opiepasswd -c
[grimreaper] ~ $ opiepasswd -f -c
Adding unfurl:
Only use this method from the console; NEVER from remote. If you are using
telnet, xterm, or a dial-in, type ^C now or exit with no password.
Then run opiepasswd without the -c parameter.
Using MD5 to compute responses.
Enter new secret pass phrase:
Again new secret pass phrase:

ID unfurl OTP key is 499 to4268
MOS MALL GOAT ARM AVID COED
```

Als `Enter new secret pass phrase:` of `Enter secret password:` op het scherm verschijnt, dient een wachtwoord of wachtwoordzin ingevoerd te worden. Dit is dus niet het aanmeldwachtwoord is, maar dit wordt gebruikt om eenmalige wachtwoorden te maken. De "ID" regel geeft de parameters van het verzoek weer: de aanmeldnaam, de iteratieteller en zaad. Bij het aanmelden kent het systeem deze parameters en worden deze weergegeven zodat ze niet onthouden hoeven te worden. Op de laatste regel staat het eenmalige wachtwoord dat overeenkomt met die parameters en het geheime wachtwoord. Als de gebruiker direct opnieuw zou aanmelden, zou hij dat eenmalige wachtwoord moeten gebruiken.

13.5.2. Onveilige verbinding initialiseren

Om een wachtwoord te initialiseren of te wijzigen over een onveilige verbinding, moet er al ergens een veilige verbinding bestaan waar de gebruiker `opiekey` kan uitvoeren. Dit kan een shellprompt zijn op een machine die vertrouwd wordt. De gebruiker moet ook een iteratieteller verzinnen (100 is wellicht een prima getal) en een eigen zaad bedenken of er een laten fabriceren. Over de onveilige verbinding (naar de machine die de gebruiker wil initialiseren) wordt het commando `opiepasswd` gebruikt:

```
% opiepasswd
```

```
Updating unfurl:
You need the response from an OTP generator.
Old secret pass phrase:
    otp-md5 498 to4268 ext
    Response: GAME GAG WELT OUT DOWN CHAT
New secret pass phrase:
    otp-md5 499 to4269
    Response: LINE PAP MILK NELL BUOY TROY

ID mark OTP key is 499 gr4269
LINE PAP MILK NELL BUOY TROY
```

Druk op `Return` om het standaardzaad te accepteren. Voor een toegangswachtwoord wordt ingevoerd, dient eerst gewisseld te worden naar de veilige verbinding en dienen dezelfde parameters ingegeven te worden:

```
% opiekey 498 to4268
Using the MD5 algorithm to compute response.
Reminder: Don't use opiekey from telnet or dial-in sessions.
Enter secret pass phrase:
GAME GAG WELT OUT DOWN CHAT
```

In de onveilige verbinding wordt nu het eenmalige wachtwoord in het relevante programma gekopieerd.

13.5.3. Een enkel eenmalig wachtwoord maken

Als OPIE eenmaal is ingesteld staat er bij het aanmelden iets als het volgende:

```
% telnet example.com
Trying 10.0.0.1...
Connected to example.com
Escape character is '^]'.

FreeBSD/i386 (example.com) (ttypa)

login: <gebruikersnaam>
otp-md5 498 gr4269 ext
Password:
```

NB: de OPIE-prompt heeft een handige optie (die hier niet te zien is): als er op `Return` wordt gedrukt bij de wachtwoordregel, wordt de echo aangezet, zodat de invoer zichtbaar is. Dit is erg handig als er met de hand een wachtwoord wordt ingegeven, zoals wanneer het wordt ingevoerd vanaf een afdruk.

Nu moet het eenmalige wachtwoord gemaakt worden om het aanmeldprompt mee te antwoorden.

Dit moet gedaan worden op een vertrouwd systeem waarop `opiekey` beschikbaar is. Er zijn ook versies voor MS-DOS®, Windows® en Mac OS®. Voor het commando moet zowel de iteratieteller als het zaad ingeven worden op de commandoregel. Deze kan zo overgenomen worden vanaf het aanmeldprompt op de machine waarop de gebruiker zich wil aanmelden.

Op het vertrouwde systeem:

```
% opiekey 498 to4268
Using the MD5 algorithm to compute response.
Reminder: Don't use opiekey from telnet or dial-in sessions.
Enter secret pass phrase:
GAME GAG WELT OUT DOWN CHAT
```

Nu het eenmalige wachtwoord er is, kan het aanmelden doorgang vinden.

13.5.4. Meerdere eenmalige wachtwoorden maken

Soms moet een gebruiker ergens naar toe gaan waar er geen toegang is tot een vertrouwde machine of een beveiligde verbinding. In dat geval is het mogelijk om met het commando `opiekey` een aantal eenmalige wachtwoorden te maken om uit te printen en mee te nemen:

```
% opiekey -n 5 30 zz99999
Using the MD5 algorithm to compute response.
Reminder: Don't use opiekey from telnet or dial-in sessions.
Enter secret pass phrase: <geheim wachtwoord>
26: JOAN BORE FOSS DES NAY QUIT
27: LATE BIAS SLAY FOLK MUCH TRIG
28: SALT TIN ANTI LOON NEAL USE
29: RIO ODIN GO BYE FURY TIC
30: GREW JIVE SAN GIRD BOIL PHI
```

Met `-n 5` worden vijf opeenvolgende sleutels aangevraagd, `30` geeft aan wat het laatste iteratiegetal moet zijn. Deze wachtwoorden worden weergegeven in *omgekeerde* volgorde voor gebruik. Als de gebruiker echt paranoïde bent kan hij ze opschrijven of hij kan er ook voor kiezen ze af te drukken met `lpr`. Op iedere regel staat dus de iteratieteller en het eenmalige wachtwoord, maar misschien is het toch handig om ze na gebruik af te strepen.

13.5.5. Gebruik van UNIX® wachtwoorden beperken

Met OPIE kan paal en perk gesteld worden aan het gebruik van UNIX® wachtwoorden op basis van het IP-adres van een aanmeldsessie. Dat kan met het bestand /etc/opieaccess dat standaard aanwezig is. Bij opieaccess(5) staat meer informatie over dit bestand en welke beveiligingsoverwegingen bestaan bij het gebruik.

Hieronder een voorbeeld voor een opieaccess bestand:

```
permit 192.168.0.0 255.255.0.0
```

In deze regel (`permit Internet`) staat dat gebruikers met een bron IP adres (wat gefingeerd kan worden) dat past binnen de aangegeven waarde en masker altijd UNIX® wachtwoorden mogen gebruiken.

Als geen van de regels uit opieaccess van toepassing is, worden standaard pogingen zonder OPIE geweigerd.

13.6. TCP Wrappers

Iedereen die bekend is met inetd(8) heeft waarschijnlijk wel eens van TCP Wrappers gehoord. Maar slechts weinigen lijken volledig te begrijpen hoe ze in een netwerkomgeving toegepast kunnen worden. Het schijnt dat iedereen een firewall wil hebben om netwerkverbindingen af te handelen. Ondanks dat een firewall veel kan, zijn er toch dingen die het niet kan, zoals tekst terugsturen naar de bron van een verbinding. De TCP Wrappers software kan dat en nog veel meer. In dit onderdeel worden de mogelijkheden van TCP Wrappers besproken en, waar dat van toepassing is, worden ook voorbeelden voor implementatie gegeven.

De TCP Wrappers software vergroot de mogelijkheden van inetd door de mogelijkheid al zijn serverdaemons te controleren. Met deze methode is het mogelijk om te loggen, berichten te zenden naar verbindingen, een daemon toe te staan alleen interne verbindingen te accepteren, etc. Hoewel een aantal van deze mogelijkheden ook ingesteld kunnen worden met een firewall, geeft deze manier niet alleen een extra laag beveiliging, maar gaat dit ook verder dan wat een firewall kan bieden.

De toegevoegde waarde van TCP Wrappers is niet dat het een goede firewall vervangt. TCP Wrappers kunnen samen met een firewall en andere beveiligingsinstellingen gebruikt worden om een extra laag van beveiliging voor het systeem te bieden.

Omdat dit een uitbreiding is op de instellingen van inetd, wordt aangenomen dat de lezer het onderdeel inetd configuratie heeft gelezen.



Hoewel programma's die onder inetd(8) draaien niet echt "daemons" zijn, heten ze traditioneel wel zo. Deze term wordt hier dus ook gebruikt.

13.6.1. Voor het eerst instellen

De enige voorwaarde voor het gebruiken van TCP Wrappers in FreeBSD is ervoor te zorgen dat de server inetd gestart wordt vanuit rc.conf met de optie `-Ww`; dit is de standaardinstelling. Er wordt vanuit gegaan dat /etc/hosts.allow juist is ingesteld, maar als dat niet zo is, dan zal syslogd(8) dat melden.



In tegenstelling tot bij andere implementaties van TCP Wrappers is het gebruik van hosts.deny niet langer mogelijk. Alle instellingen moeten in /etc/hosts.allow staan.

In de meest eenvoudige instelling worden verbindingen naar daemons toegestaan of geweigerd afhankelijk van de opties in /etc/hosts.allow. De standaardinstelling in FreeBSD is verbindingen toe te staan naar iedere daemon die met inetd is gestart. Na de basisinstelling wordt aangegeven hoe dit gewijzigd kan worden.

De basisinstelling heeft meestal de vorm `daemon : adres : actie`. `daemon` is de daemonnaam die `inetd` heeft gestart. Het `adres` kan een geldige hostnaam, een IP-adres of een IPv6-adres tussen blokhaken ([ ]) zijn. Het veld `actie` kan `allow` of `deny` zijn, afhankelijk van of toegang toegestaan of geweigerd moet worden. De instellingen werken zo dat ze worden doorlopen van onder naar boven om te kijken welke regel als eerste van toepassing is. Als een regel van toepassing is gevonden, dan stop het zoekproces.

Er zijn nog andere mogelijkheden, maar die worden elders toegelicht. Een eenvoudige instelling kan al van met deze informatie worden gemaakt. Om bijvoorbeeld POP3 verbindingen toe te staan via de mail/qpopper daemon, zouden de volgende instellingen moeten worden toegevoegd aan hosts.allow:

```
# Deze regel is nodig voor POP3-verbindingen
qpopper : ALL : allow
```

Nadat deze regel is toegevoegd moet inetd herstart worden door gebruik te maken van service(8):

```
# service inetd restart
```

13.6.2. Gevorderde instellingen

TCP Wrappers hebben ook gevorderde instellingen. Daarmee komt meer controle over de wijze waarop er met verbindingen wordt omgegaan. Soms is het een goed idee om commentaar te sturen naar bepaalde hosts of daemonverbindingen. In andere gevallen moet misschien iets in een logboekbestand geschreven worden of een email naar de beheerder gestuurd worden. Dit kan allemaal met instellingen die `wildcards`, uitbreidingskarakters (expansion characters) en het uitvoeren van externe commando's heten. De volgende twee paragrafen beschrijven deze mogelijkheden.

Externe commando's

Stel dat zich de situatie voordoet waar een verbinding geweigerd moet worden, maar er een reden gestuurd moet worden naar het individu dat die verbinding probeerde op te zetten. Hoe gaat dat? Dat is mogelijk door gebruik te maken van de optie `twist`. Als er een poging tot verbinding wordt gedaan, wordt er met `twist` een shellcommando of script uitgevoerd. Er staat al een voorbeeld in hosts.allow:

```
# De andere daemons zijn beschermd.
ALL : ALL \
    : severity auth.info \
    : twist /bin/echo "You are not welcome to use %d from %h."
```

Dit voorbeeld geeft aan dat het bericht "You are not allowed to use daemon from hostname." wordt teruggestuurd voor iedere daemon die niet al is ingesteld in het toegangsbestand. Het is erg handig om een antwoord terug te sturen naar degene die een verbinding op heeft willen zetten meteen nadat een tot stand gekomen verbinding is verbroken. Let wel dat alle berichten die gezonden worden *moeten* staan tussen " karakters. Hier zijn geen uitzonderingen op.



Het is mogelijk een ontzegging van dienst aanval uit te voeren op de server als een aanvaller, of een groep aanvallers, deze daemons kan overstromen met verzoeken om verbindingen te maken.

Het is ook mogelijk hier de optie spawn te gebruiken. Net als twist weigert de optie spawn impliciet de verbinding en kan het gebruikt worden om shellcommando's of scripts uit te voeren. Anders dan bij twist stuurt spawn geen bericht aan degene die de verbinding wilde maken. Zie bijvoorbeeld de volgende instelling:

```
# Geen verbindingen van example.com:
ALL : .example.com \
    : spawn (/bin/echo %a from %h attempted to access %d >> \
      /var/log/connections.log) \
    : deny
```

Hiermee worden alle verbindingen van het domein *.example.com geweigerd. Tegelijkertijd worden ook hostnaam, IP adres en de daemon waarmee verbinding werd gemaakt naar /var/log/connections.log geschreven.

Naast de vervangingskarakters die al zijn toegelicht, zoals %a, bestaan er nog een paar andere. In de handleiding van hosts_access(5) staat een volledige lijst.

Wildcardopties

Tot nu toe is in ieder voorbeeld ALL gebruikt. Er bestaan nog andere opties waarmee de mogelijkheden nog verder gaan. Zo kan ALL gebruikt worden om van toepassing te zijn op iedere instantie van een daemon, domein of een IP adres. Een andere wildcard die gebruikt kan worden is PARANOID. Daarmee wordt iedere host die een IP-adres geeft dat gefingeerd kan zijn aangeduid. Met andere woorden: PARANOID kan gebruikt worden om een actie aan te geven als er een IP-adres gebruikt wordt dat verschilt van de hostnaam. Het volgende voorbeeld kan wat verheldering brengen:

```
# Weiger mogelijke gespoofte verzoeken aan sendmail:
sendmail : PARANOID : deny
```

In het voorgaande voorbeeld worden alle verbindingsverzoeken aan sendmail met een IP-adres dat verschilt van de hostnaam geweigerd.



Het gebruik van de wildcard PARANOID kan nogal wat schade aanrichten als de cliënt of de server kapotte DNS-instellingen heeft. Voorzichtigheid van de

beheerder is geboden.

De handleiding van hosts_access(5) geeft meer uitleg over wildcards en de mogelijkheden die ze bieden.

Voordat de bovenstaande instellingen werken, dient de eerste regels in hosts.allow als commentaar gemarkeerd te worden.

13.7. Kerberos5

Kerberos is een netwerkdienst, protocol en systeem waarmee gebruikers zich kunnen aanmelden met behulp van een dienst op een veilige server. Diensten als op een andere server aanmelden, op afstand kopiëren, veilig tussen systemen kopiëren en andere taken met een hoog risico worden aanmerkelijk veiliger en beter controleerbaar.

Kerberos kan omschrijven worden als identiteitbevestigend proxy systeem. Het kan ook omschreven worden als een vertrouwd autenticatiesysteem van een derde partij. Kerberos vervult maar één taak: het veilig autenticeren van gebruikers op het netwerk. Het vervult geen autorisatietaken (wat gebruikers mogen) en controleert ook niets (wat gebruikers hebben gedaan). Nadat een cliënt en server Kerberos hebben gebruikt om hun identiteit vast te stellen kunnen ze ook al hun communicatie coderen om hun privacy en gegevensintegriteit te garanderen.

Daarom wordt het sterk aangeraden om Kerberos samen met andere beveiligingsmechanismen te gebruiken die autorisatie en controlemogelijkheden bieden.

De aanwijzingen die nu volgen kunnen gebruikt worden als werkinstructie om Kerberos in te stellen zoals dat wordt meegeleverd met FreeBSD. Een complete beschrijving staat in de handleiding.

Voor demonstratie van de installatie van Kerberos wordt gebruik gemaakt van de volgende naamgeving:

- Het DNS domein ("zone") is example.org.
- De Kerberos wereld is EXAMPLE.ORG.



Het advies is voor installaties van Kerberos echte domeinnamen te gebruiken, zelfs als het alleen intern wordt gebruikt. Hiermee worden DNS problemen voorkomen is een goede samenwerking met andere Kerberos werelden verzekerd.

13.7.1. Geschiedenis

Kerberos is ontworpen door MIT als oplossing voor netwerkbeveiligingsproblemen. Het Kerberos protocol gebruikt sterke codering zodat een cliënt zijn identiteit kan bewijzen aan een server (en andersom) over een onveilige netwerkverbinding.

Kerberos is zowel de naam van een netwerkautorisatieprotocol als een bijvoeglijk naamwoord om de programma's te beschrijven die gebruik maken van het programma (zoals Kerberos telnet). De huidige versie van het protocol is versie 5 en is beschreven in RFC 1510.

Er zijn een aantal vrij beschikbare implementaties van dit protocol beschikbaar voor veel systemen. Het Massachusetts Institute of Technology (MIT), waar Kerberos ooit is ontwikkeld, ontwikkelt nog steeds door aan hun Kerberos pakket. Het wordt in de VS veel gebruikt als coderingspakket en daarom wordt het ook geraakt door de exportwetgeving van de VS. Kerberos van MIT is beschikbaar als port (security/krb5). Heimdal Kerberos is een andere implementatie van versie 5 die expliciet buiten de VS is ontwikkeld om de exportwetgeving de omzeilen (en wordt daarom vaak gebruikt in niet-commerciële UNIX® varianten). De Heimdal Kerberos distributie is beschikbaar als port (security/heimdal) en er zit een minimale installatie in de basisinstallatie van FreeBSD.

Om het grootst mogelijke publiek te bereiken gaan deze instructies ervan uit dat de Heimdal distributie die bij FreeBSD zit wordt gebruikt.

13.7.2. Opzetten van een Heimdal KDC

Het Sleutel Distributie Centrum (KDC, voluit "Key Distribution Center") is de gecentraliseerde autenticatiedienst die Kerberos levert. Het is de computer die Kerberos tickets uitgeeft. Het KDC wordt "vertrouwd" door alle andere computer in de Kerberos wereld en daarom dient er een strenger beveiligingsregime op van kracht te zijn.

Hoewel het draaien van de Kerberos dienst erg weinig van een systeem vraagt, wordt het wel aangeraden om een machine in te richten exclusief voor het KDC om beveiligingsredenen.

Het opzetten van een KDC begint met de controle of de instellingen in /etc/rc.conf juist zijn om te functioneren als KDC (misschien moeten paden veranderd worden voor een eigen systeem):

```
kerberos5_server_enable="YES"
kadmind5_server_enable="YES"
```

Daarna wordt het Kerberos-instellingenbestand /etc/krb5.conf aangemaakt:

```
[libdefaults]
    default_realm = EXAMPLE.ORG
[realms]
    EXAMPLE.ORG = {
        kdc = kerberos.example.org
        admin_server = kerberos.example.org
    }
[domain_realm]
    .example.org = EXAMPLE.ORG
```

/etc/krb5.conf gaat ervan uit dat de KDC de volledig gekwalificeerde hostnaam `kerberos.example.org` heeft. Als de KDC een andere hostnaam heeft, moet er nog een CNAME (alias) toegevoegd aan de zonefile.



Voor grotere netwerken met een juist ingestelde BINDDNS server kan het bovenstaande voorbeeld ingekort worden tot:

```
[libdefaults]
      default_realm = EXAMPLE.ORG
```

Door de volgende regels toe te voegen aan het zonebestand voor `example.org`:

```
_kerberos._udp      IN  SRV     01 00 88 kerberos.example.org.
_kerberos._tcp      IN  SRV     01 00 88 kerberos.example.org.
_kpasswd._udp       IN  SRV     01 00 464 kerberos.example.org.
_kerberos-adm._tcp  IN  SRV     01 00 749 kerberos.example.org.
_kerberos           IN  TXT     EXAMPLE.ORG
```



Om cliënten de Kerberos-diensten te kunnen laten vinden, *moet* er een volledig ingestelde /etc/krb5.conf zijn of een minimaal ingestelde /etc/krb5.conf*en* een correct ingestelde DNS-server.

Nu wordt de Kerberos database aangemaakt. Deze database bevat de sleutels voor alle principals en zijn versleuteld met een hoofdwachtwoord. Dit wachtwoord hoeft niet onthouden te worden omdat het wordt opgeslagen in (/var/heimdal/m-key). De hoofdsleutel wordt aangemaakt door `kstash` te starten en een wachtwoord in te voeren.

Als de hoofdsleutel is gemaakt, kan de database ingeschakeld worden met `kadmin` met de optie `-l` (die staat voor "local"). Deze optie geeft `kadmin` de opdracht om de databasebestanden direct te wijzigingen in plaats van via de `kadmind` netwerkdienst. Hiermee wordt het kip-ei-probleem opgelost waarbij een verbinding wordt gemaakt met de database voordat hij bestaat. Op het prompt van `kadmin` kan met `init` de database met de werelden aangemaakt worden.

Tenslotte, nog steeds in `kadmin`, kan de eerste principal gemaakt worden met `add`. De standaardopties voor de principal worden nu aangehouden. Deze kunnen later altijd nog gewijzigd worden met `modify`. Met het commando `?` kunnen alle beschikbare mogelijkheden getoond worden.

Hieronder een sessie waarin een voorbeelddatabase wordt aangemaakt:

```
# kstash
Master key: xxxxxxxx
Verifying password - Master key: xxxxxxxx

# kadmin -l
kadmin> init EXAMPLE.ORG
Realm max ticket life [unlimited]:
kadmin> add tillman
Max ticket life [unlimited]:
Max renewable life [unlimited]:
Attributes []:
Password: xxxxxxxx
Verifying password - Password: xxxxxxxx
```

Nu kan de KDC dienst gestart worden met `service kerberos start` en `service kadmind start`. Op dit moment draait er nog geen enkele daemon die gebruik maakt van Kerberos. Bevestiging dat KDC draait is te krijgen door een ticket te vragen en dat uit te lezen voor de principal (gebruiker) die zojuist is aangemaakt vanaf de commandoregel van het KDC zelf:

```
% kinit tillman
tillman@EXAMPLE.ORG's Password:

% klist
Credentials cache: FILE:/tmp/krb5cc_500
    Principal: tillman@EXAMPLE.ORG

  Issued           Expires          Principal
Aug 27 15:37:58  Aug 28 01:37:58  krbtgt/EXAMPLE.ORG@EXAMPLE.ORG
```

Het ticket kan worden ingenomen wanneer u klaar bent:

```
% kdestroy
```

13.7.3. Kerberos inschakelen op een server met Heimdal diensten

Als eerste is een kopie van het instellingenbestand van Kerberos nodig, /etc/krb5.conf. Dit bestand kan eenvoudigweg op een veilige manier (met netwerkprogramma's als scp(1), of fysiek via een floppy) naar de cliëntcomputer gekopieerd worden vanaf de KDC.

Hierna is het /etc/krb5.keytab nodig. Dit is het belangrijkste verschil tussen een server die een daemons met Kerberos aanbiedt en een werkstation: de server heeft het bestand keytab nodig. Dit bestand bevat de hostsleutel van de server waardoor het werkstation en de KDC elkaars identiteit kunnen bevestigen. Dit bestand dient veilig overgebracht te worden omdat de beveiliging van de server doorbroken kan worden als de sleutel openbaar wordt gemaakt. Dit betekent expliciet dat overdracht via een protocol dat platte tekst gebruikt, bijvoorbeeld FTP, een slecht idee is.

Meestal wordt keytab naar de server gebracht met `kadmin`. Dat werkt handig omdat ook de host principal (het KDC onderdeel van krb5.keytab) aangemaakt moet worden met `kadmin`.

Let wel op dat er al een ticket moet zijn en dat dit ticket de `kadmin` interface moet mogen gebruiken in kadmind.acl. Zie "Beheer op Afstand" in de Heimdal informatiepagina's (`info heimdal`) voor details over het ontwerpen van toegangscontrole. Als `kadmin` via het netwerk geen toegang mag hebben, dan kan ook op een veilige verbinding gemaakt worden met de KDC (via het lokale console, ssh(1) of Kerberos telnet(1)) zodat alles lokaal uitgevoerd kan worden met `kadmin -l`.

Na het installeren van /etc/krb5.conf kan `kadmin` van de Kerberos server gebruikt worden. Met `add --random-key` kan de host principal toegevoegd worden en met `ext` kan de host principal van de server naar zijn eigen keytab getrokken worden. Bijvoorbeeld:

```
# kadmin
kadmin> add --random-key host/myserver.example.org
```

```
Max ticket life [unlimited]:
Max renewable life [unlimited]:
Attributes []:
kadmin> ext host/myserver.example.org
kadmin> exit
```

Let op: `ext` slaat de sleutel standaard op in /etc/krb5.keytab.

Als `kadmind` niet beschikbaar is op de KDC (wellicht om beveiligingsredenen) en er via het netwerk dus geen toegang is tot `kadmin`, dan kan de host principal (`host/myserver.EXAMPLE.ORG`) ook direct aan de KDC toegevoegd worden en daarna in een tijdelijk bestand gezet worden. Het volgende kan gebruikt worden om te voorkomen dat /etc/krb5.keytab op de KDC) wordt overschreven:

```
# kadmin
kadmin> ext --keytab=/tmp/example.keytab host/myserver.example.org
kadmin> exit
```

Hierna kan de keytab veilig gekopieerd worden naar de server (met `scp` of een floppy). Geef een niet-standaard naam op voor de keytab om te voorkomen dat de keytab op de KDC wordt overschreven.

Nu kan de server communiceren met de KDC (vanweg krb5.conf) en zijn identiteit bewijzen (vanwege krb5.keytab). Nu is de server klaar om er een aantal Kerberos diensten op te activeren. In dit voorbeeld wordt de dienst `telnet` geactiveerd door de volgende regel in /etc/inetd.conf te zetten en dan inetd(8) te herstarten met `service inetd restart`:

```
telnet    stream  tcp     nowait  root    /usr/libexec/telnetd  telnetd -a user
```

Het belangrijkste is dat de typering `-a` (van autenticatie) op user staat. Meer details zijn in telnetd(8) te vinden.

13.7.4. Kerberos activeren op een cliënt met Heimdal

Het opzetten van een cliëntcomputer is eigenlijk kinderlijk eenvoudig. Wat betreft de Kerberos instelling is alleen het Kerberos instellingenbestand (/etc/krb5.conf) nodig. Dat kan eenvoudigweg naar de cliëntcomputer gekopieerd worden vanaf de KDC.

Test de cliënt met `kinit`, `klist` en `kdestroy` vanaf de cliënt om een ticket te krijgen, te bekijken en daarna te verwijderen voor de principal die hierboven is aangemaakt. Nu moeten ook Kerberos applicaties gebruikt kunnen worden om verbindingen te maken met servers waarop Kerberos is geactiveerd. Als dat niet lukt en het verkrijgen van een ticket is wel mogelijk, dan ligt dat hoogstwaarschijnlijk aan de server en niet aan de cliënt of de KDC.

Bij het testen van een applicatie als `telnet` kan het beste een pakketsnuffelaar (bijvoorbeeld tcpdump(1)) gebruikt worden om te bevestigen dat een wachtwoord niet als tekst wordt verzonden. Gebruik `telnet` met de optie `-x`. Dan wordt de complete gegevensstroom versleuteld (vergelijkbaar met `ssh`).

Er worden standaard ook andere Kerberos applicaties op de cliënt geïnstalleerd. Hier komt de "minimalistische" natuur van de basisinstallatie van Heimdal boven drijven: `telnet` is de enige dienst waarvoor Kerberos geactiveerd is.

De port Heimdal voegt een aantal ontbrekende cliëntapplicaties toe: versies met ondersteuning voor Kerberos van `ftp`, `rsh`, `rcp`, `rlogin` en een paar minder gebruikelijke programma's. De MIT port bevat ook een volledig gamma aan Kerberos cliëntapplicaties.

13.7.5. Instellingenbestanden voor gebruikers: .k5login en .k5users

Voor gebruikers binnen een wereld wijst hun Kerberos principal (bv. `tillman@EXAMPLE.ORG`) gewoonlijk naar een lokale gebruikersaccount (bijvoorbeeld een lokale account met de naam `tillman`). Voor cliëntapplicaties als `telnet` is gewoonlijk geen gebruikersnaam of principal nodig.

Soms moet iemand zonder bijpassende Kerberos principal toch toegang hebben tot een lokale gebruikersaccount. `tillman@EXAMPLE.ORG` zou bijvoorbeeld toegang nodig kunnen hebben tot de lokale gebruikersaccount `webdevelopers`. Andere principals zouden die toegang wellicht ook nodig kunnen hebben.

De bestanden .k5login en .k5users uit de gebruikersmap kunnen op eenzelfde manier gebruikt worden als .hosts en .rhosts. Zo wordt het voorgaande probleem opgelost. Als bijvoorbeeld een .k5login met de volgende inhoud:

```
tillman@example.org
jdoe@example.org
```

in de thuismap van de lokale gebruiker `webdevelopers` gezet wordt dan zouden beide principals toegang hebben tot die account zonder dat ze een wachtwoord hoeven te delen.

We raden aan de handleidingen voor deze commando's te lezen. Let op dat de `ksu` handleiding .k5users behandelt.

13.7.6. Kerberos tips, trucs en problemen oplossen

- Als de Heimdal of MITKerberos port wordt gebruikt dan dient de `PATH` omgevingsvariabele de Kerberos versies van de cliëntapplicaties te tonen voor de systeemversies.
- Hebben alle computers in de wereld hun tijd gesynchroniseerd? Als dat niet zo is, dan slaagt de autenticatie wellicht niet. Tijd synchroniseren met NTP beschrijft hoe klokken met NTP gesynchroniseerd kunnen worden.
- MIT en Heimdal werken prima samen. Dit geldt niet voor `kadmin` omdat daarvoor geen protocolstandaard is.
- Als een hostnaam wordt gewijzigd, dan moet ook de `host/` principal aangepast en de keytab. Dit geldt ook voor bijzondere instellingen in de keytab zoals de `www/` principal voor www/mod_auth_kerb van Apache.
- Alle hosts in een wereld moeten oplosbaar (resolvable) zijn (zowel vooruit als achteruit) in de DNS (of tenminste in /etc/hosts). CNAMEs werken wel, maar de A en PTR records moeten juist en

actief zijn. De foutmelding is niet erg duidelijk: `Kerberos5 refuses authentication because Read req failed: Key table entry not found`.

- Sommige besturingssystemen van cliënten voor een KDC zetten wellicht geen setuid `root` voor `ksu`. Dit betekent dat `ksu` niet werkt. Dat is vanuit beveiligingsoogpunt een prima idee, maar wel lastig. Dit is dus geen KDC-fout.
- Als met MITKerberos een principal een ticket moet krijgen dat langer geldig is dan de standaard van tien uur, dan moet `modify_principal` in `kadmin` gebruikt worden om de maximale geldigheidsduur (maxlife) van zowel de principal waar het om gaat als de `krbtgt` principal aan te passen. Dan kan de principal `kinit`-l` gebruiken om een ticket met een langere levensduur aan te vragen.



Als een pakketsnuffelaar op de KDC draait bij om te helpen bij het oplossen van problemen en dan `kinit` vanaf een werkstation wordt gestart, dan wordt zichtbaar dat de TGT meteen wordt verstuurd als `kinit` start, zelfs nog voor het wachtwoord! De reden hiervoor is dat de Kerberos server vrijelijk een TGT (Ticket Granting Ticket) verstuurt op iedere niet geautoriseerd verzoek. Maar iedere TGT is versleuteld met een sleutel die is afgeleid van het wachtwoord van de gebruiker. Als een gebruiker zijn wachtwoord ingeeft, wordt dat dus niet naar de KDC gezonden, maar ontcijfert het de TGT die `kinit` al heeft ontvangen. Als de ontcijfering resulteert in een geldige ticket met een geldige tijdstempel, dan heeft de gebruiker geldige Kerberos rechten. Deze rechten bevatten ook een sessiesleutel voor het opzetten van beveiligde communicatie met de Kerberos server in de toekomst en de eigenlijke ticket-granting ticket, die is versleuteld met de sleutel van de Kerberos server zelf. Deze tweede laag van versleuteling is niet bekend voor de gebruiker, maar het stelt de Kerberos server in staat om de juistheid van iedere TGT te bevestigen.

- Als tickets worden gebruik die lang geldig zijn (bv. een week) en OpenSSH wordt gebruikt om een verbinding te maken met de machine waarop het ticket staat, zorg er dan voor dat de Kerberos optie `TicketCleanup` op `no` staat in sshd_config want anders worden tickets verwijderd bij afmelden.
- Hostprincipals kunnen ook een langere levensduur hebben. Als een gebruikers principal een levensduur van een week heeft, maar de host waar de verbinding mee gemaakt wordt heeft een levensduur van negen uur, dan heb staat er een verlopen host principal in de cache en dan werkt een en ander niet zoals verwacht.
- Een krb5.dict bestand om het gebruik van bepaalde slechte wachtwoorden te voorkomen (dit wordt kort behandeld in de handleiding voor `kadmind`) heeft alleen betrekking op principals waar een wachtwoordbeleid voor geldt. De opmaak van krb5.dict is eenvoudig: een rij tekens per regel. Een symbolische link maken naar /usr/shared/dict/words is misschien handig.

13.7.7. Verschillen met de MIT port

Het belangrijkste verschil tussen de MIT en Heimdal installatie heeft betrekking op `kadmin`, dat een andere (maar gelijkwaardige) set commando's kent en een andere protocol gebruikt. Dit betekent nogal wat als een KDCMIT is, omdat dan de `kadmin` van Heimdal niet gebruikt kan worden om de KDC vanaf afstand te beheren (dat geldt trouwens ook vice versa).

De cliëntapplicaties kunnen ook commandoregelopties gebruiken die een beetje verschillen, maar waarmee wel hetzelfde wordt bereikt. We raden aan de instructies op de MITKerberos website (http://web.mit.edu/Kerberos/www/) te volgen. Wees voorzichtig met paden: de MIT-port installeert standaard in /usr/local/ en dus kunnen de "normale" systeemapplicaties gestart worden in plaats van die van MIT als de `PATH` omgevingsvariabele de systeemmappen als eerste weergeeft.



Als de MITsecurity/krb5 port die bij FreeBSD zit wordt gebruikt, dan zorgt het lezen van /usr/local/shared/doc/krb5/README.FreeBSD dat bij de port wordt geïnstalleerd voor een beter begrip over waarom het aanmelden via `telnetd` en `klogind` soms wat vreemd verloopt. Als belangrijkste wijzen we erop dat het bij het corrigeren van "onjuiste rechten op het cachebestand" noodzakelijk is dat het binaire bestand `login.krb5` wordt gebruikt voor autenticatie zodat het op de juiste wijze eigenaarschap kan wijzigen voor de doorgegeven rechten.

Het bestand rc.conf moet ook gewijzigd worden zodat het de volgende configuratie bevat:

```
kerberos5_server="/usr/local/sbin/krb5kdc"
kadmind5_server="/usr/local/sbin/kadmind"
kerberos5_server_enable="YES"
kadmind5_server_enable="YES"
```

Dit is gedaan omdat de applicaties voor MIT-Kerberos binairen in de hiërarchie /usr/local installeren.

13.7.8. Beperkingen in Kerberos

Kerberos is een alles of niets aanpak

Iedere ingeschakelde dienst op het netwerk moet aangepast worden om met Kerberos te werken (of op een andere manier beschermd zijn tegen netwerkaanvallen), want anders kunnen gebruikersrechten worden gestolen en herbruikt. Een voorbeeld hier van is het inschakelen van Kerberos voor alle shells op afstand (via `rsh` en `telnet` bijvoorbeeld), maar de POP3 mailserver die wachtwoorden als platte tekst verzend ongemoeid laten.

Kerberos is bedoeld voor werkstations met een gebruiker

In een meergebruikersomgeving is Kerberos minder veilig. Dit komt doordat de tickets worden opgeslagen in de map /tmp, waar gelezen kan worden door alle gebruikers. Als een gebruiker een computer deelt met andere gebruikers op hetzelfde moment (dus multi-user), dan is het mogelijk dat een ticket van een gebruiker wordt gestolen (gekopieerd) door een andere gebruiker.

Dit kan voorkomen worden met de commandoregeloptie "`-c` bestandsnaam" of (bij voorkeur) de omgevingsvariabele `KRB5CCNAME`, maar dat wordt zelden gedaan. In principe kan het opslaan van een ticket in de thuismap van een gebruiker in combinatie met eenvoudige bestandsrechten dit probleem verhelpen.

De KDC is een single point of failure

Zoals het is ontworpen, moet de KDC zo goed mogelijk beveiligd zijn, omdat de hoofdwachtwoorddatabase erop staat. De KDC hoort geen enkele andere dienst aan te bieden en moet ook fysiek afgeschermd worden. Het gevaar is groot, omdat Kerberos alle wachtwoorden versleutelt met dezelfde sleutel (de "master" sleutel) die als een bestand op de KDC staat.

Toch is een gecompromitteerde mastersleutel niet zo'n groot probleem als wellicht wordt verondersteld. De mastersleutel wordt alleen gebruikt om de Kerberos database te versleutelen en als zaad voor de generator van willekeurige nummers. Zo lang als de toegang tot de KDC is beveiligd, kan een aanvaller niet echt iets doen met de mastersleutel.

Als de KDC niet beschikbaar is (misschien door een ontzeggen van dienst aanval of netwerkproblemen) kunnen de netwerkdiensten niet gebruikt worden omdat er geen autenticatie uitgevoerd kan worden; een recept voor een ontzeggen van dienst aanval. Dit risico kan omzeild worden door meerdere KDC's (één master en één of meer slaven) en een zorgvuldige implementatie van secundaire of fall-back autenticatie. PAM is hier uitermate geschikt voor.

Tekortkomingen van Kerberos

Kerberos stelt gebruikers, hosts en diensten in staat om elkaar te autenticeren. Maar het heeft geen mechanisme om de KDC te autenticeren aan de gebruikers, hosts of diensten. Dit betekent dat bijvoorbeeld een vervalste `kinit` alle gebruikersnamen en wachtwoorden zou kunnen afluisteren. Iets als security/tripwire of andere controle-instrumenten voor de integriteit van bestandssystemen kunnen hier verlichting brengen.

13.7.9. Bronnen en verdere informatie

- De Kerberos FAQ (Engels)
- Een Autenticatiesysteem Ontwerpen: een Dialoog in Vier Scenes (Engels)
- RFC 1510, De Kerberos Netwerk Authenticatie Dienst (V5) (Engels)
- MIT Kerberos homepage
- Heimdal Kerberos homepage

13.8. OpenSSL

Een toepassing die bij FreeBSD zit die veel gebruikers over het hoofd zien is OpenSSL. OpenSSL biedt een versleutelde transportlaag bovenop de normale communicatielaag. Daardoor biedt het de mogelijkheid met veel netwerktoepassingen en diensten verweven te raken.

Een aantal toepassingen van OpenSSL zijn versleutelde autenticatie van mailcliënten, webgebaseerde transacties als creditcardbetalingen en nog veel meer. Veel ports zoals www/apache22 en mail/claws-mail bieden tijdens het compileren ondersteuning om OpenSSL in te bouwen.



In de meeste gevallen zal de Portscollectie proberen de port security/openssl te bouwen, tenzij de make variabele `WITH_OPENSSL_BASE` expliciet naar "yes" is gezet.

De versie van OpenSSL die bij FreeBSD zit ondersteunt Secure Sockets Layer v2/v3 (SSLv2/SSLv3), Transport Layer Security v1 (TLSv1) netwerkbeveiligingsprotocollen en kan gebruikt worden als generieke versleutelingsbibliotheek.



Hoewel OpenSSL ondersteuning biedt voor het IDEA algoritme, is dat standaard uitgeschakeld in verband met patenten in de Verenigde Staten. Om het te gebruiken dient de licentie gelezen te worden en, als de restricties aanvaardbaar zijn, dient de make-variabele `MAKE_IDEA` ingesteld te worden in make.conf.

Een van de meest gebruikte toepassingen van OpenSSL is het leveren van certificaten voor gebruik met softwaretoepassingen. Deze certificaten verzekeren dat de eigenschappen van een bedrijf of individu geldig zijn en niet vervalst. Als het certificaat in kwestie niet geldig verklaard is door een van de "Certificate Authorities" of CA's, dan komt er een waarschuwing. Een Certificate Authority is een bedrijf, zoals VeriSign, dat certificaten ondertekent zodat de eigenschappen van een bedrijf of individu geldig verklaard kunnen worden. Dit proces kost geld en het is zeker geen voorwaarde voor het gebruik van certificaten. Het stelt wel de meer paranoïde gebruikers gerust.

13.8.1. Certificaten maken

Voor het maken van certificaten is het volgende commando beschikbaar:

```
# openssl req -new -nodes -out req.pem -keyout cert.pem
Generating a 1024 bit RSA private key
..............
.....................................
writing new private key to 'cert.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:PA
Locality Name (eg, city) []:Pittsburgh
Organization Name (eg, company) [Internet Widgits Pty Ltd]:My Company
Organizational Unit Name (eg, section) []:Systems Administrator
Common Name (eg, YOUR name) []:localhost.example.org
Email Address []:trhodes@FreeBSD.org

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:SOME PASSWORD
An optional company name []:Another Name
```

Let op dat het antwoord direct na "Common Name" een domeinnaam weergeeft. De prompt wil dat

er een servernaam wordt ingegeven voor het verificatieproces. Het plaatsen van iets anders dan een domeinnaam zorgt ervoor dat het certificaat waardeloos wordt. Er zijn ook andere opties als verloopdatum, andere versleutelingsalgoritmes, etc, beschikbaar. Een volledige lijst is na te lezen in de handleiding van openssl(1).

Er horen nu twee bestanden te staan in de map waarin het voorgaande commando is uitgevoerd. Het certificaatverzoek, req.pem, kan naar een certificaat autoriteit gestuurd worden die de bijgevoegde gegevens kan valideren, het verzoek kan tekenen en het certificaat kan retourneren. Het tweede bestand heet cert.pem en is de geheime sleutel voor het certificaat. Deze dient zorgvuldig beschermd te worden. Als de geheime sleutel in de handen van anderen valt kan die gebruikt worden om de identiteit van de eigenaar (of server) aan te nemen.

In gevallen waar ondertekening door een CA niet vereist is, kan een zelfondertekend certificaat gemaakt worden. Maak als eerste de RSA sleutel:

```
# openssl dsaparam -rand -genkey -out myRSA.key 1024
```

Hierna kan de CA sleutel gemaakt worden:

```
# openssl gendsa -des3 -out myca.key myRSA.key
```

Deze sleutel kan gebruikt worden om een certificaat te maken:

```
# openssl req -new -x509 -days 365 -key myca.key -out new.crt
```

Er zouden nu twee bestanden bijgekomen moeten zijn in de map: een certificaatautoriteit ondertekeningsbestand myca.key en new.crt, het certificaat zelf. Deze moeten in een map geplaatst worden, bij voorkeur onder /etc waar alleen `root` kan lezen. De rechten 0700 zijn hier prima en die kunnen ingesteld worden met `chmod`.

13.8.2. Certificaten gebruiken: een voorbeeld

En wat kunnen deze bestanden? Een prima toepassing zou het versleutelen van verbindingen naar de SendmailMTA kunnen zijn. Daardoor zouden gebruikers niet langer platte tekst hoeven te autenticeren om mail te sturen via de lokale MTA.



Dit is niet de best denkbare toepassing omdat sommige MUA's de gebruiker een foutmelding geven als ze het certificaat niet lokaal geïnstalleerd hebben. De documentatie bij de software geeft meer informatie over het installeren van certificaten.

De volgende regels moeten opgenomen worden in het lokale .mc bestand:

```
dnl SSL Options
define(`confCACERT_PATH',`/etc/certs')dnl
```

```
define(`confCACERT',`/etc/certs/new.crt')dnl
define(`confSERVER_CERT',`/etc/certs/new.crt')dnl
define(`confSERVER_KEY',`/etc/certs/myca.key')dnl
define(`confTLS_SRV_OPTIONS', `V')dnl
```

/etc/certs/ is de map die gebruikt wordt voor het lokaal opslaan van certificaten en sleutels. De laatste voorwaarde het is opnieuw aanmaken van het lokale .cf bestand. Dit gaat door eenvoudigweg `make install` te typen in de map /etc/mail. Laat dat volgen door `make install` waardoor de daemon Sendmail herstart zou moeten worden.

Als alles goed is gegaan, dan staan er geen foutmeldingen /var/log/maillog en is Sendmail zichtbaar in de proceslijst.

Maak als eenvoudige test een verbinding met de mailserver met telnet(1):

```
# telnet example.com 25
Trying 192.0.34.166...
Connected to example.com.
Escape character is '^]'.
220 example.com ESMTP Sendmail 8.12.10/8.12.10; Tue, 31 Aug 2004 03:41:22 -0400 (EDT)
ehlo example.com
250-example.com Hello example.com [192.0.34.166], pleased to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-SIZE
250-DSN
250-ETRN
250-AUTH LOGIN PLAIN
250-STARTTLS
250-DELIVERBY
250 HELP
quit
221 2.0.0 example.com closing connection
Connection closed by foreign host.
```

Als de regel "STARTTLS" verschijnt in de uitvoer dan werkt alles correct.

13.9. VPN via IPsec

Een VPN opzetten met FreeBSD gateways tussen twee netwerken die gescheiden zijn door Internet.

13.9.1. IPsec begrijpen

Deze paragraaf is een gids in het proces van het opzetten van IPsec. Voordat IPsec opgezet kan worden dient de lezer bekend te zijn met de concepten die nodig zijn om een aangepaste kernel te bouwen (zie De FreeBSD-kernel instellen).

IPsec is een protocol dat bovenop de Internet Protocol (IP) laag ligt. Hiermee kunnen twee of meer host op een veilige manier communiceren (vandaar de naam). De FreeBSD IPsec "netwerk wachtrij (stack)" is gebaseerd op de KAME-implementatie, die zowel de protocolfamilies IPv4 als de IPv6 ondersteunt.

IPsec bestaat uit twee subprotocollen:

- *Encapsulated Security Payload (ESP)* beschermt de IP-pakketdata tegen inmenging door een derde partij door de inhoud te versleutelen met symmetrische versleutelingsalgoritmes (zoals Blowfish en 3DES).
- *Authentication Header (AH)* beschermt de IP-pakketkop tegen inmenging door een derde partij en spoofing door een cryptografische checksum te berekenen en de IP-pakketkopvelden te hashen met een veilige hashfunctie. Hierna wordt een extra kop ingevoegd die de hash bevat zodat de informatie in het pakket geautenticeerd kan worden.

ESP en AH kunnen samen of apart gebruikt worden, afhankelijk van de omgeving.

IPsec kan gebruikt worden om het verkeer tussen twee hosts direct te versleutelen (dat heet *Transport Mode*) of door "virtuele tunnels" te bouwen tussen twee subnetten die gebruikt kunnen worden voor veilige communicatie tussen twee bedrijfsnetwerken (dat heet *Tunnel Mode*). De laatste versie staat beter bekend als *Virtual Private Network (VPN)*. In ipsec(4) staat gedetailleerde informatie over het IPsec subsysteem in FreeBSD.

Voor ondersteuning voor IPsec in de kernel zijn de volgende opties nodig in het kernelinstellingenbestand:

```
options   IPSEC        IP-beveiliging
device    crypto
```

Als er ook fouten in IPsec (debugging) verwijderd moeten kunnen worden, dan is de volgende optie ook nodig:

```
options   IPSEC_DEBUG  debug voor IP-beveiliging
```

13.9.2. Het probleem

Er bestaat geen standaard voor wat een VPN is. VPN's kunnen opgezet worden met behulp van een aantal verschillende technologieën die allemaal hun eigen voor- en nadelen hebben. Dit onderdeel bevat een scenario en de strategieën die gebruikt kunnen worden voor het implementeren van een VPN in iedere situatie.

13.9.3. Het scenario: twee netwerken, de ene thuisgebaseerd en de andere bedrijfgebaseerd. Beide zijn verbonden met het Internet, en er wordt van verwacht dat ze zich via dit VPN als één gedragen.

Dit is het uitgangspunt:

- Er zijn tenminste twee locaties
- Beide locaties gebruiken IP
- Beide locaties hebben een Internetverbinding via een gateway waarop FreeBSD draait.
- De gateway op ieder netwerk heeft tenminste één publiek IP-adres.
- De interne adressen van de twee netwerken mogen publieke of private IP-adressen zijn, dat maakt niet uit. Ze mogen alleen niet botsen; bijvoorbeeld: ze mogen niet beide `192.168.1.x` gebruiken.

13.9.4. IPsec configureren op FreeBSD

Om te beginnen moet de port security/ipsec-tools geïnstalleerd zijn vanuit de Portscollectie. Dit softwarepakket van een derde partij biedt een aantal applicaties die helpen de configuratie te ondersteunen.

De volgende benodigdheid is om twee gif(4) pseudo-apparaten aan te maken om de pakketten te tunnelen en beide netwerken in staat stellen om op een juiste wijze te communiceren. Draai als `root` de volgende commando's, waarbij de items *intern* en *extern* met de echte interne en externe gateways:

```
# ifconfig gif0 create
```

```
# ifconfig gif0 intern1 intern2
```

```
# ifconfig gif0 tunnel extern1 extern2
```

Het publieke IP van het LAN van de onderneming is bijvoorbeeld `172.16.5.4` en het heeft een privaat IP`10.246.38.1`. Het publieke IP van het LAN van huis is `192.168.1.12` met een intern privaat IP`10.0.0.5`.

Dit kan verwarrend lijken, dus bekijk de volgende voorbeeld van het commando ifconfig(8):

```
Gateway 1:

gif0: flags=8051 mtu 1280
tunnel inet 172.16.5.4 --> 192.168.1.12
inet6 fe80::2e0:81ff:fe02:5881%gif0 prefixlen 64 scopeid 0x6
inet 10.246.38.1 --> 10.0.0.5 netmask 0xffffff00

Gateway 2:

gif0: flags=8051 mtu 1280
tunnel inet 192.168.1.12 --> 172.16.5.4
inet 10.0.0.5 --> 10.246.38.1 netmask 0xffffff00
```

```
inet6 fe80::250:bfff:fe3a:c1f%gif0 prefixlen 64 scopeid 0x4
```

Eenmaal compleet zouden beide private IP's bereikbaar moeten zijn met het commando ping(8) zoals de volgende uitvoer suggereert:

```
priv-net# ping 10.0.0.5
PING 10.0.0.5 (10.0.0.5): 56 data bytes
64 bytes from 10.0.0.5: icmp_seq=0 ttl=64 time=42.786 ms
64 bytes from 10.0.0.5: icmp_seq=1 ttl=64 time=19.255 ms
64 bytes from 10.0.0.5: icmp_seq=2 ttl=64 time=20.440 ms
64 bytes from 10.0.0.5: icmp_seq=3 ttl=64 time=21.036 ms
--- 10.0.0.5 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 19.255/25.879/42.786/9.782 ms

corp-net# ping 10.246.38.1
PING 10.246.38.1 (10.246.38.1): 56 data bytes
64 bytes from 10.246.38.1: icmp_seq=0 ttl=64 time=28.106 ms
64 bytes from 10.246.38.1: icmp_seq=1 ttl=64 time=42.917 ms
64 bytes from 10.246.38.1: icmp_seq=2 ttl=64 time=127.525 ms
64 bytes from 10.246.38.1: icmp_seq=3 ttl=64 time=119.896 ms
64 bytes from 10.246.38.1: icmp_seq=4 ttl=64 time=154.524 ms
--- 10.246.38.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 28.106/94.594/154.524/49.814 ms
```

Zoals verwacht hebben beide kanten de mogelijkheid om ICMP-pakketten te verzenden en te ontvangen van de privaat geconfigureerde adressen. Vervolgens dient aan beide gateways verteld te worden hoe pakketten te routeren om op de juiste wijze verkeer van een van de netwerken te versturen. Het volgende commando doet dit:

```
# corp-net# route add 10.0.0.0 10.0.0.5 255.255.255.0
```

```
# corp-net# route add net 10.0.0.0: gateway 10.0.0.5
```

```
# priv-net# route add 10.246.38.0 10.246.38.1 255.255.255.0
```

```
# priv-net# route add host 10.246.38.0: gateway 10.246.38.1
```

Op dit moment dienen interne machines bereikbaar te zijn vanuit elke gateway alsook als vanuit machines achter de gateways. Dit is eenvoudig te zien aan het volgende voorbeeld:

```
corp-net# ping 10.0.0.8
```

```
PING 10.0.0.8 (10.0.0.8): 56 data bytes
64 bytes from 10.0.0.8: icmp_seq=0 ttl=63 time=92.391 ms
64 bytes from 10.0.0.8: icmp_seq=1 ttl=63 time=21.870 ms
64 bytes from 10.0.0.8: icmp_seq=2 ttl=63 time=198.022 ms
64 bytes from 10.0.0.8: icmp_seq=3 ttl=63 time=22.241 ms
64 bytes from 10.0.0.8: icmp_seq=4 ttl=63 time=174.705 ms
--- 10.0.0.8 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 21.870/101.846/198.022/74.001 ms

priv-net# ping 10.246.38.107
PING 10.246.38.1 (10.246.38.107): 56 data bytes
64 bytes from 10.246.38.107: icmp_seq=0 ttl=64 time=53.491 ms
64 bytes from 10.246.38.107: icmp_seq=1 ttl=64 time=23.395 ms
64 bytes from 10.246.38.107: icmp_seq=2 ttl=64 time=23.865 ms
64 bytes from 10.246.38.107: icmp_seq=3 ttl=64 time=21.145 ms
64 bytes from 10.246.38.107: icmp_seq=4 ttl=64 time=36.708 ms
--- 10.246.38.107 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 21.145/31.721/53.491/12.179 ms
```

De tunnels opzetten is het eenvoudige deel. Het configureren van een veilige verbinding is een veel diepgaander proces. De volgende configuratie gebruikt vooraf gedeelde (PSK) RSA-sleutels. Afgezien van de IP-adressen zijn beide bestanden /usr/local/etc/racoon/racoon.conf identiek en zien ze er ongeveer als volgt uit:

```
path    pre_shared_key  "/usr/lcoal/etc/racoon/psk.txt"; # plaats van bestand vooraf
gedeelde sleutels
log     debug;  # verbositeitsinstelling van loggen: op 'notify' zetten als testen en
debuggen klaar is

padding # opties moeten niet veranderd worden
{
        maximum_length  20;
        randomize       off;
        strict_check    off;
        exclusive_tail  off;
}

timer   # timingopties, veranderen indien nodig
{
        counter         5;
        interval        20 sec;
        persend         1;
#       natt_keepalive  15 sec;
        phase1          30 sec;
        phase2          15 sec;
}

listen  # adres [poort] waarop racoon luistert
```

```
{
        isakmp          172.16.5.4 [500];
        isakmp_natt     172.16.5.4 [4500];
}

remote  192.168.1.12 [500]
{
        exchange_mode   main,aggressive;
        doi             ipsec_doi;
        situation       identity_only;
        my_identifier   address 172.16.5.4;
        peers_identifier        address 192.168.1.12;
        lifetime        time 8 hour;
        passive         off;
        proposal_check  obey;
#       nat_traversal   off;
        generate_policy off;

                        proposal {
                                encryption_algorithm    blowfish;
                                hash_algorithm          md5;
                                authentication_method   pre_shared_key;
                                lifetime time           30 sec;
                                dh_group                1;
                        }
}

sainfo  (address 10.246.38.0/24 any address 10.0.0.0/24 any)   # adres
$netwerk/$netmasker $type adres $netwerk/$netmasker $type ( $type is any of esp)
{
        pfs_group       1;
        lifetime        time    3600 sec;
        encryption_algorithm    blowfish,3des,des;
        authentication_algorithm        hmac_md5,hmac_sha1;
        compression_algorithm   deflate;
}
```

Het uitleggen van elke beschikbare optie, samen met diegenen in deze voorbeelden valt buiten het bereik van dit document. De configuratiehandleiding van racoon staat vol relevante informatie.

De SPD-beleiden moeten geconfigureerd worden zodat FreeBSD en racoon in staat zijn om netwerkverkeer tussen hosts te versleutelen en te ontsleutelen.

Deze taak kan met een eenvoudig shellscript zoals het volgende dat op de gateway van de onderneming staat worden uitgevoerd. Dit bestand wordt gebruikt tijdens de systeeminitialisatie en dient bewaard te worden als /usr/local/etc/racoon/setkey.conf.

```
flush;
spdflush;
# Naar het thuisnetwerk
```

```
spdadd 10.246.38.0/24 10.0.0.0/24 any -P out ipsec esp/tunnel/172.16.5.4-
192.168.1.12/use;
spdadd 10.0.0.0/24 10.246.38.0/24 any -P in esp/tunnel/192.168.1.12-172.16.5.4/use;
```

Eenmaal aanwezig kan racoon op beide gateways gestart worden met het volgende commando:

```
# /usr/local/sbin/racoon -F -f /usr/local/etc/racoon/racoon.conf -l
/var/log/racoon.log
```

De uitvoer moet ongeveer gelijk zijn aan de volgende:

```
corp-net# /usr/local/sbin/racoon -F -f /usr/local/etc/racoon/racoon.conf
Foreground mode.
2006-01-30 01:35:47: INFO: begin Identity Protection mode.
2006-01-30 01:35:48: INFO: received Vendor ID: KAME/racoon
2006-01-30 01:35:55: INFO: received Vendor ID: KAME/racoon
2006-01-30 01:36:04: INFO: ISAKMP-SA established 172.16.5.4[500]-192.168.1.12[500]
spi:623b9b3bd2492452:7deab82d54ff704a
2006-01-30 01:36:05: INFO: initiate new phase 2 negotiation:
172.16.5.4[0]192.168.1.12[0]
2006-01-30 01:36:09: INFO: IPsec-SA established: ESP/Tunnel 192.168.1.12[0]-
>172.16.5.4[0] spi=28496098(0x1b2d0e2)
2006-01-30 01:36:09: INFO: IPsec-SA established: ESP/Tunnel 172.16.5.4[0]-
>192.168.1.2[0] spi=47784998(0x2d92426)
2006-01-30 01:36:13: INFO: respond new phase 2 negotiation:
172.16.5.4[0]192.168.1.12[0]
2006-01-30 01:36:18: INFO: IPsec-SA established: ESP/Tunnel 192.168.1.12[0]-
>172.16.5.4[0] spi=124397467(0x76a279b)
2006-01-30 01:36:18: INFO: IPsec-SA established: ESP/Tunnel 172.16.5.4[0]-
>192.168.1.12[0] spi=175852902(0xa7b4d66)
```

Om er zeker van te zijn dat de tunnel correct werkt, dient naar een ander console geschakeld te worden en tcpdump(1) gebruikt te worden om hiermee het netwerkverkeer te bekijken. Vervang `em0` door de netwerkinterfacekaart indien nodig.

```
# tcpdump -i em0 host 172.16.5.4 and dst 192.168.1.12
```

Gegevens lijkend op de volgende zouden op het console moeten verschijnen. Indien niet, dan is er iets aan de hand, en is het nodig om de teruggegeven gegevens te debuggen.

```
01:47:32.021683 IP corporatenetwork.com > 192.168.1.12.privatenetwork.com:
ESP(spi=0x02acbf9f,seq=0xa)
01:47:33.022442 IP corporatenetwork.com > 192.168.1.12.privatenetwork.com:
ESP(spi=0x02acbf9f,seq=0xb)
01:47:34.024218 IP corporatenetwork.com > 192.168.1.12.privatenetwork.com:
```

```
ESP(spi=0x02acbf9f,seq=0xc)
```

Op dit punt zouden beide netwerken beschikbaar moeten zijn en deel lijken van hetzelfde netwerk. Waarschijnlijk zijn beide netwerken beschermt door een firewall, zoals het hoort. Om verkeer tussen hen toe te staan, moeten er regels worden toegevoegd om pakketten heen en terug door te laten. Voeg voor de firewall ipfw(8) de volgende regels toe aan het instellingenbestand van de firewall:

```
ipfw add 00201 allow log esp from any to any
ipfw add 00202 allow log ah from any to any
ipfw add 00203 allow log ipencap from any to any
ipfw add 00204 allow log udp from any 500 to any
```



Afhankelijk van de huidige hostconfiguratie dienen de regelnummers gewijzigd te worden.

Voor gebruikers van pf(4) of ipf(8) zouden de volgende regels moeten volstaan:

```
pass in quick proto esp form any to any
pass in quick proto ah from any to any
pass in quick proto ipencap from any to any
pass in quick proto udp form any port = 500 to any port = 500
pass in quick on gif0 from any to any
pass out quick proto esp from any to any
pass out quick proto ah from any to any
pass out quick ptoto ipencap from any to any
pass out quick proto udp from any port = 500 to any port = 500
pass out quick on gif0 from any to any
```

Ter afsluiting, voeg de volgende regels toe aan /etc/rc.conf om de machine toe te staan om ondersteuning voor het VPN te starten tijdens de systeeminitialisatie:

```
ipsec_enable="YES"
ipsec_program="/usr/local/sbin/setkey"
ipsec_file="/usr/local/etc/racoon/setkey.conf" # staat toe om spd-beleiden tijdens het
opstarten op te zetten
racoon_enable="yes"
```

13.10. OpenSSH

OpenSSH is een groep netwerkverbindingsprogramma's waarmee computers via het netwerk veilig benaderd kunnen worden. Het kan ingezet worden als een directe vervanger van `rlogin`, `rsh`, `rcp` en `telnet`. Daarnaast kunnen TCP/IP-verbindingen veilig getunneld of geforward worden door SSH. OpenSSH versleutelt al het verkeer om afluisteren, het stelen van een verbinding en andere netwerkaanvallen effectief te voorkomen.

OpenSSH wordt onderhouden door het OpenBSD project en is gebaseerd op SSH v1.2.12 met alle recente bugfixes en updates. Het is compatibel met beide protocollen SSH 1 en 2.

13.10.1. Voordelen van gebruik van OpenSSH

Als gewoonlijk telnet(1) of rlogin(1) wordt gebruikt, wordt de data in platte tekst en niet versleuteld verzonden. Netwerksnuffelaars die ergens tussen de cliënt en de server meeluisteren, kunnen een gebruikersnaam en wachtwoord stelen en zien welke gegevens er worden overgezonden tijdens een sessie. OpenSSH biedt een verscheidenheid aan autenticatie en versleutelingsmethoden die het voorgaande voorkomen.

13.10.2. sshd inschakelen

De sshd is een optie die wordt aangeboden tijdens een `Standard`-installatie van FreeBSD. sshd is ingeschakeld als de volgende regel voorkomt in rc.conf:

```
sshd_enable="YES"
```

Hierdoor wordt sshd(8) geladen, het daemonprogramma voor OpenSSH, als het systeem de volgende keer opstart. Als alternatief is het mogelijk om rc(8) te gebruiken om OpenSSH te starten:

```
# service sshd start
```

13.10.3. SSH-cliënt

ssh(1) werkt net zoals rlogin(1).

```
# ssh user@example.com
Host key not found from the list of known hosts.
Are you sure you want to continue connecting (yes/no)? yes
Host 'example.com' added to the list of known hosts.
user@example.com's password: *******
```

Het aanmelden gaat nu net zoals het zou gaan als wanneer er een sessie gestart zou worden met `rlogin` of `telnet`. SSH maakt gebruik van een systeem met vingerafdrukken als sleutels voor het vaststellen met welke server verbinding wordt gemaakt op het moment dat de cliënt verbinding zoekt. De gebruiker krijgt alleen de eerste keer dat verbinding wordt gezocht met de server een vraag waarop `yes` geantwoord dient te worden. Bij volgende pogingen om aan te melden wordt de vingerafdruksleutel vergeleken met de sleutel die is opgeslagen. De SSH-cliënt alarmeert de gebruiker als de opgeslagen vingerafdruk sleutel anders is dan de sleutel die de server meldt. De vingerafdrukken worden opgeslagen in ~/.ssh/known_hosts of in ~/.ssh/known_hosts2 voor SSH v2 vingerafdrukken.

Recente OpenSSH servers staan standaard ingesteld om alleen SSH v2 connecties toe te staan. De cliënt gebruikt versie 2 als dat mogelijk is en valt anders terug op versie 1. De cliënt kan ook gedwongen worden om een van de twee protocollen te gebruiken door de optie `-1` of `-2` voor

respectievelijk versie 1 en versie 2 aan te geven. De mogelijkheid versie 1 te gebruiken blijft in de cliënt bestaan om compatibiliteit met oudere versies te behouden.

13.10.4. Veilig kopiëren

Het commando scp(1) (secure copy) werkt gelijk aan rcp(1). Het kopieert een bestand van of naar een andere machine, maar doet dat veilig.

```
#  scp user@example.com:/COPYRIGHT COPYRIGHT
user@example.com's password: *******
COPYRIGHT            100% |*****************************|  4735
00:00
#
```

Omdat de vingerafdruk al is opgeslagen voor deze host in het vorige voorbeeld, is die al geverifieerd als scp(1) gebruik wordt.

De argumenten die aan scp(1) gegeven worden zijn vrijwel gelijk aan die voor cp(1) met het bestand of de bestanden als het eerste argument en de bestemming als het tweede. Omdat het bestand over het netwerk gaat, door SSH, hebben een of meer van de bestandsargumenten de vorm `user@host:<path_to_remote_file>`.

13.10.5. Instellen

Het instellingenbestand dat voor het hele systeem geldt voor zowel de OpenSSH daemon als cliënt staat in de map /etc/ssh.

ssh_config bevat de instellingen voor de cliënt en sshd_config bevat ze voor de daemon.

Daarnaast bieden het `sshd_program` (standaard /usr/sbin/sshd) en `sshd_flags`rc.conf opties nog meer mogelijkheden voor instellingen.

13.10.6. ssh-keygen

In plaats van het gebruik van wachtwoorden kan ssh-keygen(1) gebruikt worden om DSA en RSA sleutels te maken om een gebruiker te autenticeren:

```
% ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_dsa):
Created directory '/home/user/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_dsa.
Your public key has been saved in /home/user/.ssh/id_dsa.pub.
The key fingerprint is:
bb:48:db:f2:93:57:80:b6:aa:bc:f5:d5:ba:8f:79:17 user@host.example.com
```

ssh-keygen(1) maakt een publiek en privaat sleutelpaar aan dat gebruikt kan worden voor autenticatie. De private sleutel staat opgeslagen in ~/.ssh/id_dsa of ~/.ssh/id_rsa en de publieke sleutel staat in ~/.ssh/id_dsa.pub of ~/.ssh/id_rsa.pub voor respectievelijk sleuteltypen DSA en RSA. De publieke sleutel moet voor beide RSA- of DSA-sleutels in het bestand ~/.ssh/authorized_keys van de andere machine staan om dit te laten werken.

Nu is het mogelijk een verbinding te maken met een andere machine die gebaseerd is op SSH sleutels in plaats van op wachtwoorden.

Als er een wachtwoordzin is gebruikt bij ssh-keygen(1) dan wordt de gebruiker iedere keer dat de private sleutel wordt gebruikt een wachtwoord gevraagd. ssh-agent(1) kan het ongemak van steeds opnieuw een lange wachtwoordzin moeten ingeven verlichten en wordt beschreven in het onderdeel ssh-agent en ssh-add.



Afhankelijk van de gebruikte versie van OpenSSH kunnen opties en bestanden verschillen. Het is verstandig de handleiding ssh-keygen(1) te raadplegen.

13.10.7. ssh-agent en ssh-add

De hulpprogramma's ssh-agent(1) en ssh-add(1) bieden de mogelijkheid om SSH in het geheugen te laden zodat niet iedere keer de wachtwoordzin ingegeven hoeft te worden.

Het hulpprogramma ssh-agent(1) handelt de autenticatie af voor de geheime sleutels die erin geladen zijn. ssh-agent(1) wordt gebruikt om andere programma's te starten. Bij eenvoudig gebruik kan er een shell mee gestart worden of meer complex een schermbeheerprogramma.

Voordat ssh-agent(1) in een shell gebruikt kan worden dient het eerst gestart te worden met een shell als argument. Daarna kan de identiteit toegevoegd worden daar ssh-add(1) aan te roepen en de wachtwoordzin voor de geheime sleutel op te geven. Als deze stappen zijn voltooid kan een gebruiker met ssh(1) naar iedere host waar de corresponderende publieke sleutel is geïnstalleerd:

```
% ssh-agent csh
% ssh-add
Enter passphrase for /home/user/.ssh/id_dsa:
Identity added: /home/user/.ssh/id_dsa (/home/user/.ssh/id_dsa)
%
```

Om ssh-agent(1) te gebruiken in X11 dient er een verwijzing naar ssh-agent(1) in ~/.xinitrc te staan. Dan zijn de diensten van ssh-agent(1) beschikbaar voor alle programma's die in X11 gestart worden. Een ~/.xinitrc zou er als volgt uit kunnen zien:

```
exec ssh-agent startxfce4
```

Hiermee wordt ssh-agent(1) gestart die op zijn beurt XFCE start, iedere keer dat X11 start. Als dat is gebeurd en X11 is herstart zodat de wijzigingen actief zijn, dan kan eenvoudigweg ssh-add(1) gestart worden om alle beschikbare SSH sleutels te laden.

13.10.8. SSH tunnels

OpenSSH kan een tunnel maken waarin een ander protocol ingepakt kan worden zodat er een versleutelde sessie ontstaat.

Het volgende commando geeft ssh(1) aan dat er een tunnel voor telnet gemaakt moet worden:

```
% ssh -2 -N -f -L 5023:localhost:23 user@foo.example.com
%
```

Aan het `ssh` commando worden de volgende opties meegegeven:

`-2`
: Dit dwingt `ssh` om versie 2 van het protocol te gebruiken. Gebruik van deze optie wordt afgeraden als er verbinding wordt gemaakt met oudere SSH servers.

`-N`
: Dit geeft aan dat er geen commando volgt, maar dat er een tunnel opgezet moet worden. Als deze optie niet aanwezig was, zou `ssh` een normale sessie starten.

`-f`
: Dit dwingt `ssh` om in de achtergrond te draaien.

`-L`
: Dit geeft aan dat de lokaal een tunnel wordt gemaakt in de vorm *lokale_poort:netwerk_host:netwerk_poort*.

`user@foo.example.com`
: Wijst naar een gebruiker op de SSH server op het netwerk.

Een SSH tunnel werkt doordat een luistersocket wordt gemaakt op `localhost` op de aangegeven poort. Die stuurt dan iedere ontvangen verbinding op de lokale host/poort via de SSH verbinding door naar de aangegeven host en poort op het netwerk.

In het voorbeeld wordt poort *5023* op `localhost` doorgestuurd naar poort *23* op `localhost` van de machine op het netwerk. Omdat `23` telnet is, zou dit een veilige telnet verbinding opleveren door een SSH tunnel.

Dit kan gebruikt worden om ieder willekeurig onveilig TCP protocol in te pakken als SMTP, POP3, FTP, etc.

Voorbeeld 8. SSH gebruiken om een veilige tunnel te maken voor SMTP

```
% ssh -2 -N -f -L 5025:localhost:25 user@mailserver.example.com
user@mailserver.example.com's password: *****
% telnet localhost 5025
Trying 127.0.0.1...
Connected to localhost.
```

```
Escape character is '^]'.
220 mailserver.example.com ESMTP
```

Dit kan samen met een ssh-keygen(1) en extra gebruikersaccounts gebruikt worden om een min of meer naadloze en eenvoudige SSH tunnelomgeving te maken. In plaats van wachtwoorden kunnen sleutels gebruikt worden en de tunnels kunnen in de omgeving van een aparte gebruiker draaien.

Praktische voorbeelden van een SSH tunnel

Veilige toegang tot een POP3 server

Op het werk staat een SSH server die verbindingen van buitenaf toestaat. Op hetzelfde netwerk op kantoor staat een mailserver waarop POP3 draait. Het netwerk of het netwerkpad tussen de locatie op Internet en kantoor is wellicht niet helemaal te vertrouwen. Om deze reden dient de mailserver op een veilige manier benaderd te worden. De oplossing is een SSH verbinding opzetten naar de SSH server op kantoor en dan door de tunnel heen een verbinding opzetten met de mailserver.

```
% ssh -2 -N -f -L 2110:mail.example.com:110 user@ssh-server.example.com
user@ssh-server.example.com's password:
% ******
```

Als de tunnel eenmaal draait, dan kan de mailcliënt naar `localhost` poort 2110 gewezen worden. Alle verbinding naar die poort worden veilig doorgestuurd door de tunnel naar `mail.example.com`.

Een draconische firewall omzeilen

Sommige netwerkbeheerders stellen draconische firewallregels op en filteren niet alleen inkomende verbindingen, maar ook uitgaande. Meestal mag dan alleen maar verbinding gemaakt worden met andere machines op poorten 22 en 80 voor SSH en websurfen.

Soms wil een gebruiker dan toch toegang krijgen tot andere (wellicht niet netwerkgerelateerde) diensten, zoals een Ogg Vorbis server om muziek te streamen. Als die Ogg Vorbis server streamt op een andere poort dan 22 of 80, dan kan deze niet bereikt worden.

De oplossing ligt in het opzetten van een SSH verbinding naar een machine buiten de firewall en die tunnel te gebruiken om bij de Ogg Vorbis server te komen.

```
% ssh -2 -N -f -L 2110:mail.example.com:110 user@ssh-server.example.com
user@ssh-server.example.com's password: ******
```

De streamingcliënt kan nu gewezen worden naar `localhost` poort 8888 vanwaar er wordt doorverwezen naar `music.example.com` poort 8000 en zo wordt de firewall succesvol ontwerken.

13.10.9. De optie `AllowUsers`

Vaak is het verstandig om beperkingen aan te brengen op het gebied van welke gebruikers kunnen aanmelden en van waar. De optie `AllowUsers` biedt deze mogelijkheid. Om bijvoorbeeld alleen `root` toe te staan zich aan te melden van `192.168.1.32`, kan iets als de volgende regel worden opgenomen in het bestand /etc/ssh/sshd_config:

```
AllowUsers root@192.168.1.32
```

Om de gebruiker `admin` het recht te geven zich van overal aan te melden hoeft alleen de gebruikersnaam vermeld te worden:

```
AllowUsers admin
```

Meerdere gebruikers met rechten of beperkingen horen op dezelfde regel te staan:

```
AllowUsers root@192.168.1.32 admin
```



Het is van belang dat iedere gebruiker die zich moet kunnen aanmelden wordt genoemd. De overige gebruikers worden buitengesloten.

Nadat er wijzigingen zijn gemaakt aan /etc/ssh/sshd_config dienen de bestanden in sshd(8) geladen te worden:

```
# service sshd reload
```

13.10.10. Meer informatie

OpenSSH

ssh(1) scp(1) ssh-keygen(1) ssh-agent(1) ssh-add(1) ssh_config(5)

sshd(8) sftp-server(8) sshd_config(5)

13.11. Bestandssysteem toegangscontrolelijsten (ACLs)

In combinatie met verbeteringen als snapshots, biedt FreeBSD de veiligheid van Toegangscontrolelijsten voor Bestandssystemen (Access Control Lists, ACLs).

Met toegangscontrolelijsten wordt het standaard UNIX® rechtenmodel uitgebreid op een zeer verenigbare (POSIX®.1e) manier. Deze methodes stellen een beheerder in staat om gebruik te maken en voordeel te halen uit een geraffineerder beveiligingsmodel.

Om ondersteuning voor ACLs voor bestandssystemen in te schakelen dient het volgende in de kernel gecompileerd te worden:

```
options UFS_ACL
```

Als deze optie niet aanwezig is, dan wordt er een waarschuwing weergegeven als er wordt geprobeerd een bestandssysteem aan te koppelen dat gebruik maakt van ACLs. Deze optie is al geactiveerd in de GENERIC kernel. ACLs zijn afhankelijk van uitgebreide attributen die zijn ingeschakeld op het bestandssysteem. Uitgebreide attributen worden standaard ondersteund in het volgende generatie UNIX® bestandssysteem UFS2.



Er is meer administratieve rompslomp nodig om uitgebreide attributen in te stellen op UFS1 dan op UFS2. De prestaties van uitgebreide attributen zijn op UFS2 ook veel beter. Daarom wordt UFS2 ook meestal aangeraden boven UFS1 bij het gebruik van toegangscontrolelijsten.

ACLs worden ingeschakeld door de beheersvlag `acls` op het moment van aankoppelen. Dit kan ook in /etc/fstab staan. De vlag op het moment van aankoppelen kan ook automatisch gezet worden op een persistente wijze met tunefs(8) door een superblok in de bestandssysteemkop te wijzigen. In het algemeen wordt de voorkeur gegeven aan de vlag in het superblok om een aantal redenen:

- De ACLs vlag op het moment van aankoppelen kan niet gewijzigd worden bij opnieuw aankoppelen (mount(8) `-u`), maar alleen door een volledige umount(8) en een verse mount(8). Dit betekent dat ACLs niet ingeschakeld kunnen worden op root-bestandssysteem na het opstarten. Het betekent ook dat de aard van een bestandssysteem niet veranderd kan worden als het eenmaal in gebruik is.
- Het inschakelen van de superblokvlag zorgt ervoor dat het bestandssysteem altijd wordt aangekoppeld met de ACLs ingeschakeld, zelfs als het niet in fstab staat of als de apparaten van plaats veranderen. Hiermee wordt voorkomen dat het bestandssysteem wordt gebruikt zonder dat ACLs ingeschakeld zijn, wat ervoor zou kunnen zorgen dat ACLs onjuist worden toegepast wat weer kan zorgen voor beveiligingsproblemen.



Wellicht wordt het mogelijk om de ACLs via de vlag in te schakelen zonder een compleet verse mount(8), maar de ontwikkelaars vinden het wenselijk om het per ongeluk zonder ACLs aankoppelen te ontmoedigen, omdat er bijzonder vervelende gevolgen kunnen zijn als ACLs worden ingeschakeld, daarna worden uitgezet en weer worden ingeschakeld zonder dat de uitgebreide attributen worden geschoond. In het algemeen geldt dat als ACLs eenmaal zijn ingeschakeld voor een bestandssysteem, ze niet meer uitgeschakeld moeten worden, omdat de resulterende bestandsbescherming wellicht niet compatibel is met dat wat gebruikers van het systeem nodig hebben en het opnieuw aanzetten van ACLs kan leiden tot het opnieuw koppelen van voorheen bestaande ACLs aan bestanden waarvoor de toegangsrechten sindsdien zijn aangepast, wat kan leiden tot onverwachte situaties.

Bestandssystemen waarvoor ACLs zijn ingeschakeld worden weergegeven met een `+` (plus) teken als de toegangsrechten worden bekeken:

```
drwx------  2 robert  robert  512 Dec 27 11:54 private
```

```
drwxrwx---+ 2 robert  robert  512 Dec 23 10:57 directory1
drwxrwx---+ 2 robert  robert  512 Dec 22 10:20 directory2
drwxrwx---+ 2 robert  robert  512 Dec 27 11:57 directory3
drwxr-xr-x  2 robert  robert  512 Nov 10 11:54 public_html
```

Hierboven is te zien dat mappen directory1, directory2 en directory3 allemaal gebruik maken van ACLs. De map public_html doet dat niet.

13.11.1. Gebruik maken van ACLs

De ACLs van het bestandssysteem kunnen bekeken worden met het hulpprogramma getfacl(1). Om de ACL op het bestand test te bekijken zou het volgende commando nodig zijn:

```
% getfacl test
#file:test
#owner:1001
#group:1001
user::rw-
group::r--
other::r--
```

Om de ACL op dit bestand te wijzigen wordt het hulpprogramma setfacl(1) als volgt gebruikt:

```
% setfacl -k test
```

De vlag `-k` verwijdert alle bestaande ACLs van een bestand of bestandssysteem. De methode die de voorkeur geniet is `-b` gebruiken omdat die optie de basisvelden die nodig zijn voor het laten werken van de ACLs laat staan.

```
% setfacl -m u:trhodes:rwx,group:web:r--,o::--- test
```

Bij het commando hierboven, werd de optie `-m` gebruikt om de standaard ACL aan te passen. Omdat er geen voorgedefinieerde instellingen waren, die waren verwijderd door het commando daarvoor, werden nu de standaardinstellingen hersteld en de rechten die werden aangegeven toegevoegd. Let op dat bij het toevoegen van een gebruiker of een groep die niet bekend is op het systeem een foutmelding `Invalid argument` wordt geschreven naar stdout.

13.12. Monitoren van beveiligingsproblemen met andere software

In de afgelopen jaren zijn er in de beveiligingswereld veel vorderingen gemaakt op het gebied van inzicht in kwetsbaarheden. Als er software naast het besturingssysteem wordt geïnstalleerd en ingesteld neemt op vrijwel ieder besturingssysteem het risico op inbraak toe.

Inzicht in kwetsbaarheid is een vitale factor in beveiliging en hoewel FreeBSD waarschuwingen

publiceert voor het basissysteem, gaat het publiceren van waarschuwingen voor alle overige software de scope van het FreeBSD Project te buiten. Er is een manier om inzicht te krijgen in de kwetsbaarheden voor additionele software en als beheerder gewaarschuwd te worden. Voor dit doel bestaat het FreeBSD hulpprogramma Portaudit.

De port ports-mgmt/portaudit zoekt naar bekende beveiligingsproblemen in een database die wordt bijgewerkt en onderhouden door het FreeBSD Security Team en portontwikkelaars.

Voordat Portaudit gebruikt kan worden dient het geïnstalleerd te worden uit de Portscollectie:

```
# cd /usr/ports/ports-mgmt/portaudit && make install clean
```

Tijdens het installatieproces worden de instellingenbestanden voor periodic(8) bijgewerkt, waardoor Portaudit uitvoer in de dagelijkse security runs meekomt. Het is van belang dat de emails die aan de emailaccount van `root` worden gezonden en uit de dagelijkse beveiligingsronde komen ook echt worden gelezen. Er zijn geen verdere instellingen nodig.

Na de installatie kan de beheerder de database bijwerken en bekende kwetsbaarheden in geïnstalleerde pakketten bekijken met het volgende commando:

```
# portaudit -Fda
```



De database wordt automatisch bijgewerkt tijdens de periodic(8) run; dus het voorgaande commando is volledig optioneel. Het is alleen nodig om de volgende voorbeelden na te kunnen doen.

De software de uit de Portscollectie is geïnstalleerd kan op elk moment door een beheerder ge-audit worden met:

```
# portaudit -a
```

Portaudit zal iets als het volgende produceren voor kwetsbare pakketten:

```
Affected package: cups-base-1.1.22.0_1
Type of problem: cups-base -- HPGL buffer overflow vulnerability.
Reference: <http://www.FreeBSD.org/ports/portaudit/40a3bca2-6809-11d9-a9e7-
0001020eed82.html>

1 problem(s) in your installed packages found.

You are advised to update or deinstall the affected package(s) immediately.
```

Door met een webbrowser naar de aangegeven URL te gaan kan een beheerder meer informatie over de bewust kwetsbaarheid krijgen, waaronder de versies die het betreft, volgens de FreeBSD Port versie en andere websites waarop beveiligingswaarschuwingen te lezen zijn.

In het kort is Portaudit een krachtig hulpprogramma dat bijzonder handig is als het wordt gekoppeld aan het gebruik van de port Portupgrade.

13.13. FreeBSD beveiligingswaarschuwingen

Net als veel andere kwalitatief goede productiebesturingssystemen publiceert FreeBSD "Beveiligingswaarschuwingen". Deze waarschuwingen worden meestal pas naar de beveiligingslijst gemaild en gedocumenteerd in de Errata als de van toepassing zijnde uitgaven gepatcht zijn. In deze paragraaf wordt toegelicht wat een waarschuwing is, hoe die te begrijpen en welke maatregelen er genomen moeten worden om een systeem bij te werken.

13.13.1. Hoe ziet een waarschuwing eruit?

De FreeBSD beveiligingswaarschuwingen zien er ongeveer uit als die hieronder die van de FreeBSD beveiligingswaarschuwingen mailinglijst mailinglijst komt.

```
=============================================================================
FreeBSD-SA-XX:XX.UTIL                                          Security Advisory
                                                               The FreeBSD Project

Topic:          denial of service due to some problem ①

Category:       core ②
Module:         sys ③
Announced:      2003-09-23 ④
Credits:        Person ⑤
Affects:        All releases of FreeBSD ⑥
                FreeBSD 4-STABLE prior to the correction date
Corrected:      2003-09-23 16:42:59 UTC (RELENG_4, 4.9-PRERELEASE)
                2003-09-23 20:08:42 UTC (RELENG_5_1, 5.1-RELEASE-p6)
                2003-09-23 20:07:06 UTC (RELENG_5_0, 5.0-RELEASE-p15)
                2003-09-23 16:44:58 UTC (RELENG_4_8, 4.8-RELEASE-p8)
                2003-09-23 16:47:34 UTC (RELENG_4_7, 4.7-RELEASE-p18)
                2003-09-23 16:49:46 UTC (RELENG_4_6, 4.6-RELEASE-p21)
                2003-09-23 16:51:24 UTC (RELENG_4_5, 4.5-RELEASE-p33)
                2003-09-23 16:52:45 UTC (RELENG_4_4, 4.4-RELEASE-p43)
                2003-09-23 16:54:39 UTC (RELENG_4_3, 4.3-RELEASE-p39) ⑦
CVE Name:   CVE-XXXX-XXXX ⑧

For general information regarding FreeBSD Security Advisories,
including descriptions of the fields above, security branches, and the
following sections, please visit
http://www.FreeBSD.org/security/.

I.   Background ⑨

II.  Problem Description ⑩

III. Impact ⑪
```

```
IV.  Workaround ⑫

V.   Solution ⑬

VI.  Correction details ⑭

VII. References ⑮
```

① Het veld `Topic` geeft aan wat precies het probleem is. Het is eigenlijk een inleiding op de beveiligingswaarschuwing en geeft aan welke programma kwetsbaar is.

② Het veld `Category` geeft aan welk onderdeel van het systeem kwetsbaar is. Dat kan een van de onderdelen `core`, `contrib` of `ports` zijn. De categorie `core` betekent dat de een kerncomponent van het FreeBSD besturingssysteem kwetsbaar is. De categorie `contrib` betekent dat software die toegevoegd is aan het FreeBSD Project kwetsbaar is, zoals sendmail. Tenslotte geeft de categorie `ports` aan dat een optionele component uit de Portscollectie kwetsbaar is.

③ Het veld `Module` geeft aan waar de component zich bevindt, bijvoorbeeld `sys`. In dit voorbeeld wordt het duidelijk dat de module `sys` kwetsbaar is. Hier gaat het dus om een kwetsbaar component die gebruikt wordt in de kernel.

④ Het veld `Announced` geeft aan wanneer de beveiligingswaarschuwing gepubliceerd of aangekondigd is. Dit betekent dat het beveiligingsteam heeft bevestigd dat het probleem bestaat en dat er een patch is gecommit in het depot met de broncode van FreeBSD.

⑤ In het veld `Credits` wordt iemand of een organisatie bedankt die de kwetsbaarheid heeft ontdekt en gerapporteerd.

⑥ Het veld `Affects` geeft aan welke uitgaven van FreeBSD door deze kwetsbaarheid worden getroffen. Voor de kernel kan snel gekeken worden naar de uitvoer van `ident` voor de betreffende bestanden om te bepalen welke revisie ze hebben. Voor ports is het versienummer te zien in /var/db/pkg. Als het systeem niet gelijk op loopt met het FreeBSD Subversion-depot en dagelijks herbouwd wordt, dan is de kans groot dat het systeem kwetsbaar is.

⑦ Het veld `Corrected` geeft de datum, tijd en tijdzone aan en de uitgave die is aangepast.

⑧ Gereserveerd voor de identificatie-informatie die gebruikt wordt om kwetsbaarheden in het Common Vulnerabilities Database System op te zoeken.

⑨ Het veld `Background` geeft meer informatie over wat er precies aan de hand is. Meestal staat hier waarom het programma aanwezig is in FreeBSD, waar het voor gebruikt wordt en hoe het programma is ontstaan.

⑩ Het veld `Problem Description` geeft gedetailleerde toelichting op het beveiligingsprobleem. Hier kan informatie bij staat over programmacode die fouten bevat of zelfs hoe het programma gebruikt kan worden om een beveiligingsgat te openen.

⑪ Het veld `Impact` beschrijft welke invloed het probleem kan hebben op het systeem. Dit kan bijvoorbeeld een ontzegging van dienst aanval zijn, gebruikers extra rechten geven of het verkrijgen van supergebruiker toegang voor de aanvaller zijn.

⑫ Het veld `Workaround` geeft aan hoe het mogelijk is het probleem te omzeilen (workaround) in het geval systeembeheerders niet in staat zijn om het systeem bij te werken. Dit zou te maken kunnen hebben met de tijd, beschikbaarheid van het netwerk en een hele lijst met andere

redenen. Hoe dan ook, beveiliging dient serieus genomen te worden en een systeem dat kwetsbaar is moet bijgewerkt worden of het gat in de beveiliging moet gedicht worden met de alternatieve oplossing.

⑬ Het veld `Solution` geeft instructies over hoe een systeem aangepast kan worden. Dit is een werkinstructie die getest en gecontroleerd is om een systeem aan te passen en weer veilig werkend te krijgen.

⑭ In het veld `Correction Details` staan de Subversion-takken of uitgavenamen, met de punten veranderd in een liggend streepje. Er staat ook welke revisienummer de aangetaste bestanden binnen een tak hebben.

⑮ In het veld `References` wordt gewoonlijk verwezen naar andere bronnen. Dit kunnen web-URLs, boeken, mailinglijsten en nieuwsgroepen zijn.

13.14. Procesaccounting

Procesaccounting is een beveiligingsmethode die een beheerder in staat stelt om in de gaten te houden welke systeembronnen worden gebruikt, hoe ze over gebruikers verdeeld zijn, systeemmonitoring biedt en op minimalistische wijze het gebruik van commando's door gebruikers volgt.

Deze methode heeft voordelen en nadelen. Eén van de positieve punten is dat een inbraak gevolgd kan worden tot het moment waarop die zich voordeed. Nadelen zijn de grootte van de logboeken die door procesaccounting worden gegenereerd en de schijfruimte die dat kost. In dit onderdeel wordt een beheerder de basis van procesaccounting getoond.

13.14.1. Procesaccounting inschakelen en gebruiken

Voordat procesaccounting gebruikt kan worden dient het te worden ingeschakeld met de volgende commando's:

```
# touch /var/account/acct
# accton /var/account/acct
# echo 'accounting_enable="YES"' >> /etc/rc.conf
```

Eenmaal ingeschakeld begint accounting met het bijhouden van CPU statistieken, commando's, enzovoort. Alle accounting logboeken worden in een niet leesbaar formaat bijgehouden en zijn uit te lezen met sa(8). Bij het uitvoeren zonder opties, toont `sa` informatie gerelateerd aan het aantal aanroepen per gebruiker, de totale tijd in minuten die is verstreken, de totale CPU- en gebruikerstijd in minuten, gemiddeld aantal I/O operaties, enzovoort.

Informatie over uitgevoerde commando's kan bekeken worden met lastcomm(1). Zo kan met `lastcomm` bijvoorbeeld weergegeven worden welke commando's door gebruikers op een specifieke ttys(5) zijn uitgevoerd:

```
# lastcomm ls trhodes ttyp1
```

Het bovenstaande commando toont ieder bekend gebruikt van `ls` door de gebruiker `trhodes` op terminal `ttyp1`.

Veel andere handige opties staan beschreven in lastcomm(1), acct(5) en sa(8).

Hoofdstuk 14. Jails

14.1. Overzicht

Dit hoofdstuk levert een uitleg van wat FreeBSD jails zijn en hoe ze gebruikt kunnen worden. Jails, soms ook wel bekend als een verbeterde vervanging van *chroot omgevingen*, zijn een erg krachtige tool voor systeem beheerders, maar het standaard gedrag kan ook interessant zijn voor gevorderde gebruikers.



Jails zijn een krachtig gereedschap, maar zijn geen zilveren kogel qua beveiliging. Hoewel het belangrijk is om op te merken dat het onmogelijk is voor een gevangen proces om zelf te ontsnappen, zijn er verschillende manieren waarop een ongeprivilegieerde gebruiker buiten een jail kan samenwerken met een geprivilegieerde gebruiker binnen de jail en daarmee verhoogde privileges kan krijgen in de gastheeromgeving.

De meeste van deze aanvallen kunnen worden voorkomen door ervoor te zorgen dat de jail-root niet beschikbaar is voor ongeprivilegieerde gebruikers binnen de gastheeromgeving. Buiten dat geldt als algemene regel dat onvertrouwde gebruikers met geprivilegieerde toegang tot een jail geen toegang tot de gastheeromgeving moet worden gegeven.

Na het lezen van dit hoofdstuk weet de lezer:

- Wat een jail is, en welk doel het kan dienen in een FreeBSD installatie.
- Hoe men een jail opbouwt, start en stopt.
- De basis over jail beheer, zowel van binnen in de jail, als van buitenaf.

Andere bronnen met nuttige informatie over jails zijn:

- De jail(8) handleiding. Hier kan de volledige referentie gevonden worden van het `jail` commando - de administratieve tool die in FreeBSD gebruikt kan worden om FreeBSD jails mee te beheren, te starten en te stoppen.
- De mailinglijsten en de archieven hiervan. De archieven van de FreeBSD algemene vragen mailinglijst en andere mailing lijsten die gehost worden door de FreeBSD nlijstserver bevatten reeds een rijke bron van informatie over jails. Het zou altijd aantrekkelijk moeten zijn om informatie in de archieven te zoeken, of een nieuwe vraag stellen aan de freebsd-questions mailinglijst.

14.2. Termen en begrippen van jails

Om een beter begrip te geven over de onderdelen van FreeBSD die gerelateerd zijn aan jails, de werking ervan, en hoe ze omgaan met de rest van FreeBSD worden de volgende termen gebruikt in het hoofdstuk:

chroot(8) (commando)

Hulpmiddel dat de systeemaanroep chroot(2) van FreeBSD gebruikt om de rootmap van een proces en alle afstammelingen te veranderen.

chroot(2) (omgeving)

Een omgeving van processen die draaien in een "chroot". Dit is inclusief bronnen die gebruikt worden, zoals bijvoorbeeld het gedeelte van het bestandssysteem dat zichtbaar is, de gebruiker en groep ID's welke beschikbaar zijn, netwerkkaarten en andere IPC-mechanismes, etcetera.

jail(8) (commando)

De systeem utility die het mogelijk maakt om processen binnenin een jail te starten.

host (systeem, processen, gebruiker, etc.)

Het controlerende systeem van een jail omgeving. Het host systeem heeft toegang tot alle beschikbare hardware bronnen en kan processen controleren zowel buiten als binnenin een jail. Één van de belangrijkste verschillen van het host systeem met een jail zijn de limitaties die van toepassing zijn op super-gebruiker processen binnenin een jail, niet geforceerd worden voor processen in het host systeem.

hosted (systeem, processen, gebruiker, etc.)

Een proces, gebruiker, of andere entiteit wiens toegang tot bronnen is gelimiteerd door een FreeBSD jail.

14.3. Introductie

Omdat systeem beheer een complexe en enorme taak is, zijn er vele sterke tools ontwikkeld om het leven van een systeem beheerder makkelijker te maken. Deze tools leveren meestal verbeteringen op de manier waarop systemen worden geïnstalleerd, geconfigureerd en onderhouden. Een deel van de taken waarvan verwacht wordt dat die uitgevoerd wordt door de systeem beheerder is het goed configureren van de beveiliging van een systeem, zodat het kan blijven doorgaan met het serveren van de taak, zonder dat er beveiligingsproblemen optreden.

Één van de tools welke gebruikt kan worden om de beveiliging van een FreeBSD systeem te verbeteren zijn *jails*. Jails zijn geïntroduceerd in FreeBSD 4.X door Poul-Henning Kamp <phk@FreeBSD.org>, maar zijn grotendeels verbeterd in FreeBSD 5.X om ze nog sterker en krachtiger te maken. De ontwikkeling gaat nog steeds door met verbeteringen voor het gebruik, performance, betrouwbaarheid en beveiliging.

14.3.1. Wat is een jail

BSD achtige systemen hebben sinds 4.2-BSD ondersteuning voor chroot(2). De chroot(8) utility kan gebruikt worden om de root directory van een set processen te wijzigen waardoor een veilige omgeving wordt gecreeërd voor de rest van het systeem. Processen die gemaakt worden in een chroot omgeving kunnen bestanden en bronnen daarbuiten niet benaderen. Daardoor zou een compromitering van een dienst die in een chroot omgeving draait niet direct betekenen dat het hele systeem gecompromiteerd is. De chroot(8) utility is goed genoeg voor simpele taken, waarbij flexibiliteit en geavanceerde en complexe opties niet nodig zijn. Sinds het uitvinden van het chroot concept, zijn er vele mogelijkheden gevonden om hieruit te kunnen komen en alhoewel ze

verbeterd zijn in moderne versies van FreeBSD, werd het duidelijk dat chroot(2) niet de meest ideale oplossing was voor het beveiligen van diensten. Er moest een nieuw subsysteem ontwikkeld worden.

Dit is één van de redenen waarom jails zijn ontwikkeld.

Jails zijn een verbeterd concept van de chroot(2) omgeving, in verschillende opzichten. In een traditionele chroot(2) omgeving worden processen alleen gelimiteerd in het deel van het bestandssysteem die ze kunnen benaderen. De rest van de systeem bronnen (zoals de set van systeem gebruikers, de draaiende processen of het netwerk subsysteem) worden gedeeld door het chrooted proces en de processen op het host systeem. Jails breiden dit model uit door het niet alleen virtualizeren van de toegang tot het bestandssysteem maar ook tot de set van gebruikers, het netwerk subsysteem van de FreeBSD kernel en een aantal andere delen. Een meer complete set van gespecificeerde controle mogelijkheden die beschikbaar zijn voor het personaliseren van de toegang tot een jail omgeving wordt beschreven in Optimaliseren en administratie.

Een jail heeft vier kenmerken:

- Een eigen directory structuur - het startpunt van waaruit een jail benaderd wordt. Zodra men in de jail zit, mogen processen niet buiten deze directory structuur komen. Traditionele problemen die chroot(2)'s ontwerp getart hebben, hebben geen invloed op FreeBSD jails.
- Een hostname - de hostnaam die gebruikt wordt in de jail. Jails worden met name gebruikt voor het hosten van netwerk diensten, daardoor kan het de systeembeheerder heel erg helpen als er beschrijvende hostnames worden gekozen.
- Een IP adres - deze wordt gekoppeld aan de jail en kan op geen enkele manier worden gewijzigd tijdens het leven van de jail. Het IP adres van een jail is meestal een alias op een reeds bestaande netwerk interface, maar dit is niet noodzakelijk.
- Een commando - het padnaam van een uitvoerbaar bestand in de jail. Deze is relatief aan de rootdirectory van de jail omgeving en verschilt per situatie, afhankelijk van het type van de specifieke jail omgeving.

Buiten deze kenmerken, kunnen jails hun eigen set aan gebruikers en `root` gebruiker hebben. Uiteraard zijn de mogelijkheden van de `root` gebruiker beperkt tot de jail omgeving en, vanuit het host systeem gezien, is de `root` gebruiker geen super-gebruiker. Daarnaast is het de `root` gebruiker in een jail omgeving niet toegestaan om kritieke operaties uit te voeren op het systeem buiten de gedefinieerde jail omgeving. Meer informatie over de mogelijkheden en beperkingen van de `root` gebruiker kan gevonden worden in Optimaliseren en administratiehieronder.

14.4. Creeëren en controleren van jails

Sommige beheerders kiezen ervoor om jails op te delen in de volgende twee types: "complete" jails, welke een volledig FreeBSD systeem emuleert en "service" jails, gericht op één applicatie of dienst, mogelijkerwijs draaiende met privileges. Dit is alleen een conceptuele splitsing, de manier van het opbouwen van een jail wordt hierdoor niet veranderd. De jail(8) handleiding is heel duidelijk over de procedure voor het maken van een jail:

```
# setenv D /here/is/the/jail
```

```
# mkdir -p $D ①
# cd /usr/src
# make buildworld ②
# make installworld DESTDIR=$D ③
# make distribution DESTDIR=$D ④
# mount -t devfs devfs $D/dev ⑤
```

① Het selecteren van een locatie voor een jail is het beste beginpunt. Hier zal de jail fysiek te vinden zijn binnen het bestandssysteem van het host systeem. Een goede keuze kan /usr/jailjailnaam zijn, waar *jailnaam* de naam is van de jail. Het /usr bestandssysteem heeft meestal genoeg ruimte voor het jail bestandssysteem, wat voor een "complete" jail betekend dat het eigenlijk een replica is van elk bestand dat standaard aanwezig is binnen het FreeBSD basissysteem.

② Als u uw userland al heeft herbouwd met `make world` of `make buildworld`, dan kunt u deze stap overslaan en uw bestaande userland in de nieuwe jail installeren.

③ Dit commando zal de gekozen fysieke directory vullen met de benodigde binaire bestanden, bibliotheken, handleidingen, etc.

④ Het `distribution` doel voor make installeert elk benodigd configuratie bestand. In simpelere termen, het installeert alle installeerbare bestanden in /usr/src/etc naar de /etc directory van de jail omgeving: $D/etc.

⑤ Het koppelen van het devfs(8) bestandssysteem is niet vereist in een jail. Aan de andere kant, vrijwel elke applicatie heeft toegang nodig tot minstens één apparaat, afhankelijk van het doel van het programma. Het is erg belangrijk om toegang tot apparaten te controleren binnenin een jail, omdat incorrecte instellingen een aanvaller de mogelijkheid kunnen geven om vervelende dingen in de jail te doen. De controle over devfs(8) wordt gedaan door middel van rulesets, welke beschreven worden in de devfs(8) en devfs.conf(5) handleidingen.

Zodra een jail is geïnstalleerd, kan het opgestart worden door de jail(8) applicatie. De jail(8) applicatie heeft vier benodigde argumenten welke beschreven worden in Wat is een jail. Er kunnen ook andere argumenten gebruikt worden, om bijvoorbeeld de jail te starten met de instellingen van een specifieke gebruiker. Het `commando` argument hangt af van het type jail, voor een *virtueel systeem* is /etc/rc een goede keuze, omdat het de reguliere opstart procedure nabootst van een FreeBSD systeem. Voor een *dienst* jail is het geheel afhankelijk van de dienst of applicatie die in de jail gaat draaien.

Jails worden over het algemeen gestart tegelijkertijd met de rest van het systeem. Het FreeBSD rc mechanisme levert een makkelijke manier om dat te doen:

1. Een lijst van jails die opgestart moeten worden tijdens het opstarten van het systeem, moeten worden toegevoegd aan het rc.conf(5) bestand:

   ```
   jail_enable="YES"   # Stel dit in op NO om te voorkomen dat er jails gestart
   worden
   jail_list="www"     # Door spaties gescheiden lijst van jails
   ```



De jail namen in `jail_list` mogen alleen alfanumerieke karakters bevatten.

2. Voor elke jail die gespecificeerd is in `jail_list` moet een groep van rc.conf(5) instellingen worden toegevoegd:

```
jail_www_rootdir="/usr/jail/www"     # de hoofd directory van de jail
jail_www_hostname="www.example.org"  # de hostnaam van de jail
jail_www_ip="192.168.0.10"           # het IP adres van de jail
jail_www_devfs_enable="YES"          # moet devfs wel of niet gekoppeld worden
in de jail
jail_www_devfs_ruleset="www_ruleset" # welke devfs ruleset gebruikt moet worden
voor de jail
```

   De standaard opstart variabelen in rc.conf(5) gebruiken het /etc/rc bestand om de jail op te starten, wat er vanuit gaat dat de jail een compleet virtueel systeem is. Voor service jails moet het standaard opstart commando worden gewijzigd door het aanpassen van de `jail`*`jailname`*`exec_start` optie.



Voor een complete lijst van beschikbare opties, zie de rc.conf(5) handleiding.

service(8) kan worden gebruikt om jails handmatig te starten en te stoppen, mits er een overeenkomstige verzameling regels bestaat in /etc/rc.conf.

```
# service jail start www
# service jail stop www
```

Er is op dit moment geen nette methode om een jail te stoppen. Dit komt omdat de benodigde applicaties die een nette afsluiting verzorgen, niet beschikbaar zijn in een jail. De beste manier om een jail af te sluiten is door het volgende commando van binnenin de jail uit te voeren of door middel van het jexec(8) commando:

```
# sh /etc/rc.shutdown
```

Meer informatie hierover kan gevonden worden in de jail(8) handleiding.

14.5. Optimaliseren en administratie

Er zijn meerdere opties beschikbaar die ingesteld kunnen worden voor elke jail, en er zijn meerdere mogelijkheden om een FreeBSD host systeem te combineren met jails om een betere scheiding tussen systeem en applicaties te verkrijgen. Deze sectie leert:

- Een aantal opties zijn beschikbaar voor het optimaliseren van het gedrag en beveiligings

beperkingen die geïmplementeerd worden in een jail.

- Een aantal "high-level" applicaties die gebruikt worden voor het beheren van jails, welke beschikbaar zijn via de FreeBSD Ports Collectie en kunnen gebruikt worden om een complete jail-gebaseerde oplossing te creeëren.

14.5.1. Systeem applicaties voor het optimaliseren van jails onder FreeBSD

Het goed kunnen optimaliseren van een jail configuratie wordt veelal gedaan door het instellen van sysctl(8) variabelen. Er bestaat een speciale subtak van sysctl voor het organiseren van alle relevante opties: de `security.jail.*` hierarchie binnen de FreeBSD kernel. Hieronder staat een lijst van de belangrijkste jail-gerelateerde sysctl variabelen, met informatie over de standaard waarden. De benaming zou zelf beschrijvend moeten zijn, maar voor meer informatie kunnen de jail(8) en sysctl(8) handleidingen geraadpleegd worden.

- `security.jail.set_hostname_allowed: 1`
- `security.jail.socket_unixiproute_only: 1`
- `security.jail.sysvipc_allowed: 1`
- `security.jail.enforce_statfs: 2`
- `security.jail.allow_raw_sockets: 0`
- `security.jail.chflags_allowed: 0`
- `security.jail.jailed: 0`

Deze variabelen kunnen door de systeem beheerder gebruikt worden op het *host systeem* om limitaties toe te voegen of te verwijderen, welke standaard opgedwongen worden aan de `root` gebruiker. Let op, een aantal beperkingen kan niet worden aangepast. De `root` gebruiker mag geen bestandssystemen koppelen of ontkoppelen binnenin een jail(8). De `root` gebruiker mag ook geen devfs(8) rulesets laden of ontladen, firewall rules plaatsen of andere taken uitvoeren die vereisen dat de in-kernel data wordt aangepast, zoals het aanpassen van de `securelevel` variabele in de kernel.

Het basis systeem van FreeBSD bevat een basis set van applicaties voor het inzien van de actieve jails, en voor het uitvoeren van administratieve commando's in een jail. De jls(8) en jexec(8) commando's zijn onderdeel van het basis systeem en kunnen gebruikt worden voor het uitvoeren van de volgende simpele taken:

- Het printen van een lijst van actieve jails met het corresponderende jail ID (JID), IP adres, de hostnaam en het pad.
- Het koppelen met een actieve jail vanuit het host systeem, en voor het uitvoeren van administratieve taken in de jail zelf. Dit is bijzonder handig wanneer de `root` gebruiker een jail netjes wilt afsluiten. Het jexec(8) commando kan ook gebruikt worden om een shell te starten in een jail om daarmee administratieve taken uit te voeren; bijvoorbeeld met:

```
# jexec 1 tcsh
```

14.5.2. Administratieve applicaties op hoog niveau in de FreeBSD Ports Collection.

Tussen de vele software van derde partijen voor jail beheer, is één van de meest complete en bruikbare paketten: sysutils/jailutils. Dit is een set van kleine applicaties, die bijdragen aan jail(8) beheer. Kijk op de web pagina voor meer informatie.

14.6. Toepassing van jails

14.6.1. Dienst jails

Deze sectie is gebaseerd op een idee van Simon L. B. Nielsen <simon@FreeBSD.org> op http://simon.nitro.dk/service-jails.html, en een geupdate artikel door Ken Tom locals@gmail.com. Deze sectie illusteert hoe een FreeBSD systeem opgezet kan worden met een extra laag beveiliging door gebruik te maken van jail(8). Er wordt vanuit gegaan dat het betrokken systeem minstens RELENG_6_0 draait en dat de informatie eerder in dit hoofdstuk goed begrepen is.

Ontwerp

Één van de grootste problemen met jails is het beheer van het upgrade proces. Dit is meestal een probleem omdat elke jail vanaf het begin af aan moet worden opgebouwd wanneer er geupdate wordt. Meestal is dit voor een enkele jail geen probleem, omdat het update proces redelijk simpel is, maar het kan een vervelende tijdrovende klus zijn als er meerdere jails zijn.



Deze opstelling vereist uitgebreide kennis en ervaring van FreeBSD en zijn mogelijkheden. Als onderstaande stappen te lastig lijken te zijn, wordt aangeraden om een simpeler systeem te bekijken zoals sysutils/ezjail, welke een simpele manier geeft voor het beheren van FreeBSD jails en niet zo complex is als deze opstelling.

Het idee werd geopperd om zulke problemen zoveel als mogelijk te voorkomen door zoveel als mogelijk te delen tussen de verschillende jails op een zo veilig mogelijke manier - door gebruik te maken van alleen-lezen mount_nullfs(8) koppelingen, zodat het upgraden simpeler wordt en het inzetten van jails voor enkele diensten interessanter wordt. Daarnaast geeft het een simpele manier om nieuwe jails toe te voegen of te verwijderen en om deze te upgraden.



Voorbeelden binnen deze context zijn: een HTTP server, een DNS server, een SMTP server enzovoorts.

De doelen van de opstelling zoals beschreven in dit hoofdstuk zijn:

- Het creeëren van een simpele en makkelijk te begrijpen jail structuur. Dit impliceert dat er *niet* elke keer een volledige installworld gedraaid hoeft te worden voor elke jail.
- Het makkelijk maken van het aanmaken en verwijderen van jails.
- Het makkelijk maken van het updaten en upgraden van bestaande jails.
- Het mogelijk maken van het draaien van een eigen gemaakte FreeBSD tak.

- Paranoia zijn over beveiliging, zoveel mogelijk beperken, om de kans op inbraak zo klein mogelijk te maken.
- Het zoveel mogelijk besparen van ruimte en inodes.

Zoals reeds besproken is dit ontwerp sterk afhankelijk van het hebben van een "master-template", welke alleen-lezen (beter bekend als nullfs) gekoppeld is binnen elke jail, en een beschrijfbaar apparaat per jail. Een apparaat kan hierin zijn een aparte fysieke schijf, een partitie, of een door vnodes ondersteunde md(4) apparaat. In dit voorbeeld wordt gebruik gemaakt van lezen-schrijven nullfs koppelpunten.

Het gebruikte bestandssysteem wordt beschreven door de volgende lijst:

- Elke jail zal gekopeld worden onder de /home/j directory.
- /home/j/mroot is de template voor elke jail en tevens de alleen-lezen partitie voor elke jail.
- Voor elke jail zal een lege directory structuur gemaakt worden, welke valt onder de /home/j directory.
- Elke jail heeft een /s directory, welke gekoppeld zal worden aan het beschrijfbare gedeelte van het systeem.
- Elke jail zal zijn eigen beschrijfbaar systeem hebben welke gebaseerd is op /home/j/skel.
- Elke jail ruimte (het beschrijfbare gedeelte van de jail), wordt gecreeërd in de /home/js directory.



De voorbeelden gaan er vanuit dat de jails geplaatst worden in /home partitie. Dit kan uiteraard aangepast worden, maar dan moeten de voorbeelden hieronder ook worden aangepast naar de plek die gebruikt zal worden.

De template creeëren

Deze sectie leert welke stappen er genomen moeten worden om de master-template te maken. Deze zal het alleen-lezen gedeelte vormen van de jails.

Het is altijd een goed idee om ervoor te zorgen dat het FreeBSD systeem de laatst beschikbare -RELEASE versie draait. Zie het corresponderende hoofdstuk in het Handboek om te lezen hoe dit gedaan wordt. In het geval dat het de moeite niet is om te updaten, zal een buildworld nodig zijn voordat er verder gegaan kan worden. Daarnaast is het sysutils/cpdup pakket benodigd. Er wordt gebruik gemaakt van deportsnap(8) applicatie om de FreeBSD Ports Collectie te downloaden. Het handbook met het hoofdstuk over Portsnap, is een aanrader voor nieuwe gebruikers.

1. Als eerste moet er een directory structuur gecreeërd worden voor het alleen-lezen bestandssysteem, welke de FreeBSD binaries zal bevatten voor de jails. Daarna wordt het alleen-lezen bestandssysteem geïnstalleerd vanuit de FreeBSD broncode directory in de jail template:

```
# mkdir /home/j /home/j/mroot
# cd /usr/src
```

```
# make installworld DESTDIR=/home/j/mroot
```

2. Hierna moet de FreeBSD Ports Collectie worden voorbereid, evenals de FreeBSD broncode directory, wat voor mergemaster vereist is:

```
# cd /home/j/mroot
# mkdir usr/ports
# portsnap -p /home/j/mroot/usr/ports fetch extract
# cpdup /usr/src /home/j/mroot/usr/src
```

3. Nu moet er een "skelet" gecreeërd worden voor het bechrijfbare gedeelte van het systeem:

```
# mkdir /home/j/skel /home/j/skel/home /home/j/skel/usr-X11R6
/home/j/skel/distfiles
# mv etc /home/j/skel
# mv usr/local /home/j/skel/usr-local
# mv tmp /home/j/skel
# mv var /home/j/skel
# mv root /home/j/skel
```

4. De mergemaster applictie moet gebruikt worden om de ontbrekende configuratie bestanden te installeren. Erna moeten alle overtollige directories die gecreeërd zijn door mergemaster verwijderd worden:

```
# mergemaster -t /home/j/skel/var/tmp/temproot -D /home/j/skel -i
# cd /home/j/skel
# rm -R bin boot lib libexec mnt proc rescue sbin sys usr dev
```

5. Nu moet er een symbolische link gemaakt worden tussen het beschrijfbare bestandssysteem en het alleen-lezen bestandssysteem, zorg ervoor dat de links gemaakt worden in de juiste /s directory. Als hier echte directories worden gemaakt of de directories worden op de verkeerde plak aangemaakt zal dit resulteren in een mislukte installatie:

```
# cd /home/j/mroot
# mkdir s
# ln -s s/etc etc
# ln -s s/home home
# ln -s s/root root
# ln -s ../s/usr-local usr/local
# ln -s ../s/usr-X11R6 usr/X11R6
# ln -s ../../s/distfiles usr/ports/distfiles
# ln -s s/tmp tmp
# ln -s s/var var
```

6. Als laatste stap moet er een generieke /home/j/skel/etc/make.conf gemaakt worden met de

volgende inhoud:

```
WRKDIRPREFIX?=  /s/portbuild
```

Door het gebruik van `WRKDIRPREFIX` op deze manier, is het mogelijk om per jail FreeBSD ports te compileren. Onthoud dat de ports directory onderdeel is van het alleen-lezen bestandssysteem. Het eigen pad voor `WRKDIRPREFIX` maakt het mogelijk dat port builds gedaan worden op het beschrijfbare gedeelte van elke jail.

Jails creeëren

Nu we een complete FreeBSD template hebben, kunnen we de jails opzetten en configureren in /etc/rc.conf. Dit voorbeeld demonstreert het creeëren van drie jails: "NS", "MAIL" en "WWW".

1. Zet het volgende in /etc/fstab zodat de alleen-lezen template voor de jails en de beschrijfbare partititie beschikbaar zijn in de respectievelijke jails:

```
/home/j/mroot   /home/j/ns     nullfs  ro  0   0
/home/j/mroot   /home/j/mail   nullfs  ro  0   0
/home/j/mroot   /home/j/www    nullfs  ro  0   0
/home/js/ns     /home/j/ns/s   nullfs  rw  0   0
/home/js/mail   /home/j/mail/s nullfs  rw  0   0
/home/js/www    /home/j/www/s  nullfs  rw  0   0
```



Partities die gemarkeerd zijn met een 0 als "passnummer" worden niet gecontroleerd door fsck(8) tijdens het opstarten, en partities met een "dumpnummer" van 0 worden niet geback-upped door dump(8). Het is niet gewenst dat fsck de nullfs koppelingen controleert of dat dump een back-up maakt van de alleen-lezen nullfs koppelingen van de jails. Daarom worden ze gemarkeerd met "0 0" in de laatste twee kolommen van elke fstab regel hierboven.

2. Configureer de jails in /etc/rc.conf:

```
jail_enable="YES"
jail_set_hostname_allow="NO"
jail_list="ns mail www"
jail_ns_hostname="ns.example.org"
jail_ns_ip="192.168.3.17"
jail_ns_rootdir="/usr/home/j/ns"
jail_ns_devfs_enable="YES"
jail_mail_hostname="mail.example.org"
jail_mail_ip="192.168.3.18"
jail_mail_rootdir="/usr/home/j/mail"
jail_mail_devfs_enable="YES"
```

```
jail_www_hostname="www.example.org"
jail_www_ip="62.123.43.14"
jail_www_rootdir="/usr/home/j/www"
jail_www_devfs_enable="YES"
```



De reden dat de `jailnamerootdir` variabele verwijst naar de /usr/home directory in plaats van naar /home komt doordat het fysieke pad van de /home directory op een standaard FreeBSD installatie verwijst naar /usr/home. De `jailnamerootdir` variabele mag *niet* ingesteld worden op een symbolische link, omdat dan de jail weigert te starten. Gebruik het realpath(1) programma om te zien welke waarde ingesteld moet worden voor deze variabele. Zie de FreeBSD-SA-07:11.jail waarschuwing voor meer informatie.

3. Creeër de benodigde koppelpunten voor het alleen-lezen bestandssysteem van elke jail:

```
# mkdir /home/j/ns /home/j/mail /home/j/www
```

4. Installeer de beschrijfbare template in elke jail. Let op het gebruik van sysutils/cpdup, wat helpt om een goede kopie te maken in elke directory:

```
# mkdir /home/js
# cpdup /home/j/skel /home/js/ns
# cpdup /home/j/skel /home/js/mail
# cpdup /home/j/skel /home/js/www
```

5. In deze fase zijn de jails gebouwd en voorbereid om op te starten. Koppel eerst de benodigde bestandssystemen voor elke jail, en start ze vervolgens door gebruik te maken van het rc-bestand voor de jail:

```
# mount -a
# service jail start
```

De jails zouden nu gestart moeten zijn. Om te zien of ze correct gestart zijn, wordt het jls(8) programma gebruikt. Het resultaat hiervan ziet er ongeveer als volgend uit:

```
# jls
   JID  IP Address      Hostname                      Path
     3  192.168.3.17    ns.example.org                /home/j/ns
     2  192.168.3.18    mail.example.org              /home/j/mail
     1  62.123.43.14    www.example.org               /home/j/www
```

Op dit moment, zou het mogelijk moeten zijn om op elke jail aan te loggen, nieuwe gebruikers toe te

voegen en het configureren van daemons. De `JID` kolom geeft het identificatie nummer voor elke gestarte jail. Gebruik het volgende commando om administratieve commando's uit te voeren in de jail met het `JID` 3:

```
# jexec 3 tcsh
```

Upgraden

Naarmate de tijd verstrijkt komt de noodzaak om het systeem te updaten naar een nieuwere versie van FreeBSD, danwel vanwege een veiligheids waarschuwing danwel vanwege nieuwe mogelijkheden die geïmplementeerd zijn en nuttig zijn voor de jails. Het ontwerp van deze opzet levert een makkelijke manier voor het upgraden van jails. Daarnaast minimaliseert het de "down-time", omdat de jails alleen in de allerlaatste minuut uitgeschakeld worden. Het geeft ook de mogelijkheid om terug te keren naar een oudere versie, voor het geval er problemen ontstaan.

1. De eerste stap is het upgraden van het host systeem zelf, waarna een nieuwe alleen-lezen template gemaakt wordt in /home/j/mroot2.

   ```
   # mkdir /home/j/mroot2
   # cd /usr/src
   # make installworld DESTDIR=/home/j/mroot2
   # cd /home/j/mroot2
   # cpdup /usr/src usr/src
   # mkdir s
   ```

   Het `installworld` doel creeërt een aantal onnodige directories, welke verwijderd moeten worden:

   ```
   # chflags -R 0 var
   # rm -R etc var root usr/local tmp
   ```

2. Maak opnieuw de beschrijfbare symbolische linken voor het hoofd bestandssysteem:

   ```
   # ln -s s/etc etc
   # ln -s s/root root
   # ln -s s/home home
   # ln -s ../s/usr-local usr/local
   # ln -s ../s/usr-X11R6 usr/X11R6
   # ln -s s/tmp tmp
   # ln -s s/var var
   ```

3. Dit is het juiste moment om de jails te stoppen:

```
# service jail stop
```

4. Ontkoppel de originele bestandssystemen:

```
# umount /home/j/ns/s
# umount /home/j/ns
# umount /home/j/mail/s
# umount /home/j/mail
# umount /home/j/www/s
# umount /home/j/www
```



Het beschrijfbare gedeelte van de jail is gekoppeld aan het alleen-lezen gedeelte (/s) en moet derhalve eerst ontkoppeld worden.

5. Verplaatst het oude alleen-lezen systeem en vervang het door de nieuwe systeem. Het oude systeem dient als reservekopie voor het geval er iets misgaat. De naam moet het zelfde zijn als bij de installatie van het nieuwe systeem. Verplaats de FreeBSD Ports Collectie naar het nieuwe bestandssysteem om ruimte en inodes te besparen:

```
# cd /home/j
# mv mroot mroot.20060601
# mv mroot2 mroot
# mv mroot.20060601/usr/ports mroot/usr
```

6. Op dit moment is het alleen-lezen gedeelte klaar, de enig overgebleven taak is nu om alle bestandssystemen opnieuw te koppelen en om de jails weer op te starten:

```
# mount -a
# service jail start
```

Gebruik het jls(8) programma om te zien of de jails correct zijn opgestart. Vergeet niet om in elke jail het mergemaster programma te starten. Ook moeten de configuratie bestanden en de rc.d scripts geupdate worden.

Hoofdstuk 15. Verplichte Toegangscontrole (MAC)

15.1. Overzicht

In FreeBSD 5.X werden nieuwe beveiligingsuitbreidingen geïntroduceerd uit het TrustedBSD project, dat is gebaseerd op de POSIX®.1e draft. Twee van de meest significante nieuwe beveiligingsmechanismen zijn faciliteiten voor Toegangscontrolelijsten voor bestandssystemen (ACLs) en Verplichte Toegangscontrole (Mandatory Access Control of MAC). Met Verplichte Toegangscontrole kunnen nieuwe toegangscontrolemodules geladen worden, waarmee nieuw beveiligingsbeleid opgelegd kan worden. Een aantal daarvan bieden beveiliging aan hele kleine onderdelen van het systeem, waardoor een bepaalde dienst weerbaarder wordt. Andere bieden allesomvattende gelabelde beveiliging op alle vlakken en objecten. Het verplichte deel van de definitie komt van het feit dat het opleggen van de controle wordt gedaan door beheerders en het systeem en niet wordt overgelaten aan de nukken van gebruikers, zoals wel wordt gedaan met toegangscontrole naar goeddunken (discretionary access control of DAC, de standaardrechten voor bestanden en System V IPC rechten in FreeBSD).

In dit hoofdstuk wordt de nadruk gelegd op het Verplichte Toegangscontrole Raamwerk (MAC Framework) en een verzameling van te activeren beveiligingsbeleidsmodules waarmee verschillende soorten beveiligingsmechanismen wordt ingeschakeld.

Na het lezen van dit hoofdstuk weet u:

- Welke MAC beveiligingsbeleidsmodules op dit moment in FreeBSD beschikbaar zijn en welke mechanismen daarbij horen.
- Wat MAC beveiligingsbeleidsmodules implementeren en het verschil tussen gelabeld en niet-gelabeld beleid.
- Hoe een systeem efficiënt ingesteld kan worden om met het MAC-raamwerk te werken.
- Hoe het beleid van de verschillende beveiligingsbeleidsmodules die in het MAC-raamwerk zitten ingesteld kunnen worden.
- Hoe een veiligere omgeving gemaakt kan worden met het MAC-raamwerk en de getoonde voorbeelden;
- Hoe de MAC-instellingen getest kunnen worden om er zeker van te zijn dat het raamwerk juist is geïmplementeerd.

Aangeraden voorkennis:

- Begrip van UNIX® en FreeBSD basiskennis (UNIX® beginselen);
- Bekend zijn met de beginselen van het instellen en compileren van de kernel (De FreeBSD-kernel instellen);
- Enigszins bekend zijn met beveiliging en wat dat te maken heeft met FreeBSD (Beveiliging).



Het verkeerd gebruiken van de informatie die hierin staat kan leiden tot het niet

> langer toegang hebben tot een systeem, ergernis bij gebruikers, of het niet langer kunnen gebruiken van de mogelijkheden die X11 biedt. Nog belangrijker is dat niet alleen op MAC vertrouwd moet worden voor de beveiliging van een systeem. Het MAC-raamwerk vergroot alleen het bestaande beveiligingsbeleid; zonder goede beveiligingsprocedures en regelmatige beveiligingscontroles is een systeem nooit helemaal veilig.
>
> Het is ook van belang op te merken dat de voorbeelden in dit hoofdstuk alleen voorbeelden zijn. Het is niet aan te raden ze uit te rollen op een productiesysteem. Het implementeren van de verschillende beveiligingsbeleidsmodules dient goed overdacht en getest te worden. Iemand die niet helemaal begrijpt hoe alles werkt, komt er waarschijnlijk achter dat die het complete systeem van voor naar achter en weer terug doorloopt en vele bestanden en mappen opnieuw moet instellen.

15.1.1. Wat niet wordt behandeld

In dit hoofdstuk wordt een brede reeks beveiligingsonderwerpen met betrekking tot het MAC-raamwerk behandeld. De ontwikkeling van nieuwe MAC-beveiligingsbeleidsmodules wordt niet behandeld. Een aantal modules die bij het MAC-raamwerk zitten hebben specifieke eigenschappen voor het testen en ontwikkelen van nieuwe modules. Daaronder vallen mac_test(4), mac_stub(4) en mac_none(4). Meer informatie over deze beveiligingsbeleidsmodules en de mogelijkheden die ze bieden staan in de hulppagina's.

15.2. Sleuteltermen in dit hoofdstuk

Voordat dit hoofdstuk gelezen wordt, moeten er een aantal sleuteltermen toegelicht worden. Hiermee wordt hopelijk mogelijke verwarring en de abrupte introductie van nieuwe termen en informatie voorkomen.

- *compartiment*: een compartiment is een verzameling van programma's en gegevens die gepartitioneerd of gescheiden dient te worden en waartoe gebruikers expliciet toegang moeten krijgen op een systeem. Een compartiment staat ook voor een groep, zoals een werkgroep, afdeling, project, of onderwerp. Door gebruik te maken van compartimenten is het mogelijk om een "need-to-know" beveiligingsbeleid in te stellen.
- *hoogwatermarkering*: Een hoogwatermarkeringsbeleid is een beleid dat toestaat om beveiligingsniveaus te verhogen met het doel informatie dat op een hoger niveau aanwezig is te benaderen. In de meeste gevallen wordt het originele niveau hersteld nadat het proces voltooid is. Momenteel heeft het MAC-raamwerk van FreeBSD hier geen beleid voor, maar de definitie is voor de volledigheid opgenomen.
- *integriteit*: integriteit, als sleutelconcept, is het niveau van vertrouwen dat in gegevens gesteld kan worden. Als de integriteit van gegevens wordt vergroot, dan geldt dat ook voor het vertrouwen dat in die gegevens gesteld kan worden.
- *label*: een label is een beveiligingsattribuut dat toegepast kan worden op bestanden, mappen of andere onderdelen van een systeem. Het kan gezien worden als een vertrouwelijkheidsstempel: als er een label op een bestand is geplaatst, beschrijft dat de beveiligingseigenschappen voor dat specifieke bestand en is daarop alleen toegang voor bestanden, gebruikers, bronnen, enzovoort,

met gelijke beveiligingsinstellingen. De betekenis en interpretatie van labelwaarden hangt af van de beleidsinstellingen: hoewel sommige beleidseenheden een label beschouwen als representatie van de integriteit of het geheimhoudingsniveau van een object, kunnen andere beleidseenheden labels gebruiken om regels voor toegang in op te slaan.

- *niveau*: de verhoogde of verlaagde instelling van een beveiligingsattribuut. Met het stijgen van het niveau wordt ook aangenomen dat de veiligheid stijgt.
- *laagwatermarkering*: Een laagwatermarkeringsbeleid is een beleid dat toestaat om de beveiligingsniveaus te verlagen met het doel informatie te benaderen die minder veilig is. In de meeste gevallen wordt het originele beveiligingsniveau van de gebruiker hersteld nadat het proces voltooid is. De enige beveiligingsbeleidsmodule in FreeBSD die dit gebruikt is mac_lomac(4).
- *meervoudig label*: de eigenschap `multilabel` is een optie van het bestandssysteem die in enkelegebruikersmodus met tunefs(8), tijdens het opstarten via het bestand fstab(5) of tijdens het maken van een nieuw bestandssysteem ingesteld kan worden. Met deze optie wordt het voor een beheerder mogelijk om verschillende MAC-labels op verschillende objecten toe te passen. Deze optie is alleen van toepassing op beveiligingsbeleidsmodules die labels ondersteunen.
- *object*: een object of systeemobject is een entiteit waar informatie doorheen stroomt op aanwijzing van een *subject*. Hieronder vallen mappen, bestanden, velden, schermen, toetsenborden, geheugen, magnetische opslag, printers en alle andere denkbare apparaten waarmee gegevens kunnen worden vervoerd of kunnen worden opgeslagen. In de basis is een object een opslageenheid voor gegevens of een systeembron; toegang tot een *object* betekent in feite toegang tot de gegevens.
- *beleidseenheid*: een verzameling van regels die aangeven hoe doelstellingen bereikt moeten worden. In een *beleidseenheid* staat meestal beschreven hoe bepaalde eenheden behandeld dienen te worden. In dit hoofdstuk wordt de term *beleidseenheid* in deze context gezien als een *beveiligingsbeleidseenheid*, wat zoveel wil zeggen als een verzameling regels die bepaalt hoe gegevens en informatie stroomt en aangeeft wie toegang tot welke gegevens en informatie heeft.
- *gevoeligheid*: meestal gebruikt bij het bespreken van MLS. Een gevoeligheidsniveau is een term die gebruikt wordt om te beschrijven hoe belangrijk of geheim de gegevens horen te zijn. Met het stijgen van het gevoeligheidsniveau stijgt ook het belang van de geheimhouding of de vertrouwelijkheid van de gegevens.
- *enkelvoudig label*: een enkelvoudig label wordt gebruikt als een heel bestandssysteem gebruik maakt van één label om het toegangsbeleid over de gegevensstromen af te dwingen. Als dit voor een bestandssysteem is ingesteld, wat geldt als er geen gebruik gemaakt wordt van de optie `multilabel`, dan gehoorzamen alle bestanden aan dezelfde labelinstelling.
- *subject*: een subject is een gegeven actieve entiteit die het stromen van informatie tussen *objecten* veroorzaakt, bijvoorbeeld een gebruiker, gebruikersprocessor, systeemproces, enzovoort. Op FreeBSD is dit bijna altijd een thread die in een proces namens een gebruiker optreedt.

15.3. Uitleg over MAC

Met al deze nieuwe termen in gedachten, kan overdacht worden hoe het MAC-raamwerk de complete beveiliging van een systeem kan vergroten. De verschillende beveiligingsbeleidsmodules die het MAC-raamwerk biedt zouden gebruikt kunnen worden om het netwerk en bestandssystemen te beschermen, gebruikers toegang tot bepaalde poorten en sockets kunnen ontzeggen, en nog veel meer. Misschien kunnen de beleidsmodules het beste gebruikt worden door ze samen in te zetten, door meerdere beveiligingsbeleidsmodules te laden om te komen tot een omgeving waarin de beveiliging uit meerdere lagen is opgebouwd. In een omgeving waarin de beveiliging uit meerdere lagen is opgebouwd zijn meerdere beleidsmodules actief om de beveiliging in de hand te houden. Deze aanpak is anders dan een beleid om de beveiliging sec beter te maken, omdat daarmee in het algemeen elementen in een systeem beveiligd worden dat voor een specifiek doel wordt gebruikt. Het enige nadeel is het benodigde beheer in het geval van meervoudige bestandssysteemlabels, het instellen van toegang tot het netwerk per gebruiker, enzovoort.

De nadelen zijn wel minimaal als ze worden vergeleken met het immer durende effect van het raamwerk. Zo zorgt bijvoorbeeld de mogelijkheid om te kiezen welke beleidseenheden voor een specifiek gebruik nodig zijn voor het zo laag mogelijk houden van de beheerslast. Het terugdringen van ondersteuning voor onnodige beleidseenheden kan de beschikbaarheid van systemen verhogen en ook de keuzevrijheid vergroten. Voor een goede implementatie worden alle beveiligingseisen in beschouwing genomen en daarna worden de verschillende beveiligingsbeleidsmodules effectief door het raamwerk geïmplementeerd.

Een systeem dat gebruik maakt van de mogelijkheden van MAC dient dus tenminste de garantie te bieden dat een gebruiker niet de mogelijkheid heeft naar eigen inzicht beveiligingsattributen te wijzigen. Alle gebruikersprogramma's en scripts moeten werken binnen de beperkingen die de toegangsregels voorschrijven volgens de geselecteerde beveiligingsbeleidsmodules. Het voorgaande impliceert ook dat de volledige controle over de MAC-toegangsregels bij de systeembeheerder ligt.

Het is de taak van de systeembeheerder om zorgvuldig de juiste beveiligingsbeleidsmodules te kiezen. Voor sommige omgevingen kan het nodig zijn dat de toegang tot het netwerk wordt beperkt. In dat soort gevallen zijn de beleidsmodules mac_portacl(4), mac_ifoff(4) en zelfs mac_biba(4) goede startpunten. In andere gevallen kan de strikte vertrouwelijkheid van bestandssysteemobjecten van belang zijn. Dan zijn beleidsmodules zoals mac_bsdextended(4) en mac_mls(4) voor dit doel gemaakt.

Beslissingen over beleid zouden gemaakt kunnen worden op basis van het netwerkontwerp. Wellicht wordt alleen bepaalde gebruikers toegestaan gebruik te maken van de mogelijkheden van ssh(1) om toegang te krijgen tot het netwerk of Internet. In dat geval is de juiste beleidsmodule mac_portacl(4). Maar wat te doen voor bestandssystemen? Moet alle toegang tot bepaalde mappen worden afgesneden van andere gebruikersgroepen of specifieke gebruikers, of moeten de toegang voor gebruikers of programma's tot bepaalde bestanden worden ingesteld door bepaalde objecten als geheim te bestempelen?

In het geval van het bestandssysteem, kan ervoor gekozen worden om de toegang voor sommige objecten voor bepaalde gebruikers als geheim te bestempelen, maar voor andere niet. Bijvoorbeeld: een groot ontwikkelteam wordt opgedeeld in kleinere eenheden van individuen. Ontwikkelaars in project A horen geen toegang te hebben tot objecten die zijn geschreven door ontwikkelaars in

project B. Maar misschien moeten ze wel toegang hebben tot objecten die zijn geschreven door ontwikkelaars in project C. Dat is nogal wat. Door gebruik te maken van de verschillende beveiligingsbeleidsmodules in het MAC-raamwerk kunnen gebruikers in hun groepen worden opgedeeld en kan ze toegang gegeven worden tot de juiste locaties zonder dat er angst hoeft te zijn voor het lekken van informatie.

Zo heeft dus iedere beveiligingsbeleidsmodule een unieke wijze om om te gaan met de totale beveiliging van een systeem. Het kiezen van modules hoort gebaseerd te zijn op een zorgvuldig uitgedacht beveiligingsbeleid. In veel gevallen wordt het totale beveiligingsbeleid aangepast en opnieuw toegepast op het systeem. Een goed begrip van de verschillende beveiligingsbeleidsmodules die het MAC-raamwerk biedt helpt beheerders bij het kiezen van de juiste beleidseenheden voor hun situatie.

De standaard FreeBSD-kernel kent geen ondersteuning voor het MAC-raamwerk en daarom dient de volgende kerneloptie toegevoegd te worden voordat op basis van de voorbeelden of informatie uit dit hoofdstuk wijzigen worden gemaakt:

```
options     MAC
```

Hierna dient de kernel herbouwd en opnieuw geïnstalleerd te worden.



Hoewel in de verschillende hulppagina's voor MAC-beleidsmodules staat dat ze in de kernel gebouwd kunnen worden, is het mogelijk het systeem van het netwerk af te sluiten en meer. Het implementeren van MAC is net zoiets als het implementeren van een firewall en er moet opgepast worden dat een systeem niet totaal op slot gaat. Er dient rekening gehouden te worden met het teruggaan naar een vorige instelling en het op afstand implementeren van MAC dient bijzonder voorzichtig te gebeuren.

15.4. MAC-labels begrijpen

Een MAC-label is een beveiligingsattribuut dat toegepast kan worden op subjecten en objecten die door het systeem gaan.

Bij het instellen van een label moet de gebruiker in staat zijn om precies te begrijpen wat er gebeurt. De attributen die voor een object beschikbaar zijn hangen af van de geladen beleidsmodule en die interpreteren hun attributen op nogal verschillende manieren. Het resultaat kan resulteren in onverwacht en wellicht ongewenst gedrag van een systeem als het beleid door een gebrek aan begrip verkeerd is ingesteld.

Het beveiligingslabel op een object wordt gebruikt als onderdeel van een beveiligingstoegangscontrolebeslissing door een beleidseenheid. Voor sommige beleidseenheden bevat het label zelf alle informatie die nodig is voor het maken van een beslissing; in andere modellen kunnen de labels als onderdeel van een grotere verzameling verwerkt worden, enzovoort.

Zo staat bijvoorbeeld het instellen van het label `biba/low` op een bestand voor een label dat wordt

beheerd door de beveiligingsbeleidsmodule Biba, met een waarde van "low".

Een aantal beleidsmodules die in FreeBSD de mogelijkheid voor labelen ondersteunen, bieden drie specifieke voorgedefinieerde labels: low, high en equal. Hoewel ze in verschillende beleidsmodules op een andere manier toegangscontrole afdwingen, is er de garantie dat het label `low` de laagst mogelijke instelling is, het label `equal` het subject of object uitschakelt of ongemoeid laat en het label `high` de hoogst mogelijk instelling afdwingt die beschikbaar is in de beleidsmodules Biba en MLS.

Binnen een bestandssysteemomgeving met een enkelvoudig label kan er maar één label gebruikt worden op objecten. Hiermee wordt een verzameling van toegangsrechten op het hele systeem opgelegd en dat is voor veel omgevingen voldoende. Er zijn echter een aantal gevallen waarin het wenselijk is meervoudige labels in te stellen op subjecten of objecten in het bestandssysteem. In die gevallen kan de optie `multilabel` meegegeven worden aan tunefs(8).

In het geval van Biba en MLS kan er een numeriek label gezet worden om het precieze niveau van de hiërarchische controle aan te geven. Dit numerieke niveau wordt gebruikt om informatie in verschillende groepen te partitioneren of te sorteren voor het classificeren voor het geven van toegang voor een bepaalde groep of een groep van een hoger niveau.

In de meeste gevallen stelt een beheerder alleen maar een enkelvoudig label in dat door het hele bestandssysteem wordt gebruikt.

Wacht eens, dat klinkt net als DAC! MAC gaf de controle toch strikt aan de beheerder? Dat klopt nog steeds, `root` heeft nog steeds de controle in handen en is degene die het beleid instelt zodat gebruikers in de juiste categorie en/of toegangsniveaus worden geplaatst. Daarnaast kunnen veel beleidsmodules ook de gebruiker `root` beperkingen opleggen. Dan wordt de controle overgedragen aan een groep, maar kan `root` de instellingen op ieder gewenst moment intrekken of wijzigen. Dit is het hiërarchische of toegangsmodel dat wordt afgedekt door beleidseenheden zoals Biba en MLS.

15.4.1. Labelinstellingen

Vrijwel alle aspecten voor het instellen van labelbeleid worden uitgevoerd met basissysteemprogramma's. Die commando's bieden een eenvoudige interface voor object- of subjectinstellingen of de manipulatie en verificatie van de instellingen.

Alle instellingen kunnen gemaakt worden met de hulpprogramma's setfmac(8) en setpmac(8). Het commando `setfmac` wordt gebruikt om MAC labels op systeemobjecten in te stellen en `setpmac` voor het instellen van de labels op systeemsubjecten:

```
# setfmac biba/high test
```

Als het bovenstaande commando geen foutmeldingen heeft veroorzaakt, dan komt er een prompt terug. Deze commando's geven nooit uitvoer, tenzij er een fout is opgetreden; net als bij de commando's chmod(1) en chown(8). In sommige gevallen kan de foutmelding `Permission denied` zijn en deze treedt meestal op als het label wordt ingesteld of gewijzigd op een object dat is beperkt. De systeembeheerder kan de volgende commando's gebruiken om dit probleem te voorkomen:

```
# setfmac biba/high test
Permission denied
# setpmac biba/low setfmac biba/high test
# getfmac test
test: biba/high
```

Hierboven is te zien dat `setpmac` gebruikt kan worden om aan de instellingen van een beleidsmodules voorbij te gaan door een ander label toe te wijzen aan het aangeroepen proces. Het hulpprogramma `getpmac` wordt meestal toegepast op processen die al draaien, zoals sendmail: hoewel er een proces-ID nodig is in plaats van een commando, is de logica gelijk. Als gebruikers proberen een bestand te manipuleren waar ze geen toegang tot hebben, onderhevig aan de regels van de geladen beleidsmodules, dan wordt de foutmelding `Operation not permitted` weergegeven door de functie `mac_set_link`.

Labeltypen

Met de beleidsmodules mac_biba(4), mac_mls(4) en mac_lomac(4) is het mogelijk eenvoudige labels toe te wijzen. Die kunnen hoog, gelijk aan en laag zijn. Hieronder een beschrijving van wat die labels betekenen:

- Het label `low` is de laagst mogelijke labelinstelling die een object of subject kan hebben. Deze instelling op objecten of subjecten blokkeert hun toegang tot objecten of subjecten met de markering hoog.
- Het label `equal` hoort alleen ingesteld te worden op objecten die uitgesloten moeten worden van een beleidsinstelling.
- Het label `high` geeft een object of subject de hoogst mogelijke instelling.

Afhankelijke van iedere beleidsmodule heeft iedere instelling een ander informatiestroomdirectief tot gevolg. Het lezen van de hulppagina's die van toepassing zijn geeft inzicht in de precieze eigenschappen van de standaard labelinstellingen.

Gevorderde labelinstellingen

Dit zijn de labels met numerieke graden die gebruikt worden voor `vergelijking:afdeling+afdeling`.

```
biba/10:2+3+6(5:2+3-20:2+3+4+5+6)
```

Het bovenstaande kan dus geïnterpreteerd worden als:

"Biba-beleidslabel"/"Graad 10":"Afdelingen 2, 3 en 6": ("graad 5 ...")

In dit voorbeeld is de eerste graad de "effectieve graad" met de "effectieve afdelingen", de tweede graad is de lage graad en de laatste is de hoge graad. In de meeste instellingen worden deze instellingen niet gebruikt. Ze zijn inderdaad instellingen voor gevorderden.

Als ze worden toegepast op systeemobjecten, hebben ze alleen een huidige graad/afdeling in vergelijking met systeemsubjecten, omdat ze de reikwijdte van rechten in het systeem en op

netwerkinterfaces aangeven, waar ze gebruikt worden voor toegangscontrole.

De graad en afdelingen in een subject en object paar wordt gebruikt om een relatie te construeren die "dominantie" heet, waar een subject een object domineert, geen van beiden domineert, of beiden elkaar domineren. Het geval "beiden domineren" komt voor als de twee labels gelijk zijn. Vanwege de natuur van de informatiestroom van Biba, heeft een gebruiker rechten op een verzameling van afdelingen, "need to know", die overeen zouden kunnen komen met projecten, maar objecten hebben ook een verzameling van afdelingen. Gebruikers dienen wellicht hun rechten onder te verdelen met `su` of `setpmac` om toegang te krijgen tot objecten in een afdeling die geen verboden terrein voor ze zijn.

Gebruikers en labelinstellingen

Gebruikers moeten zelf labels hebben, zodat hun bestanden en processen juist kunnen samenwerken met het beveiligingsbeleid dat op een systeem is ingesteld. Dit wordt ingesteld via het bestand login.conf door gebruik te maken van aanmeldklassen. Iedere beleidsmodule die labels gebruikt implementeert ook de instelling van de gebruikersklasse.

Een voorbeeld dat iedere instelling uit de beleidsmodule bevat is hieronder te zien:

```
default:\
        :welcome=/etc/motd:\
        :setenv=MAIL=/var/mail/$,BLOCKSIZE=K:\
        :path=~/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin:\
        :manpath=/usr/shared/man /usr/local/man:\
        :nologin=/usr/sbin/nologin:\
        :cputime=1h30m:\
        :datasize=8M:\
        :vmemoryuse=100M:\
        :stacksize=2M:\
        :memorylocked=4M:\
        :memoryuse=8M:\
        :filesize=8M:\
        :coredumpsize=8M:\
        :openfiles=24:\
        :maxproc=32:\
        :priority=0:\
        :requirehome:\
        :passwordtime=91d:\
        :umask=022:\
        :ignoretime@:\
        :label=partition/13,mls/5,biba/10(5-15),lomac/10[2]:
```

De optie `label` wordt gebruikt om het standaardlabel voor aanmeldklasse in te stellen dat door MAC wordt afgedwongen. Het wordt gebruikers nooit toegestaan deze waarde te wijzigen, dus kan het gezien worden als niet optioneel vanuit het perspectief van de gebruiker. In de echte wereld besluit een beheerder echter nooit iedere beleidsmodule te activeren. Het wordt sterk aangeraden de rest van die hoofdstuk te lezen alvorens (een deel van) de bovenstaande instellingen te implementeren.



Gebruikers kunnen hun label wijzigen na het initiële aanmelden, maar dit is wel afhankelijk van de beperkingen van een beleidsinstelling. Het bovenstaande voorbeeld vertelt de beleidseenheid Biba dat de minimale integriteit van een proces 5 en het maximum 15, maar dat het effectieve label standaard 10 is. Het proces draait op niveau 10, totdat het proces het label wijzigt, misschien door een gebruiker die `setpmac` gebruikt, bij het aanmelden beperkt tot de door Biba ingestelde reeks.

In alle gevallen dient de database met aanmeldklassemogelijkheden opnieuw gebouwd te worden met `cap_mkdb` na het wijzigen van login.conf. Dit wordt ook in alle komende voorbeelden en beschrijvingen gedaan.

Het is belangrijk op te merken dat in veel gevallen sites te maken hebben met bijzonder grote aantallen gebruikers waardoor er een aantal verschillende aanmeldklassen nodig zijn. Het is dan nodig gedetailleerd te plannen omdat dit anders bijzonder complex wordt om te onderhouden.

Netwerkinterfaces en labelinstellingen

Labels kunnen ook ingesteld worden op netwerkinterfaces om te assisteren bij het controleren van het stromen van gegevens over het netwerk. In alle gevallen werken ze op dezelfde wijze als het beleid werkt ten aanzien van objecten. Gebruikers met bijvoorbeeld een hoge instelling in `biba` krijgen geen toegang tot interfaces met een laag label.

Het `maclabel` kan meegegeven worden aan `ifconfig` als het MAC-label op netwerkinterfaces wordt ingesteld:

```
# ifconfig bge0 maclabel biba/equal
```

In het bovenstaande voorbeeld wordt het MAC-label `biba/equal` ingesteld op de interface bge(4). Als er een instelling wordt gebruikt die gelijkvormig is aan `biba/high(low-high)`, dan moet het volledige label worden ingegeven, anders treedt er een fout op.

Iedere beleidsmodule die labels ondersteunt een instelling waarmee het MAC-label op netwerkinterfaces kan worden uitgeschakeld. Het label instellen op `equal` heeft hetzelfde effect. Deze instellingen zijn na te kijken in de uitvoer van `sysctl`, de hulppagina van het beleid en zelfs later in dit hoofdstuk.

15.4.2. Enkelvoudig label of meervoudig label?

Standaard gebruikt een systeem de optie `singlelabel`. Wat betekent dit voor een beheerder? Er zijn een aantal verschillen die allemaal hun eigen voor- en nadelen hebben voor de flexibiliteit in het beveiligingsmodel voor een systeem.

Bij gebruik van `singlelabel` kan er maar één label, bijvoorbeeld `biba/high`, gebruikt worden voor ieder subject of object. Hierdoor is er minder beheer nodig, maar de flexibiliteit voor beleid dat labels ondersteunt daalt erdoor. Veel beheerders willen de optie `multilabel` gebruiken in hun beveiligingsmodel.

De optie `multilabel` staat ieder subject of object toe om zijn eigen onafhankelijke MAC-label te hebben in plaats van de standaardoptie `singlelabel`, die maar één label toestaat op een hele partitie. De labelopties `multilabel` en `single` zijn alleen verplicht voor de beleidseenheden die de mogelijkheid bieden om te labelen, waaronder de beleidsmogelijkheden van Biba, Lomac, MLS en SEBSD.

In veel gevallen hoeft `multilabel` niet eens ingesteld te worden. Stel er is de volgende situatie en beveiligingsmodel:

- FreeBSD-webserver die gebruik maakt van het MAC-raamwerk en een mengeling van verschillende beleidseenheden.
- De webserver heeft maar één label nodig, `biba/high`, voor alles in het systeem. Hier is de optie `multilabel` voor het bestandssysteem niet nodig, omdat een enkelvoudig label altijd van toepassing is.
- Maar omdat de machine als webserver dienst gaat doen, dient de webserver te draaien als `biba/low` om administratiemogelijkheden te voorkomen. Later wordt beschreven hoe de beleidseenheid Biba werkt, dus als de voorgaande opmerking wat lastig te begrijpen is, lees dan verder en kom later nog een keer terug. De server zou een aparte partitie kunnen gebruiken waarop `biba/low` van toepassing kan zijn voor de meeste, zo niet alle, runtime-statussen. Er ontbreekt veel in dit voorbeeld, bijvoorbeeld de restricties op gegevens en (gebruikers)instellingen. Dit was slechts een snel voorbeeld om de hiervoor aangehaalde stelling te ondersteunen.

Als er een niet-labelende beleidseenheid wordt gebruikt, dan is de optie `multilabel` nooit verplicht. Hieronder vallen de beleidseenheden `seeotheruids`, `portacl` en `partition`.

Bij gebruik van `multilabel` voor een partitie en het neerzetten van een beveiligingsmodel gebaseerd op `multilabel` functionaliteit gaat de deur open voor hogere administratieve rompslomp, omdat alles in een bestandssysteem een label krijgt. Hieronder vallen mappen, bestanden en zelfs apparaatknooppunten.

Het volgende commando stelt `multilabel` in op de bestandssystemen om meerdere labels te kunnen krijgen. Dit kan alleen uitgevoerd worden in enkele gebruikersmodus:

```
# tunefs -l enable /
```

Dit is geen criterium voor het wisselbestandssysteem.



Sommige gebruikers hebben problemen ondervonden met het instellen van de vlag `multilabel` op de rootpartitie. Als dit het geval is, kijk dan naar Problemen oplossen met het MAC-raamwerk van dit hoofdstuk.

15.5. De beveiligingsconfiguratie plannen

Wanneer een nieuwe technologie wordt geïmplementeerd is een planningsfase altijd een goed idee. Tijdens de planningsfases zou een beheerder in het algemeen naar de "big picture" moeten kijken, en daarbij minstens het volgende in de gaten proberen te houden:

- De implementatiebenodigdheden;
- De implementatiedoelen;

Voor MAC-installaties houden deze in:

- Hoe de beschikbare informatie en bronnen die op het doelsysteem aanwezig zijn te classificeren.
- Voor wat voor soort informatie of bronnen de toegang te beperken samen met het type van de beperkingen die dienen te worden toegepast.
- Welke MAC-module(s) nodig zullen zijn om dit doel te bereiken.

Het is altijd mogelijk om de systeembronnen en de beveiligingsinstellingen te veranderen en te herconfigureren, het komt vaak erg ongelegen om het systeem te doorzoeken en bestaande bestanden en gebruikersaccounts te repareren. Plannen helpt om zeker te zijn van een probleemloze en efficiënte systeemimplementatie. Het is vaak vitaal en zeker in uw voordeel om een proefronde van het vertrouwde systeem, inclusief de configuratie, te draaien *vóórdat* een MAC-implementatie wordt gebruikt op productiesystemen. Het idee om een systeem met MAC gewoon los te laten is als het plannen van mislukkingen.

Verschillende omgevingen kunnen verschillende behoeften en benodigdheden nodig hebben. Het opzetten van een diepgaand en compleet beveiligingsprofiel zal de noodzaak van verandering verminderen wanneer het systeem in gebruik wordt genomen. Zodoende zullen de toekomstige secties de verschillende modules die beschikbaar zijn voor beheerders behandelen; hun gebruik en configuratie beschrijven; en in sommige gevallen inzicht bieden in welke situaties ze het beste tot hun recht komen. Een webserver bijvoorbeeld zou de beleiden mac_biba(4) en mac_bsdextended(4) in gebruik nemen. In andere gevallen kan voor een machine met erg weinig lokale gebruikers mac_partition(4) een goede keuze zijn.

15.6. Module-instellingen

Iedere module uit het MAC-raamwerk kan zoals zojuist aangegeven in de kernel worden gecompileerd of als runtime-kernelmodule geladen worden. De geadviseerde methode is de naam van een module toevoegen aan het bestand /boot/loader.conf zodat die wordt geladen tijdens de eerste fase van het starten van een systeem.

In de volgende onderdelen worden de verschillende MAC-modules en hun mogelijkheden beschreven. De implementatie in een specifieke omgeving wordt ook in dit hoofdstuk beschreven. Een aantal modules ondersteunt het gebruik van labelen, wat het beperken van toegang is door een label als "dit is toegestaan en dat niet" af te dwingen. Een labelinstellingenbestand kan bepalen hoe bestanden kunnen worden benaderd, hoe netwerkcommunicatie wordt uitgewisseld, en meer. In het vorige onderdeel is beschreven hoe de vlag `multilabel` ingesteld kon worden op bestandssystemen om per bestand of per partitie toegangscontrole in te schakelen.

Een instelling met een enkelvoudig label zou maar één label over een heel systeem afdwingen, daarom wordt de optie `tunefs multilabel` genoemd.

15.7. MAC-module seeotheruids

Modulenaam: mac_seeotheruids.ko

Kernelinstelling: `options MAC_SEEOTHERUIDS`

Opstartoptie: `mac_seeotheruids_load="YES"`

De module mac_seeotheruids(4) imiteert de `sysctl`-tunables `security.bsd.see_other_uids` en `security.bsd.see_other_gids` en breidt deze uit. Voor deze optie hoeven geen labels ingesteld te worden voor de instelling en hij werkt transparant met de andere modules.

Na het laden van de module kunnen de volgende `sysctl`-tunables gebruikt worden om de opties te beheren:

- `security.mac.seeotheruids.enabled` schakelt de opties van de module in en gebruikt de standaardinstellingen. Deze standaardinstellingen ontzeggen gebruikt de mogelijkheid processen en sockets te zien die eigendom zijn van andere gebruikers.
- `security.mac.seeotheruids.specificgid_enabled` staat toe dat een bepaalde groep niet onder dit beleid valt. Om bepaalde groepen van dit beleid uit te sluiten, kan de `sysctl`-tunable `security.mac.seeotheruids.specificgid=XXX` gebruikt worden. In het bovenstaande voorbeeld dient *XXX* vervangen te worden door het numerieke ID van een groep die uitgesloten moet worden van de beleidsinstelling.
- `security.mac.seeotheruids.primarygroup_enabled` wordt gebruikt om specifieke primaire groepen uit te sluiten van dit beleid. Als deze tunable wordt gebruikt, mag `security.mac.seeotheruids.specificgid_enabled` niet gebruikt worden.

15.8. MAC-module bsdextended

Modulenaam: mac_bsdextended.ko

Kernelinstelling: `options MAC_BSDEXTENDED`

Opstartoptie: `mac_bsdextended_load="YES"`

De module mac_bsdextended(4) dwingt de bestandssysteemfirewall af. Het beleid van deze module biedt een uitbreiding van het standaard rechtenmodel voor bestandssystemen, waardoor een beheerder een firewallachtige verzameling met regels kan maken om bestanden, programma's en mappen in de bestandssysteemhiërarchie te beschermen. Wanneer geprobeerd wordt om toegang tot een object in het bestandssysteem te krijgen, wordt de lijst met regels afgelopen totdat er òf een overeenkomstige regel is gevonden òf het einde van de lijst is bereikt. Dit gedrag kan veranderd worden door het gebruik van de sysctl(8)-parameter security.mac.bsdextended.firstmatch_enabled. Net zoals andere firewall-modules in FreeBSD kan een bestand dat regels voor toegangscontrole bevat tijdens het opstarten door het systeem worden aangemaakt en gelezen door een rc.conf(5) -variabele te gebruiken.

De lijst met regels kan ingevoerd worden met het hulpprogramma ugidfw(8), dat een syntaxis heeft die lijkt op die van ipfw(8). Meer hulpprogramma's kunnen geschreven worden met de functies in

de bibliotheek libugidfw(3).

Bij het werken met deze module dient bijzondere voorzichtigheid in acht te worden genomen. Verkeerd gebruik kan toegang tot bepaalde delen van het bestandssysteem blokkeren.

15.8.1. Voorbeelden

Nadat de module mac_bsdextended(4) is geladen, kan met het volgende commando de huidige regels getoond worden:

```
# ugidfw list
0 slots, 0 rules
```

Zoals verwacht zijn er geen regels ingesteld. Dit betekent dat alles nog steeds volledig toegankelijk is. Om een regel te maken die alle toegang voor alle gebruikers behalve `root` ontzegt:

```
# ugidfw add subject not uid root new object not uid root mode n
```

Dit is een slecht idee, omdat het voorkomt dat alle gebruikers ook maar het meest eenvoudige commando kunnen uitvoeren, zoals `ls`. Een betere lijst met regels zou kunnen zijn:

```
# ugidfw set 2 subject uid gebruiker1 object uid gebruiker2 mode n
# ugidfw set 3 subject uid gebruiker1 object gid gebruiker2 mode n
```

Hiermee wordt alle toegang, inclusief het tonen van mapinhoud, tot de thuismap van `gebruiker2` ontzegd voor de gebruikersnaam `gebruiker1`.

In plaats van `gebruiker1`, zou `not uid gebruiker2` kunnen worden opgegeven. Hierdoor worden dezelfde restricties als hierboven actief voor alle gebruikers in plaats van voor slechts één gebruiker.



De gebruiker `root` blijft onaangetast door deze wijzigingen.

Met deze informatie zou een basisbegrip moeten zijn ontstaan over hoe de module mac_bsdextended(4) gebruikt kan worden om een bestandssysteem te beschermen. Meer informatie staat in de hulppagina's van mac_bsdextended(4) en ugidfw(8).

15.9. MAC-module ifoff

Modulenaam: mac_ifoff.ko

Kernelinstelling: `options MAC_IFOFF`

Opstartoptie: `mac_ifoff_load="YES"`

De module mac_ifoff(4) bestaat alleen om netwerkinterfaces tijdens het draaien uit te schakelen en

om te verhinderen dat netwerkinterfaces tijdens het initiële opstarten worden geactiveerd. Er hoeven geen labels ingesteld te worden, noch is deze module afhankelijk van andere MAC-modules.

Het meeste beheer wordt gedaan met de `sysctl`-tunables die hieronder zijn vermeld.

- `security.mac.ifoff.lo_enabled` schakelt alle verkeer op het teruglusinterface (lo(4)) in of uit.
- `security.mac.ifoff.bpfrecv_enabled` schakelt alle verkeer op het Berkeley Packet Filterinterface (bpf(4)) in of uit.
- `security.mac.ifoff.other_enabled` schakelt alle verkeer op alle andere interfaces in of uit.

mac_ifoff(4) wordt het meest gebruikt om netwerken te monitoren in een omgeving waar netwerkverkeer niet toegestaan zou moeten zijn tijdens het opstarten. Een ander voorgesteld gebruik zou het schrijven van een script zijn dat security/aide gebruikt om automatisch netwerkverkeer te blokkeren wanneer het nieuwe of veranderde bestanden in beschermde mappen vindt.

15.10. MAC-module portacl

Modulenaam: mac_portacl.ko

Kernelinstelling: `MAC_PORTACL`

Opstartoptie: `mac_portacl_load="YES"`

De module mac_portacl(4) wordt gebruikt om het binden aan lokale TCP- en UDP-poorten te begrenzen door een waaier aan `sysctl`-variabelen te gebruiken. In essentie maakt mac_portacl(4) het mogelijk om niet-`root`-gebruikers in staat te stellen om aan gespecificeerde geprivilegieerde poorten te binden, dus poorten lager dan 1024.

Eenmaal geladen zal deze module het MAC-beleid op alle sockets aanzetten. De volgende tunables zijn beschikbaar:

- `security.mac.portacl.enabled` schakelt het beleid volledig in of uit.
- `security.mac.portacl.port_high` stelt het hoogste poortnummer in waarvoor mac_portacl(4) bescherming biedt.
- `security.mac.portacl.suser_exempt` sluit de gebruiker `root` uit van dit beleid wanneer het op een waarde anders dan nul wordt ingesteld.
- `security.mac.portacl.rules` specificeert het eigenlijke beleid van mac_portacl; zie onder.

Het eigenlijke beleid van `mac_portacl`, zoals gespecificeerd in de sysctl `security.mac.portacl.rules`, is een tekststring van de vorm: `regel[,regel,⋯]` met zoveel regels als nodig. Elke regel heeft de vorm: `idtype:id:protocol:poort`. De parameter idtype kan `uid` of `gid` zijn en wordt gebruikt om de parameter id als respectievelijk een gebruikers-id of groeps-id te interpreteren. De parameter protocol wordt gebruikt om te bepalen of de regel op TCP of UDP moet worden toegepast door de parameter op `tcp` of `udp` in te stellen. De laatste parameter poort is het poortnummer waaraan de gespecificeerde gebruiker of groep zich mag binden.



Aangezien de regelverzameling direct door de kernel wordt geïnterpreteerd

> kunnen alleen numerieke waarden voor de parameters voor de gebruikers-ID, groeps-ID, en de poort gebruikt worden. Voor gebruikers, groepen, en poortdiensten kunnen dus geen namen gebruikt worden.

Standaard kunnen op UNIX®-achtige systemen poorten lager dan 1024 alleen aan geprivilegieerde processen gebonden worden, dus diegenen die als `root` draaien. Om mac_portacl(4) toe te laten staan om ongeprivilegieerde processen aan poorten lager dan 1024 te laten binden moet deze standaard UNIX®-beperking uitgezet worden. Dit kan bereikt worden door de sysctl(8)-variabelen `net.inet.ip.portange.reservedlow` en `net.inet.ip.portrange.reservedhigh` op nul te zetten.

Zie de onderstaande voorbeelden of bekijk de handleidingpagina voor mac_portacl(4) voor meer informatie.

15.10.1. Voorbeelden

De volgende voorbeelden zouden de bovenstaande discussie wat moeten toelichten:

```
# sysctl security.mac.portacl.port_high=1023
# sysctl net.inet.ip.portrange.reservedlow=0 net.inet.ip.portrange.reservedhigh=0
```

Eerst wordt mac_portacl(4) ingesteld om de standaard geprivilegieerde poorten te dekken en worden de normale bindbeperkingen van UNIX® uitgeschakeld.

```
# sysctl security.mac.portacl.suser_exempt=1
```

De gebruiker `root` zou niet beperkt moeten worden door dit beleid, stel `security.mac.portacl.suser_exempt` dus in op een waarde anders dan nul. De module mac_portacl(4) is nu ingesteld om zich op de zelfde manier te gedragen als UNIX®-achtige systemen zich standaard gedragen.

```
# sysctl security.mac.portacl.rules=uid:80:tcp:80
```

Sta de gebruiker met UID 80 (normaliter de gebruiker `www`) toe om zich aan poort 80 te binden. Dit kan gebruikt worden om de gebruiker `www` toe te staan een webserver te draaien zonder ooit `root` -rechten te hebben.

```
# sysctl security.mac.portacl.rules=uid:1001:tcp:110,uid:1001:tcp:995
```

Sta de gebruiker met UID 1001 om zich aan de TCP-poorten 110 ("pop3") en 995 ("pop3s") te binden. Dit staat deze gebruiker toe om een server te starten die verbindingen accepteert op poorten 110 en 995.

15.11. MAC-module partition

Modulenaam: mac_partition.ko

Kernelinstelling: `options MAC_PARTITION`

Opstartoptie: `mac_partition_load="YES"`

Het beleid mac_partition(4) plaatst processen in specifieke "partities" gebaseerd op hun MAC-label. Zie dit als een speciaal soort jail(8), hoewel dit nauwelijks een waardige vergelijking is.

Dit is één module die aan het bestand loader.conf(5) dient te worden toegevoegd zodat het het beleid tijdens het opstartproces laadt en aanzet.

De meeste configuratie van dit beleid wordt gedaan met het gereedschap setpmac(8), wat hieronder zal worden uitgelegd. De volgende `sysctl`-tunable is beschikbaar voor dit beleid:

- `security.mac.partition.enabled` zet het afdwingen van MAC-procespartities aan.

Wanneer dit beleid aanstaat, mogen gebruikers alleen hun eigen processen zien, en elke andere in hun partitie, maar mogen niet met gereedschappen buiten deze partitie werken. Bijvoorbeeld, een gebruiker in de klasse `insecure` heeft geen toegang tot het commando `top` noch tot vele andere commando's die een proces moeten draaien.

Gebruik het gereedschap `setpmac` om gereedschappen in te stellen of ze in een partitielabel te plaatsen:

```
# setpmac partition/13 top
```

Dit zal het commando `top` toevoegen aan het label dat voor gebruikers in de klasse `insecure` gebruikt wordt. Merk op dat alle processen gestart door gebruikers in de klasse `insecure` in het label `partition/13` zullen blijven.

15.11.1. Voorbeelden

Het volgende commando laat de partitielabel en de proceslijst zien:

```
# ps Zax
```

Het volgende commando staat toe om het procespartitielabel van een andere gebruiker en de momenteel draaiende processen van die gebruiker te zien:

```
# ps -ZU trhodes
```



Gebruikers kunnen processen in het label van `root` zien tenzij het beleid mac_seeotheruids(4) is geladen.

Een echte vakmansimplementatie zou alle diensten in /etc/rc.conf uitzetten en deze door een script met de juiste labeling laten starten.



De volgende beleiden ondersteunen integerinstellingen in plaats van de drie standaardlabels die aangeboden worden. Deze opties, inclusief hun beperkingen, worden verder uitgelegd in de handleidingpagina's van de modules.

15.12. MAC-module Multi-Level Security

Modulenaam: mac_mls.ko

Kernelinstelling: `options MAC_MLS`

Opstartoptie: `mac_mls_load="YES"`

Het beleid mac_mls(4) beheert toegang tussen subjecten en objecten in het systeem door een strikt beleid voor informatiestromen af te dwingen.

In MLS-omgevingen wordt een "toestemming"-niveau ingesteld in het label van elk subject of object, samen met compartimenten. Aangezien deze toestemmings- of zinnigheidsniveaus getallen groter dan zesduizend kunnen bereiken; zou het voor elke systeembeheerder een afschrikwekkende taak zijn om elk subject of object grondig te configureren. Gelukkig worden er al drie "kant-en-klare" bij dit beleid geleverd.

Deze labels zijn `mls/low`, `mls/equal` en `mls/high`. Aangezien deze labels uitgebreid in de handleidingpagina worden beschreven, worden ze hier slechts kort beschreven:

- Het label `mls/low` bevat een lage configuratie welke het toestaat om door alle andere objecten te worden gedomineerd. Alles dat met `mls/low` is gelabeld heeft een laag toestemmingsniveau en heeft geen toegang tot informatie van een hoger niveau. Ook voorkomt dit label dat objecten van een hoger toestemmingsniveau informatie naar hen schrijven of aan hen doorgeven.
- Het label `mls/equal` dient geplaatst te worden op objecten die geacht te zijn uitgesloten van het beleid.
- Het label `mls/high` is het hoogst mogelijke toestemmingsniveau. Objecten waaraan dit label is toegekend zijn dominant over alle andere objecten in het systeem; ze mogen echter geen informatie lekken naar objecten van een lagere klasse.

MLS biedt:

- Een hiërarchisch beveiligingsniveau met een verzameling niet-hiërarchische categoriën;
- Vaste regels: niet naar boven lezen, niet naar beneden schrijven (een subject kan leestoegang hebben naar objecten op zijn eigen niveau of daaronder, maar niet daarboven. Evenzo kan een subject schrijftoegang hebben naar objecten op zijn eigen niveau of daarboven maar niet daaronder.);
- Geheimhouding (voorkomt ongeschikte openbaarmaking van gegevens);
- Een basis voor het ontwerp van systemen die gelijktijdig gegevens op verschillende gevoeligheidsniveaus behandelen (zonder informatie tussen geheim en vertrouwelijk te

lekken).

De volgende `sysctl`-tunables zijn beschikbaar voor de configuratie van speciale diensten en interfaces:

- `security.mac.mls.enabled` wordt gebruikt om het MLS-beleid in en uit te schakelen.
- `security.mac.mls.ptys_equal` labelt alle pty(4)-apparaten als `mls/equal` wanneer ze worden aangemaakt.
- `security.mac.mls.revocation_enabled` wordt gebruikt om toegang tot objecten in te trekken nadat hun label in die van een lagere graad verandert.
- `security.mac.mls.max_compartments` wordt gebruikt om het maximaal aantal compartimentniveaus met objecten in te stellen; in feite het maximale compartimentnummer dat op een systeem is toegestaan.

Het commando setfmac(8) kan gebruikt worden om de MLS-labels te manipuleren. Gebruik het volgende commando om een label aan een object toe te kennen:

```
# setfmac mls/5 test
```

Gebruik het volgende commando om het MLS-label voor het bestand test te verkrijgen:

```
# getfmac test
```

Dit is een samenvatting van de mogelijkheden van het beleid MLS. Een andere manier is om een meesterbeleidsbestand in /etc aan te maken dat de MLS-informatie bevat en om dat bestand aan het commando `setfmac` te geven. Deze methode wordt uitgelegd nadat alle beleiden zijn behandeld.

15.12.1. Verplichte Gevoeligheid plannen

Met de beleidsmodule voor meerlaagse beveiliging plant een beheerder het beheren van gevoelige informatiestromen. Standaard zet het systeem met zijn natuur van lezen naar boven blokkeren en schrijven naar beneden blokkeren alles in een lage toestand. Alles is beschikbaar en een beheerder verandert dit langzaam tijdens de configuratiefase; waarbij de vertrouwelijkheid van de informatie toeneemt.

Buiten de bovengenoemde drie basisopties voor labels, kan een beheerder gebruikers en groepen indelen als nodig om de informatiestroom tussen hun te blokkeren. Het is misschien gemakkelijker om naar de informatie te kijken in toestemmingsniveaus waarvoor bekende woorden bestaan, zoals `Vertrouwelijk`, `Geheim` en `Strikt Geheim`. Sommige beheerders zullen verschillende groepen aanmaken gebaseerd op verschillende projecten. Ongeacht de classificatiemethode moet er een goed overwogen plan bestaan voordat zo'n berperkend beleid wordt geïmplementeerd.

Wat voorbeeldsituaties voor deze beveiligingsbeleidsmodule kunnen een e-commerce webserver, een bestandsserver die kritieke bedrijfsinformatie, en omgevingen van financiële instellingen zijn. De meest onwaarschijnlijke plaats zou een persoonlijk werkstation met slechts twee of drie gebruikers zijn.

15.13. MAC-module Biba

Modulenaam: mac_biba.ko

Kernelinstelling: `options MAC_BIBA`

Opstartoptie: `mac_biba_load="YES"`

De module mac_biba(4) laadt het beleid MAC Biba. Dit beleid werkt vaak zoals dat van MLS behalve dat de regels voor de informatiestroom lichtelijk zijn omgedraaid. Dit is gezegd om de neerwaartse stroom van gevoelige informatie te voorkomen terwijl het beleid MLS de opwaartse stroom van gevoelige informatie voorkomt; veel van deze sectie is dus op beide beleiden toepasbaar.

In Biba-omgevingen wordt een "integriteits"-label op elk subject of object ingesteld. Deze labels bestaan uit hiërarchische graden, en niet-hiërarchische componenten. Een graad van een object of subject stijgt samen met de integriteit.

Ondersteunde labels zijn `biba/low`, `biba/equal`, en `biba/high`; zoals hieronder uitgelegd:

- Het label `biba/low` wordt gezien als de laagste integriteit die een object of subject kan hebben. Dit instellen op objecten of subjecten zal hun schrijftoegang tot objecten of subjecten die als hoog zijn gemarkeerd blokkeren. Ze hebben echter nog steeds leestoegang.
- Het label `biba/equal` dient alleen geplaatst te worden op objecten die geacht te zijn uitgesloten van het beleid.
- Het label `biba/high` staat schrijven naar objecten met een lager label toe maar sluit het lezen van dat object uit. Het wordt aangeraden om dit label te plaatsen op objecten die de integriteit van het gehele systeem beïnvloeden.

Biba biedt:

- Hiërarchische integriteitsniveaus met een verzameling niet-hiërarchische integriteitscategoriën;
- Vaste regels: niet naar boven schrijven, niet naar beneden lezen (tegenovergestelde van MLS). Een subject kan schrijftoegang hebben naar objecten op hetzelfde niveau of daaronder, maar niet daarboven. Evenzo kan een subject leestoegang naar objecten op hetzelfde niveau of daarboven hebben, maar niet daaronder;
- Integriteit (voorkomt oneigenlijk wijzigen van gegevens);
- Integriteitsniveaus (in plaats van de gevoeligheidsniveaus van MLS)

De volgende `sysctl`-tunables kunnen gebruikt worden om het Biba-beleid te manipuleren.

- `security.mac.biba.enabled` kan gebruikt worden om het afdwingen van het Biba-beleid op de doelmachine aan en uit te zetten.
- `security.mac.biba.ptys_equal` kan gebruikt worden om het Biba-beleid op pty(4)-apparaten uit te zetten.
- `security.mac.biba.revocation_enabled` dwingt het herroepen van toegang tot objecten af als het label is veranderd om het subject te domineren.

Gebruik de commando's `setfmac` en `getfmac` om de instellingen van het Biba-beleid op systeemobjecten te benaderen:

```
# setfmac biba/low test
# getfmac test
test: biba/low
```

15.13.1. Verplichte Integriteit plannen

Integriteit, anders dan gevoeligheid, garandeert dat de informatie nooit door onvertrouwde gebruikers zal worden gemanipuleerd. Dit geldt ook voor informatie die tussen subjecten, objecten, of beiden wordt doorgegeven. Het verzekert dat gebruikers alleen de informatie kunnen wijzigen en in sommige gevallen zelfs benaderen die ze expliciet nodig hebben.

De beveiligingsbeleidsmodule mac_biba(4) staat een beheerder in staat om te bepalen welke bestanden en programma's een gebruiker of gebruikers mogen zien en draaien terwijl het verzekert dat de programma's en bestanden vrij zijn van dreigingen en vertrouwt zijn door het systeem voor die gebruiker of groep van gebruikers.

Tijdens de initiële planningsfase moet een beheerder bereid zijn om gebruikers in gradaties, niveaus, en gebieden in te delen. Gebruikers zal toegang tot niet alleen gegevens maar ook tot programma's en hulpmiddelen ontzegt worden zowel voordat en nadat ze beginnen. Het systeem zal standaard een hoog label instellen nadat deze beleidsmodule is ingeschakeld, en het is aan de beheerder om de verschillende gradaties en niveaus voor gebruikers in te stellen. In plaats van toestemmingsniveaus zoals boven beschreven te gebruiken, kan een goede planningsmethode onderwerpen bevatten. Bijvoorbeeld, geef alleen ontwikkelaars veranderingstoegang tot het broncoderepository, de broncodecompiler, en andere ontwikkelgereedschappen. Andere gebruikers zouden in andere groepen zoals testers, ontwerpers, of gewone gebruikers worden ingedeeld en zouden alleen leestoegang hebben.

Met zijn natuurlijke beveiligingsbeheer kan een subject van lagere integriteit niet schijven naar een subject van hogere integriteit; een subject van hogere integriteit kan geen subject van lagere integriteit observeren of lezen. Een label op de laagst mogelijke graad instellen kan het ontoegankelijk voor subjecten maken. Sommige succesvolle omgevingen voor deze beveiligingsbeheermodule zijn een beperkte webserver, een ontwikkel- en testmachine, en broncoderepositories. Minder nuttige implementaties zouden een persoonlijk werkstation, een machine gebruikt als router, of een netwerkfirewall zijn.

15.14. MAC-module LOMAC

Modulenaam: mac_lomac.ko

Kernelinstelling: `options MAC_LOMAC`

Opstartoptie: `mac_lomac_load="YES"`

In tegenstelling tot het beleid MAC Biba, staat het beleid mac_lomac(4) toegang tot objecten van lagere integriteit slechts toe nadat het integriteitsniveau is verlaagt om de integriteitsregels niet te

verstoren.

De MAC-versie van het laagwatermarkeringsintegreitsbeleid, niet te verwarren met de oudere implementatie van lomac(4), werkt bijna hetzelfde als Biba maar met de uitzondering dat er drijvende labels worden gebruikt om subjectdegradatie via een hulpcompartiment met graden te ondersteunen. Dit tweede compartiment heeft de vorm `[hulpgraad]`. Wanneer een lomac-beleid met een hulpgraad wordt toegekend, dient het er ongeveer uit te zien als: `lomac/10[2]` waar het getal twee (2) de hulpgraad is.

Het beleid MAC LOMAC berust op het overal labelen van alle systeemobjecten met integriteitslabels, waardoor subjecten wordt toegestaan om te lezen van objecten van lage integriteit en om daarna het label op subject te degraderen om toekomstig schrijven naar objecten van hoge integriteit te voorkomen. Dit is de hierboven besproken optie `[hulpgraad]`, dus biedt het beleid grotere compatibiliteit en vereist het minder initiële configuratie dan Biba.

15.14.1. Voorbeelden

Net zoals bij de beleiden Biba en MLS kunnen de commando's `setfmac` en `setpmac` gebruikt worden om labels op systeemobjecten te plaatsen:

```
# setfmac /usr/home/trhodes lomac/high[low]
# getfmac /usr/home/trhodes lomac/high[low]
```

Merk op dat de hulpgraad hier `low` is, dit is een mogelijkheid die alleen door het beleid MAC LOMAC wordt geboden.

15.15. Nagios in een MAC-jail

De volgende demonstratie zal een veilige omgeving implementeren door verschillende MAC-modules te gebruiken met juist ingestelde beleiden. Dit is slechts een test en dient niet gezien te worden als het volledige antwoord op de beveiligingszorgen van iedereen. Gewoon een beleid implementeren en het verder negeren werkt nooit en kan rampzalig zijn in een productieomgeving.

Voordat met dit proces wordt begonnen, moet de optie `multilabel` zijn geactiveerd op elk bestandssysteem zoals vermeld aan het begin van dit hoofdstuk. Nalatigheid zal in fouten resulteren. Zorg er ook voor dat de ports net-mgmt/nagios-plugins, net-mgmt/nagios, en www/apache22 allemaal geïnstalleerde en geconfigureerd zijn en correct werken.

15.15.1. Gebruikersklasse `insecure` maken

Begin de procedure door de volgende gebruikersklasse toe te voegen aan het bestand /etc/login.conf:

```
insecure:\
        :welcome=/etc/motd:\
        :setenv=MAIL=/var/mail/$,BLOCKSIZE=K:\
```

```
        :path=~/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin
        :manpath=/usr/shared/man /usr/local/man:\
        :nologin=/usr/sbin/nologin:\
        :cputime=1h30m:\
        :datasize=8M:\
        :vmemoryuse=100M:\
        :stacksize=2M:\
        :memorylocked=4M:\
        :memoryuse=8M:\
        :filesize=8M:\
        :coredumpsize=8M:\
        :openfiles=24:\
        :maxproc=32:\
        :priority=0:\
        :requirehome:\
        :passwordtime=91d:\
        :umask=022:\
        :ignoretime@:\
        :label=biba/10(10-10):
```

Voeg de volgende regel toe aan de standaard gebruikersklasse:

```
:label=biba/high:
```

Wanneer dit voltooid is, moet het volgende commando gedraaid worden om de database te herbouwen:

```
# cap_mkdb /etc/login.conf
```

15.15.2. Opstartinstellingen

Start nog niet opnieuw op, voeg alleen de volgende regels toe aan /boot/loader.conf zodat de benodigde modules worden geladen tijdens systeeminitialisatie:

```
mac_biba_load="YES"
mac_seeotheruids_load="YES"
```

15.15.3. Gebruikers instellen

Stel de gebruiker `root` in op de standaardklasse met:

```
# pw usermod root -L default
```

Alle gebruikersaccounts die geen `root` of systeemgebruikers zijn hebben nu een aanmeldklasse nodig. De aanmeldklasse is nodig om te voorkomen dat gebruikers geen toegang hebben tot gewone

commando's als vi(1). Het volgende sh-script zou moeten werken:

```
# for x in `awk -F: '($3 >= 1001) && ($3 != 65534) { print $1 }' \
    /etc/passwd`; do pw usermod $x -L default; done;
```

Laat de gebruikers nagios en www in de klasse insecure vallen:

```
# pw usermod nagios -L insecure
```

```
# pw usermod www -L insecure
```

15.15.4. Het contextbestand aanmaken

Nu dient een contextbestand aangemaakt te worden; het volgende voorbeeld dient geplaatst te worden in /etc/policy.contexts.

```
# Dit is het standaard-BIBA-beleid voor dit systeem.
# Systeem:
/var/run                        biba/equal
/var/run/*                      biba/equal

/dev                            biba/equal
/dev/*                          biba/equal

/var                            biba/equal
/var/spool                      biba/equal
/var/spool/*                    biba/equal

/var/log                        biba/equal
/var/log/*                      biba/equal

/tmp                            biba/equal
/tmp/*                          biba/equal
/var/tmp                        biba/equal
/var/tmp/*                      biba/equal

/var/spool/mqueue               biba/equal
/var/spool/clientmqueue         biba/equal

#Voor Nagios:
/usr/local/etc/nagios
/usr/local/etc/nagios/*         biba/10
/var/spool/nagios               biba/10
/var/spool/nagios/*             biba/10

#Voor Apache:
```

```
/usr/local/etc/apache           biba/10
/usr/local/etc/apache/*         biba/10
```

Dit beleid zal beveiliging afdwingen door beperkingen aan de informatiestroom te stellen. In deze specifieke configuratie mogen gebruikers, inclusief `root`, nooit toegang hebben tot Nagios. Instellingenbestanden en processen die deel zijn van Nagios zullen geheel in zichzelf of in een jail zitten.

Dit bestand kan nu in ons systeem worden gelezen door ons systeem door het volgende commando uit te voeren:

```
# setfsmac -ef /etc/policy.contexts /
# setfsmac -ef /etc/policy.contexts /
```



De bovenstaande indeling van het bestandssysteem kan afhankelijk van de omgeving verschillen; het moet echter op elk bestandssysteem gedraaid worden.

Het bestand /etc/mac.conf dient als volgt in de hoofdsectie gewijzigd te worden:

```
default_labels file ?biba
default_labels ifnet ?biba
default_labels process ?biba
default_labels socket ?biba
```

15.15.5. Het netwerk activeren

Voeg de volgende regel toe aan /boot/loader.conf:

```
security.mac.biba.trust_all_interfaces=1
```

En voeg het volgende toe aan de instellingen van de netwerkkaart opgeslagen in rc.conf. Als de primaire Internetconfiguratie via DHCP wordt gedaan, kan het nodig zijn om dit handmatig te configureren telkens nadat het systeem is opgestart:

```
maclabel biba/equal
```

15.15.6. De configuratie testen

Controleer dat de webserver en Nagios niet tijdens de systeeminitialisatie worden gestart, en start opnieuw op. Controleer dat de gebruiker `root` geen enkel bestand in de instellingenmap van Nagios kan benaderen. Als `root` het commando ls(1) op /var/spool/nagios kan uitvoeren, is er iets verkeerd. Anders zou er een fout "Permission denied" teruggegeven moeten worden.

Als alles er goed uitziet, kunnen Nagios, Apache, en Sendmail nu gestart worden op een manier die

past in het beveiligingsbeleid. De volgende commando's zorgen hiervoor:

```
# cd /etc/mail && make stop && \
setpmac biba/equal make start && setpmac biba/10\(10-10\) apachectl start && \
setpmac biba/10\(10-10\) /usr/local/etc/rc.d/nagios.sh forcestart
```

Controleer nogmaals om er zeker van te zijn dat alles juist werkt. Indien niet, controleer dan de logbestanden of de foutmeldingen. Gebruik het hulpprogramma sysctl(8) om de beveiligingsbeleidsmodule mac_biba(4) uit te schakelen en probeer om alles opnieuw op te starten, zoals gewoonlijk.



De gebruiker `root` kan zonder angst de afgedwongen beveiliging veranderen en de instellingenbestanden bewerken. Het volgende commando staat toe om het beveiligingsbeleid naar een lagere graad te degraderen voor een nieuw voortgebrachte shell:

```
# setpmac biba/10 csh
```

Om te voorkomen dat dit gebeurt, kan de gebruiker via login.conf(5) in een bereik worden gedwongen. Als setpmac(8) probeert om een commando buiten het bereik van het compartiment te draaien, zal er een fout worden teruggegeven en wordt het commando niet uitgevoerd. Zet in dit geval root op `biba/high(high-high)`.

15.16. Gebruikers afsluiten

Dit voorbeeld gaat over een relatief klein opslagsysteem met minder dan vijftig gebruikers. Gebruikers kunnen zich aanmelden, en mogen zowel gegevens opslaan als bronnen benaderen.

Voor dit scenario kunnen mac_bsdextended(4) gecombineerd met mac_seeotheruids(4) naast elkaar bestaan en zowel toegang tot systeemobjecten als tot gebruikersprocessen ontzeggen.

Begin door de volgende regel aan /boot/loader.conf toe te voegen:

```
mac_seeotheruids_load="YES"
```

Het beveiligingsbeleidsmodule mac_bsdextended(4) kan door volgende variabele in rc.conf geactiveerd worden:

```
ugidfw_enable="YES"
```

De standaardregels in /etc/rc.bsdextended zullen tijdens de systeeminitialisatie worden geladen; het kan echter nodig zijn om de standaardregels te wijzigen. Aangezien van deze machine alleen verwacht wordt dat het gebruikers bedient, kunnen alle regels uitgecommentarieerd blijven behalve de laatste twee. Deze forceren het standaard laden van systeemobjecten die eigendom zijn

van gebruikers.

Voeg de benodigde gebruikers toe aan deze machine en start opnieuw op. Probeer, voor testdoeleinden, u aan te melden als een andere gebruiker over twee consoles. Draai het commando `ps aux` om te zien of processen van andere gebruikers zichtbaar zijn. Probeer om ls(1) te draaien op de thuismap van een andere gebruiker, dit zou moeten mislukken.

Probeer niet te testen met de gebruiker `root` tenzij de specifieke `sysctl`'s om supergebruikertoegang te blokkeren zijn aangepast.



Wanneer een nieuwe gebruiker is toegevoegd, zit de mac_bsdextended(4)-regel van die gebruiker niet in de lijst van regelverzamelingen. Om de regelverzameling snel bij te werken, kan simpelweg de beveiligingsbeleidsmodule worden herladen met de gereedschappen kldunload(8) en kldload(8).

15.17. Problemen oplossen met het MAC-raamwerk

Tijdens de ontwikkeling hebben een aantal gebruikers problemen aangegeven met normale instellingen. Hieronder worden een aantal van die problemen beschreven:

15.17.1. De optie `multilabel` kan niet ingeschakeld worden op /

De vlag `multilabel` blijft niet ingeschakeld op de rootpartitie (/)!

Het lijkt er inderdaad op dat een paar procent van de gebruikers dit probleem heeft. Nadere analyse van het probleem doet vermoeden dat deze zogenaamde "bug" het resultaat is van òfwel onjuiste documentatie òfwel verkeerde interpretatie van de documentatie. Hoe het probleem ook is ontstaan, met de volgende stappen is het te verhelpen:

1. Wijzig /etc/fstab en stel de rootpartitie in op `ro` voor alleen-lezen.
2. Herstart in enkele-gebruikersmodus.
3. Draai `tunefs -l enable` op /.
4. Herstart in normale modus.
5. Draai `mount -urw`/ en wijzig `ro` terug in `rw` in /etc/fstab en start het systeem opnieuw.
6. Controleer de uitvoer van `mount` om zeker te zijn dat `multilabel` juist is ingesteld op het rootbestandssysteem.

15.17.2. X11-server start niet na MAC

Na het instellen van een beveiligde omgeving met MAC start X niet meer!

Dit kan komen door de MAC-beleidseenheid `partition` of door een verkeerde labeling van een van de MAC-labeling beleidseenheden. Probeer als volgt te debuggen:

1. Controleer de foutmelding. Als de gebruiker in de klasse `insecure` zit, kan de

beleidseenheid `partition` het probleem zijn. Zet de klasse voor de gebruiker terug naar de klasse `default` en herbouw de database met het commando `cap_mkdb`. Ga naar stap twee als hiermee het probleem niet is opgelost.

2. Controleer de labelbeleidseenheden nog een keer. Stel zeker dat het beleid voor de bewuste gebruiker, de X11-applicatie, en de onderdelen van /dev juist zijn ingesteld.
3. Als geen van beide methodes het probleem oplossen, stuur dan de foutmelding en een beschrijving van de omgeving naar de TrustedBSD-discussielijsten van de TrustedBSD website of naar de FreeBSD algemene vragen mailinglijst mailinglijst.

15.17.3. Error: _secure_path(3) cannot stat .login_conf

Bij het wisselen van de gebruiker `root` naar een andere gebruiker in het systeem, verschijnt de foutmelding `_secure_path: unable to state .login_conf`.

Deze melding komt meestal voor als de gebruiker een hogere labelinstelling heeft dan de gebruiker waarnaar wordt gewisseld. Als bijvoorbeeld gebruiker `joe` een standaardlabel `biba/low` heeft, dan kan gebruiker `root`, die een label `biba/high` heeft, de thuismap van `joe` niet zien. Dit gebeurt zonder rekening te houden met de mogelijkheid dat `root` met `su` de identiteit van `joe` heeft aangenomen. In dit scenario staat het integriteitsmodel van Biba niet toe dat `root` objecten kan zien van een lager integriteitsniveau.

15.17.4. De gebruikersnaam `root` is stuk!

In normale, of zelfs in enkelegebruikersmodus, wordt `root` niet herkend. Het commando `whoami` geeft 0 (nul) terug en `su` heeft als resultaat `who are you?`. Wat is er aan de hand?

Dit kan gebeuren als een labelbeleid is uitgeschakeld, òfwel door sysctl(8) òf doordat de beleidsmodule niet meer is geladen. Als de beleidseenheid (tijdelijk) is uitgeschakeld dan moet de database met aanmeldmogelijkheden opnieuw worden ingesteld, waarbij de optie `label` wordt verwijderd. Er dient voor te worden zorggedragen dat het bestand login.conf wordt ontdaan van alle opties met `label`, waarna de database opnieuw gebouwd kan worden met `cap_mkdb`.

Dit kan ook gebeuren als een beleid toegang verhinderd tot het bestand of de database master.passwd. Meestal wordt dit veroorzaakt door een beheerder die het bestand veranderd onder een label welke conflicteert met het globale beleid dat gebruikt wordt op het systeem. In deze gevallen wordt de gebruikersinformatie gelezen door het systeem en wordt de toegang geblokkeerd omdat het bestand het nieuwe label erft. Zet het beleid uit door middel van sysctl(8) en alles zou weer normaal moeten zijn.

Hoofdstuk 16. Security Event Auditing

16.1. Overzicht

Het besturingssysteem FreeBSD heeft ondersteuning voor diepgaande beveiligingsauditing van evenementen. Evenement auditing maakt het mogelijk dat er diepgaande en configureerbare logging van een variateit aan beveiligings-gerelateerde systeem evenementen, waaronder logins, configuratie wijzigingen, bestands- en netwerk toegang. Deze log regels kunnen erg belangrijk zijn voor live systeem monitoring, intrusion detection en postmortem analyse. FreeBSD implementeert Sun™'s gepubliceerde BSM API en bestandsformaat en is uitwisselbaar met zowel Sun™'s Solaris™ als Apple®'s Mac OS® X audit implementaties.

Dit hoofdstuk richt zich op de installatie en configuratie van evenement auditing. Het legt audit policies uit en geeft voorbeelden van audit configuraties.

Na het lezen van dit hoofdstuk weet de lezer:

- Wat evenement auditing is en hoe het werkt.
- Hoe evenement auditing geconfigureerd kan worden voor FreeBSD voor gebruikers en processen.
- Hoe de audittrail bekeken kan worden door gebruik te maken van de audit reduction en onderzoek programma's.

Voordat verder gegaan wordt moet het volgende bekend zijn:

- UNIX® en FreeBSD basishandelingen begrijpen (UNIX® beginselen).
- Bekend zijn met de basishandelingen van kernel configuratie/compilatie (De FreeBSD-kernel instellen).
- Bekend zijn met beveiliging en hoe dat relateert aan FreeBSD (Beveiliging).

De audit-faciliteiten hebben enkele bekende beperkingen waaronder dat niet alle beveiligings-relevante systeemevenementen geaudit kunnen worden en dat sommige login-mechanismes, zoals X11-gebaseerde display managers en programma's van erde partijen geen (goede) ondersteuning bieden voor het auditen van login-sessies van gebruikers.



De beveiligings evenement auditing faciliteit is in staat om erg gedetailleerde logs van systeem activiteiten op een druk systeem te genereren, trail bestands data kan erg groot worden wanneer er erg precieze details worden gevraagd, wat enkele gigabytes per week kan behalen in sommige configuraties. Beheerders moeten rekening houden met voldoende schijfruimte voor grote audit configuraties. Bijvoorbeeld het kan gewenst zijn om eigen bestandsysteem aan /var/audit toe te wijzen zo dat andere bestandssystemen geen hinder ondervinden als het audit bestandssysteem onverhoopt vol raakt.

16.2. Sleutelwoorden in dit hoofdstuk

Voordat dit hoofdstuk gelezen kan worden, moeten er een aantal audit gerelateerde termen uitgelegd worden:

- *evenement*: Een auditbaar evenement is elk evenement dat gelogged kan worden door het audit subsysteem. Voorbeelden van beveiligings gerelateerde evenementen zijn het creëeren van een bestand, het opzetten van een netwerk verbinding, of van een gebruiker die aanlogt. Evenementen zijn ofwel "attributable" wat betekend dat ze getraceerd kunnen worden naar een geauthoriseerde gebruiker, of "non-attributable" voor situaties waarin dat niet mogelijk is. Voorbeelden van non-attributable evenementen zijn elk evenement dat gebeurd voordat authorisatie plaatsvind in het login proces, zoals bij foutieve inlog pogingen.
- *class*: Evenement klassen zijn benoemde sets van gerelateerde evenementen en worden gebruikt in selectie expressies. Veel gebruikte klassen van evenementen zijn "bestands creatie" (fc), "exec" (ex) en "login_logout" (lo).
- *record*: Een record is een audit log regel die het beveiligings evenement beschrijft. Records bevatten een record evenement type, informatie over het onderwerp (de gebruiker) welke de actie uitvoerd, de datum en de tijd, informatie over de objecten of argumenten, en een conditie die aangeeft of de actie geslaagd of mislukt is.
- *trail*: Een audit trail, of log bestand bestaat uit een serie van audit records welke beveiligings evenementen beschrijft. Meestal lopen deze trails in chronologische orde, gebaseerd op de tijd dat het evenement optrad. Alleen geauthoriseerde processen mogen records toevoegen aan de audit trail.
- *selection expression*: Een selectie expressie is een string welke een lijst bevat van prefixes en audit evenement klasse namen die overeenkomen met evenementen.
- *preselection*: Het proces waarbij het systeem bepaald welke evenementen interessant zijn voor de beheerder, zodat wordt voorkomen dat er audit records worden gegenereerd voor evenementen die niet interessant zijn. De "preselection" configuratie gebruikt een serie van selectie expressies om te identificeren welke klassen van evenementen van toepassing zijn op gebruikers en globale instellingen voor zowel geauthoriseerde als ongeauthoriseerde processen.
- *reduction*: Het proces waarbij records van bestaande audit trails worden geselecteerd voor bewaring, uitprinten of analyse. Ook is dit het proces waarbij ongewenste audit records worden verwijderd uit het audit trail. Door gebruik te maken van reduction kunnen beheerders policies implementeren die het bewaren van audit data verzorgen. Bijvoorbeeld gedetailleerde audit trails kunnen één maand bewaard worden maar erna worden trails gereduceerd zodat alleen login informatie bewaard worden voor archiverings redenen.

16.3. Installeren van audit ondersteuning.

Ondersteuning in de gebruikersomgeving voor evenement auditing wordt geïnstalleerd als onderdeel van het basis FreeBSD besturingssysteem. Kernel-ondersteuning voor evenement-auditing wordt standaard meegenomen tijdens compilatie, maar moet expliciet in de kernel gecompileerd worden door de volgende regel toe te voegen aan het configuratiebestand van de kernel:

```
options AUDIT
```

Bouw en herinstalleer de kernel volgens het normale proces zoals beschreven in De FreeBSD-kernel instellen.

Zodra een audit ondersteunende kernel is gebouwd en geïnstalleerd en deze is opgestart kan de audit daemon aangezet worden door de volgende regel aan rc.conf(5) toe te voegen:

```
auditd_enable="YES"
```

Audit ondersteuning moet daarna aangezet worden door een herstart van het systeem of door het handmatig starten van de audit daemon:

```
service auditd start
```

16.4. Audit Configuratie

Alle configuratie bestanden voor beveiligings audit kunnen worden gevonden in /etc/security. De volgende bestanden moeten aanwezig zijn voor de audit daemon wordt gestart:

- audit_class - Bevat de definities van de audit klasses.
- audit_control - Controleert aspecten van het audit subsysteem, zoals de standaard audit klassen, minimale hoeveelheid diskruimte die moet overblijven op de audit log schijf, de maximale audit trail grootte, etc.
- audit_event - Tekst namen en beschrijvingen van systeem audit evenementen, evenals een lijst van klassen waarin elk evenement zich bevind.
- audit_user - Gebruiker specifieke audit benodigdheden welke gecombineerd worden met de globale standaarden tijdens het inloggen.
- audit_warn - Een bewerkbaar shell script gebruikt door de auditd applicatie welke waarschuwings berichten genereert in bijzondere situaties zoals wanneer de ruimte voor audit records te laagis of wanneer het audit trail bestand is geroteerd.



Audit configuratie bestanden moeten voorzichtig worden bewerkt en onderhouden, omdat fouten in de configuratie kunnen resulteren in het verkeerd loggen van evenementen.

16.4.1. Evenement selectie expressies

Selectie expressies worden gebruikt op een aantal plaatsen in de audit configuratie om te bepalen welke evenementen er geaudit moeten worden. Expressies bevatten een lijst van evenement klassen welke gelijk zijn aan een prefix welke aangeeft of gelijke records geaccepteerd moeten worden of genegeerd en optioneel om aan te geven of de regel is bedoeld om succesvolle of mislukte operaties te matchen. Selectie expressies worden geevalueerd van links naar rechts en

twee expressies worden gecombineerd door de één aan de ander toe te voegen.

De volgende lijst bevat de standaard audit evenement klassen welke aanwezig zijn in het audit_class bestand:

- `all` - *all* - Matched alle evenement klasses.
- `ad` - *administrative* - Administratieve acties welke uitgevoerd worden op het gehele systeem.
- `ap` - *application* - Applicatie gedefinieerde acties.
- `cl` - *file close* - Audit aanroepen naar de `close` systeem aanroep.
- `ex` - *exec* - Audit programma uitvoer. Het auditen van command line argumenten en omgevings variabelen wordt gecontroleerd via audit_control(5) door gebruik te maken van de `argv` en `envv` parameters in de `policy` setting.
- `fa` - *file attribute access* - Audit de toevoeging van object attributen zoals stat(1), pathconf(2) en gelijkwaardige evenementen.
- `fc` - *file create* - Audit evenementen waar een bestand wordt gecreëerd als resultaat.
- `fd` - *file delete* - Audit evenementen waarbij bestanden verwijderd worden.
- `fm` - *file attribute modify* - Audit evenementen waarbij bestandsattribuut wijzigingen plaatsvinden zoals bij chown(8), chflags(1), flock(2), etc.
- `fr` - *file read* - Audit evenementen waarbij data wordt gelezen, bestanden worden geopend voor lezen etc.
- `fw` - *file write* - Audit evenementen waarbij data wordt geschreven, bestanden worden geschreven of gewijzigd, etc.
- `io` - *ioctl* - Audit het gebruik van de ioctl(2) systeem aanroep.
- `ip` - *ipc* - Audit verschillende vormen van Inter-Process Communication, zoals POSIX pipes en System V IPC operaties.
- `lo` - *login_logout* - Audit login(1) en logout(1) evenementen die plaatsvinden op het systeem.
- `na` - *non attributable* - Audit non-attributable evenementen.
- `no` - *invalid class* - Matched geen enkel audit evenement.
- `nt` - *network* - Audit evenementen die gerelateerd zijn aan netwerk acties zoals connect(2) en accept(2).
- `ot` - *other* - Audit diverse evenementen.
- `pc` - *process* - Audit process operaties zoals exec(3) en exit(3)

Deze audit evenement klassen kunnen veranderd worden door het wijzigingen van de audit_class en audit_event configuratie bestanden.

Elke audit klasse in de lijst wordt gecombineerd met een voorzetsel welke aangeeft of er succesvolle of mislukte operaties hebben plaatsgevonden en of de regel wordt toegevoegd of verwijderd van het matchen van de klasse en het type.

- (none) Audit zowel succesvolle als mislukte informatie van het evenement.
- `+` Audit succesvolle evenementen in deze klasse.

- `-` Audit mislukte evenementen in deze klasse.
- `^` Audit geen enkele succesvolle of mislukte evenementen in deze klasse.
- `^+` Audit geen succesvolle evenementen in deze klasse.
- `^-` Audit geen mislukte evenementen in deze klasse.

De volgende voorbeeld selectie strings selecteren zowel succesvolle als mislukte login/logout evenementen, maar alleen succesvolle uitvoer evenementen:

```
lo,+ex
```

16.4.2. Configuratie bestanden

In de meeste gevallen moet een beheerder twee bestanden wijzigingen wanneer het audit systeem wordt geconfigureerd: audit_control en audit_user. Het eerste controleert systeem brede audit eigenschappen en policies, het tweede kan gebruikt worden om diepgaande auditing per gebruiker uit te voeren.

Het audit_control bestand

Het audit_control bestand specificeert een aantal standaarden van het audit subsysteem. Als de inhoud bekeken wordt van dit bestand is het volgende te zien:

```
dir:/var/audit
flags:lo
minfree:20
naflags:lo
policy:cnt
filesz:0
```

De `dir` optie wordt gebruikt om één of meerdere directories te specificeren die gebruikt worden voor de opslag van audit logs. Als er meer dan één directory wordt gespecificeerd, worden ze op volgorde gebruikt naarmate ze gevuld worden. Het is standaard dat audit geconfigureerd wordt dat audit logs worden bewaard op een eigen bestandssysteem, om te voorkomen dat het audit subsysteem en andere subsystemen met elkaar botsen als het bestandssysteem volraakt.

Het `flags` veld stelt de systeem brede standaard preselection maskers voor attributable evenementen in. In het voorbeeld boven worden succesvolle en mislukte login en logout evenementen geaudit voor alle gebruikers.

De `minfree` optie definieerd het minimale percentage aan vrije ruimte voor dit bestandssysteem waar de audit trails worden opgeslagen. Wanneer deze limiet wordt overschreven wordt er een waarschuwing gegenereerd. In het bovenstaande voorbeeld wordt de minimale vrije ruimte ingesteld op 20 procent.

De `naflags` optie specificeerd audit klasses welke geaudit moeten worden voor non-attributed evenementen zoals het login proces en voor systeem daemons.

De `policy` optie specificeert een komma gescheiden lijst van policy vlaggen welke diverse aspecten van het audit proces beheren. De standaard `cnt` vlag geeft aan dat het systeem moet blijven draaien ook al treden er audit fouten op (deze vlag wordt sterk aangeraden). Een andere veel gebruikte vlag is `argv`, wat het mogelijk maakt om command line argumenten aan de execve(2) systeem aanroep te auditen als onderdeel van het uitvoeren van commando's.

De `filesz` optie specificeert de maximale grootte in bytes hoeveel een audit trail bestand mag groeien voordat het automatisch getermineerd en geroteerd wordt. De standaard, 0, schakelt automatische log rotatie uit. Als de gevraagde bestands grootte niet nul is en onder de minimale 512k zit, wordt de optie genegeerd en wordt er een log bericht gegenereerd.

Het audit_user bestand

Het audit_user bestand staat de beheerder toe om verdere audit benodigdheden te specificeren voor gebruikers. Elke regel configureert auditing voor een gebruiker via twee velden, het eerste is het `alwaysaudit` veld, welke een set van evenementen specificeert welke altijd moet worden geaudit voor de gebruiker, en de tweede is het `neveraudit` veld, welke een set van evenementen specificeerd die nooit geaudit moeten worden voor de gebruiker.

Het volgende voorbeeld audit_user bestand audit login/logout evenementen en succesvolle commando uitvoer voor de `root` gebruiker, en audit bestands creatie en succesvolle commando uitvoer voor de `www` gebruiker. Als dit gebruikt wordt in combinatie met het voorbeeld audit_control bestand hierboven, is de `root` regel dubbelop en zullen login/logout evenementen ook worden geaudit voor de `www` gebruiker.

```
root:lo,+ex:no
www:fc,+ex:no
```

16.5. Het audit subsysteem beheren.

16.5.1. Audit trails inzien

Audit trails worden opgeslagen in het BSM binaire formaat, dus ondersteunende programma's moeten worden gebruikt om de informatie te wijzigen of converteren naar tekst. Het praudit(1) commando converteert trail bestanden naar een simpel tekst formaat; het auditreduce(1) commando kan gebruikt worden om de audit trail te reduceren voor analyse, archivering of voor het uitprinten van de data. `auditreduce` ondersteund een variateit aan selectie parameters, zoals evenement type, evenement klasse, gebruiker, datum of tijd van het evenement en het bestandspad of object dat gebruikt wordt.

Bijvoorbeeld, het `praudit` programma zal een dump maken van de volledige inhoud van een gespecificeerd audit log bestand in normale tekst:

```
# praudit /var/audit/AUDITFILE
```

Waar AUDITFILE het audit bestand is dat ingelezen moet worden.

Audit trails bestaan uit een serie van audit records die gevormd worden door tokens, welke `praudit` sequentieel print één per regel. Elke token is van een specifiek type, zoals een `header` welke de audit record header bevat, of `path` welke het bestandspad bevat van een lookup. Het volgende is een voorbeeld van een `execve` evenement:

```
header,133,10,execve(2),0,Mon Sep 25 15:58:03 2006, + 384 msec
exec arg,finger,doug
path,/usr/bin/finger
attribute,555,root,wheel,90,24918,104944
subject,robert,root,wheel,root,wheel,38439,38032,42086,128.232.9.100
return,success,0
trailer,133
```

Deze audit representeert een succesvolle `execve` aanroep, waarbij het commando `finger doug` is aangeroepen. Het argument token bevat beide commando's gerepresenteerd door de shell aan de kernel. Het `path` token bevat het pad naar het uitvoerbare bestand zoals opgezocht door de kernel. Het `attribute` token beschrijft de binary en om precies te zijn bevat het de bestands mode welke gebruikt kan worden om te zien of het bestand setuid was. Het `subject` token beschrijft het onderwerp proces en bevat sequentieel het audit gebruikers ID, effectieve gebruikers ID en groep ID, echte gebruikers ID, groep ID, proces ID, sessie ID, port ID en login adres. Let op dat het audit gebruikers ID en het echte gebruikers ID van elkaar verschillen omdat de gebruiker `robert` veranderd is naar de `root` gebruiker voordat het commando werd uitgevoerd, maar welke geaudit wordt als de originele geauthoriseerde gebruiker. Als laatste wordt de `return` token gebruikt om aan te geven dat er een succesvolle uitvoer is geweest en `trailer` geeft het einde aan van het record.

`praudit` ook een XML output formaat, welke geselecteerd kan worden door gebruik te maken van het `x` argument.

16.5.2. Het reduceren van audit trails

Omdat audit logs erg groot kunnen worden, zal de beheerder waarschijnlijk een subset van records willen selecteren om te gebruiken, zoals records die gekoppeld zijn aan een specifieke gebruiker:

```
# auditreduce -u trhodes /var/audit/AUDITFILE | praudit
```

Dit selecteert alle audit records die geproduceert zijn voor de gebruiker `trhodes` die opgeslagen is in het AUDITFILE bestand.

16.5.3. Delegeren van audit onderzoek rechten

Leden van de `audit` groep krijgen permissie om de audit trails te lezen in /var/audit; standaard is deze groep leeg en kan alleen de `root` gebruiker deze audit trails lezen. Gebruikers kunnen toegevoegd worden aan de `audit` groep zodat onderzoek rechten kunnen worden gedelegeerd aan de geruiker. Omdat de mogelijkheid van het inzien van audit log inhoud significante inzicht kan geven in het gedrag van gebruikers en processen, wordt het aangeraden dat de delagatie van onderzoek rechten eerst goed overdacht wordt.

16.5.4. Live monitoren door gebruik van audit pipes

Audit pipes zijn gecloonde pseudo-devices in het device bestands systeem, welke applicaties toestaat om een tap te plaatsen in de live audit record stream. Dit is primair interessant voor schrijvers van intrusion detection en systeem monitoring applicaties. Echter, voor een beheerder is het audit pipe device een makkelijke manier om live monitoring toe te staan zonder dat er problemen kunnen ontstaan met het eigenaarschap van het audit trail bestand, of dat een log rotatie de evenementen stroom in de weg zit. Om de live audit evenementen stroom te kunnen inzien is het volgende commando benodigd:

```
# praudit /dev/auditpipe
```

Standaard zijn de audit pipe device nodes alleen toegankelijk voor de `root` gebruiker. Om deze toegankelijk te maken voor leden van de `audit` groep, moet een `devfs` regel toegevoegd worden aan het devfs.rules bestand:

```
add path 'auditpipe*' mode 0440 group audit
```

Zie devfs.rules(5) voor meer informatie over het configureren van het devfs bestands systeem.



Het is makkelijk om audit evenement terugkoppeling cyclussen te creëeren, waarbij het tonen van elk audit evenement resulteert in het genereren van nog meer audit evenementen. Bijvoorbeeld, als alle netwerk I/O wordt geaudit en praudit(1) wordt gestart vanuit een SSH sessie, wordt er een grote continue stroom aan audit evenementen gegenereert doordat elk getoond evenement een nieuw evenement genereert. Het is verstandig om `praudit` te draaien op een audit pipe device voor sessies zonder diepgaande I/O auditing om te voorkomen dat dit gebeurd.

16.5.5. Het roteren van audit trail bestanden

Audit trails worden alleen beschreven door de kernel en alleen beheerd worden door de audit daemon, auditd. Beheerders mogen geen gebruik maken van newsyslog.conf(5) of soortgelijke programma's om de audit files te roteren. In plaats daarvan kan het `audit` management programma gebruikt worden om auditing te stoppen, het audit systeem te herconfigureren en log rotatie uit te voeren. Het volgende commando zorgt ervoor dat de audit daemon een nieuwe audit log maakt, en vervolgens de kernel een signaal stuurt om het nieuwe logbestand te gaan gebruiken. Het oude logbestand wordt getermineerd en hernoemd, waarna het bestand gemanipuleerd kan worden door de beheerder.

```
# audit -n
```



Als de auditd daemon op dit moment niet actief is, zal het commando falen en zal er een error bericht worden geproduceerd.

Als de volgende regel wordt toegevoegd aan het /etc/crontab bestand, zal er elke twaalf uur een rotatie plaatsvinden door middel van cron(8):

```
0     */12      *       *       *       root    /usr/sbin/audit -n
```

Deze wijziging wordt van kracht op het moment dat het nieuwe /etc/crontab bestand wordt opgeslagen.

Automatische rotatie van het audit trail bestand gebaseerd op de bestand grootte is mogelijk via de `filesz` optie in audit_control(5) en wordt beschreven in de configuratie bestanden sectie van dit hoofdstuk.

16.5.6. Audit trails comprimeren

Omdat audit trail bestanden erg groot kunnen worden, is het meestal gewenst om de trails te comprimeren of op een andere manier te archiveren zodra ze afgesloten zijn door de audit daemon. Het audit_warn script kan gebruikt worden om bewerkte operaties te doen voor een variateit aan audit gerelateerde evenementen inclusief een nette terminatie van audit trails wanneer deze geroteerd worden. Bijvoorbeeld het volgende kan worden toegevoegd aan het audit_warn script, dat de audit trails comprimeert zodra ze afgesloten worden:

```
#
# Compress audit trail files on close.
#
if [ "$1" = closefile ]; then
        gzip -9 $2
fi
```

Andere archiverings activiteiten kunnen zijn het kopieren van trail bestanden naar een gecentraliseerde server, het verwijderen van oude trail bestanden of het reduceren van de audit trail om onnodige records te verwijderen. Het script zal alleen draaien als audit trail bestanden netjes worden afgesloten, wat betekend dat het script niet uitgevoerd wordt op trails die niet netjes afgesloten zijn, waardoor bestanden corrupt kunnen raken.

Hoofdstuk 17. Opslag

17.1. Overzicht

Dit hoofdstuk behandelt het gebruik van schijven in FreeBSD. Dit omvat geheugenschijven, schijven die met het netwerk verbonden zijn, SCSI/IDE-opslagapparaten en apparaten die gebruik maken van de USB-interface.

Na het lezen van dit hoofdstuk weet de lezer:

- Welke terminologie FreeBSD gebruikt om de gegevensindeling op een fysieke schijf te beschrijven (partities en slices);
- Hoe aanvullende harde schijven aan een systeem toe te voegen;
- Hoe FreeBSD in te stellen om het gebruik te laten maken van USB-opslagapparaten;
- Hoe virtuële bestandssystemen, zoals geheugenschijven, aan te maken;
- Hoe quota te gebruiken om het schijfgebruik te beperken;
- Hoe schijven te versleutelen om ze tegen inbrekers te beschermen;
- Hoe vanuit FreeBSD CD's en DVD's aan te maken en te branden;
- Wat de verschillende mogelijkheden zijn voor opslagmedia voor back-ups;
- Hoe back-upprogramma's te gebruiken die beschikbaar zijn in FreeBSD;
- Hoe een back-up naar diskettes te maken;
- Wat bestandssysteem snapshots zijn en hoe ze efficiënt te gebruiken.

Aangeraden voorkennis:

- Hoe een nieuwe FreeBSD-kernel in te stellen en te installeren (De FreeBSD-kernel instellen).

17.2. Apparaatnamen

De volgende lijst noemt de fysieke opslagapparaten die in FreeBSD ondersteund worden, samen met de bijhorende namen.

Tabel 2. Naamconventies voor fysieke Schijven

Type medium	Apparaatnaam medium
IDE harde schijven	`ad`
IDE CD-ROM-stations	`acd`
SCSI harde schijven en USB-apparaten voor massa-opslag	`da`
SCSI CD-ROM-schijven	`cd`
Overige niet-standaard-CD-ROM-stations	`mcd` voor Mitsumi CD-ROM en `scd` voor Sony CD-ROM apparaten.

Type medium	Apparaatnaam medium
Diskettestations	`fd`
SCSI bandstations	`sa`
IDE bandstations	`ast`
Flashdrives	`fla` voor DiskOnChip® flashapparaten
RAID-schijven	`aacd` voor Adaptec® AdvancedRAID, `mlxd` en `mlyd` voor Mylex®, `amrd` voor AMI MegaRAID®, `idad` voor Compaq Smart RAID, `twed` voor 3ware® RAID.

17.3. Schijven toevoegen

De volgende sectie beschrijft hoe een nieuwe SCSI schijf aan een machine toe te voegen die slechts een enkele drive heeft. Ten eerste dient de computer uitgeschakeld te worden en dient de schijf volgens de instructies van de computer, controller en schijffabrikant geïnstalleerd te worden. Wegens de grote variéteiten om dit soort procedures uit te voeren, vallen de details buiten het bereik van dit document.

Er dient als gebruiker `root` ingelogd te worden. Nadat de schijf is toegevoegd, dient /var/run/dmesg.boot bekeken te worden om er zeker van te zijn dat de nieuwe schijf is gevonden. Volgens het voorbeeld heet de nieuw toegevoegde schijf da1 en die wordt aangekoppeld op /1 (als er een IDE-schijf wordt toegevoegd, is de apparaatnaam ad1).

FreeBSD draait op IBM-PC-compatibele computers. Daarom moet het rekening houden met de PC-BIOS-partities. Deze wijken af van de traditionele BSD-partities. Een PC-schijf bevat tot vier ingangen voor BIOS-partities. Indien de schijf geheel aan FreeBSD wordt gewijd, kan de *toegewijde* -modus gebruikt worden. In het andere geval moet FreeBSD binnen één van de vier PC-BIOS-partities draaien. De PC-BIOS-partities worden door FreeBSD *slices* genoemd om ze niet met de traditionele BSD-partities te verwarren. Slices kunnen ook op een schijf worden gebruikt die toegewijd is aan FreeBSD, maar in een computer zit die ook andere besturingssystemen heeft geïnstalleerd. Dit is een goede manier om verwarring met het programma `fdisk` van andere, niet-FreeBSD besturingssystemen te voorkomen.

Als er met slices gewerkt wordt, wordt de schijf toegevoegd als /dev/da1s1e. Dit moet worden gelezen als: SCSI-schijf, eenheid 1 (tweede SCSI-schijf), slice 1 (PC-BIOS-partitie 1) en BSD-partitie e. Als de schijf toegewijd is, wordt deze simpelweg als /dev/da1e toegevoegd.

Omdat 32-bit-integers worden gebruikt om het aantal sectoren op te slaan, is bsdlabel(8) beperkt tot 2^32-1 sectoren per schijf, wat meestal neerkomt op 2 TB. Het programma fdisk(8) staat geen hogere startsector toe dan 2^32-1 en geen grotere lengte dan 2^32-1, meestal worden hiermee partities tot 2 TB begrensd en schijven tot 4 TB. Het formaat van sunlabel(8) is beperkt tot 2^32-1 sectoren per partitie en 8 partities per schijf, in totaal dus 16 TB. Voor grotere schijven kan gpart(8) worden gebruikt om GPT-partities aan te maken. GPT heeft het bijkomende voordeel dat het niet tot 4 slices beperkt is.

17.3.1. sysinstall(8) gebruiken

1. Navigeren door sysinstall

   `sysinstall` kan gebruikt worden om een nieuwe schijf te partitioneren en te labelen met eenvoudig te gebruiken menu's. Hiervoor dient òfwel als gebruiker `root` ingelogd te zijn, òfwel gebruik te worden gemaakt van `su`. Draai `sysinstall` en ga naar het menu `Configure`. Scroll binnen het `FreeBSD Configuration Menu` naar beneden en kies de optie `Fdisk`.

2. fdisk partitie-bewerker

   Eenmaal binnen fdisk kan op `A` gedrukt worden om de gehele schijf voor FreeBSD te gebruiken. Wanneer gevraagd wordt of het systeem compatibel dient te blijven met mogelijk toekomstige besturingssystemen, dient met `YES` geantwoord te worden. Met `W` kunnen de veranderingen naar de schijf worden geschreven. Nu dient de FDISK-bewerker verlaten te worden door op `Q` te drukken. Vervolgens wordt er een vraag gesteld over het "Master Boot Record". Omdat er een schijf aan een reeds draaiend systeem wordt toegevoegd, dient hier `None` gekozen te worden.

3. Schijflabelbewerker

   Vervolgens dient sysinstall verlaten en opnieuw gestart te worden. Volg bovenstaande aanwijzingen, maar kies deze keer voor de optie `Label`. Dit geeft toegang tot de `Disk Label Editor`. Hier worden de traditionele BSD-partities aangemaakt. Een schijf kan tot acht partities bevatten, gelabeld `a-h`. Enkele partitielabels hebben een speciale functie. De partitie `a` wordt gebruikt voor de rootpartitie (/). Alleen de systeemschijf (bijvoorbeeld de schijf van waaruit opgestart wordt) moet een partitie `a` hebben. De partitie `b` wordt voor swappartities gebruikt, en het is mogelijk om vele schijven met swappartities te hebben. De partitie `c` adresseert de gehele schijf in toegewijde modus, of de gehele FreeBSD-slice in slice-modus. De andere partities zijn voor algemeen gebruik.

   sysinstall's Labelbewerker heeft een voorkeur voor de partitie `e` voor niet-root-niet-swap-partities. Binnen de Labelbewerker dient een enkel bestandssysteem te worden aangemaakt door op `C` te drukken. Kies `FS` wanneer gevraagd wordt of dit een FS (file system) of swap wordt, en geef een koppelpunt in (bijvoorbeeld /mnt). Wanneer een schijf in post-installatie-modus wordt toegevoegd, maakt sysinstall geen ingangen aan in /etc/fstab, dus dan is het opgegeven koppelpunt niet van belang.

   Nu kan het nieuwe label naar de schijf worden geschreven en er een bestandssysteem op aangemaakt worden. Dit kan gedaan worden door op `W` te drukken. Fouten van sysinstall dat de nieuwe partitie niet aankoppeld kon worden kunnen genegeerd worden. De Labelbewerker en sysinstall kunnen nu volledig verlaten worden.

4. Afronden

   De laatste stap bestaat uit het bewerken van /etc/fstab om hier een regel voor de nieuwe schijf aan toe te voegen.

17.3.2. Het gebruik van opdrachtregelgereedschappen

Het gebruik van slices

Deze installatie zorgt ervoor dat de schijf correct samenwerkt met andere besturingssystemen die eventueel op de computer zijn geïnstalleerd en dat de `fdisk`-gereedschappen van andere besturingssystemen niet verward raken. Het wordt aangeraden om deze methode te gebruiken voor de installatie van nieuwe schijven. Gebruik de `toegewijde` modus alleen als hier een goede reden voor bestaat!

```
# dd if=/dev/zero of/dev/da1 bs=1k count=1
# fdisk -BI da1 # Initialiseer de nieuwe schijf.
# bsdlabel -B -w da1s1 auto # Label de schijf.
# bsdlabel -e da1s1 # Bewerk de zojuist aangemaakte schijflabel en voeg partities toe.
# mkdir -p /1
# newfs /dev/da1s1e # Herhaal dit voor alle aangemaakte partities.
# mount /dev/da1s1e /1 # Mount de partitie(s).
# vi /etc/fstab # Voeg de juiste regel(s) aan /etc/fstab toe.
```

Vervang voor een IDE-schijf da door ad.

Toegewijd

Indien de nieuwe schijf niet met een ander besturingssysteem gedeeld wordt, kan de `toegewijde` modus gebruikt worden. Denk eraan dat deze modus besturingssystemen van Microsoft kan verwarren. Ze richten echter geen schade aan. IBM's OS/2® "fatsoeneert" echter partities die het niet begrijpt.

```
# dd if=/dev/zero of=/dev/da1 bs=1k count=1
# bsdlabel -Bw da1 auto
# bsdlabel -e da1 # Maak de `e'-partitie aan.
# newfs /dev/da1e
# mkdir -p /1
# vi /etc/fstab # Voeg een regel voor /dev/da1e toe.
# mount /1
```

Een alternatieve methode is:

```
# dd if=/dev/zero of=/dev/da1 count=2
# bsdlabel /dev/da1 | bsdlabel -BR da1 /dev/stdin
# newfs /dev/da1e
# mkdir -p /1 # Voeg een regel voor /dev/da1e toe.
# mount /1
```

17.4. RAID

17.4.1. Software RAID

Concatenated Disk Driver (CCD) instellingen

Bij het kiezen van een medium voor massa-opslag zijn de belangrijkste afwegingen snelheid, betrouwbaarheid en kosten. Het komt zelden voor dat alle drie in balans zijn. Normaalgesproken is een snel, betrouwbaar apparaat voor massa-opslag duur en kosten sparen gaat ten koste van òfwel snelheid òfwel betrouwbaarheid.

Bij het ontwerpen van het onderstaande systeem werd primair op de kosten gelet, gevolgd door snelheid en als laatste betrouwbaarheid. De overdrachtsnelheid van gegevens wordt voor dit systeem uiteindelijk beperkt door het netwerk. En hoewel betrouwbaarheid erg belangrijk is, wordt onderstaande CCD-schijf gebruikt voor het serven van on-line gegevens die reeds volledig op CD-R's zijn geback-upt en eenvoudig vervangen kunnen worden.

De eerste stap in het kiezen van een massa-opslagoplossing is het bepalen van de eigen benodigdheden. Indien snelheid belangrijker is dan betrouwbaarheid of kosten, wijkt de oplossing af van het systeem dat in deze sectie wordt beschreven.

Hardware installeren

Als aanvulling op de IDE systeemschijf zijn drie Western Digital IDE-schijven van 30 GB, 5400 RPM vanuit de kern van de onderstaande CCD-schijf aanwezig, die ongeveer 90 GB aan on-line opslag bieden. Ideaal gezien heeft iedere IDE-schijf een eigen IDE-controller en kabel, maar om de kosten te minimaliseren zijn geen aanvullende IDE-kabels gebruikt. In plaats hiervan zijn de schijven zodanig met jumpers ingesteld dat elke IDE-controller één master en één slave heeft.

Tijdens het opnieuw opstarten werd het systeem-BIOS zodanig ingesteld dat het automatisch de aangekoppelde schijven detecteerde. Het was belangrijker dat FreeBSD ze tijdens het opnieuw opstarten herkende:

```
ad0: 19574MB <WDC WD205BA> [39770/16/63] at ata0-master UDMA33
ad1: 29333MB <WDC WD307AA> [59598/16/63] at ata0-slave UDMA33
ad2: 29333MB <WDC WD307AA> [59598/16/63] at ata1-master UDMA33
ad3: 29333MB <WDC WD307AA> [59598/16/63] at ata1-slave UDMA33
```



Indien FreeBSD niet alle schijven detecteert, moet gecontroleerd worden of de jumpers juist zijn ingesteld. De meeste IDE-schijven hebben ook een jumper voor "Cable Select". Dit is *niet* de jumper voor de master/slave-instelling. Voor hulp met het identificeren van de juiste jumper dient de documentatie van de schijf geraadpleegd te worden.

CCD installeren

Het stuurprogramma ccd(4) biedt de mogelijkheid om meerdere identieke schijven aaneen te rijgen tot één logisch bestandssysteem. Om gebruik te kunnen maken van ccd(4) is een kernel met

ingebouwde ondersteuning voor ccd(4) nodig. De volgende regel dient toegevoegd te worden aan het kernelinstellingenbestand en de kernel dient opnieuw gebouwd en geïnstalleerd te worden:

```
device  ccd
```

Om ccd(4) te installeren dient eerst bsdlabel(8) gebruikt te worden om de schijven te labelen:

```
bsdlabel -w -ad1 auto
bsdlabel -w ad2 auto
bsdlabel -w ad3 auto
```

Bovenstaande maakt een schijflabel aan voor ad1c, ad2c en ad3c die de gehele schijf beslaat.

Vervolgens dient het labeltype van de schijf veranderd te worden. Voor het bewerken van de schijven kan bsdlabel(8) gebruikt worden:

```
bsdlabel -e ad1
bsdlabel -e ad2
bsdlabel -e ad3
```

Dit zorgt ervoor dat het huidige schijflabel van elke schijf met de tekstverwerker wordt geopend die door de omgevingsvariabele `EDITOR` wordt gespecificeerd, vaak vi(1).

Een ongewijzigd schijflabel ziet er ongeveer als volgt uit:

```
8 partitions:
# size  offset  fstype  [fsize  bsize bps/cpg]
c: 60074784 0 unused  0 0 0 # (Cyl. 0 - 59597)
```

Er dient een nieuwe partitie `e` toegevoegd te worden die door ccd(4) gebruikt kan worden. Deze kan gewoonlijk van partitie `c` overgenomen worden, maar het `fstype moet 4.2BSD` zijn. Het schijflabel ziet er nu ongeveer als volgt uit:

```
8 partitions:
# size  offset  fstype  [fsize  bsize bps/cpg]
c:  60074784  0 unused  0 0 0 # (Cyl. 0 - 59597)
e:  60074784  0 4.2BSD  0 0 0 # (Cyl. 0 - 59597)
```

Bestandssysteem aanmaken

Nu alle schijven gelabeld zijn, moet de ccd(4) gebouwd worden. Om dit te doen, dient ccdconfig(8) gebruikt te worden met opties die ongeveer gelijk zijn aan de volgende:

```
ccdconfig ccd0 32 0 /dev/ad1e /dev/ad2e /dev/ad3e
```

Hieronder staat het gebruik en de betekenis van elke optie:

- Het eerste argument is het in te stellen apparaat, in dit geval /dev/ccd0c. Het gedeelte /dev/ is optioneel.
- De interleave voor het bestandssysteem. De interleave definiëert de grootte van een stripe in schijfblokken, elk schijfblok is normaalgesproken 512 bytes groot. Een interleave van 32 is dus 16.384 bytes groot.
- Vlaggen voor ccdconfig(8). Indien het gewenst is om schijfspiegeling aan te zetten, kan er hier een vlag voor gespecificeerd worden. Deze opstelling biedt geen spiegeling voor ccd(4), dus is die op 0 (nul) ingesteld.
- De laatste argumenten voor ccdconfig(8) zijn de apparaten die in de rij geplaatst dienen te worden. Voor elk apparaat dient de complete padnaam gebruikt te worden.

Nadat ccdconfig(8) gedraaid is, is de ccd(4) ingesteld. Er kan een bestandssysteem worden geïnstalleerd. Er kan in newfs(8) worden gekeken voor opties, of het draaien van het onderstaande commando is ook toereikend:

```
newfs /dev/ccd0c
```

Alles automatisch maken

In het algemeen is het wenselijk om de ccd(4) telkens te mouten wanneer er opnieuw opgestart wordt. Dit dient eerst ingesteld te worden. Met het volgende commando worden de huidige instellingen naar /etc/ccd.conf geschreven:

```
ccdconfig -g > /etc/ccd.conf
```

Tijdens het opstarten draait het script `/etc/rc ccdconfig -C` indien /etc/ccd.conf bestaat. Dit stelt automatisch de ccd(4) in, zodat die kan worden aangekoppeld.



Indien er in enkele-gebruiker-modus wordt opgestart, dient het volgende commando te worden uitgevoerd om de rij in te stellen voordat de ccd(4) aangekoppeld kan worden:

```
ccdconfig -C
```

Om de ccd(4) automatisch aan te koppelen, kan er een regel voor de ccd(4) in /etc/fstab geplaatst worden, zodat die tijdens het opstarten aangekoppeld wordt:

```
/dev/ccd0c  /media  ufs rw  2   2
```

Volumebeheerder Vinum

De volumebeheerder Vinum is een blokstuurprogramma dat virtuele schijven implementeert. Het isoleert schijfhardware van de blokapparaat-interface en projecteert gegevens op een manier die de flexibiliteit, prestatie en betrouwbaarheid verhoogt in vergelijking met de traditionele slice-blik op schijfopslag. vinum(4) implementeert de modellen RAID-0, RAID-1 en RAID-5, zowel individueel als als combinatie.

17.4.2. Hardwarematige RAID

FreeBSD ondersteunt ook een verscheidenheid aan hardwarematige RAID-stuurprogramma's. Deze apparaten besturen een RAID-deelsysteem zonder dat er FreeBSD-specifieke software nodig is om de rij te beheren.

Door gebruik te maken van een BIOS die op de kaart aanwezig is, beheert de kaart de meeste schijfbewerkingen zelf. Nu volgt een korte beschrijving van een opzet waarbij een Promise IDE-stuurprogramma is gebruikt. Wanneer deze kaart geïnstalleerd en het systeem opgestart is, beeldt het een prompt af waarbij om informatie wordt gevraagd. De instructies dienen opgevolgd te worden om bij het instelscherm van de kaart te komen. Van hieruit kunnen alle aangekoppelde schijven gecombineerd worden. Nadat dit gedaan is, zien de schijven er voor FreeBSD als één enkele schijf uit. Andere RAID-niveaus kunnen overeenkomstig ingesteld worden.

17.4.3. ATA RAID1-rijen opnieuw bouwen

Met FreeBSD is het mogelijk om een defecte schijf in een rij te vervangen terwijl de computer aanstaat ("hot replace"). Hiervoor dient de schijf vóór het opnieuw opstarten vervangen te zijn.

Waarschijnlijk is zoiets als het volgende in /var/log/messages of in de uitvoer van dmesg(8) te zien:

```
ad6 on monster1 suffered a hard error.
ad6: READ command timeout tag=0 serv=0 - resetting
ad6: trying fallback to PIO mode
ata3: resetting devices .. done
ad6: hard error reading fsbn 1116119 of 0-7 (ad6 bn 1116119; cn 1107 tn 4 sn 11)\\
status=59 error=40
ar0: WARNING - mirror lost
```

Meer informatie kan met behulp van atacontrol(8) gezocht worden:

```
# atacontrol list
ATA channel 0:
    Master:      no device present
    Slave:   acd0 <HL-DT-ST CD-ROM GCR-8520B/1.00> ATA/ATAPI rev 0

ATA channel 1:
    Master:      no device present
    Slave:       no device present
```

```
ATA channel 2:
    Master:  ad4 <MAXTOR 6L080J4/A93.0500> ATA/ATAPI rev 5
    Slave:       no device present

ATA channel 3:
    Master:  ad6 <MAXTOR 6L080J4/A93.0500> ATA/ATAPI rev 5
    Slave:       no device present

# atacontrol status ar0
ar0: ATA RAID1 subdisks: ad4 ad6 status: DEGRADED
```

1. Ontkoppel eerst het ata kanaal met de falende schijf zodat deze veilig kan worden verwijderd:

   ```
   # atacontrol detach ata3
   ```

2. Vervang de schijf.
3. Koppel het ata kanaal opnieuw aan:

   ```
   # atacontrol attach ata3
   Master:  ad6 <MAXTOR 6L080J4/A93.0500> ATA/ATAPI rev 5
   Slave:   no device present
   ```

4. Voeg de nieuwe schijf toe aan de rij als reserve:

   ```
   # atacontrol addspare ar0 ad6
   ```

5. De rij dient nu opnieuw opgebouwd te worden:

   ```
   # atacontrol rebuild ar0
   ```

6. Het is mogelijk de voortgang te volgen met het volgende commando:

   ```
   # dmesg | tail -10
   [uitvoer verwijderd]
   ad6: removed from configuration
   ad6: deleted from ar0 disk1
   ad6: inserted into ar0 disk1 as spare

   # atacontrol status ar0
   ar0: ATA RAID1 subdisks: ad4 ad6 status: REBUILDING 0% completed
   ```

7. Nu moet er gewacht worden tot de bewerking voltooid is.

17.5. USB-opslagapparaten

Veel externe opslagoplossingen gebruiken tegenwoordig de Universele Seriële Bus (USB): harde schijven, USB-duimdrives, CD-R-branders, etc. FreeBSD biedt voor al dit soort apparaten ondersteuning.

17.5.1. Instellen

Het stuurprogramma umass(4) biedt de ondersteuning voor USB-opslagapparaten. Indien de kernel GENERIC wordt gebruikt, hoeft er niets aan de instellingen gewijzigd te worden. Als er een eigen kernel wordt gebruikt, dienen de volgende regels in het kernelinstellingenbestand aanwezig zijn:

```
device scbus
device da
device pass
device uhci
device ohci
device ehci
device usb
device umass
```

Het stuurprogramma umass(4) gebruikt het subsysteem SCSI om toegang te krijgen tot de USB-opslagapparaten. Het USB-apparaat wordt door het systeem als een SCSI-apparaat gezien. Afhankelijk van de chipset op het moederbord is slechts òf `device uhci` òf `device ohci` nodig voor ondersteuning van USB 1.X. Het kan echter geen kwaad om ze beiden in het kernelinstellingenbestand te hebben. Ondersteuning voor USB 2.0 wordt geleverd door het stuurprogramma ehci(4) (de regel met `device ehci`). Indien er regels zijn toegevoegd dient de kernel opnieuw gecompileerd en geïnstalleerd te worden.



Indien het USB-apparaat een CD-R- of DVD-brander is, dient het SCSI CD-ROM-stuurprogramma cd(4) met de volgende regel aan de kernel toegevoegd te worden:

```
device cd
```

Aangezien de brander als een SCSI-schijf gezien wordt, dient het stuurprogramma atapicam(4) niet in de kernelinstellingen gebruikt te worden.

17.5.2. Instellingen testen

De instellingen zijn klaar om getest te worden: het USB-apparaat dient aangesloten te worden en in de buffer voor systeemmeldingen (dmesg(8)) dient het stuurprogramma ongeveer als volgt te verschijnen:

```
umass0: USB Solid state disk, rev 1.10/1.00, addr 2
GEOM: create disk da0 dp=0xc2d74850
da0 at umass-sim0 bus 0 target 0 lun 0
```

```
da0: <Generic Traveling Disk 1.11> Removable Direct Access SCSI-2 device
da0: 1.000MB/s transfers
da0: 126MB (258048 512 byte sectors: 64H 32S/T 126C)
```

Uiteraard kunnen het merk, de apparaatnode (da0) en andere details verschillen naar gelang de instelling.

Aangezien het USB-apparaat als een SCSI-apparaat gezien wordt, kan het commando `camcontrol` gebruikt worden om de USB-opslagapparaten weer te geven die aan het systeem gekoppeld zijn:

```
# camcontrol devlist
<Generic Traveling Disk 1.11>      at scbus0 target 0 lun 0 (da0,pass0)
```

Indien er een bestandssysteem op de schijf aanwezig is, kan dat aangekoppeld worden. Schijven toevoegen biedt indien nodig hulp bij het formatteren en aanmaken van partities op de USB-drive.



Door het toestaan dat gewone gebruikers verschillende media kunnen koppelen door bijvoorbeeld het aanzetten van `vfs.usermount` zoals hieronder beschreven, zou niet als veilig beschoud moeten worden uit een beveiligings oogpunt. Veel bestandssystemen in FreeBSD zijn niet geschreven om beveiliging te bieden tegen kwaadaardige apparaten.

Om het apparaat koppelbaar te maken voor de gewone gebruiker moeten er een aantal stappen ondernomen worden. Als eerste moeten de apparaten die gecreeerd worden wanneer het USB opslag- medium wordt toegevoegd toegankelijk zijn voor de gebruiker. Een oplossing is om alle gebruikers die deze rechten nodig hebben toe te voegen aan de `operator` groep. Dit kan gedaan worden met pw(8). Daarna moet het voor de `operator` groep mogelijk zijn te lezen en te schrijven naar de gecreeerde apparaten. Dit kan bewerkstelligd worden door de volgende regels toe te voegen aan /etc/devfs.rules:

```
[localrules=5]
add path 'da*' mode 0660 group operator
```



Als er SCSI schijven in het systeem aanwezig zijn moet dit anders aangepakt worden. Stel dat het systeem reeds over de volgende schijven beschikt da0 tot en met da2, verander de regel dan in het volgende:

```
add path 'da[3-9]*' mode 0660 group operator
```

Dit sluit de reeds bestaande schijven buiten van toegang door de operator groep.

Erna moet ook de nieuwe ruleset voor devfs.rules(5) ingeschakeld worden door middel van /etc/rc.conf:

```
devfs_system_ruleset="localrules"
```

Hierna moet de kernel worden geconfigureerd zodat gewone gebruikers rechten krijgen om bestandssystemen te koppelen. De makkelijkste manier is door de volgende regel toe te voegen aan /etc/sysctl.conf:

```
vfs.usermount=1
```

Let op, deze wijziging wordt pas actief na de volgende start van het systeem. Als alternatief kan ook sysctl(8) gebruikt worden om deze variabele te zetten.

De laatste stap is het creëeren van de map waar het bestandssysteem gekoppeld wordt. Deze map moet eigendom zijn van de gebruiker die het bestandssysteem gaat koppelen. Een manier om dat te bewerkstelligen is door met de gebruiker `root` een submap aan te maken die eigendom is van de gebruiker als /mntgebruikersnaam (verander *gebruikersnaam* door de loginnaam van de daadwerkelijke gebruiker en *gebruikersgroep* door de primaire groep van de gebruiker):

```
# mkdir /mnt/gebruikersnaam
# chown gebruikersnaam:gebruikersgroep /mnt/gebruikersnaam
```

Stel dat er vervolgens een USB-stick ingeplugged wordt en er een /dev/da0s1 aangemaakt wordt. Omdat deze apparaten meestal voorgeformatteerd met een FAT-bestandssysteem komen, kan deze als volgende gekoppeld worden:

```
% mount -t msdosfs -o -m=644,-M=755 /dev/da0s1 /mnt/gebruikersnaam
```

Indien het apparaat losgekoppeld wordt (nadat de schijf afgekoppeld is), dient in de buffer voor systeemmeldingen iets als het volgende te zien te zijn:

```
umass0: at uhub0 port 1 (addr2) disconnected
(da0:umass-sim0:0:0:0): lost device
(da0:umass-sim0:0:0:0): removing device entry
GEOM: destroy disk da0 dp=0xc2d74850
umass0: detached
```

17.5.3. Referenties

Naast de onderdelen Schijven toevoegen en Bestandssystemen aan- en afkoppelen, kunnen de volgende hulppagina's ook nuttig zijn: umass(4), camcontrol(8) en usbconfig(8) voor FreeBSD 8.X of usbdevs(8) voor eerdere versies van FreeBSD.

17.6. Optische media (CD's) aanmaken en gebruiken

17.6.1. Inleiding

CD's hebben een aantal eigenschappen waardoor ze verschillen van conventionele schijven. Initieel zijn ze door de gebruiker niet beschrijfbaar. Ze zijn zó ontworpen dat ze continu, zonder vertragingen van het verplaatsen van de kop tussen tracks, gelezen kunnen worden. Ze zijn ook veel gemakkelijker tussen twee systemen te verplaatsen dan gelijksoortige media in hun tijd waren.

CD's hebben tracks, maar die verwijzen naar secties van gegevens die continu gelezen dienen te worden en niet naar fysieke eigenschappen van de schijf. Om een CD op FreeBSD te produceren, dienen de gegevensbestanden waaruit de tracks op de CD gaan bestaan te worden voorbereid, waarna de tracks op de CD worden geschreven.

Het bestandssysteem ISO 9660 is ontworpen om met deze verschillen om te gaan. Helaas codeert het bestandssysteemgrenzen die destijds gebruikelijk waren. Gelukkig biedt het een uitbreidingsmechanisme dat correct geschreven CD's toestaat om deze grenzen te overschrijden en nog steeds te werken met systemen die deze uitbreidingen niet ondersteunen.

De port sysutils/cdrtools bevat mkisofs(8), een programma dat gebruikt kan worden om een gegevensbestand aan te maken dat een ISO 9660-bestandssysteem bevat. Het bevat opties die verschillende uitbreidingen ondersteunen en wordt hieronder beschreven.

Het gereedschap om de CD te branden hangt af van het feit of de CD-brander ATAPI of iets anders is. ATAPI CD-branders gebruiken het programma `burncd` dat deel uitmaakt van het basissysteem. SCSI en USB CD-branders dienen `cdrecord` van de port sysutils/cdrtools te gebruiken. Het is ook mogelijk om `cdrecord` en andere gereedschappen voor SCSI-drives op ATAPI-hardware te gebruiken door middel van de module ATAPI/CAM.

Indien CD-brandsoftware met een grafische gebruikersinterface gewenst is, is X-CD-Roast of K3b een mogelijkheid. Deze gereedschappen zijn beschikbaar als package of vanuit de ports sysutils/xcdroast en sysutils/k3b. X-CD-Roast en K3b hebben de module ATAPI/CAM met ATAPI-hardware nodig.

17.6.2. mkisofs

Het programma mkisofs(8), dat deel uitmaakt van de port sysutils/cdrtools, maakt een ISO 9660-bestandssysteem aan dat een beeld is van een boomstructuur in de UNIX® bestandssysteem-namespace. De eenvoudigste gebruiksvorm is:

```
# mkisofs -o beeldbestand.iso /pad/naar/boomstructuur
```

Dit commando maakt een *beeldbestand.iso* aan dat een ISO 9660-bestandssysteem bevat dat een kopie is van de boomstructuur in */pad/naar/boomstructuur*. Tijdens het proces beeldt het bestandsnamen af op namen die aan de beperkingen van het standaard ISO 9660-bestandssysteem voldoen en sluit het bestanden uit die namen hebben die niet karakteristiek zijn voor ISO-bestandssystemen.

Er is een aantal opties beschikbaar om over deze beperkingen heen te komen. In het bijzonder zet `-R` de Rock Ridge-uitbreidingen aan die gangbaar zijn voor UNIX® systemen, zet `-J` de Rock Ridge-uitbreidingen aan die gebruikt worden op Microsoft-systemen en `-hfs` kan gebruikt worden om HFS-bestandssystemen aan te maken die door Mac OS® gebruikt worden.

Voor CD's die alleen op FreeBSD-systemen gebruikt gaan worden, kan `-U` gebruikt worden om alle restricties op bestandsnamen uit te zetten. Indien het met `-R` gebruikt wordt, maakt het een bestandssysteembeeld aan dat identiek is aan de FreeBSD-boomstructuur van waaruit begonnen is, alhoewel het mogelijk is dat het zich op aantal manieren niet aan de ISO 9660-standaard houdt.

De laatste optie voor algemeen gebruik is `-b`. Deze wordt gebruikt om de plaats van het opstartbeeld aan te geven om een "El Torito" opstartbare CD te maken. Deze optie heeft een argument nodig, namelijk het pad naar een opstartbeeld dat het begin van de boomstructuur die naar de CD geschreven wordt voorstelt. Gewoonlijk maakt mkisofs(8) een ISO-beeld aan in de zogenaamde "diskette-emulatie"-modus en verwacht het dus dat het beeldbestand exact 1200, 1440 of 2880 KB groot is. Sommige bootloaders, zoals degene die door de distributieschijven van FreeBSD wordt gebruikt, gebruiken de emulatiemodus niet. In dat geval dient de optie `-no-emul-boot` gebruikt te worden. Dus indien /tmp/myboot een opstartbaar FreeBSD-systeem met het beeldbestand in /tmp/myboot/boot/cdboot bevat, kan het beeld van een ISO 9660-bestandssysteem als volgt in /tmp/bootable.iso aangemaakt worden:

```
# mkisofs -R -no-emul-boot -b boot/cdboot -o /tmp/bootable.iso /tmp/myboot
```

Als dit gedaan is en md in de kernel is ingesteld, kan het bestandssysteem gekoppeld worden:

```
# mdconfig -a -t vnode -f /tmp/bootable.iso -u 0
# mount -t cd9660 /dev/md0 /mnt
```

Nu kan gecontroleerd worden of /mnt en /tmp/myboot identiek zijn.

Er zijn vele andere opties die met mkisofs(8) gebruikt kunnen worden om het gedrag af te stemmen. In het bijzonder wijzigingen aan een ISO 9660-structuur en het aanmaken van Joliet- en HFS-schijven. Details staan in mkisofs(8).

17.6.3. burncd

Indien er een ATAPI CD-brander aanwezig is, kan het commando `burncd` gebruikt worden om een ISO-beeld naar een CD te branden. `burncd` maakt deel uit van het basissysteem en is geïnstalleerd als /usr/sbin/burncd. Het gebruik is erg eenvoudig, aangezien het weinig opties heeft.

```
# burncd -f cd-apparaat gegevens beeldbestand.iso fixate
```

Het bovenstaande commando brandt een kopie van *beeldbestand.iso* naar *cd-apparaat*. Het standaardapparaat is /dev/acd0. Opties om de schrijfsnelheid in te stellen, de CD na het branden uit te werpen en geluidsgegevens te schrijven staan in burncd(8).

17.6.4. cdrecord

Indien er geen ATAPI CD-brander aanwezig is, dient `cdrecord` gebruikt te worden om CD's te branden. `cdrecord` maakt geen deel uit van het basissysteem. Het dient òfwel vanuit de port in sysutils/cdrtools òfwel als package geïnstalleerd te worden. Veranderingen in het basissysteem kunnen ervoor zorgen dat binaire versies van dit programma falen, wat mogelijk tot een "coaster" leidt. Daarom dient òfwel de port bijgewerkt te worden als het systeem wordt bijgewerkt, òwel, als -STABLE gevolgd wordt, dient de port bijgewerkt te worden wanneer er een nieuwe versie beschikbaar komt.

Hoewel `cdrecord` vele opties heeft, is het gebruik voor gewone situaties nog eenvoudiger dan dat van `burncd`. Een ISO 9660-beeld kan gebrand worden met:

```
# cdrecord dev=device beeldbestand.iso
```

Het lastige gedeelte in het gebruik van `cdrecord` is het vinden van de juiste `dev`. Om de juiste instelling te vinden, kan de vlag `-scanbus` van `cdrecord` gebruikt worden, wat resultaten zoals de onderstaande kan geven:

```
# cdrecord -scanbus
Cdrecord-Clone  2.01 (i386-unknown-freebsd7.0) Copyright (C) 1995-2004 Jörg Schilling
Using libscg version 'schily-0.1'
scsibus0:
        0,0,0     0) 'SEAGATE ' 'ST39236LW       ' '0004' Disk
        0,1,0     1) 'SEAGATE ' 'ST39173W        ' '5958' Disk
        0,2,0     2) *
        0,3,0     3) 'iomega  ' 'jaz 1GB         ' 'J.86' Removable Disk
        0,4,0     4) 'NEC     ' 'CD-ROM DRIVE:466' '1.26' Removable CD-ROM
        0,5,0     5) *
        0,6,0     6) *
        0,7,0     7) *
scsibus1:
        1,0,0   100) *
        1,1,0   101) *
        1,2,0   102) *
        1,3,0   103) *
        1,4,0   104) *
        1,5,0   105) 'YAMAHA  ' 'CRW4260        ' '1.0q' Removable CD-ROM
        1,6,0   106) 'ARTEC   ' 'AM12S          ' '1.06' Scanner
        1,7,0   107) *
```

Dit geeft de gepaste `dev`-waarden voor de apparaten in de lijst. De CD-brander dient gezocht te worden, waarna de drie getallen gescheiden door komma's gebruikt kunnen worden als de waarde voor `dev`. In dit geval is het CD-RW-apparaat 1,5,0, dus is de juiste invoer `dev=1,5,0`. Er zijn eenvoudigere manieren om deze waarde te specificeren. In cdrecord(1) staan meer details. Hier staat ook informatie over geluidstracks, de snelheid instellen en meer.

17.6.5. Audio-CD's dupliceren

Een audio-CD kan gedupliceerd worden door de geluidsgegevens van de CD naar een serie bestanden te schrijven en deze bestanden daarna naar een lege CD te schrijven. Het proces verschilt licht tussen ATAPI- en SCSI-drives.

Procedure: SCSI-drives

1. Onttrek `cdda2wav` de audio:

```
% cdda2wav -vall -D2,0 -B -Owav
```

2. Schrijf met `cdrecord` de .wav-bestanden:

```
% cdrecord -v dev=2,0 -dao -useinfo *.wav
```

   Controleer of *2,0* juist is opgegeven, zoals beschreven in cdrecord.

Procedure: ATAPI-drives



Met behulp van de ATAPI/CAM module kan `cdda2wav` ook gebruikt worden voor ATAPI-drives. Dit gereedschap is vaak een betere keuze voor de meeste gebruikers (jitter-correctie, endianness-zaken, etc.) dan de methode die hieronder wordt voorgesteld.

1. Het ATAPI CD-stuurprogramma maakt elke track beschikbaar als /dev/acddtnn, waarin *d* het stationsnummer is en *nn* het tracknummer is in twee decimale cijfers, dat indien nodig vooraf wordt gegaan door een nul. Dus is de eerste track op de eerste schijf /dev/acd0t01, de tweede /dev/acd0t02, de derde /dev/acd0t03, enzovoort.

   Controleer of de juiste bestanden in /dev bestaan. Als de benodigde namen er niet bijstaan, forceer het systeem dan om opnieuw te kijken:

```
# dd if=/dev/acd0 of=/dev/null count=1
```

2. De track kan met dd(1) onttrokken worden. Bij het onttrekken van de bestanden dient een specifieke blokgrootte gebruikt te worden.

```
# dd if=/dev/acd0t01 of=track1.cdr bs=2352
#dd if=/dev/acd0t02 of=track2.cdr bs=2352
...
```

3. Brand de onttrokken bestanden met `burncd`. Er dient opgegeven te worden dat het

geluidsbestanden zijn en dat burncd de schijf moet fixeren wanneer na afronding van het proces.

```
# burncd -f /dev/acd0 audio track1.cdr track2.cdr ... fixate
```

17.6.6. Gegevens-CD's dupliceren

Een gegevens-CD kan gekopieerd worden naar een beeldbestand dat functioneel gelijk is aan het beeldbestand dat met mkisofs(8) gemaakt is en het kan gebruikt worden om elke gegevens-CD te dupliceren. Het hier gegeven voorbeeld neemt aan dat het CD-ROM-apparaat acd0 is.

```
# dd if=/dev/acd0 of=bestand.iso bs=2048
```

Nu het beeld beschikbaar is, kan het naar CD geschreven worden zoals hierboven beschreven.

17.6.7. Gegevens-CD's gebruiken

Nu er een standaard gegevens-CD-ROM is aangemaakt moet deze waarschijnlijk aangekoppeld worden om de gegevens die er op staan te lezen. Normaalgesproken neemt mount(8) aan dat een bestandssysteem van het soort ufs is. Als zoiets als onderstaande geprobeerd wordt komt er een klacht over Incorrect super block en wordt er niet aangekoppeld:

```
# mount /dev/cd0 /mnt
```

De CD-ROM bevat geen UFS-bestandssysteem, dus pogingen om zo aan te koppelen mislukken. Er dient aan mount(8) verteld te worden dat het bestandssysteem van het soort ISO9660 is en dan werkt alles. Dit kan door de optie -t cd9660 van mount(8) op te geven. Het CD-ROM-apparaat /dev/cd0 onder /mnt aankoppelen kan zo:

```
# mount -t cd9660 /dev/cd0 /mnt
```

De apparaatnaam (in dit voorbeeld /dev/cd0) kan afwijken, afhankelijk van de interface die de CD-ROM gebruikt. Verder voert de optie -t cd9660 gewoon mount_cd9660(8) uit. Bovenstaand voorbeeld kan verkort worden tot:

```
# mount_cd9660 /dev/cd0 /mnt
```

Het is in het algemeen mogelijk om gegevens-CD-ROMs van elke fabrikant op deze manier te gebruiken. Schijven met bepaalde uitbreidingen op ISO 9660 kunnen zich echter vreemd gedragen. Joliet-schijven bijvoorbeeld, slaan alle bestandsnamen op in twee-byte Unicode-karakters. De FreeBSD-kernel spreekt geen Unicode, maar het FreeBSD CD9660 stuurprogramma is in staat om Unicode karakters direct te converteren. Als er niet-Engelse karakters verschijnen als vraagtekens, moet de lokale karakterset gedefinieerd worden met de -C optie. Zie de mount_cd9660(8)

handleiding voor meer informatie.



Om in staat te zijn om de karakter conversie te doen met behulp van de `-C` optie, heeft de kernel de cd9660_iconv.ko module nodig. Deze kan ingeladen worden door het volgende toe te voegen aan /boot/loader.conf:

```
cd9660_iconv_load="YES"
```

en daarna de machine te herstarten of door de module direct in te laden met kldload(8).

Zo nu en dan kan `Device not configured` verschijnen als geprobeerd wordt om een CD-ROM aan te koppelen. Dit betekent meestal dat het CD-ROM-station denkt dat er geen schijf in de lade ligt of dat het station niet zichtbaar is op de bus. Omdat het enkele seconden kan duren voordat een CD-ROM-station doorheeft dat er een CD-ROM in ligt, is geduld geboden.

Soms wordt een SCSI CD-ROM gemist omdat het station niet genoeg tijd had om antwoord te geven op de busreset. Indien er een SCSI CD-ROM aanwezig is, dient de volgende optie aan de kernelinstellingen toegevoegd te worden en de kernel opnieuw gebouwd te worden.

```
options SCSI_DELAY=15000
```

Dit zorgt ervoor dat de SCSI-bus 15 seconden pauzeert tijdens het opstarten opdat het CD-ROM-station elke gelegenheid krijgt om de busreset te beantwoorden.

17.6.8. Rauwe gegevens-CD's branden

Een bestand kan direct naar CD geschreven worden zonder een ISO 9660-bestandssysteem aan te maken. Sommige mensen doen dit voor back-updoeleinden. Dit gaat sneller dan een standaard-CD branden:

```
# burncd -f /dev/acd1 -s 12 gegevens archief.tar.gz fixate
```

Om de gegevens terug te halen die op zo'n CD gebrand zijn, is het noodzakelijk om gegevens van de rauwe apparaatnode te lezen:

```
# tar xzvf /dev/acd1
```

Het is niet mogelijk om deze schijf aan te koppelen zoals dat voor een normale CD-ROM gedaan wordt. Zo'n CD-ROM kan onder geen enkel besturingssysteem, behalve FreeBSD, gelezen worden. Om de CD aan te kunnen koppelen of gegevens te delen met een ander besturingssysteem, dient mkisofs(8) gebruikt te worden, zoals boven beschreven is.

17.6.9. Het ATAPI/CAM-stuurprogramma gebruiken

Dit stuurprogramma stelt ATAPI-apparaten (CD-ROM, CD-RW, DVD-stations, enzovoort) in staat om vanuit het SCSI-subsysteem benaderd te worden en maakt daarmee het gebruik van applicaties zoals sysutils/cdrdao of cdrecord(1) mogelijk.

Om dit stuurprogramma te gebruiken, is het noodzakelijk om de volgende regel aan het /boot/loader.conf bestand toe te voegen:

```
atapicam_load="YES"
```

om daarna de machine opnieuw op te starten.



Als het noodzakelijk is om atapicam(4) statisch toe te voegen aan de kernel moet de volgende regel worden toegevoegd aan het kernelinstellingenbestand:

```
device atapicam
```

Ook zijn de volgende regels in het kernelinstellingenbestand nodig, die meestal wel aanwezig zijn:

```
device ata
device scbus
device cd
device pass
```

Hierna dient de nieuwe kernel opnieuw gebouwd en geïnstalleerd te worden en dient de machine opnieuw gestart te worden.

Tijdens het opstartproces dient de brander als volgt te verschijnen:

```
acd0: CD-RW <MATSHITA CD-RW/DVD-ROM UJDA740> at ata1-master PIO4
cd0 at ata1 bus 0 target 0 lun 0
cd0: <MATSHITA CD-RW/DVD UJDA740 1.00> Removable CD-ROM SCSI-0 device
cd0: 16.000MB/s transfers
cd0: Attempt to query device size failed: NOT READY, Medium not present - tray closed
```

Het station is nu toegankelijk via de apparaatnaam /dev/cd0. Om bijvoorbeeld een CD-ROM op /mnt aan te koppelen:

```
# mount -t cd9660 /dev/cd0 /mnt
```

Als `root` kan het volgende commando gegeven worden om het SCSI-adres van de brander te verkrijgen:

```
# camcontrol devlist
<MATSHITA CD-RW/DVD UJDA740 1.00> at scbus1 target 0 lun 0 (pass0,cd0)
```

Dus `1,0,0` is het SCSI-adres dat met cdrecord(1) en andere SCSI-toepassingen gebruikt dient te worden.

Meer informatie over het ATAPI/CAM en het SCSI-systeem staat in de hulppagina's van atapicam(4) en cam(4).

17.7. Optische media (DVD's) aanmaken en gebruiken

17.7.1. Inleiding

Vergeleken met de CD behoort de DVD de tot de volgende generatie van optische media-opslagtechnologie. De DVD kan meer gegevens bevatten dan enige CD en is tegenwoordig de standaard voor videopublicatie.

Er kunnen vijf fysieke opneembare formaten gedefinieerd worden die opneembare DVD heten:

- DVD-R: dit was het eerst beschikbare opneembare DVD-formaat. De DVD-R-standaard is gedefinieerd door het DVD Forum. Dit formaat is voor eenmalig schrijven.
- DVD-RW: dit is de herschrijfbare versie van de DVD-R-standaard. Een DVD-RW kan tot ongeveer 1.000 maal herschreven worden.
- DVD-RAM: dit is ook een herschrijfbaar formaat dat door het DVD Forum ondersteund wordt. Een DVD-RAM kan gezien worden als een verwisselbare harde schijf. Dit medium is echter niet uitwisselbaar met de meeste DVD-ROM-stations en DVD-Video-spelers. Slechts enkele DVD-schrijvers ondersteunen het DVD-RAM-formaat. Lees DVD-RAM gebruiken voor meer informatie over het gebruik van DVD-RAM.
- DVD+RW: dit is het herschrijfbare formaat dat is gedefinieerd door de DVD+RW Alliance. Een DVD+RW kan tot ongeveer 1.000 maal herschreven worden.
- DVD+R: dit formaat is de eenmalig beschrijfbare versie van het DVD+RW-formaat.

Een enkellaags opneembare DVD kan maximaal 4.700.000.000 bytes bevatten, wat eigenlijk 4,38 GB of 4.485 MB is (1 kB is 1024 bytes).



Er dient onderscheid gemaakt te worden tussen het fysieke medium en de toepassing. Een DVD-Video bijvoorbeeld is een specifiek bestandsschema dat op elk fysiek opneembaar DVD-medium geschreven kan worden: DVD-R, DVD+R, DVD-RW, enzovoort. Voordat het mediumtype gekozen wordt, dient het zeker te zijn dat zowel de brander als de DVD-Video-speler (een onafhankelijke speler of een DVD-ROM-station in een computer) overweg kunnen met het overwogen medium.

17.7.2. Instellingen

Het programma growisofs(1) wordt gebruikt om DVD's op te nemen. Dit commando is deel van de dvd+rw-tools gereedschappen (sysutils/dvd+rw-tools). dvd+rw-tools ondersteunt alle types DVD-media.

Deze gereedschappen gebruiken het SCSI-subsysteem om toegang tot de apparaten te krijgen, daarvoor moet ondersteuning voor ATAPI/CAM aan de kernel toegevoegd worden. Indien de brander de USB-interface gebruikt, is deze toevoeging nutteloos en dient USB-opslagapparaten gelezen te worden voor meer details over het instellen van USB-apparaten.

De DMA-toegang voor ATAPI-apparaten dient ook aangezet te worden door de volgende regel aan het bestand /boot/loader.conf toe te voegen:

```
hw.ata.atapi_dma="1"
```

Voordat de dvd+rw-tools gebruikt kunnen worden, dienen de dvd+rw-tools' hardware compatibility notes geraadpleegd te worden voor enige informatie die betrekking heeft op de DVD-brander.



Indien een grafische gebruikersinterface gewenst is, is K3b (sysutils/k3b), die een gebruikersvriendelijke interface biedt voor growisofs(1) en vele andere brandprogramma's, het bekijken waard.

17.7.3. Gegevens-DVD's branden

Het commando growisofs(1) is een frontend voor mkisofs. Het roept mkisofs(8) aan om het bestandssysteemoverzicht aan te maken en het schrijft naar de DVD. Hierdoor is het niet nodig om een beeld van de gegevens aan te maken voordat met branden begonnen wordt.

Om de gegevens uit de map /pad/naar/gegevens op een DVD+R of een DVD-R te branden:

```
# growisofs -dvd-compat -Z /dev/cd0 -J -R /pad/naar/gegevens
```

De opties `-J -R` worden doorgegeven aan mkisofs(8) voor het aanmaken van het bestandssysteem (in dit geval een ISO 9660-bestandssysteem met Joliet en Rock Ridge uitbreidingen). Meer details staan in de hulppagina mkisofs(8).

De optie `-Z` wordt gebruikt voor het opnemen van de eerste sessie, ook bij meerdere sessies. Het DVD-apparaat, */dev/cd0*, dient aan de hand van de instellingen aangepast te worden. De parameter `-dvd-compat` sluit de schijf zodat er niets aan de opname toegevoegd kan worden. Dit zou als tegenprestatie betere uitwisselbaarheid met DVD-ROM-stations moeten geven.

Het is ook mogelijk om een vooraf gemastered beeld te branden, om bijvoorbeeld het beeld *beeldbestand.iso* te branden:

```
# growisofs -dvd-compat -Z /dev/cd0=beeldbestand.iso
```

De schrijfsnelheid moet automatisch gedetecteerd en ingesteld worden, afhankelijk van het medium en het gebruikte station. Om de schrijfsnelheid te forceren, dient de parameter `-speed=` gebruikt te worden. Meer informatie staat in de hulppagina growisofs(1).



Om bestanden groter dan 4,38GB in de compilatie op te nemen dient een UDF/ISO-9660 hybride bestandssysteem aangemaakt te worden door de aanvullende parameter `-udf -iso-level 3` aan mkisofs(8) en alle gerelateerde programma's (i.e., growisofs(1)) door te geven. Dit is alleen nodig als een ISO beeldbestand wordt aangemaakt, of als bestanden direct naar een schijf worden geschreven. Schijven die op deze manier zijn aangemaakt moeten als een UDF-bestandssysteem worden aangekoppeld met het hulpmiddel mount_udf(8), zodat het alleen bruikbaar is op een besturingssysteem dat zich van UDF bewust is, anders zal het lijken of er corrupte bestanden op staan.

Om zo'n ISO-bestand aan te maken:

```
% mkisofs -R -J -udf -iso-level 3 -o beeldbestand.iso
/pad/naar/gegevens
```

Om de bestanden direct naar een schijf te schrijven:

```
# growisofs -dvd-compat -udf -iso-level 3 -Z /dev/cd0 -J -R
/pad/naar/gegevens
```

Wanneer u een ISO-beeld heeft dat al grote bestanden bevat, zijn er geen extra opties nodig om met growisofs(1) het beeld naar een schijf te schrijven.

Zorg er ook voor dat u een actuele versie van sysutils/cdrtools heeft (welke mkisofs(8) bevat), aangezien oudere versies geen ondersteuning voor grote bestanden bieden. Als u problemen tegenkomt, gebruik dan de ontwikkelversie, sysutils/cdrtools-devel en lees de handleidingpagina mkisofs(8).

17.7.4. DVD-Video branden

Een DVD-Video is een specifiek bestandsschema dat gebaseerd is op de ISO 9660 en de micro-UDF (M-UDF) specificaties. DVD-Video heeft ook een specifieke hiërarchie voor de gegevensstructuur, de reden waarom een speciaal programma zoals multimedia/dvdauthor nodig is om de DVD te schrijven.

Indien er reeds een beeld van het bestandssysteem van de DVD-Video beschikbaar is, kan het zoals elk ander beeld gebrand worden. In de vorige sectie staat een voorbeeld. Als het resultaat voor de inhoud voor de DVD bijvoorbeeld in de map /pad/naar/video staat, kan de DVD-Video als volgt gebrand worden:

```
# growisofs -Z /dev/cd0 -dvd-video /pad/naar/video
```

De optie `-dvd-video` wordt doorgegeven aan mkisofs(8) en geeft het opdracht om een bestandssysteemschema voor een DVD-Video aan te maken. Verder impliceert de optie `-dvd-video` de optie `-dvd-compat` van growisofs(1).

17.7.5. DVD+RW gebruiken

In tegenstelling tot een CD-RW dient een nieuwe DVD+RW voor het eerste gebruik geformatteerd te worden. Het programma growisofs(1) regelt dit automatisch als nodig. Dit is de *aanbevolen* manier. Het is ook mogelijk om `dvd+rw-format` te gebruiken om een DVD+RW te formatteren:

```
# dvd+rw-format /dev/cd0
```

Deze operatie hoeft slechts één maal uitgevoerd te worden. Onthoud dat alleen nieuwe DVD+RW-media geformatteerd dienen te worden. Daarna is het mogelijk om de DVD+RW op dezelfde manier te branden zoals in bovenstaande secties staat vermeldt.

Om nieuwe gegevens op een DVD+RW te branden (een geheel nieuw bestandssysteem branden, niet wat gegevens toevoegen), is het niet nodig om deze te wissen. Het is voldoende om de vorige opname te overschrijven (tijdens het aanmaken van een initiële sessie), zoals hieronder:

```
# growisofs -Z /dev/cd0 -J -R /pad/naar/nieuwe gegevens
```

Het DVD+RW-formaat biedt de mogelijkheid om eenvoudig nieuwe gegevens aan een vorige opname toe te voegen. De operatie bestaat uit het samenvoegen van een nieuwe sessie en de bestaande. Het is geen multisessie-schrijven. growisofs(1) laat het ISO 9660-bestandssysteem dat aanwezig is op het medium *groeien*.

Om gegevens aan de vorige DVD+RW toe te voegen:

```
# growisofs -M /dev/cd0 -J -R /pad/naar/volgende gegevens
```

Dezelfde opties van mkisofs(8) die gebruikt werden om de initiële sessie te branden, dienen gebruikt te worden tijdens schrijfsessies.



De optie `-dvd-compat` kan gebruikt worden als betere uitwisselbaarheid met DVD-ROM-stations gewenst is. In het geval van een DVD+RW verhindert dit het toevoegen van gegevens niet.

Om het medium te wissen:

```
# growisofs -Z /dev/cd0=/dev/zero
```

17.7.6. DVD-RW gebruiken

Een DVD-RW accepteert twee schijfformaten: de incrementele sequentiële en beperkt overschrijven. Standaard zijn DVD-RW-schijven in het sequentiële formaat.

Een nieuwe DVD-RW kan direct beschreven worden zonder deze te formatteren. Een gebruikte DVD-RW in sequentieel formaat dient echter gewist te worden voordat het mogelijk is om een nieuwe initiële sessie te schrijven.

Om een DVD-RW in sequentiële toestand te wissen, dient het volgende gedaan te worden:

```
# dvd+rw-format -blank=full /dev/cd0
```



Volledig wissen (`-blank=full`) neemt ongeveer één uur in beslag op een 1x-medium. Het is mogelijk om snel te wissen door gebruik te maken van de optie `blank` als de DVD-RW in Disk-At-Once-modus (DAO) wordt opgenomen. Om de DVD-RW in DAO-modus te branden:

```
# growisofs -use-the-force-luke=dao -Z /dev/cd0=beeldbestand.iso
```

De optie `-use-the-force-luke=dao` is niet nodig aangezien growisofs(1) probeert om minimale (snel gewiste) media te detecteren en gebruik te maken van DAO-schrijven.

Eigenlijk moet beperkt overschrijven gebruikt worden met elke DVD-RW. Dit formaat is flexibeler dan het standaard incrementeel sequentiële.

Om gegevens op een sequentiële DVD-RW te schrijven, worden dezelfde instructies gebruikt als voor de andere DVD-formaten:

```
# growisofs -Z /dev/cd0 -J -R /pad/naar/gegevens
```

Om wat gegevens aan de vorige opname toe te voegen, dient de optie `-M` van growisofs(1) gebruikt te worden. Als echter gegevens aan een DVD-RW in incrementeel sequentiële modus worden toegevoegd, wordt een nieuwe sessie op de schijf aangemaakt wat resulteert in een multisessie schijf.

Een DVD-RW in het beperkt overschrijven formaat hoeft niet gewist te worden vóór een nieuwe initiële sessie. Het is voldoende om de schijf te overschrijven met de optie `-Z`, wat analoog is aan het geval van de DVD+RW. Het is ook mogelijk om een bestaand ISO 9660-bestandssysteem te laten groeien op soortgelijke wijze als voor een DVD+RW met de optie `-M`. Het resultaat is een enkelsessie DVD.

Om een DVD-RW in het beperkt overschrijven-formaat te zetten:

```
# dvd+rw-format /dev/cd0
```

Om terug te gaan naar het sequentiële formaat:

```
# dvd+rw-format -blank=full /dev/cd0
```

17.7.7. Multisessie

Multisessie DVD's worden door zeer weinig DVD-ROM-stations geaccepteerd en meestal lezen ze hopelijk tenminste de eerste sessie. DVD+R, DVD-R en DVD-RW kunnen in het sequentiële formaat meerdere sessies accepteren. Het idee van meerdere sessies bestaat niet voor de formaten DVD+RW en DVD-RW in beperkt overschrijven.

Om een nieuwe sessie achter een initiële (niet-gesloten) sessie op een DVD+R, DVD-R of DVD-RW in sequentieel formaat toe te voegen:

```
# growisofs -M /dev/cd0 -J -R /pad/naar/volgende gegevens
```

Het gebruik van dit commando met een DVD+RW of een DVD-RW in beperkt overschrijven-formaat voegt gegevens toe door de nieuwe sessie samen te voegen met de bestaande. Dit leidt tot een enkelsessie schijf. Deze manier kan gebruikt worden om gegevens achter een initiële sessie aan deze media toe te voegen.



Op deze media wordt wat ruimte gebruikt tussen elke sessie om het einde en begin van de sessies aan te geven. Daarom dienen sessies met grote hoeveelheden gegevens toegevoegd te worden om de mediaruimte te optimaliseren. Het aantal sessies is beperkt tot 154 voor een DVD+R, ongeveer 2000 voor een DVD-R en 127 voor een dubbellaags DVD+R.

17.7.8. Meer informatie

Om meer informatie over een DVD te verkrijgen kan het commando `dvd+rw-mediainfo /dev/cd0` met de schijf in het station gebruikt worden.

Meer informatie over dvd+rw-tools staat in de hulppagina growisofs(1), op de dvd+rw-tools website en in de archieven van de cdwrite mailing list.



De uitvoer van `dvd+rw-mediainfo` met betrekking tot de resulterende opname of het medium met problemen is verplicht voor elk probleemrapport. Zonder deze uitvoer volgt geen hulp.

17.7.9. DVD-RAM gebruiken

Configuratie

DVD-RAM schrijvers komen met of een SCSI of een ATAPI interface. DMA toegang voor ATAPI apparaten moet worden ingeschakeld, wat gedaan kan worden door de volgende regel toe te voegen aan /boot/loader.conf:

```
hw.ata.atapi_dma="1"
```

Voorbereiden van het medium

Zoals vermeld in de introductie van dit hoofdstuk kan DVD-RAM gezien worden als een verwijderbare harde schijf. Zoals elke andere harde schijf moet de DVD-RAM "voorbereid" worden voor het eerste gebruik. In het voorbeeld wordt alle beschikbare ruimte gebruikt voor een standaard UFS2 bestandssysteem:

```
# dd if=/dev/zero of=/dev/acd0 bs=2k count=1
# bsdlabel -Bw acd0
# newfs /dev/acd0
```

Het DVD apparaat, acd0 moet worden gewijzigd naar gelang de configuratie.

Het medium gebruiken

Zodra de voorgaande operaties uitgevoerd zijn op de DVD-RAM kan het gekoppeld worden net als een normale harde schijf:

```
# mount /dev/acd0 /mnt
```

Hierna zal de DVD-RAM zowel lees- als beschrijfbaar zijn.

17.8. Diskettes aanmaken en gebruiken

Soms is het opslaan van gegevens op een diskette nuttig, bijvoorbeeld als er geen andere verwijderbare opslagmedia beschikbaar zijn of als kleine hoeveelheden gegevens naar een andere computer moeten worden overgedragen.

In deze sectie wordt beschreven hoe diskettes in FreeBSD gebruikt dienen te worden. Hier worden hoofdzakelijk het formatteren en gebruik van 3,5 inch DOS-diskettes behandeld, maar de concepten zijn vergelijkbaar voor andere disketteformaten.

17.8.1. Diskettes formatteren

Het apparaat

Diskettes worden benaderd door ingangen in /dev net zoals andere apparaten. Om een rauwe floppy te benaderen gebruikt u /dev/fdN.

Formatteren

Een diskette dient op laag niveau geformatteerd te worden voordat deze kan worden gebruikt. Dit wordt meestal door de fabrikant gedaan, maar formatteren is een goede manier om de integriteit van het medium te controleren. Hoewel het mogelijk is om grotere (of kleinere) schijfgroottes te forceren, zijn de meeste diskettes ontworpen voor 1440kB.

Een diskette kan op laag niveau geformatteerd worden met fdformat(1). Dit gereedschap verwacht de apparaatnaam als parameter.

Op basis van eventuele foutmeldingen kan bepaald worden of een schijf goed of slecht is.

Formatteren van floppies

Voor het formatteren van de diskette dienen de apparaten /dev/fdN gebruikt te worden. Nadat een 3,5 inch diskette in het station is gestoken:

```
# /usr/sbin/fdformat -f 1440 /dev/fd0
```

17.8.2. Schijflabels

Nadat de diskette op laag niveau is geformatteerd, dient er schijflabel aan gekoppeld te worden. Dit schijflabel wordt later vernietigd, maar het systeem heeft het nodig om later de grootte en de geometrie van de schijf te bepalen.

Het nieuwe schijflabel neemt de gehele schijf over en bevat alle benodigde informatie over de geometrie van de diskette. De geometriewaarden van het schijflabel staan vermeld in /etc/disktab.

Nu kan bsdlabel(8) als volgt gedraaid worden:

```
# /sbin/bsdlabel -B -w /dev/fd0 fd1440
```

17.8.3. Bestandssystemen

Nu is de diskette klaar om op hoog niveau geformatteerd te worden. Hiermee wordt een nieuw bestandssysteem opgezet, wat FreeBSD in staat stelt om naar de schijf te lezen en te schrijven. Nadat het nieuwe bestandssysteem is aangemaakt, wordt het schijflabel vernietigd, dus om de schijf te herformatteren is het noodzakelijk om het schijflabel opnieuw aan te maken.

Het bestandssysteem voor diskettes kan zowel UFS als FAT zijn. FAT is over het algemeen een betere keuze voor diskettes.

Om een nieuw bestandssysteem op de diskettes te zetten:

```
# /sbin/newfs_msdos /dev/fd0
```

De schijf is nu klaar voor gebruik.

17.8.4. Diskettes gebruiken

Om de diskette te gebruiken kan mount_msdosfs(8) gebruikt worden om het medium aan te koppelen. Ook kan emulators/mtools uit de Portscollectie worden gebruikt.

17.9. Gegevensbanden aanmaken en gebruiken

Bandtechnologie is zich blijven ontwikkelen maar het is minder waarschijnlijk dat het in moderne systemen wordt gebruikt. Moderne back-upsystemen neigen om offsite gecombineerd met technologieën voor plaatselijke verwisselbare schijfstations te gebruiken. FreeBSD zal nog steeds elk bandstation dat SCSI gebruikt zoals LTO en oudere apparaten zoals DAT ondersteunen. Er is ook beperkte ondersteuning voor SATA- en USB-bandstations.

17.9.1. Seriële toegang met sa(4)

FreeBSD gebruikt het stuurprogramma sa(4), dat /dev/sa0, /dev/nsa0 en /dev/esa0 aanbiedt. Voor normaal gebruik is alleen /dev/sa0 nodig. /dev/nsa0 is fysiek hetzelfde apparaat als /dev/sa0 maar spoelt de band niet terug nadat een bestand is geschreven. Dit maakt het mogelijk om meer dan één bestand naar een band te schrijven. /dev/esa0 werpt, indien van toepassing, de band uit nadat het apparaat is gesloten.

17.9.2. Het bandstation met mt(1) beheren

mt(1) is het hulpmiddel van FreeBSD om andere bewerkingen op het bandstation uit te voeren, zoals bestanden op een band doorzoeken of controlepunten naar de band schrijven.

Als voorbeeld kunnen de eerste drie bestanden op een band bewaard worden door ze over te slaan voordat een nieuw bestand wordt geschreven:

```
# mt -f /dev/nsa0 fsf 3
```

17.9.3. tar(1) gebruiken om back-ups op banden te lezen en schrijven

tar(1) gebruiken om een enkel bestand naar band te schrijven:

```
# tar cvf /dev/sa0 bestand
```

Bestanden vanuit een tar(1)-archief op band naar de huidige map herstellen:

```
# tar xvf /dev/sa0
```

17.9.4. dump(8) en restore(8) gebruiken om back-ups aan te maken en te herstellen

Een eenvoudige back-up van /usr maken met dump(8):

```
# dump -0aL -b64 -f /dev/nsa0 /usr
```

Interactief bestanden van een dump(8)-bestand vanaf band naar de huidige map herstellen:

```
# restore -i -f /dev/nsa0
```

17.9.5. Andere bandsoftware

Er zijn programma's op hoger niveau beschikbaar om het back-uppen naar banden eenvoudiger te maken. De populairste zijn AMANDA en Bacula. Deze programma's hebben als doel om back-uppen eenvoudiger en aangenamer te maken, of om complexe back-ups van meerdere machines te automatiseren. De Portscollectie bevat deze beide en andere toepassingen om met banden te werken.

17.10. Naar diskettes back-uppen

17.10.1. Kunnen diskettes gebruikt worden om gegevens te back-uppen?

Diskettes zijn niet bepaald een geschikt medium om back-ups mee te maken, omdat:

- Het medium onbetrouwbaar is, in het bijzonder op de langere termijn;
- Het back-uppen en terugzetten erg traag is;
- Diskettes een zeer beperkte capaciteit hebben. De tijden dat een hele harde schijf naar een tiental diskettes kon worden geback-upped zijn allang verstreken.

Maar als er geen andere manier beschikbaar is om de gegevens te back-uppen, is een back-up naar diskettes beter dan helemaal geen back-up.

Gebruikte diskettes moet van goede kwaliteit zijn. Diskettes die al jaren op kantoor rondgeslingerd hebben, zijn een slechte keuze. In het ideale geval dienen nieuwe diskettes van een reputabele fabrikant gebruikt te worden.

17.10.2. Hoe de gegevens naar diskettes back-uppen?

Het beste kan naar diskettes worden geback-upped door gebruik te maken van tar(1) met de optie `-M` (meerdere volumes), die back-ups over meerdere diskettes ondersteunt.

Om alle bestanden in de huidige map en de submappen te back-uppen (als `root`):

```
# tar Mcvf /dev/fd0 *
```

Als de eerste diskette vol is, vraagt tar(1) om het volgende volume. Omdat tar(1) media-onafhankelijk is, refereert het aan volumes, in deze context diskettes.

```
Prepare volume #2 for /dev/fd0 and hit return:
```

Dit wordt herhaald (met oplopend volumenummer) totdat alle gespecificeerde bestanden zijn geback-upped.

17.10.3. Kunnen back-ups gecomprimeerd worden?

Helaas staat tar(1) het gebruik van de optie `-z` niet toe voor archieven over meerdere volumes. Het is uiteraard mogelijk om alle bestanden met gzip(1) te comprimeren, ze met tar(1) op diskettes te zetten en ze daarna met gunzip(1) weer te decomprimeren!

17.10.4. Hoe worden de back-ups teruggezet?

Om een volledige archief terug te zetten:

```
# tar Mxvf /dev/fd0
```

Er zijn twee manieren om alleen specifieke bestanden terug te zetten. Ten eerste kan met de eerste diskette begonnen worden:

```
# tar Mxvf /dev/fd0 bestandsnaam
```

Het programma tar(1) vraagt om de vervolgdiskettes totdat het benodigde bestand is gevonden.

Als alternatief kan, als bekend is op welke diskette het bestand staat, de betreffende diskette worden ingestoken en bovenstaand commando gebruikt worden. Als het eerste bestand op de diskette een vervolg is van de vorige diskette, waarschuwt tar(1) dat het bestand niet teruggezet kan worden, zelfs als hier niet om gevraagd is!

17.11. Back-up strategieën

Het eerste wat nodig is voor het ontwepken van een back-upplan, is er voor te zorgen dat de volgende mogelijke problemen worden ondervangen:

- Schijffalen
- Per ongeluk verwijderde bestanden
- Willekeurige bestandscorruptie
- Complete machinevernietiging (door bijvoorbeeld brand), inclusief de vernietiging van lokaal beschikbare back-ups.

Het is goed mogelijk dat een aantal systemen het best geholpen zijn door voor al deze problemen een andere techniek te gebruiken. Behalve voor volledig persoonlijke systemen met niet echt belangrijke gegevens, is het zelfs onwaarschijnlijk dat één techniek alle mogelijke problemen kan afvangen.

Een aantal technieken in de gereedschapskist zijn:

- Archiveren van een heel systeem op een back-up die niet lokaal wordt bewaard. Dit biedt bescherming tegen alle hierboven beschreven problemen, maar het is langzaam en onhandig om er een restore van te maken. Het is mogelijk om lokaal een kopie aan te houden en/of online, maar dan zijn er nog steeds onhandigheden, in het bijzonder voor restores voor gebruikers met beperkte rechten.
- Snapshots van bestandssystemen. Dit werkt eigenlijk alleen in het geval bestanden per ongelijk verwijderd worden, maar het kan in dat geval *erg* handig zijn en het werkt snel en eenvoudig.
- Een kopie maken van hele bestandssystemen en/of schijven (bijvoorbeeld een periodieke rsync(1) van een hele machine). Dit is in het algemeen het meest bruikbaar in netwerken met specifieke eisen. Voor algemene bescherming tegen het falen van een schijf, is het meestal minder geschikt dan RAID. Voor het herstellen van per ongeluk verwijderde bestanden is het vergelijkbaar aan UFS snapshots, maar dat hangt af van persoonlijke voorkeuren.
- RAID. Minimaliseert of voorkomt downtijd als een schijf faalt. Dit ten koste van het vaker hebben van schijven die falen (omdat er meer van zijn), maar wel met een veel lagere urgentie.
- Controleren van fingerprints van bestanden. Het hulpprogramma mtree(8) kan hier bij helpen. Hoewel dit geen back-uptechniek is, zorgt het er wel voor dat kan worden opgemerkt wanneer back-ups geraadpleegd moeten worden. Dit is in het bijzonder belangrijk voor offline back-ups en de fingerprints horen periodiek gecontroleerd te worden.

Het is makkelijk om met nog meer technieken op de proppen te komen, waaronder veel variaties op de bovengenoemde. Bijzondere eisen leiden vaak tot bijzondere oplossingen. Het back-uppen van een draaiende database vereist bijvoorbeeld een methode die toegespitst is op de gebruikte database software als tussenstap. Het is van groot belang om te onderkennen tegen welke gevaren er bescherming dient te zijn en hoe daarmee om te gaan.

17.12. Back-upbeginselen

De drie grote back-upprogramma's zijn dump(8), tar(1) en cpio(1).

17.12.1. Dump en Restore

De traditionele back-upprogramma's voor UNIX® zijn `dump` en `restore`. Deze zien het station als een verzameling van schijfblokken, onder de abstracties van bestanden, koppelingen en mappen die door de bestandssystemen worden aangemaakt. In tegenstelling tot andere back-upprogramma's, verzorgt `dump` een back-up van een compleet bestandssysteem op een apparaat. Het is niet in staat om slechts een gedeelte van een bestandssysteem of een mapstructuur die meer dan één bestandssysteem in beslag neemt te back-uppen. Het commando `dump` schrijft geen bestanden en mappen naar band, maar de rauwe gegevensblokken waaruit de bestanden en mappen bestaan. Wanneer het gebruikt wordt om gegevens te extraheren, slaat `restore` tijdelijke bestanden standaard op in /tmp/ - als u werkt vanaf een herstelschijf met een kleine map /tmp, moet u wellicht de omgevingsvariabele `TMPDIR` op een map met meer vrije ruimte instellen zodat de restore kan slagen.



Indien `dump` op een hoofdmap wordt gebruikt, wordt er geen back-up gemaakt van

> /home, /usr of van de vele andere mappen, aangezien dit typisch koppelpunten voor andere bestandssystemen of symbolische koppelingen binnen deze bestandssystemen zijn.

dump bevat eigenaardigheden die uit de begintijd in Versie 6 van AT&T UNIX® (circa 1975) zijn overgebleven. De standaardparameters zijn geschikt voor banden met 9 sporen (6.250 bpi), niet voor de media met hoge dichtheid die vandaag beschikbaar zijn (tot 62.182 ftpi). Deze standaardwaarden dienen op de opdrachtregel overschreven te worden om de capaciteit van de huidige bandstations te benutten.

Het is ook mogelijk om gegevens met rdump en rrestore over een netwerk naar een bandstation dat aan een andere computer gekoppeld is te back-uppen. Beide programma's maken gebruik van rcmd(3) en ruserok(3) om toegang tot het bandstation op afstand te krijgen. De gebruiker die de back-up uitvoert moet vermeld staat in het bestand .rhosts op de computer op afstand. De argumenten die aan rdump en rrestore gegeven worden dienen geschikt te zijn voor gebruik op de computer op afstand. Als rdump gebruikt wordt om een dump te maken van een FreeBSD computer naar een Exabyte-bandstation dat met een Sun-computer genaamd komodo verbonden is:

```
# /sbin/rdump 0dsbfu 54000 13000 126 komodo:/dev/nsa8 /dev/da0a 2>&1
```

Let op: er kleven veiligheidsbezwaren aan het toestaan van authenticatie met .rhosts. De situatie dient goed geëvalueerd te worden.

Het is ook mogelijk om dump en restore op een veiligere manier via ssh te gebruiken.

Voorbeeld 9. Het gebruik van dump via ssh

```
# /sbin/dump -0uan -f - /usr | gzip -2 | ssh -c blowfish \
      doelgebruiker@doelmachine.example.com dd of=/mijngrotebestanden/dump-usr-
10.gz
```

Ook kan de ingebouwde manier van dump gebruikt worden, door de omgevingsvariabele RSH in te stellen:

Voorbeeld 10. Het gebruik van dump via ssh met ingestelde RSH

```
# env RSH=/usr/bin/ssh /sbin/dump -0uan -f
doelgebruiker@doelmachine.example.com:/dev/sa0 /usr
```

17.12.2. tar

tar(1) stamt ook uit de tijd van Versie 6 van AT&T UNIX® (circa 1975). Het werkt samen met het bestandssysteem. tar schrijft bestanden en mappen naar band en ondersteunt niet het volledige scala aan opties dat beschikbaar is met cpio(1), maar tar heeft niet de ongebruikelijke

opdrachtpijplijn nodig die `cpio` gebruikt.

Om `tar` toe te passen op een Exabyte-bandstation die met een Sun genaamd `komodo` verbonden is:

```
# tar cf - . | rsh komodo dd of=tape-device obs=20b
```

Indien de veiligheid van back-uppen over een netwerk een punt is, dient gebruik te worden gemaakt van het commando `ssh` en niet van `rsh`.

17.12.3. `cpio`

cpio(1) is het originele UNIX® bandprogramma voor magnetische media om bestanden uit te wisselen. `cpio` heeft opties (naast vele anderen) om byte-swapping uit te voeren, een aantal verschillende archiefformaten te schrijven en de gegevens over een pijplijn naar andere programma's te voeren. Deze laatste optie maakt `cpio` een uitstekende keuze voor installatiemedia. `cpio` weet niet hoe het door een mapstructuur moet lopen. Er dient een lijst met bestanden door stdin aangeleverd te worden.

`cpio` biedt geen ondersteuning voor back-ups over het netwerk. Er kan gebruik worden gemaakt van een pijplijn en `rsh` om de gegevens naar een banddrive op afstand te sturen.

```
# for f in maplijst;  do
find $f >> back-up.lijst
done
# cpio -v -o --format=newc < back-up.lijst | ssh gebruiker@host "cat > back-
upapparaat"
```

Hier is *maplijst* een lijst van de mappen waarvan een back-up gemaakt dient te worden, *gebruiker@host* de gebruiker/hostnaam-combinatie die de back-ups uitvoert, en *back-upapparaat* het apparaat waar de back-ups naar toe geschreven te worden (bijvoorbeeld /dev/nsa0).

17.12.4. `pax`

pax(1) is het antwoord van IEEE en POSIX® op `tar` en `cpio`. In de loop der jaren zijn de verscheidene versies van `tar` en `cpio` licht incompatibel geworden. Dus in plaats van dit uit te vechten en ze volledig te standaardiseren, heeft POSIX® een nieuw archiveringsprogramma gemaakt. `pax` poogt om veel van de verscheidene formaten van `cpio` en `tar` te lezen en te schrijven, met daarbij nog nieuwe, eigen formaten. De commandoverzameling lijkt meer op die van `cpio` dan op die van `tar`.

17.12.5. Amanda

Amanda (Advanced Maryland Network Disk Archiver) is een client/server-back-upsysteem, in plaats van een enkel programma. Een Amanda server back-upt elk aantal computers dat een Amanda client en een netwerkverbinding met de Amanda server heeft naar een enkel bandstation. Een veelvoorkomend probleem bij bedrijven met een groot aantal schijven is dat de tijd die nodig is om de gegevens direct naar band te back-uppen langer is dan de tijd die voor de taak gereserveerd

is. Amanda lost dit probleem op. Amanda kan gebruik maken van een "tussenschijf" om verschillende bestandssystemen tegelijkertijd te back-uppen. Amanda maakt "archiefverzamelingen" aan, een groep banden die gedurende een tijd gebruikt wordt om volledige back-ups te maken van alle bestandssystemen die in het instellingenbestand van Amanda vermeld staan. De "archiefverzameling" bevat ook incrementele (of differentiële) back-ups van alle bestandssystemen. Voor het herstellen van een beschadigd bestandssysteem zijn de meest recente volledige back-up en de incrementele back-ups nodig.

Het instellingenbestand biedt verfijnde controle over de back-ups en het netwerkverkeer door Amanda. Amanda kan elk bovenstaand back-upprogramma gebruiken om de gegevens naar de band te schijven. Amanda is òf als port òf als package beschikbaar.

17.12.6. Nietsdoen

"Nietsdoen" is geen computerprogramma, maar de de meest gebruikte back-upstrategie. Er zijn geen initiële kosten. Er is geen back-upschema om te volgen. Zeg gewoon nee. Als er iets met gegevens gebeurt, lach erom en leef ermee!

Als tijd en gegevens weinig tot niets waard zijn, is "Nietsdoen" het meest geschikte back-upprogramma. Maar wees bedacht, UNIX® is een nuttig stuk gereedschap en er is zo maar binnen zes maanden een verzameling bestanden die wèl van waarde is.

"Nietsdoen" is de juiste back-upmethode voor /usr/obj en andere mapstructuren die zo opnieuw aangemaakt kunnen worden. Een voorbeeld zijn de bestanden waaruit de HTML- of PostScript® versie van dit Handboek bestaan. Deze documentformaten zijn vanuit XML-invoerbestanden aangemaakt. Het back-uppen van de HTML- of PostScript® bestanden is niet nodig. Van de XML-bestanden dient regelmatig een back-up gemaakt te worden.

17.12.7. Welk back-upprogramma is het beste?

dump(8). *Punt uit.*. Elizabeth D. Zwicky heeft stresstesten op alle hierboven besproken back-upprogramma's uitgevoerd. De heldere keuze voor het behouden van alle gegevens en alle eigenaardigheden van UNIX® bestandssystemen is `dump`. Elizabeth heeft bestandssystemen aangemaakt met een grote verscheidenheid aan ongewone omstandigheden (en enkele minder ongebruikelijke) en heeft elk programma getest door een back-up van die bestandssystemen uit te voeren en ze te herstellen. De eigenaardigheden omvatten bestanden met gaten, bestanden met gaten en een blok nullen, bestanden met vreemde tekens in hun namen, onleesbare en onschrijfbare bestanden, apparaten, bestanden waarvan de grootte verandert tijdens het back-uppen, bestanden die aangemaakt/verwijderd worden tijdens het back-uppen en meer. Ze presenteerde de resultaten op LISA V in oktober 1991. Zie torture-testing Backup and Archive Programs.

17.12.8. Noodterugzetprocedure

Vóór de ramp

Er zijn slechts vier stappen om te volgen bij het voorbereiden op elke ramp die voor kan komen.

Het schijflabel van elke schijf dient afgedrukt te worden (bijvoorbeeld met `bsdlabel da0 | lpr`), de

bestandssysteemtabel (/etc/fstab) en alle opstartboodschappen, alles in tweevoud.

Ten tweede dient een "livefs" CD-ROM gebrandt te worden. Deze CD-ROM biedt ondersteuning voor het opstarten in een FreeBSD "livefs" reddingsmodus die gebruiker in staat stelt om vele taken uit te voeren zoals het draaien van dump(8), restore(8), fdisk(8), bsdlabel(8), newfs(8) en meer. Een livefs CD-beeld voor FreeBSD/i386 11.2-RELEASE is beschikbaar op ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/ISO-IMAGES/11.2/FreeBSD-11.2-RELEASE-i386-livefs.iso.



Livefs CD-beelden zijn niet beschikbaar voor FreeBSD 12.0-RELEASE en nieuwer. Naast de beelden voor CDROM-installaties kunnen ook beelden voor flash-drive-installaties gebruikt worden om een systeem te redden. Het "memstick"-beelden voor FreeBSD/i386 12.0-RELEASE is beschikbaar op ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/i386/ISO-IMAGES/12.0/FreeBSD-12.0-RELEASE-i386-memstick.img.

Ten derde dienen regelmatig back-upbanden aangemaakt te worden. Alle veranderingen die na de laatste back-up zijn gemaakt kunnen onherroepelijk verloren zijn gegaan. De back-upbanden dienen beveiligd te worden tegen overschrijven.

Ten vierde dienen de "livefs" CD-ROM die in stap twee gemaakt is en de back-upbanden getest te worden. Van de handelingen dienen aantekeningen gemaakt te worden. De aantekeningen, de CD-ROM, de afdrukken en de back-upbanden dienen gezamenlijk bewaard te worden. Tijdens het herstellen kunnen de notities ervoor zorgen dat de back-upbanden vernietigd worden. Hoe? In plaats van `tar xvf /dev/sa0` kan per ongeluk `tar cvf /dev/sa0` worden ingetypt, waardoor de back-upband overschreven wordt.

Als extra veiligheidsmaatregel dienen telkens een "livefs" CD-ROM en twee back-upbanden gemaakt te worden. Eén van deze banden dient op een plaats op afstand bewaard te worden. Zo'n plaats is NIET de kelder van het zelfde kantoorgebouw. Een aantal bedrijven in het World Trade Center heeft deze les op de harde manier geleerd. Zo'n plaats dient fysiek gescheiden te zijn van de computers en de schijven door een significante afstand.

Na de ramp

De hamvraag is: heeft de hardware het overleefd? Er zijn regelmatig back-ups gemaakt, dus zorgen over de software zijn niet nodig.

Indien hardware beschadigd is, dienen kapotte onderdelen vervangen te worden voordat gepoogd wordt om een computer te gebruiken.

Plaats de "livefs" CD-ROM in de CD-ROM drive indien de hardware in orde is en start de computer op. Het originele installatiemenu wordt op het scherm getoond. Kies het land van toepassing en kies daarna Fixit — Repair mode with CDROM/DVD/floppy en kies het item CDROM/DVD — Use the live filesystem CDROM/DVD. `restore` en de andere benodigde programma's staan in /mnt2/rescue.

Herstel elk bestandssysteem apart.

Probeer de rootpartitie van de eerste schijf aan te koppelen (bijvoorbeeld `mount /dev/da0a /mnt`). Als het schijflabel beschadigd is, gebruik dan `bsdlabel` om de schijf opnieuw te partitioneren en te

labelen zodat deze overeenkomt met het afgedrukte en bewaarde label. Gebruik voor het opnieuw aanmaken van de bestandssystemen `newfs`. Koppel de rootpartitie van de schijf opnieuw aan voor lezen en schrijven (`mount -u -o rw /mnt`). Gebruik voor het herstellen van de gegevens van dit bestandssysteem het back-upprogramma en de back-upbanden (bijvoorbeeld `restore vrf /dev/sa0`). Koppel nu het bestandssysteem af (bijvoorbeeld `umount /mnt`). Herhaal dit voor elk beschadigd bestandssysteem.

Back-up de gegevens naar nieuwe banden als het systeem weer draait. De omstandigheden die verantwoordelijk waren voor de crash of het gegevensverlies kunnen weer voorkomen. Nu een extra uur investeren, kan later grote zorgen besparen.

17.13. Netwerk-, geheugen-, en bestandsgebaseerde bestandssystemen

Naast de schijven die fysiek in de computer zitten, diskettes, CD's, harde schijven, enzovoort, worden er ook andere vormen van schijven door FreeBSD begrepen: de *virtuele schijven*.

Dit omvat netwerkbestandssystemen zoals het Network File System en Coda, geheugengebaseerde bestandssystemen en bestandsgebaseerde bestandssystemen.

Nagelang de gebruikte versie van FreeBSD, zijn er andere gereedschappen voor het aanmaken en gebruiken van bestandsgebaseerde en geheugengebaseerde bestandssystemen.



Gebruik devfs(8) om de apparaatnodes transparant voor de gebruiker toe te wijzen.

17.13.1. Bestandsgebaseerd bestandssysteem

Met mdconfig(8) kunnen geheugenschijven, md(4), ingesteld worden en aangezet worden. Om mdconfig(8) te gebruiken, moet de module md(4) geladen worden of ondersteuning aan het kernelinstellingenbestand toegevoegd worden:

```
device md
```

Het commando mdconfig(8) ondersteunt drie types geheugen-gebaseerde virtuele schijven: geheugenschijven die met malloc(9) toegewezen zijn, geheugenschijven die een bestand als basis gebruiken en geheugenschijven die swapruimte als basis gebruiken. Een mogelijk gebruik is het aankoppelen van een beeld van een diskette of CD dat in een bestand bewaard wordt.

Om een bestaand beeld van een bestandssysteem aan te koppelen:

Voorbeeld 11. `mdconfig` gebruiken om een bestaand beeld van een bestandssysteem aan te koppelen

```
# mdconfig -a -t vnode -f schijfbeeld -u 0
# mount /dev/md0 /mnt
```

Om een nieuw beeld van een bestandssysteem aan te maken met mdconfig(8):

Voorbeeld 12. Nieuwe bestandsgebaseerde schijf aanmaken met `mdconfig`

```
# dd if=/dev/zero of=nieuwbeeld bs=1k count=5k
5120+0 records in
5120+0 records out
# mdconfig -a -t vnode -f nieuwbeeld -u 0
# bsdlabel -w md0 auto
# newfs md0a
/dev/md0a: 5.0MB (10240 sectors) block size 16384, fragment size 2048
    using 4 cylinder groups of 1.25MB, 80 blks, 192 inodes.
super-block backups (for fsck -b #) at:
 160, 2720, 5280, 7840
# mount /dev/md0a /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity  Mounted on
/dev/md0a       4710    4  4330     0%    /mnt
```

Indien het eenheidsnummer niet met de optie `-u` gespecificeerd wordt, gebruikt mdconfig(8) de automatische toewijzing van md(4) om een ongebruikt apparaat te selecteren. De naam van het toegewezen apparaat wordt op stdout weergegeven als md4. Meer details staan in de hulppagina van mdconfig(8).

Het commando mdconfig(8) is erg nuttig, hoewel het veel opdrachten vergt om een bestandsgebaseerd bestandssysteem aan te maken. FreeBSD wordt met mdmfs(8) geleverd. Dit programma stelt een md(4)-schijf in door gebruik te maken van mdconfig(8), zet er een bestandssysteem op door gebruik te maken van newfs(8) en koppel het aan door gebruik te maken van mount(8). Om hetzelfde bestandssysteembeeld als hierboven aan te maken en aan te koppelen:

Voorbeeld 13. Instellen en aankoppelen van een bestandsgebaseerde schijf met `mdmfs`

```
# dd if=/dev/zero of=nieuwbeeld bs=1k count=5k
5120+0 records in
5120+0 records out
# mdmfs -F nieuwbeeld -s 5m md0 /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity  Mounted on
/dev/md0        4718    4  4338     0%    /mnt
```

Als de optie `md` zonder eenheidsnummer gebruikt wordt, gebruikt mdmfs(8) de automatische toewijzing van md(4) om automatisch een ongebruikt apparaat te selecteren. Meer details staan in de hulppagina van mdmfs(8).

17.13.2. Geheugengebaseerd bestandssysteem

Voor een geheugen gebaseerd bestands systeem moet normaal gesproken "wisselbestand geheugen" gebruikt worden. Gebruik maken van wisselbestand geheugen wil niet perse zeggen dat de geheugen schijf direct in het wisselbestand gezet wordt, maar dat het bestand naar het wisselbestand geschreven kan worden indien nodig. Het is ook mogelijk om malloc(9) gebaseerde geheugen schijven te maken, maar door hiervan gebruik te maken kan het gebeuren dat het systeem crashed als de kernel uit het geheugen loopt.

Voorbeeld 14. Nieuwe geheugengebaseerde schijf aanmaken met `mdconfig`

```
# mdconfig -a -t swap -s 5m -u 1
# newfs -U md1
/dev/md1: 5.0MB (10240 sectors) block size 16384, fragment size 2048
    using 4 cylinder groups of 1.27MB, 81 blks, 192 inodes.
    with soft updates
super-block backups (for fsck -b #) at:
 160, 2752, 5344, 7936
# mount /dev/md1 /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity  Mounted on
/dev/md1        4718    4  4338     0%    /mnt
```

Voorbeeld 15. Nieuwe geheugengebaseerde schijf aanmaken met `mdmfs`

```
# mdmfs -s 5m md2 /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity  Mounted on
/dev/md2        4846    2  4458     0%    /mnt
```

17.13.3. Geheugenschijf van het systeem afkoppelen

Als een geheugen- of bestandsgebaseerd bestandssysteem niet gebruikt wordt, dienen alle bronnen aan het systeem vrijgegeven te worden. Koppel als eerste het bestandssysteem af, gebruikt daarna mdconfig(8) om de schijf van een systeem los te koppelen en de bronnen vrij te geven.

Om bijvoorbeeld alle bronnen die door /dev/md4 gebruikt worden los te koppelen en vrij te geven:

```
# mdconfig -d -u 4
```

Het is mogelijk om de informatie over ingestelde md(4) apparaten weer te geven door gebruik te maken van `mdconfig -l`.

17.14. Snapshots van bestandssystemen

FreeBSD biedt een mogelijkheid om samen met Soft Updates: snapshots van bestandssystemen.

Snapshots bieden de mogelijkheid om beelden van een gespecificeerd bestandssysteem te maken en ze als bestand te behandelen. Snapshotbestanden moeten aangemaakt worden in het bestandssysteem waarop de handeling wordt uitgevoerd en er mogen niet meer dan 20 snapshots per bestandssysteem worden aangemaakt. Actieve snapshots worden opgeslagen in het superblok zodat ze persistent zijn met afkoppel- en heraankoppelbewerkingen en met het opnieuw opstarten van het systeem. Als een snapshot niet langer nodig is, kan het met het standaardcommando rm(1) worden verwijderd. Snapshots kunnen in elke volgorde verwijderd worden, alhoewel misschien niet alle gebruikte ruimte teruggewonnen wordt omdat sommige vrijgegeven blokken mogelijk door een ander snapshot geclaimd worden.

De onveranderlijke bestandsvlag `snapshot` wordt door mksnap_ffs(8) ingesteld nadat het snapshotbestand initieel is aangemaakt. Het commando unlink(1) maakt een uitzondering voor snapshotbestanden aangezien het toestaat dat ze verwijderd worden.

Snapshotbestanden worden aangemaakt met mount(8). Om een snapshot van /var in het bestand /var/snapshot/snap te plaatsen:

```
# mount -o -o snapshot /var/snapshot/snap /var
```

Als alternatief kan mksnap_ffs(8) gebruikt worden om een snapshot aan te maken:

```
# mksnap_ffs /var /var/snapshot/snap
```

Snapshotbestanden kunnen gezocht worden op een bestandssysteem (bijvoorbeeld /var) door gebruik te maken van het commando find(1):

```
# find /var -flags snapshot
```

Nadat een snapshot is aangemaakt, kan het voor een aantal dingen gebruikt worden:

- Sommige systeembeheerders gebruiken een snapshotbestand voor back-updoeleinden, omdat het snapshot naar CD's of band overgezet kan worden;
- De bestandssysteem integriteit controle, fsck(8) kan gebruikt worden op het snapshot. Ervan uitgaande dat het bestandssysteem schoon was toen het werd aangekoppeld, zou dit altijd een schoon (en onveranderlijk) resultaat moeten opleveren. Dit is in principe wat het fsck(8) -achtergrondsproces doet;
- Het commando dump(8) draaien op het snapshot. Er wordt een dump teruggegeven die consistent is met het bestandssysteem en tijdsstempel van het snapshot. dump(8) kan ook in één commando een snapshot maken, een dumpbeeld aanmaken en daarna het snapshot verwijderen door gebruik te maken van de vlag `-L`;
- Het snapshot kan met mount(8) als bevroren beeld van het bestandssysteem worden

aangekoppeld. Om het snapshot /var/snapshot/snap aan te koppelen:

```
# mdconfig -a -t vnode -f /var/snapshot/snap -u 4
# mount -r /dev/md4 /mnt
```

Het is nu mogelijk om door de structuur van het bevroren bestandssysteem /var te lopen dat aangekoppeld is op /mnt. Alles zal initieel in dezelfde toestand verkeren als op het moment dat het snapshot werd aangemaakt. De enige uitzondering hierop is dat eerdere snapshots als bestanden met lengte nul verschijnen. Als een snapshot niet meer nodig is, kan het als volgt afgekoppeld worden:

```
# umount /mnt
# mdconfig -d -u 4
```

Meer informatie over `softupdates` en snapshots van bestandssystemen, inclusief technische documenten, staat op de website van Marshall Kirk McKusick op http://www.mckusick.com/.

17.15. Bestandssysteemquota

Quota zijn een optionele mogelijkheid van het besturingssysteem om de hoeveelheid schijfruimte en/of het aantal bestanden dat gebruikers of leden van een groep per bestandssysteem mogen gebruiken te beperken. Dit wordt het meeste gebruikt op timesharing-systemen waar het wenselijk is om het aantal bronnen dat elke gebruiker of groep van gebruikers mag gebruiken te beperken. Dit voorkomt dat één gebruiker of groep van gebruikers alle beschikbare schijfruimte in beslag neemt.

17.15.1. Schijfquota inschakelen

Controleer alvorens te proberen om schijfquota te gebruiken of quota ingesteld zijn in de kernel. Dit gebeurt door het toevoegen van de volgende regel aan het kernelinstellingenbestand:

```
options QUOTA
```

De standaardkernel GENERIC heeft deze optie niet aanstaan, dus is het nodig om een eigen kernel in te stellen, te bouwen en te installeren om gebruik te kunnen maken van schijfquota. Meer informatie over het instellen van de kernel staat in De FreeBSD-kernel instellen.

Vervolgens dienen schijfquota aangezet te worden in /etc/rc.conf. Op FreeBSD 7.X en eerder wordt deze regel toegevoegd:

```
enable_quotas="YES"
```

Voeg op FreeBSD 8.0-RELEASE en nieuwer in plaats daarvan deze regel toe:

```
quota_enable="YES"
```

Voor fijnere controle over de opstartquota zijn extra instellingsvariabelen beschikbaar. Normaalgesproken wordt de integriteit van de quota van elk bestandssysteem tijdens het opstarten door quotacheck(8) gecontroleerd. quotacheck(8) verzekert dat de gegevens in de quotadatabase een juiste afspiegeling vormen van de gegevens op het bestandssysteem. Dit proces neemt erg veel tijd in beslag en beïnvloedt de tijd die een systeem nodig heeft om op te starten significant. Om deze stap over te slaan, bestaat een variabele in /etc/rc.conf:

```
check_quotas="NO"
```

Als laatste dient /etc/fstab bewerkt te worden om schijfquota per bestandssysteem aan te zetten. Hier kunnen gebruiker- of groepquota of beide worden aangezet voor alle bestandssystemen.

Om quota per gebruiker op een bestandssysteem aan te zetten, dient de optie `userquota` aan het optieveld toegevoegd te worden aan de regel in /etc/fstab voor het bestandssysteem waar quota worden aangezet. Bijvoorbeeld:

```
/dev/da1s2g   /home    ufs rw,userquota 1 2
```

Analoog, om groepquota aan te zetten, dient de optie `groupquota` in plaats van `userquota` gebruikt te worden. Om zowel gebruikers- als groepsquota aan te zetten, dient de regel als volgt veranderd te worden:

```
/dev/da1s2g   /home    ufs rw,userquota,groupquota 1 2
```

Standaard worden de quotabestanden opgeslagen in de hoofdmap van het bestandssysteem onder de namen quota.user en quota.group voor respectievelijk gebruikers- en groepsquota. Meer informatie staat in fstab(5). Alhoewel de hulppagina fstab(5) vermeld dat een alternatieve plaats voor de quotabestanden gespecificeerd kan worden, wordt dit niet aangeraden omdat de verschillende quotagereedschappen dit niet juist schijnen af te handelen.

Hier aangekomen dient het systeem opnieuw opgestart te worden met de nieuwe kernel. /etc/rc voert automatisch de juiste commando's uit om de initiële quotabestanden aan te maken voor alle quota die in /etc/fstab zijn aangezet. Het is dus niet nodig om handmatig quotabestanden met lengte nul aan te maken.

Tijdens normale bewerkingen moet het niet nodig zijn om de commando's quotacheck(8), quotaon(8) of quotaoff(8) handmatig te draaien. Lees wel de betreffende hulppagina's om bekend te raken met de werking ervan.

17.15.2. Quotalimieten instellen

Indien het systeem ingesteld voor gebruik van quota, controleer dan of ze echt aanstaan. Een eenvoudige manier om dit te doen is de volgende:

```
# quota -v
```

Er hoort een eenregelige samenvatting te verschijnen over het schijfgebruik en de huidige quotalimieten voor elk bestandssysteem waarop quota aanstaan.

Nu kunnen quotalimieten toegewezen worden met edquota(8).

Er zijn verschillende opties om grenzen te stellen aan de hoeveelheid schijfruimte die een gebruiker of groep mag toewijzen en het aantal bestanden dat ze mogen aanmaken. Toewijzingen kunnen begrensd worden met betrekking tot schijfruimte (blokquota) of het aantal bestanden (inode-quota) of een combinatie van beide. Elk van deze limieten is op zijn beurt weer opgesplitst in twee categoriën: harde en zachte limieten.

Een harde limiet mag niet overschreden worden. Indien een gebruiker de harde limiet bereikt, mag deze geen verdere toewijzingen maken op het betreffende bestandssysteem. Indien een gebruiker bijvoorbeeld een harde limiet heeft van 500 kB op een bestandssysteem en er 490 kB van gebruikt, kan deze nog slechts 10 kB toewijzen. Een poging om 11 kB toe te wijzen zal mislukken.

Zachte limieten kunnen voor een beperkte tijd overschreden worden. Deze periode staat bekend als de gratieperiode, die standaard een week bedraagt. Als een gebruiker de zachte limiet langer dan de gratieperiode overschrijdt, verandert de zachte limiet in een harde limiet en zijn er geen verdere toewijzingen toegestaan. Als de gebruiker onder de zachte limiet komt, wordt de gratieperiode opnieuw ingesteld.

Het volgende is een voorbeeld van een mogelijk gebruik van edquota(8). Als het commando edquota(8) gestart wordt, wordt de tekstverwerker opgestart die door de omgevingsvariabele `EDITOR` gespecificeerd is, of de tekstverwerker vi als de variabele `EDITOR` niet is ingesteld. Nu kunnen de quotalimieten bewerkt worden.

```
# edquota -u test
```

```
Quotas for user test:
/usr: kbytes in use: 65, limits (soft = 50, hard = 75)
        inodes in use: 7, limits (soft = 50, hard = 60)
/usr/var: kbytes in use: 0, limits (soft = 50, hard = 75)
        inodes in use: 0, limits (soft = 50, hard = 60)
```

Normaalgesproken worden er twee regels weergegeven voor elk bestandssysteem waarvoor quota gelden: één regel voor de bloklimieten, en één voor de inode-limieten. Om de quotalimieten te veranderen dient de waarde ervan veranderd te worden. Om bijvoorbeeld de bloklimiet van een gebruiker te veranderen van een zachte limiet van 50 en een harde limiet van 75 in een zachte limiet van 500 en een harde limiet van 600, dient het volgende veranderd te worden:

```
/usr: kbytes in use: 65, limits (soft = 50, hard = 75)
```

In:

```
/usr: kbytes in use: 65, limits (soft = 500, hard = 600)
```

De nieuwe quotalimieten gelden zodra de tekstverwerker verlaten wordt.

Soms is het gewenst om quotalimieten in te stellen op een aantal UID's. Dit kan gedaan worden door de optie `-p` van edquota(8) te gebruiken. Wijs eerst de gewenste quotalimiet aan een gebruiker toe en draai daarna `edquota -p protogebruiker beginuid-einduid`. Indien bijvoorbeeld gebruiker `test` de gewenste quotalimieten heeft, kan het volgende commando gebruikt worden om deze quotalimieten te dupliceren voor UID's 10.000 tot en met 19.999:

```
# edquota -p test 10000-19999
```

Meer informatie staat in de hulppagina voor edquota(8).

17.15.3. Quotalimieten en schijfgebruik controleren

Zowel quota(1) als repquota(8) kunnen gebruikt worden om de quotalimieten en het schijfgebruik te controleren. Het commando quota(1) kan gebruikt worden om de quota van zowel individuele gebruikers als groepen en het schijfgebruik te controleren. Een gebruiker mag alleen de eigen quota en de quota van een groep waarvan deze lid is controleren. Alleen de beheerder mag alle gebruikers- en groepsquota bekijken. Het commando repquota(8) kan gebruikt worden om een overzicht te krijgen van alle quota en gebruik van bestandssystemen waarvan quota aanstaan.

Het volgende is een mogelijke uitvoer van het commando `quota -v` voor een gebruiker die quotalimieten heeft op twee bestandssystemen.

```
Disk quotas for user test (uid 1002):
     Filesystem  usage    quota   limit   grace   files   quota   limit   grace
           /usr      65*     50      75   5days       7      50      60
       /usr/var       0      50      75               0      50      60
```

Voor het bestandssysteem /usr in bovenstaand voorbeeld overschrijdt deze gebruiker de zachte limiet van 50 kB momenteel met 15 kB en heeft deze 5 dagen van de gratieperiode over. De asterisk, `*` geeft aan dat de gebruiker momenteel de quotalimiet overschrijdt.

Normaalgesproken worden bestandssystemen waarvan de gebruiker geen schijfruimte gebruikt niet weergegeven in de uitvoer van quota(1), zelfs niet als er de gebruiker een quotalimiet heeft voor dat bestandssysteem. De optie `-v` geeft deze bestandssystemen weer, zoals het bestandssysteem /usr/var in bovenstaand voorbeeld.

17.15.4. Quota over NFS

Quota worden afgedwongen door het quota-subsysteem op de NFS-server. De daemon rpc.rquotad(8) stelt quota-informatie beschikbaar aan het commando quota(1) op de NFS-cliënts,

wat de gebruikers op deze machines in staat stelt hun quota-statistieken in te zien.

`rpc.rquotad` dient als volgt in /etc/inetd.conf aangezet te worden:

```
rquotad/1      dgram rpc/udp wait root /usr/libexec/rpc.rquotad rpc.rquotad
```

Vervolgens dient `inetd` opnieuw gestart te worden:

```
# service inetd restart
```

17.16. Schijfpartities versleutelen

FreeBSD biedt uitstekende on-line bescherming tegen onbevoegde gegevenstoegang. Bestandsrechten en Mandatory Access Control (MAC) (zie Verplichte Toegangscontrole (MAC)) helpen voorkomen dat onbevoegde derde partijen toegang tot de gegevens krijgen als het besturingssysteem actief is en de computer aanstaat. De door het besturingssysteem afgedwongen rechten zijn echter niet relevant als een aanvaller fysieke toegang tot een computer heeft en deze de harde schijf van de computer in een ander systeem kan plaatsen om de gevoelige gegevens te kopiëren en te analyseren.

Afgezien van hoe een aanvaller in het bezit van een harde schijf of een uitgezette computer gekomen is, kan GEOM Based Disk Encryption (gbde) de gegevens op het bestandssysteem van de computer zelfs tegen hooggemotiveerde aanvallers met aanzienlijke middelen beschermen. In tegenstelling tot lastige versleutelmethoden die alleen losse bestanden versleutelen, versleutelt gbde gehele bestandssystemen op een transparante manier. De harde schijf komt nooit in aanraking met klare tekst.

Los van hoe een aanvaller in het bezit van een harde schijf of een uitgezette computer gekomen is, kunnen de cryptografische subsystemen GEOM Based Disk Encryption (gbde) en `geli` in FreeBSD gegevens op bestandssystemen van een computer beschermen tegen zelfs de meer gemotiveerde belagers die ook nog eens adequate middelen hebben. Anders dan met lastige versleutelingsmethoden die alleen individuele bestanden versleutelen, versleutelen `gbde` en `geli` transparant complete bestandssystemen. Er komt nooit platte tekst op een harde schijf.

17.16.1. Schijven versleutelen met gbde

1. Word `root`

   Het instellen van gbde vereist beheerdersrechten.

   ```
   % su -
   Password:
   ```

2. Voeg ondersteuning voor gbde(4) aan het kernelinstellingenbestand toe

Voeg de volgende regel toe aan het kernelinstellingenbestand:

```
options GEOM_BDE
```

Herbouw de kernel opnieuw zoals beschreven in De FreeBSD-kernel instellen.

Start op met de nieuwe kernel.

3. Een alternatief voor het hercompileren van de kernel is door gebruik te maken van kldload(8) om gbde(4) te laden:

```
# kldload geom_bde
```

Versleutelde harde schijf voorbereiden

In het volgende voorbeeld wordt aangenomen dat er een nieuwe harde schijf aan het systeem wordt toegevoegd die een enkele versleutelde partitie zal bevatten. Deze partitie wordt aangekoppeld als /private. gbde kan ook gebruikt worden om /home en /var/mail te versleutelen, maar daarvoor zijn complexere instructies nodig die buiten het bereik van deze inleiding vallen.

1. Voeg een nieuwe harde schijf toe

   Voeg de nieuwe harde schijf toe zoals beschreven in Schijven toevoegen. In dit voorbeeld is een nieuwe harde schijfpartitie toegevoegd als /dev/ad4s1c. De apparaten /dev/ad0s1* stellen bestaande standaard FreeBSD partities van het voorbeeldsysteem voor.

```
# ls /dev/ad*
/dev/ad0        /dev/ad0s1b     /dev/ad0s1e     /dev/ad4s1
/dev/ad0s1      /dev/ad0s1c     /dev/ad0s1f     /dev/ad4s1c
/dev/ad0s1a     /dev/ad0s1d     /dev/ad4
```

2. Maak een map aan voor gbde lockbestanden

```
# mkdir /etc/gbde
```

   Het lockbestand voor gbde bevat informatie die gbde nodig heeft om toegang te krijgen tot versleutelde partities. Zonder toegang tot de lockbestand is gbde niet in staat om de gegevens die op de versleutelde partitie staan te ontsleutelen zonder aanzienlijke handmatige tussenkomst die niet door de software ondersteund wordt. Elke versleutelde partitie gebruikt een ander lockbestand.

3. Initialiseer de gbde-partitie

   Een gbde-partitie dient geïnitialiseerd te worden voordat deze kan worden gebruikt. Deze initialisatie dient slechts eenmalig uitgevoerd te worden:

```
# gbde init /dev/ad4s1c -i -L /etc/gbde/ad4s1c.lock
```

gbde(8) opent een tekstverwerker om verschillende instellingen in een sjabloon te kunnen instellen. Stel de sector_size in op 2048 als UFS of UFS2 wordt gebruikt:

```
# $FreeBSD: src/sbin/gbde/template.txt,v 1.1 2002/10/20 11:16:13 phk Exp $
#
# Sector size is the smallest unit of data which can be read or written.
# Making it too small decreases performance and decreases available space.
# Making it too large may prevent filesystems from working.  512 is the
# minimum and always safe.  For UFS, use the fragment size
#
sector_size =   2048
[...]
```

gbde(8) vraagt twee keer om de wachtwoordzin voor het beveiligen van de gegevens. De wachtwoordzin dient beide keren hetzelfde te zijn. De mogelijkheid van gbde om de gegevens te beveiligen is geheel afhankelijk de gekozen wachtwoordzin.

Het commando `gbde init` maakt een lockbestand aan voor de gbde-partitie die in dit voorbeeld is opgeslagen als /etc/gbde/ad4s1c.lock. gdbde slotbestanden moeten eindigen op ".lock" om correct door het opstartscript /etc/rc.d/gbde gedetecteerd te worden.



gbde lockbestanden *moeten* samen met de inhoud van versleutelde partities geback-upped worden. Hoewel het verwijderen van een lockbestand op zich een gedreven aanvaller er niet van weerhoudt een gbde partitie te ontsleutelen, is de wettige eigenaar zonder het lockbestand niet in staat om de gegevens op de versleutelde partitie te benaderen zonder een aanzienlijke hoeveelheid werk die in het geheel niet ondersteund wordt door gbde(8) of de ontwerper ervan.

4. Koppel de versleutelde partitie aan de kernel

```
# gbde attach /dev/ad4s1c -l /etc/gbde/ad4s1c.lock
```

Er wordt om de wachtwoordzin gevraagd die gekozen is tijdens de initialisatie van de versleutelde partitie. Het nieuwe versleutelde apparaat verschijnt in /dev als /dev/apparaatnaam.bde:

```
# ls /dev/ad*
/dev/ad0        /dev/ad0s1b     /dev/ad0s1e     /dev/ad4s1
/dev/ad0s1      /dev/ad0s1c     /dev/ad0s1f     /dev/ad4s1c
/dev/ad0s1a     /dev/ad0s1d     /dev/ad4        /dev/ad4s1c.bde
```

5. Maak een bestandssysteem op het versleutelde apparaat

Nu het versleutelde apparaat aan de kernel gekoppeld is, kan een bestandssysteem op het apparaat aangemaakt worden. Met newfs(8) kan een bestandssysteem op het versleutelde apparaat aangemaakt wordne. Aangezien het veel sneller is om een nieuw UFS2 bestandssysteem te initialiseren dan om een oud UFS1 bestandssysteem te initialiseren, is het aan te raden om newfs(8) met de optie `-O2` te gebruiken.

```
# newfs -U -O2 /dev/ad4s1c.bde
```



Voer newfs(8) uit op een aangekoppelde gbde-partitie die geïndificeerd wordt door de uitbreiding *.bde op de apparaatnaam.

6. Mount de versleutelde partitie

   Maak een koppelpunt voor het versleutelde bestandssysteem aan:

   ```
   # mkdir /private
   ```

   Mount het versleutelde bestandssysteem:

   ```
   # mount /dev/ad4s1c.bde /private
   ```

7. Controleer of het versleutelde bestandssysteem beschikbaar is

   Het versleutelde bestandssysteem is nu zichtbaar met df(1) en gebruiksklaar:

   ```
   % df -H
   Filesystem        Size   Used  Avail Capacity  Mounted on
   /dev/ad0s1a      1037M    72M   883M     8%    /
   /devfs            1.0K   1.0K     0B   100%    /dev
   /dev/ad0s1f       8.1G    55K   7.5G     0%    /home
   /dev/ad0s1e      1037M   1.1M   953M     0%    /tmp
   /dev/ad0s1d       6.1G   1.9G   3.7G    35%    /usr
   /dev/ad4s1c.bde   150G   4.1K   138G     0%    /private
   ```

Bestaande versleutelde bestandssystemen aankoppelen

Elke keer nadat het systeem is opgestart dient elk versleuteld bestandssysteem opnieuw aan de kernel gekoppeld te worden, op fouten gecontroleerd te worden, en aangekoppeld te worden voordat de bestandssystemen gebruikt kunnen worden. De benodigde commando's dienen als de gebruiker `root` uitgevoerd te worden.

1. Koppel de gbde-partitie aan de kernel

```
# gbde attach /dev/ad4s1c -l /etc/gbde/ad4s1c.lock
```

Er wordt om de wachtwoordzin gevraagd die gekozen is tijdens de initialisatie van de versleutelde gbde-partitie.

2. Controleer het bestandssysteem op fouten

   Aangezien het nog niet mogelijk is om versleutelde bestandssystemen op te nemen in /etc/fstab voor automatische controle, dienen de bestandssystemen voordat ze aangekoppeld worden handmatig op fouten gecontroleerd te worden door fsck(8) uit te voeren:

   ```
   # fsck -p -t ffs /dev/ad4s1c.bde
   ```

3. Mount het versleutelde bestandssysteem

   ```
   # mount /dev/ad4s1c.bde /private
   ```

   Het versleutelde bestandssysteem is nu klaar voor gebruik.

Versleutelde partities automatisch aankoppelen

Het is mogelijk om een script aan te maken om automatisch een versleutelde partitie aan te koppelen, op fouten te controleren en aan te koppelen, maar vanwege veiligheidsredenen dient het script niet het wachtwoord voor gbde(8) te bevatten. In plaats hiervan wordt het aangeraden om zulke scripts handmatig uit te voeren en het wachtwoord via de console of ssh(1) te geven.

Als alternatief, wordt er een rc.d script bijgeleverd. De argumenten kunnen via rc.conf(5) doorgegeven worden. Bijvoorbeeld:

```
gbde_autoattach_all="YES"
gbde_devices="ad4s1c"
gbde_lockdir="/etc/gbde"
```

Hierdoor is het noodzakelijk dat de wachtwoordzin voor gbde bij het starten wordt ingegeven. Na het invoeren van de juiste wachtwoordzin wordt de met gbde versleutelde partitie automatisch aangekoppeld. Dit kan erg handig zijn bij het gebruik van gbde op notebooks.

Door gbde gebruikte cryptografische beschermingen

gbde(8) versleutelt de sectorlading door gebruik te maken van 128-bit AES in CBC-modus. Elke sector op de schijf wordt met een andere AES-sleutel versleuteld. Meer informatie over het cryptografische ontwerp van gbde, inclusief de methode die gebruikt wordt om de sectorsleutels van de door de gebruiker gegeven wachtwoordzin af te leiden, staan in gbde(4).

Compatibiliteitspunten

sysinstall(8) is niet compatibel met apparaten die met gbde versleuteld zijn. Alle *.bde apparaten moeten van de kernel ontkoppeld worden voordat sysinstall(8) gebruikt wordt om te voorkomen dat het crasht tijdens het initiële zoeken naar apparaten. Om het versleutelde apparaat dat in dit voorbeeld gebruikt wordt te ontkoppelen:

```
# gbde detach /dev/ad4s1c
```

17.16.2. Schijfversleuteling met `geli`

Een alternatieve cryptografische GEOM klassie is beschikbaar - `geli`. Deze wordt op het moment ontwikkeld door Paweł Jakub Dawidek <pjd@FreeBSD.org>. `geli` verschilt van `gbde` in de mogelijkheden en in het gebruik van een andere methode voor het versleutelen.

De meest belangrijke mogelijkheden van geli(8) zijn:

- Gebruikt het crypto(9) framework; als cryptografische hardware aanwezig is, gebruikt `geli` die automatisch;
- Ondersteunt meedere cryptografische algoritmen. Op dit moment AES, Blowfish en 3DES;
- Staat toe dat de root-partitie wordt versleuteld. De wachtwoordzin die wordt gebruikt om de root-partitie te versleutelen wordt opgevraagd tijdens het starten van een systeem;
- Staat het gebruik van twee onafhankelijke sleutels toe, bijvoorbeeld een "sleutel" en een "bedrijfssleutel");
- `geli` is snel; het werkt met sector-naar-sector versleuteling;
- Ondersteunt back-up en restore van Master Keys. Als een gebruiker sleutels moet vernietigen, is het mogelijk weer toegang te krijgen tot de gegevens door sleutels uit een back-up te halen;
- Ondersteunt het koppelen van een schijf met een willekeurige, eenmalige sleutel. Handig voor swap-partities en tijdelijke bestandssystemen.

Meer mogelijkheden van `geli` staan beschreven in de handleiding van geli(8).

De volgende stappen beschrijven hoe ondersteuning voor `geli` in de FreeBSD-kernel ingeschakeld kan worden en hoe een nieuwe `geli` versleutelingsvoorziening gemaakt kan worden.

Het is noodzakelijk super-user rechten te hebben omdat de kernel wordt aangepast.

1. Toevoegen van `geli`-ondersteuning

   Voeg de volgende regels toe aan het bestand met kernelinstellingen:

   ```
   options GEOM_ELI
   device crypto
   ```

   Herbouw de kernel zoals beschreven is in De FreeBSD-kernel instellen.

De `geli`-module kan ook bij het opstarten geladen worden. Voeg de volgende regel toe aan /boot/loader.conf:

```
geom_eli_load="YES"
```

Nu hoort geli(8) door de kernel ondersteund te worden.

2. Een Master Key genereren

   Het volgende voorbeeld beschrijft hoe een sleutelbestand te maken, dat wordt gebruikt als onderdeel van de Master Key voor de versleutelde dienst die wordt aangekoppeld onder /private. Het sleutelbestand zorgt voor wat willekeurige gegevens die worden gebruikt om de Master Key te versleutelen. De Master Key wordt ook door een wachtwoordzin beschermd. De sectorgrootte van de dienst wordt 4 kB. Ook wordt beschreven hoe de `geli` -dienst te koppelen, er een bestandsysteem op te maken, dat aan te koppelen, hoe ermee te werken en tenslotte hoe te ontkoppelen.

   Het wordt aangeraden een grotere sectorgrootte in te stellen (zoals 4 kB) voor betere prestaties.

   De Master Key wordt beschermd door een wachtwoordzin en de gegevensbron voor het sleutelbestand wordt /dev/random. De sectorgrootte van /dev/da2.eli, die als dienst wordt aangeduid, wordt 4 kB.

   ```
   # dd if=/dev/random of=/root/da2.key bs=64 count=1
   # geli init -s 4096 -K /root/da2.key /dev/da2
   Enter new passphrase:
   Reenter new passphrase:
   ```

   Het is niet verplicht om zowel een wachtwoordzijn als een sleutelbestand te gebruiken. De methodes kunnen onafhankelijk van elkaar gebruikt worden.

   Als een sleutelbestand wordt opgegeven als "-", wordt de standaardinvoer gebruikt. In het onderstaande voorbeeld wordt aangegeven hoe meer dan een sleutelbestand kan worden gebruikt.

   ```
   # cat sleutelbestand1 sleutelbestand2 sleutelbestand3 | geli init -K - /dev/da2
   ```

3. De dienst koppelen met de gemaakte sleutel

   ```
   # geli attach -k /root/da2.key /dev/da2
   Enter passphrase:
   ```

   Het nieuwe platte tekst-apparaat wordt /dev/da2.eli genoemd.

```
# ls /dev/da2*
/dev/da2  /dev/da2.eli
```

4. Het nieuwe bestandssysteem maken

   ```
   # dd if=/dev/random of=/dev/da2.eli bs=1m
   # newfs /dev/da2.eli
   # mount /dev/da2.eli /private
   ```

   Het versleutelde bestandssysteem moet nu zichtbaar zijn voor df(1) en beschikbaar zijn voor gebruik:

   ```
   # df -H
   Filesystem     Size   Used  Avail Capacity  Mounted on
   /dev/ad0s1a    248M    89M   139M    38%    /
   /devfs         1.0K   1.0K     0B   100%    /dev
   /dev/ad0s1f    7.7G   2.3G   4.9G    32%    /usr
   /dev/ad0s1d    989M   1.5M   909M     0%    /tmp
   /dev/ad0s1e    3.9G   1.3G   2.3G    35%    /var
   /dev/da2.eli   150G   4.1K   138G     0%    /private
   ```

5. De dienst afkoppelen

   Als het werk met de versleutelde partitie is afgehandeld en de /private-partitie niet langer nodig is, dan is het verstandig te overwegen de met `geli` versleutelde partitie af te koppelen van het bestandssysteem en de kernel.

   ```
   # umount /private
   # geli detach da2.eli
   ```

Meer informatie over geli(8) staat in de handleiding.

Gebruik maken van het geli rc.d script.

Bij `geli` hoort een rc.d script dat gebruikt kan worden om het gebruik van `geli` te vereenvoudigen. Een voorbeeld van hoe `geli` met rc.conf(5) ingesteld kan worden volgt:

```
geli_devices="da2"
geli_da2_flags="-p -k /root/da2.key"
```

Hiermee wordt /dev/da2 ingesteld als `geli`-dienst met Master Key-bestand /root/da2.key en `geli` gebruikt geen wachtwoordzin als de dienst wordt gekoppeld (dit kan alleen gebruikt worden als `-P` is meegegeven tijdens de `geli init` fase van `geli`). Een systeem ontkoppelt de `geli`-dienst van de kernel voordat het afsluit.

Meer informatie over het instellen van rc.d staat in het onderdeel over rc.d.

17.17. Het versleutelen van de wisselbestand ruimte

Het versleutelen van de wisselbestand ruimte is gemakkelijk met FreeBSD te configureren. Afhankelijk van welke versie er gebruikt wordt zijn er verschillende configuratie opties en instellingen mogelijk. De gbde(8) en geli(8) programma's kunnen gebruikt worden voor het versleutelen van het wisselbestand. Beide systemen maken gebruik van het encswap rc.d script.

De vorige sectie, Schijfpartities versleutelen, biedt een korte discussie over de verschillende versleutel systemen.

17.17.1. Waarom moet het wisselbestand versleuteld worden?

Net als met het versleutelen van harde schijven, wordt het versleutelen van het wisselbestand gebruikt om gevoelige data te beschermen. Stelt u eens een applicatie voor dat omgaat het wachtwoorden. Zolang deze wachtwoorden in het fysieke geheugen blijven is er niets aan de hand. Echter zodra deze verplaatst worden naar het wisselbestand om ruimte te maken voor andere applicaties, kan het gebeuren dat de wachtwoorden onbeschermd op de harde schijf geschreven worden, waardoor het makkelijk te achterhalen is voor iemand die kwaad wilt. Het versleutelen van het wisselbestand biedt hierin een mogelijke uitkomst.

17.17.2. Voorbereiding



Tot het einde van deze sectie zal ad0s1b het wisselbestand bevatten.

Tot op dit moment is het wisselbestand niet versleuteld. Het is mogelijk dat er reeds wachtwoorden of andere gevoelige data onbeschermd op de harde schijf geschreven zijn. Om dit te corrigeren, moet de data op de swap partitie overschreven worden met willekeurige data:

```
# dd if=/dev/random of=/dev/ad0s1b bs=1m
```

17.17.3. Versleutelen van het wisselbestand met gbde(8)

Er moet gebruik gemaakt worden van het `.bde` achtervoegsel aan het apparaat in de respectievelijke /etc/fstab-regel betreffende het wisselbestand:

```
# Device        Mountpoint  FStype  Options     Dump    Pass#
/dev/ad0s1b.bde     none        swap    sw      0   0
```

17.17.4. Versleutelen van het wisselbestand met geli(8)

Het opzetten van geli(8) voor het versleutelen van het wisselbestand is hetzelfde als dat van gbde(8). Hier moet echter gebruik gemaakt worden van het `.eli` achtervoegsel aan het apparaat in de respectievelijke /etc/fstab wisselbestand regel:

```
# Device        Mountpoint  FStype  Options     Dump    Pass#
/dev/ad0s1b.eli     none        swap    sw      0   0
```

geli(8) maakt standaard gebruik van het AES algoritme met een sleutellengte van 128 bits.

Optioneel kunnen deze standaardwaarden worden aangepast door gebruik te maken van de `geli_swap_flags` optie in /etc/rc.conf. De volgende regel verteld het encswap rc.d bestand om een geli(8) wisselbestand te maken met het Blowfish algoritme met een sleutel lengte van 128 bit, een sectorgrootte van 4 kilobytes en met de optie "ontkoppelen nadat de laatste afsluiting" gezet:

```
geli_swap_flags="-e blowfish -l 128 -s 4096 -d"
```

Zie de uitleg over het `onetime` commando in de geli(8) handleiding voor een lijst van mogelijke opties.

17.17.5. Controleren of het werkt

Zodra het systeem opnieuw opgestart is kan gekeken worden of alles nog goed werkt door gebruik te maken van het `swapinfo` commando.

Als gebruik gemaakt wordt van gbde(8):

```
% swapinfo
Device          1K-blocks     Used    Avail Capacity
/dev/ad0s1b.bde    542720        0   542720     0%
```

Als gebruik gemaakt wordt van geli(8):

```
% swapinfo
Device          1K-blocks     Used    Avail Capacity
/dev/ad0s1b.eli    542720        0   542720     0%
```

17.18. Highly Available Storage (HAST)

17.18.1. Overzicht

Hoge beschikbaarheid is een van de hoofdzaken in serieuze zakelijke toepassingen en hoog beschikbare opslag is een sleutelonderdeel in zulke omgevingen. Hoog beschikbare opslag, of HAST, werd ontwikkeld door Paweł Jakub Dawidek <pjd@FreeBSD.org> als een raamwerk dat transparante opslag van dezelfde gegevens toestaat over fysiek gescheiden machines die verbonden zijn door een TCP/IP-netwerk. HAST kan gezien worden als een netwerkgebaseerde RAID1 (spiegel) en is vergelijkbaar met het DRBD® opslagsysteem bekend van het GNU/Linux® platform. In combinatie met andere eigenschappen voor hoge beschikbaarheid van FreeBSD zoals CARP maakt HAST het mogelijk om een opslagcluster met hoge beschikbaarheid te bouwen dat

resistent is tegen falende hardware.

Na het lezen van deze sectie weet u:

- Wat HAST is, hoe het werkt en welke mogelijkheden het biedt.
- Hoe HAST op FreeBSD te op te zetten en te gebruiken.
- Hoe CARP en devd(8) te integreren om een robuust opslagsysteem te bouwen.

Voor het lezen van deze sectie dient u:

- De beginselen van UNIX® en FreeBSD te begrijpen (UNIX® beginselen).
- Te weten hoe de netwerkinterfaces en andere kerndeelsystemen van FreeBSD in te stellen (Instellingen en optimalisatie).
- Netwerken op FreeBSD goed te begrijpen (Netwerkcommunicatie).
- FreeBSD 8.1-RELEASE of nieuwer te gebruiken.

Het HAST-project werd gesponsord door The FreeBSD Foundation met ondersteuning van OMCnet Internet Service GmbH en TransIP BV.

17.18.2. Eigenschappen van HAST

De belangrijkste eigenschappen van HAST zijn:

- Het kan gebruikt worden om I/O-fouten op lokale harde schijven te maskeren.
- Agnostisch qua bestandssysteem, dus het werkt met elk bestandssysteem dat door FreeBSD wordt ondersteund.
- Efficiënte en snelle hersynchronisatie, alleen de blokken die zijn veranderd toen een knooppunt uitstond worden gesynchroniseerd.
- Het kan gebruikt worden in reeds uitgerolde omgevingen om aanvullende redundantie toe te voegen.
- Samen met CARP, Heartbeat of andere gereedschappen kan het worden gebruikt om een robuust en duurzaam opslagsysteem te bouwen.

17.18.3. Werking van HAST

Omdat HAST synchrone replicatie op blokniveau van elk opslagmedium naar verscheidene machines biedt, heeft het tenminste twee knooppunten (fysieke machines) nodig - het `primaire` (ook bekend als `meester`) knooppunt en het `secundaire` (`slaaf`) knooppunt. Tezamen worden deze twee machines een cluster genoemd.



HAST is momenteel beperkt tot een totaal van twee clusterknooppunten.

Aangezien HAST in een primaire-secundaire configuratie werkt, kan er op elk moment slechts één van de clusterknooppunten actief zijn. Het `primaire` knooppunt, ookwel `actief`, is degene die alle I/O-verzoeken aan apparaten die door HAST worden beheerd afhandelt. Het `secundaire` knooppunt wordt dan automatisch gesynchroniseerd vanuit het `primaire` knooppunt.

De fysieke componenten van het HAST-systeem zijn:

- lokale schijf (op primair knooppunt)
- schijf op verre machine (secundair knooppunt)

HAST werkt synchroon op blokniveau, wat het transparant maakt voor bestandssystemen en toepassingen. HAST biedt reguliere GEOM-aanbieders aan in /dev/hast/ voor zowel andere gereedschappen als toepassingen, er is dus geen verschil tussen het gebruik van apparaten die door HAST worden geleverd en rauwe schijven, partities, etc.

Elke bewerking met betrekking tot schrijven, verwijderen of spoelen wordt naar de plaatselijke schijf en over TCP/IP naar de verre schijf gestuurd. Elke leesbewerking wordt gedaan door de plaatselijke schijf, tenzij de plaatselijke schijf niet actueel is of er een I/O-fout optreed. In zulke gevallen wordt de leesbewerking naar het secundaire knooppunt gestuurd.

Synchronisatie- en replicatiemodi

HAST probeert om een snel herstel van fouten te leveren. Om deze reden is het heel belangrijk om de synchronisatietijd te verkorten nadat een knooppunt is hersteld van een uitval. Om een snelle synchronisatie te leveren, beheert HAST op de schijf een bitmap van gebruikte extents en synchroniseert het die alleen tijdens een reguliere synchronisatie (met uitzondering van de initiële synchronisatie).

Er zijn vele manieren om synchronisatie af te handelen. HAST implementeert meerdere replicatiemodi om verschillende synchronisatiemethodes af te handelen:

- *memsync*: rapporteer een schrijfbewerking als voltooid wanneer de plaatselijke schrijfbewerking klaar is en wanneer het verre knooppunt de gegevensaankomst bevestigt, maar voordat het de gegevens daadwerkelijk heeft opgeslagen. De gegevens op het verre knooppunt zullen meteen na het versturen van de bevestiging worden opgeslagen. Deze modus is bedoeld om latency te verminderen en nog steeds een zeer goede betrouwbaarheid te bieden. De replicatiemodus *memsync* is momenteel niet geïmplementeerd.
- *fullsync*: rapporteer een schrijfbewerking als voltooid wanneer zowel de plaatselijke en de verre schrijfbewerking voltooid zijn. Dit is de veiligste en traagste replicatiemodus. Dit is de standaardmodus.
- *async*: rapporteer de schrijfbewerking als voltooid wanneer de plaatselijke schrijfbewerking klaar is. Dit is de snelste en gevaarlijkste replicatiemodus. Het dient gebruikt te worden wanneer er naar een ver knooppunt wordt gerepliceerd en de latency te hoog is voor andere modi. De replicatiemodus *async* is momenteel niet geïmplementeerd.



Momenteel wordt alleen de replicatiemodus *fullsync* ondersteund.

17.18.4. HAST-configuratie

HAST heeft ondersteuning voor `GEOM_GATE` nodig om te kunnen functioneren. De kernel `GENERIC` bevat standaard *geen* `GEOM_GATE`, de laadbare module geom_gate.ko is echter beschikbaar in de standaardinstallatie van FreeBSD. Zorg ervoor dat deze module beschikbaar is voor afgeslankte systemen. Het is ook mogelijk om ondersteuning voor `GEOM_GATE` statisch in de kernel te bouwen,

door deze regel aan het kernelconfiguratiebestand toe te voegen:

```
options GEOM_GATE
```

Het HAST-raamwerk bestaat vanuit het besturingssysteem gezien uit verschillende delen:

- het daemon hastd(8) dat verantwoordelijk is voor de gegevenssynchronisatie,
- het beheerprogramma hastctl(8) voor de gebruikers,
- het configuratiebestand hast.conf(5).

Het volgende voorbeeld beschrijft hoe twee knooppunten in een `meester-slaaf` / `primaire-secundaire` opstelling te configureren door HAST te gebruiken om de gegevens tussen de twee te repliceren. De knooppunten worden `hasta` met IP-adres *172.16.0.1* en `hastb` met IP-adres *172.16.0.2* genoemd. Beide knooppunten hebben een toegewijde harde schijf /dev/ad6 van dezelfde grootte om met HAST te werken. De HAST-pool (soms ook een hulpbron genoemd, i.e., de GEOM-aanbieder in /dev/hast/) wordt test genoemd.

Het bestand /etc/hast.conf regelt de configuratie van HAST. Dit bestand dient hetzelfde te zijn op beide knooppunten. Het volgende is de eenvoudigste configuratie die mogelijk is:

```
resource test {
    on hasta {
        local /dev/ad6
        remote 172.16.0.2
    }
    on hastb {
        local /dev/ad6
        remote 172.16.0.1
    }
}
```

Raadpleeg voor geavanceerdere configuraties de handleidingpagina hast.conf(5).



Het is ook mogelijk om hostnamen in de regels met `remote` te gebruiken. Zorg er in dat geval voor dat deze hosts vindbaar zijn, bijvoorbeeld doordat ze zijn gedefinieerd in het bestand /etc/hosts of anders in het plaatselijke DNS.

Nu de configuratie op beide knooppunten aanwezig is, kan de HAST-pool aangemaakt worden . Voer deze commando's op beide knooppunten uit om de initiële metagegevens op de plaatselijke schijf te plaatsen en het hastd(8)-daemon te starten:

```
# hastctl create test
# service hastd onestart
```



Het is *niet* mogelijk om GEOM-aanbieders met een bestaand bestandssysteem te

> gebruiken (i.e., een bestaande opslag omzetten naar een door HAST beheerde pool), omdat deze procedure wat metagegevens op de aanbieder moet opslaan en er daarvoor niet genoeg beschikbare ruimte is.

De rol van een HAST-knooppunt (`primair` of `secundair`) wordt uitgekozen door een beheerder of software zoals Heartbeat dat het gereedschap hastctl(8) gebruikt. Voer het volgende commando uit op het primaire knooppunt (`hasta`):

```
# hastctl role primary test
```

Voer dit soortgelijke commando uit op het secundaire knooppunt (`hastb`):

```
# hastctl role secondary test
```



> De situatie dat de knooppunten niet met elkaar kunnen communiceren en beide geconfigureerd zijn als primaire knooppunten; wordt `split-brain` genoemd. Volg de stappen zoals beschreven in Herstellen van de Split-brain-conditie om deze situatie op te lossen.

Verifieer met het gereedschap hastctl(8) het resultaat op elk knooppunt:

```
# hastctl status test
```

De belangrijke tekst is de regel met `status` dat voor alle knooppunten `complete` dient te bevatten. Als het `degraded` bevat, is er iets verkeerd gegaan. Op dat moment is de synchronisatie tussen de knooppunten al begonnen. De synchronisatie is compleet wanneer `hastctl status` 0 bytes aan `dirty` extents rapporteert.

De volgende stap is het aanmaken van een bestandssysteem op de GEOM-aanbieder /dev/hast/test en het aan te koppelen. Dit moet op het `primaire` knooppunt gebeuren, aangezien /dev/hast/test alleen op het `primaire` knooppunt verschijnt. Het aanmaken van het bestandssysteem kan afhankelijk van de grootte van de harde schijf enkele minuten duren:

```
# newfs -U /dev/hast/test
# mkdir /hast/test
# mount /dev/hast/test /hast/test
```

Wanneer het HAST-raamwerk correct is geconfigureerd, betreft de laatste stap het ervoor zorgen dat HAST automatisch tijdens het opstarten wordt gestart. Voeg deze regel toe aan het bestand /etc/rc.conf:

```
hastd_enable="YES"
```

Failover-configuratie

Het doel van dit voorbeeld is om een robuust opslagsysteem te bouwen dat resistent is tegen het falen van alle knooppunten. Het scenario is dat een `primair` knooppunt van het cluster faalt. Als dit gebeurt, dan neemt het `secundaire` knooppunt het feilloos over, controleert het het bestandssysteem en koppelt het het bestandssysteem aan, en gaat het verder zonder dat er een bit aan gegevens ontbreekt.

Om dit voor elkaar te krijgen, is er een andere eigenschap die beschikbaar is op FreeBSD dat voorziet in automatische failover van de IP-laag - CARP. CARP (Common Address Redundancy Protocol) maakt het mogelijk dat meerdere hosts in hetzelfde netwerksegment een IP-adres delen. Stel CARP in op beide knooppunten van het cluster volgens de documentatie die beschikbaar is in Common Address Redundancy Protocol (CARP). Nadat de opzet voltooid is, heeft elk knooppunt een eigen interface carp0 met een gedeeld IP-adres *172.16.0.254*. Het primaire HAST-knooppunt van het cluster moet het meester-CARP-knooppunt zijn.

De HAST-pool die in de vorige sectie is gemaakt is nu klaar om geëxporteerd te worden naar de andere hosts op het netwerk. Dit kan gedaan worden door het te exporteren over NFS, Samba, etc., door gebruik te maken van het gedeelde IP-adres *172.16.0.254*. Het enige overgebleven probleem is een automatische failover in het geval dat het primaire knooppunt het begeeft.

Als een CARP-interface aan- of uitgaat, genereert FreeBSD een devd(8)-gebeurtenis, wat het mogelijk maakt om toestandsveranderingen op de CARP-interfaces in de gaten te houden. Een toestandsverandering op het CARP-interface geeft aan dat een van de knooppunten het begaf of weer online kwam. Deze toestandsveranderingen maken het mogelijk om een script te draaien dat automatisch de HAST-failover afhandelt.

Voeg, om toestandsverandering op de CARP-interfaces af te vangen, het volgende toe aan het bestand /etc/devd.conf op elk knooppunt:

```
notify 30 {
    match "system" "IFNET";
    match "subsystem" "carp0";
    match "type" "LINK_UP";
    action "/usr/local/sbin/carp-hast-switch master";
};

notify 30 {
    match "system" "IFNET";
    match "subsystem" "carp0";
    match "type" "LINK_DOWN";
    action "/usr/local/sbin/carp-hast-switch slave";
};
```

Herstart devd(8) op beide knooppunten om de nieuwe configuratie te laten gelden:

```
# service devd restart
```

Als het interface carp0 aan of uit gaat (i.e., de toestand van het interface verandert), genereert het systeem een notificatie wat het subsysteem devd(8) in staat stelt om een willekeurig script te draaien, in dit geval /usr/local/sbin/carp-hast-switch. Dit is het script dat de automatische failover afhandelt. Raadpleeg de handleidingpagina devd.conf(5) voor verdere uitleg over de bovenstaande configuratie van devd(8).

Dit zou een voorbeeld van zo'n script kunnen zijn:

```
#!/bin/sh
# Origineel script door Freddie Cash <fjwcash@gmail.com>
# Gewijzigd door Michael W. Lucas <mwlucas@BlackHelicopters.org>
# en Viktor Petersson <vpetersson@wireload.net>

# De namen van de HAST-hulpbronnen, zoals vermeld in /etc/hast.conf
resources="test"

# vertraging voor het aankoppelen van de HAST-hulpbron na het worden van meester
# doe een gok
delay=3

# logging
log="local0.debug"
name="carp-hast"

# einde van gebruiker-instelbare dingen

case "$1" in
    master)
        logger -p $log -t $name "Omschakelen naar primaire aanbieder voor
${resources}."
        sleep ${delay}

        # Wacht totdat de "hastd secondary" processen zijn gestopt
        for disk in ${resources}; do
            while $( pgrep -lf "hastd: ${disk} \(secondary\)" > /dev/null 2>&1 ); do
                sleep 1
            done

            # Verwissel de rol voor elke schijf
            hastctl role primary ${disk}
            if [ $? -ne 0 ]; then
                logger -p $log -t $name "Omschakelen van rol naar primair voor
hulpbron ${disk} mislukt."
                exit 1
            fi
        done

        # Wacht totdat de apparaten /dev/hast/* verschijnen
        for disk in ${resources}; do
            for I in $( jot 60 ); do
```

```
                [ -c "/dev/hast/${disk}" ] && break
                sleep 0.5
            done

            if [ ! -c "/dev/hast/${disk}" ]; then
                logger -p $log -t $name "GEOM-aanbieder /dev/hast/${disk} is niet
verschenen."
                exit 1
            fi
        done

        logger -p $log -t $name "Rollen van HAST-hulpbronnen ${resources} omgeschakeld
naar primair."

        logger -p $log -t $name "Schijven aankoppelen."
        for disk in ${resources}; do
            mkdir -p /hast/${disk}
            fsck -p -y -t ufs /dev/hast/${disk}
            mount /dev/hast/${disk} /hast/${disk}
        done

    ;;

    slave)
        logger -p $log -t $name "Omschakelen naar secundaire aanbieder voor
${resources}."

        # Schakel de rollen van de HAST-hulpbronnen om
        for disk in ${resources}; do
            if ! mount | grep -q "^/dev/hast/${disk} on "
            then
            else
                umount -f /hast/${disk}
            fi
            sleep $delay
            hastctl role secondary ${disk} 2>&1
            if [ $? -ne 0 ]; then
                logger -p $log -t $name "Omschakelen van rol naar secundair voor
hulpbron ${disk} mislukt."
                exit 1
            fi
            logger -p $log -t $name "Rol van hulpbron ${disk} omgeschakeld naar
secundair."
        done
    ;;
esac
```

In een notendop neemt het script deze acties wanneer een knooppunt `meester` / `primair` wordt:

- De HAST-pools opwaarderen naar primair op een gegeven knooppunt.

- Het bestandssysteem onder de HAST-pool controleren.
- De pools op een juiste plaats aankoppelen.

Wanneer een knooppunt `back-up` / `secundair` wordt:

- De HAST-pools afkoppelen.
- De HAST-pools degraderen naar secundair.



Houd in gedachte dat dit slechts een voorbeeldscript is om aan te tonen dat alles werkt. Het behandeld niet alle mogelijke situaties en kan op elke manier worden uitgebreid of veranderd, het kan bijvoorbeeld benodigde diensten starten en stoppen.



Voor dit voorbeeld hebben we een standaard UFS-bestandssysteem gebruikt. Om de tijd die nodig is voor herstel te verkorten, kan een bestandssysteem met UFS-journalling of ZFS worden gebruikt.

Meer gedetailleerde informatie met aanvullende voorbeelden kunnen gevonden worden op de HAST Wiki-pagina.

17.18.5. Problemen oplossen

Algemene tips om problemen op te lossen

HAST zou over het algemeen zonder problemen moeten werken. Net als met elk ander software-product zijn er momenten waarop het anders werkt dan het zou moeten. De oorzaken van de problemen kunnen verschillen, maar de vuistregel is om ervoor te zorgen dat de klokken zijn gesynchroniseerd op alle knooppunten in het cluster.

Wanneer problemen met HAST worden verholpen, dient het debug-niveau van hastd(8) verhoogd te worden door het daemon hastd(8) met het argument `-d` op te starten. Merk op dat dit argument meerdere malen kan worden opgegeven om het debug-niveau nog verder op te hogen. Op deze manier kan veel nuttige informatie worden vergaard. Overweeg ook om het argument `-F` te gebruiken, dat het daemon hastd(8) in de voorgrond zal starten.

Herstellen van de Split-brain-conditie

`Split-brain` treedt op waneer de knooppunten van het cluster niet met elkaar kunnen communiceren, en beide als primair zijn geconfigureerd. Dit is een gevaarlijke situatie omdat het beide knooppunten in staat stelt om incompatibele veranderingen aan de gegevens te maken. Dit probleem dient handmatig door de systeembeheerder te worden gecorrigeerd.

De beheerder moet besluiten welk knooppunt de belangrijkere veranderingen bevat (of ze handmatig samenvoegen) en HAST een volledige synchronisatie op het knooppunt dat de kapotte gegevens heeft laten uitvoeren. Voer hiervoor deze commando's uit op het knooppunt dat opnieuw gesynchroniseerd moet worden:

```
# hastctl role init <resource>
```

```
# hastctl create <resource>
# hastctl role secondary <resource>
```

Hoofdstuk 18. GEOM. Modulair schijftransformatie raamwerk

18.1. Overzicht

Dit hoofdstuk beschrijft het gebruik van schijven in het GEOM raamwerk in FreeBSD. Hieronder vallen de belangrijkste RAID besturingsprogramma's die het raamwerk gebruikt voor instellingen. In dit hoofdstuk wordt niet diepgaand beschreven hoe GEOM omgaat met I/O, het onderliggende subsysteem of code. Die informatie staat in het hulppagina voor geom(4) en de verscheidene "SEE ALSO" referenties. Dit hoofdstuk is ook geen definitief stuk over het instellen van RAID. Alleen de door GEOM ondersteunde RAID-classificaties worden beschreven.

Na het lezen van dit hoofdstuk weet de lezer:

- Welk type RAID-ondersteuning via GEOM beschikbaar is;
- Hoe de basisgereedschappen te gebruiken om de verschillende RAID-niveaus in te stellen, te onderhouden en te wijzigen;
- Hoe schijfapparaten via GEOM te spiegelen, aaneen te schakelen, te versleutelen en vanaf afstand schijven aan te sluiten;
- Hoe problemen op te lossen met schijven die via het GEOM raamwerk zijn aangesloten.

Veronderstelde voorkennis:

- Begrijpen hoe FreeBSD omgaat met schijfapparaten (Stockage des données);
- Weten hoe een nieuwe FreeBSD kernel in te stellen en te installeren (De FreeBSD-kernel instellen).

18.2. GEOM inleiding

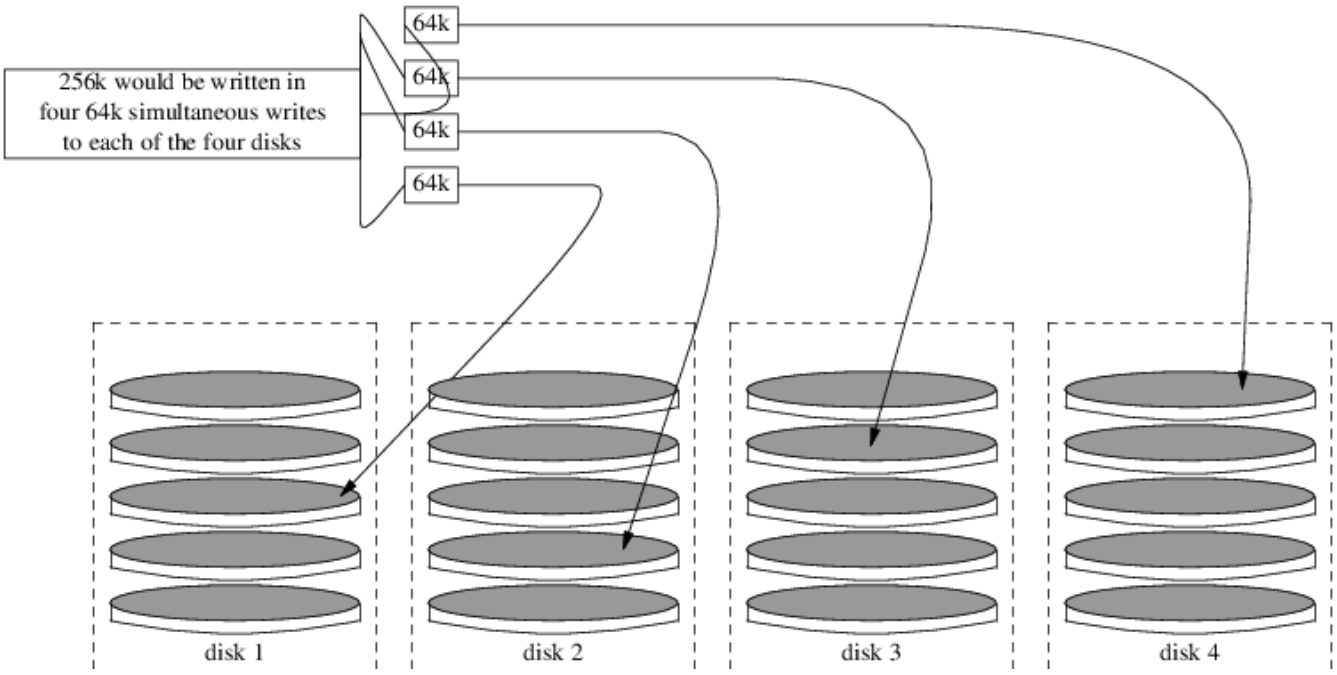
GEOM staat toegang en controle toe op klassen, Master Boot Records, BSD labels, enzovoort, door gebruik te maken van diensten of de speciale bestanden in /dev. GEOM ondersteunt verschillende software RAID instellingen en biedt transparante toegang tot het besturingssysteem en de hulpprogramma's.

18.3. RAID0 - aaneengeschakeld

Aaneenschakelen is een methode die gebruikt wordt om meerdere schijven te combineren tot een enkele volume. In veel gevallen wordt dit gedaan met hardware controllers. Het GEOM subsysteem biedt softwareondersteuning voor RAID0, ook wel bekend als aaneenschakelen ("disk striping").

In een RAID0-systeem worden gegevens opgedeeld in blokken die verdeeld worden over de schijven in een reeks. In plaats van te hoeven wachten tot een systeem 256k naar één schijf heeft geschreven, kan een RAID0-systeem tegelijkertijd 64k naar vier verschillende schijven schrijven, waardoor superieure I/O prestaties worden bereikt. Deze prestaties kunnen nog verbeterd worden door meerdere schijfcontrollers te gebruiken.

Iedere schijf in een RAID0-aaneenschakeling moet van dezelfde grootte zijn, omdat I/O-verzoeken altijd zijn opgebouwd uit precies gelijk over de schijven verdeelde verzoeken tot lezen of schrijven.



Procedure: Ongeformatteerde ATA-schijven aaneenschakelen

1. Laad de module geom_stripe.ko:

```
# kldload geom_stripe
```

2. Zorg ervoor dat er een koppelpunt beschikbaar is. Als dit volume een rootpartitie wordt, gebruikt dan tijdelijk een ander koppelpunt zoals /mnt:

```
# mkdir /mnt
```

3. Stel de apparaatnamen voor de schijven vast die aaneen worden geschakeld en maak het nieuwe apparaat aan. Om twee ongebruikte, ongepartitioneerde ATA schijven aaneen te schakelen (/dev/ad2 en /dev/ad3):

```
# gstripe label -v st0 /dev/ad2 /dev/ad3
Metadata value stored on /dev/ad2.
Metadata value stored on /dev/ad3.
Done.
```

4. Schrijf een standaard label naar de nieuwe partitie, ook wel bekend als een partitietabel en installeer de standaard opstart code:

```
# bsdlabel -wB /dev/stripe/st0
```

5. Dit proces hoort twee nieuwe apparaten gemaakt te hebben in de map /dev/stripe naast het apparaat st0, te weten st0a en st0c. Vanaf nu kan er een bestandssysteem op st0a worden gezet met behulp van de `newfs` applicatie:

   ```
   # newfs -U /dev/stripe/st0a
   ```

   Na het uitvoeren van het bovenstaande commando rollen er veel getallen over het scherm en na een aantal seconden is het proces afgerond. Het volume is gereed en klaar om aangekoppeld te worden.

Om de nieuwe aaneengeschakelde schijf handmatig te koppelen moet het volgende gedaan worden:

```
# mount /dev/stripe/st0a /mnt
```

Om dit aaneengeschakelde bestandssysteem automatisch aan te koppelen bij het opstarten wordt de volume-informatie in /etc/fstab gezet. Voor dit doel wordt een permanent koppelpunt, genaamd stripe, aangemaakt:

```
# mkdir /stripe
# echo "/dev/stripe/st0a /mnt ufs rw 2 2" \
      >> /etc/fstab
```

Laadt de module geom_stripe.ko ook automatisch bij het initialiseren van een systeem door de volgende regel toe te voegen aan /boot/loader.conf:

```
# echo 'geom_stripe_load="YES"' >> /boot/loader.conf
```

18.4. RAID1 - spiegelen

RAID1, of *spiegelen*, is de techniek om dezelfde gegevens naar meer dan één schijf te schrijven. Spiegels worden normaliter gebruikt om tegen gegevensverlies te beschermen indien een schijf kapot gaat. Elke schijf in een spiegel bevat een identieke kopie van de gegevens. Wanneer een individuele schijf het begeeft, blijft de spiegel functioneren, en levert het gegevens van de schijven die nog wel functioneren. De computer blijft draaien en de beheerder heeft tijd om de kapotte schijf te vervangen zonder onderbreking voor de gebruikers.

Twee veelvoorkomende situaties worden in deze voorbeelden getoond. Het eerste is het maken van een spiegel van twee nieuwe schijven en het als vervanging voor een bestaande enkele schijf te gebruiken. Het tweede voorbeeld maakt een spiegel op een enkele nieuwe schijf aan, kopieert de gegevens van de oude schijf er naar toe, en plaatst daarna de oude schijf in de spiegel. Hoewel deze procedure iets moeilijker is, is er maar één nieuwe schijf nodig.

Traditioneel zijn de twee schijven in een spiegel van hetzelfde model en hebben ze dezelfde capaciteit, maar gmirror(8) verplicht dit niet. Spiegels die met ongelijke schijven zijn gemaakt

zullen de capaciteit van de kleinste schijf in de spiegel aannemen. Extra schijfruimte op grotere schijven zal ongebruikt blijven. Schijven die later in de spiegel worden geplaatst moeten tenminste evenveel capaciteit hebben als de kleinste schijf die reeds in de spiegel zit.



De procedures voor het spiegelen die hier getoond worden zijn niet-destructief, maar maak zoals bij elke grote schijfoperatie eerst een volledige back-up.

18.4.1. Kwesties met meta-gegevens

Veel schijfsystemen slaan meta-gegevens op aan het einde van elke schijf. Oude meta-gegevens dienen gewist te worden voordat de schijf herbruikt wordt voor een spiegel. De meeste problemen worden veroorzaakt door twee soorten van achtergebleven meta-gegevens: GPT-partitietabellen en oude meta-gegevens van gmirror(8) van een vorige spiegel.

GPT-meta-gegevens kunnen gewist worden met gpart(8). Dit voorbeeld wist zowel de primaire als de back-up GPT-partitietabellen van schijf ada8:

```
# gpart destroy -F ada8
```

gmirror(8) kan in één stap een schijf uit een actieve spiegel halen en de meta-gegevens wissen. Hier wordt de voorbeeldschijf ada8 uit de actieve spiegel gm4 gehaald:

```
# gmirror remove gm4 ada8
```

Gebruik, als de spiegel niet draait maar er nog oude meta-gegevens van de spiegel op de schijf staan, `gmirror clear` om deze te verwijderen:

```
# gmirror clear ada8
```

gmirror(8) slaat één blok aan meta-gegevens aan het einde van de schijf op. Omdat GPT-partitieschema's ook meta-gegevens aan het einde van de schijf opslaan, wordt het spiegelen van volledige GPT-schijven met gmirror(8) niet aangeraden. Hier wordt MBR-partitionering gebruikt omdat het alleen een partitietabel aan het begin van de schijf opslaat en niet conflicteert met gmirror(8).

18.4.2. Een spiegel met twee nieuwe schijven maken

In dit voorbeeld is FreeBSD reeds op een enkele schijf ada0 geïnstalleerd. Twee nieuwe schijven, ada1 en ada2 zijn met het systeem verbonden. Er zal een nieuwe spiegel op deze twee schijven aangemaakt worden die de oude enkele schijf zal vervangen.

gmirror(8) heeft een kernelmodule geom_mirror.ko nodig, ingebouwd in de kernel of geladen tijdens het opstarten of draaien. Laadt nu handmatig de kernelmodule:

```
# gmirror load
```

Maak de spiegel aan met de twee nieuwe schijven:

```
# gmirror label -v gm0 /dev/ada1 /dev/ada2
```

gm0 is een door de gebruiker gekozen apparaatnaam die aan de nieuwe spiegel wordt toegekend. Nadat de spiegel is gestart, zal deze apparaatnaam verschijnen in de map /dev/mirror/.

Nu kunnen er met gpart(8) MBR- en bsdlabel-partitietabellen op de spiegel worden aangemaakt. Hier wordt er een traditioneel schema van een gesplitst bestandssysteem getoond, met partities voor /, swap, /var, /tmp en /usr. Dit werkt ook voor een enkel bestandssysteem met enkel / en een wisselpartitie.

Partities op de spiegel hoeven niet dezelfde grootte te hebben als die op de bestaande schijf, maar moeten groot genoeg zijn om alle gegevens die reeds op ada0 staan te kunnen bevatten.

```
# gpart create -s MBR mirror/gm0
# gpart add -t freebsd -a 4k mirror/gm0
# gpart show mirror/gm0
=>       63  156301423  mirror/gm0  MBR  (74G)
         63         63                    - free -  (31k)
        126  156301299                1  freebsd  (74G)
  156301425         61                    - free -  (30k)
```

```
# gpart create -s BSD mirror/gm0s1
# gpart add -t freebsd-ufs  -a 4k -s 2g mirror/gm0s1
# gpart add -t freebsd-swap -a 4k -s 4g mirror/gm0s1
# gpart add -t freebsd-ufs  -a 4k -s 2g mirror/gm0s1
# gpart add -t freebsd-ufs  -a 4k -s 1g mirror/gm0s1
# gpart add -t freebsd-ufs  -a 4k       mirror/gm0s1
# gpart show mirror/gm0s1
=>        0  156301299  mirror/gm0s1  BSD  (74G)
          0          2                     - free -  (1.0k)
          2    4194304                   1  freebsd-ufs  (2.0G)
    4194306    8388608                   2  freebsd-swap  (4.0G)
   12582914    4194304                   4  freebsd-ufs  (2.0G)
   16777218    2097152                   5  freebsd-ufs  (1.0G)
   18874370  137426928                   6  freebsd-ufs  (65G)
  156301298          1                     - free -  (512B)
```

Maak de spiegel opstartbaar door opstartcode in het MBR en bsdlabel te installeren en de actieve slice in te stellen:

```
# gpart bootcode -b /boot/mbr mirror/gm0
```

```
# gpart set -a active -i 1 mirror/gm0
# gpart bootcode -b /boot/boot mirror/gm0s1
```

Formatteer de bestandssystemen op de nieuwe spiegel en zet daarbij soft-updates aan.

```
# newfs -U /dev/mirror/gm0s1a
# newfs -U /dev/mirror/gm0s1d
# newfs -U /dev/mirror/gm0s1e
# newfs -U /dev/mirror/gm0s1f
```

Bestandssystemen van de originele schijf (ada0) kunnen nu met dump(8) en restore(8) naar de spiegel gekopieerd worden:

```
# mount /dev/mirror/gm0s1a /mnt
# dump -C16 -b64 -0aL -f - / | (cd /mnt && restore -rf -)
# mount /dev/mirror/gm0s1d /mnt/var
# mount /dev/mirror/gm0s1e /mnt/tmp
# mount /dev/mirror/gm0s1f /mnt/usr
# dump -C16 -b64 -0aL -f - /var | (cd /mnt/var && restore -rf -)
# dump -C16 -b64 -0aL -f - /tmp | (cd /mnt/tmp && restore -rf -)
# dump -C16 -b64 -0aL -f - /usr | (cd /mnt/usr && restore -rf -)
```

/mnt/etc/fstab moet bewerkt worden om naar de nieuwe bestandssystemen op de spiegel te wijzen:

```
# Device        Mountpoint  FStype  Options Dump    Pass#
/dev/mirror/gm0s1a  /       ufs rw  1   1
/dev/mirror/gm0s1b  none        swap    sw  0   0
/dev/mirror/gm0s1d  /var        ufs rw  2   2
/dev/mirror/gm0s1e  /tmp        ufs rw  2   2
/dev/mirror/gm0s1f  /usr        ufs rw  2   2
```

Als de kernelmodule gmirror(8) niet in de kernel is ingebouwd, wordt /mnt/boot/loader.conf bewerkt om de module tijdens het opstarten te laden:

```
geom_mirror_load="YES"
```

Herstart het systeem om de nieuwe spiegel te testen en te verifiëren dat alle gegevens zijn gekopieerd. Het BIOS zal de spiegel als twee individuele schijven zien in plaats van als een spiegel. Omdat de schijven identiek zijn, maakt het niet uit vanaf welke schijf wordt opgestart.

Bekijk de sectie Problemen oplossen als er problemen zijn tijdens het opstarten. Door de originele ada0 uit te schakelen en los te koppelen kan het als offline back-up bewaard worden.

Tijdens het gebruik zal de spiegel zich net zoals de originele enkele schijf gedragen.

18.4.3. Een spiegel met een bestaande schijf aanmaken

In dit voorbeeld is FreeBSD reeds geïnstalleerd op een enkele schijf, ada0. Een nieuwe schijf, ada1, is met het systeem verbonden. Er zal een spiegel van één schijf worden aangemaakt op de nieuwe schijf, het bestaande systeem zal ernaar worden gekopieerd, en daarna zal de oude schijf in de spiegel worden geplaatst. Deze enigszins complexe procedure is nodig omdat gmirror(8) een blok van 512 bytes aan meta-gegevens aan het einde van elke schijf moet plaatsen en de bestaande ada0 meestal alle ruimte reeds heeft toegewezen.

Laadt de kernelmodule gmirror(8):

```
# gmirror load
```

Controleer de mediagrootte van de originele schijf met diskinfo(8):

```
# diskinfo -v ada0 | head -n3
/dev/ada0
    512             # sectorsize
    1000204821504   # mediasize in bytes (931G)
```

Maak een spiegel aan op de nieuwe schijf. Om er zeker van te zijn dat de capaciteit van de spiegel niet groter is dan die van de originele schijf, wordt gnop(8) gebruikt om een nepschijf van precies dezelfde grootte aan te maken. Deze schijf slaat geen gegevens op, maar wordt alleen gebruikt om de grootte van de spiegel te begrenzen. Wanneer gmirror(8) de spiegel aanmaakt, zal het de capaciteit beperken tot de grootte van gzero.nop zelfs als de nieuwe schijf (ada1) meer ruimte heeft. Merk op dat de *1000204821504* op de tweede regel gelijk moet zijn aan de mediagrootte van ada0 zoals hierboven door diskinfo(8) is getoond.

```
# geom zero load
# gnop create -s 1000204821504 gzero
# gmirror label -v gm0 gzero.nop ada1
# gmirror forget gm0
```

gzero.nop slaat geen gegevens op, dus ziet de spiegel het niet als verbonden. De spiegel wordt verteld om componenten die niet verbonden zijn te "vergeten", waarbij referenties naar gzero.nop worden verwijderd. Het resultaat is een spiegelapparaat dat slechts één enkele schijf, ada1, bevat.

Bekijk de partitietabel van ada0 nadat gm0 is aangemaakt.

Deze uitvoer komt van een schijf van 1 TB. Als er wat niet-toegewezen ruimte aanwezig is aan het einde van de schijf, kan de inhoud direct van ada0 naar de nieuwe spiegel worden gekopieerd.

Als de uitvoer echter toont dat alle ruimte op de schijf is toegewezen zoals in de volgende lijst, is er geen ruimte over voor de 512 bytes aan meta-gegevens van gmirror(8) aan het einde van de schijf.

```
# gpart show ada0
```

```
=>        63  1953525105        ada0  MBR  (931G)
          63  1953525105           1  freebsd  [active]  (931G)
```

In dit geval moet de partitietabel worden bewerkt om de capaciteit op mirror/gm0 met één sector te verminderen. De procedure hiervoor wordt later uitgelegd.

In beide gevallen dienen de partitietabellen op de primaire schijf eerst gekopieerd te worden. Dit kan gedaan worden met de subcommando's `backup` en `restore` van gpart(8).

```
# gpart backup ada0 > table.ada0
# gpart backup ada0s1 > table.ada0s1
```

Deze subcommando's maken twee bestanden aan, table.ada0 en table.ada0s1. Dit voorbeeld komt van een schijf van 1 TB af:

```
# cat table.ada0
MBR 4
1 freebsd         63 1953525105   [active]
```

```
# cat table.ada0s1
BSD 8
1  freebsd-ufs          0    4194304
2 freebsd-swap    4194304   33554432
4  freebsd-ufs   37748736   50331648
5  freebsd-ufs   88080384   41943040
6  freebsd-ufs  130023424  838860800
7  freebsd-ufs  968884224  984640881
```

Als de gehele schijf was gebruikt in de uitvoer van gpart(8) `show`, dan moet de capaciteit in deze partitietabellen met één sector verminderd worden. Bewerk de twee bestanden zodat de grootte van zowel de slice als de laatste partitie met één verminderd wordt. Dit zijn de laatste getallen in elke lijst.

```
# cat table.ada0
MBR 4
1 freebsd         63 1953525104   [active]
```

```
# cat table.ada0s1
BSD 8
1  freebsd-ufs          0    4194304
2 freebsd-swap    4194304   33554432
4  freebsd-ufs   37748736   50331648
5  freebsd-ufs   88080384   41943040
6  freebsd-ufs  130023424  838860800
```

```
7  freebsd-ufs  968884224  984640880
```

Als er tenminste één sector aan het einde van de schijf niet was toegewezen, kunnen deze twee bestanden ongewijzigd gebruikt worden.

Herstel nu de partitietabel naar mirror/gm0.

```
# gpart restore mirror/gm0 < table.ada0
# gpart restore mirror/gm0s1 < table.ada0s1
```

Controleer de partitietabel met gpart(8) show. Dit voorbeeld heeft gm0s1a voor /, gm0s1d voor /var, gm0s1e voor /usr, gm0s1f voor /data1 en gm0s1g voor /data2.

```
# gpart show mirror/gm0
=>        63  1953525104  mirror/gm0  MBR  (931G)
          63  1953525042           1  freebsd  [active]  (931G)
  1953525105          62              - free -  (31k)

# gpart show mirror/gm0s1
=>         0  1953525042  mirror/gm0s1  BSD  (931G)
           0     2097152             1  freebsd-ufs  (1.0G)
     2097152    16777216             2  freebsd-swap  (8.0G)
    18874368    41943040             4  freebsd-ufs  (20G)
    60817408    20971520             5  freebsd-ufs  (10G)
    81788928   629145600             6  freebsd-ufs  (300G)
   710934528  1242590514             7  freebsd-ufs  (592G)
  1953525042          63                - free -  (31k)
```

Zowel de slice als de laatste partitie dienen wat vrije ruimte aan het einde van elke schijf te hebben.

Maak bestandssystemen aan op deze nieuwe partities. Het aantal partities zal variëren, overeenkomend met de partities op de originele schijf, ada0.

```
# newfs -U /dev/mirror/gm0s1a
# newfs -U /dev/mirror/gm0s1d
# newfs -U /dev/mirror/gm0s1e
# newfs -U /dev/mirror/gm0s1f
# newfs -U /dev/mirror/gm0s1g
```

Maak de spiegel opstartbaar door opstartcode in het MBR en bsdlabel te installeren en de actieve slice in te stellen:

```
# gpart bootcode -b /boot/mbr mirror/gm0
# gpart set -a active -i 1 mirror/gm0
# gpart bootcode -b /boot/boot mirror/gm0s1
```

Pas /etc/fstab aan zodat het de nieuwe partities op de spiegel gebruikt. Maak eerst een kopie van dit bestand als /etc/fstab.orig.

```
# cp /etc/fstab /etc/fstab.orig
```

Wijzig /etc/fstab door /dev/ada0 door mirror/gm0 te vervangen.

```
# Device        Mountpoint  FStype  Options Dump    Pass#
/dev/mirror/gm0s1a  /       ufs rw  1   1
/dev/mirror/gm0s1b  none        swap    sw  0   0
/dev/mirror/gm0s1d  /var        ufs rw  2   2
/dev/mirror/gm0s1e  /usr        ufs rw  2   2
/dev/mirror/gm0s1f  /data1      ufs rw  2   2
/dev/mirror/gm0s1g  /data2      ufs rw  2   2
```

Als de kernelmodule gmirror(8) niet in de kernel is gebouwd, wijzig dan /boot/loader.conf om het te laden:

```
geom_mirror_load="YES"
```

Bestandssystemen van de originele schijf kunnen nu met dump(8) en restore(8) naar de spiegel gekopieerd worden. Merk op dat het maken van een snapshot voor elk bestandssysteem dat met `dump -L` gedumpt is even kan duren.

```
# mount /dev/mirror/gm0s1a /mnt
# dump -C16 -b64 -0aL -f - /    | (cd /mnt && restore -rf -)
# mount /dev/mirror/gm0s1d /mnt/var
# mount /dev/mirror/gm0s1e /mnt/usr
# mount /dev/mirror/gm0s1f /mnt/data1
# mount /dev/mirror/gm0s1g /mnt/data2
# dump -C16 -b64 -0aL -f - /usr | (cd /mnt/usr && restore -rf -)
# dump -C16 -b64 -0aL -f - /var | (cd /mnt/var && restore -rf -)
# dump -C16 -b64 -0aL -f - /data1 | (cd /mnt/data1 && restore -rf -)
# dump -C16 -b64 -0aL -f - /data2 | (cd /mnt/data2 && restore -rf -)
```

Start het systeem opnieuw op vanaf ada1. Als alles werkt, zal het systeem opstarten vanaf mirror/gm0, wat nu dezelfde gegevens bevat die ada0 eerder bevatte. Zie de sectie Problemen oplossen als er problemen zijn met het opstarten.

Op dit moment bestaat de spiegel nog steeds alleen uit de enkele schijf ada1.

Nadat er succesvol van mirror/gm0 is opgestart, is de laatste stap het plaatsen van ada0 in de spiegel.



Als ada0 in de spiegel wordt geplaatst, zal de vorige inhoud worden overschreven door gegevens in de spiegel. Ben er zeker van dat mirror/gm0 dezelfde gegevens

> bevat als ada0 voordat ada0 aan de spiegel wordt toegevoegd. Als er iets mis is met de gegevens die door dump(8) en restore(8) gekopieerd zijn, draai dan /etc/fstab terug om de bestandssystemen op ada0 aan te koppelen, start opnieuw op, en probeer de hele procedure nogmaals.

```
# gmirror insert gm0 ada0
GEOM_MIRROR: Device gm0: rebuilding provider ada0
```

De synchronisatie tussen de twee schijven zal onmiddellijk beginnen. gmirror(8) `status` toont de voortgang.

```
# gmirror status
      Name    Status  Components
mirror/gm0  DEGRADED  ada1 (ACTIVE)
                      ada0 (SYNCHRONIZING, 64%)
```

Na een tijd zal de synchronisatie voltooid zijn.

```
GEOM_MIRROR: Device gm0: rebuilding provider ada0 finished.
# gmirror status
      Name    Status  Components
mirror/gm0  COMPLETE  ada1 (ACTIVE)
                      ada0 (ACTIVE)
```

mirror/gm0 bestaat nu uit de twee schijven ada0 en ada1, en de inhoud wordt automatisch met elkaar gesynchroniseerd. In het gebruik zal mirror/gm0 zich net zo gedragen als de originele enkele schijf.

18.4.4. Problemen oplossen

Problemen met opstarten

BIOS-instellingen

Mogelijk is het nodig om de BIOS-instellingen te wijzigen om van één van de nieuwe gespiegelde schijven op te starten. Beide spiegelschijven kunnen gebruikt worden voor het opstarten. Als componenten van een spiegel bevatten ze identieke gegevens.

Opstartproblemen

Als het opstarten met dit bericht stopt, is er iets mis met het spiegelapparaat:

```
Mounting from ufs:/dev/mirror/gm0s1a failed with error 19.

Loader variables:
  vfs.root.mountfrom=ufs:/dev/mirror/gm0s1a
```

```
  vfs.root.mountfrom.options=rw

Manual root filesystem specification:
  <fstype>:<device> [options]
      Mount <device> using filesystem <fstype>
      and with the specified (optional) option list.

    eg. ufs:/dev/da0s1a
        zfs:tank
        cd9660:/dev/acd0 ro
          (which is equivalent to: mount -t cd9660 -o ro /dev/acd0 /)

  ?               List valid disk boot devices
  .               Yield 1 second (for background tasks)
  <empty line>    Abort manual input

mountroot>
```

Het vergeten om de module geom_mirror in /boot/loader.conf te laden kan dit probleem veroorzaken. Start op vanaf een FreeBSD-9 of nieuwere CD of USB-stick en kies `Shell` op de eerste prompt om dit op te lossen. Laadt daarna de spiegelmodule en en koppel het spiegelapparaat aan:

```
# gmirror load
# mount /dev/mirror/gm0s1a /mnt
```

Voeg een regel om de spiegelmodule te laden toe aan /mnt/boot/loader.conf:

```
geom_mirror_load="YES"
```

Sla het bestand op en start opnieuw op.

Andere problemen die `error 19` veroorzaken zijn lastiger om op te lossen. Typ `ufs:/dev/ada0s1a` in op de prompt. Hoewel het systeem van ada0 zou moeten opstarten, verschijnt er een andere prompt om een shell uit te kiezen omdat /etc/fstab onjuist is. Druk op de prompt op de Enter-toets. Draai de wijzigingen tot nu toe terug door /etc/fstab terug te draaien, waardoor de bestandssystemen vanaf de originele schijf (ada0) in plaats vanaf de spiegel worden aangekoppeld. Start het systeem opnieuw op en probeer de procedure nogmaals.

```
Enter full pathname of shell or RETURN for /bin/sh:
# cp /etc/fstab.orig /etc/fstab
# reboot
```

18.4.5. Herstellen van falende schijven

Het mooie aan het spiegelen van schijven is dat een individuele schijf kan falen zonder dat de spiegel gegevens verliest.

ada0 is één van de twee schijven die de spiegel in het vorige voorbeeld vormen. Als ada0 faalt zal de spiegel blijven werken en gegevens leveren van de overgebleven werkende schijf, ada1.

Om de kapotte schijf te vervangen wordt de computer uitgezet en de kapotte schijf fysiek vervangen door een nieuwe schijf van gelijke of grotere capaciteit. Fabrikanten passen enigszins willekeurige waarden toe om schijven in gigabytes aan te duiden, de enige manier om er echt zeker van te zijn is om de totale hoeveelheid aan sectors aangegeven door `diskinfo -v` te vergelijken. Een schijf met een grotere capaciteit dan in de spiegel zal werken, alhoewel de extra ruimte op de nieuwe schijf niet gebruikt zal worden.

Nadat de computer opnieuw is aangezet, zal de spiegel in een "degraded" toestand met slechts één schijf draaien. De spiegel wordt verteld om schijven die momenteel niet verbonden zijn te vergeten:

```
# gmirror forget gm0
```

Alle oude meta-gegevens zouden van de vervangende schijf gewist moeten zijn. Daarna wordt de schijf, in dit voorbeeld ada4, in de spiegel geplaatst:

```
# gmirror insert gm0 /dev/ada4
```

De hersynchronisatie begint wanneer de nieuwe schijf in de spiegel wordt geplaatst. Het kopiëren van gegevens van de spiegel naar een nieuwe schijf kan een tijd duren. De prestaties van de spiegel zullen tijdens het kopiëren sterk verminderd zijn, dus is het het beste om nieuwe schijven in te voegen wanneer de vraag op de computer laag is.

De voortgang kan met `gmirror status` gevolgd worden, wat de schijven die gesynchroniseerd en het percentage van de voltooiing laat zien. Tijdens de hersynchronisatie zal de status `DEGRADED` zijn en veranderen in `COMPLETE` wanneer het proces is voltooid.

18.5. RAID3 - Striping op byte-niveau met toegewijde pariteit

RAID3 is een methode om verschillende schijven te combineren in een enkel volume met een toegewijde schijf voor de pariteit. In een RAID3-systeem worden de gegevens opgesplitst in een aantal bytes die over alle schijven in de rij worden geschreven, behalve naar één schijf die als een toegewijde schijf voor de pariteit dient. Dit betekent dat het lezen van 1024 kB van een RAID3-implementatie alle schijven in de rij zal benaderen. De prestatie kan worden verhoogd door meerdere schijfcontrollers te gebruiken. De RAID3-rij biedt een fouttolerantie van 1 schijf, terwijl het een capaciteit van 1 - 1/n maal de totale capaciteit biedt van alle schijven in de rij, waarbij n het aantal harde schijven in de rij is. Zulke configuraties zijn meestal geschikt voor het opslaan van gegevens van grotere groottes, bijvoorbeeld multimedia-bestanden.

Er zijn minstens 3 fysieke harde schijven nodig om een RAID3-rij te bouwen. Elke schijf moet van dezelfde grootte zijn, aangezien I/O-verzoeken worden verweven om parallel naar meerdere schijven te lezen of schrijven. Bovendien moet vanwege de aard van RAID3 het aantal schijven gelijk zijn aan 3, 5, 9, 17, enzovoorts (dus 2^n + 1).

18.5.1. Een toegewijde RAID3-rij aanmaken.

In FreeBSD is ondersteuning voor RAID3 geïmplementeerd in de GEOM-klasse graid3(8). Voor het aanmaken van een toegewijde RAID3-rij op FreeBSD zijn deze stappen nodig.



Hoewel het theoretisch mogelijk is om op FreeBSD van een RAID3-rij op te starten, is deze configuratie ongebruikelijk en niet aangeraden.

1. Laad ten eerste de kernelmodule geom_raid3.ko door de volgende opdracht uit te voeren:

   ```
   # graid3 load
   ```

   Het is ook mogelijk om handmatig de module geom_raid3.ko te laden:

   ```
   # kldload geom_raid3.ko
   ```

2. Zorg ervoor dat er een geschikt aankoppelpunt bestaat of maak het aan:

   ```
   # mkdir /multimedia/
   ```

3. Bepaal de apparaatnamen voor de schijven die aan de rij worden toegevoegd en maak het nieuwe RAID3-apparaat aan. Het laatst vermelde apparaat zal dienst doen als de toegewijde schijf voor de pariteit. Dit voorbeeld gebruikt drie ongepartitioneerde ATA-schijven: ada1 en ada2 voor gegevens en ada3 voor pariteit.

   ```
   # graid3 label -v gr0 /dev/ada1 /dev/ada2 /dev/ada3
   Metadata value stored on /dev/ada1.
   Metadata value stored on /dev/ada2.
   Metadata value stored on /dev/ada3.
   Done.
   ```

4. Partitioneer het nieuw aangemaakte apparaat gr0 en zet er een UFS-bestandssysteem op:

   ```
   # gpart create -s GPT /dev/raid3/gr0
   # gpart add -t freebsd-ufs /dev/raid3/gr0
   # newfs -j /dev/raid3/gr0p1
   ```

   Vele getallen zullen over het scherm lopen, en na wat tijd zal het proces voltooid zijn. Het volume is aangemaakt en is klaar om aangekoppeld te worden.

5. De laatste stap is het aankoppelen van het bestandssysteem:

```
# mount /dev/raid3/gr0p1 /multimedia/
```

De RAID3-rij is nu klaar voor gebruik.

Aanvullende configuratie is nodig om de bovenstaande opstelling te behouden tussen het opnieuw starten van het systeem.

1. De module geom_raid3.ko moet geladen zijn voordat de rij kan worden aangekoppeld. Voeg de volgende regel toe aan /boot/loader.conf om de kernelmodule automatisch tijdens de initialisatie van het systeem te laden:

```
geom_raid3_load="YES"
```

2. De volgende volume-informatie moet aan het bestand /etc/fstab worden toegevoegd om het bestandssysteem van de rij automatisch aan de koppelen tijdens het opstarten van het systeem:

```
/dev/raid3/gr0p1    /multimedia ufs rw  2   2
```

18.6. GEOM Gate netwerk apparaten

GEOM ondersteund het op afstand gebruiken van apparaten, zoals schijven, CD-ROMs, bestanden, etcetera door het gebruik van de gate-applicaties. Dit is vergelijkbaar met NFS.

Om te beginnen moet er een exports bestand gemaakt worden. Dit bestand specificeert wie de geëxporteerde bron mag benaderen en welke rechten er op dat moment verleend worden. Bijvoorbeeld om de vierde slice te exporteren van de eerste SCSI schijf, moet het volgende in /etc/gg.exports gezet worden:

```
192.168.1.0/24 RW /dev/da0s4d
```

Dit staat alle machines in het privé netwerk toe om het bestandssysteem op da0s4d te benaderen.

Om dit apparaat te kunnen exporteren is het van belang dat de schijf nog niet gekoppeld is en moet de ggated(8) dienst gestart worden.

```
# ggated
```

Om vervolgens het apparaat aan een client machine te koppelen moet het volgende gedaan worden:

```
# ggatec create -o rw 192.168.1.1 /dev/da0s4d
ggate0
# mount /dev/ggate0 /mnt
```

Vanaf dit moment kan de schijf benaderd worden via het koppelpunt /mnt.



Let op, dit mislukt als de schijf reeds gekoppeld is op de server machine of als deze reeds gekoppeld is aan een andere machine op het netwerk.

Zodra het apparaat niet langer nodig is, kan het veilig ontkoppeld worden met behulp van umount(8) net zoals met elke andere schijf.

18.7. Het labelen van schijven

Tijdens het initialiseren van het systeem zal de FreeBSD kernel apparaatknooppunten creëren nadat het een apparaat gevonden heeft. Deze manier om te zoeken naar apparaten levert wat problemen op bijvoorbeeld wanneer er een nieuwe schijf wordt toegevoegd via USB. Het is hoogst waarschijnlijk dat een flash apparaat een apparaatknooppunt krijgt van da0, waarna de originele da0 op schuift naar da1. Dit levert problemen op als bestandssystemen worden gekoppeld als ze gedefinieerd zijn in /etc/fstab, dit kan zelfs ertoe leiden dat het systeem niet opstart.

Een mogelijke oplossing hiervoor is om de SCSI schijven een vaste plek te geven op een bepaalde volgorde, zodat zodra er een nieuwe schijf geplaatst wordt, deze een ongebruikt apparaatknooppunt toegewezen krijgt. Maar wat als er USB apparaten zijn die de primaire SCSI schijf vervangt? Dit gebeurd omdat USB apparaten meestal eerder gevonden worden dan een SCSI kaart. Een oplossing hiervoor is om de apparaten pas toe te voegen als het systeem reeds gestart is, een andere methode kan zijn om alleen een enkele ATA schijf te koppelen en nooit SCSI schijven door middel van /etc/fstab.

Maar er is een betere oplossing beschikbaar. Door het gebruik van `glabel` kunnen beheerders en gebruikers een label toevoegen aan een schijf, en deze labels gebruiken in /etc/fstab. Omdat `glabel` het label bewaard in de laatste sector van de schijf, kan het label bewaard blijven ook na een reboot en kan het bestandssysteem altijd gekoppeld worden ongeacht welk apparaatknooppunt toegekend is aan het apparaat.



Uiteraard hoeft een label niet permanent te zijn, het `glabel` programma kan zowel tijdelijke als permanente labels aanmaken. Alleen een permanent label blijft beschikbaar ook na een reboot. Zie de glabel(8) handleiding voor meer informatie over de verschillen tussen de labeltypes.

18.7.1. Label types en voorbeelden

Er zijn twee type labels: een generiek label en een bestandssysteemlabel. Labels kunnen permanent of tijdelijk zijn. Permanente labels kunnen met de commando's tunefs(8) of newfs(8) aangemaakt worden. Ze zullen vervolgens worden aangemaakt in een submap van /dev, welke genoemd wordt naar het bestandssysteemtype. Bijvoorbeeld UFS2 labels worden geplaatst in de map /dev/ufs. Permanente labels kunnen ook worden aangemaakt met het commando `glabel label`. Deze zijn

niet specifiek voor het bestandssysteem, en zullen in de map /dev/label aangemaakt worden.

Een tijdelijk label verdwijnt na een herstart van het systeem. Deze labels worden gecreëerd in /dev/label en zijn perfect voor experimenten. Een tijdelijk kan met het commando `glabel create` worden aangemaakt. Lees voor meer informatie de handleidingpagina van glabel(8).

Om een permanent label te schrijven voor een UFS2-bestandssysteem zonder de huidige data te vernietigen:

```
# tunefs -L home /dev/da3
```



Als het bestandssyteem vol is kan dit leiden tot data corruptie; echter als het bestandssysteem vol is zou het hoofddoel moeten zijn om oude achtergebleven bestanden weg te halen in plaats van het toevoegen van labels.

Er zou nu een label moeten bestaan in /dev/ufs, welke toegevoegd kan worden aan het /etc/fstab bestand:

```
/dev/ufs/home   /home           ufs     rw              2       2
```



Het bestandssysteem mag niet aangekoppeld zijn op het moment dat `tunefs` gebruikt wordt.

Nu kan het bestandssysteem net als normaal worden gekoppeld:

```
# mount /home
```

Vanaf dit moment is het mogelijk om, zolang de geom_label.ko geladen wordt tijdens het opstarten van het systeem, of als deze is meegecompileerd door middel van de `GEOM_LABEL` optie in de kernel, het apparaatknooppunt te wijzigen zonder ernstige gevolgen voor het systeem.

Bestandssystemen kunnen ook een standaard label mee krijgen door gebruik te maken van de `-L` optie met het `newfs` commando. Zie de newfs(8) handleiding voor meer informatie.

Het volgende commando kan worden gebruikt om een label te verwijderen:

```
# glabel destroy home
```

Het volgende voorbeeld laat zien hoe de partities van een opstartschijf gelabeld worden.

Voorbeeld 16. Partities op de opstartschijf labelen

Door de partities op de opstartschijf permanent te labelen zou het systeem in staat moeten zijn om normaal door te gaan met opstarten, zelfs als de schijf verplaatst is naar een andere controller of is overgeplaatst naar een ander systeem. In dit voorbeeld wordt aangenomen dat

er een enkele ATA-schijf wordt gebruikt, die momenteel als ad0 door het systeem wordt herkend. Het wordt ook aangenomen dat het standaard partitieschema van FreeBSD wordt gebruikt, met de bestandssystemen /, /var, /usr, en /tmp, alsmede een wisselpartitie.

Start het systeem opnieuw op, en druk bij de loader(8)-prompt op 4 om in enkele gebruikersmodus op te starten. Geef dan de volgende commando's:

```
# glabel label rootfs /dev/ad0s1a
GEOM_LABEL: Label for provider /dev/ad0s1a is label/rootfs
# glabel label var /dev/ad0s1d
GEOM_LABEL: Label for provider /dev/ad0s1d is label/var
# glabel label usr /dev/ad0s1f
GEOM_LABEL: Label for provider /dev/ad0s1f is label/usr
# glabel label tmp /dev/ad0s1e
GEOM_LABEL: Label for provider /dev/ad0s1e is label/tmp
# glabel label swap /dev/ad0s1b
GEOM_LABEL: Label for provider /dev/ad0s1b is label/swap
# exit
```

Het systeem zal doorgaan met opstarten in meergebruikersmodus. Bewerk, nadat het opstarten is voltooid, /etc/fstab en vervang de conventionele namen door de respectievelijke labels. Het uiteindelijke bestand /etc/fstab zal er als volgt uitzien:

```
# Device                Mountpoint      FStype  Options         Dump    Pass#
/dev/label/swap          none            swap    sw              0       0
/dev/label/rootfs        /               ufs     rw              1       1
/dev/label/tmp           /tmp            ufs     rw              2       2
/dev/label/usr           /usr            ufs     rw              2       2
/dev/label/var           /var            ufs     rw              2       2
```

Het systeem kan nu worden herstart. Als alles goed ging, zal het normaal opstarten en zal `mount` dit laten zien:

```
# mount
/dev/label/rootfs on / (ufs, local)
devfs on /dev (devfs, local)
/dev/label/tmp on /tmp (ufs, local, soft-updates)
/dev/label/usr on /usr (ufs, local, soft-updates)
/dev/label/var on /var (ufs, local, soft-updates)
```

Beginnend met FreeBSD 7.2 ondersteunt de klasse glabel(8) een nieuw labeltype voor UFS-bestandssystemen, gebaseerd op het unieke id van het bestandssysteem, `ufsid`. Deze labels kunnen in de map /dev/ufsid gevonden worden en worden automatisch tijdens het opstarten aangemaakt. Het is mogelijk om de `ufsid`-labels te gebruiken om partities aan te koppelen door middel van de faciliteit /etc/fstab. Gebruik `glabel status` om een lijst van bestandssystemen en hun overeenkomende `ufsid`-labels te ontvangen:

```
% glabel status
                  Name  Status  Components
ufsid/486b6fc38d330916     N/A  ad4s1d
ufsid/486b6fc16926168e     N/A  ad4s1f
```

In het bovenstaande voorbeeld representeert ad4s1d het bestandssysteem /var, terwijl ad4s1f het bestandssysteem /usr representeert. Door gebruik te maken van de gegeven `ufsid`-waarden kunnen deze partities nu aangekoppeld worden met de volgende regels in /etc/fstab:

```
/dev/ufsid/486b6fc38d330916 /var    ufs rw  2   2
/dev/ufsid/486b6fc16926168e /usr    ufs rw  2   2
```

Elke partitie met een `ufsid`-label kan op deze manier worden aangekoppeld, waardoor het niet meer nodig is om handmatig permanente labels voor ze aan te maken, terwijl er nog steeds van de voordelen van apparaatnaam-onafhankelijk aankoppelen genoten kan worden.

18.8. UFS logboeken door middel van GEOM

Met de komst van FreeBSD 7.0 komt ook de langverwachte optie van UFS logboeken. De implementatie zelf is gedaan door middel van het GEOM subsysteem, welke makkelijk geconfigureerd kan worden met behulp van de gjournal(8) applicatie.

Wat is logboeken? Logboek mogelijkheden betekend het opslaan van bestandssysteem transacties, zoals wijzigingen die een complete schrijfactie zijn, voor er meta-data wordt toegevoegd en voor de wijzigingen op schijf worden gezet. Deze transactie log kan later opnieuw afgespeeld worden om te voorkomen dat er bestandssysteem inconsistenties voorkomen.

Deze methode is een extra manier om te beschermen tegen gegevensverlies en inconsistenties van het bestandssysteem. In tegenstelling tot Soft Updates, welke bijhoud welke meta-data wijzigingen er worden uitgevoerd en Snapshots, wat een beeld bestand is van het bestandssysteem, wordt er een complete log bewaard in de schijfruimte die speciaal voor deze taak is gereserveerd, en in sommige gevallen op een compleet andere schijf.

In tegenstelling tot andere logboek implementaties is de `gjournal` methode blok gebaseerd en niet geïmplementeerd als onderdeel van het bestandssysteem maar als uitbreiding op GEOM.

Om ondersteuning in te schakelen voor `gjournal`, moet de kernel over de volgende optie beschikken, welke standaard is op FreeBSD 7.X-systemen:

```
options        UFS_GJOURNAL
```

Indien gejournalde volumes tijdens het opstarten aangekoppeld moeten worden, moet de kernelmodule geom_journal.ko ook geladen zijn, door de volgende regel aan /boot/loader.conf toe te voegen:

```
geom_journal_load="YES"
```

Ook kan deze functie in een eigen kernel worden ingebouwd, door de volgende regel aan het kernelinstellingenbestand toe te voegen:

```
options    GEOM_JOURNAL
```

Het creëren van een logboek op een vrij en beschikbaar bestandssysteem kan nu gedaan worden met behulp van de volgende stappen, ervan uitgaande dat da4 de nieuwe beschikbare SCSI schijf is:

```
# gjournal load
# gjournal label /dev/da4
```

Op dit moment zou er een ad4 apparaatknooppunt en een ad4.journal apparaatknooppunt moeten zijn. Nu kan er een bestandssysteem op gezet worden:

```
# newfs -O 2 -J /dev/da4.journal
```

Het hiervoor ingevoerde commando zal een UFS2 bestandssysteem met logboek ondersteuning aanmaken.

Koppel het apparaat op een gewenst koppelpunt met:

```
# mount /dev/da4.journal /mnt
```



In het geval dat er meerdere slices zijn, zal er een logboek voor elke slice gecreëerd worden. Bijvoorbeeld, als ad4s1 en ad4s2 allebei slices zijn, dan zal `gjournal` een ad4s1.journal en een ad4s2.journal creëeren.

Voor performance doeleinden is het gewenst om het logboek op een andere schijf te bewaren. Voor deze gevallen moet de logboekleverancier of het opslagapparaat gespecificeerd worden achter het apparaat waarop de logboek functionaliteit aangebracht moet worden. De logboekfunctionaliteit kan ook worden ingeschakeld op een reeds bestaand systeem met behulp van `tunefs`. Maak echter altijd een back-up voor dat dit soort dingen uitgeprobeerd worden. In de meeste gevallen zal `gjournal` falen als het geen actueel logboek kan maken, maar het voorkomt geen dataverlies als gevolg van verkeerd gebruik van `tunefs`.

Het is ook mogelijk om een journal van de opstartschijf van een FreeBSD-systeem bij te houden. Voor gedetailleerde instructies over deze taak wordt naar het artikel Implementing UFS Journaling on a Desktop PC verwezen.

Unresolved directive in book.adoc - include::zfs/_index.adoc[leveloffset=+1]

Hoofdstuk 19. Ondersteuning van bestandssystemen

19.1. Overzicht

Bestandssystemen zijn een integraal onderdeel van ieder besturingssysteem. Ze stellen gebruikers in de gelegenheid om bestanden te uploaden en op te slaan, geven toegang tot gegevens en maken natuurlijk harde schijven bruikbaar. Verschillende besturingssystemen hebben gewoonlijk één gezamenlijk aspect, namelijk het bestandssysteem. Op FreeBSD staat dit bestandssysteem bekend onder de naam Fast File System ofwel FFS, dat is gebaseerd op het oorspronkelijke Unix™ File System, ook bekend als UFS. Dit is het oorspronkelijke bestandssysteem van FreeBSD dat op harde schijven wordt geplaatst voor gegevenstoegang.

FreeBSD ondersteunt daarnaast ook een groot aantal andere bestandssystemen om lokaal toegang tot gegevens van andere besturingssystemen te bewerkstelligen; dat wil zeggen: gegevens opgeslagen op lokaal aangesloten USB opslagapparaten, flash drives, en harde schijven. Verder is er ook ondersteuning voor vreemde bestandssystemen. Dit zijn bestandssystemen ontwikkeld voor andere besturingssystemen zoals het Linux® Extended File System (EXT) en het Sun™ Z File System (ZFS).

Er zijn verschillende gradaties van ondersteuning voor de verschillende bestandssystemen op FreeBSD. Sommigen vereisen het laden van een kernelmodule, voor anderen moet een toolset worden geïnstalleerd. Dit hoofdstuk is geschreven om gebruikers van FreeBSD te helpen om op hun systeem toegang te verkrijgen tot andere bestandssystemen, te beginnen met het Sun™ Z File System.

Na het lezen van dit hoofstuk weet de lezer:

- Het verschil tussen eigen en ondersteunde bestandssystemen.
- Welke bestandssystemen zijn ondersteund in FreeBSD.
- Hoe niet-eigen bestandssystemen geactiveerd, geconfigureerd, benaderd en gebruikt kunnen worden.

Voorafgaand aan het lezen van dit hoofdstuk dient de lezer:

- Begrip te hebben van de beginselen van UNIX® en FreeBSD (UNIX® beginselen).
- Bekend te zijn met de beginselen van kernelconfiguratie en -compilatie (De FreeBSD-kernel instellen).
- Vertrouwd te zijn met installatie van software van derden in FreeBSD (Applicaties installeren. pakketten en ports).
- Enigszins bekend te zijn met schijven, opslag en apparaatnamen in FreeBSD (Opslag).

19.2. Het Z File System (ZFS)

Het Z File System, ontwikkeld door Sun™, is een nieuwe technologie ontwikkeld om gebruik te

maken van een pool-gebaseerde opslagmethode. Dit houdt in dat ruimte pas wordt gebruikt wanneer het nodig is voor dataopslag. Verder is het ontworpen voor maximale integriteit van gegevens, ondersteuning van gegevens-snapshots, meerdere kopieën, en gegevenschecksums. Ook is een nieuw gegevensreplicatiemodel, bekend als RAID-Z, toegevoegd; RAID-Z lijkt op RAID5, maar is ontworpen om corruptie tijdens het schrijven van gegevens te voorkomen.

19.2.1. ZFS tuning

Het ZFS subsysteem maakt gebruik van veel systeembronnen waardoor het nodig kan zijn een en ander af te stellen, zodat voor het dagelijks gebruik maximale efficiëntie wordt behaald. Doordat het een experimentele eigenschap van FreeBSD is, kan dit in de nabije toekomst veranderen; op dit moment echter, worden de volgende stappen aangeraden.

Geheugen

De totale hoeveelheid systeemgeheugen dient minstens één gigabyte te zijn, maar twee gigabytes of meer wordt aanbevolen. In alle voorbeelden hier heeft het systeem één gigabyte geheugen, met verschillende andere afstelmechanismen in werking.

Sommigen hebben succes gehad met minder dan een gigabyte geheugen, maar met een dergelijke, beperkte hoeveelheid geheugen is de kans groot dat onder zware belasting een kernelpanic in FreeBSD op zal treden door uitputting van het geheugen.

Kernelconfiguratie

Het wordt aangeraden om ongebruikte stuurprogramma's en opties te verwijderen uit het kernelconfiguratiebestand. Omdat de meeste stuurprogramma's beschikbaar zijn als modules kunnen ze alsnog worden geladen door middel van het bestand /boot/loader.conf.

Gebruikers van de i386™-architectuur dienen de volgende optie aan hun kernelconfiguratiebestand toe te voegen, de kernel opnieuw te compileren, en opnieuw op te starten:

```
options     KVA_PAGES=512
```

Deze optie vergroot de kerneladresruimte, waarmee het mogelijk wordt gemaakt om de `vm.kvm_size` afstelling hoger dan de huidige limiet van 1 GB (2 GB voor PAE) in te stellen. Deel, om de meest geschikte waarde voor deze optie te vinden, de gewenste hoeveelheid adresruimte door vier (4). In dit geval is dat `512` voor 2 GB.

Loader tunables

De kmem adresruimte dient te worden vergroot op alle FreeBSD architecturen. Op het testsysteem met één gigabyte fysiek geheugen werd succes behaald met de volgende opties, die in het bestand /boot/loader.conf geplaatst dienen te worden, waarna het systeem opnieuw moet worden opgestart:

```
vm.kmem_size="330M"
vm.kmem_size_max="330M"
vfs.zfs.arc_max="40M"
```

```
vfs.zfs.vdev.cache.size="5M"
```

Zie voor een meer gedetailleerde lijst van aanbevelingen aangaande ZFS-afstelling: http://wiki.freebsd.org/ZFSTuningGuide .

19.2.2. Gebruik maken van ZFS

Er is een opstartmechanisme dat FreeBSD in staat stelt om ZFS pools te mounten tijdens initialisatie van het systeem. Voer de volgende commando's uit om dit in te stellen:

```
# echo 'zfs_enable="YES"' >> /etc/rc.conf
# service zfs start
```

In het resterende deel van dit document wordt aangenomen dat er drie SCSI-schijven beschikbaar zijn, en dat hun apparaatnamen respectievelijk da0, da1 en da2 zijn. Gebruikers van IDE-hardware kunnen de ad apparaten gebruiken in plaats van SCSI-apparaten.

Een pool op een enkele schijf

Voer het commando `zpool` uit om een simpele, niet-redundante ZFS-pool op een enkele schijf aan te maken:

```
# zpool create example /dev/da0
```

Bestudeer de uitvoer van het commando `df` om de nieuwe pool te zien:

```
# df
Filesystem  1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a   2026030  235230  1628718    13%    /
devfs               1       1        0   100%    /dev
/dev/ad0s1d  54098308 1032846 48737598     2%    /usr
example      17547136       0 17547136     0%    /example
```

In deze uitvoer wordt duidelijk dat de `example`-pool niet alleen is aangemaakt, maar ook direct *gemount* is. Hij is ook toegankelijk, net als een gewoon bestandssysteem; er kunnen bestanden op worden aangemaakt en gebruikers kunnen er op rondkijken zoals in het volgende voorbeeld:

```
# cd /example
# ls
# touch testfile
# ls -al
total 4
drwxr-xr-x   2 root  wheel    3 Aug 29 23:15 .
drwxr-xr-x  21 root  wheel  512 Aug 29 23:12 ..
-rw-r--r--   1 root  wheel    0 Aug 29 23:15 testfile
```

Helaas benut deze pool nog geen ZFS-mogelijkheden. Maak een bestandssysteem aan op deze pool en activeer er compressie op:

```
# zfs create example/compressed
# zfs set compression=gzip example/compressed
```

`example/compressed` is nu een gecomprimeerd ZFS-bestandssysteem. Probeer er een paar grote bestanden naartoe te kopiëren door ze naar /example/compressed te kopiëren.

De compressie kan nu worden uitgeschakeld met:

```
# zfs set compression=off example/compressed
```

Voer het volgende commando uit om het bestandssysteem te unmounten, en controleer dat daarna met `df`:

```
# zfs umount example/compressed
# df
Filesystem  1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a   2026030  235232  1628716    13%    /
devfs               1       1        0   100%    /dev
/dev/ad0s1d  54098308 1032864 48737580     2%    /usr
example      17547008       0 17547008     0%    /example
```

Mount het bestandssysteem opnieuw om het weer toegankelijk te maken en controleer met `df`:

```
# zfs mount example/compressed
# df
Filesystem         1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a          2026030  235234  1628714    13%    /
devfs                      1       1        0   100%    /dev
/dev/ad0s1d         54098308 1032864 48737580     2%    /usr
example             17547008       0 17547008     0%    /example
example/compressed  17547008       0 17547008     0%    /example/compressed
```

De pool en het bestandssysteem zijn ook zichtbaar in de uitvoer van `mount`:

```
# mount
/dev/ad0s1a on / (ufs, local)
devfs on /dev (devfs, local)
/dev/ad0s1d on /usr (ufs, local, soft-updates)
example on /example (zfs, local)
example/data on /example/data (zfs, local)
example/compressed on /example/compressed (zfs, local)
```

Zoals is te zien kunnen ZFS-bestandssystemen, nadat ze zijn gecreëerd, net als gewone bestandssystemen worden gebruikt; er zijn echter ook vele andere mogelijkheden beschikbaar. In het volgende voorbeeld wordt er een nieuw bestandssysteem `data` gecreëerd. Er zullen belangrijke bestanden op worden bewaard, dus het bestandssysteem wordt zodanig ingesteld dat het twee kopieën van ieder gegevensblok opslaat:

```
# zfs create example/data
# zfs set copies=2 example/data
```

Het is nu mogelijk om het gegevens- en ruimtegebruik te bekijken door `df` opnieuw te draaien:

```
# df
Filesystem         1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a          2026030  235234  1628714    13%    /
devfs                      1       1        0   100%    /dev
/dev/ad0s1d         54098308 1032864 48737580     2%    /usr
example             17547008       0 17547008     0%    /example
example/compressed  17547008       0 17547008     0%    /example/compressed
example/data        17547008       0 17547008     0%    /example/data
```

Merk op dat ieder bestandssysteem in de pool dezelfde hoeveelheid vrije ruimte heeft. Dit is de reden dat `df` steeds wordt gebruikt tussen de voorbeelden door, om te laten zien dat de bestandssystemen slechts zoveel ruimte gebruiken als ze nodig hebben en allemaal putten uit dezelfde pool. Het ZFS bestandssysteem elimineert concepten als volumes en partities, en staat verschillende bestandssystemen toe om in dezelfde pool te bestaan. Verwijder nu de bestandssystemen en verwijder daarna de pool, omdat deze niet meer nodig zijn:

```
# zfs destroy example/compressed
# zfs destroy example/data
# zpool destroy example
```

Schijven gaan slechter werken en begeven het, een onvermijdelijke eigenschap. Wanneer de schijf stukgaat zullen de gegevens verloren gaan. Een methode om gegevensverlies ten gevolge van een kapotte harde schijf te vermijden is het implementeren van RAID. ZFS ondersteunt deze mogelijkheid in zijn pool-ontwerp en wordt beschreven in de volgende sectie.

ZFS RAID-Z

Zoals eerder opgemerkt wordt in deze sectie aangenomen dat er drie SCSI-schijven bestaan als de apparaten da0, da1 en da2 (of ad0 en hoger als IDE-schijven worden gebruikt). Voer het volgende commando uit om een RAID-Z-pool te creëren:

```
# zpool create storage raidz da0 da1 da2
```



Sun™ raadt aan om tussen de drie en negen schijven te gebruiken voor een RAID-

> Z-configuratie. Overweeg, als u een enkele pool met 10 of meer schijven nodig heeft, om deze te splitsen in kleine RAID-Z-groepen. Overweeg, als u slechts twee schijven heeft en nog steeds redundantie nodig heeft, om in plaats hiervan een ZFS-spiegel te gebruiken. Bekijk de handleidingpagina zpool(8) voor meer details.

De `storage` zpool zou gecreëerd moeten zijn. Dit kan worden geverifieerd met de mount(8) en df(1) commando's zoals eerder. Er kunnen meer schijfapparaten worden toegewezen door ze aan het einde van de bovenstaande lijst toe te voegen. Maak een nieuw bestandssysteem in de pool, genaamd `home`, waar op den duur de gebruikersbestanden geplaatst zullen worden:

```
# zfs create storage/home
```

Het is nu mogelijk om compressie in te schakelen en extra kopieën te bewaren van de gebruikersmappen en -bestanden. Dit kan net als eerder worden bewerkstelligd door de volgende commando's uit te voeren:

```
# zfs set copies=2 storage/home
# zfs set compression=gzip storage/home
```

Kopieer, om dit als de nieuwe home-map voor gebruikers in te stellen, de gebruikersgegevens naar deze map en creëer de benodigde links:

```
# cp -rp /home/* /storage/home
# rm -rf /home /usr/home
# ln -s /storage/home /home
# ln -s /storage/home /usr/home
```

De gebruikersgegevens zouden nu op het nieuw aangemaakte /storage/home bestandssysteem moeten staan. Test dit door een nieuwe gebruiker aan te maken en daarmee in te loggen.

Probeer een snapshot te maken dat later weer hersteld kan worden:

```
# zfs snapshot storage/home@08-30-08
```

Merk op dat de snapshot-optie alleen een echt bestandssysteem vastlegt, geen mappen of bestanden. Het `@`-karakter wordt gebruikt als scheidingsteken tussen de naam van het bestandssysteem of de naam van het volume. Wanneer de home-map van een gebruiker wordt weggegooid, kan deze worden hersteld met:

```
# zfs rollback storage/home@08-30-08
```

Voer `ls` in de .zfs/snapshot directory van het bestandssysteem uit om een lijst van alle beschikbare snapshots te krijgen. Voer, om bijvoorbeeld het zojuist gemaakte snapshot te zien, het volgende commando uit:

```
# ls /storage/home/.zfs/snapshot
```

Het is mogelijk om een script te schrijven dat maandelijks een snapshot van de gebruikersgegevens maakt; na verloop van tijd kunnen snapshots echter een grote hoeveelheid schrijfruimte in beslag nemen. Het vorige snapshot kan worden verwijderd met het volgende commando:

```
# zfs destroy storage/home@08-30-08
```

Na al dit testen is er geen reden om /storage/home in zijn huidige staat nog te bewaren. Maak er het echte /home bestandssysteem van:

```
# zfs set mountpoint=/home storage/home
```

Het uitvoeren van de commando's `df` en `mount` laat zien dat het systeem ons bestandssysteem nu als de echte /home behandelt:

```
# mount
/dev/ad0s1a on / (ufs, local)
devfs on /dev (devfs, local)
/dev/ad0s1d on /usr (ufs, local, soft-updates)
storage on /storage (zfs, local)
storage/home on /home (zfs, local)
# df
Filesystem   1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a    2026030  235240  1628708    13%    /
devfs                1       1        0   100%    /dev
/dev/ad0s1d   54098308 1032826 48737618     2%    /usr
storage       26320512       0 26320512     0%    /storage
storage/home  26320512       0 26320512     0%    /home
```

Hiermee is de RAID-Z configuratie compleet. Voer het volgende commando uit om status-updates van de gecreëerde bestandssystemen te krijgen tijdens het draaien van de nachtelijke periodic(8):

```
# echo 'daily_status_zfs_enable="YES"' >> /etc/periodic.conf
```

Het herstellen van RAID-Z

Iedere software-RAID heeft een methode om zijn `status` te inspecteren. ZFS is geen uitzondering. De status van RAID-Z-apparaten kan worden geïnspecteerd met het volgende commando:

```
# zpool status -x
```

Als alle pools in orde zijn en alles is normaal, dan wordt het volgende bericht weergegeven:

```
all pools are healthy
```

Als er een probleem is, misschien een schijf die offine is gegaan, dan wordt de status van de pool weergegeven en dat zal er als volgt uitzien:

```
  pool: storage
 state: DEGRADED
status: One or more devices has been taken offline by the administrator.
    Sufficient replicas exist for the pool to continue functioning in a
    degraded state.
action: Online the device using 'zpool online' or replace the device with
    'zpool replace'.
 scrub: none requested
config:

    NAME        STATE     READ WRITE CKSUM
    storage     DEGRADED     0     0     0
      raidz1    DEGRADED     0     0     0
        da0     ONLINE       0     0     0
        da1     OFFLINE      0     0     0
        da2     ONLINE       0     0     0

errors: No known data errors
```

Hier staat dat het apparaat offline is gezet door de beheerder. Dat is waar voor dit specifieke voorbeeld. Om de schijf offline te zetten werd het volgende commando gebruikt:

```
# zpool offline storage da1
```

Het is nu mogelijk om de schijf da1 te vervangen nadat het systeem uitgeschakeld is. Zodra het systeem weer opgestart is, kan het volgende commando worden uitgevoerd om de schijf te vervangen:

```
# zpool replace storage da1
```

Nu kan de status opnieuw geïnspecteerd worden, dit keer zonder de `-x` vlag, om de statusinformatie op te vragen:

```
# zpool status storage
 pool: storage
 state: ONLINE
 scrub: resilver completed with 0 errors on Sat Aug 30 19:44:11 2008
config:

    NAME        STATE     READ WRITE CKSUM
```

```
    storage     ONLINE       0     0     0
      raidz1    ONLINE       0     0     0
        da0     ONLINE       0     0     0
        da1     ONLINE       0     0     0
        da2     ONLINE       0     0     0

errors: No known data errors
```

Zoals te zien in dit voorbeeld lijkt alles normaal te zijn.

Gegevensverificatie

Zoals eerder opgemerkt gebruikt ZFS `checksums` om de integriteit van opgeslagen gegevens te verifiëren. Ze worden automatisch ingeschakeld bij het creëeren van bestandssystemen en kunnen worden uitgeschakeld door middel van het volgende commando:

```
# zfs set checksum=off storage/home
```

Dit is echter geen verstandig idee, omdat checksums zeer weinig opslagruimte innemen en nuttiger zijn wanneer ze zijn ingeschakeld. Het lijkt daarnaast ook geen merkbare invloed op de prestaties te hebben wanneer ze zijn ingeschakeld. Wanneer ze aanstaan is het mogelijk om ZFS gegevensintegriteit te laten controleren door middel van checksum-verificatie. Dit proces staat bekend als "scrubbing". Voer het volgende commando uit om de gegevensintegriteit van de `storage` -pool te controleren:

```
# zpool scrub storage
```

Dit proces kan, afhankelijk van de hoeveelheid opgeslagen gegevens, een aanzienlijke hoeveelheid tijd in beslag nemen. Het is daarnaast ook zeer I/O-intensief, zozeer dat slechts één van deze operaties tegelijkertijd uitgevoerd kan worden. Nadat de scrub is voltooid wordt de status bijgewerkt en kan deze worden bekeken door een statusaanvraag te doen:

```
# zpool status storage
 pool: storage
 state: ONLINE
 scrub: scrub completed with 0 errors on Sat Aug 30 19:57:37 2008
config:

    NAME        STATE     READ WRITE CKSUM
    storage     ONLINE       0     0     0
      raidz1    ONLINE       0     0     0
        da0     ONLINE       0     0     0
        da1     ONLINE       0     0     0
        da2     ONLINE       0     0     0

errors: No known data errors
```

De voltooiingstijd is in dit voorbeeld duidelijk zichtbaar. Deze eigenschap helpt om gegevensintegriteit te garanderen gedurende een langere tijdsperiode.

Er zijn vele andere opties voor het Z-bestandssysteem, zie de handleidingpagina's zfs(8) en zpool(8).

ZFS quota

ZFS ondersteunt verschillende soorten quota: de refquota, de algemene quota, de gebruikersquota en de groepsquota. Deze sectie legt de beginselen van ieder van deze uit en bevat wat instructies voor gebruik.

Quota beperken de hoeveelheid ruimte die een gegevensverzameling en zijn afstammelingen kunnen gebruiken en dwingen een limiet af op de hoeveelheid ruimte dat gebruikt wordt door bestandssystemen en snapshots voor deze afstammelingen. Vanuit gebruikers zijn quota handig om de hoeveelheid ruimte die een bepaalde gebruiker kan gebruiken te beperken.



Quota kunnen niet op volumes worden ingesteld, aangezien de eigenschap `volsize` als een impliciet quotum optreedt.

De refquota, `refquota=grootte`, beperkt de hoeveelheid ruimte die een gegevensverzameling in beslag kan nemen door een harde grens aan de gebruikte ruimte te stellen. Deze harde grens bevat echter niet de ruimte gebruikt door afstammelingen, zoals bestandssystemen of snapshots.

Gebruik het volgende om een algemeen quotum van 10 GB voor /home/storage/bob af te dwingen:

```
# zfs set quota=10G storage/home/bob
```

Gebruikersquota beperken de hoeveelheid ruimte die door de aangegeven gebruiker kan worden gebruikt. Het algemene formaat is `userquota@gebruiker=grootte` waarbij de gebruikersnaam in één van de volgende formaten dient te zijn:

- Naam compatibel met POSIX (bijvoorbeeld *jan*).
- Numeriek POSIX-ID (bijvoorbeeld *789*).
- SID-naam (bijvoorbeeld *jan.bloggs@example.com*).
- Numeriek SID-ID (bijvoorbeeld *S-1-123-456-789*).

Gebruik het volgende om bijvoorbeeld een quotum van 50 GB voor een gebruiker *jan* af te dwingen:

```
# zfs set userquota@jan=50G
```

Gebruik in plaats hiervan, om het quotum te verwijderen of er zeker van te zijn dat er geen is ingesteld:

```
# zfs set userquota@jan=none
```

Eigenschappen van gebruikersquota worden niet weergegeven door `zfs get all`. Niet-`root` gebruikers kunnen alleen hun eigen quota zien tenzij het privilege `userquota` aan ze is gegeven. Gebruikers met dit privilege kunnen ieders quota bekijken en instellen.

Groepsquota beperken de hoeveelheid ruimte die de gespecificeerde gebruikersgroep in beslag kan nemen. Het algemene formaat is `groupquota@groep=grootte`.

Gebruik om het quotum voor de groep *eerstegroep* op 50 GB in te stellen:

```
# zfs set groupquota@eerstegroep=50G
```

Gebruik in plaats hiervan, om het quotum voor de groep *eerstegroep* te verwijderen of om er voor te zorgen dat deze niet is ingesteld:

```
# zfs set groupquota@eerstegroep=none
```

Net zoals bij de eigenschappen van gebruikersquota kunnen niet-`root`-gebruikers alleen de quota zien die geassocieerd zijn met de gebruikersgroepen waar ze bij horen, een `root`-gebruiker of een gebruiker met het privilege `groupquota` kan alle quota voor alle groepen bekijken en instellen.

Het deelcommando `zfs userspace` geeft de hoeveelheid ruimte weer die door elke gebruiker op de snapshot van het gespecificeerde bestandssysteem in beslag wordt genomen, tezamen met alle ingestelde quota. Het deelcommando `zfs groupspace` doet hetzelfde voor groepen. Bekijk zfs(1) voor meer informatie over ondersteunde opties of het weergegeven van specifieke opties.

Gebruik het volgende om de quota voor storage/home/bob weer te geven, als u de juiste privileges heeft of `root` bent:

```
# zfs get quota storage/home/bob
```

Reserveringen in ZFS

ZFS ondersteunt twee soorten van ruimtereserveringen. Deze sectie legt de beginselen van elk van de twee uit en bevat enkele instructies voor gebruik.

De eigenschap `reservation` maakt het mogelijk om een gegarandeerde minimale hoeveelheid ruimte voor een gegevensverzameling en zijn afstammelingen te reserveren. Dit betekent dat als er een reservering van 10 GB is ingesteld voor storage/home/bob en de schijfruimte op raakt, er tenminste 10 GB aan ruimte is gereserveerd voor deze gegevensverzameling. De eigenschap `reservation` stelt de minimale hoeveelheid ruimte in die gegarandeerd is voor een gegevensverzameling exclusief afstammelingen zoals snapshots, of geeft deze aan. Als er bijvoorbeeld een snapshot is genomen van storage/home/bob moet er genoeg schijfruimte zijn buiten de `refreservation` hoeveelheid om de operatie te laten slagen omdat afstammelingen van de hoofdgegevensverzameling niet worden meegeteld in de `refreservation` hoeveelheid en dus niet stiekem de vastgestelde ruimte wijzigen.

Reserveringen kunnen in allerlei situaties nuttig zijn, bijvoorbeeld voor het plannen en testen van

de geschiktheid van het toewijzen van schijfruimte in een nieuw systeem, of om ervoor te zorgen dat er genoeg schijfruimte beschikbaar is op bestandsssystemen voor systeemherstelprocedures en bestanden.

Het algemene formaat van de eigenschap `reservation` is `reservation=grootte`, dus gebruik het onderstaande commando om een reservering van 10 GB op storage/home/bob te plaatsen:

```
# zfs set reservation=10G storage/home/bob
```

Gebruik, om te controleren of er geen reservatie is geplaatst of om een reservatie te verwijderen:

```
# zfs set reservation=none storage/home/bob
```

Het zelfde principe kan worden toegepast op de eigenschap `refreservation` om een refreservation in te stellen, met het algemene formaat `refreservation=grootte`.

Gebruik één van de volgende commando's om te kijken of er een reservatie of refreservation bestaat op storage/home/bob:

```
# zfs get reservation storage/home/bob
# zfs get refreservation storage/home/bob
```

19.3. Linux® bestandssystemen

Deze sectie beschrijft enkele van de Linux® bestandssystemen die door FreeBSD worden ondersteund.

19.3.1. Ext2FS

De kernelimplementatie van het ext2fs(5) bestandssysteem was geschreven door Godmar Back, het eerste stuurprogramma verscheen in FreeBSD 2.2. In FreeBSD 8 en eerder is de code gelicenseerd onder de GNU Public License, onder FreeBSD 9 is de code echter herschreven en nu beschikbaar onder de BSD-licentie.

Het stuurprogramma ext2fs(5) stelt de FreeBSD-kernel in staat om ext2 bestandssystemen te lezen en er naar te schrijven.

Laad ten eerste de kernelmodule:

```
# kldload ext2fs
```

Koppel daarna een ext2fs(5)-volume aan dat zich op /dev/ad1s1 bevindt:

```
# mount -t ext2fs /dev/ad1s1 /mnt
```

19.3.2. XFS

Het X-bestandssysteem, XFS, is origineel geschreven door SGI voor het besturingssysteem IRIX, ze hebben het overgebracht naar Linux®. De broncode is vrijgegeven onder de GNU Public License. Kijk op deze pagina voor meer details. De FreeBSD-port werd gestart door Russel Cattelan, Alexander Kabaev <kan@FreeBSD.org> en Craig Rodrigues <rodrigc@FreeBSD.org>.

Om XFS als een kernelmodule te laden:

```
# kldload xfs
```

Het stuurprogramma xfs(5) stelt de FreeBSD-kernel in staat om XFS-bestandssystemen te benaderen. Momenteel is echter alleen ondersteuning voor lezen aanwezig. Schrijven naar een volume is niet mogelijk.

Om een xfs(5)-volume wat op /dev/ad1s1 aan te koppelen:

```
# mount -t xfs /dev/ad1s1 /mnt
```

Merk op dat de port sysutils/xfsprogs het gereedschap `mkfs.xfs` bevat wat het mogelijk maakt om XFS-bestandssystemen aan te maken, en verder gereedschappen om ze te analyseren en repareren.

De vlag `-p` van `mkfs.xfs` kan worden gebruikt om een xfs(5)-bestandssysteem aan te maken welke bevolkt wordt met bestanden en andere meta-gegevens. Dit kan worden gebruikt om snel een alleen-lezen bestandssysteem aan te maken welke op FreeBSD getest kan worden.

19.3.3. ReiserFS

Het Reiser bestandssysteem, ReiserFS, was overgebracht naar FreeBSD door Jean-Sébastien Pédron <dumbbell@FreeBSD.org> en is vrijgegeven onder de GNU Public License.

Het stuurprogramma voor ReiserFS stelt de FreeBSD-kernel momenteel in staat om ReiserFS bestandssystemen te benaderen en hun inhoud te lezen, maar het kan ze momenteel niet beschrijven.

Laad ten eerste eerst de kernelmodule:

```
# kldload reiserfs
```

Om ten tweede een ReiserFS-volume dat zich op /dev/ad1s1 aan te koppelen:

```
# mount -t reiserfs /dev/ad1s1 /mnt
```

Hoofdstuk 20. Virtualisatie

20.1. Overzicht

Virtualisatiesoftware maakt het mogelijk om meerdere besturingssystemen gelijktijdig op dezelfde computer te draaien. Zulke softwaresystemen voor PC's gebruiken vaak een gastheer-besturingssysteem dat de virtualisatiesoftware draait en dat elk aantal gast-besturingssystemen ondersteunt.

Aan het einde van dit hoofdstuk weet de lezer:

- Het verschil tussen een gastheer-besturingssysteem en een gast-besturingssysteem.
- Hoe FreeBSD op een Intel®-gebaseerde Apple® Macintosh® computer te installeren.
- Hoe FreeBSD op Microsoft® Windows® te installeren met Virtual PC.
- Hoe een FreeBSD-systeem in te stellen voor de beste prestaties tijdens virtualisatie.

Voor het lezen van dit hoofdstuk, dient de lezer:

- De beginselen van UNIX® en FreeBSD (UNIX® beginselen) te begrijpen.
- Te weten hoe FreeBSD te installeren (FreeBSD installeren op FreeBSD 8.X en eerder).
- Te weten hoe een netwerkverbinding te installeren (Geavanceerd netwerken).
- Te weten hoe aanvullende software van derde partijen te installeren (Applicaties installeren. pakketten en ports).

20.2. FreeBSD als een gast-besturingssysteem

20.2.1. Parallels op Mac OS®

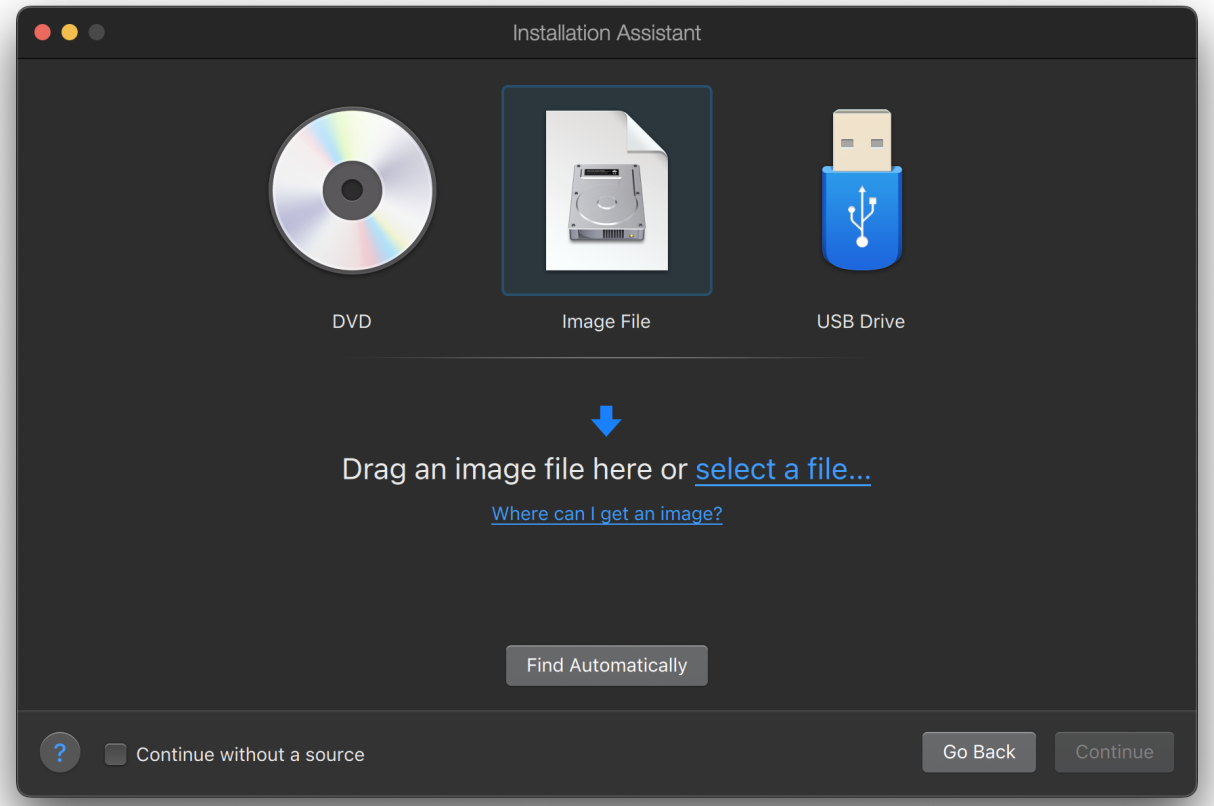
Parallels Desktop voor Mac OS® is een commercieel softwareprodukt voor Intel®-gebaseerde Apple® Mac® computers die Mac OS® 10.4.6 of nieuwer draaien. FreeBSD is een volledig ondersteund gast-besturingssysteem. Nadat Parallels is geïnstalleerd op Mac OS® X dient de gebruiker een virtuele machine in te stellen en daarna het gewenste gast-besturingssysteem te installeren.

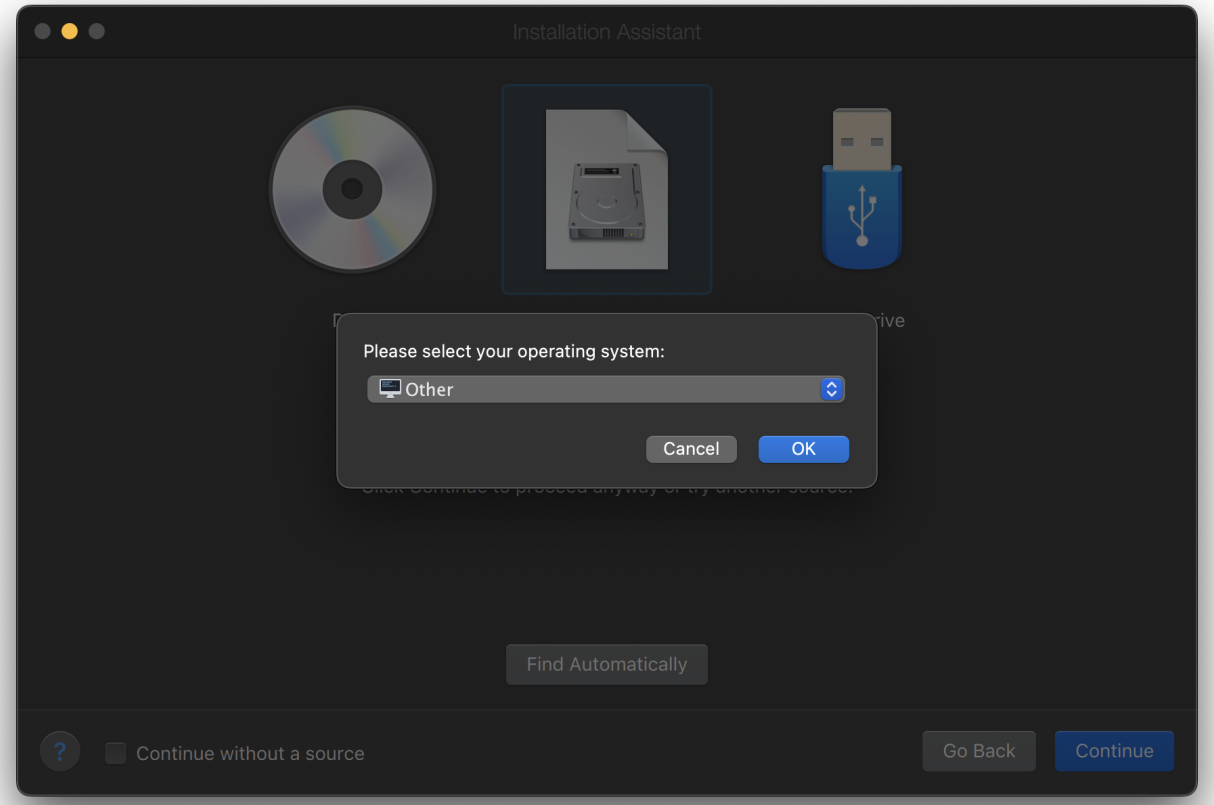
FreeBSD installeren op Parallels/Mac OS® X

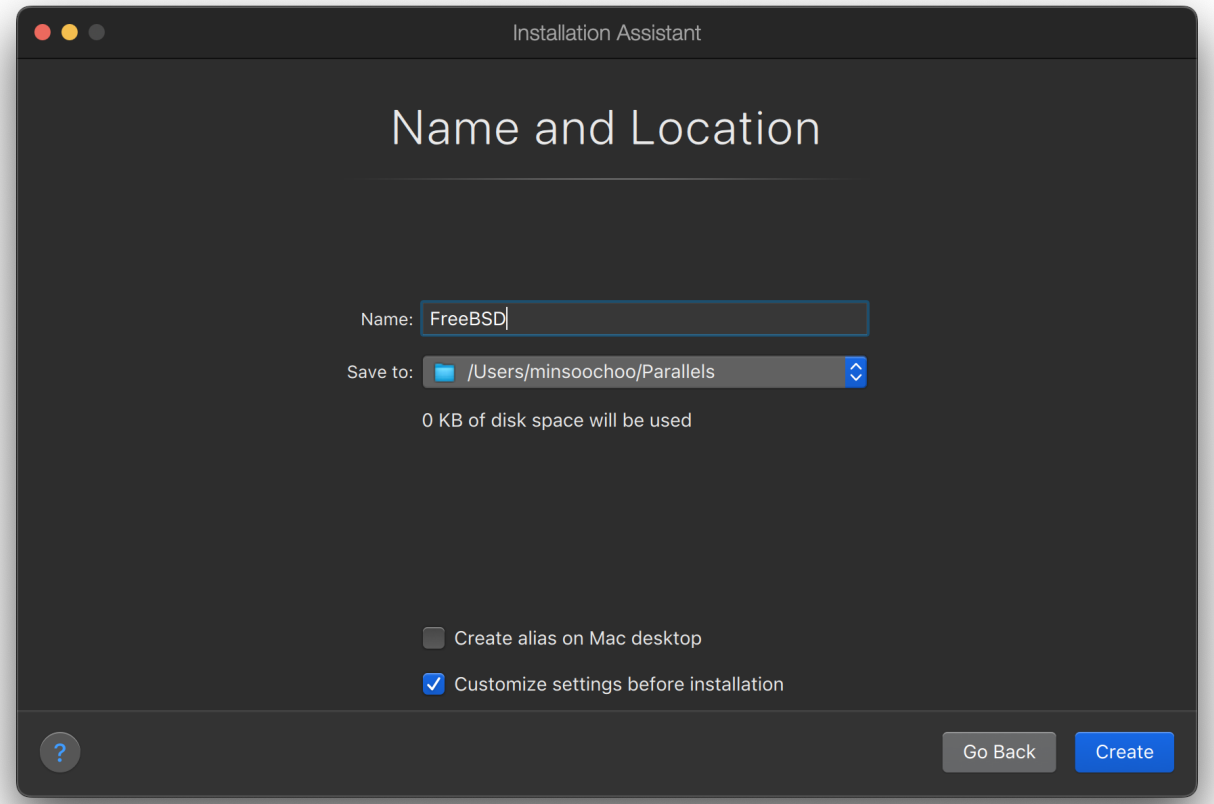
De eerste stap in het installeren van FreeBSD op Mac OS® X Parallels is het aanmaken van een nieuwe virtuele machine voor het installeren van FreeBSD. Selecteer FreeBSD als het Guest OS Type wanneer dit gevraagd wordt:

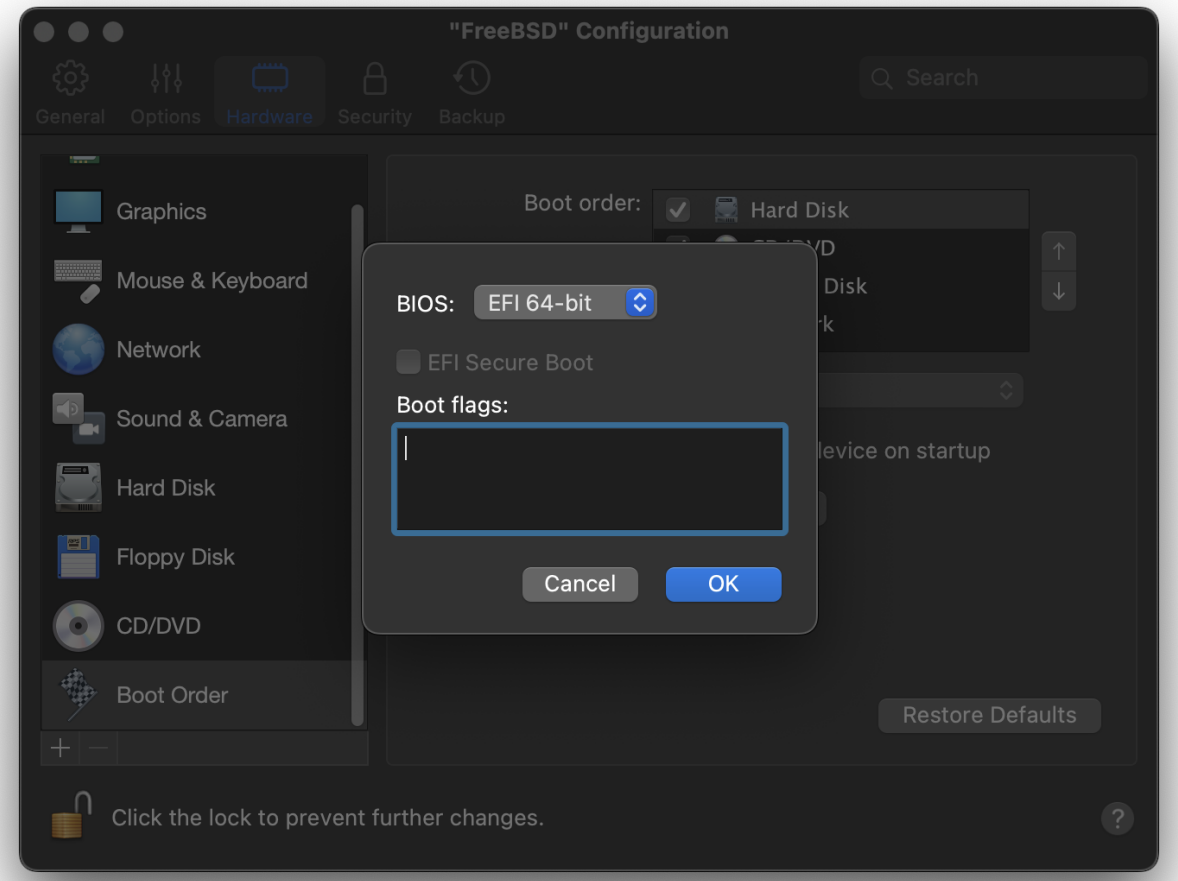


Kies verder een hoeveelheid aan schijf- en geheugenruimte afhankelijk van de plannen voor deze virtuele instantie van FreeBSD. 4GB aan schijfruimte en 512MB aan RAM werken goed voor de meeste gebruikers van FreeBSD onder Parallels:

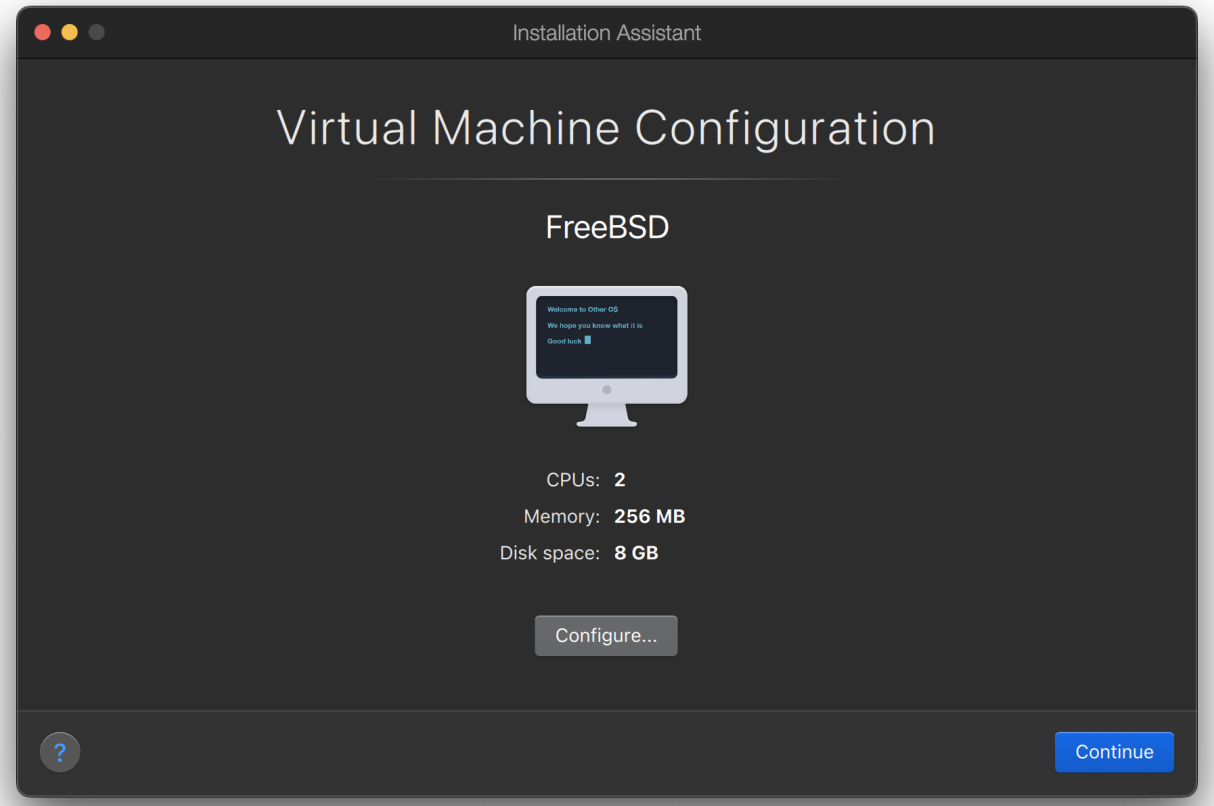

Installation Assistant
DVD
Image File
USB Drive
Drag an image file here or select a file...
Where can I get an image?
Find Automatically
Continue without a source
Go Back
Continue


Installation Assistant
Please select your operating system:
Other
Cancel
OK
Find Automatically
Continue without a source
Go Back
Continue


Installation Assistant
Name and Location
Name:
FreeBSD
Save to:
/Users/minsoochoo/Parallels
0 KB of disk space will be used
Create alias on Mac desktop
Customize settings before installation
Go Back
Create

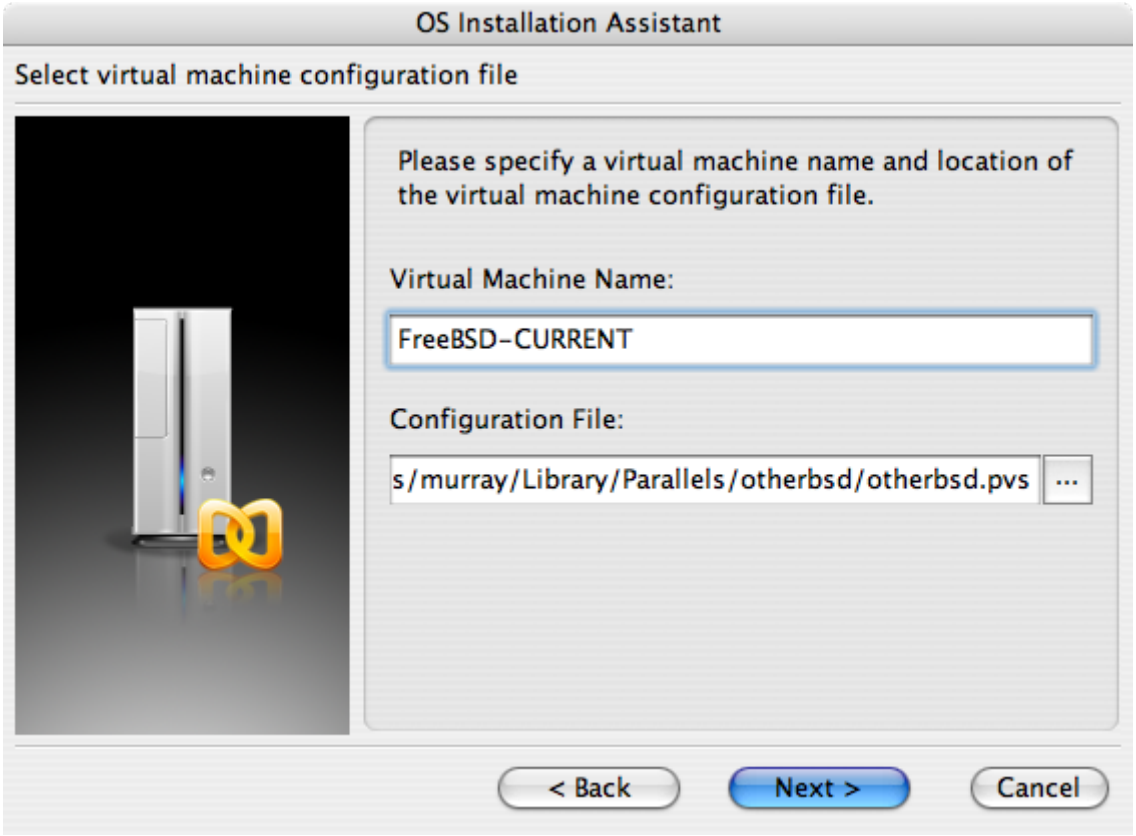


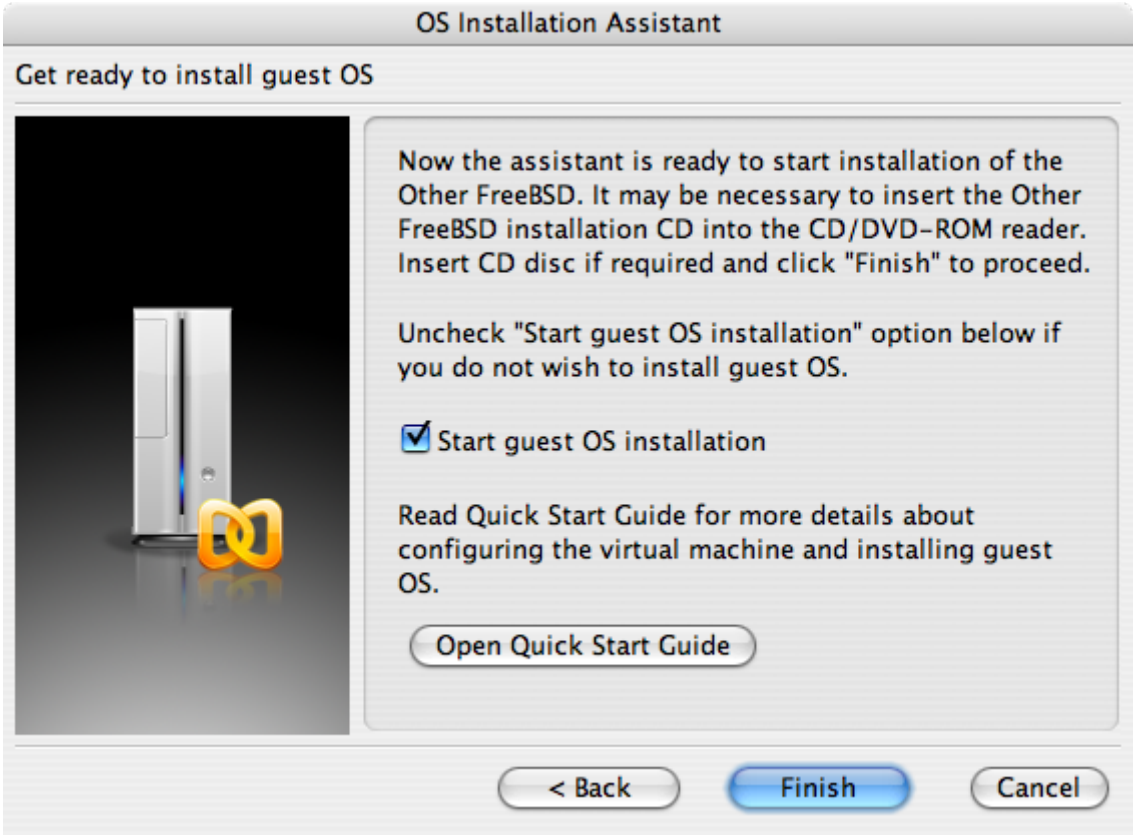
Selecteer het type netwerk en een netwerkinterface:


Installation Assistant
Virtual Machine Configuration
FreeBSD
Welcome to Other OS
We hope you know what it is
Good luck
CPUs: 2
Memory: 256 MB
Disk space: 8 GB
Configure...
Continue

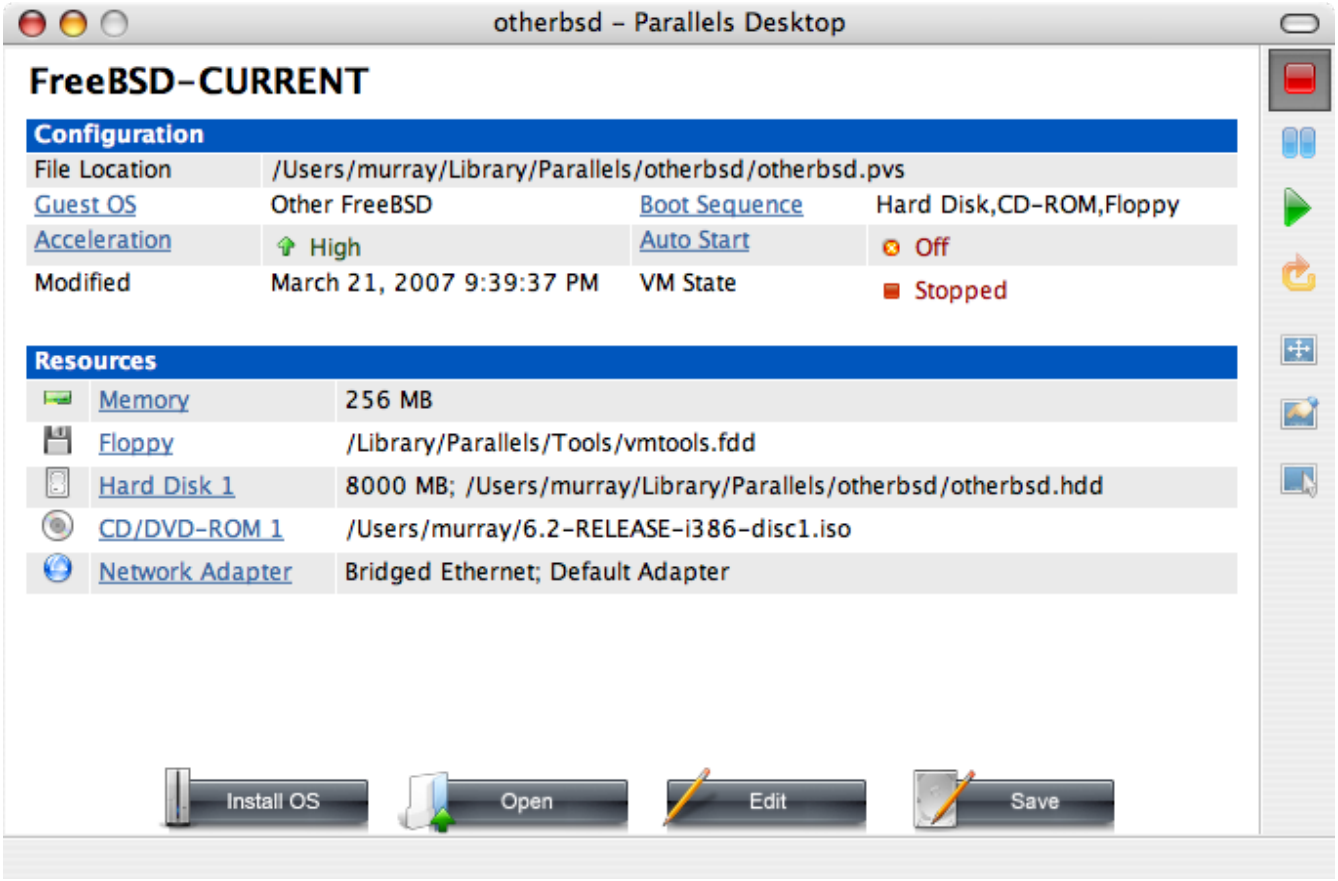


Bewaar de instellingen en sluit af:

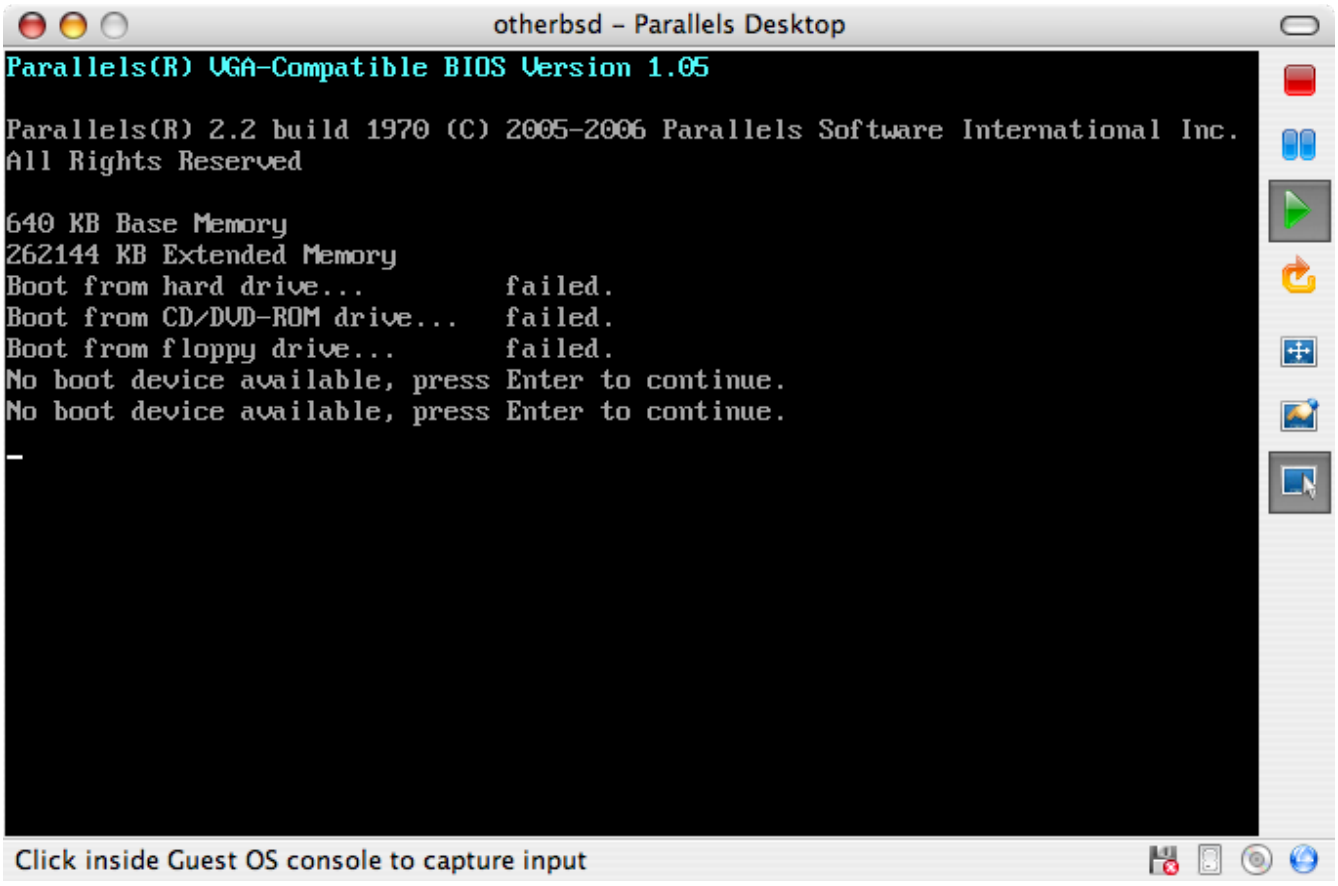




Nadat de virtuele FreeBSD-machine is aangemaakt, dient er FreeBSD op geïnstalleerd te worden. Dit gaat het beste met een officiële FreeBSD CDROM of met een ISO-beeld dat is gedownload van een officiële FTP-site. Wanneer het juiste ISO-beeld op het plaatselijke Mac®-bestandssysteem of een CDROM in de CD-drive van de Mac® aanwezig is, dient op het schijfikoon in de rechteronderhoek van het FreeBSD Parallels-scherm geklikt te worden. Dit zal een scherm tonen dat het mogelijk maakt om de CDROM-drive in de virtuele machine te associëren met een ISO-bestand op schijf of met een echte CDROM-drive.

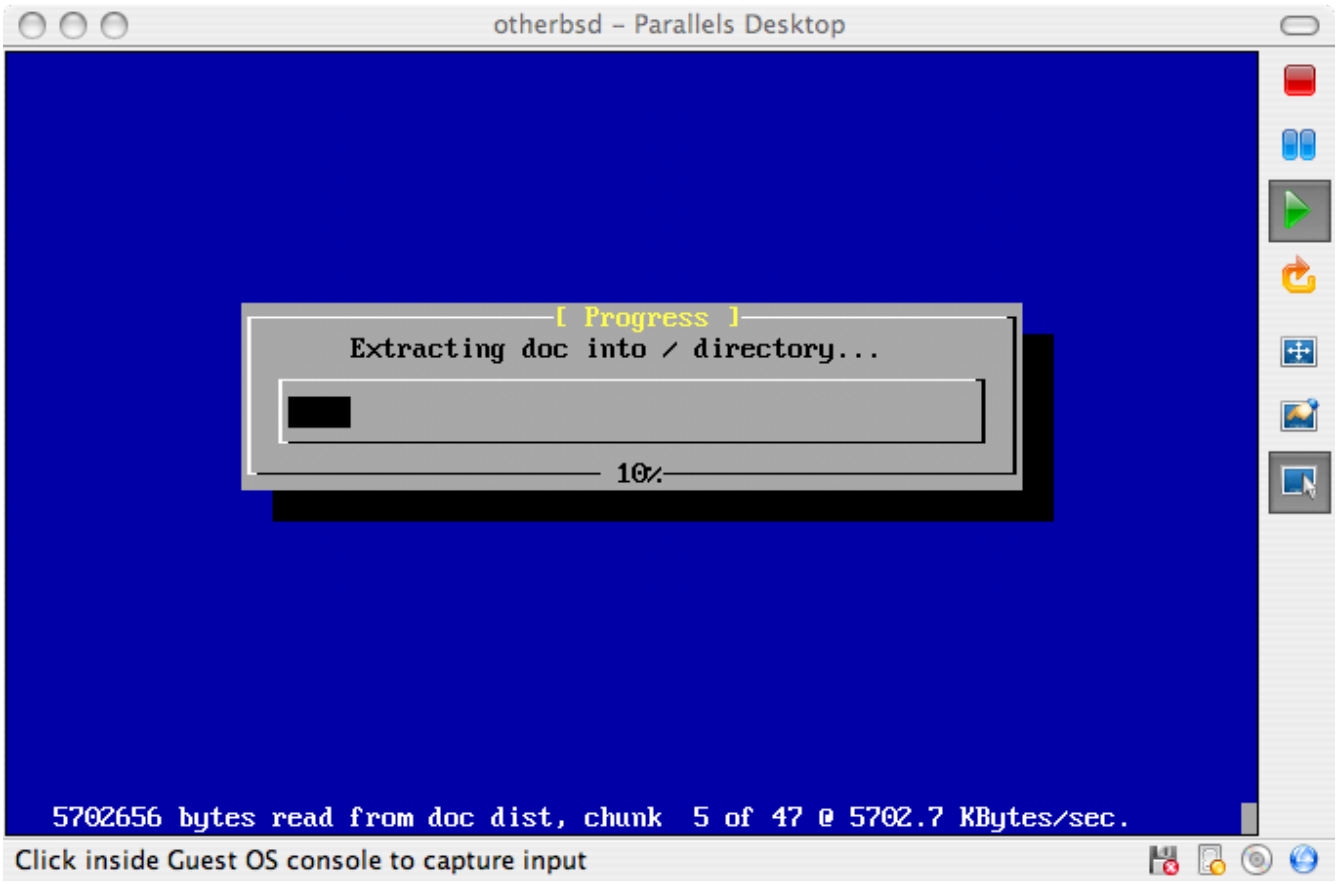


Nadat deze associatie met de CDROM-bron is gemaakt, dient de virtuele FreeBSD-machine herstart te worden door op het herstart-icoon te klikken. Parallels zal herstarten met een speciale BIOS dat eerst controleert of er een CDROM aanwezig is, net zoals een normale BIOS zou doen.

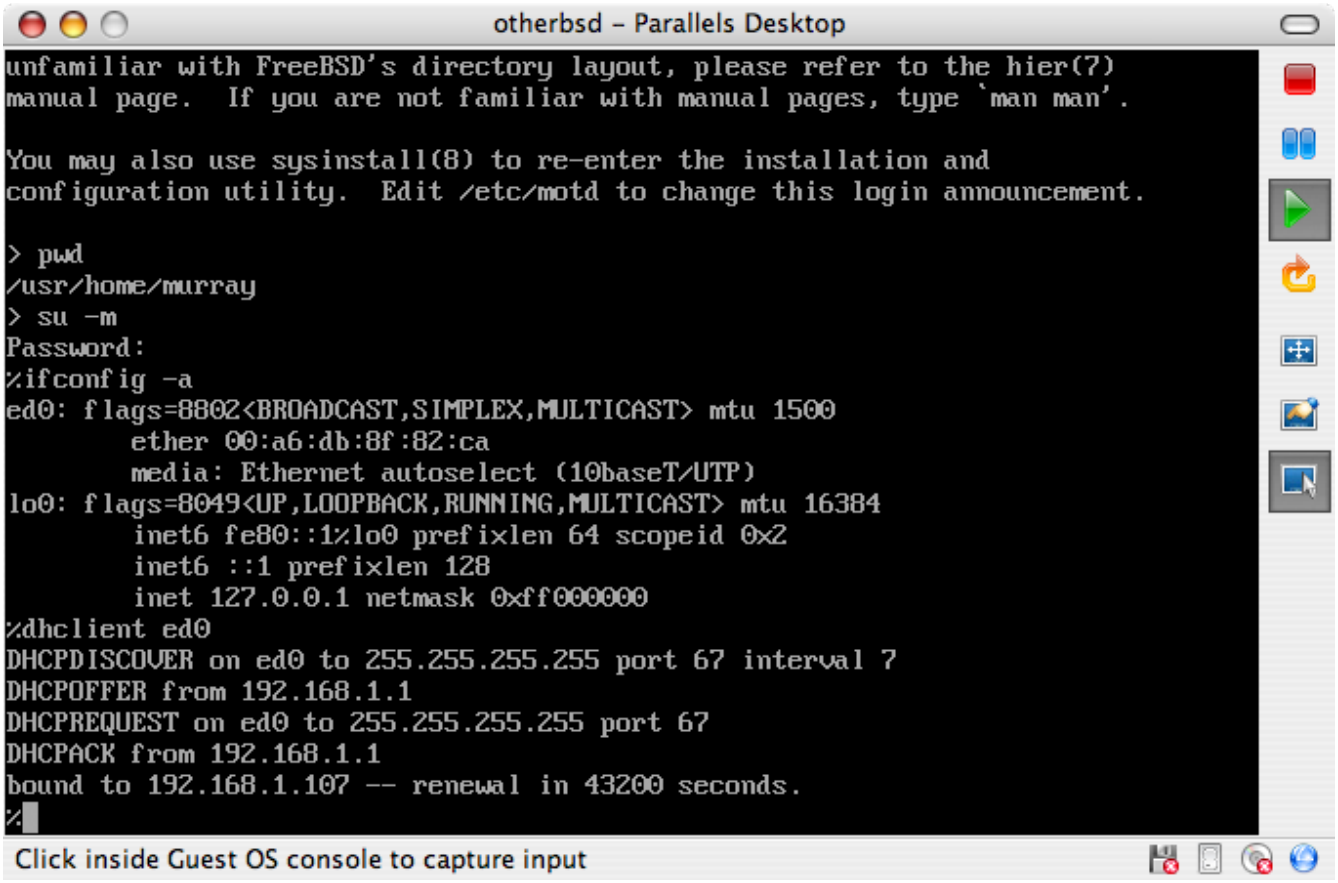


In dit geval zal het de installatiemedia van FreeBSD vinden en een normale installatie gebaseerd op

sysinstall beginnen zoals beschreven in FreeBSD installeren op FreeBSD 8.X en eerder. X11 kan nu geïnstalleerd, maar nog niet ingesteld, worden.



Nadat de installatie is voltooid, kan naar de nieuw geïnstalleerde virtuele FreeBSD-machine herstart worden.



FreeBSD instellen op Mac OS® X/Parallels

Nadat FreeBSD succesvol op Mac OS® X met Parallels is geïnstalleerd, zijn er een aantal instellingen die gewijzigd kunnen worden om het systeem voor virtuele werking te optimaliseren.

1. De variabelen voor de bootloader instellen

   De belangrijkste stap is om de tunable `kern.hz` te verlagen om het CPU-gebruik van FreeBSD onder de Parallels-omgeving te verminderen. Dit kan bereikt worden door de volgende regel aan /boot/loader.conf toe te voegen:

   ```
   kern.hz=100
   ```

   Zonder deze instelling zal een rustend FreeBSD Parallels gast-besturingssysteem ongeveer 15% van de CPU van een enkele iMac®-processor gebruiken. Na deze wijziging zal het gebruik slechts ongeveer 5% zijn.

2. Een nieuw instellingenbestand voor de kernel aanmaken

   Alle stuurprogramma's voor SCSI, FireWire, en USB kunnen verwijderd worden. Parallels biedt een virtuele netwerkadapter die door het stuurpogramma ed(4) wordt gebruikt, dus kunnen alle andere netwerkapparaten behalve ed(4) en miibus(4) uit de kernel verwijderd worden.

3. Het netwerk instellen

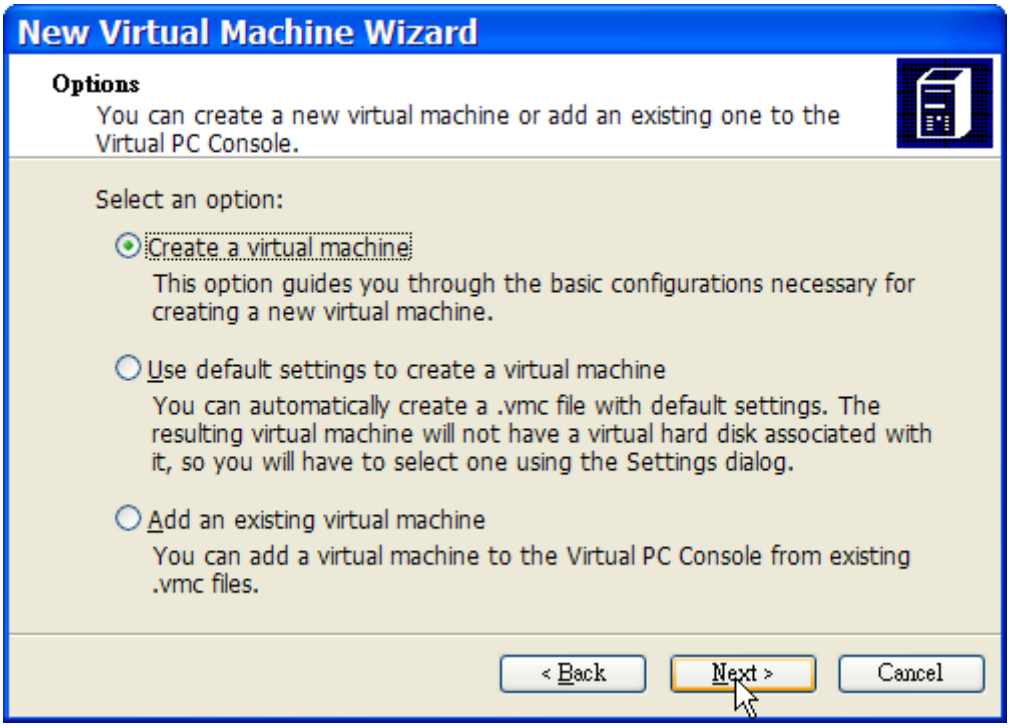
   De eenvoudigste netwerkinstallatie omvat het gebruik van DHCP om de virtuele machine met hetzelfde LAN te verbinden als het Mac®-gastheer. Dit kan bereikt worden door `ifconfig_ed0="DHCP"` aan /etc/rc.conf toe te voegen. Meer geavanceerde netwerkinstallaties staan beschreven in Netwerken voor gevorderden.

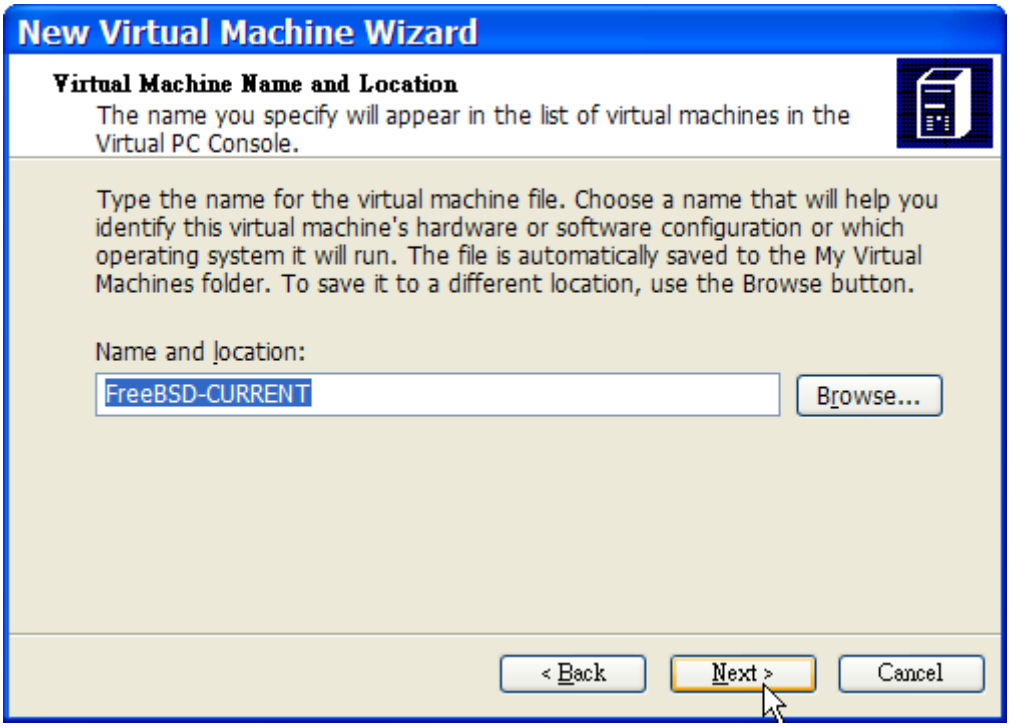
20.2.2. Virtual PC op Windows®

Virtual PC voor Windows® is een softwareprodukt van Microsoft® dat gratis gedownload kan worden . Zie systeemeisen. Nadat Virtual PC is geïnstalleerd op Microsoft® Windows®, dient de gebruiker een virtuele machine in te stellen en daarna het gewenste gast-besturingssysteem te installeren.

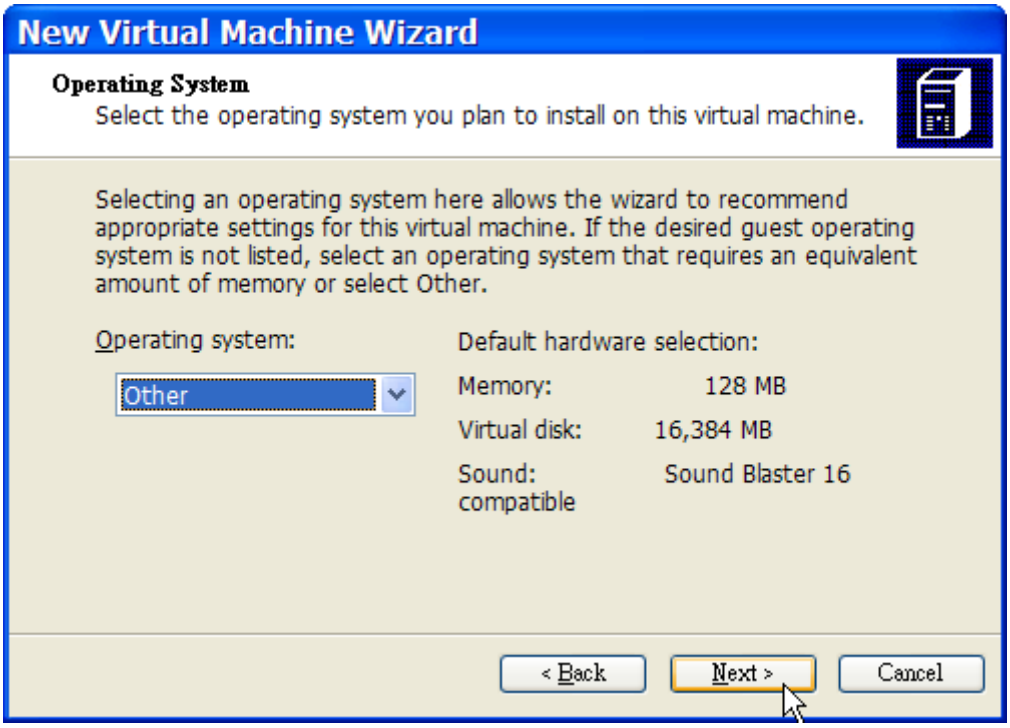
FreeBSD installeren op Virtual PC/Microsoft® Windows®

De eerste stap in het installeren van FreeBSD op Microsoft® Windows®/Virtual PC is het aanmaken van een nieuwe virtuele machine voor het installeren van FreeBSD. Kies Create a virtual machine wanneer daarom wordt gevraagd:

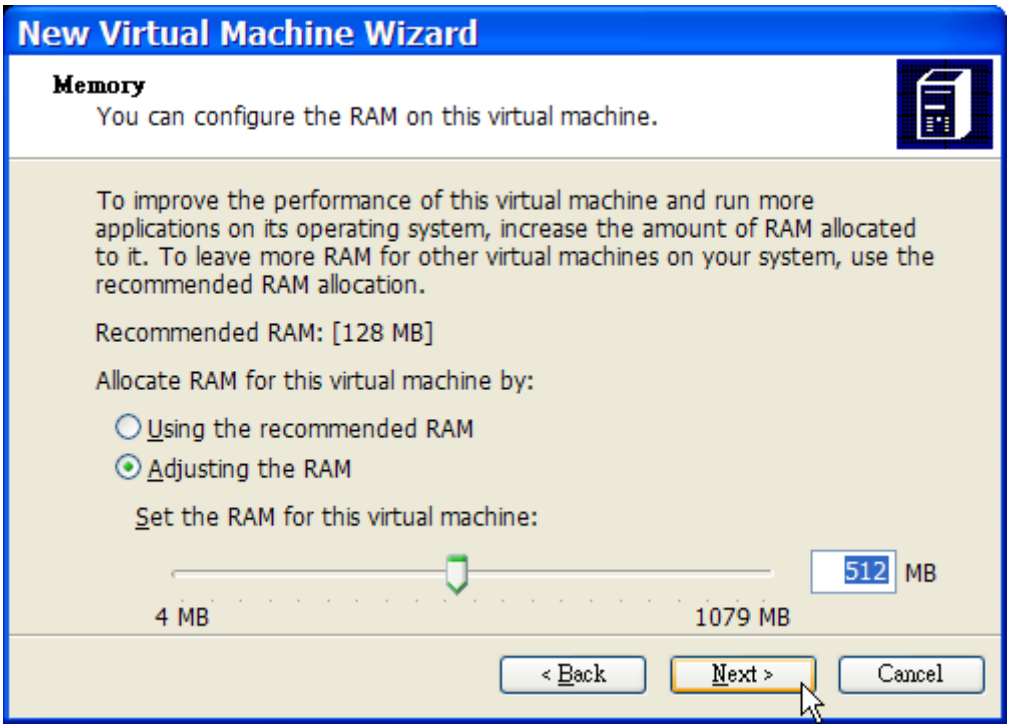


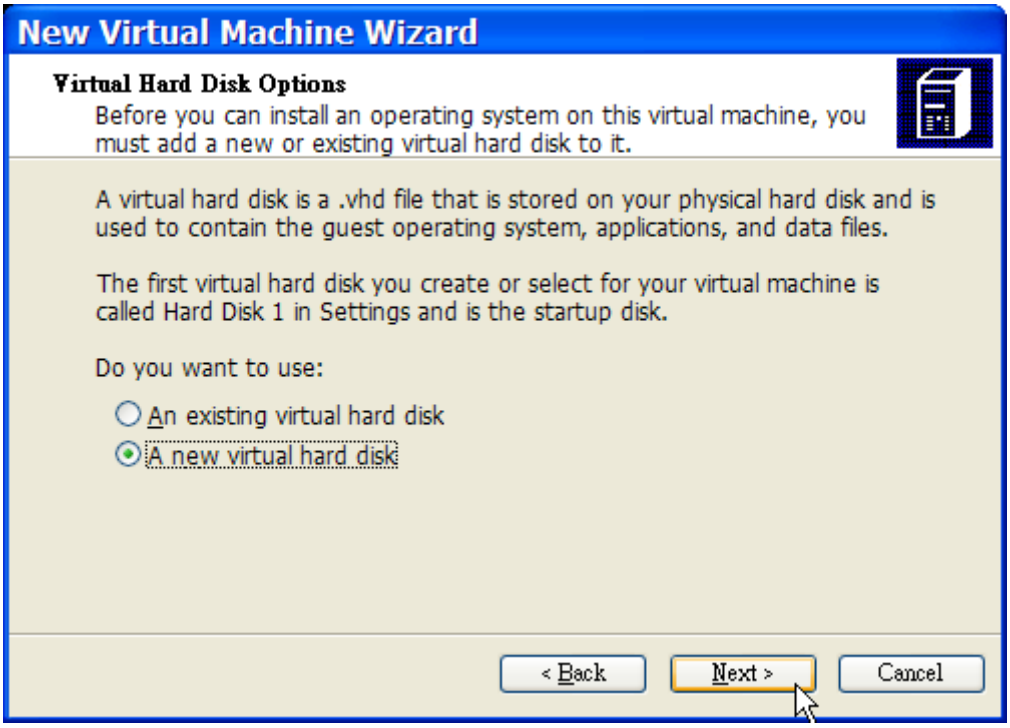


Selecteer Other als het Operating system wanneer dat gevraagd wordt:

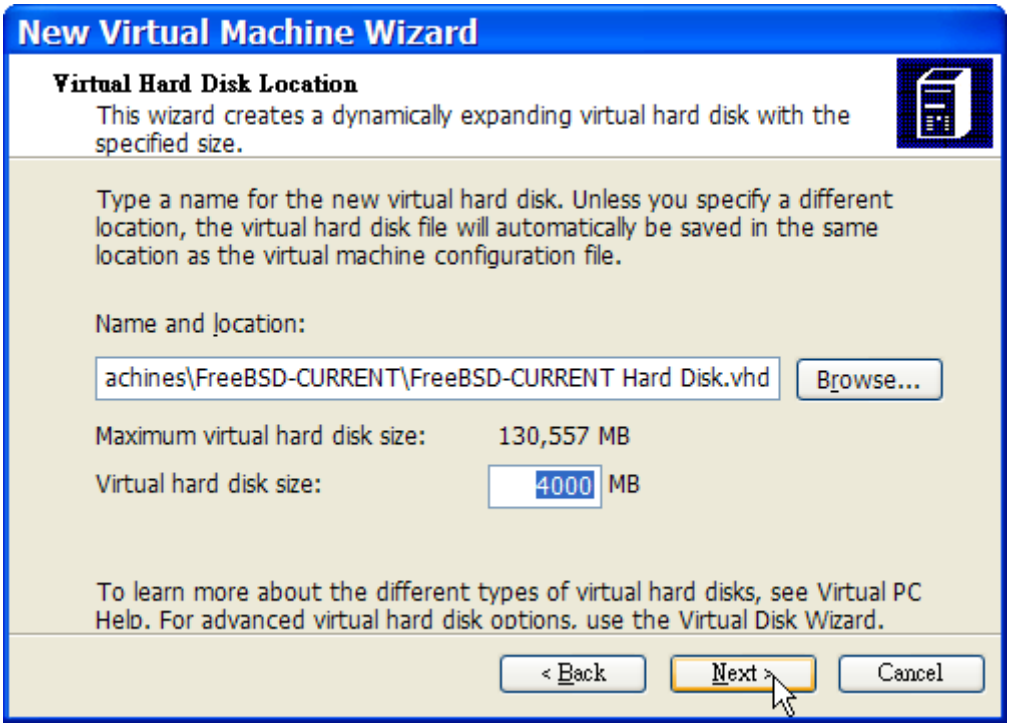


Kies vervolgens een gepaste hoeveelheid aan schijf- en geheugenruimte afhankelijk van de plannen voor deze virtuele instantie van FreeBSD. 4GB aan schijfruimte en 512MB aan RAM werken goed voor de meeste gebruikers van FreeBSD onder Virtual PC:

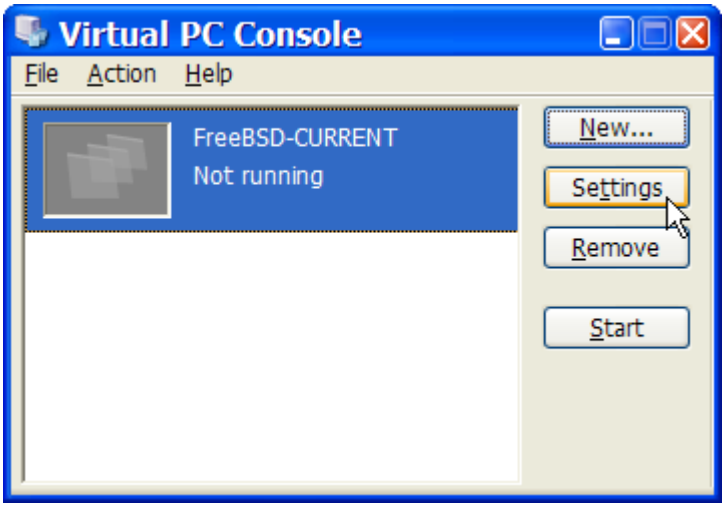


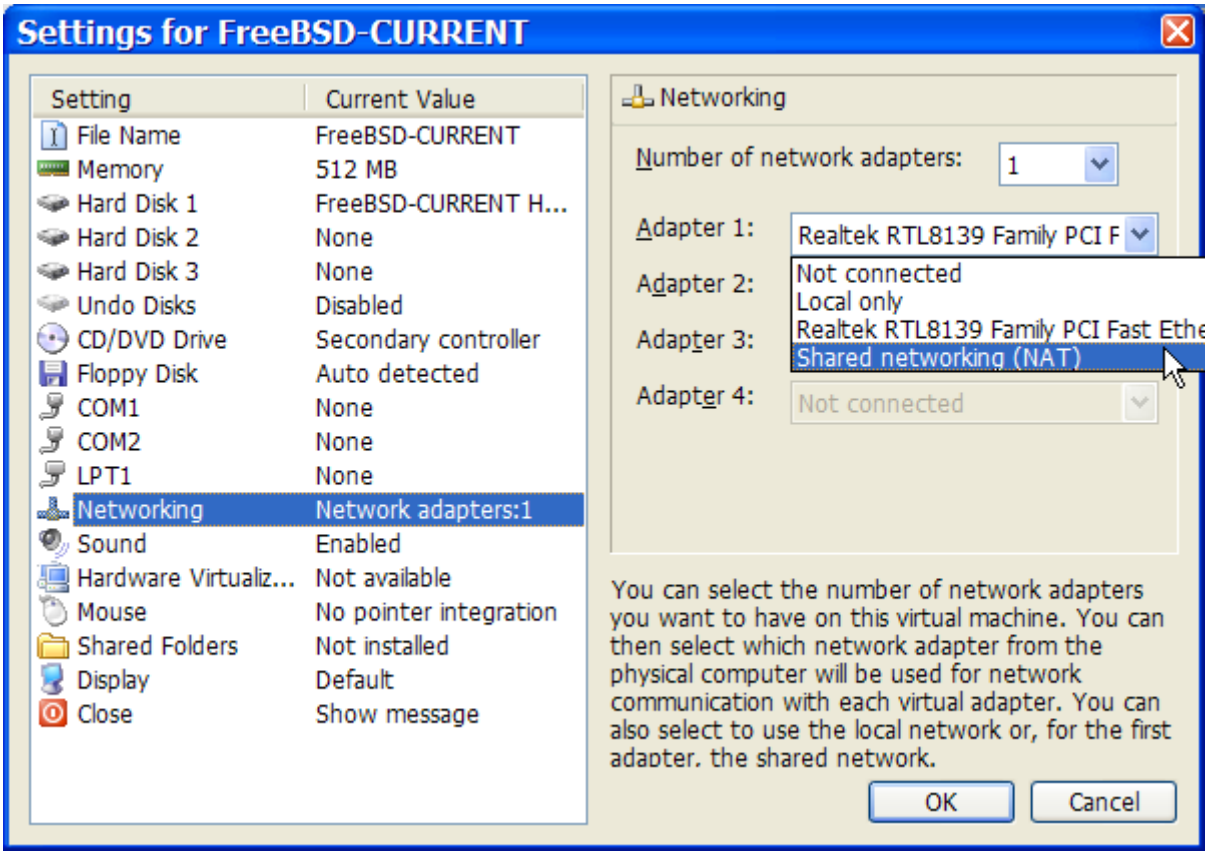


Bewaar de instellingen en sluit ze af:

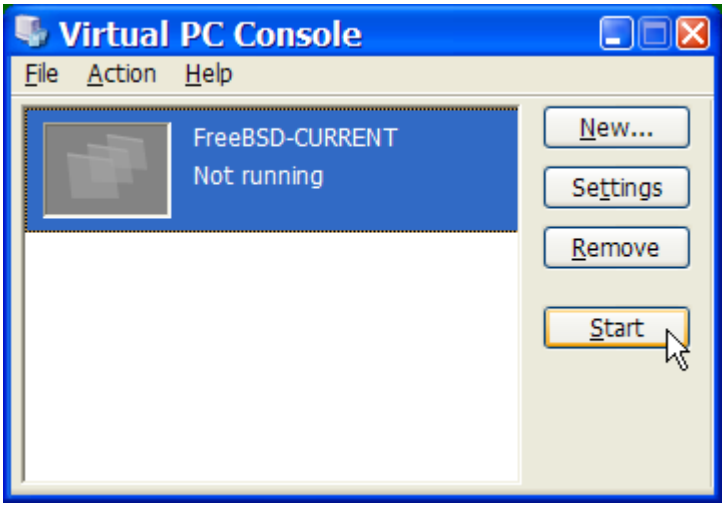


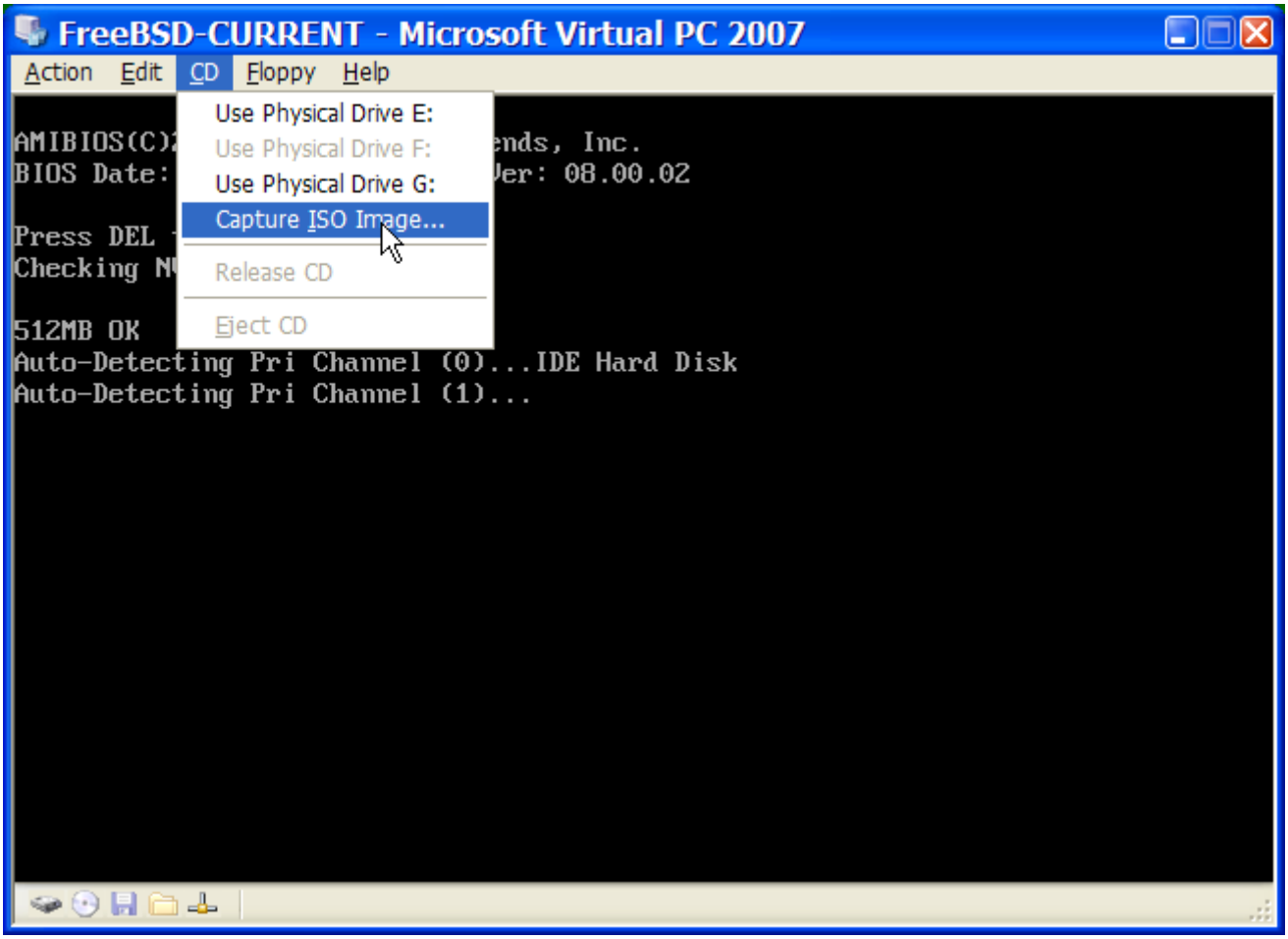
Selecteer de virtuele FreeBSD-machine en klik op **Settings**, stel daarna het type netwerk en een netwerkinterface in:



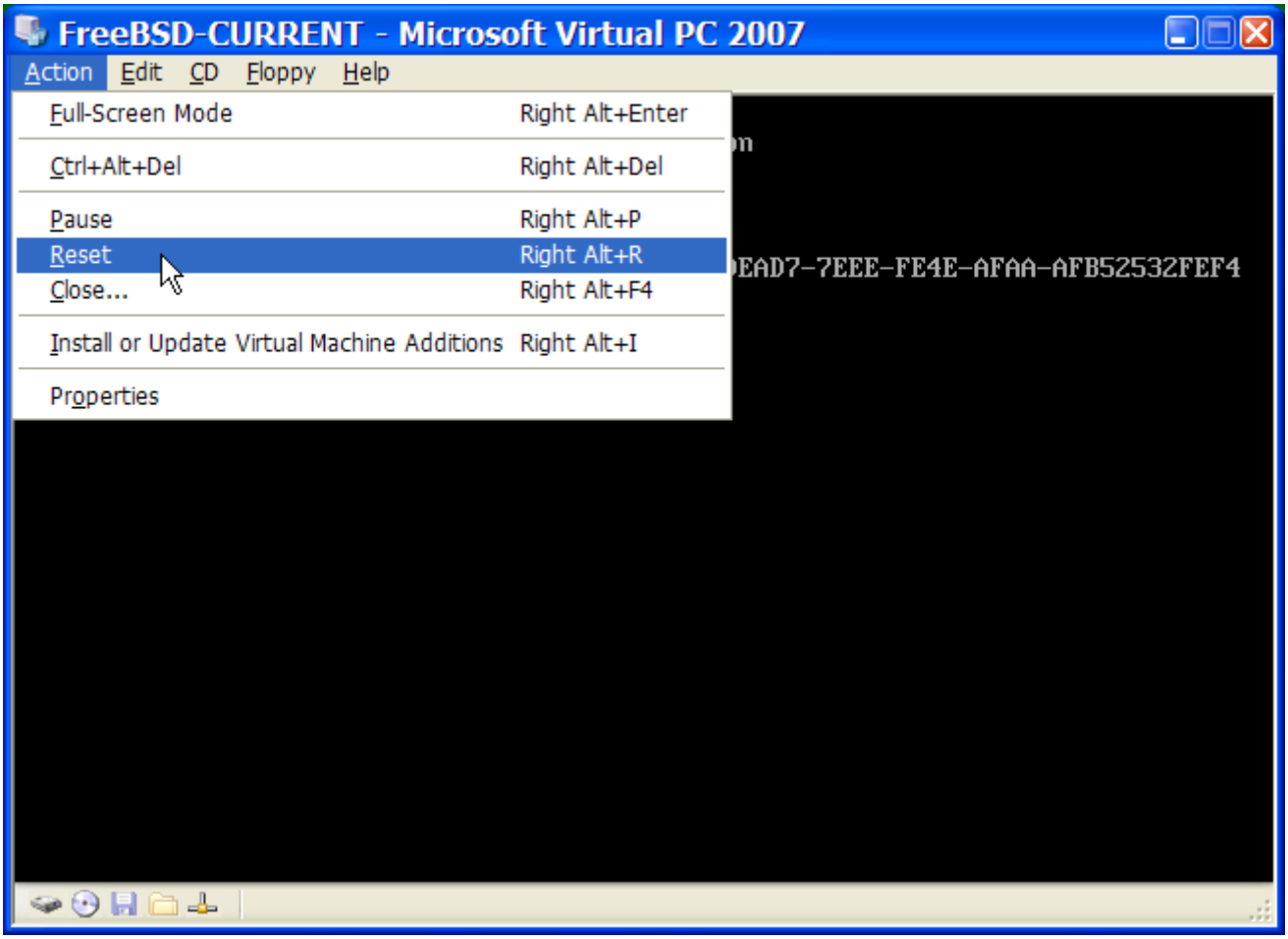


Nadat de virtuele FreeBSD-machine is aangemaakt, dient FreeBSD erop geïnstalleerd te worden. Dit gaat het beste met een officiële FreeBSD-CDROM of met een ISO-beeld dat van een officiële FTP-site is gedownload. Wanneer het juiste ISO-beeld op het lokale bestandssysteem van Windows® staat of er een CDROM in de CD-drive zit, dubbelklik dan op de virtuele FreeBSD-machine om op te starten. Klik daarna op **CD** en kies menu:Capture ISO Image... [] in het venster van Virtual PC . Dit toont een scherm dat het mogelijk maakt om de CDROM-drive in de virtuele machine te associëren met een ISO-bestand op schijf of met een echte CDROM-drive.

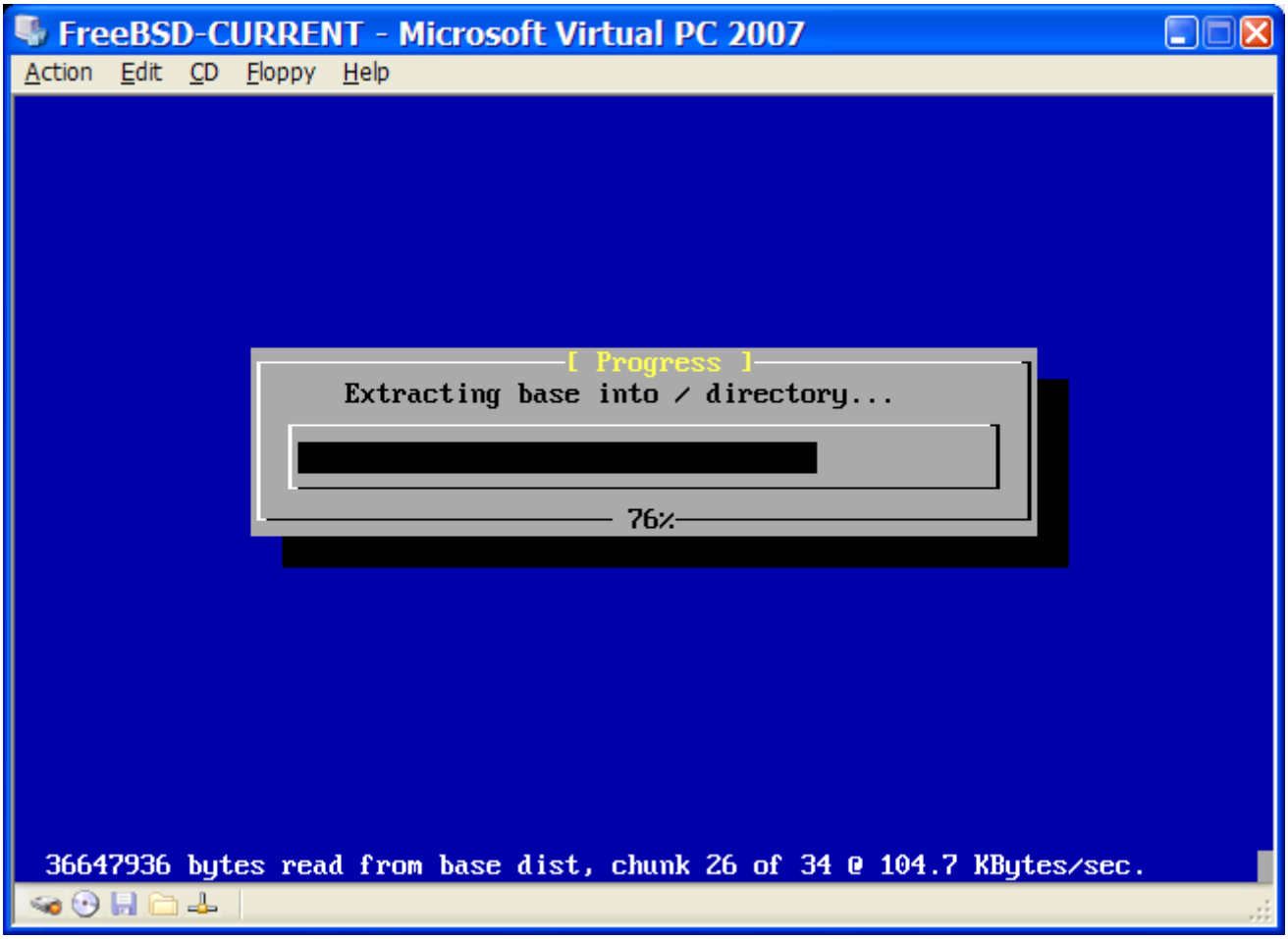




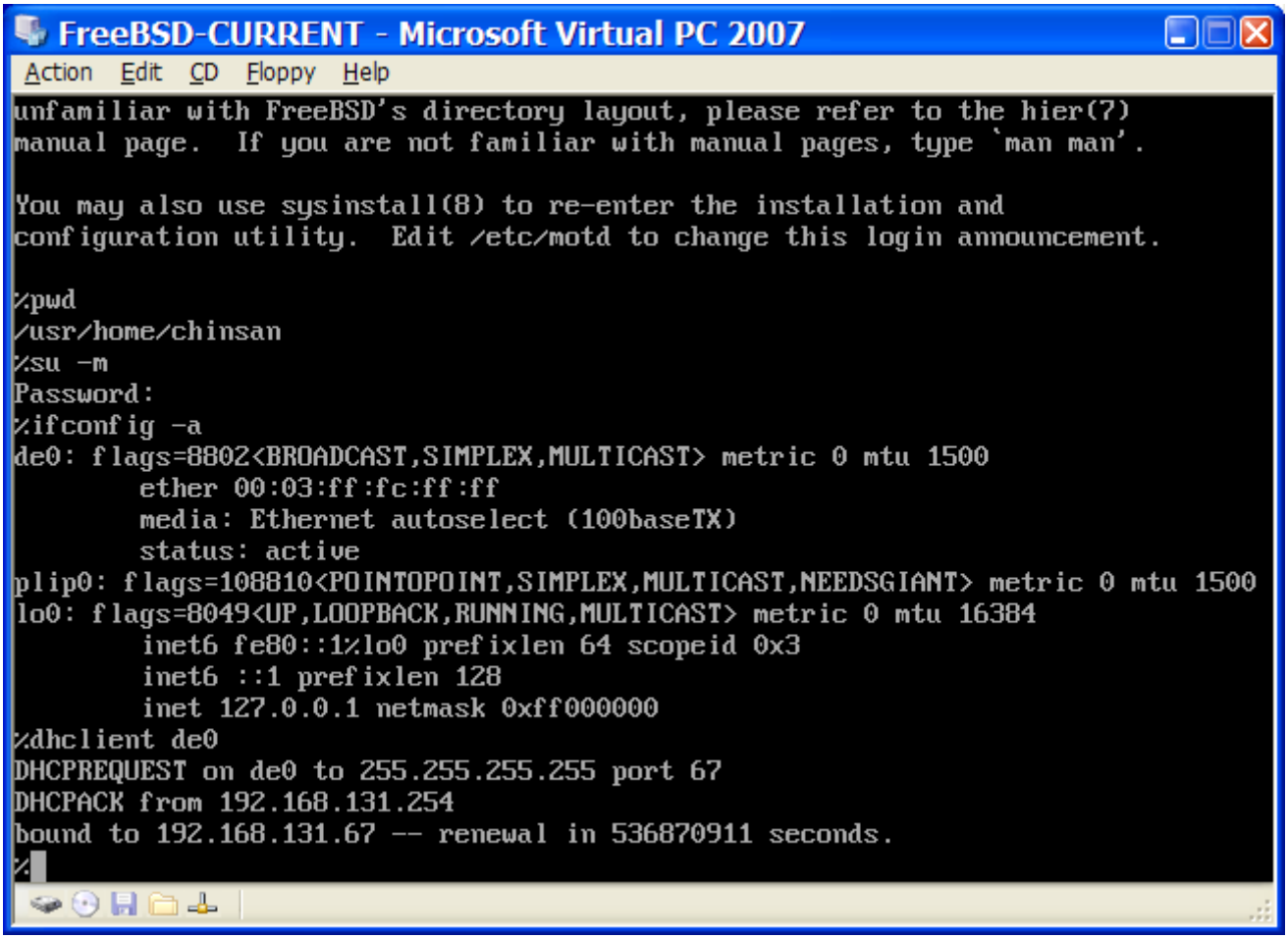
Start, nadat deze associatie met de CDROM-bron is gemaakt, de virtuele FreeBSD-machine opnieuw op zoals gewoonlijk door op **Action** en **Reset** te klikken. Virtual PC zal herstarten met een speciale BIOS dat eerst controleert of er een CDROM aanwezig is, net zoals eeen normale BIOS dat zou doen.



In dit geval zal het de installatiemedia van FreeBSD vinden en een normale installatie gebaseerd op sysinstall beginnen zoals beschreven in FreeBSD installeren op FreeBSD 8.X en eerder. X11 kan nu geïnstalleerd, maar nog niet ingesteld, worden.



Denk eraan om de CDROM of het ISO-beeld te verwijderen nadat de installatie voltooid is. Start als laatste op naar de nieuw geïnstalleerde virtuele FreeBSD-machine.



FreeBSD instellen op Microsoft® Windows®/Virtual PC

Nadat FreeBSD succesvol is geïnstalleerd op Microsoft® Windows® met Virtual PC zijn er een aantal instellingen die aangepast kunnen worden om het system te optimaliseren voor virtueel gebruik.

1. De variabelen voor de bootloader instellen

   De belangrijkste stap is om de tunable `kern.hz` te verlagen om zo het CPU-gebruik van FreeBSD in de omgeving van Virtual PC te verminderen. Dit kan bereikt worden door de volgende regel aan /boot/loader.conf toe te voegen:

   ```
   kern.hz=100
   ```

   Zonder deze instelling zal een FreeBSD als gast-besturingssysteem voor Virtual PC in rust ongeveer 40% van de CPU van een computer met een enkele processor gebruiken. Na deze verandering zal het gebruik slechts rond de 3% liggen.

2. Een nieuw instellingenbestand voor de kernel aanmaken

   Alle stuurprogramma's voor SCSI, FireWire, en USB kunnen verwijderd worden. Virtual PC beidt een virtuele netwerkadapter dat door het stuurprogramma de(4) gebruikt wordt, dus kunnen alle netwerkapparaten behalve de(4) en miibus(4) uit de kernel verwijderd worden.

3. Het netwerk instellen

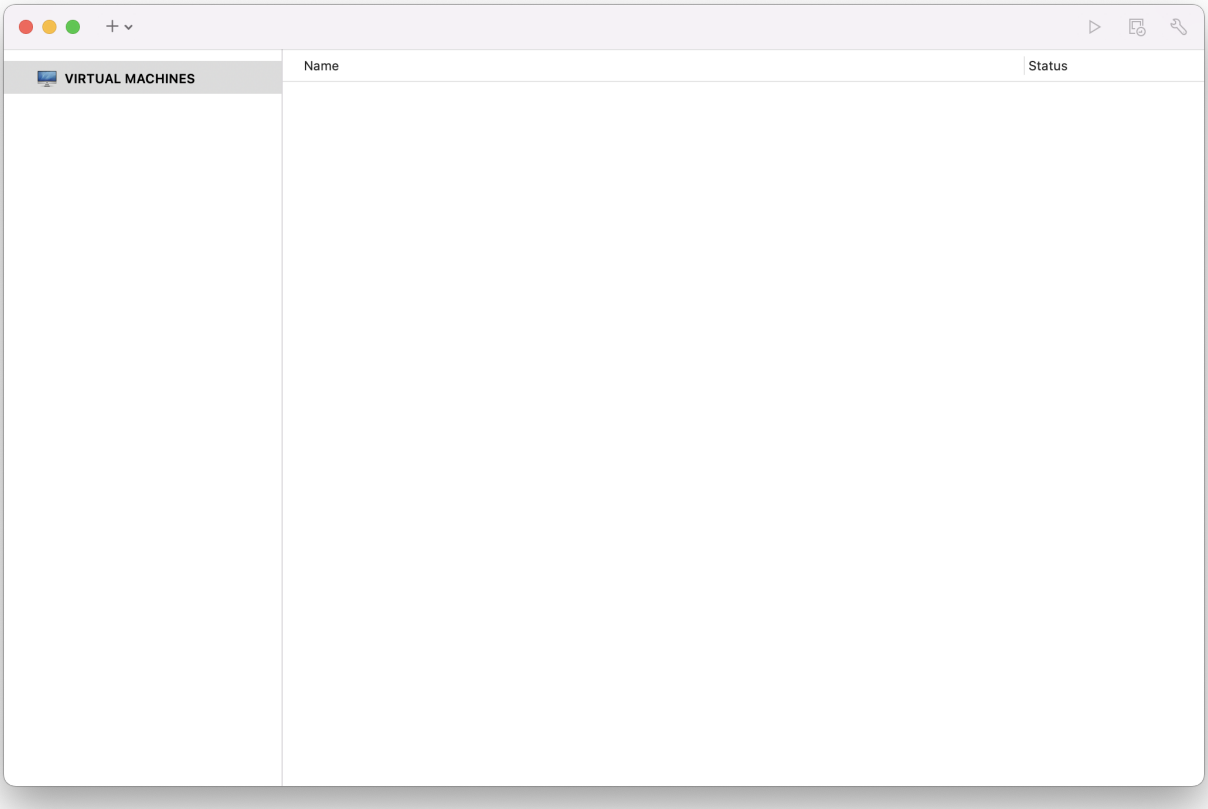
   De eenvoudigste netwerkinstallatie omvat het gebruik van DHCP om de virtuele machine met het zelfde LAN te verbinden als de Microsoft® Windows®-gastheer. Dit kan bereikt worden door `ifconfig_de0="DHCP"` toe te voegen aan /etc/rc.conf. Geavanceerdere netwerkinstallaties staan beschreven in Netwerken voor gevorderden.

20.2.3. VMware op Mac OS®

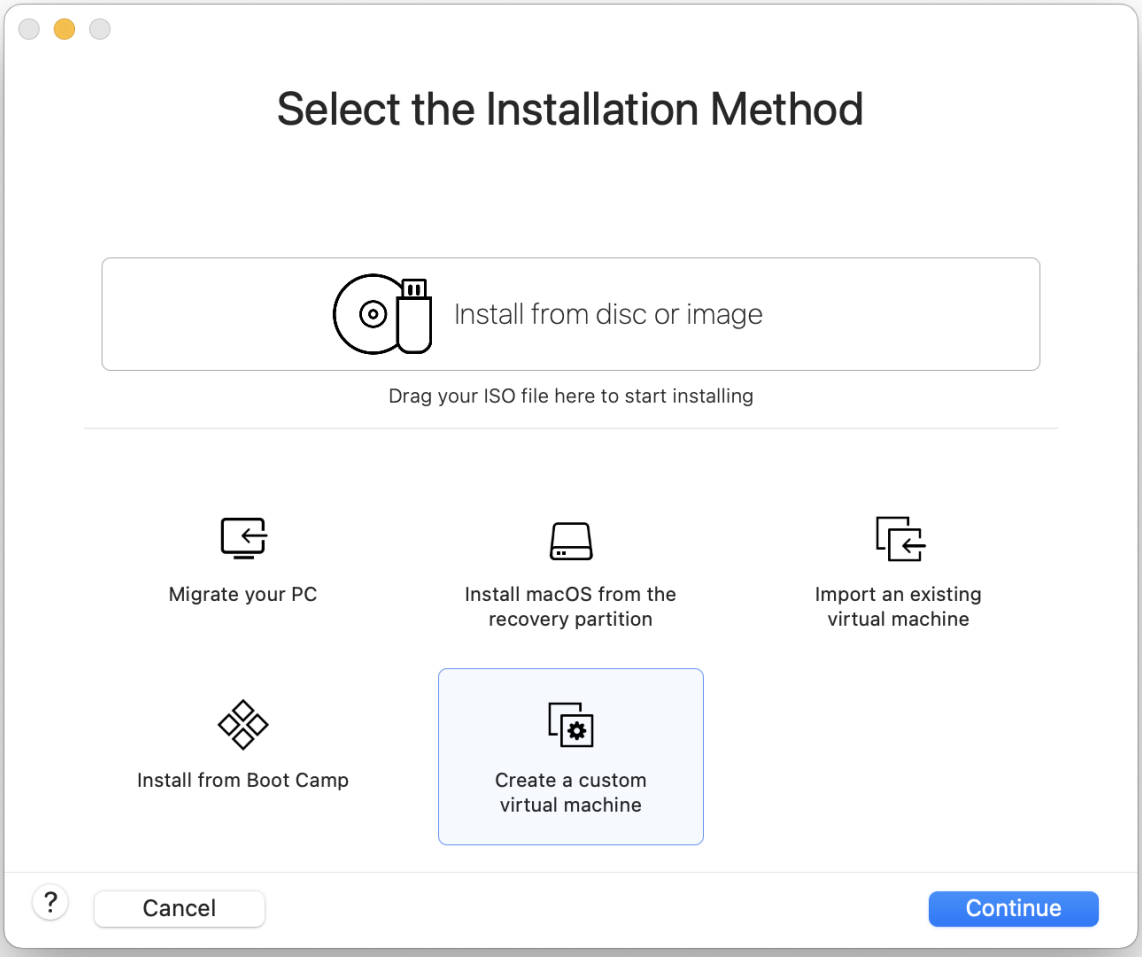
VMware Fusion voor Mac® is een comercieel softwareprodukt beschikbaar voor op Intel® gebaseeerde Mac®-computers die Mac OS® 10.4.9 of nieuwer draaien. FreeBSD is een volledig ondersteund gast-besturingssysteem. Nadat VMware Fusion is geïnstalleerd op Mac OS® X dient de gebruiker een virtuele machine in te stellen en daarna het gewenste gast-besturingssysteem te installeren.

FreeBSD installeren op VMware/Mac OS® X

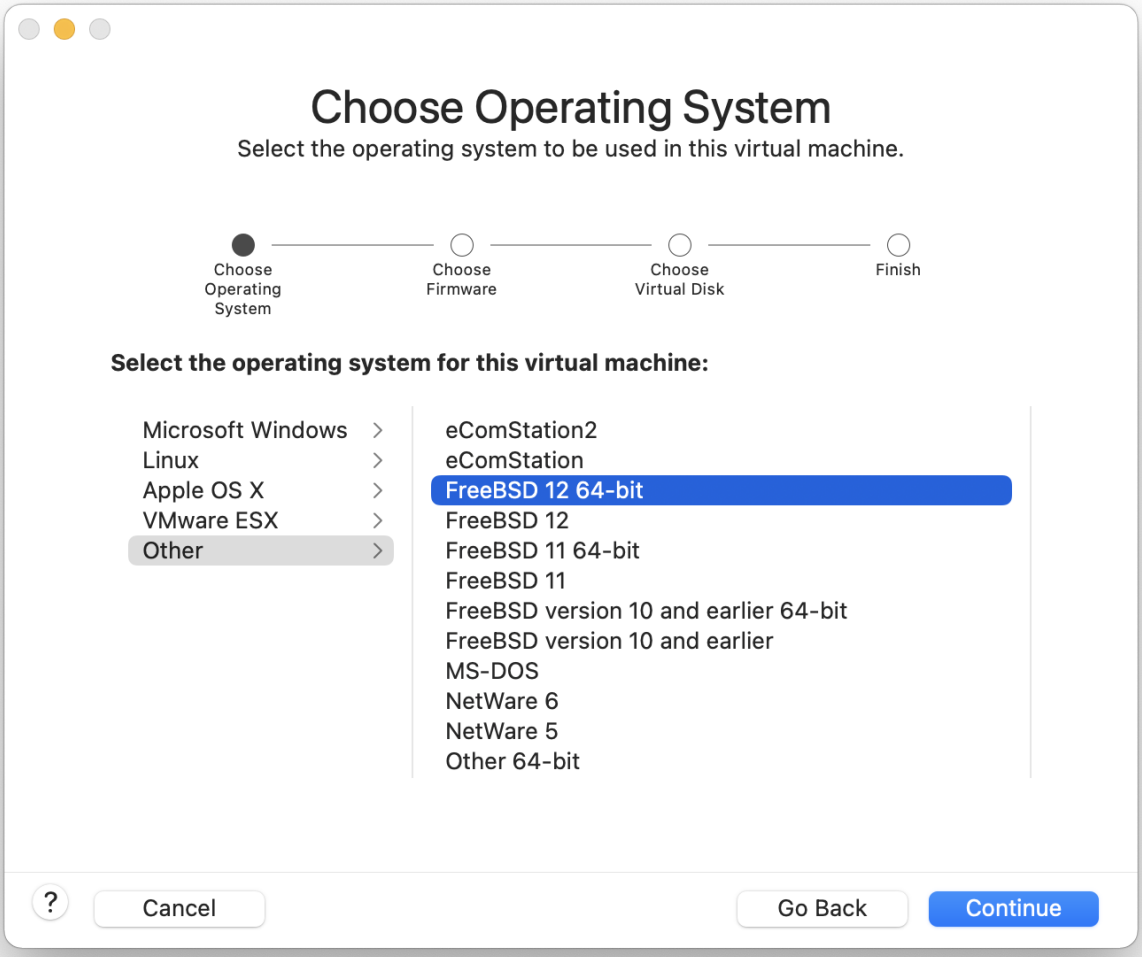
De eerste stap is om VMware Fusion te laden, de Virtual Machine Library zal geladen worden. Klik op "New" om de VM aan te maken:



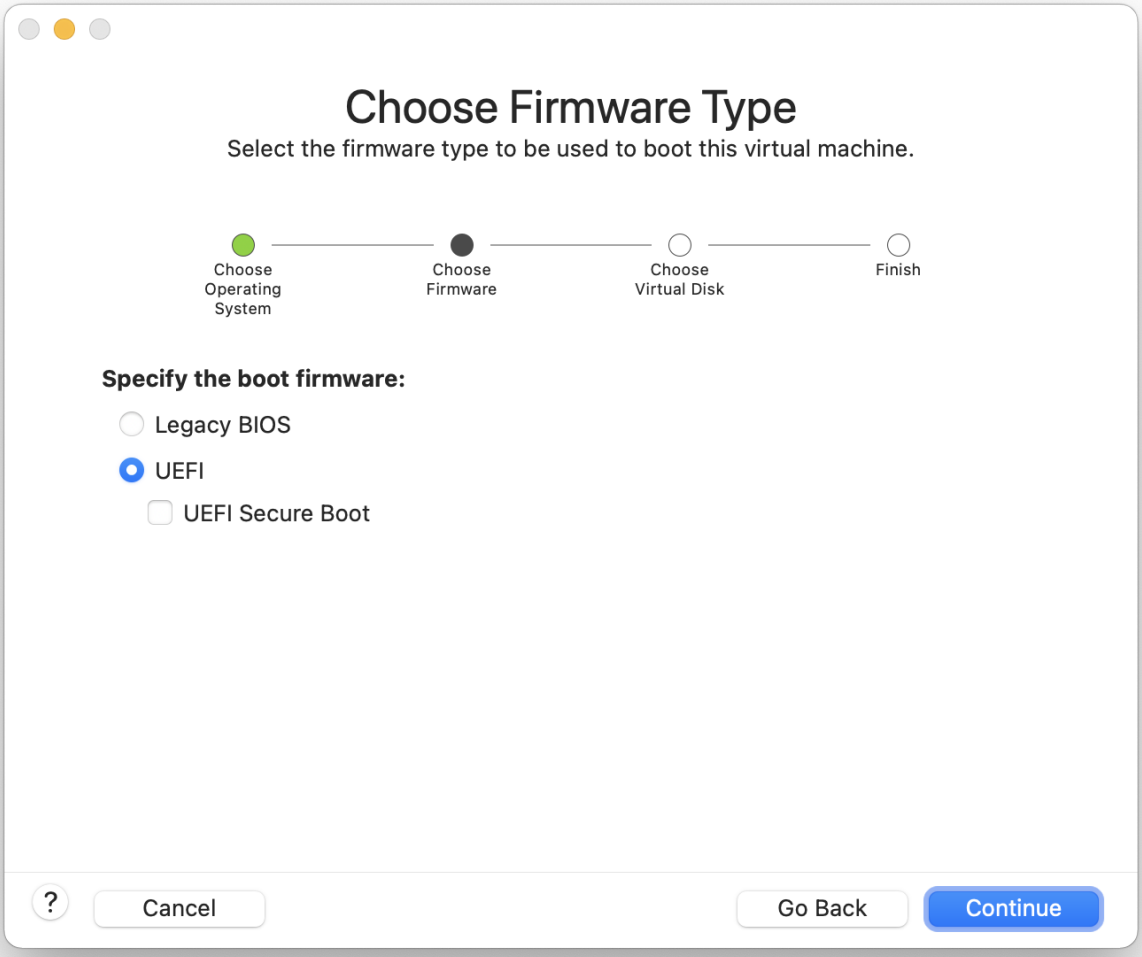
Dit laadt de New Virtual Machine Assistant dat helpt om de VM aan te maken, klik op Continue om verder te gaan:



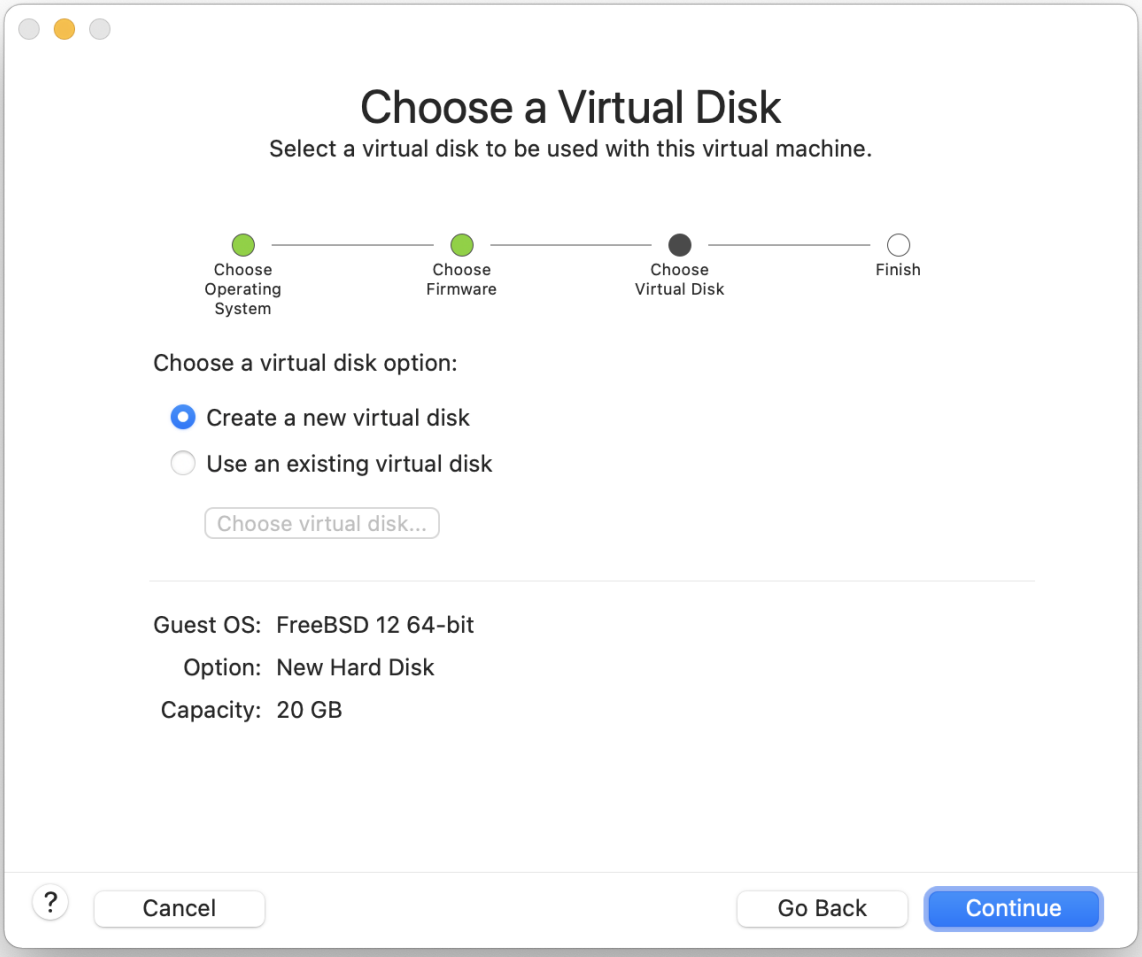
Selecteer Other als het Operating System en FreeBSD of FreeBSD 64-bit, afhankelijk van de wens voor ondersteuning voor 64-bit, als de Version wanneer dat gevraagd wordt:



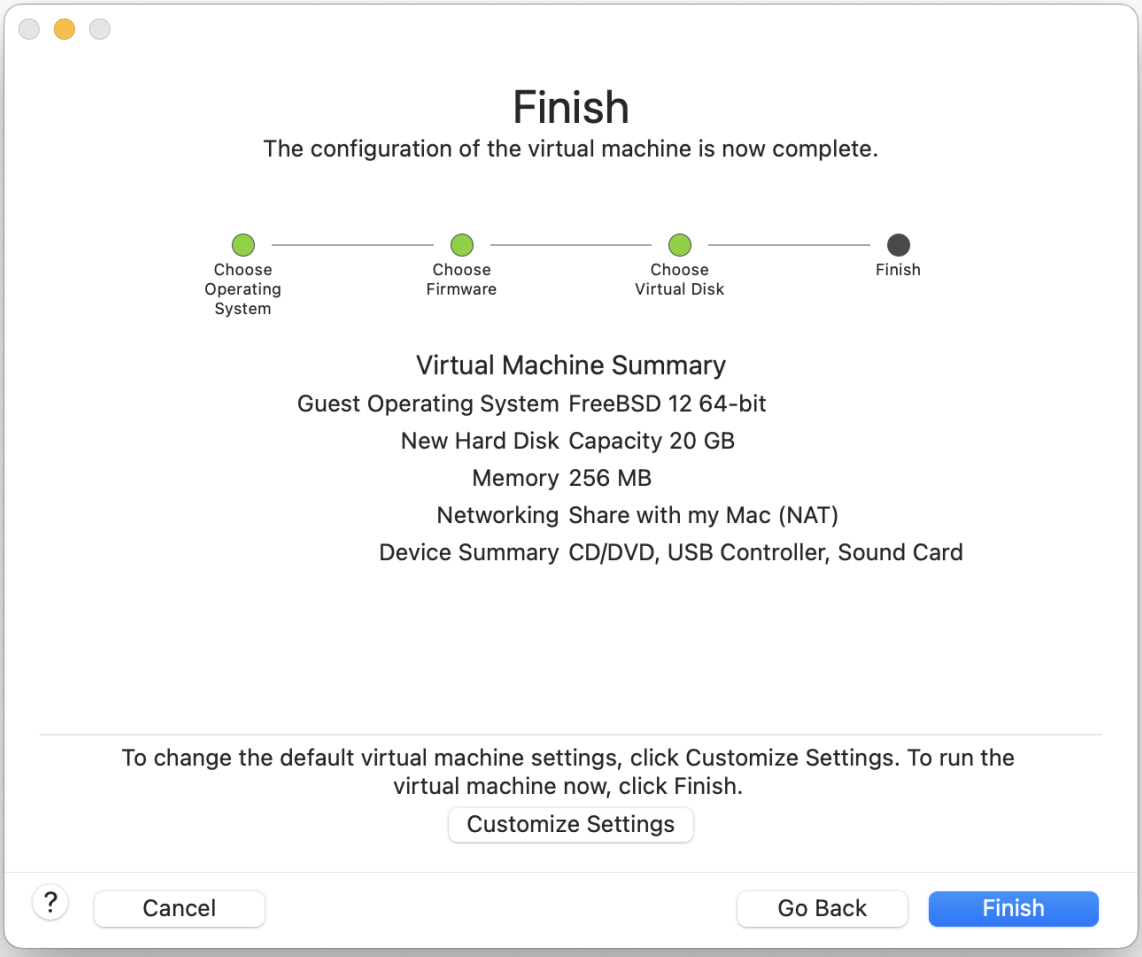
Kies de naam van het VM-beeld en de map waarin het bewaard dient te worden:



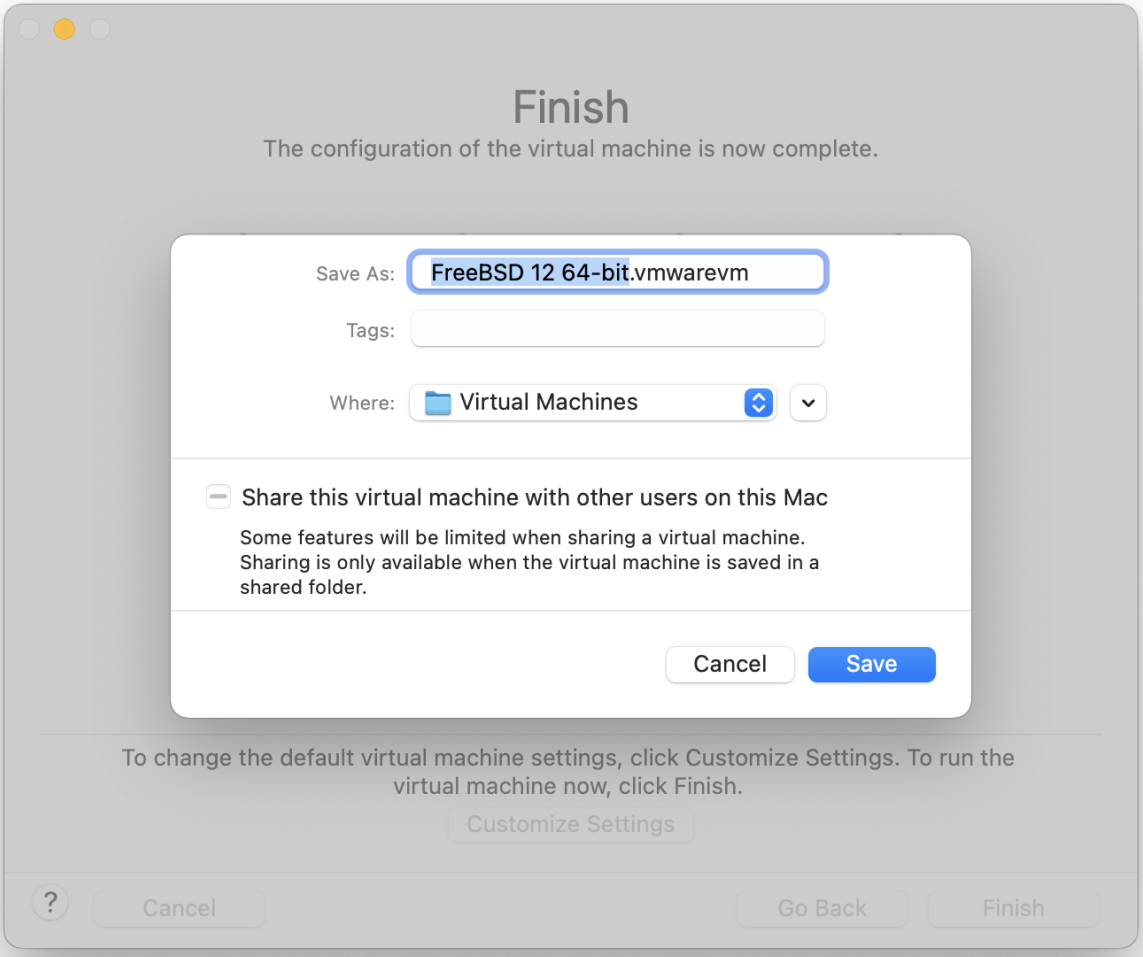
Kies de grootte van de virtuele harde schijf voor de VM:



Kies de manier om de VM te installeren, van een ISO-beeld of van een CD:



Nadat op Finish is geklikt, zal de VM opstarten:



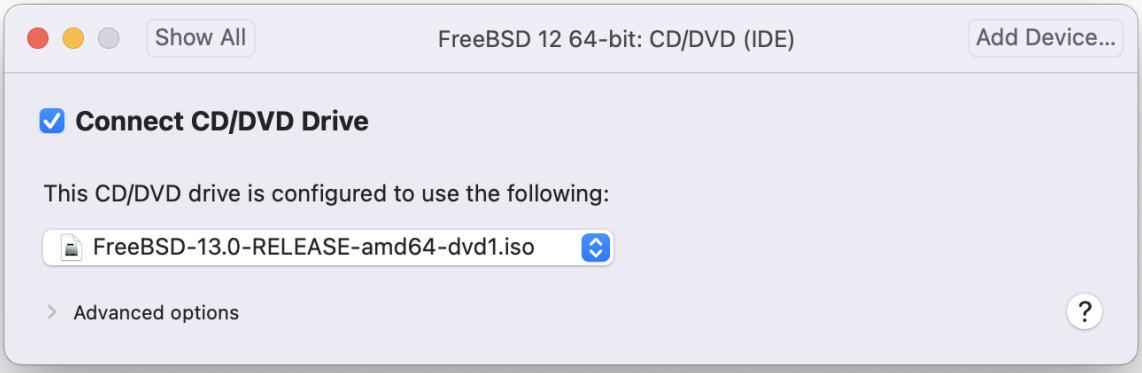
Installeer FreeBSD zoals gewoonlijk, of door de aanwijzingen in FreeBSD installeren op FreeBSD 8.X en eerder op te volgen:



Nadat de installatie voltooid is kunnen de instellingen van de VM aangepast worden, zoals het geheugengebruik:



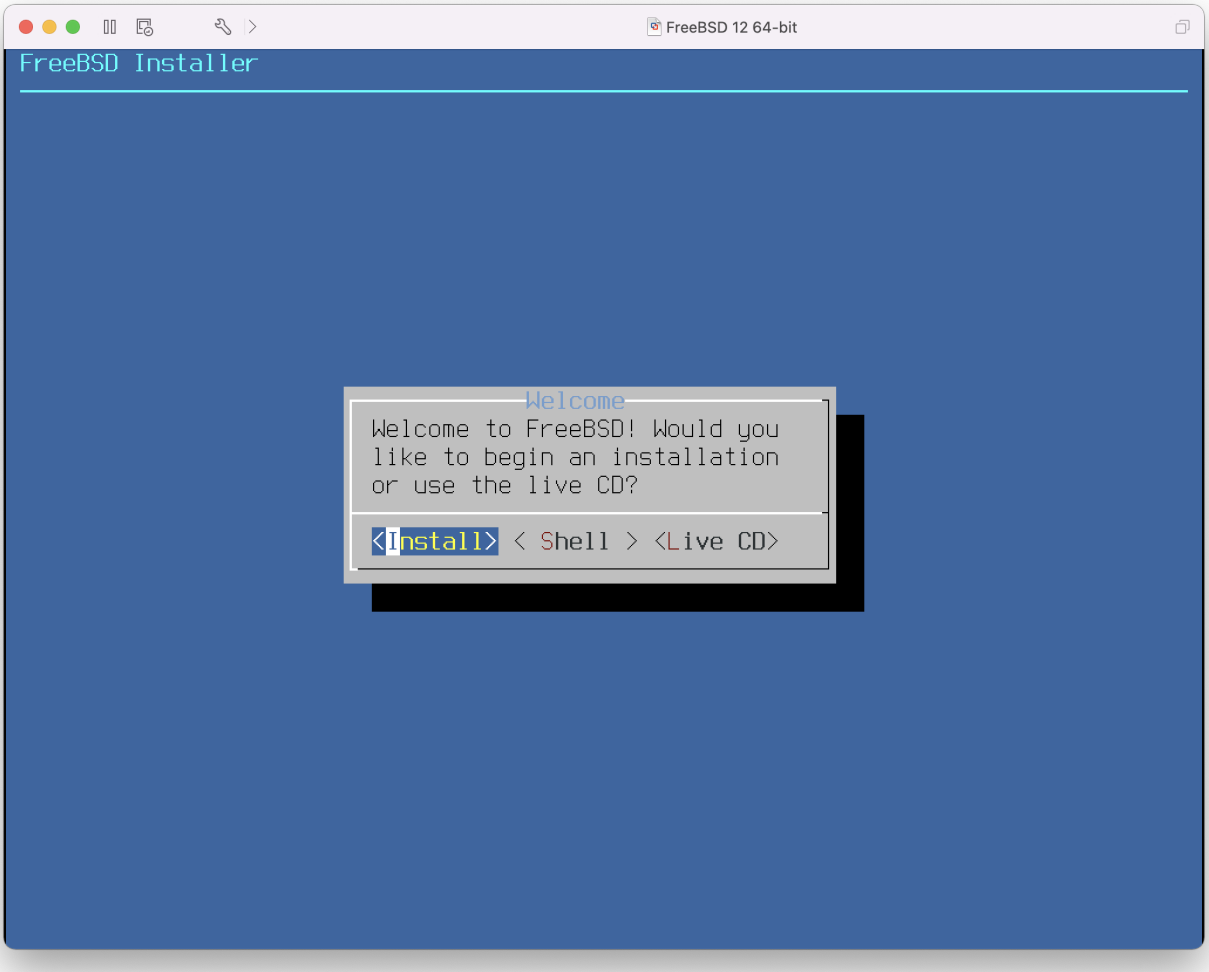
De instellingen van de systeemhardware van de VM kunnen niet veranderd worden zolang de VM draait.



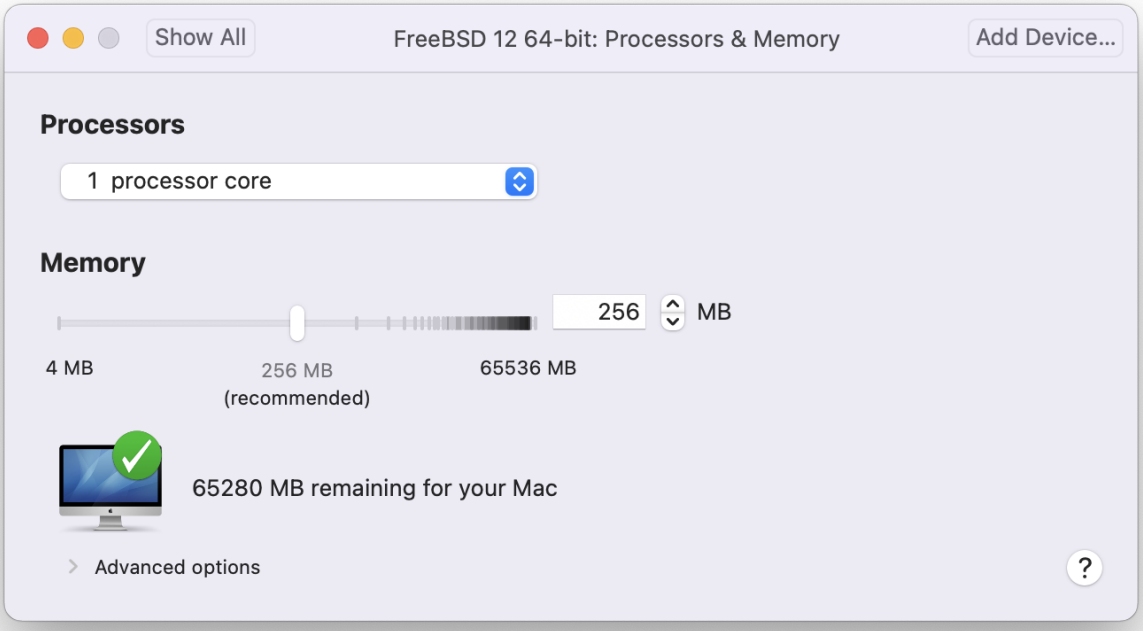
Het aantal CPU's waartoe de VM toegang heeft:



De status van het CD-ROM-apparaat. Gewoonlijk kan de CD-ROM of het ISO-beeld van de VM worden losgekoppeld wanneer het niet meer nodig is.



Het laatste om te veranderen is de manier waarop de VM verbinding met het netwerk maakt. Indien verbindingen naar de VM van andere machines naast de gastheer gewenst zijn, dient Connect directly to the physical network (Bridged) gekozen te worden. In andere situaties is Share the host's internet connection (NAT) te verkiezen, zodat de VM toegang kan hebben tot het Internet, maar dat het netwerk geen toegang heeft tot de VM.



Herstart de nieuw geïnstalleerde virtuele FreeBSD-machine nadat alle instellingen zijn aangepast.

FreeBSD instellen op Mac OS® X/VMware

Nadat FreeBSD succesvol is geïnstalleerd op Mac OS® X met VMware, zijn er een aantal instellingen die gewijzigd kunnen worden op het systeem te optimaliseren voor virtueel gebruik.

1. De variabelen voor de bootloader instellen

   De belangrijkste stap is het verlagen van de tunable `kern.hz` om het CPU-gebruik van FreeBSD in de omgeving van VMware te verminderen. Dit kan bereikt worden door de volgende regel aan /boot/loader.conf toe te voegen:

   ```
   kern.hz=100
   ```

   Zonder deze instelling gebruikt FreeBSD als VMware gast-besturingssysteem ongeveer 15% van de CPU van een enkele iMac®-processor. Na deze verandering zal het gebruik dichter bij 5% liggen.

2. Een nieuw instellingenbestand voor de kernel aanmaken

   Alle stuurprogramma's voor FireWire en USB kunnen verwijderd worden. VMware biedt een virtuele netwerkadapter dat door het stuurpogramma em(4) gebruikt wordt, dus alle netwerkapparaten behalve em(4) kunnen uit de kernel verwijderd worden.

3. Het netwerk instellen

   De eenvoudigste netwerkinstallatie omvat het gebruik van DHCP om de virtuele machine

met hetzelfde LAN te verbinden als de Mac®-gastheer. Dit kan bereikt worden door `ifconfig_em0="DHCP"` toe te voegen aan /etc/rc.conf. Geavanceerdere netwerkinstallaties staan beschreven in Netwerken voor gevorderden.

20.2.4. VirtualBox™ gasttoevoegingen op een FreeBSD-gast

De gasttoevoegingen van VirtualBox™ bieden ondersteuning voor:

- Het delen van het prikbord
- Integratie van de muiscursor
- Synchronisatie met de tijd van de gastheer
- Het schalen van vensters
- Naadloze modus



De volgende commando's worden gedraaid in de FreeBSD-gast.

Installeer ten eerste het pakket emulators/virtualbox-ose-additions in de FreeBSD-gast.

```
# cd /usr/ports/emulators/virtualbox-ose-additions && make install clean
```

Voeg deze regels toe aan /etc/rc.conf:

```
vboxguest_enable="YES"
vboxservice_enable="YES"
```

Als ntpd(8) of ntpdate(8) gebruikt, dient de synchronisatie met de tijd van de gastheer te worden uitgeschakeld:

```
vboxservice_flags="--disable-timesync"
```

De `vboxvideo_drv` zou herkend moeten worden door `Xorg -configure`. Als dit niet zo is, dient xorg.conf gewijzigd te worden voor de videokaart van VirtualBox™:

```
Section "Device"
    ### Available Driver options are:-
    ### Values: <i>: integer, <f>: float, <bool>: "True"/"False",
    ### <string>: "String", <freq>: "<f> Hz/kHz/MHz"
    ### [arg]: arg optional
    Identifier "Card0"
    Driver "vboxvideo"
    VendorName "InnoTek Systemberatung GmbH"
    BoardName "VirtualBox Graphics Adapter"
    BusID "PCI:0:2:0"
```

```
EndSection
```

Pas het gedeelte over de muis in xorg.conf aan om `vboxmouse_drv` te gebruiken:

```
Section "InputDevice"
    Identifier "Mouse0"
    Driver "vboxmouse"
EndSection
```

Gebruikers van HAL dienen dit bestand aan te maken als /usr/local/etc/hal/fdi/policy/90-vboxguest.fdi of het te kopiëren van /usr/local/hal/fdi/policy/10osvendor/90-vboxguest.fdi:

```
<?xml version="1.0" encoding="utf-8"?>
<!--
# Sun VirtualBox
# Hal driver description for the vboxmouse driver
# $Id$

       Copyright (C) 2008-2009 Sun Microsystems, Inc.

       This file is part of VirtualBox Open Source Edition (OSE, as
       available from http://www.virtualbox.org. This file is free software;
       you can redistribute it and/or modify it under the terms of the GNU
       General Public License (GPL) as published by the Free Software
       Foundation, in version 2 as it comes in the "COPYING" file of the
       VirtualBox OSE distribution. VirtualBox OSE is distributed in the
       hope that it will be useful, but WITHOUT ANY WARRANTY of any kind.

       Please contact Sun Microsystems, Inc., 4150 Network Circle, Santa
       Clara, CA 95054 USA or visit http://www.sun.com if you need
       additional information or have any questions.
-->
<deviceinfo version="0.2">
  <device>
    <match key="info.subsystem" string="pci">
      <match key="info.product" string="VirtualBox guest Service">
    <append key="info.capabilities" type="strlist">input</append>
    <append key="info.capabilities" type="strlist">input.mouse</append>
    <merge key="input.x11_driver" type="string">vboxmouse</merge>
    <merge key="input.device" type="string">/dev/vboxguest</merge>
      </match>
    </match>
  </device>
</deviceinfo>
```

20.3. FreeBSD als een gastheer-besturingssysteem

Voor een aantal jaren werd FreeBSD niet officieel ondersteund als een gastheer-besturingssysteem door de beschikbare virtualisatiepakketten. Sommige mensen gebruikten oudere en meestal overbodige versies van VMware (zoals emulators/vmware3), die gebruik maakten van de Linux® binaire compatibiliteitslaag. Kort na de uitgave van FreeBSD 7.2 verscheen Suns VirtualBox™ in de Ports; Collectie als een programma dat voor FreeBSD zelf bedoeld is.

VirtualBox™ is een actief ontwikkeld en compleet virtualisatiepakket dat beschikbaar is voor de meeste besturingssystemen waaronder Windows®, Mac OS®, Linux® en FreeBSD. Het kan evengoed Windows® of UNIX® als gast draaien. Het is gerealiseerd als een open-source pakket met gesloten-source componenten beschikbaar in een apart uitbreidingspakket. Deze componenten bevatten onder andere ondersteuning voor USB-2.0-apparaten. Meer informatie kan gevonden worden op de pagina "Downloads" van de VirtualBox™-wiki op http://www.virtualbox.org/wiki/Downloads. Momenteel zijn deze uitbreidingen niet beschikbaar voor FreeBSD.

20.3.1. VirtualBox™ installeren

VirtualBox™ is beschikbaar als een FreeBSD-port in emulators/virtualbox-ose. Zorg ervoor, aangezien VirtualBox™ erg actief ontwikkeld wordt, dat uw ports bijgewerkt is voordat u met de installatie begint. Installeer het met deze commando's:

```
# cd /usr/ports/emulators/virtualbox-ose
# make install clean
```

Een nuttige optie in het configuratiescherm is de verzameling `GuestAdditions` programma's. Deze bieden een aantal nuttige mogelijkheden in gastbesturingssystemen, zoals integratie van de muiscursor (wat het mogelijk maakt om de muis te delen tussen de gast en de gastheer zonder dat er een speciale toetsencombinatie hoeft te worden gebruikt voor het omwisselen) en snellere video-rendering, met name in Windows®-gasten. De gastaanvullingen zijn beschikbaar in het menu **Devices**, nadat de installatie van het gastbesturingssysteem is voltooid.

Er zijn enkele aanpassingen aan de instellingen nodig voordat VirtualBox™ voor het eerst wordt gestart. De port installeert een kernelmodule in /boot/modules welke in de draaiende kernel geladen moet worden:

```
# kldload vboxdrv
```

Voeg de volgende regel toe aan /boot/loader.conf om er zeker van te zijn dat de module altijd na een herstart wordt geladen:

```
vboxdrv_load="YES"
```

Voeg het volgende aan /etc/rc.conf toe en herstart de computer om de kernelmodules te gebruiken

die in gebridgede of gastheer-only netwerken voorzien:

```
vboxnet_enable="YES"
```

De groep `vboxusers` wordt tijdens de installatie van VirtualBox™ aangemaakt. Alle gebruikers die toegang tot VirtualBox™ nodig hebben moeten als lid van deze groep worden toegevoegd. Met het commando `pw` kunnen nieuwe leden worden toegevoegd:

```
# pw groupmod vboxusers -m uwgebruikersnaam
```

De standaardpermissies voor /dev/vboxnetctl zijn restrictief en moeten veranderd worden voor gebridged netwerken.

Om het tijdelijk te testen:

```
# chown root:vboxusers /dev/vboxnetctl
# chmod 0660 /dev/vboxnetctl
```

Voeg deze regels toe aan /etc/devfs.conf om de permissiewijziging permanent te maken:

```
own vboxnetctl root:vboxusers
perm    vboxnetctl 0660
```

Gebruik de optie Sun VirtualBox van het menu van de grafische omgeving of typ het volgende in een terminal om VirtualBox™ te starten:

```
% VirtualBox
```

Bezoek de officiële website op http://www.virtualbox.org voor meer informatie over het configureren en gebruiken van VirtualBox™. Aangezien de FreeBSD-port erg nieuw is, wordt het nog volop ontwikkeld. Kijk voor de laatste informatie en instructies om problemen op te lossen op de relevantie pagina in de FreeBSD-wiki op http://wiki.FreeBSD.org/VirtualBox.

20.3.2. USB-ondersteuning in VirtualBox™



Voor deze stappen is VirtualBox 4.0.0 of nieuwer nodig.

Om van UBS-apparaten te kunnen lezen en ernaar te kunnen schrijven dienen gebruikers lid te zijn van de groep operator:

```
# pw groupmod operator -m jerry
```

Voeg vervolgens het volgende toe aan /etc/devfs.rules (maak het aan als het nog niet bestaat):

```
[system=10]
add path 'usb/*' mode 0660 group operator
```

Voeg het volgende aan toe aan /etc/rc.conf om deze nieuwe regels te laden:

```
devfs_system_ruleset="system
```

Herstart vervolgens devfs:

```
# service devfs restart
```

USB kan nu in het gast-besturingssysteem worden aangezet. USB-apparaten zouden zichtbaar moeten zijn in de voorkeuren van VirtualBox™.

20.3.3. DVD/CD-toegang van de gastheer in VirtualBox™

Toegang tot de CD/DVD-stations van de gastheer wordt bereikt door het delen van de fysieke stations. In de GUI kan dit vanuit het scherm Opslag in de Instellingen van de virtuele machine worden ingesteld. Maak eerst een leeg IDE CD-/DVD-apparaat aan. Kies daarna het Gastheerstation van het popup-menu voor het kiezen van het virtuele CD-/DVD-station. Later zal er een checkbox genaamd `Passthrough` verschijnen. Dit stelt de virtuele machine in staat om de hardware direct te gebruiken. Audio-CDs en branders bijvoorbeeld werken alleen als deze optie is aangezet.

HAL moet draaien om de DVD/CD-functies van VirtualBox™ te laten werken, zet het dus aan in /etc/rc.conf en start het (als het niet reeds draait):

```
hald_enable="YES"
```

```
# service hald start
```

Om gebruikers de DVD/CD-functionaliteit van VirtualBox™ te laten gebruiken, dienen ze toegang te hebben tot /dev/xpt0, /dev/cdN en /dev/passN. Dit wordt normaliter gedaan door de gebruiker van VirtualBox™ lid te maken van de groep operator, wat ook de standaardgroep is voor bovengenoemde apparaten. De rechten van deze apparaten dienen gecorrigeerd te worden door de volgende regels aan /etc/devfs.conf toe te voegen:

```
perm cd* 0600
perm xpt0 0660
perm pass* 0660
```

```
# service devfs restart
```

Hoofdstuk 21. Lokalisatie - I18N/L10N gebruiken en instellen

21.1. Overzicht

FreeBSD is een zeer gedistribueerd project met gebruikers over de gehele wereld. Dit hoofdstuk behandelt de internationalisatie- en lokalisatie-eigenschappen van FreeBSD die niet-Engelssprekende gebruikers echt werk laten verzetten. Er zitten veel aspecten van de i18n-implementatie in zowel de systeem- als applicatieniveaus, dus waar mogelijk wordt de lezer verwezen naar meer specifieke bronnen.

Na dit hoofdstuk weet de lezer:

- Hoe verschillende talen en locales gecodeerd zijn op moderne besturingssystemen.
- Hoe de locale in te stellen voor een login-shell.
- Hoe de console voor niet-Engelse talen in te stellen.
- Hoe het X Window systeem effectief met meerdere talen te gebruiken.
- Waar meer informatie te vinden over het schrijven van i18n-respecterende applicaties.

Veronderstelde voorkennis:

- Weten hoe aanvullende applicaties van derde partijen geïnstalleerd worden (Applicaties installeren. pakketten en ports).

21.2. Beginselen

21.2.1. Wat is I18N/L10N?

Ontwikkelaars hebben internationalisatie ("internationalization" afgekort tot de term I18N, de eerste en de laatste letter en het aantal tussenliggende letters. L10N gebruikt hetzelfde schema voor naamgeving en komt van "localization". Samen staan I18N/L10N methoden, protocollen en applicaties gebruikers toe de taal van hun keuze te gebruiken.

I18N-applicaties zijn geprogrammeerd door gebruik te maken van I18N-gereedschappen van bibliotheken. Daarmee kunnen ontwikkelaars een eenvoudig bestand schrijven en menu’s en teksten weergeven in elke taal. Programmeurs worden door het FreeBSD Project sterk aangemoedigd deze conventie te volgen.

21.2.2. Waarom I18N/L10N gebruiken?

I18N/L10N wordt gebruikt als een gebruiker gegevens wil bekijken, invoeren of verwerken in niet-Engelse talen.

21.2.3. Welke talen worden ondersteund door I18N?

I18N en L10N zijn niet FreeBSD specifiek. Momenteel kan er gekozen worden uit de meeste grote wereldtalen, inclusief maar niet beperkt tot: Chinees, Duits, Japans, Koreaans, Frans, Russisch en Vietnamees.

21.3. Lokalisatie gebruiken

In al zijn pracht is I18N niet FreeBSD specifiek maar een conventie. Het FreeBSD Project moedigt iedereen aan FreeBSD te helpen deze conventie te gebruiken.

Lokalisatie-instellingen zijn gebaseerd op drie hoofdtermen: Taalcode, Landcode en Codering. Localenamen zijn als volgt opgebouwd:

```
Taalcode_Landcode.Codering
```

21.3.1. Taal- en landcodes

Om een FreeBSD-systeem (of een ander I18N-ondersteunend UNIX® achtig systeem) te lokaliseren naar een bepaalde taal, moet de gebruiker de codes voor het specifieke land en taal achterhalen. Landcodes geven applicaties aan welke variatie van de gegeven taal gebruikt moet worden. Ook webbrowsers, SMTP/POP-servers, webservers, enzovoorts maken beslissingen gebaseerd op die codes. Hieronder staan voorbeelden van taal- en landcodes:

Taal- en landcode	Omschrijving
en_US	Engels - Verenigde Staten
ru_RU	Russisch voor Rusland
zh_TW	Traditioneel Chinees voor Taiwan

Een complete lijst van beschikbare locales is beschikbaar via:

```
% locale -a
```

21.3.2. Coderingen

Sommige talen gebruiken andere ASCII-coderingen dan 8-bit, wijde of multibyte karakters, zie multibyte(3). Oudere programma's herkennen die niet en interpreteren ze foutief als controlekarakters aan. Afhankelijk van de implementatie moeten gebruikers eventueel een applicatie met wijde of multibyte karakterondersteuning compileren, of hem correct instellen. Om wijde of multibyte karakters in te kunnen voeren en te kunnen verwerken levert de FreeBSD Portscollectie voor elke taal programma's. Hiervoor staat I18N-documentatie in de respectievelijke FreeBSD Port.

Voor het bouwen van een gewenste applicatie met lokalisatie is het verstandig de applicatiedocumentatie te bekijken om te bepalen hoe de juiste waarden doorgegeven kunnen

worden naar configure, Makefile of de compiler.

Houd rekening met:

- Taalspecifieke enkele C-karakters karakterverzamelingen (zie multibyte(3)), bijvoorbeeld ISO8859-1, ISO-8859-15, KOI8-R of CP437.
- Wijde of multibyte coderingen, bijvoorbeeld EUC of Big5.

Een lijst met actieve karakterverzamelingen staat bij de IANA Registry.



FreeBSD gebruikt in plaats hiervan X11-compatible locale-coderingen.

21.3.3. I18N applicaties

In het FreeBSD Ports en Package systeem hebben I18N-applicaties `I18N` in hun naam zodat ze eenvoudig herkend kunnen worden. Toch ondersteunen ze niet altijd iedere mogelijk gewenste taal.

21.3.4. Locale instellen

Meestal is het voldoende om de waarde van de localenaam te exporteren als `LANG` in de login-shell. Dit kan door die waarde in ~/.login_conf van de gebruiker of in ~/.profile, ~/.bashrc of ~/.cshrc) van de gebruiker te zetten. Het is niet nodig om localedeelverzamelingen als `LC_CTYPE` of `LC_CTIME` in te stellen. Bij de taalspecifieke FreeBSD documentatie staat vaak nog informatie.

De twee volgende omgevingsvariabelen moeten in de instellingenbestanden ingesteld worden:

- `LANG` voor de POSIX® setlocale(3) functies.
- `MM_CHARSET` voor de MIME karakters voor applicaties.

Dit is inclusief het instellen van de gebruikers-shell, het instellen van de specifieke applicatie en de instellingen voor X11.

Methoden om locale in te stellen

Er zijn twee methoden om de locale in te stellen en beiden worden hieronder beschreven. De eerste (aanbevolen) methode is door middel van het toekennen van omgevingsvariabelen in de loginklasse en de tweede is mogelijk door middel van het toevoegen van de omgevingsvariabelen aan het opstartbestand van de systeem-shell.

Methode loginklasse

Deze methode biedt de mogelijkheid om omgevingsvariabelen die nodig zijn voor de localenaam en MIME karakterverzamelingen éénmalig voor elke mogelijke shell toe te kennen in plaats van door toekenning via het opstartbestand van elke shell. Gebruikersinstellingen kunnen door de gebruiker zelf worden gemaakt en voor Beheerdersinstellingen zijn superuser-rechten nodig.

Gebruikersinstellingen

Hieronder staat een minimaal voorbeeld van een .login_conf bestand in de thuismap van een gebruiker die beide variabelen heeft ingesteld op Latin-1 codering:

```
me:\
  :charset=ISO-8859-1:\
  :lang=de_DE.ISO8859-1:
```

Hieronder staat is een voorbeeld van een .login_conf die variabelen instelt voor traditioneel Chinees in BIG-5 codering. Er zijn veel andere variabelen ingesteld zijn omdat sommige software localevariabelen niet correct respecteert voor Chinees, Japans, en Koreaans.

```
# Gebruikers die geen valuta eenheden of tijdformaten voor Taiwan
# willen gebruiken kunnen handmatig elke variabele wijzigen.
me:\
  :lang=zh_TW.Big5:\
  :setenv=LC_ALL=zh_TW.Big5:\
  :setenv=LC_COLLATE=zh_TW.Big5:\
  :setenv=LC_CTYPE=zh_TW.Big5:\
  :setenv=LC_MESSAGES=zh_TW.Big5:\
  :setenv=LC_MONETARY=zh_TW.Big5:\
  :setenv=LC_NUMERIC=zh_TW.Big5:\
  :setenv=LC_TIME=zh_TW.Big5:\
  :charset=big5:\
  :xmodifiers="@im=gcin": # Stel gcin in als XIM invoerserver
```

Zie Beheerdersinstellingen en login.conf(5) voor meer details.

Beheerdersinstellingen

Er dient gecontroleerd te worden of loginklasse voor gebruikers, /etc/login.conf, de juiste taal instelt door de volgende instellingen in /etc/login.conf:

```
taalnaam|accountstitel:\
  :charset=MIME_karakterverzameling:
  :lang=localenaam:\
  :tc=default:
```

Voor het bovenstaande voorbeeld dat gebruik maakt van Latin-1 ziet dat er als hieronder uit:

```
german|Duitse gebruikersaccounts:\
  :charset=ISO-8859-1:\
  :lang=de_DE.ISO8859-1:\
  :tc=default:
```

Voer voordat de gebruikers login class wordt gewijzigd het volgende uit:

```
# cap_mkdb /etc/login.conf
```

om de nieuwe configuratie in /etc/login.conf zichtbaar te maken voor het systeem.

Loginklasse wijzigen met vipw(8)

Met `vipw` kunnen nieuwe gebruikers toegevoegd worden en de instellingen dienen ongeveer als volgt uit te zien:

```
gebruiker:wachtwoord:1111:11:taal:0:0:Gebruikersnaam:/home/gebruiker:/bin/sh
```

Loginklasse wijzigen met adduser(8)

Met `adduser` kunnen nieuwe gebruikers toegevoegd worden. Hierna dient één van de volgende stappen uitgevoerd te worden:

- `defaultclass = taal` instellen in /etc/adduser.conf. In dit geval dient er voor alle gebruikers van andere talen een `default` klasse ingevoerd te worden.
- Er kan ook gekozen worden voor een antwoord op de vraag over taal vanuit adduser(8):

  ```
  Enter login class: default []:
  ```

- Ook kan het volgende gebruikt worden voor elke gebruiker die een andere taal gebruikt:

  ```
  # adduser -class taal
  ```

Loginklasse wijzigen met pw(8)

Als pw(8) wordt gebruikt om nieuwe gebruikers toe te voegen:

```
# pw useradd gebruikersnaam -L taal
```

Methode opstartbestand shell



Deze methode wordt niet aanbevolen omdat er instellingenen nodig zijn voor elke mogelijke shell. Het advies is de Methode Loginklasse te gebruiken.

Om de localenaam en MIME karakterverzameling toe te voegen kunnen gewoon twee omgevingsvariabelen ingesteld worden, zoals hieronder te zien is, in /etc/profile en/of /etc/csh.login opstartbestanden voor shells. Hier wordt de Duitse taal als voorbeeld gebruikt:

In /etc/profile:

```
LANG=de_DE.ISO8859-1; export LANG
MM_CHARSET=ISO-8859-1; export MM_CHARSET
```

Of in /etc/csh.login:

```
setenv LANG de_DE.ISO8859-1
setenv MM_CHARSET ISO-8859-1
```

Het is ook mogelijk de bovenstaande instructies toe te toevoegen /usr/shared/skel/dot.profile (ongeveer gelijk aan wat hierboven in /etc/profile is gebruikt) of aan /usr/shared/skel/dot.login (ongeveer gelijk aan wat hierboven in /etc/csh.login is gebruikt).

Voor X11:

In $HOME/.xinitrc:

```
LANG=de_DE.ISO8859-1; export LANG
```

Of:

```
setenv LANG de_DE.ISO8859-1
```

Afhankelijk van de shell (zie boven).

21.3.5. Console instellen

Voor alle enkele C-karakters karakterverzamelingen worden de juiste lettertypen voor het console ingesteld in /etc/rc.conf voor de taal in kwestie met:

```
font8x16=lettertypenaam
font8x14=fontnaam
font8x8=fontnaam
```

De *lettertypenaam* komt uit de map /usr/shared/syscons/fonts zonder het achtervoegsel .fnt.

De gebruiker dient ervoor te zorgen dat indien nodig de juiste enkele C-karakters karakterverzameling wordt ingesteld met /stand/sysinstall. In sysinstall kan Configure en Console gekozen worden. Het is ook mogelijk het volgende aan /etc/rc.conf toe te voegen:

```
scrnmap=schermmappingnaam
keymap=toetsenmappingnaam
keychange="fkey_nummer sequentie"
```

schermmappingnaam komt uit de map /usr/shared/syscons/scrnmaps zonder het achtervoegsel .scm. Meestal is een schermmapping met een overeenkomstig gemapt lettertype nodig als workaround om bit 8 naar bit 9 uit te breiden op een lettertype-karaktermatrix van een VGA-adapter in pseudografische gebieden, dat wil zeggen om letters uit dat gebied te halen als het schermlettertype een bit 8 kolom gebruikt.

Als de moused daemon is ingeschakeld met de onderstaande regel in /etc/rc.conf, dan wordt aangeraden de muiscursorinformatie in de volgende paragraaf te bekijken.

```
moused_enable="YES"
```

Standaard neemt de muiscursor van het syscons(4) stuurprogramma het bereik 0xd0-0xd3 van de tekenverzameling in beslag. Als een ingestelde taal dit bereik gebruikt, moet het cursorbereik hierbuiten gehaald worden. Om de workaround voor FreeBSD aan te zetten kan de volgende regel aan /etc/rc.conf toegevoegd worden:

```
mousechar_start=3
```

De *toetsenmappingnaam* komt uit de map /usr/shared/syscons/keymaps zonder het achtervoegsel .kbd. Als niet precies duidelijk is welke toetsenmapping te gebruiken, kan de toetsenmapping getest worden met kbdmap(1) zonder opnieuw op te starten.

`keychange` is nodig om functietoetsen zo te programmeren dat ze overeenkomen met het geselecteerde terminaltype omdat functietoetssequenties niet in de toetsenmapping gedefinieerd kunnen worden.

Er dient ook een controle te zijn op een juiste instelling van het juiste terminaltype voor het console in /etc/ttys voor alle `ttyv*` regels. De huidige instellingen zijn:

Karakterverzameling	**Terminaltype**
ISO8859-1 of ISO-8859-15	`cons25l1`
ISO8859-2	`cons25l2`
ISO8859-7	`cons25l7`
KOI8-R	`cons25r`
KOI8-U	`cons25u`
CP437 (VGA standaardinstelling)	`cons25`
US-ASCII	`cons25w`

Voor wijde of multibyte karaktertalen kan je juiste FreeBSD port in de map /usr/ports/taal gebruikt worden. Sommige ports verschijnen als console terwijl het systeem ze als serieële vtty ziet. Er dienen dus voldoende vtty's gereserveerd te zijn voor zowel X11 als de pseudo-serieële console. Hier is een gedeeltelijke lijst van applicaties voor het gebruik van andere talen in console:

Taal	Locatie
traditioneel Chinees (BIG-5)	chinese/big5con
Japans	japanese/kon2-16dot of japanese/mule-freewnn
Koreaans	korean/han

21.3.6. X11 instellen

Hoewel X11 geen deel is van het FreeBSD Project wordt het hier wel besproken voor FreeBSD gebruikers. Meer details zijn te vinden op de Xorg website of op de website van een andere X11 server die gebruikt wordt.

In ~/.Xresources kunnen applicatiespecifieke I18N instellingen gemaakt worden als lettertypen, menu's, enzovoort.

Lettertypen weergeven

Eerst moet Xorg server (x11-servers/xorg-server), geïnstalleerd worden en daarna de TrueType® lettertypen van de taal. Door de gewenste locale in te stellen worden de menu's en dergelijke in de gekozen taal weergegeven.

Niet-Engelse karakters invoeren

Het X11 Input Method (XIM) protocol is een nieuwe standaard voor alle X11-cliënts. Alle X11-applicaties horen geschreven te worden als XIM-cliënts die invoer aannemen van de XIM-invoerservers. Er zijn meerdere XIM-servers beschikbaar voor verschillende talen.

21.3.7. Printerinstellingen

Sommige enkele C-karakters karakterverzamelingen zijn standaard hardware-gecodeerd in printers. Voor wijde of multibyte karakterverzamelingen is een speciale installatie nodig en het gebruik van apsfilter wordt dan aangeraden. Een document kan ook naar PostScript® of PDF formaat omgezet worden door gebruik te maken van taalspecifieke conversieprogramma's.

21.3.8. Kernel en bestandssystemen

Het FreeBSD Snelle Bestandssysteem (FFS) is 8-bit schoon, dus het kan gebruikt worden met elke enkele C-karakters karakterverzameling (zie multibyte(3)), maar er is geen karakterverzamelingnaam opgeslagen in het bestandssysteem. Het is dus rauw 8-bit en het weet niets van coderingsbevelen. Officieel ondersteunt FFS nog geen enkele vorm van wijde of multibyte karakterverzamelingen. Toch hebben sommige wijde of multibyte karakterverzamelingen onafhankelijke patches voor FFS die ondersteuning inschakelen. Dit zijn tijdelijke oplossingen of hacks die niet overdraagbaar zijn en daarom is besloten ze niet in de source tree op te nemen. Op de websites van de talen staan de patchbestanden en meer informatie.

Voor het FreeBSD MS-DOS® bestandssysteem kan geschakeld worden tussen MS-DOS®, Unicode karakterverzamelingen en gekozen FreeBSD bestandssysteem-karakterverzamelingen. mount_msdosfs(8) beschijft de details.

21.4. I18N-programma's compileren

Veel FreeBSD Ports zijn geschikt gemaakt voor FreeBSD met I18N-ondersteuning. Een aantal daarvan zijn gemarkeerd met "-I18N" in de portnaam. Deze en nog veel andere programma's hebben ingebouwde ondersteuning voor I18N en behoeven geen speciale aandacht.

Toch is het voor sommige applicaties zoals MySQL nodig dat de Makefile ingesteld is met de specifieke karakterverzameling. Dit wordt normaliter gedaan in de Makefile of door middel van het doorgeven van een waarde aan configure in de broncode.

21.5. FreeBSD lokaliseren naar talen

21.5.1. Russisch (KOI8-R codering)

Voor meer informatie over KOI8-R codering, zie de KOI8-R References (Russian Net Character Set).

Locale instellen

Voeg de volgende regels toe aan ~/.login_conf bestand:

```
me:Mijn account:\
  :charset=KOI8-R:\
  :lang=ru_RU.KOI8-R:
```

Zie eerder in dit hoofdstuk voor voorbeelden over het opzetten van de locale.

Console instellen

- Voeg de volgende regel toe aan /etc/rc.conf:

  ```
  mousechar_start=3
  ```

- Gebruik ook de volgende instellingen in /etc/rc.conf:

  ```
  keymap="ru.utf-8"
  scrnmap="utf-82cp866"
  font8x16="cp866b-8x16"
  font8x14="cp866-8x14"
  font8x8="cp866-8x8"
  ```

- Voor elke `ttyv*` regel in /etc/ttys, gebruik `cons25r` als het terminaltype.

Zie eerder in dit hoofdstuk voor voorbeelden over het opzetten van de console.

Printer instellen

Aangezien de meeste printers met Russische karakters met hardware-codepagina CP866 komen, is een speciaal uitvoerfilter nodig om KOI8-R om te zetten in CP866. Zo'n filter is standaard geïnstalleerd als /usr/libexec/lpr/ru/koi2alt. Een /etc/printcap regel voor een Russische printer moet er uit zien als:

```
lp|Russische lokale lijnprinter:\
  :sh:of=/usr/libexec/lpr/ru/koi2alt:\
  :lp=/dev/lpt0:sd=/var/spool/output/lpd:lf=/var/log/lpd-errs:
```

Zie printcap(5) voor een gedetailleerde beschrijving.

MS-DOS® bestandssysteem en Russische bestandsnamen

De volgende voorbeeld fstab(5) regel zet ondersteuning aan voor Russische bestandsnamen gekoppeld op MS-DOS® bestandssystemen:

```
/dev/ad0s2  /dos/c  msdos rw,-Wkoi2dos,-Lru_RU.KOI8-R 0 0
```

De `-L` optie selecteert de te gebruiken localenaam, en `-W` stelt de karakteromzettabel in. Om de `-W` te gebruiken moet /usr gemount zijn voor de MS-DOS® partitie omdat de omzettabellen zich bevinden in /usr/libdata/msdosfs. mount_msdosfs(8) geeft verdere uitleg.

X11 instellen

1. Voer eerst de niet-X lokale instellingen uit zoals beschreven.
2. Installeer bij gebruik van Xorg het package x11-fonts/xorg-fonts-cyrillic.

   Controleer de `"Files"` sectie in /etc/X11/xorg.conf bestand. Zorg dat de volgende regel *vóór* andere `FontPath` regels staan:

   ```
   FontPath  "/usr/local/lib/X11/fonts/cyrillic"
   ```

   Zie de Ports Collectie voor meer cyrillic fonts.

3. Om een Russisch toetsenbord te activeren dient het volgende in het `"Keyboard"` gedeelte van xorg.conf te staan:

   ```
   XkbLayout "ru"
   XkbOptions  "grp:caps_toggle"
   ```

   Voor Xorg:

   ```
   Option  "XkbLayout" "us,ru"
   ```

```
Option  "XkbOptions"  "grp:caps_toggle"
```

Ook moet daar `XkbDisable` uitgeschakeld (uitgecomment) zijn.

Voor `grp:toggle` is de RUS/LAT-schakelaar `Rechter Alt` voor de `grp:ctrl_shift_toggle` schakelaar zal dat `Ctrl` + `Shift` zijn. Voor `grp:caps_toggle` zal de RUS/LAT-schakelaar `CapsLock` zijn. De oude `CapsLock` functie is nog steeds beschikbaar via `Shift` + `CapsLock` (alleen in LAT-modus). `grp:caps_toggle` werkt om onbekende reden niet in Xorg.

Als er "Windows®" toetsen op een toetsenbord zitten en het blijkt dat sommige niet-alfabetische toetsen verkeerd gemapt zijn in RUS-modus, dan kan de volgende regel aan xorg.conf toegevoegd worden:

```
Option  "XkbVariant"  ",winkeys"
```



Het Russische XKB toetsenbord hoeft niet te werken met niet-gelokaliseerde applicaties.



Minimaal gelokaliseerde applicaties moeten vroeg in het programma een aanroep naar de `XtSetLanguageProc (NULL, NULL,);` functie doen.

In KOI8-R for X Window staan meer instructies over het lokaliseren van X11-applicaties.

21.5.2. Traditioneel Chinees voor Taiwan

Het FreeBSD-Taiwan Project heeft een Chinese HOWTO voor FreeBSD op http://netlab.cse.yzu.edu.tw/~statue/zh-l10n-tut/ die gebruik maakt van veel Chinese ports. De huidige redacteur voor de `FreeBSD Chinese HOWTO` is Shen Chuan-Hsing statue@freebsd.sinica.edu.tw.

Chuan-Hsing Shen heeft de Chinese FreeBSD Collection (CFC) gemaakt met gebruik van FreeBSD-Taiwan's `zh-L10N-tut`. De packages en scriptbestanden zijn beschikbaar op ftp://freebsd.csie.nctu.edu.tw/pub/taiwan/CFC.

21.5.3. Duits (alle ISO 8859-1 talen)

Slaven Rezic eserte@cs.tu-berlin.de heeft een tutorial geschreven over het gebruik van umlauten op een FreeBSD-machine. De tutorial is in het Duits geschreven en staat op ttp://user.cs.tu-berlin.de/~eserte/FreeBSD/doc/umlaute/umlaute.html.

21.5.4. Grieks

Nikos Kokkalis nickkokkalis@gmail.com heeft een compleet artikel over Griekse ondersteuning in FreeBSD geschreven. Het is beschikbaar als deel van de officiële Griekse FreeBSD-documentatie, in http://www.freebsd.org/doc/el/articles/greek-language-support/. Merk opdat dit *alleen* in het Grieks beschikbaar is.

21.5.5. Japans en Koreaans

Japanse lokalisatie staat beschreven op http://www.jp.FreeBSD.org/ en de Koreaanse lokalisatie staat op http://www.kr.FreeBSD.org/.

21.5.6. Niet-Engelstalige FreeBSD documentatie

Sommige delen van de FreeBSD-documentatie zijn naar andere talen vertaald. Hiernaar staan links op de hoofdsite of in /usr/shared/doc.

Hoofdstuk 22. FreeBSD updaten en upgraden

22.1. Overzicht

FreeBSD wordt ontwikkeld tussen de verschillende versies in. Sommige mensen prefereren om de officieel uitgegeven versies te draaien, terwijl anderen gesynchroniseerd willen blijven met de nieuwste ontwikkelingen. Zelfs officiële uitgaven echter worden vaak bijgewerkt met veiligheids- en andere kritieke reparaties. Ongeacht de gebruikte versie biedt FreeBSD alle noodzakelijke gereedschappen om uw systeem bijgewerkt te houden, en maakt FreeBSD het upgraden tussen versies ook gemakkelijk. Dit hoofdstuk helpt om een keuze te maken of het wenselijk is het ontwikkelsysteem te volgen of één van de uitgegeven versies. De basisgereedschappen om uw systeem bijgewerkt te houden worden ook gepresenteerd.

Na het lezen van dit hoofdstuk weet de lezer:

- Welke gereedschappen gebruikt kunnen worden om het systeem en de Portscollectie te updaten.
- Hoe een systeem bijgewerkt kan worden met freebsd-update, CVSup, CVS of CTM;
- Hoe de toestand van een geïnstalleerd systeem met een bekende maagdelijke kopie te vergelijken.
- Hoe uw documentatie bijgewerkt te houden met CVSup of documentatie-ports.
- De verschillen tussen de ontwikkeltakken FreeBSD-STABLE en FreeBSD-CURRENT;
- Hoe een basissysteem opnieuw te compileren en te herinstalleren met `make buildworld`, enzovoort.

Veronderstelde criteria:

- Een juist ingesteld netwerk (Geavanceerd netwerken);
- Weten hoe software van derden te installeren (Applicaties installeren. pakketten en ports).



Door dit hoofdstuk heen wordt `cvsup` gebruikt om de broncode van FreeBSD te verkrijgen en bij te werken. Om het te gebruiken, dient u de port of het pakket voor net/cvsup te installeren (als u niet de grafische `cvsup`-cliënt wilt installeren, kunt u de port net/cvsup-without-gui installeren. U kunt ervoor kiezen om dit te vervangen door csup(1) welke onderdeel is van het basissysteem.

22.2. FreeBSD Update

Het toepassen van beveiligingspatches is een belangrijk onderdeel van het beheren van computersoftware, met name het besturingssysteem. Dit was voor een lange tijd geen gemakkelijk proces op FreeBSD. Er moesten patches op de broncode worden toegepast, de code moest herbouwd worden tot binairen, en daarna moesten de binairen worden geherinstalleerd.

Dit is niet langer het geval aangezien FreeBSD nu een gereedschap heeft dat eenvoudigweg `freebsd-update` heet. Dit gereedschap biedt twee gescheiden functies. Ten eerste voorziet het in het

toepassen van binaire beveiligings- en errata-updates op het basissysteem van FreeBSD zonder de eis om te bouwen en te installeren. Ten tweede ondersteunt het gereedschap kleine en grote uitgave-upgrades.



Binaire updates zijn beschikbaar voor alle architecturen en uitgaveaankondigingen dienen gelezen te worden aangezien deze belangrijke informatie over de gewenste uitgave kunnen bevatten. De aankondigingen kunnen op de volgende koppelin bekeken worden: http://www.FreeBSD.org/releases/.

Als er een `crontab` bestaat die de mogelijkheden van `freebsd-update` gebruikt, moet het uitgeschakeld worden voordat aan de volgende operatie wordt begonnen.

22.2.1. Het configuratiebestand

Sommige gebruikers willen het standaard configuratiebestand optimaliseren, waardoor het proces beter gecontroleerd kan worden. De opties zijn goed gedocumenteerd, maar voor de volgenden is wat extra uitleg nodig:

```
# Componenten van het basissysteem die bijgewerkt moeten blijven
Components src world kernel
```

Deze parameter bepaalt welke delen van FreeBSD bijgewerkt blijven. Standaard wordt de broncode bijgewerkt, het hele basissysteem, en de kernel. Dezelfde componenten als tijdens de installatie zijn beschikbaar, het toevoegen van bijvoorbeeld `world/games` zou de spelpatches toepassen. Het gebruik van `src/bin` zou de broncode in src/bin bijgewerkt houden.

Het beste kan dit op de standaardwaarde blijven aangezien het veranderen hiervan om specifieke items te bevatten de gebruiker dwingt om alle items die bijgewerkt dienen te worden op te noemen. Dit kan rampzalige gevolgen hebben aangezien de broncode en de binairen asynchroon kunnen raken.

```
# Paden die beginnen met iets wat overeenkomt met een regel in een IgnorePaths
# statement zullen genegeerd worden.
IgnorePaths
```

Voeg paden, zoals /bin of /sbin toe om deze specifieke mappen ongemoeid te laten tijdens het updateproces. Deze optie kan gebruikt worden om te voorkomen dat `freebsd-update` lokale wijzigingen overschrijft.

```
# Paden die beginnen met iets wat overeenkomt met een regel in een UpdateIfUnmodified
# statement zullen alleen worden bijgewerkt als de inhoud van het bestand niet is
# gewijzigd door de gebruiker (tenzij veranderingen zijn samengevoegd; zie beneden).
UpdateIfUnmodified /etc/ /var/ /root/ /.cshrc /.profile
```

Werk configuratiebestanden in de aangegeven mappen alleen bij als ze niet zijn gewijzigd. Alle veranderingen die door de gebruiker zijn gemaakt maken het automatisch bijwerken van deze

bestanden ongeldig. Er is een andere optie, `KeepModifiedMetadata`, die `freebsd-update` instrueert om de veranderingen tijdens het samenvoegen te bewaren.

```
# Wanneer naar een nieuwe uitgave van FreeBSD wordt ge-upgraded, worden lokale
veranderingen van bestanden die overeenkomen met MergeChanges
# samengevoegd in de versie van de nieuwe uitgave.
MergeChanges /etc/ /var/named/etc/
```

Lijst van mappen met instellingenbestanden waar `freebsd-update` moet proberen om in samen te voegen. Het proces van bestanden samenvoegen is een serie van diff(1)-patches die ongeveer gelijk is aan mergemaster(8) met minder opties, de samenvoegingen worden ofwel geaccepteerd, of openen een tekstverwerker, of zorgen ervoor dat `freebsd-update` afbreekt. Maak in geval van twijfel een reservekopie van /etc en accepteer de samenvoegingen. In `mergemaster` staat meer informatie over het commando `mergemaster`.

```
# Map waarin de gedownloade updates en tijdelijke
bestanden
# die door een FreeBSD Update worden gebruikt worden opgeslagen.
# WorkDir /var/db/freebsd-update
```

Dit is de map waarin alle patches en tijdelijke bestanden worden geplaatst. In het geval dat de gebruiker een versie-upgrade uitvoert, dient deze locatie tenminste een gigabyte aan vrije schijfruimte te hebben.

```
# Wanneer tussen uitgaven wordt ge-upgraded, dient de lijst van Componenten dan
# strikt gelezen te worden (StrictComponents yes) of slechts als een lijst van
componenten

# die geïnstalleerd *kunnen* worden en waarvan FreeBSD Update uit dient te zoeken
# welke daadwerkelijk zijn geïnstalleerd en die te upgraden (StrictComponents no)?
# StrictComponents no
```

Wanneer ingesteld op `yes`, zal `freebsd-update` aannemen dat de lijst `Components` compleet is en zal het niet proberen om wijzigingen buiten de lijst te maken. Effectief zal `freebsd-update` proberen om elk bestand bij te werken dat op de lijst `Components` staat.

22.2.2. Beveiligingspatches

Beveiligingspatches staan op een verre machine en kunnen met het volgende commando gedownload en geïnstalleerd worden:

```
# freebsd-update fetch
# freebsd-update install
```

Als er kernelpatches zijn toegepast moet het systeem opnieuw opgestart worden. Als alles goed is

gegaan dient het systeem gepatcht te zijn en kan `freebsd-update` als een nachtelijke cron(8)-taak gedraaid worden. Een regel in /etc/crontab zou genoeg moeten zijn om deze taak te volbrengen:

```
@daily                                  root    freebsd-update cron
```

Deze regel verklaart dat eenmaal per dag het commando `freebsd-update` gedraaid zal worden. Op deze manier, door het argument `cron` te gebruiken, zal het gereedschap `freebsd-update` alleen kijken of er updates bestaan. Als er patches bestaan, zullen ze automatisch worden gedownload naar de plaatselijke schijf maar niet worden toegepast. Er zal een email aan de gebruiker `root` worden verstuurd zodat ze handmatig geïnstalleerd kunnen worden.

Als er iets misging, heeft `freebsd-update` de mogelijkheid om de laatste verzamelingen veranderingen terug te draaien met het volgende commando:

```
# freebsd-update rollback
```

Eenmaal voltooid, dient het systeem herstart te worden als de kernel of enige kernelmodule is gewijzigd. Dit stelt FreeBSD in staat om de nieuwe binairen in het geheugen te laden.

Het gereedschap `freebsd-update` kan alleen de kernel GENERIC automatisch bijwerken. Als een eigen kernel wordt gebruikt, moet het herbouwd en geherinstalleerd worden nadat `freebsd-update` klaar is met het installeren de rest van de updates. `freebsd-update` zal echter de kernel GENERIC in /boot/GENERIC detecteren en bijwerken (als het bestaat), zelfs als het niet de huidige (draaiende) kernel van het systeem is.



Het is een goed idee om altijd een kopie van de kernel GENERIC in /boot/GENERIC te bewaren. Het kan van pas komen bij het vaststellen van een keur aan problemen, en bij het uitvoeren van versie-upgrades met `freebsd-update` zoals beschreven in Grote en kleine upgrades.

Tenzij de standaardconfiguratie in /etc/freebsd-update.conf is gewijzigd, zal `freebsd-update` de bijgewerkte kernelbronnen samen met de rest van de updates installeren. Het herbouwen en herinstalleren van uw nieuwe eigen kernel kan daarna op de gebruikelijke manier gedaan worden.



De updates die via `freebsd-update` verspreid worden hebben niet altijd betrekking op de kernel. Het is niet nodig om uw eigen kernel te herbouwen als de kernelbronnen niet zijn aangepast door het uitvoeren van `freebsd-update install`. `freebsd-update install` zal echter altijd het bestand /usr/src/sys/conf/newvers.sh bijwerken. Het huidige patchniveau (zoals aangegeven door het `-p`-nummer gerapporteerd door `uname -r`) wordt uit dit bestand gehaald. Het herbouwen van uw eigen kernel, zelfs als er niets veranderd is, stelt uname(1) in staat om het huidige patchniveau van het systeem accuraat te rapporteren. Dit is in het bijzonder behulpzaam wanneer meerdere systemen onderhouden worden, aangezien hierdoor snel de geïnstalleerde updates op elk ervan kunnen worden nagegaan.

22.2.3. Grote en kleine upgrades

Dit proces ruimt oude objectbestanden en bibliotheken op waardoor de meeste applicaties van derde partijen kapot gaan. Het wordt aangeraden dat alle geïnstalleerde poorten ofwel verwijderd en geherinstalleerd worden of later ge-upgraded worden met het hulpmiddel ports-mgmt/portupgrade. De meeste gebruikers zullen willen proefdraaien met het volgende commando:

```
# portupgrade -af
```

Dit zorgt ervoor dat alles juist wordt geherinstalleerd. Merk op dat het instellen van de omgevingsvariabele `BATCH` op `yes` het antwoord `yes` zal geven op alle prompts tijdens dit proces, waardoor het niet nodig is om handmatig in het bouwproces in te grijpen.

Als een eigen kernel wordt gebruikt, is het upgradeproces iets ingewikkelder. Een kopie van de kernel GENERIC is nodig en dient in /boot/GENERIC geplaatst te worden. Als de kernel GENERIC niet reeds op het systeem aanwezig is, moet het met één van de volgende methoden verkregen worden:

- Als er slechts eenmaal een eigen kernel is gebouwd, dan is de kernel in /boot/kernel.old eigenlijk de GENERIC. Hernoem deze map naar /boot/GENERIC.
- Aannemende dat fysieke toegang tot de machine mogelijk is, kan een kopie van de kernel GENERIC van het CD-ROM-medium worden geïnstalleerd. Laad de installatieschijf en geef de volgende commando's:

  ```
  # mount /cdrom
  # cd /cdrom/X.Y-RELEASE/kernels
  # ./install.sh GENERIC
  ```

  Vervang X.Y-RELEASE met de versie van de uitgave die u gebruikt. De kernel GENERIC zal standaard in /boot/GENERIC worden geïnstalleerd.

- Als al het bovenstaande niet lukt, kan de kernel GENERIC herbouwd en geherinstalleerd worden vanaf de broncode:

  ```
  # cd /usr/src
  # env DESTDIR=/boot/GENERIC make kernel
  # mv /boot/GENERIC/boot/kernel/* /boot/GENERIC
  # rm -rf /boot/GENERIC/boot
  ```

  Om deze kernel door `freebsd-update` als GENERIC te laten herkennen, mag het configuratiebestand voor GENERIC niet op enige wijze veranderd zijn. Het is ook aan te raden dat het zonder andere speciale opties wordt gebouwd (bij voorkeur met een leeg /etc/make.conf).

Opnieuw opstarten naar de kernel GENERIC is in dit stadium niet nodig.

Updates van grote en kleine versies kunnen worden uitgevoerd door een uitgaveversie als doel aan

freebsd-update op te geven, het volgende commando zal bijvoorbeeld updaten naar FreeBSD 8.1:

```
# freebsd-update -r 8.1-RELEASE upgrade
```

Nadat het commando is ontvangen, zal freebsd-update het instellingenbestand en het huidige systeem evalueren in een poging om de benodigde informatie te verzamelen om het systeem te updaten. Een lijst op het scherm zal aangeven welke componenten zijn gedetecteerd en welke niet. Bijvoorbeeld:

```
Looking up update.FreeBSD.org mirrors... 1 mirrors found.
Fetching metadata signature for 8.0-RELEASE from update1.FreeBSD.org... done.
Fetching metadata index... done.
Inspecting system... done.

The following components of FreeBSD seem to be installed:
kernel/smp src/base src/bin src/contrib src/crypto src/etc src/games
src/gnu src/include src/krb5 src/lib src/libexec src/release src/rescue
src/sbin src/secure src/share src/sys src/tools src/ubin src/usbin
world/base world/info world/lib32 world/manpages

The following components of FreeBSD do not seem to be installed:
kernel/generic world/catpages world/dict world/doc world/games
world/proflibs

Does this look reasonable (y/n)? y
```

Nu zal freebsd-update proberen om alle bestanden die nodig zijn voor de upgrade te downloaden. In sommige gevallen kan de gebruiker worden gevraagd wat te installeren of hoe verder te gaan.

Wanneer een eigen kernel wordt gebruikt, zal de bovenstaande stap een waarschuwing geven die lijkt op de volgende:

```
WARNING: This system is running a "MIJNKERNEL" kernel, which is not a
kernel configuration distributed as part of FreeBSD 8.0-RELEASE.
This kernel will not be updated: you MUST update the kernel manually
before running "/usr/sbin/freebsd-update install"
```

Deze waarschuwing kan op dit moment veilig worden genegeerd. De bijgewerkte kernel GENERIC zal als tussenliggende stap in het upgradeproces worden gebruikt.

Nadat alle patches zijn gedownload naar het plaatselijke systeem zullen ze worden toegepast. Dit proces kan afhankelijk van de snelheid en werklast van de machine even duren. Hierna zullen instellingenbestanden worden samengevoegd - voor dit gedeelte van het proces is enige tussenkomst van de gebruiker nodig aangezien een bestand kan worden samengevoegd of omdat er een tekstverwerker op het scherm kan verschijnen om het bestand handmatig samen te voegen. Het resultaat van elke succesvolle samenvoeging zal aan de gebruiker worden getoond naarmate het proces verder gaat. Een mislukte of genegeerde samenvoegpoging zal het proces afbreken. Het

is mogelijk voor gebruikers om een reservekopie van /etc te maken en belangrijke bestanden, zoals master.passwd of group, later samen te voegen.



Het systeem is nog niet veranderd, al het patchen en samenvoegen gebeurt in een andere map. Wanneer alle patches succesvol zijn toegepast, alle instellingenbestanden zijn samengevoegd en het erop lijkt dat het proces soepel verloopt, dienen de veranderingen verzegeld te worden door de gebruiker.

Als dit proces eenmaal voltooid is, kan de upgrade aan de schijf toevertrouwd worden met het volgende commando.

```
# freebsd-update install
```

De kernel en kernelmodules zullen als eerste gepatcht worden. Nu moet de machine opnieuw opgestart worden. Als het systeem een eigen kernel draaide, gebruik dan het commando nextboot(8) om de kernel voor de volgende keer dat opgestart wordt in te stellen op /boot/GENERIC (welke is bijgewerkt):

```
# nextboot -k GENERIC
```



Voordat er met de kernel GENERIC wordt opgestart, dient te worden gecontroleerd dat het alle stuurprogramma's bevat om uw systeem juist te laten opstarten (en met het netwerk te verbinden, als de machine die bijgewerkt wordt van afstand wordt benaderd). In het bijzonder, als de vorige kernel die draaide ingebouwde functionaliteit bevatte die normaalgesproken door kernelmodules wordt geleverd, zorg er dan voor dat deze modules tijdelijk in de kernel GENERIC worden geladen door de faciliteit /boot/loader.conf te gebruiken. U kunt er ook voor kiezen om niet-essentiële diensten, schijf- en netwerkkoppelingen, enzovoorts uit te zetten totdat het upgradeproces voltooid is.

De machine dient nu te worden herstart met de bijgewerkte kernel:

```
# shutdown -r now
```

Als het systeem weer actief is, moet `freebsd-update` nogmaals gestart worden. De toestand van het proces is opgeslagen en dus zal `freebsd-update` niet vooraan beginnen, maar zal het alle oude gedeelde bibliotheken en objectbestanden verwijderen. Geef het volgende commando om verder te gaan op dit punt:

```
# freebsd-update install
```



Afhankelijk van het feit of er versienummers van bibliotheken zijn opgehoogd, kunnen er slechts twee in plaats van drie installatiefasen zijn.

Alle software van derde partijen dient nu opnieuw gebouwd en geïnstalleerd te worden. Dit is nodig omdat geïnstalleerde software van bibliotheken afhankelijk kan zijn die tijdens het upgradeproces zijn verwijderd. Het commando ports-mgmt/portupgrade kan gebruikt worden om dit proces te automatiseren. Dit proces kan met de volgende commando's gestart worden:

```
# portupgrade -f ruby
# rm /var/db/pkg/pkgdb.db
# portupgrade -f ruby18-bdb
# rm /var/db/pkg/pkgdb.db /usr/ports/INDEX-*.db
# portupgrade -af
```

Voltooi, nadat dit voltooid is, het upgradeproces met een laatste aanroep naar `freebsd-update`. Geef het volgende commando om alle losse eindjes in het upgradeproces samen te knopen:

```
# freebsd-update install
```

Als de kernel GENERIC tijdelijk werd gebruikt, is dit het moment om een nieuwe eigen kernel op de gebruikelijke manier te bouwen en installeren.

Start de machine opnieuw op in de nieuwe FreeBSD-versie. Het proces is voltooid.

22.2.4. Het vergelijken van systeemtoestanden

Het gereedschap `freebsd-update` kan gebruikt worden om de toestand van de geïnstalleerde versie van FreeBSD met een bekende goede kopie te vergelijken. Deze optie evalueert de huidige versie van systeemgereedschappen, bibliotheken, en instellingenbestanden. Geef het volgende commando om met de vergelijking te beginnen:

```
# freebsd-update IDS >> uitvoerbestand.ids
```



Hoewel de commandonaam IDS is, is het in geen geval een vervanging voor een indringdetectiesysteem zoals security/snort. Aangezien `freebsd-update` gegevens op schijf opslaat, is de mogelijkheid om te knoeien duidelijk. Hoewel deze mogelijkheid verminderd kan worden door de instelling `kern.securelevel` te gebruiken en de gegevens van `freebsd-update` op een bestandssysteem dat alleen gelezen kan worden op te slaan wanneer deze niet gebruikt worden, zou een betere oplossing zijn om het systeem met een veilige schijf te vergelijken, zoals een DVD of een veilig opgeslagen externe USB-schijf.

Het systeem zal nu geïnspecteerd worden, en er zal een lijst van hun sha256(1)-hashwaarden, zowel de bekende waarde in de uitgave en de huidige geïnstalleerde waarde, afgebeeld worden. Hierom wordt de uitvoer naar het bestand uitvoerbestand.ids gezonden. Het scrollt te snel voorbij om het met het oog te vergelijken, en het vult al snel de gehele consolebuffer op.

Deze regels zijn ook extreem lang, maar het uitvoerformaat kan vrij eenvoudig geparsed worden. Geef, om bijvoorbeeld een lijst van alle bestanden te krijgen die verschillen van die in de uitgave,

het volgende commando:

```
# cat uitvoerbestand.ids | awk '{ print $1 }' | more
/etc/master.passwd
/etc/motd
/etc/passwd
/etc/pf.conf
```

Deze uitvoer is afgekapt, er bestaan veel meer bestanden. Sommige van deze bestanden hebben natuurlijke veranderingen, het /etc/passwd is gewijzigd omdat er gebruikers aan het systeem zijn toegevoegd. In sommige gevallen kunnen er andere bestanden zijn, zoals kernelmodules, die verschillen aangezien `freebsd-update` ze ge-updated kan hebben. Voeg, om bepaalde bestanden of mappen uit te sluiten, deze toe aan de optie `IDSIgnorePaths` in /etc/freebsd-update.conf.

Dit systeem kan gebruikt worden als deel van een uitgebreide upgrademethode, afgezien van de eerder besproken versie.

22.3. Portsnap: een updategereedschap voor de Portscollectie

Het basissysteem van FreeBSD bevat ook een gereedschap om de Portscollectie bij te werken: het hulpmiddel portsnap(8). Wanneer het wordt uitgevoerd, zal het een verbinding maken met een verre site, de veilige sleutel controleren, en een nieuwe kopie van de Portscollectie downloaden. De sleutel wordt gebruikt om de integriteit van alle gedownloade bestanden te controleren, om er zeker van te zijn dat ze niet tijdens het downloaden zijn gewijzigd. Geef het volgende commando om de nieuwste versie van de bestanden van de Portscollectie te downloaden:

```
# portsnap fetch
Looking up portsnap.FreeBSD.org mirrors... 9 mirrors found.
Fetching snapshot tag from geodns-1.portsnap.freebsd.org... done.
Fetching snapshot metadata... done.
Updating from Tue May 22 02:12:15 CEST 2012 to Wed May 23 16:28:31 CEST 2012.
Fetching 3 metadata patches.. done.
Applying metadata patches... done.
Fetching 3 metadata files... done.
Fetching 90 patches.....10....20....30....40....50....60....70....80....90. done.
Applying patches... done.
Fetching 133 new ports or files... done.
```

Dit voorbeeld laat zien dat portsnap(8) verscheidene patches heeft gevonden en deze met de huidige portsgegevens heeft gecontroleerd. Het geeft ook aan dat het gereedschap eerder is gedraaid, als het voor de eerste keer was gedraaid, had het simpelweg de collectie gedownload.

Wanneer portsnap(8) succesvol een `fetch`-operatie afrondt, bestaan de Portscollectie en de vervolgpatches die de verificatie doorstaan hebben op het plaatselijke systeem. Gebruik de eerste keer dat `portsnap` wordt uitgevoerd `extract` om de gedownloade bestanden te installeren:

```
# portsnap extract
/usr/ports/.cvsignore
/usr/ports/CHANGES
/usr/ports/COPYRIGHT
/usr/ports/GIDs
/usr/ports/KNOBS
/usr/ports/LEGAL
/usr/ports/MOVED
/usr/ports/Makefile
/usr/ports/Mk/bsd.apache.mk
/usr/ports/Mk/bsd.autotools.mk
/usr/ports/Mk/bsd.cmake.mk
...
```

Om een reeds geïnstalleerde Ports Collectie te updaten kan er gebruik worden gemaakt van het commando `portsnap update`:

```
# portsnap update
```

Het proces is nu compleet, en applicaties kunnen met de bijgewerkte Portscollectie worden geïnstalleerd of worden bijgewerkt.

De bewerkingen `fetch` en `extract` of `update` kunnen achter elkaar uitgevoerd worden, zoals het volgende voorbeeld laat zien:

```
# portsnap fetch update
```

Dit commando zal de laatste versie van de Ports Collectie downloaden en de lokale versie bijwerken in de /usr/ports.

22.4. De documentatie bijwerken

Naast het basissysteem en de Portscollectie is documentatie een integraal onderdeel van het besturingssysteem FreeBSD. Hoewel een actuele versie van de FreeBSD-documentatie altijd beschikbaar is op de FreeBSD website, hebben sommige gebruikers een langzame of helemaal geen permanente netwerkverbinding. Gelukkig zijn er verschillende manieren om de documentatie die bij elke uitgave wordt geleverd bij te werken door een lokale kopie van de nieuwste FreeBSD-documentatie bij te houden.

22.4.1. Subversion gebruiken om de documentatie bij te werken

De bronnen van de FreeBSD-documentatie kunnen met Subversion worden bijgewerkt. Deze sectie beschrijft:

- Hoe de documentatiegereedschappen, de gereedschappen die nodig zijn om de FreeBSD-documentatie vanuit de broncode te herbouwen, te installeren.

- Hoe een kopie van de documentatiebronnen in /usr/doc te downloaden door Subversion te gebruiken.
- Hoe de FreeBSD-documentatie vanuit de broncode te herbouwen en onder /usr/shared/doc te installeren.
- Sommige bouwopties die door het bouwsysteem van de documentatie ondersteund worden, i.e., de opties die slechts enkele van de verschillende vertalingen van de documentatie bouwen of de opties die een specifiek uitvoerformaat selecteren.

22.4.2. Subversion en de documentatiegereedschappen installeren

Voor het herbouwen van de FreeBSD-documentatie vanuit de broncode is een aardig grote verzameling gereedschappen nodig. Deze gereedschappen zijn geen deel van het basissysteem van FreeBSD omdat ze een grote hoeveelheid schijfruimte nodig hebben en niet voor alle FreeBSD-gebruikers nuttig zijn; ze zijn alleen nuttig voor die gebruikers die actief nieuwe documentatie voor FreeBSD schrijven of regelmatig hun documentatie vanuit de broncode bijwerken.

Alle benodigde gereedschappen zijn beschikbaar als deel van de Portscollectie. De port textproc/docproj is een meester-port die door het FreeBSD Documentatieproject is ontwikkeld om de installatie en toekomstige updates van deze gereedschappen makkelijker te maken.



Wanneer er geen PostScript®- of PDF-documentatie nodig is, kan men overwegen om in plaats hiervan de port textproc/docproj-nojadetex te installeren. Deze versie van de documentatiegereedschappen bevat alles behalve de typesetting-engine teTeX. teTeX is een erg grote verzameling van gereedschappen, dus kan het zinvol zijn om de installatie ervan achterwege te laten als PDF-uitvoer niet echt nodig is.

Subversion wordt geïnstalleerd met de port textproc/docproj.

22.4.3. De documentatiebroncode bijwerken

Het programma Subversion kan een schone kopie van de documentatiebroncode ophalen door het volgende te typen:

```
# svn checkout svn://svn.FreeBSD.org/doc/head /usr/doc
```

De initiële download van de documentatiebroncode kan een tijd duren. Laat het draaien totdat het voltooid is.

Toekomstige updates van de documentatiebroncode kunnen opgehaald worden door het volgende commando te draaien:

```
# svn update /usr/doc
```

Nadat de broncode is uitgecheckt, wordt een alternatieve manier om de documentatie bij te werken ondersteund door Makefile van de map /usr/doc door het volgende te draaien:

```
# cd /usr/doc
# make update
```

22.4.4. Instelbare opties van de documentatiebroncode

Het bijwerk- en bouwsysteem van de FreeBSD-documentatie ondersteunt enkele opties die het proces om de documentatie alleen gedeeltelijk bij te werken, of om specifieke vertalingen te bouwen, makkelijker maken. Deze opties kunnen of als systeemwijde opties in het bestand /etc/make.conf worden ingesteld, of als opdrachtregelopties aan het hulpmiddel make(1) worden doorgegeven.

De volgende opties zijn er enkelen van:

`DOC_LANG`

De lijst van te bouwen en te installeren talen en coderingen, bijvoorbeeld `en_US.ISO8859-1` voor alleen de Engelse documentatie.

`FORMATS`

Een enkel formaat of een lijst van uitvoerformaten die gebouwd moeten worden. Momenteel worden `html`, `html-split`, `txt`, `ps`, `pdf`, en `rtf` ondersteund.

`DOCDIR`

Waar de documentatie te installeren. Dit staat standaard op /usr/shared/doc.

Bekijk make.conf(5) voor meer make-variabelen die als systeemwijde opties in FreeBSD worden ondersteund.

Voor meer make-variabelen die door het bouwsysteem van de FreeBSD-documentatie ondersteund worden, wordt naar het FreeBSD Documentation Project Primer for New Contributors verwezen.

22.4.5. De FreeBSD-documentatie vanuit de broncode installeren

Wanneer er een actueel snapshot van de documentatiebroncode is opgehaald in /usr/doc, is alles gereed om de geïnstalleerde documentatie bij te werken.

Het volledig bijwerken van alle talen die in de Makefile-optie `DOC_LANG` zijn gedefinieerd kan worden gedaan door te typen:

```
# cd /usr/doc
# make install clean
```

Als alleen het bijwerken van een specifieke taal gewenst is, dan kan make(1) worden aangeroepen in een taalspecifieke submap van /usr/doc, i.e.:

```
# cd /usr/doc/en_US.ISO8859-1
# make update install clean
```

De te installeren uitvoerformaten kunnen worden gespecificeerd door de make-variabele FORMATS in te stellen, i.e.:

```
# cd /usr/doc
# make FORMATS='html html-split' install clean
```

22.4.6. Documentatieports gebruiken

In de vorige sectie werd er een methode voor het bijwerken van de FreeBSD-documentatie vanaf de broncode gepresenteerd. Het bijwerken gebaseerd op broncode is echter niet voor alle FreeBSD-systemen haalbaar of praktisch. Voor het bouwen van de documentatiebronnen zijn een redelijk grote verzameling van gereedschappen, de *documentatie gereedschapskist*, een bepaald niveau van bekendheid met Subversion en checkouts van broncode vanuit een reservoir nodig, en een aantal handmatige stappen om de uitgecheckte broncode te bouwen. In deze sectie wordt een alternatieve manier beschreven om de geïnstalleerde kopiën van de FreeBSD-documentatie bij te werken; een die de Ports Collectie gebruikt en het mogelijk maakt om:

- Voorgebouwde versies van de documentatie te downloaden en te installeren, zonder iets lokaal te hoeven bouwen (op deze manier wordt de noodzaak voor een installatie van de gehele documentatie-gereedschapskist voorkomen).
- De documentatiebronnen te bouwen en ze via het ports-raamwerk te bouwen (de stappen van het uitchecken en bouwen worden iets eenvoudiger gemaakt).

Deze twee methoden om de FreeBSD-documentatie bij te werken worden ondersteund door een verzameling van *documentatie-ports* die maandelijks door het Documentatie Engineering Team <doceng@FreeBSD.org> worden bijgewerkt. Deze zijn vermeld in de FreeBSD Ports Collectie onder de virtuele categorie docs.

Documentatie-ports bouwen en installeren

De documentatie-ports gebruiken het bouwraamwerk van de ports om het bouwen van documentatie eenvoudiger te maken. Ze automatiseren het proces van het uitchecken van de broncode van de documentatie, het draaien van make(1) met de juiste omgevingsinstellingen en opdrachtregelopties, en ze maken de installatie of deïnstallatie van documentatie net zo eenvoudig als de installatie van elke andere FreeBSD-port of -pakket.



Als een extra eigenschap registreren de documentatie-ports, wanneer ze lokaal zijn gebouwd, een afhankelijkheid naar de ports van de *documentatie-gereedschapskist*, zodat de laatste ook automatisch is geïnstalleerd.

De organisatie van de documentatie-ports is als volgt:

- Er is een "meester-port", misc/freebsd-doc-en, waar de bestanden van de documentatie-ports gevonden kunnen worden. Het is de basis van alle documentatie-ports. Standaard bouwt het alleen de Engelstalige documentatie.
- Er is een "alles-in-één port", misc/freebsd-doc-all, en het bouwt en installeert alle documentatie in alle beschikbare talen.

- Ten slotte is er een "slaaf-port" voor elke vertaling, bijvoorbeeld misc/freebsd-doc-hu voor de documenten in het Hongaars. Ze zijn allemaal afhankelijk van de meester-port en installeren de vertaalde documentatie van de respectievelijke taal.

Gebruik de volgende commando's (als `root`) om een documentatieport vanaf de broncode te installeren:

```
# cd /usr/ports/misc/freebsd-doc-en
# make install clean
```

Dit zal de Engelstalige documentatie in gesplitst HTML-formaat (hetzelfde als dat op http://www.FreeBSD.org wordt gebruikt) in de map /usr/local/shared/doc/freebsd bouwen en installeren.

Algemene knoppen en opties

Er zijn vele opties om het standaardgedrag van de documentatie-ports aan te passen. Het volgende is slechts een korte lijst:

`WITH_HTML`
: Staat bouwen van het HTML-formaat toe: een enkel HTML-bestand per document. De opgemaakte documentatie wordt naar gelang in een bestand genaamd article.html, of book.html, met afbeeldingen opgeslagen.

`WITH_PDF`
: Staat bouwen van het Adobe® Portable Document Format toe, te gebruiken met Adobe® Acrobat Reader®, Ghostscript, of andere PDF-lezers. De opgemaakte documentatie wordt naar gelang opgeslagen in een bestand genaamd article.pdf of book.pdf opgeslagen.

`DOCBASE`
: Waar de documentatie te installeren. Standaard is dit /usr/local/shared/doc/freebsd.



Merk op dat de standaard doelmap afwijkt van de map die door de Subversion-methode wordt gebruikt. Dit komt omdat er een port wordt geïnstalleerd, en ports worden normaliter onder de map /usr/local geïnstalleerd. Dit kan veranderd worden door de variabele `PREFIX` toe te voegen.

Hier is een kort voorbeeld over hoe de bovengenoemde variabelen te gebruiken om de Hongaarse documentatie in Portable Document Format te installeren:

```
# cd /usr/ports/misc/freebsd-doc-hu
# make -DWITH_PDF DOCBASE=share/doc/freebsd/hu install clean
```

Documentatiepakketten gebruiken

Voor het bouwen van de documentatie-ports vanaf broncode, zoals beschreven in de vorige sectie, is een lokale installatie van de documentatie-gereedschapskist en wat schijfruimte voor het bouwen

van de ports nodig. Wanneer de bronnen voor het installeren van de documentatie-gereedschapskist niet aanwezig zijn, of wanneer het bouwen vanaf broncode te veel schijfruimte in beslag neemt, is het nog steeds mogelijk om de vooraf gebouwde versies van de documentatie-ports te installeren.

Het Documentatie Engineering Team <doceng@FreeBSD.org> bereidt maandelijkse versies van de FreeBSD documentatiepakketten voor. Deze binaire pakketten kunnen met elk van de meegeleverde pakketgereedschappen, zoals pkg_add(1), pkg_delete(1), enzovoorts gebruikt worden.



Wanneer binaire pakketten worden gebruikt, zal de FreeBSD documentatie in *alle* beschikbare formaten voor de gegeven taal geïnstalleerd worden.

Het volgende commando bijvoorbeeld zal het nieuwste vooraf gebouwde pakket van de Hongaarse documentatie installeren:

```
# pkg_add -r hu-freebsd-doc
```



Pakketten hebben het volgende naamformaat welke afwijkt van de naam van de overeenkomstige port: `taal-freebsd-doc`. Hier is *taal* het korte formaat van de taalcode, i.e., `hu` voor Hongaars, of `zh_cn` voor Vereenvoudigd Chinees.

Documentatieports bijwerken

Voor het bijwerken van een eerder geïnstalleerde documentatieport is elk gereedschap voor het bijwerken van ports geschikt. Het volgende commando bijvoorbeeld werkt de geïnstalleerde Hongaarse documentatie bij via het gereedschap ports-mgmt/portupgrade door alleen pakketten te gebruiken:

```
# portupgrade -PP hu-freebsd-doc
```

22.5. Een ontwikkelingstak volgen

Er zijn twee ontwikkeltakken voor FreeBSD: FreeBSD-CURRENT en FreeBSD-STABLE. Deze sectie licht beiden toe en beschrijft hoe een systeem bijgewerkt te houden met elke tak. FreeBSD-CURRENT wordt eerst behandeld, daarna FreeBSD-STABLE.

22.5.1. Bijblijven met FreeBSD

Bedenk dat FreeBSD-CURRENT het "nieuwste van het nieuwste" is van FreeBSD ontwikkeling. Van FreeBSD-CURRENT gebruikers wordt verwacht dat ze veel technische kennis hebben en capabel zijn om zelfstandig lastige systeemproblemen op te lossen. Nieuwe gebruikers van FreeBSD kunnen het beste twee keer nadenken alvorens het te installeren.

Wat is FreeBSD-CURRENT?

FreeBSD-CURRENT is de laatste werkende set broncode voor FreeBSD. Dit bevat werk in uitvoering,

experimentele wijzigingen en overgangsmechanismes die mogelijk wel of niet meegenomen worden in de volgende officiële uitgave van het besturingssysteem. Alhoewel veel FreeBSD-ontwikkelaars de broncode van FreeBSD-CURRENT dagelijks compileren, zijn er periodes dat de broncode niet compileerbaar is. Deze problemen worden zo snel mogelijk gerepareerd, maar het is mogelijk dat FreeBSD-CURRENT een ramp veroorzaakt in plaats van dat het de gewenste functionaliteit levert. Dit ligt geheel aan het moment waarop de broncode is opgehaald.

Wie heeft FreeBSD-CURRENT nodig?

FreeBSD-CURRENT is beschikbaar voor drie primaire aandachtsgroepen:

1. Leden van de FreeBSD-gemeenschap die actief werken aan een deel van de broncode voor wie "current" een echte eis is.
2. Leden van de FreeBSD-gemeenschap die actief testen en tijd hebben om problemen op te lossen om zeker te stellen dat FreeBSD-CURRENT zo gezond als mogelijk is. Er zijn ook mensen die actuele suggesties maken over wijzigingen en de algemene richting van FreeBSD en die patches opsturen om deze te implementeren.
3. Diegenen die alleen een oogje in het zeil willen houden of de huidige bronnen gebruiken ter referentie (bijvoorbeeld voor het *lezen* en niet het draaien). Deze mensen geven ook regelmatig commentaar of dragen bij in de code.

Wat is FreeBSD-CURRENT *niet*?

1. Een snelle manier om pre-release versies te krijgen omdat bekend is dat er een aantal leuke nieuwe mogelijkheden in zitten en het leuk is deze als eerste te gebruiken. Het als eerste gebruiken van nieuwe mogelijkheden betekent ook de eerste zijn die nieuwe bugs ontdekt.
2. Een snelle manier om bugfixes te krijgen. Elke willekeurige versie van FreeBSD-CURRENT heeft waarschijnlijk net zoveel nieuwe bugs als dat er bugs opgelost zijn.
3. Op welke manier dan ook "officieel ondersteund". We doen onze best om mensen echt te helpen in één van de drie "legitieme" FreeBSD-CURRENT groepen maar er is simpelweg *niet genoeg tijd* om technische ondersteuning te leveren. Dit is niet omdat we gemene en vervelende mensen zijn die anderen niet willen helpen (we zouden niet eens aan FreeBSD werken als we dat durfden). De ontwikkelaars kunnen simpelweg geen honderd berichten per dag beantwoorden *én* aan FreeBSD werken. Bij de keuze tussen het verbeteren van FreeBSD en vragen beantwoorden over experimentele code, kiezen ontwikkelaars voor het eerste.

FreeBSD-CURRENT gebruiken

1. Neem een abonnement op de mailinglijsten FreeBSD-CURRENT mailinglijst en SVN-commitberichten voor de src-structuur voor head/-current. Dit is niet alleen een goed idee, het is *essentieel*. Geen berichten ontvangen van de lijst *FreeBSD-CURRENT mailinglijst* betekent geen commentaar zien dat mensen maken over de huidige staat van het systeem en dus waarschijnlijk struikelen over problemen die anderen al gevonden en opgelost hebben. Nog belangrijker is het missen van belangrijke informatie die kritisch kan zijn voor een systeem.

   De lijst SVN-commitberichten voor de src-structuur voor head/-current biedt de mogelijkheid de wijzigingsboodschap te zien voor elke wijziging die gemaakt wordt, samen met relevante informatie over mogelijke bijwerkingen.

Ga om op deze lijsten of één van de andere beschikbare lijsten te abonneren naar https://lists.freebsd.org en klik op de gewenste lijst. Instructies over de rest van de procedure zijn daar beschikbaar. Als u geïnteresseerd bent in het volgen van veranderingen voor de gehele broncodeboom, raden wij u aan een abonnement te nemen op de lijst SVN-commitberichten voor de gehele src-structuur (behalve voor "user" en "projects").

2. Haal de broncode van een FreeBSD mirrorsite. Dit kan op de volgende twee manieren:

   a. Gebruik het programma cvsup met de supfile genaamd standard-supfile uit /usr/shared/examples/cvsup. Dit is de geadviseerde methode, omdat de gehele collectie in één keer wordt binnengehaald en daarna alleen hetgeen wat gewijzigd is. Veel mensen draaien `cvsup` vanuit de `cron` en houden daarmee hun broncode automatisch bijgewerkt. De voorbeeld supfile dient aangepast te worden om cvsup in te stellen voor uw omgeving.



Het voorbeeld standard-supfile is bedoeld om een specifieke beveiligingstak van FreeBSD te volgen, niet FreeBSD-CURRENT. U moet dit bestand bewerken en de volgende regel vervangen:

```
*default release=cvs tag=RELENG_X_Y
```

door deze:

```
*default release=cvs tag=.
```

Voor een gedetailleerde uitleg over bruikbare tags wordt naar de sectie CVS Tags van het Handboek verwezen.

   b. Gebruik de faciliteit CTM. Bij een "slechte verbinding", dure connecties of alleen e-mail toegang, is CTM een optie. Het werkt echter lastig en geeft mogelijk corrupte bestanden. Dit zorgt ervoor dat het zelden gebruikt wordt, dat de kans verhoogt dat het niet werkt voor redelijk lange periodes. Het advies is CVSup te gebruiken.

3. Als de broncode wordt opgehaald om te draaien en niet alleen om naar te kijken, haal dan *alles* op van FreeBSD-CURRENT en niet alleen geselecteerde delen. De reden hiervoor is dat verschillende delen van de code afhangen van updates op andere plekken en het compileren van een onderdeel gegarandeerd problemen oplevert.

   Voordat FreeBSD-CURRENT gecompileerd wordt is het raadzaam om de Makefile in /usr/src aandachtig te bekijken. Het is handig om de eerste keer op zijn minst de kernel en de "wereld" opnieuw te bouwen als onderdeel van het updateproces. Via de FreeBSD-CURRENT mailinglijst en /usr/src/UPDATING is het mogelijk op de hoogte te blijven van mogelijke wijzigingen in de opstartprocedures die soms nodig zijn tussen verschillende versies.

4. Wees actief! Ervaringen van FreeBSD-CURRENT-gebruikers zijn belangrijk, zeker als het gaat om suggesties voor verbeteringen of bugfixes. Suggesties met bijbehorende code worden enthousiast ontvangen!

22.5.2. FreeBSD stabiel houden

Wat is FreeBSD-STABLE?

FreeBSD-STABLE is de ontwikkeltak waaruit grote releases gemaakt worden. Wijzigingen in deze tak gaan in een ander tempo en met de algemene aanname dat ze eerst in FreeBSD-CURRENT worden ingebracht ter test. Dit is *nog steeds* een ontwikkeltak, echter dit betekent dat op elk gegeven moment de code voor FreeBSD-STABLE wel of niet geschikt is voor een speciaal doel. Het is simpelweg een andere ontwikkelomgeving en geen bron voor eindgebruikers.

Wie heeft FreeBSD-STABLE nodig?

Bij interesse in het bijhouden van of bijdragen aan het FreeBSD-ontwikkelproces, speciaal als het gerelateerd is aan de volgende versie van FreeBSD, is het volgen van FreeBSD-STABLE het overwegen waard.

Ondanks dat security fixes ook in de FreeBSD-STABLE-tak komen, hoeft dit *niet* per se. In elke beveiligingswaarschuwing voor FreeBSD wordt uitgelegd uit hoe het probleem opgelost kan worden voor de release die het betreft. Het volgen van de volledige ontwikkeltak alleen om veiligheidsredenen levert ongetwijfeld ongewenste wijzigingen op.

Ondanks het voornemen ervoor te zorgen dat de FreeBSD-STABLE-tak compileert en altijd draait, wordt dit niet gegarandeerd. Terwijl code ontwikkeld wordt in FreeBSD-CURRENT voordat die in FreeBSD-STABLE verwerkt wordt, draaien meer mensen FreeBSD-STABLE dan FreeBSD-CURRENT, dus het is onontkoombaar dat bugs en randgevallen soms in FreeBSD-STABLE gevonden worden die niet in FreeBSD-CURRENT bekend waren.

Om deze redenen wordt *niet* aangeraden FreeBSD-STABLE blindelings te volgen en het is extra belangrijk geen productieservers bij te werken naar FreeBSD-STABLE zonder de code te testen in een testomgeving.

Als de mogelijkheden om dit te doen niet beschikbaar zijn, dan is het advies de meest recente release van FreeBSD te draaien en dan de binaire update methode te hanteren om bij te werken tussen verschillende releases.

FreeBSD-STABLE gebruiken

1. Neem een abonnement op de lijst FreeBSD-STABLE; mailinglijst. Deze biedt informatie over onderdelen van de build die mogelijk verschijnen in FreeBSD-STABLE of eventuele andere kwesties die speciale aandacht vereisen. Ontwikkelaars kondigen in deze mailinglijst ook aan wanneer ze overwegen om een controversiële fix of aanpassing willen maken, waardoor de gebruikers een kans hebben om te reageren als ze goede redenen hebben tegen de voorgestelde wijziging.

   Wordt lid van de relevante SVN-lijst voor de tak die u volgt. Als u bijvoorbeeld de tak 7-STABLE volgt, wordt u lid van de svn-src-stable-7 lijst. Dit stelt u in staat om het commit-log-bericht te bekijken voor elke verandering die is gemaakt, tezamen met relevante informatie over mogelijke bijwerkingen.

   Ga om te abonneren op deze lijsten, of één van de andere beschikbare lijsten naar

https://lists.freebsd.org en klik op de lijst waarop een abonnement gewenst is. Instructies over de rest van de procedure zijn daar beschikbaar. Als u geïnteresseerd bent in het volgen van veranderingen voor de gehele broncodeboom, raden wij u aan een abonnement te nemen op de SVN-commitberichten voor de gehele src-structuur (behalve voor "user" en "projects") lijst.

2. Kijk op de webpagina Snapshots om een systeem te installeren van een maandelijkse snapshot van FreeBSD-STABLE. Het is ook mogelijk om de meest recente FreeBSD-STABLE release te installeren van de mirrorsites. Volg de onderstaande instructies om een systeem bij te werken naar de meest recente FreeBSD-STABLE broncode.

   Als al een vorige release van FreeBSD draait en bijgewerkt moet worden via de broncodes dan kan dat via de FreeBSD mirrorsites. Dit kan op één van de twee volgende manieren:

   a. Gebruik het programma cvsup met de supfile stable-supfile uit de map /usr/shared/examples/cvsup. Dit is de aanbevolen methode omdat het hiermee mogelijk is de volledige collectie te downloaden en daarna alleen hetgeen wat veranderd is. Veel mensen draaien `cvsup` vanuit de `cron` om de broncodes automatisch bij te werken. Het voorbeeld van de supfile dient aangepast en ingesteld te worden voor de omgeving waarin het instellingenbestand gebruikt wordt.

   b. Gebruik CTM als er geen snelle, goedkope verbinding is met internet. Dan is dit de methode om te gebruiken.

3. Als er snelle on-demand toegang nodig is tot de broncode en bandbreedte is geen overweging, gebruik dan `cvsup` of `ftp`. Gebruik anders CTM.

4. Lees alvorens FreeBSD-STABLE te compileren goed de Makefile in /usr/src. Het is handig om de eerste keer op zijn minst de kernel en de "wereld" opnieuw te bouwen als onderdeel van het updateproces. Via de FreeBSD-STABLE; mailinglijst en /usr/src/UPDATING is het mogelijk op de hoogte te blijven van mogelijke wijzigingen in de opstartprocedures die soms nodig zijn tussen verschillende releases.

22.6. Broncode synchroniseren

Er zijn verschillende manieren om een internet (of e-mail) verbinding te gebruiken om bij te blijven met elk onderdeel van de FreeBSD projectbronnen of alle onderdelen, afhankelijk van het interessegebied. De primaire diensten zijn Anonieme CVS en CTM.



Ondanks dat het mogelijk is om alleen delen van de broncode bij te werken, is de enige ondersteunde methode de totale broncode bijwerken en zowel userland (alle programma's die in gebruikersruimte draaien, zoals programma's in /bin en /sbin) als de kernel opnieuw compileren. Als alleen delen van de broncode worden bijgewerkt, alleen de kernel of alleen het userland, resulteert dat vaak in problemen. Deze problemen kunnen verschillen van compileerfouten tot kernel panics of corruptie van gegevens.

Anonieme CVS en CVSup gebruiken het *pull* model om broncode bij te werken. In het geval van CVSup start de gebruiker (of een `cron` script) het programma `cvsup` waarbij het communiceert met een `cvsupd` server om bestanden bij te werken. De ontvangen updates zijn op de minuut nauwkeurig en ze komen alleen wanneer dat is ingesteld. Updates kunnen eenvoudig beperkt

worden tot specifieke bestanden of mappen uit een interessegebied. Updates worden automatisch gegenereerd door een server, aan de hand van wat is ingesteld. Anonieme CVS is veel eenvoudiger dan CVSup omdat dat alleen een uitbreiding is van CVS die de mogelijkheid biedt om wijzigingen direct van een CVS repository op afstand te halen. CVSup kan dit veel efficiënter doen, maar anonieme CVS is makkelijker in het gebruik.

CTM aan de andere kant maakt geen vergelijking tussen de aanwezige bronnen en die op de master server. In plaats daarvan wordt een script uitgevoerd dat wijzigingen in bestanden ziet sinds de vorige keer dat is bijgewerkt en die meerdere keren per dag worden uitgevoerd op de master CTM machine. Elke ontdekte wijziging wordt gecomprimeerd, krijgt een volgnummer toegekend en wordt gecodeerd voor verzending via e-mail (in leesbare ASCII). Deze "CTM delta's" kunnen dan aangeleverd worden aan ctm_rmail(1) die ze automatisch decodeert, controleert en toepast in de gebruikerskopie van de bronnen. Dit proces is veel efficiënter dan CVSup en claimt minder systeembronnen omdat het model *push* in plaats van *pull* is.

Er zijn andere nadelen. Als per ongeluk een deel van het archief wordt verwijderd, kan CVSup dat detecteren en het beschadigde deel repareren. CTM doet dit niet en als een deel van de broncode wordt verwijderd (en er geen back-up is), dan moet er opnieuw begonnen worden (vanaf de meest recente CVS "base delta" en moet alles opnieuw opgebouwd worden met CTM. Met Anonymous CVS kan simpelweg het slechte deel verwijderd worden alvorens weer te synchroniseren.

22.7. De "wereld" opnieuw bouwen

Zodra de lokale broncode gesynchroniseerd is met een bepaalde versie van FreeBSD (FreeBSD-STABLE, FreeBSD-CURRENT, enzovoort) kan de broncode gebruikt worden om een systeem te herbouwen.



Maak een back-up

Het kan niet vaak genoeg verteld worden hoe belangrijk het is om een back-up te maken van een systeem *vóór* deze taak uit te voeren. Ook al is het opnieuw bouwen van de wereld vrij simpel (als deze instructies gevolgd worden), er worden ongetwijfeld ooit fouten gemaakt, misschien zelfs in de broncode, die het onmogelijk maken om een systeem op te starten.

Wees ervan verzekerd dat er een back-up gemaakt is en dat er een reparatiediskette of cd-rom bij de hand is. Deze wordt waarschijnlijk nooit gebruikt maar "better safe than sorry".



Abonneer op de juiste mailinglijsten

De FreeBSD-STABLE en FreeBSD-CURRENT takken zijn van nature *in ontwikkeling*. Mensen die bijdragen aan FreeBSD zijn menselijk en foutjes ontstaan regelmatig.

Soms zijn deze foutjes onschadelijk, ze geven dan hooguit een nieuwe diagnostische waarschuwing weer. Maar de wijziging kan ook catastrofaal zijn en ervoor zorgen dat een systeem niet meer opstart of bestandssystemen vernietigt (of erger).

Als problemen zoals deze voorkomen wordt er een "heads up" naar de juiste

mailinglijst gestuurd, waarin uitgelegd wordt wat het probleem is en welke systemen het raakt. Er wordt een "all clear" bericht gestuurd als het probleem is opgelost.

FreeBSD-STABLE of FreeBSD-CURRENT volgen zonder de FreeBSD-STABLE; mailinglijst of FreeBSD-CURRENT mailinglijst te volgen is vragen om problemen.



Gebruik geen `make world`

Veel oudere documentatie raadt aan om `make world` te gebruiken. In dat geval worden er belangrijke stappen overgeslagen en gebruik het commando alleen als er voldoende kennis over aanwezig is. In bijna alle omstandigheden is `make world` verkeerd en de procedure die hier beschreven is hoort in plaats daarvan gebruikt te worden.

22.7.1. De universele wijze om een systeem bij te werken

Om uw systeem bij te werken, dient u /usr/src/UPDATING te controleren op eventuele pre-buildworld stappen die nodig zijn voor uw versie van de broncode en daarna de procedure te gebruiken die hier beschreven staat.

Deze bijwerkstappen nemen aan dat u nu een oude versie van FreeBSD gebruikt, die uit een oude compiler, een oude kernel, een oude wereld en oude instellingenbestanden bestaat. Onder "wereld" worden de binairen, bibliotheken, en programmeerbestanden van het kernsysteem verstaan. De compiler is deel van "wereld", maar heeft enkele speciale aandachtspunten.

We nemen ook aan dat u reeds de broncode van een nieuwer systeem heeft verkregen. Bekijk, als de bronnen op een bepaald systeem ook oud zijn, Broncode synchroniseren voor uitgebreide hulp over het synchroniseren ervan naar een nieuwere versie.

Het bijwerken van het systeem vanaf de broncode is wat subtieler dan het op het eerste gezicht lijkt, en de ontwikkelaars van FreeBSD vonden het in de loop der jaren nodig om de aangeraden methode redelijk drastisch te veranderen met het aan het licht komen van nieuwe soorten onontwijkbare afhankelijkheden. De rest van deze sectie beschrijft de rationale achter de huidige aanbevolen bijwerkmethode.

Elke succesvolle bijwerkmethode krijgt te maken met de volgende punten:

- Het kan voorkomen dat de oude compiler de nieuwe kernel niet kan compileren. (Oude compilers bevatten soms bugs.) De nieuwe kernel dient dus met de nieuwe compiler gebouwd te worden. In het bijzonder moet de nieuwe compiler gebouwd worden voordat de nieuwe kernel gebouwd wordt. Dit betekent niet per se dat de nieuwe compiler *geïnstalleerd* moet worden voordat de nieuwe kernel gebouwd wordt.
- De nieuwe wereld kan afhankelijk zijn van mogelijkheden van de nieuwe kernel. Dus moet de nieuwe kernel worden geïnstalleerd voordat de nieuwe wereld wordt geïnstalleerd.

De eerste twee gevallen zijn de basis voor de methode `buildworld`, `buildkernel`, `installkernel`, `installworld` die we in de volgende paragrafen beschrijven. Dit is geen uitputtende lijst van alle redenen waarom het huidige aanbevolen bijwerkproces de voorkeur verdient. Wat minder voor de

hand liggende redenen worden hieronder genoemd:

- Het kan zijn dat de oude wereld niet correct draait op de nieuwe kernel, dus moet de nieuwe wereld onmiddellijk na het installeren van de nieuwe kernel geïnstalleerd worden.
- Sommige instellingen moeten veranderd worden voordat de nieuwe wereld wordt geïnstalleerd, maar anderen kunnen de oude wereld kapot maken. Vandaar dat over het algemeen twee verschillende bijwerkstappen voor de instellingen nodig zijn.
- Voor het grootste gedeelte houdt het bijwerkproces zich alleen bezig met het vervangen of toevoegen van bestanden; bestaande oude bestanden worden niet verwijderd. Dit kan in sommige gevallen problemen geven. Als een gevolg zal de bijwerkprocedure soms aangeven dat bepaalde bestanden tijdens bepaalde stappen handmatig verwijderd dienen te worden. Dit kan in de toekomst eventueel geautomatiseerd worden.

Deze zorgen hebben tot het volgende aanbevolen bijwerkproces geleid. Merk op dat het gedetailleerde proces voor bepaalde updates aanvullende stappen nodig kan hebben, maar dit kernproces zou de komende tijd ongewijzigd moeten blijven:

1. `make buildworld`

   Dit compileert eerst de nieuwe compiler en enkele aanverwante gereedschappen, daarna wordt de nieuwe compiler gebruikt om de rest van de nieuwe wereld te compileren. Het resultaat komt in /usr/obj te staan.

2. `make buildkernel`

   In tegenstelling tot de oude aanpak, die config(8) en make(1) gebruikt, gebruikt dit de *nieuwe* compiler die in /usr/obj verblijft. Dit beschermt u tegen mismatches tussen de compiler en de kernel.

3. `make installkernel`

   Plaatst de nieuwe kernel en kernelmodules op de schijf, waardoor het mogelijk wordt om met de nieuw bijgewerkte kernel op te starten.

4. Start opnieuw op in enkele-gebruikersmodus.

   De enkele-gebruikersmodus minimaliseert problemen met het bijwerken van software die al draait. Het minimaliseert ook problemen die opduiken door een oude wereld op een nieuwe kernel te draaien.

5. `mergemaster -p`

   Dit voert wat initiële updates aan instellingenbestanden uit ter voorbereiding op de nieuwe wereld. Het kan bijvoorbeeld nieuwe gebruikersgroepen aan het systeem, of nieuwe gebruikersnamen aan de wachtwoorddatabase toevoegen. Dit is vaak nodig wanneer er nieuwe groepen of speciale accounts voor systeemgebruikers zijn toegevoegd sinds de laatste keer bijwerken, zodat de stap `installworld` zonder problemen de nieuw geïnstalleerde namen van systeemgebruikers of systeemgroepen kan gebruiken.

6. `make installworld`

Kopieert de wereld van /usr/obj. U heeft nu een nieuwe kernel en een nieuwe wereld op schijf staan.

7. `mergemaster`

   Nu kunt u de overgebleven instellingenbestanden bijwerken, aangezien u een nieuwe wereld op schijf heeft staan.

8. Start opnieuw op.

   Een volledige nieuwe start van de machine is nodig om de nieuwe kernel en de nieuwe wereld met nieuwe instellingenbestanden te laden.

Merk op dat als u van de ene uitgave van dezelfde tak van FreeBSD bijwerkt naar een recentere uitgave van dezelfde tak, i.e. van 7.0 naar 7.1, dat deze procedure dan niet absoluut nodig is, aangezien het onwaarschijnlijk is dat u serieuze problemen krijgt met de compiler, kernel, gebruikersland en instellingenbestanden. De oudere aanpak met `make world` gevolgd door het bouwen en installeren van een nieuwe kernel kan voor kleine updates goed genoeg zijn.

Maar mensen die deze procedure niet volgen tijdens het bijwerken tussen grote uitgaven kunnen wat problemen verwachten.

Het is ook goed om op te merken dat veel upgrades (i.e. 4.*X* naar 5.0) wat specifieke aanvullende stappen nodig hebben (bijvoorbeeld het hernoemen of verwijderen van specifieke bestanden voorafgaand aan installworld). Lees het bestand /usr/src/UPDATING zorgvuldig, met name het einde, waar het huidig aangeraden bijwerkproces expliciet wordt beschreven.

Deze procedure is in de loop der tijd veranderd aangezien de ontwikkelaars zagen dat het onmogelijk was om bepaalde mismatch-problemen volledig te voorkomen. Hopelijk blijft de huidige procedure voor een lange tijd stabiel.

Samengevat is de huidige aanbevolen manier om FreeBSD vanaf broncode bij te werken:

```
# cd /usr/src
# make buildworld
# make buildkernel
# make installkernel
# shutdown -r now
```



Er zijn een aantal zeldzame gevallen waarin `mergemaster -p` nog een keer moet draaien voor de stap met `buildworld`. Deze staan beschreven in UPDATING. In het algemeen kan deze stap echter zonder risico worden overgeslagen als er niet tussen een of meer hoofdversies wordt bijgewerkt.

Nadat `installkernel` succesvol is afgerond, dient er in single-user modus opgestart te worden (met `boot -s` vanaf de loaderprompt). Draai dan:

```
# mount -u /
# mount -a -t ufs
```

```
# adjkerntz -i
# mergemaster -p
# cd /usr/src
# make installworld
# mergemaster
# reboot
```



Lees verdere uitleg

De hierboven beschreven volgorde is alleen een korte samenvatting. Ook de volgende secties lezen geeft een beter beeld van elke stap, met name als er een op maat gemaakte kernelinstelling wordt gebruikt.

22.7.2. /usr/src/UPDATING lezen

Lees voor verder te gaan /usr/src/UPDATING (of het gelijknamige bestand waar de kopie van de broncode ook staat). Dit bestand kan belangrijke informatie bevatten over mogelijke problemen of specificeert de volgorde waarin bepaalde commando's gestart moeten worden. Als UPDATING tegenstrijdig is met wat hier wordt beschreven, heeft UPDATING voorrang.



UPDATING lezen is geen acceptabele vervanging voor het abonneren op de correcte mailinglijst zoals eerder beschreven. De twee vullen elkaar aan en zijn niet exclusief.

22.7.3. /etc/make.conf controleren

Controleer /usr/shared/examples/etc/make.conf en /etc/make.conf. Het eerste bestand bevat standaard definities, waarvan de meeste uitgecommentarieerd zijn. Om hiervan gebruik te maken als het systeem opnieuw opgebouwd wordt vanuit de broncode, moeten ze toegevoegd worden aan /etc/make.conf. Bedenk dat alles wat toegevoegd wordt aan /etc/make.conf ook gebruikt wordt bij elk `make` commando. Het is dus verstandig om daar redelijke waardes in te vullen voor een systeem.

Een typische gebruiker wil waarschijnlijk de regel `NO_PROFILE` uit /usr/shared/examples/etc/make.conf kopiëren naar /etc/make.conf en het commentaar verwijderen.

Bekijk de andere definities, zoals `NOPORTDOCS` en bepaal of deze relevant zijn.

22.7.4. /etc bijwerken

De map /etc bevat een groot deel van de systeeminstellingen en scripts die gestart worden tijdens de systeemstart. Sommige van deze scripts verschillen van versie tot versie in FreeBSD.

Sommige van de instellingenbestanden worden dagelijks gebruikt voor het draaien van een systeem. In het bijzonder /etc/group.

Er zijn gevallen geweest waarbij het installatiegedeelte van `make installworld` een aantal gebruikersnamen of groepen verwachtte. Als er een upgrade wordt uitgevoerd is het waarschijnlijk dat deze gebruikers of groepen niet bestaan. Dit levert problemen op bij upgraden. In sommige gevallen controleert `make buildworld` of deze gebruikers of groepen bestaan.

Een voorbeeld hiervan is het toevoegen van de gebruiker `smmsp`. Gebruikers hadden een falend installatieproces toen mtree(8) probeerde om /var/spool/clientmqueue te creëren.

mergemaster(8) kan in voorbereidende modus gedraaid worden als de optie `-p` wordt meegegeven. Dan worden alleen de bestanden vergeleken die essentieel zijn voor het succes van `buildworld` of `installworld`:



In "paranoide beheerdersmodus" kan er gecontroleerd worden welke bestanden op een systeem eigendom zijn van de groep die wordt hernoemd of verwijderd:

```
# find / -group GID -print
```

Dit commando toont alle bestanden die eigendom zijn van de groep *GID* (een groepsnaam of een numeriek groeps-ID).

22.7.5. Systeem naar single-user modus brengen

Het kan zijn dat een systeem in single-user modus gecompileerd moet worden. Buiten het duidelijke voordeel dat de operatie iets sneller verloopt, is het voordeel dat bij een herinstallatie van een systeem een aantal belangrijke systeembestanden waaronder binaire systeembestanden, bibliotheken, include bestanden, enzovoort, worden aangepast, iets wat op een actief systeem vragen om problemen is (zeker als er actieve gebruikers op een systeem aanwezig zijn).

Een andere methode is het systeem compileren in multi-user modus en daarna naar single-user modus gaan voor de installatie. Bij deze methode moeten de volgende stappen gevolgd worden. Het overschakelen naar single-user modus kan uitgesteld worden tot en met `installkernel` of `installworld`.

Een supergebruiker kan als volgt een draaiend systeem naar single-user modus overgeschakelen:

```
# shutdown now
```

Als alternatief kan tijdens het opstarten de optie `single user` worden gekozen. Het systeem start dan in single-user modus. Op de shell prompt moet dan worden ingegeven:

```
# fsck -p
# mount -u /
# mount -a -t ufs
# swapon -a
```

Hierdoor worden de bestandssystemen gecontroleerd, / met lees en schrijf rechten opnieuw gemount, worden alle andere UFS bestandssystemen die in /etc/fstab staan gemount en wordt swap ingeschakeld.



Als de CMOS-klok ingesteld is naar de lokale tijd en niet naar GMT (dit is waar als het resultaat van date(1) niet de correcte tijd en zone weergeeft), dan is het

misschien handig om het volgende commando te starten:

```
# adjkerntz -i
```

Dit zorgt ervoor dat de lokale tijdzoneinstellingen correct ingesteld worden. Zonder deze instelling kunnen er later problemen ontstaan.

22.7.6. /usr/obj verwijderen

Als delen van een systeem opnieuw gebouwd worden, worden ze standaard geplaatst in mappen onder /usr/obj. Deze mappen schaduwen de mappen onder /usr/src.

Het proces `make buildworld` kan versneld worden en problemen met afhankelijkheden kunnen voorkomen worden als deze map wordt verwijderd.

Sommige bestanden onder /usr/obj hebben mogelijk de optie "niet aanpassen" ingesteld (zie chflags(1)) die eerst verwijderd moet worden:

```
# cd /usr/obj
# chflags -R noschg *
# rm -rf *
```

22.7.7. Broncode van het basissysteem hercompileren

Uitvoer bewaren

Het is een goed idee om de uitvoer van make(1) te bewaren in een ander bestand. Als er iets misgaat is er een kopie van de foutmelding aanwezig. Hoewel dit misschien niet helpt in de diagnose van wat er fout is gegaan, kan het anderen helpen als het probleem wordt aangegeven in een FreeBSD mailinglijst.

De makkelijkste manier om dit te doen is door het commando script(1) te gebruiken, met een parameter die de naam specificeert waar de uitvoer naartoe moet. Dit moet direct gedaan worden vóór het herbouwen van de wereld, zodat het proces klaar is moet `exit` worden ingegeven:

```
# script /var/tmp/mw.out
Script started, output file is /var/tmp/mw.out
# make TARGET
… compile, compile, compile …
# exit
Script done, …
```

Bewaar de uitvoer in deze stap *niet* in /tmp. Deze map wordt mogelijk opgeschoond tijdens de volgende herstart. Een betere plaats om dit bestand te bewaren is de map /var/tmp (zoals in het vorige voorbeeld) of in de thuismap van `root`.

Basissysteem compileren

Ga naar de map /usr/src, tenzij de broncode ergens anders staat, in welk geval naar die map gegaan moet worden:

```
# cd /usr/src
```

Om de wereld opnieuw te bouwen moet het commando make(1) gebruikt worden. Dit commando leest zijn instructies uit het bestand Makefile, dat beschrijft hoe de programma's die samen FreeBSD vormen moeten worden gebouwd, in welke volgorde ze gebouwd moeten worden, enzovoort.

Het algemene formaat van de commandoregel die gebruikt moet worden is als volgt:

```
# make -x -DVARIABELE doel
```

In dit voorbeeld is de optie `-x` een optie die wordt meegegeven aan make(1). In de hulppagina voor make(1) staat een voorbeeld van de opties die meegegeven kunnen worden.

`-D_VARIABELE_` geeft een variabele door aan Makefile. Het gedrag van Makefile wordt beïnvloed door deze variabele. Dit zijn dezelfde variabelen die ingesteld worden in /etc/make.conf. Deze optie biedt een alternatief om deze opties in te stellen.

```
# make -DNO_PROFILE doel
```

Het bovenstaande commando is een andere manier om aan te geven dat geprofileerde bibliotheken niet gebouwd moeten worden en correspondeert met de onderstaande regel in /etc/make.conf:

```
NO_PROFILE=    true #    Avoid compiling profiled libraries
```

doel geeft make(1) aan wat er gedaan moet worden. Elke Makefile definieert een aantal van verschillende doelen en het gekozen doel bepaalt wat er gebeurt.

Sommige doelen staan vermeld in het bestand Makefile, maar zijn niet geschikt om direct te starten. Integendeel, deze worden gebruikt door het bouwproces om de benodigde stappen onder te verdelen.

In veel gevallen hoeven er geen parameters te worden meegegeven aan make(1) en dus ziet de commando regel er als volgt uit:

```
# make doel
```

Waar *doel* een van de vele bouw opties is. De eerste target moet echter altijd `buildworld` zijn.

Zoals de namen impliceren bouwt `buildworld` een compleet nieuwe boom onder /usr/obj en

`installworld`, een andere target, installeert deze boom op de huidige machine.

Het hebben van verschillende opties is handig om twee redenen. Als eerste biedt het de mogelijkheid om de bouw veilig te doen met de wetenschap dat geen enkel draaiend onderdeel van een systeem geraakt wordt. De bouw is "zelf ondersteunend". Hierdoor kan veilig in multi-user modus `buildworld` gedraaid worden. Het wordt echter nog steeds aangeraden om `installworld` in single-user modus te starten.

Ten tweede geeft het de mogelijkheid om NFS-mounts te gebruiken om meerdere machines in het netwerk bij te werken. Als er drie machines zijn, `A`, `B` en `C`, die bijgewerkt moeten worden, dan kunnen `make buildworld` en `make installworld` gedraaid worden op `A` waarna `B` en `C` een NFS-mount kunnen opzetten naar /usr/src en /usr/obj op machine `A` waarna `make installworld` gedraaid kan worden op `B` en `C` om de resultaten de installeren.

Alhoewel het doel `world` nog wel bestaat wordt het gebruik ervan sterk *afgeraden*.

Voer het volgende commando uit:

```
# make buildworld
```

Het is mogelijk om de optie `-j` mee te geven aan `make`, wat resulteert in meerdere processen die tegelijkertijd draaien. Dit heeft het meeste effect op machines met meerdere processoren. Echter, omdat het compilatieproces meer IO-gericht is dan processorgericht, kan het ook nuttig zijn op systemen met één processor.

Start als volgt op een systeem met één processor:

```
# make -j4 buildworld
```

make(1) draait dan maximaal 4 processen tegelijkertijd. In het algemeen blijkt uit de mailinglijsten dat dit de beste resultaten geeft.

Als er meerdere processoren in een systeem zitten en gebruik gemaakt wordt van een SMP kernel, probeer dan waardes tussen de 6 en 10 en bekijk hoe het systeem reageert.

Doorlooptijd

Veel factoren bepalen de doorlooptijd van het bouwen van een boom, maar redelijk recente machines doen er maar 1 tot 2 uur over om de FreeBSD-STABLE boom te bouwen. zonder extra trucjes. Een FreeBSD-CURRENT boom kan wat langer duren.

22.7.8. Nieuwe kernel compileren en installeren

Om volledig gebruik te maken van het nieuwe systeem moet de kernel opnieuw gecompileerd worden. Dit is bijna altijd nodig omdat sommige geheugenstructuren mogelijkerwijs veranderd zijn en programma's als ps(1) en top(1) niet werken totdat de kernel en de broncode dezelfde versie hebben.

De simpelste en makkelijkste manier om dit te doen is om een kernel te maken die gebaseerd is op GENERIC. Ondanks dat GENERIC mogelijk niet alle benodigde apparaten heeft voor een systeem, hoort het alles te bevatten dat nodig is om een systeem te starten in single-user modus. Dit is een goede test op de correcte werking van een nieuw systeem. Na het opstarten van GENERIC en een systeemcontrole kan erna een nieuwe kernel gebouwd worden gebaseerd op een aangepast kernelinstellingenbestand.

Op FreeBSD is het belangrijk om de wereld opnieuw te bouwen voordat een nieuwe kernel gebouwd wordt.



Als een aangepaste kernel gemaakt moet worden en er reeds een instellingenbestand aanwezig is, gebruik dan `KERNCONF=MYKERNEL` als volgt:

```
# cd /usr/src
# make buildkernel KERNCONF=MYKERNEL
# make installkernel KERNCONF=MYKERNEL
```

Let op dat als `kern.securelevel` een waarde hoger dan 1 heeft *of* `noschg` of gelijksoortige opties geplaatst zijn op het binaire kernelbestand, is het misschien nodig om terug te gaan naar single-user modus om `installkernel` uit te voeren. In andere gevallen moet het mogelijk zijn om deze commando's zonder problemen uit te voeren in multi-user modus. Zie init(8) voor meer informatie over `kern.securelevel` en chflags(1) voor informatie over diverse bestandsopties.

22.7.9. Opnieuw opstarten in single-user modus

Start met de instructies in Systeem naar single-user modus brengen in single-user modus op om te testen of de nieuwe kernel werkt.

22.7.10. Nieuwe binaire systeembestanden installeren

Na het draaien van `make buildworld` kan nu `installworld` gebruikt worden om de nieuwe binaire systeembestanden te installeren.

Voer de volgende commando's uit:

```
# cd /usr/src
# make installworld
```



Als er variabelen gespecificeerd zijn op de commandoregel van `make buildworld` moeten dezelfde variabelen gebruikt worden op de commandoregel van `make installworld`. Dit is niet per se waar voor opties zoals `-j`, die nooit gebruikt mogen worden met `installworld`.

Als bijvoorbeeld het volgende commando is uitgevoerd:

```
# make -DNO_PROFILE buildworld
```

Dan moet het resultaat geïnstalleerd worden met:

```
# make -DNO_PROFILE installworld
```

Anders wordt geprobeerd geprofileerde bibliotheken te installeren die niet gebouwd zijn tijdens de fase `make buildworld`.

22.7.11. Bestanden bijwerken die niet bijgewerkt zijn door `make installworld`

Het herbouwen van de wereld werkt bepaalde mappen niet bij (in het bijzonder /etc, /var en /usr) met nieuwe of gewijzigde instellingenbestanden.

De simpelste manier om deze bestanden bij te werken is door mergemaster(8) te gebruiken, maar het is ook mogelijk dit handmatig te doen. Welke manier er ook gekozen wordt, zorg er altijd voor dat een back-up van /etc beschikbaar is voor het geval er iets misgaat.

`mergemaster`

Het hulpprogramma mergemaster(8) is een Bourne script dat helpt bij het bepalen van de verschillen tussen de instellingenbestanden in /etc en de instellingenbestanden in de broncodeboom /usr/src/etc. Deze methode wordt aangeraden om instellingenbestanden van een systeem bijgewerkt te houden met de bestanden die in de broncodeboom staan.

Het programma wordt gestart met `mergemaster` op de commandoregel en geeft dan resultaten weer. `mergemaster` bouwt dan een tijdelijke root omgeving vanaf / en vult deze met diverse instellingenbestanden voor een systeem. Deze bestanden worden vergeleken met de bestanden die geïnstalleerd zijn op een systeem. Op dit punt worden de bestanden getoond die verschillen in het diff(1)-formaat, met een `+` voor toegevoegde of gewijzigde regels en een `-` voor regels die verwijderd of vervangen zijn. In de hulppagina voor diff(1) staat meer informatie over de syntaxis van diff(1) en hoe bestandsverschillen getoond worden.

mergemaster(8) toont dan elk bestand dat verschilt en op dit moment is er de mogelijkheid om of het nieuwe bestand te verwijderen (ofwel het tijdelijke bestand), het tijdelijke bestand te installeren zonder enige wijzigingen, het verwerken van het oude bestand in het nieuwe bestand of de resultaten van diff(1) nogmaals te tonen.

Als gekozen wordt om het tijdelijke bestand te verwijderen, geeft dit mergemaster(8) aan dat het huidige bestand niet gewijzigd dient te worden en de nieuwe versie verwijderd kan worden. Deze optie wordt niet aangeraden, behalve als er geen reden is om het huidige bestand aan te passen. Op ieder moment kunnen hulpteksten getoond worden door `?` in te geven op de prompt van mergemaster(8). Als een bestand wordt overgeslagen, dan wordt het weer getoond als alle overige bestanden verwerkt zijn.

Bij de keuze om het ongewijzigde tijdelijke bestand te installeren wordt het huidige bestand vervangen door het nieuwe. Voor de meeste ongewijzigde bestanden is dit de beste optie.

Als ervoor gekozen wordt om de wijzigingen te verwerken wordt er een tekstverwerker gestart die de inhoud van beide bestanden toont. De verschillen kunnen verwerkt worden terwijl beide bestanden naast elkaar op het scherm staan. Hier kunnen delen gekozen worden die gezamenlijk een nieuw bestand opleveren. Als de bestanden zij aan zij vergeleken worden, wordt met de toets `l` de inhoud links geselecteerd en met de toets `r` de inhoud rechts geselecteerd. Het eindresultaat bestaat uit delen van beide bestanden die erna geinstalleerd kunnen worden. Deze optie wordt voornamelijk gebruikt voor bestanden die gewijzigd zijn door de beheerder.

Als ervoor gekozen wordt om de diff(1) resultaten nog een keer te tonen, worden dezelfde verschillen getoond zoals mergemaster(8) deed voordat een optie gevraagd werd.

Zodra mergemaster(8) klaar is met de systeembestanden worden er andere opties getoond. mergemaster(8) kan vragen of het wachtwoordbestand opnieuw gebouwd moet worden. Als laatste wordt een optie getoond om alle overgebleven tijdelijke bestanden te verwijderen.

Handmatig bijwerken

Bij handmatig bijwerken kunnen de bestanden van /usr/src/etc niet zomaar naar /etc gekopieerd worden om een werkend systeem te krijgen. Sommige van deze bestanden moeten eerst "geïnstalleerd" worden. Dit omdat de map /usr/src/etc*geen* kopie is van /etc. Daarnaast staan er in /etc bestanden die niet in /usr/src/etc staan.

Als mergemaster(8) gebruikt wordt (zoals aangeraden), kan doorgegaan worden met het volgende onderdeel.

De simpelste manier om met de hand bij te werken, is de bestanden in een nieuwe map installeren en daarna naar verschillen tussen de bestanden te zoeken.



Back-up maken van /etc

Ondanks dat, in theorie, niets in deze map automatisch wordt aangepast, is het altijd beter om daar zeker van te zijn. Dus kopieer de bestaande /etc naar een veilige locatie. Zoals bijvoorbeeld met het volgende commando:

```
# cp -Rp /etc /etc.old
```

`-R` maakt een recursieve kopie, `-p` bewaart tijden, eigenaarschap, enzovoorts op bestanden.

Er moet een dummyset van mappen gemaakt worden om de nieuwe /etc en andere bestanden in te installeren. /var/tmp/root is een redelijke keuze en er zijn hier een aantal benodigde submappen aanwezig:

```
# mkdir /var/tmp/root
# cd /usr/src/etc
# make DESTDIR=/var/tmp/root distrib-dirs distribution
```

Dit maakt de benodigde mappenstructuur en installeert de bestanden. Een groot deel van de

submappen die gemaakt zijn in /var/tmp/root zijn leeg en moeten verwijderd worden. De simpelste manier om dit te doen is:

```
# cd /var/tmp/root
# find -d . -type d | xargs rmdir 2>/dev/null
```

Dit verwijderd alle lege mappen. De standaardfout wordt omgeleid naar /dev/null om waarschuwingen te voorkomen over mappen die niet leeg zijn.

/var/tmp/root bevat nu alle bestanden die geplaatst zouden moeten worden op de juiste locaties in /. Er moet nu in de bestanden gekeken worden om te bepalen of deze verschillen met de huidige betanden.

Let op dat sommige van de bestanden die geïnstalleerd zijn in /var/tmp/root beginnen met een ".". Op het moment van schrijven hebben alleen shell opstartscripts in /var/tmp/root en /var/tmp/root/root dit, maar er kunnen ook andere zijn. Zorg ervoor dat `ls -a` gebruikt wordt om deze bestanden te zien.

De simpelste manier om twee bestanden te vergelijken is diff(1) gebruiken:

```
# diff /etc/shells /var/tmp/root/etc/shells
```

Dit toont de verschillen tussen de huidige /etc/shells en de nieuwe /var/tmp/root/etc/shells. Gebruik dit om te bepalen of de wijzigingen gemigreerd moeten worden of dat het oude bestand gekopieërd moet worden.

Voeg aan de naam van de nieuwe rootmap (/var/tmp/root) een tijdsindicatie toe zodat makkelijk verschillen tussen versies bepaald kunnen worden

Als de wereld regelmatig wordt herbouwd moeten bestanden in /etc ook regelmatig bijgewerkt moeten worden, wat een vervelend werkje kan zijn.

Dit proces kan versneld worden door een kopie te bewaren van de bestanden die gemigreerd zijn naar /etc. De volgende procedure geeft een idee over hoe dit gedaan kan worden.



1. Maak de wereld zoals normaal. Als /etc en de andere mappen bijgewerkt moeten worden, geef dan de doelmap een naam gebaseerd op de huidige datum. Op 14 februari 1998 wordt dat als volgt gedaan:

   ```
   # mkdir /var/tmp/root-19980214
   # cd /usr/src/etc
   # make DESTDIR=/var/tmp/root-19980214 \
       distrib-dirs distribution
   ```

2. Migreer de wijzigingen van deze map zoals hierboven beschreven.

Verwijder de map /var/tmp/root-19980214*niet* na afronden.

3. Als de laatste versie van de broncode gedownload en opnieuw gemaakt is, volg stap 1. Dit geeft een nieuwe map die wellicht /var/tmp/root-19980221 heet (als er een week zit tussen het bijwerken).
4. De verschillen die gemaakt zijn in de tussenliggende week kunnen nu getoond worden door met diff(1) een recursieve diff te maken tussen de twee mappen:

   ```
   # cd /var/tmp
   # diff -r root-19980214 root-19980221
   ```

   Vaak is dit een kleinere set aan verschillen dan tussen /var/tmp/root-19980221/etc en /etc. Omdat de set verschillen kleiner is, is het makkelijker om deze te migreren naar de map /etc.
5. De oudste van de twee /var/tmp/root-*-mappen kan nu verwijderd worden:

   ```
   # rm -rf /var/tmp/root-19980214
   ```
6. Herhaal dit proces elke keer als er wijzigingen gemigreerd moeten worden naar /etc.

Met date(1) kan het maken van de mappen geautomatiseerd worden:

```
# mkdir /var/tmp/root-`date "+%Y%m%d"`
```

22.7.12. Herstarten

Dit was het. Na een controle of alles op de juiste plaats staat kan het systeem herstart worden. Dan kan met een simpele shutdown(8):

```
# shutdown -r now
```

22.7.13. Klaar

Het FreeBSD systeem is nu succesvol bijgewerkt. Gefeliciteerd!

Als er dingen misgingen is het makkelijk om een deel van het systeem opnieuw te bouwen. Als bijvoorbeeld per ongeluk /etc/magic verwijderd is als onderdeel van de upgrade of door het samenvoegen van /etc, dan werkt file(1) niet meer. Dat kan als volgt opgelost worden:

```
# cd /usr/src/usr.bin/file
# make all install
```

22.7.14. Vragen

Moet de wereld opnieuw gemaakt worden voor elke wijziging?

Op deze vraag bestaat geen eenvoudig antwoord, omdat dit afhangt van de aard van de wijziging. Als bijvoorbeeld net CVSup is gedraaid en de onderstaande bestanden zijn bijgewerkt, dan is het waarschijnlijk niet de moeite waard om de volledige wereld te herbouwen:

```
src/games/cribbage/instr.c
src/games/sail/pl_main.c
src/release/sysinstall/config.c
src/release/sysinstall/media.c
src/shared/mk/bsd.port.mk
```

Dan is het handiger om naar de juiste submappen te gaan, daar `make all install` uit te voeren en dat is het zo'n beetje. Maar als er iets wezenlijks is veranderd, bijvoorbeeld src/lib/libc/stdlib, dan dient ofwel de wereld herbouwd te worden of tenminste die delen die statisch gelinkt zijn (en ook al het andere dat statisch gelinkt is en onderdeel is van een systeem).

Uiteindelijk beslist een beheerder zelf. Misschien vindt die het prettig iedere twee weken de wereld te herbouwen terwijl de wijzigingen in die twee weken binnenkomen. Een andere beheerder herbouwt alleen die onderdelen die veranderd zijn en vertrouwt erop dat hij alle afhankelijkheden in de gaten heeft.

Natuurlijk hangt het ook af van de keuze hoe vaak het wenselijk is bij te werken en of FreeBSD-STABLE of FreeBSD-CURRENT wordt bijgehouden.

Het compileren gaat fout met veel meldingen van signal 11signal 11 (of andere signalnummers). Wat is er aan de hand?

Dit wijst meestal op hardwareproblemen. Het (her)bouwen van de wereld is een prima manier om een stresstest op hardware uit te voeren en hierdoor komen vaak geheugenproblemen bovendrijven. Die resulteren vaak in een compiler die op mysterieuze wijze overlijdt na het ontvangen van vreemde signalen.

Dit probleem is nog duidelijker als na het herstarten van de make het proces opnieuw stopt op een ander punt.

Hier biedt niets anders uitkomst dan componenten in een systeem wisselen om uit te zoeken welk component er faalt.

Kan /usr/obj verwijderd worden na afloop?

Het korte antwoord is ja.

/usr/obj bevat alle objectbestanden die tijdens het compileren zijn gemaakt. Normaliter is een van de eerste stappen in het `make buildworld` proces deze map verwijderen en een verse start maken. In dit geval heeft het behouden van /usr/obj na het afronden weinig zin en geeft het ook nogal wat extra vrije schijfruimte (ongeveer 2 GB).

Als er veel kennis aanwezig is bij een beheerder, dan kan `make buildworld` aangegeven worden deze stap over te slaan. Hierdoor draaien volgende builds veel sneller, omdat veel broncode niet opnieuw gecompileerd hoeft te worden. De andere kant van de medaille is dat er subtiele afhankelijkheidsproblemen kunnen ontstaan, waardoor een build op bijzondere wijze kan falen. Hierdoor onstaat regelmatig ruis op FreeBSD mailinglijsten als er iemand klaagt dat zijn build faalt, terwijl hij zich niet realiseert dat dit komt doordat hij zijn updateproces niet volgens het boekje heeft uitgevoerd.

Kunnen onderbroken builds gecontinueerd worden?

Dit hangt af van hoever een systeem was voordat een probleem gevonden werd.

Normaal gesproken (en dit is geen vaste regel) maakt het proces `make buildworld` nieuwe kopieën van essentiele hulpprogramma's (zoals gcc(1) en make(1)) en de systeembibliotheken. Deze hulpprogramma's en bibliotheken worden daarna geïnstalleerd. De nieuwe hulpprogramma's en bibliotheken worden daarna gebruikt om zichzelf opnieuw op te bouwen en wederom te installeren. Het complete systeem (nu met gewone programma's zoals ls(1) en grep(1)) wordt daarna opnieuw gebouwd met de nieuwe systeembestanden.

Als een systeem in de laatste fase zit (wat uit de uitvoer blijkt) kan dit redelijk veilig gedaan worden:

```
··· fix the problem ···
# cd /usr/src
# make -DNO_CLEAN all
```

Dit maakt het werk van de vorige `make buildworld` niet ongedaan.

Als het onderstaande bericht in de uitvoer van `make buildworld` staat, dan is het redelijk veilig om het te doen:

```
--------------------------------------------------------------
Building everything..
--------------------------------------------------------------
```

Als dat bericht er niet is, of er is onzekerheid over, dan is het altijd beter om de build opnieuw te starten vanaf het begin.

Kan kan de wereld bouwen versneld worden?

- Draai in single-user modus;
- Zet de mappen /usr/src en /usr/obj op aparte bestandssystemen die op aparte schijven staan.

Hang deze schijven als mogelijk aan aparte schijfcontrollers;

- Nog beter, verspreid de bestandssystemen over meerdere schijven via het apparaat ccd(4) (concatenated disk driver);
- Zet profiling uit (voeg "NO_PROFILE=true" toe aan /etc/make.conf). Het is zeer waarschijnlijk niet nodig;
- Geef de optie `-jn` mee aan make(1) om meerdere processen parallel te laten lopen. Dit helpt in de meeste gevallen, onafhankelijk of er gewerkt wordt op een systeem met één of meerdere processoren;
- Het bestandssysteem dat /usr/src bevat, kan (opnieuw) gemount worden met de optie `noatime`. Dit voorkomt dat het bestandssysteem de toegangsmomenten registreert. Deze informatie is waarschijnlijk toch niet nodig.

```
# mount -u -o noatime /usr/src
```



In dit voorbeeld wordt aangenomen dat /usr/src op zijn eigen bestandssysteem staat. Als dit niet het geval is (bijvoorbeeld als het onderdeel is van /usr), dan moet het mountpunt voor dat bestandssysteem gebruikt moeten worden en niet /usr/src;

- Het bestandssysteem dat /usr/obj gevat kan (opnieuw) worden gemount met de optie `async`. Dit zorgt ervoor dat schrijfacties naar een schijf asynchroon plaatsvinden. In andere woorden: de schrijfactie wordt direct uitgevoerd en de gegevens worden later naar de schijf geschreven. Dit stelt het systeem in staat om data geclusterd weg te schrijven, wat een grote prestatieverbetering kan opleveren.



Houd er rekening mee dat deze optie het bestandssysteem kwetsbaarder maakt. Met deze optie is er een vergrote kans dat, indien er een stroomstoring optreed, het bestandssysteem in een niet meer te herstellen staat komt als de machine herstart.

Als op dit bestandssysteem alleen /usr/obj staat, is dit geen probleem. Als er andere belangrijke gegevens op hetzelfde bestandssysteem staan, zorg er dan voor dat er verse back-ups zijn voordat deze optie aangezet wordt.

```
# mount -u -o async /usr/obj
```



Zorg ervoor, zoals al eerder is aangegeven, dat als /usr/obj niet op een eigen bestandssysteem staat, het juiste mountpunt wordt gebruikt.

Wat te doen als er iets mis gaat?

Zorg ervoor dat het systeem geen rommel meer bevat van eerdere builds. Het volgende helpt daarbij:

```
# chflags -R noschg /usr/obj/usr
# rm -rf /usr/obj/usr
# cd /usr/src
# make cleandir
# make cleandir
```

Inderdaad, `make cleandir` moet twee keer gedraaid worden.

Herstart daarna het complete proces vanaf `make buildworld`.

Als er nog steeds problemen zijn, stuur dan de foutmelding en de uitvoer van `uname -a` naar de FreeBSD algemene vragen mailinglijst. Wees bereid aanvullende vragen over het systeem te beantwoorden!

22.8. Het verwijderen van overbodige bestanden, directories en bibliotheken

Als onderdeel van de FreeBSD ontwikkel levenscyclus kan het van tijd tot tijd gebeuren dat bestanden en de inhoud ervan overbodig worden. Dit kan komen doordat de functionaliteit ergens anders geïmplementeerd is, het versienummer van de bibliotheek veranderd is of hij is totaal van het systeem verdwenen. Dit is inclusief oude bestanden, bibliotheken en directories welke verwijderd moeten worden bij het updaten van het systeem. Het voordeel voor de gebruiker is dat het systeem niet vervuild wordt met oude bestanden die onnodig ruimte innemen op het opslag (en back-up) systeem. Ook is het zo dat als de oude bibliotheek een beveiligings of stabiliteits probleem had, er moet worden geupdate naar de nieuwere bibliotheek om het systeem veilig te houden en te voorkomen dat er crashes komen door de oude implementatie van de bibliotheek. De bestanden, directories en bibliotheken welke als overbodig worden gezien zijn beschreven in /usr/src/ObsoleteFiles.inc. De volgende instructies zullen helpen om deze verouderde bestanden te verwijderen tijdens het systeem upgrade proces.

Er wordt aangenomen dat de stappen gevolgd worden zoals uitgelegd in De universele wijze om een systeem bij te werken. Na het `make installworld` commando en het daarop volgende `mergemaster` commando succesvol uitgevoerd zijn kan er op de volgende manier gecontroleerd worden voor verouderde bestanden en bibliotheken:

```
# cd /usr/src
# make check-old
```

Als er verouderde bestanden gevonden worden kunnen deze verwijderd worden door het volgende commando:

```
# make delete-old
```



Zie het /usr/src/Makefile bestand voor meer interessante targets.

Er wordt een prompt getoond voordat elk verouderd bestand wordt verwijderd. Deze prompt kan worden overgeslagen en het systeem deze bestanden automatisch laten verwijderen door gebruik te maken van de `BATCH_DELETE_OLD_FILES` make variabele als volgt:

```
# make -DBATCH_DELETE_OLD_FILES delete-old
```

Dit kan ook worden gedaan door deze commando's door `yes` te pipen als volgt:

```
# yes|make delete-old
```



Waarschuwing

Het verwijderen van verouderde bestanden zal applicaties stuk maken die nog gebruik maken van de overbodige bestanden. Dit is zeker waar voor oude bibliotheken. In de meeste gevallen moeten de programma's, ports of bibliotheken opnieuw gecompileerd worden voordat `make delete-old-libs` wordt uitgevoerd.

Gereedschappen om gedeelde bibliotheek afhankelijkheden te controleren zijn beschikbaar in de Ports Collectie in sysutils/libchk of sysutils/bsdadminscripts.

Overbodige gedeelde bibliotheken kunnen conflicteren met nieuwere bibliotheken welke berichten zoals deze kunnen veroorzaken:

```
/usr/bin/ld: warning: libz.so.4, needed by /usr/local/lib/libtiff.so, may conflict
with libz.so.5
/usr/bin/ld: warning: librpcsvc.so.4, needed by /usr/local/lib/libXext.so, may
conflict with librpcsvc.so.5
```

Om deze problemen op te lossen moet bepaald worden welke port deze bibliotheek heeft geïnstalleerd:

```
# pkg_info -W /usr/local/lib/libtiff.so
/usr/local/lib/libtiff.so was installed by package tiff-3.9.4
# pkg_info -W /usr/local/lib/libXext.so
/usr/local/lib/libXext.so was installed by package libXext-1.1.1,1
```

Deïnstalleer, herbouw en herinstalleer de port. De ports-mgmt/portmaster en ports-mgmt/portupgrade gereedschappen kunnen gebruikt worden om deze processen te automatiseren. Nadat zeker is dat alle ports opnieuw gebouwd zijn, en de oude bibliotheken niet meer gebruikt worden, kunnen deze verwijderd worden met het volgende commando:

```
# make delete-old-libs
```

22.9. Meerdere machines bijwerken

Als er meerdere machines zijn die dezelfde broncode bijhouden, lijkt het downloaden van alle broncode en alles overal opnieuw bouwen zonde van de bronnen: harde schijfruimte, netwerk bandbreedte, en processorbelasting. Dit klopt en de oplossing is om alles op één machine te doen terwijl de overige machines het uitgevoerde werk benaderen via NFS. Nu wordt een methode beschreven waarmee dit gedaan kan worden.

22.9.1. Benodigdheden

Als eerste moet er een groep van machines gekozen worden die dezelfde set aan binaire bestanden zal draaien, hier een *bouwgroep*. Elke machine kan een eigen afwijkende kernel hebben maar moet dezelfde binaire gebruikersbestanden draaien. Uit die groep moet een machine gekozen worden die de *bouwmachine* wordt. Dit wordt de machine waar de wereld en kernel op gebouwd worden. In het meest ideale geval is dit een snelle machine die genoeg processorkracht vrij heeft om `make buildworld` en `make buildkernel` te draaien. Er moet ook een machine gekozen worden die de *testmachine* wordt waarop alle bijgewerkte software wordt test voordat die in productie wordt genomen. Dit *moet* een machine zijn die voor langere tijd down mag zijn. Dit kan de bouwmachine zijn maar dat hoeft niet per se.

Alle machines in deze bouwgroep moeten ingesteld worden om /usr/obj en /usr/src vanaf dezelfde machine te mounten op hetzelfde punt. In het meest ideale geval zijn dit twee verschillende schijven op de bouwmachine, maar ze kunnen ook door middel van NFS op die machine gemount zijn. Als er meerdere bouwgroepen zijn, dan moet /usr/src op één bouwmachine staan en door middel van NFS gemount worden op de overige machines.

Zorg er als laatste voor dat /etc/make.conf en /etc/src.conf op alle machines in de bouwgroep het eens zijn met de bouwmachine. Dat betekent dat de bouwmachine alle delen van het basissysteem moet bouwen die elke machine in de bouwgroep installeert. Ook heeft elke bouwmachine zijn kernelnaam ingesteld met `KERNCONF` in /etc/make.conf en de bouwmachine moet ze allemaal hebben in `KERNCONF`, zijn eigen kernel eerst. De bouwmachine moet de instellingenbestanden voor elke machine in /usr/src/sys/arch/conf hebben als deze machine de kernels voor de overige machines gaat bouwen.

22.9.2. Basissysteem

Nu kan één systeem alles bouwen. Bouw de kernel en wereld zoals beschreven in Basissysteem compileren op de bouwmachine, maar installeer niets. Zodra de bouw klaar is, moet op de testmachine de kernel geïnstalleerd en getest worden. Als deze machine /usr/src en /usr/obj mount via NFS, moet na een herstart in single-user modus het netwerk ingeschakeld worden zodat de mounts opnieuw gemaakt kunnen worden. De makkelijkste manier om dit te doen is om te starten in multi-user modus en daar `shutdown now` starten om in single-user modus te komen. Eenmaal daar aangekomen kunnen de nieuwe kernel en de wereld geïnstalleerd worden en kan daarna normaal `mergemaster` gestart worden. Zodra dit klaar is, kan de machine opnieuw gestart worden om naar multi-user modus terug te keren.

Nadat zeker is dat alles op de testmachine correct werkt, kan dezelfde procedure gebruikt worden om de nieuwe software op elke machine te installeren in de bouwgroep.

22.9.3. Ports

Dezelfde ideeën kunnen gebruikt worden voor de ports. De eerste kritieke stap is om /usr/ports te mounten op alle machines in de bouwgroep. Daarna kan /etc/make.conf correct ingesteld worden om de distfiles te delen. De variabele `DISTDIR` moet wijzen naar een gedeelde map waarin geschreven kan worden door de gebruiker waar `root` naar wijst in de NFS mounts. Op elke machine moet `WRKDIRPREFIX` naar een lokale bouwmap wijzen. Als er pakketten gebouwd en gedistribueerd worden moet `PACKAGES` naar een map wijzen gelijkvormig aan de instelling voor `DISTDIR`.

Hoofdstuk 23. DTrace

23.1. Overzicht

DTrace, ook bekend als Dynamic Tracing, was ontwikkeld door Sun™ als een gereedschap om prestatie-bottlenecks in productie- en preproductiesystemen op te sporen. Het is in geen enkel opzicht een debug-gereedschap, maar een gereedschap voor real-time analyse om prestatie- en andere zaken op te sporen.

DTrace is een opmerkelijk profileringsgereedschap, met een indrukwekkende verzameling mogelijkheden om systeemzaken te diagnosticeren. Het kan ook worden gebruikt om vooraf geschreven scripts te draaien om zo voordeel te halen uit de mogelijkheden. Gebruikers kunnen zelfs hun eigen middelen schrijven door gebruik te maken van de DTrace D Language, wat ze in staat stelt om hun profilering aan te passen aan hun specifieke behoeften.

Na het lezen van dit hoofdstuk weet u:

- Wat DTrace is en welke mogelijkheden het biedt.
- De verschillen tussen de DTrace-implementatie van Solaris™ en degene die door FreeBSD wordt aangeboden.
- Hoe DTrace op FreeBSD aan te zetten en te gebruiken.

Voordat u dit hoofdstuk leest, dient u:

- De beginselen van UNIX® en FreeBSD te begrijpen (UNIX® beginselen).
- Bekend te zijn met de beginselen van kernelconfiguratie en -compilatie (De FreeBSD-kernel instellen).
- Wat bekendheid te hebben met beveiliging en hoe het zich verhoudt tot FreeBSD (Beveiliging).
- Te begrijpen hoe de broncode van FreeBSD te verkrijgen en te herbouwen (FreeBSD updaten en upgraden).



Deze mogelijkheid wordt als experimenteel beschouwd. Van sommige opties kan er functionaliteit ontbreken, andere delen kunnen in het geheel niet werken. In de loop der tijd zal deze mogelijkheid als productierijp worden beschouwd en zal deze documentatie worden aangepast om die situatie te representeren.

23.2. Implementatieverschillen

Hoewel DTrace in FreeBSD erg lijkt op degene die in Solaris™ zit, zijn er verschillen die uitgelegd moeten worden voordat er verder wordt gegaan. Het primaire verschil dat gebruikers zullen zien is dat DTrace specifiek moet worden aangezet op FreeBSD. Er zijn kernelopties en modulen die aangezet moeten worden om DTrace juist te laten werken. Deze zullen later worden uitgelegd.

Er is een kerneloptie `DDB_CTF` die gebruikt wordt om ondersteuning voor het laden van CTF-gegevens van kernelmodulen en de kernel zelf. CTF is het Compact C Type Format van Solaris™

welke een beperkte vorm van debuginformatie bevat die vergelijkbaar is met DWARF en de befaamde stabs. Deze CTF-gegevens worden door de bouwmiddelen `ctfconvert` en `ctfmerge` aan de binairen toegevoegd. Het hulpmiddel `ctfconvert` parseert DWARFELF-debug-secties die door de compiler zijn aangemaakt en `ctfmerge` voegt CTFELF-secties van objecten samen in hun executables of gedeelde bibliotheken. Meer informatie over hoe dit voor de bouw van de kernel en FreeBSD aan te zetten komt eraan.

Sommige aanbieders voor FreeBSD verschillen van die voor Solaris™. De meest opmerkelijke is de aanbieder `dtmalloc`, welke het volgen van `malloc()` op soort in de FreeBSD-kernel toestaat.

Alleen `root` mag DTrace op FreeBSD gebruiken. Dit heeft te maken met beveiligingsverschillen, Solaris™ heeft enkele beveiligingscontroles op laag niveau die nog niet bestaan in FreeBSD. Hierom is /dev/dtrace/dtrace strikt beperkt tot `root`.

Tenslotte valt de DTrace-software onder de CDDL-licentie van Sun™. De `Common Development and Distribution License` wordt bij FreeBSD geleverd, zie /usr/src/cddl/contrib/opensolaris/OPENSOLARIS.LICENSE of bekijk het online op http://www.opensolaris.org/os/licensing.

Deze licentie houdt in dat een FreeBSD-kernel met de DTrace-opties nog steeds onder de BSD-licentie valt; de CDDL komt echter op de proppen wanneer de modulen in binaire vorm worden verspreid, of wanneer de binairen zijn geladen.

23.3. Ondersteuning voor DTrace aanzetten

Voeg de volgende regels toe aan het kernelinstellingenbestand om ondersteuning voor DTrace aan te zetten:

```
options         KDTRACE_HOOKS
options         DDB_CTF
```



Gebruikers van de AMD64-architectuur zullen de volgende regel aan hun kernelinstellingenbestand willen toevoegen:

```
options         KDTRACE_FRAME
```

Deze optie biedt ondersteuning voor de mogelijkheid FBT. DTrace zal zonder deze optie werken; er zal echter beperkte ondersteuning zijn voor het volgen van functiegrenzen.

Alle broncode moet herbouwd en geherinstalleerd worden met de CTF-opties. Om deze taak te volbrengen, wordt de FreeBSD-broncode herbouwd met:

```
# cd /usr/src
```

```
# make WITH_CTF=1 kernel
```

Het systeem moet opnieuw gestart worden.

Nadat opnieuw is opgestart en de nieuwe kernel in het geheugen is geladen, dient ondersteuning voor de Korn-shell te worden toegevoegd. Dit is nodig omdat de verschillende hulpmiddelen van DTraceToolkit in `ksh` zijn geschreven. Installeer shells/ksh93. Het is ook mogelijk om deze hulpmiddelen in shells/pdksh of shells/mksh te draaien.

Als laatste dient de huidige DTraceToolkit verkregen te worden. Indien u FreeBSD 10 draait, vindt u de DTraceToolkit in /usr/shared/dtrace. In andere gevallen kunt u de DTraceToolkit installeren via de port sysutils/dtrace-toolkit.

23.4. DTrace gebruiken

Voordat er gebruik wordt gemaakt van de functionaliteit van DTrace, moet het DTrace-apparaat bestaan. Geef het volgende commando om het apparaat te laten:

```
# kldload dtraceall
```

Ondersteuning van DTrace zou nu beschikbaar moeten zijn. De beheerder kan het volgende commando uitvoeren om alle sondes te bekijken:

```
# dtrace -l | more
```

Alle uitvoer wordt aan het hulpmiddel `more` doorgegeven omdat het snel de schermbuffer zal laten overstromen. DTrace kan nu als werkend worden beschouwd. Het is nu tijd om de gereedschapskist te bekijken.

De gereedschapskist is een verzameling van kant-en-klare scripts die met DTrace gedraaid kunnen worden om informatie over het systeem te verzamelen. Er zijn scripts om open bestanden, geheugen, CPU-gebruik, en nog veel meer te controleren. Pak de scripts uit met het volgende commando:

```
# gunzip -c DTraceToolkit* | tar xvf -
```

Ga naar die map met `cd` en zet de uitvoerpermissies voor alle bestanden waarvan de naam uit kleine letters bestaat, op `755`.

De inhoud van al deze scripts moet veranderd worden. Degenen die naar /usr/bin/ksh verwijzen dienen naar /usr/local/bin/ksh te verwijzen, de anderen die /usr/bin/sh gebruiken dienen gewijzigd te worden om /bin/sh te gebruiken, en tenslotte dienen degenen die /usr/bin/perl gebruiken veranderd te worden om /usr/local/bin/perl te gebruiken.



Op dit moment is het voorzichtig om de lezer eraan te herinneren dat de

> ondersteuning voor DTrace in FreeBSD *niet compleet* en *experimenteel* is. Veel van deze scripts zullen niet werken omdat ze of te Solaris™-specifiek zijn of omdat ze sondes gebruiken die momenteel niet ondersteund worden.

Op het moment van schrijven worden slechts twee scripts van de DTrace Toolkit volledig ondersteund in FreeBSD: de scripts hotkernel en procsystime. Dit zijn de twee die we in de volgende gedeelten van deze sectie zullen bekijken.

De hotkernel is ontworpen om te identificeren welke functie de meeste kerneltijd gebruikt. Als het normaal gedraaid wordt, zal het uitvoer die op de volgende lijkt produceren:

```
# cd /usr/shared/dtrace/toolkit
# ./hotkernel
Sampling... Hit Ctrl-C to end.
```

De systeembeheerder moet de toetsencombinatie `Ctrl` + `C` gebruiken om het proces te stoppen. Nadat het gestopt is, zal het script een lijst van kernelfuncties en timinginformatie weergeven, waarbij de uitvoer in volgorde van toenemende tijd is gesorteerd:

```
kernel`_thread_lock_flags                             2   0.0%
0xc1097063                                            2   0.0%
kernel`sched_userret                                  2   0.0%
kernel`kern_select                                    2   0.0%
kernel`generic_copyin                                 3   0.0%
kernel`_mtx_assert                                    3   0.0%
kernel`vm_fault                                       3   0.0%
kernel`sopoll_generic                                 3   0.0%
kernel`fixup_filename                                 4   0.0%
kernel`_isitmyx                                       4   0.0%
kernel`find_instance                                  4   0.0%
kernel`_mtx_unlock_flags                              5   0.0%
kernel`syscall                                        5   0.0%
kernel`DELAY                                          5   0.0%
0xc108a253                                            6   0.0%
kernel`witness_lock                                   7   0.0%
kernel`read_aux_data_no_wait                          7   0.0%
kernel`Xint0x80_syscall                               7   0.0%
kernel`witness_checkorder                             7   0.0%
kernel`sse2_pagezero                                  8   0.0%
kernel`strncmp                                        9   0.0%
kernel`spinlock_exit                                 10   0.0%
kernel`_mtx_lock_flags                               11   0.0%
kernel`witness_unlock                                15   0.0%
kernel`sched_idletd                                 137   0.3%
0xc10981a5                                        42139  99.3%
```

Het script werkt ook met kernelmodules. Draai het script met de vlag `-m` om deze mogelijkheid te gebruiken:

```
# ./hotkernel -m
Sampling... Hit Ctrl-C to end.
^C
MODULE                                                  COUNT   PCNT
0xc107882e                                                  1   0.0%
0xc10e6aa4                                                  1   0.0%
0xc1076983                                                  1   0.0%
0xc109708a                                                  1   0.0%
0xc1075a5d                                                  1   0.0%
0xc1077325                                                  1   0.0%
0xc108a245                                                  1   0.0%
0xc107730d                                                  1   0.0%
0xc1097063                                                  2   0.0%
0xc108a253                                                 73   0.0%
kernel                                                    874   0.4%
0xc10981a5                                             213781  99.6%
```

Het script procsystime vangt en beeldt het tijdsgebruik van systeemaanroepen af voor een gegeven PID of procesnaam. In het volgende voorbeeld wordt er een nieuwe instantie van /bin/csh gedraaid. Het procsystime werd uitgevoerd en bleef wachten terwijl er enkele commando's op de andere instantie van csh werden getypt. Dit zijn de resultaten van deze test:

```
# ./procsystime -n csh
Tracing... Hit Ctrl-C to end...
^C

Elapsed Times for processes csh,

         SYSCALL          TIME (ns)
          getpid               6131
       sigreturn               8121
           close              19127
           fcntl              19959
             dup              26955
         setpgid              28070
            stat              31899
       setitimer              40938
           wait4              62717
       sigaction              67372
     sigprocmask             119091
    gettimeofday             183710
           write             263242
          execve             492547
           ioctl             770073
           vfork            3258923
      sigsuspend            6985124
            read         3988049784
```

Zoals te zien is, lijkt de systeemaanroep `read()` de meeste tijd in nanoseconden te gebruiken en gebruikte de systeemaanroep `getpid()` de minste hoeveelheid tijd.

23.5. De taal D

De DTrace-gereedschapskist bevat vele scripts in de speciale taal van DTrace. Deze taal wordt "de taal D" genoemd door de documentatie van Sun™, en lijkt sterk op C++. Een diepgaande discussie over de taal valt buiten het bereik van dit document. Het wordt uitgebreid behandeld op http://wikis.oracle.com/display/DTrace/Documentation.

Unresolved directive in book.adoc - include::usb-device-mode/_index.adoc[leveloffset=+1]

Deel IV: Netwerkcommunicatie

FreeBSD is een van de meest gebruikte besturingssystemen voor hoogwaardige, snelle netwerkservers. De hoofdstukken in dit deel behandelen:

- Seriële communicatie
- PPP en PPP over Ethernet
- E-mail
- Netwerkdiensten
- Firewalls
- Andere geavanceerde netwerkonderwerpen

Deze hoofdstukken kunnen worden gelezen als naslagwerk. Ze kunnen in willekeurige volgorde worden gelezen en niet alle hoofdstukken hoeven gelezen te worden om FreeBSD te kunnen gaan gebruiken in een netwerkomgeving.

Hoofdstuk 24. Seriële communicatie

24.1. Overzicht

UNIX® heeft altijd ondersteuning geboden voor seriële communicatie. Het is een feit dat de allereerste UNIX®-machines afhankelijk waren van seriële kabels voor gebruikersinvoer en -uitvoer. De dingen zijn flink veranderd sinds de tijd dat de gemiddelde "terminal" uit een 10-tekens-per-seconde seriële printer en een toetsenbord bestond. Dit hoofdstuk beschrijft enkele manieren waarop FreeBSD gebruik maakt van seriële communicatie.

Na het lezen van dit hoofdstuk weet de lezer:

- Hoe terminals met een FreeBSD-systeem te verbinden;
- Hoe een modem te gebruiken om naar computers op afstand te bellen;
- Hoe het mogelijk te maken voor gebruikers op afstand om met een modem op een systeem aan te melden;
- Hoe een systeem van een seriële console op te starten.

Veronderstelde voorkennis:

- Weten hoe een nieuwe kernel ingesteld en geïnstalleerd wordt (De FreeBSD-kernel instellen);
- Begrijpen hoe rechten en processen in UNIX® werken (UNIX® beginselen);
- De beschikking hebben over het technische handboek voor de hardware (modem of meerpoortige kaart) die gebruikt wordt met FreeBSD.

24.2. Inleiding



Vanaf FreeBSD 8.0 zijn de seriële poorten hernoemd van /dev/cuadN naar /dev/cuauN en van /dev/ttydN naar /dev/ttyuN. FreeBSD 7.X gebruikers moeten de documentatie aanpassen naar deze wijzigingen.

24.2.1. Terminologie

bps

Bits per seconde: de snelheid waarmee gegevens verstuurd worden

DTE

Data Terminal Equipment (apparatuur voor gegevensterminal): bijvoorbeeld een computer

DCE

Data Communications Equipment (apparatuur voor gegevenscommunicatie): een modem

RS-232

EIA standaard voor hardwarematige seriële communicatie

Deze sectie gebruikt niet de term "baud" als er over snelheden van gegevenscommunicatie gesproken wordt. Baud verwijst naar het aantal elektrische toestandsovergangen dat binnen een tijdsperiode gemaakt mag worden, "bps" (bits per seconde) is de *correcte* term om te gebruiken (de oude mopperkonten schijnen zich er niet erg druk over te maken).

24.2.2. Kabels en poorten

Om een modem of terminal met een FreeBSD-systeem te verbinden, dienen een seriële poort op een computer en een kabel om verbinding te maken met een serieel apparaat aanwezig te zijn. Indien kennis over hardware en de benodigde kabel reeds aanwezig is, kan deze sectie veilig worden overgeslagen.

Kabels

Er zijn verschillende soorten seriële kabels. De twee meest voorkomende types in deze context zijn nulmodem-kabels en standaard ("rechte") RS-232-kabels. De documentatie van de hardware beschrijft het type kabel dat nodig is.

Nulmodem-kabels

Een nulmodem-kabel geeft sommige signalen, zoals "Aardesignaal" recht door, maar kruist andere signalen. Bijvoorbeeld, de "Verzonden Gegevens"-pin aan de ene kant gaat naar de "Ontvangen Gegevens"-pin aan de andere kant.

Een nulmodem-kabel voor het gebruik met terminals kan ook zelf worden gemaakt (bijvoorbeeld voor kwaliteitsdoeleinden). Deze tabel toont de RS-232C signalen en de pinnummers op een DB-25-aansluiting. De standaard vereist ook een *Aardebescherming* rechte lijn van pin 1 naar pin 1, maar deze wordt vaak weggelaten. Sommige terminals werken goed met slechts pin 2, 3 en 7, terwijl andere instellingen eisen die afwijken van die in de onderstaande voorbeelden.

Tabel 3. DB-25 naar DB-25 nulmodem-kabel

Signaal	Pin #		Pin #	Signaal
SG	7	verbonden met	7	SG
TD	2	verbonden met	3	RD
RD	3	verbonden met	2	TD
RTS	4	verbonden met	5	CTS
CTS	5	verbonden met	4	RTS
DTR	20	verbonden met	6	DSR
DTR	20	verbonden met	8	DCD

In de onderstaande tabellen volgen twee schema's die momenteel meer gebruikelijk zijn:

Tabel 4. DB-9 naar DB-9 nulmodem-kabel

Signaal	Pin #		Pin #	Signaal
RD	2	verbonden met	3	TD

Signaal	Pin #		Pin #	Signaal
TD	3	verbonden met	2	RD
DTR	4	verbonden met	6	DSR
DTR	4	verbonden met	1	DCD
SG	5	verbonden met	5	SG
DSR	6	verbonden met	4	DTR
DCD	1	verbonden met	4	DTR
RTS	7	verbonden met	8	CTS
CTS	8	verbonden met	7	RTS

Tabel 5. DB-9 naar DB-25 nulmodem-kabel

Signaal	Pin #		Pin #	Signaal
RD	2	verbonden met	2	TD
TD	3	verbonden met	3	RD
DTR	4	verbonden met	6	DSR
DTR	4	verbonden met	8	DCD
SG	5	verbonden met	7	SG
DSR	6	verbonden met	20	DTR
DCD	1	verbonden met	20	DTR
RTS	7	verbonden met	5	CTS
CTS	8	verbonden met	4	RTS



Als een pin aan het ene eind verbonden is met een pinnenpaar aan het andere eind, is dit meestal geïmplementeerd met een korte draad tussen het pinnenpaar in de stekker en een lange draad naar de andere, enkele pin.

Bovenstaande ontwerpen lijken het populairst. In een andere variatie (uitgelegd in het boek *RS-232 Made Easy*) worden de volgende verbindingen gemaakt: SG met SG, TD met RD, RTS en CTS met DCD, DTR met DSR en vice-versa.

Standaard RS-232C-kabels

Een standaard seriële kabel laat alle RS-232C-signalen recht door. Dit betekent dat de "Verzonden Gegevens"-pin aan de ene kant naar de "Verzonden Gegevens"-pin aan de andere kant gaat. Dit type kabel wordt gebruikt om een modem met een FreeBSD-systeem te verbinden en is ook geschikt voor sommige terminals.

Poorten

Seriële poorten zijn apparaten die gebruikt worden om gegevens te versturen tussen een FreeBSD gastcomputer en een terminal. Deze sectie beschrijft de bestaande soorten poorten en hoe deze

aangesproken worden in FreeBSD.

Soorten poorten

Er bestaan verschillende soorten seriële poorten. Controleer of een kabel past op de poorten van een terminal en een FreeBSD-systeem alvorens deze te kopen of te maken.

De meeste terminals hebben DB-25-poorten. PC's, inclusief PC's die FreeBSD draaien, hebben DB-25- of DB-9-poorten. Indien een meerpoortige seriële kaart voor een PC beschikbaar is, kan het zijn dat er RJ-12- of RJ-45-poorten aanwezig zijn.

In documentatie die bij hardware zit, staan specificaties over het soort poort dat gebruikt wordt. Vaak volstaat ook een visuele inspectie van een poort.

Poortnamen

In FreeBSD wordt elke seriële poort benaderd door een ingang in de map /dev. Er zijn twee verschillende soorten ingangen:

- Inbelpoorten heten /dev/ttyuN waarbij *N* het poortnummer is, beginnend met nul. In het algemeen kunnen inbelpoorten voor terminals gebruikt worden. Inbelpoorten stellen de eis dat een seriële kabel ervoor zorgt dat het data carrier detect (DCD) signaal correct werkt.
- Uitbelpoorten heten /dev/cuauN. In het algemeen worden uitbelpoorten niet voor terminals maar voor modems gebruikt. Gebruik een uitbelpoort als een seriële kabel of terminal het carrier detect-signaal niet ondersteunt.

Als er een terminal met de eerste seriële poort (COM1 in MS-DOS®) verbonden is, wordt /dev/ttyu0 gebruikt om naar de terminal te verwijzen. Als een terminal op de tweede seriële poort is aangesloten (ook bekend als COM2), dient /dev/ttyu1 gebruikt te worden, enzovoort.

24.2.3. Kernelinstellingen

FreeBSD ondersteunt standaard vier seriële poorten. In de wereld van MS-DOS® staan ze bekend als COM1, COM2, COM3 en COM4. FreeBSD ondersteunt momenteel "domme" meerpoortige seriële interfacekaarten, zoals de BocaBoard 1008 en 2016, alsook intelligentere meerpoortige kaarten van fabrikanten als Digiboard en Stallion Technologies. De kernel kijkt echter alleen naar de standaard COM-poorten.

Bekijk de boodschappen tijdens het opstarten van de kernel om te zien of de kernel seriële poorten herkent of gebruik het commando `/sbin/dmesg` om de opstartboodschappen van de kernel te herhalen. Kijk in het bijzonder naar boodschappen die met de tekens `uart` beginnen als u FreeBSD 8.0 of nieuwer gebruikt, of `sio` voor FreeBSD 7.4 of ouder.



Gebruik het volgende commando om alleen de boodschappen die het woord `sio` bevatten te zien:

```
# /sbin/dmesg | grep 'uart'
# /sbin/dmesg | grep 'sio'
```

Voor bijvoorbeeld een FreeBSD 7.*X* systeem met vier seriële poorten zijn dit de opstartboodschappen van de kernel die specifiek zijn voor de seriële poorten:

```
sio0 at 0x3f8-0x3ff irq 4 on isa
sio0: type 16550A
sio1 at 0x2f8-0x2ff irq 3 on isa
sio1: type 16550A
sio2 at 0x3e8-0x3ef irq 5 on isa
sio2: type 16550A
sio3 at 0x2e8-0x2ef irq 9 on isa
sio3: type 16550A
```

Als een kernel niet alle seriële poorten herkent, dan dient waarschijnlijk de kernel aangepast te worden in het bestand /boot/device.hints. Het is ook mogelijk regels uit te schakelen of volledig te verwijderen voor apparaten die niet aanwezig zijn.

Zie de hulppagina sio(4) voor meer informatie over het instellen van seriële poorten en meerpoortige kaarten. Bij gebruik van een instellingenbestand dat eerder voor een andere versie van FreeBSD werd gebruikt is voorzichtigheid geboden omdat de apparaatvlaggen en de syntaxis tussen de versies veranderd zijn.



`port IO_COM1` is een substitutie voor `port 0x3f8`, `IO_COM2` is `0x2f8`, `IO_COM3` is `0x3e8` en `IO_COM4` is `0x2e8`, welke redelijk algemene poortadressen zijn voor hun overeenkomstige seriële poorten. Interrupts 4, 3, 5 en 9 zijn redelijk algemene interruptlijnen. Reguliere seriële poorten kunnen *geen* interrupts delen op ISA-bus-PC's (meerpoortige kaarten hebben elektronica die alle 16550A's op een kaart in staat stellen om één of twee interruptlijnen te delen).

24.2.4. Speciale apparaatbestanden

De meeste apparaten in de kernel worden benaderd met "speciale apparaatbestanden" die in de map /dev staan. De apparaten sio worden benaderd met de apparaten /dev/ttyuN (inbellen) en /dev/cuauN (uitbellen). FreeBSD biedt ook initialisatie-apparaten (/dev/ttyuN.init en /dev/cuauN.init) en slotapparaten (/dev/ttyuN.lock en /dev/cuadN.lock). De initialisatie-apparaten worden gebruikt om telkens als een poort wordt geopend de parameters van de communicatiepoorten te initialiseren, zoals `crtscts` voor modems die gebruik maken van `RTS/CTS`-signalering voor gegevensstroombeheer. De slotapparaten worden gebruikt om vlaggen op poorten op slot te zetten om te voorkomen dat gebruikers of programma's bepaalde parameters veranderen. In de hulppagina's termios(4), sio(4) en stty(1) staat informatie over respectievelijk terminalinstellingen, apparaten op slot zetten en initialiseren en terminalopties instellen.

24.2.5. De seriële poort instellen

Het apparaat ttyuN (of cuauN) is het gebruikelijke apparaat dat geopend dient te worden voor de applicaties. Wanneer een proces het apparaat opent, heeft het een standaardverzameling aan terminal I/O-instellingen. Bekijk deze instellingen met het volgende commando:

```
# stty -a -f /dev/ttyu1
```

Als de instellingen van dit apparaat veranderd worden, blijven de instellingen geldig totdat het apparaat gesloten wordt. Als het heropend wordt, gaat het terug naar de standaardverzameling. Om de standaardverzameling te veranderen, dient het apparaat voor de "initiële toestand" geopend te worden en die instellingen veranderd te worden. Om bijvoorbeeld de `CLOCAL`-modus, 8-bits-communicatie en `XON/XOFF`-gegevensstroombeheer voor apparaat ttyu5 standaard aan te zetten:

```
# stty -f /dev/ttyu5.init clocal cs8 ixon ixoff
```

De systeembrede initialisatie van de seriële apparaten wordt beheerd in /etc/rc.d/serial. Dit bestand heeft invloed op de standaardinstellingen van seriële apparaten.

Om te voorkomen dat bepaalde instellingen door een applicatie worden veranderd, dienen wijzigingen aan het "slottoestand"-apparaat te worden aangebracht. Om bijvoorbeeld de snelheid van ttyu5 vast te zetten op 57600 bps:

```
# stty -f /dev/ttyu5.lock 57600
```

Nu blijft een applicatie die ttyu5 en de snelheid van de poort probeert te veranderen zitten op 57600 bps.

Uiteraard dienen de apparaten voor de initiële toestand en de slottoestand alleen voor het account `root` schrijfbaar te zijn.

24.3. Terminals



Vanaf FreeBSD 8.0 zijn de seriële poorten hernoemd van /dev/cuadN naar /dev/cuauN en van /dev/ttydN naar /dev/ttyuN. FreeBSD 7.X gebruikers moeten de documentatie aanpassen naar deze wijzigingen.

Terminals bieden een handige en goedkope manier om een FreeBSD systeem te benaderen als de console van of een netwerk naar een computer niet beschikbaar is. Deze sectie beschrijft hoe terminals met FreeBSD te gebruiken.

24.3.1. Types terminals en ze gebruiken

De originele UNIX®-systemen hadden geen consoles. In plaats daarvan werd er aangemeld en werden programma's via terminals gedraaid die verbonden waren met de seriële poorten van een computer. Het is goed vergelijkbaar met het gebruik van een modem en terminalsoftware om op een systeem op afstand in te bellen en werk te doen wat alleen uit tekst bestaat.

De consoles van hedendaagse PC's kunnen grafische uitvoer van hoge kwaliteit produceren, maar de mogelijkheid om een aanmeldsessie op een seriële poort tot stand te brengen bestaat nog steeds op bijna elk hedendaags UNIX®-achtig systeem. FreeBSD is geen uitzondering. Door gebruik te

maken van een terminal die aangesloten is op een ongebruikte seriële poort, kan er aangemeld worden en kan bijna elk tekstprogramma gedraaid worden dat normaalgesproken op de console of in een `xterm`-venster in het X Window-systeem gedraaid wordt.

Een zakelijke gebruiker kan vele terminals aan een FreeBSD-systeem koppelen en deze op de bureaus van medewerkers neerzetten. Een thuisgebruiker kan een reservecomputer, zoals een oudere IBM PC of een Macintosh®, met de terminal verbinden met een krachtigere computer die FreeBSD draait. Op deze manier kan wat anders een computer voor een enkele gebruiker zou zijn, worden veranderd in een krachtig systeem voor meerdere gebruikers.

Er zijn drie soorten terminals voor FreeBSD:

- Domme terminals;
- PC's die als terminals dienen;
- X-terminals.

De overige subsecties beschrijven elk van deze soorten.

Domme terminals

Domme terminals zijn gespecialiseerde stukken hardware die computers door seriële kabels kunnen verbinden. Ze worden "dom" genoemd omdat ze alleen maar tekst kunnen weergeven, verzenden en ontvangen. Het is niet mogelijk om programma's op deze terminals te draaien. De computer waar ze op zijn aangesloten heeft de benodigde kracht om tekstverwerkers, compilers, e-mail, spellen, enzovoort te draaien.

Er zijn honderden soorten domme terminals gemaakt door vele fabrikanten, inclusief de VT-100 van Digital Equipment Corporation en de WY-75 van Wyse. Bijna elke soort werkt met FreeBSD. Sommige terminals uit de hoogste klasse kunnen zelfs grafisch weergeven, maar slechts bepaalde softwarepakketten kunnen gebruik maken van deze geavanceerde mogelijkheden.

Domme terminals zijn ook populair in werkomgevingen waarin gebruikers geen toegang tot grafische applicaties nodig hebben, zoals die door het X Window systeem worden geleverd.

PC's die als terminal dienen

Indien een domme terminal net genoeg mogelijkheden heeft om tekst weer te geven, te verzenden en te ontvangen, dan kan zeker elke reserve-PC een domme terminal zijn. De enige benodigdheden zijn de juiste kabel en wat *terminal-emulatie* software om op de computer te draaien.

Zo'n opstelling is populair in thuissituaties. Indien bijvoorbeeld persoon A werkt op de console van een FreeBSD-systeem, kan persoon B wat alleen-tekst-werk verrichten op een minder krachtige PC die als terminal met het FreeBSD-systeem verbonden is.

Er zijn minstens twee applicaties beschikbaar in het basissysteem van FreeBSD welke gebruikt kunnen worden om te communiceren door een seriële connectie: cu(1) en tip(1).

Om een connectie op te zetten vanaf een systeem dat FreeBSD draait naar een seriële connectie van een andere machine kan het volgende gedaan worden:

```
# cu -l serial-port-device
```

Hierbij is "serial-port-device" de naam is van de speciale apparaatnode die gebruikt wordt voor de seriële poort op het systeem. Deze bestanden heten /dev/cuauN.

Het "N" gedeelte van de apparaatnaam is het nummer van de seriële poort.



Let op, de apparaatnummers beginnen in FreeBSD bij nul en niet bij één (zoals ze bijvoorbeeld wel doen bij MS-DOS® gebaseerde systemen). Dit betekend dat wat MS-DOS® gebaseerde systemen COM1 noemt bij FreeBSD meestal /dev/cuau0 genoemd wordt.



Sommige mensen preferen andere programma's die beschikbaar zijn via de Ports Collectie. De ports bevatten een aantal programma's die hetzelfde kunnen werken als cu(1) en tip(1), zoals comms/minicom.

X-terminals

X-terminals behoren tot de meest geavanceerde terminalsoort die beschikbaar is. In plaats van dat ze verbinding maken met een seriële poort, maken ze meestal verbinding met een netwerk zoals Ethernet. In plaats van dat ze alleen tekstapplicaties weergeven, kunnen ze elke X-applicatie weergeven.

X-terminals worden slechts voor de compleetheid geïntroduceerd. Dit hoofdstuk behandelt echter *niet* de installatie, het instellen of het gebruik van X-terminals.

24.3.2. Instellen

Deze sectie beschrijft wat in te stellen op een os;-systeem om een aanmeldsessie op een terminal mogelijk te maken. De sectie gaat ervan uit dat er al een kernel is ingesteld met ondersteuning voor een seriële poort waar de terminal op is aangesloten en dat deze verbonden is.

In Het FreeBSD opstartproces staat beschreven dat het proces `init` verantwoordelijk is voor het beheer van alle processen en voor de initialisatie tijdens het opstarten van een systeem. Eén van de taken die door `init` wordt uitgevoerd is het lezen van het bestand /etc/ttys en het starten van een proces `getty` op de beschikbare terminals. Het proces `getty` is verantwoordelijk voor het lezen van een aanmeldnaam en het starten van het programma `login`.

Voer volgende stappen als `root` uit om terminals voor een FreeBSD-systeem in te stellen:

1. Voeg een regel aan /etc/ttys toe voor de ingang in de map /dev voor een seriële poort als deze er nog niet is;
2. Specificeer dat `/usr/libexec/getty` uitgevoerd moet worden op de poort en het juiste type *getty* van het bestand /etc/gettytab;
3. Specificeer het standaard terminaltype;
4. Stel de poort in op "on";

5. Specificeer of de poort "secure" dient te zijn;
6. `init` dient /etc/ttys opnieuw te lezen.

Als optionele stap kan het wenselijk zijn om een eigen type *getty* aan te maken voor stap 2 door een ingang in /etc/gettytab te maken. Dit wordt hier niet beschreven. Meer informatie staat in de hulppagina's gettytab(5) en getty(8).

Een regel aan /etc/ttys toevoegen

Het bestand /etc/ttys bevat alle poorten op een FreeBSD-systeem waar aanmelden is toegestaan. De eerste virtuele console ttyv0 staat bijvoorbeeld in dit bestand vermeld. Met deze vermelding kan er op de console worden aangemeld. Dit bestand bevat ook vermeldingen voor de andere virtuele consoles, seriële poorten en pseudo-tty's. Vermeld voor een vast aangesloten terminal de /dev-regel van de seriële poort zonder het /dev-gedeelte (/dev/ttyv0 wordt bijvoorbeeld ttyv0).

Een standaard FreeBSD installatie bevat een bestand /etc/ttys met ondersteuning voor de eerste vier seriële poorten: ttyu0 tot en met ttyu3. Indien er aan een van deze poorten een terminal wordt gekoppeld is het niet nodig om een regel toe te voegen.

Voorbeeld 17. Terminalregels aan /etc/ttys toevoegen

Stel dat er twee terminals verbonden moeten worden met een systeem: een Wyse-50 en een oude 286 IBM-PC waarop Procomm terminalsoftware draait dat een VT-100 terminal emuleert. De Wyse wordt met de tweede seriële poort verbonden en de 286 met de zesde seriële poort (een poort op een meerpoortige seriële kaart). De overeenkomstige regels in /etc/ttys zien er als volgt uit:

```
ttyu1  "/usr/libexec getty std.38400"  wy50  on  insecure
ttyu5   "/usr/libexec/getty std.19200"  vt100  on  insecure
```

- Het eerste veld specificeert normaalgesproken de naam van het speciale terminalbestand zoals dat in /dev staat.
- Het tweede veld bevat het commando dat voor deze regel uitgevoerd moet worden, meestal is dit getty(8). `getty` initialiseert en opent een lijn, stelt een snelheid in, vraagt om een gebruikersnaam en draait daarna het programma login(1).Het programma `getty` accepteert één (optionele) parameter op de opdrachtregel, het type *getty*. Een type *getty* stelt karakteristieken op een terminallijn in, zoals de bps-snelheid en de pariteit. Het programma `getty` leest deze karakteristieken uit het bestand /etc/gettytab.Het bestand /etc/gettytab bevat een hoop regels voor zowel oude als nieuwe terminallijnen. In bijna alle gevallen werken de regels die met de tekst `std` beginnen voor vast aangesloten terminals. Deze regels negeren pariteit. Er is een `std`-regel voor elke bps-snelheid van 110 tot en met 115200. Uiteraard kunnen eigen regels aan dit bestand worden toegevoegd. De hulppagina gettytab(5) biedt meer informatie.Zorg er tijdens het instellen van het type *getty* in het bestand /etc/ttys voor dat de communicatie-instellingen op de terminal ermee over komen.In bovenstaand voorbeeld gebruikt de Wyse-50 geen pariteit en maakt deze verbinding met 38400 bps. De 286 PC gebruikt geen pariteit en maakt verbinding met

19200 bps.

- Het derde veld bevat het type terminal dat normaalgesproken is verbonden met de tty-lijn. Voor inbelpoorten wordt voor dit veld normaalgesproken `unknown` of `dialup` gebruikt omdat gebruikers bijna elk type terminal of software gebruiken om in te bellen. Voor terminals met een vaste aansluiting verandert het type terminal niet, dus kan in dit veld een echt terminaltype uit het databasebestand termcap(5) worden gebruikt.In bovenstaand voorbeeld gebruikt de Wyse-50 het echte terminaltype, terwijl de 286 PC die Procomm draait zo ingesteld wordt dat deze een VT-100 emuleert.
- Het vierde veld geeft aan of de poort aan moet staan. Indien hier `on` staat, start `init` het programma in het tweede veld, `getty`, op. Indien hier `off` staat wordt `getty` niet uitgevoerd en kan er daarom niet op de poort worden aangemeld.
- Het laatste veld geeft aan of de poort veilig is. Indien deze poort als veilig is aangemerkt betekent dit dat er genoeg vertrouwen is om de gebruiker `root` (of iedere andere account met een gebruikers-id 0) aan te laten melden via deze poort. Onveilige poorten staan aanmelden door `root` niet toe. Meld op onveilige poorten eerst aan een account zonder rechten en gebruik daarna su(1) of een soortgelijk mechanisme om rootrechten te verkrijgen.Het wordt sterk aangeraden om "insecure" zelfs voor terminals achter gesloten deuren te gebruiken. Het is vrij gemakkelijk om aan te melden en `su` te gebruiken indien rootrechten nodig zijn.

`init` forceren om /etc/ttys opnieuw te lezen

Stuur na het maken van de benodigde veranderingen aan het bestand /etc/ttys een SIGHUP-signaal (ophangen) naar het proces `init` om het te dwingen het instellingenbestand opnieuw te lezen:

```
# kill -HUP 1
```



`init` is altijd het eerste proces dat op een systeem gedraaid wordt, daarom heeft het altijd PID 1.

Indien alles juist is ingesteld, alle kabels juist zijn aangesloten en alle terminals aanstaan, draait er op elke terminal een proces `getty` en is er een aanmeldprompt zichtbaar op de terminals.

24.3.3. Problemen met een verbinding oplossen

Zelfs met de grootste aandacht voor details kan er nog steeds iets mis gaan met het instellen van een terminal. Hier is een lijst van symptomen en mogelijke oplossingen.

Er verschijnt geen aanmeldprompt

Controleer of de terminal is aangesloten en aan staat. Indien het een PC is die als terminal fungeert, controleer of de terminalemulatiesoftware op de juiste seriële poort draait.

Controleer of de kabel stevig verbonden is met zowel de terminal als de FreeBSD computer en dat de kabel van het juiste soort is.

Controleer of de terminal en FreeBSD dezelfde bps-snelheid en pariteit gebruiken. Indien de terminal een beeldscherm is, controleer dan of de video- en helderheidsniveaus zijn ingesteld. Indien de terminal een printer is, controleer of er voldoende papier en inkt aanwezig zijn.

Controleer of er een proces `getty` draait dat de terminal bedient. Om bijvoorbeeld een lijst van draaiende `getty`-processen te krijgen:

```
# ps -axww|grep getty
```

Er zou een regel voor de terminal zichtbaar moeten zijn. Het volgende scherm geeft bijvoorbeeld weer dat `getty` op de tweede seriële poort ttyu1 draait en de regel `std.38400` in /etc/gettytab gebruikt:

```
22189  d1  Is+   0:00.03 /usr/libexec/getty std.38400 ttyu1
```

Indien er geen proces `getty` draait, controleer dan of de poort in /etc/ttys aan staat. Draai `kill -HUP 1` nadat het bestand ttys is gewijzigd.

Indien het proces `getty` draait maar de terminal nog steeds geen aanmeldprompt weergeeft of als het een prompt weergeeft maar er niet getypt kan worden, kan het zijn dat de terminal of de kabel hardwarematige handshaking niet ondersteunt. Probeer om de regel in /etc/ttys van `std.38400` in `3wire.38400` te veranderen (draai na het wijzigen van /etc/ttys`kill -HUP 1`). De regel `3wire` is vergelijkbaar met de regel `std`, maar negeert hardwarematige handshaking. Het kan nodig zijn om de baudsnelheid te verlagen of om softwarematig doorvoerbeheer aan te zetten als `3wire` gebruikt wordt, om overspoelde buffers te voorkomen.

Als er rommel in plaats van een aanmeldprompt verschijnt

Controleer of de terminal en FreeBSD dezelfde bps-snelheid en pariteit gebruiken. Controleer de `getty`-processen op het gebruik van het juiste type `getty`. Indien dit niet het geval is, wijzig dan /etc/ttys en draai `kill -HUP 1`.

Tekens verschijnen dubbel en/of het wachtwoord verschijnt tijdens de invoer

Wijzig de terminal (of de terminalemulatiesoftware) van "half duplex" of "local echo" naar "full duplex".

24.4. Inbeldienst



Vanaf FreeBSD 8.0 zijn de seriële poorten hernoemd van /dev/cuadN naar /dev/cuauN en van /dev/ttydN naar /dev/ttyuN. FreeBSD 7.X gebruikers moeten de documentatie aanpassen naar deze wijzigingen.

Het instellen van het FreeBSD-systeem voor inbeldiensten is vrijwel gelijk aan het verbinden van terminals, behalve dat er met modems in plaats van terminals wordt gewerkt.

24.4.1. Externe en interne modems

Externe modems lijken gemakkelijker voor het inbellen, omdat externe modems vaak semi-permanent ingesteld kunnen worden via parameters die in een niet-vluchtig RAM worden opgeslagen en ze hebben gewoonlijk LED's die de toestand van belangrijke RS-232-signalen weergeven. Knipperende LED's maken indruk op bezoekers, maar LED's zijn ook zeer nuttig om te zien of een modem goed functioneert.

Interne modems hebben vaak geen niet-vluchtig RAM en het kan dus voorkomen dat de instelmogelijkheden beperkt zijn tot het instellen van DIP-schakelaars. Als een intern modem al indicatie-LED's voor signalen heeft, zijn ze moeilijk te zien in de behuizing van een systeem.

Modems en kabels

Bij gebruik van een extern modem is uiteraard een juiste kabel nodig. Een standaard RS-232C seriële kabel moet voldoen zolang alle normale signalen zijn aangesloten.

Tabel 6. Signaalnamen

Acroniemen	Namen
RD	Received Data (ontvangen gegevens)
TD	Transmitted Data (verzonden gegevens)
DTR	Data Terminal Ready (gegevensterminal gereed)
DSR	Data Set Ready (gegevensverzameling gereed)
DCD	Data Carrier Detect (RS-232's detector voor signaal lijn-ontvangen)
SG	Signal Ground (signaalaarde)
RTS	Request to Send (verzoek om te zenden)
CTS	Clear to Send (gereed om te zenden)

FreeBSD heeft de signalen RTS en CTS nodig voor doorstroombeheer bij snelheden van meer dan 2400 bps, het signaal CD om te bepalen wanneer een oproep beantwoord of geannuleerd is, en het signaal DTR om een modem opnieuw in te stellen nadat een sessie voltooid is. Op sommige kabels ontbreken sommige benodigde signalen. Dus als zich problemen voordoen, zoals een aanmeldsessie die niet weggaat nadat de verbinding verbroken is, kan dit aan de kabel liggen.

Net als andere UNIX®-achtige besturingssystemen gebruikt FreeBSD hardwaresignalen om te bepalen of een oproep beantwoord of weggedrukt is en om met het modem op te hangen en dit opnieuw in te stellen na een oproep. FreeBSD vermijdt het versturen van commando's naar een modem en het bekijken van de toestand van een modem. Dit kan vreemd lijken als bekend is hoe modems met PC-gebaseerde prikbordsystemen (BBS) verbinden.

24.4.2. Overwegingen voor de seriële interface

FreeBSD ondersteunt EIA RS-232C (CCITT V.24) communicatie-interfaces gebaseerd op NS8250, NS1645, NS16550 en NS16550A. De 8250- en 16450-apparaten hebben buffers van een enkel karakter. Het 16550-apparaat biedt een buffer van 16 karakters, wat betere systeemprestaties

toestaat. Door fouten in platte 16550's is het niet mogelijk de buffer van 16 karakters te gebruiken, dus gebruik indien mogelijk 16550A's. Omdat apparaten met een buffer van een enkel karakter meer werk door het besturingssysteem vereisen dan apparaten met een buffer van 16 karakters, ligt de voorkeur bij seriële interfacekaarten gebaseerd op de 16550A. Indien een systeem veel actieve seriële poorten heeft of zwaar belast wordt, zijn kaarten gebaseerd op de 16550A beter voor communicatie met een lage foutenratio.

24.4.3. Snel overzicht

Net als met terminals zet `init` een `getty`-proces op voor elke seriële poort die voor inbelverbindingen is ingesteld. Indien bijvoorbeeld een modem aan /dev/ttyu0 is gekoppeld, kan het commando `ps ax` het volgende weergeven:

```
4850 ??  I     0:00.09 /usr/libexec/getty V19200 ttyu0
```

Wanneer een gebruiker naar de modemlijn belt en de modems verbinding maken, wordt de CD-lijn (Carrier Detect) door het modem gerapporteerd. De kernel merkt op dat een draaggolf is gesignaleerd en laat `getty` het openen van de poort voltooien. `getty` stuurt een prompt `login:` met de initieel gespecificeerde lijnsnelheid. `getty` bekijkt of er geldige karakters zijn ontvangen en probeert, in een typische opstelling, indien het rommel aantreft (waarschijnlijk omdat de snelheid waarmee het modem verbindt afwijkt van de snelheid van `getty`) de lijnsnelheden aan te passen totdat het redelijke karakters ontvangt.

Nadat een gebruikersnaam is opgegeven voert `getty`/usr/bin/login uit, die het aanmelden voltooit door te vragen naar het wachtwoord van een gebruiker en daarna de shell van een gebruiker op te starten.

24.4.4. Instellingenbestanden

Er zijn drie systeeminstellingenbestanden in de map /etc die waarschijnlijk gewijzigd moeten worden om inbellen op een FreeBSD-systeem toe te staan. Het eerste bestand, /etc/gettytab, bevat informatie om de daemon /usr/libexec/getty in te stellen. Het tweede bestand, /etc/ttys bevat informatie voor /sbin/init dat vertelt op welke tty apparaten een proces `getty` moet draaien. Als laatste kunnen in het script /etc/rc.d/serial commando's geplaatst worden om poorten te initialiseren.

Er bestaan twee stromingen met betrekking tot inbelmodems op UNIX®. De ene houdt ervan om modems en systemen in te stellen zodat de lokale computer-naar-modem RS-232-interface met een vaste snelheid werkt, ongeacht de snelheid waarmee een gebruiker-op-afstand inbelt. Het voordeel van deze instelling is dat een gebruiker-op-afstand altijd meteen een aanmeldprompt van een systeem ziet. Het nadeel is dat een systeem niet weet wat de werkelijke gegevenssnelheid van een gebruiker is en dus passen programma's die met een volledig scherm werken, zoals Emacs, hun methode om het scherm te tekenen niet aan om hun reactie beter te maken voor langzame verbindingen.

De andere stroming stelt de RS-232-interface van een modem zo in dat de snelheid ervan varieert met de verbindingssnelheid van een gebruiker-op-afstand. Zo zorgen V.32bis-verbindingen (14,4 kbps) met een modem ervoor dat een modem de RS-232-interface op 19,2 kbps laat draaien, terwijl

verbindingen op 2400 bps ervoor zorgen dat de RS-232-interface van een modem op 2400 bps draait. Omdat `getty` meldingen over de verbindingssnelheid van een gegeven modem niet begrijpt, geeft `getty` een bericht `login:` op een initiële snelheid en kijkt het naar de karakters die als antwoord terugkomen. Als een gebruiker rommel ziet, wordt ervan uitgegaan dat deze weet dat de `Enter` toets ingedrukt moet worden totdat een herkenbaar prompt zichtbaar is. Indien de gegevenssnelheden niet overeenkomen, ziet `getty` alles wat een gebruiker intypt als "rommel", probeert het op de volgende snelheid over te gaan en het geeft opnieuw het prompt `login:`. Deze procedure kan ad nauseam doorgaan, maar normaal gesproken zijn er slechts een stuk of twee toetsaanslagen nodig voordat een gebruiker een juist prompt ziet. Het is duidelijk dat deze aanmeldprocedure er niet zo mooi uitziet als de methode "vaste-snelheid", maar een gebruiker met een langzame verbinding zou betere interactiviteit moeten beleven met programma's die met een volledig scherm werken.

Deze sectie poogt om neutrale informatie over instellingen te geven, maar is geneigd om de gegevenssnelheid van het modem af te laten hangen van de verbindingssnelheid.

/etc/gettytab

/etc/gettytab is een bestand met informatie over instellingen voor getty(8) in de stijl van termcap(5). In de hulppagina van gettytab(5) staat de volledige informatie over het formaat van het bestand en de lijst met mogelijkheden.

Vaste snelheid instellen

Indien de snelheid van een modem om gegevens te communiceren op een bepaalde waarde wordt vastgezet, is het waarschijnlijk niet nodig om wijzigingen aan te brengen in /etc/gettytab.

Overeenkomstige snelheid instellen

In /etc/gettytab dient een regel ingesteld te worden om `getty` informatie te geven over de snelheden die voor het modem gewenst zijn. Indien een 2400 bps modem aanwezig is, kan waarschijnlijk de bestaande regel `D2400` gebruikt worden.

```
#
# Voor snelle inbelterminals, 2400/1200/300 roterend (er kan met beide kanten begonnen
worden)
#
D2400|d2400|Fast-Dial-2400:\
    :nx=D1200:tc=2400-baud:
3|D1200|Fast-Dial-1200:\
    :nx=D300:tc=1200-baud:
5|D300|Fast-Dial-300:\
    :nx=D2400:tc=300-baud:
```

Indien er een modem voor hogere snelheden aanwezig is, dient er waarschijnlijk een regel aan /etc/gettytab toegevoegd te worden. Hieronder staat een regel die gebruikt kan worden voor een 14,4 kbps modem met een maximale interface-snelheid van 19,2 kbps:

```
#
# Toevoegingen voor een V.32bis modem
#
um|V300|High Speed Modem at 300,8-bit:\
    :nx=V19200:tc=std.300:
un|V1200|High Speed Modem at 1200,8-bit:\
    :nx=V300:tc=std.1200:
uo|V2400|High Speed Modem at 2400,8-bit:\
    :nx=V1200:tc=std.2400:
up|V9600|High Speed Modem at 9600,8-bit:\
    :nx=V2400:tc=std.9600:
uq|V19200|High Speed Modem at 19200,8-bit:\
    :nx=V9600:tc=std.19200:
```

Dit resulteert in 8-bits verbindingen zonder pariteit.

Het bovenstaande voorbeeld begint met een communicatiesnelheid van 19,2 kbps (voor een V.32bis-verbinding), daarna doorloopt het 9600 bps (voor V.32), 2400 bps, 1200 bps, 300 bps en daarna weer 19,2 kbps. Het doorlopen van de communicatiesnelheid is met de mogelijkheid `nx=` ("volgende tabel") geïmplementeerd. Elk van deze regels gebruikt een regel `tc=` ("tabel continuëren") om de rest van de "standaard"-instellingen voor een bepaalde gegevenssnelheid op te pikken.

Indien er een 28,8 kbps modem aanwezig is en/of het gewenst is om voordeel uit de compressie met een 14,4 kbps te halen, is het nodig om hogere communicatiesnelheden dan 19,2 kbps te gebruiken. Hieronder staat een voorbeeld van een regel voor gettytab die begint met 57,6 kbps.

```
#
# Toevoegingen voor een V.32bis of V.34 modem
# beginnend bij 57,6 kbps
#
vm|VH300|Very High Speed Modem at 300,8-bit:\
    :nx=VH57600:tc=std.300:
vn|VH1200|Very High Speed Modem at 1200,8-bit:\
    :nx=VH300:tc=std.1200:
vo|VH2400|Very High Speed Modem at 2400,8-bit:\
    :nx=VH1200:tc=std.2400:
vp|VH9600|Very High Speed Modem at 9600,8-bit:\
    :nx=VH2400:tc=std.9600:
vq|VH57600|Very High Speed Modem at 57600,8-bit:\
    :nx=VH9600:tc=std.57600:
```

Indien een CPU langzaam of een systeem zwaar belast is en er geen seriële poorten gebaseerd op 16550A aanwezig zijn, kunnen er `sio` "silo"-fouten optreden bij 57,6 kbps.

/etc/ttys

Het instellen van het bestand /etc/ttys staat beschreven in Terminalregels aan /etc/ttys toevoegen.

Het instellen van modems is vergelijkbaar maar er moet een ander argument aan `getty` doorgegeven worden en er moet een ander type terminal doorgegeven te worden. Het algemene formaat voor zowel vaste snelheid als overeenkomstige snelheid is:

```
ttyu0   "/usr/libexec/getty xxx"   dialup on
```

Het eerste item op bovenstaande regel is het speciale apparaatbestand. ttyu0 betekent dat /dev/ttyu0 het bestand is dat door `getty` in de gaten wordt gehouden. Het tweede item, `"/usr/libexec/gettyxxx"` (*xxx* wordt vervangen door de initiële mogelijkheden van gettytab) is het proces dat door `init` op het apparaat gedraaid wordt. Het derde item, `dialup`, is het standaard terminaltype. De vierde parameter, `on`, geeft aan `init` aan dat de lijn operationeel is. Er kan een vijfde parameter zijn, `secure`, maar gebruik deze alleen voor terminals die fysiek veilig zijn (zoals de systeemconsole).

Het standaard terminaltype (`dialup` in bovenstaand voorbeeld) mag afhangen van lokale voorkeuren. Het traditionele standaard terminaltype voor inbellijnen is `dialup`, zodat gebruikers hun aanmeldscripts kunnen aanpassen om op te merken wanneer het terminal `dialup` is en automatisch hun terminaltype kunnen aanpassen. Wellicht is het makkelijker om `vt102` als het standaard terminaltype te specificeren, aangezien gebruikers gewoon VT102-emulatie op hun systemen-op-afstand gebruiken.

Nadat /etc/ttys gewijzigd is, kan aan het proces `init` een signaal HUP gestuurd worden om het bestand opnieuw te laten lezen. Gebruik volgende opdracht om het signaal te versturen:

```
# kill -HUP 1
```

Indien een systeem voor de eerste keer wordt geïnstalleerd, is het verstandig te wachten totdat een modem juist ingesteld en verbonden is voordat het signaal aan `init` verstuurd wordt.

Vaste snelheid instellen

Voor het instellen van een vaste snelheid dient de regel in ttys een vaste snelheid door te geven aan `getty`. Voor een modem met een vaste poortsnelheid van 19,2 kbps kan de regel in ttys er als volgt uitzien:

```
ttyu0   "/usr/libexec/getty std.19200"   dialup on
```

Indien een modem op een andere gegevenssnelheid is ingesteld, dient de juiste waarde voor `std.snelheid` in plaats van `std.19200` ingesteld te worden. Gebruik een geldig type dat in /etc/gettytab vermeld staat.

Overeenkomstige snelheid instellen

Voor het instellen van een overeenkomstige snelheid dient de regel in ttys te verwijzen naar regel met de juiste begin-"auto-baud" (sic). Indien bijvoorbeeld de boven voorgestelde regel voor een modem met een overeenkomstige snelheid die begint met 19,2 kbps wordt toegevoegd (de regel in

gettytab die het beginpunt V19200), kan de regel in ttys er als volgt uitzien:

```
ttyu0   "/usr/libexec/getty V19200"   dialup on
```

/etc/rc.d/serial

Hogesnelheidsmodems, zoals V.32, V.32bis, en V.34 modems, moeten gebruik maken van hardwarematig (RTS/CTS) gegevensstroombeheer. Er kunnen stty-opdrachten aan /etc/rc.d/serial toegevoegd worden om de vlag voor hardwarematig gegevensstroombeheer in de kernel van FreeBSD voor modempoorten in te stellen.

Om bijvoorbeeld de termios-vlag crtscts op de apparaten die de in- en uitbelapparaten initialiseren op de eerste seriële poort (COM2) in te stellen, kunnen de volgende regels aan /etc/rc.d/serial worden toegevoegd:

```
# Seriële poort initieel instellen
stty -f /dev/ttyu1.init crtscts
stty -f /dev/cuau1.init crtscts
```

24.4.5. Modeminstellingen

Bij gebruik van een modem waarvan de parameters permanent in niet-vluchtig RAM ingesteld kunnen worden, is er een terminalprogramma (zoals Telix onder MS-DOS® of tip onder FreeBSD) nodig om parameters in te stellen. Maak een verbinding met een modem met dezelfde communicatiesnelheid als de initiële snelheid die door getty gebruikt wordt en stel het niet-vluchtige RAM van een modem in zodat aan deze voorwaarden voldaan wordt:

- CD geldt tijdens verbindingen;
- DTR geldt tijdens gebruik; het loslaten van DTR hangt de verbinding op en stelt het modem opnieuw in;
- gegevensstroombeheer door CTS verzonden;
- gegevensstroombeheer met XON/XOFF uitgezet;
- gegevensstroombeheer door RTS ontvangen;
- Stille modus (geen resultaatcodes);
- Geen opdrachtecho.

Kijk in de documentatie van een modem voor de benodigde opdrachten en/of instellingen van DIP-schakelaars.

Om de bovenstaande parameters bijvoorbeeld op een U.S. Robotics® Sportster® 14.400 extern modem in te stellen, kunnen de volgende opdrachten aan het modem gegeven worden:

```
ATZ
AT&C1;&D2&H1&I0&R2&W
```

In deze fase kunnen ook andere modeminstellingen aangepast worden, zoals of het V.42bis en/of MNP5 compressie wordt gebruiken.

Een U.S. Robotics®Sportster® 14.400 externe modem heeft ook enkele DIP-schakelaars die ingesteld moeten worden. Voor andere modems kunnen deze instellingen wellicht als voorbeeld dienen:

- Schakelaar 1: UP: DTR Normal
- Schakelaar 2: N/A (Verbal Result Codes/Numeric Result Codes)
- Schakelaar 3: UP: Suppress Result Codes
- Schakelaar 4: DOWN: Geen echo, offline opdrachten
- Schakelaar 5: UP: Auto Answer
- Schakelaar 6: UP: Carrier Detect Normal
- Schakelaar 7: UP: Load NVRAM Defaults
- Schakelaar 8: N/A (Smart Mode/Dumb Mode)

Schakel resultaatcodes voor alle inbelmodems uit of onderdruk ze om problemen te voorkomen die kunnen optreden als `getty` abusievelijk een prompt `login:` geeft aan een modem dat in opdrachtmodus staat en het modem de opdracht echoot of een resultaatcode teruggeeft. Deze sequentie kan tot een uitgebreide, onnozele discussie tussen `getty` en het modem leiden.

Vaste snelheid instellen

Stel voor een vaste snelheid een modem zodanig in dat die een constante gegevenssnelheid naar de computer, onafhankelijk van de communicatiesnelheid, behoudt. Op een U.S. Robotics® Sportster® 14.400 extern modem zetten de volgende opdrachten de gegevenssnelheid naar de computer vast op de snelheid die gebruikt werd om de opdrachten te geven:

```
ATZ
AT&B1&W
```

Overeenkomstige snelheid instellen

Stel voor een variabele snelheid een modem zodanig in dat het de gegevenssnelheid van zijn seriële poort aanpast aan de snelheid van een binnenkomende oproep. Op een U.S. Robotics® Sportster® 14.400 extern modem zetten de volgende opdrachten de gegevenssnelheid van het modem, die op fouten gecorrigeerd wordt, vast op de snelheid die gebruikt werd om de opdrachten te geven, maar staan ze toe dat de snelheid van de seriële poort varieert voor verbindingen die niet op fouten gecorrigeerd worden:

```
ATZ
AT&B2&W
```

De modeminstellingen controleren

De meeste modems die op hoge snelheid werken, bieden opdrachten om de huidige werkparameters van een modem in een min of meer voor mensen leesbare vorm te bekijken. Op het U.S. Robotics® Sportster® 14.400 extern modem beeldt de opdracht `ATI5` de instellingen af die in het niet-vluchtige RAM zijn opgeslagen. Gebruik om de werkelijke werkparameters van een modem te zien (zoals beïnvloed door de stand van de DIP-schakelaars van een modem) de opdrachten `ATZ` gevolgd door `ATI4`.

Kijk in de handleiding van een modem als er met een ander merk modem gewerkt wordt voor het controleren van de parameters voor het instellen van dat modem.

24.4.6. Problemen oplossen

Hier volgen wat stappen die gevolgd kunnen worden om een inbelmodem op een systeem te controleren.

Een FreeBSD-systeem controleren

Verbind een modem met een FreeBSD-systeem, start het systeem op en kijk, indien het modem lampjes bevat die de toestand aangeven, of de DTR-indicator oplicht als het prompt `login:` op de systeemconsole verschijnt. Als het oplicht zou dit betekenen dat FreeBSD een `getty`-proces heeft gestart op de juiste communicatiepoort en wacht op het modem om een gesprek aan te nemen.

Geef als de DTR-indicator niet oplicht, na aanmelden op de console, de opdracht `ps ax` om te zien of FreeBSD probeert een `getty`-proces op de juiste poort te draaien. Er dienen tussen de weergegeven processen regels zoals de onderstaande te verschijnen:

```
114 ??  I      0:00.10 /usr/libexec/getty V19200 ttyu0
115 ??  I      0:00.10 /usr/libexec/getty V19200 ttyu1
```

Er kan ook iets als het volgende verschijnen:

```
114 d0  I      0:00.10 /usr/libexec/getty V19200 ttyu0
```

Als het modem nog geen gesprek heeft aangenomen, betekent dit dat `getty` het openen van de communicatiepoort voltooid heeft. Dit kan duiden op een probleem met de bekabeling of op een verkeerd ingesteld modem omdat `getty` niet in staat zou moeten zijn om de communicatiepoort te openen totdat CD (kiestoon) door het modem is bevestigd.

Indien er geen enkel `getty`-proces verschijnt dat wacht op het openen van de gewenste poort ttyuN, controleer dan de regels in /etc/ttys op vergissingen. Controleer ook het logboekbestand /var/log/messages om te zien of er logboekberichten van `init` of `getty` met betrekking tot problemen zijn. Indien er problemen zijn, controleer dan nogmaals de instellingenbestanden /etc/ttys en /etc/gettytab, alsook de betreffende speciale apparaatbestanden /dev/ttyuN, op vergissingen, ontbrekende regels of ontbrekende speciale apparaatbestanden.

Proberen om in te bellen

Probeer in te bellen op een systeem. Controleer of op het systeem-op-afstand 8 bits, geen pariteit en 1 stopbit gebruikt wordt. Probeer, indien er niet meteen een prompt verschijnt of als er rommel verschijnt, ongeveer eens per seconde op `Enter` te drukken. Probeer, indien er na een tijd nog geen prompt `login:` verschijnt, een `BREAK` te versturen. Probeer, indien er een modem wordt gebruikt dat op hoge snelheid werkt om te bellen, opnieuw in te bellen nadat de interfacesnelheid van het bellende modem is vastgezet (bijvoorbeeld via `AT&B1` op een U.S. Robotics® Sportster® modem).

Controleer, indien er nog steeds geen prompt `login:` verschijnt, nogmaals /etc/gettytab en controleer of:

- De initiële specificatie die in /etc/ttys voor de lijn staat overeenkomt met een naam van een specificatie in /etc/gettytab;
- Elke regel `nx=` overeenkomt met een naam van een andere specificatie in gettytab;
- Elke regel `tc=` overeenkomt met een naam van een andere specificatie in gettytab.

Controleer, indien er gebeld wordt maar het modem op het FreeBSD-systeem niet reageert, of het modem ingesteld is om de telefoon te beantwoorden als DTR bevestigd is. Controleer, indien het modem juist ingesteld lijkt te zijn, of de DTR-lijn bevestigd is door de indicatielampjes van het modem te controleren (indien die aanwezig zijn).

Neem een pauze en probeer het later nog eens indien alles meerdere malen is geprobeerd en het nog steeds niet werkt. Indien het nog steeds niet werkt, stuur dan een e-mail naar de FreeBSD algemene vragen mailinglijst met een beschrijving van het modem en het probleem en de mensen op de lijst zullen proberen te helpen.

24.5. Uitbeldienst



Vanaf FreeBSD 8.0 zijn de seriële poorten hernoemd van /dev/cuadN naar /dev/cuauN en van /dev/ttydN naar /dev/ttyuN. FreeBSD 7.X gebruikers moeten de documentatie aanpassen naar deze wijzigingen.

De volgende tips dienen voor het maken van een verbinding met een andere computer per modem. Dit is geschikt voor het opzetten van een terminalsessie met een gast op afstand.

Dit is nuttig bij het aanmelden op een BBS.

Dit soort verbinding kan extreem behulpzaam zijn om een bestand op het Internet te krijgen indien er problemen zijn met PPP. Indien FTP ergens voor nodig is en PPP kapot is, kan de terminalsessie voor FTP gebruikt worden. Vervolgens kan zmodem gebruikt worden om het naar de machine te verzenden.

24.5.1. Een gewone Hayes-modem wordt niet ondersteund, wat nu?

In feite is de hulppagina voor `tip` verouderd. Er is al een generiek belprogramma voor Hayes ingebouwd. Gebruik `at=hayes` in het bestand /etc/remote.

Het stuurprogramma voor Hayes is niet slim genoeg om enkele geavanceerde eigenschappen van nieuwere modems te herkennen. Berichten als `BUSY` (in gesprek), `NO DIALTONE` (geen kiestoon) of `CONNECT 115200` (verbinden 115200) verwarren het stuurprogramma. Schakel deze berichten uit bij gebruik van `tip` (door middel van `ATX0&W`).

Verder is de beltimeout voor `tip` 60 seconden. Het modem dient een lagere waarde te gebruiken om te voorkomen dat `tip` denkt dat er een probleem met de communicatie is. Probeer `ATS7=45&W`.

24.5.2. Hoe deze `AT`-commando's in te geven?

Maak een zogenaamde "directe" regel in het bestand /etc/remote aan. Als het modem bijvoorbeeld aan de eerste seriële poort, /dev/cuau0, is gekoppeld, voeg dan de volgende regel toe:

```
cuau0:dv=/dev/cuau0:br#19200:pa=none
```

Gebruik voor de mogelijkheid `br` de hoogst ondersteunde snelheid van het modem in bps. Typ hierna `tip cuau0` om een verbinding met het modem te maken.

Als alternatief kan `cu` als `root` met het volgende commando gebruikt worden:

```
# cu
      -llijn
      -ssnelheid
```

De waarde *lijn* is de seriële poort (bijvoorbeeld /dev/cuau0) en *snelheid* is de snelheid (bijvoorbeeld `57600`). Als alle AT-commando's zijn ingevoerd, voer dan `~.` in om het programma te verlaten.

24.5.3. Het teken `@` voor de optie pn werkt niet!

Het teken `@` in de telefoonnummermogelijkheid vertelt `tip` om in /etc/phones naar een telefoonnummer te kijken. Maar het teken `@` is ook een speciaal teken in specificatiebestanden als /etc/remote. Gebruik een backslash om hieraan te ontsnappen:

```
pn=\@
```

24.5.4. Hoe een telefoonnummer op de opdrachtregel te draaien?

Voeg een zogenaamde "generieke" regel aan het bestand /etc/remote toe. Bijvoorbeeld:

```
tip115200|Bel elk telefoonnummer met 115200 bps:\
        :dv=/dev/cuau0:br#115200:at=hayes:pa=none:du:
tip57600|Bel elk telefoonnummer met 57600 bps:\
        :dv=/dev/cuau0:br#57600:at=hayes:pa=none:du:
```

Hierna zijn onder andere de volgende mogelijkheden beschikbaar:

```
# tip -115200 5551234
```

Indien `cu` boven `tip` geprefereerd wordt, dient een generieke regel voor `cu` gebruikt te worden:

```
cu115200|Gebruik cu om elk nummer met 115200bps te bellen:\
    :dv=/dev/cuau1:br#57600:at=hayes:pa=none:du:
```

Voer in:

```
# cu 5551234 -s 115200
```

24.5.5. Dient de bps-snelheid telkens ingevoerd te worden?

Voeg een regel toe voor `tip1200` of `cu1200`, maar gebruik een bps-snelheid die geschikt is voor de br-mogelijkheid. `tip` meent dat 1200 bps een goede standaardwaarde is, hierdoor zoekt het naar een regel `tip1200`. Uiteraard hoeft 1200 bps niet gebruikt te worden.

24.5.6. Een aantal hosts met een terminalserver benaderen

Om niet iedere keer te hoeven wachten totdat er verbinding is en `CONNECT host` in te typen, kan de mogelijkheid `cm` van `tip` gebruikt worden. Als voorbeeld bieden de onderstaande regels in /etc/remote de mogelijkheid om `tip pain` of `tip muffin` in te typen om met de hosts `pain` of `muffin` te verbinden, en `tip deep13` om naar de terminalserver te gaan:

```
pain|pain.deep13.com|Forresters machine:\
        :cm=CONNECT pain\n:tc=deep13:
muffin|muffin.deep13.com|Franks machine:\
        :cm=CONNECT muffin\n:tc=deep13:
deep13:Gizmonics Institute terminalserver:\
        :dv=/dev/cuau2:br#38400:at=hayes:du:pa=none:pn=5551234:
```

24.5.7. Kan tip meer dan één lijn voor elke site proberen?

Dit is een vaak een probleem als een universiteit een handvol modemlijnen en enkele duizenden studenten heeft die ze proberen te gebruiken.

Voeg een regel voor de universiteit toe in /etc/remote en gebruik `@` voor de mogelijkheid `pn`:

```
grote-universiteit:\
        :pn=\@:tc=dialout
dialout:\
        :dv=/dev/cuau3:br#9600:at=courier:du:pa=none:
```

Voeg hierna de telefoonnummers voor de universiteit toe aan /etc/phones:

```
grote-universiteit 5551111
grote-universiteit 5551112
grote-universiteit 5551113
grote-universiteit 5551114
```

Het commando `tip` probeert elk nummer in de volgorde van de lijst alvorens op te geven. Om de pogingen te herhalen, kan `tip` in een while-lus gedraaid worden.

24.5.8. Waarom moet `Ctrl` + `P` tweemaal worden ingedrukt om `Ctrl` + `P` éénmaal te versturen?

`Ctrl` + `P` is het standaard "forceer"-karakter, dat gebruikt wordt om `tip` te vertellen dat het volgende karakter letterlijk genomen dient te worden. Het forceerkarakter kan met de ontsnapping `~s`, wat "stel een variabele in" betekent, op elk ander karakter ingesteld worden.

Typ `~sforce=enkel-karakter` in gevolgd door een nieuwe regel. *enkel-karakter* is elk enkel karakter. Indien *enkel-karakter* weggelaten wordt, is het forceerkarakter het nulkarakter, wat door middel van `Ctrl` + `2` of `Ctrl` + `spatie` verkregen kan worden. Een redelijke standaardwaarde voor *enkel-karakter* is `Shift` + `Ctrl` + `6`, die slechts op enkele terminalservers gebruikt wordt.

Het forceerkarakter kan op elk gewenst karakter ingesteld worden door het volgende op te nemen in het bestand $HOME/.tiprc:

```
force=enkel-karakter
```

24.5.9. Alle ingevoerde tekst staat opeens in hoofdletters?

Waarschijnlijk is `Ctrl` + `A` ingedrukt, het "raisechar" van `tip`, dat speciaal voor mensen met een kapotte caps-lock toets is ontworpen. Gebruik `~s` zoals boven is aangegeven en stel de variabele `raisechar` op iets redelijks in. Het kan zelfs op hetzelfde als het forceerkarakter worden ingesteld, indien het onwaarschijnlijk is dat een van deze mogelijkheden ooit gebruikt wordt.

Hier volgt een voorbeeld voor het bestand .tiprc dat perfect is voor gebruikers van Emacs die `Ctrl` + `2` en `Ctrl` + `A` vaak moeten gebruiken:

```
force=^^
raisechar=^^
```

De `^^` is `Shift` + `Ctrl` + `6`.

24.5.10. Hoe kan met `tip` bestanden worden verstuurd?

In de communicatie met een ander UNIX®-systeem kunnen bestanden verzonden en ontvangen worden met de commando's `~p` (put) en `~t` (take). Deze commando's draaien `cat` en `echo` op een systeem op afstand om bestanden aan te nemen en te verzenden. De syntaxis is:

`~p` lokaal-bestand [bestand-op-afstand]

`~t` bestand-op-afstand [lokaal-bestand]

Er wordt niet op fouten gecontroleerd, het is dus verstandig om een ander protocol te gebruiken, zoals zmodem.

24.5.11. Hoe kan zmodem samen met `tip` draaien?

Start om bestanden te ontvangen het verstuurprogramma aan de andere kant. Typ daarna `~C rz` om ze lokaal te ontvangen.

Start om bestanden te versturen het ontvangprogramma aan de andere kant. Typ daarna `~C sz bestanden` om ze naar het systeem aan de andere kant te versturen.

24.6. Seriële console opzetten



Vanaf FreeBSD 8.0 zijn de seriële poorten hernoemd van /dev/cuadN naar /dev/cuauN en van /dev/ttydN naar /dev/ttyuN. FreeBSD 7.X gebruikers moeten de documentatie aanpassen naar deze wijzigingen.

24.6.1. Inleiding

FreeBSD biedt de mogelijkheid om op een systeem op te starten met slechts een domme terminal en een seriële poort als console. Dit soort opstellingen is handig voor twee soorten mensen: voor systeembeheerders die FreeBSD willen installeren op machines die geen toetsenbord of beeldscherm hebben en voor ontwikkelaars die de kernel of apparaatstuurprogramma's willen debuggen.

Zoals beschreven in Het FreeBSD opstartproces, gebruikt FreeBSD drie fasen voor het opstarten. De eerste twee fasen bevinden zich in de code van het opstartblok dat zich aan het begin van de opstartslice van FreeBSD op de opstartschijf bevindt. Het opstartblok laadt vervolgens de opstartlader (/boot/loader) en draait als de code van de derde fase.

Om de seriële console gereed te maken moeten de code in het opstartblok, de code van de opstartlader en de kernel worden ingesteld.

24.6.2. De seriële console instellen, korte versie

Deze sectie neemt aan dat de standaard opstelling wordt gebruikt en dat een kort overzicht voor het opzetten van de seriële console gewenst is.

1. Verbind de seriële kabel met COM1 en de leidende terminal;
2. Om alle opstartmeldingen op de seriële console te zien, dient het volgende commando als supergebruiker uitgevoerd te worden:

```
# echo 'console="comconsole"' >> /boot/loader.conf
```

3. Bewerk /etc/ttys en wijzig `off` in `on` en `dialup` in `vt100` voor de regel ttyu0. Indien dit niet gebeurt is er geen wachtwoord nodig om met de seriële console te verbinden, wat tot een mogelijk beveiligingslek leidt;
4. Start het systeem opnieuw op om te zien of de veranderingen effect hebben.

Indien een andere instelling nodig is, is er een diepgaandere uitleg over instellingen beschikbaar in De seriële console instellen.

24.6.3. De seriële console instellen

1. Bereid een seriële kabel voor.

   Benodigd zijn een nulmodem-kabel òf een standaard seriële kabel samen met een nulmodem-adapter. Zie Kabels en poorten voor een beschrijving van seriële kabels.

2. Ontkoppel het toetsenbord.

   De meeste PC-systemen zoeken naar het toetsenbord tijdens de Power-On Self-Test (POST) en geven een foutmelding als het toetsenbord niet is gevonden. Sommige systemen klagen luid over het ontbreken van een toetsenbord en gaan niet verder met opstarten totdat het is aangesloten.

   Indien de computer klaagt over de fout, maar desondanks opstart, is het niet nodig iets speciaals te doen. Sommige machines waarop Phoenix BIOS is geïnstalleerd melden enkel `Toetsenbord faalde` en gaan normaal door met opstarten.

   Indien de machine weigert zonder toetsenbord op te starten dient het BIOS ingesteld te worden zodat het deze fout negeert (als het dit kan). Raadpleeg het handboek van het moederbord voor verdere aanwijzingen.

   Stel het toetsenbord in op "Niet geïnstalleerd" in de BIOS-instellingen. Het is dan nog steeds mogelijk om het toetsenbord te gebruiken. Dit zorgt er alleen voor dat het BIOS niet naar een toetsenbord zoekt tijdens het aanzetten. Het BIOS dient niet te klagen als het toetsenbord ontbreekt. Het is mogelijk om het toetsenbord aangesloten te laten, zelfs als deze vlag is ingesteld op "Niet geïnstalleerd" en het toetsenbord werkt nog steeds. Kijk, als de bovenstaande optie niet in het BIOS aanwezig is, naar een optie "Halt on Error". Het instellen van deze optie op "All but keyboard" of zelfs op "No Errors" zal hetzelfde effect hebben.

   Als een systeem een PS/2®-muis heeft, is het goed mogelijk dat naast het toetsenbord ook de muis losgekoppeld moet worden. Dit komt doordat PS/2®-muizen wat hardware met het toetsenbord delen. Als de muis aangesloten blijft, kan het zoeken naar het toetsenbord als resultaat

> hebben dat het toetsenbord er nog steeds is. Een Gateway 2000 Pentium 90 MHz systeem met een AMI BIOS schijnt zich op deze manier te gedragen. Over het algemeen is dit geen probleem aangezien een muis zonder toetsenbord sowieso weinig nut heeft.

3. Sluit een domme terminal aan op COM1 (sio0).

   Indien er geen domme terminal aanwezig is, kan een oude PC met een modemprogramma of de seriële poort van een andere UNIX® machine gebruikt worden. Indien er geen COM1 (sio0) aanwezig is dient deze geregeld te worden. Op dit moment is er geen manier om een andere poort dan COM1 voor de opstartblokken te selecteren, afgezien van deze opnieuw te compileren. Indien COM1 al voor een ander apparaat gebruikt wordt, verwijder dat apparaat dan tijdelijk en installeer een nieuw opstartblok en een nieuwe kernel zodra FreeBSD werkt. Er wordt aangenomen dat COM1 sowieso beschikbaar is op een bestands-/reken-/terminalserver. Als COM1 ècht voor iets anders nodig is (en het niet mogelijk is om dat op COM2 (sio1) over te zetten), is het sowieso al onverstandig om hiermee bezig te zijn.)

4. Controleer of het instellingenbestand van de kernel de juiste vlaggen ingesteld heeft voor COM1 (sio0).

   Relevante vlaggen zijn:

   `0x10`

   Zet console-ondersteuning voor deze eenheid aan. De andere consolevlaggen worden genegeerd tenzij deze is aangezet. Momenteel kan ten hoogste één eenheid console-ondersteuning hebben. De eerste (in de volgorde van het instellingenbestand) waarvan deze vlag is aangezet heeft de voorkeur. Deze optie zelf maakt de seriële poort geen console. Stel de volgende vlag in of gebruik de onderstaande optie `-h` samen met deze vlag.

   `0x20`

   Dwingt deze eenheid om de console te zijn (tenzij er een andere console met hogere prioriteit is), ongeacht de onderstaande optie `-h`. De vlag `0x20` dient samen met de vlag `0x10` gebruikt te worden.

   `0x40`

   Reserveert deze eenheid (in samenwerking met `0x10`) en maakt de eenheid ontoegankelijk voor normale toegang. Deze vlag dient niet aangezet te worden op de seriële poort van de eenheid die als seriële console gebruikt gaat worden. De enige functie van deze vlag is de eenheid voor het debuggen van de kernel op afstand aan te merken. Zie het Ontwikkelaarshandboek voor meer informatie over debuggen op afstand.

   Voorbeeld:

   ```
   device sio0 at isa? port IO_COM1 flags 0x10 irq 4
   ```

   Zie de hulppagina sio(4) voor meer details.

Indien de vlaggen niet waren ingesteld, dient UserConfig gedraaid te worden (op een andere console) of de kernel opnieuw gecompileerd te worden.

5. Maak boot.config aan in de hoofdmap van de partitie a van de opstartschijf.

   Dit bestand instrueert de code op het opstartblok hoe het systeem opgestart dient te worden. Om de seriële console te activeren, zijn één of meer van de volgende opties nodig. Indien meerdere opties gewenst zijn, dienen ze allemaal op dezelfde regel te staan:

   `-h`

   Wisselt tussen de interne en de seriële console. Indien bijvoorbeeld vanaf de interne (video)console opgestart wordt, kan `-h` gebruikt worden om het console-apparaat van de opstartlader en de kernel om te leiden naar de seriële console. Indien vanaf de seriële poort opgestart wordt, kan `-h` gebruikt worden om de opstartlader en de kernel het videoscherm als console te laten gebruiken.

   `-D`

   Wisselt tussen opstellingen met een enkele en een dubbele console. In opstellingen met een enkele console is de console òfwel de interne console (videoscherm) òfwel de seriële poort, afhankelijk van bovenstaande optie `-h`. In opstellingen met een dubbele console worden zowel het videoscherm als de seriële poort tegelijkertijd console, ongeacht de toestand van de optie `-h`. De opstelling met een dubbele console heeft alleen effect als het opstartblok draait. Zodra de opstartlader het overneemt, wordt de console die met de optie `-h` gespecificeerd is de enige console.

   `-P`

   Zorgt ervoor dat het opstartblok naar het toetsenbord zoekt. Als er geen toetsenbord wordt gevonden, worden de opties `-D` en `-h` automatisch ingesteld.

   > Vanwege ruimtebeperkingen in de huidige versie van het opstartblok, is de optie `-P` alleen in staat om uitgebreide toetsenborden te detecteren. Toetsenborden met minder dan 101 toetsen (en zonder de toetsen F11 en F12) worden mogelijk niet gedetecteerd. Toetsenborden op sommige laptops worden vanwege deze beperking mogelijk niet correct gevonden. Indien dit het geval is met een systeem, vermijd dan de optie `-P`. Helaas is er geen mogelijkheid om dit probleem te omzeilen.

   Gebruik om de console automatisch te selecteren òfwel de optie `-P` òfwel de optie `-h` om de seriële console te activeren.

   De andere opties beschreven in boot(8) kunnen ook gebruikt worden.

   De opties, behalve `-P`, worden aan de opstartlader (/boot/loader doorgegeven. De opstartlader bepaalt of de interne videopoort of de seriële poort de console wordt door enkel naar de toestand van de optie `-h` te kijken. Dit betekent dat als de optie `-D`, maar niet de optie `-h` in /boot.config gespecificeerd wordt, de seriële poort alleen tijdens het opstartblok als console gebruikt kan worden, de opstartlader gebruikt het interne videoscherm als console.

6. Start de machine op.

   Als FreeBSD gestart wordt, tonen de opstartblokken de inhoud van /boot.config op de console. Bijvoorbeeld:

   ```
   /boot.config: -P
   Keyboard: no
   ```

   De tweede regel verschijnt alleen als `-P` in /boot.config staat en aangegeven wordt of het toetsenbord aanwezig of afwezig is. Deze berichten gaan of naar de seriële of interne console of naar beide, afhankelijk van de optie in /boot.config.

Opties	Bericht gaat naar
geen	interne console
`-h`	seriële console
`-D`	seriële en interne consoles
`-Dh`	seriële en interne consoles
`-P`, toetsenbord aanwezig	interne console
`-P`, toetsenbord afwezig	seriële console

   Na de bovenstaande berichten is er een korte pauze voordat de opstartblokken doorgaan met het laden van de opstartlader en voordat er verdere berichten op de console worden afgebeeld. Normaalgesproken hoeven de opstartblokken niet onderbroken te worden, maar het kan gedaan worden om er zeker van te zijn dat alles goed is ingesteld.

   Om het opstartproces te onderbreken, kan op elke andere toets dan `Enter` gedrukt worden. De opstartblokken vragen dan om verdere actie. Er verschijnt iets als het volgende:

   ```
   >> FreeBSD/i386 BOOT
   Default: 0:ad(0,a)/boot/loader
   boot:
   ```

   Controleer of de bovenstaande boodschap naar de seriële of interne console of beide gaat, naar gelang de opties in /boot.config. Indien de boodschap op de juiste console verschijnt kan op `Enter` gedrukt worden om het opstartproces voort te zetten.

   Als de seriële console gewenst is maar de prompt niet op de seriële terminal verschijnt, is er iets mis met de instellingen. Voer in de tussentijd `-h` in en druk op `Enter` of `Return` (indien mogelijk) om aan het opstartblok (en vervolgens de opstartlader en de kernel) te vertellen dat de seriële poort console moet worden. Controleer als het systeem draait wat er verkeerd ging.

Nadat de opstartlader is geladen en het derde stadium van het opstartproces bereikt is, kan er nog steeds gewisseld worden tussen de interne console en de seriële console door de juiste

omgevingsvariabelen in de opstartlader in te stellen. Zie De console vanuit de opstartlader veranderen.

24.6.4. Samenvatting

Hieronder volgt een samenvatting van de verschillende instellingen die in deze sectie en de uiteindelijk gekozen console beschreven zijn.

Geval 1: vlaggen ingesteld op 0x10 voor sio

```
device sio0 at isa? port IO_COM1 flags 0x10 irq 4
```

Opties in /boot.config	Console tijdens de opstartblokken	Console tijdens de opstartlader	Console in kernel
niets	intern	intern	intern
`-h`	serieel	serieel	serieel
`-D`	serieel en intern	intern	intern
`-Dh`	serieel en intern	serieel	serieel
`-P`, toetsenbord aanwezig	intern	intern	intern
`-P`, toetsenbord afwezig	serieel en intern	serieel	serieel

Geval 2: vlaggen ingesteld op 0x30 voor sio

```
device sio0 at isa? port IO_COM1 flags 0x30 irq 4
```

Opties in /boot.config	Console tijdens de opstartblokken	Console tijdens de opstartlader	Console in kernel
niets	intern	intern	serieel
`-h`	serieel	serieel	serieel
`-D`	serieel en intern	intern	serieel
`-Dh`	serieel en intern	serieel	serieel
`-P`, toetsenbord aanwezig	intern	intern	serieel
`-P`, toetsenbord afwezig	serieel en intern	serieel	serieel

24.6.5. Tips voor de seriële console

Een hogere snelheid voor de seriële poort instellen

Standaard zijn de instellingen van de seriële poort: 9600 baud, 8 bits, geen pariteit, en 1 stopbit.

Indien het wenselijk is om de snelheid te veranderen, zijn de volgened opties beschikbaar:

- Hercompileer de opstart blokken met `BOOT_COMCONSOLE_SPEED` ingesteld op de nieuwe console snelheid. Zie Een andere seriële poort dan sio0 voor de console gebruiken voor gedetailleerde instructies over het bouwen en installeren van nieuwe opstartblokken.

  Als de seriële poort anders is gespecificeerd dan met `-h` bij het opstarten, of als de seriële console die gebruikt wordt door de kernel anders is dan die gebruikt wordt door de opstart blokken, dan moet de volgende optie aan het kernel instellingen bestand worden toegevoegd en moet de kernel opnieuw gecompileerd worden:

  ```
  options CONSPEED=19200
  ```

- Gebruik de `-S` opstartoptie van de kernel. De optie `-S` kan worden toegevoegd aan het bestand /boot.config. Zie de handleiding boot(8) voor een beschrijving over hoe opties kunnen worden toegevoegd aan /boot.conf, en welke opties ondersteund worden.
- Zet de `comconsole_speed` optie in het /boot/loader.conf bestand.

  Deze optie is ervan afhankelijk dat de `console`, `boot_serial` en `boot_multicons` ingesteld staan in /boot/loader.conf. Een voorbeeld van hoe `comconsole_speed` gebruikt kan worden om de console snelheid aan te passen:

  ```
  boot_multicons="YES"
  boot_serial="YES"
  console_speed="115200"
  console="comconsole,vidconsole"
  ```

Een andere seriële poort dan sio0 voor de console gebruiken

Het gebruik van een andere poort dan sio vergt wat hercompileren. Indien het gewenst is om een andere seriële poort te gebruiken, hercompileer dan de opstartblokken, de opstartlader en de kernel als volgt:

1. De broncode van de kernel moet beschikbaar zijn. Zie FreeBSD updaten en upgraden;
2. Bewerk /etc/make.conf en stel `BOOT_COMCONSOLE_PORT` in op het adres van de te gebruiken poort (0x3F8, 0x2F8, 0x3E8 of 0x2E8). Alleen sio0 tot en met sio3 (COM1 tot en met COM4) zijn te gebruiken. Seriële kaarten met meerdere poorten werken niet. Interrupts instellen is niet nodig;
3. Maak een aangepast kernelinstellingenbestand aan en voeg de juiste vlaggen toe voor de te gebruiken seriële poort. Als bijvoorbeeld sio1 (COM2) de console moet worden:

   ```
   device sio1 at isa? port IO_COM2 flags 0x10 irq 3
   ```

   Alternatief:

```
device sio1 at isa? port IO_COM2 flags 0x30 irq 3
```

Stel de consolevlaggen voor de andere seriële poorten niet in;

4. Hercompileer en installeer de opstartblokken en de opstartlader:

```
# cd /sys/boot
# make clean
# make
# make install
```

5. Herbouw en installeer de kernel;
6. Schrijf de opstartblokken met disklabel(8) naar de opstartschijf en start met de nieuwe kernel op.

De debugger DDB gebruiken via de seriële verbinding

Als het wenselijk is om vanuit de seriële console in de kerneldebugger te vallen - nuttig voor diagnose op afstand, maar ook gevaarlijk indien een onbedoelde BREAK op de seriële poort wordt gegenereerd! - compileer de kernel dan met de volgende opties:

```
options BREAK_TO_DEBUGGER
options DDB
```

Een aanmeldprompt op de seriële console krijgen

Hoewel dit niet nodig is, kan het gewenst zijn om een *aanmeld*prompt over de seriële lijn te krijgen, nu het mogelijk is om opstartboodschappen te zien en de kerneldebugsessie door de seriële console betreden kan worden. Hier volgt hoe het te doen.

Open het bestand /etc/ttys met een tekstverwerker en zoek de volgende regels:

```
ttyu0 "/usr/libexec/getty std.9600" unknown off secure
ttyu1 "/usr/libexec/getty std.9600" unknown off secure
ttyu2 "/usr/libexec/getty std.9600" unknown off secure
ttyu3 "/usr/libexec/getty std.9600" unknown off secure
```

ttyu0 tot en met ttyu3 komen overeen met COM1 tot en met COM4. Wijzig `off` in `on` voor de gewenste poort. Als de snelheid van de seriële poort is gewijzigd, wijzig dan `std.9600` zodat het met de huidige instelling overeenkomt, bijvoorbeeld `std.19200`.

Het kan ook wenselijk zijn om het terminaltype te wijzigen van `unknown` naar het eigenlijke type van de seriële terminal.

Voer `kill -HUP 1` uit na het wijzigen van het bestand om de wijzigingen actief te maken.

24.6.6. De console vanuit de opstartlader veranderen

De vorige secties beschreven hoe de seriële console ingesteld kan worden door het instellen van het opstartblok. Deze sectie toont dat het mogelijk is om de console te specificeren door het invoeren van enkele opdrachten en omgevingsvariabelen in de opstartlader. Aangezien de opstartlader tijdens het derde stadium van het opstartproces wordt geactiveerd, na het opstartblok, overheersen de instellingen in de opstartlader de instellingen in het opstartblok.

De seriële console instellen

Het is mogelijk om de opstartlader en de kernel gebruik te laten maken van de seriële console door slechts één regel naar /boot/loader.conf te schrijven:

```
console="comconsole"
```

Dit heeft effect ongeacht de instellingen in het opstartblok die in de vorige sectie zijn besproken.

Het is verstandig om bovenstaande regel de eerste regel van /boot/loader.conf te maken om de opstartboodschappen zo vroeg mogelijk op de seriële console te kunnen zien.

Evenzo kan de interne videoconsole worden gespecificeerd met:

```
console="vidconsole"
```

Indien de omgevingsvariabele `console` van de opstartlader niet ingesteld wordt, gebruikt de opstartlader, en vervolgens de kernel, de console die door de optie `-h` in het opstartblok wordt aangegeven.

De console kan worden gespecificeerd in /boot/looader.conf.local of in /boot/loader.conf.

Zie loader.conf(5) voor meer informatie.



Momenteel heeft de opstartlader een optie die gelijk is aan de optie `-P` van het opstartblok en is er geen voorziening om automatisch de interne console en de seriële console te selecteren afhankelijk van de aanwezigheid van een toetsenbord.

Een andere seriële poort dan sio voor de console gebruiken

Compileer de opstartlader opnieuw om een andere seriële poort dan sio voor de seriële console te gebruiken. Volg de procedure zoals beschreven in Een andere seriële poort dan sio0 voor de console gebruiken.

24.6.7. Valkuilen

De doelstelling van dit stuk is beheerders in staat te stellen om toegewijde servers te installeren die geen grafische hardware of aangesloten toetsenborden nodig hebben. Hoewel de meeste systemen zonder toetsenbord opstarten, zijn er helaas aardig wat die niet zonder een grafische adapter opstarten. Machines met een AMI BIOS kunnen ingesteld worden om zonder grafische adapter op

te starten door de instelling "graphics adapter" in de CMOS-instellingen te wijzigen in "Not installed".

De meeste systemen ondersteunen deze optie echter niet en weigeren om zonder weergavehardware op te starten. Voor deze machines is het nodig om een of andere grafische kaart in een systeem te laten (zelfs al is het een aftandse monochrome kaart) hoewel het niet nodig is om een beeldscherm aan te sluiten. Ook kan geprobeerd worden om een AMI BIOS te installeren.

Hoofdstuk 25. PPP en SLIP

25.1. Overzicht

FreeBSD heeft een aantal manieren om de ene computer met de andere te verbinden. Om een netwerk of internet verbinding op te zetten door een inbelmodem, of om anderen toe te staan dit te doen door de machine heen vereist het gebruik van PPP en SLIP. Dit hoofdstuk beschrijft het opzetten van op modems gebaseerde diensten in meer detail.

Na het lezen van dit hoofdstuk weet u:

- Hoe gebruikers PPP opgezet kan worden.
- Hoe kernel-PPP opgezet kan worden (alleen voor FreeBSD 7.X).
- Hoe PPPoE opgezet kan worden (PPP over Ethernet).
- Hoe PPPoA opgezet kan worden (PPP over ATM).
- Hoe een SLIP-server en cliënt opgezet kan worden en hoe dat geconfigureerd wordt (alleen voor FreeBSD 7.X).

Voordat dit hoofdstuk gelezen wordt, moet u:

- Bekend zijn met basis netwerk terminologie.
- De basis en doeleinden van een inbel verbinding en van PPP en/of SLIP.

U kunt zich afvragen wat het verschil is tussen gebruiker-PPP en kernel-PPP. Het antwoord is simpel: gebruiker-PPP verwerkt inkomend en uitgaande data in het gebruikersland in plaats van in de kernel. Dit is duur in de zin van het kopiëren van de data tussen de kernel en het gebruikersland, maar levert meer mogelijkheden voor de PPP implementatie. Gebruikers PPP gebruikt het tun apparaat om te communiceren met de buitenwereld. Kernel-PPP maakt gebruik van het ppp apparaat.



Voor de rest van dit hoofdstuk, zal gebruiker-PPP gebruikt worden als ppp tenzij er onderscheid gemaakt moet worden met andere PPP software zoals pppd. Tenzij anders vermeld moeten alle uitgelegde commando's in dit hoofdstuk gestart worden als de `root` gebruiker.

25.2. Gebruikmaken van gebruiker-PPP

25.2.1. Gebruiker-PPP

Vereisten

Dit document gaat er vanuit dat u de volgende punten beschikbaar heeft:

- Een account bij een Internet Service Provider (ISP) waarmee verbinding gemaakt wordt door middel van PPP.

- Een modem of een ander apparaat verbonden met uw PC en correct geconfigureerd zodat u verbinding kan maken met uw ISP.
- De inbelnummers van uw ISP.
- Uw loginnaam en wachtwoord (danwel een combinatie van een standaard UNIX®-stijl login en wachtwoord of een PAP of CHAP login en wachtwoordcombinatie).
- Het IP-adres van één of meerdere naamservers. Normaal gesproken krijgt u twee IP adressen van uw ISP om te gebruiken. Als u er echter geen één gekregen heeft, kunt u het commando `enable dns` gebruiken in ppp.conf en ppp zal de naamservers voor u configureren. Deze optie is afhankelijk van de PPP implementatie van de ISP, welke DNS onderhandeling moet ondersteunen.

De volgende informatie kan aangeleverd worden door uw ISP maar is niet echt noodzakelijk:

- Het IP-adres van de router van uw ISP. De router is de machine waarmee u verbinding maakt en welke ingesteld wordt als de *standaard route*. Als u deze informatie niet heeft, kunt u een willekeurig adres verzinnen waarna de PPP server van de ISP het juiste adres vertelt zodra u verbinding maakt.

  Dit IP-adres wordt door ppp `HISADDR` genoemd.

- Het netwerkmasker wat gebruikt moet worden. Als uw ISP deze niet heeft opgegeven, kan `255.255.255.255` gebruikt worden.
- Als uw ISP u een vast IP-adres en hostnaam levert, kunt u deze invoeren. In andere gevallen bepaalt de andere kant welk adres er uitgegeven wordt.

Als u niet in bezit bent van de vereiste informatie, moet u contact opnemen met uw ISP.



Door de rest van dit hoofdstuk worden in veel van de voorbeelden configuratie bestanden genummerd per regel. Deze nummers dienen alleen als hulp voor de presentatie en discussie en zijn verder niet bedoeld om daadwerkelijk geïmplementeerd te worden. Een juiste inspringing met tabs en spaties zijn daarbij ook belangrijk.

Automatische configuratie van PPP

Zowel `ppp` als `pppd` (de implementatie van PPP op kernel niveau) gebruiken de configuratie bestanden die zich in de map /etc/ppp bevinden. Voorbeelden configuraties voor gebruiker-PPP kunnen gevonden worden in /usr/shared/examples/ppp/.

Het configureren van `ppp` vereist dat u een aantal bestanden bewerkt, afhankelijk van uw eisen. Wat u moet invoeren is deels afhankelijk van wat uw ISP u aanbied met oog op statische IP-adressen (lees u krijgt een statisch adres welke u altijd gebruikt) of dynamisch (lees: uw IP-adres veranderd elke keer als u verbinding maakt met uw ISP).

PPP en statische IP-adressen

U moet het /etc/ppp/ppp.conf bewerken. Het zou dan als volgend eruit moeten zien:



Regels die eindigen met een : starten in de eerste kolom (het begin van de regel) - alle andere regels moeten inspringen zoals getoond door middel van spaties of tabs.

```
1     default:
2       set log Phase Chat LCP IPCP CCP tun command
3       ident user-ppp VERSION (built COMPILATIONDATE)
4       set device /dev/cuau0
5       set speed 115200
6       set dial "ABORT BUSY ABORT NO\\sCARRIER TIMEOUT 5 \
7                 \"\" AT OK-AT-OK ATE1Q0 OK \\dATDT\\T TIMEOUT 40 CONNECT"
8       set timeout 180
9       enable dns
10
11    provider:
12      set phone "(123) 456 7890"
13      set authname foo
14      set authkey bar
15      set login "TIMEOUT 10 \"\" \"\" gin:--gin: \\U word: \\P col: ppp"
16      set timeout 300
17      set ifaddr x.x.x.x y.y.y.y 255.255.255.255 0.0.0.0
18      add default HISADDR
```

Regel 1

Deze regel identificeert de standaard regel. Commando's in deze regel worden automatisch gestart zodra ppp gestart wordt.

Regel 2

Zet de log paramaters aan. Zodra de configuratie naar verwachting werkt, moet deze regel aangepast worden naar:

```
set log phase tun
```

om te voorkomen dat er extreem grote log files gemaakt worden.

Regel 3

Vertelt PPP hoe het zich moet identificeren aan de router aan de andere kant, als deze problemen heeft met het onderhandelen en het opzetten van de link en het leveren van informatie die de beheerders van de andere kant nuttig kunnen vinden om zulke problemen te onderzoeken.

Regel 4

Identificeert het apparaat waarmee het modem verbonden is. COM1 is /dev/cuau0 en COM2 is /dev/cuau1.

Regel 5

Stelt de snelheid in waarmee verbinding gemaakt wordt. Als 115200 niet werkt (wat wel zou moeten kunnen met elk nieuw modem), probeert u dan de instelling van 38400.

Regels 6 & 7

De inbelregel. Gebruiker-PPP gebruikt een "expect-send" syntax wat vergelijkbaar is met het chat(8) programma. Bekijk de handleiding voor meer informatie over de mogelijkheden van deze taal.

Let op dat dit commando doorgaat op de volgende regel zodat deze leesbaar blijft. Elk commando in ppp.conf kan dit doen als het laatste karakter op een regel, het \ karakter is.

Regel 8

Stelt de idle timeout in voor een link. 180 seconden is standaard, dus deze regel is puur cosmetisch.

Regel 9

Vertelt PPP om de andere kant te vragen om een bevestiging van de lokale naamserver instellingen. Als u een lokale naamserver draait moet deze regel uitgecommentarieërd of verwijderd worden.

Regel 10

Een blanco regel voor de leesbaarheid. Blanco regels worden door PPP genegeerd.

Regel 11

Identificeert een sectie voor de provider die "provider" genoemd wordt. Dit kan gewijzigd worden in de naam van uw provider zodat er later gebruik gemaakt van worden bij de optie `load provider` om een verbinding op te zetten.

Regel 12

Stelt het telefoonnummer in voor deze provider. Meerdere telefoonnummers kunnen gespecificeerd worden door gebruik te maken van de dubbele punt (`:`) of het pipe karakter (`|`) als scheidingsteken. Het verschil tussen de twee scheidingstekens wordt beschreven in de ppp(8) handleiding. Om samen te vatten, als u wilt rouleren tussen de nummers gebruikt u dan een dubbelepunt. Als u altijd het eerste nummer als eerste wilt draaien en alleen de andere nummers wilt draaien als het eerste nummer niet werkt, gebruik dan het pipe karakter. Quote altijd de hele set van telefoonnummers zoals getoond.

U moet het telefoonnummer citeren met dubbele quotes (`"`) als er enige intentie is in het gebruik van spaties in het telefoonummer. Dit kan een simpele, maar subtiele fout creeëren.

Regels 13 & 14

Identificeert de gebruikersnaam en het wachtwoord. Wanneer gebruik gemaakt wordt van een UNIX® stijl login worden deze waarden verwezen door het `set login` commando door gebruik te maken van de \U en \P variabelen. Wanneer er verbinding gemaakt wordt door PPP en CHAP worden deze waardes gebruikt tijdens het authenticeren.

Regels 15

Als u gebruik maakt van PPP en CHAP, zal er geen login op dit moment zijn, en moet deze regel uitgecommentarieërd of verwijderd worden. Zie het PAP en CHAP authenticatie hoofdstuk voor meer details.

De login regel is hetzelfde als de chat-achtige syntax van de inbelregel. In dit voorbeeld werkt de reegl voor een dienst wiens login sessie als volgt eruit ziet:

```
J. Random Provider
login: foo
password: bar
protocol: ppp
```

U moet dit script aanpassen om aan uw behoeften te voldoen. Wanneer u dit script voor het eerst schrijft, moet u ervoor zorgen dat u de "chat" log optie heeft aangezet zodat u kunt bepalen of de communicatie gaat zoals verwacht.

Regel 16

Selt de standaard idle timeout in (in seconden) voor de connectie. Hier wordt de connectie automatisch afgesloten na 300 seconden van inactiviteit. Als u nooit een timeout wilt krijgen, kunt u de waarde op nul zetten of gebruik maken van de optie `-ddial` op de commando regel.

Regel 17

Stelt het interface adres in. De regel *x.x.x.x* moet vervangen worden door het IP-adres dat uw provider aan u heeft uitgegeven. De regel *y.y.y.y* moet vervangen worden door het IP-adres dat uw provider aan u heeft gegeven voor de router (de machine waarmee u verbinding maakt). Als uw ISP u geen router adres heeft gegeven, gebruik dan `10.0.0.2/0`. Als u gebruik moet maken van een "gegokt", zorg ervoor dat er een regel staat in /etc/ppp/ppp.linkup zoals beschreven in de instructies voor PPP en dynamische IP adressen. Als deze regel weggelaten wordt kan `ppp` niet in `-auto` mode starten.

Regel 18

Voegt een standaard routering toe naar uw providers router. Het speciale `HISADDR` woord, wordt vervangen door het router adres zoals gespecificeerd op regel 17. Het is belangrijk dat deze regel na regel 17 komt, anders is `HISADDR` nog niet geïnitialiseerd.

Als u ppp niet in `-auto` mode wilt draaien, moet deze regel verplaatst worden naar het ppp.linkup bestand.

Het is niet nodig om een regel toe te voegen aan ppp.linkup wanneer u een statisch IP-adres krijgt en ppp met de `-auto` mode gestart is omdat uw routerings tabel al correcte regels heeft voordat u verbinding maakt. U kunt echter een regel aanmaken om programma's te starten nadat de verbinding opgezet is. Dit wordt later uitgelegd met een voorbeeld over sendmail.

Voorbeeld van configuratiebestanden kunnen gevonden worden in de map usr/shared/examples/ppp.

PPP en dynamische IP-adressen

Als uw provider geen statisch IP-adres aanlevert kan `ppp` geconfigureerd worden om het lokale en het remote adres te onderhandelen. Dit wordt gedaan door het "gokken" van een IP-adres en PPP toestaan dit adres te corrigeren door gebruik te maken van het IP Configuration Protocol (IPCP) nadat er een verbinding opgezet is. De ppp.conf configuratie is verders hetzelfde als voor de PPP en statische IP adressen, met de volgende wijziging:

```
17      set ifaddr 10.0.0.1/0 10.0.0.2/0 255.255.255.255 0.0.0.0
```

Nogmaals, het regelnummer hoeft niet te worden toegevoegd, deze dient puur ter referentie. Indentatie van minstens één spatie is vereist.

Regel 17

Het nummer achter het `/` karakter is het aantal netwerk master bits van het adres die ppp eist. Het is mogelijk dat u IP-adressen wilt gebruiken die meer van toepassing zijn op uw situatie, maar bovenstaand voorbeeld zal altijd werken.

Het laatste argument (`0.0.0.0`) vertelt PPP om te onderhandelen met het adres `0.0.0.0` in plaats van met `10.0.0.1` en is benodigd voor sommige ISPs. Gebruik `0.0.0.0` niet als eerste argument voor het commando `set ifaddr`, omdat dit ervoor zorgt dat PPP geen initiële route kan opzetten in `-auto` mode.

Als u niet draait in `-auto` mode, moet u een nieuwe regel toevoegen aan /etc/ppp/ppp.linkup. ppp.linkup wordt uitgevoerd nadat een connectie is opgezet. Op dit moment krijgt `ppp` het interface adres en is het mogelijk om regels toe te voegen aan de route tabel:

```
1     provider:
2      add default HISADDR
```

Regel 1

Bij het tot stand brengen van een verbinding zal `ppp` kijken voor een corresponderende regel in ppp.linkup volgens de volgende criteria: Als eerste, probeert het hetzelfde label te vinden zoals gebruikt in ppp.conf. Als dat mislukt, zoek dan een regel waarin het IP-adres van onze router in voorkomt. Deze regel bevat een IP stijl van 4 octetten. Als nu nog steeds geen corresponderende regel gevonden is wordt er gezocht naar de `HISADDR` regel.

Regel 2

Deze regel verteld `ppp` om een standaard routering toe te voegen die wijst richting `HISADDR`. `HISADDR` wordt vervangen door het IP-adres van de router zoals onderhandeld door IPCP.

Zie de `pmdemand` regel in de bestanden /usr/shared/examples/ppp/ppp.conf.sample en /usr/shared/examples/ppp/ppp.linkup.sample voor een gedetailleerd voorbeeld.

Het ontvangen van binnenkomende gesprekken

Wanneer ppp geconfigureerd is om inkomende gesprekken te ontvangen op een machine die verbonden is met een LAN, moet u beslissen of er pakketten worden doorgestuurd naar het LAN.

Als u dat doet, moet u de andere kant een IP-adres geven uit het subnet van uw LAN, en zult u gebruik moeten maken van het command `enable proxy` in het /etc/ppp/ppp.conf bestand. U zult ook moeten controleren of het /etc/rc.conf bestand het volgende bevat:

```
gateway_enable="YES"
```

Welke getty?

Het configureren van FreeBSD voor inbel diensten levert een goede beschrijving van het inschakelen van inbeldiensten door gebruik te maken van getty(8).

Een alternatief voor `getty` is mgetty (van de port comms/mgetty+sendfax), een betere versie van `getty` ontworpen voor onder andere inbellijnen.

De voordelen van het gebruik van `mgetty` is dat het actief *communiceert* met modems, wat betekent dat als de port uitgeschakeld is in /etc/ttys, het modem de telefoon niet zal beantwoorden.

Latere versies van `mgetty` (vanaf 0.99beta en later) ondersteunen ook het automatisch detecteren van PPP stromen waardoor cliënten zonder extra scripting toegang kunnen krijgen tot uw server.

Raadpleeg naar Mgetty en AutoPPP voor meer informatie over `mgetty`.

PPP Permissies

Het `ppp` commando moet normaal gesproken gestart worden door de `root` gebruiker. Als u echter wilt toestaan dat `ppp` in server mode gestart wordt door een normale gebruiker door het uitvoeren van `ppp`, zoals beschreven hieronder, moet deze gebruiker permissie krijgen om `ppp` te starten. Dit kan gedaam worden door de gebruiker toe te voegen aan de `network` groep van het /etc/group bestand.

U moet de gebruiker ook toegang geven tot één of meerdere secties van het configuratie bestand door gebruik te maken van het `allow` commando:

```
allow users fred mary
```

Als dit commando wordt gebruikt in de `default` sectie, geeft ppp alle opgegeevn gebruikers toegang tot alle opties.

PPP shells voor dynamische IP-gebruikers

Creeër een bestand genaamd /etc/ppp/ppp-shell welke de volgende gegevens bevat:

```
#!/bin/sh
IDENT=`echo $0 | sed -e 's/^.*-\(.*\)$/\1/'`
CALLEDAS="$IDENT"
TTY=`tty`

if [ x$IDENT = xdialup ]; then
```

```
        IDENT=`basename $TTY`
fi

echo "PPP voor $CALLEDAS op $TTY"
echo "Starten van PPP voor $IDENT"

exec /usr/sbin/ppp -direct $IDENT
```

Dit script moet uitvoerbaar zijn. Ook moet er een symbolische link gemaakt worden naar dit script met de naam ppp-dialup door gebruik te maken van de volgende commando's:

```
# ln -s ppp-shell /etc/ppp/ppp-dialup
```

U moet dit script gebruiken als de *shell* voor al uw inbel gebruikers. Dit is een voorbeeld uit /etc/passwd voor een PPP inbelgebruiker met de gebruikersnaam `pchilds` (Let op, u mag niet direct het wachtwoord bestand bewerken, gebruik daarom het programma vipw(8)).

```
pchilds:*:1011:300:Peter Childs PPP:/home/ppp:/etc/ppp/ppp-dialup
```

Creeër vervolgens een map /home/ppp die door iedereen gelezen en beschreven kan worden en zet daar de volgende 0 byte grote bestanden in:

```
-r--r--r--   1 root     wheel           0 May 27 02:23 .hushlogin
-r--r--r--   1 root     wheel           0 May 27 02:22 .rhosts
```

welke voorkomen dat /etc/motd getoond wordt.

PPP shells voor statische IP-gebruikers

Creeër het ppp-shell bestand zoals hierboven, en voor elk account met een statisch toegewezen IP-adres creeërt u een symbolische link naar ppp-shell.

Als u bijvoorbeeld drie inbel gebruikers hebt genaamd `fred`, `sam` en `mary` waar u een /24 CIDR netwerk voor routeert, moet u het volgende typen:

```
# ln -s /etc/ppp/ppp-shell /etc/ppp/ppp-fred
# ln -s /etc/ppp/ppp-shell /etc/ppp/ppp-sam
# ln -s /etc/ppp/ppp-shell /etc/ppp/ppp-mary
```

Elk van deze inbelgebruikers moet de shell ingesteld hebben op de symbolische link die hierboven is gecreeërd (bijvoorbeeld `mary`'s shell moet zijn /etc/ppp/ppp-mary).

Het instellen van ppp.conf voor dynamische IP-gebruikers

Het /etc/ppp/ppp.conf bestand moet iets zoals hieronder bevatten:

```
default:
  set debug phase lcp chat
  set timeout 0

ttyu0:
  set ifaddr 203.14.100.1 203.14.100.20 255.255.255.255
  enable proxy

ttyu1:
  set ifaddr 203.14.100.1 203.14.100.21 255.255.255.255
  enable proxy
```



Het inspringen is belangrijk.

De `default:` sectie wordt altijd geladen. Voor elke inbellijn die ingeschakeld is in /etc/ttys moet een soortgelijke regel worden gemaakt als die voor `ttyu0:` hierboven. Elke regel moet een uniek IP-adres krijgen van uw pool van IP-adressen voor dynamische gebruikers.

Het instellen van ppp.conf voor statische IP-gebruikers.

Samen met de inhoud van het voorbeeld /usr/shared/examples/ppp/ppp.conf bestand hierboven moet een sectie aangemaakt worden voor elke van de statisch ingestelde inbelgebruikers. We gaan door met ons `fred`, `sam` en `mary` voorbeeld.

```
fred:
  set ifaddr 203.14.100.1 203.14.101.1 255.255.255.255

sam:
  set ifaddr 203.14.100.1 203.14.102.1 255.255.255.255

mary:
  set ifaddr 203.14.100.1 203.14.103.1 255.255.255.255
```

Het /etc/ppp/ppp.linkup bestand moet ook informatie over routeringen bevatten voor elke statische IP-gebruiker waar nodig. De regel hieronder voegt een routering toe voor het `203.14.201.0/24` netwerk via de ppp link van de gebruiker.

```
fred:
  add 203.14.101.0 netmask 255.255.255.0 HISADDR

sam:
  add 203.14.102.0 netmask 255.255.255.0 HISADDR

mary:
  add 203.14.103.0 netmask 255.255.255.0 HISADDR
```

`mgetty` en AutoPPP

Standaard staat de optie `AUTO_PPP` in de port comms/mgetty+sendfax welke `mgetty` in staat stelt om de LCP fase van PPP connecties te detecteren en aan de hand daarvan automatisch een ppp shell te creeëren. Echter, de standaard login procedure vindt in deze mode niet plaats, waardoor het nodig is om de gebruikers te authenticeren door middel van PAP of CHAP.

De volgende sectie gaat er vanuit dat u succesvol de port comms/mgetty+sendfax op uw systeem heeft gecompileerd en geïnstalleerd.

Zorg ervoor dat uw /usr/local/etc/mgetty+sendfax/login.config bestand de volgende inhoud heeft:

```
/AutoPPP/ -     -            /etc/ppp/ppp-pap-dialup
```

Dit verteld `mgetty` om het ppp-pap-dialup script te starten wanneer er een PPP connectie gedetecteerd wordt.

Creeër een bestand genaamd /etc/ppp/ppp-pap-dialup met de volgende inhoud (het bestand moet uitvoerbaar zijn):

```
#!/bin/sh
exec /usr/sbin/ppp -direct pap$IDENT
```

Voor elke inbelregel die ingeschakeld is in /etc/ttys, creeër een corresponderende regel in /etc/ppp/ppp.conf. Dit gaat goed samen met de definities die hierboven gedaan zijn.

```
pap:
  enable pap
  set ifaddr 203.14.100.1 203.14.100.20-203.14.100.40
  enable proxy
```

Elke gebruiker die op deze manier inlogt moet een gebruikersnaam en wachtwoord hebben in het /etc/ppp/ppp.secret bestand of de volgende optie moet worden toegevoegd om gebruikers te authenticeren via PAP vanuit het /etc/passwd bestand.

```
enable passwdauth
```

Als u een aantal gebruikers een statisch IP-adres wilt geven, kan dat gespecificeerd worden als het derde argument in /etc/ppp/ppp.secret. Zie /usr/shared/examples/ppp/ppp.secret.sample voor een voorbeeld.

Microsoft® Extensies

Het is mogelijk om PPP dusdanig te configuren dat deze DNS en NetBIOS naamserver adressen meegeeft.

Om deze extensies in te schakelen met PPP versie 1.x kunnen de volgende regels toegevoegd worden aan de relevante sectie in /etc/ppp/ppp.conf:

```
enable msext
set ns 203.14.100.1 203.14.100.2
set nbns 203.14.100.5
```

en voor PPP versie 2 en hoger:

```
accept dns
set dns 203.14.100.1 203.14.100.2
set nbns 203.14.100.5
```

Dit verteld de cliënt het primaire en secundaire naamserver adres, en geeft een NetBIOS naamserver adres.

In versie 2 en hoger zal PPP gebruik maken van de instellingen in /etc/resolv.conf als de regel `set dns` niet wordt gevonden.

PAP en CHAP authenticatie

Sommige providers stellen hun systemen dusdanig in dat het authenticatie gedeelte van uw verbinding wordt afgehandeld door het PAP of CHAP authenticatie mechanisme. Als dit het geval is zal uw provider u niet voorzien van een `login:` prompt wanneer u verbinding maakt maar zal deze meteen gaan communiceren over het PPP protocol.

PAP is minder veilig dan CHAP, maar beveiliging is meestal geen probleem omdat wachtwoorden, ook al worden deze in platte tekst verstuurd met PAP, alleen worden verstuurd via een seriële lijn. Hier is maar weinig ruimte voor crackers om stiekem mee te luisteren.

Terug verwijzende naar de PPP en statische IP-adressen of PPP en dynamische IP-adressen secties moeten de volgende aanpassingen gedaan worden:

```
13      set authname Mijngebruikersnaam
14      set authkey Mijnwachtwoord
15      set login
```

Regel 13

Deze regel geeft uw PPP/CHAP gebruikersnaam aan. U moet de juiste waardes invullen voor *Mijngebruikersnaam*.

Regel 14

Deze regel geeft uw PPP/CHAP wachtwoord aan. U moet de juiste waarde invullen voor *Mijnwachtwoord*. Misschien wilt u een extra regel toevoegen als:

```
16      accept PAP
```

of

```
16      accept CHAP
```

om duidelijk te maken op welke manier u wilt authenticeren, standaard worden zowel PAP als CHAP geaccepteerd.

Regel 15

Uw ISP zal normaal gesprokken niet eisen dat u op de server aanlogt als u gebruik maakt van PAP of CHAP. Daarom moet u de "set login" regel uitschakelen.

Het aanpassen van uw `ppp` configuratie terwijl deze in gebruik is

Het is mogelijk om tegen met het `ppp` programma te communiceren terwijl deze in gebruik is op de achtergrond, maar dat kan alleen als er een geschikte diagnostische poort ingesteld is. Om dit te kunnen doen moet de volgende regel worden toegevoegd aan de configuratie:

```
set server /var/run/ppp-tun%d DiagnosticPassword 0177
```

Dit vertelt PPP om te luisteren naar het gespecificeerde UNIX® domein socket, waarbij de cliënten gevraagd worden om het opgegeven wachtwoord voordat toegang verleend kan worden. Het `%d` in de naam wordt vervangen door het tun apparaat dat gebruikt wordt voor de verbinding.

Zodra een socket ingesteld is kan het pppctl(8) programma gebruikt worden in scripts die het draaiende programma willen bewerken.

De vertaalmogelijkheden van PPP voor netwerkadressen gebruiken

PPP heeft de mogelijkheid om interne NAT te gebruiken zonder dat de kernel hiervoor iets hoeft te doen. Deze functionaliteit kan worden ingeschakeld door de volgende regel in /etc/ppp/ppp.conf:

```
nat enable yes
```

Ook kan PPP NAT ingeschakeld worden door de optie `-nat`. Er is ook een /etc/rc.conf optie genaamd `ppp_nat` welke standaard ingeschakeld is.

Als u gebruik wilt maken van deze optie, kunt u de volgende /etc/ppp/ppp.conf opties ook nuttig vinden om binnenkomende connecties door te sturen:

```
nat port tcp 10.0.0.2:ftp ftp
nat port tcp 10.0.0.2:http http
```

of als u niets vertrouwd vanaf buitenaf:

```
nat deny_incoming yes
```

Laatste systeemconfiguratie

U heeft nu `ppp` geconfigurerd, maar er moeten nog een aantal dingen gedaan worden voordat deze klaar is om te kunnen werken. Hiervoor moeten een aantal aanpassingen gedaan worden in het bestand /etc/rc.conf.

Van boven naar beneden kijkende zorgen we er als eerste voor dat de `hostname=` regel ingesteld is met bijvoorbeeld:

```
hostname="foo.example.com"
```

Als uw provider u een statisch adres en een naam heeft gegeven is het waarschijnlijk handig dat u deze naam gebruikt als uw hostnaam.

Zoek naar de `network_interfaces` variabele. Als u uw systeem wilt configuren om in te bellen bij uw provider wanneer nodig, zorg er dan voor dat het tun0 apparaat is toegevoegd aan deze lijst. Haal deze anders weg.

```
network_interfaces="lo0 tun0"
ifconfig_tun0=
```



De `ifconfig_tun0` variabele moet leeg zijn, en een bestand genaamd /etc/start_if.tun0 moet aangemaakt worden met de volgende inhoud:

```
ppp -auto mysystem
```

Dit script wordt uitgevoerd tijdens de netwerk configuratie, waarbij uw ppp daemon wordt gestart in automatische mode. Als u een LAN heeft waarvoor deze machine een router is wilt u wellicht ook de `-alias` meegeven. Bekijk de handleiding voor verdere details.

Zorg ervoor dat het router programma is ingesteld op `NO` door middel van de volgende regel in uw /etc/rc.conf bestand:

```
router_enable="NO"
```

Het is belangrijk dat de `routed` daemon niet gestart wordt, omdat `routed` de neiging heeft om de standaard routeringtabel regels die gemaakt worden door `ppp` te verwijderen.

Het is waarschijnlijk een goed idee om te zorgen dat de `sendmail_flags` regel de `-q` optie niet wordt

meegenomen, anders zal `sendmail` periodiek een zoek actie verrichten op het netwerk, wat ervoor zorgt dat uw machine gaat uitbellen. U kunt het volgende instellen:

```
sendmail_flags="-bd"
```

Het nadeel hiervan is dat u `sendmail` moet forceren om de mailqueue periodiek te bekijken zodra de ppp link op is door het typen van:

```
# /usr/sbin/sendmail -q
```

U wilt wellicht gebruik maken van het `!bg` commando in ppp.linkup om dit automatisch te doen:

```
1     provider:
2       delete ALL
3       add 0 0 HISADDR
4       !bg sendmail -bd -q30m
```

Als u dit niet wilt doen, is het mogelijk om een "dfiler" in te stellen welke SMTP verkeer blokkeert. Raadpleeg naar de voorbeeld bestanden voor verdere details.

Alles wat nu nog nodig is, is het herstarten van de machine. Na het herstarten kunt het volgende typen:

```
# ppp
```

en daarna `dial provider` om de PPP sessie te starten, of u indien u dat wilt kan `ppp` automatisch sessies opzetten wanneer er uitgaand verkeer is (en wanneer u geen start_if.tun0 script heeft aangemaakt), typt u:

```
# ppp -auto provider
```

Samenvatting

Om samen te vatten zijn de volgende stappen benodigd om PPP voor de eerste keer in te stellen:

Aan de cliënt zijde:

1. Zorg ervoor dat het tun apparaat is ingeschakeld in uw kernel.
2. Zorg ervoor dat het apparaatbestand tunN beschikbaar is in de map /dev.
3. Creeër een regel in /etc/ppp/ppp.conf. Het pmdemand voorbeeld zou moeten volstaand voor de meeste providers.
4. Als u dynamische IP-adressen heeft, creeër een regel in /etc/ppp/ppp.linkup.

5. Update uw /etc/rc.conf bestand.
6. Creeër een start_if.tun0 script als u op verzoek wilt inbellen.

Aan de server zijde:

1. Zorg ervoor dat het tun apparaat is ingeschakeld in uw kernel.
2. Zorg ervoor dat het apparaatbestand tunN beschikbaar is in de map /dev.
3. Creeër een regel in /etc/passwd (door gebruik te maken van het vipw(8) programma).
4. Creeër een profiel in deze gebruikers home directory die `ppp -direct direct-server` start of iets in die trant.
5. Creeër een regel in /etc/ppp/ppp.conf. Het direct-server voorbeeld zou moeten volstaan.
6. Creeër een regel in /etc/ppp/ppp.linkup.
7. Update uw /etc/rc.conf bestand.

25.3. Kernel-PPP gebruiken

25.3.1. Het opzetten van kernel-PPP



Deze sectie geldt en is alleen geldig voor FreeBSD 7.X.

Voordat u begint met het opzetten van PPP op uw machine, zorg ervoor dat het `pppd` commando zich bevindt in de map /usr/sbin en dat de map /etc/ppp bestaat.

`pppd` kan in twee verschillende modes werken:

1. Als een "cliënt" - u wilt uw machine verbinden met de buitenwereld via een seriële PPP-verbinding of een modemlijn.
2. Als een "server" - uw machine bevind zich in het netwerk en wordt gebruikt om andere computers te verbinden door middel van PPP.

In beide gevallen moet u een bestand met opties instellen (/etc/ppp/options of ~/.ppprc als er meer dan één gebruiker is op uw machine die gebruik maakt van PPP).

U heeft ook enige modem/seriële software nodig (comms/kermit wordt aanbevolen), zodat u de andere kant kunt bellen en een verbinding kunt opzetten.

25.3.2. Gebruik maken van `pppd` als cliënt

De volgende /etc/ppp/options kan gebruikt worden om met een Cisco terminal server PPP lijn verbinding te maken.

```
crtscts         # Schakel hardware flow controle in
modem           # modem controle lijn
```

```
noipdefault     # De PPP-server aan de andere kant moet uw IP-adres
                # opgeven, als de machine aan de andere kant uw IP
                # adres niet meegeeft tijdens de IPCP onderhandeling
                # moet deze optie worden verwijderd
passive         # Wacht op LCP pakketten
domain ppp.foo.com      # Vul uw domein naam hier in

:remote_ip    # Vul het IP-adres van de PPP
                # server in deze wordt gebruikt om pakketten te
                # routeren via de PPP link.  Als u de noipdefault optie
                # niet heeft aangegeven verander dan deze regel in
                # local_ip:remote_ip

defaultroute    # Vul dit in als u wilt dat de PPP server de standaard
                # router wordt
```

Om verbinding te maken:

1. Bel naar de machine aan de andere kant door middel van Kermit (of een ander modem programma), en vul uw gebruikersnaam en wachtwoord in (of wat er ook nodig is om de verbinding op te brengen met de machine aan de andere kant).
2. Stop Kermit (zonder de lijn op te hangen).
3. Type het volgende:

   ```
   # /usr/sbin/pppd /dev/tty01 19200
   ```

   Wees er zeker van dat de juiste snelheid en het juiste apparaat wordt aangesproken.

Uw computer is nu verbonden met PPP. Als de connectie faalt, kan de `debug` optie worden meegegeven in het /etc/ppp/options bestand waarna op de console berichten kunnen worden geraadpleegd om het probleem te achterhalen.

Het volgende /etc/ppp/pppup script zal alle drie de stappen automatisch doen:

```
#!/bin/sh
pgrep -l pppd
pid=`pgrep pppd`
if [ "X${pid}" != "X" ] ; then
        echo 'killing pppd, PID=' ${pid}
        kill ${pid}
fi
pgrep -l kermit
pid=`pgrep kermit`
if [ "X${pid}" != "X" ] ; then
        echo 'killing kermit, PID=' ${pid}
        kill -9 ${pid}
```

```
fi

ifconfig ppp0 down
ifconfig ppp0 delete

kermit -y /etc/ppp/kermit.dial
pppd /dev/tty01 19200
```

/etc/ppp/kermit.dial is een Kermit script dat belt en alle benodigde authorisaties doet op de machine aan de andere kant (een voorbeeld van zo'n script is bijgevoegd aan het einde van dit document).

Gebruik het volgende /etc/ppp/pppdown script om de PPP lijn af te breken:

```
#!/bin/sh
pid=`pgrep pppd`
if [ X${pid} != "X" ] ; then
        echo 'killing pppd, PID=' ${pid}
        kill -TERM ${pid}
fi

pgrep -l kermit
pid=`pgrep kermit`
if [ "X${pid}" != "X" ] ; then
        echo 'killing kermit, PID=' ${pid}
        kill -9 ${pid}
fi

/sbin/ifconfig ppp0 down
/sbin/ifconfig ppp0 delete
kermit -y /etc/ppp/kermit.hup
/etc/ppp/ppptest
```

Controleer of `pppd` nog steeds draait door het uitvoeren van /usr/etc/ppp/ppptest, welke er als volgend uitziet:

```
#!/bin/sh
pid=`pgrep pppd`
if [ X${pid} != "X" ] ; then
        echo 'pppd running: PID=' ${pid-NONE}
else
        echo 'No pppd running.'
fi
set -x
netstat -n -I ppp0
ifconfig ppp0
```

Om het modem op te hangen, voer het /etc/ppp/kermit.hup script uit welke het volgende bevat:

```
set line /dev/tty01 ; vul hier uw modem in
set speed 19200
set file type binary
set file names literal
set win 8
set rec pack 1024
set send pack 1024
set block 3
set term bytesize 8
set command bytesize 8
set flow none

pau 1
out +++
inp 5 OK
out ATH0\13
echo \13
exit
```

Hier is een alternatieve methode welke gebruik maakt van `chat` in plaats van `kermit`:

De volgende twee regels zijn voldoende om een `pppd` verbinding op te zetten.

/etc/ppp/options:

```
/dev/cuad1 115200

crtscts         # Schakel hardware flow controle in
modem           # modem controle lijn
connect "/usr/bin/chat -f /etc/ppp/login.chat.script"
noipdefault     # De PPP server aan de andere kant moet uw IP-adres
                # opgeven, als de machine aan de andere kant uw IP
                # adres niet meegeeft tijdens de IPCP onderhandeling
                # moet deze optie worden verwijderd
passive         # Wacht op LCP pakketten
domain your.domain   # Vul uw domein naam hier in

:remote_ip    # Vul het IP-adres van de PPP
                # server in deze wordt gebruikt om pakketten te
                # routeren via de PPP link.  Als u de noipdefault optie
                # niet heeft aangegeven verander dan deze regel in
                # local_ip:remote_ip

defaultroute    # Vul dit in als u wilt dat de PPP server de standaard
                # router wordt
```

/etc/ppp/login.chat.script:



Het volgende moet op één regel.

```
ABORT BUSY ABORT 'NO CARRIER' "" AT OK ATDTtelefoon.nummer
  CONNECT "" TIMEOUT 10 ogin:-\\r-ogin: login-id
  TIMEOUT 5 sword: password
```

Zodra deze zijn geïnstalleerd en correct aangepast is het enige dat gedaan moet worden, het starten van pppd zoals volgt:

```
# pppd
```

25.3.3. Gebruik maken van pppd als server

/etc/ppp/options moet ongeveer het volgende bevatten:

```
crtscts                         # Hardware flow controle
netmask 255.255.255.0           # netmask (niet vereist)
192.114.208.20:192.114.208.165  # IP's van lokale en niet lokale hosten
                                # het lokale IP moet anders zijn dan
                                # degeen die is toegewezen aan de
                                # Ethernet (of andere) interface op uw
                                # machine.  remote IP is het IP-adres
                                # dat wordt toegewezen aan de machine
                                # aan de andere kant
domain ppp.foo.com              # uw domein
passive                         # Wacht op LCP
modem                           # modem lijn
```

Het volgende /etc/ppp/pppserv script zal pppd vertellen zich te gedragen als server:

```
#!/bin/sh
pgrep -l pppd
pid=`pgrep pppd`
if [ "X${pid}" != "X" ] ; then
        echo 'killing pppd, PID=' ${pid}
        kill ${pid}
fi
pgrep -l kermit
pid=`pgrep kermit`
if [ "X${pid}" != "X" ] ; then
        echo 'killing kermit, PID=' ${pid}
        kill -9 ${pid}
fi

# reset ppp interface
ifconfig ppp0 down
```

```
ifconfig ppp0 delete

# enable autoanswer mode
kermit -y /etc/ppp/kermit.ans

# run ppp
pppd /dev/tty01 19200
```

Gebruik dit /etc/ppp/pppservdown script om de server te stoppen:

```
#!/bin/sh
pgrep -l pppd
pid=`pgrep pppd`
if [ "X${pid}" != "X" ] ; then
        echo 'killing pppd, PID=' ${pid}
        kill ${pid}
fi
pgrep -l kermit
pid=`pgrep kermit`
if [ "X${pid}" != "X" ] ; then
        echo 'killing kermit, PID=' ${pid}
        kill -9 ${pid}
fi
ifconfig ppp0 down
ifconfig ppp0 delete

kermit -y /etc/ppp/kermit.noans
```

Het volgende Kermit script (/etc/ppp/kermit.ans) zal het automatisch beantwoorden van uw modem in of uitschakelen. Het moet eruit zien als volgend:

```
set line /dev/tty01
set speed 19200
set file type binary
set file names literal
set win 8
set rec pack 1024
set send pack 1024
set block 3
set term bytesize 8
set command bytesize 8
set flow none

pau 1
out +++
inp 5 OK
out ATH0\13
inp 5 OK
echo \13
```

```
out ATS0=1\13   ; Verander dit in out ATS0=0\13 als u automatisch
                ; beantwoorden wilt uitschakelen
inp 5 OK
echo \13
exit
```

Een script genaamd /etc/ppp/kermit.dial wordt gebruikt voor het bellen en authenticeren van de machine aan de andere kant. U moet deze aanpassen aan uw wensen. Vul in dit script uw inlognaam en wachtwoord in, u moet ook het input statement aanpassen afhankelijk hoe uw modem antwoordt en de communicatie van de machine aan de andere kant.

```
;
; Vul de seriële lijn in welke verbonden is met het modem
;
set line /dev/tty01
;
; Stel het modem snelheid in:
;
set speed 19200
set file type binary            ; volledige 8 bit bestands xfer
set file names literal
set win 8
set rec pack 1024
set send pack 1024
set block 3
set term bytesize 8
set command bytesize 8
set flow none
set modem hayes
set dial hangup off
set carrier auto                ; Daarna stel SET CARRIER in indien nodig
set dial display on             ; Stel daarna SET DIAL in indien nodig
set input echo on
set input timeout proceed
set input case ignore
def \%x 0                       ; login prompt teller
goto slhup

:slcmd                          ; stel het modem in op commandomodus
echo Stel het modem in op commandomodus.
clear                           ; Verwijder ongelezen karakters uit de input buffer
pause 1
output +++                      ; hayes escape sequence
input 1 OK\13\10                ; wacht op OK
if success goto slhup
output \13
pause 1
output at\13
input 1 OK\13\10
if fail goto slcmd              ; Als het modem niet antwoordt met OK, probeer het
```

```
opnieuw

:slhup                          ; hang de telefoon op
clear                           ; Verwijder ongelezen karakters uit de input buffer
pause 1
echo De telefoon wordt opgehangen.
output ath0\13                  ; hayes command voo on hook
input 2 OK\13\10
if fail goto slcmd              ; Als er geen OK antwoord is, stel het modem in op
commandomodus

:sldial                         ; Draai het nummer
pause 1
echo Bellen.
output atdt9,550311\13\10               ; put phone number here
assign \%x 0                    ; zero the time counter

:look
clear                           ; Verwijder ongelezen karakters uit de input buffer
increment \%x                   ; Tel de seconden
input 1 {CONNECT }
if success goto sllogin
reinput 1 {NO CARRIER\13\10}
if success goto sldial
reinput 1 {NO DIALTONE\13\10}
if success goto slnodial
reinput 1 {\255}
if success goto slhup
reinput 1 {\127}
if success goto slhup
if < \%x 60 goto look
else goto slhup

:sllogin                        ; login
assign \%x 0                    ; Stel de tijd teller in op nul
pause 1
echo Zoeken naar de login prompt

:slloop
increment \%x                   ; Tel de seconden
clear                           ; Verwijder ongelezen karakters uit de input buffer
output \13
;
; Stel hier de verwachte login prompt in:
;
input 1 {Username: }
if success goto sluid
reinput 1 {\255}
if success goto slhup
reinput 1 {\127}
if success goto slhup
```

```
if < \%x 10 goto slloop         ; Probeer 10 x om een login prompt te krijgen
else goto slhup                 ; Hang op en probeer het nogmaals als er 10 mislukte
pogingen zijn

:sluid
;
; Vul hier uw gebruikersnaam in:
;
output ppp-login\13
input 1 {Password: }
;
; Vul hier uw wachtwoord in:
;
output ppp-password\13
input 1 {Entering SLIP mode.}
echo
quit

:slnodial
echo \7Er is geen kiestoon, controleer de telefoon lijn!\7
exit 1

; local variables:
; mode: csh
; comment-start: "; "
; comment-start-skip: "; "
; end:
```

25.4. Het problemen oplossen van PPP-verbindingen

Deze sectie behandelt een paar problemen die kunnen optreden wanneer PPP wordt gebruikt over een modemverbinding. Bijvoorbeeld, misschien moet u exact weten wat de prompt is die het systeem waarop u inbelt presenteert. Sommige providers presenteren de `ssword` prompt terwijl anderen `password` tonen als het `ppp` script niet goed geschreven is en de inlogin poging faalt. De meest standaard manier om `ppp` verbindingen te onderzoeken op problemen is door handmatig een connectie op te zetten. De volgende informatie helpt u om stap voor stap een handmatige connectie op te zetten.

25.4.1. Controleer de apparaatknooppunten

Als er een eigen kernel gebruikt wordt, vergeet dan niet om de volgende regel in uw kernelinstellingenbestand op te nemen:

```
device  uart
```

Het apparaat uart is al in de kernel `GENERIC` opgenomen, dus zijn er in dit geval geen extra stappen nodig. Controleer de resultaten van het commando `dmesg` voor het modemapparaat door middel van:

```
# dmesg | grep uart
```

U zou enige informatie moeten ontvangen over de uart apparaten. Deze bevinden zich op de COM-poorten die we nodig hebben. Als uw modem zich gedraagt als een standaard seriële poort zou u deze moeten vinden als zijnde uart1 of COM2. Als dat klopt hoeft u de kernel niet opnieuw te bouwen. Wanneer u de sio-apparaten controleert en het modem is op uart1 te vinden of als COM2 als u zich onder MS-DOS® bevindt, dan is uw modemapparaat /dev/cuau1.

25.4.2. Handmatig verbinding maken

Verbinding maken met het internet door handmatig controle te hebben over ppp is snel, makkelijk en een geweldige manier om problemen te vinden bij een verbinding of zelfs voor alleen het verkrijgen van inforamtie over hoe uw provider de ppp cliënt verbindingen behandelt. Laten we starten met PPP vanaf de commando regel. Let op dat in al onze voorbeelden we gebruik maken van *example* als hostnaam van de machine die PPP draait. U start ppp door enkel het commando ppp te typen:

```
# ppp
```

We hebben nu ppp gestart.

```
ppp ON example> set device /dev/cuau1
```

We stellen ons modem in, in dit geval is dat cuau1.

```
ppp ON example> set speed 115200
```

We stellen de verbindings snelheid in, in dit geval gebruiken we 115,200 kbps.

```
ppp ON example> enable dns
```

Vertel ppp om onze naam vertaler te configuren, en de juiste naamserver regels toe te voegen aan /etc/resolv.conf. Als ppp onze hostnaam niet kan bepalen, kunnen we deze later instellen.

```
ppp ON example> term
```

Wissel naar "terminal" mode zodat we handmatig het modem kunnen bedienen.

```
deflink: Entering terminal mode on /dev/cuau1
type '~h' for help
```

```
at
OK
atdt123456789
```

Gebruik `at` om het modem te initialiseren, en daarna `atdt` en het nummer voor uw provider om het inbel proces te beginnen.

```
CONNECT
```

Bevestiging van de verbinding, als we tegen problemen aanlopen met de verbinding, welke niet gerelateerd zijn aan de hardware, is dit de plek om te beginnen om de problemen op te lossen.

```
provider login:myusername
```

Hier wordt u gevraagd om een gebruikersnaam. Geef de gebruikersnaam op welke aangeleverd is door de provider.

```
provider pass:mypassword
```

Deze keer worden we gevraagd voor een wachtwoord. Vul uw wachtwoord in welke u is aangeleverd door de provider. Net zoals het aanloggen op FreeBSD zal het wachtwoord niet getoond worden.

```
Shell or PPP:ppp
```

Afhankelijk van uw provider wordt deze prompt wellicht nooit getoond. Hier wordt ons gevraagd of we een shell willen starten op de host van de provider, of dat we `ppp` willen starten. In dit geval is er gekozen voor `ppp` omdat we een internet verbinding willen.

```
Ppp ON example>
```

Let op dat in dit voorbeeld de eerste `p` een hoofdletter geworden is. Dit geeft aan dat we succesvol verbonden zijn met de provider.

```
PPp ON example>
```

We hebben ons succesvol geauthenticeerd bij onze provider en we wachten op een IP-adres dat ons wordt toegewezen.

```
PPP ON example>
```

We hebben een IP adres verkregen en hebben succesvol een verbinding opgebouwd.

```
PPP ON example>add default HISADDR
```

Hier wordt een standaard route toegevoegd. Deze moet worden toegevoegd voordat we kunnen communiceren met de buitenwereld aangezien de enige verbinding op dit moment met de andere machine is. Als dit niet lukt omdat er al een route bestaat, kan er een "bang" karakter (`!`) geplaatst worden voor de `add` optie. Als alternatief kan dit ook gedaan worden voordat de verbinding opgezet wordt, waarna een nieuwe route onderhandeld wordt.

Als alles goed gegaan is, zou er nu een actieve verbinding moeten zijn met het internet, welke in de achtergrond gezet kan worden door `CTRL` + `z` te gebruiken. Als u ziet dat het commando `PPP` terugkeert naar `ppp` is de verbinding afgebroken. Dit is goed om te weten, aangezien dit de status van de verbinding toont. Hoofdletter P's betekenen dat er een verbinding is met de provider, en kleine letters betekend dat de verbinding verloren is gegaan om welke reden dan ook. `ppp` kent alleen deze twee statussen.

Debuggen

Als u een directe lijn heeft en geen verbinding kan maken, zet dan hardware flow CTS/RTS uit met de `set ctsrts off` optie. Dit is meestal het geval voor een PPP terminal server waar PPP hangt wanneer deze probeert te schrijven naar uw communicatie link, dus moet deze wachten op een CTS of een Clear To Send signaal welke misschien nooit komt. Als u deze optie gebruikt, moet u ook de `set accmap` optie gebruiken welke benodigd kan zijn om hardware afhankelijkheden te omzeilen door bepaalde karakters over en weer te sturen, meestal XON/XOFF. Zie de ppp(8) handleiding voor meer informatie over deze optie en hoe deze gebruikt kan worden.

Als u een ouder modem heeft, kan het voorkomen dat u ook de `set parity even` optie moet gebruiken. De parity is standaard ingesteld op none maar wordt gebruikt voor fout controle (met als gevolg een grote verhoging van de hoeveelheid data) bij oudere modems en sommige providers. Dit is bijvoorbeeld een benodigde optie bij de Compuserve provider.

Het kan voorkomen dat PPP niet terugkeert naar de commando mode, wat meestal betekent dat er een onderhandelings fout is waarbij de provider wacht op uw kant om de onderhandeling te kunnen beginnen. Op dit moment kunt u gebruik maken van het `~p` commando om ppp te forceren om de configuratie informatie te versturen.

Als u nooit een inlogin prompt krijgt is het zeer waarschijnlijk dat u PAP of CHAP authenticatie moet gebruiken in plaats van de UNIX® stijl in het voorbeeld hierboven. Om gebruik te maken van PAP of CHAP voegt u het volgende opties toe aan PPP voordat u de terminal mode ingaat:

```
ppp ON example> set authname mijngebruikersnaam
```

Waarbij *mijngebruikersnaam* moet worden vervangen met de gebruikersnaam die wordt toegewezen door de provider.

```
ppp ON example> set authkey mijnwachtwoord
```

Waarbij *mijnwachtwoord* moet worden vervangen door het wachtwoord wat u is toegewezen door de provider.

Als u een goed werkende verbinding kunt maken maar het onmogelijk lijkt om een domeinnaam te vinden, probeert u dan ping(8) te gebruiken met een IP adres en kijk of er enige informatie terugkomt. Als u 100 procent (100%) packet loss ziet is het zeer waarschijnlijk dat u geen default route heeft gekregen. Controleer nogmaals of de optie `add default HISADDR` ingesteld is tijdens de connectie. Als u verbinding kunt maken met een extern IP adres is het mogelijk dat een naamserver niet is toegevoegd aan het /etc/resolv.conf bestand. Dit bestand moet er uitzien als volgend:

```
domain example.com
nameserver x.x.x.x
nameserver y.y.y.y
```

Waar *x.x.x.x* en *y.y.y.y* moet worden vervangen door het IP adres van uw providers naamservers. Deze informatie kan mogelijk wel of niet geleverd zijn toen u zich inschreef, maar een snel telefoontje naar uw provider zou hierin uitkomst kunnen bieden.

U kunt ook syslog(3) gebruiken om een log functie voor PPP aan te maken. Voeg het volgende toe aan /etc/syslog.conf:

```
!ppp
*.*     /var/log/ppp.log
```

Deze functionaliteit bestaat in de meeste gevallen al.

25.5. PPP gebruiken over Ethernet (PPPoE)

Deze sectie beschrijft hoe PPP over Ethernet opgezet kan worden (PPPoE).

25.5.1. Het configureren van de kernel

Inmiddels is het niet langer benodigd om de kernel configuratie aan te passen voor PPPoE. Als de benodigde netgraph ondersteuning niet in de kernel aanwezig is zal deze dynamisch geladen worden door ppp.

25.5.2. Het instellen van ppp.conf

Hieronder volgt een voorbeeld van een werkende ppp.conf:

```
default:
  set log Phase tun command # Er kan meer gedetailleerde logging ingeschakeld worden
indien gewenst
```

```
  set ifaddr 10.0.0.1/0 10.0.0.2/0

name_of_service_provider:
  set device PPPoE:xl1 # Vervang x11 met uw ethernet apparaat
  set authname UWLOGINNAAM
  set authkey UWWACHTWOORD
  set dial
  set login
  add default HISADDR
```

25.5.3. Het draaien van ppp

Als de `root` gebruiker kunt u het volgende draaien:

```
# ppp -ddial name_of_service_provider
```

25.5.4. Het pstarten van ppp tijdens het opstarten

Voeg het volgende toe aan uw /etc/rc.conf bestand:

```
ppp_enable="YES"
ppp_mode="ddial"
ppp_nat="YES"   # Indien u nat wilt inschakelen voor het lokale netwerk, gebruik
anders NO
ppp_profile="naam_van_service_provider"
```

25.5.5. Gebruik maken van een PPPoE service label

Soms is het nodig om een service tag te gebruiken om verbinding te kunnen maken. Service tags worden gebruikt om onderscheid te maken tussen de verschillende PPPoE servers die verbonden zijn met een netwerk.

Uw provider zou u de juiste service tag gegevens verstrekt moeten hebben in de documentatie die opgeleverd is. Als u deze niet kunt vinden in de documentatie moet u deze opvragen bij uw technische support afdeling van uw provider.

Als allerlaatste optie kunt u de aangerade methode gebruiken van het Roaring Penguin PPPoE programma welke gevonden kan worden in de Ports Collectie. Houd u echter in uw achterhoofd dat dit uw modem ernstige schade kan toebrengen, dus denkt u er goed over na voordat u het uitprobeert. Installeer simpelweg het programma dat is meegeleverd bij het modem door uw provider. Open hierna het **System** menu vanuit het programma. De naam van uw profiel moet hier te vinden zijn. Meestal is deze *ISP*.

De naam van het profiel (servicetag) zal worden gebruikt in de PPPoE configuratie regel van ppp.conf in het provider gedeelte van het `set device` commando (zie de ppp(8) handleiding voor meer informatie hierover). Dit zou er als volgend uit moeten zien:

```
set device PPPoE:xl1:ISP
```

Vergeet u niet om *xl1* te vervangen door het juiste apparaat voor uw Ethernet kaart.

Vergeet u niet om *ISP* te vergangen door het profiel wat hierboven ingesteld is.

Voor meer informatie zie:

- Cheaper Broadband with FreeBSD on DSL door Renaud Waldura.

25.5.6. PPPoE met een 3Com® HomeConnect™ ADSL Modem Dual Link

Dit modem volgt RFC 2516 niet (*Een methode voor het versturen van PPP over Ethernet (PPPoE)* geschreven door by L. Mamakos, K. Lidl, J. Evarts, D. Carrel, D. Simone, and R. Wheeler). Daarentegen is een ander type pakket code gebruikt voor de Ethernet frames. Klaagt u alstublift bij 3Com als u vind dat ze zich aan de PPPoE specificatie moeten houden.

Om FreeBSD in staat te stellen om te communiceren met dit apparaat, moet er een sysctl ingesteld worden. Dit kan automatisch tijdens het opstarten gedaan worden door het bewerken van /etc/sysctl.conf:

```
net.graph.nonstandard_pppoe=1
```

Dit kan ook direct gedaan worden met het commando:

```
# sysctl net.graph.nonstandard_pppoe=1
```

Helaas is het, doordat dit een systeem brede instelling is, niet mogelijk om tegelijkertijd met een normale PPPoE cliënt of server en een 3Com® HomeConnect™ ADSL-modem te communiceren.

25.6. Gebruik maken van PPP over ATM (PPPoA)

Het volgende beschrijft hoe PPP over ATM (PPPoA) opgezet kan worden. PPPoA is een populaire keuze binnen Europese DSL providers.

25.6.1. Gebruik maken van PPPoA met de Alcatel SpeedTouch™ USB

PPPoA ondersteuning voor dit apparaat wordt geleverd door middel van een port in FreeBSD omdat de firmware wordt gedistribueerd onder Alcatel's licentie overeenkomst en mag derhalve niet vrijelijk verspreid worden met het basis systeem van FreeBSD.

Om de software te installeren, wordt simpelweg de Ports Collectie gebruikt. Installeer de net/pppoa port en volg de instructies die meegeleverd worden.

Zoals de meeste USB apparaten moet de Alcatel SpeedTouch™ USB zijn firmware downloaden van de host computer om correct te kunnen werken. Het is mogelijk om dit proces te automatiseren

binnen FreeBSD zodat deze overdracht elke keer gebeurd als het apparaat in een USB poort wordt gestoken. De volgende informatie kan worden toegevoegd aan het /etc/usbd.conf bestand om deze automatische overdracht in te schakelen. Dit bestand moet bewerkt worden door de `root` gebruiker.

```
device "Alcatel SpeedTouch USB"
    devname "ugen[0-9]+"
    vendor 0x06b9
    product 0x4061
    attach "/usr/local/sbin/modem_run -f /usr/local/libdata/mgmt.o"
```

Om de USB daemon, usbd, te starten moet de volgende regel toegevoegd worden aan /etc/rc.conf:

```
usbd_enable="YES"
```

Het is ook mogelijk om ppp op te zetten om in te bellen tijdens het opstarten. Om dit te doen moet de volgende regel worden toegevoegd aan /etc/rc.conf. Voor deze procedure moet er ook aangelogt zijn als de `root` gebruiker.

```
ppp_enable="YES"
ppp_mode="ddial"
ppp_profile="adsl"
```

Om dit correct te laten werken moet het voorbeeld ppp.conf bestand gebruikt worden welke geleverd wordt door de net/pppoa port.

25.6.2. Gebruik maken van mpd

U kunt mpd gebruiken om met een variateit aan diensten verbinding te maken, in het bijzonder PPTP diensten. U kunt mpd vinden in de Ports Collectie, net/mpd. Veel ADSL-modems vereisen dat er een PPTP tunnel wordt gecreeërd tussen het modem en de computer, een voorbeeld van zo'n modem is de Alcatel SpeedTouch™ Home.

Eerst moet u de port installeren waarna mpd geconfigureerd kan worden om uw eisen en provider instellingen op te geven. De port plaatst een verzameling voorbeeldconfiguratiebestanden welke goed gedocumenteerd zijn in PREFIX/etc/mpd/. Let op dat *PREFIX* betekend dat dit de directory is waar uw ports in worden geïnstalleerd. Standaard is dit /usr/local/. Een complete handleiding om mpd te configureren is beschikbaar in HTML formaat zodra de port geïnstalleerd is. Deze wordt geplaatst in PREFIX/shared/doc/mpd/. Hieronder staat een voorbeeld configuratie om verbinding te maken met een ADSL dienstverlener door het gebruik van mpd. De configuratie is verspreid over twee bestanden, allereerst het mpd.conf bestand:



Dit voorbeeld van het bestand mpd.conf werkt alleen met mpd 4.X.

```
default:
    load adsl
```

```
adsl:
    new -i ng0 adsl adsl
    set bundle authname gebruikersnaam ①
    set bundle password wachtwoord ②
    set bundle disable multilink

    set link no pap acfcomp protocomp
    set link disable chap
    set link accept chap
    set link keep-alive 30 10

    set ipcp no vjcomp
    set ipcp ranges 0.0.0.0/0 0.0.0.0/0

    set iface route default
    set iface disable on-demand
    set iface enable proxy-arp
    set iface idle 0

    open
```

① De gebruikersnaam die gebruikt wordt om uzelf te authenticeren aan uw provider.

② Het wachtwoord wat gebruikt wordt om uzelf te authenticeren aan uw provider.

Het mpd.links bestand bevat informatie over de link, of linken waarmee u verbinding wilt maken. Een voorbeeld mpd.links wat bij bovenstaand voorbeeld hoort is hieronder gegeven:

```
adsl:
    set link type pptp
    set pptp mode active
    set pptp enable originate outcall
    set pptp self 10.0.0.1 ①
    set pptp peer 10.0.0.138 ②
```

① Het IP-adres van uw FreeBSD computer waar vanaf mpd gebruikt wordt.

② Het IP-adres van uw ADSL-modem. Voor de Alcatel SpeedTouch™ Home is dit adres standaard `10.0.0.138`.

Het is mogelijk om de verbinding makkelijk te initialiseren door het volgende commando als `root` uit te voeren:

```
# mpd -b adsl
```

U kunt de status van de verbinding zien met het volgende commando:

```
% ifconfig ng0
```

```
ng0: flags=88d1<UP,POINTOPOINT,RUNNING,NOARP,SIMPLEX,MULTICAST> mtu 1500
     inet 216.136.204.117 --> 204.152.186.171 netmask 0xffffffff
```

Het gebruik van mpd is de aangeraden manier om met een ADSL dienst te verbinden met FreeBSD.

25.6.3. Gebruik maken van pptpclient

Het is ook mogelijk om FreeBSD te gebruiken om naar een andere PPPoA dienstenm verbinding te maken door middel van de net/pptpclient port.

Om gebruik te maken van net/pptpclient om naar een DSL dienst verbinding te maken, installeert u de port of package en bewerkt u /etc/ppp/ppp.conf U moet dit onder de `root` gebruiker doen, om beide acties uit te voeren. Een voorbeeld sectie van ppp.conf is hieronder gegeven. Voor meer informatie over ppp.conf consulteert u de ppp(8) handleiding.

```
adsl:
 set log phase chat lcp ipcp ccp tun command
 set timeout 0
 enable dns
 set authname gebruikersnaam ①
 set authkey wachtwoord ②
 set ifaddr 0 0
 add default HISADDR
```

① De gebruikersnaam van uw account bij uw DSL provider.

② Het wachtwoord voor uw account.



Omdat u het wachtwoord van uw account in het ppp.conf bestand in leesbare vorm moet plaatsen, moet u ervoor zorgen dat niemand anders de inhoud van dit bestand kan lezen. De volgende serie van commando's zorgt ervoor dat het bestand alleen leesbaar is door de `root` gebruiker. Raadpleeg de handleidingen van chmod(1) en chown(8) voor verdere informatie.

```
# chown root:wheel /etc/ppp/ppp.conf
# chmod 600 /etc/ppp/ppp.conf
```

Dit opent een tunnel voor een PPP sessie naar uw DSL router. Ethernet DSL-modems hebben een voor geconfigureerd LAN IP adres waarmee u verbinding maakt. In het geval van de Alcatel SpeedTouch™ home is `10.0.0.138` het adres. Uw router documentatie vertelt u welk adres uw apparaat gebruikt. Om de tunnel te openen en om een PPP sessie op te zetten, start u het volgende commando:

```
# pptp address adsl
```



Het kan wenselijk zijn om een ampersand ("&") toe te voegen aan het einde van

het vorige commando, omdat pptp uw prompt niet teruggeeft.

Er wordt een tun virtueel tunnel apparaat gecreeërd voor interactie tussen de pptp en ppp processen. Zodra u terugbent op uw prompt, of als pptp bevestigd dat er een verbinding is, kunt u de tunnel als volgend inzien:

```
% ifconfig tun0
tun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1500
        inet 216.136.204.21 --> 204.152.186.171 netmask 0xffffff00
        Opened by PID 918
```

Als het niet mogelijk is om verbinding te maken, controleert u de configuratie van uw router, welke meestal bereikbaar is door middel van telnet of via een web browser. Als u nog steeds geen verbinding kunt maken moet u de resultaten van het `pptp` onderzoeken en de inhoud van het ppp log bestand, /var/log/ppp.log voor meer hints over wat er mis kan zijn.

25.7. Gebruik maken van SLIP



Deze sectie geldt en is alleen geldig voor FreeBSD 7.X.

25.7.1. Het opzetten van een SLIP-cliënt

Het volgende is één manier om een FreeBSD machine in te stellen voor gebruik met SLIP op een statisch host netwerk. Voor dynamische hostnaam toewijzing (uw adres veranderd elke keer als u inbelt), heeft u waarschijnlijk een meer complexe opzet nodig.

Bepaal eerst aan welke seriële poort uw modem verbonden is. Veel mensen gebruiken hiervoor een symbolische link zoals /dev/modem welke verwijst naar de echte naam van het apparaat /dev/cuadN. Dit geeft de mogelijkheid om naam abstract te houden, voor het geval het modem ooit verplaatst wordt naar een andere poort. Het kan best een vervelende klus zijn wanneer er een aantal bestanden in /etc en .kermrc bestanden verspreid over het gehele systeem gerepareerd moeten worden!



/dev/cuad0 is COM1, cuad1 is COM2, etc.

Zorg ervoor dat u het volgende in uw kernel configuratie bestand hebt:

```
device   sl
```

Deze is standaard opgenomen in de GENERIC kernel, dus dat zou geen problemen moeten opleveren tenzij u deze verwijderd heeft.

Dingen die u maar eenmalig hoeft uit te voeren

1. Voeg uw machine, de router en de naamservers toe aan uw /etc/hosts bestand. Ons bestand

ziet er als volgend uit:

```
127.0.0.1               localhost loghost
136.152.64.181          water.CS.Example.EDU water.CS water
136.152.64.1            inr-3.CS.Example.EDU inr-3 slip-gateway
128.32.136.9            ns1.Example.EDU ns1
128.32.136.12           ns2.Example.EDU ns2
```

2. Zorg ervoor dat u `files` voor `dns` in de `hosts:` sectie van uw /etc/nsswitch.conf bestand. Zonder deze parameters zouden er interessante dingen kunnen gebeuren.
3. Bewerk het /etc/rc.conf bestand.
   a. Stel uw hostnaam in door de regel te bewerken die aangeeft:

   ```
   hostname="myname.my.domain"
   ```

   De volledig gekwalificeerde internet hostnaam moet hier geplaatst worden.

   b. Stel de standaard router in door het aanpassen van de volgende regel van:

   ```
   defaultrouter="NO"
   ```

   naar:

   ```
   defaultrouter="slip-gateway"
   ```

4. Creeër en bestand genaamd /etc/resolv.conf welke het volgende bevat:

   ```
   domain CS.Example.EDU
   nameserver 128.32.136.9
   nameserver 128.32.136.12
   ```

   Zoals u kunt zien, stellen deze de naamserver hosten in. Uiteraard is het echte domein en adres afhankelijk van uw omgeving.

5. Stel het wachtwoord in voor de `root` en de `toor` gebruikers (en elke andere gebruiker die geen wachtwoord heeft).
6. Herstart de machine en controleeer of deze opkomt met de correcte hostnaam.

Het opzetten van een SLIP-verbinding

1. Bel in, type `slip` op de prompt en voer uw machine naam en wachtwoord in. Wat is vereist, is afhankelijk van uw omgeving. Als u gebruik maakt van Kermit kan een script als de

volgende gebruikt worden:

```
# kermit setup
set modem hayes
set line /dev/modem
set speed 115200
set parity none
set flow rts/cts
set terminal bytesize 8
set file type binary
# De volgende macro zal inbellen en ons inloggen.
define slip dial 643-9600, input 10 =>, if failure stop, -
output slip\x0d, input 10 Username:, if failure stop, -
output silvia\x0d, input 10 Password:, if failure stop, -
output ***\x0d, echo \x0aCONNECTED\x0a
```

Uiteraard moet u uw gebruikersnaam en wachtwoord wijzigen zodat deze overeenkomen met die van u. Nadat dit gedaan is kunt u `slip` invullen op de Kermit prompt om verbinding te maken.



Het achterlaten van uw wachtwoord in leesbare tekst waar dan ook op het bestandssysteem is zeker een *slecht* idee. Doe dit op eigen risico.

2. Laat Kermit daar (het programma kan tijdelijk uitgeschakeld worden door `Ctrl` + `z`) en type vervolgens als `root`:

```
# slattach -h -c -s 115200 /dev/modem
```

Als u in staat bent om andere hosten met `ping` te benaderen aan de andere kant van de router, bent u verbonden! Als dit niet werkt kunt u wellicht de `-a` gebruiken in plaats van de `-c` als argument voor `slattach`.

Hoe de verbinding afgebroken moet worden

Doe het volgende

```
# kill -INT `cat /var/run/slattach.modem.pid`
```

om `slattach` te stoppen. Houd in uw achterhoofd dat u dit als `root` moet doen. Ga hierna terug naar `kermit` (door het intypen van `fg` als u deze tijdelijk uitgeschakeld had) en verlaat de applicatie (`q`).

De slattach(8) handleiding zegt dat `ifconfig sl0 down` uitgevoerd moet worden om de interface uit te schakelen, maar dit lijkt geen verschil op te leveren. (`ifconfig sl0` lijkt hetzelfde resultaat te geven.)

Soms kan het gebeuren dat het modem weigert om de carrier los te laten. Start in dat geval

simpelweg `kermit` en stop deze wederom. Meestal stopt het met de tweede poging.

Problemen oplossen

Als dit niet werkt, voelt u zich dan vrij om rond te vragen op de freebsd-net mailing lijst. De volgende dingen zijn mensen al eens tegen aangelopen:

- Het niet gebruiken van de `-c` of `-a` optie voor `slattach` (Dit zou niet fataal moeten zijn, maar sommige mensen hebben aangegeven dat het de problemen oplostte.)
- Het intypen van `s10` in plaats van `sl0` (het verschil is wellicht lastig te zien met sommige fonts).
- Probeer `ifconfig sl0` uit te voeren om de interface status te zien. U kunt bijvoorbeeld krijgen:

```
# ifconfig sl0
sl0: flags=10<POINTOPOINT>
        inet 136.152.64.181 --> 136.152.64.1 netmask ffffff00
```

- Als u `no route to host` krijgt van het ping(8) commando, is er mogelijk een probleem met uw route tabel. U kunt het `netstat -r` commando uitvoeren om de huidige routes te zien:

```
# netstat -r
Routing tables
Destination      Gateway            Flags     Refs     Use  IfaceMTU    Rtt
Netmasks:

(root node)
(root node)

Route Tree for Protocol Family inet:
(root node) =>
default          inr-3.Example.EDU  UG          8   224515  sl0 -      -
localhost.Exampl localhost.Example. UH          5    42127  lo0 -       0.438
inr-3.Example.ED water.CS.Example.E UH          1        0  sl0 -      -
water.CS.Example localhost.Example. UGH        34 47641234  lo0 -       0.438
(root node)
```

  Het voorgaand voorbeeld komt van een relatief druk systeem. De getallen op uw systeem zullen anders zijn naar gelang de netwerk activiteiten.

25.7.2. Het opzetten van een SLIP-server

Dit document levert suggesties voor het opzetten van een SLIP-server op een FreeBSD systeem, welke meestal betekent het configureren van uw systeem om automatisch verbindingen op te zetten wanneer er wordt ingelogt met remote SLIP cliënten.

Eisen vooraf

Deze sectie is vrij technisch van aard, dus achtergrond informatie is vereist. Er wordt aangenomen

dat u bekend bent met het TCP/IP-netwerk protocol, en in dan in het bijzonder met netwerk en node adresseringen, netwerk adres maskers, subnetten, routes en dynamische routering protocollen zoals RIP. Het configureren van een SLIP-dienst op een inbel server vereist kennis van deze concepten en als u daarmee niet bekend bent, leest u dan aub een versie van of Craig Hunt's *TCP/IP Network Administration* gepubliceerd door O'Reilly & Associates, Inc. (ISBN Number 0-937175-82-X), of Douglas Comer's boeken over het TCP/IP protocol.

Daarnaast wordt er vanuit gegaan dat u reeds uw modem(s) heeft geconfigureerd en dat u de juiste systeem bestanden heeft aangepast zodat er logins mogelijk zijn door uw modem(s) heen. Als u dat nog niet heeft gedaan, zie dan Inbeldienst voor details over het opzetten van inbel diensten. Wellicht wilt u ook de handleiding bekijken voor sio(4) voor meer informatie over de seriële port device driver en de ttys(5), gettytab(5), getty(8), & init(8) handleidingen voor informatie die relevant zijn voor het configureren van het systeem zodat logins mogelijk worden op modems, en wellicht stty(1) voor informatie over het instellen van de seriële poort (zoals `clocal` voor direct verbonden seriële interfaces).

Snel overzicht

In een typische configuratie, werkt het gebruik van FreeBSD als een SLIP-server als volgend: een SLIP-gebruiker belt in op uw FreeBSD SLIP-server systeem en logt in met een speciaal SLIP-login ID dat gebruik maakt van /usr/sbin/sliplogin. Het `sliplogin` programma leest door het /etc/sliphome/slip.hosts bestand om een corresponderende regel te vinden voor de speciale gebruiker en als deze een match vindt verbind het de seriële lijn met een beschikbare SLIP-interface waarna het shellscript /etc/sliphome/slip.login wordt uitgevoerd om de SLIP-interface te configureren.

Een voorbeeld van SLIP-server login

Bijvoorbeeld, als een SLIP-user-ID `Shelmerg` is, kan `Shelmerg`'s regel in /etc/master.passwd er als volgt uitzien:

```
Shelmerg:password:1964:89::0:0:Guy Helmer -
SLIP:/usr/users/Shelmerg:/usr/sbin/sliplogin
```

Wanneer `Shelmerg` inlogt, zoekt het `sliplogin` programma in het /etc/sliphome/slip.hosts bestand voor een regel dat een corresponderende user ID heeft, er kan bijvoorbeeld een regel staan in /etc/sliphome/slip.hosts dat eruit ziet als volgend:

```
Shelmerg        dc-slip sl-helmer       0xfffffc00        autocomp
```

`sliplogin` zal de corresponderende regel vinden en de seriële lijn koppelen aan de eerste beschikbare SLIP-interface, waarna /etc/sliphome/slip.login wordt uitgevoerd zoals volgend:

```
/etc/sliphome/slip.login 0 19200 Shelmerg dc-slip sl-helmer 0xfffffc00 autocomp
```

Als alles goed gaat, zal /etc/sliphome/slip.login een `ifconfig` commando uitvoeren voor de SLIP

interface waaraan `sliplogin` zichzelf koppelt (SLIP-interface 0 zoals in bovenstaand voorbeeld was de eerste parameter in de lijst welke gegeven is aan slip.login) om een lokaal IP-adres in te stellen (`dc-slip`), een remote IP adres (`sl-helmer`), een netwerk master voor de SLIP-interface (`0xffffc00`), en enkele additionele vlaggen (`autocomp`). Als er iets misgaat zal `sliplogin` meestal voldoende goede informatie loggen via de syslogd daemon faciliteiten, welke meestal logt naar /var/log/messages (zie de handleidingen van syslogd(8) en syslog.conf(5) en controleer het /etc/syslog.conf bestand om te zien wat syslogd logt en waar dit naartoe gelogt wordt).

Kernel-configuratie

FreeBSD's standaard kernel (GENERIC) heeft reeds ondersteuning voor SLIP (sl(4)), in het geval van een custom kernel moet de volgende regel worden toegevoegd aan de kernel configuratie:

```
device   sl
```

Standaard zal uw FreeBSD machine geen pakketten doorsturen. Als u wilt dat uw FreeBSD SLIP-server zich gedraagt als router zult u het bestand /etc/rc.conf moeten bewerken en de instelling van de `gateway_enable` variabele moeten aanpassen naar `YES`. Dit zorgt ervoor dat de machine na een herstart zich zal blijven gedragen als router.

Om de instellingen meteen actief te maken kunt u het volgende commando als `root` uitvoeren:

```
# service routing start
```

Raadpleeg aub De FreeBSD-kernel instellen over het configureren van de FreeBSD kernel voor meer hulp over het herconfigureren van uw kernel.

Sliplogin-configuratie

Zoals eerder vermeld, zijn er drie bestanden in de map /etc/sliphome die onderdeel zijn van de configuratie voor /usr/sbin/sliplogin (zie sliplogin(8) voor de actuele handleiding voor `sliplogin`): slip.hosts, welke de SLIP-gebruikers definieert en de gekoppelde IP adressen; slip.login, welke meestal de SLIP-interface configureert en (optioneel) slip.logout, welke de effecten van slip.login ongedaan maakt wanneer de seriële verbinding verbroken wordt.

slip.hosts configuratie

/etc/sliphome/slip.hosts bevat regels welke minstens vier onderdelen heeft die gescheiden worden door een spatie:

- SLIP-gebruikers login ID
- Lokale adres (lokaal voor de SLIP-server) van de SLIP-link
- Remote adres van de SLIP-link
- Netwerk masker

De lokale en remote adressen mogen host namen zijn (Welke naar IP-adressen vertaald kunnen worden door /etc/hosts of door de DNS diensten, afhankelijk van uw specificaties in het

/etc/nsswitch.conf, het netwerk masker mag een naam zijn dat vertaald kan worden door een zoek opdracht in /etc/networks. Op een voorbeeld systeem ziet het /etc/sliphome/slip.hosts bestand er als volgt uit:

```
#
# login local-addr      remote-addr     mask            opt1    opt2
#                                               (normal,compress,noicmp)
#
Shelmerg  dc-slip       sl-helmerg      0xfffffc00      autocomp
```

Aan het einde van deze regel staan één of meerdere opties:

- `normal` - geen compressie van de header
- `compress` - comprimeer headers
- `autocomp` - comprimeer de headers als de remote kant dit accepteert
- `noicmp` - schakelt ICMP pakketten uit (dus alle "ping" pakketten worden geweigerd in plaats van dat deze bandbreedte verbruiken)

Uw keuze van een lokaal en remote adres voor uw SLIP verbindingen is afhankelijk van of u een speciaal toegewezen TCP/IP-subnet gebruikt, of dat u gebruik gaat maken van "proxy ARP" op uw SLIP-server (het is geen echte "proxy ARP", maar het is de terminologie welke in deze sectie gebruikt wordt om het te beschrijven). Als u niet zeker bent welke methode u moet kiezen, of hoe u IP-adressen moet toewijzen, raadpleegt u dan de TCP/IP boeken die vermeld worden in de SLIP vereisten (Eisen vooraf) en/of vraag uw IP-netwerk manager om hulp.

Als u gebruik gaat maken van een separaat subnet voor uw SLIP-cliënten, moet u een subnet alloceren uit de voor u toegewezen IP-ruimte, en elke SLIP-cliënt een IP-adres geven uit dat subnet. Daarna moet u waarschijnlijk een statische route configureren voor uw SLIP-subnet via uw SLIP-server naar de dichtsbijzijnde IP-router.

In het andere geval moet u gebruik maken van de "proxy ARP" methode, u moet elke SLIP cliënt een IP-adres geven uit het Ethernet-subnet van uw SLIP-server, daarnaast moet u het /etc/sliphome/slip.login en het /etc/sliphome/slip.logout script aanpassen om gebruik te maken van arp(8) om de "proxy ARP" regels te beheren in de SLIP servers ARP tabel.

slip.login configuratie

Een typisch /etc/sliphome/slip.login bestand ziet er als volgend uit:

```
#!/bin/sh -
#
#       @(#)slip.login  5.1 (Berkeley) 7/1/90

#
# generiek loginbestand voor een SLIP-lijn.  sliplogin voert deze uit
# met de volgende parameters:
#      1        2         3        4          5         6     7-n
```

```
#   slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 inet $4 $5 netmask $6
```

Dit slip.login bestand start alleen het `ifconfig` commando voor de betreffende SLIP-interace met het lokale en remote adres met het netwerkmasker van de SLIP-interface.

Als u besloten heeft om gebruik te maken van de "proxy ARP" methode (in plaats van het gebruiken van een apart subnet voor uw SLIP-cliënten) moet u het /etc/sliphome/slip.login bestand aanpassen zodat deze er ongeveer als volgend uitziet:

```
#!/bin/sh -
#
#       @(#)slip.login  5.1 (Berkeley) 7/1/90

#
# generiek loginbestand voor een SLIP-lijn.  sliplogin voert deze uit
# met de volgende parameters:
#      1        2         3        4          5         6     7-n
#   slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 inet $4 $5 netmask $6
# Beantwoord ARP Verzoeken voor de SLIP-cliënt met ons Ethernet
# adres
/usr/sbin/arp -s $5 00:11:22:33:44:55 pub
```

De extra regel in het slip.login bestand, `arp -s $5 00:11:22:33:44:55 pub`, creeërt een ARP-regel in de ARP-tabel van de SLIP-server. Deze ARP regel zorgt ervoor dat de SLIP-server antwoord geeft met het Ethernet MAC adres van de SLIP-server wanneer een andere IP-node op het Ethernet vraagt om te communiceren met het IP-adres van de SLIP-cliënt.

Wanneer u gebruik maakt van het voorbeeld hierboven, wees u er dan zeker van dat u het Ethernet MAC adres (`00:11:22:33:44:55`) veranderd in het MAC adres van uw systeem's Ethernet kaart, anders werkt uw "proxy ARP" zeker niet! U kunt het Ethernet MAC adres van uw SLIP-server achterhalen door het bekijken van het resultaat van `netstat -i`; de tweede regel met resultaten moet er ongeveer als volgend uitzien:

```
ed0   1500  <Link>0.2.c1.28.5f.4a         191923    0   129457     0   116
```

Dit geeft aan dat het specifieke Ethernet MAC adres van het systeem is `00:02:c1:28:5f:4a` - de punten in het Ethernet MAC adres welke gegeven wordt door `netstat -i` moet worden veranderd in dubbele punten (":") en voorloop nullen moeten worden toegevoegd aan elk enkel hexadecimaal getal om het adres te converteren naar de vorm die arp(8) wenst; zie de handleiding van arp(8) voor een compleet overzicht van het gebruik hiervan.



Wanneer u /etc/sliphome/slip.login en /etc/sliphome/slip.logout, aanmaakt moet het "uitvoerbare" bitje gezet zijn (bijvoorbeeld `chmod 755 /etc/sliphome/slip.login`

> `/etc/sliphome/slip.logout`) anders is `sliplogin` niet in staat om deze uit te voeren.

slip.logout configuratie

/etc/sliphome/slip.logout is niet strict noodzakelijk (tenzij u "proxy ARP" implementeert), maar als beslist om deze aan te maken is dit een voorbeeld basis slip.logout script:

```
#!/bin/sh -
#
#       slip.logout

#
# uitlogbestand voor een SLIP-regel.  sliplogin voert deze uit met de
# parameters:
#      1        2         3        4          5         6     7-n
#   slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 down
```

Als u gebruik maakt van "proxy ARP" wilt u waarschijnlijk dat het /etc/sliphome/slip.logout bestand de ARP regel weghaalt voor de SLIP-cliënt:

```
#!/bin/sh -
#
#       @(#)slip.logout

#
# uitlogbestand voor een SLIP-regel.  sliplogin voert deze uit met de
# parameters:
#      1        2         3        4          5         6     7-n
#   slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 down
# Stop met het beantwoorden van ARP verzoeken voor de SLIP-cliënt
/usr/sbin/arp -d $5
```

Het `arp -d $5` verwijderd de ARP regel, die "proxy ARP" heeft toegevoegd toen de SLIP-cliënt inlogde.

Het is belangrijk om dit te herhalen: zorg ervoor /etc/sliphome/slip.logout het uitvoerbare bitje heeft gezet nadat deze gecreeërd is (b.v. `chmod 755 /etc/sliphome/slip.logout`).

Routering-overwegingen

Als u geen gebruik maakt van de "proxy ARP" voor het routeren van pakketten tussen uw SLIP-cliënten en de rest van uw netwerk (en wellicht het internet), moeten er misschien een aantal statische routeringen naar de best bereikbare standaard router ingesteld worden om uw SLIP cliënts te routeren via uw SLIP0server.

Statische routeringen

Het toevoegen van statische routeringen naar de dichts bijzijnde router kan problematisch zijn (of zelfs onmogelijk als u niet de bevoegdheden heft om dit te doen). Als u een netwerk heeft met meerdere routers binnen uw organisatie kan het zijn dat sommige routers, zoals die van Cisco en Proteon , niet alleen geconfigureerd moeten worden met de statische route naar het SLIP-subnet, maar deze moeten dan ook geconfigureerd worden over welke statische routes aan andere routers verteld moeten worden. Enige expertise en fine tunen kan nodig zijn om statische routing te laten werken.

Hoofdstuk 26. Elektronische mail

26.1. Overzicht

"Elektronische Mail", beter bekend als email, is tegenwoordig een van de meest gebruikte vormen van communicatie. Dit hoofdstuk geeft een algemene inleiding in het opzetten van een mailserver op FreeBSD, alsmede een introductie in het verzenden en ontvangen van email op FreeBSD; het is echter geen complete referentie en veel belangrijke overwegingen zullen buiten beschouwing worden gelaten. Voor een completere behandeling wordt de lezer gewezen op de vele uitstekende boeken welke worden vermeld in Bibliografie.

In dit hoofdstuk wordt behandeld:

- Welke software (componenten) gebruikt wordt(en) bij het verzenden en ontvangen van email.
- Waar algemene sendmail instellingsbestanden worden opgeslagen in FreeBSD.
- Het verschil tussen lokale en postbussen op-afstand.
- Hoe spammers te verhinderen dat ze de mailserver illegaal gebruiken als "relay".
- Hoe een andere MTA (Mail Transfer Agent) te installeren en configureren op het systeem, ter vervanging van sendmail.
- Hoe veel voorkomende problemen met mail servers worden opgelost.
- Hoe SMTP met UUCP te gebruiken.
- Hoe een systeem in te stellen om alleen mail te verzenden.
- Hoe email te gebruiken met een inbelverbinding.
- Hoe SMTP Authenticatie in te stellen voor verhoogde beveiliging.
- Hoe een Mail User Agent zoals mutt te installeren om email te verzenden en te ontvangen.
- Hoe mail te downloaden van een POP of IMAP server op afstand.
- Hoe automatisch filters en sorteerregels op inkomende email toe te passen.

Voordat dit hoofdstuk gelezen wordt, dienen:

- De netwerkverbindingen correct ingesteld te zijn (Geavanceerd netwerken).
- De juiste DNS-informatie ingesteld te zijn voor de mailserver (Netwerkdiensten).
- Bekend te zijn hoe software van derde partijen te installeren (Applicaties installeren. pakketten en ports).

26.2. Gebruik maken van elektronische mail

Er zijn vijf belangrijke componenten betrokken bij het uitwisselen van email. Dit zijn: het gebruikersprogramma, de serverdaemon, DNS, "een postbus, lokaal of op afstand ", en natuurlijk de mailhost zelf.

26.2.1. Het gebruikersprogramma

Dit omvat opdrachtregelprogramma's zoals mutt, alpine, elm, en `mail`, en GUI programma's zoals balsa, xfmail, en iets "geavanceerders" zoals een webbrowser. Deze programma's doen niets anders dan de mail bezorgen bij de lokale "mailhost", door deze af te leveren of bij een van de beschikbare serverdiensten, of via TCP.

26.2.2. Mailhost Server Daemon

FreeBSD wordt standaard geleverd met de sendmail, maar ondersteunt meerdere andere mailserver daemons, zoals:

- exim;
- postfix;
- qmail.

De server daemon heeft meestal twee functies-het is verantwoordelijk voor het ontvangen van inkomende mail en het bezorgen van uitgaande mail. Het is *niet* verantwoordelijk voor het verzamelen van mail door gebruik te maken van protocollen zoals POP of IMAP om mail te lezen, noch staat het toe om een verbinding te maken met een lokale mbox of Maildir postbus. Het is mogelijk dat daarvoor een extra daemon voor nodig is.



Oudere versies van sendmail hebben serieuze beveiligingslekken welke kunnen leiden tot een situatie waarbij een aanvaller lokale of toegang van afstand tot de machine kan verkrijgen. Draai een actuele versie om deze problemen te voorkomen. Optioneel kan een alternatieve MTA van de FreeBSD Portscollectie geïnstalleerd worden.

26.2.3. Email en DNS

Het Domein Naam Systeem (DNS) en de daemon `named` spelen een grote rol in het bezorgen van email. Om het mogelijk te maken email van de deze lokatie naar een andere lokatie te bezorgen, zal de serverdaemon de andere lokatie opzoeken in het DNS om zo de host te bepalen die de email voor de bestemming in ontvangst zal nemen. Dit gebeurt ook als email verzonden wordt vanaf een andere host naar de lokale mailserver.

DNS is verantwoordelijk voor het koppelen van hostnamen aan IP-adressen, en voor het opslaan van specifieke informatie voor het bezorgen van mail, bekend als MX-regels. De MX-regel (Mail eXchanger) specificeert welke host(s) mail zullen ontvangen voor een specifiek domein. Als er geen MX-regel is voor deze hostnaam of dit domein, dan zal de mail direct bij de host worden afgeleverd, mits er een A-regel is die deze hostnaam aan dit IP-adres koppelt.

De MX-regels van een willekeurig domein kunnen worden bekeken door gebruik te maken van het commando host(1), zoals te zien is in het onderstaande voorbeeld:

```
% host -t mx FreeBSD.org
FreeBSD.org mail is handled (pri=10) by mx1.FreeBSD.org
```

26.2.4. Mail ontvangen

De mailhost verzorgt het ontvangen van mail voor het domein. Deze zal alle mail verzonden aan het domein verzamelen en deze afhankelijk van de configuratie opslaan in òf mbox (de standaardmanier om mail op te slaan) òf in Maildir-formaat. Wanneer de mail eenmaal is opgeslagen, kan het òf lokaal gelezen worden door toepassingen als mail(1) of mutt , of op afstand bekeken en verzameld worden middels protocollen als POP of IMAP. Dit betekent, dat als mail alleen lokaal wordt gelezen, er geen POP- of IMAP-server geïnstalleerd hoeft te worden.

Op afstand toegang tot de postbus krijgen door gebruik te maken van POP en IMAP

Om op afstand toegang te krijgen tot postbussen is het nodig toegang te hebben tot een POP- of IMAP -server. Deze protocollen stellen gebruikers in staat hun postbus gemakkelijk op afstand te benaderen. Hoewel zowel POP als IMAP gebruikers in staat stellen op afstand een postbus te bereiken, biedt IMAP veel voordelen, waaronder:

- IMAP kan berichten zowel op de server op afstand opslaan als ze ophalen.
- IMAP ondersteunt gelijktijdig actualiseren.
- IMAP kan uitstekend worden gebruikt over langzame verbindingen omdat het gebruikers in staat stelt de structuur van berichten te bekijken zonder deze binnen te halen; het kan ook worden gebruikt om te zoeken op de server om zo de gegevensoverdracht tussen client en server te minimaliseren.

Om een POP- of IMAP- server te installeren, zijn de volgende stappen nodig:

1. Kies een IMAP- of POP -server die het beste aan de eisen voldoet. De volgende POP- en IMAP -servers zijn zeer bekend en zijn goede voorbeelden:
   - qpopper;
   - teapop;
   - imap-uw;
   - courier-imap;
   - dovecot;
2. Installeer de gewenste POP- of IMAP-daemon vanuit de Portscollectie.
3. Wijzig indien nodig /etc/inetd.conf om de POP- of IMAP - server te laden.



Let er wel op dat zowel POP en IMAP informatie, waaronder gegevens over gebruikersnaam en wachtwoord, onversleuteld versturen. Dit betekent, dat wanneer het gewenst is dat de uitwisseling van gegevens over deze protocollen versleuteld is, het verstandig is om te overwegen de sessies over ssh(1) te tunnelen of SSL te gebruiken. Het tunnelen van sessies wordt beschreven in SSH tunnels en SSL in OpenSSL.

Toegang tot lokale postbussen

Postbussen kunnen lokaal benaderd worden door direct op de server waarop de postbus wordt bewaard MUAs te gebruiken. Dit kan gedaan worden door programma's zoals mutt of mail(1) te gebruiken.

26.2.5. De mailhost

De mailhost is de naam van de server welke verantwoordelijk is voor het afleveren en ontvangen van mail voor de server en mogelijk voor het netwerk.

26.3. sendmail instellen

sendmail(8) is de standaard Mail Transfer Agent (MTA) in FreeBSD. sendmail's taak is het accepteren van mail van gebruikersprogramma's (MUA) en deze te bezorgen bij de juiste mailer zoals gedefinieerd in het betreffende configuratiebestand. sendmail kan ook netwerkverbindingen accepteren en mail in lokale postbussen afleveren of bezorgen bij een ander programma.

sendmail gebruikt de volgende configuratiebestanden:

Bestandsnaam	Functie
/etc/mail/access	bestand met de toegangsdatabase van sendmail
/etc/mail/aliases	Aliases voor postbussen
/etc/mail/local-host-names	Lijst van servers waarvoor sendmail mail accepteert
/etc/mail/mailer.conf	Configuratie voor het mailerprogramma
/etc/mail/mailertable	Aflevertabel voor de mailer
/etc/mail/sendmail.cf	Hoofdconfiguratiebestand van sendmail
/etc/mail/virtusertable	Tabellen voor virtuele gebruikers en domeinen

26.3.1. /etc/mail/access

De toegangsdatabase definieert welke host(s) of IP-adressen toegang hebben tot de lokale mailserver en wat voor soort toegang ze hebben. Hosts kunnen in de lijst als `OK`, `REJECT`, of `RELAY` staan, of worden doorgevoerd naar de foutafhandelingsprocedure van sendmail met een bepaalde mailerfout. Hosts welke vermeld staan als `OK`, wat de standaard is, kunnen mail versturen naar deze host zolang de eindbestemming van de mail de lokale machine is. Hosts welke vermeld staan als `REJECT` worden voor alle verbindingen geweigerd. Hosts met een `RELAY` vermelding wordt toegestaan om via deze server mail naar elke bestemming te sturen.

Voorbeeld 18. Configureren van de sendmail toegangsdatabase

```
cyberspammer.com    550 We accepteren geen mail van spammers
FREE.STEALTH.MAILER@    550 We accepteren geen mail van spammers
another.source.of.spam  REJECT
```

```
okay.cyberspammer.com   OK
128.32  RELAY
```

In dit voorbeeld staan vijf vermeldingen. Mailafzenders die overeenkomen met de linkerzijde van de tabel worden beïnvloed door de actie die vermeld staan aan de rechterzijde van de tabel. De eerste twee voorbeelden geven een foutcode af aan de foutafhandelingsroutine van sendmail. Het bericht wordt bij de externe host bekend gemaakt wanneer een mail voldoet aan de linkerzijde van de tabel. De volgende regel weigert mail van een specifieke host op het Internet, `another.source.of.spam`. De volgende regel accepteert mailverbindingen van een host `okay.cyberspammer.com`, welke nauwkeuriger is dan de regel met `cyberspammer.com` erboven. Specifiekere regels vervangen minder specifieke. De laatste regel staat het doorsturen van elektronische mail toe vanaf hosts waarvan de IP-adressen beginnen met `128.32`. Deze hosts zijn dan in staat om via deze mailserver naar een andere bestemming mail te versturen.

Wanneer dit bestand is bijgewerkt, dient `make` in /etc/mail/ te gedraaid te worden om de database bij te werken.

26.3.2. /etc/mail/aliases

De aliasdatabase bevat een lijst met virtuele postbussen die verwijzen naar andere gebruiker(s), bestand(en), programma('s) of andere aliassen. Hier zijn een paar voorbeelden die gebruikt kunnen worden in /etc/mail/aliases:

Voorbeeld 19. Mailaliassen

```
root: localuser
ftp-bugs: joe,eric,paul
bit.bucket:  /dev/null
procmail: "|/usr/local/bin/procmail"
```

Het bestandsformaat is simpel; de postbusnaam aan de linkerzijde van de dubbele punt wordt verder uitgewerkt naar de doel(en) aan de rechterzijde. Het eerste voorbeeld breidt de postbus van `root` uit naar de postbus `localuser`, welke dan vervolgens weer wordt opgezocht in de aliasdatabase. Als er geen verdere overeenkomst wordt gevonden, dan wordt het bericht afgeleverd bij de lokale gebruiker `localuser`. Het volgende voorbeeld toont een mailinglijst. Mail voor de postbus `ftp-bugs` wordt doorverwezen naar de drie lokale postbussen `joe`, `eric` en `paul`. Merk op dat een externe postbus gespecificeerd kan worden als user@example.com. Het volgende voorbeeld toont het schrijven van mail naar een bestand, in dit geval /dev/null. Het laatste voorbeeld toont het sturen van mail naar een programma, in dit geval wordt het mailbericht doorgestuurd naar de standaard invoer van /usr/local/bin/procmail via een UNIX® pijp.

Wanneer dit bestand is bijgewerkt, dient `make` in /etc/mail/ gedraaid te worden om de database bij te werken.

26.3.3. /etc/mail/local-host-names

Dit is een lijst van hostnamen die sendmail(8) moet accepteren als de lokale hostnaam. Hierin dienen alle hostnamen geplaatst te worden waarvoor sendmail mail moet ontvangen. Als deze mailserver mail moet ontvangen voor het domein `example.com` en de hostnaam is `mail.example.com`, dan ziet local-host-names er ongeveer zo uit:

```
example.com
mail.example.com
```

Wanneer dit bestand is bijgewerkt, dient sendmail(8) opnieuw gestart te worden zodat het de veranderingen kan lezen.

26.3.4. /etc/mail/sendmail.cf

Het hoofdinstellingenbestand van sendmail , sendmail.cf controleert het algemene gedrag van sendmail, inclusief alles van het herschrijven van emailadressen tot het sturen van weigeringsberichten naar externe mailservers. Met zo'n diverse rol is dit instellingenbestand redelijk complex en vallen de details buiten het bereik van dit hoofdstuk. Gelukkig hoeft dit bestand maar zelden aangepast te worden voor standaard mailservers.

Het hoofdinstellingenbestand van sendmail kan gebouwd worden met m4(1) macro's die het gedrag en de mogelijkheden van sendmail specificeren. Lees /usr/src/contrib/sendmail/cf/README voor meer details.

Wanneer dit bestand is bijgewerkt, dient sendmail(8) opnieuw gestart te worden om de wijzigingen door te voeren.

26.3.5. /etc/mail/virtusertable

De virtusertable verbindt mailadressen voor virtuele domeinen en postbussen met echte postbussen. Deze postbussen kunnen lokaal, op afstand, aliassen gedefinieerd in /etc/mail/aliases, of bestanden zijn.

Voorbeeld 20. Voorbeeld van een mailtabel voor een virtueel domein

```
root@example.com    root
postmaster@example.com  postmaster@noc.example.net
@example.com    joe
```

In het voorbeeld hierboven staat een tabel voor een domein `example.com`. Dit bestand wordt van boven naar beneden verwerkt, en de eerste overeenkomende regel wordt gebruikt. De eerste regel verbindt root@example.com met de lokale postbus `root`. De volgende regel verbindt postmaster@example.com met de postbus `postmaster` op de host `noc.example.net`. Als geen van de vorige regels van `example.com` overeenkomen, zal de laatste regel gebruikt worden, die alle andere post geadresseerd aan iemand bij `example.com` opvangt en naar de lokale postbus `joe` stuurt.

26.4. De Mail Transfer Agent vervangen

Zoals eerder vermeld wordt FreeBSD geleverd met sendmail voorgeïnstalleerd als MTA (Mail Transfer Agent). Daarom regelt het standaard uitgaande en binnenkomende mail.

In sommige gevallen willen systeembeheerders wegens uiteenlopende redenen hun MTA vervangen. Deze redenen variëren van het uitproberen van een andere MTA tot het installeren van een bepaalde functionaliteit of pakket dat afhankelijk is van een andere MTA.

26.4.1. Een nieuwe MTA installeren

Er is een waaier van MTA's beschikbaar. Een goed startpunt is de FreeBSD Ports Collectie waar er veel gevonden kunnen worden. Het is natuurlijk mogelijk iedere MTA te gebruiken vanaf iedere locatie, zolang het draait op FreeBSD.

Begin met het installeren van de nieuwe MTA. Als de MTA eenmaal geïnstalleerd is wordt er de kans gegeven te beslissen of de nieuwe MTA echt voldoet aan de eisen, en is het mogelijk de nieuwe software te configureren voordat deze het werk van sendmail overneemt. Bevestig voordat de MTA geïnstalleerd wordt dat de nieuwe software geen poging onderneemt systeemtoepassingen zoals /usr/bin/sendmail te overschrijven, anders wordt de nieuwe software onmiddellijk in gebruik genomen voordat het is geconfigureerd.

Neem de documentatie van de gekozen MTA door voor meer informatie over het configureren van de software.

26.4.2. sendmail uitschakelen



Als sendmail's uitgaande emaildienst uitgeschakeld wordt, is het belangrijk dat het vervangen wordt door een alternatief systeem. Als ervoor gekozen wordt dit niet te doen, zullen systeemfunctionaliteiten zoals periodic(8) niet in staat zijn hun resultaten te bezorgen per email, zoals ze normaliter verwachten te kunnen doen. Vele delen van het systeem zullen verwachten een werkend systeem aan te treffen dat compatibel is met sendmail. Als toepassingen binaries van sendmail blijven gebruiken om mail te versturen nadat deze uitgeschakeld werden, kan de mail in een inactieve sendmail wachtrij geplaatst worden, en nooit bezorgd worden.

Om sendmail volledig uit te schakelen, inclusief de uitgaande emaildienst, dient

```
sendmail_enable="NO"
sendmail_submit_enable="NO"
sendmail_outbound_enable="NO"
sendmail_msp_queue_enable="NO"
```

toegevoegd te worden aan /etc/rc.conf.

Als enkel sendmail's ingaande emaildienst uitgeschakeld dient te worden, dient

```
sendmail_enable="NO"
```

toegevoegd te worden aan /etc/rc.conf. Meer informatie over de opstartopties van sendmail is beschikbaar in de hulppagina rc.sendmail(8).

26.4.3. De nieuwe MTA starten tijdens het opstarten

De nieuwe MTA kan gestart worden door deze instellingsregel toe te voegen aan /etc/rc.conf, zoals het volgende voorbeeld voor postfix:

```
# echo 'postfix_enable="YES"' >> /etc/rc.conf
```

De MTA zal nu automatisch tijdens het opstarten worden gestart.

26.4.4. sendmail vervangen als de standaard systeemmailer

Het programma sendmail is zo vanzelfsprekend als standaard software op UNIX® systemen dat sommige softwarepakketten ervan uitgaan dat sendmail reeds geïnstalleerd en geconfigureerd is. Daarom voorzien vele alternatieve MTA's in compatibele implementaties van de opdrachtregelinterface van sendmail; dit vergemakkelijkt het gebruik van alternatieve MTA's als vervanging voor sendmail.

Bij het gebruiken van een alternatieve MTA moet men er zeker van zijn dat software die probeert de standaardtoepassingen van sendmail zoals /usr/bin/sendmail te gebruiken, ook daadwerkelijk de gekozen alternatieve mailer gebruikt. Gelukkig heeft FreeBSD hiervoor een systeem, mailwrapper(8), dat deze taak van de systeembeheerder overneemt.

Als sendmail werkt zoals origineel geïnstalleerd, bevat /etc/mail/mailer.conf bij benadering het volgende:

```
sendmail    /usr/libexec/sendmail/sendmail
send-mail   /usr/libexec/sendmail/sendmail
mailq       /usr/libexec/sendmail/sendmail
newaliases  /usr/libexec/sendmail/sendmail
hoststat    /usr/libexec/sendmail/sendmail
purgestat   /usr/libexec/sendmail/sendmail
```

Dit wil zeggen dat wanneer een van deze algemene opdrachten (zoals sendmail zelf) uitgevoerd wordt, het systeem in werkelijkheid een kopie van de mailwrapper genaamd sendmail uitvoert, dat mailer.conf controleert en /usr/libexec/sendmail/sendmail uitvoert. Dit systeem maakt het eenvoudiger te specificeren welke toepassingen daadwerkelijk uitgevoerd worden wanneer deze standaard sendmail functies aangeroepen worden.

Als men bijvoorbeeld wil dat /usr/local/supermailer/bin/sendmail-compat uitgevoerd wordt in plaats van sendmail, kan men /etc/mail/mailer.conf als volgt aanpassen:

```
sendmail    /usr/local/supermailer/bin/sendmail-compat
send-mail   /usr/local/supermailer/bin/sendmail-compat
mailq       /usr/local/supermailer/bin/mailq-compat
newaliases  /usr/local/supermailer/bin/newaliases-compat
hoststat    /usr/local/supermailer/bin/hoststat-compat
purgestat   /usr/local/supermailer/bin/purgestat-compat
```

26.4.5. Afwerking

Wanneer alles correct geconfigureerd is, dienen ofwel alle ongebruikte sendmail processen gestopt te worden en de processen behorend aan de nieuwe software gestart te worden, ofwel dient het systeem opnieuw gestart te worden. Herstarten geeft ook de mogelijkheid te controleren of de nieuwe MTA correct geconfigureerd is om tijdens het opstartproces gestart te worden.

26.5. Problemen oplossen

26.5.1. Waarom is het nodig om de FQDN te gebruiken voor hosts op de site?

Het is waarschijnlijk dat de host zich in een ander domein bevindt; bijvoorbeeld als het gewenst is om host `mompel` in het domein `bar.edu` vanuit domein `foo.bar.edu` te bereiken, is het nodig om er met de volledig gekwalificeerde domeinnaam naar te verwijzen, `mompel.bar.edu`, in plaats van slechts `mompel`.

Traditioneel werd dit door BSD BIND resolvers toegestaan. De huidige versie van BIND die met FreeBSD wordt geleverd levert niet langer standaard afkortingen voor onvolledig gekwalificeerde domeinnamen anders dan het huidige domein. Dus moet een ongekwalificeerde host `mompel` òf als `mompel.foo.bar.edu` gevonden worden, òf wordt er naar gezocht in het root-domein.

Dit verschilt van het vorige gedrag, waar de zoektocht doorging over `mompel.bar.edu`, en `bar.edu`. Zie RFC 1535 voor de redenen waarom dit als een slechte gewoonte en zelfs als beveiligingslek werd beschouwd.

Als een goede tussenoplossing kan deze regel:

```
search foo.bar.edu bar.edu
```

in plaats van het voorgaande:

```
domain foo.bar.edu
```

in /etc/resolv.conf geplaatst worden. Ben er echter zeker van dat de zoekvolgorde niet verder gaat dan de "grens tussen lokale en publieke regelgeving", zoals RFC 1535 het noemt.

26.5.2. sendmail zegt mail loops back to myself

Dit wordt in de FAQ van sendmail als volgt beantwoord:

```
Deze foutmeldingen verschijnen:

553 MX list for domain.net points back to relay.domain.net
554 <user@domain.net>... Local configuration error

Hoe kan dit probleem worden opgelost?

Er is gevraagd om mail van het domein (bijvoorbeeld domain.net) naar een
specifieke host door te sturen (in dit geval relay.domain.net) door
gebruik te maken van een MX-regel, maar de machine die het door moet
sturen herkent zichzelf niet als domain.net.  Voeg domain.net toe aan
/etc/mail/local-host-names [bekend als /etc/sendmail.cw voor versies
eerder dan 8.10] (als FEATURE(use_cw_file) gebruikt wordt) of voeg
Cw domain.net toe aan /etc/mail/sendmail.cf.
```

De FAQ van sendmail is te vinden op http://www.sendmail.org/faq/ en wordt aangeraden om te lezen indien enig "tweaken " van de mailinstallatie gewenst is.

26.5.3. Hoe kan een mailserver op een inbel-PPP-host gedraaid worden?

Het is gewenst om een FreeBSD-computer in een LAN met het Internet te verbinden. De FreeBSD-computer zal en mail-gateway voor het LAN zijn. De PPP-verbinding is niet toegewijd.

Er zijn minstens twee manieren om dit te doen. Eén manier is om UUCP te gebruiken.

Een andere manier is ervoor te zorgen dat een server die altijd met het Internet verbonden is secundaire MX-diensten voor het domein biedt. Als het domein bijvoorbeeld `example.com` is en de internetprovider `example.net` heeft ingesteld om secundaire MX-diensten voor het domein te bieden:

```
example.com.        MX  10  example.com.
            MX  20  example.net.
```

Er dient slechts één host als de uiteindelijke ontvanger gespecificeerd te worden (voeg `CW example.com` toe aan /etc/mail/sendmail.cf op `example.com`>).

Wanneer de verzendende `sendmail` probeert om mail af te leveren zal het proberen met `example.com` te verbinden via de modemverbinding. Waarschijnlijk zal dit een time-out geven omdat de computer niet online is. Het programma sendmail zal het automatisch aan de secundaire MX-site, de internetprovider (`example.net`) afleveren. De secundaire MX zal dan periodiek proberen om een verbinding te maken met de computer en de mail aan de primaire MX-host leveren (`example.com`).

Het kan wenselijk zijn om iets als het onderstaande als inlogscript te gebruiken:

```
#!/bin/sh
# Zet mij in /usr/local/bin/pppmyisp
( sleep 60 ; /usr/sbin/sendmail -q ) &
/usr/sbin/ppp -direct pppmyisp
```

Indien er een apart inlogscript voor een gebruiker wordt aangemaakt, kan `sendmail -qRexample.com` gebruikt worden in plaats van het bovenstaande script. Dit zorgt ervoor dat alle mail in de mailrij voor `example.com` onmiddellijk verwerkt wordt.

Een verdere verfijning van de situatie is deze:

Bericht gestolen van de FreeBSD Internet service provider mailinglijst.

```
> we bieden de secundaire MX voor een klant.
> De klant maakt automatisch verschillende keren per dag een verbinding
> met onze diensten om de mailberichten naar zijn primaire MX te
> sturen (we bellen zijn site niet indien er een mail voor zijn
> domein arriveert).  Onze sendmail verstuurt de mailrij om de 30
> minuten.  Op het moment moet de klant 30 minuten online blijven om
> er zeker van te zijn dat alle mail naar de primaire MX is gegaan.
>
> Is er een commando dat sendmail er toe aanzet om alle mailberichten
> nu te versturen?  De gebruiker heeft uiteraard geen root-rechten op
> onze machine.

In de sectie privacy flags van sendmail.cf staat een
definitie Opgoaway,restrictqrun

Verwijder restrictqrun om niet-root-gebruikers toe te staan te beginnen
de rij te verwerken.  Het kan ook wenselijk zijn om de MXs opnieuw te
rangschikken.  Wij zijn zo de eerste MX voor onze klanten, en we hebben
dit gedefinieerd:

# Als we de beste MX voor een host zijn, probeer direct in plaats van
# een lokale configuratiefout te genereren.
OwTrue

Op deze manier zal een site op afstand rechtstreeks hier afleveren,
zonder de verbinding van de klant te proberen.  Vervolgens wordt er naar
de klant verstuurd.  Dit werkt alleen voor hosts, dus
dient de klant hun mailcomputer customer.com te noemen en
hostname.customer.com in de DNS de plaatsen.  Plaats een
A-regel in de DNS voor customer.com.
```

26.5.4. Waarom blijven er fouten als Relaying Denied verchijnen wanneer er mail van andere hosts wordt verstuurd?

In standaard FreeBSD-installaties is sendmail geconfigureerd om alleen mail te versturen van de

host waarop het draait. Als bijvoorbeeld een POP-server beschikbaar is, kunnen gebruikers mail controleren vanuit school, werk, of andere lokaties op afstand, maar zullen ze nog steeds niet in staat zijn om uitgaande emails van lokaties van buitenaf te versturen. Gewoonlijk zal er na enkele ogenblikken na de poging een email van MAILER-DAEMON worden verzonden met een foutbericht `5.7 Relaying Denied`.

Er zijn verschillende manieren om dit te omzeilen. De oplossing die het meest voor de hand ligt, is om het adres van de internetprovider in een bestand relay-domains op /etc/mail/relay-domains te zetten. Een snelle manier om dit te doen is:

```
# echo "your.isp.example.com" > /etc/mail/relay-domains
```

Nadat dit bestand is aangemaakt of bewerkt dient sendmail opnieuw gestart te worden. Dit werkt prima indien u een serverbeheerder bent en het niet wenselijk is om mail lokaal te verzenden, of indien het gewenst is om een point-en-click client/systeem op een andere machine of zelfs bij een andere internetprovider te gebruiken. Het is ook erg bruikbaar indien er slechts enkele email-accounts zijn aangemaakt. Als er een groot aantal adressen dient te worden toegevoegd, kan dit bestand in een tekstverwerker worden geopend en de domeinen worden toegevoegd, één per regel:

```
your.isp.example.com
other.isp.example.net
users-isp.example.org
www.example.org
```

Nu zal het verzenden van elke mail door dit systeem, verstuurd door elke host in deze lijst, lukken (aangenomen dat de gebruiker een account op het systeem heeft). Dit is een aardige manier om gebruikers toe te staan op afstand mail vanaf het systeem te verzenden zonder dat mensen wordt toegestaan om spam vanaf het systeem te verzenden.

26.6. Geavanceerde onderwerpen

De volgende sectie behandelt meer ervaren onderwerpen zoals mailinstellingen en het instellen van mail voor het gehele domein.

26.6.1. Basisinstellingen

Het verzenden van email naar externe hosts zou onmiddellijk moeten werken, zolang /etc/resolv.conf is aangemaakt of zolang er een nameserver wordt gedraaid. Indien het gewenst is dat mail voor de host aan de MTA (bijvoorbeeld sendmail) geleverd dient te worden op de FreeBSD-host, zijn er twee methoden:

- Draai een eigen nameserver op een eigen domein, bijvoorbeeld `FreeBSD.org`
- Zorg ervoor dat mail direct aan de host geleverd wordt. Dit wordt gedaan door mail direct aan de huidige DNS-naam voor de machine, bijvoorbeeld `example.FreeBSD.org`, te leveren.

Onafhankelijk van de hierboven gekozen methode, dient de host, om er direct mail aan geleverd te

krijgen, een permanent statisch IP-adres te hebben (niet een dynamisch adres, zoals dat bij de meeste PPP-inbelverbindingen het geval is). Indien er een firewall actief is, dient het SMTP-verkeer naar de host door te geven. Indien het gewenst is dat de host direct mail ontvangt, dient één van de twee onderstaande dingen geregeld te zijn:

- Zorg ervoor dat de (laagstgenummerde) MX-regel in het DNS naar het IP-adres van de host wijst.
- Zorg ervoor dat er geen MX-regel in het DNS is voor de host.

Met elk van de bovenstaanden kan mail direct op de host ontvangen worden.

Probeer dit:

```
# hostname
example.FreeBSD.org
# host example.FreeBSD.org
example.FreeBSD.org has address 204.216.27.XX
```

Indien dit verschijnt, zal mail die direct naar yourlogin@example.FreeBSD.org zonder problemen moeten werken (aangenomen dat sendmail correct werkt op `example.FreeBSD.org`).

Indien in plaats daarvan zoiets als dit verschijnt:

```
# host example.FreeBSD.org
example.FreeBSD.org has address 204.216.27.XX
example.FreeBSD.org mail is handled (pri=10) by hub.FreeBSD.org
```

zal alle mail die naar de host (`example.FreeBSD.org`) verzameld worden op `hub` onder dezelfde gebruikersnaam in plaats van direct naar de host verstuurd te worden.

Bovenstaande informatie wordt door de DNS-server afgehandeld. De DNS-regel die informatie over het routen van mail bevat is de *M*ail e*X*change regel. Indien er geen MX-regel is, zal mail direct aan de host worden afgeleverd door middel van het IP-adres.

De MX-regel voor `freefall.FreeBSD.org` zag er eens als volgt uit:

```
freefall        MX  30  mail.crl.net
freefall        MX  40  agora.rdrop.com
freefall        MX  10  freefall.FreeBSD.org
freefall        MX  20  who.cdrom.com
```

Te zien is dat `freefall` vele MX-regels had. Het laagste MX-getal hoort bij de host die de mail direct ontvangt indien beschikbaar; indien het om een of andere reden niet beschikbaar is, accepteren de anderen (soms " reserve-MXs" genoemd) tijdelijk berichten en geven ze die door wanneer een lager-genummerde host beschikbaar is, om uiteindelijk aan de laagstgenummerde host af te leveren.

Alternatieve MX-sites zouden andere Internetverbindingen dan die van de host moeten hebben om

het nuttigst te zijn. De internetprovider of een andere vriendelijke site zouden geen problemen moeten hebben met het leveren van deze dienst.

26.6.2. Mail voor het domein

Om een "mailhost" (ook bekend als een mailserver) te installeren, is het nodig om mail die verzonden wordt naar de verschillende werkstations ernaar toe te leiden. In principe dient alle mail voor elke hostnaam in het domein (in dit geval `*.FreeBSD.org`) geclaimd te worden en naar de mailserver omgeleid te worden zodat gebruikers hun mail op de hoofdmailserver kunnen ontvangen.

Het gemakkelijkste is het indien er een gebruikersaccount met dezelfde *gebruikersnaam* op beide machines bestaat. Hiervoor dient adduser(8) gebruikt te worden.

De mailhost die het meest gebruikt zal worden is de toegewezen mailuitwisselaar voor elk werkstation in het netwerk. Dit wordt in de DNS-instellingen als volgt gedaan:

```
example.FreeBSD.org A   204.216.27.XX   ; werkstation
        MX  10  hub.FreeBSD.org ; mailhost
```

Dit zal mail voor het werkstation naar de mailhost leiden onafhankelijk van waar de A-regel naar toe wijst. De mail wordt naar de MX-host verzonden.

Om dit te doen is het nodig om een eigen DNS-server te draaien. Neem, indien dit niet het geval is of het niet mogelijk is om een eigen DNS-server te draaien, contact op met degene die de DNS levert.

De volgende informatie is nuttig indien email virtueel gehost wordt. In dit voorbeeld wordt aangenomen dat er een klant is met een eigen domein, in dit geval `customer1.org`, en dat alle mail voor `customer1.org` naar de mailhost `mail.myhost.com` verzonden dient te worden. De regel in het DNS dient er als volgt uit te zien:

```
customer1.org       MX  10  mail.myhost.com
```

Het is *niet* nodig om een A-regel voor `customer1.org` te hebben als er voor dat domein alleen email afgehandeld dient te worden.



Let erop dat `customer1.org` pingen niet werkt tenzij er een A-regel voor bestaat.

Als laatste dient sendmail op de mailhost te weten voor welke domeinen en/of hostnamen het mail dient te accepteren. Er bestaan enkele verschillende manieren om dit te doen. Elk van de volgende manieren zal werken:

- Voeg de hosts toe aan het bestand /etc/mail/local-host-names indien `FEATURE(use_cw_literal)`. Indien er een versie van sendmail wordt gebruikt die ouder is dan 8.10, is het te gebruiken bestand /etc/sendmail.cw.
- Voeg een regel met `Cwyour.host.com` toe aan /etc/sendmail.cf of aan /etc/mail/sendmail.cf indien versie 8.10 of nieuwer van sendmail wordt gebruikt.

26.7. SMTP met UUCP

De instellingen van sendmail die met FreeBSD worden geleverd zijn ontworpen voor sites die een directe verbinding met het Internet hebben. Sites waarvoor de mail via UUCP willen uitwisselen dienen een ander instellingenbestand voor sendmail te installeren.

Het handmatig bijstellen van /etc/mail/sendmail.cf is een geavanceerd onderwerp. Versie 8 van sendmail genereert instellingenbestanden via m4(1) preprocessing, waarbij het eigenlijke instellen op een hoger abstractieniveau plaatsvindt. De instellingenbestanden voor m4(1) kunnen onder /usr/shared/sendmail/cf gevonden worden. Het bestand README in de map cf kan dienen als een basisintroductie tot het instellen van m4(1).

De beste manier om UUCP te ondersteunen is het gebruiken van de eigenschap `mailertable`. Dit maakt een database aan die sendmail kan gebruiken om beslissingen over routes te nemen.

Als eerste dient het .mc-bestand aangemaakt te worden. De map /usr/shared/sendmail/cf/cf bevat enkele voorbeelden. Indien het bestand foo.mc heet, hoeft slechts het volgende gedaan te worden om het in een geldig sendmail.cf om te zetten:

```
# cd /etc/mail
# make foo.cf
# cp foo.cf /etc/mail/sendmail.cf
```

Een typisch .mc-bestand kan er als volgt uitzien:

```
VERSIONID(`Uw versienummer') OSTYPE(bsd4.4)

FEATURE(accept_unresolvable_domains)
FEATURE(nocanonify)
FEATURE(mailertable, `hash -o /etc/mail/mailertable')

define(`UUCP_RELAY', uw.uucp.relay)
define(`UUCP_MAX_SIZE', 200000)
define(`confDONT_PROBE_INTERFACES')

MAILER(local)
MAILER(smtp)
MAILER(uucp)

Cw  uw.alias.host.naam
Cw  uwuucpnodenaam.UUCP
```

De regels die de eigenschappen `accept_unresolvable_domains`, `nocanonify`, en `confDONT_PROBE_INTERFACES` bevatten zorgen ervoor dat er geen gebruik wordt gemaakt van het DNS tijdens het afleveren van mail. De clausule `UUCP_REPLAY` is nodig om UUCP-aflevering te ondersteunen. Hier dient een hostnaam op het Internet ingevuld te worden die .UUCP pseudo-domeinadressen kan afhandelen, waarschijnlijk zal dit de mailrelay van de Internetprovider zijn.

Nadat dit gedaan is, is er een bestand /etc/mail/mailertable nodig. Indien er slechts één verbinding naar buiten is die voor alle mails gebruikt wordt, zal het volgende bestand volstaan:

```
#
# makemap hash /etc/mail/mailertable.db < /etc/mail/mailertable
.               uucp-dom:uw.uucp.relay
```

Een complexer voorbeeld kan er als volgt uitzien:

```
#
# makemap hash /etc/mail/mailertable.db < /etc/mail/mailertable
#
horus.interface-business.de    uucp-dom:horus
.interface-business.de         uucp-dom:if-bus
interface-business.de          uucp-dom:if-bus
.heep.sax.de                   smtp8:%1
horus.UUCP                     uucp-dom:horus
if-bus.UUCP                    uucp-dom:if-bus
.                              uucp-dom:
```

De eerste drie regels behandelen speciale gevallen waarbij domein-geadresseerde mail niet naar de standaardroute verzonden dient te worden, maar in plaats daarvan naar een UUCP-buur om het afleverpad "af te snijden". De volgende regel handelt mail naar het lokale Ethernetdomein die met SMTP afgeleverd kan worden af. Als laatste worden UUCP-buren in de .UUCP-pseudodomeinnotatie genoemd, om een `uucp-buur!ontvanger`-overname toe te staan. De laatste regel bestaat altijd uit een enkele punt, dat met al het andere matcht, met UUCP-aflevering naar een UUCP-buur die als universele mail-gateway naar de wereld dient. Alle nodenamen achter het sleutelwoord `uucp-dom:` dienen geldige UUCP-buren te zijn, dat met het commando `uuname` gecontroleerd kan worden.

Dit bestand dient naar een DBM-database omgezet te worden voor gebruik. De opdrachtregel om dit te doen kan het beste als commentaar bovenaan het bestand mailertable gezet worden. Deze opdracht dient telkens wanneer het bestand mailertable wordt gewijzigd uitgevoerd te worden.

Laatste tip: indien de werking van een zekere mailroute niet zeker is, kan de optie `-bt` van sendmail gebruikt worden. Het start sendmail in *adrestestmodus* op; voer `3,0` gevolgd door het adres dat voor de mailrouting getest dient te worden in. De laatste regel bevat de gebruikte interne mailagent, de bestemmingshost waarmee deze agent aangeroepen wordt, en het (mogelijk vertaalde) adres. Deze modus kan door het typen van `Ctrl` + `D` verlaten worden.

```
% sendmail -bt
ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter <ruleset> <address>
> 3,0 foo@example.com
canonify           input: foo @ example . com
...
parse            returns: $# uucp-dom $@ uw.uucp.relay $: foo < @ example . com . >
```

```
> ^D
```

26.8. Instellen om alleen te versturen

Er zijn veel gevallen waarbij het gewenst is om enkel mail te verzenden via een relay. Voorbeelden hiervan zijn:

- De computer is een desktop, maar het is gewenst om programma's als send-pr(1) te gebruiken. Hiervoor dient de mailrelay van de internetprovider gebruikt te worden.
- De computer is een server welke mail niet lokaal verwerkt, maar alle mail voor verwerking doorstuurt.

Zowat elke MTA kan deze specifieke taak vervullen. Helaas kan het erg moeilijk zijn om een MTA met alle mogelijkheden correct in te stellen om alleen uitgaande mail te behandelen. Programma's als sendmail en postfix zijn hiervoor grotendeels overbodig.

Ook kan het zijn dat de overeenkomst van een typisch internetabonnement het draaien van een "mail server" verbiedt.

De gemakkelijkste manier om aan deze behoeften te voldoen is door de port mail/ssmtp te installeren. Voer als `root` de volgende opdrachten uit:

```
# cd /usr/ports/mail/ssmtp
# make install replace clean
```

Eenmaal geïnstalleerd kan mail/ssmtp door middel van het vier-regelige bestand /usr/local/etc/ssmtp/ssmtp.conf ingesteld worden:

```
root=uwechteemail@example.com
mailhub=mail.example.com
rewriteDomain=example.com
hostname=_HOSTNAME_
```

Let erop dat het echte emailadres voor `root` gebruikt wordt. Vervang `mail.example.com` door de uitgaande mail relay van de internetprovider (ook wel de "uitgaande mailserver" of "SMTP-server" genoemd).

Let erop dat sendmail uitgeschakeld wordt, inclusief de uitgaande maildienst. Raadpleeg sendmail uitschakelen voor details.

mail/ssmtp heeft nog meer mogelijkheden. Raadpleeg het voorbeeldinstelbestand /usr/local/etc/ssmtp of de hulppagina van ssmtp voor enkele voorbeelden en meer informatie.

Door ssmtp op deze manier in te stellen kan alle software op de computer welke mail dient te versturen correct functioneren, zonder dat het beleid van de internetprovider geschonden wordt of dat de computer gekaapt kan worden om spam mee te versturen.

26.9. Mail gebruiken met een inbelverbinding

Indien het IP-adres statisch is, is het niet nodig om de standaardwaarden aan te passen. De toegewezen Internetnaam dient als hostnaam gebruikt te worden waarna sendmail de rest kan doen.

Indien het IP-adres dynamisch is en er een inbelverbinding naar het Internet gebruikt wordt, is de postbus waarschijnlijk op de mailserver van de Internetprovider geplaatst. Stel dat het domein van de Internetprovider `example.net` is, dat de gebruikersnaam `gebruiker` is, dat de machine `bsd.home` is, en dat volgens de Internetprovider `relay.example.net` als mailrelay gebruikt kan worden.

Om mail van de postbus te ontvangen, dient er een ontvangstagent geïnstalleerd te worden. Het gereedschap fetchmail is een goede keuze omdat het veel verschillende protocollen ondersteunt. Dit programma is als pakket of vanuit de Portscollectie (mail/fetchmail) beschikbaar. Normaliter levert de Internetprovider POP. Indien gebruikers-PPP gebruikt wordt, kan de mail automatisch worden opgehaald wanneer er een verbinding met Internet tot stand is gebracht door middel van de volgende regel in /etc/ppp/ppp.linkup:

```
MYADDR:
!bg su gebruiker -c fetchmail
```

Indien sendmail gebruikt wordt (zoals hieronder te zien is) om mail aan niet-lokale accounts af te leveren, is het waarschijnlijk gewenst dat sendmail de mailrij verwerkt zodra er een Internetverbinding tot stand is gebracht. Hiervoor dient de volgende opdracht na de `fetchmail` -opdracht in /etc/ppp/ppp.linkup geplaatst te worden:

```
  !bg su gebruiker -c "sendmail -q"
```

Aangenomen wordt dat er een account voor `gebruiker` op `bsd.home` aanwezig is. In de thuismap van `gebruiker` op `bsd.home` dient een bestand .fetchmailrc aangemaakt te worden:

```
poll example.net protocol pop3 fetchall pass MijnGeheim
```

Dit bestand dient alleen voor `gebruiker` leesbaar te zijn aangezien dit bestand het wachtwoord `MijnGeheim` bevat.

Om mail met de correcte `from:`-header te versturen, dient sendmail gebruiker@example.net in plaats van gebruiker@bsd.home te gebruiken. Het kan ook wenselijk zijn om sendmail alle mail via `relay.example.net` te versturen, om sneller mail te verzenden.

Het volgende .mc zou voldoende moeten zijn:

```
VERSIONID(`bsd.home.mc version 1.0')
OSTYPE(bsd4.4)dnl
FEATURE(nouucp)dnl
MAILER(local)dnl
```

```
MAILER(smtp)dnl
Cwlocalhost
Cwbsd.home
MASQUERADE_AS(`example.net')dnl
FEATURE(allmasquerade)dnl
FEATURE(masquerade_envelope)dnl
FEATURE(nocanonify)dnl
FEATURE(nodns)dnl
define(`SMART_HOST', `relay.example.net')
Dmbsd.home
define(`confDOMAIN_NAME', `bsd.home')dnl
define(`confDELIVERY_MODE', `deferred')dnl
```

In de vorige sectie staan de details over het omzetten van een .mc-bestand in bestand sendmail.cf. Ook dient sendmail herstart te worden na het wijzigen van sendmail.cf.

26.10. SMTP-authenticatie

Het hebben van SMTP-authenticatie op een mailserver heeft een aantal voordelen. SMTP-authenticatie kan een extra beveiligingslaag toevoegen aan sendmail, en het geeft mobiele gebruikers die van hosts wisselen de mogelijkheid om dezelfde mailserver te gebruiken zonder dat ze telkens de instellingen van hun mailclient moeten veranderen.

1. Installeer security/cyrus-sasl2 vanuit de ports. Deze port is te vinden in security/cyrus-sasl2. De port security/cyrus-sasl2 ondersteund een aantal opties tijdens de compilatie. Voor de SMTP-authenticatiemethode die hier gebruikt wordt, dient de optie `LOGIN` te zijn uitgezet.
2. Voeg nadat security/cyrus-sasl2 is geïnstalleerd deze regel toe aan /usr/local/lib/sasl2/Sendmail.conf:

   ```
   pwcheck_method: saslauthd
   ```

3. Installeer vervolgens security/cyrus-sasl2-saslauthd, en voeg de volgende regel toe aan /etc/rc.conf:

   ```
   saslauthd_enable="YES"
   ```

   en start vervolgens het saslauthd-daemon op:

   ```
   # service saslauthd start
   ```

   Deze daemon fungeert als een onderhandelaar voor sendmail die zich tegen de FreeBSD passwd-database authenticeert. Dit bespaart de moeite van het opnieuw creëren van een nieuwe verzameling gebruikersnamen en wachtwoorden voor elke gebruiker die SMTP-

authenticatie nodig heeft, en het houdt de wachtwoorden voor het inloggen en de mail hetzelfde.

4. Voeg de volgende regels toe aan /etc/make.conf:

```
SENDMAIL_CFLAGS=-I/usr/local/include/sasl -DSASL
SENDMAIL_LDFLAGS=-L/usr/local/lib
SENDMAIL_LDADD=-lsasl2
```

   Deze regels geven sendmail de juiste instelopties om tijdens het compileren met cyrus-sal2 te linken. Zorg ervoor dat cyrus-sasl2 is geïnstalleerd voordat sendmail wordt gehercompileerd.

5. Hercompileer sendmail door de volgende opdrachten uit te voeren:

```
# cd /usr/src/lib/libsmutil
# make cleandir && make obj && make
# cd /usr/src/lib/libsm
# make cleandir && make obj && make
# cd /usr/src/usr.sbin/sendmail
# make cleandir && make obj && make && make install
```

   Het compileren van sendmail zou geen problemen moeten geven indien /usr/src niet veel veranderd is en dat de benodigde gedeelde bibliotheken aanwezig zijn.

6. Nadat sendmail is gecompileerd en opnieuw is gecompileerd, dient /etc/mail/freebsd.mc (of het plaatselijke .mc-bestand) gewijzigd te worden. Veel beheerders kiezen ervoor om de uitvoer van hostname(1) als .mc-bestandsnaam te gebruiken vanwege de uniciteit. Voeg deze regels toe:

```
dnl set SASL options
TRUST_AUTH_MECH(`GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN')dnl
define(`confAUTH_MECHANISMS', `GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN')dnl
```

   Deze opties stellen de verschillende beschikbare methoden voor sendmail in om gebruikers te authenticeren. Gebruik de bijgeleverde documentatie indien een andere methode dan pwcheck gewenst is.

7. Voer als laatste make(1) in /etc/mail uit. Hierdoor wordt het nieuwe .mc-bestand uitgevoerd en wordt een bestand freebsd.cf (of de plaatselijke variant ervan) aangemaakt. Voer hierna de opdracht `make install restart` uit, wat het bestand naar sendmail.cf kopieert en sendmail op de juiste manier herstart. In /etc/mail/Makefile staat meer informatie over dit proces.

Indien alles goed is gegaan, moet het mogelijk zijn om de inloginformatie in de mailclient in te voeren en een testbericht te versturen. Zet voor verdere onderzoekingen de `LogLevel` van sendmail

op 13 en houdt /var/log/maillog in de gaten voor foutmeldingen.

Refereer naar de sendmail-pagina betreffende SMTP-authenticatie voor meer informatie.

26.11. Mail User Agents

Een mail user agent (MUA) is een toepassing die wordt gebruikt om email te versturen en te ontvangen. Bovendien, omdat email "evolueert" en steeds complexer wordt, worden MUAs steeds krachtiger in de manier waarop ze met email omgaan; dit biedt gebruikers verhoogde functionaliteit en flexibiliteit. FreeBSD ondersteunt verschillende mail user agents die allemaal eenvoudig geïnstalleerd kunnen worden door de FreeBSD Ports Collectie te gebruiken. Gebruikers kunnen kiezen tussen grafische emailclients zoals evolution of balsa, op de console gebaseerde clients zoals mutt, alpine of `mail`, of de webinterface die door sommige grote organisaties wordt gebruikt.

26.11.1. mail

mail(1) is de standaard mail user agent (MUA) in FreeBSD. Het is een consolegebaseerde MUA die alle basisfunctionaliteit biedt die nodig is om tekstgebaseerde email te verzenden en te ontvangen, maar het is beperkt in de mogelijkheden om met bijlagen om te gaan en het ondersteunt alleen plaatselijke postbussen.

Hoewel `mail` van huis uit geen ondersteuning voor POP- of IMAP -servers biedt, kunnen deze postbussen gedownload worden naar een lokaal mbox-bestand door een toepassing als fetchmail te gebruiken, welke later in dit hoofdstuk behandeld wordt (fetchmail gebruiken).

Draai `mail` om email te versturen en te ontvangen:

```
% mail
```

De inhoud van de gebruikerspostbus in /var/mail wordt automatisch gelezen door het programma `mail`. Indien de postbus leeg is, eindigt het programma het een melding dat er geen mail gevonden kon worden. Wanneer de postbus is gelezen, wordt de applicatie-interface gestart, en wordt er een berichtenlijst weergegeven. Berichten worden automatisch genummerd, zoals in het volgende voorbeeld te zien is:

```
Mail version 8.1 6/6/93.  Type ? for help.
"/var/mail/marcs": 3 messages 3 new
>N  1 root@localhost        Mon Mar  8 14:05  14/510   "test"
 N  2 root@localhost        Mon Mar  8 14:05  14/509   "user account"
 N  3 root@localhost        Mon Mar  8 14:05  14/509   "sample"
```

Berichten kunnen nu worden gelezen door middel van het commando `t` van `mail`, gevolgd door het gewenste berichtnummer. In dit voorbeeld wordt de eerste email gelezen:

```
& t 1
```

```
Message 1:
From root@localhost  Mon Mar  8 14:05:52 2004
X-Original-To: marcs@localhost
Delivered-To: marcs@localhost
To: marcs@localhost
Subject: test
Date: Mon,  8 Mar 2004 14:05:52 +0200 (SAST)
From: root@localhost (Charlie Root)

This is a test message, please reply if you receive it.
```

Zoals in bovenstaand voorbeeld te zien is, zorgt de toets `t` ervoor dat het bericht met volledige headers wordt getoond. Om de berichtenlijst nogmaals weer te geven, dient de toets `h` gebruikt te worden.

Er kan met `mail` op een email gereageerd worden, door gebruik te maken één van de toetsen `R` of `r`. De toets `R` vertelt `mail` dat er alleen aan de verzender van het bericht geantwoord dient te worden, terwijl de toets `r` niet alleen aan de verzender antwoordt, maar ook aan andere ontvangers van het bericht. Het is ook mogelijk om achter deze commando's het berichtnummer te plaatsen waarop gereageerd dient te worden. Nadat dit gedaan is , dient het antwoord gegeven te worden, en dient het einde van het bericht aangegeven te worden met een enkele `.` op een nieuwe regel. Een voorbeeld staat hieronder:

```
& R 1
To: root@localhost
Subject: Re: test

Thank you, I did get your email.
.
EOT
```

Om een nieuwe email te verzenden, dient de toets `m` gebruikt te worden, gevolgd door het adres van de ontvanger. Er kunnen meerdere ontvangers gespecificeerd worden door ze met een `,` te scheiden. Hierna kan het onderwerp van het bericht worden gegeven, gevolgd door de inhoud van het bericht. Het einde van het bericht dient te worden aangegeven door een enkele `.` op een nieuwe regel te plaatsen.

```
& mail root@localhost
Subject: I mastered mail

Now I can send and receive email using mail ... :)
.
EOT
```

Binnen het programma `mail` kan op elk moment de opdracht `?` gebruikt worden om hulp weer te geven, hiervoor kan ook de hulppagina mail(1) worden geraadpleegd.



Zoals eerder is aangegeven, is het programma mail(1) van origine niet ontworpen om met bijlagen om te gaan, dus behandelt het deze slecht. Nieuwere MUAs zoals mutt gaan veel intelligenter met bijlagen om. Maar indien het programma `mail` nog steeds geprefereerd wordt, kan de port converters/mpack van aanzienlijk nut zijn.

26.11.2. mutt

mutt is een kleine doch zeer krachtige mail user agent, met uitstekende mogelijkheden, waaronder:

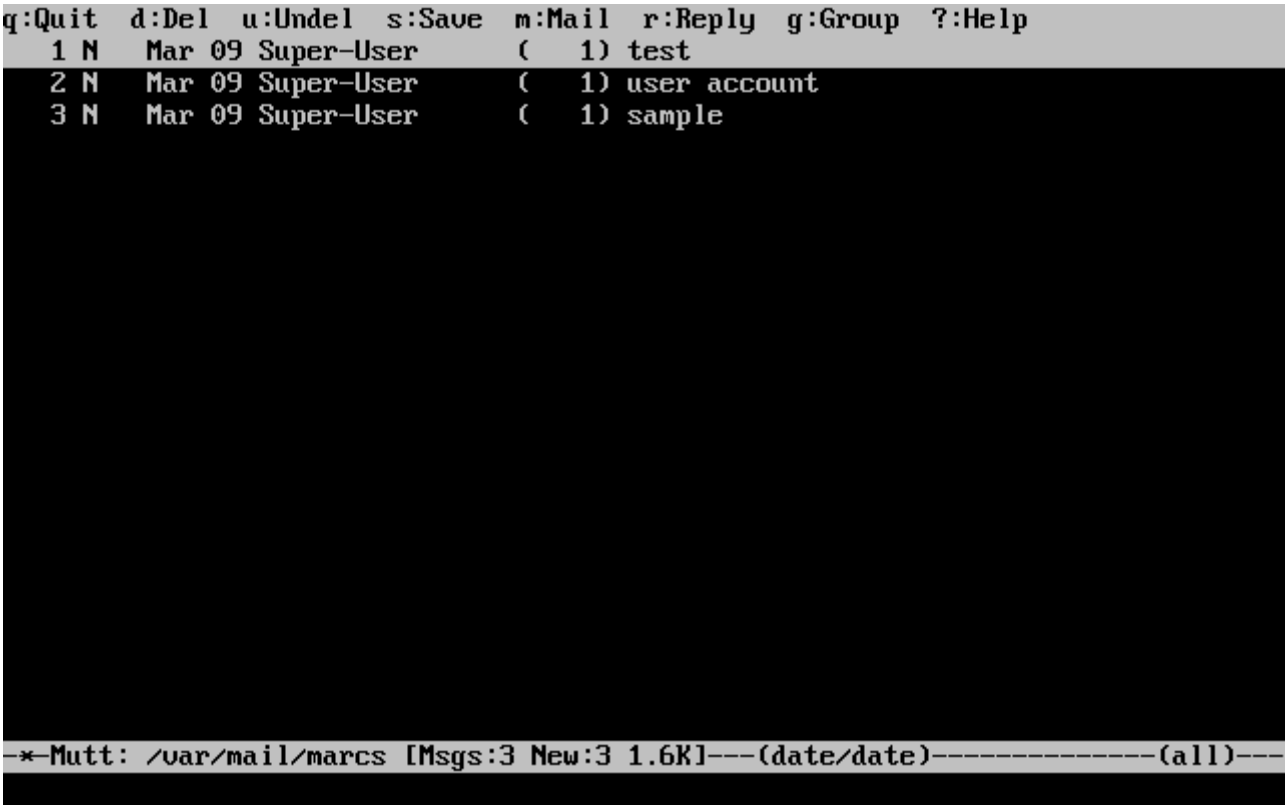
- De mogelijkheid om berichten te threaden;
- PGP-ondersteuning voor het digitaal ondertekenen en versleutelen van email;
- MIME-ondersteuning;
- Maildir-ondersteuning;
- Erg goed aan te passen.

Al deze eigenschappen zorgen ervoor dat mutt een van de meest geavanceerde beschikbare mail user agents is. Op http://www.mutt.org staat meer informatie.

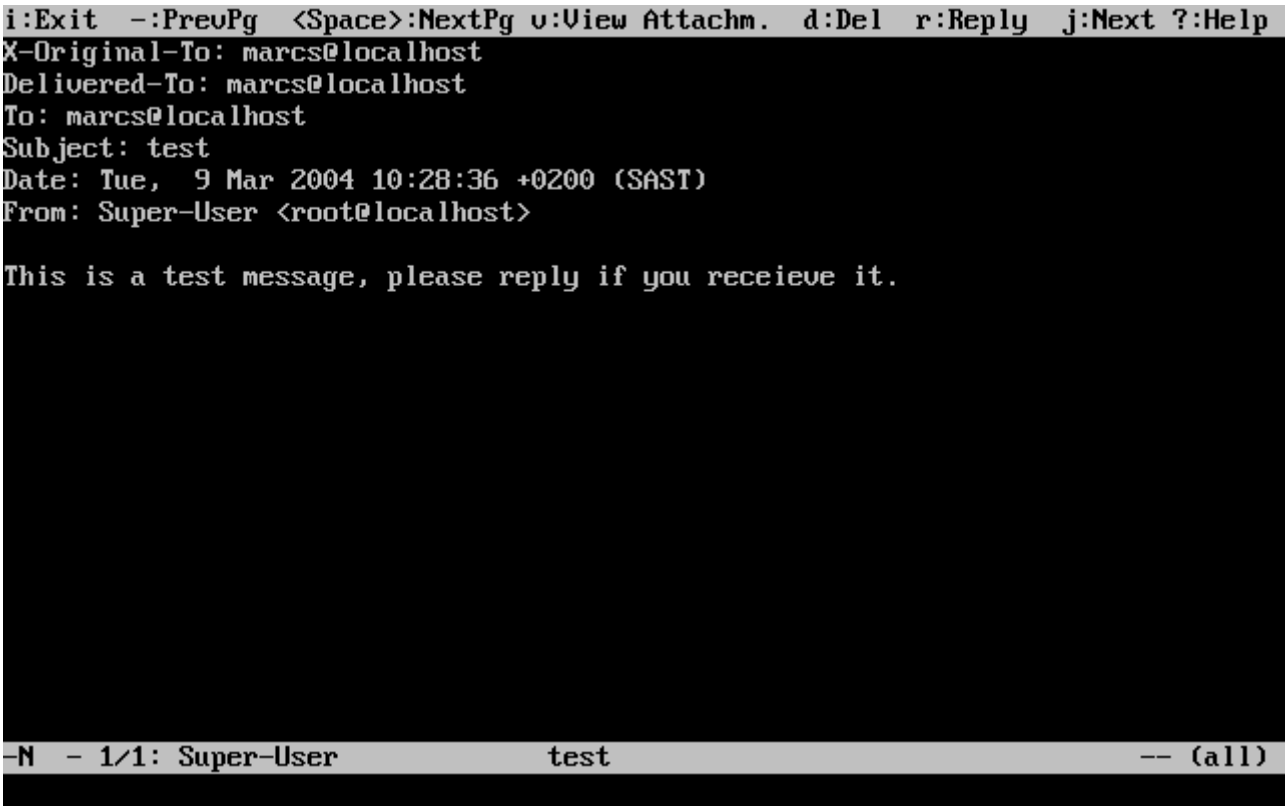
De stabiele versie van mutt kan geïnstalleerd worden door de port mail/mutt te gebruiken, terwijl de huidige ontwikkelaarsversie geïnstalleerd kan worden via de port mail/mutt-devel. Nadat de port is geïnstalleerd, kan mutt gestart worden met het volgende commando:

```
% mutt
```

mutt zal automatisch de inhoud van de gebruikerspostbus in /var/mail lezen en de inhoud weergeven indien van toepassing. Indien er geen mails gevonden zijn in de gebruikerspostbus, zal mutt wachten voor opdrachten van de gebruiker. Het onderstaande voorbeeld laat zien hoe mutt een lijst berichten weergeeft:



Selecteer om een email te lezen deze met de cursortoetsen, en sla de toets `Enter` aan. Een voorbeeld waarbij mutt email laat zien staat hieronder:



Net zoals het commando mail(1) staat mutt gebruikers toe om alleen de afzender alsook alle ontvangers te beantwoorden. Om alleen de afzender van de email te antwoorden, wordt de toets `r` gebruikt. Om aan een groep te antwoorden, welke aan zowel de originele afzender als aan alle berichtontvangers wordt gestuurd, wordt de toets `g` gebruikt.



mutt maakt gebruikt van het programma vi(1) als tekstverwerker voor het aanmaken en beantwoorden van emails. De gebruiker kan dit aanpassen door een eigen .muttrc aan te maken in hun thuismap en de variabele `editor` of de omgevingsvariabele `EDITOR` aan te passen. Zie http://www.mutt.org/ voor meer informatie over het instellen van mutt.

Voor het opstellen van een nieuw mailbericht wordt de toets `m` gebruikt. Nadat er een geldig bericht is gegeven, start mutt vi(1) op en kan de mail geschreven worden. Nadat de inhoud van de mail is geschreven, zal mutt nadat `vi` verlaten is, zichzelf hervatten en een overzichtsscherm van de te verzenden mail afbeelden. Om de mail te versturen wordt de toets `y` gebruikt. Een voorbeeld van het overzichtsscherm is hieronder te zien:

```
y:Send  q:Abort  t:To  c:CC  s:Subj  a:Attach file  d:Descrip  ?:Help
    From: Marc Silver <marcs@localhost>
      To: Super-User <root@localhost>
      Cc:
     Bcc:
 Subject: Re: test
Reply-To:
     Fcc:
Security: Clear


-- Attachments
- I     1 /tmp/mutt-bsd-c0hobscQ              [text/plain, 7bit, us-ascii, 1.1K]




-- Mutt: Compose  [Approx. msg size: 1.1K   Atts: 1]----------------------------
```

mutt bevat ook uitgebreide hulp, welke in de meeste menu's geactiveerd kan worden door de toets `?` aan te slaan. De bovenste regel geeft ook de relevante toetsen aan.

26.11.3. alpine

alpine richt zich op de beginnende gebruiker, maar bevat ook geavanceerde mogelijkheden.



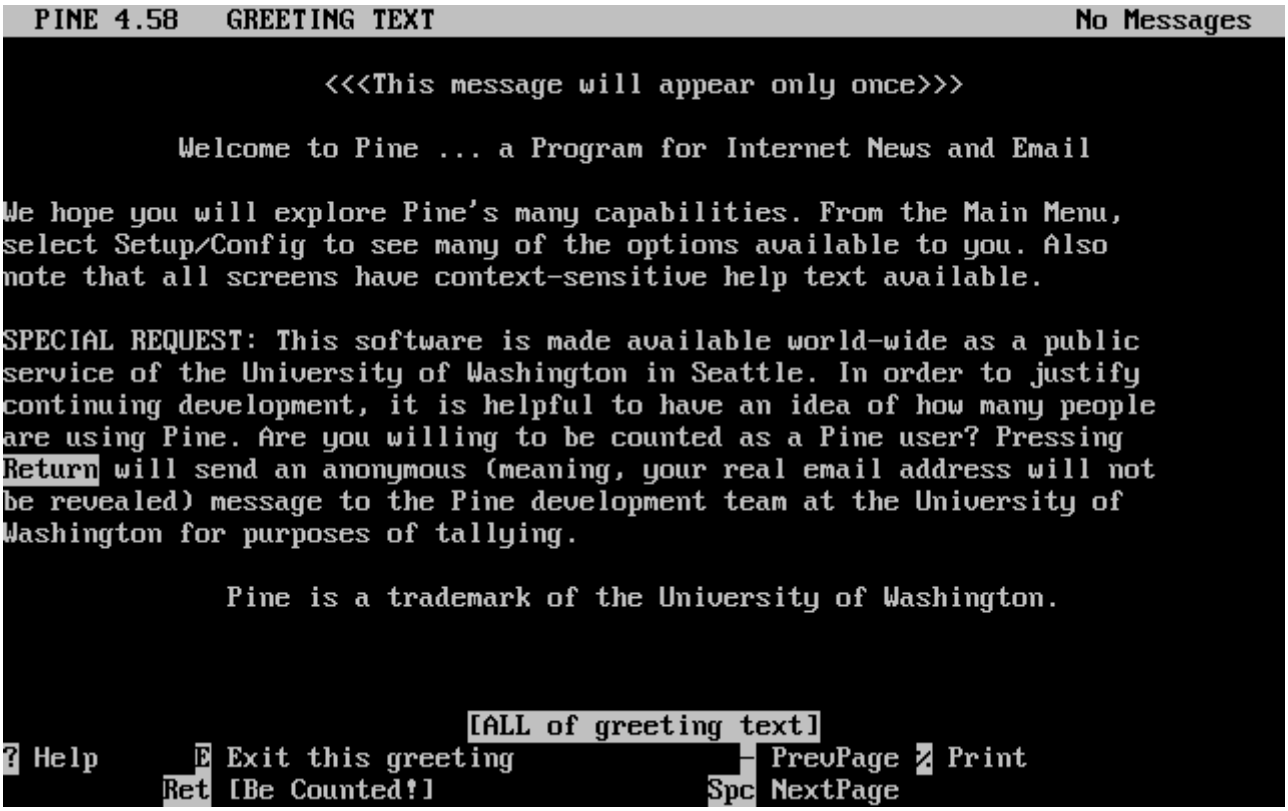
Er zijn in het verleden verschillende kwetsbaarheden voor alpine ontdekt, welke aanvallers op afstand in staat stelden om willekeurige code als gebruikers op het lokale systeem uit te voeren, door een speciaal voorbereide email te versturen. Alle *bekende* problemen van dit type zijn gerepareerd, maar de code van alpine is op een zeer onveilige manier geschreven en de beveiligingsofficier van FreeBSD gelooft dat het waarschijnlijk is dat er nog meer onontdekte kwetsbaarheden zijn. Installeer alpine op eigen risico.

De huidige versie van alpine kan door middel van de port mail/alpine geïnstalleerd worden.

Wanneer de port geïnstalleerd is, kan alpine met het volgende commando gestart worden:

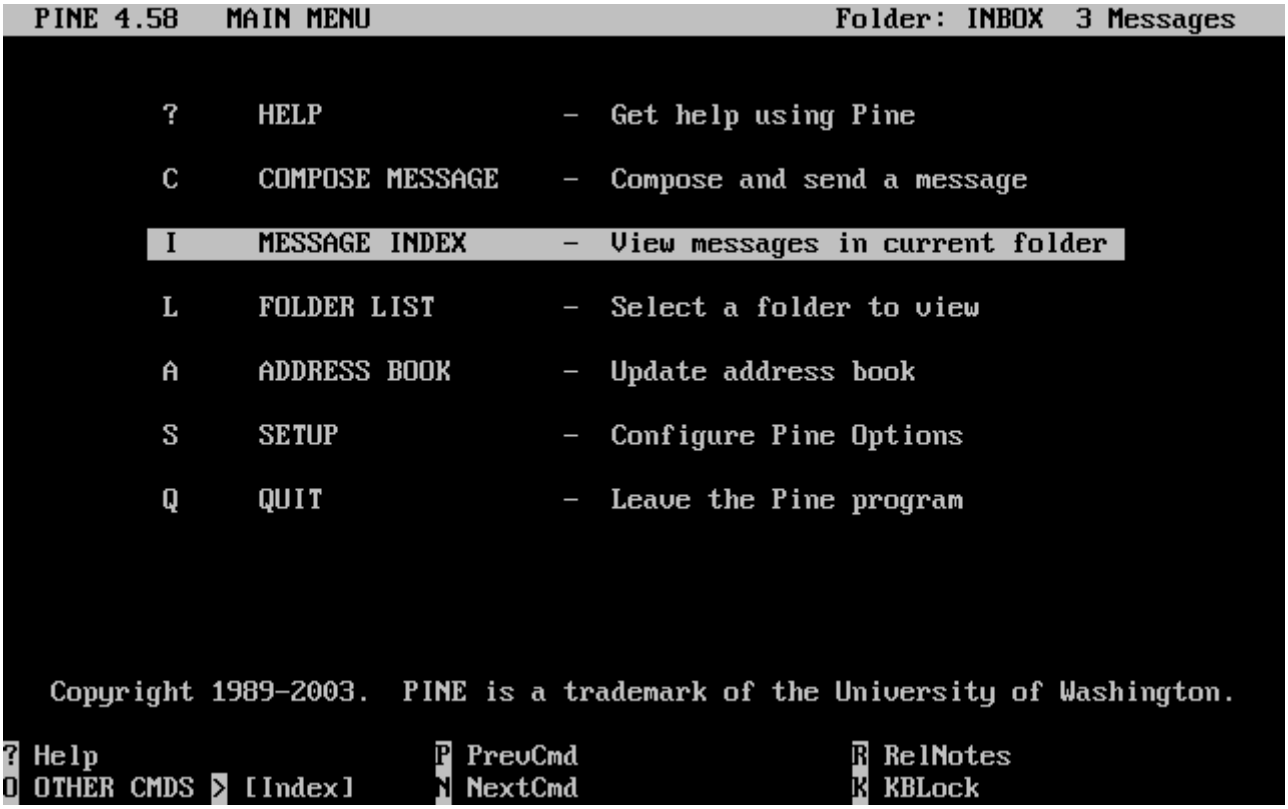
```
% alpine
```

De eerste keer dat alpine wordt gedraaid geeft het een welkomstpagina met een korte introductie weer, alsmede een verzoek van het ontwikkelteam van alpine om een anoniem emailbericht te versturen wat ze in staat stelt om te beoordelen hoeveel gebruikers hun client gebruiken. Druk op `Enter` om dit anonieme bericht te versturen, of druk op `E` om het welkomstscherm te verlaten zonder een anoniem bericht te versturen. Een voorbeeld van het welkomstscherm is hieronder te zien:



Vervolgens wordt het hoofdmenu getoond, waarin gemakkelijk met de cursortoetsen kan worden genavigeerd. Dit hoofdmenu biedt afkortingen voor het schrijven van nieuwe mail, het doorbladeren van mailmappen, en zelfs het beheren van het adresboek. Onder het hoofdmenu worden relevante toetscombinaties voor de huidige taak getoond.

De standaardmap die door alpine wordt geopend is de inbox. Gebruik de toets `I` om de berichtenindex te zien, of selecteer de optie MESSAGE INDEX zoals hieronder te zien is:



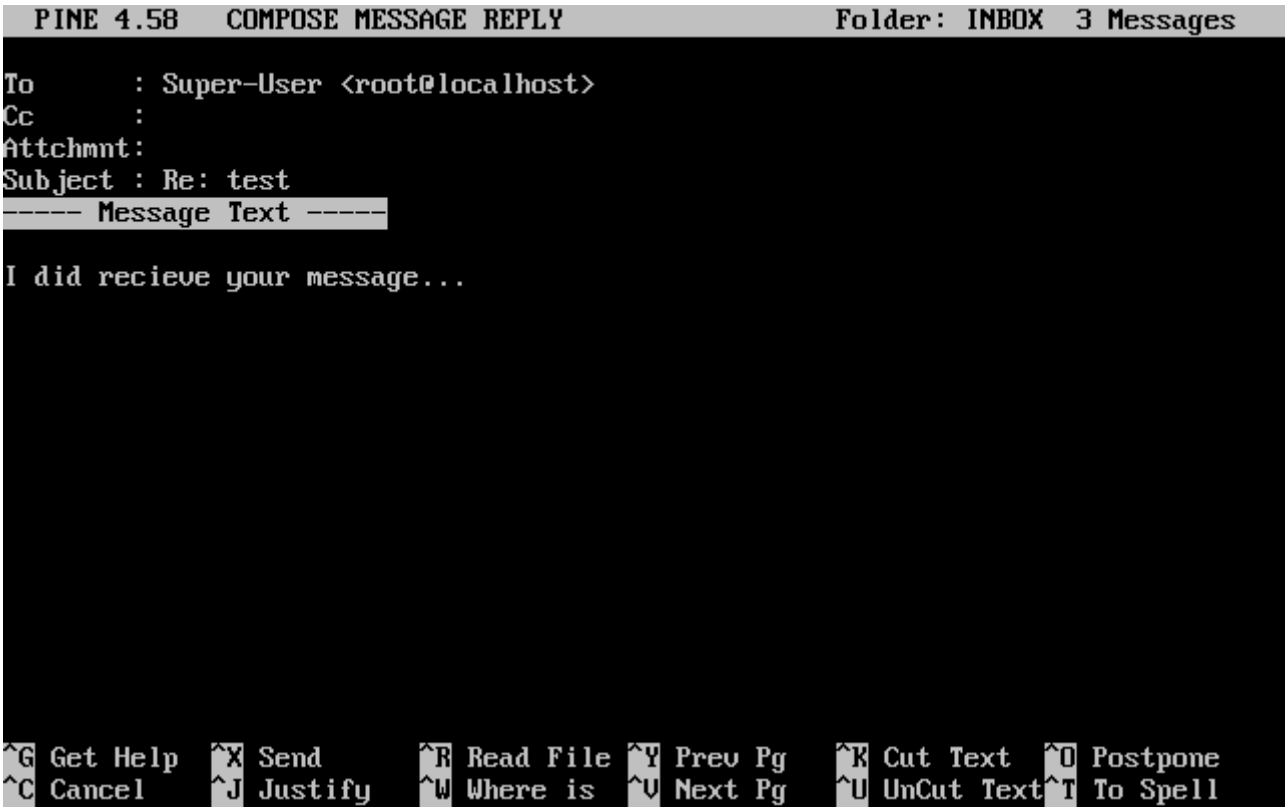
De berichtenindex geeft de berichten in de huidige map weer, en kan met de cursortoetsen worden genavigeerd. Gemarkeerde berichten kunnen worden gelezen door op `Enter` te drukken.



In onderstaand screenshot wordt een voorbeeldbericht door alpine weergegeven. Toetsencombinaties worden ter referentie aan de onderkant van het scherm weergegeven. Een voorbeeld van een van deze combinaties is de toets `r`, welke de MUA vertelt op het huidige bericht te antwoorden.



Voor het beantwoorden van een bericht wordt in alpine gebruikt gemaakt van de tekstverwerker pico, welke standaard bij alpine wordt geïnstalleerd. Het programma pico maakt het gemakkelijk om in het bericht te navigeren en is meer vergevingsgezind voor nieuwe gebruikers dan vi(1) of mail(1). Wanneer het antwoord voltooid is, kan het bericht worden verzonden door `Ctrl` + `X` te gebruiken. Het programma alpine zal om bevestiging vragen.



Het programma alpine kan worden aangepast door de optie SETUP van het hoofdmenu te gebruiken. Raadpleeg http://www.washington.edu/pine/ voor meer informatie.

26.12. fetchmail gebruiken

fetchmail is een volwaardige client voor IMAP en POP welke gebruikers in staat stelt om automatisch mail van IMAP- en POP-servers op afstand naar plaatselijke postbussen te downloaden; daar kan het gemakkelijker worden benaderd. fetchmail kan met de port mail/fetchmail worden geïnstalleerd, en biedt verschillende mogelijkheden, waaronder:

- Ondersteuning voor POP3, APOP, KPOP, IMAP, ETRN, en ODMR protocollen.
- De mogelijkheid om mail via SMTP door te sturen, wat filteren, doorsturen, en aliassen toestaat om normaal te functioneren.
- Kan in daemon-modus gedraaid worden om periodiek op nieuwe berichten te controleren.
- Kan verschillende postbussen ophalen en ze afhankelijk van de instellingen naar verschillende plaatselijke gebruikers doorsturen.

Hoewel het niet de bedoeling van dit document is om alle mogelijkheden van fetchmail uit te leggen, zullen sommige basismogelijkheden worden uitgelegd. Het gereedschap fetchmail heeft een instellingenbestand .fetchmailrc nodig om correct te kunnen werken. Dit bestand bevat zowel informatie over de server als de inloggegevens. Vanwege de gevoelige aard van de inhoud van dit bestand is het aan te raden om het met het volgende commando alleen leesbaar te maken voor de eigenaar ervan :

```
% chmod 600 .fetchmailrc
```

Het volgende .fetchmailrc dient als een voorbeeld voor het downloaden van een postbus van een enkele gebruiker via POP. Het vertelt fetchmail om met `example.com` te verbinden als gebruiker `joesoap` met wachtwoord `XXX`. Dit voorbeeld gaat ervan uit dat de gebruiker `joesoap` ook een gebruiker is op het plaatselijke systeem.

```
poll example.com protocol pop3 username "joesoap" password "XXX"
```

Het volgende voorbeeld legt verbinding met meerdere POP- en IMAP-servers en stuurt de mail door naar verschillende plaatselijke gebruikers indien van toepassing:

```
poll example.com proto pop3:
user "joesoap", with password "XXX", is "jsoap" here;
user "andrea", with password "XXXX";
poll example2.net proto imap:
user "john", with password "XXXXX", is "myth" here;
```

Het gereedschap fetchmail kan in daemon-modus worden gedraaid met de vlag `-d` gevolgd door het interval (in seconden) waarmee fetchmail de servers die in het bestand .fetchmailrc vermeld staan dient te vragen. Het volgende voorbeeld zorgt ervoor dat fetchmail elke 600 seconden vraagt:

```
% fetchmail -d 600
```

Meer informatie over fetchmail is te vinden op http://fetchmail.berlios.de/.

26.13. procmail gebruiken

Het gereedschap procmail is een zeer krachtig gereedschap voor het filteren van binnenkomende mail. Het stelt gebruikers in staat om "regels" te definiëren welke aan binnenkomende mail gekoppeld kunnen worden om specifieke taken uit te voeren of om de mail naar alternatieve postbussen en/of emailadressen door te sturen. procmail kan met de port mail/procmail geïnstalleerd worden. Eenmaal geïnstalleerd kan het direct met de meeste MTAs geïntegreerd worden; raadpleeg de documentatie van de MTA voor meer informatie. Als alternatief kan procmail geïntegreerd worden door de volgende regel aan het bestand .forward in de thuismap van de gebruiker die procmail gebruikt toe te voegen:

```
"|exec /usr/local/bin/procmail || exit 75"
```

De volgende sectie geeft wat basisregels van procmailmet een korte beschrijving ervan. Deze, en andere, regels dienen in het bestand .procmailrc geplaatst te worden, welke zich in de thuismap van de gebruiker dient te bevinden.

De meerderheid van deze regels kan ook in de hulppagina procmailex(5) gevonden worden.

Stuur alle mail van user@example.com door naar het externe adres goodmail@example2.com:

```
:0
* ^From.*user@example.com
! goodmail@example2.com
```

Stuur alle mails korten dan 1000 bytes door naar het externe adres goodmail@example2.com:

```
:0
* < 1000
! goodmail@example2.com
```

Stuur alle mail verzonden aan alternate@example.com door naar een postbus alternate:

```
:0
* ^TOalternate@example.com
alternate
```

Stuur alle mail met het onderwerp "Spam" door naar /dev/null:

```
:0
^Subject:.*Spam
/dev/null
```

Een handig recept dat binnenkomende `FreeBSD.org` mailinglijsten parseert en elke lijst in en eigen postbus plaatst:

```
:0
* ^Sender:.owner-freebsd-\/[^@]+@FreeBSD.ORG
{
    LISTNAME=${MATCH}
    :0
    * LISTNAME??^\/[^@]+
    FreeBSD-${MATCH}
}
```

Hoofdstuk 27. Netwerkdiensten

27.1. Overzicht

Dit hoofdstuk behandelt een aantal veelgebruikte netwerkdiensten op UNIX® systemen. Er wordt ingegaan op de installatie, het instellen, testen en beheren van verschillende typen netwerkdiensten. Overal in dit hoofdstuk staan voorbeeldbestanden met instellingen waar de lezer zijn voordeel mee kan doen.

Na het lezen van dit hoofdstuk weet de lezer:

- Hoe om te gaan met de inetd daemon;
- Hoe een netwerkbestandssysteem opgezet kan worden;
- Hoe een netwerkinformatiedienst (NIS) opgezet kan worden voor het delen van gebruikersaccounts;
- Hoe automatische netwerkinstellingen gemaakt kunnen worden met DHCP;
- Hoe een domeinnaam server opgezet kan worden;
- Hoe een Apache HTTP Server opgezet kan worden;
- Hoe een File Transfer Protocol (FTP) Server opgezet kan worden;
- Hoe een bestand-- en printserver voor Windows® cliënten opgezet kan worden met Samba;
- Hoe datum en tijd gesynchroniseerd kunnen worden en hoe een tijdserver opgezet kan worden met het NTP-protocol.
- Hoe het standaard log-daemon `syslogd` in te stellen om logs van hosts op afstand te accepteren.

Veronderstelde voorkennis:

- Basisbegrip van de scripts in /etc/rc;
- Bekend zijn met basis netwerkterminologie;
- Kennis van de installatie van software van derde partijen (Applicaties installeren. pakketten en ports).

27.2. De inetd"Super-Server"

27.2.1. Overzicht

inetd(8) wordt soms de "Internet Super-Server" genoemd, omdat het verbindingen voor meerdere diensten beheert. Als door inetd een verbinding wordt ontvangen, bepaalt die voor welk programma de verbinding bedoeld is, splitst het dat proces af en delegeert de socket (het programma wordt gestart met de socket van de dienst als zijn standaardinvoer, -uitvoer en -foutbeschrijvingen). Het draaien van inetd voor servers die niet veel gebruikt worden kan de algehele werklast verminderen in vergelijking met het draaien van elke daemon individueel in stand-alone modus.

inetd wordt primair gebruikt om andere daemons aan te roepen, maar het handelt een aantal triviale protocollen direct af, zoals chargen, auth en daytime.

In deze paragraaf worden de basisinstellingen van inetd behandeld met de opties vanaf de commandoregel en met het instellingenbestand /etc/inetd.conf.

27.2.2. Instellingen

inetd wordt gestart door het rc(8)-systeem. De optie `inetd_enable` staat standaard op `NO`, maar kan tijdens de installatie door sysinstall worden aangezet. Door het plaatsen van

```
inetd_enable="YES"
```

of

```
inetd_enable="NO"
```

in /etc/rc.conf wordt inetd bij het opstarten van een systeem wel of niet ingeschakeld. Het commando:

```
# service inetd rcvar
```

kan gedraaid worden om de huidige effectieve instellingen weer te geven.

Dan kunnen er ook nog een aantal commandoregelopties aan inetd meegegeven worden met de optie `inetd_flags`.

27.2.3. Commandoregelopties

Zoals de meeste serverdaemons heeft inetd een aantal opties die doorgegeven kunnen worden om het gedrag aan te passen. Zie de handleidingpagina inetd(8) voor een volledige lijst van de opties.

Opties kunnen door middel van de optie `inetd_flags` in /etc/rc.conf aan inetd worden doorgegeven. Standaard staat `inetd_flags` ingesteld op `-wW -C 60`, dat TCP-wrapping aanzet voor de diensten van inetd, en voorkomt dat elk enkelvoudig IP-adres enige dienst meer dan 60 keer per minuut opvraagt.

Ook al worden er hieronder rate-limiting opties besproken, beginnende gebruikers kunnen blij zijn met het feit dat deze parameters gewoonlijk niet hoeven te worden aangepast. Deze opties kunnen interessant zijn wanneer er een buitensporige hoeveelheid verbindingen worden opgezet. Een volledige lijst van opties staat in de hulppagina inetd(8).

-c maximum

Geeft het maximale aantal gelijktijdige verzoeken voor iedere dienst aan. De standaard is ongelimiteerd. Kan per dienst ter zijde geschoven worden met de parameter `max-child`.

-C rate

Geeft het maximale aantal keren aan dat een dienst vanaf een bepaald IP-adres per minuut aangeroepen kan worden. Kan per dienst ter zijde geschoven worden met de parameter `max-connections-per-ip-per-minute`.

-R rate

Geeft het maximale aantal keren aan dat een dienst per minuut aangeroepen kan worden. De standaard is 256. De instelling `0` geeft aan dat er geen limiet is.

-s maximum

Specificeert het maximaal aantal keer per minuut dat een dienst aangeroepen kan worden vanuit een enkelvoudig IP-adres; de standaard is onbeperkt. Kan worden overstemd op een per-dienst-basis met de parameter `max-child-per-ip`.

27.2.4. inetd.conf

De instellingen van inetd worden beheerd in /etc/inetd.conf.

Als er een wijziging wordt aangebracht in /etc/inetd.conf, dan kan inetd gedwongen worden om de instellingen opnieuw in te lezen door dit commando te draaien:

Voorbeeld 21. Het instellingenbestand van inetd herladen

```
# service inetd reload
```

Iedere regel in het bestand met instellingen heeft betrekking op een individuele daemon. Commentaar wordt vooraf gegaan door een `#`. De opmaak van elke regel van /etc/inetd.conf is als volgt:

```
service-name
socket-type
protocol
{wait|nowait}[/max-child[/max-connections-per-ip-per-minute[/max-child-per-ip]]]
user[:group][/login-class]
server-program
server-program-arguments
```

Een voorbeeldregel voor de daemon ftpd(8) met IPv4 kan eruit zien als:

```
ftp     stream  tcp     nowait  root    /usr/libexec/ftpd       ftpd -l
```

service-name

Dit is de dienstnaam van een daemon. Die moet overeenkomen met een dienst uit /etc/services. Hiermee kan de poort waarop inetd moet luisteren aangegeven worden. Als er een nieuwe dienst wordt gemaakt, moet die eerst in /etc/services gezet worden.

socket-type

Dit is `stream`, `dgram`, `raw` of `seqpacket`. `stream` moet gebruikt worden voor verbindingsgebaseerde TCP-daemons, terwijl `dgram` wordt gebruikt voor daemons die gebruik maken van het transportprotocol UDP.

protocol

Een van de volgende:

Protocol	Toelichting
tcp, tcp4	TCP IPv4
udp, udp4	UDP IPv4
tcp6	TCP IPv6
udp6	UDP IPv6
tcp46	Zowel TCP IPv4 als v6
udp46	Zowel UDP IPv4 als v6

{wait|nowait}[/max-child[/max-connections-per-ip-per-minute[/max-child-per-ip]]]

`wait|nowait` geeft aan of de daemon die door inetd wordt aangesproken zijn eigen sockets kan afhandelen of niet. `dgram` sockettypen moeten de optie `wait` gebruiken, terwijl streamsocket daemons, die meestal multi-threaded zijn, de optie `nowait` horen te gebruiken. `wait` geeft meestal meerdere sockets aan een daemon, terwijl `nowait` een kinddaemon draait voor iedere nieuwe socket.

Het maximum aantal kinddaemons dat inetd mag voortbrengen kan ingesteld worden met de optie `max-child`. Als een limiet van tien instanties van een bepaalde daemon gewenst is, dan zou er `/10` achter `nowait` gezet worden. Door `/0` wordt een onbeperkt aantal kinderen toegestaan.

Naast `max-child` zijn er nog twee andere opties waarmee het maximale aantal verbindingen van een bepaalde plaats naar een daemon ingesteld kan worden. `max-connections-per-ip-per-minute` beperkt het aantal verbindingen per minuut voor enig IP-adres, een waarde van tien betekent hier dat er van ieder IP-adres maximaal tien verbindingen naar een bepaalde dienst tot stand gebracht kunnen worden. `max-child-per-ip` beperkt het aantal kindprocessen dat namens enig IP-adres op enig moment gestart kan worden. Deze opties kunnen zijn nuttig om bedoeld en onbedoeld buitensporig bronnengebruik van en Denial of Service (DoS) aanvallen op een machine te voorkomen.

In dit veld is één van `wait` of `nowait` verplicht. `max-child`, `max-connections-per-ip-per-minute` en `max-child-per-ip` zijn optioneel.

Een stream-type multi-threaded daemon zonder één van de limieten `max-child`, `max-connections-per-ip-per-minute` of `max-child-per-ip` is eenvoudigweg: `nowait`.

Dezelfde daemon met een maximale limiet van tien daemons zou zijn: `nowait/10`.

Dezelfde instellingen met een limiet van twintig verbindingen per IP-adres per minuut en een totaal maximum van tien kinddaemons zou zijn: `nowait/10/20`.

Deze opties worden allemaal gebruikt door de standaardinstellingen van de daemon fingerd(8):

```
finger stream  tcp     nowait/3/10 nobody /usr/libexec/fingerd fingerd -s
```

Als afsluiting, een voorbeeld in dit veld met een maximum van 100 kinderen in totaal, met een maximum van 5 voor enig IP-adres zou zijn: `nowait/100/0/5`.

user

Dit is de gebruikersnaam waar een daemon onder draait. Daemons draaien meestal als de gebruiker `root`. Om veiligheidsredenen draaien sommige daemons onder de gebruiker `daemon` of de gebruiker met de minste rechten: `nobody`.

server-program

Het volledige pad van de daemon die uitgevoerd moet worden als er een verbinding wordt ontvangen. Als de daemon een dienst is die door inetd intern wordt geleverd, dan moet de optie `internal` gebruikt worden.

server-program-arguments

Deze optie werkt samen met de optie `server-program` en hierin worden de argumenten ingesteld, beginnend met `argv[0]`, die bij het starten aan de daemon worden meegegeven. Als `mijndaemon -d` de commandoregel is, dan zou `mijndaemon -d` de waarde van `server-program-arguments` zijn. Hier geldt ook dat als de daemon een interne dienst is, hier de optie `internal` moet worden.

27.2.5. Beveiliging

Afhankelijk van keuzes gemaakt tijdens de installatie, kunnen veel van de diensten van inetd standaard ingeschakeld zijn. Het is verstandig te overwegen om een daemon dat niet noodzakelijk is uit te schakelen. Plaats een `#` voor de daemon in /etc/inetd.conf en herlaad vervolgens de instellingen van inetd. Sommige daemons, zoals fingerd, zijn wellicht helemaal niet gewenst omdat ze informatie geven die nuttig kan zijn voor een aanvaller.

Sommige daemons zijn zich niet echt bewust van beveiliging en hebben lange of niet bestaande timeouts voor verbindingspogingen. Hierdoor kan een aanvaller langzaam veel verbindingen maken met een daemon en zo beschikbare bronnen verzadigen. Het is verstandig voor die daemons de limietopties `max-connections-per-ip-per-minute`, `max-child` of `max-child-per-ip` te gebruiken als ze naar uw smaak teveel verbindingen hebben.

TCP-wrapping staat standaard aan. Er staat meer informatie over het zetten van TCP-restricties op de verschillende daemons die door inetd worden aangesproken in hosts_access(5).

27.2.6. Allerlei

daytime, time, echo, discard, chargen en auth zijn allemaal interne diensten van inetd.

De dienst auth biedt identiteitsnetwerkdiensten en is tot op een bepaald niveau instelbaar, terwijl de anderen eenvoudigweg aan of uit staan.

Meer diepgaande informatie staat in inetd(8).

27.3. Netwerkbestandssysteem (NFS)

Het Netwerkbestandssysteem (Network File System) is een van de vele bestandssystemen die FreeBSD ondersteunt. Het staat ook wel bekend als NFS. Met NFS is het mogelijk om mappen en bestanden met anderen in een netwerk te delen. Door het gebruik van NFS kunnen gebruikers en programma's bij bestanden op andere systemen op bijna dezelfde manier als bij hun eigen lokale bestanden.

De grootste voordelen van NFS zijn:

- Lokale werkstations gebruiken minder schijfruimte omdat veel gebruikte data op één machine opgeslagen kan worden en nog steeds toegankelijk is voor gebruikers via het netwerk;
- Gebruikers hoeven niet op iedere machine een thuismap te hebben. Thuismappen kunnen op de NFS server staan en op het hele netwerk beschikbaar zijn;
- Opslagapparaten als floppydisks, CD-ROM drives en Zip® drives kunnen door andere machines op een netwerk gebruikt worden. Hierdoor kan het aantal drives met verwijderbare media in een netwerk verkleind worden.

27.3.1. Hoe NFS werkt

NFS bestaat uit tenminste twee hoofdonderdelen: een server en een of meer cliënten. De cliënt benadert de gegevens die op een servermachine zijn opgeslagen via een netwerk. Om dit mogelijk te maken moeten er een aantal processen ingesteld en gestart worden.

Op de server moeten de volgende daemons draaien:

Daemon	Beschrijving
nfsd	De NFS-daemon die verzoeken van de NFS cliënten afhandelt.
mountd	De NFS koppeldaemon die doorgestuurde verzoeken van nfsd(8) uitvoert.
rpcbind	Deze daemon geeft voor NFS-cliënten aan welke poort de NFS-server gebruikt.

Op de cliënt kan ook een daemon draaien: nfsiod. De daemon nfsiod handelt verzoeken van de NFS-server af. Dit is optioneel en kan de prestaties verbeteren, maar het is niet noodzakelijk voor een normale en correcte werking. Meer informatie staat in nfsiod(8).

27.3.2. NFS instellen

NFS instellen gaat redelijk rechtlijnig. Alle processen die moeten draaien kunnen meestarten bij het opstarten door een paar wijzigingen in /etc/rc.conf.

Op de NFS server dienen de volgende opties in /etc/rc.conf te staan:

```
rpcbind_enable="YES"
```

```
nfs_server_enable="YES"
mountd_flags="-r"
```

mountd start automatisch als de NFS server is ingeschakeld.

Op de cliënt dient de volgende optie in /etc/rc.conf te staan:

```
nfs_client_enable="YES"
```

In het bestand /etc/exports staat beschreven welke bestandssystemen NFS moet exporteren (soms heet dat ook wel delen of "sharen"). Iedere regel in /etc/exports slaat op een bestandssysteem dat wordt geëxporteerd en welke machines toegang hebben tot dat bestandssysteem. Samen met machines die toegang hebben, kunnen ook toegangsopties worden aangegeven. Er zijn veel opties beschikbaar, maar hier worden er maar een paar beschreven. Alle opties staan beschreven in exports(5).

Nu volgen een aantal voorbeelden voor /etc/exports:

Het volgende voorbeeld geeft een beeld van hoe een bestandssysteem te exporteren, hoewel de instellingen afhankelijk zijn van de omgeving en het netwerk. Om bijvoorbeeld de map /cdrom te exporteren naar drie machines die dezelfde domeinnaam hebben als de server (vandaar dat de machinenamen geef domeinachtervoegsel hebben) of in /etc/hosts staan. De vlag `-ro` exporteert het bestandssysteem als alleen-lezen. Door die vlag kan een ander systeem niet schrijven naar het geëxporteerde bestandssysteem.

```
/cdrom -ro host1 host2 host3
```

Het volgende voorbeeld exporteert /home naar drie hosts op basis van IP-adres. Dit heeft zin als er een privaat netwerk bestaat, zonder dat er een DNS server is ingesteld. Optioneel kan /etc/hosts gebruikt worden om interne hostnamen in te stellen. Er is meer informatie te vinden in hosts(5). Met de vlag `-alldirs` mogen submappen ook koppelpunten zijn. De submap wordt dan niet feitelijk aangekoppeld, maar de cliënt koppelt dan alleen de submappen aan die verplicht of nodig zijn.

```
/home  -alldirs  10.0.0.2 10.0.0.3 10.0.0.4
```

Het volgende voorbeeld exporteert /a zo dat twee cliënten uit verschillende domeinen bij het bestandssysteem mogen. Met de vlag `-maproot=root` mag de gebruiker op het andere systeem gegevens naar het geëxporteerde bestandssysteem schrijven als `root`. Als de vlag `-maproot=root` niet wordt gebruikt, dan kan een gebruiker geen bestanden wijzigen op het geëxporteerde bestandssysteem, zelfs niet als een gebruiker daar `root` is.

```
/a  -maproot=root  host.example.com box.example.org
```

Om een cliënt toegang te geven tot een geëxporteerd bestandssysteem, moet die cliënt daar rechten voor hebben. De cliënt moet daarvoor genoemd worden in /etc/exports.

In /etc/exports staat iedere regel voor de exportinformatie van één bestandssysteem naar één host. Per bestandssysteem mag een host maar één keer genoemd worden en mag maar één standaard hebben. Stel bijvoorbeeld dat /usr een enkel bestandssysteem is. Dan is de volgende /etc/exports niet geldig:

```
# Werkt niet als /usr 1 bestandssysteem is
/usr/src   client
/usr/ports client
```

Eén bestandssysteem, /usr, heeft twee regels waarin exports naar dezelfde host worden aangegeven, `client`. In deze situatie is de juiste instelling:

```
/usr/src /usr/ports  client
```

De eigenschappen van een bestandssysteem dat naar een bepaalde host wordt geëxporteerd moeten allemaal op één regel staan. Regels waarop geen cliënt wordt aangegeven worden behandeld als een enkele host. Dit beperkt hoe bestandssysteem geëxporteerd kunnen worden, maar dat blijkt meestal geen probleem te zijn.

Het volgende voorbeeld is een geldige exportlijst waar /usr en /exports lokale bestandssystemen zijn:

```
# Exporteer src en ports naar client01 en client02,
# maar alleen client01 heeft er rootprivileges
/usr/src /usr/ports -maproot=root    client01
/usr/src /usr/ports               client02
# De cliëntmachines hebben rootrechten en kunnen overal aankoppelen
# op /exports. Iedereen in de wereld kan /exports/obj als alleen-lezen aankoppelen.
/exports -alldirs -maproot=root      client01 client02
/exports/obj -ro
```

De daemon mountd moet gedwongen worden om het bestand /etc/exports te controleren steeds wanneer het is aangepast, zodat de veranderingen effectief kunnen worden. Dit kan worden bereikt door òfwel een HUP-signaal naar de draaiende daemon te sturen:

```
# kill -HUP `cat /var/run/mountd.pid`
```

of door het rc(8) script `mountd` met de juiste parameter aan te roepen:

```
# service mountd onereload
```

Raadpleeg Gebruik van rc met FreeBSD voor meer informatie over het gebruik van rc-scripts.

Het is ook mogelijk een machine te herstarten, zodat FreeBSD alles netjes in kan stellen, maar dat is

niet nodig. Het uitvoeren van de volgende commando's als `root` hoort hetzelfde resultaat te hebben.

Op de NFS server:

```
# rpcbind
# nfsd -u -t -n 4
# mountd -r
```

Op de NFS cliënt:

```
# nfsiod -n 4
```

Nu is alles klaar om feitelijk het netwerkbestandssysteem aan te koppelen. In de volgende voorbeelden is de naam van de server `server` en de naam van de cliënt is `client`. Om een netwerkbestandssysteem slechts tijdelijk aan te koppelen of om alleen te testen, kan een commando als het onderstaande als `root` op de cliënt uitgevoerd worden:

```
# mount server:/home /mnt
```

Hiermee wordt de map /home op de server aangekoppeld op /mnt op de cliënt. Als alles juist is ingesteld, zijn nu in /mnt op de cliënt de bestanden van de server zichtbaar.

Om een netwerkbestandssysteem iedere keer als een computer opstart aan te koppelen, kan het bestandssysteem worden toegevoegd aan het bestand /etc/fstab:

```
server:/home    /mnt    nfs rw  0   0
```

Alle beschikbare opties staan in fstab(5).

27.3.3. Op slot zetten

Voor sommige applicaties (b.v. mutt) is het nodig dat bestanden op slot staan om correct te werken. In het geval van NFS, kan rpc.lockd worden gebruikt voor het op slot zetten van bestanden. Voeg het volgende toe aan het bestand /etc/rc.conf op zowel de cliënt als de server om het aan te zetten (het wordt aangenomen dat de NFS-cliënt en -server reeds zijn geconfigureerd):

```
rpc_lockd_enable="YES"
rpc_statd_enable="YES"
```

Start de applicatie met:

```
# service lockd start
# service statd start
```

Als echt op slot zetten tussen de NFS-cliënten en de NFS-server niet nodig is, is het mogelijk om de NFS-cliënt bestanden lokaal op slot te laten zetten door `-L` aan mount_nfs(8) door te geven. In de handleidingpagina mount_nfs(8) staan verdere details.

27.3.4. Mogelijkheden voor gebruik

NFS is voor veel doeleinden in te zetten. Een aantal voorbeelden:

- Een aantal machines een CD-ROM of andere media laten delen. Dat is goedkoper en vaak ook handiger, bijvoorbeeld bij het installeren van software op meerdere machines;
- Op grote netwerken kan het praktisch zijn om een centrale NFS server in te richten, waarop alle thuismappen staan. Die thuismappen kunnen dan geëxporteerd worden, zodat gebruikers altijd dezelfde thuismap hebben, op welk werkstation ze ook aanmelden;
- Meerdere machines kunnen een gezamenlijke map /usr/ports/distfiles hebben. Dan is het mogelijk om een port op meerdere machines te installeren, zonder op iedere machine de broncode te hoeven downloaden.

27.3.5. Automatisch aankoppelen met amd

amd(8) (de automatic mounter daemon) koppelt automatisch netwerkbestandssystemen aan als er aan een bestand of map binnen dat bestandssysteem wordt gerefereerd. amd ontkoppelt ook bestandssystemen die een bepaalde tijd niet gebruikt worden. Het gebruikt van amd is een aantrekkelijk en eenvoudig alternatief ten opzichte van permanente koppelingen, die meestal in /etc/fstab staan.

amd werkt door zichzelf als NFS-server te koppelen aan de mappen /host en /net. Als binnen die mappen een bestand wordt geraadpleegd, dan zoekt amd de bijbehorende netwerkkoppeling op en koppelt die automatisch aan. /net wordt gebruikt om een geëxporteerd bestandssysteem van een IP-adres aan te koppelen, terwijl /host wordt gebruikt om een geëxporteerd bestandssysteem van een hostnaam aan te koppelen.

Het raadplegen van een bestand in /host/foobar/usr geeft amd aan dat die moet proberen de /usr export op de host `foobar` aan te koppelen.

Voorbeeld 22. Een export aankoppelen met amd

De beschikbare koppelingen van een netwerkhost zijn te bekijken met `showmount`. Om bijvoorbeeld de koppelingen van de host `foobar` te bekijken:

```
% showmount -e foobar
Exports list on foobar:
/usr                               10.10.10.0
/a                                 10.10.10.0
% cd /host/foobar/usr
```

Zoals in het bovenstaande voorbeeld te zien is, toont `showmount`/usr als een export. Als er naar de map /host/foobar/usr wordt gegaan, probeert amd de hostnaam `foobar` te resolven en de gewenste

export automatisch aan te koppelen.

amd kan gestart worden door de opstartscript door de volgende regel in /etc/rc.conf te plaatsen:

```
amd_enable="YES"
```

Er kunnen ook nog opties meegegeven worden aan amd met de optie `amd_flags`. Standaard staat `amd_flags` ingesteld op:

```
amd_flags="-a /.amd_mnt -l syslog /host /etc/amd.map /net /etc/amd.map"
```

In het bestand /etc/amd.map staan de standaardinstellingen waarmee exports aangekoppeld worden. In het bestand /etc/amd.conf staan een aantal van de meer gevorderde instellingen van amd.

In amd(8) en amd.conf(8) staat meer informatie.

27.3.6. Problemen bij samenwerking met andere systemen

Bepaalde Ethernet adapters voor ISA PC systemen kennen limieten die tot serieuze netwerkproblemen kunnen leiden, in het bijzonder met NFS. Dit probleem is niet specifiek voor FreeBSD, maar het kan op FreeBSD wel voor komen.

Het probleem ontstaat bijna altijd als (FreeBSD) PC-systemen netwerken met hoog presterende werkstations, zoals van Silicon Graphics, Inc. en Sun Microsystems, Inc. De NFS-koppeling werkt prima en wellicht lukken een aantal acties ook, maar dan ineens lijkt de server niet meer te reageren voor de cliënt, hoewel verzoeken van en naar andere systemen gewoon verwerkt worden. Dit gebeurt op een cliëntsysteem, of de cliënt nu het FreeBSD systeem is of het werkstation. Op veel systemen is er geen manier om de cliënt netjes af te sluiten als dit probleem is ontstaan. Vaak is de enige mogelijkheid een reset van de cliënt, omdat het probleem met NFS niet opgelost kan worden.

Hoewel de enige "correcte" oplossing de aanschaf van een snellere en betere Ethernet adapter voor het FreeBSD systeem is, is er zo om het probleem heen te werken dat het werkbaar is. Als FreeBSD de *server* is, kan de optie `-w=1024` gebruikt worden bij het aankoppelen door de cliënt. Als het FreeBSD systeem de *cliënt* is, dan dient het NFS-bestandssysteem aangekoppeld te worden met de optie `r=1024`. Deze opties kunnen het vierde veld zijn in een regel in fstab voor automatische aankoppelingen en bij handmatige aankoppelingen met mount(8) kan de parameter `-o` gebruikt worden.

Soms wordt een ander probleem voor dit probleem versleten, als servers en cliënten zich op verschillende netwerken bevinden. Als dat het geval is, dan dient *vastgesteld* te worden dat routers de UDP informatie op de juiste wijze routeren, omdat er anders nooit NFS-verkeer gerouteerd kan worden.

In de volgende voorbeelden is `fastws` de host(interface)naam van een hoog presterend werkstation en `freebox` is de host(interface)naam van een FreeBSD systeem met een Ethernet adapter die mindere prestaties levert. /sharedfs wordt het geëxporteerde NFS-bestandssysteem (zie exports(5))

en /project wordt het koppelpunt voor het geëxporteerde bestandssysteem op de cliënt.



In sommige gevallen kunnen applicaties beter draaien als extra opties als `hard` of `soft` en `bg` gebruikt worden.

Voorbeelden voor het FreeBSD systeem (`freebox`) als de cliënt in /etc/fstab op `freebox`:

```
fastws:/sharedfs /project nfs rw,-r=1024 0 0
```

Als een handmatig aankoppelcommando op `freebox`:

```
# mount -t nfs -o -r=1024 fastws:/sharedfs /project
```

Voorbeelden voor het FreeBSD systeem als de server in /etc/fstab op `fastws`:

```
freebox:/sharedfs /project nfs rw,-w=1024 0 0
```

Als een handmatig aankoppelcommando op `fastws`:

```
# mount -t nfs -o -w=1024 freebox:/sharedfs /project
```

Bijna iedere 16-bit Ethernet adapter werkt zonder de hierboven beschreven restricties op de lees- en schrijfgrootte.

Voor wie het wil weten wordt nu beschreven wat er gebeurt als de fout ontstaan, wat ook duidelijk maakt waarom het niet hersteld kan worden. NFS werkt meestal met een "block"grootte van 8 K (hoewel het mogelijk is dat er kleinere fragmenten worden verwerkt). Omdat de maximale grootte van een Ethernet pakket rond de 1500 bytes ligt, wordt een "block" opgesplitst in meerdere Ethernetpakketten, hoewel het hoger in de code nog steeds één eenheid is, en wordt ontvangen, samengevoegd en *bevestigd* als een eenheid. De hoog presterende werkstations kunnen de pakketten waaruit een NFS-eenheid bestaat bijzonder snel naar buiten pompen. Op de kaarten met minder capaciteit worden de eerdere pakketten door de latere pakketten van dezelfde eenheid ingehaald voordat ze bij die host zijn aangekomen en daarom kan de eenheid niet worden samengesteld en bevestigd. Als gevolg daarvan ontstaat er op het werkstation een timeout en probeert die de eenheid opnieuw te sturen, maar dan weer de hele eenheid van 8 K, waardoor het proces wordt herhaald, ad infinitum.

Door de grootte van de eenheid kleiner te houden dan de grootte van een Ethernet pakket, is het zeker dat elk Ethernetpakket dat compleet is aangekomen bevestigd kan worden, zodat de deadlock niet ontstaat.

Toch kan een PC systeem nog wel overrompeld worden als hoog presterende werkstations er op inhakken, maar met de betere netwerkkaarten valt het dan in ieder geval niet om door de NFS "eenheden". Als het systeem toch wordt overrompeld, dan worden de betrokken eenheden opnieuw verstuurd en dan is de kans groot dat ze worden ontvangen, samengevoegd en bevestigd.

27.4. Netwerkinformatiesysteem (NIS/YP)

27.4.1. Wat is het?

NIS, dat staat voor Netwerkinformatiediensten (Network Information Services), is ontwikkeld door Sun Microsystems om het beheer van UNIX® (origineel SunOS™) systemen te centraliseren. Tegenwoordig is het eigenlijk een industriestandaard geworden. Alle grote UNIX® achtige systemen (Solaris™, HP-UX, AIX®, Linux®, NetBSD, OpenBSD, FreeBSD, enzovoort) ondersteunen NIS.

NIS stond vroeger bekend als Yellow Pages, maar vanwege problemen met het handelsmerk heeft Sun de naam veranderd. De oude term, en yp, wordt nog steeds vaak gebruikt.

Het is een op RPC-gebaseerd cliënt/serversysteem waarmee een groep machines binnen een NIS-domein een gezamenlijke verzameling met instellingenbestanden kan delen. Hierdoor kan een beheerder NIS-systemen opzetten met een minimaal aantal instellingen en vanaf een centrale lokatie instellingen toevoegen, verwijderen en wijzigen.

Het is te vergelijken met het Windows NT® domeinsysteem en hoewel de interne implementatie van de twee helemaal niet overeenkomt, is de basisfunctionaliteit vergelijkbaar.

27.4.2. Termen en processen om te onthouden

Er zijn een aantal termen en belangrijke gebruikersprocessen die een rol spelen bij het implementeren van NIS op FreeBSD, zowel bij het maken van een NIS-server als bij het maken van een systeem dan NIS-cliënt is:

Term	Beschrijving
NIS-domeinnaam	Een NIS-masterserver en al zijn cliënten (inclusief zijn slave master) hebben een NIS-domeinnaam. Vergelijkbaar met een Windows NT® domeinnaam, maar de NIS-domeinnaam heeft niets te maken met DNS.
rpcbind	Moet draaien om RPC (Remote Procedure Call in te schakelen, een netwerkprotocol dat door NIS gebruikt wordt). Als rpcbind niet draait, dan kan er geen NIS-server draaien en kan een machine ook geen NIS-cliënt zijn.
ypbind	"Verbindt" een NIS-cliënt aan zijn NIS-server. Dat gebeurt door met de NIS-domeinnaam van het systeem en door het gebruik van RPC te verbinden met de server. ypbind is de kern van cliënt-server communicatie in een NIS-omgeving. Als ypbind op een machine stopt, dan kan die niet meer bij de NIS-server komen.

Term	Beschrijving
ypserv	Hoort alleen te draaien op NIS-servers. Dit is het NIS-serverproces zelf. Als ypserv(8) stopt, dan kan de server niet langer reageren op NIS-verzoeken (hopelijk is er dan een slaveserver om het over te nemen). Er zijn een aantal implementaties van NIS, maar niet die op FreeBSD, die geen verbinding met een andere server proberen te maken als de server waarmee ze verbonden waren niet meer reageert. In dat geval is vaak het enige dat werkt het serverproces herstarten (of zelfs de hele server) of het ypbind-proces op de cliënt.
rpc.yppasswdd	Nog een proces dat alleen op NIS-masterservers hoort te draaien. Dit is een daemon waarbij NIS-cliënten hun NIS-wachtwoorden kunnen wijzigen. Als deze daemon niet draait, moeten gebruikers zich aanmelden op de NIS-masterserver en daar hun wachtwoord wijzigen.

27.4.3. Hoe werkt het?

Er zijn drie typen hosts in een NIS-omgeving: master servers, slaveservers en cliënten. Servers zijn het centrale depot voor instellingen voor een host. Masterservers bevatten de geautoriseerd kopie van die informatie, terwijl slaveservers die informatie spiegelen voor redundantie. Cliënten verlaten zich op de servers om hun die informatie ter beschikking te stellen.

Op deze manier kan informatie uit veel bestanden gedeeld worden. De bestanden master.passwd, group en hosts worden meestal via NIS gedeeld. Als een proces op een cliënt informatie nodig heeft die normaliter in een van die lokale bestanden staat, dan vraagt die het in plaats daarvan aan de NIS-servers waarmee hij verbonden is.

Soorten machines

- Een *NIS-masterserver*. Deze server onderhoudt, analoog aan een Windows NT® primaire domeincontroller, de bestanden die door alle NIS-cliënten gebruikt worden. De bestanden passwd, group en andere bestanden die door de NIS-cliënten gebruikt worden staan op de masterserver.



Het is mogelijk om één machine master server te laten zijn voor meerdere NIS-domeinen. Dat wordt in deze inleiding echter niet beschreven, omdat die uitgaat van een relatief kleine omgeving.

- *NIS-slaveservers*. Deze zijn te vergelijken met Windows NT® backup domain controllers. NIS-slaveservers beheren een kopie van de bestanden met gegevens op de NIS-master. NIS-slaveservers bieden redundantie, die nodig is in belangrijke omgevingen. Ze helpen ook om de belasting te verdelen met de master server: NIS-cliënten maken altijd een verbinding met de

NIS-server die het eerst reageert en dat geldt ook voor antwoorden van slaveservers.

- *NIS-cliënten*. NIS-cliënten authenticeren, net als de meeste Windows NT® werkstations, tegen de NIS-server (of de Windows NT® domain controller in het geval van Windows NT® werkstations) bij het aanmelden.

27.4.4. NIS/YP gebruiken

Dit onderdeel behandelt het opzetten van een NIS-voorbeeldomgeving.

Plannen

Er wordt uitgegaan van een beheerder van een klein universiteitslab. Dat lab, dat bestaat uit FreeBSD machines, kent op dit moment geen centraal beheer. Iedere machine heeft zijn eigen /etc/passwd en /etc/master.passwd. Die bestanden worden alleen met elkaar in lijn gehouden door handmatige handelingen. Als er op dit moment een gebruiker aan het lab wordt toegevoegd, moet `adduser` op alle 15 machines gedraaid worden. Dat moet natuurlijk veranderen en daarom is besloten het lab in te richten met NIS, waarbij twee machines als server worden gebruikt.

Het lab ziet er ongeveer als volgt uit:

Machinenaam	IP-adres	Rol Machine
`ellington`	`10.0.0.2`	NIS-master
`coltrane`	`10.0.0.3`	NIS-slave
`basie`	`10.0.0.4`	Wetenschappelijk werkstation
`bird`	`10.0.0.5`	Cliënt machine
`cli[1-11]`	`10.0.0.[6-17]`	Andere cliënt machines

Bij het voor de eerste keer instellen van een NIS-schema is het verstandig eerst na te denken over hoe dat opgezet moet worden. Hoe groot een netwerk ook is, er moeten een aantal beslissingen gemaakt worden.

Een NIS-domeinnaam kiezen

Dit is wellicht niet de bekende "domeinnaam". Daarom wordt het ook de "NIS-domeinnaam" genoemd. Bij de broadcast van een cliënt om informatie wordt ook de naam van het NIS-domein waar hij onderdeel van uitmaakt meegezonden. Zo kunnen meerdere servers op een netwerk bepalen of er antwoord gegeven dient te worden op een verzoek. De NIS-domeinnaam kan voorgesteld worden als de naam van een groep hosts die op een of andere manier aan elkaar gerelateerd zijn.

Sommige organisaties kiezen hun Internet-domeinnaam als NIS-domeinnaam. Dat wordt niet aangeraden omdat het voor verwarring kan zorgen bij het debuggen van netwerkproblemen. De NIS-domeinnaam moet uniek zijn binnen een netwerk en het is handig als die de groep machines beschrijft waarvoor hij geldt. Zo kan bijvoorbeeld de financiële afdeling van Acme Inc. als NIS-domeinnaam "acme-fin" hebben. In dit voorbeeld wordt de naam `test-domain` gekozen.

Sommige besturingssystemen gebruiken echter (met name SunOS™) hun NIS-domeinnaam als hun

Internet-domeinnaam. Als er machines zijn op een netwerk die deze restrictie kennen, dan *moet* de Internet-domeinnaam als de naam voor het NIS-domeinnaam gekozen worden.

Systeemeisen

Bij het kiezen van een machine die als NIS-server wordt gebruikt zijn er een aantal aandachtspunten. Een van de onhandige dingen aan NIS is de afhankelijkheid van de cliënten van de server. Als een cliënt de server voor zijn NIS-domein niet kan bereiken, dan wordt die machine vaak onbruikbaar. Door het gebrek aan gebruiker- en groepsinformatie bevriezen de meeste systemen. Daarom moet er een machine gekozen worden die niet vaak herstart hoeft te worden of wordt gebruikt voor ontwikkeling. De NIS-server is in het meest ideale geval een alleenstaande server die als enige doel heeft NIS-server te zijn. Als een netwerk niet zwaar wordt gebruikt, kan de NIS-server op een machine die ook andere diensten aanbiedt gezet worden, maar het blijft belangrijk om ervan bewust te zijn dat als de NIS-server niet beschikbaar is, dat nadelige invloed heeft op *alle* NIS-cliënten.

NIS-servers

De hoofdversies van alle NIS-informatie staan opgeslagen op één machine die de NIS-masterserver heet. De databases waarin de informatie wordt opgeslagen heten NIS-afbeeldingen. In FreeBSD worden die afbeeldingen opgeslagen in /var/yp/[domeinnaam] waar [domeinnaam] de naam is van het NIS-domein dat wordt bediend. Een enkele NIS-server kan tegelijkertijd meerdere NIS-domeinen ondersteunen en het is dus mogelijk dat er meerdere van zulke mappen zijn, een voor ieder ondersteund domein. Ieder domein heeft zijn eigen onafhankelijke verzameling afbeeldingen.

In NIS-master- en -slaveservers worden alle NIS-verzoeken door de daemon `ypserv` afgehandeld. `ypserv` is verantwoordelijk voor het ontvangen van inkomende verzoeken van NIS-cliënten, het vertalen van de gevraagde domeinnaam en mapnaam naar een pad naar het corresponderende databasebestand en het terugsturen van de database naar de cliënten.

Een NIS-masterserver opzetten

Het opzetten van een master NIS-server kan erg eenvoudig zijn, afhankelijk van de behoeften. FreeBSD heeft ondersteuning voor NIS als basisfunctie. Alleen de volgende regels hoeven aan /etc/rc.conf toegevoegd te worden en FreeBSD doet de rest:

```
nisdomainname="test-domain"
```

1. Deze regel stelt de NIS-domeinnaam in op `test-domain` bij het instellen van het netwerk (bij het opstarten).

   ```
   nis_server_enable="YES"
   ```

2. Dit geeft FreeBSD aan de NIS-serverprocessen te starten als het netwerk de volgende keer wordt opgestart.

```
nis_yppasswdd_enable="YES"
```

3. Dit schakelt de daemon `rpc.yppasswdd` in die, zoals al eerder aangegeven, cliënten toestaat om hun NIS-wachtwoord vanaf een cliënt-machine te wijzigen.



Afhankelijk van de inrichting van NIS, kunnen er nog meer instellingen nodig zijn. In het onderdeel NIS-servers die ook NIS-cliënten zijn staan meer details.

Draai na het instellen van bovenstaande regels het commando `/etc/netstart` als supergebruiker. Het zal alles voor u instellen, gebruikmakende van de waarden die u in /etc/rc.conf heeft ingesteld. Start als laatste stap, voor het initialiseren van de NIS-afbeeldingen, de daemon ypserv handmatig:

```
# service ypserv start
```

NIS-afbeeldingen initialiseren

Die *NIS-afbeeldingen* zijn databasebestanden die in de map /var/yp staan. Ze worden gemaakt uit de bestanden met instellingen uit de map /etc van de NIS-master, met één uitzondering: /etc/master.passwd. Daar is een goede reden voor, want het is niet wenselijk om de wachtwoorden voor `root` en andere administratieve accounts naar alle servers in het NIS-domein te sturen. Daar moet voor het initialiseren van de NIS-afbeeldingen het volgende uitgevoerd worden:

```
# cp /etc/master.passwd /var/yp/master.passwd
# cd /var/yp
# vi master.passwd
```

Dan horen alle systeemaccounts verwijderd te worden (`bin`, `tty`, `kmem`, `games`, enzovoort) en alle overige accounts waarvoor het niet wenselijk is dat ze op de NIS-cliënten terecht komen (bijvoorbeeld `root` en alle andere UID 0 (supergebruiker) accounts).



/var/yp/master.passwd hoort niet te lezen te zijn voor een groep of voor de wereld (dus modus 600)! Voor het aanpassen van de rechten kan `chmod` gebruikt worden.

Als dat is gedaan, kunnen de NIS-afbeeldingen geïnitialiseerd worden. Bij FreeBSD zit een script `ypinit` waarmee dit kan (in de hulppagina staat meer informatie). Dit script is beschikbaar op de meeste UNIX® besturingssystemen, maar niet op allemaal. Op Digital UNIX/Compaq Tru64 UNIX heet het `ypsetup`. Omdat er afbeeldingen voor een NIS-master worden gemaakt, wordt de optie `-m` meegegeven aan `ypinit`. Aangenomen dat de voorgaande stappen zijn uitgevoerd, kunnen de NIS-afbeeldingen gemaakt worden op de volgende manier:

```
ellington# ypinit -m test-domain
Server Type: MASTER Domain: test-domain
Creating an YP server will require that you answer a few questions.
Questions will all be asked at the beginning of the procedure.
```

```
Do you want this procedure to quit on non-fatal errors? [y/n: n] n
Ok, please remember to go back and redo manually whatever fails.
If you don't, something might not work.
At this point, we have to construct a list of this domains YP servers.
rod.darktech.org is already known as master server.
Please continue to add any slave servers, one per line. When you are
done with the list, type a <control D>.
master server   :  ellington
next host to add:  coltrane
next host to add:  ^D
The current list of NIS servers looks like this:
ellington
coltrane
Is this correct?  [y/n: y] y

[..uitvoer van het maken van de afbeeldingen..]

NIS Map update completed.
ellington has been setup as an YP master server without any errors.
```

`ypinit` hoort /var/yp/Makefile gemaakt te hebben uit /var/yp/Makefile.dist. Als dit bestand is gemaakt, neemt dat bestand aan dat er in een omgeving met een enkele NIS-server wordt gewerkt met alleen FreeBSD-machines. Omdat `test-domain` ook een slaveserver bevat, dient /var/yp/Makefile gewijzigd te worden:

```
ellington# vi /var/yp/Makefile
```

Als de onderstaande regel niet al uitgecommentarieerd is, dient dat alsnog te gebeuren:

```
NOPUSH = "True"
```

Een NIS-slaveserver opzetten

Het opzetten van een NIS-slaveserver is nog makkelijker dan het opzetten van de master. Dit kan door aan te melden op de slaveserver en net als voor de masterserver /etc/rc.conf te wijzigen. Het enige verschil is dat nu de optie `-s` gebruikt wordt voor het draaien van `ypinit`. Met de optie `-s` moet ook de naam van de NIS-master meegegeven worden. Het commando ziet er dus als volgt uit:

```
coltrane# ypinit -s ellington test-domain

Server Type: SLAVE Domain: test-domain Master: ellington

Creating an YP server will require that you answer a few questions.
Questions will all be asked at the beginning of the procedure.

Do you want this procedure to quit on non-fatal errors? [y/n: n]  n
```

```
Ok, please remember to go back and redo manually whatever fails.
If you don't, something might not work.
There will be no further questions. The remainder of the procedure
should take a few minutes, to copy the databases from ellington.
Transferring netgroup...
ypxfr: Exiting: Map successfully transferred
Transferring netgroup.byuser...
ypxfr: Exiting: Map successfully transferred
Transferring netgroup.byhost...
ypxfr: Exiting: Map successfully transferred
Transferring master.passwd.byuid...
ypxfr: Exiting: Map successfully transferred
Transferring passwd.byuid...
ypxfr: Exiting: Map successfully transferred
Transferring passwd.byname...
ypxfr: Exiting: Map successfully transferred
Transferring group.bygid...
ypxfr: Exiting: Map successfully transferred
Transferring group.byname...
ypxfr: Exiting: Map successfully transferred
Transferring services.byname...
ypxfr: Exiting: Map successfully transferred
Transferring rpc.bynumber...
ypxfr: Exiting: Map successfully transferred
Transferring rpc.byname...
ypxfr: Exiting: Map successfully transferred
Transferring protocols.byname...
ypxfr: Exiting: Map successfully transferred
Transferring master.passwd.byname...
ypxfr: Exiting: Map successfully transferred
Transferring networks.byname...
ypxfr: Exiting: Map successfully transferred
Transferring networks.byaddr...
ypxfr: Exiting: Map successfully transferred
Transferring netid.byname...
ypxfr: Exiting: Map successfully transferred
Transferring hosts.byaddr...
ypxfr: Exiting: Map successfully transferred
Transferring protocols.bynumber...
ypxfr: Exiting: Map successfully transferred
Transferring ypservers...
ypxfr: Exiting: Map successfully transferred
Transferring hosts.byname...
ypxfr: Exiting: Map successfully transferred

coltrane has been setup as an YP slave server without any errors.
Don't forget to update map ypservers on ellington.
```

Nu hoort er een map /var/yp/test-domain te zijn waarin kopieë van de NIS-masterserver afbeeldingen staan. Die moeten bijgewerkt blijven. De volgende regel in /etc/crontab op de

slaveservers regelt dat:

```
20      *       *       *       *       root    /usr/libexec/ypxfr passwd.byname
21      *       *       *       *       root    /usr/libexec/ypxfr passwd.byuid
```

Met de bovenstaande twee regels wordt de slave gedwongen zijn afbeeldingen met de afbeeldingen op de masterserver te synchroniseren. Dit is niet verplicht omdat de masterserver automatisch probeert veranderingen aan de NIS-afbeeldingen door te geven aan zijn slaves. Echter, vanwege het belang van correcte wachtwoordinformatie op andere cliënten die van de slaveserver afhankelijk zijn, is het aanbevolen om specifiek de wachtwoordafbeeldingen vaak tot bijwerken te dwingen. Dit is des te belangrijker op drukke netwerken, omdat daar het bijwerken van afbeeldingen niet altijd compleet afgehandeld hoeft te worden.

Nu kan ook op de slaveserver het commando `/etc/netstart` uitgevoerd worden, dat op zijn beurt de NIS-server start.

NIS-cliënten

Een NIS-cliënt maakt wat heet een verbinding (binding) met een NIS-server met de daemon `ypbind`. `ypbind` controleert het standaarddomein van het systeem (zoals ingesteld met `domainname`) en begint met het broadcasten van RPC-verzoeken op het lokale netwerk. Die verzoeken bevatten de naam van het domein waarvoor `ypbind` een binding probeert te maken. Als een server die is ingesteld om het gevraagde domein te bedienen een broadcast ontvangt, dan antwoordt die aan `ypbind` dat dan het IP-adres van de server opslaat. Als er meerdere servers beschikbaar zijn, een master en bijvoorbeeld meerdere slaves, dan gebruikt `ypbind` het adres van de eerste server die antwoord geeft. Vanaf dat moment stuurt de cliënt alle NIS-verzoeken naar die server. `ypbind` "pingt" de server zo nu en dan om te controleren of die nog draait. Als er na een bepaalde tijd geen antwoord komt op een ping, dan markeert `ypbind` het domein als niet verbonden en begint het broadcasten opnieuw, in de hoop dat er een andere server wordt gelocaliseerd.

Een NIS-cliënt opzetten

Het opzetten van een FreeBSD machine als NIS-cliënt is redelijk doorzichtig:

1. Wijzig /etc/rc.conf en voeg de volgende regels toe om de NIS-domeinnaam in te stellen en `ypbind` mee te laten starten bij het starten van het netwerk:

   ```
   nisdomainname="test-domain"
   nis_client_enable="YES"
   ```

2. Om alle mogelijke regels voor accounts uit de NIS-server te halen, dienen alle gebruikersaccounts uit /etc/master.passwd verwijderd te worden en dient met `vipw` de volgende regel aan het einde van het bestand geplaatst te worden:

   ```
   +:::::::::
   ```



Door deze regel wordt alle geldige accounts in de wachtwoordafbeelding van de NIS-server toegang gegeven. Er zijn veel manieren om de NIS-cliënt in te stellen door deze regel te veranderen. In het onderdeel netgroepen hieronder staat meer informatie. Zeer gedetailleerde informatie staat in het boek `NFS en NIS beheren` van O'Reilly.



Er moet tenminste één lokale account behouden blijven (dus niet geïmporteerd via NIS) in /etc/master.passwd en die hoort ook lid te zijn van de groep `wheel`. Als er iets mis is met NIS, dan kan die account gebruikt worden om via het netwerk aan te melden, `root` te worden en het systeem te repareren.

3. Om alle groepen van de NIS-server te importeren, kan de volgende regel aan /etc/group toegevoegd worden:

```
+:*::
```

Voer, om de NIS-cliënt onmiddelijk te starten, de volgende commando's als supergebruiker uit:

```
# /etc/netstart
# service ypbind start
```

Na het afronden van deze stappen zou met `ypcat passwd` de passwd map van de NIS-server te zien moeten zijn.

27.4.5. NIS-beveiliging

In het algemeen kan iedere netwerkgebruiker een RPC-verzoek doen uitgaan naar ypserv(8) en de inhoud van de NIS-afbeeldingen ontvangen, mits die gebruiker de domeinnaam kent. Omdat soort ongeautoriseerde transacties te voorkomen, ondersteunt ypserv(8) de optie "securenets", die gebruikt kan worden om de toegang te beperken tot een opgegeven aantal hosts. Bij het opstarten probeert ypserv(8) de securenets informatie te laden uit het bestand /var/yp/securenets.



Dit pad kan verschillen, afhankelijk van het pad dat opgegeven is met de optie `-p`. Dit bestand bevat regels die bestaan uit een netwerkspecificatie en een netwerkmasker, gescheiden door witruimte. Regels die beginnen met `#` worden als commentaar gezien. Een voorbeeld van een securenetsbestand zou er zo uit kunnen zien:

```
# allow connections from local host -- mandatory
127.0.0.1     255.255.255.255
# allow connections from any host
# on the 192.168.128.0 network
192.168.128.0 255.255.255.0
```

```
# allow connections from any host
# between 10.0.0.0 to 10.0.15.255
# this includes the machines in the testlab
10.0.0.0        255.255.240.0
```

Als ypserv(8) een verzoek ontvangt van een adres dat overeenkomt met een van de bovenstaande regels, dan wordt dat verzoek normaal verwerkt. Als er geen enkele regel op het verzoek van toepassing is, dan wordt het verzoek genegeerd en wordt er een waarschuwing gelogd. Als het bestand /var/yp/securenets niet bestaat, dan accepteert `ypserv` verbindingen van iedere host.

Het programma `ypserv` ondersteunt ook het pakket TCP Wrapper van Wietse Venema. Daardoor kan een beheerder de instellingenbestanden van TCP Wrapper gebruiken voor toegangsbeperking in plaats van /var/yp/securenets.



Hoewel beide methoden van toegangscontrole enige vorm van beveiliging bieden, zijn ze net als de geprivilegieerde poorttest kwetsbaar voor "IP spoofing" aanvallen. Al het NIS-gerelateerde verkeer hoort door een firewall tegengehouden te worden.

Servers die gebruik maken van /var/yp/securenets kunnen wellicht legitieme verzoeken van NIS-cliënten weigeren als die gebruik maken van erg oude TCP/IP-implementaties. Sommige van die implementaties zetten alle host bits op nul als ze een broadcast doen en/of kijken niet naar het subnetmasker als ze het broadcastadres berekenen. Hoewel sommige van die problemen opgelost kunnen worden door de instellingen op de cliënt aan te passen, zorgen andere problemen voor het noodgedwongen niet langer kunnen gebruiker van NIS voor die cliënt of het niet langer gebruiken van /var/yp/securenets.

Het gebruik van /var/yp/securenets op een server met zo'n oude implementatie van TCP/IP is echt een slecht idee en zal leiden tot verlies van NIS-functionaliteit voor grote delen van een netwerk.

Het gebruik van het pakket TCP Wrapper leidt tot langere wachttijden op de NIS-server. De extra vertraging kan net lang genoeg zijn om een timeout te veroorzaken in cliëntprogramma's, in het bijzonder als het netwerk druk is of de NIS-server traag is. Als een of meer cliënten last hebben van dat symptoom, dan is het verstandig om de cliëntsysteem in kwestie NIS-slaveserver te maken en naar zichzelf te laten wijzen.

27.4.6. Aanmelden voor bepaalde gebruikers blokkeren

In het lab staat de machine `basie`, die alleen faculteitswerkstation hoort te zijn. Het is niet gewenst die machine uit het NIS-domein te halen, maar het passwd bestand op de master NIS-server bevat nu eenmaal accounts voor zowel de faculteit als de studenten. Hoe kan dat opgelost worden?

Er is een manier om het aanmelden van specifieke gebruikers op een machine te weigeren, zelfs als ze in de NIS-database staan. Daarvoor hoeft er alleen maar `-gebruikersnaam` met het juiste aantal dubbele punten (zoals bij andere regels) aan het einde van /etc/master.passwd op de cliëntmachine

toegevoegd te worden, waar *gebruikersnaam* de gebruikersnaam van de gebruiker die niet mag aanmelden is. De regel met de geblokkeerde gebruiker moet voor de regel met `+` staan om NIS-gebruikers toe te staan. Dit gebeurt bij voorkeur met `vipw`, omdat `vipw` de wijzigingen aan /etc/master.passwd controleert en ook de wachtwoord database opnieuw bouwt na het wijzigen. Om bijvoorbeeld de gebruiker `bill` te kunnen laten aanmelden op `basie`:

```
basie# vipw
[voeg -bill::::::::: aan het einde toe, exit]
vipw: rebuilding the database...
vipw: done

basie# cat /etc/master.passwd

root:[password]:0:0::0:0:The super-user:/root:/bin/csh
toor:[password]:0:0::0:0:The other super-user:/root:/bin/sh
daemon:*:1:1::0:0:Owner of many system processes:/root:/sbin/nologin
operator:*:2:5::0:0:System &:/:/sbin/nologin
bin:*:3:7::0:0:Binaries Commands and Source,,,:/:/sbin/nologin
tty:*:4:65533::0:0:Tty Sandbox:/:/sbin/nologin
kmem:*:5:65533::0:0:KMem Sandbox:/:/sbin/nologin
games:*:7:13::0:0:Games pseudo-user:/usr/games:/sbin/nologin
news:*:8:8::0:0:News Subsystem:/:/sbin/nologin
man:*:9:9::0:0:Mister Man Pages:/usr/shared/man:/sbin/nologin
bind:*:53:53::0:0:Bind Sandbox:/:/sbin/nologin
uucp:*:66:66::0:0:UUCP pseudo-user:/var/spool/uucppublic:/usr/libexec/uucp/uucico
xten:*:67:67::0:0:X-10 daemon:/usr/local/xten:/sbin/nologin
pop:*:68:6::0:0:Post Office Owner:/nonexistent:/sbin/nologin
nobody:*:65534:65534::0:0:Unprivileged user:/nonexistent:/sbin/nologin
-bill:::::::::
+:::::::::

basie#
```

27.4.7. Netgroups gebruiken

De methode uit het vorige onderdeel werkt prima als er maar voor een beperkt aantal gebruikers en/of machines speciale regels nodig zijn. Op grotere netwerken *gebeurt* het gewoon dat er wordt vergeten om een aantal gebruikers de aanmeldrechten op gevoelige machines te ontnemen of dat zelfs iedere individuele machine aangepast moet worden, waardoor het voordeel van NIS teniet wordt gedaan: *centraal* beheren.

De ontwikkelaars van NIS hebben dit probleem opgelost met *netgroepen*. Het doel en de semantiek kunnen vergeleken worden met de normale groepen die gebruikt worden op UNIX® bestandssystemen. De belangrijkste verschillen zijn de afwezigheid van een numeriek ID en de mogelijkheid om een netgroep aan te maken die zowel gebruikers als andere netgroepen bevat.

Netgroepen zijn ontwikkeld om gebruikt te worden voor grote, complexe netwerken met honderden gebruikers en machines. Aan de ene kant is dat iets Goeds. Aan de andere kant is het wel complex en bijna onmogelijk om netgroepen met een paar eenvoudige voorbeelden uit te

leggen. Dat probleem wordt in de rest van dit onderdeel duidelijk gemaakt.

Stel dat de succesvolle implementatie van NIS in het lab de interesse heeft gewekt van een centrale beheerclub. De volgende taak is het uitbreiden van het NIS-domein met een aantal andere machines op de campus. De onderstaande twee tabellen bevatten de namen van de nieuwe gebruikers en de nieuwe machines met een korte beschijving.

Gebruikersnamen	Beschrijving
`alpha`, `beta`	Gewone medewerkers van de IT-afdeling
`charlie`, `delta`	Junior medewerkers van de IT-afdeling
`echo`, `foxtrott`, `golf`, …	Gewone medewerkers
`able`, `baker`, …	Stagiairs

Machinenamen	Beschrijving
`war`, `death`, `famine`, `pollution`	De belangrijkste servers. Alleen senior medewerkers van de IT-afdeling mogen hierop aanmelden.
`pride`, `greed`, `envy`, `wrath`, `lust`, `sloth`	Minder belangrijke servers. Alle leden van de IT-afdeling mogen aanmelden op deze machines.
`one`, `two`, `three`, `four`, …	Gewone werkstations. Alleen *echte* medewerkers mogen zich op deze machines aanmelden.
`trashcan`	Een erg oude machine zonder kritische data. Zelfs de stagiair mag deze doos gebruiken.

Als deze restricties ingevoerd worden door iedere gebruiker afzonderlijk te blokkeren, dan wordt er een `-user` regel per systeem toegevoegd aan de passwd voor iedere gebruiker die niet mag aanmelden op dat systeem. Als er maar één regel wordt vergeten, kan dat een probleem opleveren. Wellicht lukt het nog dit juist in te stellen bij de bouw van een machine, maar het wordt *echt* vergeten de regels toe te voegen voor nieuwe gebruikers in de productiefase. Murphy was tenslotte een optimist.

Het gebruik van netgroepen biedt in deze situatie een aantal voordelen. Niet iedere gebruiker hoeft separaat afgehandeld te worden. Een gebruik kan aan een of meer groepen worden toegevoegd en aanmelden kan voor alle leden van zo'n groep worden toegestaan of geweigerd. Als er een nieuwe machine wordt toegevoegd, dan hoeven alleen de aanmeldrestricties voor de netgroepen te worden ingesteld. Als er een nieuwe gebruiker wordt toegevoegd, dan hoeft die alleen maar aan de juiste netgroepen te worden toegevoegd. Die veranderingen zijn niet van elkaar afhankelijk: geen "voor iedere combinatie van gebruiker en machine moet het volgende …". Als de NIS-opzet zorgvuldig is gepland, dan hoeft er maar één instellingenbestand gewijzigd te worden om toegang tot machines te geven of te ontnemen.

De eerst stap is het initialiseren van de NIS-afbeelding netgroup. ypinit(8) van FreeBSD maakt deze map niet standaard, maar als die is gemaakt, ondersteunt de NIS-implementatie hem wel. Een lege map wordt als volgt gemaakt:

```
ellington# vi /var/yp/netgroup
```

Nu kan hij gevuld worden. In het gebruikte voorbeeld zijn tenminste vier netgroepen: IT-medewerkers, IT-junioren, gewone medewerkers en stagiars.

```
IT_MW   (,alpha,test-domain)    (,beta,test-domain)
IT_APP  (,charlie,test-domain)  (,delta,test-domain)
USERS   (,echo,test-domain)     (,foxtrott,test-domain) \
        (,golf,test-domain)
STAGS   (,able,test-domain)     (,baker,test-domain)
```

`IT_MW`, `IT_APP` enzovoort, zijn de namen van de netgroepen. Iedere groep tussen haakjes bevat een of meer gebruikersnamen voor die groep. De drie velden binnen een groep zijn:

1. De naam van de host of namen van de hosts waar de volgende onderdelen geldig zijn. Als er geen hostnaam wordt opgegeven dan is de regel geldig voor alle hosts. Als er wel een hostnaam wordt opgegeven, dan wordt een donker, spookachtig en verwarrend domein betreden.
2. De naam van de account die bij deze netgroep hoort.
3. Het NIS-domein voor de account. Er kunnen accounts uit andere NIS-domeinen geïmporteerd worden in een netgroep als een beheerder zo ongelukkig is meerdere NIS-domeinen te hebben.

Al deze velden kunnen jokerkarakters bevatten. Details daarover staan in netgroup(5).



De naam van een netgroep mag niet langer zijn dan acht karakters, zeker niet als er andere besturingssystemen binnen een NIS-domein worden gebruikt. De namen zijn hoofdlettergevoelig: alleen hoofdletters gebruiken voor de namen van netgroepen is een makkelijke manier om onderscheid te kunnen maken tussen gebruikers-, machine- en netgroepnamen.

Sommige NIS-cliënten (andere dan die op FreeBSD draaien) kunnen niet omgaan met netgroepen met veel leden. Sommige oudere versies van SunOS™ gaan bijvoorbeeld lastig doen als een netgroep meer dan 15 *leden* heeft. Dit kan omzeild worden door meerdere subnetgroepen te maken met 15 gebruikers of minder en een echte netgroep die de subnetgroepen bevat:

```
BIGGRP1  (,joe1,domain)  (,joe2,domain)  (,joe3,domain) [...]
BIGGRP2  (,joe16,domain)  (,joe17,domain) [...]
BIGGRP3  (,joe31,domain)  (,joe32,domain)
BIGGROUP  BIGGRP1 BIGGRP2 BIGGRP3
```

Dit proces kan herhaald worden als er meer dan 225 gebruikers in een netgroep moeten.

Het activeren en distribueren van de nieuwe NIS-map is eenvoudig:

```
ellington# cd /var/yp
ellington# make
```

Hiermee worden drie nieuwe NIS-afbeeldingen gemaakt: netgroup, netgroup.byhost en netgroup.byuser. Met ypcat(1) kan bekeken worden op de nieuwe NIS-afbeeldingen beschikbaar zijn:

```
ellington% ypcat -k netgroup
ellington% ypcat -k netgroup.byhost
ellington% ypcat -k netgroup.byuser
```

De uitvoer van het eerste commando hoort te lijken op de inhoud van /var/yp/netgroup. Het tweede commando geeft geen uitvoer als er geen host-specifieke netgroepen zijn ingesteld. Het derde commando kan gebruikt worden om een lijst van netgroepen voor een gebruiker op te vragen.

Het instellen van de cliënt is redelijk eenvoudig. Om de server `war` in te stellen hoeft alleen met vipw(8) de volgende regel in de regel daarna vervangen te worden:

```
+:::::::::
```

Vervang de bovenstaande regel in de onderstaande.

```
+@IT_MW:::::::::
```

Nu worden alleen de gebruikers die in de netgroep `IT_MW` geïmporteerd in de wachtwoorddatabase van de host `war`, zodat alleen die gebruikers zich kunnen aanmelden.

Helaas zijn deze beperkingen ook van toepassing op de functie `~` van de shell en alle routines waarmee tussen gebruikersnamen en numerieke gebruikers ID's wordt gewisseld. Met andere woorden: `cd ~user` werkt niet, `ls -l` toont het numerieke ID in plaats van de gebruikersnaam en `find . -user joe -print` faalt met de foutmelding `No such user`. Om dit te repareren moeten alle gebruikers geïmporteerd worden, *zonder ze het recht te geven aan te melden op een server*.

Dit kan gedaan worden door nog een regel aan /etc/master.passwd toe te voegen:

```
+:::::::::/sbin/nologin
```

Dit betekent "importeer alle gebruikers, maar vervang de shell door /sbin/nologin". Ieder veld in een `passwd` regel kan door een standaardwaarde vervangen worden in /etc/master.passwd.



De regel `:::::::::/sbin/nologin` moet na `@IT_MW:::::::::` komen. Anders krijgen alle gebruikers die uit NIS-komen /sbin/nologin als aanmeldshell.

Na deze wijziging hoeft er nog maar één NIS-afbeelding gewijzigd te worden als er een nieuwe

medewerker komt bij de IT-afdeling. Dezelfde aanpak kan gebruikt worden voor de minder belangrijke servers door de oude regel `+:::::::::` in de lokale versie van /etc/master.passwd door iets als het volgende te vervangen:

```
+@IT_MW:::::::::
+@IT_APP:::::::::
+:::::::::/sbin/nologin
```

Voor normale werkstations zijn het de volgende regels:

```
+@IT_MW:::::::::
+@USERS:::::::::
+:::::::::/sbin/nologin
```

En dat zou allemaal leuk en aardig zijn als er niet na een paar weken een beleidsverandering komt: de IT-afdeling gaat stagiairs aannemen. De IT-stagiairs mogen de normale werkstations en de minder belangrijke servers gebruiken en de juniorbeheerders mogen gaan aanmelden op de hoofdservers. Dat kan door een nieuwe groep `IT_STAG` te maken en de nieuwe IT-stagiairs toe te voegen aan die netgroep en dan de instellingen op iedere machine te gaan veranderen. Maar zoals het spreekwoord zegt: "Fouten in een centrale planning leiden tot complete chaos."

Deze situaties kunnen voorkomen worden door gebruik te maken van de mogelijkheid in NIS om netgroepen in netgroepen op te nemen. Het is mogelijk om rolgebaseerde netgroepen te maken. Er kan bijvoorbeeld een netgroep `BIGSRV` gemaakt worden om het aanmelden op de belangrijke servers te beperken en er kan een andere netgroep `SMALLSRV` voor de minder belangrijke servers zijn en een derde netgroep met de naam `USERBOX` voor de normale werkstations. Al die netgroepen kunnen de netgroepen bevatten die op die machines mogen aanmelden. De nieuwe regels in de NIS-afbeelding netgroup zien er dan zo uit:

```
BIGSRV    IT_MW  IT_APP
SMALLSRV  IT_MW  IT_APP  ITSTAG
USERBOX   IT_MW  ITSTAG USERS
```

Deze methode voor het instellen van aanmeldbeperkingen werkt redelijk goed als er groepen van machines gemaakt kunnen worden met identieke beperkingen. Helaas blijkt dat eerder uitzondering dan regel. Meestal moet het mogelijk zijn om per machine in te stellen wie zich wel en wie zich niet mogen aanmelden.

Daarom is het ook mogelijk om via machinespecifieke netgroepen de hierboven aangegeven beleidswijziging op te vangen. In dat scenario bevat /etc/master.passwd op iedere machine twee regels die met "+" beginnen. De eerste voegt de netgroep toe met de accounts die op de machine mogen aanmelden en de tweede voegt alle andere accounts toe met /sbin/nologin als shell. Het is verstandig om als naam van de netgroep de machinenaam in "HOOFDLETTERS" te gebruiken. De regels zien er ongeveer als volgt uit:

```
+@MACHINENAAM:::::::::
```

```
+:::::::::/sbin/nologin
```

Als dit voor alle machines is gedaan, dan hoeven de lokale versies van /etc/master.passwd nooit meer veranderd te worden. Alle toekomstige wijzigingen kunnen dan gemaakt worden door de NIS-afbeelding te wijzigen. Hieronder staat een voorbeeld van een mogelijke netgroep map voor het beschreven scenario met een aantal toevoegingen:

```
# Definieer eerst de gebruikersgroepen
IT_MW     (,alpha,test-domain)    (,beta,test-domain)
IT_APP    (,charlie,test-domain)  (,delta,test-domain)
DEPT1     (,echo,test-domain)     (,foxtrott,test-domain)
DEPT2     (,golf,test-domain)     (,hotel,test-domain)
DEPT3     (,india,test-domain)    (,juliet,test-domain)
ITSTAG    (,kilo,test-domain)     (,lima,test-domain)
D_STAGS   (,able,test-domain)     (,baker,test-domain)
#
# En nu een aantal groepen op basis van rollen
USERS     DEPT1   DEPT2     DEPT3
BIGSRV    IT_MW   IT_APP
SMALLSRV  IT_MW   IT_APP    ITSTAG
USERBOX   IT_MW   ITSTAG    USERS
#
# Een een groep voor speciale taken.
# Geef echo en golf toegang tot de anti-virus machine.
SECURITY  IT_MW   (,echo,test-domain)  (,golf,test-domain)
#
# Machinegebaseerde netgroepen
# Hoofdservers
WAR       BIGSRV
FAMINE    BIGSRV
# Gebruiker india heeft toegang tot deze server nodig.
POLLUTION  BIGSRV  (,india,test-domain)
#
# Deze is erg belangrijk en heeft strengere toegangseisen nodig.
DEATH     IT_MW
#
# De anti-virus machine als hierboven genoemd.
ONE       SECURITY
#
# Een machine die maar door 1 gebruiker gebruikt mag worden.
TWO       (,hotel,test-domain)
# [...hierna volgen de andere groepen]
```

Als er een soort database wordt gebruikt om de gebruikersaccounts te beheren, dan is het in ieder geval nodig dat ook het eerste deel van de afbeelding met de databaserapportagehulpmiddelen gemaakt kan worden. Dan krijgen nieuwe gebruikers automatisch toegang tot de machines.

Nog een laatste waarschuwing: het is niet altijd aan te raden gebruik te maken van machinegebaseerde netgroepen. Als er tientallen of zelfs honderden gelijke machines voor

bijvoorbeeld studentenruimtes worden uitgerold, dan is het verstandiger rolgebaseerde netgroepen te gebruiken in plaats van machinegebaseerde netgroepen om de grootte van de NIS-afbeelding binnen de perken te houden.

27.4.8. Belangrijk om te onthouden

In een NIS-omgeving werken een aantal dingen wel anders.

- Als er een gebruiker toegevoegd moet worden, dan moet die *alleen* toegevoegd worden aan de master NIS-server en *mag niet vergeten worden dat de NIS-afbeeldingen herbouwd moeten worden*. Als dit wordt vergeten, dan kan de nieuwe gebruiker nergens anders aanmelden dan op de NIS-master. Als bijvoorbeeld een nieuwe gebruiker `jsmith` toegevoegd moet worden:

  ```
  # pw useradd jsmith
  # cd /var/yp
  # make test-domain
  ```

  Er kan ook `adduser jsmith` in plaats van `pw useradd jsmith` gebruikt worden.

- *De beheeraccounts moeten buiten de NIS-afbeeldingen gehouden worden*. Het is niet handig als de beheeraccounts en wachtwoorden naar machines waarop gebruikers zich aanmelden die geen toegang tot die informatie horen te hebben zouden gaan.
- *De NIS-master en slave moeten veilig blijven en zo min mogelijk niet beschikbaar zijn*. Als de machine wordt gehackt of als hij wordt uitgeschakeld, dan kunnen er in theorie nogal wat mensen niet meer aanmelden.

  Dit is de belangrijkste zwakte van elk gecentraliseerd beheersysteem. Als de NIS-servers niet goed beschermd worden, dan worden veel gebruikers boos!

27.4.9. NIS v1-compatibiliteit

ypserv voor FreeBSD biedt wat ondersteuning voor NIS v1 cliënten. De NIS-implementatie van FreeBSD gebruikt alleen het NIS v2 protocol, maar andere implementaties bevatten ondersteuning voor het v1 protocol voor achterwaartse compatibiliteit met oudere systemen. De ypbind-daemons die bij deze systemen zitten proberen een binding op te zetten met een NIS v1 server, hoewel dat niet per se ooit nodig is (en ze gaan misschien nog wel door met broadcasten nadat ze een antwoord van een v2 server hebben ontvangen). Het is belangrijk om te melden dat hoewel ondersteuning voor gewone cliëntoproepen aanwezig is, deze versie van ypserv geen overdrachtsverzoeken voor v1-afbeeldingen af kan handelen. Daarom kan ypserv niet gebruikt worden als master of slave in combinatie met oudere NIS-servers die alleen het v1 protocol ondersteunen. Gelukkig worden er in deze tijd niet meer zoveel van deze servers gebruikt.

27.4.10. NIS-servers die ook NIS-cliënten zijn

Het is belangrijk voorzichtig om te gaan met het draaien van ypserv in een multi-server domein waar de server machines ook NIS-cliënten zijn. Het is in het algemeen verstandiger om de servers te dwingen met zichzelf te binden dan ze toe te staan een bindverzoek te broadcasten en het risico te lopen dat ze een binding met elkaar maken. Er kunnen vreemde fouten optreden als een van de

servers plat gaat als er andere servers van die server afhankelijk zijn. Na verloop van tijd treedt op de cliënten wel een timeout op en verbinden ze met een andere server, maar de daarmee gepaard gaande vertraging kan aanzienlijk zijn en de foutmodus is nog steeds van toepassing, omdat de servers dan toch weer opnieuw een verbinding met elkaar kunnen vinden.

Het is mogelijk een host aan een specifieke server te binden door aan `ypbind` de vlag `-S` mee te geven. Om dit niet iedere keer handmatig na een herstart te hoeven uitvoeren, kan de volgende regel worden opgenomen in /etc/rc.conf van de NIS-server:

```
nis_client_enable="YES" # start ook het cliënt gedeelte
nis_client_flags="-S NIS domain,server"
```

In ypbind(8) staat meer informatie.

27.4.11. Wachtwoordformaten

Een van de meest voorkomende problemen bij het implementeren van NIS is de compatibiliteit van het wachtwoordformaat. Als een NIS-server wachtwoorden gebruikt die met DES gecodeerd zijn, dan kunnen alleen cliënten die ook DES gebruiken ondersteund worden. Als er bijvoorbeeld Solaris™ NIS-cliënten in een netwerk zijn, dan moet er vrijwel zeker gebruik gemaakt worden van met DES gecodeerde wachtwoorden.

Van welk formaat cliënten en servers gebruik maken is te zien in /etc/login.conf. Als een host gebruik maakt van met DES gecodeerde wachtwoorden, dan staat er in de klasse `default` een regel als de volgende:

```
default:\
    :passwd_format=des:\
    [Overige regels weggelaten]
```

Andere mogelijke waarden voor `passwd_format` zijn `blf` en `md5` (respectievelijk voor Blowfish en MD5 gecodeerde wachtwoorden).

Als er wijzigingen gemaakt zijn aan /etc/login.conf dan moet de login capability database herbouwd worden door het volgende commando als `root` uit te voeren:

```
# cap_mkdb /etc/login.conf
```



Het formaat van de wachtwoorden die al in /etc/master.passwd staan worden niet bijgewerkt totdat een gebruiker zijn wachtwoord voor de eerste keer wijzigt *nadat* de login capability database is herbouwd.

Om te zorgen dat de wachtwoorden in het gekozen formaat zijn gecodeerd, moet daarna gecontroleerd worden of de waarde `crypt_default` in /etc/auth.conf de voorkeur geeft aan het gekozen formaat. Om dat te realiseren dient het gekozen formaat vooraan gezet te worden in de lijst. Als er bijvoorbeeld gebruik gemaakt wordt van DES gecodeerde wachtwoorden, dan hoort de

regel er als volgt uit te zien:

```
crypt_default   =   des blf md5
```

Als de bovenstaande stappen op alle FreeBSD gebaseerde NIS-servers en cliënten zijn uitgevoerd, dan is het zeker dat ze het allemaal eens zijn over welk wachtwoordformaat er op het netwerk wordt gebruikt. Als er problemen zijn bij de authenticatie op een NIS-cliënt, dan is dit een prima startpunt voor het uitzoeken waar de problemen vandaan komen. Nogmaals: als er een NIS-server in een heterogene omgeving wordt geplaatst, dan is het waarschijnlijk dat er gebruik gemaakt moet worden van DES op alle systemen, omdat dat de laagst overeenkomende standaard is.

27.5. Automatisch netwerk instellen (DHCP)

27.5.1. Wat is DHCP?

DHCP, het Dynamic Host Configuration Protocol, schrijft voor hoe een systeem verbinding kan maken met een netwerk en hoe het de benodigde informatie kan krijgen om met dat netwerk te communiceren. FreeBSD gebruikt de OpenBSD `dhclient` welke uit OpenBSD 3.7 komt. Alle informatie over `dhclient` kan zowel voor de ISC als de OpenBSD DHCP-cliënt gebruikt worden. De DHCP-server zit bij de ISC-distributie.

27.5.2. Wat behandeld wordt

In dit onderdeel worden de cliëntcomponenten van de ISC en OpenBSD DHCP-cliënt en de servercomponenten van het ISC DHCP-systeem beschreven. Het programma voor de cliënt, `dhclient`, zit standaard in FreeBSD en de server is beschikbaar via de port net/isc-dhcp42-server. Naast de onderstaande informatie, zijn de hulppagina's van dhclient(8), dhcp-options(5) en dhclient.conf(5) bruikbare bronnen.

27.5.3. Hoe het werkt

Als `dhclient`, de DHCP-cliënt, wordt uitgevoerd op een cliëntmachine, dan begint die met het broadcasten van verzoeken om instellingeninformatie. Standaard worden deze verzoeken op UDP poort 68 gedaan. De server antwoordt op UDP 67 en geeft de cliënt een IP-adres en andere relevante netwerkinformatie, zoals een netmasker, router en DNS-servers. Al die informatie komt in de vorm van een DHCP "lease" en is voor een bepaalde tijd geldig (die is ingesteld door de beheerder van de DHCP-server). Op die manier kunnen IP-adressen voor cliënten die niet langer met het netwerk verbonden zijn (stale) automatisch weer ingenomen worden.

DHCP-cliënten kunnen veel informatie van de server krijgen. Er staat een uitputtende lijst in dhcp-options(5).

27.5.4. FreeBSD integratie

FreeBSD integreert de OpenBSD DHCP-cliënt `dhclient` volledig. Er is ondersteuning voor de DHCP-cliënt in zowel het installatieprogramma als in het basissysteem, waardoor het niet noodzakelijk is om kennis te hebben van het maken van netwerkinstellingen voor het netwerk waar een DHCP-

server draait.

DHCP wordt ondersteund door sysinstall. Bij het instellen van een netwerkinterface binnen sysinstall is de tweede vraag: "Wil je proberen de interface met DHCP in te stellen?" Als het antwoord bevestigend luidt, dan wordt `dhclient` uitgevoerd en als dat succesvol verloopt, dan worden de netwerkinstellingen automatisch ingevuld.

Voor het gebruiken van DHCP bij het opstarten van het systeem zijn twee instellingen nodig:

- Het apparaat bpf moet in de kernel gecompileerd zijn. Dit kan door `device bpf` aan het bestand met kernelinstellingen toe te voegen en de kernel te herbouwen. Meer informatie over het bouwen van een kernel staat in De FreeBSD-kernel instellen.

  Het apparaat bpf is al onderdeel van de GENERIC kernel die bij FreeBSD zit, dus als er geen sprake is van een aangepaste kernel, dan hoeft er geen nieuwe gemaakt te worden om DHCP aan te praat te krijgen.

  Voor de lezer die bijzonder begaan is met beveiliging, is het belangrijk aan te geven dat bpf ook het apparaat is waardoor pakketsnuffelaars hun werk kunnen doen (hoewel ze nog steeds als `root` moeten draaien). bpf*is* noodzakelijk voor DHCP, maar als beveiliging bijzonder belangrijk is, dan hoort bpf waarschijnlijk niet in een kernel te zitten omdat de verwachting dat er in de toekomst ooit DHCP gebruikt gaat worden.

- Standaard draait de DHCP-synchronisatie op FreeBSD in de achtergrond, of *asynchroon*. Andere opstartscripten gaan verder terwijl DHCP wordt voltooid, wat het opstarten van het systeem versnelt.

  DHCP in de achtergrond werkt goed als de DHCP-server snel op verzoeken reageert en het DHCP-configuratieproces snel gaat. Op sommige systemen kan het lang duren voordat DHCP klaar is. Als netwerkdiensten proberen te draaien voordat DHCP voltooid is, zullen ze falen. Door DHCP in *synchrone* modus te draaien wordt dit probleem voorkomen en wordt het opstarten gepauzeerd totdat de DHCP-configuratie voltooid is.

  Gebruik om in de achtergrond verbinding te maken met een DHCP-server terwijl andere opstartscripts verder gaan (asynchrone modus) de waarde “DHCP” in /etc/rc.conf:

  ```
  ifconfig_fxp0="DHCP"
  ```

  Gebruik om het opstarten te pauzeren totdat DHCP voltooid is de synchrone modus met waarde “SYNDHCP”:

  ```
  ifconfig_fxp0="SYNDHCP"
  ```

  Vervang *fxp0* zoals getoond in deze voorbeelden met de naam van de interface dat dynamisch geconfigureerd moet worden, zoals getoond in Netwerkkaarten instellen.

Als er een andere lokatie voor `dhclient` wordt gebruikt of als er extra parameters aan `dhclient` meegegeven moeten worden, dan dient ook iets als het volgende toegevoegd te worden:

```
dhclient_program="/sbin/dhclient"
dhclient_flags=""
```

De DHCP-server, dhcpd, zit bij de port net/isc-dhcp42-server in de Portscollectie. Deze port bevat de ISC DHCP-server en documentatie.

27.5.5. Bestanden

- /etc/dhclient.conf

  Voor `dhclient` is een instellingenbestand /etc/dhclient.conf nodig. Dat bestand bevat meestal alleen maar commentaar, omdat de standaardinstellingen redelijk zinvol zijn. Dit bestand wordt beschreven in dhclient.conf(5).

- /sbin/dhclient

  `dhclient` is statisch gelinkt en staat in /sbin. Er staat meer informatie over `dhclient` in dhclient(8).

- /sbin/dhclient-script

  `dhclient-script` is het FreeBSD-specifieke DHCP-cliënt instellingenscript. Het wordt beschreven in dhclient-script(8), maar het is niet nodig het te wijzigen om goed te werken.

- /var/db/dhclient.leases.interface

  De DHCP-cliënt houdt in dit bestand een database bij van geldige leases, die naar een logboekbestand worden geschreven. In dhclient.leases(5) staat een iets uitgebreidere beschrijving.

27.5.6. Verder lezen

Het DHCP-protocol staat volledig beschreven in RFC 2131. Er is nog een bron van informatie ingesteld op http://www.dhcp.org/.

27.5.7. Een DHCP-server installeren en instellen

Wat behandeld wordt

In dit onderdeel wordt beschreven hoe een FreeBSD systeem zo ingesteld kan worden dat het opereert als DHCP-server door gebruik te maken van de ISC (Internet Systems Consortium) implementatie van de DHCP-server.

De server wordt niet geleverd als deel van FreeBSD en om deze dienst aan te bieden dient de port net/isc-dhcp42-server geïnstalleerd te worden. In Applicaties installeren. pakketten en ports staat meer informatie over de Portscollectie.

DHCP-serverinstallatie

Om een FreeBSD systeem in te stellen als DHCP-server moet het apparaat bpf(4) in de kernel zijn opgenomen. Om dit te doen dient `device bpf` aan het bestand met kernelinstellingen toegevoegd te worden en dient de kernel herbouwd te worden. Meer informatie over het bouwen van kernels staat in De FreeBSD-kernel instellen.

Het apparaat bpf is al onderdeel van de GENERIC kernel die bij FreeBSD, dus het is meestal niet nodig om een aangepaste kernel te bouwen om DHCP aan de praat te krijgen.



Het is belangrijk te vermelden dat bpf ook het apparaat is waardoor pakketsnuffelaars kunnen werken (hoewel de programma's die er gebruik van maken wel bijzondere toegang nodig hebben). bpf*is* verplicht voor DHCP, maar als beveiliging van belang is, dan is het waarschijnlijk niet verstandig om bpf in een kernel op te nemen alleen omdat er in de toekomst misschien ooit DHCP gebruikt gaat worden.

Hierna dient het standaardbestand dhcpd.conf dat door de port net/isc-dhcp42-server is geïnstalleerd gewijzigd te worden. Standaard is dit /usr/local/etc/dhcpd.conf.sample en dit bestand dient gekopieerd te worden naar /usr/local/etc/dhcpd.conf voordat de wijzigingen worden gemaakt.

De DHCP-server instellen

dhcpd.conf is opgebouwd uit declaraties over subnetten en hosts en is wellicht het meest eenvoudig te beschrijven met een voorbeeld:

```
option domain-name "example.com"; ①
option domain-name-servers 192.168.4.100; ②
option subnet-mask 255.255.255.0; ③

default-lease-time 3600; ④
max-lease-time 86400; ⑤
ddns-update-style none; ⑥

subnet 192.168.4.0 netmask 255.255.255.0 {
  range 192.168.4.129 192.168.4.254; ⑦
  option routers 192.168.4.1; ⑧
}

host mailhost {
  hardware ethernet 02:03:04:05:06:07; ⑨
  fixed-address mailhost.example.com; ⑩
}
```

① Deze optie geeft het domein aan dat door cliënten als standaard zoekdomein wordt gebruikt. In resolv.conf(5) staat meer over wat dat betekent.

② Deze optie beschrijft een door komma's gescheiden lijst met DNS-servers die de cliënt moet gebruiken.

③ Het netmasker dat aan de cliënten wordt voorgeschreven.

④ Een cliënt kan om een bepaalde duur vragen die een lease geldig is. Anders geeft de server aan wanneer de lease vervalt (in seconden).

⑤ Dit is de maximale duur voor een lease die de server toestaat. Als een cliënt vraagt om een langere lease, dan wordt die wel verstrekt, maar is de maar geldig gedurende `max-lease-time` seconden.

⑥ Deze optie geeft aan of de DHCP-server moet proberen de DNS-server bij te werken als een lease is geaccepteerd of wordt vrijgegeven. In de ISC implementatie is deze optie *verplicht*.

⑦ Dit geeft aan welke IP-adressen in de groep met adressen zitten die zijn gereserveerd om uitgegeven te worden aan cliënten. Alle IP-adressen tussen de aangegeven adressen en die adressen zelf worden aan cliënten uitgegeven.

⑧ Geeft de default gateway aan die aan de cliënten wordt voorgeschreven.

⑨ Het hardware MAC-adres van een host, zodat de DHCP-server een host kan herkennen als die een verzoek doet.

⑩ Geeft een host aan die altijd hetzelfde IP-adres moet krijgen. Hier kan een hostnaam gebruikt worden, omdat de DHCP-server de hostnaam zelf opzoekt voordat de lease-informatie terug wordt gegeven.

Wanneer u klaar bent met het schrijven van uw dhcpd.conf, dient u de DHCP-server in /etc/rc.conf aan te zetten, door het volgende toe te voegen:

```
dhcpd_enable="YES"
dhcpd_ifaces="dc0"
```

Vervang de interfacenaam `dc0` door de interface (of interfaces, gescheiden door witruimtes) waarop uw DHCP-server moet luisteren naar DHCP-verzoeken van cliënten.

Daarna kunt u doorgaan met het starten van de server door het volgende commando te geven:

```
# service isc-dhcpd start
```

Als er later wijzigingen in de instellingen gemaakt moeten worden, dan is het belangrijk te onthouden dat het sturen van een `SIGHUP` signaal naar dhcpd_niet_ resulteert in het opnieuw laden van de instellingen, zoals voor de meeste daemons geldt. Voor deze daemon dient een signaal `SIGTERM` gestuurd te worden om het proces te stoppen. Daarna dient de daemon met het hiervoor beschreven commando weer gestart worden.

Bestanden

- /usr/local/sbin/dhcpd

  dhcpd is statisch gelinkt en staat in /usr/local/sbin. In de hulppagina voor dhcpd(8) die meekomt met de port staat meer informatie over dhcpd.

- /usr/local/etc/dhcpd.conf

dhcpd heeft een instellingenbestand, /usr/local/etc/dhcpd.conf, nodig voordat de daemon diensten aan cliënten kan leveren. Het bestand moet alle informatie bevatten die aan cliënten gegeven moet worden en de informatie die nodig is voor het draaien van de dienst. Dit instellingenbestand staat beschreven in de hulppagina voor dhcpd.conf(5) die meekomt met de port.

- /var/db/dhcpd.leases

  De DHCP-server houdt in dit bestand een database bij met leases die zijn uitgegeven en die naar een logboek worden geschreven. In de hulppagina dhcpd.leases(5) die bij de port zit wordt dit uitvoeriger beschreven.

- /usr/local/sbin/dhcrelay

  dhcrelay wordt in uitgebreidere omgevingen gebruikt waar de ene DHCP-server een verzoek van een cliënt naar een andere DHCP-server op een ander netwerk doorstuurt. Als deze functionaliteit nodig is, kan die beschikbaar komen door de port net/isc-dhcp42-relay te installeren. De hulppagina voor dhcrelay(8) die bij de port zit bevat meer details.

27.6. Domeinnaamsysteem (DNS)

27.6.1. Overzicht

FreeBSD gebruikt standaard een versie van BIND (Berkeley Internet Name Domain), wat de meest gebruikte implementatie van het DNS-protocol is. DNS is het protocol waarmee namen aan IP-adressen gebonden worden en vice versa. Zo wordt bijvoorbeeld op een zoekopdracht voor `www.FreeBSD.org` geantwoord met het IP-adres van de webserver van het FreeBSD Project en op een zoekopdracht voor `ftp.FreeBSD.org` wordt geantwoord met het IP-adres van de bijbehorende FTP-machine. Het tegenovergestelde kan ook gebeuren. Een zoekopdracht voor een IP-adres kan de bijbehorende hostnaam opleveren. Het is niet nodig om een naamserver te draaien om op een systeem zoekopdrachten met DNS uit te voeren.

FreeBSD wordt momenteel standaard geleverd met de BIND9 DNS-serversoftware. Onze installatie biedt verbeterde beveilingsmogelijkheden, een nieuwe indeling van het bestandssysteem en geautomatiseerde configuratie van chroot(8).

DNS wordt op Internet onderhouden door een enigszins complex systeem van autoritaire root, Top Level Domain (TLD), en andere kleinschaligere naamservers die individuele domeininformatie hosten en cachen.

Op dit moment wordt BIND beheerd door het Internet Systems Consortium https://www.isc.org/.

27.6.2. Terminologie

Om dit document te begrijpen moeten een aantal termen gerelateerd aan DNS begrepen worden.

Term	Definitie
Voorwaartse DNS	Het afbeelden van hostnamen op IP-adressen.

Term	Definitie
Herkomst (origin)	Verwijst naar het domein dat door een bepaald zonebestand wordt gedekt.
named, BIND	Vaak gebruikte namen voor het naamserverpakket BIND in FreeBSD.
Resolver	Een systeemproces waarmee een machine zoekopdrachten om zoneinformatie aan een naamserver geeft.
Reverse DNS	Het afbeelden van IP-adressen op hostnamen.
Rootzone	Het begin van de Internet zonehiërarchie. Alle zones vallen onder de rootzone, net zoals alle bestanden in een bestandssysteem onder de rootmap vallen.
Zone	Een individueel domein, subdomein of een deel van de DNS die door dezelfde autoriteit wordt beheerd.

Voorbeelden van zones:

- `.` is hoe de rootzone normaliter in de documentatie genoemd wordt.
- `org.` is een Top Level Domain (TLD) onder de rootzone.
- `example.org.` is een zone onder het TLD `org.`.
- `1.168.192.in-addr.arpa` is een zone die naar alle IP-adressen verwijst die onder de IP-adresruimte `192.168.1.*` vallen.

Zoals te zien is staat het specifiekere deel van een hostnaam aan de linkerkant. Zo is bijvoorbeeld `example.org.` specifieker dan `org.` en is `org.` specifieker dan de rootzone. De indeling van ieder deel van een hostnaam lijkt veel op een bestandssysteem: de map /dev valt onder de root, enzovoort.

27.6.3. Redenen om een naamserver te draaien

Naamservers bestaan in het algemeen in twee smaken: autoratieve naamservers en caching (ook bekend als resolving) naamservers.

Er is een autoratieve naamserver nodig als:

- Het gewenst is om DNS-informatie aan te bieden aan de wereld om met autoriteit op verzoeken te antwoorden.
- Een domein, zoals `example.org`, is geregistreerd en er IP-adressen aan hostnamen die daaronder liggen toegewezen moeten worden.
- Een IP-adresblok omgekeerde DNS-ingangen nodig heeft (IP naar hostnaam).
- Een omgekeerde of tweede naamserver, die een slaaf wordt genoemd, moet antwoorden op verzoeken.

Er is een caching naamserver nodig als:

- Een lokale DNS-server kan cachen en wellicht sneller kan antwoorden dan een naamserver die verder weg staat.

Als er een verzoek wordt gedaan voor `www.FreeBSD.org`, dan doet de resolver meestal een verzoek bij de naamserver van de ISP die de uplink levert en ontvangt daarop een antwoord. Met een lokale, caching DNS-server hoeft het verzoek maar één keer door de caching DNS-server naar de buitenwereld gedaan te worden. Voor aanvullende verzoeken hoeft niet buiten het lokale netwerk te gaan omdat het al lokaal in de cache staat.

27.6.4. Hoe het werkt

De daemon BIND heet in FreeBSD named.

Bestand	Beschrijving
named(8)	De daemon BIND.
rndc(8)	Naamserverbeheerprogramma.
/etc/namedb	Map waar zoneinformatie van BIND staat.
/etc/namedb/named.conf	Instellingenbestand van de daemon.

Afhankelijk van hoe en gegeven zone op de server is geconfigureerd, staan de bestanden gerelateerd aan die zone in de submappen master, slave, of dynamic van de map /etc/namedb. Deze bestanden bevatten de DNS-informatie die door de naamserver als antwoord op zoekopdrachten gegeven zal worden.

27.6.5. BIND starten

Omdat BIND standaard wordt geïnstalleerd, is het instellen relatief eenvoudig.

De standaardconfiguratie van named is die van een eenvoudige resolverende naamserver, draaiende in een chroot(8)-omgeving, en beperkt tot het luisteren op het lokale IPv4-teruglusadres (127.0.0.1). Gebruik het volgende commando om de server eenmaal met deze configuratie te starten:

```
# service named onestart
```

Om er zeker van te zijn dat de daemon named elke keer bij het opstarten gestart wordt, moet de volgende regel in /etc/rc.conf gezet worden:

```
named_enable="YES"
```

Het is duidelijk dat er vele instelopties voor /etc/namedb/named.conf zijn die buiten het bereik van dit document vallen. Als u echter geïnteresseerd bent in de opstartopties voor named op FreeBSD, bekijk dan de `named_*`-vlaggen in /etc/defaults/rc.conf en raadpleeg de handleidingpagina rc.conf(5).

De sectie Gebruik van rc met FreeBSD is ook nuttig om te lezen.

27.6.6. Instellingenbestanden

Instellingenbestanden voor named bevinden zich momenteel in /etc/namedb en moeten gewijzigd worden voor gebruik, tenzij er alleen een eenvoudige resolver nodig is. Hier vindt de meeste configuratie plaats.

/etc/namedb/named.conf

```
// $FreeBSD$
//
// In de handleidingpagina's named.conf(5) en named(8), en in de
// documentatie in /usr/shared/doc/bind9 zijn meer details te vinden.
//
// Voor het opzetten van een autoratieve server is een grondig begrip
// van de werking van DNS noodzakelijk.  Zelfs eenvoudige fouten kunnen // de werking
verstoren voor beïnvloede partijen of veel onnodig
// Internetverkeer veroorzaken.

options {
       // Alle namen van bestanden en paden zijn relatief aan de chroot-map,
       // indien aanwezig, en moeten volledig gekwalificeerd zijn.
       directory       "/etc/namedb/working";
       pid-file        "/var/run/named/pid"
       dump-file       "/var/dump/named_dump.db"
       statistics-file "/var/stats/named.stats"

// Als named alleen als een lokale resolver gebruikt wordt, is dit een
// veilige standaardinstelling.  Om named toegang tot het netwerk te
// verschaffen, dient deze optie gecommentarieerd te worden, het
// juiste IP-adres opgegeven te worden, of dient deze optie verwijderd
// te worden.
       listen-on       { 127.0.0.1; };

// Als u IPv6 aan heeft staan op dit systeem, dient deze optie
// uitgecommentarieerd te worden om als lokale resolver te dienen.  Om
// toegang tot het netwerk te verschaffen, dient een IPv6-adres of het
// sleutelwoord "any" gegeven te worden.
//     listen-on-v6    { ::1; };

// Deze zones zijn reeds opgenomen door de lege zones die hieronder
// staan.  Als u de gerelateerde lege zones hieronder verwijdert,
// dienen deze regels uitgecommentarieerd te worden.
        disable-empty-zone "255.255.255.255.IN-ADDR.ARPA";
        disable-empty-zone
"0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.ARPA";
        disable-empty-zone
"1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.ARPA";
```

```
// Als er een DNS-server beschikbaar is bij een upstream provider dan
// kan het IP-adres op de regel hieronder ingegeven worden en kan die
// geactiveerd worden.  Hierdoor wordt voordeel gehaald uit de cache,
// waardoor het algehele DNS-verkeer op het Internet vermindert.
/*
       forwarders {
               127.0.0.1;
       };

*/

// Als de 'forwarders'-clausule niet leeg is, is de standaard om "forward
// first" te gebruiken, welke terug zal vallen op het versturen van een
// verzoek naar uw lokale server als de naamservers in 'forwarders' het
// antwoord niet weten.  Als alternatief kunt u uw naamserver dwingen om
// nooit zelf verzoeken in te dienen door de volgende regel aan te
// zetten:
//     forward only;

// Als u forwarding automatisch wilt configureren gebaseerd op de regels
// in /etc/resolv.conf, verwijder dan het commentaar van de volgende
// regel en stel in /etc/rc.conf named_auto_forward=yes in.  U kunt ook
// named_auto_forward_only aanzetten (het effect hiervan is hierboven
// beschreven).
//      include "/etc/namedb/auto_forward.conf";
```

Zoals al in het commentaar staat kan van een cache in de uplink geprofiteerd worden als `forwarders` ingeschakeld worden. Onder normale omstandigheden maakt een naamserver recursief verzoeken tot het Internet op zoek naar zekere naamservers tot er een antwoord komt waar het naar op zoek is. Door de bovenstaande optie in te schakelen wordt eerst de uplink naamserver (of de opgegeven naamserver) gevraagd, waardoor er gebruik gemaakt kan worden van de cache van die server. Als die uplink naamserver een drukke, snelle naamserver is, kan het erg de moeite waard zijn om dit aan te zetten.



`127.0.0.1` werkt hier *niet*. Verander dit IP-adres in een naamserver in de uplink.

```
/*
      Moderne versies van BIND gebruiken standaard een random
      UDP-poort voor elk uitgaand verzoek om de kans op cache
      poisoning drastisch te verminderen.  Alle gebruikers wordt met
      klem verzocht om deze mogelijkheid te gebruiken en hun
      firewalls overeenkomstig aan te passen.

      ALS EEN LAATSTE UITVLUCHT om een beperkende firewall te
      omzeilen kunt u proberen om onderstaande optie aan te zetten.
      Het gebruik van deze optie vermindert uw kans om een cache
      poisoning aanval te weerstaan aanzienlijk, en dient indien
      mogelijk te worden vermeden.
```

```
      Vervang NNNNN in het voorbeeld door een getal tussen 49160 en
      65530.
    */
    // query-source address * port NNNNN;
};

// Als er een lokale naamserver wordt gebruikt, vergeet dan niet om
// eerst 127.0.0.1 in /etc/resolv.conf te zetten zodat die gevraagd
// wordt.  Controleer ook dat het in /etc/rc.conf is aangezet.

// Het traditionele root-hint-mechanisme.  Gebruik dit OF de
// onderstaande slaafzones.
zone "." { type hint; file "/etc/namedb/named.root"; };

/*  Het slaaf maken van de volgende zones vanaf de root-naamservers
    heeft een aantal aanzienlijke voordelen:
    1. Snellere lokale resolutie voor uw gebruikers
    2. Geen vals verkeer dat vanaf uw netwerk naar de roots wordt verzonden
    3. Betere weerstand tegen elke mogelijk falen van de rootserver/DDoS

    Wel is het zo dat deze methode meer toezicht vraagt dan het
    hintbestand om er zeker van te zijn dat een onverwachte
    faalmodus uw server niet heeft lamgelegd.  Naamservers die
    veel clienten serveren zullen meer voordeel uit deze aanpak
    halen dan individuele hosts.  Met zorg gebruiken.

    Verwijder het commentaar uit de onderstaande regels en
    commentarieer de bovenstaande hintzone om dit mechanisme te
    gebruiken.

    Zoals gedocumenteerd op http://dns.icann.org/services/axfr/ zijn deze
    zones: "." (de root), ARPA, IN-ADDR.ARPA, IP6.ARPA en ROOT-SERVERS.NET
    beschikbaar voor AXFR van deze servers op IPv4 en IPv6:
    xfr.lax.dns.icann.org, xfr.cjr.dns.icann.org
*/

zone "." {
    type slave;
    file "/etc/namedb/slave/root.slave";
    masters {
        192.5.5.241;    // F.ROOT-SERVERS.NET.
    };
    notify no;
};

zone "arpa" {
    type slave;
    file "/etc/namedb/slave/arpa.slave";
    masters {
        192.5.5.241;    // F.ROOT-SERVERS.NET.
    };
```

```
    notify no;
};

/*  Het lokaal serveren van de volgende zones voorkomt dat enig
    verzoek voor deze zones uw netwerk verlaat en naar de
    root-naamservers gaat.  Dit heeft twee aanzienlijke voordelen:
    1. Snellere lokale resolutie voor uw gebruikers
    2. Er zal geen vals verkeer vanaf uw netwerk naar de roots worden verzonden
*/
// RFCs 1912 en 5735 (en BCP32 voor localhost)
zone "localhost"    { type master; file "/etc/namedb/master/localhost-forward.db"; };
zone "127.in-addr.arpa" { type master; file "/etc/namedb/master/localhost-reverse.db";
};
zone "255.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };

// RFC 1912-stijl zone voor IPv6 localhost adres
zone "0.ip6.arpa"   { type master; file "/etc/namedb/master/localhost-reverse.db"; };

// "Dit" netwerk (RFCs 1912 en 5735)
zone "0.in-addr.arpa"   { type master; file "/etc/namedb/master/empty.db"; };

// Netwerken voor privaat gebruik (RFC 1918 en 5735)
zone "10.in-addr.arpa"  { type master; file "/etc/namedb/master/empty.db"; };
zone "16.172.in-addr.arpa"  { type master; file "/etc/namedb/master/empty.db"; };
zone "17.172.in-addr.arpa"  { type master; file "/etc/namedb/master/empty.db"; };
zone "18.172.in-addr.arpa"  { type master; file "/etc/namedb/master/empty.db"; };
zone "19.172.in-addr.arpa"  { type master; file "/etc/namedb/master/empty.db"; };
zone "20.172.in-addr.arpa"  { type master; file "/etc/namedb/master/empty.db"; };
zone "21.172.in-addr.arpa"  { type master; file "/etc/namedb/master/empty.db"; };
zone "22.172.in-addr.arpa"  { type master; file "/etc/namedb/master/empty.db"; };
zone "23.172.in-addr.arpa"  { type master; file "/etc/namedb/master/empty.db"; };
zone "24.172.in-addr.arpa"  { type master; file "/etc/namedb/master/empty.db"; };
zone "25.172.in-addr.arpa"  { type master; file "/etc/namedb/master/empty.db"; };
zone "26.172.in-addr.arpa"  { type master; file "/etc/namedb/master/empty.db"; };
zone "27.172.in-addr.arpa"  { type master; file "/etc/namedb/master/empty.db"; };
zone "28.172.in-addr.arpa"  { type master; file "/etc/namedb/master/empty.db"; };
zone "29.172.in-addr.arpa"  { type master; file "/etc/namedb/master/empty.db"; };
zone "30.172.in-addr.arpa"  { type master; file "/etc/namedb/master/empty.db"; };
zone "31.172.in-addr.arpa"  { type master; file "/etc/namedb/master/empty.db"; };
zone "168.192.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };

// Lokale link/APIPA (RFCs 3927 en 5735)
zone "254.169.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };

// IETF protocol-toewijzingen (RFCs 5735 en 5736)
zone "0.0.192.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };

// TEST-NET-[1-3] voor documentatie (RFCs 5735 en 5737)
zone "2.0.192.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "100.51.198.in-addr.arpa"  { type master; file "/etc/namedb/master/empty.db"; };
zone "113.0.203.in-addr.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
```

```
// IPv6-bereik voor documentatie (RFC 3849)
zone "8.b.d.0.1.0.0.2.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; };

// Domeinnamen voor documentatie en testen (BCP 32)
zone "test" { type master; file "/etc/namedb/master/empty.db"; };
zone "example" { type master; file "/etc/namedb/master/empty.db"; };
zone "invalid" { type master; file "/etc/namedb/master/empty.db"; };
zone "example.com" { type master; file "/etc/namedb/master/empty.db"; };
zone "example.net" { type master; file "/etc/namedb/master/empty.db"; };
zone "example.org" { type master; file "/etc/namedb/master/empty.db"; };

// Router benchmarken (RFC 2544 en 5735)
zone "18.198.in-addr.arpa"  { type master; file "/etc/namedb/master/empty.db"; }
zone "19.198.in-addr.arpa"  { type master; file "/etc/namedb/master/empty.db"; }

// Gereserveerd door IANA - oude ruimte van klasse E (RFC 5735)
zone "240.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; }
zone "241.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; }
zone "242.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; }
zone "243.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; }
zone "244.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; }
zone "245.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; }
zone "246.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; }
zone "247.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; }
zone "248.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; }
zone "249.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; }
zone "250.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; }
zone "251.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; }
zone "252.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; }
zone "253.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; }
zone "254.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; }

// Niet-toegewezen IPv6-adressen (RFC 4291)
zone "1.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "2.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "3.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "4.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "5.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "6.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "7.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "8.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "9.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "a.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "b.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "c.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "d.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "e.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "0.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; }
zone "1.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; }
zone "2.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; }
```

```
zone "3.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; }
zone "4.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; }
zone "5.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; }
zone "6.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; }
zone "7.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; }
zone "8.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; }
zone "9.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; }
zone "a.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; }
zone "b.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; }
zone "0.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "1.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "2.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "3.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "4.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "5.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "6.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "7.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }

// IPv6 ULA (RFC 4193)
zone "c.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; }
zone "d.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; }

// IPv6 lokale link (RFC 4291)
zone "8.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "9.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "a.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "b.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }

// IPv6 verouderde site-lokale adressen (RFC 3879)
zone "c.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "d.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "e.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }
zone "f.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; }

// IP6.INT is verouderd (RFC 4159)
zone "ip6.int"  { type master; file "/etc/namedb/master/empty.db"; }

// NB: De IP-adressen hieronder zijn bedoeld als voorbeeld en dienen
//     niet gebruikt te worden!
//
// Voorbeeld instellingen voor slaafzones.  Het kan handig zijn om
// tenminste slaaf te worden voor de zone waar de host onderdeel van
// uitmaakt.  Bij uw netwerkbeheerder kan het IP-adres van de
// verantwoordelijke meester-naamserver nagevraagd worden.
//
// Vergeet niet om de omgekeerde lookup-zone op te nemen!
// Dit is genoemd na de eerste bytes van het IP-adres, in omgekeerde
// volgorde, met daarachter ".IN-ADDR.ARPA", of "IP6.ARPA" voor IPv6.
//
// Het is van groot belang om de werking van DNS en BIND te begrijpen
// voordat er een meester-zone wordt opgezet.  Er zijn nogal wat
```

```
// onverwachte valkuilen.  Het opzetten van een slaafzone is
// gewoonlijk eenvoudiger.
//
// NB: Zet de onderstaande voorbeelden niet blindelings aan. :-)
// Gebruik in plaats hiervan echte namen en adressen.
/* Een voorbeeld van een dynamische zone
key "exampleorgkey" {
       algorithm hmac-md5;
       secret "sf87HJqjkqh8ac87a02lla==";
};

zone "example.org" {
       type master;
       allow-update {
               key "exampleorgkey";
       };
       file "/etc/namedb/dynamic/example.org";
};
*/

/* Voorbeeld van een omgekeerde slaafzone
zone "1.168.192.in-addr.arpa" {
       type slave;
       file "/etc/namedb/slave/1.168.192.in-addr.arpa";
       masters {
               192.168.1.1;
       };
};
*/
```

In named.conf zijn dit voorbeelden van slaafregels voor een voorwaartse en een omgekeerde zone.

Voor iedere nieuwe zone die wordt aangeboden dient een nieuwe instelling voor de zone aan named.conf toegevoegd te worden.

De eenvoudigste instelling voor de zone `example.org` kan er als volgt uitzien:

```
zone "example.org" {
    type master;
    file "master/example.org";
};
```

De zone is een master, zoals aangegeven door het statement `type`, waarvan de zoneinformatie in /etc/namedb/example.org staat, zoals het statement `file` aangeeft.

```
zone "example.org" {
    type slave;
    file "slave/example.org";
```

```
};
```

In het geval van de slaaf wordt de zoneinformatie voor een zone overgedragen van de master naamserver en opgeslagen in het ingestelde bestand. Als de masterserver het niet meer doet of niet bereikbaar is, dan heeft de slaveserver de overgedragen zoneinformatie nog en kan het die aanbieden.

Zonebestanden

Een voorbeeldbestand voor een masterzone voor `example.org` (bestaande binnen /etc/namedb/master/example.org) ziet er als volgt uit:

```
$TTL 3600        ; 1 uur standaard TTL
example.org.     IN      SOA      ns1.example.org. admin.example.org. (
                                  2006051501      ; Serienummer
                                  10800           ; Verversen
                                  3600            ; Opnieuw proberen
                                  604800          ; Verlopen
                                  300             ; Negatieve antwoord-TTL
                         )

; DNS Servers
                 IN      NS      ns1.example.org.
                 IN      NS      ns2.example.org.

; MX Records
                 IN      MX 10   mx.example.org.
                 IN      MX 20   mail.example.org.

                 IN      A       192.168.1.1

; Machinenamen
localhost        IN      A       127.0.0.1
ns1              IN      A       192.168.1.2
ns2              IN      A       192.168.1.3
mail             IN      A       192.168.1.4
mx               IN      A       192.168.1.5

; Aliases
www              IN      CNAME   example.org.
```

Iedere hostnaam die eindigt op een "." is een exacte hostnaam, terwijl alles zonder een "." op het einde relatief is aan de oorsprong. Zo wordt `ns1` bijvoorbeeld vertaald naar `ns1.example.org.`.

De regels in een zonebestand volgen de volgende opmaak:

```
recordnaam      IN recordtype   waarde
```

De meest gebruikte DNS-records:

SOA
: begin van autoriteit (start of authority)

NS
: een bevoegde (autoratieve) name server

A
: een hostadres

CNAME
: de canonieke naam voor een alias

MX
: mail exchanger

PTR
: een domeinnaam pointer (gebruikt in omgekeerde DNS)

```
example.org. IN SOA ns1.example.org. admin.example.org. (
                        2006051501      ; Serienummer
                        10800           ; Ververs na 3 uur
                        3600            ; Opnieuw proberen na 1 uur
                        604800          ; Verlopen na 1 week
                        300             ; Negatieve antwoord-TTL
```

`example.org.`
: de domeinnaam, ook de oorsprong voor dit zonebestand.

`ns1.example.org.`
: de primaire/bevoegde naamserver voor deze zone.

`admin.example.org.`
: de persoon die verantwoordelijk is voor deze zone, emailadres met "@" vervangen. admin@example.org wordt `admin.example.org`.

`2006051501`
: het serienummer van het bestand. Dit moet iedere keer als het zonebestand wordt aangepast opgehoogd worden. Tegenwoordig geven veel beheerders de voorkeur aan de opmaak `yyyymmddrr` voor het serienummer. `2006051501` betekent dan dat het voor het laatst is aangepast op 15-05-2006, de laatste `01` betekent dat het zonebestand die dag voor het eerst is aangepast. Het serienummer is belangrijk omdat het slaafnaamservers aangeeft dat een zone is bijgewerkt.

```
        IN NS           ns1.example.org.
```

Hierboven staat een NS-regel. Voor iedere naamserver die bevoegde antwoorden moet geven voor de zone hoort er zo'n regel te zijn.

```
localhost       IN      A       127.0.0.1
ns1             IN      A       192.168.1.2
ns2             IN      A       192.168.1.3
mx              IN      A       192.168.1.4
mail            IN      A       192.168.1.5
```

Een A-record geeft een machinenaam aan. Hierboven is te zien dat `ns1.example.org` zou resolven naar `192.168.1.2`.

```
        IN      A       192.168.1.1
```

Deze regel kent IP-adres `192.168.1.1` toe aan de huidige oorsprong, in dit geval `example.org`.

```
www             IN CNAME        @
```

Een canoniek naamrecord wordt meestal gebruikt voor het geven van aliassen aan een machine. In het voorbeeld is `www` een alias naar de "master" machine waarvan de naam gelijk is aan de domeinnaam `example.org` (`192.168.1.1`). CNAME's kunnen nooit samen met een ander soort record voor dezelfde hostnaam gebruikt worden.

```
                IN MX   10      mail.example.org.
```

MX records geven aan welke mailservers verantwoordelijk zijn voor het afhandelen van inkomende mail voor de zone. `mail.example.org` is de hostnaam van een mailserver en 10 is de prioriteit voor die mailserver.

Het is mogelijk meerdere mailservers in te stellen met prioriteiten 10, 20, enzovoorts. Een mailserver die probeert mail af te leveren voor `example.org` probeert dat eerst bij de MX met de hoogste prioriteit (het record met het laagste prioriteitsnummer), daarna de tweede hoogste, enzovoort, totdat de mail afgeleverd kan worden.

Voor in-addr.arpa zonebestanden (omgekeerd DNS) wordt dezelfde opmaak gebruikt, maar dan met PTR-regels in plaats van A of CNAME.

```
$TTL 3600

1.168.192.in-addr.arpa. IN SOA ns1.example.org. admin.example.org. (
                        2006051501      ; Serienummer
                        10800           ; Ververs
                        3600            ; Opnieuw proberen
                        604800          ; Verlopen
                        300 )           ; Negatieve antwoord-TTL
```

```
        IN      NS      ns1.example.org.
        IN      NS      ns2.example.org.

1       IN      PTR     example.org.
2       IN      PTR     ns1.example.org.
3       IN      PTR     ns2.example.org.
4       IN      PTR     mx.example.org.
5       IN      PTR     mail.example.org.
```

Dit bestand geeft de juiste IP-adressen voor hostnamen in het voorbeelddomein hierboven.

Het is het vernoemen waard dat alle namen aan de rechterkant van een PTR-record volledig gekwalificeerd dienen te zijn (i.e., met een "." eindigen).

27.6.7. Caching naamserver

Een caching naamserver is een naamserver wiens primaire rol het oplossen van recursieve verzoeken is. Het dient simpelweg zelf verzoeken in en onthoudt de antwoorden voor later gebruik.

27.6.8. DNSSEC

Domain Name Security System Extensions, ofwel DNSSEC, is een verzameling van specificaties om resolvende naamservers te beschermen tegen valse DNS-gegevens, zoals vervalste DNS-records. Door digitale handtekeningen te gebruiken kan een resolver de integriteit van een record controleren. Merk op dat DNSSEC alleen integriteit biedt via het digitaal ondertekenen van het Resource Record (RRs). Het biedt noch betrouwbaarheid noch bescherming tegen onjuiste aannames van eindgebruikers. Dit betekent dat het mensen niet kan beschermen tegen het bezoeken van `example.net` in plaats van `example.com`. Het enige wat DNSSEC doet is authenticeren dat de gegevens niet tijdens het transport zijn gecompromitteerd. De beveiliging van DNSSEC is een belangrijke stap in het beveiligen van het internet in het algemeen. De relevante RFCs zijn een goed beginpunt voor meer gedetailleerde gegevens over hoe DNSSEC werkt. Raadpleeg de lijst in Verder lezen.

De volgende secties laten zien hoe DNSSEC voor een autoratieve DNS-server en een recursieve (of caching) DNS-server die BIND 9 draait kan worden bewerkstelligd. Hoewel alle versies van BIND 9 DNSSEC ondersteunen, is tenminste versie 9.6.2 nodig om gebruik te kunnen maken van de ondertekende rootzones tijdens het valideren van DNS-verzoeken. Dit komt doordat eerdere versies de benodigde algoritmes om validatie met de sleutel voor de rootzone te uit te voeren niet hebben. Het wordt sterk aangeraden om de nieuwste versie van BIND 9.7 te gebruiken om gebruik te kunnen maken van automatische sleutel-updates voor de rootsleutel en van andere mogelijkheden om zones ondertekend en sleutel up-to-date te houden. Wanneer configuraties tussen 9.6.2 en 9.7 en later verschillen, zullen deze worden toegelicht.

Configuratie van een recursieve DNS-server

Het aanzetten van DNSSEC-validatie van verzoeken die door een recursieve DNS-server worden uitgevoerd heeft enkele aanpassingen aan named.conf nodig. Voordat deze wijzigingen worden gemaakt dient de rootzone-sleutel, of vertrouwensanker, te worden opgehaald. Momenteel is de

rootzone-sleutel niet beschikbaar in een bestandsformaat dat BIND begrijpt, dus moet het handmatig in het juiste formaat omgezet worden. De sleutel zelf kan verkregen worden door de rootzone ervoor met dig te ondervragen. Door

```
% dig +multi +noall +answer DNSKEY . > root.dnskey
```

te draaien, wordt de sleutel in root.dnskey opgeslagen. De inhoud dient er ongeveer als volgt uit te zien:

```
. 93910 IN DNSKEY 257 3 8 (
    AwEAAagAIKlVZrpC6Ia7gEzahOR+9W29euxhJhVVLOyQ
    bSEW0O8gcCjFFVQUTf6v58fLjwBd0YI0EzrAcQqBGCzh
    /RStIoO8g0NfnfL2MTJRkxoXbfDaUeVPQuYEhg37NZWA
    JQ9VnMVDxP/VHL496M/QZxkjf5/Efucp2gaDX6RS6CXp
    oY68LsvPVjR0ZSwzz1apAzvN9dlzEheX7ICJBBtuA6G3
    LQpzW5hOA2hzCTMjJPJ8LbqF6dsV6DoBQzgul0sGIcGO
    Yl7OyQdXfZ57relSQageu+ipAdTTJ25AsRTAoub8ONGc
    LmqrAmRLKBP1dfwhYB4N7knNnulqQxA+Uk1ihz0=
    ) ; key id = 19036
. 93910 IN DNSKEY 256 3 8 (
    AwEAAcaGQEA+OJmOzfzVfoYN249JId7gx+OZMbxy69Hf
    UyuGBbRN0+HuTOpBxxBCkNOL+EJB9qJxt+0FEY6ZUVjE
    g58sRr4ZQ6Iu6b1xTBKgc193zUARk4mmQ/PPGxn7Cn5V
    EGJ/1h6dNaiXuRHwR+7oWh7DnzkIJChcTqlFrXDW3tjt
    ) ; key id = 34525
```

Schrik niet als de verkregen sleutels anders zijn dan in dit voorbeeld. Ze kunnen zijn veranderd nadat deze instructies voor het laatst waren bijgewerkt. De uitvoer bevat in feite twee sleutels. De eerste sleutel, met de waarde 257 na het DNSKEY-recordtype, is degene die nodig is. Deze waarde geeft aan dat dit een Secure Entry Point (SEP) is, beter bekend als een Key Signing Key (KSK). De tweede sleutel, met de waarde 256, is een deelsleutel, beter bekend als een Zone Signing Key (ZSK). Meer over de verschillende soorten sleutels komt aan bod in Configuratie van een autoratieve DNS-server.

Nu moet de sleutel gecontroleerd en geformatteerd worden zodat BIND deze kan gebruiken. Maak om de sleutel te controleren een DS - RR-paar aan. Maak een bestand aan dat deze RRs bevat aan met

```
% dnssec-dsfromkey -f root-dnskey . > root.ds
```

Deze records gebruiken respectievelijk SHA-1 en SHA-256, en dienen er als het volgende voorbeeld uit te zien, waarbij het langere record SHA-256 gebruikt.

```
. IN DS 19036 8 1 B256BD09DC8DD59F0E0F0D8541B8328DD986DF6E
. IN DS 19036 8 2 49AAC11D7B6F6446702E54A1607371607A1A41855200FD2CE1CDDE32F24E8FB5
```

Het SHA-256 RR kan nu worden vergeleken met de digest in https://data.iana.org/root-anchors/root-anchors.xml. Om er absoluut zeker van te zijn dat er niet geknoeid is met de sleutel kunnen de gegevens in het XML-bestand worden gecontroleerd met de PGP-handtekening in https//data.iana.org/root-anchors/root-anchors.asc.

Vervolgens dient de sleutel juist geformateerd te worden. Dit verschilt een beetje tussen versie 9.6.2 en versie 9.7 en later van BIND. In versie 9.7 is ondersteuning toegevoegd om automatisch veranderingen aan de sleutel te volgen en deze bij te werken indien nodig. Dit wordt gedaan met `managed-keys` zoals in het volgende voorbeeld te zien is. Als de oudere versie gebruikt wordt, wordt de sleutel toegevoegd met een commando `trusted-keys` en dient deze handmatig bijgewerkt te worden. Voor BIND 9.6.2 ziet het formaat er uit als:

```
trusted-keys {
    "." 257 3 8
    "AwEAAagAIKlVZrpC6Ia7gEzahOR+9W29euxhJhVVLOyQbSEW0O8gcCjF
    FVQUTf6v58fLjwBd0YI0EzrAcQqBGCzh/RStIoO8g0NfnfL2MTJRkxoX
    bfDaUeVPQuYEhg37NZWAJQ9VnMVDxP/VHL496M/QZxkjf5/Efucp2gaD
    X6RS6CXpoY68LsvPVjR0ZSwzz1apAzvN9dlzEheX7ICJBBtuA6G3LQpz
    W5hOA2hzCTMjJPJ8LbqF6dsV6DoBQzgul0sGIcGOYl7OyQdXfZ57relS
    Qageu+ipAdTTJ25AsRTAoub8ONGcLmqrAmRLKBP1dfwhYB4N7knNnulq
    QxA+Uk1ihz0=";
};
```

Voor versie 9.7 ziet het formaat er echter zo uit:

```
managed-keys {
    "." initial-key 257 3 8
    "AwEAAagAIKlVZrpC6Ia7gEzahOR+9W29euxhJhVVLOyQbSEW0O8gcCjF
    FVQUTf6v58fLjwBd0YI0EzrAcQqBGCzh/RStIoO8g0NfnfL2MTJRkxoX
    bfDaUeVPQuYEhg37NZWAJQ9VnMVDxP/VHL496M/QZxkjf5/Efucp2gaD
    X6RS6CXpoY68LsvPVjR0ZSwzz1apAzvN9dlzEheX7ICJBBtuA6G3LQpz
    W5hOA2hzCTMjJPJ8LbqF6dsV6DoBQzgul0sGIcGOYl7OyQdXfZ57relS
    Qageu+ipAdTTJ25AsRTAoub8ONGcLmqrAmRLKBP1dfwhYB4N7knNnulq
    QxA+Uk1ihz0=";
};
```

De rootsleutel kan nu aan named.conf worden toegevoegd, ofwel direct of door een bestand dat de sleutel bevat te includen. Stel na deze stappen BIND in zodat het DNSSEC-validatie uitvoert op verzoeken door named.conf te bewerken en het volgende aan de directief `options` toe te voegen:

```
dnssec-enable yes;
dnssec-validation yes;
```

Om te controleren dat het ook echt werkt, kan dig gebruikt worden om een verzoek op een ondertekende zone uit te voeren met de zojuist geconfigureerde resolver. Een succesvol antwoord zal de vlag `AD` bevatten om aan te geven dat de gegevens zijn geautenticeerd. Een verzoek als

```
% dig @resolver +dnssec se ds
```

zou het DSRR paar voor de `.se`-zone moeten teruggeven. In de sectie `flags:` moet de vlag `AD` te zien zijn, als in:

```
...
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1
...
```

De resolver is nu in staat om DNS-verzoeken te autenticeren.

Configuratie van een autoratieve DNS-server

Om een autoratieve naamserver een met DNSSEC ondertekende zone te laten serveren is wat meer werk nodig. Een zone wordt ondertekend met cryptografische sleutels die aangemaakt moeten worden. Het is mogelijk om hier slechts één sleutel voor te gebruiken. De methode die de voorkeur verdient is echter om een sterke, goed beschermde Key Signing Key (KSK) die niet vaak wordt geroteerd en een Zone Signing Key (ZSK) die vaker wordt geroteerd te hebben. Informatie over aanbevolen procedures staat in RFC 4641: DNSSEC Operational Practices. Procedures betreffende de rootzone staan in DNSSEC Practice Statement for the Root Zone KSK operator en DNSSEC Practice Statement for the Root Zone ZSK operator. De KSK wordt gebruikt om een autoriteitsketen voor de te valideren gegevens op te bouwen en wordt daarom ook een Secure Entry Point (SEP)-sleutel genoemd. Een bericht-digest van deze sleutel, dat Delegation Signer (DS)-record genoemd wordt, moet gepubliceerd zijn in de ouderzone om een vertrouwensketen op te bouwen. Hoe dit bereikt wordt hangt af van de eigenaar van de ouderzone. De ZSK wordt gebruikt om de zone te ondertekenen, en hoeft alleen daar gepubliceerd te worden.

Om DNSSEC aan te zetten voor de zone `example.com` zoals beschreven in de voorgaande voorbeelden, dient als eerste dnssec-keygen gebruikt te worden om het sleutelpaar met de KSK en ZSK te genereren. Dit sleutelpaar kan verschillende cryptografische algoritmes gebruiken. Het wordt aanbevolen om RSA/SHA-256 voor de sleutels te gebruiken, een sleutellengte van 2048 bits zou voldoende moeten zijn. Om de KSK voor `example.com` te genereren:

```
% dnssec-keygen -f KSK -a RSASHA256 -b 2048 -n ZONE example.com
```

en om de ZSK te genereren:

```
% dnssec-keygen -a RSASHA256 -b 2048 -n ZONE example.com
```

dnssec-keygen maakt twee bestanden, de publieke en private sleutels in bestanden met namen als Kexample.com.+005+nnnnn.key (publiek) en Kexample.com.+005+nnnnn.private (privaat). Het gedeelte `nnnnn` van de bestandsnaam is een sleutel-ID van vijf cijfers. Houd bij welke sleutel-ID bij welke sleutel hoort. Dit is in het bijzonder van belang wanneer er meerdere sleutels per zone zijn. Het is ook mogelijk om de sleutels te hernoemen. Voor elk KSK-bestand:

```
% mv Kexample.com.+005+nnnnn.key Kexample.com.+005+nnnn.KSK.key
% mv Kexample.com.+005+nnnnn.private Kexample.com.+005+nnnnn.KSK.private
```

Voor ZSK-bestanden dient KSK waar nodig door ZSK vervangen te worden. De bestanden kunnen nu worden opgenomen in het zonebestand, door de opdracht $include te gebruiken. Het zou er ongeveer als volgt uit moeten zien:

```
$include Kexample.com.+005+nnnnn.KSK.key ; KSK
$include Kexample.com.+005+nnnnn.ZSK.key ; ZSK
```

Onderteken tenslotte de zone en vertel BIND om het ondertekende zonebestand te gebruiken. Voor het ondertekenen van een zone wordt dnssec-signzone gebruikt. Het commando om de zone example.com, dat zich in example.com.db bevindt, zou er ongeveer zo uit moeten zien:

```
% dnssec-signzone -o example.com -k Kexample.com.+005+nnnnn.KSK example.com.db
Kexample.com.+005+nnnnn.ZSK.key
```

De sleutel die aan het argument -k wordt meegegeven is de KSK en het andere sleutelbestand is de ZSK dat bij het ondertekenen gebruikt moet worden. Het is mogelijk om meer dan één KSK en ZSK op te geven, wat tot gevolg heeft dat de zone met alle meegegeven sleutels wordt ondertekend. Dit kan nodig zijn om zonegegevens aan te leveren die met meerdere algoritmes zijn ondertekend. De uitvoer van dnssec-signzone is een zonebestand met daarin alle RRs ondertekend. Deze uitvoer komt in een bestand met de extensie .signed terecht, zoals example.com.db.signed. De DS-records worden ook naar een apart bestand dsset-example.com geschreven. Om deze ondertekende zone te gebruiken hoeft alleen de zone-directief in named.conf veranderd te worden om example.com.db.signed. Standaard zijn de ondertekeningen slechts 30 dagen geldig, wat betekent dat de zone over ongeveer 15 dagen hertekend moet worden om er zeker van te zijn dat resolvers geen records met oude ondertekeningen cachen. Het is mogelijk om hiervoor een script en een crontaak te maken. Bekijk de relevante handleidingen voor details.

Zorg ervoor dat de private sleutels veilig blijven, zoals met alle cryptografische sleutels. Bij het veranderen van een sleutel kan het beste de nieuwe sleutel in de zone opgenomen worden, en nog met de oude sleutel te ondertekenen, en om daarna over te stappen op de nieuwe sleutel. Nadat deze handelingen zijn voltooid kan de oude sleutel uit de zone worden verwijderd. Wanneer dit niet wordt gedaan kunnen de DNS-gegevens tijdelijk onbeschikbaar zijn totdat de nieuwe sleutel door de DNS-hiërarchie is gepropageerd. Meer informatie over sleutelwisselingen en andere praktijken rondom DNSSEC staan in RFC 4641: DNSSEC Operational practices.

Automatisering met BIND 9.7 of nieuwer

In versie 9.7 van BIND is een nieuwe mogelijkheid genaamd *Smart Signing* geïntroduceerd. Deze mogelijkheid heeft als doel om het sleutelbeheer en ondertekenproces eenvoudiger te maken door delen van deze taken te automatiseren. Door de sleutels in een *sleutelreservoir* te stoppen en de nieuwe optie auto-dnssec te gebruiken, is het mogelijk om een dynamische zone aan te maken welke opnieuw getekend wordt indien dat nodig is. Gebruik om deze zone bij te werken nsupdate met de nieuwe -l. rndc kan nu ook zones ondertekenen met sleutels uit het sleutelreservoir door de

optie `sign` te gebruiken. Voeg, om BIND dit automatische ondertekenen en bijwerken van zones te laten gebruiken voor `example.com`, het volgende aan named.conf toe:

```
zone example.com {
    type master;
    key-directory "/etc/named/keys";
    update-policy local;
    auto-dnssec maintain;
    file "/etc/named/dynamic/example.com.zone";
};
```

Nadat deze veranderingen gemaakt zijn, dienen de sleutels voor de zone aangemaakt te worden zoals uitgelegd in Configuratie van een autoratieve DNS-server, deze sleutels in het sleutelreservoir gestopt te worden dat als argument aan de `key-directory` in het zoneconfiguratie is meegegeven, waarna de zone automatisch zal worden ondertekend. Zones die op deze manier zijn geconfigureerd dienen met nsupdate te worden gedaan, dat voor het opnieuw ondertekenen van de zone met de nieuw toegevoegde gegevens zal zorgen. Zie voor meer details Verder lezen en de BIND-documentatie.

27.6.9. Beveiliging

Hoewel BIND de meest gebruikte implementatie van DNS is, is er altijd nog het beveiligingsvraagstuk. Soms worden er mogelijke en te misbruiken beveiligingsgaten gevonden.

Hoewel FreeBSD named automatisch in een chroot(8)-omgeving plaatst; zijn er verschillende andere beveiligingsmechanismen actief die zouden kunnen helpen om mogelijke aanvallen op de DNS-dienst af te wenden.

Het is altijd verstandig om de CERT beveiligingswaarschuwingen te lezen en een abonnement te nemen op de FreeBSD beveiligingswaarschuwingen mailinglijst om bij te blijven met de beveiligingsproblemen wat betreft Internet en FreeBSD.



Als er problemen ontstaan, kan het bijwerken van broncode en het opnieuw bouwen van named hulp bieden.

27.6.10. Verder lezen

BIND/named hulppagina's: rndc(8) named(8) named.conf(8) nsupdate(8) dnssec-signzone(8) dnssec-keygen(8)

- Officiële ISC BIND pagina
- Officieel ISC BIND Forum
- O'Reilly DNS en BIND 5e Editie
- Root DNSSEC
- DNSSEC Trust Anchor Publication for the Root Zone
- RFC1034 - Domain Names - Concepts and Facilitities

- RFC1035 - Domain Names - Implementation and Specification
- RFC4033 - DNS Security Introduction and Requirements
- RFC4034 - Resource Records for the DNS Security Extensions
- RFC4035 - Protocol Modifications for the DNS Security Extensions
- RFC4641 - DNSSEC Operational Practices
- RFC5011 - Automated Updates of DNS Security (DNSSEC Trust Anchors)

27.7. Apache HTTP server

27.7.1. Overzicht

FreeBSD wordt gebruikt om een paar van de drukste websites ter wereld te draaien. De meeste webservers op Internet maken gebruik van de Apache HTTP Server. Apache softwarepakketten staan op de FreeBSD installatiemedia. Als Apache niet bij de oorspronkelijke installatie van FreeBSD is meegeïnstalleerd, dan kan dat vanuit de port www/apache22.

Als Apache succesvol is geïnstalleerd, moeten er instellingen gemaakt worden.



In dit onderdeel wordt versie 2.2.X van de Apache HTTP Server behandeld omdat die het meest gebruikt wordt op FreeBSD. Meer gedetailleerde informatie over Apache 2.X dat buiten het bereik van dit document valt is te vinden op http://httpd.apache.org/.

27.7.2. Instellen

Het belangrijkste bestand met instellingen voor de Apache HTTP Server op FreeBSD is /usr/local/etc/apache22/httpd.conf. Dit bestand is een typisch UNIX® tekstgebaseerd instellingenbestand waarin regels met commentaar beginnen met het karakter `#`. Het uitputtend beschrijven van alle mogelijke instellingen valt buiten het bereik van dit boek, dus worden alleen de meest gebruikte directieven beschreven.

`ServerRoot "/usr/local"`

Hierin wordt de standaard mappenhiërarchie voor de Apache installatie aangegeven. Binaire bestanden staan in de submappen bin en sbin van de serverroot en bestanden met instellingen staan in etc/apache.

`ServerAdmin beheerder@beheer.adres`

Het adres waaraan problemen met de server gemaild kunnen worden. Dit adres verschijnt op een aantal door de server gegenereerde pagina's, zoals documenten met foutmeldingen.

`ServerName www.example.com`

Met `ServerName` kan een hostnaam ingesteld worden die wordt teruggezonden aan de cliënten als de naam van de server anders is dan diegene is ingesteld (gebruik bijvoorbeeld `www` in plaats van de echte hostnaam).

`DocumentRoot "/usr/local/www/apache22/data"`

`DocumentRoot`: de map waaruit de documenten worden geserveerd. Standaard worden alle verzoeken uit deze map gehaald, maar er kunnen symbolische links en aliassen gebruikt worden om naar andere locaties te wijzen.

Het is altijd een goed idee om reservekopieën te maken van het instellingenbestand voor Apache vóór het maken van wijzigingen. Als de juiste instellingen gemaakt zijn, kan Apache gestart worden.

27.7.3. Apache draaien

De port www/apache2 installeert een rc(8)-script dat helpt met het starten, stoppen en herstarten van Apache en is te vinden in /usr/local/etc/rc.d/.

Om Apache met het systeem mee te starten kan de volgende regel aan /etc/rc.conf worden toegevoegd:

```
apache22_enable="YES"
```

Als het nodig is Apache met afwijkende opties op te starten, kan de volgende regel aan /etc/rc.conf worden toegevoegd:

```
apache22_flags=""
```

De configuratie van Apache kan worden getest op fouten voordat het daemon `httpd` voor de eerste keer wordt gestart, of na het maken van wijzigingen aan de instellingen terwijl `httpd` draait. Dit kan direct door het rc(8)-script worden gedaan, of door het gereedschap service(8) door één van de volgende commando's op te geven:

```
# service apache22 configtest
```



Het is belangrijk om op te merken dat `configtest` geen rc(8)-standaard is, verwacht niet dat het met alle rc(8)-opstartscripts werkt.

Als Apache geen instellingsfouten meldt, kan Apache `httpd` gestart worden met service(8):

```
# service apache22 start
```

De dienst `httpd` kan getest worden door `http://localhost` in een webbrowser te typen, waarbij *localhost* door de volledig gekwalificeerde domeinnaam wordt vervangen van de machine die `httpd` draait, als het niet de lokale machine is. De standaard webpagina die afgebeeld wordt is /usr/local/www/apache22/data/index.html.

27.7.4. Virtuele hosting

Apache ondersteunt twee verschillende manieren van Virtuele Hosting. De eerste methode is Naamgebaseerde Virtuele Hosting. Naamgebaseerde Virtuele Hosting gebruikt de HTTP/1.1 headers van de cliënten om de hostnaam uit te zoeken. Hierdoor kunnen meerdere domeinen hetzelfde IP-adres delen.

Om Apache gebruik te laten maken van Naamgebaseerde Virtuele Hosting kan een regel als de volgende in httpd.conf worden opgenomen:

```
NameVirtualHost *
```

Als een webserver `www.domein.tld` heet en er moet een virtueel domein voor `www.anderdomein.tld` gaan draaien, dan kunnen de volgende regels aan httpd.conf worden toegevoegd:

```
<VirtualHost *>
    ServerName www.domein.tld
    DocumentRoot /www/domein.tld
</VirtualHost>

<VirtualHost *>
    ServerName www.anderdomein.tld
    DocumentRoot /www/anderdomein.tld
</VirtualHost>
```

De adressen en de paden uit dit voorbeeld kunnen in echte implementaties uiteraard gewijzigd worden.

Meer informatie over het opzetten van virtuele hosts staat in de officiële documentatie voor Apache op http://httpd.apache.org/docs/vhosts/

27.7.5. Apache modules

Er zijn veel verschillende Apache modules die functionaliteit toevoegen aan de basisdienst. De FreeBSD Portscollectie biedt op een eenvoudige manier de mogelijkheid om Apache samen met de meeste populaire add-on modules te installeren.

mod_ssl

De module mod_ssl gebruikt de bibliotheek OpenSSL om sterke cryptografie te leveren via de protocollen Secure Sockets Layer (SSL v2/v3) en Transport Layer Security (TLS v1). Deze module levert alles wat nodig is om een getekend certificaat aan te vragen bij een vertrouwde certificaatautoriteit om een veilige webserver onder FreeBSD te kunnen draaien.

De module mod_ssl wordt standaard gebouwd, maar kan worden aangezet door tijdens het compileren `-DWITH_SSL` op te geven.

Taalbindingen

Er zijn Apache-modules beschikbare voor de meeste grote scriptingtalen. Deze modules maken het typisch mogelijk om Apache-modules geheel in een scriptingtaal te schrijven. Ze worden ook vaak gebruikt als een persistente interpreter die in de server zit en die de rompslomp van het starten van een externe interpreter en de opstartvertraging voor dynamische websites vermijdt, zoals beschreven in de volgende sectie.

27.7.6. Dynamische websites

In het afgelopen decennium hebben steeds meer bedrijven zich op Internet gericht om hun omzet te verhogen en hun zichtbaarheid te vergroten. Hiermee is ook de behoefte aan interactieve webinhoud toegenomen. Hoewel sommige bedrijven zoals Microsoft® oplossingen hebben geïntroduceerd voor hun eigen (propriëtaire) producten, heeft ook de open source gemeenschap een antwoord op de vraag gegeven. Moderne opties voor dynamische webinhoud zijn onder andere Django, Ruby on Rails, mod_perl2, en mod_php.

Django

Django is een BSD-gelicenseerd raamwerk ontworpen om ontwikkelaars in staat te stellen om snel hoog presterende, elegante webapplicaties te schrijven. Het biedt een vertaling van objecten naar relaties zodat datatypes ontwikkeld kunnen worden als Python-objecten, en er een rijke dynamische databasetoegang voor die objecten kan worden geboden zonder dat de ontwikkelaar ooit SQL hoeft te schrijven. Het biedt ook een uitbreidbaar sjabloonsysteem zodat de applicatielogica is gescheiden van de HTML-presentatie.

Django is afhankelijk van mod_python, Apache, en een SQL-database-engine naar keuze. De FreeBSD-port zal al deze vereisten met de juiste vlaggen voor u installeren.

Voorbeeld 23. Django installeren met Apache2, mod_python3 en PostgreSQL

```
# cd /usr/ports/www/py-django; make all install clean -DWITH_MOD_PYTHON3
-DWITH_POSTGRESQL
```

Als Django en deze vereisten eenmaal zijn geïnstalleerd, dient u een Django-projectmap te maken en vervolgens Apache te configureren om de ingebakken Python-interpreter te gebruiken om uw applicatie voor specifieke URL's op uw site aan te roepen.

Voorbeeld 24. Apache-configuratie voor Django/mod_python

U moet een regel aan het Apache-bestand httpd.conf toevoegen om Apache in te stellen om verzoeken voor bepaalde URL's aan uw webapplicatie door te geven:

```
<Location "/">
    SetHandler python-program
    PythonPath "['/map/naar/uw/django-pakketten/'] + sys.path"
    PythonHandler django.core.handlers.modpython
```

```
    SetEnv DJANGO_SETTINGS_MODULE mijnsite.settings
    PythonAutoReload On
    PythonDebug On
</Location>
```

Ruby on Rails

Ruby on Rails is een ader opensource webraamwerk dat een volledige ontwikkelstack biedt en geoptimaliseerd is om webontwikkelaars productiever te maken en snel krachtige applicaties te laten ontwikkelen. Het kan eenvoudig vanuit het portssysteem geïnstalleerd worden.

```
# cd /usr/ports/www/rubygem-rails; make all install clean
```

mod_perl2

Het Apache/Perl integratieproject brengt de volledige kracht van de programmeertaal Perl en de Apache HTTP Server samen. Met de module mod_perl2 is het mogelijk om Apache-modules volledig in Perl te schrijven. Daarnaast voorkomt een ingebouwde persistente interpreter in de server de rompslomp van het starten van een externe interpreter en de nadelen van de opstarttijd van Perl.

mod_perl2 is beschikbaar in de port www/mod_perl2.

mod_php

PHP, ook bekend als "PHP: Hypertext Preprocessor", is een algemene scripttaal die bijzonder geschikt is voor webontwikkeling. Het is mogelijk de taal in te bedden in HTML en de syntaxis is afgeleid van C, Java™ en Perl met de bedoeling webontwikkelaars in staat te stellen om snel dynamisch samengestelde pagina's te schrijven.

Om ondersteuning voor PHP5 toe te voegen aan de Apache webserver kan eerst de port lang/php5 geïnstalleerd worden.

Als de port lang/php5 voor het eerst geïnstalleerd wordt, worden automatisch de beschikbare `OPTIONS` weergegeven. Als er geen menu wordt weergegeven, omdat de port lang/php5 reeds in het verleden is geïnstalleerd, is het altijd mogelijk om het optiedialoog weer te laten verschijnen door

```
# make config
```

uit te voeren in de map van de port.

Controleer in het optiedialoog dat de optie `APACHE` mod_php5 als een laadbare module voor de webserver Apache bouwt.



Een heleboel sites draaien nog steeds PHP4 om verschillende redenen (compatibiliteitszaken of reeds in gebruik genomen webapplicaties). Als mod_php4 nodig is in plaats van mod_php5, gebruik dan de port lang/php4. De port lang/php4

> ondersteunt een groot deel van de configuratie- en bouwopties van de port lang/php5.

Hiermee worden de modules die nodig zijn voor de ondersteuning van dynamische PHP-applicaties geïnstalleerd en ingesteld. Controleer dat de volgende secties aan /usr/local/etc/apache22/httpd.conf zijn toegevoegd:

```
LoadModule php5_module        libexec/apache/libphp5.so
```

```
AddModule mod_php5.c
    IfModule mod_php5.c
        DirectoryIndex index.php index.html
    /IfModule
    IfModule mod_php5.c
        AddType application/x-httpd-php .php
        AddType application/x-httpd-php-source .phps
    /IfModule
```

Na voltooiing is een eenvoudige aanroep van het commando `apachectl` voor een nette herstart nodig om de module PHP te laden:

```
# apachectl graceful
```

Voor toekomstig bijwerken van PHP zal het commando `make config` niet nodig zijn; de geselecteerde `OPTIONS` worden automatisch bewaard door het FreeBSD Ports raamwerk.

De ondersteuning voor PHP in FreeBSD is extreem modulair waardoor de basisinstallatie zeer beperkt is. Het is heel gemakkelijk om ondersteuning toe te voegen door de port lang/php5-extensions te gebruiken. Deze port biedt een menugestuurde interface voor de installatie van PHP-uitbreidingen. Als alternatief kunnen individuele uitbreidingen worden geïnstalleerd door de juiste port te gebruiken.

Om bijvoorbeeld ondersteuning voor de MySQL databaseserver aan PHP5 toe te voegen kan gewoonweg de port databases/php5-mysql geïnstalleerd worden:

Na de installatie van een uitbreiding moet de Apache-server herladen worden om de nieuwe veranderingen in de configuratie op te pikken:

```
# apachectl graceful
```

27.8. File Transfer Protocol (FTP)

27.8.1. Overzicht

Het File Transfer Protocol (FTP) biedt gebruikers een eenvoudige manier om bestanden van en naar een FTP server te verplaatsen. FreeBSD bevat FTP server software, ftpd, in het basissysteem. Hierdoor is het opzetten en beheren van een FTP server op FreeBSD erg overzichtelijk.

27.8.2. Instellen

De belangrijkste stap bij het instellen is de beslissing welke accounts toegang krijgen tot de FTP server. Een normaal FreeBSD systeem heeft een aantal systeemaccounts die gebruikt worden voor daemons, maar onbekende gebruikers mag niet toegestaan worden van die accounts gebruikt te maken. In /etc/ftpusers staat een lijst met gebruikers die geen FTP toegang hebben. Standaard staan daar de voorgenoemde accounts in, maar het is ook mogelijk om daar gebruikers toe te voegen die geen FTP toegang mogen hebben.

Het kan ook wenselijk zijn de FTP toegang voor sommige gebruikers te beperken, maar niet onmogelijk te maken. Dit kan met /etc/ftpchroot. In dat bestand staan gebruikers en groepen waarop FTP toegangsbeperkingen van toepassing zijn. In ftpchroot(5) staan alle details die hier niet beschreven zijn.

Om anonieme FTP toegang voor een server in te schakelen, dient er een gebruiker `ftp` op een FreeBSD systeem aangemaakt te worden. Dan kunnen gebruikers op de server aanmelden met de gebruikersnaam `ftp` of `anonymous` en met ieder wachtwoord (de geldende conventie schrijft voor dat dit een emailadres van de gebruiker is). De FTP server roep bij een anonieme aanmelding chroot(2) aan, zodat er alleen toegang is tot de thuismap van de gebruiker `ftp`.

Er zijn twee tekstbestanden waarin welkomstberichten voor de FTP-cliënten gezet kunnen worden. De inhoud van /etc/ftpwelcome wordt getoond voordat gebruikers een aanmeldprompt zien. Na een succesvolle aanmelding wordt de inhoud van /etc/ftpmotd getoond. Het genoemde pad is relatief ten opzichte van de aanmeldomgeving, dus voor anonieme gebruikers wordt ~ftp/etc/ftpmotd getoond.

Als een FTP server eenmaal correct is ingesteld, moet die ingeschakeld worden in /etc/inetd.conf. Daar moet het commentaarkarakter `#` voor de bestaande ftpd regel verwijderd worden:

```
ftp stream  tcp nowait  root    /usr/libexec/ftpd   ftpd -l
```

Zoals is uitgelegd in Het instellingenbestand van inetd herladen, moet de configuratie van inetd worden herladen nadat dit instellingenbestand is gewijzigd. Details over het aanzetten van inetd op uw systeem staan in Instellingen.

Als alternatief kan ftpd ook gestart worden als een op zichzelf staande dienst. In dat geval volstaat het om de juiste variabele in te stellen in /etc/rc.conf:

```
ftpd_enable="YES"
```

Na het instellen van de bovenstaande variabele zal de op zichzelf staande server gestart worden

nadat de computer opnieuw is opgestart, of het kan handmatig worden gestart door het volgende commando als `root` uit te voeren:

```
# service ftpd start
```

Nu kan aangemeld worden op de FTP-server met:

```
% ftp localhost
```

27.8.3. Beheren

De ftpd daemon gebruikt syslog(3) om berichten te loggen. Standaard plaatst de systeemlogdaemon berichten over FTP in /var/log/xferlog. De lokatie van het FTP logboek kan gewijzigd worden door de volgende regels in /etc/syslog.conf te wijzigen:

```
ftp.info      /var/log/xferlog
```

Het is verstandig na te denken over de gevaren die op de loer liggen bij het draaien van een anonieme FTP server. Dat geldt in het bijzonder voor het laten uploaden ven bestanden. Het is dan goed mogelijk dat een FTP site een forum wordt om commerciële software zonder licenties uit te wisselen of erger. Als anonieme uploads toch nodig zijn, dan horen de rechten op die bestanden zo te staan dat ze niet door andere anonieme gebruikers gelezen kunnen worden tot er door een beheerder naar gekeken is.

27.9. Bestands- en printdiensten voor Microsoft® Windows® cliënten (Samba)

27.9.1. Overzicht

Samba is een populair open source softwarepakket dat bestands- en printdiensten voor Microsoft® Windows® cliënten biedt. Die cliënten kunnen dan ruimte op een FreeBSD bestandssysteem gebruiken alsof het een lokale schijf is en FreeBSD printers gebruiken alsof het lokale printers zijn.

Samba softwarepakketten horen op de FreeBSD installatiemedia te staan. Als Samba bij de basisinstallatie niet mee is geïnstalleerd, dan kan dat alsnog via de net/samba34 port of met het pakket.

27.9.2. Instellen

Een standaardbestand met instellingen voor Samba wordt geïnstalleerd als /usr/local/shared/examples/samba34/smb.conf.default. Dit bestand dient gekopieerd te worden naar /usr/local/etc/smb.conf en voordat Samba gebruikt kan worden, moeten er aanpassingen aan worden gemaakt.

smb.conf bevat de instellingen voor Samba, zoals die voor de printers en de "gedeelde

bestandssystemen" die gedeeld worden met Windows® cliënten. Het pakket Samba bevat een webgebaseerde beheermodule die swat heet, waarmee smb.conf op een eenvoudige manier ingesteld kan worden.

De Samba webbeheermodule gebruiken (SWAT)

De Samba Webbeheermodule (SWAT) draait als een daemon vanuit inetd. Daarom dient inetd aangezet te worden zoals beschreven in De inetd"Super-Server" en dient voor de volgende regel uit /etc/inetd.conf het commentaarkarakter verwijderd te worden voordat swat gebruikt kan worden om Samba in te stellen:

```
swat   stream  tcp     nowait/400      root    /usr/local/sbin/swat    swat
```

Zoals is uitgelegd in Het instellingenbestand van inetd herladen, moet de configuratie van inetd worden herladen nadat dit instellingenbestand is gewijzigd.

Als swat is ingeschakeld in inetd.conf, kan de module gebruikt worden door met een browser een verbinding te maken met http://localhost:901. Er dient aangemeld te worden met het `root` account van het systeem.

Na succesvol aanmelden op de hoofdpagina voor de Samba instellingen, is het mogelijk de systeemdocumentatie te bekijken of te starten door op het tabblad **Globals** te klikken. Het onderdeel **Globals** correspondeert met de sectie `[global]` in /usr/local/etc/smb.conf.

Systeembrede instellingen

Of Samba nu wordt ingesteld door /usr/local/etc/smb.conf direct te bewerken of met swat, de eerste instellingen die gemaakt moeten worden zijn de volgende:

`workgroup`

: NT Domeinnaam of Werkgroepnaam voor de computers die verbinding gaan maken met de server.

`netbiosnaam`

: Hiermee wordt de NetBIOS naam waaronder de Samba server bekend zal zijn ingesteld. Standaard is de naam het eerste gedeelte van de DNS-naam van een host.

`server string`

: Hiermee wordt de string ingesteld die te zien is als het commando `net view` en een aantal andere commando's die gebruik maken van de beschrijvende tekst voor de server gebruikt worden.

Beveiligingsinstellingen

Twee van de belangrijkste instellingen in /usr/local/etc/smb.conf zijn het gekozen beveiligingsmodel en het wachtwoord voor cliëntgebruikers. Deze worden met de volgende instellingen gemaakt:

`security`

: De twee meest gebruikte mogelijkheden hier zijn `security = share` en `security = user`. Als de

cliënten gebruikersnamen hebben die overeenkomen met hun gebruikersnaam op de FreeBSD machine, dan is het verstandig om te kiezen voor beveiliging op gebruikersniveau. Dit is het standaard beveiligingsbeleid en kent als voorwaarde dat gebruikers zich eerst moeten aanmelden voordat ze toegang krijgen tot gedeelde bronnen.

Bij beveiliging op shareniveau hoeft een cliënt niet met een geldige gebruikersnaam en wachtwoord aan te melden op de server voor het mogelijk is om een verbinding te proberen te krijgen met een gedeelde bron. Dit was het standaardbeveiligingsmodel voor oudere versies van Samba.

`passdb backend`

Samba kent aan de achterkant verschillende authenticatiemodellen. Cliënten kunnen authenticeren met LDAP, NIS+, een SQL-database of een aangepast wachtwoordbestand. De standaard authenticatiemethode is `smbpasswd`. Meer wordt hier niet behandeld.

Als aangenomen wordt dat de standaard achterkant `smbpasswd` wordt gebruikt, dan moet /usr/local/etc/samba/smbpasswd gemaakt worden om Samba in staat te stellen cliënten te authenticeren. Als het gewenst is om uw UNIX® gebruikersaccounts toegang te geven vanaf Windows® cliënten, gebruik dan het volgende commando:

```
# smbpasswd -a gebruikersnaam
```



De aanbevolen backend is nu `tdbsam`, en het volgende command moet gebruikt worden om gebruikersaccounts toe te voegen:

```
# pdbedit -a -u gebruikersnaam
```

In de Official Samba HOWTO staat meer informatie over instelopties. Met de hier gegeven basisuitleg moet het mogelijk zijn Samba draaiende te krijgen.

27.9.3. Samba starten

De port net/samba34 voegt een nieuw opstartscript toe, dat gebruikt kan worden om Samba te beheren. Om dit script te activeren, zodat het bijvoorbeeld gebruikt kan worden om Samba te starten, stoppen, of te herstarten, dient de volgende regel aan /etc/rc.conf toegevoegd te worden:

```
samba_enable="YES"
```

Of, voor fijnkorrelig beheer:

```
nmbd_enable="YES"
```

```
smbd_enable="YES"
```



Dit stelt Samba ook in om automatisch tijdens het opstarten te starten.

Vervolgens is het mogelijk om Samba op elk moment te starten door dit te typen:

```
# service samba start
Starting SAMBA: removing stale tdbs :
Starting nmbd.
Starting smbd.
```

Refereer aan Gebruik van rc met FreeBSD voor meer informatie over het gebruikt van rc-scripts.

Samba bestaat feitelijk uit drie afzonderlijke daemons. Het script samba start de daemons nmbd en smbd. Als de winbind naamresolutiediensten in smb.conf zijn ingeschakeld, dan start ook de daemon winbindd.

Samba kan op ieder moment gestopt worden met:

```
# service samba stop
```

Samba is een complexe softwaresuite met functionaliteit waarmee verregaande integratie met Microsoft® Windows® netwerken mogelijk wordt. Informatie die verder gaat dan de basisinstallatie staat op http://www.samba.org.

27.10. Tijd synchroniseren met NTP

27.10.1. Overzicht

Na verloop van tijd gaat de tijd van een computer meestal uit de pas lopen. Het Netwerk Tijd Protocol (NTP) kan ervoor zorgen dat de tijd accuraat blijft.

Veel diensten op Internet zijn afhankelijk, of hebben veel voordeel, van het betrouwbaar zijn van de tijd. Zo ontvangt een webserver bijvoorbeeld veel verzoeken om een bestand te sturen als dat gewijzigd is sinds een bepaald moment. In een LAN-omgeving is het van groot belang dat computers die bestanden delen van eenzelfde server gesynchroniseerde tijd hebben zodat de tijdstempels consistent blijven. Diensten zoals cron(8) zijn ook afhankelijk van een betrouwbare systeemtijd om commando's op het ingestelde moment uit te voeren.

Bij FreeBSD zit de ntpd(8) NTP server die gebruikt kan worden om bij andere NTP servers de tijd op te vragen om de eigen klok gelijk te zetten of om de juiste tijd te verstrekken aan andere apparaten.

27.10.2. Passende NTP-servers kiezen

Om de tijd te synchroniseren moeten er één of meer NTP-servers beschikbaar zijn. Een lokale systeembeheerder of een ISP heeft wellicht een NTP-server voor dit doel opgezet. Het is verstandig om documentatie te raadplegen en te bekijken of dat het geval is. Er is een online lijst van publiek toegankelijke NTP-servers waarop een NTP-server gezocht kan worden die in geografische zin

dichtbij een te synchroniseren computer ligt. Het is belangrijk te voldoen aan het beleid voor de betreffende server en toestemming te vragen als dat in de voorwaarden staat.

Het is verstandig meerdere, niet van elkaar afhankelijke, NTP-servers te kiezen voor het geval een van de servers niet langer betrouwbaar is of niet bereikbaar is. ntpd(8) gebruikt de antwoorden die van andere servers ontvangen worden op intelligente wijze: betrouwbare servers krijgen voorrang boven onbetrouwbare servers.

27.10.3. Machine instellen

Basisinstellingen

Als het alleen de bedoeling is de tijd te synchroniseren bij het opstarten van een machine, dan kan ntpdate(8) gebruikt worden. Dit kan van toepassing zijn op desktops die regelmatig herstart worden en niet echt regelmatig gesynchroniseerd hoeven te worden. Op sommige machines hoort echter ntpd(8) te draaien.

Het gebruik van ntpdate(8) bij het opstarten is ook een goed idee voor machines waarop ntpd(8) draait. De ntpd(8) wijzigt de tijd geleidelijk, terwijl ntpdate(8) gewoon de tijd instelt, hoe groot het verschil tussen de bestaande tijd van een machine en de correcte tijd ook is.

Om ntpdate(8) tijdens het opstarten in te schakelen kan `ntpdate_enable="YES"` aan /etc/rc.conf worden toegevoegd. Alle voor de synchronisatie te gebruiken servers moeten dan, samen met eventuele opties voor ntpdate(8), in `ntpdate_flags` aangegeven worden.

Algemene instellingen

NTP wordt ingesteld met het bestand /etc/ntp.conf in het formaat dat beschreven staat in ntp.conf(5). Hieronder volgt een eenvoudig voorbeeld:

```
server ntplocal.example.com prefer
server timeserver.example.org
server ntp2a.example.net

driftfile /var/db/ntp.drift
```

De optie `server` geeft aan welke servers er gebruikt moeten worden, met op elke regel een server. Als de server wordt ingesteld met het argument `prefer`, zoals bij `ntplocal.example.com`, dan krijgt die server de voorkeur boven de andere. Een antwoord van een voorkeursserver wordt genegeerd als dat significant afwijkt van de antwoorden van de andere servers. In andere gevallen wordt het gebruikt zonder rekening te houden met de andere antwoorden. Het argument `prefer` wordt meestal gebruikt voor NTP-servers waarvan bekend is dat ze erg betrouwbaar zijn, zoals die met speciale tijdbewakingshardware.

De optie `driftfile` geeft aan welk bestand gebruikt wordt om de offset van de klokfrequentie van het systeem op te slaan. ntpd(8) gebruikt die om automatisch te compenseren voor het natuurlijke afwijken van de tijd, zodat er zelfs bij gebrek aan externe bronnen een redelijke accurate tijdsinstelling mogelijk is.

De optie `driftfile` geeft aan welk bestand gebruikt wordt om informatie over eerdere antwoorden van NTP-servers die gebruikt worden op te slaan. Dit bestand bevat interne informatie voor NTP. Het hoort niet door andere processen gewijzigd te worden.

Toegang tot een server instellen

Een NTP-server is standaard toegankelijk voor alle hosts op een netwerk. De optie `restrict` in /etc/ntp.conf maakt het mogelijk om aan te geven welke machines de dienst mogen benaderen.

Voor het blokkeren van toegang voor alle andere machines kan de volgende regel aan /etc/ntp.conf toegevoegd worden:

```
restrict default ignore
```



Dit zal ook toegang van uw server naar alle servers die vermeld staan in uw lokale configuratie verhinderen. Als u uw NTP-server moet synchroniseren met een externe NTP-server, dient u deze specifieke server toe te staan. Lees de handleiding voor ntp.conf(5) voor meer informatie.

Om alleen machines op bijvoorbeeld het lokale netwerk toe te staan hun tijd te synchroniseren met een server, maar ze tegelijkertijd niet toe te staan om de server te draaien of de server als referentie voor synchronisatie te gebruiken, kan de volgende regel toegevoegd worden:

```
restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap
```

Hierboven is `192.168.1.0` een IP-adres op een LAN en `255.255.255.0` is het bijbehorende netwerkmasker.

/etc/ntp.conf mag meerdere regels met `restrict` bevatten. Meer details staan in het onderdeel `Access Control Support` van ntp.conf(5).

27.10.4. De NTP-server draaien

De NTP-server kan bij het opstarten gestart worden door de regel `ntpd_enable="YES"` aan /etc/rc.conf toe te voegen. Om extra opties aan ntpd(8) mee te geven kan de parameter `ntpd_flags` in /etc/rc.conf gebruikt worden.

Om de server zonder een herstart van de machine te starten kan `ntpd` uitgevoerd worden, met toevoeging van de parameters uit `ntpd_flags` in /etc/rc.conf. Bijvoorbeeld:

```
# ntpd -p /var/run/ntpd.pid
```

27.10.5. ntpd gebruiken met een tijdelijke Internetverbinding

ntpd(8) heeft geen permanente verbinding met een netwerk nodig om goed te werken. Maar als er gebruik gemaakt wordt van een inbelverbinding, is het wellicht verstandig om ervoor te zorgen dat

uitgaande NTP-verzoeken geen uitgaande verbinding kunnen starten. Als er gebruik gemaakt wordt van gebruikers-PPP, kunnen er `filter` commando's ingesteld worden in /etc/ppp/ppp.conf. Bijvoorbeeld:

```
set filter dial 0 deny udp src eq 123
# NTP-verkeer zorgt niet voor uitbellen
set filter dial 1 permit 0 0
set filter alive 0 deny udp src eq 123
# Inkomend NTP-verkeer houdt de verbinding niet open
set filter alive 1 deny udp dst eq 123
# Uitgaand NTP-verkeer houdt de verbinding niet open
set filter alive 2 permit 0/0 0/0
```

Meer details staan in de sectie `PACKET FILTERING` in ppp(8) en in de voorbeelden in /usr/shared/examples/ppp/.



Sommige Internetproviders blokkeren lage poorten, waardoor NTP niet kan werken omdat er nooit een antwoord ontvangen kan worden door een machine.

27.10.6. Meer informatie

HTML-documentatie voor de NTP-server staat in /usr/shared/doc/ntp/.

27.11. Hosts op afstand loggen met `syslogd`

Het omgaan met systeemlogs is een cruciaal aspect van zowel beveiligings- als systeembeheer. Het in de gaten houden van logbestanden van meerdere hosts kan nogal onhandelbaar worden als deze hosts over (middel)grote netwerken zijn verspreid, of wanneer ze deel zijn van verschillende soorten netwerken. In deze gevallen kan het op afstand loggen het gehele proces een stuk aangenamer maken.

Het centraal loggen naar een specifieke loghost kan wat van de administratieve last van het beheren van logbestanden wegnemen. Het aggregeren, samenvoegen, en roteren van logbestanden kan op één enkele plaats worden ingesteld, door gebruik te maken van de eigen gereedschappen van FreeBSD, zoals syslogd(8) en newsyslog(8). In de volgende voorbeeldconfiguratie zal host `A`, genaamd `logserv.example.com`, loginformatie voor het plaatselijke netwerk verzamelen. Host `B`, genaamd `logclient.example.com`, zal loginformatie aan het serversysteem doorgeven. In echte configuraties hebben beide hosts degelijke voor- en terugwaartse DNS of regels in /etc/hosts nodig. Anders worden de gegevens geweigerd door de server.

27.11.1. Configuratie van de logserver

Logservers zijn machines die zijn geconfigureerd om loginformatie van hosts op afstand te accepteren. In de meeste gevallen is dit om de configuratie te vergemakkelijken, in andere gevallen kan het gewoon een beheersbeslissing zijn. Ongeacht de reden zijn er enkele eisen voordat er verder wordt gegaan.

Een juist geconfigureerde logserver voldoet aan de volgende minimale eisen:

- De regels van de firewall staan toe dat UDP wordt doorgegeven op poort 514 van zowel de cliënt als de server;
- syslogd is ingesteld om berichten op afstand van cliëntmachines te accepteren;
- De syslogd-server en alle cliëntmachines moeten geldige regels hebben voor zowel voorwaartse als terugwaartse DNS, of correct zijn geconfigureerd in /etc/hosts.

Om de logserver te configureren, moet de cliënt vermeld zijn in /etc/syslog.conf, en moet de logfaciliteit zijn gespecificeerd:

```
+logclient.example.com
*.* /var/log/logclient.log
```



Meer informatie over de verschillende ondersteunde en beschikbare *faciliteiten* kan gevonden worden in de handleidingpagina syslog.conf(5).

Eenmaal toegevoegd worden alle `faciliteits`-berichten gelogd naar het eerder gespecificeerde bestand, /var/log/logclient.log.

De servermachine moet ook het volgende in /etc/rc.conf hebben staan:

```
syslogd_enable="YES"
syslogd_flags="-a logclient.example.com -v -v"
```

De eerste optie zet de daemon `syslogd` aan tijdens het opstarten, en de tweede regel staat toe dat gegevens van de cliënt op deze server worden geaccepteerd. Het laatste gedeelte, dat `-v -v` gebruikt, verhoogt de verbositeit van gelogde berichten. Dit is extreem handig voor het optimaal instellen van faciliteiten aangezien beheerders kunnen zien welk soort berichten onder welke faciliteit worden gelogd.

Er kunnen meerdere opties `-a` worden gespecificeerd om logging vanuit meerdere cliënten toe te staan. IP-adressen en hele netblokken mogen ook worden gespecificeerd, bekijk de hulppagina syslog(3) voor een volledige lijst van mogelijke opties.

Als laatste dient het logbestand gecreëerd te worden. De gebruikte manier maakt niet uit, maar touch(1) werkt prima in dit soort situaties:

```
# touch /var/log/logclient.log
```

Nu dient het `syslogd`-daemon herstart en geverifieerd worden:

```
# service syslogd restart
# pgrep syslog
```

Als er een PID wordt teruggegeven, dan is de server succesvol herstart, en kan er begonnen worden met de configuratie van de cliënt. Raadpleeg de log /var/log/messages voor uitvoer als de server niet is herstart.

27.11.2. Configuratie van de logcliënt

Een logcliënt is een machine die loginformatie naar een logserver verstuurt en daarnaast lokale kopieën bewaart.

Net als logservers moeten logcliënten ook aan enkele minimumeisen voldoen:

- syslogd(8) moet zijn ingesteld om berichten van bepaalde soorten naar een logserver te sturen, die ze moet accepteren;
- De firewall moet UDP-pakketten doorlaten op poort 514;
- Zowel voorwaartse als terugwaartse DNS moeten geconfigureerd zijn of juiste regels in /etc/hosts hebben.

De configuratie van cliënten is wat soepeler dan die van servers. De cliëntmachine moet de volgende regels in /etc/rc.conf hebben:

```
syslogd_enable="YES"
syslogd_flags="-s -v -v"
```

Net als eerder zullen deze regels de daemon `syslogd` tijdens het opstarten aanzetten, en de verbositeit van gelogde berichten verhogen. De optie `-s` voorkomt dat logs van deze cliënt vanuit andere hosts worden geaccepteerd.

Faciliteiten beschrijven het systeemgedeelte waarvoor een bericht is gegenereerd. ftp en ipfw bijvoorbeeld zijn beide faciliteiten. Wanneer er logberichten worden gegenereerd voor deze twee diensten, zullen ze normaalgesproken deze twee gereedschappen in elk logbericht opnemen. Faciliteiten worden vergezeld van een prioriteit of niveau, welke wordt gebruikt om aan te geven hoe belangrijk een logbericht is. De meest voorkomende zullen `warning` en `info` zijn. Bekijk de handleidingpagina syslog(3) voor een volledige lijst van beschikbare faciliteiten en prioriteiten.

De logserver moet in /etc/syslog.conf van de cliënt zijn gedefinieerd. In dit geval wordt het symbool `@` gebruikt om loggegevens naar een server op afstand te sturen en zou er ongeveer als de volgende regel uit moeten zien:

```
*.* @logserv.example.com
```

Eenmaal toegevoegd moet `syslogd` worden herstart zodat de veranderingen effect hebben:

```
# service syslogd restart
```

Om te testen of logberichten over het netwerk worden verzonden, wordt logger(1) op de cliënt gebruikt om een bericht naar `syslogd` te sturen:

```
# logger "Testbericht van logclient"
```

Dit bericht dient nu zowel in /var/log/messages op de cliënt als /var/log/logclient.log op de logserver te staan.

27.11.3. Logservers debuggen

In bepaalde gevallen kan het nodig zijn om te debuggen als berichten niet door de logserver worden ontvangen. Er zijn verschillende redenen waarom dit kan gebeuren; de twee meest voorkomende zijn echter voorvallen met de netwerkverbinding en DNS. Om deze gevallen te testen, dient te worden nagegaan dat beide hosts elkaar kunnen bereiken door de hostnaam in /etc/rc.conf te gebruiken. Als dit juist lijkt te werken, dient de optie `syslogd_flags` in /etc/rc.conf te worden veranderd.

In het volgende voorbeeld is /var/log/logclient.log leeg, en noemt /var/log/messages geen reden waarom het mislukt. Verander de optie `syslogd_flags` zoals in het volgende voorbeeld en herstart om de debuguitvoer te verhogen:

```
syslogd_flags="-d -a logclien.example.com -v -v"
```

```
# service syslogd restart
```

Debuggegevens zoals de volgende zullen meteen na de herstart over het scherm vliegen:

```
logmsg: pri 56, flags 4, from logserv.example.com, msg syslogd: restart
syslogd: restarted
logmsg: pri 6, flags 4, from logserv.example.com, msg syslogd: kernel boot file is
/boot/kernel/kernel
Logging to FILE /var/log/messages
syslogd: kernel boot file is /boot/kernel/kernel
cvthname(192.168.1.10)
validate: dgram from IP 192.168.1.10, port 514, name logclient.example.com;
rejected in rule 0 due to name mismatch.
```

Het is duidelijk dat de berichten worden geweigerd wegens een niet-overeenkomende naam. Na de configuratie grondig te hebben herzien, lijkt het of een typefout in de volgende regel in /etc/rc.conf een probleem heeft:

```
syslogd_flags="-d -a logclien.example.com -v -v"
```

De regel dient `logclient`, niet `logclien` te bevatten. Nadat de juiste wijzigingen zijn gemaakt, wordt er herstart met de verwachte resultaten:

```
# service syslogd restart
logmsg: pri 56, flags 4, from logserv.example.com, msg syslogd: restart
syslogd: restarted
logmsg: pri 6, flags 4, from logserv.example.com, msg syslogd: kernel boot file is
/boot/kernel/kernel
syslogd: kernel boot file is /boot/kernel/kernel
logmsg: pri 166, flags 17, from logserv.example.com,
msg Dec 10 20:55:02 <syslog.err> logserv.example.com syslogd: exiting on signal 2
cvthname(192.168.1.10)
validate: dgram from IP 192.168.1.10, port 514, name logclient.example.com;
accepted in rule 0.
logmsg: pri 15, flags 0, from logclient.example.com, msg Dec 11 02:01:28 trhodes: Test
message 2
Logging to FILE /var/log/logclient.log
Logging to FILE /var/log/messages
```

Nu worden de berichten juist ontvangen en in het correcte bestand geplaatst.

27.11.4. Beveiligingsoverwegingen

Zoals bij alle netwerkdiensten, dienen beveiligingseisen in acht te worden genomen voordat deze configuratie wordt geïmplementeerd. Soms kunnen logbestanden gevoelige gegevens bevatten over diensten die aanstaan op de lokale host, gebruikersaccounts, en configuratiegegevens. Netwerkgegevens die van de cliënt naar de server worden verzonden worden niet versleuteld noch met een wachtwoord beveiligd. Als versleuteling nodig is, kan security/stunnel worden gebruikt, wat gegevens over een versleutelde tunnel verstuurt.

Aan lokale beveiliging moet ook gedacht worden. Logbestanden worden niet versleuteld tijdens gebruik of na logrotatie. Lokale gebruikers kunnen deze bestanden benaderen om aanvullende inzichten over de systeemconfiguratie op te doen. In deze gevallen is het van kritiek belang om de juiste rechten op deze bestanden in te stellen. Het gereedschap syslogd(8) ondersteunt het instellen van rechten op nieuw aangemaakte en geroteerde logbestanden. Het instellen van logbestanden op modus `600` dient al het ongewenste spieken door lokale gebruikers te verhinderen.

Hoofdstuk 28. Firewalls

28.1. Inleiding

Firewalls bieden de mogelijkheid om inkomend en uitgaand verkeer op een systeem te filteren. Een firewall gebruikt daarvoor een of meer groepen regels ("rules") om netwerkpakketten te inspecteren als ze binnenkomen of weggaan door netwerkverbindingen en staat dat verkeer dan toe of blokkeert het. De regels van een firewall kunnen één of meerdere eigenschappen van pakketten onderzoeken waaronder, maar niet uitsluitend, het protocol, het bron- of bestemmingsadres en de bron- en bestemmingspoort.

Firewalls kunnen de veiligheid van een host of netwerk enorm vergroten. Ze kunnen één of meer van de volgende dingen doen:

- Applicaties, diensten en machines op een intern netwerk te beschermen tegen ongewild verkeer van het Internet.
- Toegang tot Internet voor interne hosts te limiteren of uitschakelen.
- Ondersteuning bieden voor netwerkadres vertaling ("network address translation" of NAT), waarmee er vanaf een intern netwerk met private IP adressen een Internetverbinding gedeeld kan worden met één IP adres of met een groep van publieke adressen die automatisch wordt toegewezen.

Na het lezen van dit hoofdstuk weet de lezer:

- Hoe pakketfilteringsregels op de juiste wijze samengesteld kunnen worden;
- De verschillen tussen de firewalls die bij FreeBSD worden geleverd;
- Hoe de OpenBSD firewall PF te gebruiken en in te stellen;
- Hoe IPFILTER te gebruiken en in te stellen;
- Hoe IPFW te gebruiken en in te stellen.

Er wordt aangenomen dat de lezer van dit hoofdstuk:

- Basisbegrip heeft van FreeBSD en Internetconcepten.

28.2. Firewallconcepten

Er zijn twee basismogelijkheden om sets met regels te maken voor firewalls: "inclusief" of "exclusief". Een exclusieve firewall staat al het verkeer door behalve het verkeer dat past bij de set met regels. Een inclusieve firewall doet het tegenovergestelde. Die staat alleen verkeer toe dat past bij de regels en blokkeert al het overige verkeer.

Een inclusieve firewall biedt veel betere controle over het uitgaande verkeer, waardoor het een betere keuze is voor systemen die diensten op het publieke Internet aanbieden. Het beheert ook het type verkeer dat van het publieke Internet afkomt en toegang heeft tot uw privé-netwerk. Al het verkeer dat niet aan de regels voldoet wordt geblokkeerd en gelogd, dat is zo ontworpen. Inclusieve firewalls zijn over het algemeen veiliger dan exclusieve firewalls omdat ze het risico dat ongewenst

verkeer door ze heen gaat aanzienlijk verminderen.



Tenzij anders aangegeven, creëeren alle configuraties en voorbeelden van regelverzamelingen in dit hoofdstuk inclusieve firewalls.

De beveiliging kan nog verder vergroot worden met een "stateful firewall". Dit type firewall houdt bij welke connecties er door de firewall tot stand zijn gekomen en laat alleen verkeer door dat bij een bestaande connectie hoort of onderdeel is van een connectie in opbouw. Het nadeel van een stateful firewall is dat die kwetsbaar kan zijn voor Ontzegging van Dienst (DoS) aanvallen als er een groot aantal nieuwe verbindingen binnen korte tijd wordt opgezet. Met de meeste firewalls is het mogelijk een combinatie te maken van stateful en niet stateful gedrag om een optimale firewall voor een site te maken.

28.3. Firewallsoftware

FreeBSD heeft drie soorten firewallsoftware in de basisinstallatie. Dat zijn: IPFILTER (ook bekend als IPF), IPFIREWALL (ook bekend als IPFW) en de pakketfilter van OpenBSD (ook bekend als PF). FreeBSD heeft ook twee ingebouwde pakketten voor het regelen van verkeer (in de basis het beheersen van bandbreedtegebruik): altq(4) en dummynet(4). Dummynet is traditioneel sterk verbonden met IPFW en ALTQ met PF. Het vormgeven van verkeer voor IPFILTER kan momenteel gedaan worden met IPFILTER voor NAT en filtering en IPFW met dummynet(4) *of* door PF met ALTQ te gebruiken. IPFW en PF gebruiken allemaal regels om de toegang van pakketten tot een systeem te regelen, hoewel ze dat op andere manieren doen en ze een andere regelsyntaxis hebben.

De reden dat er meerdere firewallpakketten in FreeBSD zitten is dat verschillende mensen verschillende eisen en voorkeuren hebben. Geen enkel firewallpakket is het beste.

De schrijver van dit artikel geeft de voorkeur aan IPFILTER omdat daarmee stateful regels minder complex zijn toe te passen in een omgeving waar NAT wordt gebruikt en IPF heeft een ingebouwde FTP proxy waardoor de regels voor het veilig gebruiken van FTP eenvoudiger worden.

Omdat alle firewalls gebaseerd zijn op het inspecteren van aangegeven controlevelden in pakketten, moet iemand die sets van firewallregels opstelt begrijpen hoe TCP/IP werkt, welke waarde de controlevelden kunnen hebben en hoe die waarden gebruikt worden in normaal verkeer. Op de volgende webpagina wordt een prima uitleg gegeven: http://www.ipprimer.com/overview.cfm.

28.4. De OpenBSD Packet Filter (PF) en ALTQ

Vanaf juli 2003 is de OpenBSD firewalltoepassing PF geporteerd naar FreeBSD en beschikbaar gekomen in de FreeBSD Portscollectie. In 2004 was FreeBSD 5.3 de eerste release die PF bevatte is integraal onderdeel van het basissysteem. PF is een complete en volledige firewall die optioneel ALTQ bevat (Alternate Queuing). ALTQ biedt Quality of Service (QoS) functionaliteit.

het OpenBSD Project levert een uitstekend werk wat betreft het onderhouden van de PF FAQ. Zodoende zal deze sectie van het handboek zich richten op PF met betrekking tot FreeBSD terwijl het ook wat algemene informatie over het gebruik zal geven. Voor gedetailleerde gebruikersinformatie wordt naar de PF FAQ verwezen.

Meer informatie over PF voor FreeBSD staat op http://pf4freebsd.love2party.net/.

28.4.1. De laadbare kernelmodules voor PF gebruiken

Voeg de volgende regel toe aan /etc/rc.conf om de kernelmodule PF te laden:

```
pf_enable="YES"
```

Draai vervolgens het opstartscript om de module te laden:

```
# service pf start
```

Merk op dat de PF module niet laadt als het het instellingenbestand met de regelverzameling niet kan vinden. De standaardlocatie is /etc/pf.conf. Als de regelverzameling voor PF zich elders bevindt, kan PF worden verteld om daar te kijken een regel analoog aan de volgende aan /etc/rc.conf toe te voegen:

```
pf_rules="/pad/naar/pf.conf"
```

Het voorbeeld pf.conf bestand kan gevonden worden in /usr/shared/examples/pf

De module PF kan ook handmatig vanaf de opdrachtregel geladen worden:

```
# kldload pf.ko
```

Logondersteuning voor PF wordt geleverd door `pflog.ko` en kan worden geladen door de volgende regel aan /etc/rc.conf toe te voegen:

```
pflog_enable="YES"
```

Draai vervolgens het opstartscript om de module te laden:

```
# service pflog start
```

Als u andere mogelijkheden van PF nodig heeft dient u ondersteuning voor PF in de kernel te compileren.

28.4.2. Kernelopties voor PF

Hoewel het niet nodig is om ondersteuning voor PF in de kernel te compileren, biedt dit wel de mogelijkheid om van een van PF's geavanceerde mogelijkheden gebruik te maken die niet in de laadbare module zitten, namelijk pfsync(4), dat een pseudo-apparaat is dat zekere veranderingen aan de toestandstabel die door PF wordt gebruikt prijsgeeft. Het kan worden gecombineerd met

carp(4) om failover firewalls aan te maken die gebruik maken van PF. Meer informatie over CARP kan gevonden worden in Common Address Redundancy Protocol (CARP) van het Handboek.

De kernelopties voor PF kunnen gevonden worden in /usr/src/sys/conf/NOTES en zijn hieronder gereproduceerd:

```
device pf
device pflog
device pfsync
```

De optie `device pf` schakelt ondersteuning voor de "Packet Filter" firewall (pf(4)) in.

De optie `device pflog` schakelt het optionele pflog(4) pseudo-netwerkapparaat in dat gebruikt kan worden om verkeer te loggen naar een bpf(4) descriptor. De pflogd(8) daemon kan gebruikt worden om de logboekinformatie naar schijf te schrijven.

De optie `device pfsync` schakelt het optionele pfsync(4) pseudo netwerkapparaat in waarmee de toestandswijzigingen gemonitord kunnen worden.

28.4.3. Beschikbare opties voor rc.conf

De volgende rc.conf(5) statements stellen PF en pflog(4) in tijdens het opstarten:

```
pf_enable="YES"                 # Schakel PF in (laad module als nodig)
pf_rules="/etc/pf.conf"         # bestand met regels voor pf
pf_flags=""                     # aanvullende vragen voor opstarten pfctl
pflog_enable="YES"              # start pflogd(8)
pflog_logfile="/var/log/pflog"  # waar pflogd het logboekbestand moet opslaan
pflog_flags=""                  # aanvullende vlaggen voor opstarten pflogd
```

Als er een LAN achter de firewall staat en er pakketten doorgestuurd moeten worden naar computers op het LAN of als NAT actief is, dan is de volgende optie ook nodig:

```
gateway_enable="YES"            # Schakel in als LAN gateway
```

28.4.4. Filterregels aanmaken

PF leest de instelregels van pf.conf(5) (standaard /etc/pf.conf) en het verandert, verwijdert, of geeft pakketten door aan de hand van de regels of definities die daar zijn gespecificeerd. De FreeBSD-installatie bevat een aantal voorbeeldbestanden in /usr/shared/examples/pf/. In de PF FAQ staat een complete behandeling van de PF regels.



Houd tijdens het doornemen van de PF FAQ in de gaten dat verschillende versies van FreeBSD verschillende versies van PF kunnen bevatten. Momenteel gebruikt FreeBSD 8.*X* dezelfde versie van PF als OpenBSD 4.1. FreeBSD 9.*X* en hoger gebruiken dezelfde versie van PF als OpenBSD 4.5.

De FreeBSD pakketfilter mailinglijst is een goede plaats om vragen over het instellen en draaien van de PF firewall te stellen. Vergeet niet de mailinglijstarchieven te controleren alvorens vragen te stellen!

28.4.5. Werken met PF

Gebruik pfctl(8) om PF te beheren. Hieronder staan wat nuttige commando's (bekijk de hulppagina pfctl(8) voor alle beschikbare opties):

Commando	Doel		
`pfctl -e`	PF aanzetten		
`pfctl -d`	PF uitzetten		
`pfctl -F all -f /etc/pf.conf`	Spoel alle regels door (nat, filter, toestand, tabel, etc.) en herlaad vanuit het bestand /etc/pf.conf	`pfctl -s [ rules | nat | state ]`	Rapporteer over de filterregels, NAT-regels, of toestandstabel
`pfctl -vnf /etc/pf.conf`	Controleer /etc/pf.conf op fouten, maar laad de regelverzameling niet		

28.4.6. ALTQ inschakelen

ALTQ is alleen beschikbaar ondersteuning ervoor in de FreeBSD Kernel te compileren. ALTQ wordt niet door alle netwerkkaartstuurprogramma's ondersteund. In altq(4) staat een lijst met ondersteunde stuurprogramma's voor de betreffende versie.

Met de volgende opties wordt ALTQ ingeschakeld en additionele functionaliteit toegevoegd:

```
options         ALTQ
options         ALTQ_CBQ        # Class Bases Queuing (CBQ)
options         ALTQ_RED        # Random Early Detection (RED)
options         ALTQ_RIO        # RED In/Out
options         ALTQ_HFSC       # Hierarchical Packet Scheduler (HFSC)
options         ALTQ_PRIQ       # Priority Queuing (PRIQ)
options         ALTQ_NOPCC      # Required for SMP build
```

`options ALTQ` schakelt het ALTQ raamwerk in.

`options ALTQ_CBQ` schakelt *Class Based Queuing* (CBQ) in. Met CBQ kan de bandbreedte van een verbinding worden opgedeeld in verschillende klassen of wachtrijen om verkeer te prioriteren op basis van filterregels.

`options ALTQ_RED` schakelt *Random Early Detection* (RED) in. RED wordt gebruikt om netwerkverstopping te voorkomen. RED doet dit door de lengte van de wachtrij te meten en die te vergelijken met de minimale en maximale drempelwaarden voor de wachtrij. Als de wachtrij

groter is dan de maximale waarde worden alle nieuwe pakketten genegeerd. Het werkt naar zijn naam, dus RED negeert willekeurig pakketten van verschillende verbindingen.

`options ALTQ_RIO` schakelt *Random Early Detection In and Out* in.

`options ALTQ_HFSC` schakelt de *Hierarchical Fair Service Curve Packet Scheduler* in. Meer informatie over HFSC staat op http://www-2.cs.cmu.edu/~hzhang/HFSC/main.html.

`options ALTQ_PRIQ` schakelt *Priority Queuing* (PRIQ) in. PRIQ laat verkeer dat in een hogere wachtrij staat altijd eerder door.

`options ALTQ_NOPCC` schakelt SMP ondersteuning voor ALTQ in. Deze optie is verplicht op SMP systemen.

28.5. De IPFILTER (IPF) firewall

Darren Reed is de auteur van IPFILTER, dat niet afhankelijk is van één besturingssysteem. Het is een open source applicatie die is geporteerd naar FreeBSD, NetBSD, OpenBSD, SunOS, HP/UX en Solaris besturingssystemen. IPFILTER wordt actief ondersteund en onderhouden en er worden regelmatig nieuwe versies uigebracht.

IPFILTER is gebaseerd op een firewall aan de kernelkant en een NAT mechanisme dat gecontroleerd en gemonitord kan worden door programma's in userland. De firewallregels kunnen ingesteld of verwijderd worden met het hulpprogramma ipf(8). De NAT regels kunnen ingesteld of verwijderd worden met ipnat(8). Het programma ipfstat(8) kan actuele statistieken leveren voor de kernelonderdelen van IPFILTER. ipmon(8) kan acties van IPFILTER wegschrijven naar logboekbestanden van het systeem.

IPF is oorspronkelijk geschreven met logica die regels verwerkte volgens het principe "de laatst passende regel wint" en gebruikte toen alleen staatloze regels. In de loop der tijd is IPF verbeterd en zijn de opties `quick` en `keep state` toegevoegd waarmee de logica van het verwerken van regels drastisch is gemoderniseerd. In de officiële documentatie van IPF worden alleen de regels en verwerkingslogica behandeld. De moderne functies worden alleen behandeld als opties, waardoor hun nut dat er een veel betere en veiligere firewall mee te maken volledig onderbelicht blijft.

De instructies in dit hoofdstuk zijn gebaseerd op regels die gebruik maken van de optie `quick` en de stateful optie `keep state`. Dit is het raamwerk waarmee een set van inclusieve firewallregels wordt samengesteld.

Voor een gedetailleerde uitleg over de verwerking van de verouderde regels zie http://www.munk.me.uk/ipf/ipf-howto.html en http://coombs.anu.edu.au/~avalon/ip-filter.html.

De IPF FAQ is te vinden op http://www.phildev.net/ipf/index.html.

Een doorzoekbaar archief van de open-source IPFilter mailing lijst is beschikbaar op http://marc.theaimsgroup.com/?l=ipfilter.

28.5.1. IPF inschakelen

IPF zit in de basisinstallatie van FreeBSD als een aparte "run time" laadbare module. Een systeem

laadt de IPF kernel laadbare module dynamisch als `ipfilter_enable="YES"` in rc.conf staat. Voor de laadbare module zijn de opties `logging` en `default pass all` ingeschakeld. IPF hoeft niet in de kernel gecompileerd te worden om het standaardgedrag te wijzigen naar `block all`. Dat is mogelijk door op het einde van de regelverzameling een regel `block all` toe te voegen die al het verkeer blokkeert.

28.5.2. Kernelopties

Het is niet verplicht om IPF in te schakelen door de volgende opties in de FreeBSD kernel te compileren. Dit wordt alleen beschreven als achtergrondinformatie. Door IPF in de kernel te compileren wordt de laadbare module niet gebruikt.

Voorbeeld kernelinstellingen voor IPF staan beschreven in de /usr/src/sys/i386/conf/LINT in de kernelbroncode en worden hier beschreven:

```
options IPFILTER
options IPFILTER_LOG
options IPFILTER_DEFAULT_BLOCK
```

`options IPFILTER` schakelt ondersteuning voor de "IPFILTER" firewall in.

`options IPFILTER_LOG` schakelt de optie in waarmee IPF verkeer kan loggen door het naar het ipl pakketloggende pseudo-apparaat te schrijven voor iedere regel met het sleutelwoord `log` erin.

`options IPFILTER_DEFAULT_BLOCK` wijzigt het standaardgedrag zodat ieder pakket waarop geen enkele `pass` regel van toepassing is wordt geblokkeerd.

Deze instelling worden pas actief nadat een kernel waarvoor deze instellingen zijn gemaakt is gebouwd en geïnstalleerd.

28.5.3. Beschikbare opties voor rc.conf

De volgende instellingen moeten in /etc/rc.conf staan om IPF bij het opstarten te activeren:

```
ipfilter_enable="YES"             # Start ipf firewall
ipfilter_rules="/etc/ipf.rules"   # laad regels uit het doelbestand
ipmon_enable="YES"                # Start IP monitor log
ipmon_flags="-Ds"                 # D = start als daemon
                                  # s = log naar syslog
                                  # v = log tcp window, ack, seq
                                  # n = vertaal IP & poort naar namen
```

Als er een LAN achter de firewall staat dat gebruik maakt van IP-adressen uit de private reeks, dan moet de volgende optie ook ingesteld worden om NAT-functionaliteit in te schakelen:

```
gateway_enable="YES"              # Schakel in als LAN gateway
ipnat_enable="YES"                # Start ipnat functie
ipnat_rules="/etc/ipnat.rules"    # bestand met regels voor ipnat
```

28.5.4. IPF

Het commando ipf(8) wordt gebruikt om het bestand met firewallregels te laden. Gewoonlijk wordt er een bestand aangemaakt waarin de situatieafhankelijke regels staan waarmee in één keer de bestaande regels kunnen worden vervangen:

```
# ipf -Fa -f /etc/ipf.rules
```

`-Fa`: verwijder alle interne tabellen met regels.

`-f`: laad het aangegeven bestand met regels.

Hiermee wordt het mogelijk wijzigingen te maken aan het bestand met eigen regels en met ipf(8) de firewall aan te passen met verse regels zonder het systeem te booten. Deze methode is erg handig om nieuwe regels te testen omdat dit zo vaak als nodig gedaan kan worden.

In ipf(8) worden alle opties die beschikbaar zijn toegelicht.

ipf(8) verwacht dat het bestand met regels een standaard tekstbestand is. Het accepteert geen bestand met regels dat is opgesteld als een script dat gebruik maakt van substitutie.

Er is wel een mogelijkheid om IPF regels op te stellen en gebruik te maken van substitutie. Meer informatie staat in Script met regels met substitutie bouwen.

28.5.5. IPFSTAT

ipfstat(8) haalt de totalen van de statistieken op die horen bij de firewall sinds die is gestart en toont deze. Het kan ook zijn dat de tellers in tussentijd op nul zijn gesteld met `ipf -Z`.

In ipfstat(8) worden alle details behandeld.

Standaard ziet ipfstat(8) uitvoer er ongeveer als volgt uit:

```
input packets: blocked 99286 passed 1255609 nomatch 14686 counted 0
output packets: blocked 4200 passed 1284345 nomatch 14687 counted 0
input packets logged: blocked 99286 passed 0
output packets logged: blocked 0 passed 0
packets logged: input 0 output 0
log failures: input 3898 output 0
fragment state(in): kept 0 lost 0
fragment state(out): kept 0 lost 0
packet state(in): kept 169364 lost 0
packet state(out): kept 431395 lost 0
ICMP replies: 0 TCP RSTs sent: 0
Result cache hits(in): 1215208 (out): 1098963
IN Pullups succeeded: 2 failed: 0
OUT Pullups succeeded: 0 failed: 0
Fastroute successes: 0 failures: 0
TCP cksum fails(in): 0 (out): 0
```

```
Packet log flags set: (0)
```

Als er als optie `-i` voor inkomend of `-o` voor uitgaand wordt meegegeven, dan zal het commando de juiste lijst met regels die de kernel op dat moment gebruikt wordt weergeven.

`ipfstat -in` toont de tabel met regels voor inkomend verkeer met regelnummers

`ipfstat -on` toont de tabel met regels voor uitgaand verkeer met regelnummers

De uitvoer ziet er ongeveer als volgt uit:

```
@1 pass out on xl0 from any to any
@2 block out on dc0 from any to any
@3 pass out quick on dc0 proto tcp/udp from any to any keep state
```

`ipfstat -ih` toont de tabel met regels voor inkomend verkeer, waarbij voor iedere regel staat hoe vaak die van toepassing was.

`ipfstat -oh` toont de tabel met regels voor uitgaand verkeer, waarbij voor iedere regel staat hoe vaak die van toepassing was.

De uitvoer ziet er ongeveer als volgt uit:

```
2451423 pass out on xl0 from any to any
354727 block out on dc0 from any to any
430918 pass out quick on dc0 proto tcp/udp from any to any keep state
```

Een van de belangrijkste functies van `ipfstat` is de vlag `-t` waarmee de staat-tabel wordt getoond op een wijze die vergelijkbaar is met de wijze waarop top(1) de draaiende FreeBSD procestabel toont. Als een firewall wordt aangevallen, dan geeft deze functie de mogelijkheid om de pakketten van de aanvaller te identificeren en nader te onderzoeken. De optionele subvlaggen bieden de mogelijkheid om een bron of bestemmings IP adres, poort of protocol aan te geven dat gemonitord moet worden. Details zijn na te lezen in ipfstat(8).

28.5.6. IPMON

Om ipmon(8) te laten werken zoals bedoeld, moet de kerneloptie `IPFILTER_LOG` aan staan. Dit commando kan op twee verschillende wijzen gebruikt worden. De standaard is van toepassing als het commando op de commandoregel wordt ingegeven zonder de optie `-D`.

De daemon wordt gebruikt als continu een systeemlogboek bijgewerkt moet worden zodat het mogelijk is om gebeurtenissen in het verleden te bekijken. Zo zijn FreeBSD en IPFILTER ingesteld om samen te werken. FreeBSD heeft ingebouwde mogelijkheden om automatisch syslogs te roteren. Daarom is het beter om de uitvoer naar syslogd(8) te schrijven dan naar een gewoon bestand. In de standaardversie van rc.conf is te zien dat de instelling `ipmon_flags` de waarde `-Ds` heeft:

```
ipmon_flags="-Ds" # D = start als daemon
```

```
                # s = log naar syslog
                # v = log tcp window, ack, seq
                # n = vertaal IP & poort naar namen
```

De voordelen van loggen zijn duidelijk. Het biedt de mogelijkheid om na het feit informatie na te zien als: welke pakketten heeft de firewall laten vallen, waar kwamen ze vandaan en waar gingen ze heen? Dit zijn allemaal voordelen als het gaat om uitvinden waar een aanvaller vandaan komt en wat deze heeft geprobeerd.

Zelfs als loggen is ingeschakeld, logt IPF nog niets uit zichzelf. De beheerder van de firewall beslist welke regels in de regelverzameling iets weg moeten schrijven door het sleutelwoord `log` aan die regels toe te voegen. Gewoonlijk worden alleen `deny` regels gelogd.

Het is heel normaal om als laatste regel een `deny` regel aan de set met regels toe te voegen waar het sleutelwoord `log` in staat. Zo krijgt een beheerder alle pakketten te zien waarop geen enkele regel van toepassing was.

28.5.7. Loggen met IPMON

Syslogd heeft een eigen methode om logboekgegevens te scheiden. Het maakt gebruik van speciale groepen die "facility" en "level" heten. ipmon(8) in `-Ds` mode gebruikt `local0` als "facility"naam. Alle door ipmon(8) gelogde gegevens gaan standaard naar de naam `security`. De nu volgende levels kunnen gebruikt worden om de gelogde gegevens nog verder uit elkaar te trekken als dat gewenst is.

```
LOG_INFO – pakketten gelogd met het sleutelwoord "log" als actie in plaats van pass of
block.
LOG_NOTICE – gelogde pakketten die ook zijn doorgelaten
LOG_WARNING – gelogde pakketten die ook geblokkeerd zijn
LOG_ERR – gelogde pakketten die een verkeerde opbouw hebben, "short"
```

Om IPFILTER alle gelogde gegevens naar /var/log/ipfilter.log te laten schrijven, dient dat bestand vooraf te bestaan. Dat kan met het volgende commando:

```
# touch /var/log/ipfilter.log
```

De functionaliteit van syslogd(8) wordt beheerd met instellingen in /etc/syslog.conf. Dit bestand biedt aanzienlijke flexibiliteit in hoe syslog omgaat met systeemberichten die door softwaretoepassingen als IPF worden gegeven.

Zo kan de volgende instelling toegevoegd worden aan /etc/syslog.conf:

```
local0.* /var/log/ipfilter.log
```

Het deel `local0.*` betekent dat alle logberichten naar de aangegeven plaats geschreven moeten worden.

Om de wijzigingen in /etc/syslog.conf actief te maken kan er opnieuw opgestart worden of is het mogelijk de daemon syslogd(8) een schop te geven zodat /etc/syslog.conf opnieuw wordt ingelezen met `/etc/rc.d/syslogd reload`. Het PID (procesnummer) is te achterhalen door een overzicht van taken te tonen met `ps -ax`. Het PID is het nummer in de linker kolom voor de regel waarop "syslog" staat.

Vaak wordt vergeten /etc/newsyslog.conf te wijzigen om het nieuw aangemaakte logboekbestand te laten roteren.

28.5.8. De opmaak van gelogde berichten

Berichten die door `ipmon` wordt gezonden bestaan uit velden die gescheiden worden door een spatie. Velden die in alle berichten zitten zijn:

1. De datum waarop het pakket is ontvangen.
2. De tijd waarop het pakket is ontvangen weergegeven als HH:MM:SS.F voor uren, minuten, seconden en fracties van een seconde. De fractie kan meerdere cijfers lang zijn.
3. De naam van de interface waarop het pakket is ontvangen, bijvoorbeeld dc0.
4. De groep en regelnummer van de regel, bijvoorbeeld `@0:17`.

Deze kunnen ingezien worden met `ipfstat -in`.

1. De acties: `p` voor doorgelaten ("passed"), `b` voor geblokkeerd ("blocked"), `S` voor een verkeerd pakket ("short packet"), `n` voor dat er geen enkele regel van toepassing was, `L` voor een logboekregel. De volgorde waarin deze acties getoond worden is: S, p, b, n, L. Een hoofdletter `P` of `B` betekent dat het pakket gelogd is vanwege een globale instelling, niet vanwege één regel in het bijzonder.
2. De adressen. Dit zijn eigenlijk drie velden: het bronadres en poort gescheiden door een komma, het symbool → en het bestemmingsadres en poort, bijvoorbeeld: `209.53.17.22,80 → 198.73.220.17,1722`.
3. Achter `PR` staat de naam van het protocol of het nummer, bijvoorbeeld `PR tcp`.
4. Achter `len` staan de lengte van de pakketkop en de totale lengte van het pakket, bijvoorbeeld `len 20 40`.

Als het pakket een TCP pakket is, dan is er nog een veld dat begint met een verbindingsstreepje met daarachter letters die overeenkomen met vlaggen die ingeschakeld waren. In ipf(5) is een lijst met letters en bijbehorende vlaggen te vinden.

Als het pakket een ICMP pakket is, dan worden aan het einde twee velden toegevoegd. Het eerste is altijd `ICMP` en het volgende het ICMP bericht en subbericht type, gescheiden door een slash, bijvoorbeeld `ICMP 3/3` voor "een poort niet bereikbaar" bericht.

28.5.9. Script met regels met substitutie bouwen

Geoefende gebruikers van IPF maken een bestand dat de regels bevat en stellen dat op zo'n manier op dat het uitgevoerd kan worden als een script met substitutie. Het grote voordeel van deze werkwijze is dat er dan alleen de waarde geassocieerd met een symbolische naam gewijzigd hoeft

te worden en dat als het script opnieuw wordt uitgevoerd, op alle plaatsen waar de variabele wordt gebruikt, de nieuwe waarde in de regels wordt opgenomen. Omdat het een script is, kan substitutie gebruik worden om vaak voorkomende waarden de definiëren zodat ze in meerdere regels vervangen kunnen worden. Dit wordt geïllustreerd in het onderstaande voorbeeld.

De syntaxis die in het script wordt gebruikt is compatibel met de shells sh(1), csh(1) en tcsh(1).

Velden waarvoor substitutie van toepassing is worden vooraf gegaan door het dollarteken $.

Definities worden niet vooraf gegaan door het voorvoegsel $.

De waarden van een definitie moet omsloten worden door dubbele aanhalingstekens (").

Een set regels begint wellicht als volgt:

```
############### Begin IPF regels script ##########################
oif="dc0"            # naam van de uitgaande interface
odns="192.0.2.11"    # IP adres van DNS server ISP
myip="192.0.2.7"     # statische IP adres gekregen van ISP
ks="keep state"
fks="flags S keep state"

# Er kan gekozen worden om dit script te gebruiken om een eigen
# /etc/ipf.rules script te maken of dit script kan gebruikt worden
# "as is"
#
# Haal bij één van deze regels het commentaarteken weg
# en plaats hem bij de ander.
#
# 1) Deze kan gebruikt worden om /etc/ipf.rules te maken:
#cat > /etc/ipf.rules << EOF
# 2) Deze kan gebruikt worden om het script "as is" te starten:
# Let op: er moet een lege regel zijn na het EOF teken.
/sbin/ipf -Fa -f - << EOF

# Verleen toegang tot de DNS van de ISP.
pass out quick on $oif proto tcp from any to $odns port = 53 $fks
pass out quick on $oif proto udp from any to $odns port = 53 $ks

# Sta uitgaand verkeer voor niet beveiligd www verkeer toe
pass out quick on $oif proto tcp from $myip to any port = 80 $fks

# Sta uitgaand verkeer voor beveiligd www verkeer toe (https over TLS SSL)
pass out quick on $oif proto tcp from $myip to any port = 443 $fks
EOF
################## Einde IPF regels script ########################
```

Dat is alles. De regels zijn niet van belang in dit voorbeeld, maar tonen hoe substitutievelden worden gedefinieerd en hoe ze worden gebruikt. Als het bovenstaande voorbeeld de inhoud van /etc/ipf.rules.script was, dan konden deze regels herladen worden door het vanaf de

commandoregel aan te roepen:

```
# sh /etc/ipf.rules.script
```

Er is wel een probleem met het gebruik van regels in combinatie met substitutie. IPF snapt het niet en kan deze scripts niet direct lezen.

Dit script kan gebruikt worden op één van de volgende twee manieren:

- Haal het commentaarteken weg bij de regel die begint met `cat` en zet het commentaarteken bij de regel die begint met `/sbin/ipf`. Plaats `ipfilter_enable="YES"` in /etc/rc.conf zoals gewoonlijk en start het script eenmalig na elke wijziging om /etc/ipf.rules te maken of bij te werken.
- Schakel IPFILTER uit in de systeem opstart scripts door `ipfilter_enable="NO"` toe te voegen aan /etc/rc.conf (dit is de standaardwaarde).

  Voeg een script zoals de volgende toe aan de opstartmap /usr/local/etc/rc.d. Het script zou een duidelijke naam moeten hebben zoals ipf.loadrules.sh. De uitbreiding .sh is noodzakelijk.

  ```
  #!/bin/sh
  sh /etc/ipf.rules.script
  ```

  De permissies op dit script moeten zijn: lezen,schrijven en uitvoeren voor de gebruiker `root`.

  ```
  # chmod 700 /usr/local/etc/rc.d/ipf.loadrules.sh
  ```

Als het systeem nu herstart, worden de regels via het script gestart.

28.5.10. Sets van IPF regels

Een set regels is een groep IPF-regels die is gemaakt om pakketten toe te staan of te blokkeren op basis van de eigenschappen van dat pakket. De bi-directionele uitwisseling van pakketten tussen hosts bestaat uit een gesprek dat een sessie heet. De set van firewallregels verwerkt zowel de pakketten die arriveren van het publieke Internet, als de pakketten die door het systeem zijn geproduceerd als een antwoord erop. Elke TCP/IP-dienst (telnet, www, mail, enzovoorts) is vooraf gedefinieerd door een protocol en bevoorrechte (luister)poort. Pakketten bedoeld voor een speciale dienst beginnen bij het bronadres gebruik makend van een onbevoorrechte (hogere orde) poort en komen aan bij de specifieke dienstpoort op het bestemmingsadres. Alle bovengenoemde parameters (poorten en adressen) kunnen gebruikt worden als selectiecriteria om regels aan te maken die diensten zullen toestaan of blokkeren.

IPF is oorspronkelijk geschreven met logica die regels verwerkte volgens het principe "de laatst passende regel wint" en gebruikte toen alleen staatloze regels. In de loop der tijd is IPF verbeterd en zijn de opties "quick" en "keep state" toegevoegd waarmee de logica van het verwerken van regels drastisch is gemoderniseerd.

De instructies in dit hoofdstuk zijn gebaseerd op regels die gebruik maken van de optie "quick" en

de stateful optie "keep state". Dit is het raamwerk waarmee een set van inclusieve firewallregels wordt samengesteld.



Werk bij het wijzigen van firewallregels *zeer voorzichtig*. Met sommige instellingen is een server *niet meer bereikbaar*. Om het veilig te spelen is het aan te raden de eerste instellingen vanaf het console te maken, in plaats van via ssh.

28.5.11. Regelsyntaxis

De regelsyntaxis die hier wordt besproken is versimpeld door alleen de moderne stateful regels en de "eerste van toepassing zijnde regel wint" te belichten. De complete regelsyntaxis is na te lezen in ipf(8).

Het karakter `#` wordt gebruikt om het begin van een opmerking te markeren en zowel op een eigen regel als achter een firewallregel staan. Lege regels worden genegeerd.

Regels bevatten sleutelwoorden die in een bepaalde volgorde van links naar rechts op een regel horen te staan. Sleutelwoorden worden vet weergegeven. Sommige sleutelwoorden hebben subopties die zelf ook weer sleutelwoorden hebben die ook weer subopties kunnen hebben. Alle opties die hier direct onder staan, worden daaronder uitgebreid weergegeven en verderop in dit hoofdstuk in een aparte paragraaf behandeld.

ACTIE IN/UIT OPTIES SELECTIE STATEFUL PROTO BRON_ADR,BEST_ADR OBJECT POORT_NUM TCP_VLAG STATEFUL

ACTIE = block | pass

IN/UIT = in | out

OPTIES = log | quick | on interfacenaam

SELECTIE = protowaarde | bron/bestemming IP | poort = nummer | flags flag-value

PROTO = tcp/udp | udp | tcp | icmp

BRON_ADR,BEST_ADR = all | from object to object

OBJECT = IP adres | any

POORT_NUM = poortnummer

TCP_VLAG = S

STATEFUL = keep state

ACTIE

De actie geeft aan wat er met het pakket gedaan moet worden als het van toepassing is op de rest van de filterregel. Iedere regel *moet* een actie hebben. De volgende acties zijn mogelijk:

`block` geeft aan dat het pakket moet verdwijnen als de parameters van toepassing zijn op het

pakket.

`pass` geeft aan dat het pakket doorgelaten moet worden als de parameters van toepassing zijn op het pakket.

IN/UIT

Een verplicht onderdeel voor iedere filterregel waarin expliciet wordt aangegeven op welke zijde van de in/uit deze van toepassing is. Het volgende sleutelwoord moet `in` of `out` zijn en één van de twee moet gecodeerd worden, anders is de regel syntactisch onjuist.

`in` betekent dat de regel van toepassing is op inkomende pakketten.

`out` betekent dat de regel van toepassing is op inkomende pakketten.

OPTIES



Deze opties moeten in de volgorde waarin ze hier beschreven staan gebruikt worden.

`log` geeft aan dat het pakket naar het ipl logboekbestand geschreven moeten worden (zoals verderop beschreven staat in de paragraaf "Loggen") als de regel van toepassing is op het pakket.

`quick` geeft aan dat als een regel van toepassing is, dat de laatste regel moet zijn die wordt gecontroleerd, waardoor er een pad wordt "kortgesloten" waardoor de volgende regels voor dat pakket niet meer gecontroleerd worden. Deze optie is voor de moderne regels eigenlijk verplicht.

`on` geeft de interface aan die in de parameters meegenomen moet worden. De namen van interfaces kunnen getoond worden met ifconfig(8). Als deze optie wordt gebruikt, kan een regel alleen van toepassing zijn als het pakket door de aangegeven interface gaat in de richting die is aangegeven (`in`/`out`). Ook deze optie is verplicht voor de moderne regels.

Als een pakket wordt gelogd, dan worden de koppen van het pakket weggeschreven naar het ipl pakketloggende pseudo-apparaat. Direct na het sleutelwoord `log` mogen de volgende opties gebruikt worden (in de aangegeven volgorde):

`body` geeft aan dat de eerste 128 bytes van de inhoud van het pakket worden opgeslagen na de kop.

`first`; als het sleutelwoord `log` samen met een optie `keep state` wordt gebruikt, wordt het aangeraden om deze optie ook te gebruiken zodat alleen het pakket dat als eerste in de sessie van toepassing was en niet ook alle pakketten die daarna in de sessie volgens `keep state` van toepassing zijn.

SELECTIE

De sleutelwoorden in deze paragraaf worden gebruikt om attributen van het pakket dat wordt geïnspecteerd te beschrijven om te bepalen of een regel wel of niet van toepassing is. Er is een sleutelwoord `subject` en er zijn subopties waarvan er één of meer gekozen moeten worden. De volgende attributen zijn beschikbaar voor het proces en moeten in de aangegeven volgorde worden gebruikt:

PROTO

`proto` is het `subject` sleutelwoord dat moet worden aangegeven samen met een van de sleutelwoorden uit de subopties. De waarde geeft een bepaald protocol aan dat van toepassing moet zijn. Ook deze optie is verplicht voor de moderne regels.

`tcp/udp`, `tcp`, `udp`, `icmp` of ieder ander protocol dat in /etc/protocols staat wordt herkend en kan gebruikt worden. Het bijzondere protocolsleutelwoord `tcp/udp` kan gebruikt worden om zowel voor TCP- als UDP-pakketten van toepassing te laten zijn. Het is toegevoegd voor het gemak om vrijwel gelijke regels te voorkomen.

BRON_ADR/BEST_ADR

Het sleutelwoord `all` is in feite hetzelfde als `from any to any` zonder overige parameters.

`from bron to bestemming`; de sleutelwoorden `from` en `to` worden gebruikt om te testen op IP-adressen. In regels moet *zowel* een bron- *als* bestemmings-IP-adres aangegeven worden. `any` is een bijzonder sleutelwoord dat van toepassing is op ieder IP-adres. Voorbeelden van gebruik: `from any to any` of `from 0.0.0.0/0 to any` of `from any to 0.0.0.0/0` of `from 0.0.0.0 to any` of `from any to 0.0.0.0`.

Het is vaak lastig om te komen tot een reeks IP-adressen die zich niet gemakkelijk laten uitdrukken met de gepunte numerieke vorm/ maskerlengte notatie. De port net-mgmt/ipcalc kan gebruikt worden om de berekeningen te vereenvoudigen. Aanvullende informatie is beschikbaar op de webpagina van het gereedschap: http://jodies.de/ipcalc.

POORT

Als in een regel op een poort wordt gecontroleerd, voor bron- of bestemmingspoort of beiden, dan is dat alleen van toepassing op TCP- en UDP-pakketten. Bij het maken van poortvergelijkingen kunnen zowel de dienstnamen uit /etc/services als een uit een natuurlijk getal bestaand poortnummer ingesteld worden. Als de poort onderdeel is van het `from` object dan wordt het vergeleken met het poortnummer van de bron en als het onderdeel is van het `to` object, dan wordt het vergeleken met het poortnummer van de bestemming. Het gebruik van het `to` object is in de moderne regels verplicht en neemt de vorm aan van `from any to any port = 80`.

Enkelvoudige poortvergelijkingen kunnen op verschillende manieren gedaan worden met een aantal verschillende operatoren. Er kunnen ook reeksen van poorten ingesteld worden.

poort "=" | "!=" | "<" | ">" | "⇐" | ">=" | "eq" | "ne" | "lt" | "gt" | "le" | "ge"

Reeksen van poorten worden met de volgende optie aangegeven: poort <> | ><



De volgende twee parameters die betrekking hebben op bron en bestemming, zijn verplicht in de moderne regels.

TCP_VLAG

Vlaggen zijn alleen beschikbaar voor het filteren van TCP. De letters staan voor de mogelijke vlaggen die bekeken kunnen worden in de kop van een TCP-pakket.

In de moderne regels wordt de optie `flags S` gebruikt om het verzoek tot het starten van een TCP sessie.

STATEFUL

`keep state` geeft aan dat in een regel met `pass` voor alle pakketten die van toepassing zijn stateful gefilterd moet worden.



Deze optie is voor moderne regels verplicht.

28.5.12. Stateful filteren

Met stateful filteren wordt verkeer benaderd als een uitwisseling van pakketten tussen twee kanten die een sessie zijn. Als het is ingeschakeld, dan maakt het `keep state` mechanisme dynamisch interne regels voor pakketten die in de sessie horen te volgen. Het kan bekijken of de karakteristieken van de sessie tussen verzender en ontvanger de juiste procedure volgen. Alle pakketten die niet passen in de sessie, worden automatisch geblokkeerd.

`keep state` staat ook ICMP-pakketten toe die gerelateerd zijn aan een TCP- of UDP-sessie. Dus als er een ICMP-type 3 code 4 komt in antwoord op websurfen, dat wordt toegestaan van binnen naar buiten door een `keep state` regel, dan wordt dat toegelaten. Pakketten waarvan IPF zeker is dat ze onderdeel zijn van de sessie worden toegelaten, zelfs als ze van een ander protocol zijn.

Wat er gebeurt: pakketten die naar buiten gaan op de interface die met Internet is verbonden worden eerst vergeleken met de dynamische staattabel. Als een pakket voldoet aan de verwachting van het volgende pakket in de sessie, dan mag het de firewall verlaten en wordt de toestand van de sessie in de dynamische toestandstabel bijgewerkt. Pakketten die niet bij een reeds actieve sessie horen, worden tegen de uitgaande regelverzameling gecontroleerd.

Pakketten die binnenkomen op de interface die met Internet is verbonden worden eerst vergeleken met de dynamische staattabel. Als een pakket voldoet aan de verwachting van het volgende pakket in de sessie, dan mag het de firewall verlaten en wordt de toestand van de sessie in de dynamische toestandstabel bijgewerkt. Pakketten die niet bij een reeds actieve sessie horen, worden vergeleken met de regelverzameling voor binnenkomend verkeer.

Als de sessie wordt beëindigd wordt het uit de dynamische staattabel verwijderd.

Met stateful filteren is het mogelijk om de focus te leggen op het blokkeren of toestaan van nieuwe sessies. Als een nieuwe sessie tot stand mag komen, dan worden alle volgende pakketten automatisch doorgelaten en al het vervalste verkeer wordt automatisch tegengehouden. Als een nieuwe sessie wordt geweigerd, dan wordt geen enkel pakket doorgelaten. Met stateful filteren zijn er uitgebreide mogelijkheden voor onderzoek om bescherming te bieden tegen de veelheid aan aanvallen die tegenwoordig door aanvallers worden uitgevoerd.

28.5.13. Voorbeeld van inclusieve regels

De onderstaande regels zijn een voorbeeld van hoe een erg veilige inclusieve firewall opgezet kan worden. Een inclusieve firewall staat alleen diensten toe die passen bij de `pass`-regels en blokkeert al het overige verkeer. Firewalls die bedoeld zijn om andere machines te beschermen, ook wel

"netwerk-firewalls" genoemd, dienen tenminste twee interfaces te hebben, die over het algemeen zijn ingesteld om de ene kant te vertrouwen (het LAN) maar niet de andere (het publieke Internet). Ook kan een firewall worden ingesteld om alleen het systeem te beschermen waarop het draait-dit wordt een "host-gebaseerde firewall" genoemd, en is in het bijzonder geschikt voor servers op een onvertrouwd netwerk.

Alle UNIX® systemen en dus ook FreeBSD zijn zo ontworpen dat ze voor interne communicatie de interface lo0 en IP adres `127.0.0.1` gebruiken. De firewall moet dit interne verkeer gewoon doorgang laten vinden.

Voor de interface die is verbonden met het publieke Internet worden regels gemaakt waarmee de toegang voor uitgaande en binnenkomende verbindingen worden geautoriseerd en beheerst. Dit kan de PPP-interface tun0 zijn of de netwerkkaart die is verbonden met een xDSL- of kabelmodem.

In gevallen dat er één of meer netwerkkaarten zijn aangesloten op private netwerksegmenten kunnen er regels op de firewall nodig zijn om pakketten die van die LAN-interfaces afkomen vrije doorgang te geven naar elkaar en/of naar buiten (het Internet).

De regels worden opgedeeld in drie onderdelen: eerst de vertrouwde interfaces, dan het publieke uitgaande interface en als laatste het onvertrouwde publieke binnenkomende interfaces.

In iedere sectie moeten zo staan dat de regels die het meest gebruikt worden vóór de regels die minder vaak gebruikt worden staan. De laatste regel van een onderdeel geeft aan dat al het overige verkeer op die interface in die richting geblokkeerd en gelogd moet worden.

In het onderdeel Uitgaand staan alleen regels met `pass` die parameters bevatten om uniek individuele diensten identificeren die het publieke Internet mogen benaderen. Bij al die regels staan de opties `quick`, `on`, `proto`, `port` en `keep state` aan. De regels met `proto tcp` maken ook gebruik van de optie `flag` om te bekijken of het een pakket betreft voor het opzetten van een sessie om de stateful functionaliteit aan te sturen.

In het onderdeel Inkomend staan eerst alle regels voor het blokkeren van ongewenste pakketten, om twee redenen. Als eerste kan het zo zijn dat kwaadaardige pakketten gedeeltelijk overeenkomen met legitiem verkeer. Deze pakketten moeten worden weggegooid in plaats van binnengelaten te worden, gebaseerd op hun gedeeltelijke match met de `allow`-regels. De tweede reden is dat bekende en oninteressante verwerpingen stil geblokkeerd kunnen worden in plaats van gevangen en gelogd te worden door de laatste regels in de sectie. De laatste regel in elke sectie blokkeert en logt alle pakketten en kan worden gebruikt voor het wettelijke bewijs nodig om degenen die uw systeem aanvallen aan te klagen.

Waar ook gezorgd voor moet worden is dat al het verkeer dat wordt geweigerd geen antwoord verstuurd. Ongeldige pakketten dienen gewoon te verdwijnen. Zo weet een aanvaller niet of een pakket het doelsysteem wel heeft bereikt. Zo kan een aanvaller geen informatie verzamelen over een systeem: hoe minder informatie er over een systeem beschikbaar is, hoe meer tijd iemand erin moet steken voordat er iets slechts gedaan kan worden. Regels die een optie `log first` bevatten, zullen alleen de eerste keer dat de gebeurtenis voorkomt de gebeurtenis loggen. Deze optie is opgenomen in de voorbeeldregel `nmap OS fingerpint`. Het gereedschap security/nmap wordt vaak door aanvallers gebruikt om het besturingssysteem van uw server proberen te achterhalen.

We raden aan om telkens als er logmeldingen van een regel met de optie `log first` komen, `ipfstat`

`-hio` uit te voeren om te bekijken hoe vaak de regel van toepassing is geweest. Een groot aantal overeenkomsten geeft gewoonlijk aan dat de firewall overspoeld wordt, met andere woorden aangevallen wordt.

Het bestand /etc/services kan gebruikt worden om onbekende poortnummers op te zoeken. Ook kan http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers worden bezocht en het poortnummer worden opgezocht om het doel van een bepaalde poort uit te vinden.

Op de volgende link worden poortnummers van Trojans beschreven: http://www.sans.org/security-resources/idfaq/oddports.php.

De onderstaande set regels is een complete en erg veilige `inclusieve` set met regels voor een firewall die is getest op productiesystemen. Deze set met regels is eenvoudig aan te passen voor uw eigen systeem. Maak gewoon commentaar van elke `pass`-regel voor een dienst die niet gewenst is.

Logberichten die niet gewenst zijn, zijn uit te sluiten door een `block`-regel toe te voegen in het begin van het onderdeel Inkomend.

Voor de onderstaande regels dient de dc0 interfacenaam in iedere regel vervangen te worden door de echte interfacenaam van de netwerkkaart in het systeem die met het publieke Internet is verbonden. Voor gebruikers van PPP zou dat tun0 zijn.

Dit zou de inhoud van /etc/ipf.rules kunnen zijn:

```
######################################################################
# Geen beperkingen op de interface aan de LAN kant.
# Niet nodig als er geen LAN is.
######################################################################

#pass out quick on xl0 all
#pass in quick on xl0 all

######################################################################
# Geen beperkingen op de loopback interface
######################################################################
pass in quick on lo0 all
pass out quick on lo0 all

######################################################################
# Interface aan het publieke Internet (onderdeel Uitgaand).
# Inspecteer verzoeken om een sessie te starten van achter de
# firewall op het private netwerk of vanaf deze gateway-server
# naar het publieke Internet.
######################################################################

# Geef toegang tot de DNS server van de ISP.
# xxx moet het IP adres van de DNS van de ISP zijn.
# Dupliceer deze regels als een ISP meerdere DNS servers heeft.
# Haal het IP adres evt. uit /etc/resolv.conf.
pass out quick on dc0 proto tcp from any to xxx port = 53 flags S keep state
```

```
pass out quick on dc0 proto udp from any to xxx port = 53 keep state

# Geef toegang tot de DHCP server van de ISP voor kabel- en
# xDSL-netwerken. Deze regel is niet nodig als gebruik gemaakt worden
# van PPP naar het publieke Internet. In dat geval kan de hele groep
# verwijderd worden. Gebruik de volgende regel en controleer het
# logboek voor het IP adres. Wijzig dan het IP adres in de regel
# commentaar hieronder en verwijder de eerste regel.
pass out log quick on dc0 proto udp from any to any port = 67 keep state
#pass out quick on dc0 proto udp from any to z.z.z.z port = 67 keep state

# Sta niet beveiligd www verkeer toe.
pass out quick on dc0 proto tcp from any to any port = 80 flags S keep state

# Sta beveiligd www verkeer over TLS SSL toe.
pass out quick on dc0 proto tcp from any to any port = 443 flags S keep state

# Sta het verzenden en ontvangen van e-mail toe.
pass out quick on dc0 proto tcp from any to any port = 110 flags S keep state
pass out quick on dc0 proto tcp from any to any port = 25 flags S keep state

# Sta Time toe.
pass out quick on dc0 proto tcp from any to any port = 37 flags S keep state

# Sta uitgaand NNTP nieuws toe.
pass out quick on dc0 proto tcp from any to any port = 119 flags S keep state

# Sta uitgaande lokale niet beveiligde FTP (ook van LAN-gebruikers) toe
# (zowel passieve als actieve modes).  Deze functie maakt gebruik van
# de in IP-NAT ingebouwde FTP-proxy die in het bestand met NAT-regels
# staat om dit in één regel te laten werken. Als er met
# pkg_add pakketten toegevoegd moeten kunnen worden op een systeem, dan
# is deze regel nodig.
pass out quick on dc0 proto tcp from any to any port = 21 flags S keep state

# Sta uitgaande SSH/SFTP/SCP toe (vervangingen van telnet/rlogin/FTP)
# Deze functie maakt gebruik van SSH (secure shell)
pass out quick on dc0 proto tcp from any to any port = 22 flags S keep state

# Sta uitgaande niet beveiligde telnet toe.
pass out quick on dc0 proto tcp from any to any port = 23 flags S keep state

# Sta de FreeBSD CVSUP-functie toe.
pass out quick on dc0 proto tcp from any to any port = 5999 flags S keep state

# Sta ping toe naar het publieke Internet.
pass out quick on dc0 proto icmp from any to any icmp-type 8 keep state

# Sta whois toe vanaf het LAN naar het publieke Internet.
pass out quick on dc0 proto tcp from any to any port = 43 flags S keep state
```

```
# Blokkeer en log het eerste voorkomen van al het andere dat probeert
# buiten te komen. Deze regel implementeert de standaard-blokkade.
block out log first quick on dc0 all

######################################################################
# Interface aan het publieke Internet (onderdeel Inkomend).
# Inspecteert pakketten die van het publieke Internet komen
# met als bestemming deze gateway-server of het private netwerk.
######################################################################

# Blokkeer al het verkeer voor niet-routeerbare of gereserveerde
# adresreeksen.
block in quick on dc0 from 192.168.0.0/16 to any    #RFC 1918 privaat IP
block in quick on dc0 from 172.16.0.0/12 to any     #RFC 1918 privaat IP
block in quick on dc0 from 10.0.0.0/8 to any        #RFC 1918 privaat IP
block in quick on dc0 from 127.0.0.0/8 to any       #loopback
block in quick on dc0 from 0.0.0.0/8 to any         #loopback
block in quick on dc0 from 169.254.0.0/16 to any    #DHCP auto-config
block in quick on dc0 from 192.0.2.0/24 to any      #gereserveerd voor documentatie
block in quick on dc0 from 204.152.64.0/23 to any   #Sun cluster interconnect
block in quick on dc0 from 224.0.0.0/3 to any       #Klasse D & E multicast

##### Blokkeer wat vervelende dingen #############
# die niet in de logboeken moeten komen.

# Blokkeer fragmenten.
block in quick on dc0 all with frags

# Block korte TCP pakketten.
block in quick on dc0 proto tcp all with short

# Blokkeer source gerouteerde pakketten.
block in quick on dc0 all with opt lsrr
block in quick on dc0 all with opt ssrr

# Blokkeer pogingen voor nmap OS fingerprint.
# Blokkeer het eerste voorkomen ervan voor de IP-adressen
block in log first quick on dc0 proto tcp from any to any flags FUP

# Blokkeer alles met speciale opties.
block in quick on dc0 all with ipopts

# Blokkeer publieke pings.
block in quick on dc0 proto icmp all icmp-type 8

# Blokkeer ident.
block in quick on dc0 proto tcp from any to any port = 113

# Blokkeer alle Netbios diensten. 137=naam, 138=datagram, 139=sessie.
# Netbios is de Windows(R) bestandsdeeldienst.
# Blokkeer Windows(R) hosts2 name server verzoeken 81.
```

```
block in log first quick on dc0 proto tcp/udp from any to any port = 137
block in log first quick on dc0 proto tcp/udp from any to any port = 138
block in log first quick on dc0 proto tcp/udp from any to any port = 139
block in log first quick on dc0 proto tcp/udp from any to any port = 81

# Sta inkomend verkeer toe van de DHCP server van de ISP. Deze regel
# moet het IP adres van de DHCP server van de ISP bevatten omdat die
# de enige toegestane bron van dit type pakketten moet zijn. Alleen
# van belang voor kabel en xDSL instellingen. Deze regel is niet nodig
# voor PPP verbindingen naar het publieke Internet.  Dit is hetzelfde
# IP adres dat in het Uitgaande onderdeel is opgezocht.
pass in quick on dc0 proto udp from z.z.z.z to any port = 68 keep state

# Sta inkomend webverkeer toe omdat er een Apache server draait.
pass in quick on dc0 proto tcp from any to any port = 80 flags S keep state

# Sta niet beveiligde telnet sessie toe vanaf het publieke Internet.
# Dit heeft het label niet veilig omdat gebruikersnaam en
# wachtwoord als platte tekst over Internet gaan. Als er geen telnet
# server draait, hoeft deze regel niet actief te zijn.
#pass in quick on dc0 proto tcp from any to any port = 23 flags S keep state

# Sta beveiligde FTP, telnet en SCP toe vanaf Internet.
# Deze functie gebruikt SSH (secure shell).
pass in quick on dc0 proto tcp from any to any port = 22 flags S keep state

# Blokkeer en log het eerste voorkomen van al het andere dat probeert
# binnen te komen. Het loggen van alleen het eerste voorkomen stopt
# een ontzegging van dienst aanval die gericht is op het laten
# vollopen van de partitie waarop de logboeken staan. Deze regel implementeert
# de standaard blokkade.
block in log first quick on dc0 all
##################### Einde van de regels ###################################
```

28.5.14. NAT

NAT staat voor *Network Address Translation* (netwerkadres vertaling). In Linux® heet dit IP Masquerading. Een van de vele mogelijkheden die IPF NAT kan bieden is het delen van één IP adres op het publieke Internet met een LAN achter een firewall.

De vraag zou kunnen rijzen waarom iemand dat zou willen. ISP's wijzen normaliter namelijk dynamisch een IP adres toe aan hun niet-commerciële gebruikers. Dynamisch betekent hier dat het IP-adres iedere dat er wordt ingebeld of dat het kabel- of xDSL-modem uit- en aangeschakeld wordt anders kan zijn. Dit dynamische IP-adres wordt gebruikt om uw systeem op het publieke Internet te identificeren.

Stel dat er vijf PC's in een huis staan en iedere computer in dat huis heeft toegang tot Internet nodig. Dan zouden er bij een ISP vijf individuele accounts moeten zijn en vijf telefoonlijnen om dat te realiseren.

Met NAT is er maar één account bij een ISP nodig. De andere vier PC's moeten met kabels op een switch worden aangesloten waarop ook een FreeBSD systeem is aangesloten dat binnen uw LAN als gateway gaat opereren. NAT zal automatisch de private LAN IP adressen van alle PC's vertalen naar een enkel publiek IP-adres als de pakketten de firewall naar het Internet verlaten.

Er is een speciale reeks van IP-adressen gereserveerd voor NAT op private LANs. Volgens RFC 1918 kunnen de volgende reeksen IP-adressen gebruikt worden op private netwerken die nooit direct op het publieke Internet gerouteerd worden.

Eerste IP	-	Laatste IP
`10.0.0.0`	-	`10.255.255.255`
`172.16.0.0`	-	`172.31.255.255`
`192.168.0.0`	-	`192.168.255.255`

28.5.15. IPNAT

NAT regels worden geladen met `ipnat`. De NAT regels worden vaak opgeslagen in /etc/ipnat.rules. Meer details staan in ipnat(8).

Bij het maken van wijzigingen aan de NAT-regels nadat NAT gestart is, wordt aangeraden de wijziging aan het bestand met regels te maken en daarna `ipnat -CF` te gebruiken om alle actieve NAT-regels te wissen. Daarna kunnen de regels uit het bestand weer als volgt geladen worden:

```
# ipnat -CF -f /etc/ipnat.rules
```

Gebruiksgegevens over NAT kunnen getoond worden met:

```
# ipnat -s
```

De huidige inhoud van de NAT tabellen kan getoond worden met:

```
# ipnat -l
```

Met het volgende commando kan de uitgebreide rapportage worden ingeschakeld en dan wordt informatie over het verwerken van verkeer en de actieve regels getoond:

```
# ipnat -v
```

28.5.16. IPNAT regels

NAT regels zijn erg flexibel en er kunnen veel dingen mee gedaan worden om behoeften van bedrijven en thuisgebruikers in te vullen.

De syntaxis van de regels die hier wordt toegelicht is vereenvoudigd om te passen bij een niet-

commerciële omgeving. De complete syntaxis is na te lezen in ipnat(5).

De syntaxis voor een NAT regel ziet er ongeveer als volgt uit:

```
map IF LAN_IP_REEKS -> PUBLIEK_ADRES
```

De regel begint met het sleutelwoord `map`.

IF dient vervangen te worden door de aanduiding van de externe interface.

LAN_IP_REEKS is de reeks die clients op een LAN gebruiken, meestal iets van `192.168.1.0/24`.

PUBLIEK_ADRES kan het publieke IP adres zijn of een speciaal sleutelwoord `0.32`, wat betekent dat het IP adres van *IF* gebruikt moet worden.

28.5.17. Hoe NAT werkt

Een pakket komt vanaf het LAN aan bij de firewall en heeft een publieke bestemming. Het wordt verwerkt door de filterregels voor inkomend verkeer en daarna krijgt NAT de kans zijn regels op het pakket toe te passen. De regels worden van boven naar beneden toegepast en de eerste regel die van toepassing is wint. NAT controleert voor alle regels het pakket op interfacenaam en bron IP adres. Als de interfacenaam van een pakket past bij een NAT regel dan wordt het bron IP adres van dat pakket gecontroleerd, dat is dus een IP adres op het private LAN, om te bekijken of het valt in de reeks die is opgegeven aan de linkerkant van een NAT regel. Als ook dat klopt, dan wordt het bron IP adres van het pakket vervangen ("rewritten") door een publiek IP adres dat verkregen kan zijn met het sleutelwoord `0.32`. NAT werkt dan zijn interne NAT tabel bij, zodat als er een pakket uit die sessie terugkomt van het publieke Internet, dat pakket weer gepast kan worden bij het originele private IP adres en door de firewallregels gefilterd kan worden om daarna, als dat mag, naar een client gestuurd te worden.

28.5.18. IPNAT inschakelen

Voor IPNAT zijn de onderstaande instellingen in /etc/rc.conf beschikbaar.

Om verkeer tussen interfaces te kunnen routeren:

```
gateway_enable="YES"
```

Om IPNAT automatisch te starten:

```
ipnat_enable="YES"
```

Om aan te geven waar de IPNAT regels staan:

```
ipnat_rules="/etc/ipnat.rules"
```

28.5.19. NAT voor een groot LAN

Voor netwerken met grote aantallen PC's of netwerken met meerdere LAN's kan het een probleem worden om al die private IP adressen met één enkel publiek IP adres te vervangen, omdat vaak dezelfde poortnummers gebruikt worden. Er zijn twee manieren om dit probleem op te lossen.

Aangeven welke poorten te gebruiken

Een normale regel voor NAT ziet er als volgt uit:

```
map dc0 192.168.1.0/24 -> 0.32
```

Met de bovenstaande regel blijft de bronpoort ongewijzigd als het pakket door IPNAT gaat. Door gebruik te maken van het sleutelwoord `portmap` kan IPNAT ingesteld worden om alleen bronpoorten in de aangegeven reeks te gebruiken. Zo stelt de onderstaande regel in dat IPNAT de bronpoort aanpast naar een poortnummer dat in de aangegeven reeks valt:

```
map dc0 192.168.1.0/24 -> 0.32 portmap tcp/udp 20000:60000
```

Het kan nog eenvoudiger door gebruik te maken van het sleutelwoord `auto` zodat IPNAT zelf bepaalt welke poorten gebruikt kunnen worden:

```
map dc0 192.168.1.0/24 -> 0.32 portmap tcp/udp auto
```

Meerdere publieke adressen gebruiken

In grote netwerken komt er een moment waarop er gewoon te veel adressen zijn om te bedienen met één IP adres. Als er een blok van publiekelijke IP adressen beschikbaar is, dan kunnen deze adressen gebruikt worden in een "poel", welke door IPNAT gebruikt kan worden om één van de adressen te gebruiken als uitgaand adres.

Bijvoorbeeld om alle pakketten te verstoppen achter één een enkel IP adres:

```
map dc0 192.168.1.0/24 -> 204.134.75.1
```

Een reeks van publiekelijke IP adressen kan gespecificeerd worden met een netwerkmasker:

```
map dc0 192.168.1.0/24 -> 204.134.75.1-10
```

of door gebruik van de CIDR notatie:

```
map dc0 192.168.1.0/24 -> 204.134.75.0/24
```

28.5.20. Poorten omleiden

Het is erg gebruikelijk om een webserver, mailserver, database server en DNS server op verschillende computers op een LAN te draaien. Het uitgaande verkeer van die servers kan dan met NAT afgehandeld worden, maar er moet ook ingesteld worden dat inkomend verkeer bij de juiste computer terecht komt. IPNAT gebruikt daarvoor de opties in NAT waarmee verkeer omgeleid kan worden. Als bijvoorbeeld een webserver op het LAN-adres `10.0.10.25` draait en het enkele publieke IP adres zou `20.20.20.5` zijn, dan zou de regel er als volgt uit zien:

```
rdr dc0 20.20.20.5/32 port 80 -> 10.0.10.25 port 80
```

of:

```
rdr dc0 0.0.0.0/32 port 80 -> 10.0.10.25 port 80
```

Voor een DNS server op een LAN die ook vanuit Internet bereikbaar met zijn en die draait op `10.0.10.33` zou de regel er als volgt uit zien:

```
rdr dc0 20.20.20.5/32 port 53 -> 10.0.10.33 port 53 udp
```

28.5.21. FTP en NAT

FTP is dinosaurus uit het tijdperk van voor Internet was zoals het nu is, toen onderzoeksinstellingen met elkaar verbonden waren via huurlijnen en FTP de aangewezen methode was om bestanden met elkaar uit te wisselen. Maar bij het gebruik van FTP worden gebruikersnaam en wachtwoord als platte tekst verzonden en het protocol is nooit aangepast. FTP is er in twee smaken: actief en passief. Het verschil zit 'm in hoe het datakanaal wordt opgezet. De passieve variant is veiliger voor een gebruiker omdat bij deze variant beide communicatiekanalen door de cliënt zelf worden opgezet. Op de volgende pagina zijn details over FTP na te lezen: http://www.slacksite.com/other/ftp.html.

IPNAT-regels

IPNAT heeft een speciale FTP-proxy ingebouwd die kan worden ingeschakeld met een NAT-`map` -regel. Die kan al het uitgaande verkeer monitoren wat betreft opstartverzoeken voor sessies voor actieve en passieve FTP en dynamisch tijdelijke filterregels maken die alleen het poortnummer dat echt in gebruik is voor het datakanaal doorlaten. Hiermee wordt een veiligheidsrisico dat normaal gepaard gaat met FTP, namelijk het toestaan van grote reeksen hoge poortnummers, weggenomen.

De volgende regel handelt al het FTP verkeer van het LAN af:

```
map dc0 10.0.10.0/29 -> 0/32 proxy port 21 ftp/tcp
```

De regel hieronder handelt het FTP verkeer van de gateway zelf af:

```
map dc0 0.0.0.0/0 -> 0/32 proxy port 21 ftp/tcp
```

Deze laatste regel handelt al het niet-FTP verkeer voor het LAN af:

```
map dc0 10.0.10.0/29 -> 0/32
```

De FTP-afbeeldregel hoort voor de normale regels te staan. Alle pakketten worden als eerste vergeleken met de eerste regel en zo verder. Eerst wordt gekeken over de interfacenaam overeenkomt, daarna het bron IP adres van het LAN en dan of het een FTP pakket is. Als dat allemaal klopt, dan maakt de speciale FTP proxy een tijdelijke filterregel die de pakketten uit de FTP sessie naar binnen en buiten doorlaat en ook NAT toepast op de FTP pakketten. Alle pakketten van het LAN die niet van het protocoltype FTP zijn en dus niet bij de eerste regel passen, worden tegen de derde regel gehouden die van toepassing is vanwege de interface en bron IP adres, zodat er dan NAT op toegepast wordt.

IPNAT FTP filterregels

Als de NAT-FTP-proxy wordt gebruikt is er maar één filterregel voor FTP nodig. Zonder de FTP-proxy zouden er drie regels nodig zijn:

```
# Sta LAN client toe te FTP-en naar Internet
# Actieve en passieve modes
pass out quick on rl0 proto tcp from any to any port = 21 flags S keep state

# Sta opzetten van het datakanaal voor passieve mode toe voor hoge poorten
pass out quick on rl0 proto tcp from any to any port > 1024 flags S keep state

# Laat het datakanaal van de FTP server binnen voor actieve mode
pass in quick on rl0 proto tcp from any to any port = 20 flags S keep state
```

28.6. IPFW

IPFIREWALL (IPFW) is een firewall die binnen FreeBSD wordt ontwikkeld en onderhouden door vrijwillige leden van de staf. Het maakt gebruik van verouderde staatloze regels en een verouderde techniek om te realiseren wat eenvoudige stateful logica zou kunnen heten.

De verzameling voorbeeldregels van IPFW (die in /etc/rc.firewall en /etc/rc.firewall6 staan) uit de standaard FreeBSD-installatie is redelijk eenvoudig en niet voorbereid om zonder wijzigingen gebruikt te worden. Het voorbeeld maakt geen gebruik van stateful filteren, wat een voordeel is in de meeste situaties. Daarom worden deze regels niet als basis gebruikt in dit onderdeel.

De staatloze syntaxis van IPFW is krachtig door de technisch geavanceerde mogelijkheden van de regelsyntaxis die de kennis van de gemiddelde gebruiker van firewalls ver overstijgt. IPFW is gericht op de professionele gebruiker of de gevorderde thuisgebruiker die hoge eisen stelt aan de wijze waarop er met pakketten wordt omgegaan. Voordat de kracht van de IPFW regels echt ingezet kan worden, moet de gebruiker veel weten over de verschillende protocollen en de wijze waarop

pakketten in elkaar zitten. Het tot op dat niveau behandelen van stof valt buiten de doelstellingen van dit Handboek.

IPFW bestaat uit zeven componenten: de verwerkingseenheid voor de firewallregels, verantwoording, loggen, regels met `divert` (omleiden) waarmee NAT gebruikt kan worden en de speciale gevorderde mogelijkheden voor bandbreedtebeheer met DUMMYNET, de `fwd rule` forward-mogelijkheid, de bridge-mogelijkheden en de ipstealth-mogelijkheden. IPFW ondersteunt zowel IPv4 als IPv6.

28.6.1. IPFW inschakelen

IPFW zit bij de basisinstallatie van FreeBSD als een losse tijdens runtime laadbare module. Het systeem laadt de kernelmodule dynamisch als in rc.conf de regel `firewall_enable="YES"` staat. IPFW hoeft niet in de FreeBSD kernel gecompileerd te worden.

Na het rebooten van een systeem met `firewall_enable="YES"` in rc.conf is het volgende bericht op het scherm te zien tijdens het booten:

```
ipfw2 initialized, divert disabled, rule-based forwarding disabled, default to deny,
logging disabled
```

In de laadbare module zit de mogelijkheid om te loggen gecompileerd. Er is een knop in /etc/sysctl.conf om loggen aan te zetten en de uitgebreide loglimiet in te stellen. Door deze regels toe te voegen, staat loggen aan bij toekomstige herstarts:

```
net.inet.ip.fw.verbose=1
net.inet.ip.fw.verbose_limit=5
```

28.6.2. Kernelopties

Het is niet verplicht om IPFW in te schakelen door het mee te compileren in de FreeBSD kernel. Dit wordt alleen beschreven als achtergrondinformatie.

```
options    IPFIREWALL
```

Met `IPFIREWALL` wordt IPFW ingeschakeld als deel van de kernel.

```
options    IPFIREWALL_VERBOSE
```

Met `IPFIREWALL_VERBOSE` wordt het loggen van pakketten die worden verwerkt met IPFW mogelijk die het sleutelwoord `log` in een regel hebben staan.

```
options    IPFIREWALL_VERBOSE_LIMIT=5
```

Limiteert het aantal pakketten dat per regel wordt gelogd via syslogd(8). Deze optie kan gebruikt worden in vijandige omgevingen waar de activiteit van een firewall gelogd moet worden. Hierdoor kan een mogelijke ontzegging van dienst aanval door het vol laten lopen van syslog voorkomen worden.

```
options    IPFIREWALL_DEFAULT_TO_ACCEPT
```

Met `IPFIREWALL_DEFAULT_TO_ACCEPT` wordt standaard alles door de firewall doorgelaten. Dit wordt aangeraden als iemand voor het eerst een firewall opzet.

```
options    IPDIVERT
```

Met `IPDIVERT` wordt de NAT functionaliteit ingeschakeld.



De firewall zal alle binnenkomende en uitgaande pakketten blokkeren als de kerneloptie `IPFIREWALL_DEFAULT_TO_ACCEPT` of een regel om deze verbindingen expliciet toe te staan ontbreekt.

28.6.3. /etc/rc.conf opties

Start de firewall:

```
firewall_enable="YES"
```

Om één van de standaard firewall types die geleverd wordt door FreeBSD te selecteren, lees /etc/rc.firewall, maak een selectie en plaats het in de volgende regel:

```
firewall_type="open"
```

Beschikbare waardes voor deze instelling zijn:

- `open` - laat al het verkeer door.
- `client` - beschermt alleen deze machine.
- `simple` - beschermt het hele netwerk.
- `closed` - blokkeert alle IP-verkeer, behalve voor lokaal verkeer.
- `UNKNOWN` - voorkomt het laden de firewall-regels.
- bestandsnaam - absoluut pad naar een bestand dat firewall-regels bevat.

Het is mogelijk om twee verschillende manieren te gebruiken voor speciaal gemaakte regels voor de ipfw firewall. één daarvan is door het zetten van de `firewall_type` variabele naar een absoluut pad van een bestand, welke *firewall-regels* bevat, zonder enige specifieke opties voor ipfw(8). Het volgende is een eenvoudig voorbeeld van een bestand met regelverzamelingen dat al het inkomend en uitgaand verkeer blokkeert:

```
add deny in
add deny out
```

Aan de andere kant is het mogelijk om de variabele `firewall_script` in te stellen op een absoluut pad van een uitvoerbaar script, welke inclusief `ipfw` commando's uitgevoerd wordt tijdens het opstarten van het systeem. Een geldig script met regels dat gelijkwaardig is aan het bestand met regels hierboven, zou het volgende zijn:

```
#!/bin/sh

ipfw -q flush

ipfw add deny in
ipfw add deny out
```



Als `firewall_type` is gezet naar `client` of `simple` moeten de standaard regels die gevonden kunnen worden in /etc/rc.firewall gecontroleerd worden om te zien of deze configuratie voldoet voor de machine. Let ook op dat alle voorbeelden die gebruikt zijn in dit hoofdstuk ervan uitgaan dat de `firewall_script` variabele gezet is naar /etc/ipfw.rules.

Om loggen in te schakelen:

```
firewall_logging="YES"
```



Het enige dat de variabele `firewall_logging` doet is de sysctl variabele `net.inet.ip.fw.verbose` op de waarde `1` zetten (zie IPFW inschakelen). Er is geen variabele in rc.conf om logboeklimieten in te stellen, maar dat kan ingesteld worden via een sysctl variabele, handmatig of via /etc/sysctl.conf:

```
net.inet.ip.fw.verbose_limit=5
```

Als de machine in kwestie een gateway is, dus Network Address Translation (NAT) diensten levert via natd(8), dan staat in Network Address Translation meer informatie over de benodigde instellingen voor /etc/rc.conf.

28.6.4. Het commando IPFW

Gewoonlijk wordt `ipfw` gebruikt om met de hand enkelvoudige regels toe te voegen of te verwijderen als IPFW actief is. Het probleem met deze methode is dat, als het systeem wordt uitgezet alle regels die gewijzigd of verwijderd zijn verloren gaan. Door alle regels in een bestand op te nemen dat bij het booten wordt geladen of door het bestand waarin de wijzigingen zijn gemaakt als een machine draait te laden bestaat die probleem niet.

Met `ipfw` kunnen de actieve regels van de firewall op het scherm getoond worden. De verantwoordingsmogelijkeden van ipfw(8) maken dynamisch tellers aan voor iedere regel en houden die bij voor alle pakketten die van toepassing zijn op die regel. Tijdens het testen van een regel is het afbeelden van de regel met zijn teller een van de manieren om te bepalen of de regel werkt.

Om alle regels in volgorde te tonen:

```
# ipfw list
```

Om alle regels te tonen met de tijd waarop deze voor het laatst van toepassing was:

```
# ipfw -t list
```

Het volgende commando kan gebruikt worden om de verantwoordingsinformatie, pakkettellers en de regel zelf te tonen. De eerste kolom is het regelnummer met daarachter het aantal keren dat de regel van toepassing was voor inkomend verkeer, gevolgd door het aantal keren dat de regel van toepassing was voor uitgaand verkeer. Als laatste wordt de regel zelf getoond:

```
# ipfw -a list
```

Ook kunnen onder de statische regels de dynamische regels getoond worden:

```
# ipfw -d list
```

En de dynamische regels die verlopen zijn:

```
# ipfw -d -e list
```

De tellers op nul gesteld worden:

```
# ipfw zero
```

Alleen de tellers voor regel met nummer *NUM* op nul stellen:

```
# ipfw zero NUM
```

28.6.5. Sets van IPFW regels

Een verzameling regels is een groep IPFW-regels die is gemaakt om pakketten toe te staan of te blokkeren op basis van de inhoud van dat pakket. De bi-directionele uitwisseling van pakketten tussen hosts bestaat uit een gesprek dat een sessie heet. De verzameling van firewallregels

beoordeelt zowel de pakketten die aankomen van de host op het publieke Internet als de pakketten die op het systeem ontstaan als antwoord daarop. Iedere TCP/IP-dienst als telnet, www, mail, etc, heeft zijn eigen protocol en bevoorrechte (luister)poort. Pakketten bestemd voor een specifieke poort verlaten het bronadres via een onbevoorrechte (hogere) poort en doelen op de specifieke dienstpoort op het bestemmingsadres. Alle bovenstaande parameters (poorten en adressen) kunnen gebruikt worden als selectiecriteria om regels aan te maken die diensten doorlaten of blokkeren.

Als een pakket de firewall binnenkomt wordt het vergeleken met de eerste regel in de set regels en zo gaat dat voor iedere regel vanaf boven tot beneden. Als een regel van toepassing is op een pakket, dan wordt het actieveld van de regel uitgevoerd. Dit wordt de "de eerst passende regel wint" zoekmethode genoemd. Als een pakket bij geen enkele regel past, dan wordt de verplichte standaardregel 65535 van IPFW toegepast, die alle pakketten weigert zonder een antwoord terug te sturen naar de verzender.



Het zoeken gaat door na regels met `count`, `skipto` en `tee`.

De instructies in dit onderdeel zijn gebaseerd op regels die gebruik maken van de stateful opties `keep state`, `limit`, `in`, `out` en `via`. Dit is het raamwerk waarmee een set van inclusieve firewallregels wordt samengesteld.



Wees voorzichtig tijdens het werken met firewall-regels, het is gemakkelijk om uzelf uit te sluiten.

Regelsyntaxis

De regelsyntaxis zoals hier toegelicht is vereenvoudigd door alleen te tonen wat nodig is om een standaard inclusieve set met firewallregels te maken. De complete beschrijving van alle mogelijkheden staat in ipfw(8).

Regels bevatten sleutelwoorden die in een bepaalde volgorde van links naar rechts op een regel horen te staan. Sleutelwoorden worden vet weergegeven. Sommige sleutelwoorden hebben subopties die zelf ook weer sleutelwoorden hebben die ook weer subopties kunnen hebben.

Het karakter # wordt gebruikt om het begin van een opmerking te markeren en kan zowel op een eigen regel als achter een firewallregel staan. Lege regels worden genegeerd.

CMD REGEL_NUMMER ACTIE LOGGEN SELECTIE STATEFUL

CMD

Iedere regel moet beginnen met add om hem toe te voegen aan de tabel met regels.

REGEL_NUMMER

Elke regel is geassocieerd met een regel_nummer van 1 tot en met 65535.

ACTIE

Bij een regel kunnen één of meer acties horen die worden uitgevoerd als een regel geldt voor een

pakket.

allow | accept | pass | permit

Deze opties betekenen allemaal hetzelfde: als de regel geldt voor een pakket, laat dat pakket dan door en stop met het zoeken naar geldende regels.

check-state

Vergelijkt het pakket met de tabel met dynamische regels. Als het erin staat, dan wordt de actie van de dynamisch door deze regel gemaakte regel uitgevoerd. Anders wordt er verder gezocht door de regels. Een regel met check-state heeft geen selectiecriteria. Als er geen regel met check-state in de set met regels staat, dan wordt de tabel met dynamische regels bij het eerste voorkomen van keep-state of limit gecontroleerd.

deny | drop

Deze opties betekenen hetzelfde: als de regel geldt voor een pakket, blokkeer dat pakket dan en stop met het zoeken naar geldende regels.

Loggen

log of logamount

Als een regel met het sleutelwoord `log` van toepassing is op een pakket, dan wordt er een bericht naar syslogd(8) geschreven met de faciliteitsnaam SECURITY. Er wordt alleen een bericht geschreven als het aantal voor die regel gelogde pakketten niet groter is dan de instelling van de parameter logamount. Als er geen `logamount` is ingesteld, dan wordt de limiet uit de sysctl(8) variabele `net.inet.ip.fw.verbose_limit` gehaald. In beide gevallen bestaat er in het geval de waarde nul is geen limiet. Als de limiet is bereikt, dan kan het loggen weer ingeschakeld worden door de teller voor het loggen weer op nul te zetten voor die regel met `ipfw reset log`.



Er wordt gelogd als een pakket zeker past bij een regel, maar voordat de actie (bijvoorbeeld accept of `deny`) op een pakket wordt toegepast. Uiteindelijk bepaalt de gebruiker zelf voor welke regels loggen wordt ingeschakeld.

Selectie

De sleutelwoorden in deze paragraaf beschrijven de attributen van een pakket die gecontroleerd worden bij het bepalen of een regel wel of niet op een pakket van toepassing is. De attributen waarop gecontroleerd kan worden moeten in de beschreven volgorde gebruikt worden.

udp | tcp | icmp

Naast de hierboven aangegeven protocollen kunnen alle in /etc/protocols beschreven protocollen gebruikt worden. De waarde die wordt opgegeven is het protocol dat van toepassing moet zijn. Dit attribuut is verpicht.

from bron to best

De sleutelwoorden `from` en `to` worden gebruikt om te bekijken of een regel van toepassing is op IP-

adressen. Een regel moet *zowel* bron- als bestemmingsadressen bevatten. `any` is een bijzonder sleutelwoord dat van toepassing is op alle IP-adressen. `me` is een bijzonder sleutelwoord dat van toepassing is op alle IP-adressen die ingesteld zijn op interfaces van een FreeBSD systeem om de PC waarop de firewall draait te vertegenwoordigen (deze machine). Zo kan dit onderdeel bijvoorbeeld de volgende vormen aannemen: `from me to any`, `from any to me`, `from 0.0.0.0/0 to any`, `from any to 0.0.0.0/0`, `from 0.0.0.0 to any`, `from any to 0.0.0.0` of `from me to 0.0.0.0`. IP-adressen mogen ingevoerd worden in de vorm numeriek, door punten gescheiden adres/maskerlengte (CIDR-notatie) of als een enkelvoudig IP-adres in de vorm numeriek, door punten gescheiden. De port net-mgmt/ipcalc kan gebruikt worden om de berekeningen e vereenvoudigen. Aanvullende informatie is beschikbaar op de webpagina van het programma: http://jodies.de/ipcalc.

poortnummer

Wordt gebruikt voor protocollen die poortnummers ondersteunen (als TCP en UDP). Het gebruik van een poortnummer is verplicht. Er mogen ook dienstnamen uit /etc/services gebruikt worden in plaats van nummers.

in | out

Is op respectievelijk inkomende of uitgaande pakketten van toepassing. De sleutelwoorden `in` of `out` zijn verplicht in een regel.

via IF

Deze parameter geeft aan op welke interface de regel van toepassing is, waarbij `IF` de exacte naam van de bedoelde interface is.

setup

Dit is een verplicht sleutelwoord waarmee wordt aangegeven dat er gezocht wordt naar een pakket met het verzoek tot het opstarten van een TCP sessie.

keep-state

Dit is een verplicht sleutelwoord. Als er een pakket op een regel met keep-state van toepassing is, dan wordt er door de firewall een dynamische regel gemaakt die bi-directioneel verkeer zal toestaan tussen bron en bestemming en de bijbehorende poorten voor hetzelfde protocol.

limit {bron-adr | bron-poort | best-adr | best-poort}

De firewall staat maar *N* verbindingen toe met dezelfde groep parameters uit een regel. Er kunnen één of meer van de parameters bron- of bestemmingsadres en bron- en bestemmingspoort gebruikt worden. limit en keep-state kunnen niet in dezelfde regel gebruikt worden. De optie limit geeft dezelfde mogelijkheden als keep-state en voegt daar zijn eigen mogelijkheden aan toe.

Regeloptie stateful

Bij stateful filteren wordt verkeer bekeken als bi-directioneel verkeer dat samen een sessie vormt. Het heeft de mogelijkheid om te bepalen of de sessie tussen de zender en de ontvanger op de juiste wijze voortgaat. Alle pakketten die niet precies in de verwachting van een sessie passen worden automatisch als fout geblokkeerd.

De optie `check-state` wordt gebruikt om aan te geven waar IPFW-regels tegen de mogelijkheden voor dynamische regels gehouden moeten worden. Als er een passende regel bij een pakket wordt gevonden, dan kan dat pakket de firewall verlaten en wordt een nieuwe regel gemaakt voor het volgende pakket dat wordt verwacht in de sessie. Als er geen regel van toepassing is op het pakket, dan wordt de volgende regel in de groep regels getest.

De mogelijkheden voor dynamische regels zijn kwetsbaar voor een aanval die SYN-flood heet, waarmee wordt geprobeerd een zeer groot aantal regels aan te laten maken. Om deze aanval tegen te gaan, is de optie `limit` beschikbaar. Met deze optie kan het maximaal aantal simultane sessies geregeld worden op basis van bron en bestemmingsvelden. Als het aantal sessies gelijk aan het maximale aantal sessies is, wordt een pakket voor een nieuwe sessie geweigerd.

Firewallberichten loggen

De voordelen van loggen zijn duidelijk. Het biedt de mogelijkheid om na het feit informatie na te zien als: welke pakketten heeft de firewall laten vallen, waar kwamen ze vandaan en waar gingen ze heen. Dit zijn allemaal voordelen als het gaat om uitvinden waar een aanvaller vandaan komt en wat hij heeft geprobeerd.

Zelfs als logging is ingeschakeld logt IPFW nog niets uit zichzelf. De beheerder van de firewall beslist welke actieve regels iets weg moeten schrijven door het sleutelwoord `log` aan die regels toe te voegen. Gewoonlijk worden alleen `deny`-regels gelogd. Dit geldt bijvoorbeeld voor de `deny`-regel voor inkomende ICMP pings. Het is gebruikelijk om de standaardregel "ipfw default deny everything" te dupliceren, daar `log` in op te nemen, en deze als laatste in de verzameling met regels te plaatsen. Zo zijn alle pakketten te zien die niet voldeden aan ook maar één regel.

Loggen heeft ook mogelijke nadelen. Het is mogelijk om te veel te loggen en dan om te komen in logboekgegevens die uiteindelijk een schijf kunnen vullen. Een DoS aanval om een schijf met logs te vullen is een van de oudst bekende typen DoS aanvallen. Logberichten van de firewall worden niet alleen naar syslogd geschreven, maar ook op het `root` console getoond waar ze snel erg vervelend kunnen worden.

De kerneloptie `IPFIREWALL_VERBOSE_LIMIT=5` beperkt het aantal opeenvolgende berichten dat naar syslogd(8) wordt geschreven voor één specifieke regel. Als deze optie is ingeschakeld, worden in dit geval maximaal vijf berichten voor dezelfde regel gemeld. Als er meer berichten op dezelfde regel zouden zijn, zou dat als volgt aan syslogd gemeld worden:

```
last message repeated 45 times
```

Standaard worden alle gelogde pakketten weggeschreven naar /var/log/security, wat is ingesteld in /etc/syslog.conf.

Regelscript bouwen

De meeste ervaren gebruikers van IPFW maken een bestand waarin de regels staan en stellen dat zo op dat het als script uitgevoerd kan worden. Het grootste voordeel van deze methode is dat de firewallregels allemaal vervangen kunnen worden zonder dat het systeem opnieuw gestart moet worden. Deze methode is ook erg geschikt voor het testen van regels omdat de procedure zo vaak

als nodig uitgevoerd kan worden. Omdat het een script is, kan er gebruik gemaakt worden van substitutie zodat veel gebruikte waarden verduidelijkt en in meerdere regels toegepast kunnen worden. In het volgende voorbeeld wordt hier gebruik van gemaakt.

De syntaxis die in het script wordt gebruikt is compatibel met de shells sh(1), csh(1) en tcsh(1). Velden waarvoor substitutie van toepassing is worden vooraf gegaan door het dollarteken $. Definities worden niet vooraf gegaan door het voorvoegsel $. De waarden van een substitutie moet omsloten worden door "dubbele aanhalingstekens".

Een bestand met regels kan als volgt beginnen:

```
################ begin voorbeeldscript ipfw regels ###############
#
ipfw -q -f flush       # Verwijder alle bestaande regels.
# Stel standaarden in.
oif="tun0"             # uitgaande interface.
odns="192.0.2.11"      # IP adres DNS server ISP.
cmd="ipfw -q add "     # Voorvoegsel voor regel.
ks="keep-state"        # Te lui om iedere keer in te typen.
$cmd 00500 check-state
$cmd 00502 deny all from any to any frag
$cmd 00501 deny tcp from any to any established
$cmd 00600 allow tcp from any to any 80 out via $oif setup $ks
$cmd 00610 allow tcp from any to $odns 53 out via $oif setup $ks
$cmd 00611 allow udp from any to $odns 53 out via $oif $ks
#################### einde voorbeeldscript ipfw regels ############
```

Dat is alles. De feitelijke functie van de regels is in dit voorbeeld van ondergeschikt belang. Dit was slechts een voorbeeld om het gebruik van substitutie te illustreren.

Als het bovenstaande voorbeeld de inhoud van /etc/ipfw.rules was, dan kon het herladen worden met het volgende commando:

```
# sh /etc/ipfw.rules
```

/etc/ipfw.rules zou overal kunnen staan met iedere gewenste naam.

Wat in het bovenstaande voorbeeld met een bestand is gerealiseerd, kan ook met de hand:

```
# ipfw -q -f flush
# ipfw -q add 00500 check-state
# ipfw -q add 00502 deny all from any to any frag
# ipfw -q add 00501 deny tcp from any to any established
# ipfw -q add 00600 allow tcp from any to any 80 out via tun0 setup keep-state
# ipfw -q add 00610 allow tcp from any to 192.0.2.11 53 out via tun0 setup keep-state
# ipfw -q add 00611 allow udp from any to 192.0.2.11 53 out via tun0 keep-state
```

Verzameling van stateful regels

De volgende verzameling van regels, waarin geen gebruik gemaakt wordt van NAT, is een voorbeeld van hoe een erg veilige inclusieve firewall kan worden opgezet. Een inclusieve firewall laat alleen diensten toe waarvoor `pass` regels van toepassing zijn en blokkeert al het andere verkeer. Firewalls die ontworpen zijn om hele netwerksegmenten te beschermen hebben tenminste twee interfaces waarvoor regels moeten zijn die de firewall in staat stellen zijn werk te doen.

Alle UNIX® systemen en dus ook FreeBSD zijn zo ontworpen dat ze voor interne communicatie de interface lo0 en IP adres `127.0.0.1` gebruiken. De firewall moet dit interne verkeer gewoon doorgang laten vinden.

Voor de interface die is verbonden met het publieke Internet worden regels gemaakt waarmee sessies naar het Internet mogelijk gemaakt worden en toegang wordt gegeven voor pakketten die uit die sessies terug komen. Dit kan de gebruikers-PPP-interface tun0 zijn of de netwerkkaart die is verbonden met een xDSL of kabelmodem.

In gevallen dat er meer dan één netwerkkaart is aangesloten op het private netwerk achter de firewall, moeten er op de firewall-regels zijn om het verkeer tussen die interfaces vrije doorgang te geven.

De regels worden opgedeeld in drie onderdelen: alle interfaces met vrije doorgang, uitgaand op publieke interfaces en inkomend op publieke interfaces.

De volgorde van de regels in iedere sectie voor publieke interfaces moet zo zijn dat de regels die het meest gebruikt worden vóór de regels staan die minder vaak gebruikt worden. De laatste regel van een onderdeel geeft aan dat al het overige verkeer op die interface in die richting geblokkeerd en gelogd moet worden.

In het onderdeel Uitgaand van de volgende verzameling regels staan alleen regels met `allow` die parameters bevatten om individuele diensten beschikbaar te maken die publieke toegang tot Internet mogen hebben Al die regels moeten gebruik maken van de opties `proto`, `port`, `in/out`, `via` en `keep-state`. De regels met `proto tcp` maken ook gebruik van `setup` om te bekijken of het een pakket betreft voor het opzetten van een sessie om de stateful functionaliteit aan te sturen.

In het onderdeel Inkomend staan als eerste alle regels voor het blokkeren van ongewenste pakketten, om twee redenen. Als eerste kan het zo zijn dat kwaadaardige pakketten gedeeltelijk overeenkomen met legitiem verkeer. Deze regels moeten worden geblokkeerd in plaats van te worden binnengelaten, gebaseerd op hun gedeeltelijke overeenkomst met `allow`-regels. De tweede reden is dat nu ongewenste pakketten die vaak voorkomen en die bij voorkeur niet in de logboeken voorkomen niet meer van toepassing zijn op de laatste regel van het onderdeel waarin ze zouden worden gelogd. Met de laatste regel van dit onderdeel worden alle overige pakketten geblokkeerd en gelogd en ze kunnen bewijsmateriaal zijn in een zaak tegen iemand die heeft geprobeerd een systeem aan te vallen.

Iets waarop u ook moet letten is dat voor al het verkeer dat wordt geweigerd geen antwoord wordt gestuurd. Die pakketten verdwijnen gewoon. Zo weet een aanvaller niet of een pakket het doelsysteem wel heeft bereikt. Zo kan een aanvaller geen informatie verzamelen over een systeem: hoe minder informatie er over een systeem beschikbaar is, hoe veiliger het is. Als er pakketten

gelogd worden met een onbekend poortnummer, dan is de functie van dat poortnummer na te zoeken in /etc/services of op http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers. Op de volgende link worden poortnummers van Trojans beschreven: http://www.sans.org/security-resources/idfaq/oddports.php.

Voorbeeld van een set inclusieve regels

Het volgende voorbeeld is een complete inclusieve verzameling van regels die geen gebruik maakt van NAT. Deze verzameling van regels is veilig om deze regels op uw eigen systemen te gebruiken. Dit kan door commentaar te maken van een `pass`-regel voor een dienst die niet gewenst is. Logberichten die niet gewenst zijn, zijn uit te sluiten door een `deny`-regel toe te voegen aan het onderdeel Inkomend. Voor de onderstaande regels dient de interfacenaam dc0 in iedere regel vervangen te worden door de interfacenaam van de netwerkkaart in het systeem die met het publieke Internet is verbonden. Voor gebruikers van PPP zou dat tun0 zijn.

Er zit een merkbare structuur in het gebruik van deze regels:

- Alle regels die een verzoek zijn voor het opzetten van een sessie gebruiken `keep-state`.
- Alle diensten die vanaf Internet bereikbaar zijn gebruiken de optie `limit` om "flooding" te voorkomen.
- Alle regels gebruiken `in` of `out` om de richting aan te geven.
- Alle regels gebruiken `via interfacenaam` om aan te geven op welke interface de regel van toepassing is.

De volgende regels zouden in /etc/ipfw.rules kunnen staan:

```
################## Begin bestand met IPFW regels ###############################
# Verwijder eerst de bestaande regels.
ipfw -q -f flush

# Stel commando voorvoegsel in.
cmd="ipfw -q add"
pif="dc0"     # Interfacenaam van NIC die verbinding
              # met het publieke Internet heeft.

###############################################################
# Geen beperkingen op de interface aan de LAN kant. Alleen nodig
# als er een LAN is. Wijzig xl0 naar de gebruikte interfacenaam.
###############################################################
#$cmd 00005 allow all from any to any via xl0

###############################################################
# Geen beperkingen op de loopback interface.
###############################################################
$cmd 00010 allow all from any to any via lo0

###############################################################
# Sta het pakket toe als het aan de tabel met dynamische regels
# was toegevoegd met een 'allow keep-state' commando.
```

```
#####################################################################
$cmd 00015 check-state

#####################################################################
# Interface aan het publieke Internet (onderdeel Uitgaand).
# Inspecteer verzoeken om een sessie te starten van achter de
# firewall op het private netwerk of vanaf de server zelf naar
# het publieke Internet.
#####################################################################

# Geef toegang tot de DNS server van de ISP.
# x.x.x.x moet het IP adres van de DNS van de ISP zijn.
# Dupliceer deze regels als een ISP meerdere DNS servers heeft.
# Haal het IP adres evt. uit /etc/resolv.conf
$cmd 00110 allow tcp from any to x.x.x.x 53 out via $pif setup keep-state
$cmd 00111 allow udp from any to x.x.x.x 53 out via $pif keep-state

# Geef toegang tot de DHCP server van de ISP voor kabel- en
# xDSL-netwerken. Deze regel is niet nodig als gebruik gemaakt worden
# van PPP naar het publieke Internet. In dat geval kan de hele groep
# verwijderd worden. Gebruik de volgende regel en controleer het
# logboek voor het IP adres. Wijzig dan het IP adres in de regel
# commentaar hieronder en verwijder de eerste regel.
$cmd 00120 allow log udp from any to any 67 out via $pif keep-state
#$cmd 00120 allow udp from any to x.x.x.x 67 out via $pif keep-state

# Sta niet beveiligd www verkeer toe.
$cmd 00200 allow tcp from any to any 80 out via $pif setup keep-state

# Sta beveiligd www verkeer over TLS SSL toe.
$cmd 00220 allow tcp from any to any 443 out via $pif setup keep-state

# Sta het verzenden en ontvangen van e-mail toe.
$cmd 00230 allow tcp from any to any 25 out via $pif setup keep-state
$cmd 00231 allow tcp from any to any 110 out via $pif setup keep-state

# Sta de FreeBSD CVSUP functie toe voor uid root.
$cmd 00240 allow tcp from me to any out via $pif setup keep-state uid root

# Sta ping toe.
$cmd 00250 allow icmp from any to any out via $pif keep-state

# Sta Time toe naar buiten.
$cmd 00260 allow tcp from any to any 37 out via $pif setup keep-state

# Sta NNTP nieuws toe naar buiten.
$cmd 00270 allow tcp from any to any 119 out via $pif setup keep-state

# Sta beveiligde FTP, Telnet en SCP toe naar buiten.
# Deze functie maakt gebruik van SSH (secure shell).
$cmd 00280 allow tcp from any to any 22 out via $pif setup keep-state
```

```
# Sta whois toe naar buiten.
$cmd 00290 allow tcp from any to any 43 out via $pif setup keep-state

# Blokkeer en log al het andere dat probeert buiten te komen.
# Deze regel dwingt de 'block all' logica af.
$cmd 00299 deny log all from any to any out via $pif

#####################################################################
# Interface aan het publieke Internet (onderdeel Inkomend).
# Inspecteert pakketten die van het publieke Internet komen
# met als bestemming de host zelf of het private netwerk.
#####################################################################

# Blokkeer al het verkeer voor niet-routeerbare of gereserveerde
# adresreeksen.
$cmd 00300 deny all from 192.168.0.0/16 to any in via $pif   #RFC 1918 privaat IP
$cmd 00301 deny all from 172.16.0.0/12 to any in via $pif     #RFC 1918 privaat IP
$cmd 00302 deny all from 10.0.0.0/8 to any in via $pif        #RFC 1918 privaat IP
$cmd 00303 deny all from 127.0.0.0/8 to any in via $pif       #loopback
$cmd 00304 deny all from 0.0.0.0/8 to any in via $pif         #loopback
$cmd 00305 deny all from 169.254.0.0/16 to any in via $pif    #DHCP auto-config
$cmd 00306 deny all from 192.0.2.0/24 to any in via $pif      #gereserveerd voor
documentatie
$cmd 00307 deny all from 204.152.64.0/23 to any in via $pif   #Sun cluster
interconnect
$cmd 00308 deny all from 224.0.0.0/3 to any in via $pif       #Klasse D & E multicast

# Blokkeer publieke pings.
$cmd 00310 deny icmp from any to any in via $pif

# Blokkeer ident.
$cmd 00315 deny tcp from any to any 113 in via $pif

# Blokkeer alle Netbios diensten. 137=naam, 138=datagram, 139=sessie.
# Netbios is de Windows(R) bestandsdeeldienst.
# Blokkeer Windows hosts2 name server verzoeken 81.
$cmd 00320 deny tcp from any to any 137 in via $pif
$cmd 00321 deny tcp from any to any 138 in via $pif
$cmd 00322 deny tcp from any to any 139 in via $pif
$cmd 00323 deny tcp from any to any 81 in via $pif

# Blokkeer gefragmenteerde pakketten.
$cmd 00330 deny all from any to any frag in via $pif

# Blokkeer ACK pakketten die niet in de tabel met dynamische regels
# staan.
$cmd 00332 deny tcp from any to any established in via $pif

# Geef toegang tot de DHCP server van de ISP voor kabel- en
# xDSL-netwerken. Deze regel is niet nodig als gebruik gemaakt worden
```

```
# van PPP naar het publieke Internet. In dat geval kan de hele groep
# verwijderd worden. Hier wordt hetzelfde IP adres gebruikt als in de
# sectie voor Uitgaand verkeer.
#$cmd 00360 allow udp from any to x.x.x.x 67 in via $pif keep-state

# Sta inkomend webverkeer toe omdat er een Apache server draait.
$cmd 00400 allow tcp from any to me 80 in via $pif setup limit src-addr 2

# Sta beveiligde FTP, telnet en SCP toe vanaf Internet.
$cmd 00410 allow tcp from any to me 22 in via $pif setup limit src-addr 2

# Sta niet beveiligde telnet sessie toe vanaf het publieke Internet.
# Dit heeft het label ``niet veilig'' omdat gebruikersnaam en
# wachtwoord als platte tekst over Internet gaan. Als er geen telnet
# server draait, hoeft deze regel niet actief te zijn.
$cmd 00420 allow tcp from any to me 23 in via $pif setup limit src-addr 2

# Weiger en log alle niet toegestane inkomende verbindingen van buiten.
$cmd 00499 deny log all from any to any in via $pif

# Al het andere verkeer wordt standaard geblokkeerd. Weiger en log alle
# pakketten die tot hier zijn gekomen om te bekijken welke het waren.
$cmd 00999 deny log all from any to any
################## Einde bestand met IPFW regels ##########################
```

Voorbeeld NAT en stateful regels

Om NAT met IPFW te gebruiken moeten een extra aantal instellingen gemaakt worden. In het instellingenbestand voor de kernel moet `option IPDIVERT` toegevoegd worden aan de andere opties van IPFIREWALL.

Naast de normale IPFW opties in /etc/rc.conf zijn de volgende nodig:

```
natd_enable="YES"                   # Schakel NATD in
natd_interface="rl0"                # interfacenaam voor de publieke Internet NIC
natd_flags="-dynamic -m"            # -m = behoud poortnummers als mogelijk
```

Stateful regels samen met de regel `divert natd` (Network Address Translation) gebruiken maakt het schrijven van regels veel gecompliceerder. De plaats van de regels met `check-state` en `divert natd` zijn van kritiek belang. De logica bestaat niet langer uit het eenvoudigweg van boven naar beneden doorwerken van de regels. Er wordt dan ook een nieuw type actie gebruik: `skipto`. Bij het gebruik van `skipto` is het verplicht iedere regel te nummeren zodat duidelijk is waar een `skipto` precies heen springt.

Hieronder staat een groep regels zonder commentaar waarin een manier om pakketten door de groep regels te leiden wordt aangegeven.

De verwerking begint met de eerste regel en er wordt steeds een volgende regel gecontroleerd tot het einde wordt bereikt of totdat een regel op het gecontroleerde pakket van toepassing is, en het

pakket uit de firewall wordt vrijgelaten. In het voorbeeld zijn de regels 100, 101, 450, 500, en 510 van belang. Die regels regelen de vertaling van inkomende en uitgaande pakketten zodat er in de tabel met de dynamische `keep-state`-regels altijd het private IP-adres staat. Daarnaast is het van belang op te merken dat er in alle `allow`- en `deny`-regels de richting van het pakket wordt gecontroleerd (inkomend of uitgaand) en over welke interface het pakket gaat. Merk ook op dat alle uitgaande verzoeken voor het starten van een sessie met een `skipto` naar regel 500 gaan voor NAT.

Stel dat een gebruiker zijn webbrowser gebruikt om een webpagina op te halen. Webpagina's worden over poort 80 verzonden. Er komt een pakket de firewall binnen dat niet past bij regel 100 omdat het naar buiten gaat en niet naar binnen. Het komt voorbij regel 101 omdat dit het eerste pakket is en er dus nog niets over in de dynamische keep-state tabel staat. Als het pakket bij 125 aankomt blijkt het te passen bij die regel. Het gaat naar buiten door de interface aan het publieke Internet. Het pakket heeft dan nog steeds het bron-IP-adres van het private LAN. Als blijkt dat deze regel geldt, dan gebeuren er twee dingen: door `keep-state` wordt er een regel in de dynamische keep-state tabel gezet en wordt de aangegeven actie uitgevoerd. De actie is onderdeel van de informatie uit de dynamische tabel. In dit geval is het `skipto rule 500`. In regel 500 wordt NAT op het IP-adres van het pakket toegepast en dan kan het weg. Dit is van groot belang. Dit pakket komt aan op zijn bestemming en als er een pakket als antwoord terug komt, dan begint de verwerking van het antwoordpakket weer van voor af aan. Nu voldoet het aan regel 100 en dus wordt het bestemmingsadres vertaald naar het bijbehorende IP-adres op het LAN. Daarna past het bij de `check-state`-regel en wordt een vermelding in de tabel gevonden wat betekent dat er een bestaande sessie is en wordt het doorgelaten naar het LAN. Het gaat dan naar de PC op het LAN die als eerste een pakket heeft verzonden en die verstuurt een nieuw pakket met de vraag om een volgend segment met gegevens naar de server. Nu blijkt bij controle van de `check-state`-regel dat die op het pakket van toepassing moet zijn en er staat een vermelding in de tabel voor uitgaand verkeer. Daarom wordt de bijbehorende actie `skipto rule 500` uitgevoerd. Het pakket springt naar regel 500, er wordt NAT op toegepast en het kan zijn weg vervolgen.

Wat betreft binnenkomende pakketten wordt alles dat onderdeel is van een bestaande sessie automatisch afgehandeld door de `check-state`-regel en de correct geplaatste `divert natd`-regels. Nu hoeven alleen de foute pakketten nog geweigerd te worden en moeten de inkomende geauthoriseerde diensten doorgelaten worden. In dit geval draait er een Apache server op de firewall-machine die vanaf Internet bereikbaar moet zijn. Het nieuwe inkomende pakket past bij regel 100 en het IP-adres wordt aangepast aan het interne IP-adres van de firewall-machine. Dat pakket wordt dan gecontroleerd op alle ongewenste eigenschappen en komt uiteindelijk aan bij regel 425 die van toepassing blijkt te zijn. In dat geval kunnen er twee dingen gebeuren: de pakketregel wordt in de dynamische keep-state tabel gezet, maar nu wordt het aantal nieuwe sessies dat van het bron IP-adres komt gelimiteerd tot twee. Dit is een bescherming tegen DoS-aanvallen op de dienst die op dat poortnummer wordt aangeboden. De actie is `allow`, dus het pakket wordt tot het LAN toegelaten. Voor het pakket dat als antwoord wordt verstuurd herkent de `check-state` regel dat het pakket bij een bestaande sessie hoort. Het stuurt het naar regel 500 voor NAT en stuurt het via de uitgaande interface weg.

Voorbeeld Set Regels #1:

```
#!/bin/sh
cmd="ipfw -q add"
skip="skipto 500"
```

```
pif=rl0
ks="keep-state"
good_tcpo="22,25,37,43,53,80,443,110,119"

ipfw -q -f flush

$cmd 002 allow all from any to any via xl0  # exclude LAN traffic
$cmd 003 allow all from any to any via lo0  # exclude loopback traffic

$cmd 100 divert natd ip from any to any in via $pif
$cmd 101 check-state

# Toegestaan uitgaand verkeer.
$cmd 120 $skip udp from any to xx.168.240.2 53 out via $pif $ks
$cmd 121 $skip udp from any to xx.168.240.5 53 out via $pif $ks
$cmd 125 $skip tcp from any to any $good_tcpo out via $pif setup $ks
$cmd 130 $skip icmp from any to any out via $pif $ks
$cmd 135 $skip udp from any to any 123 out via $pif $ks

# Blokkeer al het verkeer voor niet-routeerbare of gereserveerde
# adresreeksen.
$cmd 300 deny all from 192.168.0.0/16  to any in via $pif  #RFC 1918 privaat IP
$cmd 301 deny all from 172.16.0.0/12   to any in via $pif  #RFC 1918 privaat IP
$cmd 302 deny all from 10.0.0.0/8      to any in via $pif  #RFC 1918 privaat IP
$cmd 303 deny all from 127.0.0.0/8     to any in via $pif  #loopback
$cmd 304 deny all from 0.0.0.0/8       to any in via $pif  #loopback
$cmd 305 deny all from 169.254.0.0/16  to any in via $pif  #DHCP auto-config
$cmd 306 deny all from 192.0.2.0/24    to any in via $pif  #gereserveerd voor
documentatie
$cmd 307 deny all from 204.152.64.0/23 to any in via $pif  #Sun cluster
$cmd 308 deny all from 224.0.0.0/3     to any in via $pif  #Klasse D & E multicast

# Toegestaan inkomend verkeer.
$cmd 400 allow udp from xx.70.207.54 to any 68 in $ks
$cmd 420 allow tcp from any to me 80 in via $pif setup limit src-addr 1

$cmd 450 deny log ip from any to any

# Dit is de 'skipto' locatie voor de uitgaande stateful regels.
$cmd 500 divert natd ip from any to any out via $pif
$cmd 510 allow ip from any to any

########################## Einde regels  ###################
```

Het volgende voorbeeld doet vrijwel hetzelfde als het bovenstaande, maar volgt een zelfdocumenterende stijl voor het opstellen van regels en commentaar waardoor minder ervaren gebruikers beter kunnen begrijpen wat de regels doen.

Voorbeeld Set Regels #2:

```
#!/bin/sh
################ Begin bestand met IPFW regels ###############################
# Verwijder eerst de bestaande regels.
ipfw -q -f flush

# Stel commando voorvoegsel in.
cmd="ipfw -q add"
skip="skipto 800"
pif="rl0"     # Interfacenaam van NIC die verbinding
              # met het publieke Internet heeft.

######################################################################
# Geen beperkingen op de interface aan de LAN kant.
# Wijzig xl0 naar de gebruikte interfacenaam.
######################################################################
$cmd 005 allow all from any to any via xl0

######################################################################
# Geen beperkingen op de loopback interface.
######################################################################
$cmd 010 allow all from any to any via lo0

######################################################################
# Controleer of pakket inkomend is. NAT in dat geval.
######################################################################
$cmd 014 divert natd ip from any to any in via $pif

######################################################################
# Sta het pakket toe als het aan de tabel met dynamische regels
# was toegevoegd met een 'allow keep-state' commando.
######################################################################
$cmd 015 check-state

######################################################################
# Interface aan het publieke Internet (onderdeel Uitgaand).
# Inspecteer verzoeken om een sessie te starten van achter de
# firewall op het private netwerk of vanaf de server zelf naar
# het publieke Internet.
######################################################################

# Geef toegang tot de DNS server van de ISP.
# x.x.x.x moet het IP adres van de DNS van de ISP zijn.
# Dupliceer deze regels als een ISP meerdere DNS servers heeft.
# Haal het IP adres evt. uit /etc/resolv.conf
$cmd 020 $skip tcp from any to x.x.x.x 53 out via $pif setup keep-state

# Geef toegang tot de DHCP server van de ISP voor kabel en xDSL.
$cmd 030 $skip udp from any to x.x.x.x 67 out via $pif keep-state

# Sta niet beveiligd www verkeer toe.
```

```
$cmd 040 $skip tcp from any to any 80 out via $pif setup keep-state

# Sta beveiligd www verkeer over TLS SSL toe.
$cmd 050 $skip tcp from any to any 443 out via $pif setup keep-state

# Sta het verzenden en ontvangen van e-mail toe.
$cmd 060 $skip tcp from any to any 25 out via $pif setup keep-state
$cmd 061 $skip tcp from any to any 110 out via $pif setup keep-state

# Sta de FreeBSD CVSUP functie toe voor uid root.
$cmd 070 $skip tcp from me to any out via $pif setup keep-state uid root

# Sta ping toe naar het publieke Internet.
$cmd 080 $skip icmp from any to any out via $pif keep-state

# Sta Time toe.
$cmd 090 $skip tcp from any to any 37 out via $pif setup keep-state

# Sta NNTP nieuws toe.
$cmd 100 $skip tcp from any to any 119 out via $pif setup keep-state

# Sta beveiligde FTP, Telnet en SCP toe.
# Deze functie maakt gebruik van SSH (secure shell).
$cmd 110 $skip tcp from any to any 22 out via $pif setup keep-state

# Sta whois toe.
$cmd 120 $skip tcp from any to any 43 out via $pif setup keep-state

# Sta NPT tijdserver toe.
$cmd 130 $skip udp from any to any 123 out via $pif keep-state

########################################################################
# Interface aan het publieke Internet (onderdeel Inkomend).
# Inspecteert pakketten die van het publieke Internet komen met
# als bestemming deze gateway-server zelf of het private netwerk.
########################################################################

# Blokkeer al het verkeer voor niet-routeerbare of gereserveerde
# adresreeksen.
$cmd 300 deny all from 192.168.0.0/16  to any in via $pif  #RFC 1918 privaat IP
$cmd 301 deny all from 172.16.0.0/12   to any in via $pif  #RFC 1918 privaat IP
$cmd 302 deny all from 10.0.0.0/8      to any in via $pif  #RFC 1918 privaat IP
$cmd 303 deny all from 127.0.0.0/8     to any in via $pif  #loopback
$cmd 304 deny all from 0.0.0.0/8       to any in via $pif  #loopback
$cmd 305 deny all from 169.254.0.0/16  to any in via $pif  #DHCP auto-config
$cmd 306 deny all from 192.0.2.0/24    to any in via $pif  #gereserveerd voor
documentatie
$cmd 307 deny all from 204.152.64.0/23 to any in via $pif  #Sun cluster
$cmd 308 deny all from 224.0.0.0/3     to any in via $pif  #Klasse D & E multicast

# Blokkeer ident.
```

```
$cmd 315 deny tcp from any to any 113 in via $pif

# Blokkeer alle Netbios diensten. 137=naam, 138=datagram, 139=sessie.
# Netbios is de Windows® bestandsdeeldienst.
# Blokkeer Windows hosts2 name server verzoeken 81.
$cmd 320 deny tcp from any to any 137 in via $pif
$cmd 321 deny tcp from any to any 138 in via $pif
$cmd 322 deny tcp from any to any 139 in via $pif
$cmd 323 deny tcp from any to any 81  in via $pif

# Blokkeer gefragmenteerde pakketten.
$cmd 330 deny all from any to any frag in via $pif

# Blokkeer ACK pakketten die niet in de tabel met dynamische regels
# staan.
$cmd 332 deny tcp from any to any established in via $pif

# Geef toegang tot de DHCP server van de ISP voor kabel- en
# xDSL-netwerken. Deze regel is niet nodig als gebruik gemaakt worden
# van PPP naar het publieke Internet. In dat geval kan de hele groep
# verwijderd worden. Hier wordt hetzelfde IP adres gebruikt als in de
# sectie voor Uitgaand verkeer.
$cmd 360 allow udp from x.x.x.x to any 68 in via $pif keep-state

# Sta inkomend webverkeer toe omdat er een Apache server draait.
$cmd 370 allow tcp from any to me 80 in via $pif setup limit src-addr 2

# Sta beveiligde FTP, telnet en SCP toe vanaf Internet.
$cmd 380 allow tcp from any to me 22 in via $pif setup limit src-addr 2

# Sta niet beveiligde telnet sessie toe vanaf het publieke Internet.
# Dit heeft het label ``niet veilig'' omdat gebruikersnaam en
# wachtwoord als platte tekst over Internet gaan. Als er geen telnet
# server draait, hoeft deze regel niet actief te zijn.
#$cmd 390 allow tcp from any to me 23 in via $pif setup limit src-addr 2

# Weiger en log alle niet toegestane inkomende verbindingen vanaf het
# publieke Internet.
$cmd 400 deny log all from any to any in via $pif

# Weiger en log alle niet toegestane uitgaande verbindingen naar
# Internet.
$cmd 450 deny log all from any to any out via $pif

# Dit is de 'skipto' locatie voor de uitgaande stateful regels
$cmd 800 divert natd ip from any to any out via $pif
$cmd 801 allow ip from any to any

# Al het andere verkeer wordt standaard geblokkeerd. Weiger en log alle
# pakketten die tot hier zijn gekomen om te bekijken welke het waren.
$cmd 999 deny log all from any to any
```

```
################ Einde bestand met IPFW regels ########################
```

Hoofdstuk 29. Geavanceerde netwerken

29.1. Samenvatting

Dit hoofdstuk zal een aantal onderwerpen over geavanceerde netwerken behandelen.

Na het lezen van dit hoofdstuk is bekend:

- De beginselen van gateways en routes.
- Hoe IEEE® 802.11- en Bluetooth®-apparaten te installeren.
- Hoe FreeBSD als een bridge te laten werken.
- Hoe een schijfloze machine vanaf het netwerk op te starten.
- Hoe opstarten met netwerk-PXE en een NFS-root-bestandssysteem te installeren.
- Hoe Network Address Translation te installeren.
- Hoe IPv6 op een FreeBSD-machine te installeren.
- Hoe ATM in te stellen.
- Hoe de mogelijkheden van CARP, het Common Address Redundancy Protocol, aan te zetten en te benutten.

Voordat dit hoofdstuk gelezen wordt, dient de lezer:

- De beginselen van de scripts in /etc/rc te begrijpen.
- Bekend te zijn met basisnetwerktermen.
- Te weten hoe een nieuwe FreeBSD-kernel in te stellen en te installeren (De FreeBSD-kernel instellen).
- Te weten hoe aanvullende software van derde partijen te installeren (Applicaties installeren. pakketten en ports).

29.2. Gateways en routes

Indien een machine een andere machine over een netwerk wil vinden, dient er een mechanisme te zijn dat beschrijft hoe van de ene naar de andere machine te gaan. Dit wordt *routen* genoemd. Een "route" is een gedefinieerd adressenpaar: een "bestemming" en een "gateway". Het paar geeft aan dat door deze *gateway* gecommuniceerd moet worden om bij deze *bestemming* aan te komen. Er zijn drie soorten bestemmingen: individuele host, subnetten en "standaard". De "standaardroute" wordt gebruikt indien geen van de andere routes van toepassing zijn. Verderop wordt verder op standaardroutes ingegaan. Er zijn ook drie soorten gateways: individuele hosts, interfaces (ook wel "verbindingen " genoemd), en Ethernet-hardware-adressen (MAC-adressen).

29.2.1. Een voorbeeld

Om de verschillende aspecten van routen te illustreren, wordt het volgende voorbeeld van `netstat` gebruikt:

```
% netstat -r
Routing tables

Destination      Gateway            Flags     Refs     Use     Netif Expire

default          outside-gw         UGSc       37      418      ppp0
localhost        localhost          UH          0      181       lo0
test0            0:e0:b5:36:cf:4f   UHLW        5    63288       ed0     77
10.20.30.255     link#1             UHLW        1     2421
example.com      link#1             UC          0        0
host1            0:e0:a8:37:8:1e    UHLW        3     4601       lo0
host2            0:e0:a8:37:8:1e    UHLW        0        5       lo0 =>
host2.example.com link#1             UC          0        0
224              link#1             UC          0        0
```

De eerste twee regels geven de standaardroute (die behandeld wordt in de volgende sectie) en de `localhost`-route aan.

De interface (kolom `Netif`) dat deze routeertabel aangeeft om voor `localhost` te gebruiken is lo0, ook bekend als het teruglusapparaat. Dit geeft aan dat alle verkeer voor deze bestemming intern gehouden moet worden, in plaats van het over het LAN te sturen, aangezien het alleen aankomt op de plaats waar het verzonden werd.

Het volgende dat opvalt zijn de adressen die beginnen met `0:e0:`. Dit zijn Ethernet-hardware adressen, ook bekend als MAC-adressen. FreeBSD zal automatisch elke host (`test0` in het voorbeeld) op het lokale Ethernet identificeren en een route voor die host toevoegen, direct van deze host over de Ethernet-interface, ed0. Er is ook een timeout (kolom `Expire`) met deze routesoort geassocieerd, die gebruikt wordt indien er binnen een bepaalde tijd geen bericht komt van de host. Indien dit gebeurt, wordt de route naar deze host automatisch verwijderd. Deze hosts worden geïdentificeerd door middel van een mechanisme dat bekend staat als RIP (Routing Information Protocol), dat routes naar lokale hosts bepaald door middel van een kortste-pad algoritme.

FreeBSD zal ook subnetroutes voor het lokale subnet toevoegen (`10.20.30.255` is het broadcast-adres voor het subnet `10.20.30`, en `example.com` is de domeinnaam die bij dat subnet hoort). De aanduiding `link#1` verwijst naar de eerste Ethernetkaart in de machine. Merk op dat voor hen geen aanvullende interface is gespecificeerd.

Voor beide groepen (lokale netwerkhosts en lokale subnetten) worden de routes automatisch ingesteld door een daemon genaamd routed. Indien dit niet draait, zullen alleen routes die statisch gedefinieerd (i.e., expliciet vermeld zijn) bestaan.

De regel met `host1` verwijst naar deze host, het kent deze door het Ethernetadres. Aangezien het de zendende host is, weet FreeBSD dat het de teruglus-interface (lo0) moet gebruiken, in plaats van het over de Ethernet-interface te verzenden.

De twee regels met `host2` geven een voorbeeld van wat er gebeurt als een alias met ifconfig(8) gebruikt wordt (in de sectie over Ethernet staan redenen waarom dit gedaan wordt). Het symbool ⇒ na de interface lo0 zegt dat niet alleen de teruglus gebruikt wordt (aangezien dit adres ook verwijst naar de lokale host), maar specifiek dat dit een alias is. Zulke routes verschijnen alleen op

de hosts die de alias ondersteunen; alle andere hosts op het lokale netwerk vermelden simpelweg een regel met `link#1` voor zulke routes.

De laatste regel (bestemming subnet `224`) heeft te maken met multicasten, wat in een andere sectie besproken wordt.

Als laatste staan in de kolom `Flags` verschillende attributen. Hieronder staat een korte tabel met enkele van deze vlaggen en hun betekenis:

U	Up: De route is actief.
H	Host: De bestemming van de route is een enkele host.
G	Gateway: Stuur alles voor deze bestemming door naar dit verre systeem, dat zoekt daar uit waar het verder naar te sturen.
S	Statisch: Deze route was handmatig ingesteld, dus niet automatisch door het systeem aangemaakt.
C	Kloon: Maakt op basis van deze route een nieuwe route aan voor machines waarmee verbinding wordt gemaakt. Dit soort routes wordt gewoonlijk in lokale netwerken gebruikt.
W	WasGekloond: Geeft aan dat een route automatisch was ingesteld gebaseerd op een LAN (kloon)-route.
L	Verbinding: De route maakt gebruik van verwijzingen naar Ethernet-hardware.

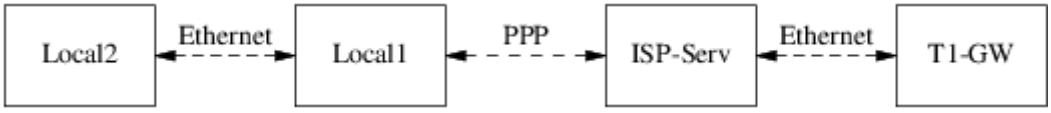
29.2.2. Standaardroutes

Wanneer het lokale systeem een verbinding met een verre host moet maken, controleert het de routeertabel op reeds bekende paden. Indien de verre host binnen een subnet valt waarvan bekend is hoe het bereikt kan worden (gekloonde routes), controleert het systeem of het met de daarbij behorende interface verbinding kan maken.

Indien alle bekende paden falen, heeft het systeem één laatste mogelijkheid: de "standaardroute". Deze route is een speciaal soort gateway-route (gewoonlijk de enig aanwezige in het systeem) en is altijd gemarkeerd met een `c` in het vlaggenveld. Voor hosts op een LAN staat deze gateway ingesteld op de machine die een directe verbinding met de buitenwereld heeft (via een PPP-verbinding, DSL, kabelmodem, T1, of een ander netwerkinterface).

Indien de standaardroute wordt ingesteld voor een machine die zelf als gateway naar de buitenwereld werkt, zal de standaardroute de gateway-machine van de internetprovider zijn.

Hieronder volgt een voorbeeld van standaardroutes. Dit is een veelgebruikte opstelling:



De hosts `Lokaal1` en `Lokaal2` staan op deze site. `Lokaal1` is verbonden met een internetprovider via een inbel-PPP-verbinding. Deze PPP-server is door een LAN verbonden met een andere gateway-computer door een externe interface naar de Internet-feed van de internetprovider.

De standaardroutes voor de machines zijn:

Host	Standaard gateway	Interface
Lokaal2	Lokaal1	Ethernet
Lokaal1	T1-GW	PPP

Een veelvoorkomende vraag is "Waarom (of hoe) moet worden ingesteld dat `T1-GW` de standaard gateway is voor `Lokaal1`, in plaats van de server van de internetprovider waarmee het verbonden is?".

Onthoud dat, aangezien de PPP-interface een adres gebruikt op het lokale netwerk van de internetprovider voor deze kant van de verbinding, routes voor alle andere machines op het lokale netwerk van de internetprovider automatisch aangemaakt worden. Daarom is het al bekend hoe de machine `T1-GW` bereikt kan worden, dus is de tussenstap dat het verkeer eerst naar de server van de internetprovider gestuurd wordt niet nodig.

Het is gebruikelijk om het adres `X.X.X.1` te gebruiken als het gateway-adres voor het lokale netwerk. Dus (gebruikmakend van hetzelfde voorbeeld), indien de lokale klasse-C adresruimte `10.20.30` was en de internetprovider `10.9.9` gebruikte, zouden de standaardroutes als volgt zijn:

Host	Standaardroute
Lokaal2 (10.20.30.2)	Lokaal1 (10.20.30.1)
Lokaal1 (10.20.30.1, 10.9.9.30)	T1-GW (10.9.9.1)

De standaardroute kan eenvoudig in /etc/rc.conf gedefinieerd worden. In dit voorbeeld werd de volgende regel aan /etc/rc.conf van `Lokaal2` toegevoegd:

```
defaultrouter="10.20.30.1"
```

Het is ook mogelijk dit met het commando route(8) direct vanaf de opdrachtregel te doen:

```
# route add default 10.20.30.1
```

Voor meer informatie over het handmatig manipuleren van netwerkrouteertabellen kan de hulppagina route(8) geraadpleegd worden.

29.2.3. Dual Homed machines

Er is nog één andere soort opstelling die behandeld dient te worden, en dat is een host die in twee verschillende netwerken zit. Technisch gezien telt elke machine die als gateway dienst doet (in bovenstaand voorbeeld door een PPP-verbinding te gebruiken) als een dual-homed host. Maar de term wordt echt alleen gebruikt om naar een machine te verwijzen die in twee LAN's zit.

In het ene geval heeft de machine twee Ethernetkaarten, waarbij elke kaart een adres op de gescheiden subnetten heeft. Een alternatief is dat de machine slechts één Ethernetkaart heeft en gebruikt maakt van ifconfig(8) aliasing. Het eerste wordt gebruikt indien er twee fysiek gescheiden Ethernet-netwerken in gebruik zijn, het laatste indien er één fysiek netwerksegment is, maar er twee logisch gescheiden subnetten zijn.

In beide gevallen worden er routeertabellen aangemaakt zodat elk subnet weet dat deze machine de gedefinieerde gateway (ingaande route) naar het andere subnet is. Deze opstelling, waarbij de machine dienst doet als router tussen de twee subnetten, wordt vaak gebruikt voor het implementeren van pakketfilters of firewall-beveiliging in één of beide richtingen.

Om deze machine daadwerkelijk pakketten te laten forwarden tussen de twee interfaces, moet aan FreeBSD verteld worden dat het deze mogelijkheid aan moet zetten. In de volgende sectie staan meer details over hoe dit te doen.

29.2.4. Een router bouwen

Een netwerkrouter is simpelweg een systeem dat pakketten van de ene naar de andere interface doorstuurt. Internetstandaarden en goede ontwerppraktijken verhinderen het FreeBSD Project dit standaard in FreeBSD aan te zetten. Deze mogelijkheid kan worden aangezet door de volgende variabele in rc.conf(5) op `YES` in te stellen:

```
gateway_enable="YES"    # Op YES instellen indien deze host een gateway is
```

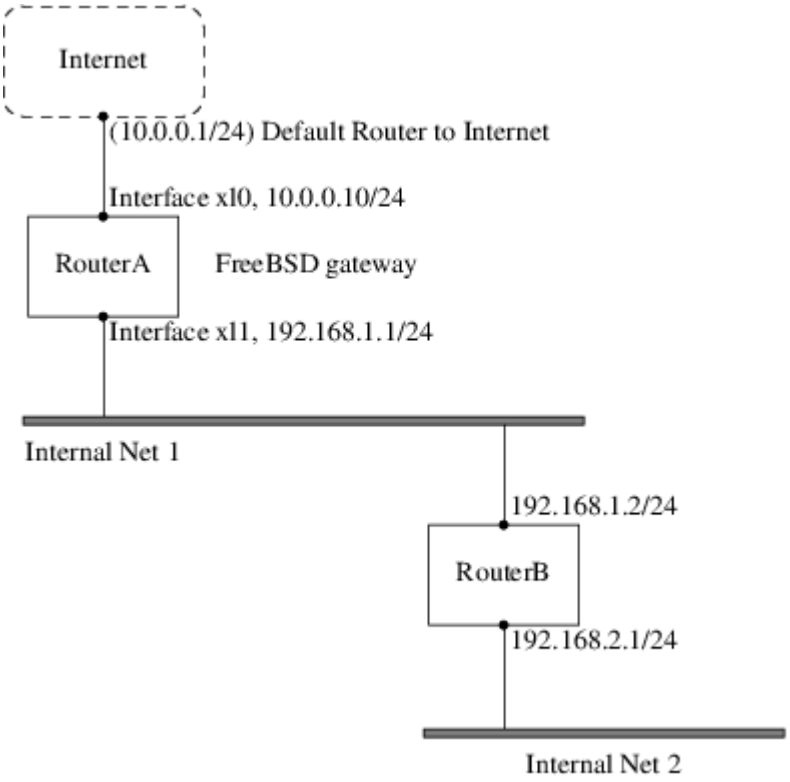
Deze optie stelt de sysctl(8) variabele `net.inet.ip.forwarding` in op `1`. Indien het nodig is om het routen tijdelijk te stoppen, kan deze variabele tijdelijk op `0` worden teruggezet.

De nieuwe router heeft routes nodig om te weten waar het het verkeer naar toe moet sturen. Voor een eenvoudig netwerk kunnen statische routes gebruikt worden. FreeBSD wordt met het standaard BSD routeer-daemon routed(8) geleverd, dat RIP (zowel versie 1 en versie 2) en IRDP spreekt. Ondersteuning voor BGP v4, OSPF v2, en andere slimme routeerprotocollen is beschikbaar via het pakket net/zebra. Ook zijn commerciële producten als GateD® beschikbaar voor complexere netwerkrouteer-oplossingen.

29.2.5. Statische routes opzetten

Handmatige configuratie

Er wordt van het volgende netwerk uitgegaan:



In dit scenario is `RouterA` een FreeBSD-machine die dienst doet als router naar de rest van het Internet. Het heeft een standaardroute ingesteld op `10.0.0.1`, dat het in staat stelt om verbindingen met de buitenwereld te maken. Er wordt aangenomen dat `RouterB` reeds juist is ingesteld en dat het weet hoe het waar naar toe moet gaan. (In dit plaatje is dit simpel. Voeg een standaardroute op `RouterB` toe door `192.168.1.1` als gateway te gebruiken.)

De routeertabel voor `RouterA` zou er ongeveer als volgt uitzien:

```
% netstat -nr
Routing tables

Internet:
Destination        Gateway            Flags    Refs      Use  Netif  Expire
default            10.0.0.1           UGS         0    49378    xl0
127.0.0.1          127.0.0.1          UH          0        6    lo0
10.0.0.0/24          link#1             UC          0        0    xl0
192.168.1.0/24       link#2             UC          0        0    xl1
```

Met de huidige routeertabel is `RouterA` niet in staat om Intern Net 2 te bereiken. Het heeft geen route voor `192.168.2.0/24`. Een manier om dit te verhelpen is om de route handmatig toe te voegen. Het volgende commando voegt het netwerk Intern Net 2 toe aan de routeertabel van `RouterA` door `192.168.1.2` als de volgende hop te gebruiken:

```
# route add -net 192.168.2.0/24 192.168.1.2
```

Nu kan `RouterA` elke host op het netwerk `192.168.2.0/24` bereiken.

Persistente configuratie

Bovenstaand voorbeeld is perfect voor het instellen van een statische route op een draaiend systeem. Een probleem is dat de routeerinformatie verdwijnt indien de FreeBSD-machine opnieuw wordt opgestart. Aanvullende statische routes kunnen in /etc/rc.conf opgenomen worden:

```
# Voeg Intern Net 2 als een statische route toe
static_routes="internnet2"
route_internnet2="-net 192.168.2.0/24 192.168.1.2"
```

De instellingsvariabele `static_routes` is een lijst van strings gescheiden door een spatie. Elke string verwijst naar een routenaam. Bovenstaand voorbeeld heeft slechts één string in `static_routes`. Dit is de string *internnet2*. Vervolgens wordt een instellingsvariabele `route_internnet2` toegevoegd waarin alle instellingsparameters staan die aan het commando route(8) moeten worden doorgegeven. Voor bovenstaand voorbeeld zou het volgende commando zijn gebruikt:

```
# route add -net 192.168.2.0/24 192.168.1.2
```

Dus is `"-net 192.168.2.0/24 192.168.1.2"` nodig.

Zoals hierboven is vermeld is het mogelijk om meerdere strings in `static_routes` te hebben. Dit maakt het mogelijk om meerdere statische routes aan te maken. De volgende regels geven een voorbeeld van het toevoegen van statische routes voor de netwerken `192.168.0.0/24` en `192.168.1.0/24` op een denkbeeldige router:

```
static_routes="net1 net2"
route_net1="-net 192.168.0.0/24 192.168.0.1"
route_net2="-net 192.168.1.0/24 192.168.1.1"
```

29.2.6. Routes propageren

Er is al gesproken over hoe routes naar de buitenwereld te definiëren, maar niet over hoe de buitenwereld ons kan vinden.

Het is al bekend dat routeertabellen aangemaakt kunnen worden zodat al het verkeer voor een bepaalde adresruimte (in ons voorbeeld een klasse-C subnet) naar een bepaalde host op dat netwerk gezonden kan worden, dat de ingaande pakketten doorgeeft.

Wanneer een adresruimte aan een site wordt toegewezen, stelt de serviceprovider al hun routeertabellen zodanig in dat al het verkeer voor het bijhorende subnet naar de PPP-verbinding van de site gezonden wordt. Maar hoe weten sites door het land heen hoe naar de internetprovider van deze site te versturen?

Er bestaat een systeem (dat veel lijkt op de gedistribueerde DNS-informatie) dat alle toegewezen adresruimtes bijhoudt, en hun verbindingspunt met de Internet Backbone definieert. De "Backbone" zijn de grote kabels die Internetverkeer door het land en over de wereld sturen. Elke backbone-machine heeft een kopie van een master-verzameling van tabellen, die verkeer voor een

bepaald netwerk naar een bepaalde backbone-carrier sturen, en van daaruit naar een keten van serviceproviders totdat het netwerk van de site bereikt is.

Het is de taak van de serviceprovider om bij de backbone-sites aan te geven dat zij het verbindingspunt (en dus het ingaande pad) zijn voor de site. Dit staat bekend als routepropagatie.

29.2.7. Problemen oplossen

Soms is er een probleem met routepropagatie en kunnen sommige sites geen verbinding maken. Misschien is het nuttigste commando om proberen uit te zoeken waar het routen misgaat traceroute(8). Het is ook nuttig als er geen verbinding mogelijk lijkt met een verre machine (dus als ping(8) faalt).

Het commando traceroute(8) wordt gedraaid met de naam van de verre host waarmee geprobeerd wordt te verbinden. Het laat de gateway-hosts zien langs het gepoogde pad, dat uiteindelijk de doelhost bereikt, of wegens een gebrek aan verbinding afgebroken wordt.

Raadpleeg voor meer informatie de hulppagina voor traceroute(8).

29.2.8. Multicast routen

FreeBSD ondersteunt zowel multicast-applicaties als multicast routen van huis uit. Voor multicast-applicaties is geen speciale configuratie van FreeBSD nodig; applicaties draaien over het algemeen als geleverd. Voor multicast routen dient ondersteuning in de kernel gecompileerd te worden:

```
options MROUTING
```

Ook dient de multicast-routeer-daemon mrouted(8) ingesteld worden zodat het tunnels en DVMRP via /etc/mrouted.conf aanmaakt. Kijk voor meer details over multicast-instellingen in de hulppagina voor mrouted(8).



De mrouted(8) multicast-routeer-daemon implementeert het multicast-routeer-protocol DVRMP welke in veel multicast-installaties grotendeels is vervangen door pim(4). mrouted(8) en de gerelateerde map-mbone(8) en mrinfo(8) gereedschappen zijn beschikbaar in de FreeBSD Ports Collectie als net/mrouted.

29.3. Draadloze netwerken

29.3.1. De beginselen van draadloos netwerken

De meeste draadloze netwerken zijn op de IEEE® 802.11 standaarden gebaseerd. Een eenvoudig draadloos netwerk bestaat uit meerdere stations die met radio's communiceren die in de 2,4GHz of de 5GHz band uitzenden (alhoewel dit regionaal varieert en het ook verandert om communicatie in de 2,3GHz en de 4,9GHz banden mogelijk te maken).

802.11-netwerken zijn op twee manieren georganiseerd: in *infrastructuurmodus* treedt één station als meester op, alle andere stations associëren met dit station; dit netwerk staat bekend als een BSS

en het meesterstation heet een toegangspunt (AP). In een BSS gaat alle communicatie via het AP; zelfs als een station met een ander draadloos station wil communiceren gaan de boodschappen door het AP. In de tweede netwerkvorm is er geen meester en communiceren de stations direct. Deze netwerkvorm is een IBSS en staat gewoonlijk bekend als een *ad-hoc netwerk*.

802.11 netwerken begonnen in de 2,4GHz band waarbij gebruik werd gemaakt van protocollen die door de IEEE® 802.11 en 802.11b standaarden worden gedefinieerd. Deze specificaties omvatten de werkfrequenties, karakteristieken van de MAC-lagen waaronder frame- en zendsnelheden (communicatie kan met verschillende snelheden plaatsvinden). Later definieerde de 802.11a-standaard het werken in de 5GHz band, inclusief andere mechanismen voor signalering en hogere zendsnelheden. Nog later werd de 802.11g-standaard gedefinieerd om gebruik te kunnen maken van de signalerings- en zendmechanismen van 802.11a in de 2,4GHz band zodanig dat het met terugwerkende kracht werkt op 802.11b-netwerken.

Afgezien van de onderliggende zendtechnieken beschikken 802.11-netwerken over een verscheidenheid aan beveiligingstechnieken. De originele 802.11-specificaties definieerden een eenvoudig beveiligingsprotocol genaamd WEP. Dit protocol maakt gebruik van een vaste, van te voren gedeelde sleutel en het cryptografische algoritme RC4 om de gegevens die over het netwerk verstuurd worden te coderen. Alle stations dienen dezelfde sleutel te gebruiken om te kunnen communiceren. Het is bewezen dat dit mechanisme eenvoudig te kraken is en wordt nu, afgezien om voorbijgaande gebruikers te ontmoedigen het netwerk te gebruiken, nog zelden gebruikt. De huidige beveiligingsmethoden worden gegeven door de IEEE® 802.11i specificatie dat nieuwe cryptografische algoritmen en een aanvullend protocol om stations aan een toegangspunt te authenticeren en om sleutels voor gegevenscommunicatie uit te wisselen definieert. Verder worden cryptografische sleutels periodiek ververst en zijn er mechanismen om indringpogingen te detecteren (en om indringpogingen tegen te gaan). Een andere specificatie van een veelgebruikt beveiligingsprotocol in draadloze netwerken is WPA. Dit was een voorloper op 802.11i en gedefinieerd door een industriegroep als een tussenmaatregel terwijl er gewacht werd op de ratificatie van 802.11i. WPA specificeert een deel van de eisen van 802.11i en is ontworpen voor implementatie op verouderde hardware. In het bijzonder vereist WPA alleen de TKIP-sleutel die van de originele WEP-sleutel is afgeleid. 802.11i staat het gebruik van TKIP toe maar vereist ook ondersteuning voor een sterkere sleutel, AES-CCM, om gegevens te versleutelen. (De AES-sleutel was niet nodig in WPA omdat het rekenkundig te kostbaar werd geacht voor implementatie op verouderde hardware.)

Afgezien van de bovenstaande protocolstandaarden is de andere belangrijke standaard waarvan bewustzijn belangrijk is 802.11e. Deze standaard definieert het opstellen van multimediatoepassingen zoals gestroomde video en voice over IP (VoIP) binnen een 802.11-netwerk. Net als 802.11i heeft ook 802.11e een voorgaande specificatie genaamd WME (later hernoemd tot WMM) die door een industriegroep is gedefinieerd als een deelverzameling van 802.11e die nu kan worden gebruikt om multimediatoepassingen mogelijk te maken terwijl er gewacht wordt op de uiteindelijke ratificatie van 802.11e. Het belangrijkste om over 802.11e en WME/WMM te weten is dat ze gepriotiseerd verkeersgebruik van een draadloos netwerk mogelijk maken door middel van Quality of Service (QoS) protocollen en protocollen voor verbeterde mediatoegang. Een juiste implementatie van deze protocollen maken snelle gegevensbursts en gepriotiseerde verkeersstromen mogelijk.

FreeBSD ondersteunt netwerken die met 802.11a, 802.11b, en 802.11g werken. Ook worden de veiligheidsprotocollen WPA en 802.11i ondersteund (samen met 11a, 11b, of 11g) en QoS en de

verkeersprioritisatieprotocollen die nodig zijn voor de protocollen WME/WMM worden voor een beperkte verzameling draadloze apparatuur ondersteund.

29.3.2. Basisinstallatie

Kernelinstellingen

Om van een draadloos netwerk gebruik te maken is het nodig om een draadloze netwerkkaart te hebben en om de kernel met de juiste ondersteuning voor draadloze netwerken in te stellen. Het laatste is verdeeld in meerdere modulen zodat alleen de software ingesteld hoeft te worden die daadwerkelijk gebruikt zal worden.

Ten eerste is een draadloos netwerkapparaat nodig. De meestgebruikte apparaten zijn degenen die onderdelen van Atheros gebruiken. Deze apparaten worden ondersteund door het stuurprogramma ath(4) en voor hen dient de volgende regel aan /boot/loader.conf toegevoegd te worden:

```
if_ath_load="YES"
```

Het stuurprogramma voor Atheros is opgedeeld in drie verschillende delen: het eigenlijke stuurprogramma (ath(4)), de ondersteuningslaag voor de hardware die chip-specifieke functies afhandelt (ath_hal(4)), en een algoritme om de snelheid om frames te verzenden te kiezen uit een reeks mogelijke waarden (hier ath_rate_sample). Indien deze ondersteuning als kernelmodules wordt geladen, zullen de afhankelijkheden automatisch afgehandeld worden. Voor andere apparaten dan die van Atheros dient de module voor dat stuurprogramma geladen te worden; bijvoorbeeld:

```
if_wi_load="YES"
```

voor apparaten die op onderdelen van Intersil Prism zijn gebaseerd (stuurprogramma wi(4)).



In de rest van dit document zal een ath(4) apparaat gebruikt worden, de naam van het apparaat in de voorbeelden dient aangepast te worden aan de lokale installatie. Een lijst van beschikbare draadloze stuurprogramma's en ondersteunde adapters staat in de FreeBSD Hardware Notes. Kopieën hiervan voor verschillende uitgaven en architecturen zijn beschikbaar op de Uitgave Informatie pagina van de FreeBSD website. Indien er geen origineel stuurprogramma voor het draadloze apparaat bestaat, is het mogelijk om te proberen om direct het stuurprogramma van Windows® proberen te gebruiken met behulp van de stuurprogramma-wrapper NDIS.

Daarvoor zijn ook de modules nodig die cryptografische ondersteuning implementeren voor de te gebruiken veiligheidsprotocollen. Het is de bedoeling dat ze dynamisch door de module wlan(4) worden geladen maar momenteel dienen ze handmatig ingesteld te worden. De volgende modules zijn beschikbaar: wlan_wep(4), wlan_ccmp(4), en wlan_tkip(4). Zowel de stuurprogramma's wlan_ccmp(4) en wlan_tkip(4) zijn alleen nodig indien het veiligheidsprotocol WPA en/of 802.11i

gebruikt wordt. Indien het netwerk encryptieloos dient te zijn, is de ondersteuning van wlan_wep(4) niet nodig. Om deze modules tijdens het opstarten te laden, dienen de volgende regels aan /boot/loader.conf toegevoegd te worden:

```
wlan_wep_load="YES"
wlan_ccmp_load="YES"
wlan_tkip_load="YES"
```

Nadat deze informatie aan het instellingenbestand om het systeem op te starten (i.e., /boot/loader.conf) is toegevoegd, is het noodzakelijk om de FreeBSD-computer opnieuw op te starten. Indien het ongewenst is om de computer nu opnieuw op te starten, kunnen de modules ook handmatig worden geladen door kldload(8) te gebruiken.



Indien het gebruik van modules ongewenst is, kunnen deze stuurprogramma's in de kernel worden gecompileerd door de volgende regels aan het kernelinstellingenbestand toe te voegen:

```
device wlan              # 802.11 ondersteuning
device wlan_wep          # 802.11 WEP-ondersteuning
device wlan_ccmp         # 802.11 CCMP-ondersteuning
device wlan_tkip         # 802.11 TKIP-ondersteuning
device wlan_amrr     # AMRR controle-algoritme voor zendsnelheid
device ath           # Atheros PCI/Cardbus netwerkkaarten
device ath_hal       # Ondersteuning voor PCI/cardbus chips
options AH_SUPPORT_AR5146 # zet AR5146 tx/rx descriptors aan
device ath_rate_sample   # SampleRate verzendsnelheid-controle voor ath
```

Met deze informatie in het kernelinstellingenbestand kan de kernel opnieuw gecompileerd en de FreeBSD-computer opnieuw opgestart worden.

Wanneer het systeem draait, is het mogelijk om enige informatie over de draadloze apparaten in de opstartboodschappen te vinden, zoals:

```
ath0: <Atheros 5212> mem 0x88000000-0x8800ffff irq 11 at device 0.0 on cardbus1
ath0: [ITHREAD]
ath0: AR2413 mac 7.9 RF2413 phy 4.5
```

29.3.3. Infrastructuurmodus

De infrastructuur- of BSS-modus is de modus die normaliter gebruikt wordt. In deze modus zijn een aantal draadloze toegangspunten verbonden met een bedraad netwerk. Elk draadloos netwerk heeft een eigen naam, deze naam wordt de SSID van het netwerk genoemd. Draadloze cliënten verbinden zich met de draadloze toegangspunten.

FreeBSD cliënten

Hoe toegangspunten te vinden

Voor het scannen van netwerken wordt het commando `ifconfig` gebruikt. Het kan even duren voordat dit verzoek is afgehandeld aangezien het systeem op elke beschikbare draadloze frequentie naar toegangspunten moet zoeken. Alleen de super-gebruiker kan zo'n scan opzetten:

```
# ifconfig wlan0 create wlandev ath0
# ifconfig wlan0 up scan
SSID/MESH ID  BSSID              CHAN RATE   S:N     INT CAPS
dlinkap       00:13:46:49:41:76   11   54M -90:96 100 EPS  WPA WME
freebsdap     00:11:95:c3:0d:ac    1   54M -83:96 100 EPS  WPA
```



De interface dient als up te worden gemarkeerd voordat het scannen begint. Voor verdere scans is het niet nodig om de interface als up te markeren.

De uitvoer van een scanverzoek vermeld elk gevonden BSS/IBSS-netwerk. Naast de naam van het netwerk, `SSID`, staat het `BSSID`, wat het MAC-adres van het toegangspunt is. Het veld `CAPS` identificeert het type van elk netwerk en de mogelijkheden van de stations die daar werkzaam zijn:

Tabel 7. Station Capability Codes

Capability Code	Betekenis
`E`	Uitgebreide dienstenverzameling (ESS). Geeft aan dat het station deel uitmaakt van een infrastructuurnetwerk (in tegenstelling tot een IBSS-/ ad-hoc-netwerk).
`I`	IBSS-/ad-hoc-netwerk. Geeft aan dat het station deel uitmaakt van een ad-hoc-netwerk (in tegenstelling tot een ESS-netwerk).
`P`	Privacy. Vertrouwelijkheid is vereist voor alle gegevensframes die binnen het BSS worden uitgewisseld. Dit betekent dat dit BSS eist dat het station cryptografische middelen als WEP, TKIP of AES-CCMP dient te gebruiken om de gegevensframes die met anderen worden uitgewisseld te versleutelen en te ontsleutelen.
`S`	Korte preambule. Geeft aan dat het netwerk korte preambules gebruikt (gedefinieerd in 802.11b Hoge Snelheid/DSSS PHY, korte preambule gebruikt een 56-bits synchronisatieveld in tegenstelling tot een 128-bits dat bij lange preambules wordt gebruikt).

Capability Code	Betekenis
`S`	Korte slottijd. Geeft aan dat het 802.11g-netwerk een korte slottijd gebruikt omdat er geen verouderde (802.11b) stations aanwezig zijn.

Het is ook mogelijk om de huidige lijst van bekende netwerken weer te geven met:

```
# ifconfig scan0 list scan
```

Deze informatie kan automatisch bijgewerkt worden door de adapter of handmatig met een `scan` verzoek. Oude gegevens worden automatisch uit de cache verwijderd, dus kan deze lijst na verloop van tijd korter worden tenzij er meer scanverzoeken gedaan worden.

Basisinstellingen

Deze sectie geeft een eenvoudig voorbeeld hoe de draadloze netwerkadapter in FreeBSD zonder encryptie aan de praat te krijgen. Nadat deze concepten bekend zijn, wordt het sterk aangeraden om WPA te gebruiken om de draadloze netwerken op te zetten.

Er zijn drie basisstappen om een draadloos netwerk in te stellen: een toegangspunt kiezen, het station authenticeren, en een IP-adres instellen. De volgende secties behandelen elk een stap.

Een toegangspunt kiezen

In de meeste gevallen is het voldoende om het systeem een toegangspunt gebaseerd op de ingebouwde heuristieken te laten kiezen. Dit is het standaardgedrag wanneer een interface als up wordt gemarkeerd of als een interface wordt ingesteld door het te noemen in /etc/rc.conf, bijvoorbeeld:

```
wlans_ath0="wlan0"
ifconfig_wlan0="DHCP"
```

Indien er meerdere toegangspunten zijn en het gewenst is om een specifieke te kiezen, kan dit met het SSID:

```
wlans_ath0="wlan0"
ifconfig_wlan0="ssid uw_ssid_hier DHCP"
```

In een omgeving waar meerdere toegangspunten hetzelfde SSID hebben (vaak gedaan om roamen eenvoudiger te maken) kan het nodig zijn om met één specifiek apparaat te associëren. In dit geval kan ook het BSSID van het toegangspunt gespecificeerd worden (het SSID kan ook weggelaten worden):

```
wlans_ath0="wlan0"
ifconfig_wlan0="ssid uw_ssid_hier bssid xx:xx:xx:xx:xx:xx DHCP
```

Er zijn andere manieren om de keuze van een toegangspunt te beperken zoals het beperken van het aantal frequenties waarop het systeem scant. Dit kan handig zijn bij multi-band-netwerkkaarten aangezien het scannen van alle mogelijke kanalen tijdrovend kan zijn. Om de werking tot een specifieke band te beperken kan de parameter `mode` gebruikt worden; bijvoorbeeld:

```
wlans_ath0="wlan0"
ifconfig_wlan0="mode 11g ssid uw_ssid_hier DHCP"
```

zal de kaart forceren om te werken in 802.11g welke alleen voor 2,4GHz frequenties is gedefinieerd dus de 5GHz kanalen blijven buiten beschouwing. Andere manieren om dit te doen zijn de parameter `channel`, om bewerkingen op één specifieke frequentie vast te zetten, en de parameter `chanlist`, om een lijst van te scannen kanalen te specificeren. Meer informatie over deze parameters kan in de hulppagina ifconfig(8) gevonden worden.

Authenticatie

Nadat er een toegangspunt is gekozen moet het station zich authenticeren voordat het gegevens kan versturen. Authenticatie kan op verschillende manieren gebeuren. Het meest gebruikte schema wordt open authenticatie genoemd en staat doe dat elk station aan het netwerk deelneemt en communiceert. Deze manier van authenticatie dient gebruikt te worden voor testdoeleinden tijdens het voor de eerste keer opzetten van een draadloos netwerk. Andere schema's vereisen dat cryptografische overeenkomsten voltooid worden voordat gegevensverkeer kan stromen; ofwel door vooraf gedeelde sleutels of geheimen te gebruiken, of door complexere schema's te gebruiken welke achterliggende diensten zoals RADIUS betrekken. De meeste gebruikers zullen open authenticatie gebruiken welke de standaardinstelling is. De dan meest voorkomende opstelling is WPA-PSK, ook bekend als WPA Personal, welke hieronder beschreven is.



Indien er een Apple® AirPort® Extreme basisstation als toegangspunt wordt gebruikt kan het nodig zijn om gedeelde-sleutel-authenticatie samen met een WEP-sleutel in te stellen. Dit kan gedaan worden in het bestand /etc/rc.conf of door het programma wpa_supplicant(8) te gebruiken. Indien er een enkel AirPort® basisstation wordt gebruikt kan de toegang met zoiets als het volgende worden ingesteld:

```
wlans_ath0="wlan0"
ifconfig_wlan0="authmode shared wepmode on weptxkey 1 wepkey 01234567
DHCP"
```

Over het algemeen dient authenticatie via gedeelde sleutels worden voorkomen omdat het materiaal van de WEP-sleutel op een zeer afgedwongen manier gebruikt wordt wat het zelfs gemakkelijker maakt om de sleutel te kraken. Indien WEP gebruikt moet worden (bijvoorbeeld voor compatibiliteit met verouderde apparaten) is het beter om WEP met `open` authenticatie te gebruiken. Meer informatie met betrekking tot WEP kan gevonden worden in WEP.

Een IP-adres verkrijgen met DHCP

Nadat het toegangspunt is gekozen en de parameters voor de authenticatie zijn ingesteld, dient er een IP-adres ter communicatie verkregen worden. In de meeste gevallen wordt het draadloze IP-adres verkregen via DHCP. Om dat te bereiken, dient /etc/rc.conf bewerkt te worden en `DHCP` aan de instellingen voor het apparaat toegevoegd te worden zoals in de verschillende bovenstaande voorbeelden is laten zien:

```
wlans_ath0="wlan0"
ifconfig_wlan0="DHCP"
```

Op dit moment kan de draadloze interface geactiveerd worden:

```
# service netif start
```

Wanneer de interface draait, kan `ifconfig` gebruikt worden om de status van de interface ath0 te zien:

```
# ifconfig wlan0
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether 00:11:95:d5:43:62
    inet 192.168.1.00 netmask 0xffffff00 broadcast 192.168.1.255
        media: IEEE 802.11 Wireless Ethernet OFDM/54Mbps mode 11g
        status: associated
        ssid dlinkap channel 11 (2462 Mhz 11g) bssid 00:13:46:49:41:76
    country US ecm authmode OPEN privacy OFF txpower 21.5 bmiss 7
    scanvalid 60 bgscan bgscanintvl 300 bgscanidle 250 roam:rssi 7
    roam:rate 5 protmode CTS wme burst
```

Het `status: associated` betekent dat er verbinding is met een draadloos netwerk (in dit geval met het netwerk `dlinkap`). Het gedeelte `bssid 00:13:46:49:41:76` is het MAC-adres van het toegangspunt; de gedeelte met `authmode` vertelt dat de communicatie niet versleuteld is.

Statisch IP-adres

In het geval dat het niet mogelijk is om een IP-adres van een DHCP-server te krijgen, kan er een vast IP-adres worden ingesteld. Vervang het sleutelwoord `DHCP` van hierboven met de adresinformatie. Zorg ervoor dat de andere parameters voor het selecteren van een toegangspunt behouden blijven:

```
wlans_ath0="wlan0"
ifconfig_wlan0="inet 192.168.1.100 netmask 255.255.255.0 ssid uw_ssid_hier"
```

WPA

WPA (Wi-Fi Protected Access) is een beveiligingsprotocol dat samen met 802.11-netwerken wordt gebruikt om het gebrek aan degelijke authenticatie en de zwakte van WEP te benadrukken. WPA

verbetert het 802.1X-authenticatieprotocol en gebruikt een sleutel gekozen uit meerdere in plaats van WEP voor gegevensintegriteit. De enige sleutel welke WPA vereist is TKIP (Temporary Key Integrity Protocol). TKIP is een sleutel dat de basis-RC4-sleutel welke door WEP wordt gebruikt uitbreidt door integriteitscontroles, knoeidetectie, en maatregelen om op elke gedetecteerde inbraak te reageren toe te voegen. TKIP is ontworpen om op verouderde hardware met enkel wijzigingen in software te draaien; het representeert een compromis dat de veiligheid verbetert maar nog steeds niet geheel immuun is tegen aanvallen. WPA specificeert ook de sleutel AES-CCMP als een alternatief voor TKIP welke te verkiezen is indien mogelijk; voor deze specificatie wordt gewoonlijk de term WPA2 (of RSN) gebruikt.

WPA definieert protocollen voor authenticatie en versleuteling. Authenticatie gebeurt het meeste door één van deze twee technieken te gebruiken: door 802.1X en een achterliggende authenticatiedienst zoals RADIUS, of door een minimale overeenkomst tussen het station en het toegangspunt door een van te voren gedeeld geheim te gebruiken. Het eerste wordt vaak WPA Enterprise genoemd en het laatste staat bekend als WPA Personal. Aangezien de meeste mensen geen achterliggende RADIUS-server voor hun draadloos netwerk zullen opzetten, is WPA-PSK veruit de meest gebruikte configuratie voor WPA.

Het beheer van de draadloze verbinding en de authenticatie (sleutelonderhandeling of authenticatie met een server) gebeurt met het gereedschap wpa_supplicant(8). Dit programma vereist dat er een instellingenbestand, /etc/wpa_supplicant.conf, draait. Meer informatie over dit bestand kan in de hulppagina wpa_supplicant.conf(5) worden gevonden.

WPA-PSK

WPA-PSK, ook bekend als WPA-Personal, is gebaseerd op een vooraf gedeelde sleutel (PSK) gegenereerd vanuit een gegeven wachtwoord die gebruikt zal worden als de hoofdsleutel in het draadloze netwerk. Dit betekent dat alle draadloze gebruikers dezelfde sleutel zullen delen. WPA-PSK is bedoeld voor kleine netwerken waar het gebruik van een authenticatieserver niet mogelijk of gewenst is.



Gebruik altijd sterke wachtwoorden welke voldoende lang zijn en opgebouwd zijn uit een grote tekenverzameling zodat ze niet gemakkelijk worden geraden of aangevallen.

De eerste stap is het instellen van het bestand /etc/wpa_supplicant.conf met het SSID en de vooraf gedeelde sleutel van het netwerk:

```
network={
  ssid="freebsdap"
  psk="freebsdmall"
}
```

Daarna zal in /etc/rc.conf worden aangegeven dat de draadloze configuratie met WPA zal gebeuren en dat het IP-adres met DHCP zal worden verkregen:

```
wlans_ath0="wlan0"
```

```
ifconfig_wlan0="WPA DHCP"
```

Hierna kan de interface geactiveerd worden:

```
# service netif start
Starting wpa_supplicant.
DHCPDISCOVER on wlan0 to 255.255.255.255 port 67 interval 5
DHCPDISCOVER on wlan0 to 255.255.255.255 port 67 interval 6
DHCPOFFER from 192.168.0.1
DHCPREQUEST on wlan0 to 255.255.255.255 port 67
DHCPACK from 192.168.0.1
bound to 192.168.0.254 -- renewal in 300 seconds.
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet OFDM/36Mbps mode 11g
      status: associated
      ssid freebsdap channel 1 (2412 Mhz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
      AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
      bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
      wme burst roaming MANUAL
```

Ook kan gepoogd worden dit handmatig in te stellen door hetzelfde /etc/wpa_supplicant.conf als hierboven te gebruiken, en dit te draaien:

```
# wpa_supplicant -i wlan0 -c /etc/wpa_supplicant.conf
Trying to associate with 00:11:95:c3:0d:ac (SSID='freebsdap' freq=2412 MHz)
Associated with 00:11:95:c3:0d:ac
WPA: Key negotiation completed with 00:11:95:c3:0d:ac [PTK=CCMP GTK=CCMP]
CTRL-EVENT-CONNECTED - Connection to 00:11:95:c3:0d:ac completed (auth) [id=0 idstr=]
```

De volgende stap is het lanceren van het commando `dhclient` om een IP-adres van de DHCP-server te krijgen:

```
# dhclient wlan0
DHCPREQUEST on wlan0 to 255.255.255.255 port 67
DHCPACK from 192.168.0.1
bound to 192.168.0.254 -- renewal in 300 seconds.
# ifconfig wlan0
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet OFDM/36Mbps mode 11g
      status: associated
      ssid freebsdap channel 1 (2412 Mhz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode WPA2/802.11i privacy ON defxkey UNDEF
```

```
        AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
        bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
        wme burst roaming MANUAL
```



/etc/rc.conf heeft een regel `ifconfig_wlan0` met de tekst `DHCP` (zoals `ifconfig_wlan0="DHCP"`), `dhclient` zal automatisch gestart worden nadat `wpa_supplicant` geassocieerd is met het toegangspunt.

Als DHCP niet mogelijk of gewenst is, kan een statisch IP-adres worden ingesteld nadat `wpa_supplicant` het station heeft geauthenticeerd:

```
# ifconfig wlan0 inet 192.168.0.100 netmask 255.255.255.0
# ifconfig wlan0
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether 00:11:95:d5:43:62
        inet 192.168.0.100 netmask 0xffffff00 broadcast 192.168.0.255
        media: IEEE 802.11 Wireless Ethernet OFDM/36Mbps mode 11g
        status: associated
        ssid freebsdap channel 1 (2412 Mhz 11g) bssid 00:11:95:c3:0d:ac
        country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
        AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
        bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
        wme burst roaming MANUAL
```

Indien DHCP niet wordt gebruikt, dienen ook de standaard gateway en de naamserver handmatig ingesteld te worden:

```
# route add default uw_standaard_router
# echo "nameserver uw_DNS_server" >> /etc/resolv.conf
```

WPA met EAP-TLS

De tweede manier om WPA te gebruiken is met een achterliggende 802.1X-authenticatieserver. In dit geval wordt het WPA-Enterprise genoemd om het verschil met het minder veilige WPA-Personal met de vooraf gedeelde sleutel aan te duiden. Authenticatie is in WPA-Enterprise gebaseerd op EAP (Extensible Authentication Protocol).

EAP wordt niet met een encryptiemethode geleverd. In plaats daarvan was het besloten om EAP in een versleutelde tunnel te omsluiten. Er bestaan vele EAP-authenticatiemethodes, de meest voorkomende zijn EAP-TLS, EAP-TTLS, en EAP-PEAP.

EAP-TLS (EAP met Transport Layer Security) is een zeer goed ondersteund authenticatieprotocol in de draadloze wereld aangezien het de eerste EAP-methode was die gecertificeerd werd door de Wi-Fi alliantie. EAP-TLS vereist dat er drie certificaten draaien: het CA-certificaat (geïnstalleerd op alle machines), het servercertificaat voor de authenticatieserver, en een cliëntcertificaat voor elke draadloze cliënt. Bij deze EAP-methode authenticeren zowel de authenticatieserver als de draadloze cliënt elkaar door hun respectievelijke certificaten te laten zien, en ze controleren dat

deze certificaten zijn getekend door de certificatenauthoriteit (CA) van de organisatie.

Zoals voorheen gebeurt het instellen via /etc/wpa_supplicant.conf:

```
network={
  ssid="freebsdap" ①
  proto=RSN  ②
  key_mgmt=WPA-EAP ③
  eap=TLS ④
  identity="loader" ⑤
  ca_cert="/etc/certs/cacert.pem" ⑥
  client_cert="/etc/certs/clientcert.pem" ⑦
  private_key="/etc/certs/clientkey.pem" ⑧
  private_key_passwd="freebsdmallclient" ⑨
}
```

① Dit veld geeft de naam van het netwerk (SSID) aan.

② Hier wordt het RSN (IEEE® 802.11i) protocol gebruikt, ofwel WPA2.

③ De regel `key_mgmt` verwijst naar het gebruikte sleutelbeheerprotocol. In dit geval is het WPA dat EAP-authenticatie gebruikt: `WPA-EAP`.

④ In dit veld wordt de EAP-methode voor de verbinding genoemd.

⑤ Het veld `identity` bevat de identiteitsstring voor EAP.

⑥ Het veld `ca_cert` geeft de padnaam van het CA-certificaatbestand aan. Dit bestand is nodig om het servercertificaat te controleren.

⑦ De regel `client_cert` geeft de padnaam van het cliëntcertificaatbestand aan. Dit certificaat is uniek voor elke draadloze cliënt van het netwerk.

⑧ Het veld `private_key` is de padnaam naar het bestand dat de privésleutel van het cliëntcertificaat bevat.

⑨ Het veld `private_key_passwd` bevat het wachtwoord voor de privésleutel.

Voeg vervolgens de volgende regels toe aan /etc/rc.conf:

```
wlans_ath0="wlan0"
ifconfig_wlan0="WPA DHCP"
```

De volgende stap is het activeren van de interface:

```
# service netif start
Starting wpa_supplicant.
DHCPREQUEST on wlan0 to 255.255.255.255 port 67
DHCPREQUEST on wlan0 to 255.255.255.255 port 67
DHCPACK from 192.168.0.20
bound to 192.168.0.254 -- renewal in 300 seconds.
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 00:11:95:d5:43:62
```

```
        inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
        media: IEEE 802.11 Wireless Ethernet DS/11Mbps mode 11g
        status: associated
        ssid freebsdap channel 1 (2412 Mhz 11g) bssid 00:11:95:c3:0d:ac
        country US ecm authmode WPA/802.11i privacy ON deftxkey UNDEF
        AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
        bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
        wme burst roaming MANUAL
```

Zoals eerder is laten zien, is het ook mogelijk om de interface handmatig te activeren met zowel de commando's `wpa_supplicant` en `ifconfig`.

WPA met EAP-TTLS

Bij EAP-TLS hebben zowel de authenticatieserver als de cliënt een certificaat nodig, met EAP-TTLS (EAP-Tunneled Transport Layer Security) is een cliëntcertificaat optioneel. Deze methode komt in de buurt van wat sommige beveiligde websites doen, waar de webserver een veilige SSL-tunnel kan aanmaken zelfs als de bezoekers geen certificaten aan de cliëntkant hebben. EAP-TTLS zal de versleutelde TLS-tunnel gebruiken voor het veilig transporteren van de authenticatiegegevens.

De instellingen worden gedaan via het bestand /etc/wpa_supplicant.conf:

```
network={
  ssid="freebsdap"
  proto=RSN
  key_mgmt=WPA-EAP
  eap=TTLS ①
  identity="test" ②
  password="test" ③
  ca_cert="/etc/certs/cacert.pem" ④
  phase2="auth=MD5" ⑤
}
```

① Dit veld noemt de EAP-methode voor de verbinding.

② Het veld `identity` bevat de identiteitsstring voor EAP-authenticatie binnen de versleutelde TLS-tunnel.

③ Het veld `password` bevat het wachtwoord voor de EAP-authenticatie.

④ Het veld `ca_cert` wijst naar de padnaam van het CA-certificaatbestand. Dit bestand is nodig om het servercertificaat te controleren.

⑤ Dit veld noemt de gebruikte authenticatiemethode in de versleutelde TLS-tunnel. In dit geval is EAP met MD5-Challenge gebruikt. De "binnenste authenticatie"-fase wordt vaak "phase2" genoemd.

Ook dienen de volgende regels toegevoegd te worden aan /etc/rc.conf:

```
wlans_ath0="wlan0"
```

```
ifconfig_ath0="WPA DHCP"
```

De volgende stap is het activeren van de interface:

```
# service netif start
Starting wpa_supplicant.
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 7
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 15
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 21
DHCPACK from 192.168.0.20
bound to 192.168.0.254 -- renewal in 300 seconds.
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet autoselect DS/11Mbps mode 11g
      status: associated
      ssid freebsdap channel 1 (2412 Mhz) bssid 00:11:95:c3:0d:ac
      country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
      AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
      bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
      wme burst roaming MANUAL
```

WPA met EAP-PEAP



PEAPv0/EAP-MSCHAPv2 is de meest gebruikelijke PEAP-methode. In de rest van dit document wordt de term PEAP gebruikt om naar die methode te verwijzen.

PEAP (Beveiligd EAP) is ontworpen als een alternatief voor EAP-TTLS, en is de meest gebruikte EAP-standaard na EAP-TLS. Met andere woorden, als u een netwerk met verschillende besturingssystemen heeft, zou PEAP de meest ondersteunde standaard moeten zijn na EAP-TLS.

PEAP is soortgelijk aan EAP-TTLS: het gebruikt een server-side certificaat om de cliënten te authenticeren door een beveiligde TLS-tunnel tussen de cliënt en de authenticatieserver aan te maken, welke de uitwisseling van de authenticatie-informatie beschermt. Vanuit een beveiligingsoogpunt gezien is het verschil tussen EAP-TTLS en PEAP dat PEAP-authenticatie de gebruikersnaam onversleuteld uitzendt, alleen het wachtwoord wordt in de beveiligde TLS-tunnel verzonden. EAP-TTLS gebruikt de TLS-tunnel voor zowel de gebruikersnaam als het wachtwoord.

Het bestand /etc/wpa_supplicant.conf dient gewijzigd te worden om de EAP-PEAP-gerelateerde instellingen toe te voegen:

```
network={
  ssid="freebsdap"
  proto=RSN
  key_mgmt=WPA-EAP
  eap=PEAP ①
  identity="test" ②
  password="test" ③
```

```
  ca_cert="/etc/certs/cacert.pem" ④
  phase1="peaplabel=0" ⑤
  phase2="auth=MSCHAPV2" ⑥
}
```

① Dit veld noemt de EAP-methode voor de verbinding.

② Het veld `identity` bevat de identiteitsstring voor EAP-authenticatie binnen de versleutelde TLS-tunnel.

③ Het veld `password` bevat het wachtwoord voor de EAP-authenticatie.

④ Het veld `ca_cert` wijst naar de padnaam van het CA-certificaatbestand. Dit bestand is nodig om het servercertificaat te controleren.

⑤ Dit veld bevat de parameters voor de eerste fase van authenticatie (de TLS-tunnel). Afhankelijk van de gebruikte authenticatieserver moet er een specifiek label voor authenticatie worden opgegeven. In de meeste gevallen zal het label "client EAP encryption" zijn welke ingesteld is door `peaplabel=0` te gebruiken. Meer informatie kan in de hulppagina wpa_supplicant.conf(5) gevonden worden.

⑥ Dit veld noemt het authenticatieprotocol dat in de versleutelde TLS-tunnel gebruikt wordt. In het geval van PEAP is dit `auth=MSCHAPV2`.

Het volgende dient te worden toegevoegd aan /etc/rc.conf:

```
wlans_ath0="wlan0"
ifconfig_wlan0="WPA DHCP"
```

Hierna kan de interface worden geactiveerd:

```
# service netif start
Starting wpa_supplicant.
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 7
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 15
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 21
DHCPACK from 192.168.0.20
bound to 192.168.0.254 -- renewal in 300 seconds.
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet autoselect DS/11Mbps mode 11g
      status: associated
      ssid freebsdap channel 1 (2412 Mhz) bssid 00:11:95:c3:0d:ac
      country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
      AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
      bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
      wme burst roaming MANUAL
```

WEP

WEP (Wired Equivalent Privacy) maakt deel uit van de oorspronkelijke 802.11 standaard. Er is geen authenticatiemechanisme, slechts een zwakke vorm van toegangscontrole, en het is gemakkelijk te kraken.

WEP kan worden opgezet met `ifconfig`:

```
# ifconfig wlan0 create wlandev ath0
# ifconfig wlan0 inet 192.168.1.100 netmask 255.255.255.0 \
        ssid mijn_net wepmode on weptxkey 3 wepkey 3:0x3456789012
```

- De `weptxkey` geeft aan welke WEP-sleutel zal worden gebruikt tijdens het verzenden. Hier wordt de derde sleutel gebruikt. Dit dient overeen te komen met de instelling in het toegangspunt. Probeer, indien onbekend is welke sleutel door het toegangspunt wordt gebruikt, `1` (i.e., de eerste sleutel) voor deze waarde te gebruiken.
- De `wepkey` selecteert één van de WEP-sleutels in. Het dient in het formaat *index:sleutel* te zijn. Sleutel `1` wordt als standaard gebruikt; de index hoeft alleen ingesteld te worden als we een andere dan de eerste sleutel gebruiken.



De `0x3456789012` dient vervangen te worden door de sleutel die ingesteld is voor gebruik met het toegangspunt.

Het wordt aangeraden om de hulppagina ifconfig(8) te lezen voor verdere informatie.

De faciliteit `wpa_supplicant` kan ook gebruikt worden om de draadloze interface in te stellen voor WEP. Het bovenstaande voorbeeld kan worden ingesteld door de volgende regels toe te voegen aan /etc/wpa_supplicant.conf:

```
network={
  ssid="mijn_net"
  key_mgmt=NONE
  wep_key3=3456789012
  wep_tx_keyidx=3
}
```

Daarna:

```
# wpa_supplicant -i wlan0 -c /etc/wpa_supplicant.conf
Trying to associate with 00:13:46:49:41:76 (SSID='dlinkap' freq=2437 MHz)
Associated with 00:13:46:49:41:76
```

29.3.4. Ad-hoc-modus

IBSS-modus, ook ad-hoc-modus genoemd, is ontworpen voor point-to-point-verbindingen. Om bijvoorbeeld een ad-hoc-netwerk tussen de machine `A` en de machine `B` op te zetten, is het slechts

nodig om twee IP-adressen en een SSID te kiezen.

Op machine A:

```
# ifconfig wlan0 create wlandev ath0 wlanmode adhoc
# ifconfig wlan0 inet 192.168.0.1 netmask 255.255.255.0 ssid freebsdap
  wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
      ether 00:11:95:c3:0d:ac
      inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <adhoc>
      status: running
      ssid freebsdap channel 2 (2417 Mhz 11g) bssid 02:11:95:c3:0d:ac
      country US ecm authmode OPEN privacy OFF txpower 21.5 scanvalid 60
      protmode CTS wme burst
```

De parameter adhoc geeft aan dat de interface in de IBSS-modus draait.

Op B zal het mogelijk moeten zijn om A te detecteren:

```
# ifconfig wlan0 create wlandev ath0 wlanmode adhoc
# ifconfig wlan0 up scan
  SSID/MESH ID    BSSID              CHAN RATE   S:N     INT CAPS
  reebsdap        02:11:95:c3:0d:ac    2   54M -64:-96  100 IS   WME
```

De I in de uitvoer bevestigt dat machine A in ad-hoc-modus verkeert. Het is slechts nodig om B met een ander IP-adres in te stellen:

```
# ifconfig wlan0 inet 192.168.0.2 netmask 255.255.255.0 ssid freebsdap
# ifconfig wlan0 ssid freebsdap mediaopt adhoc inet 192.168.0.2 netmask 255.255.255.0
# ifconfig wlan0
  wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.2 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <adhoc>
      status: running
      ssid freebsdap channel 2 (2417 Mhz 11g) bssid 02:11:95:c3:0d:ac
      country US ecm authmode OPEN privacy OFF txpower 21.5 scanvalid 60
      protmode CTS wme burst
```

Zowel A als B zijn nu klaar om informatie uit te wisselen.

29.3.5. FreeBSD Host Toegangspunten

FreeBSD kan als toegangspunt (AP) functioneren wat de noodzaak om een hardwarematig AP te kopen of een ad-hoc-netwerk te draaien wegneemt. Dit kan bijzonder nuttig zijn indien de FreeBSD-machine als gateway naar een ander netwerk (bijvoorbeeld het Internet) functioneert.

Basisinstellingen

Voordat de FreeBSD-machine als een AP wordt ingesteld, dient de kernel te worden ingesteld met de juiste ondersteuning voor draadloos netwerken voor de draadloze kaart. Ook dient er ondersteuning voor de te gebruiken beveiligingsprotocollen te worden toegevoegd. Meer details staan in Basisinstallatie.



Momenteel staan de NDIS-stuurprogrammawrapper en de stuurprogramma's van Windows® het werken als AP niet toe. Alleen originele draadloze FreeBSD-stuurprogramma's ondersteunen AP-modus.

Wanneer de ondersteuning voor draadloos netwerken is geladen, kan gecontroleerd worden of het draadloze apparaat de hostgebaseerde toegangspuntmodus ondersteunt (ook bekend als hostap-modus):

```
# ifconfig wlan0 create wlandev ath0
# ifconfig wlan0 list caps
drivercaps=6f85edc1<STA,FF,TURBOP,IBSS,HOSTAP,AHDEMO,TXPMGT,SHSLOT,SHPREAMBLE,MONITOR,
MBSS,WPA1,WPA2,BURST,WME,WDS,BGSCAN,TXFRAG>
cryptocaps=1f<WEP,TKIP,AES,AES_CCM,TKIPMIC>
```

Deze uitvoer geeft de mogelijkheden van de kaart weer, het woord HOSTAP bevestigt dat deze draadloze kaart als toegangspunt kan functioneren. Ook worden verschillende ondersteunde versleutelmethoden genoemd: WEP, TKIP, AES, enzovoorts. Deze informatie is belangrijk om te weten welke beveiligingsprotocollen gebruikt kunnen worden op het toegangspunt.

Het draadloze apparaat kan enkel in hostap-modus worden gezet tijdens het creeëren van het netwerk pseudo-device dus een vooraf aangemaakt apparaat moet eerst verwijderd worden:

```
# ifconfig wlan0 destroy
```

waarna deze opnieuw aangemaakt kan worden met de juiste parameters:

```
# ifconfig wlan0 create wlandev ath0 wlanmode hostap
# ifconfig wlan0 inet 192.168.0.1 netmask 255.255.255.0 ssid freebsdap mode 11g
channel 1
```

Gebruik nogmaals ifconfig om de status van de interface wlan0 te zien:

```
# ifconfig wlan0
  wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
      ether 00:11:95:c3:0d:ac
      inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
      status: running
      ssid freebsdap channel 1 (2412 Mhz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode OPEN privacy OFF txpower 21.5 scanvalid 60
```

```
      protmode CTS wme burst dtimperiod 1 -dfs
```

De parameter `hostap` geeft aan dat de interface in hostgebaseerde toegangspuntmodus draait.

Het instellen van de interface kan automatisch tijdens het opstarten gedaan worden door de volgende regels aan /etc/rc.conf toe te voegen:

```
wlans_ath0="wlan0"
create_args_wlan0="wlanmode hostap"
ifconfig_wlan0="inet 192.168.0.1 netmask 255.255.255.0 ssid freebsdap mode 11g channel
1"
```

Hostgebaseerde toegangspunt zonder authenticatie of versleuteling

Hoewel het niet aangeraden wordt om een AP zonder enige vorm van authenticatie of encryptie te draaien, is dit een eenvoudige manier om te controleren of het AP werkt. Deze configuratie is ook belangrijk voor het debuggen van problemen met cliënten.

Nadat het AP is ingesteld als eerder is laten zien, is het mogelijk om van een andere draadloze machine een scan te beginnen om het AP te vinden:

```
# ifconfig wlan0 create wlandev ath0
# ifconfig wlan0 up scan
SSID/MESH ID    BSSID              CHAN RATE   S:N     INT CAPS
freebsdap       00:11:95:c3:0d:ac    1   54M -66:-96  100 ES   WME
```

De cliëntmachine heeft het AP gevonden en kan ermee geassocieerd worden:

```
# ifconfig ath0 ssid freebsdap inet 192.168.0.2 netmask 255.255.255.0
  ifconfig wlan0
  wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.2 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet OFDM/54Mbps mode 11g
      status: associated
      ssid freebsdap channel 1 (2412 Mhz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode OPEN privacy OFF txpower 21.5 bmiss 7
      scanvalid 60 bgscan bgscanintvl 300 bgscanidle 250 roam:rssi 7
      roam:rate 5 protmode CTS wme burst
```

WPA hostgebaseerde toegangspunt

Deze sectie zal zich richten op opzetten van een FreeBSD toegangspunt dat het beveiligingsprotocol WPA gebruikt. Meer details over WPA en het instellen van op WPA gebaseerde draadloze cliënten kan gevonden worden in WPA.

De daemon hostapd wordt gebruikt om cliëntauthenticatie en sleutelbeheer op het toegangspunt

met WPA af te handelen.

In het volgende zullen alle instellingsbewerkingen worden uitgevoerd op de FreeBSD-machine die als AP dienst doet. Wanneer het AP correct werkt, zou hostapd automatisch tijdens het opstarten aangezet moeten worden met de volgende regel in /etc/rc.conf:

```
hostapd_enable="YES"
```

Zorg ervoor dat voordat geprobeerd wordt om hostapd in te stellen, de basisinstellingen die in Basisinstellingen zijn geïntroduceerd zijn uitgevoerd.

WPA-PSK

WPA-PSK is bedoeld voor kleine netwerken waar het gebruik van een achterliggende authenticatieserver niet mogelijk of gewenst is.

Het instellen wordt gedaan in het bestand /etc/hostapd.conf:

```
interface=wlan0 ①
debug=1 ②
ctrl_interface=/var/run/hostapd ③
ctrl_interface_group=wheel ④
ssid=freebsdap ⑤
wpa=1 ⑥
wpa_passphrase=freebsdmall ⑦
wpa_key_mgmt=WPA-PSK ⑧
wpa_pairwise=CCMP TKIP ⑨
```

① Dit veld geeft aan welke draadloze interface voor het toegangspunt wordt gebruikt.

② Dit veld stelt het verbositeitsniveau in dat tijdens het draaien van hostapd wordt gebruikt. Een waarde van 1 vertegenwoordigt het minimale niveau.

③ Het veld `ctrl_interface` geeft de padnaam van de door hostapd gebruikte map om de domeinsocketbestanden voor communicatie met externe programma's zoals hostapd_cli(8) in op te slaan. Hier wordt de standaardwaarde gebruikt.

④ De regel `ctrl_interface_group` stelt de groep in (hier is het de groep `wheel`) die toegang heeft tot de controle interfacebestanden.

⑤ Het veld `wpa` maakt WPA mogelijk en specificeert welk WPA-authenticatieprotocol nodig zal zijn. De waarde 1 stelt het AP in op WPA-PSK.

⑥ Het veld `wpa_passphrase` bevat het ASCII-wachtwoord voor de WPA-authenticatie.

⑦ De regel `wpa_key_mgmt` verwijst naar het gebruikte sleutelbeheerprotocol. In dit geval is dat WPA-PSK.

⑧ Het veld `wpa_pairwise` geeft aan welke versleutelingsalgoritmes door het toegangspunt worden geaccepteerd. Hier worden zowel de versleuteling TKIP (WPA) en CCMP (WPA2) geaccepteerd. De versleuteling CCMP is een alternatief voor TKIP en wordt sterk aangeraden indien mogelijk; TKIP dient alleen gebruikt te worden voor stations die geen CCMP aankunnen.

De volgende stap is het starten van hostapd:

```
# service hostapd forcestart
```

```
# ifconfig wlan0
  wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 2290
      inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255dddd
      inet6 fe80::211:95ff:fec3:dac%ath0 prefixlen 64 scopeid 0x4
      ether 00:11:95:c3:0d:ac
      media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <hostap>
      status: associated
      ssid freebsdap channel 1 bssid 00:11:95:c3:0d:ac
      authmode WPA2/802.11i privacy MIXED deftxkey 2 TKIP 2:128-bit txpowmax 36
protmode CTS dtimperiod 1 bintval 100
```

Het toegangspunt draait nu, de cliënten kunnen er nu mee worden geassocieerd, zie WPA voor meer details. Het is mogelijk om de stations die met het AP geassocieerd zijn te zien door het commando `ifconfig wlan0 list` te gebruiken.

WEP hostgebaseerd toegangspunt

Het wordt niet aangeraden om WEP te gebruiken om een toegangspunt op te zetten aangezien er geen authenticatiemechanisme is en het gemakkelijk is te kraken. Sommige verouderde draadloze kaarten ondersteunen alleen WEP als een beveiligingsprotocol, met deze kaarten is het alleen mogelijk om een AP zonder authenticatie of encryptie of een AP dat het WEP-protocol gebruikt op te zetten.

Het draadloze apparaat kan nu in hostap-modus worden gezet en ingesteld worden met het juiste SSID en IP-adres:

```
# ifconfig wlan0 create wlandev ath0 wlanmode hostap
# ifconfig wlan0 inet 192.168.0.1 netmask 255.255.255.0 \
  ssid freebsdap wepmode on weptxkey 3 wepkey 3:0x3456789012 mode 11g
```

- Het `weptxkey` geeft aan welke WEP-sleutel tijdens het zenden zal worden gebruikt. Hier wordt de derde sleutel gebruikt (merk op dat de nummering van de sleutels bij 1 begint). Deze parameter moet gespecificeerd worden om de gegevens daadwerkelijk te versleutelen.
- Het `wepkey` geeft aan dat de geselecteerde WEP-sleutel wordt ingesteld. Het dient in het formaat *index:key* te zijn, indien de index niet is gegeven, wordt sleutel 1 gebruikt. Dus indien een andere sleutel dan de eerste wordt gebruikt dient de index te worden ingesteld.

Weer wordt `ifconfig` gebruikt om de status van de interface wlan0 te zien:

```
# ifconfig wlan0
  wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
      ether 00:11:95:c3:0d:ac
```

```
        inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
        media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <hostap>
        status: running
        ssid freebsdap channel 4 (2427 Mhz 11g) bssid 00:11:95:c3:0d:ac
        country US ecm authmode OPEN privacy ON deftxkey 3 wepkey 3:40-bit
        txpower 21.5 scanvalid 60 protmode CTS wme burst dtimperiod 1 -dfs
```

Vanaf een andere draadloze machine is het mogelijk om een scan te beginnen om het AP te vinden:

```
# ifconfig wlan0 create wlandev ath0
# ifconfig wlan0 up scan
SSID            BSSID              CHAN RATE  S:N   INT CAPS
freebsdap       00:11:95:c3:0d:ac    1   54M 22:1   100 EPS
```

De cliëntmachine heeft het toegangspunt gevonden en kan ermee geassocieerd worden door de juiste parameters (sleutel, enz.) te gebruiken, zie WEP voor meer details.

29.3.6. Zowel de bekabelde als de draadloze verbinding gebruiken

Een bekabelde verbinding biedt betere prestaties en betrouwbaarheid, terwijl een draadloze verbinding meer flexibiliteit en mobiliteit biedt; laptop-gebruikers zullen dit willen combineren en naadloos tussen de twee overschakelen.

In FreeBSD is het mogelijk om twee of meer netwerkinterfaces te combineren in een "failover"-opstelling, dit houdt in dat de meest geprefereerde en best beschikbare verbinding van een groep van netwerkinterfaces wordt gebruikt, en het besturingssysteem automatisch te laten overschakelen wanneer de status van de verbinding verandert.

Link-aggregatie en failover worden behandeld in Verbindingsaggregatie en failover, een voorbeeld voor het gebruik van zowel een bekabelde als een draadloze verbinding wordt gegeven in Failover-modus tussen bekabelde en draadloze interfaces.

29.3.7. Problemen verhelpen

Indien er problemen met het draadloos netwerk zijn, zijn er een aantal stappen die genomen kunnen worden om het probleem te helpen verhelpen.

- Indien het toegangspunt niet vermeld wordt tijdens het scannen, controleer dan of het draadloze apparaat niet is ingesteld op een beperkt aantal kanalen.
- Indien het niet mogelijk is om met een toegangspunt te associëren, controleer dan of de instellingen van het station overeenkomen met die van het toegangspunt. Dit omvat het authenticatieschema en de beveiligingsprotocollen. Versimpel de configuratie zoveel mogelijk. Indien een beveiligingsprotocol als WPA of WEP wordt gebruikt, stel het toegangspunt dan in voor open authenticatie en geen beveiliging en kijk of er verkeer door kan.
- Wanneer er met het toegangspunt geassocieerd kan worden, stel dan een diagnose over alle beveiligingsinstellingen met eenvoudige gereedschappen zoals ping(8).

`wpa_supplicant` biedt veel ondersteuning voor debuggen; probeer het handmatig te draaien met de optie `-dd` en controleer de systeemlogs.

- Er zijn ook veel debug-gereedschappen op lagere niveaus. Het is mogelijk om debugberichten in de laag die het 802.11 protocol ondersteunt aan te zetten door het programma `wlandebug` te gebruiken dat gevonden wordt in /usr/src/tools/tools/net80211. Bijvoorbeeld:

  ```
  # wlandebug -i ath0 +scan+auth+debug+assoc
    net.wlan.0.debug: 0 => 0xc80000<assoc,auth,scan>
  ```

  kan worden gebruikt om consoleberichten aan te zetten die te maken hebben met het scannen van toegangspunten en het uitvoeren van 802.11 handshakes die nodig zijn om communicatie te regelen.

  Er worden ook veel nuttige statistieken door de 802.11 laag bijgehouden; het gereedschap `wlanstats` geeft deze informatie weer. Deze statistieken zouden alle fouten die door de 802.11 laag zijn geïdentificeerd moeten identificeren. Let erop dat sommige fouten worden geïdentificeerd in de apparaatstuurprogramma's die onder de 802.11 laag liggen zodat ze niet verschijnen. Voor het diagnosticeren van apparaatspecifieke problemen dient de documentatie van het stuurprogramma geraadpleegd te worden.

Indien de bovenstaande informatie niet helpt om het probleem te verhelderen, stuur dan een probleemrapport op inclusief de uitvoer van de bovenstaande gereedschappen.

29.4. Bluetooth

29.4.1. Introductie

Bluetooth is een draadloze technologie om persoonlijke netwerken aan te maken die in de vrije 2,4GHz-band werken binnen een straal van 10 meter. Deze netwerken worden gewoonlijk ad-hoc gevormd en bestaan uit draagbare apparaten zoals mobiele telefoons, handhelds en laptops. In tegenstelling tot die andere populaire draadloze techniek, Wi-Fi, biedt Bluetooth een hoger niveau van serviceprofielen, zoals FTP-achtige bestandsservers, pushing van bestanden, stemtransport, emulatie van seriële lijnen, en meer.

De Bluetooth stack is in FreeBSD geïmplementeerd door gebruik te maken van het Netgraph-raamwerk (zie netgraph(4)). Veel van de Bluetooth USB-dongles worden ondersteund door het stuurprogramma ng_ubt(4). Apparaten gebaseerd op de Broadcom BCM2033 chip worden ondersteund door de stuurprogramma's ubtbcmfw(4) en ng_ubt(4). De 3Com Bluetooth PC Card 3CRWB60-A wordt ondersteund door het stuurprogramma ng_bt3c(4). Seriële en op UART gebaseerde Bluetooth-apparaten worden ondersteund via sio(4), ng_h4(4), en hcseriald(8). Deze sectie beschrijft het gebruik van de USB Bluetooth-dongle.

29.4.2. Het apparaat inprikken

Standaard zijn stuurprogramma's voor Bluetooth-apparaten beschikbaar als kernelmodules. Voordat een apparaat wordt aangekoppeld, dient het stuurprogramma in de kernel geladen te

worden:

```
# kldload ng_ubt
```

Indien het Bluetooth-apparaat tijdens het opstarten van het systeem in het systeem aanwezig is, kan de module vanuit /boot/loader.conf geladen worden:

```
ng_ubt_load="YES"
```

Prik de USB-dongle in. Uitvoer vergelijkbaar aan de onderstaande zal op de console (of in syslog) verschijnen:

```
ubt0: vendor 0x0a12 product 0x0001, rev 1.10/5.25, addr 2
ubt0: Interface 0 endpoints: interrupt=0x81, bulk-in=0x82, bulk-out=0x2
ubt0: Interface 1 (alt.config 5) endpoints: isoc-in=0x83, isoc-out=0x3,
      wMaxPacketSize=49, nframes=6, buffer size=294
```

service(8) wordt gebruikt om de Bluetooth-stack te starten en te stoppen. Het is een goed idee om de stack te stoppen voordat het apparaat wordt losgekoppeld, maar het is (gewoonlijk) niet fataal. Tijdens het starten van de stack verschijnt er uitvoer vergelijkbaar met de onderstaande:

```
# service bluetooth start ubt0
BD_ADDR: 00:02:72:00:d4:1a
Features: 0xff 0xff 0xf 00 00 00 00 00
<3-Slot> <5-Slot> <Encryption> <Slot offset>
<Timing accuracy> <Switch> <Hold mode> <Sniff mode>
<Park mode> <RSSI> <Channel quality> <SCO link>
<HV2 packets> <HV3 packets> <u-law log> <A-law log> <CVSD>
<Paging scheme> <Power control> <Transparent SCO data>
Max. ACL packet size: 192 bytes
Number of ACL packets: 8
Max. SCO packet size: 64 bytes
Number of SCO packets: 8
```

29.4.3. Host Controller Interface (HCI)

Het Host Controller Interface (HCI) biedt een opdrachtinterface naar de controller van de basisband en de verbindingsbeheerder, en toegang tot hardwarestatus en controleregisters. Deze interface biedt een uniforme manier om de mogelijkheden van de basisband van Bluetooth te benaderen. De HCI-laag op de gastheer wisselt gegevens en opdrachten uit met de HCI-firmware in de Bluetooth-hardware. Het stuurprogramma voor de Host Controller Transport Layer (i.e., de fysieke bus) biedt aan beide HCI-lagen de mogelijkheid om informatie met elkaar uit te wisselen.

Voor een enkel Bluetooth-apparaat wordt een enkele Netgraph knoop van het type *hci* aangemaakt. De HCI-knoop is normaliter verbonden met de knoop van het Bluetooth-apparaatstuurprogramma

(naar beneden toe) en de L2CAP-knoop (naar boven toe). Alle HCI-bewerkingen dienen te worden uitgevoerd op de HCI-knoop en niet op de knoop van het apparaatstuurprogramma. De standaardnaam voor de HCI-knoop is "devicehci". Kijk voor meer details in de hulppagina ng_hci(4).

Eén van de meest voorkomende taken is het ontdekken van Bluetooth-apparaten binnen radiobereik. Deze bewerking wordt *ondervragen* genoemd. Ondervragen en andere HCI-gerelateerde bewerkingen worden uitgevoerd met het programma hccontrol(8). Het onderstaande voorbeeld laat zien hoe kan worden uitgezocht welke Bluetooth-apparaten zich binnen het bereik bevinden. De lijst met apparaten zou binnen enkele seconden moeten binnenkomen. Bedenk dat een apparaat op afstand alleen antwoord op de ondervraging zal geven indien het in *ontdekbare* modus staat.

```
% hccontrol -n ubt0hci inquiry
Inquiry result, num_responses=1
Inquiry result #0
       BD_ADDR: 00:80:37:29:19:a4
       Page Scan Rep. Mode: 0x1
       Page Scan Period Mode: 00
       Page Scan Mode: 00
       Class: 52:02:04
       Clock offset: 0x78ef
Inquiry complete. Status: No error [00]
```

BD_ADDR is een uniek adres van een Bluetooth-apparaat, vergelijkbaar met een MAC-adres van een netwerkkaart. Dit adres is nodig voor verdere communicatie met een apparaat. Het is mogelijk om een menselijk leesbare naam aan een BD_ADDR toe te kennen. Het bestand /etc/bluetooth/hosts bevat informatie over de bekende Bluetooth-gastheren. Het volgende voorbeeld laat zien hoe de menselijk leesbare naam dat aan het apparaat op afstand was toegekend te verkrijgen is:

```
% hccontrol -n ubt0hci remote_name_request 00:80:37:29:19:a4
BD_ADDR: 00:80:37:29:19:a4
Name: Pav's T39
```

Tijdens het uitvoeren van een ondervraging op een Bluetooth-apparaat op afstand zal het de computer als "uw.gastheer.naam (ubt0)" vinden. De naam die aan het lokale apparaat is toegekend, kan altijd gewijzigd worden.

Het Bluetooth-systeem biedt een punt-naar-punt-verbinding (slechts twee Bluetooth-eenheden betrokken), of een punt-naar-veelpunt-verbinding. Bij een punt-naar-veelpunt-verbinding wordt de verbinding met meerdere Bluetooth-apparaten gedeeld. Het volgende voorbeeld laat zien hoe de lijst met actieve basisbandverbindingen voor het lokale apparaat te verkrijgen is:

```
% hccontrol -n ubt0hci read_connection_list
Remote BD_ADDR    Handle Type Mode Role Encrypt Pending Queue State
00:80:37:29:19:a4     41  ACL    0 MAST    NONE       0     0 OPEN
```

Een *verbindingshandvat* is nuttig indien het beëindigen van de basisbandverbinding noodzakelijk is. Normaalgesproken is het niet nodig om dit handmatig te doen. De stack zal automatisch niet-actieve basisbandverbindingen beëindigen.

```
# hccontrol -n ubt0hci disconnect 41
Connection handle: 41
Reason: Connection terminated by local host [0x16]
```

Raadpleeg `hccontrol help` voor een volledige lijst van beschikbare HCI-opdrachten. Voor de meeste HCI-opdrachten zijn geen beheerdersrechten nodig.

29.4.4. Logical Link Control and Adaptation Protocol (L2CAP)

Het Logical Link Control and Adaptation Protocol (L2CAP) biedt verbindingsgeoriënteerde en verbindingsloze gegevensdiensten met mogelijkheden om protocollen te multiplexen en mogelijkheden voor segmentatie/herassemblage voor protocollen in hogere lagen. L2CAP staat toe dat protocollen en toepassingen in hogere lagen L2CAP-gegevenspakketten met een maximale lengte van 64 kB te verzenden en ontvangen.

L2CAP is op het concept van *kanalen* gebaseerd. Een kanaal is een logische verbinding bovenop een basisbandverbinding. Elk kanaal is op een veel-op-één manier aan een enkel protocol gebonden. Aan hetzelfde protocol kunnen meerdere kanalen worden gebonden, maar één kanaal kan niet aan meerdere protocollen worden gebonden. Elk L2CAP-pakket dat op een kanaal wordt ontvangen, wordt naar het juiste hogere protocol doorgestuurd. Meerdere kanalen kunnen dezelfde basisbandverbinding delen.

Voor elk Bluetooth-apparaat wordt een enkele Netgraph-knoop van het soort *l2cap* aangemaakt. De L2CAP-knoop is normaalgesproken verbonden met de Bluetooth HCI-knoop (naar beneden toe) en de knopen van de stopcontacten voor Bluetooth (naar boven toe). De standaardnaam voor de L2CAP-knoop is "devicel2cap". Zie voor meer details de hulppagina ng_l2cap(4).

Een nuttig commando is l2ping(8), dat gebruikt kan worden om andere apparaten te pingen. Sommige Bluetooth-implementaties geven niet alle verzonden gegevens terug, dus is `0 bytes` normaal in het volgende voorbeeld.

```
# l2ping -a 00:80:37:29:19:a4
0 bytes from 0:80:37:29:19:a4 seq_no=0 time=48.633 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=1 time=37.551 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=2 time=28.324 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=3 time=46.150 ms result=0
```

Met het programma l2control(8) kunnen verschillende bewerkingen op L2CAP-knopen worden uitgevoerd. Dit voorbeeld laat zien hoe de lijst met logische verbindingen (kanalen) en de lijst met basisbandverbindingen voor het lokale apparaat verkregen kunnen worden:

```
% l2control -a 00:02:72:00:d4:1a read_channel_list
L2CAP channels:
```

```
Remote BD_ADDR     SCID/ DCID   PSM  IMTU/ OMTU State
00:07:e0:00:0b:ca    66/   64     3   132/  672 OPEN
% l2control -a 00:02:72:00:d4:1a read_connection_list
L2CAP connections:
Remote BD_ADDR    Handle Flags Pending State
00:07:e0:00:0b:ca     41 O           0 OPEN
```

Een ander diagnostisch programma is btsockstat(1). Het heeft ongeveer hetzelfde doel als netstat(1), maar dan voor Bluetooth-netwerkgerelateerde gegevensstructuren. Het onderstaande voorbeeld laat dezelfde logische verbinding zien als die van l2control(8) hierboven.

```
% btsockstat
Active L2CAP sockets
PCB      Recv-Q Send-Q Local address/PSM       Foreign address   CID   State
c2afe900      0      0 00:02:72:00:d4:1a/3     00:07:e0:00:0b:ca 66    OPEN
Active RFCOMM sessions
L2PCB    PCB      Flag MTU   Out-Q DLCs State
c2afe900 c2b53380 1    127   0     Yes  OPEN
Active RFCOMM sockets
PCB      Recv-Q Send-Q Local address     Foreign address   Chan DLCI State
c2e8bc80      0    250 00:02:72:00:d4:1a 00:07:e0:00:0b:ca 3    6    OPEN
```

29.4.5. Het RFCOMM-protocol

Het RFCOMM-protocol biedt emulatie van seriële poorten over het L2CAP-protocol. Het protocol is gebaseerd op de ETSI-standaard TS 07.10. RFCOMM is een eenvoudig transportprotocol, met aanvullende voorzieningen om de 9 circuits van RS-232- (EIATIA-232-E-) seriële poorten te emuleren. Het RFCOMM-protocol ondersteunt tot 60 gelijktijdige verbindingen (RFCOMM-kanalen) tussen twee Bluetooth-apparaten.

Het is de bedoeling van RFCOMM dat in een volledig communicatiepad twee toepassingen op verschillende apparaten draaien (de eindpunten van de communicatie) met daartussen een communicatiesegment. RFCOMM is bedoeld om de toepassingen te beheren die gebruik maken van de seriële poorten van de apparaten waarop ze zijn geïnstalleerd. Het communicatiesegment is een directe Bluetooth-verbinding van het ene apparaat naar het andere.

RFCOMM houdt zich alleen bezig met de verbinding tussen twee apparaten bij directe verbindingen, of tussen het apparaat en een modem in het geval van een netwerk. RFCOMM kan andere opstellingen ondersteunen, zoals modules die via draadloze Bluetooth-technologie communiceren aan de ene kant, en een draadinterface aanbieden aan de andere kant.

In FreeBSD is het RFCOMM-protocol in de laag van de Bluetooth-stopcontacten geïmplementeerd.

29.4.6. Het paren van apparaten

Standaard is Bluetooth-communicatie niet geauthenticeerd en kan elk apparaat met elk ander apparaat praten. Een Bluetooth-apparaat (bijvoorbeeld een mobiele telefoon) kan ervoor kiezen dat voor bepaalde diensten authenticatie nodig is (bijvoorbeeld voor de inbeldienst). Bluetooth-

authenticatie geschied normaalgesproken met *PIN-codes*. Een PIN-code is een ACII-reeks van maximaal 16 tekens lang. De gebruiker dient dezelfde PIN-code op beide apparaten in te voeren. Nadat de gebruiker de PIN-code heeft ingevoerd, zullen beide apparaten een *verbindingssleutel* aanmaken. Hierna kan de verbindingssleutel òfwel in de apparaten zelf, òfwel in een permanente opslag worden opgeslagen. De volgende keer zullen beide apparaten de van tevoren aangemaakte verbindingssleutel gebruiken. Bovenstaande procedure wordt *paren* genoemd. Merk op dat indien een apparaat de verbindingssleutel verliest, het paren moet worden herhaald.

De daemon hcsecd(8) is verantwoordelijk voor het behandelen van alle verzoeken voor Bluetooth-authenticatie. Het standaard instellingenbestand is /etc/bluetooth/hcsecd.conf. Een voorbeeldsectie voor een mobiele telefoon waarvan de PIN-code willekeurig op "1234" is hieronder beschreven:

```
device {
    bgaddr  00:80:37:29:19:a4;
    name    "Pav's T39";
    key nokey;
    pin "1234";
      }
```

Er is geen limiet voor PIN-codes (behalve de lengte). Voor sommige apparaten (bijvoorbeeld Bluetooth-headsets) kan de PIN-code vast zijn ingebouwd. De schakelaar `-d` dwingt de daemon hcsecd(8) om op de voorgrond te blijven, zodat het gemakkelijk is om te zien wat er gebeurt. Stel het andere apparaat in om paarverzoeken te ontvangen en initialiseer de Bluetooth-verbinding naar het andere apparaat. Het apparaat moet zeggen dat het paarverzoek geaccepteerd is en om de PIN-code vragen. Geef dezelfde PIN-code op als in hcsecd.conf. Nu zijn de PC en het andere apparaat gepaard. Als alternatief kan paren op het andere apparaat worden geïnitialiseerd.

De volgende regel kan aan het bestand /etc/rc.conf worden toegevoegd om hcsecd automatisch met het systeem op te starten:

```
hcsecd_enable="YES"
```

Het volgende is een voorbeeld van de uitvoer van de daemon hcsecd:

```
hcsecd[16484]: Got Link_Key_Request event from 'ubt0hci', remote bdaddr
0:80:37:29:19:a4
hcsecd[16484]: Found matching entry, remote bdaddr 0:80:37:29:19:a4, name 'Pav's T39',
link key doesn't exist
hcsecd[16484]: Sending Link_Key_Negative_Reply to 'ubt0hci' for remote bdaddr
0:80:37:29:19:a4
hcsecd[16484]: Got PIN_Code_Request event from 'ubt0hci', remote bdaddr
0:80:37:29:19:a4
hcsecd[16484]: Found matching entry, remote bdaddr 0:80:37:29:19:a4, name 'Pav's T39',
PIN code exists
hcsecd[16484]: Sending PIN_Code_Reply to 'ubt0hci' for remote bdaddr 0:80:37:29:19:a4
```

29.4.7. Service Discovery Protocol (SDP)

Het Service Discovery Protocol (SDP) biedt voor cliënttoepassingen de mogelijkheid om diensten te ontdekken die door servertoepassingen worden aangeboden alsook de kenmerken van deze diensten. De kenmerken van een dienst omvatten de soort of klasse van de aangeboden dienst en de informatie over het mechanisme of protocol dat nodig is om de dienst te gebruiken.

SDP omvat communicatie tussen een SDP-server en een SDP-cliënt. De server houdt een lijst van dienstenregistraties bij die de eigenschappen van de diensten beschrijven die met de server geassocieerd zijn. Elke dienstregistratie bevat informatie over een enkele dienst. Een cliënt kan informatie over een dienstregistratie opvragen die door de SDP-server wordt bijgehouden door een SDP-verzoek in te dienen. Indien de cliënt, of een toepassing die met de cliënt geassocieerd is, besluit om de dienst te gebruiken, moet het een aparte verbinding naar de aanbieder van de dienst openen om de dienst te gebruiken. SDP biedt een mechanisme om diensten en hun attributen te ontdekken, maar het biedt geen mechanisme om die diensten te gebruiken.

Normaalgesproken zoekt een SDP-cliënt naar diensten naar aanleiding van enkele gewenste eigenschappen van die diensten. Soms is het echter wenselijk om te ontdekken welke soorten diensten door de dienstregistraties van een SDP-server worden beschreven zonder enige voorkennis van deze diensten. Dit kijken naar alle aangeboden diensten wordt *browsen* genoemd.

De Bluetooth SDP-server sdpd(8) en de opdrachtregelcliënt sdpcontrol(8) zitten in de standaard FreeBSD-installatie. Het volgende voorbeeld laat zien hoe een SDP-browse query uit te voeren.

```
% sdpcontrol -a 00:01:03:fc:6e:ec browse
Record Handle: 00000000
Service Class ID List:
        Service Discovery Server (0x1000)
Protocol Descriptor List:
        L2CAP (0x0100)
                Protocol specific parameter #1: u/int/uuid16 1
                Protocol specific parameter #2: u/int/uuid16 1

Record Handle: 0x00000001
Service Class ID List:
        Browse Group Descriptor (0x1001)

Record Handle: 0x00000002
Service Class ID List:
        LAN Access Using PPP (0x1102)
Protocol Descriptor List:
        L2CAP (0x0100)
        RFCOMM (0x0003)
                Protocol specific parameter #1: u/int8/bool 1
Bluetooth Profile Descriptor List:
        LAN Access Using PPP (0x1102) ver. 1.0
```

i. enzovoorts. Merk op dat elke dienst een lijst met attributen heeft (bijvoorbeeld een RFCOMM-kanaal). Afhankelijk van de dienst kan het nodig zijn om een aantekening van sommige

attributen te maken. Sommige Bluetooth-implementaties ondersteunen dienst-browsen niet en zullen een lege lijst teruggeven. In dit geval is het mogelijk om naar de specifieke dienst te zoeken. Het onderstaande voorbeeld laat zien hoe naar de dienst OBEX Object Push (OPUSH) gezocht kan worden:

```
% sdpcontrol -a 00:01:03:fc:6e:ec search OPUSH
```

Het aanbieden van diensten op FreeBSD aan Bluetooth-cliënten wordt gedaan met de server sdpd(8). De volgende regel kan aan het bestand /etc/rc.conf worden toegevoegd:

```
sdpd_enable="YES"
```

Het daemon sdpd kan worden gestart met:

```
# service sdpd start
```

De plaatselijke servertoepassing die Bluetooth-diensten wil aanbieden aan verre cliënten zal de dienst registreren bij de plaatselijke SDP-daemon. Een voorbeeld van zo'n toepassing is rfcomm_pppd(8). Nadat het gestart is zal het de Bluetooth LAN-dienst bij de plaatselijke SDP-daemon registreren.

De lijst met diensten die bij de plaatselijke SDP-server zijn geregistreerd kan worden opgevraagd door te SDP-browsen via het plaatselijke controlekanaal:

```
# sdpcontrol -l browse
```

29.4.8. Dial-Up Networking (DUN) en netwerktoegang met PPP (LAN) profielen

Het inbelnetwerk (DUN) profiel wordt het meeste gebruikt met modems en mobiele telefoons. De volgende scenario's worden in dit profiel behandeld:

- het gebruik van een mobiele telefoon of modem door een computer als een draadloze modem voor het verbinden met een inbelserver voor Internet-toegang, of voor andere inbeldiensten;
- het gebruik van een mobiele telefoon of modem door een computer om gegevensoproepen te ontvangen.

Het profiel voor netwerktoegang met PPP (LAN) kan in de volgende situaties gebruikt worden:

- LAN-toegang voor een enkel Bluetooth-apparaat;
- LAN-toegang voor meerdere Bluetooth-apparaten;
- PC naar PC (door PPP-netwerken over een seriële kabel te emuleren).

Op FreeBSD zijn beide profielen geïmplementeerd met ppp(8) en rfcomm_pppd(8) - een wrapper

die een RFCOMM Bluetooth-verbinding omzet in iets waar PPP mee overweg kan. Voordat een profiel gebruikt kan worden, dient een nieuw PPP-label in het bestand /etc/ppp/ppp.conf te worden aangemaakt. Raadpleeg de hulppagina rfcomm_pppd(8) voor voorbeelden.

In het volgende voorbeeld zal rfcomm_pppd(8) gebruikt worden om RFCOMM-verbinding met een ver apparaat met BD_ADDR 00:80:37:29:19:a4 op een DUN RFCOMM-kanaal te maken. Het eigenlijke RFCOMM-kanaalnummer wordt via SDP van het verre apparaat verkregen. Het is mogelijk om het RFCOMM-kanaal handmatig op te geven, en in dat geval zal rfcomm_pppd(8) het SDP-verzoek niet uitvoeren. Gebruik sdpcontrol(8) om het RFCOMM-kanaal op het verre apparaat te achterhalen.

```
# rfcomm_pppd -a 00:80:37:29:19:a4 -c -C dun -l rfcomm-dialup
```

Om netwerktoegang met PPP (LAN) aan te bieden moet de server sdpd(8) draaien. Er dient een nieuwe regel voor LAN-cliënten in het bestand /etc/ppp/ppp.conf aangemaakt te worden. Raadpleeg de hulppagina rfcomm_pppd(8) voor voorbeelden. Tenslotte dient de RFCOMM PPP-server op een geldig RFCOMM-kanaal gestart te worden. De RFCOMM PPP-server zal automatisch de Bluetooth LAN-dienst bij de plaatselijke SDP-daemon registreren. Het volgende voorbeeld laat zien hoe een RFCOMM PPP-server te starten:

```
# rfcomm_pppd -s -C 7 -l rfcomm-server
```

29.4.9. Het OBEX Object Push (OPUSH) profiel

OBEX is een veelgebruikt protocol voor eenvoudige bestandsoverdrachten tussen mobiele apparaten. Het primaire gebruik is infraroodcommunicatie, waar het wordt gebruikt voor generieke bestandsoverdrachten tussen notebooks of PDA's, en om visitekaarten en kalenderregels tussen mobiele telefoons en andere apparaten met PIM-toepassingen over te dragen.

De OBEX-server en clieënt zijn geïmplenteerd als een pakket van derde partij, obexapp, dat beschikbaar is als de port comms/obexapp.

De OBEX-cliënt wordt gebruikt om objecten naar en/of van de OBEX-server te duwen/trekken. Een object kan bijvoorbeeld een visitekaart of een afspraak zijn. De OBEX-cliënt kan het RFCOMM-kanaalnummer van het verre apparaat via SDP opvragen. Dit kan gedaan worden door de dienstnaam in plaats van het RFCOMM-kanaalnummer op te geven. De ondersteunde dienstnamen zijn: IrMC, FTRN, en OPUSH. Het is mogelijk om het RFCOMM-kanaal als een nummer op te geven. Het onderstaande is een voorbeeld van een OBEX-sessie, waar een apparaatinformatie-object van de mobiele telefoon wordt getrokken, en een nieuw object (een visitekaart) in de gids van de telefoon wordt geduwd:

```
% obexapp -a 00:80:37:29:19:a4 -C IrMC
obex> get telecom/devinfo.txt devinfo-t39.txt
Success, response: OK, Success (0x20)
obex> put new.vcf
Success, response: OK, Success (0x20)
obex> di
```

```
Success, response: OK, Success (0x20)
```

Om de dienst OBEX Object Push aan te bieden, moet de server sdpd(8) draaien. Er moet een hoofdmap worden aangemaakt waarin alle binnenkomende objecten worden opgeslagen. Het standaardpad naar de hoofdmap is /var/spool/obex. Tenslotte moet de OBEX-server op een geldig RFCOMM-kanaal worden gestart. De OBEX-server zal automatisch de dienst OBEX Object Push bij de plaatselijke SDP-daemon registeren. Het onderstaande voorbeeld laat zien hoe de OBEX-server gestart wordt:

```
# obexapp -s -C 10
```

29.4.10. Serial Port Profile (SPP)

Het Seriële Poort Profiel (SPP) zorgt ervoor dat Bluetooth-apparaten RS232 (of gelijkwaardige) seriële kabels kunnen emuleren. Het scenario dat dit profiel behandelt zorgt ervoor dat oude toepassingen Bluetooth kunnen gebruiken als vervanging van kabels, door gebruik te maken van een virtuele seriële poort.

Het programma rfcomm_sppd(1) implementeert het Seriële Poort profiel. Een pseudo-tty wordt gebruikt als abstractie voor een virtuele seriële poort. Onderstaand voorbeeld laat zien hoe met een Seriële Poortdienst voor verre apparaten te verbinden. Merk op dat het niet nodig is om een RFCOMM-kanaal te kiezen - rfcomm_sppd(1) kan het via SDP van het verre apparaat verkrijgen. Dit kan worden overruled door een RFCOMM-kanaal op de opdrachtregel te specificeren.

```
# rfcomm_sppd -a 00:07:E0:00:0B:CA -t /dev/ttyp6
rfcomm_sppd[94692]: Starting on /dev/ttyp6...
```

Als er een verbinding is, kan de pseudo-tty als seriële poort worden gebruikt:

```
# cu -l ttyp6
```

29.4.11. Problemen oplossen

Een apparaat op afstand kan geen verbinding maken

Sommige oudere Bluetooth-apparaten ondersteunen het wisselen van rol niet. Standaard probeert FreeBSD, wanneer het een nieuwe verbinding accepteert, een rolwisseling uit te voeren en meester te worden. Apparaten die dit niet ondersteunen zullen niet kunnen verbinden. Merk op dat van rol wordt gewisseld wanneer een nieuwe verbinding wordt gemaakt, dus het is niet mogelijk om het verre apparaat te vragen of het rolwisseling ondersteunt. Er is een HCI-optie om rolwisselen aan de plaatselijke kant uit te zetten:

```
# hccontrol -n ubt0hci write_node_role_switch 0
```

Er gaat iets mis, kan ik precies zien wat er gebeurt?

Ja, dit is mogelijk. Gebruik het pakket hcidump, dat beschikbaar is als de port comms/hcidump. Het gereedschap hcidump is vergelijkbaar met tcpdump(1). Het kan gebruikt worden om de inhoud van Bluetooth-pakketten op de terminal te laten zien en om de Bluetooth-pakketten naar een bestand te schrijven.

29.5. Bridging

29.5.1. Introductie

Soms is het handig om één fysiek netwerk (zoals een Ethernet-segment) in twee gescheiden netwerksegmenten te verdelen zonder de noodzaak om een IP-subnet aan te maken en een router te gebruiken om de segmenten met elkaar te verbinden. Een apparaat dat twee netwerken op deze manier met elkaar verbindt wordt een "bridge (brug)" genoemd. Een FreeBSD-systeem met twee netwerkkaarten kan als bridge dienen.

De bridge werkt door de adressen van de MAC-laag (Ethernetadressen) van de apparaten op elke netwerkinterface te leren. Het stuurt alleen verkeer tussen twee netwerken door indien de bron en het doel zich op verschillende netwerken bevinden.

In vele opzichten is een bridge als een Ethernet-switch met erg weinig poorten.

29.5.2. Situaties waarin bridging juist is

Er zijn vandaag de dag veel situaties waarin een bridge gebruikt wordt.

Netwerken verbinden

Het basisgebruik van een bridge is het met elkaar verbinden van twee of meer netwerksegmenten. Er zijn vele redenen om een hostgebaseerde bridge te gebruiken in plaats van simpele netwerkapparaten zoals kabelbeperkingen, firewalling of het verbinden van pseudonetwerken zoals een interface van een virtuële machine. Een bridge kan ook een draadloze interface die in hostap-modus draait met een bedraad netwerk verbinden en als een toegangspunt dienen.

Filtering/Bandbreedtebeheersende firewall

Een gebruikelijke situatie dient zich voor wanneer de functionaliteit van een firewall nodig is zonder routing of network address translation (NAT).

Een voorbeeld is een klein bedrijf dat via DSL of ISDN met hun internetprovider verbonden is. Dit bedrijf heeft 13 wereldwijd bereikbare IP-adressen van de internetprovider en 10 PC’s op hun netwerk. In deze situatie is een firewall die op een router gebaseerd is lastig wegens subnet-problemen.

Een firewall die op een bridge gebaseerd is kan ingesteld en net na de DSL- of ISDN-router geplaatst worden zonder dat er problemen met IP-nummers optreden.

Netwerktap

Een bridge kan twee netwerksegmenten verbinden en kan gebruikt worden om alle Ethernetframes die tussen dezen voorbijkomen te inspecteren. Dit kan òfwel vanuit het gebruik van bpf(4)/tcpdump(1) op de bridge-interface òfwel door een kopie van alle frames naar een extra interface (overspanpoort) te versturen.

Laag 2 VPN

Twee Ethernetnetwerken kunnen over een IP-verbinding verbonden worden door de netwerken naar een EtherIP-tunnel te bridgen of met een oplossing gebaseerd po tap(4) zoals OpenVPN.

Laag 2 Redundancy

Een netwerk kan met meerdere verbindingen verbonden worden en het Spanning Tree Protocol gebruiken om overbodige paden te blokkeren. Een Ethernetnetwerk kan alleen juist functioneren indien er slechts één actief pad bestaat tussen twee apparaten, Spanning Tree zal lussen detecteren en de overbodige verbindingen in een geblokkeerde toestand zetten. Indien een van de actieve verbindingen faalt zal het protocol een andere boom berekenen en een van de geblokkeerde paden weer activeren om de verbindingen naar alle punten in het netwerk te herstellen.

29.5.3. De kernel instellen

Deze sectie behandelt de bridges geïmplementeerd met if_bridge(4), een stuurprogramma dat bridges met netgraph implementeert is ook beschikbaar, zie voor meer informatie de hulppagina ng_bridge(4).

Het bridge-stuurprogramma is een kernelmodule en zal automatisch door ifconfig(8) worden geladen wanneer er een bridge-interface wordt aangemaakt. Het is mogelijk om de bridge in de kernel te compileren door `device if_bridge` aan het kernelinstellingenbestand toe te voegen.

Pakketfiltering kan met elk firewall-pakket worden gebruikt dat via het raamwerk pfil(9) aankoppelt. De firewall kan als een module worden geladen of in de kernel worden gecompileerd.

De bridge kan als met altq(4) of dummynet(4) als een verkeersregelaar worden gebruikt.

29.5.4. De bridge inschakelen

De bridge wordt aangemaakt door interfaces te klonen. Om een bridge aan te maken wordt ifconfig(8) gebruikt, indien het bridge-stuurprogramma niet in de kernel aanwezig is zal het automatisch worden geladen.

```
# ifconfig bridge create
# ifconfig bridge0
bridge0: flags=8802<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
        ether 96:3d:4b:f1:79:7a
        id 00:00:00:00:00:00 priority 32768 hellotime 2 fwddelay 15
        maxage 20 holdcnt 6 proto rstp maxaddr 100 timeout 1200
        root id 00:00:00:00:00:00 priority 0 ifcost 0 port 0
```

Een bridge-interface is aangemaakt en er is automatisch een random gegenereerd Ethernetadres aan toegekend. De parameters `maxaddr` en `timeout` bepalen hoeveel MAC-adressen de bridge in de doorstuurtabel houdt en hoeveel seconden voordat elke regel wordt verwijderd nadat het voor het laatst gezien is. De andere parameters bepalen hoe Spanning Tree werkt.

Voeg de netwerkinterfaces die lid zijn aan de bridge toe. Om de bridge pakketten te laten doorsturen dienen alle lidinterfaces en de bridge actief te zijn:

```
# ifconfig bridge0 addm fxp0 addm fxp1 up
# ifconfig fxp0 up
# ifconfig fxp1 up
```

De bridge stuurt nu Ethernet-frames door tussen fxp0 en fxp1. De overeenkomstige configuratie in /etc/rc.conf zodat de bridge tijdens het opstarten wordt aangemaakt is:

```
cloned_interfaces="bridge0"
ifconfig_bridge0="addm fxp0 addm fxp1 up"
ifconfig_fxp0="up"
ifconfig_fxp1="up"
```

Indien de bridge-gastheer een IP-adres nodig heeft dan is de juiste plaats om dit in te stellen op de bridge-interface zelf in plaats van op een van de lidinterfaces. Dit kan statisch of via DHCP worden ingesteld:

```
# ifconfig bridge0 inet 192.168.0.1/24
```

Het is ook mogelijk om een IPv6-adres aan een bridge-interface toe te kennen.

29.5.5. Firewalls gebruiken

Wanneer pakketten worden gefilterd, zullen gebridgede pakketten het filter inbound op de vertrekkende interface passeren, op de bridge-interface en outbound op de bestemde interface. Elke stap kan uitgezet worden. Wanneer de richting van het pakketverkeer belangrijk is, kan de firewall het beste op de lidinterfaces draaien en niet op de bridge zelf.

De bridge heeft verschillende aanpasbare instellingen voor het doorlaten van non-IP- en ARP-pakketten, en een laag 2 firewall met IPFW. Zie if_bridge(4) voor meer informatie.

29.5.6. Opspannende boom

Het bridge-stuurprogramma implementeert het Rapid Spanning Tree Protocol (RSTP of 802.1w) met terugwaartse compatibiliteit met het verouderde Spanning Tree Protocol (STP). Spanning Tree wordt gebruikt om lussen in een netwerktopologie te detecteren en verwijderen. RSTP biedt snellere convergentie naar een opspannende boom dan het verouderde STP, het protocol wisselt informatie met naburige switches uit om snel naar forwarding over te gaan zonder lussen te creëren. FreeBSD ondersteunt RSTP en STP als opties, waarbij RSTP de standaard is.

Spanning Tree kan op lidinterfaces worden geactiveerd met het commando `stp`. Voor een bridge met fxp0 en fxp1 alle huidige interfaces, wordt STP met het volgende geactiveerd:

```
# ifconfig bridge0 stp fxp0 stp fxp1
bridge0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        ether d6:cf:d5:a0:94:6d
        id 00:01:02:4b:d4:50 priority 32768 hellotime 2 fwddelay 15
        maxage 20 holdcnt 6 proto rstp maxaddr 100 timeout 1200
        root id 00:01:02:4b:d4:50 priority 32768 ifcost 0 port 0
        member: fxp0 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
                port 3 priority 128 path cost 200000 proto rstp
                role designated state forwarding
        member: fxp1 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
                port 4 priority 128 path cost 200000 proto rstp
                role designated state forwarding
```

De bridge heeft spanning tree ID `00:01:02:4b:d4:50` en prioriteit `32768`. Aangezien het `root id` hetzelfde is geeft dit aan dat dit de hoofdbridge voor de boom is.

Een andere bridge in het netwerk heeft spanning tree ook geactiveerd:

```
bridge0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        ether 96:3d:4b:f1:79:7a
        id 00:13:d4:9a:06:7a priority 32768 hellotime 2 fwddelay 15
        maxage 20 holdcnt 6 proto rstp maxaddr 100 timeout 1200
        root id 00:01:02:4b:d4:50 priority 32768 ifcost 400000 port 4
        member: fxp0 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
                port 4 priority 128 path cost 200000 proto rstp
                role root state forwarding
        member: fxp1 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
                port 5 priority 128 path cost 200000 proto rstp
                role designated state forwarding
```

De reegl `root id 00:01:02:4b:d4:50 priority 32768 ifcost 400000 port 4` geeft aan dat de hoofdbridge `00:01:02:4b:d4:50` is zoals boven en dat de padkosten `400000` zijn vanaf deze bridge, het pad naar de hoofdbridge gaat via `port 4` welke fxp0 is.

29.5.7. Geavanceerd bridgen

Verkeersstromen reconstrueren

De bridge ondersteunt monitormodus, waarin de pakketten worden verwijderd nadat ze door bpf(4) zijn verwerkt, en ze niet verder verwerkt of doorgestuurd worden. Dit kan worden gebruikt om de invoer van twee of meer interfaces naar een enkele bpf(4)-stroom te multiplexen. Dit is nuttig voor het reconstrueren van het verkeer voor netwerktaps welke de RX/TX-signalen over twee verschillende interfaces uitzenden.

Om de invoer van vier netwerkinterfaces als één stroom te lezen:

```
# ifconfig bridge0 addm fxp0 addmfxp1 addm fxp2 addm fxp3 monitor up
# tcpdump -i bridge0
```

SPAN poorten

Van elk Ethernet-frame dat door de bridge wordt ontvangen wordt er een kopie naar de aangewezen SPAN-poort verstuurd. Het aantal geconfigureerde SPAN-poorten op een bridge is onbeperkt, indien een interface aangewezen is als SPAN-poort kan het niet ook als gewone bridgepoort gebruikt worden. Dit is het nuttigste voor het passief afluisteren van een gebridged netwerk op een andere host die met een van de SPAN-poorten van de bridge verbonden is.

Om een kopie van alle frames naar de interface fxp4 te versturen:

```
# ifconfig bridge0 span fxp4
```

Privé-interfaces

Een privé-interface stuurt geen verkeer door naar poorten die niet ook een privé-interface zijn. Het verkeer wordt onvoorwaardelijk geblokkeerd, dus worden er geen Ethernetframes doorgestuurd, inclusief ARP. Indien verkeer selectief dient te worden geblokkeerd dient er in plaats hiervan een firewall gebruikt te worden.

Klevende interfaces

Indien een lidinterface van een bridge als klevend is gemarkeerd worden dynamisch geleerde adresregels als statisch behandelt wanneer ze in de doorstuurcache komen. Klevende interfaces vallen nooit uit de cache en worden nooit vervangen, zelfs niet als het adres op een andere interface wordt gezien. Dit biedt het voordeel van statische adresregels zonder dat de doorstuurtabel van te voren gevuld hoeft te worden, cliënten die geleerd zijn op een bepaald segment van de bridge kunnen niet roamen naar een ander segment.

Een ander voorbeeld voor het gebruik van klevende adressen zou het combineren van de bridge met VLANs zijn om een router te creëren waar klantnetwerken geïsoleerd zijn zonder dat IP-adresruimte verspild wordt. Neem aan dat `KlantA` op `vlan100` zit en `KlantB` op `vlan101`. De bridge heeft het adres `192.168.0.1` en is tevens een internet-router.

```
# ifconfig bridge0 addm vlan100 sticky vlan100 addm vlan101 sticky vlan101
# ifconfig bridge0 inet 192.168.0.1/24
```

Beide cliënten zien `192.168.0.1` als hun standaard gateway en aangezien de bridge-cache kleverig is kunnen ze niet het MAC-adres van de andere klant spoofen om hun verkeer op te vangen.

Alle communicatie tussen de VLANs kan geblokkeerd worden door het gebruik van privé-interfaces (of een firewall):

```
# ifconfig bridge0 private vlan100 private vlan101
```

De klanten zijn compleet geïsoleerd van elkaar, het volledige `/24` adresruimte kan zonder subnetten toegewezen worden.

Adresbeperkingen

Het aantal unieke bron-MAC-adressen achter een interface kan beperkt zijn. Wanneer de limiet bereikt is worden pakketten met een onbekend bronadres gedropt totdat een bestaande ingang in de host-cache vervalt of wordt verwijderd.

Het volgende voorbeeld stelt het maximum aantal Ethernetapparaten voor `KlantA` op `vlan100` in op 10.

```
# ifconfig bridge0 ifmaxaddr vlan100 10
```

SNMP-monitoring

De bridge-interface en STP-parameters kunnen gemonitord worden via het SNMP-daemon dat met het basis FreeBSD-systeem wordt meegeleverd. De geëxporteerde bridge-MIBs houden zich aan de standaarden van de IETF zodat elke SNMP-cliënt of monitorpakket kan worden gebruikt om de gegevens te verzamelen.

Op de bridge-machine dient de regel `begemotSnmpdModulePath."bridge" = "/usr/lib/snmp_bridge.so"` van /etc/snmp.config geactiveerd te worden en het daemon bsnmpd gestart te worden. Andere instellingen zoals gemeenschapsnamen en toegangslijsten dienen eventueel aangepast te worden. Zie bsnmpd(1) en snmp_bridge(3) voor meer informatie.

Het volgende voorbeeld gebruikt de software Net-SNMP (net-mgmt/net-snmp om een bridge te ondervragen, de port net-mgmt/bsnmptools kan ook worden gebruikt. Voeg de volgende regels toe aan $HOME/.snmp/snmp.conf op de SNMP-cliënt-host om de MIB-definities van de bridge in Net-SNMP te importeren:

```
mibdirs +/usr/shared/snmp/mibs
mibs +BRIDGE-MIB:RSTP-MIB:BEGEMOT-MIB:BEGEMOT-BRIDGE-MIB
```

Om een enkele bridge via de IETF BRIDGE-MIB (RFC4188) te monitoren:

```
% snmpwalk -v 2c -c public bridge1.example.com mib-2.dot1dBridge
BRIDGE-MIB::dot1dBaseBridgeAddress.0 = STRING: 66:fb:9b:6e:5c:44
BRIDGE-MIB::dot1dBaseNumPorts.0 = INTEGER: 1 ports
BRIDGE-MIB::dot1dStpTimeSinceTopologyChange.0 = Timeticks: (189959) 0:31:39.59 centi-
seconds
BRIDGE-MIB::dot1dStpTopChanges.0 = Counter32: 2
BRIDGE-MIB::dot1dStpDesignatedRoot.0 = Hex-STRING: 80 00 00 01 02 4B D4 50
...
```

```
BRIDGE-MIB::dot1dStpPortState.3 = INTEGER: forwarding(5)
BRIDGE-MIB::dot1dStpPortEnable.3 = INTEGER: enabled(1)
BRIDGE-MIB::dot1dStpPortPathCost.3 = INTEGER: 200000
BRIDGE-MIB::dot1dStpPortDesignatedRoot.3 = Hex-STRING: 80 00 00 01 02 4B D4 50
BRIDGE-MIB::dot1dStpPortDesignatedCost.3 = INTEGER: 0
BRIDGE-MIB::dot1dStpPortDesignatedBridge.3 = Hex-STRING: 80 00 00 01 02 4B D4 50
BRIDGE-MIB::dot1dStpPortDesignatedPort.3 = Hex-STRING: 03 80
BRIDGE-MIB::dot1dStpPortForwardTransitions.3 = Counter32: 1
RSTP-MIB::dot1dStpVersion.0 = INTEGER: rstp(2)
```

De waarde `dot1dStpTopChanges.0` is twee wat betekent dat de topologie van de STP-bridge twee maal veranderd is, een topologieverandering houdt in dat één of meerdere links in het netwerk zijn veranderd of hebben gefaald en dat er een nieuwe boom is berekend. De waarde `dot1dStpTimeSinceTopologyChange.0` laat zien wanneer dit gebeurde.

Om meerdere bridge-interfaces te monitoren kan men het privé BEGEMOT-BRIDGE-MIB gebruiken:

```
% snmpwalk -v 2c -c public bridge1.example.com
enterprises.fokus.begemot.begemotBridge
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseName."bridge0" = STRING: bridge0
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseName."bridge2" = STRING: bridge2
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseAddress."bridge0" = STRING: e:ce:3b:5a:9e:13
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseAddress."bridge2" = STRING: 12:5e:4d:74:d:fc
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseNumPorts."bridge0" = INTEGER: 1
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseNumPorts."bridge2" = INTEGER: 1
...
BEGEMOT-BRIDGE-MIB::begemotBridgeStpTimeSinceTopologyChange."bridge0" = Timeticks:
(116927) 0:19:29.27 centi-seconds
BEGEMOT-BRIDGE-MIB::begemotBridgeStpTimeSinceTopologyChange."bridge2" = Timeticks:
(82773) 0:13:47.73 centi-seconds
BEGEMOT-BRIDGE-MIB::begemotBridgeStpTopChanges."bridge0" = Counter32: 1
BEGEMOT-BRIDGE-MIB::begemotBridgeStpTopChanges."bridge2" = Counter32: 1
BEGEMOT-BRIDGE-MIB::begemotBridgeStpDesignatedRoot."bridge0" = Hex-STRING: 80 00 00 40
95 30 5E 31
BEGEMOT-BRIDGE-MIB::begemotBridgeStpDesignatedRoot."bridge2" = Hex-STRING: 80 00 00 50
8B B8 C6 A9
```

Om de bridge-interface die via de subboom `mib-2.dot1dBridge` wordt gemonitord te veranderen:

```
% snmpset -v 2c -c private bridge1.example.com
BEGEMOT-BRIDGE-MIB::begemotBridgeDefaultBridgeIf.0 s bridge2
```

29.6. Verbindingsaggregatie en failover

29.6.1. Introductie

De interface lagg(4) maakt het mogelijk om meerdere netwerkinterfaces te aggregeren in één

virtueel interface voor het bieden van fout-tolerante en zeer snelle verbindingen.

29.6.2. Werkmodi

Failover

Zendt en ontvangt verkeer alleen door de meesterpoort. Wanneer de meesterpoort niet beschikbaar is, wordt de volgende actieve poort gebruikt. De eerste toegevoegde interface is de meesterpoort; alle interfaces die hierna zijn toegevoegd worden gebruikt als failover-apparaten. Als failover naar een niet-meesterpoort plaatsvindt, dan wordt de originele poort de meester wanneer deze weer beschikbaar wordt.

Cisco® Fast EtherChannel®

Cisco® Fast EtherChannel® (FEC), is een statische installatie en onderhandelt niet over aggregatie met de peer noch wisselt het frames uit om de verbinding te monitoren. Indien de switch LACP ondersteunt dient dat gebruikt te worden.

FEC balanceert uitgaand verkeer over de actieve poorten gebaseerd op gehashde informatie over protocolheaders en accepteert inkomend verkeer van elke actieve poort. De hash bevat het Ethernet bron- en doeladres, en indien beschikbaar, de VLAN-tag, en de IPv4/IPv6 bron- en doeladressen.

LACP

Het IEEE® 802.3ad Link Aggregation Control Protocol (LACP) en het Marker Protocol. LACP onderhandelt met de peer over een verzameling aggregeerbare verbindingen in één of meerdere Link Aggregated Groups (LAG). Elke LAG is opgebouwd uit poorten die dezelfde snelheid hebben, ingesteld op full-duplex werking. Het verkeer zal over de poorten in de LAG gebalanceerd worden met de hoogste totaalsnelheid, in de meeste gevallen zal er slechts één LAG zijn die alle poorten bevat. Wanneer er fysieke verbindingen veranderen, zal Link Aggregation snel naar een nieuwe opstelling convergeren.

LACP balanceert uitgaand verkeer over de actieve poorten gebaseerd op gehashde informatie over protocolheaders en accepteert inkomend verkeer van elke actieve poort. De hash bevat het Ethernet bron- en doeladres, en indien beschikbaar, de VLAN-tag, en de IPv4/IPv6 bron- en doeladressen.

Loadbalance

Dit is een alias van de *FEC* modus.

Round-Robin

Distribueert uitgaand verkeer door middel van een round-robin scheduler over alle actieve poorten en accepteert inkomend verkeer van elke actieve poort. Deze modus schendt Ethernet frame-ordering en dient met zorg gebruikt te worden.

29.6.3. Voorbeelden

Voorbeeld 25. LACP-aggregatie met een Cisco® switch

Dit voorbeeld verbindt twee interfaces op een FreeBSD-machine met de switch als een enkele

loadgebalanceerde en fout-tolerante verbinding. Er kunnen meer interfaces worden toegevoegd om de doorvoer en fouttolerantie te verhogen. Aangezien frame-ordering verplicht is op Ethernetverbindingen stroomt al het verkeer tussen twee stations altijd over dezelfde fysieke verbinding zodat de maximum snelheid beperkt wordt tot die van één interface. Het verzendalgoritme probeert zoveel mogelijk informatie te gebruiken voor het onderscheiden van verschillende verkeersstromen en deze over de beschikbare interfaces te balanceren.

Voeg op de Cisco® switch de interfaces *FastEthernet0/1* en *FastEthernet0/2* aan de kanaalgroep *1* toe:

```
interface FastEthernet0/1
 channel-group 1 mode active
 channel-protocol lacp
!
interface FastEthernet0/2
 channel-group 1 mode active
 channel-protocol lacp
```

Maak de lagg(4)-interface aan met *fxp0* en *fxp1* en activeer de interface met IP-adres *10.0.0.3/24*:

```
# ifconfig fxp0 up
# ifconfig fxp1 up
# ifconfig lagg0 create
# ifconfig lagg0 up laggproto lacp laggport fxp0 laggport fxp1 10.0.0.3/24
```

Bekijk de interfacestatus van ifconfig:

```
# ifconfig lagg0
```

Poorten die als *ACTIVE* zijn gemarkeerd zijn lid van de actieve aggregatiegroep waarover onderhandeld is met de verre switch en waarover verkeer zal worden verzonden en ontvangen. Gebruik de uitgebreide uitvoer van ifconfig(8) om de LAG-identifiers te bekijken.

```
lagg0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=8<VLAN_MTU>
        ether 00:05:5d:71:8d:b8
        media: Ethernet autoselect
        status: active
        laggproto lacp
        laggport: fxp1 flags=1c<ACTIVE,COLLECTING,DISTRIBUTING>
        laggport: fxp0 flags=1c<ACTIVE,COLLECTING,DISTRIBUTING>
```

Gebruik, om de toestand van de poorten op de switch te bekijken, `show lacp neighbor`.

```
switch# show lacp neighbor
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode

Channel group 1 neighbors

Partner's information:

                LACP port                       Oper    Port     Port
Port      Flags   Priority  Dev ID          Age     Key     Number   State
Fa0/1     SA      32768     0005.5d71.8db8  29s     0x146   0x3      0x3D
Fa0/2     SA      32768     0005.5d71.8db8  29s     0x146   0x4      0x3D
```

Gebruik voor meer detail het commando `show lacp neighbor detail`.

Voeg de volgende regels aan /etc/rc.conf toe om deze informatie na het opnieuw starten te behouden:

```
ifconfig_fxp0="up"
ifconfig_fxp1="up"
cloned_interfaces="lagg0"
ifconfig_lagg0="laggproto lacp laggport fxp0 laggport fxp1 10.0.0.3/24"
```

Voorbeeld 26. Failover-modus

Failover-modus kan worden gebruikt om op een secondaire interface over te schakelen wanneer de verbinding op de meesterinterface verloren is. Activeer de onderliggende fysieke interface. Creëer de interface `lagg0`, met *fxp0* als de meesterinterface en *fxp1* als de secondaire interface en ken er IP-adres *10.0.0.15/24* aan toe:

```
# ifconfig fxp0 up
# ifconfig fxp1 up
# ifconfig lagg0 create
# ifconfig lagg0 up laggproto failover laggport fxp0 laggport fxp1 10.0.0.15/24
```

De interface zal er ongeveer als volgt uitzien, de grote verschillen zullen het MAC-adres en de apparaatnamen zijn:

```
# ifconfig lagg0
lagg0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=8<VLAN_MTU>
        ether 00:05:5d:71:8d:b8
        inet 10.0.0.15 netmask 0xffffff00 broadcast 10.0.0.255
        media: Ethernet autoselect
        status: active
```

```
        laggproto failover
        laggport: fxp1 flags=0<>
        laggport: fxp0 flags=5<MASTER,ACTIVE>
```

Het verkeer zal worden verzonden en ontvangen op *fxp0*. Indien de verbinding op *fxp0* verloren is, zal *fxp1* de actieve verbinding worden. Indien de verbinding op de meesterinterface hersteld is, zal het weer de actieve verbinding worden.

Voeg de volgende regels aan /etc/rc.conf toe om deze informatie na het opnieuw starten te behouden:

```
ifconfig_fxp0="up"
ifconfig_fxp1="up"
cloned_interfaces="lagg0"
ifconfig_lagg0="laggproto lacp laggport fxp0 laggport fxp1 10.0.0.15/24"
```

Voorbeeld 27. Failover-modus tussen bekabelde en draadloze interfaces

Voor laptop-gebruikers is het normaliter wenselijk om het draadloze interface als secundair interface te gebruiken indien het bekabelde interface niet beschikbaar is. Met lagg(4) is het mogelijk om één IP-adres te gebruiken en het bekabelde interface voor zowel prestatie als veiligheid te prefereren terwijl de mogelijkheid behouden blijft om de draadloze verbinding te gebruiken.

In deze opstelling dient het MAC-adres van het onderliggende draadloze interface overschreven te worden om met dat van lagg(4) overeen te komen, welke afkomstig is van het primaire interface dat wordt gebruikt, het bekabelde interface.

In deze opstelling wordt het bekabelde interface, *bge0* als meester gebruikt, en het draadloze interface, *wlan0*, als het failover-interface. *wlan0* was aangemaakt vanuit *iwn0* voor welke het MAC-adres van de bekabelde verbinding zal worden gebruikt. De eerste stap is om het MAC-adres van het bekabelde interface te verkrijgen:

```
#  ifconfig bge0
bge0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
      options=19b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM,TSO4>
      ether 00:21:70:da:ae:37
      inet6 fe80::221:70ff:feda:ae37%bge0 prefixlen 64 scopeid 0x2
      nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
      media: Ethernet autoselect (1000baseT <full-duplex>)
      status: active
```

bge0 kan vervangen worden door het eigenlijke interface, er zal een andere regel met `ether` verschijnen, dit is het MAC-adres van het bekabelde interface. Om het onderliggende draadloze interface, *iwn0* te wijzigen:

```
# ifconfig iwn0 ether 00:21:70:da:ae:37
```

Activeer het draadloze interface maar geef er nog geen IP-adres aan:

```
# ifconfig wlan0 create wlandev iwn0 ssid mijn_router up
```

Activeer de interface *bge0*. Maak het lagg(4)-interface aan met *bge0* als meester, en met failover naar *wlan0* indien nodig:

```
# ifconfig bge0 up
# ifconfig lagg0 create
# ifconfig lagg0 up laggproto failover laggport bge0 laggport wlan0
```

Het interface zal er ongeveer als volgt uitzien, de grootste verschillen zullen het MAC-adres en de apparaatnamen zijn:

```
# ifconfig lagg0
lagg0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
      options=8<VLAN_MTU>
      ether 00:21:70:da:ae:37
      media: Ethernet autoselect
      status: active
      laggproto failover
      laggport: wlan0 flags=0<>
      laggport: bge0 flags=5<MASTER,ACTIVE>
```

Start vervolgens de DHCP-cliënt om een IP-adres te verkrijgen:

```
# dhclient lagg0
```

Om deze configuratie bij het opstarten te behouden, kan het volgende aan /etc/rc.conf worden toegevoegd:

```
ifconfig_bge0="up"
ifconfig_iwn0="ether 00:21:70:da:ae:37"
wlans_iwn0="wlan0"
ifconfig_wlan0="WPA"
cloned_interfaces="lagg0"
ifconfig_lagg0="laggproto failover laggport bge0 laggport wlan0 DHCP"
```

29.7. Schijfloos werken

Een FreeBSD-machine kan over het netwerk opstarten en zonder een plaatselijke schijf werken, door gebruik te maken van bestandssystemen die van een NFS-server aangekoppeld worden. Er zijn geen systeemwijzigingen nodig anders dan de standaard instellingenbestanden. Dit soort systemen is relatief eenvoudig op te zetten omdat alle noodzakelijke elementen al aanwezig zijn:

- Er zijn minstens twee manieren om de kernel over het netwerk te laden:
  - PXE: De Intel® Preboot eXecution Environment is een vorm een smart boot ROM dat in sommige netwerkkaarten en moederborden is ingebouwd. Bekijk de hulppagina pxeboot(8) voor meer informatie.
  - De poort Etherboot (net/etherboot) maakt code aan dat naar een ROM geschreven kan worden en dat kernels over het netwerk opstart. De code kan òfwel naar een opstart-PROM op een netwerkkaart geflashed worden, òfwel van een floppy (of harde) schijf geladen worden, òfwel van een draaiend MS-DOS® systeem geladen worden. Vele netwerkkaarten worden ondersteund.
- Een voorbeeldscript (/usr/shared/examples/diskless/clone_root) vergemakkelijkt het aanmaken en beheren van het root bestandssysteem van het werkstation op de server. Het kan nodig zijn dat het script wat aangepast moet worden, maar het zorgt voor een snelle start.
- Er bestaan standaardbestanden voor het opstarten van het systeem in /etc om een systeemstart zonder schijf te detecteren en te ondersteunen.
- Het gebruik van een wisselbestand, indien nodig, kan worden gedaan naar òfwel een NFS bestand òfwel naar een plaatselijke schijf.

Er zijn vele manieren om een schijfloos werkstation op te zetten. Hierbij zijn veel elementen betrokken, en vele kunnen aan de eigen smaak worden aangepast. Het volgende beschrijft variaties met betrekking tot het installeren van een compleet systeem, waarbij de nadruk ligt op de eenvoud en de compatibiliteit met de standaard opstartscripts van FreeBSD. Het beschreven systeem heeft de volgende eigenschappen:

- De schijfloze werkstations gebruiken een gedeeld bestandssysteem voor /, dat alleen gelezen kan worden, en een gedeeld bestandssysteem voor /usr, dat eveneens alleen gelezen kan worden.

  Het root-bestandssysteem is een kopie van een standaard root-bestandssysteem voor FreeBSD (typisch van een server), waarbij enkele instellingenbestanden zijn overschreven door versies die specifiek zijn voor een schijfloos systeem of, mogelijk, door het werkstation horen waar ze bij horen.

  De delen van het root-bestandssysteem die beschrijfbaar moeten zijn, zijn overdekt met md(4) bestandssystemen. Alle veranderingen gaan verloren indien het systeem opnieuw wordt opgestart.

- De kernel is overgedragen en òfwel met Etherboot òfwel met PXE geladen, aangezien sommige situaties het gebruik van één van de methodes kan eisen.



Het systeem zoals hierboven beschreven is onveilig. Het dient in een beschermd

> gebied van een netwerk te functioneren, en niet vertrouwd te worden door andere hosts.

Alle informatie in deze sectie is getest met FreeBSD 5.2.1-RELEASE.

29.7.1. Achtergrondinformatie

Het installeren van schijfloze werkstations is zowel vrij rechttoe-rechtaan als foutgevoelig. Deze fouten zijn soms moeilijk vast te stellen wegens een aantal redenen. Bijvoorbeeld:

- Opties die tijdens het compileren zijn opgegeven kunnen verschillend gedrag tonen tijdens het draaien.
- Foutmeldingen zijn vaak cryptisch of geheel afwezig.

Op dit gebied is het bezit van wat achtergrondkennis over de gebruikte mechanismen zeer nuttig om mogelijke problemen op te lossen.

Voor een succesvol opstarten dienen verschillende handelingen uitgevoerd te worden:

- De machine moet een aantal initiële parameters zoals het IP-adres, de bestandsnaam van de executable, de naam van de server, en het root-pad verkrijgen. Dit wordt gedaan door gebruik te maken van de DHCP of BOOTP protocollen. DHCP is een compatible uitbreiding van BOOTP, het gebruikt dezelfde poorten en het pakketformaat heeft dezelfde basis.

  Het is mogelijk om een systeem in te stellen zodat het alleen BOOTP gebruikt. Het serverprogramma bootpd(8) wordt met het basissysteem van FreeBSD meegeleverd.

  DHCP biedt echter een aantal voordelen boven BOOTP (fijnere instellingenbestanden, mogelijkheid om PXE te gebruiken, en vele anderen die niet direct verband houden met schijfloos werken), er zal hoofdzakelijk een opstelling met DHCP worden beschreven, met analoge voorbeelden voor bootpd(8) indien mogelijk. De voorbeeldopstelling zal het softwarepakket van ISC DHCP gebruiken (versie 3.0.1.r12 was geïnstalleerd op de testserver).

- De machine moet één of meerdere programma's naar het plaatselijke geheugen versturen. Eén van TFTP of NFS wordt gebruikt. De keuze tussen TFTP en NFS is op verschillende plaatsen een optie tijdens het compileren. Een veelgemaakte fout is het opgeven van bestandsnamen voor het verkeerde protocol: TFTP verstuurd typisch alle bestanden vanuit één map op de server, en verwacht dat alle bestandsnamen relatief aan deze map zijn; NFS verwacht absolute bestandspaden.
- De mogelijke tussentijdse opstartprogramma's en de kernel dienen geïnitialiseerd en uitgevoerd te worden. Er zijn enkele belangrijke variaties op dit gebied:
  - PXE zal pxeboot(8) laden, wat een aangepaste versie is van de lader voor stage drie van FreeBSD. loader(8) zal de meeste parameters verkrijgen die noodzakelijk zijn om het systeem op te starten, en zal ze in de kernelomgeving laten staan voordat het de controle overdraagt. Het is in dit geval mogelijk om een GENERIC kernel te gebruiken.
  - Etherboot zal met minder voorbereiding direct de kernel laden. Hiervoor is het noodzakelijk om een kernel met specifieke opties te bouwen.

PXE en Etherboot werken beide even goed; echter, omdat kernels normaalgesproken meer werk overlaten aan loader(8), is PXE de te verkiezen methode.

Indien het BIOS en de netwerkkaarten PXE ondersteunen, dient dat waarschijnlijk gebruikt te worden.

- Tenslotte: de machine heeft toegang tot de bestandssystemen nodig. NFS wordt in alle gevallen gebruikt.

Zie ook de hulppagina diskless(8).

29.7.2. Installatie-instructies

Instellen met behulp van ISC DHCP

De ISC DHCP server kan zowel verzoeken voor BOOTP als DHCP beantwoorden.

ISC DHCP 4.2 maakt geen deel uit van het basissysteem. Eerst dient de poort net/isc-dhcp42-server of het corresponderende pakket geïnstalleerd te worden.

Wanneer ISC DHCP is geïnstalleerd, heeft het een instellingenbestand nodig om te draaien (normaliter /usr/local/etc/dhcpd.conf genoemd). Hieronder volgt een voorbeeld met commentaar, waarbij host `margaux` gebruik maakt van Etherboot en `corbieres` gebruik maakt van PXE:

```
default-lease-time 600;
max-lease-time 7200;
authoritative;

option domain-name "example.com";
option domain-name-servers 192.168.4.1;
option routers 192.168.4.1;

subnet 192.168.4.0 netmask 255.255.255.0 {
  use-host-decl-names on; ①
  option subnet-mask 255.255.255.0;
  option broadcast-address 192.168.4.255;

  host margaux {
    hardware ethernet 01:23:45:67:89:ab;
    fixed-address margaux.example.com;
    next-server 192.168.4.4; ②
    filename "/data/misc/kernel.diskless"; ③
    option root-path "192.168.4.4:/data/misc/diskless"; ④
  }
  host corbieres {
    hardware ethernet 00:02:b3:27:62:df;
    fixed-address corbieres.example.com;
    next-server 192.168.4.4;
    filename "pxeboot";
    option root-path "192.168.4.4:/data/misc/diskless";
```

```
  }
}
```

① Deze optie vertelt dhcpd om de waarde die in de verklaringen voor `host` staan te versturen als de hostnaam voor de schijfloze host. Een andere mogelijkheid is om `option host-name margaux` binnen de verklaringen voor `host` op te nemen.

② De aanwijzing `next-server` bepaalt de TFTP of NFS server die gebruikt moet worden voor het laden van het lader- of kernelbestand (standaard wordt dezelfde host als voor de DHCP-server gebruikt).

③ De aanwijzing `filename` bepaalt het bestand dat Etherboot of PXE gebruikt voor de volgende uitvoerstap. Het dient gespecificeerd te worden volgens de gebruikte verzendmethode. Voor Etherboot kan tijdens het compileren worden opgegeven of het NFS of TFTP moet gebruiken. De FreeBSD-poort stelt standaard NFS in. PXE gebruikt TFTP, vandaar dat hier een relatieve bestandsnaam wordt gebruikt (dit kan afhangen van de instellingen van de TFTP-server, maar het is de gewoonte). Verder geldt dat PXE pxeboot en niet de kernel laadt. Er zijn andere interessante mogelijkheden, zoals het laden van pxeboot vanuit de map /boot van een FreeBSD CD-ROM (aangezien pxeboot(8) de GENERIC kernel kan laden, bestaat de mogelijkheid om PXE te gebruiken om van een CDROM op afstand op te starten.

④ De optie `root-path` definieert het pad naar het root-bestandssysteem, in de gebruikelijke notatie van NFS. Indien PXE gebruikt wordt, is het mogelijk om het IP-adres van de host weg te laten zolang de kerneloptie BOOTP niet geactiveerd is. De NFS-server is dan dezelfde als die van TFTP.

Configuratie door gebruik van BOOTP

Hieronder staan de equivalente instellingen voor bootpd (gereduceerd tot één cliënt). Dit staat in /etc/bootptab.

Merk op dat Etherboot gecompileerd dient te worden met de afwijkende optie `NO_DHCP_SUPPORT` om BOOTP te gebruiken, en dat PXEDHCP *nodig heeft*. Het enige duidelijke voordeel van bootpd is dat het in het basissysteem zit.

```
.def100:\
  :hn:ht=1:sa=192.168.4.4:vm=rfc1048:\
  :sm=255.255.255.0:\
  :ds=192.168.4.1:\
  :gw=192.168.4.1:\
  :hd="/tftpboot":\
  :bf="/kernel.diskless":\
  :rp="192.168.4.4:/data/misc/diskless":

margaux:ha=0123456789ab:tc=.def100
```

Een opstartprogramma voorbereiden met Etherboot

De website van Etherboot bevat uitgebreide documentatie die over het algemeen is bedoeld voor Linux-systemen, maar die desalniettemin bruikbare informatie bevat. Het volgende geeft een samenvatting over hoe Etherboot op een FreeBSD-systeem te gebruiken.

Ten eerste dient het pakket of de poort net/etherboot geïnstalleerd te worden.

De instellingen van Etherboot (i.e., om TFTP in plaats van NFS te gebruiken) kunnen gewijzigd worden door het bestand Config in de bronmap van Etherboot te bewerken.

Hieronder zal een opstartdiskette gebruikt worden. Raadpleeg voor andere methoden (PROM, of een MS-DOS®-programma) de documentatie van Etherboot.

Om een opstartdiskette te maken, dient er een diskette in het diskettestation van de machine aanwezig te zijn waarop Etherboot is geïnstalleerd, daarna dient er naar de map src in de mapboom van Etherboot gegaan te worden, en het volgende ingetypt te worden:

```
# gmake bin32/apparaatsoort.fd0
```

apparaatsoort hangt af van het soort Ethernetkaart dat in het schijfloze werkstation aanwezig is. Raadpleeg het bestand NIC in dezelfde map om het juiste *apparaatsoort* te bepalen.

Opstarten met PXE

Standaard laadt de lader pxeboot(8) de kernel via NFS. Het kan zodanig gecompileerd worden dat het TFTP gebruikt door de optie `LOADER_TFTP_SUPPORT` in /etc/make.conf te specificeren. Raadpleeg het commentaar in /usr/shared/examples/etc/make.conf voor instructies.

Er zijn nog twee andere opties voor make.conf die nuttig kunnen zijn bij het opzetten van een schijfloze machine die als seriële console gebruikt wordt: `BOOT_PXELDR_PROBE_KEYBOARD`, en `BOOT_PXELDR_ALWAYS_SERIAL`.

Om PXE bij het opstarten van de machine te gebruiken, is het gewoonlijk nodig om de optie `Boot from network` in het BIOS te selecteren, of om een functietoets tijdens de initialisatie van de PC in te typen.

De TFTP en NFS servers instellen

Indien PXE of Etherboot gebruikt wordt, welke is ingesteld om TFTP te gebruiken, is het nodig om tftpd op de bestandsserver aan te zetten:

1. Maak een map aan van waaruit tftpd de bestanden serveert, bijvoorbeeld /tftpboot.
2. Voeg deze regel toe aan /etc/inetd.conf:

   ```
   tftp    dgram   udp wait    root    /usr/libexec/tftpd  tftpd -l -s /tftpboot
   ```

   Het schijnt dat sommige versies van PXE de TCP-versie van TFTP vereisen. In dit geval dient een tweede regel toegevoegd te worden, waarbij `dgram udp` door `stream tcp` vervangen wordt.

3. inetd dient de instellingenbestanden opnieuw te lezen. De regel `inetd_enable="YES"` dient in het bestand /etc/rc.conf aanwezig te zijn voor de juiste werking van deze opdracht:

```
# service inetd restart
```

De map tftpboot kan overal op de server geplaatst worden. De plaats dient zowel in inetd.conf als in dhcpd.conf ingesteld te worden.

In alle gevallen dient er ook voor gezorgd te worden dat NFS aanstaat en dat het juiste bestandssysteem op de NFS-server geëxporteerd wordt.

1. Voeg het volgende toe aan /etc/rc.conf:

   ```
   nfs_server_enable="YES"
   ```

2. Exporteer het bestandssysteem waar de schijfloze root-map zich bevindt door het volgende aan /etc/exports toe te voegen (pas het aankoppelpunt van het volume aan en vervang *margaux corbieres* door de namen van de schijfloze werkstations):

   ```
   /data/misc -alldirs -ro margaux corbieres
   ```

3. mountd dient het instellingenbestand opnieuw te lezen. Indien het nodig was om NFS in /etc/rc.conf tijdens de eerste stap aan te zetten, is het waarschijnlijk gewenst om in plaats hiervan opnieuw op te starten.

   ```
   # service mountd restart
   ```

Een schijfloze kernel bouwen

Indien Etherboot gebruikt wordt, is het nodig om een kernelinstellingenbestand voor de schijfloze cliënt met de volgende opties (naast de gebruikelijke) aan te maken:

```
options BOOTP       # Gebruik BOOTP om het IP-adres en de hostnaam te verkrijgen
options BOOTP_NFSROOT   # NFS-mount het root-bestandssysteem door gebruik te maken van
de informatie van BOOTP
```

Het kan ook gewenst zijn om `BOOTP_NFSV3`, `BOOT_COMPAT`, en `BOOTP_WIRED_TO` te gebruiken (raadpleeg hiervoor NOTES).

De namen van deze opties zijn historisch en enigszins misleidend aangezien ze eigenlijk onverschillig gebruik van DHCP en BOOTP in de kernel mogelijk maken (het is ook mogelijk om strikt gebruik van BOOTP of DHCP te forceren).

De kernel dient gebouwd te worden (zie De FreeBSD-kernel instellen) en gekopieerd te worden naar de plaats die in dhcpd.conf is aangegeven.



Indien PXE gebruikt wordt, is het bouwen van een kernel met bovenstaande opties niet strikt noodzakelijk (maar wel aangeraden). Door deze opties aan te zetten zullen er meer verzoeken voor DHCP tijdens het opstarten van de kernel verstuurd worden, met in sommige speciale gevallen een klein risico op inconsistentie tussen de nieuwe waarden en degenen die door pxeboot(8) zijn ontvangen. Het voordeel van het gebruik van deze opties is dat de hostnaam als een bijverschijnsel wordt ingesteld. In de andere gevallen dient de hostnaam op een andere manier ingesteld te worden, bijvoorbeeld in een cliënt-specifiek bestand rc.conf.



Om laadbaar te zijn met Etherboot, dienen de apparaataanwijzingen in de kernel gecompileerd te worden. Normaalgesproken wordt hiervoor de volgende optie in het instellingenbestand gebruikt (zie het instellingencommentaarbestand NOTES):

```
hints       "GENERIC.hints"
```

Het root-bestandssysteem voorbereiden

Er dient een root-bestandssysteem voor de schijfloze werkstations op de plaats die als `root-path` in dhcpd.conf staat aangegeven aangemaakt te worden.

`make world` gebruiken om het root-bestandssysteem te bevolken

Deze methode is snel en installeert een compleet maagdelijk systeem (niet alleen het root-bestandssysteem) in `DESTDIR`. Hiervoor dient slechts het volgende script uitgevoerd te worden:

```
#!/bin/sh
export DESTDIR=/data/misc/diskless
mkdir -p ${DESTDIR}
cd /usr/src; make buildworld && make buildkernel
make installworld && make installkernel
cd /usr/src/etc; make distribution
```

Nadat dit gedaan is, kunnen /etc/rc.conf en /etc/fstab die in `DESTDIR` geplaatst zijn naar behoefte worden aangepast.

Swapruimte instellen

Indien nodig kan een wisselbestand dat zich op de server bevindt via NFS worden benaderd.

Swapruimte via NFS

De kernel biedt geen ondersteuning om swapruimte via NFS tijdens het opstarten aan te zetten. De swapruimte moet door de opstartscripts worden aangezet, door een beschrijfbaar bestandssysteem aan te koppelen en een wisselbestand aan te maken en aan te zetten. De volgende opdracht maakt een wisselbestand van de juiste grootte aan:

```
# dd if=/dev/zero of=/pad/naar/wisselbestand bs=1k count=1 oseek=100000
```

Om het aan te zetten dient de volgende regel aan /etc/rc.conf te worden toegevoegd:

```
swapfile=/pad/naar/wisselbestand
```

Diverse problemen

Draaien met een alleen-lezen /usr

Indien het schijfloze werkstation is ingesteld om X te draaien, is het nodig om het instellingenbestand van XDM te wijzigen, dat standaard het foutenlogboek in /usr plaatst.

Gebruik maken van een niet-FreeBSD-server

Indien de server voor het root-bestandssysteem geen FreeBSD draait, is het nodig om het root-bestandssysteem op een FreeBSD-machine aan te maken, en het daarna naar de bestemming te kopiëren, door gebruik te maken van `tar` of `cpio`.

In deze situatie zijn er af en toe problemen met de speciale bestanden in /dev, vanwege verschillen in de groottes van grote/kleine integers. Een oplossing voor dit probleem is om een map van de niet-FreeBSD-server te exporteren, deze map op een FreeBSD-machine aan te koppelen, en devfs(5) te gebruiken om de apparaatknooppunten transparant voor de gebruiker toe te wijzen.

29.8. Met PXE en een NFS-root-bestandssysteem opstarten

Het Preboot eXecution Environment (PXE) van Intel® maakt het mogelijk om het besturingssysteem over het netwerk op te starten. Ondersteuning voor PXE wordt normaliter aangeboden in het BIOS van moderne moederborden, waar het kan worden aangezet in de instellingen van het BIOS wat opstarten over het netwerk mogelijk maakt. Een volledig werkende PXE-opstelling vereist ook correct geconfigureerde DHCP- en TFTP-servers.

Wanneer de gastheercomputer opstart, krijgt het informatie over DHCP over waar de intiële bootloader staat via TFTP. Nadat de gastheercomputer deze informatie heeft ontvangen, downloadt het de bootloader via TFTP en voert het vervolgens de bootloader uit. Dit is gedocumenteerd in sectie 2.2.1 van de Preboot Execution Environment (PXE) Specification. In FreeBSD is de bootloader die tijdens het PXE-proces wordt opgehaald /boot/pxeboot. Terwijl /boot/pxeboot wordt uitgevoerd, wordt de kernel van FreeBSD geladen en wordt er verder gegaan met de rest van de opstartprocedure van FreeBSD. Kijk voor meer informatie over het opstartproces van FreeBSD in Het FreeBSD opstartproces.

29.8.1. De `chroot`-omgeving voor het NFS-root-bestandssysteem instellen

1. Kies een map uit voor een installatie van FreeBSD die over NFS aangekoppeld kan worden.

Bijvoorbeeld een map als /b/tftpboot/FreeBSD/install.

```
# export NFSROOTDIR=/b/tftpboot/FreeBSD/install
# mkdir -p ${NFSROOTDIR}
```

2. Stel de NFS-server in door de instructies in NFS instellen op te volgen.
3. Exporteer de map via NFS door het volgende aan /etc/exports toe te voegen:

```
/b -ro -alldirs
```

4. Herstart de NFS-server:

```
# service nfsd restart
```

5. Stel inetd(8) in door de stappen zoals in Instellingen beschreven op te volgen.
6. Voeg de volgende regel toe aan /etc/inetd.conf:

```
tftp dgram udp wait root /usr/libexec/tftpd tftpd -l -s /b/tftpboot
```

7. Herstart inetd:

```
# service inetd restart
```

8. Herbouw de kernel en userland van FreeBSD:

```
# cd /usr/src
# make buildworld
# make buildkernel
```

9. Installeer FreeBSD in de map die over NFS is aangekoppeld:

```
# make installworld DESTDIR=${NFSROOTDIR}
# make installkernel DESTDIR=${NFSROOTDIR}
# make distribution DESTDIR=${NFSROOTDIR}
```

10. Test dat de TFTP-server werkt en dat het de bootloader dat via PXE verkregen zal worden kan downloaden:

```
# tftp localhost
tftp> get FreeBSD/install/boot/pxeboot
```

```
Received 264951 bytes in 0.1 seconds
```

11. Voeg een regel aan ${NFSROOTDIR}/etc/fstab toe om het root-bestandssysteem over NFS aan te koppelen:

```
# Device                                        Mountpoint    FSType
Options  Dump Pass
mijnhost.example.com:/b/tftpboot/FreeBSD/install      /         nfs      ro
0    0
```

    Vervang *mijnhost.example.com* door de hostnaam of het IP-adres van uw NFS-server. In dit voorbeeld wordt het root-bestandssysteem als alleen-lezen aangekoppeld om te voorkomen dat NFS-cliënten per ongeluk de inhoud van het root-bestandssysteem wissen.

12. Stel het root-wachtwoord in voor de chroot(8)-omgeving.

```
# chroot ${NFSROOTDIR}
# passwd
```

    Dit stelt het root-wachtwoord in voor cliëntmachines die over PXE opstarten.

13. Maak root-logins over SSH mogelijk voor cliëntmachines die met PXE opstarten door ${NFSROOTDIR}/etc/ssh/sshd_config te bewerken en de optie `PermitRootLogin` aan te zetten. Dit is gedocumenteerd in sshd_config(5).
14. Pas andere wijzigingen toe aan de chroot(8)-omgeving in ${NFSROOTDIR}. Deze wijzigingen zouden het toevoegen van pakketten met pkg_add(1), het bewerken van het wachtwoordbestand met vipw(8) of het bewerken van amd.conf(8)-projecties voor automatisch aankoppelen kunnen zijn. Bijvoorbeeld:

```
# chroot ${NFSROOTDIR}
# pkg_add -r bash
```

29.8.2. Geheugenbestandssystemen die gebruikt worden door /etc/rc.initdiskless configureren

Als u vanaf een NFS-rootvolume opstart, detecteert /etc/rc dat u over NFS opstartte en draait het het script /etc/rc.initdiskless. Lees het commentaar in dit script om te begrijpen wat er gebeurt. Het is nodig om /etc en /var geheugen-backed te maken omdat deze mappen schrijfbaar moeten zijn, maar de NFS-rootmap is alleen-lezen.

```
# chroot ${NFSROOTDIR}
# mkdir -p conf/base
# tar -c -v -f conf/base/etc.cpio.gz --format cpio --gzip etc
```

```
# tar -c -v -f conf/base/var.cpio.gz --format cpio --gzip var
```

Wanneer het systeem opstart, zullen er geheugen-bestandssystemen voor /etc en /var worden aangemaakt en aangekoppeld, en zal de inhoud van de cpio.gz-bestanden er naartoe worden gekopieerd.

29.8.3. Een DHCP-server prepareren

PXE heeft een geprepareerde TFTP-server en DHCP-server nodig. De DHCP-server hoeft niet per së dezelfde machine te zijn als de TFTP-server, maar het dient bereikbaar te zijn in uw netwerk.

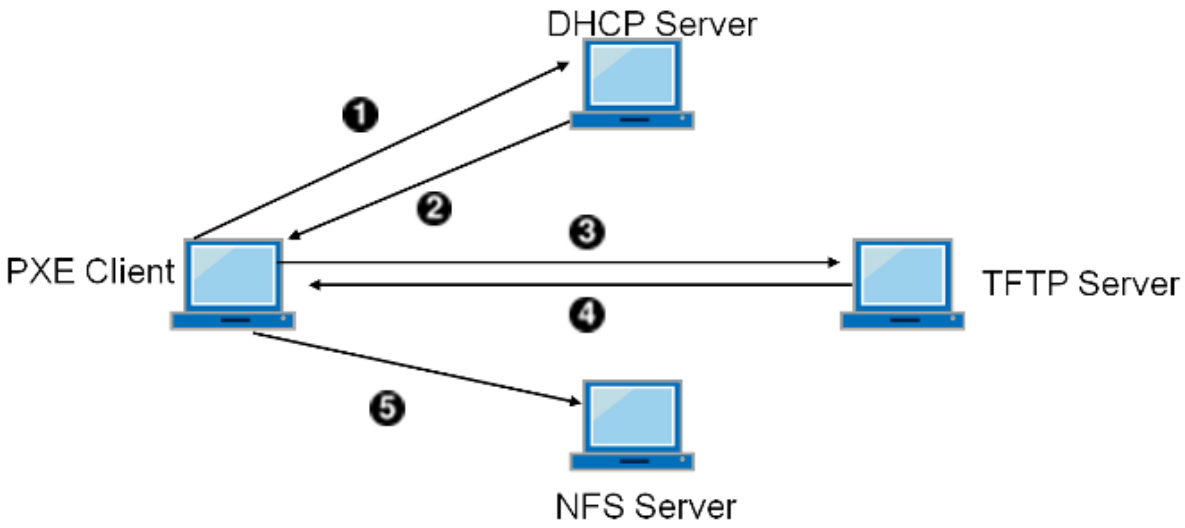
1. Installeer de DHCP-server door de instructies op te volgen zoals beschreven in Een DHCP-server installeren en instellen. Zorg ervoor dat /etc/rc.conf en /usr/local/etc/dhcpd.conf correct zijn geconfigureerd.
2. Stel in /usr/local/etc/dhcpd.conf`next-server`, `filename` en `option root-path` in om het IP-adres van uw TFTP-server, het pad naar /boot/pxeboot en het pad naar het NFS-root-bestandssysteem op te geven. Hier is een voorbeeld van de instellingen voor dhcpd.conf:

```
subnet 192.168.0.0 netmask 255.255.255.0 {
   range 192.168.0.2 192.168.0.3 ;
   option subnet-mask 255.255.255.0 ;
   option routers 192.168.0.1 ;
   option broadcast-address 192.168.0.255 ;
   option domain-name-server 192.168.35.35, 192.168.35.36 ;
   option domain-name "example.com";

   # IP-adres van TFTP server
   next-server 192.168.0.1 ;

   # pad van bootloader verkregen via TFTP
   filename "FreeBSD/install/boot/pxeboot" ;

   # pxeboot bootloader zal proberen om deze map te NFS-mounten voor root-FS
   option root-path "192.168.0.1:/b/tftpboot/FreeBSD/install/" ;
}
```

29.8.4. De PXE-cliënt configureren en verbindingsproblemen opsporen

1. Ga naar het BIOS-configuratiemenu wanneer de cliëntmachine opstart. Stel het BIOS zo in dat het van het netwerk opstart. Indien alle vorige configuratiestappen correct zijn, zou alles "gewoon" moeten werken.
2. Gebruik de poort net/wireshark om netwerkverkeer met betrekking tot het PXE-opstartproces te debuggen, wat geïllustreerd is in onderstaand diagram. In Een DHCP-server prepareren is een voorbeeldconfiguratie gegeven waarbij de DHCP-, TFTP- en NFS-servers op dezelfde machine staan. Deze servers kunnen echter op verschillende machines

staan.



Figuur 1. PXE-opstartproces met NFS-root-mount

3. Controleer dat het bestand pxeboot via TFTP kan worden verkregen. Kijk op uw TFTP-server in /var/log/xferlog om er zeker van de zijn dat het bestand pxeboot van de juiste locatie is opgehaald. Om de configuratie met bovenstaande dhcpd.conf te testen:

```
# tftp 192.168.0.1
tftp> get FreeBSD/install/boot/pxeboot
Received 264951 bytes in 0.1 seconds
```

   Lees tftpd(8) en tftp(1). De BUGS secties in deze pagina's documenteren enkele beperkingen van TFTP.

4. Controleer dat het root-bestandssysteem via NFS kan worden aangekoppeld. Om de configuratie met bovenstaande dhcpd.conf te testen:

```
# mount -t nfs 192.168.0.1:/b/tftpboot/FreeBSD/install /mnt
```

5. Lees de code in src/sys/boot/i386/libi386/pxe.c om te begrijpen hoe de pxeboot-lader variabelen als `boot.nfsroot.server` en `boot.nfsroot.path` instelt. Deze variabelen worden vervolgens gebruikt in de root-aankoppelcode voor diskvrij NFS in src/sys/nfsclient/nfs_diskless.c.
6. Lees pxeboot(8) en loader(8).

29.9. ISDN

Een goede bron voor informatie over de technologie van en hardware over ISDN is Dan Kegel's ISDN Page.

Hieronder staat een snelle eenvoudige handleiding voor ISDN:

- Indien u in Europa leeft is het raadzaam om de sectie over ISDN-kaarten te bestuderen.
- Indien het plan is om ISDN hoofdzakelijk te gebruiken om via een niet-toegewijde inbellijn een verbinding met het Internet te maken, zijn Terminal Adapters wellicht een optie. Dit biedt de meeste flexibiliteit, en de minste problemen bij het wisselen van providers.
- Indien twee LANs met elkaar verbonden worden, of indien er een toegewijde ISDN-verbinding wordt gebruikt om met het Internet te verbinden, is het gebruik van een zelfstandige router/bridge te overwegen.

Financiële kosten zijn een belangrijke factor in de uiteindelijke oplossing. De volgende opties zijn gesorteerd in volgorde van oplopende kosten.

29.9.1. ISDN-kaarten

De ISDN-implementatie in FreeBSD biedt alleen ondersteuning voor de DSS1/Q.931 (of Euro-ISDN) standaard indien passieve kaarten gebruikt worden. Sommige actieve kaarten worden ondersteund indien de firmware ook ondersteuning voor andere signaleringsprotocollen biedt; dit omvat ook de eerst ondersteunde Primary Rate (PRI) ISDN-kaart.

De isdn4bsd-software biedt de mogelijkheid om met andere ISDN-routers te verbinden door òfwel IP over rauwe HDLC òfwel synchrone PPP te gebruiken: òfwel via kernel-PPP met `isppp`, een aangepast stuurprogramma voor sppp(4), òfwel via het gebruikersprogramma ppp(8). Door het gebruikersprogramma ppp(8) te gebruiken, is het combineren van twee of meer ISDN B-kanalen mogelijk. Ook zijn een toepassing die de telefoon beantwoordt en vele gereedschappen zoals een 300 Baud-modem in software beschikbaar.

Een groeiend aantal ISDN-kaarten voor de PC wordt door FreeBSD ondersteund en volgens de rapportages wordt het succesvol in heel Europa en in vele andere delen van de wereld gebruikt.

De ondersteunde passieve ISDN-kaarten zijn meestal uitgerust met de Infineon (voormalig Siemens) ISAC/HSCX/IPAC ISDN-chipsets, maar ook worden ISDN-kaarten ondersteund met chips van Cologne Chip (alleen ISA-bus), PCI-kaarten met Winbond W6692-chips, enkele kaarten met combinaties van Tiger300/320/ISAC chipsets en enkele kaarten die gebaseerd zijn op fabrikantspecifieke chipsets zoals de AVM Fritz!Card PCI V.1.0 en de AVM Fritz!Card PnP.

Momenteel zijn de actieve ISDN-kaarten die ondersteund worden de AVM B1 (ISA en PCI) BRI-kaarten en de AVM T1 PCI PRI-kaarten.

Kijk voor documentatie over isdn4bsd op de homepage van isdn4bsd, welke ook verwijzingen naar tips, errata, en veel meer documentatie zoals het isdn4bsd handboek bevat.

Indien er interesse is om ondersteuning voor een ander ISDN-protocol, een momenteel niet-ondersteunde ISDN-kaart voor de PC, of een andere verbetering voor isdn4bsd toe te voegen, dient er contact opgenomen te worden met Hellmuth Michaelis <hm@FreeBSD.org>.

Voor vragen over het installeren, instellen, en problemen met isdn4bsd oplossen is er een mailinglijst, freebsd-isdn, beschikbaar.

29.9.2. ISDN Terminal Adapters

Terminal adapters (TA) zijn voor ISDN wat modems voor gewone telefoonlijnen zijn.

De meeste TA's gebruiken de standaard opdrachtenverzameling van de Hayes-modem, en kunnen direct als vervanging van een modem gebruikt worden.

Een TA zal als een gewoon modem werken behalve dat de verbindings- en doorvoersnelheden veel hoger zullen zijn dan van het oude modem. Het is noodzakelijk om PPP precies hetzelfde als voor het modem in te stellen. Zorg ervoor dat de seriële snelheid zo hoog mogelijk wordt ingesteld.

Het grootste voordeel van met een TA met een internetprovider te verbinden is de mogelijkheid tot dynamisch PPP. Aangezien IP-adresruimte steeds schaarser wordt, zijn de meeste providers niet meer bereid om een statisch IP te geven. De meeste zelfstandige routers zijn niet in staat tot dynamische IP-toewijzing.

TA's zijn geheel afhankelijk van het PPP-daemon dat gedraaid wordt voor hun mogelijkheden en stabiliteit van de verbinding. Dit maakt het mogelijk om gemakkelijk om op een FreeBSD-machine van een modem naar ISDN over te gaan, indien PPP reeds is ingesteld. Echter, dezelfde problemen die er waren met het PPP-programma zullen blijven voorkomen.

Indien maximale stabiliteit gewenst is, dient de kernel PPP-, niet de gebruikers-PPP-optie gebruikt te worden.

Van de volgende TA's is bekend dat ze met FreeBSD werken:

- Motorola BitSurfer en BitSurfer Pro
- Adtran

De meeste andere TA's zullen waarschijnlijk ook werken, TA-verkopers proberen er zeker van te zijn dat hun product het meeste van de AT-opdrachtverzameling van het standaardmodem accepteert.

Het echte probleem met externe TA's is dat, net zoals bij modems, een goede seriële kaart in de computer nodig is.

Voor een goed begrip van seriële apparaten dient de tutorial FreeBSD Serial Hardware en de verschillen tussen asynchrone en synchrone seriële poorten gelezen te worden.

Een TA die op een standaard seriële poort (asynchroon) van een PC draait beperkt de snelheid tot 115.2 Kbps, zelfs als er een 128 Kbps-verbinding beschikbaar is. Om de volledige 128 Kbps waartoe ISDN in staat is te gebruiken, dient de TA op een synchrone seriële kaart overgeplaatst te worden.

Het kopen van een interne TA voorkomt het probleem van synchroon/asynchroon niet. Interne TA's hebben simpelweg een seriële poortchip van een standaard PC ingebouwd. Dit ontlast de gebruiker alleen van het kopen van nog een seriële kabel en het vinden van nog een leeg elektronisch uitbreidingsslot.

Een synchrone kaart met een TA is minstens zo snel als een zelfstandige router, en wanneer het door een eenvoudige 386 met FreeBSD erop wordt aangestuurd, waarschijnlijk flexibeler.

De keuze tussen synchrone kaart/TA en zelfstandige router is grotendeels religieus. Hierover zijn wat discussies in de mailinglijsten gevoerd. Het wordt aangeraden om de archieven te doorzoeken voor de volledige discussie.

29.9.3. Zelfstandige ISDN bridges/routers

ISDN-bridges of -routers zijn in het geheel niet specifiek voor FreeBSD of enig ander besturingssysteem. Raadpleeg voor een volledigere beschrijving van de technologie van routing en bridging een referentieboek over netwerken.

In deze sectie zullen de termen router en bridge door elkaar worden gebruikt.

Aangezien de prijzen van eenvoudige ISDN-routers/-bridges zakken, zal dit waarschijnlijk een steeds populairdere keuze worden. Een ISDN-router is en kleine doos die direct in het plaatselijke Ethernetnetwerk geprikt wordt, en zijn eigen verbinding met de andere bridge/router beheert. Het heeft ingebouwde software om via PPP en andere populaire protocollen te communiceren.

Een router staat veel snellere doorvoer dan een standaard-TA toe, aangezien het een volledig synchrone ISDN-verbinding zal gebruiken.

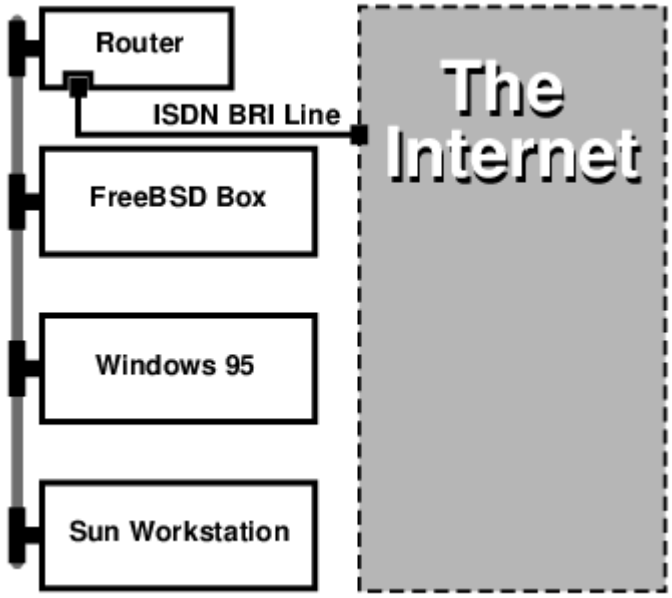
Het grootste probleem met ISDN-routers en -bridges is dat samenwerking tussen fabrikanten nog steeds een probleem kan zijn. Indien er plannen zijn om met een internetprovider te verbinden, is het raadzaam de wensen met hen te bespreken.

Indien er gepland is om twee LAN-segmenten met elkaar te verbinden, zoals het thuis-LAN en het kantoor-LAN, is dit de eenvoudigste en onderhoudarmste oplossing. Aangezien de apparatuur voor beide kanten van de verbinding wordt gekocht is het zeker dat de verbinding zal werken.

De volgende installatie kan worden gebruikt om bijvoorbeeld een thuiscomputer of een netwerk van een afdelingskantoor met een netwerk van het hoofdkantoor te verbinden:

Voorbeeld 28. Netwerk van afdelingskantoor of thuis

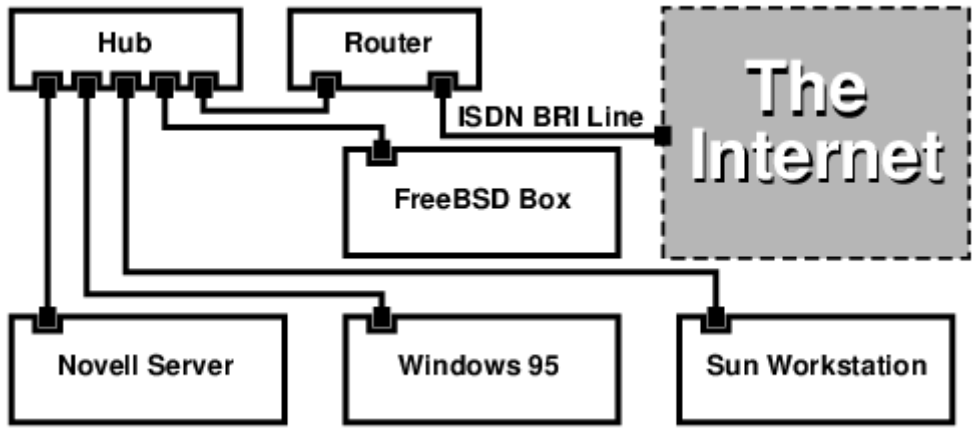
Het netwerk gebruikt een topologie gebaseerd op een bus met een 10 base 2 Ethernet ("thinnet"). Verbind indien nodig de router met de netwerkkabel met een AUI/10BT transceiver.



Wanneer het thuis-/afdelingskantoornetwerk uit slechts één computer bestaat kan een twisted-pair crossover-kabel gebruikt worden om direct met de zelfstandige router te verbinden.

Voorbeeld 29. Hoofdkantoor- of ander LAN

Het netwerk gebruikt een stertopologie met 10 base T Ethernet ("Twisted Pair").



Een groot voordeel van de meeste routers/bridges is dat ze *gelijktijdig* 2 *gescheiden onafhankelijke* PPP-verbindingen met 2 gescheiden sites toestaan. Dit wordt door de meeste TA's niet ondersteund, behalve voor specifieke (gewoonlijk dure) modellen die twee seriële poorten hebben. Dit dient niet met kanaalbinding, MPP, etcetera verward te worden.

Dit kan een erg handige eigenschap zijn indien, bijvoorbeeld, er een toegewijde ISDN-verbinding op kantoor is en het gewenst is om deze af te tappen, maar een andere ISDN-lijn op het werk ongewenst is. Een router op kantoor kan een toegewijde B-kanaal verbinding (64 Kbps) met het Internet beheren en het andere B-kanaal voor een gescheiden gegevensverbinding gebruiken. Het tweede B-kanaal kan voor inbellen, uitbellen, of dynamisch binden (MPP, etcetera) gebruikt worden met het eerste B-kanaal voor meer bandbreedte.

Een Ethernet-bridge staat ook toe om meer dan alleen IP-verkeer te verzenden. Het is ook mogelijk om IPX/SPX of enig ander protocol te gebruiken.

29.10. Network Address Translation

29.10.1. Overzicht

Het Network Address Translation daemon van FreeBSD, in het algemeen bekend als natd(8), is een daemon dat rauwe binnenkomende IP-pakketten accepteert, de bron naar die van de plaatselijke machine verandert en de pakketten terug in de uitgaande IP-pakketstroom injecteert. natd(8) doet dit door het IP-adres en de poort van de bron zo te veranderen dat wanneer de gegevens weer ontvangen worden, het in staat is om de originele plaats van de gegevens te achterhalen en ze door te sturen naar de originele aanvrager.

NAT wordt het meest gebruikt wat in het algemeen bekend is als het delen van een Internetverbinding.

29.10.2. Installatie

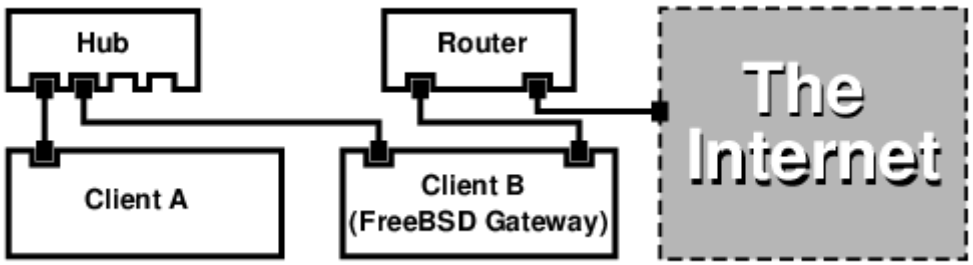
Wegens de krimpende IP-ruimte in IPv4, en het groeiend aantal gebruikers van consumentenlijnen op hoge snelheid zoals kabel of DSL, hebben steeds meer mensen een oplossing als het delen van een Internetverbinding nodig. Vanwege de mogelijkheid om meerdere computers online te verbinden door één verbinding en IP-adres is natd(8) een redelijke keuze.

In de meeste gevallen heeft een gebruiker een machine verbonden met een kabel- of DSL-lijn met één IP-adres en is het gewenst om deze ene verbonden computer te gebruiken om Internettoegang aan meerdere computers over een LAN te geven.

Hiervoor dient de FreeBSD-machine op het Internet dienst doen als gateway. Deze gateway-machine heeft twee NICs nodig - één voor de verbinding met de Internetrouter, de andere voor de verbinding met het LAN. Alle machines op het LAN zijn verbonden door een hub of switch.



Er zijn vele manieren om een LAN via een FreeBSD-gateway met het Internet te verbinden. Dit voorbeeld behandelt slechts een gateway met tenminste twee NICs.



Dit soort installaties wordt in het algemeen gebruikt om een Internetverbinding te delen. Eén van de LAN-machines is verbonden met het Internet. De rest van de machines hebben internettoegang via die "gateway"-machine.

29.10.3. Bootloader-configuratie

De mogelijkheden van de kernel voor network address translation met natd(8) staan niet aan in GENERIC, maar ze kunnen worden voorgeladen tijdens het opstarten door enkele opties aan /boot/loader.conf toe te voegen:

```
ipfw_load="YES"
ipdivert_load="YES"
```

Ook moet de tunable `net.inet.ip.fw.default_to_accept` op `1` worden gezet:

```
net.inet.ip.fw.default_to_accept="1"
```



Het is een goed idee om deze optie aan te zetten tijdens de eerste pogingen om een firewall en NAT gateway te installeren. Op deze manier zal het standaardbeleid van ipfw(8) `allow ip from any to any` zijn in plaats van het minder vrije `deny ip from any to any`, en zal het iets moeilijker zijn om buitengesloten te worden net na het opnieuw opstarten van het systeem.

29.10.4. Kernelconfiguratie

Wanneer modules geen optie zijn of wanneer het gewenst is om alle benodigde mogelijkheden in de draaiende kernel te bouwen, dienen de volgende opties in het kernelinstellingenbestand aanwezig te zijn:

```
options IPFIREWALL
options IPDIVERT
```

De volgende opties kunnen ook van pas komen:

```
options IPFIREWALL_DEFAULT_TO_ACCEPT
options IPFIREWALL_VERBOSE
```

29.10.5. Systeeminstellingen voor het opstarten

Om de firewall en NAT tijdens het opstarten aan te zetten, moet het volgende in /etc/rc.conf staan:

```
gateway_enable="YES" ①
firewall_enable="YES" ②
firewall_type="OPEN" ③
natd_enable="YES"
natd_interface="fxp0" ④
natd_flags="" ⑤
```

① Stelt de machine in om dienst te doen als gateway. Het draaien van `sysctl net.inet.ip.forwarding=1` heeft hetzelfde effect.

② Activeert de firewall-regels in /etc/rc.firewall tijdens het opstarten.

③ Dit specificeert een vooraf gedefinieerde verzameling van firewall-regels die alles binnenlaat. Raadpleeg /etc/rc.firewall voor aanvullende types.

④ Geeft aan welke interface te gebruiken om pakketten naar door te sturen (de interface die met het Internet verbonden is).

⑤ Alle aanvullende instelopties die tijdens het opstarten aan natd(8) worden doorgegeven.

Het gedefinieerd hebben van de bovenstaande opties in /etc/rc.conf zal `natd -interface fxp0` draaien tijdens het opstarten. Dit kan ook handmatig worden gedraaid.

Het is ook mogelijk om een instellingenbestand voor natd(8) te gebruiken als er teveel opties zijn om door te geven. In dit geval dient het instellingenbestand te worden gedefinieerd door de volgende regel aan /etc/rc.conf toe te voegen:

```
natd_flags="-f /etc/natd.conf"
```



Het bestand /etc/natd.conf zal een lijst met instelopties bevatten, één per regel. Het geval in de volgende sectie bijvoorbeeld zal het volgende bestand gebruiken:

```
redirect_port tcp 192.168.0.2:6667 6667
redirect_port tcpc 192.168.0.3:80 80
```

Raadpleeg voor meer informatie over het instellingenbestand het gedeelte over de optie `-f` van de hulppagina natd(8).

Elke machine en interface achter het LAN dient een IP-adres in de privé-netwerkruimte toegewezen te krijgen zoals gedefinieerd in RFC 1918 en een standaard gateway van het interne IP-adres van de natd-machine hebben.

Bijvoorbeeld, cliënt `A` en `B` achter het LAN hebben IP-adressen `192.168.0.2` en `192.168.0.3`, terwijl de LAN-interface van de natd-machine IP-adres `192.168.0.1` heeft. De standaard gateway van cliënt `A` en `B` dient ingesteld te worden op die van de natd-machine, `192.168.0.1`. Voor de externe, of Internet-interface van de natd-machine zijn geen speciale wijzigingen nodig om natd(8) te laten werken.

29.10.6. Poorten omleiden

Het nadeel van natd(8) is dat de LAN-cliënten niet vanaf het Internet toegankelijk zijn. Cliënten op het LAN kunnen uitgaande verbinden naar de wereld maken maar kunnen geen inkomende verbindingen ontvangen. Dit vormt een probleem wanneer geprobeerd wordt om Internetdiensten op een van de LAN-cliëntmachines te draaien. Een eenvoudige om dit te omzeilen is om bepaalde Internetpoorten op de natd-machine om te leiden naar een LAN-cliënt.

Bijvoorbeeld, er draait een IRC-server op cliënt `A`, en er draait een webserver op cliënt `B`. Om dit goed te laten werken, dienen verbindingen die worden ontvangen op poorten 6667 (IRC) en 80 (web) te worden omgeleid naar de respectievelijke machines.

De optie `-redirect_port` dient aan natd(8) met de juiste opties te worden doorgegeven. De syntaxis is als volgt:

```
    -redirect_port proto doelIP:doelPOORT[-doelPOORT]
                [aliasIP:]aliasPOORT[-aliasPOORT]
                [verIP[:verrePOORT[-verrePOORT]]]
```

In het bovenstaand voorbeeld dienen de argumenten te zijn:

```
    -redirect_port tcp 192.168.0.2:6667 6667
-redirectport tcp 192.168.0.3:80 80
```

Dit zal de juiste *tcp*-poorten naar de LAN-cliënt-machines omleiden.

Het argument `-redirect_port` kan worden gebruikt om poortbereiken over individuele poorten aan te geven. Bijvoorbeeld, *tcp 192.168.0.2:2000-3000 2000-3000* zal alle verbindingen die op poorten 2000 tot 3000 worden ontvangen omleiden naar poorten 2000 tot 3000 op cliënt `A`.

Deze opties kunnen worden gebruikt wanneer natd(8) direct wordt gedraaid, wanneer ze zijn geplaatst in de optie `natd_flags=""` van /etc/rc.conf, of wanneer ze via een instellingenbestand worden doorgegeven.

Raadpleeg voor meer instelopties natd(8).

29.10.7. Adressen omleiden

Adressen omleiden is handig wanneer er verschillende IP-adressen beschikbaar zijn, maar ze op één machine moeten zitten. Hiermee kan natd(8) aan elke LAN-cliënt een eigen extern IP-adres toewijzen. Vervolgens overschrijft natd(8) de uitgaande pakketten van de LAN-cliënten met het juiste IP-adres en leidt het al het binnenkomende verkeer op dat ene IP-adres terug naar de specifieke LAN-cliënt. Dit staat ook bekend als statisch NAT. Bijvoorbeeld, de IP-adressen `128.1.1.1`, `128.1.1.2`, en `1281.2..3` behoren toe aan de natd gateway-machine. `128.1.1.1` kan gebruikt worden als het externe IP-adres van de natd gateway-machine, terwijl `128.1.1.2` en `128.1.1.3` terug worden gestuurd naar de LAN-cliënten `A` en `B`.

De syntaxis van `-redirect_address` is als volgt:

```
-redirect_address lokaalIP publiekIP
```

lokaalIP	Het interne IP-adres van de LAN-cliënt.
publiekIP	Het externe IP-adres overeenkomend met de LAN-cliënt.

In het voorbeeld zou dit argument zijn:

```
-redirect_address 192.168.0.2 128.1.1.2
-redirect_address 192.168.0.3 128.1.1.3
```

Net zoals `-redirect_port` worden ook deze argumenten geplaatst in de optie `natd_flags=""` van /etc/rc.conf, of doorgegeven via een instellingenbestand. Met adresomleiding is het omleiden van poorten niet nodig aangezien alle gegevens die op een bepaald IP-adres worden ontvangen worden omgeleidt.

Het externe IP-adres op de natd machine dient actief en naar een externe interface gealiased te zijn. In rc.conf(5) staat hoe dit te doen.

29.11. IPv6

IPv6 (ook bekend als IPng "IP next generation") is de nieuwe versie van het welbekende IP-protocol (ook bekend als IPv4). Net zoals de andere huidige *BSD-systemen, bevat FreeBSD de referentie-implementatie van KAME IPv6. Het FreeBSD-systeem wordt dus geleverd met alles wat nodig is om met IPv6 te experimenteren. Deze sectie richt zich op het ingesteld en draaiend krijgen van IPv6.

In de vroege jaren 1990 werden mensen zich bewust van de snel krimpende adresruimte van IPv4. De uitbreidingssnelheid van het Internet baarde twee grote zorgen:

- Geen adresruimte meer. Tegenwoordig is dit niet zo'n probleem meer aangezien RFC1918 voor privé-adresruimte (`10.0.0.0/8`, `172.16.0.0/12`, en `192.168.0.0/16`) en Network Address Translation (NAT) worden gebruikt.
- De regels in de routeertabellen werden te groot. Dit is tegenwoordig nog steeds een probleem.

IPv6 behandelt deze en vele andere zaken:

- 128-bits adresruimte. Met andere woorden, er zijn theoretisch 340.282.366.920.938.463.463.374.607.431.768.211.456 adressen beschikbaar. Dit betekent dat er ongeveer 6,67 * 10^27 IPv6-adressen per vierkante meter op onze planeet beschikbaar zijn.
- Routers zullen alleen netwerkaggregatie-adressen in hun routeertabellen opslaan en dus de gemiddelde ruimte van een routeertabel verkleinen tot 8192 regels.

IPv6 heeft ook vele andere nuttige eigenschappen zoals:

- Automatische adresconfiguratie (RFC2462)
- Anycast-adressen ("ééen-van-velen")
- Verplichte multicast-adressen
- IPsec (IP security)
- Versimpelde structuur van de headers
- Mobiele IP
- Overgangsmechanismen voor IPv6 naar IPv4

Bekijk voor meer informatie:

- IPv6-overzicht op playground.sun.com
- KAME.net

29.11.1. Achtergrond over IPv6 adressen

Er zijn verschillende soorten IPv6-adressen: unicast, anycast, en multicast.

Unicast-adressen zijn de bekende adressen. Een pakket dat naar een unicast-adres wordt verzonden arriveert precies op de interface dat bij dat adres hoort.

Anycast-adressen zijn syntactisch niet van unicast-adressen te onderscheiden maar ze adresseren een groep interfaces. Een pakket dat bestemd is voor een anycast-adres zal bij de dichtstbijzijnde interface arriveren (in router-metrieken). Anycast-adressen mogen alleen door routers worden gebruikt.

Multicast-adressen identificeren een groep interfaces. Een pakket dat bestemd is voor en multicast-adres zal bij alle interfaces die bij de multicast-groep horen arriveren.



Het broadcast-adres van IPv4 (gewoonlijk `xxx.xxx.xxx.255`) wordt in IPv6 met multicast-adressen uitgedrukt.

Tabel 8. Gereserveerde IPv6-adressen

IPv6-adres	Prefixlengte (bits)	Beschrijving	Opmerkingen
`::`	128 bits	niet gespecificeerd	cf. `0.0.0.0` in IPv4
`::1`	128 bits	teruglusadres	cf. `127.0.0.1` in IPv4
`::00:xx:xx:xx:xx`	96 bits	ingebouwd IPv4	De laagste 32 bits zijn het IPv4-adres. Ook "IPv4 compatibel IPv6-adres" genoemd.
`::ff:xx:xx:xx:xx`	96 bits	IPv4-afgebeeld IPv6-adres	De laagste 32 bits zijn het IPv4-adres. Voor hosts die geen IPv6 ondersteunen.
`fe80::` - `feb::`	10 bits	link-lokaal	cf. teruglusadres in IPv4
`fec0::` - `fef::`	10 bits	site-lokaal	
`ff::`	8 bits	multicast	
`001` (base 2)	3 bits	globale unicast	Alle globale unicast-adressen worden vanuit deze pool toegewezen. De eerste 3 bits zijn "001".

29.11.2. IPv6-adressen lezen

De canonieke vorm wordt weergegeven als: `x:x:x:x:x:x:x:x`, waarbij elke "x" een 16-bits hexadecimale waarde is. Bijvoorbeeld `FEBC:A574:382B:23C1:AA49:4592:4EFE:9982`

Vaak bevat een adres lange deelstrings van allen nullen, daarom kan per adres één zo'n deelstring worden afgekort als "::". Ook kunnen maximaal drie voorlopende "0"'s per hexadecimaal viertal worden weggelaten. Bijvoorbeeld, `fe80::1` komt overeen met de canonieke vorm `fe80:0000:0000:0000:0000:0000:0000:0001`.

Een derde vorm is het schrijven van de laatste 32 bits in de bekende (decimale) IPv4-stijl met punten "." als scheidingstekens. Bijvoorbeeld, `2002::10.0.0.1` komt overeen met de (hexadecimale) canonieke representatie `2002:0000:0000:0000:0000:0000:0a00:0001` wat weer hetzelfde is als `2002::a00:1`.

Op dit punt dient de lezer het volgende te begrijpen:

```
# ifconfig
```

```
rl0: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        inet 10.0.0.10 netmask 0xffffff00 broadcast 10.0.0.255
        inet6 fe80::200:21ff:fe03:8e1%rl0 prefixlen 64 scopeid 0x1
        ether 00:00:21:03:08:e1
    media: Ethernet autoselect (100baseTX )
        status: active
```

`fe80::200:21ff:fe03:8e1%rl0` is een automatisch ingesteld link-lokaal adres. Het is als deel van de automatische instelling vanuit het MAC-adres aangemaakt.

Kijk voor verdere informatie over de structuur van IPv6-adressen op RFC3513.

29.11.3. Verbinding krijgen

Er zijn momenteel vier manieren om met andere IPv6-hosts en -netwerken te verbinden:

- Neem contact op met de Internetprovider om te zien of ze al IPv6 aanbieden.
- SixXS biedt wereldwijd tunnels met eindpunten aan.
- Tunnelen via 6-naar-4 (RFC3068)
- Gebruik de poort net/freenet6 indien er een inbelverbinding wordt gebruikt.

29.11.4. DNS in de IPv6-wereld

Er waren twee soorten DNS-records voor IPv6. De IETF heeft A6-records overbodig verklaard. AAAA-records zijn nu de standaard.

AAAA-records gebruiken gaat rechttoe-rechtaan. Wijs de hostnaam toe aan het nieuwe IPv6-adres dat net ontvangen is door het volgende aan de DNS-bestand voor primaire zones toe te voegen:

```
MIJNHOSTNAAM          AAAA    MIJNIPv6ADRES
```

Vraag het aan de DNS-provider indien de DNS-zones niet zelf worden geserveerd. De huidige versies van bind (versie 8.3 en 9) en dns/djbdns (met de IPv6-patch) ondersteunen AAAA-records.

29.11.5. De benodigde wijzigingen doorvoeren in /etc/rc.conf

IPv6-cliëntinstellingen

Deze instellingen helpen bij het configureren van een machine in het LAN die als cliënt in plaats van router dienst zal doen. Om rtsol(8) automatisch de interface tijdens het opstarten te laten configureren op FreeBSD 9.*X* en nieuwer dient het volgende aan rc.conf toegevoegd te worden:

```
ipv6_prefer="YES"
```

Voeg voor FreeBSD 8.*X* en ouder het volgende toe:

```
ipv6_enable="YES"
```

Voeg het volgende toe om statisch een IP-adres zoals 2001:471:1f11:251:290:27ff:fee0:2093 aan de interface fxp0 toe te voegen voor FreeBSD 9.*X*:

```
ifconfig_fxp0_ipv6="2001:471:1f11:251:290:27ff:fee0:2093 prefixlen 64"
```



Zorg ervoor dat *prefixlen 64* wordt vervangen door de juiste waarde voor het subnet van de computer.

Voeg voor FreeBSD 8.*X* het volgende toe:

```
ipv6_ifconfig_fxp0="2001:471:1f11:251:290:27ff:fee0:2093"
```

Voeg het volgende aan /etc/rc.conf toe om een standaardrouter 2001:471:1f11:251::1 toe te wijzen:

```
ipv6_defaultrouter="2001:471:1f11:251::1"
```

IPv6 router/gateway instellingen

Deze paragraaf helpt bij het opvolgen van de aanwijzingen die de tunnelprovider heeft gegeven en ze om te zetten in instellingen die blijven na een herstart. Om de tunnel tijdens het opstarten te herstellen kan het volgende in /etc/rc.conf gebruikt worden:

Noem de generieke tunnelinterfaces die zullen worden ingesteld, bijvoorbeeld gif0:

```
gif_interfaces="gif0"
```

Om de interface met een lokaal eindpunt *MIJN_IPv4_ADRES* in te stellen naar een ver eindpunt *VER_IPv4_ADRES*:

```
gifconfig_gif0="MIJN_IPv4_ADRES VER_IPv4_ADRES"
```

Voeg het volgende toe om het IPv6-adres dat is toegewezen als het eindpunt van de IPv6-tunnel te gebruiken voor FreeBSD 9.*X* en nieuwer:

```
ifconfig_gif0_ipv6="inet6 MIJN_TOEGEWEZEN_IPv6_TUNNEL_EINDPUNT_ADRES"
```

Voeg voor FreeBSD 8.*X* en eerder het volgende toe:

```
ipv6_ifconfig_gif0="MIJN_TOEGEWEZEN_IPv6_TUNNEL_EINDPUNT_ADRES"
```

Nu hoeft alleen de standaardroute voor IPv6 ingesteld te worden. Dit is de andere kant van de IPv6-tunnel:

```
ipv6_defaultrouter="MIJN_IPv6_VER_TUNNEL_EINDPUNT_ADRES"
```

IPv6-tunnelinstellingen

Indien de server gebruikt wordt om IPv6 tussen de rest van het netwerk en de wereld te routen, is ook de volgende instelling in /etc/rc.conf nodig:

```
ipv6_gateway_enable="YES"
```

29.11.6. Routeradvertentie en automatische hostconfiguratie

Deze sectie helpt bij het instellen van rtadvd(8) om de standaard IPv6-route te adverteren.

Het volgende is nodig in /etc/rc.conf om rtadvd(8) aan te zetten:

```
rtadvd_enable="YES"
```

Het is belangrijk om de interface te specificeren waarop het IPv6-routerverzoek plaatsvindt. Om bijvoorbeeld rtadvd(8) te vertellen om fxp0 te gebruiken:

```
rtadvd_interfaces="fxp0"
```

Nu dient het instellingenbestand /etc/rtadvd.conf aangemaakt te worden. Hier is een voorbeeld:

```
fxp0:\
    :addrs#1:addr="2001:471:1f11:246::":prefixlen#64:tc=ether:
```

Vervang fxp0 door de interface die gebruikt gaat worden.

Vervang vervolgens `2001:471:1f11:246::` met de prefix van uw toewijzing.

Indien een `/64` subnet is toegewezen, hoeft er verder niets veranderd te worden. In andere gevallen dient de juiste waarde voor `prefixlen#` gebruikt te worden.

29.12. Asynchronous Transfer Mode (ATM)

29.12.1. Klassiek IP configureren over ATM (PVCs)

Klassiek IP over ATM (CLIP) is de eenvoudigste methode om Asynchronous Transfer Mode (ATM) met IP te gebruiken. Het kan met geswitchte verbindingen (SVCs) en met permanente verbindingen (PVCs) gebruikt worden. Deze sectie beschrijft hoe een netwerk gebaseerd op PVCs op te zetten.

Volledig geschakelde configuraties

De eerste methode om een CLIP met PVCs op te zetten is om elke machine met elke andere machine in het netwerk te verbinden via een toegewijde PVC. Hoewel dit eenvoudig te configureren is, wordt het onpraktisch voor een groter aantal machines. Dit netwerk gaat ervan uit dat er vier machines in het netwerk zijn, allen verbonden met het ATM netwerk met een ATM adapterkaart. De eerste stap is het plannen van de IP-adressen en de ATM verbindingen tussen de machines. Het volgende wordt gebruikt:

Host	IP-adres
hostA	192.168.173.1
hostB	192.168.173.2
hostC	192.168.173.3
hostD	192.168.173.4

Om een volledig geschakeld net te bouwen is er een ATM-verbinding nodig tussen elk paar machines:

Machines	VPI.VCI koppel
hostA - hostB	0.100
hostA - hostC	0.101
hostA - hostD	0.102
hostB - hostC	0.103
hostB - hostD	0.104

Machines	VPI.VCI koppel
hostC - hostD	0.105

De VPI- en VCI-waarde kunnen aan beide kanten van de verbinding verschillen, maar voor de eenvoud wordt aangenomen dat ze hetzelfde zijn. Vervolgens dienen de ATM-interfaces op elke host geconfigureerd te worden:

```
hostA# ifconfig hatm0 192.168.173.1 up
hostB# ifconfig hatm0 192.168.173.2 up
hostC# ifconfig hatm0 192.168.173.3 up
hostD# ifconfig hatm0 192.168.173.4 up
```

aannemende dat de ATM-interface op alle hosts hatm0 is. Nu dienen de PVCs op hostA geconfigureerd te worden (er wordt aangenomen dat ze reeds op de ATM-switches zijn geconfigureerd, raadpleeg de handleiding van de switch hoe dit te doen).

```
hostA# atmconfig natm add 192.168.173.2 hatm0 0 100 llc/snap ubr
hostA# atmconfig natm add 192.168.173.3 hatm0 0 101 llc/snap ubr
hostA# atmconfig natm add 192.168.173.4 hatm0 0 102 llc/snap ubr

hostB# atmconfig natm add 192.168.173.1 hatm0 0 100 llc/snap ubr
hostB# atmconfig natm add 192.168.173.3 hatm0 0 103 llc/snap ubr
hostB# atmconfig natm add 192.168.173.4 hatm0 0 104 llc/snap ubr

hostC# atmconfig natm add 192.168.173.1 hatm0 0 101 llc/snap ubr
hostC# atmconfig natm add 192.168.173.2 hatm0 0 103 llc/snap ubr
hostC# atmconfig natm add 192.168.173.4 hatm0 0 105 llc/snap ubr

hostD# atmconfig natm add 192.168.173.1 hatm0 0 102 llc/snap ubr
hostD# atmconfig natm add 192.168.173.2 hatm0 0 104 llc/snap ubr
hostD# atmconfig natm add 192.168.173.3 hatm0 0 105 llc/snap ubr
```

Uiteraard kunnen ook andere verkeerscontracten dan UBR worden gebruikt indien de ATM-adapter die ondersteunt. In dit geval wordt de naam van het verkeerscontract gevolgd door de parameters van het verkeer. Hulp voor het gereedschap atmconfig(8) kan verkregen worden met:

```
# atmconfig help natm add
```

of in de hulppagina atmconfig(8).

Dezelfde configuratie kan ook bereikt worden via /etc/rc.conf. Voor hostA wordt dit:

```
network_interfaces="lo0 hatm0"
ifconfig_hatm0="inet 192.168.173.1 up"
natm_static_routes="hostB hostC hostD"
route_hostB="192.168.173.2 hatm0 0 100 llc/snap ubr"
```

```
route_hostC="192.168.173.3 hatm0 0 101 llc/snap ubr"
route_hostD="192.168.173.4 hatm0 0 102 llc/snap ubr"
```

De huidige toestand van alle CLIP routes kan worden verkregen met:

```
hostA# atmconfig natm show
```

29.13. Common Address Redundancy Protocol (CARP)

Het Common Address Redundancy Protocol, of CARP, staat toe dat meerdere hosts hetzelfde IP-adres gebruiken. In sommige opstellingen wordt dit gebruikt voor beschikbaarheid of loadbalancing. Hosts kunnen ook gescheiden IP-adressen gebruiken, zoals in het voorbeeld dat hier is gegeven.

Om ondersteuning voor CARP aan te zetten, dient de FreeBSD-kernel herbouwd zoals beschreven in De FreeBSD-kernel instellen met de volgende optie:

```
device  carp
```

Als alternatief kan de if_carp.ko module geladen worden tijdens het opstarten. Voeg de volgende regel toe aan /boot/loader.conf:

```
if_carp_load="YES"
```

De functionaliteit van CARP zou nu beschikbaar moeten zijn en kan met verschillende `sysctl`-OIDs worden bijgesteld:

OID	Beschrijving
`net.inet.carp.allow`	Accepteer inkomende CARP pakketten. Staat standaard aan.
`net.inet.carp.preempt`	Deze optie zet alle CARP interfaces down op de host wanneer er een down gaat. Staat standaard uit.
`net.inet.carp.log`	De waarde `0` zet alle logging uit. De waarde `1` zet het loggen van slechte CARP-pakketten aan. Waardes hoger dan `1` zet het loggen van toestandsveranderingen van de CARP interfaces aan. De standaardwaarde is `1`.
`net.inet.carp.arpbalance`	Balanceer lokaal netwerkverkeer met ARP. Staat standaard uit.

OID	Beschrijving
net.inet.carp.suppress_preempt	Een alleen-lezen OID die de toestand van preëmptie-onderdrukking weergeeft. Preëmptie kan worden onderdrukt wanneer de verbinding op een interface afwezig is. De waarde 0 betekent dat preëmptie niet onderdrukt is. Elk probleem verhoogt deze OID.

De CARP-apparaten zelf kunnen met het commando ifconfig worden aangemaakt:

```
# ifconfig carp0 create
```

In een echte omgeving hebben deze interfaces unieke identificatienummers, bekend als een VHID, nodig. Dit VHID of Virtual Host Identification zal worden gebruikt om de hosts op het netwerk te onderscheiden.

29.13.1. CARP gebruiken voor serverbeschikbaarheid

Eén gebruik van CARP, zoals boven aangegeven, is serverbeschikbaarheid. Dit voorbeeld geeft failover-ondersteuning voor drie hosts, met allemaal een uniek IP-adres en dezelfde webinhoud. Deze machines zullen samen met een Round Robin DNS configuratie dienst doen. De failover-machine zal twee aanvullende CARP-interfaces hebben, één voor elk van de IP's van de contentservers. Wanneer er een storing optreedt, zou de failover-server het IP-adres van de falende machine moeten oppikken. Dit betekent dat de storing geheel onmerkbaar zou moeten zijn voor de gebruiker. De failover-server heeft dezelfde inhoud en diensten nodig als de andere contentservers waarvoor het moet invallen.

De twee machines dienen identiek geconfigureerd te worden op de gegeven hostnamen en VHIDs na. Dit voorbeeld noemt deze machines respectievelijk hosta.example.org en hostb.example.org. Ten eerste dienen de benodigde regels voor een CARP-configuratie aan rc.conf te worden toegevoegd. Voor hosta.example.org dient het bestand rc.conf de volgende regels te bevatten:

```
hostname="hosta.example.org"
ifconfig_fxp0="inet 192.168.1.3 netmask 255.255.255.0"
cloned_interfaces="carp0"
ifconfig_carp0="vhid 1 pass testpass 192.168.1.50/24"
```

Op hostb.example.org dienen de volgende regels in rc.conf te staan:

```
hostname="hostb.example.org"
ifconfig_fxp0="inet 192.168.1.4 netmask 255.255.255.0"
cloned_interfaces="carp0"
ifconfig_carp0="vhid 2 pass testpass 192.168.1.51/24"
```



Het is erg belangrijk dat de wachtwoorden die met de optie pass aan ifconfig

gegeven zijn, identiek zijn. De carp apparaten zullen alleen luisteren naar en advertenties accepteren van machines met het juiste wachtwoord. Het VHID dient ook verschillend te zijn voor elke machine.

De derde machine, `provider.example.org`, dient voorbereidt te worden op het afhandelen van failover van beide hosts. Deze machine heeft twee carp apparaten nodig, één om elke host af te handelen. De juiste instelregels voor rc.conf zullen ongeveer gelijk zijn aan de volgende:

```
hostname="provider.example.org"
ifconfig_fxp0="inet 192.168.1.5 netmask 255.255.255.0"
cloned_interfaces="carp0 carp1"
ifconfig_carp0="vhid 1 advskew 100 pass testpass 192.168.1.50/24"
ifconfig_carp1="vhid 2 advskew 100 pass testpass 192.168.1.51/24"
```

Met twee carp apparaten is `provider.example.org` in staat om het IP-adres van de andere machine op te pikken wanneer de ene niet meer antwoordt.



De standaard FreeBSD-kernel *kan* preëmptie geactiveerd hebben. In dat geval hoeft `provider.example.org` het IP-adres niet terug te geven aan de originele contentserver. In dit geval kan het nodig zijn dat een beheerder handmatig het IP terug aan de meester moet geven. Het volgende commando dient op `provider.example.org` gegeven te worden:

```
# ifconfig carp0 down && ifconfig carp0 up
```

Dit dient gedaan te worden op de carp interface die met de juiste host overeenkomt.

Op dit moment dient CARP volledig actief en beschikbaar voor testen te zijn. Voor het testen dienen òfwel het netwerken herstart te worden, òf de machines dienen opnieuw opgestart te worden.

Meer informatie is altijd beschikbaar in de hulppagina carp(4)

Deel V: Bijlagen

Bijlage A: FreeBSD verkrijgen

A.1. CD-ROM en DVD uitgevers

A.1.1. Winkelproducten in doos

FreeBSD is beschikbaar in een doos (FreeBSD CD-ROMs, additionele software en gedrukte documentatie) bij verschillende verkopers:

- Frys Electronics
  WWW: http://www.frys.com/

A.1.2. CD-ROMs en DVD's

FreeBSD CD-ROMs en DVD's zijn te koop bij veel online winkels:

- FreeBSD Mall, Inc.
  700 Harvest Park Ste F
  Brentwood, CA 94513
  Verenigde Staten
  Telefoon: +1 925 240-6652
  Fax: +1 925 674-0821
  E–mail: <info@freebsdmall.com>
  WWW: http://www.freebsdmall.com/
- Dr. Hinner EDV
  St. Augustinus-Str. 10
  D-81825 München
  Duitsland
  Telefoon: (089) 428 419
  WWW: http://www.hinner.de/linux/freebsd.html
- JMC Software
  Ierland
  Telefoon: 353 1 6291282
  WWW: http://www.thelinuxmall.com
- Linux Distro UK
  42 Wharfedale Road
  Margate
  CT9 2TB
  Verenigd Koninkrijk
  WWW: https://linux-distro.co.uk/
- The Linux Emporium
  Hilliard House, Lester Way
  Wallingford
  OX10 9TA
  Verenigd Koninkrijk

Telefoon: +44 1491 837010
Fax: +44 1491 837016
WWW: http://www.linuxemporium.co.uk/products/bsd/

- Linux+ DVD Magazine
  Lewartowskiego 6
  Warsaw
  00-190
  Polen
  Telefoon: +48 22 860 18 18
  E–mail: <editors@lpmagazine.org>
  WWW: http://www.lpmagazine.org/
- Linux System Labs Australia
  21 Ray Drive
  Balwyn North
  VIC - 3104
  Australië
  Telefoon: +61 3 9857 5918
  Fax: +61 3 9857 8974
  WWW: http://www.lsl.com.au
- LinuxCenter.Ru
  Galernaya Street, 55
  Saint-Petersburg
  190000
  Rusland
  Telefoon: +7-812-3125208
  E–mail: <info@linuxcenter.ru>
  WWW: http://linuxcenter.ru/shop/freebsd

A.1.3. Distributeurs

Wederverkopers die FreeBSD CD-ROM producten willen verkopen kunnen contact opnemen met een distributeur:

- Ingram Micro
  1600 E. St. Andrew Place
  Santa Ana, CA
  92705-4926 Verenigde Staten
  Telefoon: 1 (800) 456-8000
  WWW: http://www.ingrammicro.com/
- Kudzu, LLC
  7375 Washington Ave. S.
  Edina, MN 55439
  Verenigde Staten
  Telefoon: +1 952 947-0822
  Fax: +1 952 947-0876
  E–mail: <sales@kudzuenterprises.com>

- LinuxCenter.Ru
  Galernaya Street, 55
  Sint-Petersburg
  190000
  Rusland
  Telefoon: +7-812-3125208
  E–mail: <info@linuxcenter.ru>
  WWW: http://linuxcenter.ru/freebsd

A.2. FTP sites

De officiële broncode voor FreeBSD is beschikbaar via anoniem toegankelijke FTP in de hele wereld via vele mirrorsites. De site ftp://ftp.FreeBSD.org/pub/FreeBSD/ heeft een goede verbinding en staat veel verbindingen toe, maar het is waarschijnlijk beter om een mirrorsite te zoeken die "dichterbij" is (zeker als het doel is ook een soort mirrorsite op te zetten).

FreeBSD is beschikbaar via de onderstaande anonieme FTP mirror sites. Bij het kiezen van anonieme FTP voor het verkrijgen van FreeBSD wordt aangeraden een site die dichtbij ligt te kiezen. De mirrorsites die in de lijst staan als "Primaire Mirrorsites" hebben meestal het complete FreeBSD archief (alle beschikbare versies voor alle architecturen) maar downloads zijn waarschijnlijk sneller van een site die in het land of de regio van de gebruiker staat. De regionale sites hebben de meeste recente versies voor de meest populaire architecturen, maar hebben wellicht niet het complete archief. Alle sites geven toegang via anonieme FTP, maar een aantal sites hebben ook andere toegangsmogelijkheden. De toegangsmogelijkheden voor iedere site staan tussen haakjes achter de hostnaam.

Central Servers, Primary Mirror Sites, Armenia, Australia, {mirrors-austria}, Brazil, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hong Kong, Ireland, Japan, Korea, Latvia, Lithuania, Netherlands, New Zealand, Norway, Poland, Russia, Saudi Arabia, Slovenia, South Africa, Spain, Sweden, Switzerland, Taiwan, Ukraine, United Kingdom, United States of America.

(as of UTC)

Central Servers

ftp://ftp.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp.FreeBSD.org/pub/FreeBSD/ / http://ftp.FreeBSD.org/pub/FreeBSD/)

Primary Mirror Sites

In case of problems, please contact the hostmaster <mirror-admin@FreeBSD.org> for this domain.

- ftp://ftp1.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp4.FreeBSD.org/pub/FreeBSD/ / http://ftp4.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp5.FreeBSD.org/pub/FreeBSD/ (ftp)

- ftp://ftp6.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp7.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp10.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp10.FreeBSD.org/pub/FreeBSD/ / http://ftp10.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp11.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp13.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp14.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp14.FreeBSD.org/pub/FreeBSD/)

Armenia

In case of problems, please contact the hostmaster <hostmaster@am.FreeBSD.org> for this domain.

- ftp://ftp1.am.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp1.am.FreeBSD.org/pub/FreeBSD/ / rsync)

Australia

In case of problems, please contact the hostmaster <hostmaster@au.FreeBSD.org> for this domain.

- ftp://ftp.au.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.au.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.au.FreeBSD.org/pub/FreeBSD/ (ftp)

{mirrors-austria}

In case of problems, please contact the hostmaster <{mirrors-austria-email}> for this domain.

- {mirrors-austria-ftp} (ftp / ftpv6 / {mirrors-austria-ftp-http} / {mirrors-austria-ftp-httpv6})

Brazil

In case of problems, please contact the hostmaster <hostmaster@br.FreeBSD.org> for this domain.

- ftp://ftp2.br.FreeBSD.org/FreeBSD/ (ftp / http://ftp2.br.FreeBSD.org/)
- ftp://ftp3.br.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)
- ftp://ftp4.br.FreeBSD.org/pub/FreeBSD/ (ftp)

Czech Republic

In case of problems, please contact the hostmaster <hostmaster@cz.FreeBSD.org> for this domain.

- ftp://ftp.cz.FreeBSD.org/pub/FreeBSD/ (ftp / ftp://ftp.cz.FreeBSD.org/pub/FreeBSD/ / http://ftp.cz.FreeBSD.org/pub/FreeBSD/ / http://ftp.cz.FreeBSD.org/pub/FreeBSD/ / rsync / rsyncv6)
- ftp://ftp2.cz.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp2.cz.FreeBSD.org/pub/FreeBSD/)

Denmark

In case of problems, please contact the hostmaster <staff@dotsrc.org> for this domain.

- ftp://ftp.dk.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp.dk.FreeBSD.org/pub/FreeBSD/ / http://ftp.dk.FreeBSD.org/pub/FreeBSD/)

Estonia

In case of problems, please contact the hostmaster <hostmaster@ee.FreeBSD.org> for this domain.

- ftp://ftp.ee.FreeBSD.org/pub/FreeBSD/ (ftp)

Finland

In case of problems, please contact the hostmaster <hostmaster@fi.FreeBSD.org> for this domain.

- ftp://ftp.fi.FreeBSD.org/pub/FreeBSD/ (ftp)

France

In case of problems, please contact the hostmaster <hostmaster@fr.FreeBSD.org> for this domain.

- ftp://ftp.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp1.fr.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp1.fr.FreeBSD.org/pub/FreeBSD/ / rsync)
- ftp://ftp3.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.fr.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)
- ftp://ftp7.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.fr.FreeBSD.org/pub/FreeBSD/ (ftp)

Germany

In case of problems, please contact the hostmaster <de-bsd-hubs@de.FreeBSD.org> for this domain.

- ftp://ftp.de.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp1.de.FreeBSD.org/freebsd/ (ftp / http://www1.de.FreeBSD.org/freebsd/ / rsync://rsync3.de.FreeBSD.org/freebsd/)
- ftp://ftp2.de.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp2.de.FreeBSD.org/pub/FreeBSD/ / rsync)
- ftp://ftp4.de.FreeBSD.org/FreeBSD/ (ftp / http://ftp4.de.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp5.de.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp7.de.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp7.de.FreeBSD.org/pub/FreeBSD/)

Greece

In case of problems, please contact the hostmaster <hostmaster@gr.FreeBSD.org> for this domain.

- ftp://ftp.gr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.gr.FreeBSD.org/pub/FreeBSD/ (ftp)

Hong Kong

ftp://ftp.hk.FreeBSD.org/pub/FreeBSD/ (ftp)

Ireland

In case of problems, please contact the hostmaster <hostmaster@ie.FreeBSD.org> for this domain.

- ftp://ftp3.ie.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)

Japan

In case of problems, please contact the hostmaster <hostmaster@jp.FreeBSD.org> for this domain.

- ftp://ftp.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp7.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp9.jp.FreeBSD.org/pub/FreeBSD/ (ftp)

Korea

In case of problems, please contact the hostmaster <hostmaster@kr.FreeBSD.org> for this domain.

- ftp://ftp.kr.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)
- ftp://ftp2.kr.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp2.kr.FreeBSD.org/pub/FreeBSD/)

Latvia

In case of problems, please contact the hostmaster <hostmaster@lv.FreeBSD.org> for this domain.

- ftp://ftp.lv.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.lv.FreeBSD.org/pub/FreeBSD/)

Lithuania

In case of problems, please contact the hostmaster <hostmaster@lt.FreeBSD.org> for this domain.

- ftp://ftp.lt.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.lt.FreeBSD.org/pub/FreeBSD/)

Netherlands

In case of problems, please contact the hostmaster <hostmaster@nl.FreeBSD.org> for this domain.

- ftp://ftp.nl.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.nl.FreeBSD.org/os/FreeBSD/ / rsync)
- ftp://ftp2.nl.FreeBSD.org/pub/FreeBSD/ (ftp)

New Zealand

- ftp://ftp.nz.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.nz.FreeBSD.org/pub/FreeBSD/)

Norway

In case of problems, please contact the hostmaster <hostmaster@no.FreeBSD.org> for this domain.

- ftp://ftp.no.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)

Poland

In case of problems, please contact the hostmaster <hostmaster@pl.FreeBSD.org> for this domain.

- ftp://ftp.pl.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp2.pl.FreeBSD.org

Russia

In case of problems, please contact the hostmaster <hostmaster@ru.FreeBSD.org> for this domain.

- ftp://ftp.ru.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.ru.FreeBSD.org/FreeBSD/ / rsync)
- ftp://ftp2.ru.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp2.ru.FreeBSD.org/pub/FreeBSD/ / rsync)
- ftp://ftp5.ru.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp5.ru.FreeBSD.org/pub/FreeBSD/ / rsync)
- ftp://ftp6.ru.FreeBSD.org/pub/FreeBSD/ (ftp)

Saudi Arabia

In case of problems, please contact the hostmaster <ftpadmin@isu.net.sa> for this domain.

- ftp://ftp.isu.net.sa/pub/ftp.freebsd.org (ftp)

Slovenia

In case of problems, please contact the hostmaster <hostmaster@si.FreeBSD.org> for this domain.

- ftp://ftp.si.FreeBSD.org/pub/FreeBSD/ (ftp)

South Africa

In case of problems, please contact the hostmaster <hostmaster@za.FreeBSD.org> for this domain.

- {mirrors-south-africa-ftp} (ftp)
- ftp://ftp2.za.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.za.FreeBSD.org/pub/FreeBSD/ (ftp)

Spain

In case of problems, please contact the hostmaster <hostmaster@es.FreeBSD.org> for this domain.

- ftp://ftp.es.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.es.FreeBSD.org/pub/FreeBSD/)

- ftp://ftp3.es.FreeBSD.org/pub/FreeBSD/ (ftp)

Sweden

In case of problems, please contact the hostmaster <hostmaster@se.FreeBSD.org> for this domain.

- ftp://ftp.se.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.se.FreeBSD.org/pub/FreeBSD/ (ftp / rsync://ftp2.se.FreeBSD.org/)
- ftp://ftp3.se.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.se.FreeBSD.org/pub/FreeBSD/ (ftp / ftp://ftp4.se.FreeBSD.org/pub/FreeBSD/ / http://ftp4.se.FreeBSD.org/pub/FreeBSD/ / http://ftp4.se.FreeBSD.org/pub/FreeBSD/ / rsync://ftp4.se.FreeBSD.org/pub/FreeBSD/ / rsync://ftp4.se.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp6.se.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp6.se.FreeBSD.org/pub/FreeBSD/)

Switzerland

In case of problems, please contact the hostmaster <hostmaster@ch.FreeBSD.org> for this domain.

- ftp://ftp.ch.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.ch.FreeBSD.org/pub/FreeBSD/)

Taiwan

In case of problems, please contact the hostmaster <hostmaster@tw.FreeBSD.org> for this domain.

- ftp://ftp.ch.FreeBSD.org/pub/FreeBSD/ (ftp / ftp://ftp.tw.FreeBSD.org/pub/FreeBSD/ / rsync / rsyncv6)
- ftp://ftp2.tw.FreeBSD.org/pub/FreeBSD/ (ftp / ftp://ftp2.tw.FreeBSD.org/pub/FreeBSD/ / http://ftp2.tw.FreeBSD.org/pub/FreeBSD/ / http://ftp2.tw.FreeBSD.org/pub/FreeBSD/ / rsync / rsyncv6)
- ftp://ftp4.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.tw.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp6.tw.FreeBSD.org/ / rsync)
- ftp://ftp7.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp11.tw.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp11.tw.FreeBSD.org/FreeBSD/)
- ftp://ftp12.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp13.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp14.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp15.tw.FreeBSD.org/pub/FreeBSD/ (ftp)

Ukraine

- ftp://ftp.ua.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.ua.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp6.ua.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp6.ua.FreeBSD.org/pub/FreeBSD /

rsync://ftp6.ua.FreeBSD.org/FreeBSD/)

- ftp://ftp7.ua.FreeBSD.org/pub/FreeBSD/ (ftp)

United Kingdom

In case of problems, please contact the hostmaster <hostmaster@uk.FreeBSD.org> for this domain.

- ftp://ftp.uk.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.uk.FreeBSD.org/pub/FreeBSD/ (ftp / rsync://ftp2.uk.FreeBSD.org/ftp.freebsd.org/pub/FreeBSD/)
- ftp://ftp3.uk.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.uk.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.uk.FreeBSD.org/pub/FreeBSD/ (ftp)

United States of America

In case of problems, please contact the hostmaster <hostmaster@us.FreeBSD.org> for this domain.

- ftp://ftp1.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.us.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp4.us.FreeBSD.org/pub/FreeBSD/ / http://ftp4.us.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp5.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp10.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp11.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp13.us.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp13.us.FreeBSD.org/pub/FreeBSD/ / rsync)
- ftp://ftp14.us.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp14.us.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp15.us.FreeBSD.org/pub/FreeBSD/ (ftp)

A.3. Subversion-sites

Sinds juli 2012 gebruikt FreeBSD Subversion als het primaire versiebeheersysteem om alle broncode van FreeBSD, de documentatie, en de Portscollectie op te slaan.



Subversion is hoofdzakelijk een gereedschap voor ontwikkelaars. De meeste gebruikers dienen FreeBSD Update te gebruiken om het basissysteem van FreeBSD bij te werken, en Portsnap om de FreeBSD Portscollectie bij te werken.

Het spiegelsite-netwerk voor Subversion van FreeBSD bevindt zich nog in de beginfase en zal waarschijnlijk veranderen. Reken er niet op dat deze lijst van spiegelsites statisch is. In het

bijzonder zullen de SSL-certificaten van de servers op een gegeven moment veranderen.

In Subversion worden URLs gebruikt om een depot aan te duiden in de vorm van `protocol://hostnaam/pad`. Spiegelsites kunnen verschillende protocollen ondersteunen zoals hieronder is gespecificeerd. Het eerste gedeelte van het pad is het FreeBSD-depot wat benaderd moet worden. Er zijn drie verschillende depots, `base` voor de broncode van het basissysteem van FreeBSD, `ports` voor de Portscollectie, en `doc` voor de documentatie. De URL `svn://svn0.us-east.FreeBSD.org/ports/head/` specificeert de hoofdtak van het ports-depot op de spiegelsite `svn0.us-east.FreeBSD.org`, gebruikmakend van het svn-protocol.

Alle spiegelsites bevatten alle depots.

De FreeBSD Subversion hoofdserver, `svn.FreeBSD.org`, is publiekelijk toegankelijk als alleen-lezen. Dit kan in de toekomst veranderen, dus gebruikers worden aangeraden om een van de officiële spiegelsites te gebruiken. Gebruik http://svnweb.FreeBSD.org om de Subversion-depots van FreeBSD met een webbrowser te bekijken.

Naam	Protocollen	Locatie	SSL-vingerafdruk
`svn0.us-west.FreeBSD.org`	svn, http, https	Verenigde Staten, Californië	SHA1 `79:35:8F:CA:6D:34:D9:30:44:D1:00:AF:33:4D:E6:11:44:4D:15:EC`
`svn0.us-east.FreeBSD.org`	svn, http, https	Verenigde Staten, New Jersey	SHA1 `06:D1:23:DE:5E:7A:F7:2B:7A:7E:74:95:5F:54:8D:5C:B0:D6:2E:8F`

A.4. Anonieme CVS

A.4.1. Inleiding

Anonieme CVS (of ook wel bekend als *anoncvs*) is een functie die beschikbaar is met de hulpprogramma's die bij FreeBSD zitten om te synchroniseren met een elders aanwezig CVS depot. Het staat gebruikers van FreeBSD onder andere toe om zonder bijzondere rechten alleen-lezen operaties uit te voeren op een van de officiële anoncvs servers van het FreeBSD project. Om het te kunnen gebruiken dient de omgevingsvariabele `CVSROOT` zo ingesteld te worden dat hij wijst naar de gewenste anoncvs server, dient het bekende wachtwoord "anoncvs" bij het commando `cvs login` opgegeven te worden en kan daarna cvs(1) gebruikt worden om het te benaderen als ieder lokaal aanwezig depot.



Het commando `cvs login` slaat de wachtwoorden die voor aanmelden bij de CVS server op in een bestand met de naam .cvspass in de map `HOME`. Als dit bestand niet bestaat, is het mogelijk dat er een foutmelding wordt gegeven als `cvs login` de eerste keer wordt gebruikt. Dat kan opgelost worden door een leeg bestand .cvspass te maken en dan opnieuw aan te melden.

Hoewel de diensten CVSup en *anoncvs* beiden vrijwel dezelfde functie invullen, zijn er redenen die de keuze voor de synchronisatiemethode beïnvloeden. In een notendop is CVSup veel efficiënter in

het gebruik van netwerkbronnen en is het de meest geavanceerde van de twee, maar daar staat iets tegenover. Voor het gebruik van CVSup moet eerst een speciale client geïnstalleerd en ingesteld worden voordat er bits kunnen gaan stromen en dat kan dan alleen in de redelijk grote brokken die in CVSup_collections_ heten.

Anoncvs kan daarentegen gebruikt worden om alles te bekijken van een individueel bestand tot aan een specifiek programma (als `ls` of `grep`) door aan de naam van de CVS module te refereren. Ook anoncvs is alleen geschikt voor alleen-lezen operaties op het CVS depot, dus als het de bedoeling is om lokaal ontwikkelwerk en hetzelfde depot met delen uit het FreeBSD project te combineren, dan biedt alleen CVSup daar een oplossing voor.

A.4.2. Anonieme CVS gebruiken

Het instellen van cvs(1) om gebruik te maken van een Anoniem CVS depot is een kwestie van het instellen van de omgevingsvariabele `CVSROOT` op een van de *anoncvs* servers van het FreeBSD project. Op het moment van schrijven zijn de volgende servers beschikbaar:

- *Frankrijk*: :pserver:anoncvs@anoncvs.fr.FreeBSD.org:/home/ncvs (Gebruik `cvs login` voor pserver-modus en voer het wachtwoord "anoncvs" in wanneer het gevraagd wordt. Voor ssh is geen wachtwoord nodig.)
- *Taiwan*: :pserver:anoncvs@anoncvs.tw.FreeBSD.org:/home/ncvs (Gebruik `cvs login` voor pserver-modus en voer elk willekeurig wachtwoord in wanneer het gevraagd wordt. Voor ssh is geen wachtwoord nodig.)

  ```
  SSH2 HostKey: 1024 02:ed:1b:17:d6:97:2b:58:5e:5c:e2:da:3b:89:88:26
  /etc/ssh/ssh_host_rsa_key.pub
  SSH2 HostKey: 1024 e8:3b:29:7b:ca:9f:ac:e9:45:cb:c8:17:ae:9b:eb:55
  /etc/ssh/ssh_host_dsa_key.pub
  ```

Omdat met CVS vrijwel iedere versie die ooit beschikbaar is geweest "uitgecheckt" kan worden, is het van belang op de hoogte te zijn van de cvs(1) vlag voor revisie (`-r`) en welke waarden zie zoal kan aannemen in het FreeBSD Project depot.

Er zijn twee soorten labels (tags): revisielabels en taklabels (branch). Een revisielabel refereert aan een specifieke revisie. De betekenis blijft van dag tot dag gelijk. Aan de andere kant refereert een taklabel aan de laatste revisie in een bepaalde ontwikkellijn op een bepaald moment. Omdat een taklabel niet refereert aan een specifieke revisie, kan die morgen anders zijn dan vandaag.

CVS labels bevat revisielabels waar gebruikers in geïnteresseerd kunnen zijn. Nogmaals: deze zijn allemaal niet geldig voor de Portscollectie omdat de Portscollectie geen meerdere ontwikkel takken kent.

Als een specifiek taklabel wordt aangegeven, worden als alles goed gaat, de laatste revisies uit een bepaalde ontwikkellijn ontvangen. Als er een oudere versie opgehaald moet worden, kan dat door met de vlag `-D datum` een datum aan te geven. In cvs(1) staan meer details.

A.4.3. Voorbeelden

Hoewel het sterk wordt aangeraden eerst de hulppagina's voor cvs(1) grondig door te lezen, volgen hier een aantal snelle voorbeelden die feitelijk aangeven hoe Anonieme CVS gebruikt kan worden.

Voorbeeld 30. SSH gebruiken om de src/ tree uit te checken:

```
% cvs -d anoncvs@anoncvs1.FreeBSD.org:/home/ncvs co src
The authenticity of host 'anoncvs1.freebsd.org (216.87.78.137)' can't be
established.
DSA key fingerprint is 53:1f:15:a3:72:5c:43:f6:44:0e:6a:e9:bb:f8:01:62.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'anoncvs1.freebsd.org' (DSA) to the list of known
hosts.
```

Voorbeeld 31. Iets uitchecken uit -CURRENT (ls(1)):

```
% setenv CVSROOT :pserver:anoncvs@anoncvs.tw.FreeBSD.org:/home/ncvs
% cvs login
Op de prompt, voer een willekeurig wachtwoord in “wachtwoord”.
% cvs co ls
```

Voorbeeld 32. SSH gebruiken om de src/ structuur uit te checken:

```
% cvs -d freebsdanoncvs@anoncvs.FreeBSD.org:/home/ncvs co src
The authenticity of host 'anoncvs.freebsd.org (128.46.156.46)' can't be
established.
DSA key fingerprint is 52:02:38:1a:2f:a8:71:d3:f5:83:93:8d:aa:00:6f:65.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'anoncvs.freebsd.org' (DSA) to the list of known hosts.
```

Voorbeeld 33. De versie van ls(1) in de 8-STABLE tak uitchecken:

```
% setenv CVSROOT :pserver:anoncvs@anoncvs.tw.FreeBSD.org:/home/ncvs
% cvs login
Op de prompt, voer een willekeurig wachtwoord in “wachtwoord”.
% cvs co -rRELENG_8 ls
```

Voorbeeld 34. Een lijst wijzigingen maken (als unified diffs) voor ls(1)

```
% setenv CVSROOT :pserver:anoncvs@anoncvs.tw.FreeBSD.org:/home/ncvs
```

```
% cvs login
Op de prompt, voer een willekeurig wachtwoord in “wachtwoord”.
% cvs rdiff -u -rRELENG_8_0_0_RELEASE -rRELENG_8_1_0_RELEASE ls
```

Voorbeeld 35. Uitzoeken welke modulenamen gebruikt kunnen worden:

```
% setenv CVSROOT :pserver:anoncvs@anoncvs.tw.FreeBSD.org:/home/ncvs
% cvs login
Op de prompt, voer een willekeurig wachtwoord in “wachtwoord”.
% cvs co modules
% more modules/modules
```

A.4.4. Andere bronnen

De volgende bronnen kunnen bijdragen aan een beter begrip van CVS:

- CVS Tutorial van California Polytechnic State University.
- CVS Home, de CVS gemeenschap voor ontwikkeling en ondersteuning.
- CVSweb is de FreeBSD Project webinterface voor CVS.

A.5. CTM gebruiken

CTM is een methode om een map elders gesynchroniseerd te houden met een centrale. Het is ontwikkeld voor gebruik met de FreeBSD broncode, hoewel sommigen het ook voor andere doeleinden handig vinden. Er bestaat op dit moment weinig tot geen documentatie over het proces van het maken van delta’s. Voor informatie over het gebruik van CTM kan het beste contact gezocht worden met de ctm-users-desc mailinglijst.

A.5.1. Waarom CTM gebruiken?

CTM geeft een lokale kopie van de FreeBSD broncode. Die is in een aantal "smaken" beschikbaar. Of het gaat om slechts één tak of de complete CVS structuur, CTM kan het bieden. CTM is gewoon gemaakt voor actieve ontwikkelaars die met FreeBSD werken, maar geen of een slechte Internetverbinding hebben of gewoon automatisch de laatste wijzigingen willen ontvangen. De meest actieve takken kennen op z’n hoogst drie delta’s per dag. Het is het overwegen waard om ze per automatische mail te laten sturen. De grootte van de updates wordt altijd zo klein mogelijk gehouden. Meestal kleiner dan 5 K en soms (in tien procent van de gevallen) is het 10-50 K. In uitzonderlijke gevallen komt het voor dat een mail van 100 K of meer wordt gestuurd.

Het is wel van belang op de hoogte te zijn van de valkuilen die een rol spelen bij het direct werken met broncode in plaats van met een voorverpakte release. Dit geldt nog meer als wordt gewerkt met de "current" code. Het lezen van Bijblijven met FreeBSD wordt sterk aangeraden.

A.5.2. Wat is er nodig om CTM te gebruiken?

Voor het gebruik van CTM zijn twee dingen nodig: het CTM programma en de initiële delta's om de applicatie te voeden en naar een "current" niveau te komen.

CTM is al onderdeel van FreeBSD sinds versie 2.0 is uitgebracht en is te vinden in /usr/src/usr.sbin/ctm, als de broncode aanwezig is.

De "delta's" voor CTM kunnen op twee manieren komen: met FTP of per e-mail. De volgende FTP sites bieden ondersteuning voor CTM:

ftp://ftp.FreeBSD.org/pub/FreeBSD/CTM/

Er staan er nog meer in de paragraaf mirrors.

FTP de relevante map en download het bestand README vanaf daar.

Voor delta's via e-mail:

Er dient een abonnement genomen te worden op een van de CTM distributielijsten. ctm-src-cur-desc ondersteunt de complete Subversion structuur. ctm-src-cur-desc ondersteunt het hoofd van de ontwikkeltak. ctm-src-9-desc ondersteunt de 9.X release tak, enzovoort. Om te abonneren kan geklikt worden op de bovenstaande links of via https://lists.freebsd.org kan in een lijst geklikt worden op de lijst waarvoor een abonnement gewenst is. De lijstpagina bevat instructies over hoe te abonneren.

Na het ontvangen van CTM updates per mail, kan `ctm_rmail` gebruikt worden voor het uitpakken en verwerken. `ctm_rmail` kan zelfs direct vanuit /etc/aliases gebruikt worden om het proces volledig automatisch te laten verlopen. In de hulppagina van `ctm_rmail` staan meer details.



Welke methode ook gebruikt wordt voor de CTM delta's, het is belangrijk een abonnement te nemen op de ctm-announce-desc mailinglijst. In de toekomst worden alleen op die lijst aankondigingen gedaan over het CTM systeem. Abonneren kan door op de link hierboven te klikken en de instructies te volgen.

A.5.3. CTM de eerste keer gebruiken

Voordat de CTM delta's gebruikt kunnen worden, moet er een startpunt voor bepaald worden.

Eerst moet bepaald worden wat er al is. Het is mogelijk te beginnen vanuit een "lege" map. Dan moet een initiële "Empty" delta gebruikt worden om een door CTM ondersteunde structuur te starten. Het is de bedoeling dat deze "start" delta's ooit voor het gemak op de CD-ROM komen te staan, maar dit is nog niet het geval.

Omdat de structuren tientallen megabytes groot zijn, heeft het de voorkeur om al met iets te beginnen. Als er een -RELEASE CD-ROM beschikbaar is, kan de initiële broncode gekopieerd of uitgepakt worden. Dit bespaart nogal wat dataverkeer.

De "start" delta's kunnen herkend worden aan de `X` die aan het nummer is toegevoegd (bijvoorbeeld src-cur.3210XEmpty.gz). De nummering achter de `X` komt overeen met de oorsprong van het initiële

"zaad". Empty is een lege map. Er wordt in het algemeen iedere honderd delta's een basistransitie voor `Empty` gemaakt. Die zijn trouwens groot: 70 tot 80 Megabytes `gzip` data is normaal voor de XEmpty delta's.

Als er een delta als startpunt is gekozen, zijn ook alle delta's met hogere volgnummers nodig.

A.5.4. CTM in het dagelijks leven gebruiken

Om de delta's toe te passen:

```
# cd /where/ever/you/want/the/stuff
# ctm -v -v /where/you/store/your/deltas/src-xxx.*
```

CTM begrijpt delta's in `gzip` formaat, dus het niet nodig om eerst `gunzip` te gebruiken. Dat spaart diskruimte.

Tenzij het zeker is van de veiligheid van het proces, doet CTM niets met de structuur. Om een delta te verifiëren kan ook de vlag `-c` gebruikt worden en dan komt CTM ook niet aan een structuur. Dan wordt alleen de integriteit van de delta gecontroleerd en of die zonder problemen op de huidige structuur kan worden toegepast.

CTM kent nog meer opties die in de hulppagina's worden besproken.

Meer is er niet. Iedere keer dat er een delta wordt ontvangen, moet die door CTM gehaald worden om de broncode bijgewerkt te houden.

Delta's kunnen het beste niet verwijderd worden als het lastig is ze opnieuw te downloaden. Dan kunnen ze het beste bewaard worden voor het geval er eens iets gebeurt. Zelfs als er alleen floppy's beschikbaar zijn, is het wellicht verstandig die te gebruiken met `fdwrite`.

A.5.5. Lokale wijzigingen behouden

Een ontwikkelaar wil graag experimenteren met bestanden in de structuur en die bestanden veranderen. CTM ondersteunt lokale wijzigingen in beperkte mate: alvorens te kijken of bestand foo bestaat, zoekt het eerst naar foo.ctm. Als dat bestand bestaat, past CTM de wijzigigen daarop toe in plaats van op foo.

Dit gedrag biedt een eenvoudige mogelijkheid om lokale wijzigingen bij te houden. Dat kan dus door bestanden die gewijzigd gaan worden te kopiëren naar een bestand met dezelfde naam met de toevoeging .ctm. Dan kan er vrijelijk gespeeld worden met de code, terwijl CTM het bestand .ctm bijwerkt.

A.5.6. Andere interessante mogelijkheden van CTM

Uitvinden wat precies wordt veranderd met bijwerken

Het is mogelijk een lijst met wijzigingen te maken die CTM zou maken op het broncodedepot met de optie `-l`.

Dit is nuttig als het gewenst is om een logboek bij te houden van de wijzigingen, de te wijzigen bestanden voor- of na te bewerken op welke manier dan ook, of als de gebruiker gewoon een beetje paranoïde is.

Back-ups maken vóór bijwerken

Soms kan het wenselijk zijn om een back-up te maken van alle bestanden die gewijzigd gaan worden door een CTM update.

Met `-B back-upbestand` back-upt CTM alle bestanden die gewijzigd gaan worden door een CTM delta naar back-upbestand.

Te wijzigen bestanden door bijwerken beperken

Soms is het wenselijk de reikwijdte voor een CTM update te beperken of kan het wenselijk zijn om maar een paar bestanden bij te werken uit een aantal delta's.

Een lijst met bestanden die CTM mag bewerken kan aangegeven worden met de opties `-e` en `-x` en het opgeven van regular expressions.

Om bijvoorbeeld een bijgewerkte kopie van lib/libc/Makefile te maken uit de verzameling met opgeslagen CTM delta's, kan het volgende commando uitgevoerd worden:

```
# cd /where/ever/you/want/to/extract/it/
# ctm -e '^lib/libc/Makefile' ~ctm/src-xxx.*
```

Voor ieder te wijzigen bestand in een CTM delta worden de opties `-e` en `-x` toegepast in de volgorde waarin ze op de commandoregel staan. Het bestand wordt alleen door CTM verwerkt als het passend is bevonden na het toepassen van alle parameters in `-e` en `-x`.

A.5.7. Toekomstige plannen voor CTM

Die zijn er:

- Een of andere vorm van authenticatie in het CTM systeem bouwen zodat vervalste CTM updates afgevangen kunnen worden;
- De opties voor CTM opruimen omdat ze verwarrend zijn geworden.

A.5.8. Nog meer

Er zijn ook delta's voor de `ports`collectie, maar daar is nog niet zo veel belangstelling voor.

A.5.9. CTM mirrors

CTM/FreeBSD is op de volgende mirrorsites via anonieme FTP beschikbaar. Als voor CTM anonieme FTP wordt gebruikt, heeft het de voorkeur een site die in geografische zin dichtbij is te gebruiken.

Bij problemen kan contact gezocht worden met de ctm-users-desc mailinglijst.

Californië, Bay Area, officiële bron

- ftp://ftp.FreeBSD.org/pub/FreeBSD/development/CTM/

Zuid-Afrika, back-upserver voor oude delta's

- ftp://ftp.za.FreeBSD.org/pub/FreeBSD/CTM/

Taiwan/R.O.C.

- ftp://ctm.tw.FreeBSD.org/pub/FreeBSD/development/CTM/
- ftp://ctm2.tw.FreeBSD.org/pub/FreeBSD/development/CTM/
- ftp://ctm3.tw.FreeBSD.org/pub/FreeBSD/development/CTM/

Als er geen mirror dichtbij is of als die incompleet is, kan een zoekmachine als allтheweb gebruikt worden.

A.6. CVSup gebruiken

A.6.1. Inleiding

CVSup is een softwarepakket voor het verspreiden en bijwerken van broncodestructuren vanaf een master CVS depot op een andere server. De FreeBSD broncode wordt beheerd in een broncode depot op een centrale ontwikkelmachine in Californië. Met CVSup kunnen FreeBSD gebruikers op eenvoudige wijze hun broncode bijwerken.

CVSup gebruikt een zogenaamd *pull* model voor het bijwerken. In het pull-model vraagt iedere client de server om updates als die nodig zijn. De server wacht passief op een verzoek om updates van zijn clients. Alle updates worden dus op initiatief van de client gedaan. De server stuurt nooit ongevraagde updates. Gebruikers moeten de CVSup client handmatig draaien om te updaten of een `cron` taak instellen om op regelmatige basis bij te werken.

De term CVSup, op de gegeven wijze geschreven, doelt op het complete softwarepakket. De belangrijkste componenten zijn de client `cvsup`, die op de machine van een gebruiker draait, en de server `cvsupd`, die op alle FreeBSD mirrorsites draait.

In de FreeBSD documentatie en op de mailinglijsten zijn referenties aan sup te vinden. Sup was de voorloper van CVSup en diende hetzelfde doel. CVSup wordt op dezelfde manier gebruikt als sup en gebruikt zelfs bestanden met instellingen die ook te gebruiken zijn met `sup`. Sup wordt niet langer gebruikt in het FreeBSD project omdat CVSup sneller en flexibeler is.



De csup applicatie is een herschreven versie van CVSup in de C taal. Het grootste voordeel ervan is dat het sneller is en dat het niet afhankelijk is van de Modula-3 taal, dus dat hoeft niet geïnstalleerd te worden als afhankelijkheid. Sterker nog de applicatie wordt standaard meegeleverd. als ervoor gekozen is om csup te gebruiken, sla dan de installatie stappen voor CVSup over en vervang de referenties naar CVSup met csup terwijl de rest van het artikel gevolgd wordt.

A.6.2. Installatie

De meest eenvoudige wijze van installatie van CVSup is met het voorgecompileerde pakket

net/cvsup uit de FreeBSD pakkettencollectie. Als het gewenst is, kan CVSup ook uit de broncode gebouwd worden in net/cvsup. De port net/cvsup is afhankelijk van het Modula-3 systeem en dat kan wel even duren en er is ook nogal wat schijfruimte voor nodig om het te downloaden en te bouwen.



Als CVSup gebruikt gaat worden op een machine waarop geen Xorg staat, zoals een server, dan dient de port waar geen CVSupGUI bij zit geïnstalleerd te worden: net/cvsup-without-gui.

A.6.3. CVSup instellingen

De werking van CVSup wordt gestuurd door een bestand met instellingen met de naam supfile. Er staan een aantal supfiles als voorbeeld in de map /usr/shared/examples/cvsup/.

De informatie in een supfile beantwoordt de volgende vragen voor CVSup:

- Welke bestanden moeten ontvangen worden?
- Welke versies daarvan moeten ontvangen worden?
- Waar moeten ze vandaan komen?
- Waar moeten ze komen te staan?
- Waar moet `cvsup` zijn statusbestanden bijhouden?

In de volgende paragrafen wordt een supfile bestand opgebouwd door achtereenvolgens alle gestelde vragen te beantwoorden. Als eerste wordt de algemene structuur van een supfile beschreven.

Een supfile is een tekstbestand. Commentaar begint met een `#` en loopt tot het einde van de regel. Lege regels en regels die alleen commentaar bevatten worden genegeerd.

Iedere regel die overblijft slaat op een groep bestanden die ontvangen moet worden. De regel begint met de naam van een "collectie", een logische groep bestanden op de server. De naam van de collectie geeft de server aan welke bestanden er gestuurd moeten worden. Na de naam van de collectie komen er geen of meer velden die gescheiden worden door witruimte. Die velden beantwoorden de hierboven gestelde vragen. Er zijn twee soorten velden: vlagvelden en waardevelden. Een vlagveld bestaat uit een alleenstaand sleutelwoord, bijvoorbeeld `delete` of `compress`. Een waardeveld begint ook met een sleutelwoord, maar het sleutelwoord wordt direct (zonder witruimte) gevolgd door `=` en een tweede woord. `release=cvs` is bijvoorbeeld een waardeveld.

In een supfile wordt meestal aangegeven dat er meerdere collecties ontvangen moeten worden. Het is mogelijk om een supfile te structureren door expliciet alle relevante velden aan te geven voor iedere collectie, maar dat maakt de regels in de supfile nogal lang en het is onhandig omdat de meeste velden hetzelfde zijn voor alle collecties in een supfile. CVSup biedt een systeem met standaardinstellingen om dit probleem te omzeilen. Regels die beginnen met de speciale pseudo-collectienaam `*default` kunnen gebruikt worden om standaarden in te stellen voor de collecties die er in de supfile achteraan komen. Een standaardwaarde kan voor individuele collecties overschreven worden door een andere waarde in de collectie zelf aan te geven. Standaarden

kunnen ook middenin het bestand gewijzigd of aangevuld worden met extra `*default` regels.

Na deze achtergronden wordt er nu een supfile samengesteld voor het ontvangen en bijwerken van de hoofd broncodestructuur van FreeBSD-CURRENT.

- Welke bestanden moeten ontvangen worden?

  De bestanden die via CVSup beschikbaar zijn, zijn beschikbaar in groepen die "collecties" heten. De beschikbare collecties staan beschreven in de volgende paragraaf. In dit voorbeeld is het de bedoeling dat de hele hoofd broncodestructuur voor FreeBSD wordt ontvangen. Daar is één grote collectie voor: `src-all`. De eerste stap in het maken van een supfile is het opsommen van de gewenste collecties, één per regel (in dit geval maar één regel):

  ```
  src-all
  ```

- Welke versies daarvan moeten ontvangen worden?

  Met CVSup kan vrijwel iedere versie van de broncode die ooit heeft bestaan opgehaald worden. Dat kan omdat de cvsupd server direct vanaf het CVS depot werkt, dat alle versies bevat. Er kan aangegeven welke ontvangen moeten worden met de waardevelden `tag=` en `date=`.

  Voorzichtigheid is geboden bij het correct aangeven van velden met `tag=`. Sommige labels zijn alleen geldig voor bepaalde collecties of bestanden. Als ze incorrect worden aangeven of als er een spelfout wordt gemaakt in een label, verwijdert CVSup bestanden waarvan dat waarschijnlijk niet de bedoeling is. Het label `tag=.` dient eigenlijk *alleen* gebruikt te worden voor de `ports-*` collecties.

  Het veld `tag=` benoemt een symbolisch label in het depot. Er zijn twee soorten labels: revisielabels en taklabels. Een revisielabel refereert aan een specifieke revisie. De betekenis blijft altijd hetzelfde. Een taklabel refereert echter aan de laatste revisie van een gegeven ontwikkellijn op een gegeven moment. Omdat een taklabel niet refereert aan een specifieke revisie, kan het morgen iets anders betekenen dan vandaag.

  CVS labels beschrijft de meest interessante taklabels. Als er in het instellingenbestand van CVSup een label wordt aangegeven, moet dat vooraf gegaan worden door `tag=` (`RELENG_8` zal `tag=RELENG_8` worden). Voor de Portscollectie is alleen `tag=.` relevant.

  Labels dienen exact zo ingegeven te worden als ze staan beschreven. CVSup kan geen onderscheid maken tussen geldige en ongeldige labels. Als er een spelfout in een label wordt gemaakt, doet CVSup alsof er een geldig label is ingegeven dat aan geen enkel bestand refereert. Dan zal CVSup de bestaande broncode wissen.

  Bij het aangeven van een taklabel wordt meestal de laatste versie van de bestanden voor een bepaalde ontwikkellijn ontvangen. Om een oudere versie te ontvangen kan in het veld `date=` een datum opgegeven worden. In cvsup(1) staat hoe dat werkt.

Om bijvoorbeeld FreeBSD-CURRENT te ontvangen dient het volgende aan het begin van supfile toegevoegd te worden:

```
*default tag=.
```

Er ontstaat een belangrijk speciaal geval als er geen velden met `tag=` of `date=` worden aangegeven. In dat geval worden de eigenlijke RCS bestanden direct uit het CVS depot van de server ontvangen in plaats van dat een bepaalde versie wordt ontvangen. Ontwikkelaars geven in het algemeen de voorkeur aan deze optie. Door zelf een kopie van de broncode op hun systeem te hebben, krijgen ze de mogelijkheid om zelf door eerdere versies van bestanden te bladeren en de geschiedenis ervan te bekijken. Dit voordeel kost wel veel schijfruimte.

- Waar moeten ze vandaan komen?

  Het veld `host=` wordt gebruikt om `cvsup` aan te geven waar de updates vandaan moeten komen. Dat kan van elke CVSup mirrorsite, hoewel er wordt aangeraden een site die geografisch dichtbij ligt te kiezen. In dit voorbeeld wordt een fictieve FreeBSD distributiesite gebruikt, `cvsup99.FreeBSD.org`:

  ```
  *default host=cvsup99.FreeBSD.org
  ```

  In een werkelijke situatie dient de hostnaam gewijzigd te worden in een host die echt bestaat voordat CVSup gaat draaien. Iedere keer dat `cvsup` wordt gestart, kan er een andere host op de commandoregel opgegeven worden met de optie `-h_hostname_`.

- Waar moeten ze komen te staan?

  Het veld `prefix=` geeft `cvsup` aan waar de ontvangen bestanden terecht moeten komen. In dit voorbeeld worden de bestanden direct in de hoofd broncodestructuur /usr/src geplaatst. De map src is al impliciet in de gekozen collecties, vandaar dat het onderstaande de juiste instelling is:

  ```
  *default prefix=/usr
  ```

- Waar moet `cvsup` zijn statusbestanden bijhouden?

  De CVSup client houdt statusbestanden bij in een map die "base" wordt genoemd. Die bestanden helpen CVSup efficiënter te werken door bij te houden welke updates al eerder zijn ontvangen. Hier wordt de standaard basemap gebruikt, /var/db:

  ```
  *default base=/var/db
  ```

  De bovenstaande instelling wordt standaard gebruikt als die niet wordt aangegeven in de supfile, dus hij is eigenlijk niet nodig.

  Als de basemap niet al bestaat, moet die gemaakt worden. De `cvsup` client weigert te draaien als

de basemap niet bestaat.

- Allerlei supfile instellingen:

  Er is nog een regel die in een supfile moet staan:

  ```
  *default release=cvs delete use-rel-suffix compress
  ```

  `release=cvs` geeft de server aan dat de informatie uit het FreeBSD hoofd CVS depot moet komen. Dat is eigenlijk altijd het geval, maar er zijn mogelijkheden die buiten het bereik van dit handboek vallen.

  `delete` geeft CVSup het recht om bestanden te verwijderen. Dit moet altijd aangegeven worden zodat CVSup de broncode altijd kan bijwerken. CVSup gaat voorzichtig om met het verwijderen van bestanden waar het verantwoordelijk voor is. Extra bestanden in de structuur worden met rust gelaten.

  `use-rel-suffix` is nogal geheimzinnig. Voor de nieuwsgierigen staat er meer over in cvsup(1). Anders kan het gewoon ingesteld worden zonder erover na te denken.

  `compress` schakelt het gebruikt van gzip compressie in voor het communicatiekanaal. Als de verbinding een E1 of sneller is, hoeft er geen compressie gebruikt te worden. Anders helpt het aanzienlijk.

- Alles combinerend:

  Hieronder staat de hele supfile uit het voorbeeld:

  ```
  *default tag=.
  *default host=cvsup99.FreeBSD.org
  *default prefix=/usr
  *default base=/var/db
  *default release=cvs delete use-rel-suffix compress

  src-all
  ```

Het bestand refuse

Zoals hierboven al is aangegeven, gebruikt CVSup een *pull methode*. Dat betekent eigenlijk dat er een verbinding wordt gemaakt met de CVSup server en die zegt dan: "Dit kan er van mij gedownload worden...", en dan antwoordt de client met: "Oké, ik wil dit en dat en zus en zo." Met de standaardinstellingen haalt de CVSup client alle bestanden die bij een collectie en het label horen dat in het bestand met de instellingen is opgegeven. Maar dat is niet altijd wenselijk, in het bijzonder als de doc, ports of www structuren worden gesynchroniseerd. De meeste mensen kunnen geen vier of vijf talen lezen en die hebben de taalspecifieke bestanden dus niet nodig. Als de Portscollectie met CVSup wordt opgehaald, is het mogelijk om iedere collectie apart aan te geven (bijvoorbeeld *ports-astrology*, *ports-biology*, enzovoort, in plaats van eenvoudigweg *ports-all*). Maar omdat de doc en www structuren geen taalspecifieke collecties hebben, moet er gebruik gemaakt

worden van een van de vele mooie mogelijkheden van CVSup: het bestand refuse.

Het bestand refuse geeft CVSup in feite aan dat niet ieder bestand uit een collectie opgehaald moet worden. Het geeft dus aan dat de client bepaalde bestanden van de server moet *weigeren*. Het bestand refuse staat in (of kan gemaakt worden in) base/sup/. *base* staat ingesteld in supfile. De standaardlocatie voor *base* is /var/db. De standaardplaats voor refuse is dus /var/db/sup/refuse.

Het bestand refuse heeft een erg eenvoudige opmaak. Het bevat de namen van de bestanden die niet gedownload mogen worden. Als een gebruiker bijvoorbeeld geen andere talen spreekt dan Engels en Nederlands, maar de Nederlandse vertaling van de documentatie hoeft niet binnengehaald te worden, dan kan het volgende in het bestand refuse gezet worden:

```
doc/bn_*
doc/da_*
doc/de_*
doc/el_*
doc/es_*
doc/fr_*
doc/hu_*
doc/it_*
doc/ja_*
doc/mn_*
doc/nl_*
doc/no_*
doc/pl_*
doc/pt_*
doc/ru_*
doc/sr_*
doc/tr_*
doc/zh_*
```

Dit gaat zo door voor de andere talen. De volledige lijst staat in het FreeBSD CVS depot.

Met deze handige eigenschap kunnen gebruikers met langzamere verbindingen of zij die per minuut voor hun Internetverbinding betalen waardevolle tijd besparen omdat er geen bestanden meer gedownload worden die nooit gebruikt worden. Meer informatie over refuse bestanden en andere leuke mogelijkheden van CVSup staat in de handleiding.

A.6.4. CVSup draaien

Nu kan het bijwerken beginnen. Het commando is best wel eenvoudig:

```
# cvsup supfile
```

De supfile is de naam van het supfile bestand dat gebruikt moet worden. Aangenomen dat er X11 draait op een machine, toont `cvsup` een GUI venster met wat knoppen om de bekende acties uit te voeren. Het proces start na het klikken op de knop btn[go].

Omdat in dit voorbeeld de werkelijke structuur in /usr/src wordt bijgewerkt, moet het programma als `root` uitgevoerd worden, zodat `cvsup` de rechten heeft die het nodig heeft om de bestanden bij te werken. Het is voorstelbaar dat de benodigde rechten, het net gemaakte bestand met instellingen en het voor de eerste keer draaien van een programma zorgt voor wat onrust. Daarom is het mogelijk proef te draaien zonder dat er bestanden gewijzigd worden. Dat kan door ergens een lege map te maken en een extra argument mee te geven op de commandoregel:

```
# mkdir /var/tmp/dest
# cvsup supfile /var/tmp/dest
```

De opgegeven map is de bestemming voor alle bestandsupdates. CVSup bekijkt wel de bestanden in /usr/src, maar wijzigt ze niet. Alle updates belanden in /var/tmp/dest/usr/src. CVSup werkt ook de statusbestanden niet bij als het op deze wijze wordt uitgevoerd. De nieuwe versies van de bestanden worden naar de aangegeven map geschreven. Als er maar leestoegang is tot /usr/src, hoeft een gebruiker zelfs geen `root` te zijn bij het uitvoeren van dit experiment.

Als er geen X11 draait of als het niet wenselijk is een GUI te gebruiken, dan kunnen daarvoor opties op de commandoregel meegegeven worden bij het draaien van `cvsup`:

```
# cvsup -g -L 2 supfile
```

De optie `-g` geeft CVSup aan dat de GUI niet gebruikt hoeft te worden. Dit gebeurt automatisch als X11 niet draait, maar anders moet het aangegeven worden.

De optie `-L 2` geeft CVSup aan dat details getoond moeten worden over alle bestanden die bijgewerkt worden. Er zijn drie niveaus van uitvoerigheid, van `-L 0` tot `-L 2`. Standaard is het 0, wat betekent dat er geen enkel bericht wordt getoond, met uitzondering van foutmeldingen.

Er zijn nog veel andere opties beschikbaar. Met `cvsup -H` wordt een lijst met korte uitleg getoond. Beschrijvingen met meer details staan in de handleiding.

Als het bijwerken op de gewenste manier loopt, kan het regulier draaien van CVSup met cron(8) ingesteld worden. Natuurlijk hoort CVSup zonder GUI te draaien als het programma vanuit de cron(8) draait.

A.6.5. CVSup bestandscollecties

De via CVSup beschikbare bestandscollecties zijn hiërarchisch georganiseerd. Er zijn een paar grote collecties en die zijn opgedeeld in kleinere subcollecties. Het ontvangen van een collectie is hetzelfde als het ontvangen van alle subcollecties. De hiërarchische relatie tussen de collecties wordt hieronder aangegeven door het niveau van inspringen.

De meest gebruikte collecties zijn `src-all` en `ports-all`. De andere collecties worden door kleine groepen mensen gebruikt voor bijzondere doeleinden en sommige mirrorsites hebben ze niet allemaal.

`cvs-all release=cvs`

Het FreeBSD CVS hoofddepot, inclusief de cryptografische code.

`distrib release=cvs`

Bestanden die betrekking hebben op het verspreiden en spiegelen van FreeBSD.

`ports-all release=cvs`

De FreeBSD Portscollectie.



Als `ports-all` (het complete portssysteem) niet bijgewerkt hoeft te worden, maar enkele van de onderstaande subcollecties, dan moet *altijd* ook de `ports-base` subcollectie bijgewerkt worden! Als er iets wijzigt in de infrastructuur van de ports waar `ports-base` voor staat, is het vrijwel zeker dat die wijzigingen heel snel door "echte" ports gebruikt gaan worden. Dus als alleen de "echte" ports bijgewerkt worden en als die gebruik maken van nieuwe mogelijkheden, dan is de kans groot dat het bouwen daarvan foutloopt met een vage foutmelding. Het *eerste* dat gedaan moeten worden is ervoor zorgen dat de `ports-base` subcollectie is bijgewerkt.



Bij het zelf bouwen van een lokale kopie van ports/INDEX*moet* `ports-all` geaccepteerd worden (de hele port structuur). Het bouwen van ports/INDEX met een gedeeltelijke structuur wordt niet ondersteund. Zie ook de FAQ.

`ports-accessibility release=cvs`

Software voor minder valide gebruikers.

`ports-arabic release=cvs`

Ondersteuning voor de Arabische taal.

`ports-archivers release=cvs`

Archiveringshulpmiddelen.

`ports-astro release=cvs`

Astronomie ports.

`ports-audio release=cvs`

Geluidsondersteuning.

`ports-base release=cvs`

De infrastructuur van de Portscollectie. Bestanden uit de mappen Mk/ en Tools/ van /usr/ports.



Zie ook de belangrijke waarschuwing hierboven: deze subcollectie dient *altijd* bijgewerkt te worden als er een onderdeel van de FreeBSD Portscollectie wordt bijgewerkt!

`ports-benchmarks release=cvs`

Benchmarks.

`ports-biology release=cvs`

Biologie.

`ports-cad release=cvs`

Computer aided design programma's.

`ports-chinese release=cvs`

Ondersteuning voor de Chinese taal.

`ports-comms release=cvs`

Communicatiesoftware.

`ports-converters release=cvs`

Karaktercode omzetters.

`ports-databases release=cvs`

Databases.

`ports-deskutils release=cvs`

Dingen die op een bureaublad stonden voordat computers waren uitgevonden.

`ports-devel release=cvs`

Ontwikkelhulpmiddelen.

`ports-dns release=cvs`

DNS gerelateerde software.

`ports-editors release=cvs`

Editors.

`ports-emulators release=cvs`

Emulatoren voor besturingssystemen.

`ports-finance release=cvs`

Monetaire, financiële en gerelateerde applicaties.

`ports-ftp release=cvs`

FTP client en server programma's.

`ports-games release=cvs`

Spelletjes.

`ports-german release=cvs`

Ondersteuning voor de Duitse taal.

`ports-graphics release=cvs`
: Grafische programma's.

`ports-hebrew release=cvs`
: Ondersteuning voor de Hebreeuwse taal.

`ports-hungarian release=cvs`
: Ondersteuning voor de Hongaarse taal.

`ports-irc release=cvs`
: Internet Relay Chat hulpprogramma's.

`ports-japanese release=cvs`
: Ondersteuning voor de Japanse taal.

`ports-java release=cvs`
: Java™ programma's.

`ports-korean release=cvs`
: Ondersteuning voor de Koreaanse taal.

`ports-lang release=cvs`
: Programmeertalen.

`ports-mail release=cvs`
: Mailsoftware.

`ports-math release=cvs`
: Numerieke rekensoftware.

`ports-misc release=cvs`
: Verschillende programma's.

`ports-multimedia release=cvs`
: Multimedia software.

`ports-net release=cvs`
: Netwerksoftware.

`ports-net-im release=cvs`
: Berichtenuitwisseling.

`ports-net-mgmt release=cvs`
: Netwerkbeheersoftware.

`ports-net-p2p release=cvs`
: Peer to Peer Netwerken

`ports-news release=cvs`

USENET news software.

`ports-palm release=cvs`

Softwareondersteuning voor Palm™ apparatuur.

`ports-polish release=cvs`

Ondersteuning voor de Poolse taal.

`ports-ports-mgmt release=cvs`

Programma's om ports en pakketten te beheren.

`ports-portuguese release=cvs`

Ondersteuning voor de Portugese taal.

`ports-print release=cvs`

Printsoftware.

`ports-russian release=cvs`

Ondersteuning voor de Russische taal.

`ports-science release=cvs`

Wetenschappelijk.

`ports-security release=cvs`

Beveiligingsprogramma's.

`ports-shells release=cvs`

Commandoregelshells.

`ports-sysutils release=cvs`

Systeemprogramma's.

`ports-textproc release=cvs`

Tekstverwerkingsprogramma's (zonder desktop publishing).

`ports-ukrainian release=cvs`

Ondersteuning voor de Oekraïense taal.

`ports-vietnamese release=cvs`

Ondersteuning voor de Viëtnamese taal.

`ports-www release=cvs`

Software gerelateerd aan het Wereldwijde Web.

`ports-x11 release=cvs`

Ports voor het X windowsysteem.

`ports-x11-clocks release=cvs`

X11 klokken.

`ports-x11-drivers release=cvs`

X11-stuurprogramma's

`ports-x11-fm release=cvs`

X11 bestandsbeheerders.

`ports-x11-fonts release=cvs`

X11 lettertypen en lettertypeprogramma's.

`ports-x11-toolkits release=cvs`

X11 hulpprogramma's.

`ports-x11-servers release=cvs`

X11 servers.

`ports-x11-themes`

X11 thema's.

`ports-x11-wm release=cvs`

X11 vensterbeheerprogramma's.

`projects-all release=cvs`

Broncode's voor de FreeBSD projecten repository.

`src-all release=cvs`

De hoofdbroncode van FreeBSD, inclusief de cryptografische code.

`src-base release=cvs`

Verschillende bestanden bovenin de /usr/src structuur.

`src-bin release=cvs`

Gebruikersprogramma's die wellicht nodig zijn in single-user modus (/usr/src/bin).

`src-cddl release=cvs`

Programma's en bibliotheken die uitgegeven zijn onder de CDDL licentie (/usr/src/cddl).

`src-contrib release=cvs`

Programma's en bibliotheken van buiten het FreeBSD project die vrijwel ongewijzigd gebruikt worden (/usr/src/contrib).

`src-crypto release=cvs`

Cryptografische programma's en bibliotheken van buiten het FreeBSD project, die vrijwel ongewijzigd worden gebruikt (/usr/src/crypto).

`src-eBones release=cvs`

Kerberos en DES (/usr/src/eBones). Niet gebruikt in recente uitgaves van FreeBSD.

`src-etc release=cvs`

Bestanden met systeeminstellingen (/usr/src/etc).

`src-games release=cvs`

Spelletjes (/usr/src/games).

`src-gnu release=cvs`

Programma's die onder de GNU Public License vallen (/usr/src/gnu).

`src-include release=cvs`

Headerbestanden (/usr/src/include).

`src-kerberos5 release=cvs`

Kerberos5 beveiligingspakket (/usr/src/kerberos5).

`src-kerberosIV release=cvs`

KerberosIV beveiligingspakket (/usr/src/kerberosIV).

`src-lib release=cvs`

Bibliotheken (/usr/src/lib).

`src-libexec release=cvs`

Systeemprogramma's die meestal door andere programma's worden uitgevoerd (/usr/src/libexec).

`src-release release=cvs`

Bestanden die nodig zijn voor het maken van een FreeBSD release (/usr/src/release).

`src-release release=cvs`

Statisch gelinkte programma's voor nood onderhoud, zie rescue(8) (/usr/src/rescue).

`src-sbin release=cvs`

Systeemprogramma's voor single-user modus (/usr/src/sbin).

`src-secure release=cvs`

Cryptografische bibliotheken en commando's (/usr/src/secure).

`src-share release=cvs`

Bestanden die tussen meerdere systemen gedeeld kunnen worden (/usr/src/share).

`src-sys release=cvs`

De kernel (/usr/src/sys).

`src-sys-crypto release=cvs`

Cryptografische kernelcode (/usr/src/sys/crypto).

`src-tools release=cvs`
: Verschillende hulpprogramma's voor het onderhoud van FreeBSD (/usr/src/tools).

`src-usrbin release=cvs`
: Gebruikersprogramma's (/usr/src/usr.bin).

`src-usrsbin release=cvs`
: Systeemprogramma's (/usr/src/usr.sbin).

`distrib release=self`
: De instellingenbestanden van de CVSup server zelf. Gebruikt door de CVSup mirrorsites.

`gnats release=current`
: De GNATS bug-tracking database.

`mail-archive release=current`
: FreeBSD mailinglijstarchief.

`www release=current`
: De voorbewerkte FreeBSD websitebestanden (niet de broncode). Gebruikt door WWW mirrorsites.

A.6.6. Voor meer informatie

De CVSup FAQ en andere informatie over CVSup is te vinden op De CVSup Homepage.

De meeste FreeBSD-gerelateerde discussie over CVSup vindt plaats op de FreeBSD technische discussie mailinglijst. Daar worden nieuwe versies van de software aangekondigd, net als op de FreeBSD aankondigingen mailinglijst.

Voor vragen en foutrapporten moet een kijkje genomen worden op de CVSup FAQ

A.6.7. CVSup sites

CVSup servers voor FreeBSD draaien op de onderstaande sites.

A.7. CVS labels

Bij het ophalen of bijwerken van broncode met cvs of CVSup moet een revisielabel meegegeven worden. Een revisielabel refereert aan een specifieke lijn in de FreeBSD ontwikkeling of aan een specifiek moment in de tijd. Het eerste type heet "taklabel" (branch tag) en het tweede type heet "releaselabel" (release tag).

A.7.1. Taklabels

Deze zijn, met uitzondering van `HEAD` (dat altijd een geldig label is), alleen van toepassing op de src/ structuur. De ports/, doc/ en www/ structuren kennen geen takken.

HEAD

Symbolische naam voor de hoofdlijn van FreeBSD-CURRENT. Ook de standaard als geen revisie is aangegeven.

In CVSup wordt dit label aangegeven met een `.` (dat is dus geen interpunctie, maar een echt `.` karakter).



In CVS is dit de standaard als er geen revisielabel is aangegeven. Het is meestal *geen* goed idee om een checkout of update van CURRENT broncode op een STABLE machine te doen, tenzij dat expliciet de bedoeling is.

RELENG_9

De ontwikkellijn voor FreeBSD-9.X, ook bekend als FreeBSD 9-STABLE.

RELENG_9_0

De uitgavetak voor FreeBSD-9.0, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_8

De ontwikkellijn voor FreeBSD-8.X, ook bekend als FreeBSD 8-STABLE.

RELENG_8_3

De uitgavetak voor FreeBSD-8.3, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_8_2

De uitgavetak voor FreeBSD-8.2, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_8_1

De uitgavetak voor FreeBSD-8.1, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_8_0

De uitgavetak voor FreeBSD-8.0, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_7

De ontwikkellijn voor FreeBSD-7.X, ook bekend als FreeBSD 7-STABLE.

RELENG_7_4

De uitgavetak voor FreeBSD-7.3, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_7_3

De uitgavetak voor FreeBSD-7.3, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_7_2

De uitgavetak voor FreeBSD-7.2, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_7_1

De uitgavetak voor FreeBSD-7.1, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_7_0

De uitgavetak voor FreeBSD-7.0, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_6

De ontwikkellijn voor FreeBSD-6.X, ook bekend als FreeBSD 6-STABLE.

RELENG_6_4

De uitgavetak voor FreeBSD-6.4, alleen gebruikt voor beveiligingsadviezen en andere kritieke reparaties.

RELENG_6_3

De uitgavetak voor FreeBSD-6.3, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_6_2

De releasetak voor FreeBSD-6.2, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_6_1

De releasetak voor FreeBSD-6.1, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_6_0

De releasetak voor FreeBSD-6.0, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_5

De ontwikkellijn voor FreeBSD-5.X, ook bekend als FreeBSD 5-STABLE.

RELENG_5_5

De releasetak voor FreeBSD-5.5, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_5_4

De releasetak voor FreeBSD-5.4, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_5_3

De releasetak voor FreeBSD-5.3, alleen gebruikt voor beveiligingswaarschuwingen en andere

kritische aanpassingen.

RELENG_5_2

De releasetak voor FreeBSD-5.2 en FreeBSD-5.2.1, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_5_1

De releasetak voor FreeBSD-5.1, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_5_0

De releasetak voor FreeBSD-5.0, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_4

De ontwikkellijn voor FreeBSD-4.X, ook bekend als FreeBSD 4-STABLE.

RELENG_4_11

De releasetak voor FreeBSD-4.11, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_4_10

De releasetak voor FreeBSD-4.10, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_4_9

De releasetak voor FreeBSD-4.9, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_4_8

De releasetak voor FreeBSD-4.8, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_4_7

De releasetak voor FreeBSD-4.7, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_4_6

De releasetak voor FreeBSD-4.6 en FreeBSD-4.6.2, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_4_5

De releasetak voor FreeBSD-4.5, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_4_4

De releasetak voor FreeBSD-4.4, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_4_3

De releasetak voor FreeBSD-4.3, alleen gebruikt voor beveiligingswaarschuwingen en andere kritische aanpassingen.

RELENG_3

De ontwikkellijn voor FreeBSD-3.X, ook bekend als 3.X-STABLE.

RELENG_2_2

De ontwikkellijn voor FreeBSD-2.2.X, ook bekend als 2.2-STABLE. Deze tak is sterk verouderd.

A.7.2. Releaselabels

Deze labels refereren aan een specifiek moment in de tijd waarop een versie van FreeBSD is uitgegeven. Het proces om tot een release te komen is gedetailleerder beschreven in de Release Engineering Informatie en Release Proces documenten. De src structuur gebruikt labelnamen die beginnen met `RELENG_` labels. De ports en doc structuren gebruiken labels waarvan de naam begint met het label `RELEASE`. De www tenslotte, is niet gemarkeerd met een bijzondere naam bij uitgaven.

RELENG_9_0_0_RELEASE

FreeBSD 9.0

RELENG_8_3_0_RELEASE

FreeBSD 8.3

RELENG_8_2_0_RELEASE

FreeBSD 8.2

RELENG_8_1_0_RELEASE

FreeBSD 8.1

RELENG_8_0_0_RELEASE

FreeBSD 8.0

RELENG_7_4_0_RELEASE

FreeBSD 7.4

RELENG_7_3_0_RELEASE

FreeBSD 7.3

RELENG_7_2_0_RELEASE

FreeBSD 7.2

RELENG_7_1_0_RELEASE

FreeBSD 7.1

RELENG_7_0_0_RELEASE

FreeBSD 7.0

RELENG_6_4_0_RELEASE
FreeBSD 6.4

RELENG_6_3_0_RELEASE
FreeBSD 6.3

RELENG_6_2_0_RELEASE
FreeBSD 6.2

RELENG_6_1_0_RELEASE
FreeBSD 6.1

RELENG_6_0_0_RELEASE
FreeBSD 6.0

RELENG_5_5_0_RELEASE
FreeBSD 5.5

RELENG_5_4_0_RELEASE
FreeBSD 5.4

RELENG_4_11_0_RELEASE
FreeBSD 4.11

RELENG_5_3_0_RELEASE
FreeBSD 5.3

RELENG_4_10_0_RELEASE
FreeBSD 4.10

RELENG_5_2_1_RELEASE
FreeBSD 5.2.1

RELENG_5_2_0_RELEASE
FreeBSD 5.2

RELENG_4_9_0_RELEASE
FreeBSD 4.9

RELENG_5_1_0_RELEASE
FreeBSD 5.1

RELENG_4_8_0_RELEASE
FreeBSD 4.8

RELENG_5_0_0_RELEASE
FreeBSD 5.0

RELENG_4_7_0_RELEASE
FreeBSD 4.7

RELENG_4_6_2_RELEASE
FreeBSD 4.6.2

RELENG_4_6_1_RELEASE
FreeBSD 4.6.1

RELENG_4_6_0_RELEASE
FreeBSD 4.6

RELENG_4_5_0_RELEASE
FreeBSD 4.5

RELENG_4_4_0_RELEASE
FreeBSD 4.4

RELENG_4_3_0_RELEASE
FreeBSD 4.3

RELENG_4_2_0_RELEASE
FreeBSD 4.2

RELENG_4_1_1_RELEASE
FreeBSD 4.1.1

RELENG_4_1_0_RELEASE
FreeBSD 4.1

RELENG_4_0_0_RELEASE
FreeBSD 4.0

RELENG_3_5_0_RELEASE
FreeBSD-3.5

RELENG_3_4_0_RELEASE
FreeBSD-3.4

RELENG_3_3_0_RELEASE
FreeBSD-3.3

RELENG_3_2_0_RELEASE
FreeBSD-3.2

RELENG_3_1_0_RELEASE
FreeBSD-3.1

RELENG_3_0_0_RELEASE

FreeBSD-3.0

RELENG_2_2_8_RELEASE

FreeBSD-2.2.8

RELENG_2_2_7_RELEASE

FreeBSD-2.2.7

RELENG_2_2_6_RELEASE

FreeBSD-2.2.6

RELENG_2_2_5_RELEASE

FreeBSD-2.2.5

RELENG_2_2_2_RELEASE

FreeBSD-2.2.2

RELENG_2_2_1_RELEASE

FreeBSD-2.2.1

RELENG_2_2_0_RELEASE

FreeBSD-2.2.0

A.8. rsync sites

De volgende sites bieden FreeBSD aan via het protocol rsync. Het programma rsync werkt vrijwel hetzelfde als rcp(1), maar kent meer mogelijkheden en gebruikt het rsync remote-update protocol, dat alleen verschillen tussen twee groepen bestanden overbrengt, waardoor het synchroniseren via een netwerk drastisch wordt versneld. Dit kan het beste gedaan worden als er een mirrorsite voor de FreeBSD FTP server of het FreeBSD CVS depot draait. De rsync suite is voor veel besturingssystemen beschikbaar. Voor FreeBSD kan het pakket of de port uit net/rsync geïnstalleerd worden.

Tsjechië

rsync://ftp.cz.FreeBSD.org/

Beschikbare collecties:

- ftp: een gedeeltelijke mirror van de FreeBSD FTP server.
- FreeBSD: een volledige mirror van de FreeBSD FTP server.

Nederland

rsync://ftp.nl.FreeBSD.org/

Beschikbare collecties:

- FreeBSD: een volledige mirror van de FreeBSD FTP server.

Rusland

rsync://ftp.mtu.ru/

Beschikbare collecties:

- FreeBSD: een volledige spiegel van de FTP-server van FreeBSD.
- FreeBSD-gnats: De GNATS bug-tracking database.
- FreeBSD-archief: spiegel van de FreeBSD Archive FTP-server.

Zweden

rsync://ftp4.se.freebsd.org/

Beschikbare verzamelingen:

- FreeBSD: een volledige spiegel van de FTP-server van FreeBSD.

Taiwan

rsync://ftp.tw.FreeBSD.org/

rsync://ftp2.tw.FreeBSD.org/

rsync://ftp6.tw.FreeBSD.org/

Beschikbare collecties:

- FreeBSD: een volledige mirror van de FreeBSD FTP server.

Verenigd Koninkrijk

rsync://rsync.mirrorservice.org/

Beschikbare collecties:

- sites/ftp.freebsd.org: een volledige mirror van de FreeBSD FTP server.

Verenigde Staten van Amerika

rsync://ftp-master.FreeBSD.org/

Deze server mag alleen gebruikt worden door FreeBSD primaire mirrorsites.

Beschikbare collecties:

- FreeBSD: het masterarchief van de FreeBSD FTP server.
- acl: de FreeBSD master ACL lijst.

  rsync://ftp13.FreeBSD.org/

  Beschikbare collecties:

- FreeBSD: een volledige mirror van de FreeBSD FTP server.

Bijlage B: Bibliography

Hoewel de handleiding de juiste referentie is voor individuele stukken van het FreeBSD besturingssysteem, staan ze erom bekend niet te illustreren hoe de stukken in elkaar vallen om het hele besturingssysteem soepel te laten draaien. Daarom wordt er gesteld dat er geen vervanger is voor een goed boek over UNIX® systeembeheer en een goede gebruikershandleiding.

B.1. Boeken & tijdschriften over FreeBSD

Internationale boeken & Tijdschriften:

- Using FreeBSD (Traditioneel Chinees), gepubliceerd door Drmaster, 1997. ISBN 9-578-39435-7.
- FreeBSD Unleashed (Versimpelde Chinese vertaling), gepubliceerd door China Machine Press. ISBN 7-111-10201-0.
- FreeBSD From Scratch Second Edition (Versimpeld Chinees), gepubliceerd door China Machine Press. ISBN 7-111-10286-X.
- FreeBSD Handbook Second Edition (Versimpeld Chinese vertaling), gepubliceerd door Posts & Telecom Press. ISBN 7-115-10541-3.
- FreeBSD & Windows (Versimpeld Chinees), gepubliceerd door gepubliceerd door China Railway Publishing House. ISBN 7-113-03845-X
- FreeBSD Internet Services HOWTO (Versimpeld Chinees), gepubliceerd door China Railway Publishing House. ISBN 7-113-03423-3
- FreeBSD (Japans), gepubliceerd door CUTT. ISBN 4-906391-22-2 C3055 P2400E.
- Complete Introduction to FreeBSD (Japans), gepubliceerd door Shoeisha Co., Ltd. ISBN 4-88135-473-6 P3600E.
- Personal UNIX Starter Kit FreeBSD (Japans), gepubliceerd door ASCII. ISBN 4-7561-1733-3 P3000E.
- FreeBSD Handbook (Japanse vertaling), gepubliceerd door ASCII. ISBN 4-7561-1580-2 P3800E.
- FreeBSD mit Methode (Duits), gepubliceerd door Computer und Literatur Verlag/Vertrieb Hanser, 1998. ISBN 3-932311-31-0.
- FreeBSD de Luxe (Duits), gepubliceerd door Verlag Modere Industrie, 2003. ISBN 3-8266-1343-0.
- FreeBSD Install and Utilization Manual (Japans), gepubliceerd door Mainichi Communications Inc..
- Onno W Purbo, Dodi Maryanto, Syahrial Hubbany, Widjil Widodo *Building Internet Server with FreeBSD* (Indonesisch), published by Elex Media Komputindo.
- Absolute BSD: The Ultimate Guide to FreeBSD (Traditioneel Chinese vertaling), gepubliceerd door GrandTech Press, 2003. ISBN 986-7944-92-5.
- The FreeBSD 6.0 Book (Traditioneel Chinees), gepubliceerd door Drmaster, 2006. ISBN 9-575-27878-X.

Engelstalige boeken & Tijdschriften:

- Absolute FreeBSD, 2e editie: The Complete Guide to FreeBSD, gepubliceerd door No Starch Press, 2007. ISBN: 978-1-59327-151-0
- The Complete FreeBSD, gepubliceerd door O'Reilly, 2003. ISBN: 0596005164
- The FreeBSD Corporate Networker's Guide, gepubliceerd door Addison-Wesley, 2000. ISBN: 0201704811
- FreeBSD: An Open-Source Operating System for Your Personal Computer, gepubliceerd door The Bit Tree Press, 2001. ISBN: 0971204500
- Teach Yourself FreeBSD in 24 Hours, gepubliceerd door Sams, 2002. ISBN: 0672324245
- FreeBSD unleashed, gepubliceerd door Sams, 2006. ISBN: 0672328755
- FreeBSD: The Complete Reference, gepubliceerd door McGrawHill, 2003. ISBN: 0072224096
- BSD Magazine, gepubliceerd door Software Press Sp. z.o.o. SK. ISSN 1898-9144

B.2. Voor gebruikers

- Ohio State University heeft een UNIX Introductiecursus geschreven welke online in HTML en in PostScript® formaat beschikbaar is.

  Een Italiaanse vertaling van dit document is beschikbaar als onderdeel van het FreeBSD Italian Documentation Project.

- Jpman Project, Japan FreeBSD Users Group. FreeBSD User's Reference Manual (Japanse vertaling). Mainichi Communications Inc., 1998. ISBN4-8399-0088-4 P3800E.
- Edinburgh University heeft een Online Guide geschreven voor nieuwkomers in de UNIX® omgeving.

B.3. Voor beheerders

- Jpman Project, Japan FreeBSD Users Group. FreeBSD System Administrator's Manual (Japanse vertaling). Mainichi Communications Inc., 1998. ISBN4-8399-0109-0 P3300E.
- Dreyfus, Emmanuel. Cahiers de l'Admin: BSD 2nd Ed. (Frans), Eyrolles, 2004. ISBN 2-212-11463-X

B.4. Voor programmeurs

- Computer Systems Research Group, UC Berkeley. *4.4BSD Programmer's Reference Manual*. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-078-3
- Computer Systems Research Group, UC Berkeley. *4.4BSD Programmer's Supplementary Documents*. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-079-1
- Harbison, Samuel P. and Steele, Guy L. Jr. *C: A Reference Manual*. 4th ed. Prentice Hall, 1995. ISBN 0-13-326224-3
- Kernighan, Brian and Dennis M. Ritchie. *The C Programming Language.*. 2nd Ed. PTR Prentice Hall, 1988. ISBN 0-13-110362-8
- Lehey, Greg. *Porting UNIX Software*. O'Reilly & Associates, Inc., 1995. ISBN 1-56592-126-7

- Plauger, P. J. *The Standard C Library*. Prentice Hall, 1992. ISBN 0-13-131509-9
- Spinellis, Diomidis. Code Reading: The Open Source Perspective. Addison-Wesley, 2003. ISBN 0-201-79940-5
- Marshall Kirk McKusick, George V. Neville-Neil. *The Design and Implementation of the FreeBSD UNIX Operating System*. Boston, Mass. : Addison-Wesley, 2004. ISBN 0-201-70245-2
- Spinellis, Diomidis. Code Quality: The Open Source Perspective. Addison-Wesley, 2006. ISBN 0-321-16607-8
- Stevens, W. Richard. *Advanced Programming in the UNIX Environment*. Reading, Mass. : Addison-Wesley, 2005. ISBN 0-201-43307-9
- Stevens, W. Richard. *UNIX Network Programming*. 2nd Ed, PTR Prentice Hall, 1998. ISBN 0-13-490012-X

B.5. Dieper in het besturingssysteem

- Andleigh, Prabhat K. *UNIX System Architecture*. Prentice-Hall, Inc., 1990. ISBN 0-13-949843-5
- Jolitz, William. "Porting UNIX to the 386". *Dr. Dobb's Journal*. January 1991-July 1992.
- Leffler, Samuel J., Marshall Kirk McKusick, Michael J Karels en John Quarterman *The Design and Implementation of the 4.3BSD UNIX Operating System*. Reading, Mass. : Addison-Wesley, 1989. ISBN 0-201-06196-1
- Leffler, Samuel J., Marshall Kirk McKusick, *The Design and Implementation of the 4.3BSD UNIX Operating System: Answer Book*. Reading, Mass. : Addison-Wesley, 1991. ISBN 0-201-54629-9
- McKusick, Marshall Kirk, Keith Bostic, Michael J Karels, and John Quarterman. *The Design and Implementation of the 4.4BSD Operating System*. Reading, Mass. : Addison-Wesley, 1996. ISBN 0-201-54979-4

  Hoofdstuk 2 is online beschikbaar als onderdeel van het FreeBSD Documentatie Project.

- Stevens, W. Richard. *TCP/IP Illustrated, Volume 1: The Protocols*. Reading, Mass. : Addison-Wesley, 1996. ISBN 0-201-63346-9
- Schimmel, Curt. *Unix Systems for Modern Architectures*. Reading, Mass. : Addison-Wesley, 1994. ISBN 0-201-63338-8
- Stevens, W. Richard. *TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP and the UNIX Domain Protocols*. Reading, Mass. : Addison-Wesley, 1996. ISBN 0-201-63495-3
- Vahalia, Uresh. *UNIX Internals — The New Frontiers*. Prentice Hall, 1996. ISBN 0-13-101908-2
- Wright, Gary R. and W. Richard Stevens. *TCP/IP Illustrated, Volume 2: The Implementation*. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-63354-X

B.6. Over beveiliging

- Cheswick, William R. and Steven M. Bellovin. *Firewalls and Internet Security: Repelling the Wily Hacker*. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-63357-4
- Garfinkel, Simson. *PGP Pretty Good Privacy* O'Reilly & Associates, Inc., 1995. ISBN 1-56592-098-8

B.7. Over hardware

- Anderson, Don and Tom Shanley. *Pentium Processor System Architecture*. 2nd Ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-40992-5
- Ferraro, Richard F. *Programmer's Guide to the EGA, VGA, and Super VGA Cards*. 3rd ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-62490-7
- Intel Corporation publiceert documentatie over haar CPU's, chipsets en standaarden op haar ontwikkelaars website, gewoonlijk als PDF bestanden.
- Shanley, Tom. *80486 System Architecture*. 3rd ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-40994-1
- Shanley, Tom. *ISA System Architecture*. 3rd ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-40996-8
- Shanley, Tom. *PCI System Architecture*. 4th ed. Reading, Mass. : Addison-Wesley, 1999. ISBN 0-201-30974-2
- Van Gilluwe, Frank. *The Undocumented PC*, 2nd Ed. Reading, Mass: Addison-Wesley Pub. Co., 1996. ISBN 0-201-47950-8
- Messmer, Hans-Peter. *The Indispensable PC Hardware Book*, 4th Ed. Reading, Mass: Addison-Wesley Pub. Co., 2002. ISBN 0-201-59616-4

B.8. UNIX® geschiedenis

- Lion, John *Lion's Commentary on UNIX, 6th Ed. With Source Code*. ITP Media Group, 1996. ISBN 1573980137
- Raymond, Eric S. *The New Hacker's Dictionary, 3rd edition*. MIT Press, 1996. ISBN 0-262-68092-0. Ook bekend als het Jargon Bestand
- Salus, Peter H. *A quarter century of UNIX*. Addison-Wesley Publishing Company, Inc., 1994. ISBN 0-201-54777-5
- Simon Garfinkel, Daniel Weise, Steven Strassmann. *The UNIX-HATERS Handbook*. IDG Books Worldwide, Inc., 1994. ISBN 1-56884-203-1. Het is niet meer te leveren, maar wel online beschikbaar.
- Don Libes, Sandy Ressler *Life with UNIX* - special edition. Prentice-Hall, Inc., 1989. ISBN 0-13-536657-7
- *The BSD family tree*. http://www.FreeBSD.org/cgi/cvsweb.cgi/src/shared/misc/bsd-family-tree of /usr/shared/misc/bsd-family-tree op een FreeBSD machine.
- *Networked Computer Science Technical Reports Library*. http://www.ncstrl.org/
- *Oude BSD releases van de Computer Systems Research group (CSRG)*. http://www.mckusick.com/csrg/: De set van 4 cd-roms bevat alle versies van BSD van 1BSD to 4.4BSD en 4.4BSD-Lite2 (maar helaas 2.11BSD niet). Op de laatste disk staan ook de laatste broncode en de SCCS bestanden.

B.9. Tijdschriften en periodieken

- *The C/C++ Users Journal*. R&D Publications Inc. ISSN 1075-2838
- *Sys Admin - The Journal for UNIX System Administrators* Miller Freeman, Inc., ISSN 1061-2688
- *freeX - Das Magazin für Linux - BSD - UNIX* (Duits) Computer- und Literaturverlag GmbH, ISSN 1436-7033

Bijlage C: Bronnen op Internet

Door de snelle ontwikkeling van FreeBSD zijn gedrukte media niet zo praktisch om de laatste ontwikkelingen te volgen. Elektronische bronnen zijn de beste, en vaak de enige, om op de hoogte te blijven van de laatste ontwikkelingen. Omdat FreeBSD draait op de inzet van vrijwilligers, is de gebruikersgemeenschap vaak een soort "technische ondersteuningsgroep", die heeft ontdekt dat email, webfora, en USENET de meeste effectieve manieren zijn om de gebruikersgemeenschap te bereiken.

Hieronder staan de meest belangrijke contactmogelijkheden met de FreeBSD gebruikersgemeenschap beschreven. Mochten er andere bronnen zijn die hier niet beschreven zijn, laat die dan weten aan de FreeBSD documentatieproject mailinglijst, zodat ze hier ook beschreven kunnen worden.

C.1. Mailinglijsten

De mailinglijsten zijn de meest directe manier om vragen te stellen aan of een technische discussie te beginnen met een geconcentreerd FreeBSD-publiek. Er is een grote verscheidenheid aan lijsten met betrekking tot verschillende FreeBSD-onderwerpen. Door uw vragen aan de meest geschikte mailinglijst te stellen bent u ongetwijfeld verzekerd van een sneller en accurater antwoord.

De doelstellingen van de verschillende lijsten staan onderaan dit document. *Lees alstublieft de doelstellingen alvorens lid te worden of mail te sturen*. De meeste leden ontvangen tegenwoordig vaak honderden FreeBSD-gerelateerde berichten per dag, en door de doelstellingen en gebruiksregels op te stellen wordt gestreefd om zo min mogelijk ruis op de lijn te krijgen. Door de voorgaande adviezen te negeren zouden de mailinglijsten op termijn falen als een effectief communicatiemedium over het project.



Als u wilt testen of u naar de FreeBSD lijsten email kunt versturen, stuur dan een bericht naar FreeBSD test mailinglijst. Stuur alstublieft geen testberichten naar andere lijsten.

Bij twijfel over naar welke lijst te posten, kan de pagina Hoe de beste resultaten uit de FreeBSD-vragen mailinglijst te halen wellicht helpen.

Alvorens naar enige lijst te posten, is het verstandig te leren hoe de mailinglijsten het beste gebruikt kunnen worden. Hoe bijvoorbeeld zich vaak herhalende discussies voorkomen kunnen worden door het document Veel Gestelde Mailinglijstvragen (FAQ) te lezen.

Voor alle mailinglijsten worden archieven bijgehouden die doorzocht kunnen worden op de FreeBSD World Wide Web server. De met sleutelwoorden te doorzoeken archieven bieden een voortreffelijke methode om antwoorden te vinden op vaak gestelde vragen en horen geraadpleegd te worden voordat er vragen op een lijst worden gesteld. Merk op dat dit ook betekent dat berichten die naar de mailinglijsten van FreeBSD worden verzonden tot in de oneindigheid worden gearchiveerd. Overweeg, wanneer het beschermen van privacy belangrijk is, om een tweede emailadres dat weggegooid kan worden te gebruiken en om alleen publieke informatie te posten.

C.1.1. Lijstsamenvatting

Algemene lijsten: De volgende zijn algemene lijsten waarop vrijelijk (en aangemoedigd) geabonneerd kan worden:

Lijst	Doel
freebsd-advocacy	FreeBSD Evangelisatie
FreeBSD aankondigingen mailinglijst	Belangrijke gebeurtenissen en projectdoelen (gemodereerd)
freebsd-arch	Architectuur en ontwerp discussies
freebsd-bugbusters	Discussie over het onderhoud van de FreeBSD probleemrapportendatabase en aanverwante zaken
freebsd-bugs	Bugbeschrijvingen
freebsd-chat	Niet-technische onderwerpen met betrekking tot de FreeBSD-gemeenschap
freebsd-chromium	FreeBSD specifieke Chromium problemen
FreeBSD-CURRENT mailinglijst	Discussie over het gebruik van FreeBSD-CURRENT
freebsd-isp	Zaken voor Internet Service Providers die FreeBSD gebruiken
freebsd-jobs	Werk en mogelijkheden voor het geven van advies met betrekking tot FreeBSD
freebsd-questions	Gebruikersvragen en technische ondersteuning
FreeBSD beveiligingswaarschuwingen mailinglijst	Beveiligingswaarschuwingen (gemodereerd)
FreeBSD-STABLE; mailinglijst	Discussies over het gebruik van FreeBSD-STABLE
FreeBSD test mailinglijst	Hier kunnen testberichten heengestuurd worden in plaats van naar de eigenlijke lijsten

Technische lijsten: De volgende lijsten zijn voor technische discussie. Het is van belang de doelstellingen te lezen alvorens lid te worden of mail te sturen omdat de richlijnen voor het gebruik en de inhoud erg strikt zijn.

Lijst	Doel
FreeBSD ACPI mailinglijst	Ontwikkeling van ACPI en energiebeheer
freebsd-afs	Porten van AFS naar FreeBSD
freebsd-aic7xxx	Ontwikkeling van stuurprogramma's voor de Adaptec® AIC 7xxx

Lijst	Doel
freebsd-amd64	Porten van FreeBSD naar AMD64-systemen (gemodereerd)
freebsd-apache	Discussie over ports met betrekking tot Apache
freebsd-arm	Porten van FreeBSD naar ARM®-processors
freebsd-atm	Het gebruik van ATM-netwerken met FreeBSD
freebsd-bluetooth	Bluetooth® technologie gebruiken in FreeBSD
freebsd-cluster	FreeBSD gebruiken in een geclusterde omgeving
freebsd-cvsweb	CVSweb onderhoud
freebsd-database	Discussie over het gebruik en de ontwikkeling van databases met FreeBSD
freebsd-desktop	FreeBSD gebruiken op en verbeteren voor bureaubladen
freebsd-doc	Het maken van FreeBSD-gerelateerde documenten
freebsd-drivers	Apparaatstuurprogramma's schrijven voor FreeBSD
freebsd-dtrace	DTrace op FreeBSD gebruiken en ontwikkelen
freebsd-eclipse	Gebruikers van Eclipse IDE, hulpprogramma's, cliëntapplicaties en ports
freebsd-eol	Ondersteuning voor FreeBSD-gerelateerde software welke niet langer ondersteund worden door het FreeBSD-project.
freebsd-embedded	FreeBSD gebruiken in embedded applicaties.
freebsd-emulation	Emulatie van andere systemen zoals Linux®, MS-DOS®, en Windows®
freebsd-firewire	FreeBSD FireWire® (iLink, IEEE 1394) technische discussie
freebsd-fs	Bestandssystemen
freebsd-gecko	Discussies over de Gecko Rendering Engine
freebsd-geom	GEOM-specifieke discussies en implementaties
freebsd-gnome	Porten van GNOME en GNOME applicaties
freebsd-hackers	Algemene technische discussies
freebsd-hardware	Algemene discussies over hardware voor het draaien van FreeBSD
freebsd-i18n	FreeBSD Internationalisatie
freebsd-ia32	FreeBSD op het IA-32 (Intel® x86) platform

Lijst	Doel
freebsd-ia64	Porten van FreeBSD naar Intel®'s IA64 systemen
freebsd-infiniband	Infiniband op FreeBSD
freebsd-ipfw	Technische discussie over het herontwerp van de IP-firewallcode
freebsd-isdn	ISDN-ontwikkelaars
freebsd-jail	Discussies over de jail(8)-faciliteiten.
freebsd-java	Java™ ontwikkelaars en mensen die JDK™s porten naar FreeBSD
freebsd-kde	Porten van KDE en KDE applicaties
freebsd-lfs	Porten van LFS naar FreeBSD
freebsd-mips	Porten van FreeBSD naar MIPS®
freebsd-mobile	Discussie over mobiel computeren
freebsd-mono	Mono en C# applicaties op FreeBSD
freebsd-mozilla	Porten van Mozilla naar FreeBSD
FreeBSD multimedia mailinglijst	Multimedia-applicaties
freebsd-new-bus	Technische discussies over busarchitecturen
freebsd-net	Discussies over netwerken en TCP/IP-broncode
freebsd-numerics	Discussies over implementaties van hoge kwaliteit van functies in libm
freebsd-office	Kantoortoepassingen op FreeBSD
freebsd-performance	Optimalisatie van prestaties voor installaties met hoge prestaties en/of load
freebsd-perl	Onderhoud van een aantal ports met betrekking tot Perl
freebsd-pf	Discussies en vragen voor het pakketfilter firewallsysteem
freebsd-platforms	Ports naar niet Intel®-architectuurplatformen
freebsd-ports	Discussie over de Portscollectie
freebsd-ports-announce	Belangrijk nieuws en belangrijke instructies over de Portscollectie (gemodereerd)
freebsd-ports-bugs	Discussie over bugs in ports en PR’s
freebsd-ppc	Porten van FreeBSD naar de PowerPC®
freebsd-proliant	Technische discussie over FreeBSD op HP Proliant serverplatforms
freebsd-python	FreeBSD-specifieke zaken over Python

Lijst	Doel
freebsd-realtime	Ontwikkeling van realtime-uitbreidingen voor FreeBSD
freebsd-rc	Discussie over het rc.d-systeem en de ontwikkeling daarvan
freebsd-ruby	FreeBSD-specifieke discussies over Ruby
freebsd-scsi	Het SCSI-subsysteem
FreeBSD beveiliging mailinglijst	Beveiligingsonderwerpen betreffende FreeBSD
freebsd-small	FreeBSD gebruiken in embedded toepassingen, verouderd, gebruik in plaats hiervan freebsd-embedded
freebsd-snapshots	Aankondigingen van ontwikkel-snapshots van FreeBSD
freebsd-sparc64	Porten van FreeBSD naar op SPARC® gebaseerde systemen
freebsd-standards	Volgen van de C99- en de POSIX® standaarden door FreeBSD
freebsd-sysinstall	Ontwikkeling van sysinstall(8)
freebsd-tcltk	FreeBSD-specifieke discussies over Tcl/Tk
freebsd-testing	Testen op FreeBSD
freebsd-tex	TeX en haar toepassingen op FreeBSD overzetten
freebsd-threads	Threading in FreeBSD
freebsd-tilera	FreeBSD porteren naar de Tilera CPU-familie
freebsd-tokenring	Ondersteuning voor Token Ring in FreeBSD
freebsd-toolchain	Onderhoud van de ingebouwde toolchain van FreeBSD
freebsd-usb	Discussie over FreeBSD-ondersteuning voor USB
freebsd-virtualization	Discussies over verscheidene virtualisatietechnieken ondersteund door FreeBSD
freebsd-vuxml	Discussie over VuXML-infrastructuur
freebsd-x11	Onderhoud en ondersteuning voor X11 op FreeBSD
freebsd-xen	Discussies over het overbrengen van FreeBSD naar Xen™ - implementatie en gebruik
freebsd-xfce	Overbrengen en onderhouden van XFCE voor FreeBSD

Lijst	Doel
freebsd-zope	Zope voor FreeBSD - overbrengen en onderhouden

Beperkte lijsten: De volgende lijsten zijn voor meer gespecialiseerd publiek en algemene gebruikers hebben er waarschijnlijk niets aan. Het is verstandig om eerst naam te maken in de technische lijsten alvorens lid te worden van een van de onderstaande beperkte lijsten, zodat de gebruiken op die lijst bekend zijn.

Lijst	Doel
freebsd-hubs	Mensen die mirrorsites draaien (infrastructurele ondersteuning)
freebsd-user-groups	Gebruikersgroepcoördinatie
freebsd-wip-status	FreeBSD Werk-In-Uitvoering status
freebsd-wireless	Discussies over de ontwikkeling van de 802.11-stack, gereedschappen en stuurprogramma's.

Verkorte versie van lijsten (digest): Alle hierboven beschreven lijsten zijn beschikbaar in verkorte vorm. Na het lid worden van een lijst zijn de digest opties te wijzigen bij de accountopties.

SVN-lijsten: De volgende lijsten zijn voor mensen met interesse in het zien van logboekberichten voor wijzigingen in verschillende onderdelen van de broncodeboom. Het zijn *Alleen-lezen*-lijsten waar geen email heen gezonden hoort te worden.

Lijst	Broncodegebied	Broncodebeschrijving
svn-doc-all	/usr/doc	Alle wijzigingen aan het doc-Subversion-reservoir (behalve user, projects en translations)
svn-doc-head	/usr/doc	Alle wijzigingen aan de tak "head" van het doc-Subversion-reservoir
svn-doc-projects	/usr/doc/projects	Alle wijzigingen in het projects-gebied van het doc-Subversion-reservoir
svn-doc-svnadmin	/usr/doc	Alle wijzigingen aan de administratieve scripts, haken en andere configuratiegegevens van het doc-Subversion-reservoir
svn-ports-all	/usr/ports	Alle wijzigingen aan het ports-Subversion-reservoir

Lijst	Broncodegebied	Broncodebeschrijving
svn-ports-head	/usr/ports	Alle wijzigingen aan de tak "head" van het ports-Subversion-reservoir
svn-ports-svnadmin	/usr/ports	Alle wijzigingen aan de administratieve scripts, haken en andere configuratiegegevens van het ports-Subversion-reservoir
SVN-commitberichten voor de gehele src-structuur (behalve voor "user" en "projects")	/usr/src	Alle wijzigingen in het src-Subversion-repository (behalve user en projects)
SVN-commitberichten voor de src-structuur voor head/-current	/usr/src	Alle wijzigingen aan de "head"-tak van het src-Subversion-repository (de tak FreeBSD-CURRENT)
svn-src-projects	/usr/projects	Alle wijzigingen aan het gebied projects van het src-Subversion-repository
svn-src-release	/usr/src	Alle veranderingen aan het gebied releases van het src-Subversion-repository
svn-src-releng	/usr/src	Alle veranderingen aan de takken releng van het src-Subversion-repository (de beveiligings- / uitgavetakken)
svn-src-stable	/usr/src	Alle veranderingen aan alle stable-takken van het src-Subversion-repository
svn-src-stable-6	/usr/src	Alle veranderingen aan de stable/6-tak van het src-Subversion-repository
svn-src-stable-7	/usr/src	Alle veranderingen aan de stable/7-tak van het src-Subversion-repository
svn-src-stable-8	/usr/src	Alle veranderingen aan de stable/8-tak van het src-Subversion-repository
SVN-commitberichten voor alleen de 9-stable-src-structuur	/usr/src	Alle veranderingen aan de stable/9-tak van het src-Subversion-repository

Lijst	Broncodegebied	Broncodebeschrijving
svn-src-stable-other	/usr/src	Alle veranderingen aan de oudere stable-takken van het src-Subversion-repository
svn-src-svnadmin	/usr/src	Alle veranderingen aan de administratieve scripts, haken, en andere configuratiegegevens van het src-Subversion-repository
svn-src-user	/usr/src	Alle veranderingen aan het experimentele gebied user van het src-Subversion-repository
svn-src-vendor	/usr/src	Alle wijzigingen aan het verkoperswerkgebied van het src-Subversion-repository

C.1.2. Hoe abonneren

Om te abonneren op een lijst kan geklikt worden op de naam van de lijst hierboven of kan op https://lists.freebsd.org geklikt worden op de lijst waarin interesse bestaat. De pagina waarop de lijsten staan beschreven bevat alle informatie die nodig is om te abonneren.

Om te posten op een lijst kan een email gestuurd worden naar lijstnaam@FreeBSD.org. Daarna wordt die doorgestuurd aan leden van de lijst in de hele wereld.

Om het abonnement op een lijst op te zeggen kan op de URL die onderaan iedere email van een lijst staat geklikt worden. Het is ook mogelijk om een email te sturen naar lijstnaam-unsubscribe@FreeBSD.org om een abonnement op te zeggen.

Hierbij nogmaals het advies om discussies op de technische mailinglijsten technisch te houden. Als er alleen interesse bestaat in belangrijke mededelingen dan wordt aangeraden te abonnneren op FreeBSD aankondigingen mailinglijst, waarop zelden verkeer voorkomt.

C.1.3. Lijstdoelstellingen

Alle FreeBSD-mailinglijsten hebben eigen regels waaraan voldaan dient te worden bij gebruik. Als daaraan niet wordt voldaan, resulteert dat in maximaal twee (2) schriftelijke waarschuwingen van de FreeBSD Postmaster postmaster@FreeBSD.org, waarna na de derde overtreding de poster verwijderd wordt van alle FreeBSD-mailinglijsten en alle toekomstige mail van het adres van de verzender wordt uitgefilterd. Helaas zijn deze regels nodig, omdat het Internet van vandaag de dag een onvriendelijke omgeving is en slechts weinigen zich bewust zijn van hoe fragiel sommige mechanismen zijn.

Standaardregels:

- Het onderwerp van iedere mail dient te voldoen aan de basisdoelstellingen van de lijst waarnaar wordt gepost. Als de lijst bijvoorbeeld over technische onderwerpen gaat, dan hoort

een post ook over iets technisch te gaan. Ruis en flaming doen alleen af aan de waarde van een mailinglijst voor alle leden en dat wordt niet getolereerd. Voor vrije discussie dient de FreeBSD babbel mailinglijst gebruikt te worden die daar speciaal voor is ingesteld.

- Bijdragen horen niet naar meer dan twee mailinglijsten verzonden te worden en alleen dan naar twee als het helder en duidelijk is dat daarvoor de noodzaak bestaat. Voor de meeste lijsten bestaat er al veel overlap in de leden en met uitzondering van de meer esoterische lijsten, zoals bijvoorbeeld "-stable & -scsi", is er eigenlijk slechts zelden aanleiding om naar meer dan een lijst te posten. Als een bericht zo is verzonden dat er meerdere mailinglijsten op de regel `Cc` staan, dan hoort de regel `Cc` weer ingekort te worden in een eventueel antwoord. *De verzender is verantwoordelijk voor zijn eigen kruisposten, wie ook een eerdere zender was.*
- Persoonlijke aanvallen en profane taal (in de context van een geschil) zijn niet toegestaan. Dit geldt zowel voor gebruikers als ontwikkelaars. Grove schending van de netiquette, zoals kopiëren uit of het volledig doorsturen van persoonlijke email zonder dat daarvoor toestemming is gegeven, wordt niet op prijs gesteld. Er zijn hoe dan ook zeer weinig gevallen waarin zoiets dergelijks wel binnen de doelstelling van een lijst valt, waardoor dat soort emails op grond van de inhoud alleen al vaak reden zijn voor een waarschuwing (of ban).
- Adverteren voor niet-FreeBSD-gerelateerde producten is streng verboden en heeft direct een ban tot gevolg als helder is dat de overtreder adverteert door middel van spam.

Individuele lijstdoelstellingen:

FreeBSD ACPI mailinglijst

ACPI en energiebeheerontwikkeling

freebsd-afs

Andrew Bestandssysteem (Andrew File System)

Deze lijst is voor onderwerpen over het porten en gebruik van AFS van CMU/Transarc

FreeBSD aankondigingen mailinglijst

Belangrijke gebeurtenissen en projectdoelen

Dit is de mailinglijst voor hen die alleen interesse hebben in gelegenheidsmededelingen of belangrijke FreeBSD-gebeurtenissen. Hieronder vallen aankondigingen over snapshots en andere uitgaven. De lijst omvat ook aankondigingen over nieuwe mogelijkheden binnen FreeBSD. Er kunnen ook oproepen gedaan worden voor vrijwilligers, enzovoort. Deze lijst kent een laag volume en is volledig gemodereerd.

freebsd-arch

Discussie van architectuur en ontwerp

Deze lijst is bedoeld voor het bespreken van de FreeBSD-architectuur. Berichten zijn in het algemeen strikt technisch van aard. Voorbeelden van geschikte onderwerpen zijn:

- Hoe het buildsysteem bijgewerkt kan worden zodat meerdere aanpaste builds tegelijkertijd kunnen lopen.
- Wat moet er aan VPS aangepast worden om Heidemann-lagen te laten werken.

- Hoe kan de apparataatstuurprogramma interface aangepast worden zodat dezelfde stuurprogramma's netjes op vele bussen en architecturen gebruikt kunnen worden.
- Hoe een netwerkstuurprogramma geschreven kan worden.

freebsd-bluetooth

Bluetooth® in FreeBSD

Dit is het forum waar gebruikers van Bluetooth® op FreeBSD samenkomen. Gespreksstof op het gebied van ontwerp, implementatiedetails, patches, probleemrapportages, statusrapportages, verzoeken voor nieuwe mogelijkheden en al het andere dat met Bluetooth® te maken heeft is geschikt materiaal.

freebsd-bugbusters

Coördinatie afhandeling Problem Reports

Het doel van deze lijst is een platform zijn voor de coördinatie en discussie voor de Bugmeister, zijn Bugbusters en anderen die interesse hebben in de PR-database. Deze lijst is niet bedoeld voor discussies over specifieke bugs, patches of PR's.

freebsd-bugs

Bug reports

Dit is de mailinglijst voor het rapporteren van bugs in FreeBSD. Waar mogelijk dienen bugs ingezonden te worden via send-pr(1) of via de Webinterface daarvan.

freebsd-chat

Niet-technische onderwerpen met betrekking tot de FreeBSD-gemeenschap

Deze lijst bevat alle onderwerpen waar op andere lijsten geen ruimte voor is wat betreft niet-technische en sociale informatie. Er wordt gesproken over de moord op Van Gogh, of er in onderkast of kapitalen geschreven dient te worden, wie er te veel koffie drinkt, waar het beste bier vandaan komt, enzovoort. Belangrijke gebeurtenissen (zoals feestjes, bruiloften, geboorten, nieuwe banen, enzovoort) kunnen op de technische lijsten aangekondigd worden, maar antwoorden dienen naar deze -chat lijst te gaan.

freebsd-chromium

FreeBSD specifieke Chromium problemen

Dit is een lijst voor het bespreken van Chromium ondersteuning voor FreeBSD. Dit is een technische lijst om de ontwikkelingen en installatie van Chromium te bespreken.

Core Team

FreeBSD Kernteam

Dit is een interne mailinglijst die wordt gebruikt door de kernleden. Er kunnen berichten naar gestuurd worden als een belangrijke FreeBSD-gerelateerde zaak arbitrage nodig heeft of een onderzoekende blik op hoog niveau nodig is.

FreeBSD-CURRENT mailinglijst

Discussie over het gebruikt van FreeBSD-CURRENT

Dit is de mailinglijst voor gebruikers van FreeBSD-CURRENT. Er staan waarschuwingen op over nieuwe mogelijkheden in -CURRENT die impact hebben op gebruikers en instructies over de te nemen stappen om -CURRENT te blijven. Iedereen die "CURRENT" draait, zou zich moeten abonneren. Dit is een technische mailinglijst waarop strikt technische berichten worden verwacht.

freebsd-cvsweb

FreeBSD CVSweb Project

Technische discussie over het gebruik, de ontwikkeling en het beheer van FreeBSD-CVSweb.

freebsd-desktop

FreeBSD gebruiken op en verbeteren voor bureaubladen

Dit is een forum voor het bespreken van FreeBSD op desktops. Het is vooral een plaats voor porters en gebruikers van bureaubladomgevingen om zaken te bespreken en de ondersteuning van FreeBSD op het bureaublad te verbeteren.

freebsd-doc

Documentatieproject

Dit is de mailinglijst voor het bespreken van onderwerpen en projecten die te maken hebben met het maken van documentatie voor FreeBSD. De leden van deze mailinglijst worden samen "The FreeBSD Documentation Project" genoemd. Het is een open lijst waarop zonder problemen een abonnement genomen kan worden en bijdragen zeer op prijs worden gesteld!

freebsd-drivers

Apparaatstuurprogramma's schrijven voor FreeBSD

Dit is een forum voor technische discussie met betrekking tot apparaatstuurprogramma's op FreeBSD. Het is vooral een plaats voor schrijvers van apparaatstuurprogramma's om vragen te stellen over hoe apparaatstuurprogramma's te schrijven met de API's in de kernel van FreeBSD.

freebsd-dtrace

DTrace op FreeBSD gebruiken en ontwikkelen

DTrace is een geïntegreerd component van FreeBSD dat een raamwerk biedt om de kernel en de gebruikersprogramma's tijdens het draaien te begrijpen. De mailinglijst is een gearchiveerde discussie voor ontwikkelaars van de code en voor de gebruikers ervan.

freebsd-eclipse

Gebruikers van Eclipse IDE, hulpprogramma's, cliëntapplicaties en ports

De doelstelling van deze lijst is wederzijdse ondersteuning bieden voor alles dat te maken heeft met het kiezen, installeren, gebruiken, ontwikkelen, en onderhouden van Eclipse IDE, hulpprogramma's en cliëntapplicaties op het FreeBSD-platform en te ondersteunen bij het

porten van Eclipse IDE en plugins naar de FreeBSD-omgeving.

Het is ook de bedoeling om het uitwisselen van informatie tussen de Eclipse gemeenschap en de FreeBSD-gemeenschap te bevorderen zodat beiden ervan kunnen profiteren.

Hoewel deze lijst voornamelijk is gericht op de behoeften van gebruikers van Eclipse, wordt ook een forum geboden voor hen die FreeBSD-specifieke applicaties willen ontwikkelen met het Eclipse raamwerk.

freebsd-eol

Ondersteuning van FreeBSD gerelateerde software welke niet meer ondersteund wordt door het FreeBSD-project.

Deze lijst is voor degenen die geïnteresseerd zijn in het leveren of gebruiken van ondersteuning voor FreeBSD-gerelateerde software voor welke het FreeBSD-project geen ondersteuning meer biedt (in de vorm van beveiligingsadviezen en patches).

freebsd-embedded

FreeBSD gebruiken in embedded applicaties

Deze lijst heeft tot doel om te discussieren over FreeBSD in embedded systemen. Dit is een technische mailinglijst waarbij men alleen technische inhoud verwacht. Voor het belang van deze lijst definiëren we embedded systemen als computersystemen die geen desktop-systemen zijn en meestal slechts één doel hebben ten opzichte van gewone systemen. Voorbeelden bevatten onder andere: diverse soorten telefoonsets, netwerkapparatuur zoals routers, switches en PBX'en, op afstand bestuurbare meetapparatuur, PDA's, Point of Sale systemen etc.

freebsd-emulation

Emulatie van andere systemen zoals Linux®, MS-DOS® en Windows®

Dit is een forum voor technische discussie met betrekking tot het draaien van programma's op FreeBSD die zijn geschreven voor andere besturingssystemen.

freebsd-firewire

FireWire® (iLink, IEEE 1394)

Dit is de mailinglijst voor het bespreken van het ontwerp en de implementatie van een FireWire® (ook wel IEEE 1394 of iLink) subsysteem voor FreeBSD. Relevante onderwerpen omvatten de standaarden, busapparaten en hun protocollen, adapter boards/kaarten/chipssets en de architectuur en implementatie van code voor een juiste ondersteuning.

freebsd-fs

Bestandssystemen

Discussie over FreeBSD-bestandssystemen. Dit is een technische mailinglijst waarop slechts strikt technische bijdragen worden verwacht.

freebsd-gecko

Gecko Rendering Engine

Dit is een forum over Gecko applicaties die FreeBSD gebruiken.

De discussie concentreert zich op toepassingen van Gecko Ports, hun installatie, hun ontwikkeling en hun ondersteuning binnen FreeBSD.

freebsd-geom

GEOM

Discussie specifiek over GEOM en gerelateerde implementaties. Dit is een technische mailinglijst waarop slechts strikt technische bijdragen worden verwacht.

freebsd-gnome

GNOME

Discussie over de bureaubladomgeving GNOME voor FreeBSD. Dit is een technische mailinglijst waarop slechts strikt technische bijdragen worden verwacht.

freebsd-infiniband

Infiniband op FreeBSD

Technische mailinglijst over Infiniband, OFED en OpenSM op FreeBSD.

freebsd-ipfw

IP Firewall

Dit is het forum voor technische bespreking van het herontwerp van de IP-firewallcode in FreeBSD. Dit is een technische mailinglijst waarop slechts strikt technische bijdragen worden verwacht.

freebsd-ia64

Porten van FreeBSD naar IA64

Dit is een technische mailinglijst voor individuen die actief werken aan het porten van FreeBSD naar het platform IA-64 van Intel®, om problemen op tafel te leggen of alternatieve oplossingen te bespreken. Geïnteresseerden die alleen de technische bespreking willen volgen zijn ook welkom.

freebsd-isdn

ISDN-communicatie

Dit is de mailinglijst voor discussie over de ontwikkeling van ISDN-ondersteuning voor FreeBSD.

freebsd-java

Java™ Ontwikkeling

Dit is de mailinglijst voor het bespreken van de ontwikkeling van significante Java™ applicaties voor FreeBSD en het porten en het beheer van JDK™'s.

freebsd-jobs

Banen in de aanbieding en gezocht

Dit is een forum voor vacatures en CV's specifiek gerelateerd aan FreeBSD, bijvoorbeeld als er FreeBSD-gerelateerd werk wordt gezocht of in de aanbieding is. Dit is *geen* mailinglijst voor algemene werkonderwerpen omdat daarvoor al elders ruimte staat.

Ook deze lijst wordt net als alle andere `FreeBSD.org` mailinglijsten wereldwijd verspreid. Daarom dient duidelijk vermeld te worden om welke locatie het gaat en onder welke voorwaarden telewerken of bijdragen in huisvesting mogelijk zijn.

Email dient alleen open formaten te bevatten. Bij voorkeur platte tekst, maar standaard Portable Document Format (PDF), HTML, en een aantal andere, zijn acceptabel voor lezers. Gesloten formaten, zoals Microsoft® Word (.doc), worden door de mailinglijstserver geweigerd.

freebsd-kde

KDE

Discussie over KDE op FreeBSD-systemen. Dit is een technische mailinglijst waarop slechts strikt technische bijdragen worden verwacht.

freebsd-hackers

Technische discussies

Dit is een forum voor technische discussie met betrekking tot FreeBSD. Dit is de leidende technische mailinglijst die is bestemd voor mensen die actief aan FreeBSD werken om problemen aan het voetlicht te brengen of alternatieve oplossingen te bespreken. Geïnteresseerden die alleen de technische bespreking willen volgen zijn ook welkom. Dit is een technische mailinglijst waarop slechts strikt technische bijdragen worden verwacht.

freebsd-hardware

Algemene discussie over FreeBSD-hardware

Algemene discussie over de typen hardware waar FreeBSD op draait en problemen en oplossingen over wat te kopen en wat vooral niet.

freebsd-hubs

Mirrorsites

Aankondigingen en discussie voor beheerders van FreeBSD-mirrorsites.

freebsd-isp

Onderwerpen voor Internet Service Providers

Deze mailinglijst is voor het bespreken van relevante onderwerpen voor Internet Service Providers (ISP's) die FreeBSD gebruiken. Dit is een technische mailinglijst waarop slechts strikt technische bijdragen worden verwacht.

freebsd-mono

Mono en C# applicaties op FreeBSD

Dit is een lijst voor discussies met betrekking tot het Mono-ontwikkelraamwerk op FreeBSD. Dit is een technische mailinglijst. Het is bedoeld voor individuen die actief werken aan het

overbrengen van Mono of C# applicaties naar FreeBSD, om problemen naar voren te brengen of alternatieve oplossingen te bespreken. Individuen die geïnteresseerd zijn in het volgen van de technische discussie zijn ook welkom.

freebsd-office

Kantoortoepassingen op FreeBSD

De discussie richt zich op kantoortoepassingen, hun installatie, hun ontwikkeling en hun ondersteuning binnen FreeBSD.

freebsd-ops-announce

Aankondigingen over de projectinfrastructuur

Deze mailinglijst is bedoeld voor mensen die geïnteresseerd zijn in veranderingen en zaken die te maken hebben met de infrastructuur van het FreeBSD.org project.

Deze gemodereerde lijst is strict voor aankondigingen: geen antwoorden, verzoeken, discussies of meningen.

freebsd-performance

Discussie over het optimaliseren of versnellen van FreeBSD

Deze mailinglijst is een platform voor hackers, beheerders en/of andere belanghebbenden om FreeBSD- en prestatiegerelateerde onderwerpen te bespreken. De onderwerpen die besproken kunnen worden omvatten FreeBSD-installaties met een hoge load, systemen met prestatieproblemen of systemen die tegen de limieten van FreeBSD aan zitten. Zij die willen meewerken om de prestaties van FreeBSD te verbeteren worden sterk aangemoedigd zich op deze lijst te abonneren. Deze lijst is bijzonder technisch en bijzonder geschikt voor ervaren FreeBSD-gebruikers, hackers en beheerders die FreeBSD snel, robuust, en schaalbaar willen houden. Deze lijst is geen vraag-en-antwoord lijst die dient als vervanging voor het lezen van documentatie, maar hier worden bijdragen geleverd of vragen gesteld over nog niet eerder beschreven prestatiegerelateerde onderwerpen.

freebsd-pf

Discussie en vragen over het pakketfilter firewallsysteem

Discussie over het pakketfilter (pf) firewallsysteem met betrekking tot FreeBSD. Technische discussie en gebruikersvragen zijn beiden welkom. Deze lijst is ook de plaats om het raamwerk ALTQ QoS te bespreken.

freebsd-pkg

Discussies over binair pakketbeheer en pakketgereedschappen

Discussies over alle aspecten over het beheren van FreeBSD-systemen door middel van het gebruik van binaire pakketten om software te installeren, inclusief de gereedschappen en formaten van binaire pakketten, hun ontwikkeling en ondersteuning binnen FreeBSD, het beheer van pakketreservoirs en pakketten van derde partijen.

Merk op dat discussies over poorten die onjuiste pakketten genereren over het algemeen als

problemen met poorten moet worden gezien en dus ongeschikt zijn voor deze lijst.

freebsd-platforms

Porten van niet-Intel® platforms

Cross-platform FreeBSD-zaken, algemene discussie en voorstellen voor niet-Intel® FreeBSD ports. Dit is een technische mailinglijst waarop slechts strikt technische bijdragen worden verwacht.

freebsd-ports

Discussie over "ports"

Discussie over de "Portscollectie" (/usr/ports) van FreeBSD, de Ports infrastructuur en algemene coördinatie aangaande ports. Dit is een technische mailinglijst waarop slechts strikt technische bijdragen worden verwacht.

freebsd-ports-announce

Belangrijk nieuws en belangrijke instructies over FreeBSD "Portscollectie"

Belangrijk nieuws voor ontwikkelaars, porters en gebruikers van de "Portscollectie" (/usr/ports), waaronder veranderingen aan de architectuur/infrastructuur, nieuwe mogelijkheden, kritische opwaardeerinstructies, en uitgave-informatie. Dit is een mailinglijst met een laag volume, bedoeld voor aankondigingen.

freebsd-ports-bugs

Discussie over "ports" bugs

Discussie over probleemrapportages voor de FreeBSD "Portscollectie" (/usr/ports), voorgestelde ports of aanpassingen aan ports. Dit is een technische mailinglijst waarop slechts strikt technische bijdragen worden verwacht.

freebsd-proliant

Technische discussie over FreeBSD op HP Proliant serverplatforms

Deze mailinglijst wordt gebruikt voor technische discussie over het gebruik van FreeBSD op HP ProLiant servers, inclusief het bespreken van ProLiant-specifieke stuurprogramma's, beheersoftware, gereedschappen voor instellingen en BIOS-updates. Dit is daaarom ook de uitgesproken plaats voor het bespreken van de modules hpasmd, hpasmcli, en hpacucli.

freebsd-python

Python op FreeBSD

Dit is een lijst voor discussie gerelateerd aan het verbeteren van ondersteuning voor Python op FreeBSD. Dit is een technische mailinglijst voor mensen die aan het porten van Python, aanverwante modules en Zope-dingen naar FreeBSD werken.

freebsd-questions

Gebruikersvragen

Dit is de mailinglijst voor vragen over FreeBSD. Er horen geen "how to" vragen op de technische

mailinglijsten thuis, tenzij een vraag erg technisch van aard is.

freebsd-ruby

FreeBSD-specifieke discussies over Ruby

Dit is een lijst voor discussies gerelateerd aan de Ruby-ondersteuning op FreeBSD. Dit is een technische mailinglijst. Het is bedoeld voor individuen die aan Ruby-ports, bibliotheken van derde partijen, en raamwerken werken.

Individuen die geïnteresseerd zijn in de technische discussie zijn ook welkom.

freebsd-scsi

SCSI-subsysteem

Dit is de mailinglijst voor mensen die aan het SCSI-subsysteem voor FreeBSD werken. Dit is een technische mailinglijst waarop slechts strikt technische bijdragen worden verwacht.

FreeBSD beveiliging mailinglijst

Beveiligingsonderwerpen

FreeBSD-computerbeveiligingsonderwerpen (DES, Kerberos, bekende beveiligingsgaten, oplossingen, enzovoort). Dit is een technische mailinglijst waarop slechts strikt technische bijdragen worden verwacht. Dit is zeker geen vraag-en-antwoord lijst, maar bijdragen voor de FAQ (zowel *vraag* als *antwoord*) zijn welkom.

FreeBSD beveiligingswaarschuwingen mailinglijst

Beveiligingswaarschuwingen

Waarschuwingen voor FreeBSD beveiligingsproblemen en oplossingen. Dit is geen discussielijst. De discussielijst is FreeBSD beveiliging mailinglijst.

freebsd-small

FreeBSD gebruiken in embedded toepassingen

Op deze lijst worden onderwerpen gerelateerd aan ongebruikelijk kleine en embedded FreeBSD-installaties besproken. Dit is een technische mailinglijst waarop slechts strikt technische bijdragen worden verwacht.

Deze lijst is vervangen door freebsd-embedded

freebsd-snapshots

Aankondigingen van ontwikkel-snapshots van FreeBSD

Deze lijst houdt u op de hoogte over de beschikbaarheid van nieuwe ontwikkel-snapshots voor de takken head/ en stable/ van FreeBSD.

FreeBSD-STABLE; mailinglijst

Discussie over het gebruik van FreeBSD-STABLE

Dit is de mailinglijst voor gebruikers van FreeBSD-STABLE. Er worden ook waarschuwingen op

gepost over nieuwe opties in -STABLE die invloed op de systemen van gebruikers kunnen hebben en instructies over de te nemen stappen om -STABLE te blijven. Iedereen die "STABLE" draait hoort zich op deze lijst te abonneren. Dit is een technische mailinglijst waarop slechts strikt technische bijdragen worden verwacht.

freebsd-standards

Conformeren C99 & POSIX

Dit is een forum voor technische bespreking gerelateerd aan het conformeren van FreeBSD aan de C99- en de POSIX-standaarden.

freebsd-testing

Testen op FreeBSD

Technische mailinglijst voor discussies over testen op FreeBSD, inclusief ATF/Kyua, infrastructuren voor testbuilds, het testen van ports naar FreeBSD van andere besturingssystemen (NetBSD, ...), enzovoorts.

freebsd-tex

TeX en haar toepassingen op FreeBSD overzetten

Dit is een technische mailinglijst voor discussies over TeX en haar toepassingen op FreeBSD. Het is bedoeld voor degenen die actief werken aan het overzetten van TeX op FreeBSD, om problemen te bespreken of alternatieve oplossingen aan te dragen. Personen die geïnteresseerd zijn in het volgen van de technische discussie zijn ook welkom.

freebsd-toolchain

Onderhoud van de ingebouwde toolchain van FreeBSD

Dit is de mailinglijst bedoeld voor discussies over het onderhoud van de toolchain die met FreeBSD wordt geleverd. Dit zou de toestand van Clang en GCC kunnen omvatten, maar ook software als assemblers, linkers en debuggers.

freebsd-usb

Discussie over FreeBSD ondersteuning voor USB

Dit is de mailinglijst voor technische bespreking van onderwerpen gerelateerd aan FreeBSD ondersteuning voor USB.

freebsd-user-groups

Gebruikersgroep Coördinatie Lijst

Dit is de mailinglijst voor coördinatoren voor alle lokale gebruikersgroepen, zodat ze met elkaar en een lid van het Kernteam zaken kunnen bespreken. Deze lijst hoort beperkt te blijven tot een overzicht van overleggen en de coördinatie van projecten waarbij meerdere gebruikersgroepen betrokken zijn.

freebsd-virtualization

Discussies over verscheidene virtualisatietechnieken ondersteund door FreeBSD

Een lijst om de verscheidene virtualisatietechnieken die door FreeBSD worden ondersteund te bespreken. Aan de ene kant zal de nadruk liggen op de implementatie van de basale functionaliteit alsook op het toevoegen van nieuwe mogelijkheden. Aan de andere kant zullen gebruikers een forum hebben om om hulp te vragen bij problemen of om hun usecases te bespreken.

freebsd-wip-status

FreeBSD Werk-In-Uitvoering status

Deze mailinglijst kan gebruikt worden om de schepping en voortgang van uw FreeBSD-gerelateerd werk aan te kondigen. Berichten zullen gemodereerd worden. Het wordt gesuggereerd om het bericht "Aan:" een FreeBSD-mailinglijst dat het onderwerp beter dekt te sturen en deze lijst alleen te "BCC:"-en. Op deze manier kan uw werk-in-uitvoering ook op de onderwerpslijst worden bediscussieerd, aangezien discussies op deze lijst niet zijn toegestaan.

Kijk in de archieven voor voorbeelden van geschikte berichten.

Een redactioneel overzicht van de berichten aan deze lijst kan om de paar maanden naar de FreeBSD-website gezonden worden als deel van de Status Reports . Meer voorbeelden en oude rapportages zijn daar ook te vinden.

freebsd-wireless

Discussies over de 802.11-stack, de ontwikkeling van gereedschappen voor stuurprogramma's

De FreeBSD-wireless lijst richt zich op de 802.11-stack (sys/net80211) en de ontwikkeling van stuurprogramma's en gereedschappen. Dit omvat bugs, nieuwe eigenschappen en onderhoud.

freebsd-xen

Discussies over het porteren van FreeBSD naar Xen™ - implementatie en gebruik

Een lijst die zich richt op de FreeBSD Xen™ port. De verwachte hoeveelheid verkeer is laag genoeg zodat het voor zowel technische discussies over de implementatie- en ontwerpdetails als voor zaken over administratief gebruik bedoeld is.

freebsd-xfce

XFCE

Dit is een forum voor discussies gerelateerd aan de XFCE-omgeving voor FreeBSD. Dit is een technische mailinglijst. Het is bedoeld voor degenen die actief werken aan het porten van XFCE naar FreeBSD, om problemen naar voren te brengen of alternatieve oplossingen te bespreken. Personen die geïnteresseerd zijn in het volgen van de technische discussie zijn ook welkom.

freebsd-zope

Zope

Dit is een forum voor discussies die verwant zijn aan het brengen van de Zope-omgeving naar FreeBSD. Dit is een technische mailinglijst. Het is bedoeld voor individuen die actief werken aan het overbrengen van Zope naar FreeBSD, om problemen aan te dragen of alternatieve oplossingen te bespreken. Individuen die geïnteresseerd zijn in het volgen van de technische

discussie zijn ook welkom.

C.1.4. Filters op de mailinglijsten

De FreeBSD-mailinglijsten worden op verschillende manieren gefilterd om het doorsturen van spam, virussen, en andere ongewenste email te beperken. De hieronder beschreven filteracties bevatten niet alle genomen acties voor de beveiliging van de mailinglijsten.

Er is een beperkt aantal typen bijlagen toegestaan op de mailinglijsten. Alle bijlagen met een MIME-inhoudtype dat niet in de onderstaande lijst staat worden verwijderd voordat de mail wordt doorgestuurd naar de mailinglijsten.

- application/octet-stream
- application/pdf
- application/pgp-signature
- application/x-pkcs7-signature
- message/rfc822
- multipart/alternative
- multipart/related
- multipart/signed
- text/html
- text/plain
- text/x-diff
- text/x-patch



Sommige mailinglijsten staan wellicht bijlagen toe met andere MIME-inhoudtypen, maar de bovenstaande lijst zal gelden voor de meeste mailinglijsten.

Als een email zowel een HTML- als een platte tekstversie bevat, dan wordt de HTML-versie verwijderd. Als een mail alleen een HTML-versie bevat, dan wordt die omgezet naar platte tekst.

C.2. Usenet-nieuwsgroepen

Naast de twee specifieke FreeBSD-nieuwsgroepen zijn er nog vele andere waarin FreeBSD wordt besproken of die anderszins relevant zijn voor gebruikers van FreeBSD.

C.2.1. Specifieke BSD nieuwsgroepen

- comp.unix.bsd.freebsd.announce
- comp.unix.bsd.freebsd.misc
- de.comp.os.unix.bsd (Duits)
- fr.comp.os.bsd (Frans)

- it.comp.os.freebsd (Italiaans)

C.2.2. Overige interessante UNIX®-nieuwsgroepen

- comp.unix
- comp.unix.questions
- comp.unix.admin
- comp.unix.programmer
- comp.unix.shell
- comp.unix.user-friendly
- comp.security.unix
- comp.sources.unix
- comp.unix.advocacy
- comp.unix.misc
- comp.unix.bsd

C.2.3. X Window systeem

- comp.windows.x.i386unix
- comp.windows.x
- comp.windows.x.apps
- comp.windows.x.announce
- comp.windows.x.intrinsics
- comp.windows.x.motif
- comp.windows.x.pex
- comp.emulators.ms-windows.wine

C.3. World wide webservers

C.3.1. Fora, blogs, en sociale netwerken

- The FreeBSD Forums bieden een webgebaseerd discussieforum voor vragen en technische discussies over FreeBSD.
- Planet FreeBSD biedt een samengestelde feed van tientallen blogs geschreven door FreeBSD-ontwikkelaars. Vele ontwikkelaars gebruiken dit om korte notities te posten over waaraan ze werken, nieuwe patches, en ander werk in uitvoering.
- Het BSDConferences YouTube Channel biedt een verzameling van video's van hoge kwaliteit van wereldwijde BSD-conferenties. Dit is een prima manier om presentaties van hoofdontwikkelaars over nieuw werk in FreeBSD te zien.

C.3.2. Officiële spiegels

Central Servers, Armenia, Australia, {mirrors-austria}, Czech Republic, Denmark, Finland, France, Germany, Hong Kong, Ireland, Japan, Latvia, Lithuania, Netherlands, Norway, Russia, Slovenia, South Africa, Spain, Sweden, Switzerland, Taiwan, United Kingdom, United States of America.

(as of UTC)

Central Servers

- https://www.FreeBSD.org/

Armenia

- http://www1.am.FreeBSD.org/ (IPv6)

Australia

- http://www.au.FreeBSD.org/
- http://www2.au.FreeBSD.org/

{mirrors-austria}

- http://www1.am.FreeBSD.org/ (IPv6)

Czech Republic

- http://www.cz.FreeBSD.org/ (IPv6)

Denmark

- http://www.dk.FreeBSD.org/ (IPv6)

Finland

- http://www.fi.FreeBSD.org/

France

- http://www1.fr.FreeBSD.org/

Germany

- http://www.de.FreeBSD.org/

Hong Kong

- http://www.hk.FreeBSD.org/

Ireland

- http://www.ie.FreeBSD.org/

Japan

- http://www.jp.FreeBSD.org/www.FreeBSD.org/ (IPv6)

Latvia

- http://www.lv.FreeBSD.org/

Lithuania

- http://www.lt.FreeBSD.org/

Netherlands

- http://www.nl.FreeBSD.org/

Norway

- http://www.no.FreeBSD.org/

Russia

- http://www.ru.FreeBSD.org/ (IPv6)

Slovenia

- http://www.si.FreeBSD.org/

South Africa

- http://www.za.FreeBSD.org/

Spain

- http://www.es.FreeBSD.org/
- http://www2.es.FreeBSD.org/

Sweden

- http://www.se.FreeBSD.org/

Switzerland

- http://www.ch.FreeBSD.org/ (IPv6)
- http://www2.ch.FreeBSD.org/ (IPv6)

Taiwan

- http://www.tw.FreeBSD.org/
- http://www2.tw.FreeBSD.org/
- http://www4.tw.FreeBSD.org/

- http://www5.tw.FreeBSD.org/ (IPv6)

United Kingdom

- http://www1.uk.FreeBSD.org
- http://www3.uk.FreeBSD.org/

United States of America

- http://www5.us.FreeBSD.org/ (IPv6)

C.4. Email-adressen

De onderstaande gebruikersgroepen bieden FreeBSD-gerelateerde email-adressen aan voor hun leden. De aangegeven beheerders behouden zich het recht voor om een account te verwijderen als die op enigerlei wijze wordt misbruikt.

Domein	Faciliteiten	Gebruikersgroep	Beheerder
ukug.uk.FreeBSD.org	Alleen forwarden	ukfreebsd@uk.FreeBSD.org	Lee Johnston lee@uk.FreeBSD.org

Bijlage D: PGP sleutels

In het geval een handtekening van een van de beambten of ontwikkelaars gecontroleerd moet worden of er een versleutelde e-mail aan ze gezonden moet worden, worden hier voor het gemak een aantal sleutels weergegeven. Een complete sleutelring van FreeBSD.org gebruikers kan op de volgende link gedownload worden: pgpkeys.txt.

D.1. Beambten

D.1.1. Beveiligingsbeambtenteam <security-officer@FreeBSD.org>

```
pub   rsa4096/D9AD2A18057474CB 2022-12-11 [C] [expires: 2030-01-24]
      Key fingerprint = 0BE3 3275 D74C 953C 79F8  1107 D9AD 2A18 0574 74CB
uid                           FreeBSD Security Officer <security-officer@freebsd.org>
sub   rsa4096/6E58DE901F001AEF 2022-12-11 [S] [expires: 2028-01-24]
sub   rsa4096/46DB26D62F6039B7 2022-12-11 [E] [expires: 2028-01-24]
```

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGOVdeUBEADHF5VGg1iPbACB+7lomX6aDytUf0k2k2Yc/Kp6lfYv7JKU+1nr
TcNF7Gt1YkajPSeWRKNZw/X94g4w5TEOHbJ6QQWx9g+N7RjEq75actQ/r2N5zY4S
ujfFTepbvgR55mLTxlxGKFBmNrfNbpHRyh4GwFRgPlxf5Jy9SB+0m54yFS4QlSd0
pIzO0CLkjHUFy/8S93oSK2zUkgok5gLWruBXom+8VC3OtBElkWswPkE1pKZvMQCv
VyM+7BS+MCFXSdZczDZZoEzpQJGhUYFsdg0KqlLv6z1rP+HsgUYKTkRpcrumDQV0
MMuCE4ECU6nFDDTnbR8Wn3LF5oTt0GtwS0nWf+nZ1SFTDURcSPR4Lp/PKjuDAkOS
P8BaruCNx1ItHSwcnXw0gS4+h8FjtWNZpsawtzjjgApcl+m9KP6dkBcbN+i1DHm6
NG6YQVtVWyN8aOKmoC/FEm1CWh1bv+ri9XOkF2EqT/ktbjbT1hFoFGBkS9/35y1G
3KKyWtwKcyF4OXcArl6sQwGgiYnZEG3sUMaGrwQovRtMf7le3cAYsMkXyiAnEufa
deuabYLD8qp9L/eNo+9aZmhJqQg4EQb+ePH7bGPNDZ+M5oGUwReX857FoWaPhs4L
dAKQ1YwASxdKKh8wnaamjIeZSGP5TCjurH7pADAIaB3/D+ZNl2a7od+C1wARAQAB
tDdGcmVlQlNEIFNlY3VyaXR5IE9mZmljZXIgPHNlY3VyaXR5LW9mZmljZXJAZnJl
ZWJzZC5vcmc+iQJSBBMBCgA8AhsBBAsJCAcEFQoJCAUWAgMBAAIeBQIXgBYhBAvj
MnXXTJU8efgRB9mtKhgFdHTLBQJjlXeQBQkF3u+rAAoJENmtKhgFdHTLOVoQALS3
cj7rqYkHiV4zDYrgPEp9O1kAyGI8VdfGAMkDVTqr+wP4v/o7LIUrgwZl5qxesVFB
VknFr0Wp5g9h0iAjasoI5sDd6tH2SmumhBHXFVdftzDQhrugxH6fWRhHs0SaFYCk
Qt5nFbcpUfWgtQ35XTbsL8iENdYpjKXsSFQrJneGSwxIjWYTFn6ps/AI3gwR8+Bn
OffEFdYugJ049O6Vu6YBFJHrnMO7NbF4v95dVYuLtpMIaXWM+V9KITmhaBzFz5fM
Q7UOzcLlbxOYKNIWcp8QQk429mayKW5VUeUExUD1ZzBHn+P6ZG7QTMDu/RmBqiHo
ewCMVz4n9uXT5BiOngE4CvS0WQwHzK+k9MLpG2u/Bo9+LT0Ceh9Ou1rfU5+0tRwl
GyOFFjf3INS7I7gkcAwxQ7dzDItN/UQPZpg8y9mABU2x4enz0AvTnb61d/1dnTEr
tdNgU433he0ZnD1HurZCjBEWC656wv6iMdWcD8gjhMbmEpPmjvXcYlTO6zhEygSM
DiwdQCWK2W4++YJerA6ULBi3niNWBpofOFH8XylV56ruhjtHCo7+/3carcMoPOJv
lVZ1zCKxLro3TRBT15JTFBGqblRyTopFK3PuxW//GTnZOtpQEOV6yL4RAXcWeC1d
1hb5k/YxUmRF6XsDNEH4b08T8ZO8dV3dAV43Wh1oiQEzBBABCAAdFiEEuyjUCzYO
7pNq7RVv5fe8y6O93fgFAmObXVYACgkQ5fe8y6O93fiBlwf/W8y1XXJIx1ZA3n6u
f7aS70rbP9KFPr4U0dixwKE/gbtIQ9ckeNXrDDWz0v0NCz4qS+33IPiJg1WcY3vR
W90e7QgAueCo5TdZPImPbCs42vadpa5byMXS4Pw+xyT+d/yp2oLKYbj3En4bg1GM
```

```
w71DezIjvV+e01UR++u1t9yZ8LOWM5Kumz1zyQLZDZ8qIKt1bBfpa+E0cEqtNQWu
iGhQE3AHI8eWV+jBkg5y2zHRIevbWb1UPsj43lgkFtAGHk9rrM8Rmgr4AXr531iD
srBwauKZ/MElcF3MINuLH+gkPPaFHw/YIpLRLaZXZVsw3Xi1RNXI2n2ea29dvs/C
Lcf1vYkCMwQQAQgAHRYhBPwOh4rlr+eIAo1jVdOXkvSep+XCBQJjm14FAAoJENOX
kvSep+XC0DcP/1ZB7k9p1T+9QbbZZE1PJiHby3815ccH3XKexbNmmakHIn3L6Cet
F891Kqt9ssbhFRMNtyZ/k/8y8Hv5bKxVep5/HMyK+8aqfDFN0WMrqZh0/CiR6DJh
gnAmPNw/hAVHMHaYGII9kCrFfPFJ02FKoc81g9F08odb7TV+UlvRjkErhRxF+dGS
wQoO0RCbf0Z1cs7nd0Vb2z4IJh4XMxBjWc/uQ2Q9dH/0uRzwpAnR4YX+MG5YrX7Z
zBvDyR0r76iQwRSDKgioNgkr6R3rq1NZGdaj+8b0LzdOqtzKJ/eupDe3+H67e/EN
qymtreGjrubpiU9bKvYArisUqhE5KtguryvR6Qz9bj87nPg33DT3WWGVrwFRxBox
dbWzjQFv0wug8m4GAwVF7fPR5/eW7IHw8zvgn0vSPcZz7MZ4e6Y5jN4kA5/xWJYZ
Sps54qQWB+FA30unIXN68KqdIzONIbtaY3W4/JjJUCm4T+wEjKaH+wJX8w1DMjlg
mkTmGh/UrTyC1vXbPgk9Sy3cRTICR1T9z7W8UlmTtnKrUklrjlFR7SXzrEXzLGOX
Fm+NEHpHNXqzcm6c3QfzY/yQ9HSAQ/t7SUQ9caRePbDz3/msyPxtGFor9roQv6VN
wRXCyRgkH4Y5tPhJAQ8G/FxX+VXFb93QL0lfelb23/BBu6cUwW63SRn5iHUEExYI
AB0WIQQeB2Johg/5/ikUnJwDU9SVF1S13AUCZISO3wAKCRADU9SVF1S13NnqAP95
LA1Om9XSAKl76VtV+L3JPDdAwIdbNa0OsRT4Wm7U3wD/YoFrdHXVHHQFKwYeUUhj
XZcxnZLe9Ixo0/JP+RVFVw6JAlIEEwEKADwCGwEECwkIBwQVCgkIBRYCAwEAAh4F
AheAFiEEC+MydddMlTx5+BEH2a0qGAV0dMsFAmluhFoFCQ1lXf0ACgkQ2a0qGAV0
dMv8ixAAhYn3uxnjTCBZ5BJ6FMdwFXgGZMZVnL1qEoHA7HWB0ZoQlu29sFzBX97x
rkVZfWiU82V9/PYyHh4EY0hN9oOJMDDhSgi+DfdoEablIcIXR9uBBL/Snnak6uEF
gOD6VFH1T3o7iBmS/ixee2YFXq5QxAwEiANSAsxDm2rlyztPc5xM76fz5eEcgp1m
UDI7qFjoEkJVvRraJGVy3AUh4sr/94VAXPuCa7JnC6a8j9SD8XtzqIfcuqNBXXfA
m4ji4fN+KcDHtZPdkNLDBu1DEXQkOR/1S++NmXHYCEdQEuVOwLVsy6vQqgc+unju
Xh9sOrdiXUWylHh2hvgmwkjhMGOREiIRwNoA09m0OKZ2jo2dSlhNdVFRD5yhD92t
X9BarQbbjQtyCvoWnCJMFfw/vDnxfDdUdV3GxLhaOWvR8ItWQ7ts8tEZSjR/G76/
fHj1JRNuHRdhwype6ehieewJSMyzByhT0A3Njv3KLI4nwQj//MYzB5FJ9uOmKLcG
w7PB0yVjW4oRbG62+Qv6IQybCyrAL+12qgbUXbMsTqMwmsmdgtsqr/7OJNo3PBxX
2rUEio/mez8wuzNB0mJtlMynoMqXtbFX7AAXHk16fOGd6L6guW8+IhPedT3TvrjS
hqiQVvjOhM6rEYZ/YUfcIdkWJVpmE6+MRPVUufyMWeHaWqsKXoa5Ag0EY5V2yQEQ
AOqjzPpMUCGu8eElXnAd2PruC6hi+lc/yC90KqizxIuW6qLQBaAkTCWq7suYpDqo
ygn7YM3rL50S285WAECAXrcst/cVAqr0UH/e6p4iJCUIiXcfjd/wq20RnN/+VuvL
hjpCFLY5czfVS31D7Uh9MbC+zUTz8nVTiNCsAao0qSdfJDIzB4nSO+9xIsme/dLs
I5QlU5PdxOBV6HdEhCUXOoratJCbKAOlLxtPwyMKxmv4oZ7Mqlt10peKjhpBb97q
cIzJhHxujQZD0OmzIA6xoQ2eSCGdxCEDsZ09kr3Esw1AwKnQ51xmWpFWNFk6627M
1bo8+hzOz81CrTZhYrgE+1JXv6V1L2A9lMsimdE1BHNycDS+dBOpIB9qxXCwAab4
ykfvNxox/ZPDUrTy7v7mDI5uDNTNCYYsKCj1UidycOKzSziB9Oa2uvmMJ5XstgNB
f7Z8Cky1dtVd4ol6bU9L5nos9tbYeSXFi4bmcWB7AJiVCMq6N+LBbUKWGLglB4TU
1qhttpqv31X9V6ges5gARY/RuRTKsVyhwsn7SDcqmNKRy0im2AYakwEp7hT07ula
hOSLxjP+5hCf+nSJlwbxJ8ozwjjbzeN2yLlJSI00klkIFBNUDt3wzFRW/n6qlf+/
lepgzekfNrYMtfPB8AT07Z2A3U4xlgiV346dZymbY/EjABEBAAGJBHIEGAEKACYW
IQQL4zJ110yVPHn4EQfZrSoYBXR0ywUCY5V2yQIbAgUJAgIpAAJACRDZrSoYBXR0
y8F0IAQZAQoAHRYhBLYVJ36BCH33XIGDO25Y3pAfABrvBQJjlXbJAAoJEG5Y3pAf
ABrvuBsQAOlQFPXhx6whO4yw5ZizIS02YHhSVMVYKS2T9jPIKi1qxnEiEw9eKH0b
W0Oj0TEhZPyM2NJID7DRWK5r8+KsMu8jwm1fUmIrefAx6fCVfCWRECT1MlbL3jhh
6AcX/nK2e3Bn8vgExhzczO3JlvD6wPCc0FkpiY7yDB9ihu1+gbE5Hg6dvftttRXD
rbEdAifbNp9KYxDigxdlOb0S14hjCBysLWH5Su/khcIlkeuqZcI8TmDldnUb2OqT
CVpFhaNwsPSrHBzmb0s2sXo4FL03pLsOdwhi31W6kjk4KvW5FKrOpoEwUMKVNMf5
0DHdvonUoUHRSIc/cV5NqUWHwvc0T5031qk0CCRRa/+/iij/p2RG7c1mx7ZECj+j
ZfmvjSqT+WHJ1BFlNJMWyK4fdVRZWyCaoAecdbukwzDwUCUqHJFIWeFtbut7SOPx
cwg7sbnKNApAKdi491dvH75s/U/OwRYO/2P+ymHlqtyix2jq0ReSVYcQPXswQ8i2
```

```
ifX41F+xTSl4RWCBBeXB1Nxk3+HsV4Jnnp1zAJZ0KlKW/oJxbNFdI1TImkpr2p8i
oFf+aiePLvDkgeaG8vABgjoihPXWHVAMR8Z+GvBY/A6OdexpibkTvC/zDr0/Exs4
lsylZKDwvvFbctcpHVXBeCBQLX5vfLrsTkaCLWF/SV9OdMykvYKU7ZAP+gKEwhp+
HPFuOHZbOBhqFUdkfeCkdzX/QGdzTuz349roRhgz2vRfN7MtbtuzA6NWWhEWt5Dc
UgX/Y5I3Q4Z2bt3JiXQ6WJMgMMOXAr+XxtxyRRykc1HV3DQ/cq8OWYubNnIbgebP
NIFr2OIWKsR9yDaucZzpmLfzaMZUAu5hWmU9fIw5SIKGnQABBnNMhilfD+CkETp6
baTvjTK4rpaobjJdeCTrsWgfXRNC8x3hDvcrjPD70MylOGVQdx8GYChWJnCKXsLT
GX7KwdfxkjclTyzWvdcCemp0eLhamLGb9y1dtWdNIDcVCvZJy0lipHVUdFYYxb4i
LZJANL631tlPM6AA8sO1/L4mqEGnAIHVrUQd+2QkSiOl9mKlpgaR/fJz683BR5Qe
n9ywX0JPtBupqPW3t9VbO/uNxUqlHCeAhPi9NLOpujpYLfgW5QAfS3u0nkp5nrbk
CoQUua2q0Oj7J0mFmtWtcE1c9+THmFJVb8j2G9yQw3ADe3Qp9ALazP5nVDVri8NZ
BhHK1/KuBmRYZtcyfqXUnKoiiWAlm5rHaRiztW7e3wqm2oJu/RkEAagybutEuBWh
2Ej2+gDxjEKKtIKGu54lif4kqTwwjKTcN1ekGihwwgCMUkBSBeNXk1ClkzLFHwES
JCcFwdEgpVYQTKFsu0emYISyco3IpUajGzfUiQRyBBgBCgAmAhsCFiEEC+MydddM
lTx5+BEH2a0qGAV0dMsFAmWFy28FCQPyfaYCQMF0IAQZAQoAHRYhBLYVJ36BCH33
XIGDO25Y3pAfABrvBQJjlXbJAAoJEG5Y3pAfABrvuBsQAOlQFPXhx6whO4yw5Ziz
IS02YHhSVMVYKS2T9jPIKi1qxnEiEw9eKH0bW0Oj0TEhZPyM2NJID7DRWK5r8+Ks
Mu8jwm1fUmIrefAx6fCVfCWRECT1MlbL3jhh6AcX/nK2e3Bn8vgExhzczO3JlvD6
wPCc0FkpiY7yDB9ihu1+gbE5Hg6dvftttRXDrbEdAifbNp9KYxDigxdlOb0S14hj
CBysLWH5Su/khcIlkeuqZcI8TmDldnUb2OqTCVpFhaNwsPSrHBzmb0s2sXo4FL03
pLsOdwhi31W6kjk4KvW5FKrOpoEwUMKVNMf50DHdvonUoUHRSIc/cV5NqUWHwvc0
T5031qk0CCRRa/+/iij/p2RG7c1mx7ZECj+jZfmvjSqT+WHJ1BFlNJMWyK4fdVRZ
WyCaoAecdbukwzDwUCUqHJFIWeFtbut7SOPxcwg7sbnKNApAKdi491dvH75s/U/O
wRYO/2P+ymHlqtyix2jq0ReSVYcQPXswQ8i2ifX41F+xTSl4RWCBBeXB1Nxk3+Hs
V4Jnnp1zAJZ0KlKW/oJxbNFdI1TImkpr2p8ioFf+aiePLvDkgeaG8vABgjoihPXW
HVAMR8Z+GvBY/A6OdexpibkTvC/zDr0/Exs4lsylZKDwvvFbctcpHVXBeCBQLX5v
fLrsTkaCLWF/SV9OdMykvYKUCRDZrSoYBXR0yyOqEACitDvbkbfjaton6izr4T8Q
U2yvhJHkf4B6KeVDbKY1J4784OxXp2bJgPeF53SYBe8gm3YHjp8ULh4A/19U4hsw
yE8ymcm5nIs8OLyBdxkuBZJGEnzxH3woiyYqWH7991kzhEjUkuMgKLuTI1HiO0oL
MuPQNhUHOnWafSVPC0XO/tIL12OmoUuc7ligY9Z9AceFjTZOuHamixHAAc6hpxdI
W+yhC/qTpc2VK0niWeuQfq3453iRTf9MnR5Beztl3ZYRWcx7UiFuKGwZwBibNnNm
Us6GyQcJ5UTa1oeJcLqHi0Lf/r0jXo3wgJq7EZjjVyU+GI2ZVoDOaS6c4/OvLm62
XoeSlnn/dQxUcjUki+x8lb69IxSF1xAgsC/oNtFZYd5rHdlnqIBUYK0lLtSCXBkz
VeivSiQa0hL5on8LDu1nw2bXyW61yt/YxVb4FanMxAqdYVBhOfU0RaPNifH01rbb
4TwC9bTZN1LQ1KI/Swb/SruUE0RyT28fhYRtsReS2PnUODghJSFDJbwFbBZf6RKI
16q1xqKRRvxIWPm+lMOi1NLOKR9P+OKy9HmChMw0UJUcVl1cJ2xtRl3wi5t6AA6H
oNv/TrLeYVgMR9wYmKlpvjTQ5jTdrbHD1XP5jGsp8QsJMGja1m/7cryReCpcVxvI
meReaOdgz+zDmQqq3O5zuIkEcgQYAQoAJgIbAhYhBAvjMnXXTJU8efgRB9mtKhgF
dHTLBQJnhEEJBQkF0/JAAkDBdCAEGQEKAB0WIQS2FSd+gQh991yBgztuWN6QHwAa
7wUCY5V2yQAKCRBuWN6QHwAa77gbEADpUBT14cesITuMsOWYsyEtNmB4UlTFWCkt
k/YzyCotasZxIhMPXih9G1tDo9ExIWT8jNjSSA+w0Viua/PirDLvI8JtX1JiK3nw
MenwlXwlkRAk9TJWy944YegHF/5ytntwZ/L4BMYc3MztyZbw+sDwnNBZKYmO8gwf
YobtfoGxOR4Onb37bbUVw62xHQIn2zafSmMQ4oMXZTm9EteIYwgcrC1h+Urv5IXC
JZHrqmXCPE5g5XZ1G9jqkwlaRYWjcLD0qxwc5m9LNrF6OBS9N6S7DncIYt9VupI5
OCr1uRSqzqaBMFDClTTH+dAx3b6J1KFB0UiHP3FeTalFh8L3NE+dN9apNAgkUWv/
v4oo/6dkRu3NZse2RAo/o2X5r40qk/lhydQRZTSTFsiuH3VUWVsgmqAHnHW7pMMw
8FAlKhyRSFnhbW7re0jj8XMIO7G5yjQKQCnYuPdXbx++bP1PzsEWDv9j/sph5arc
osdo6tEXklWHED17MEPIton1+NRfsU0peEVggQXlwdTcZN/h7FeCZ56dcwCWdCpS
lv6CcWzRXSNUyJpKa9qfIqBX/monjy7w5IHmhvLwAYI6IoT11h1QDEfGfhrwWPwO
jnXsaYm5E7wv8w69PxMbOJbMpWSg8L7xW3LXKR1VwXggUC1+b3y67E5Ggi1hf0lf
TnTMpL2ClAkQ2a0qGAV0dMsNxRAAsuW1aLh+hgydW+iH6DmdQRMESsB1kE02kO14
```

```
62TAQaziIAvNoxw5h48xvyEnrDA8d+9IDMyxdrLmAbndUlSveMa9+EPiGHwr6VTy
FL8nA5F7DcFi4mjEyGKe18JcaAlYUtvHgWH6EjiX2iSXpsrJFEhtfFNolZ5sp9LF
I6hOBihSJxZK4sbMR7Q6IkDuAVpTFLiejBRlsXpFvTGL6040CtxbL5cqkVMYP38r
FMTuc3pGGJA4wb5EC1dGjUi6XjbYH7kuCAFyXqV9eQQP61x7K9W8qnXW+weCIMKf
SX7AcCtH1jXBAM6lqpPrh6amc+/rbg2eNA7DmgJnEY4apIcDB/b4khRMga2ozeGW
WyIvOaVvR2R7ALQ+Rgut85cM+4+Vl2PHmOzW/yYdHVb5REQItFR5COb/mGUqYhkC
tiV3nXo/K0uOQKu5SBbNzLuNvuwdn+Eimxjl18VnrGG7sjtUa0MLmtr62GiEVrhr
Dqa/biHp8LdWkAQjLZ4aTRh2XZiggaVFZHmkw3ILPyKKM21UXdM0YRk3TGVK8ODQ
y58ebPS4v9yYT9gUA9UDkDYeGcF2qjoDPVNvcG6H8jCSsPRl1KZwtqqITCOSAIAP
I4Nu97kO6nbOyQpYlwjd1MhvVXnP66mHSvmqaxbNGX1mF9B/yERkBkooNZrKuJSv
BTC2J1qJBHIEGAEKACYCGwIWIQQL4zJ110yVPHn4EQfZrSoYBXR0ywUCaW6EcgUJ
CaGkeQJAwXQgBBkBCgAdFiEEthUnfoEIffdcgYM7bljekB8AGu8FAmOVdskACgkQ
bljekB8AGu+4GxAA6VAU9eHHrCE7jLDlmLMhLTZgeFJUxVgpLZP2M8gqLWrGcSIT
D14ofRtbQ6PRMSFk/IzY0kgPsNFYrmvz4qwy7yPCbV9SYit58DHp8JV8JZEQJPUy
VsveOGHoBxf+crZ7cGfy+ATGHNzM7cmW8PrA8JzQWSmJjvIMH2KG7X6BsTkeDp29
+221FcOtsR0CJ9s2n0pjEOKDF2U5vRLXiGMIHKwtYflK7+SFwiWR66plwjxOYOV2
dRvY6pMJWkWFo3Cw9KscHOZvSzaxejgUvTekuw53CGLfVbqSOTgq9bkUqs6mgTBQ
wpU0x/nQMd2+idShQdFIhz9xXk2pRYfC9zRPnTfWqTQIJFFr/7+KKP+nZEbtzWbH
tkQKP6Nl+a+NKpP5YcnUEWU0kxbIrh91VFlbIJqgB5x1u6TDMPBQJSockUhZ4W1u
63tI4/FzCDuxuco0CkAp2Lj3V28fvmz9T87BFg7/Y/7KYeWq3KLHaOrRF5JVhxA9
ezBDyLaJ9fjUX7FNKXhFYIEF5cHU3GTf4exXgmeenXMAlnQqUpb+gnFs0V0jVMia
SmvanyKgV/5qJ48u8OSB5oby8AGCOiKE9dYdUAxHxn4a8Fj8Do517GmJuRO8L/MO
vT8TGziWzKVkoPC+8Vty1ykdVcF4IFAtfm98uuxORoItYX9JX050zKS9gpQJENmt
KhgFdHTLozcQAJCdYMPo1SPgxxRI40BCD4ekuA6ZMzdTuuso9XTTLf3moh+eu0Y+
Pv6ISOFI93g09V34H3XnuaOnA16DyApO6mCPZ58DunjRRYHv4wxbeff6vw9twPQb
bcdNgI2ZXSKjUef4FygkpUhy6075m6zoUfcsR/22Ggzd8Qw3chfoqiBZs7SlGwEH
sxkB+dJvgCMwYaLGqS9lNbHSEFnETb0ihFSHKDrK6pYrFnfsuArJN5owuTN+dzim
8g0agB3Pzs4A6+9Hpt9sN78VEY+ZzFhQD8knEgqSNbFIaNxsG5/L8/RcXuNyPvvz
5MIkhfacOX37YFfAhCORfN3PaVWYQ7oDnv8Nh69i9SkxXKUcwX0/iYaim8pwwM6a
HXhTlqBgKR5b5Hcx6Cwx1GTRUpdbJYThmz7LQgzjBIhmoGbxR3vyJjSyPGBZa3Ih
jAETYzMt7j0qGPiJUx7kcHeK7apAH1xB4DnfuPLSC1GjWptb31A22wjWy/75SJk8
0FoyYKPj5RqiQF+5CcSLRwTJEEi+9ggM8mUJiFNUOtfZgATgmCyEKG4fL1N4iyrj
fBha11/EO3LcH4t0rj5i8SkT+w8A0bgpOliLhnKpFQMy+cRewkzNIS2ynGlHqnKA
biuBEr1ji0iHNHczTMuqSYwraVHis+098D+1me5+9SaDp8yEIKgVuTUAuQINBGOV
dwcBEADKRVXM2aPWTzbIFu/dVFuaOsJ653dZyDvkuK3N+6tkxM/ChhwrsaOTcH32
drei59OASOeKW2qYNe2+2tIT5Gt+xxZC9vJ+FcjHcdArAJt0BH1LHSuscCldkrnG
HlXK1dbqXDeD3Vw+7LtSO/wOAllxSRtOO8ydGfPZSWYkBIgKu4IbxKIpwtLJvtGu
mqP6HKgoFgijNcgjzssyV5Xka79vUkZ2M4jEfqtlNMfcYPV5ehjMFJhcGyXPZzkV
byx+321Ov3bvBIKAGK4JRThWRv00qJUhlkUjImr89CUg/4vqxDeGiE+FnwXvi/Vs
tQJqzcYgDvsypsx0LtyUsQ53K/svoXHfltIKsNdBaCPY1i3C+5EfumtIjo7iUl0q
Katpdw2n4j8bQhZq+AhkIcwRzOnuP7/zTXdxdF6VWyekUKYe5ZH738/Ktpb1JR4X
sydKdcV5oX3/Qh43VTIZgJs+fopgQF5NpqLbTXym9KRRHp9cWNyv1DwOIPqD7Eo+
PeTLLyIPXiDAnAt/89EhFDFh/7Lcsicz6n6cnxsGXk3CI/b4NqPkZxr5nC1WAPxB
k6D5PdOZL/3Cu96MvG2cjgdMzIO0/T2loC1DtLKo6qNDMN5JxckoN6EV5XIpT0L3
z3V9Cb8C6XPQ1pa9ftRpAlydPJVroCXPegiULqSkW9SGwtKFFwARAQABiQI8BBgB
CgAmFiEEC+MydddMlTx5+BEH2a0qGAV0dMsFAmOVdwcCGwwFCQICKQAACgkQ2a0q
GAV0dMsMLQ/8CA5hHB3RZchUU6JlAY23itrs3GSIZJIC+fJXOt1D5XuYidhQb5mE
vtK0Eud7FRLHHJ+Sn5DsE68ipZb1m5dCM7oqn2sMQC3H1qX2SR80zEpwx34F7zjs
hTEK7qkQwBwK+zzQtAp209LBBdmRc2nfDPXy6rZF1MeJoBAjRieAlaoO+SmPLlM7
/wZkcBH+fOxU+f4gCHneTIxQdlRQ1/RgAXL002x8QmSMZ8Zd3LNeUsU9JQKH2drn
lOpFmFmUlxXmhDP+MpjycLNppldAJGyYIS1wp/TMd51+jU2WfR2q267vkc1t/x4N
```

```
w2XdCfb5KPD2nz9JdTVrYMl++Aqr04I808V056q0V83HMyu34XAtruItKaz7V13M
sou0sKaC4pGmQixRYVJjsfpkuU47YBchWQTcZzu4gMJM33E3AvbD6FmhZokH+quR
CGMxvlkKZ3I/NJsZdx1jj+qBY/4WSUWLHinjJLycy06PO6xXTOgrDKy8BE0envml
jNG7TxYpda11QAjTo7MtG8U1fDiRH8sZ2Jso75wbwdJHYmlRA3VYcYEqsywF+JoF
2uPf1ejP7szK/p9/wNSXZOFen1ZvEuXBSF8aFGrd3AeMRmk/Tlg96NTT6+nkvVaN
BLCZUZfNUhuvH0vkLZNn19u0+vGBuAGt8zH2cVLFbhS7U7kG0QvnyaSJAjwEGAEK
ACYCGwwWIQQL4zJ110yVPHn4EQfZrSoYBXR0ywUCZYXLqgUJA/J9aAAKCRDZrSoY
BXR0y6DtEACbExPX5JXm+2MzxLj/Rid48fDxnVpu5/vFDXa9uJEuoDl1VHz2a1mu
UqNtYXCMGW+D2Q6h0lUkJ0XIioUpxu4KZWDzRmePhVmy6/5gUWGq45FRS1Lgj3sU
iI8VdSkhiZ1rPWE1QCmQM++RqQjC+ow+m4gjkAkPOsKjnVr2SGUfGL7q68UeuAAQ
dK427GhvHDTlF9XMAH3S8LYX+uHUvtpt5ajRgx5D+XjS9ESMQuYF2BKdJb5pTB92
bjuVq3x+9z0w4PPW8V3roaxZkhMH+MVputN6IvP9qkL0ff+429KPre9TULlq1iBT
jWVloGQVBrKPrsk914GOGKihkd89kMyh2JyejO0r0mKaku0bXMXAlxdBxjXxEvHA
ILSlaRbW7UsF16fDvKC8o5lSDvzY1KWyzhnRSLGq7HsyxVfX9n6metyd+IclEoPC
62FVdXwOQyVKXm21+IS6XZNDDLYBuNwcCiBpSU6S+UZEP7kFN2FtpGoq8u6PtYpg
w9E0c0MbXY1TiOb55UzXHyWSCHy8eHawAQNdjr6V93Xz1etkfvAQJSRU+N9AhugC
EiSAaiURb7opJp2RlLxB5jlLPxS222l0z5CnO7UdliMJY0bzFfkJPOHFnF+mHITL
JTLogzNZrZUGYoN+FhxkhPswlp1wLJSB5UsKMe6n3TtCs9l53EmyAIkCPAQYAQoA
JgIbDBYhBAvjMnXXTJU8efgRB9mtKhgFdHTLBQJnhEEuBQkF0/ICAAoJENmtKhgF
dHTLJtUP/1dPcTrrUm1lOOCkbq3ZxirJOegxZal7YxmCagz6zg6V9zPCA39jLkyD
dsnzzL/fOGwGzgPUS+xGhlZUFFvUFj1ETLGShBUJY4X8USTvSaBg2CO7kQ50hIul
o1b4FJOToiIdChCOIxHGiws73iBHLMK1Q0gJ/u5xr3sD/Ea8e6jFDbywJbcYjMnx
l53mr0bN7H7rM3W/S002PJQlNMa7JWl6673wpGkW7JBYt0t9SPtWQf3sLgIR6n3N
d3kQYz8EDmpyRXqQMz+BheNX1UDfRrbLAEwg/nVejhYWI31iWziHD79u+8ArxiBb
nDPV0Y25AiGBUBOVvtTv2nLdzm49bAfdAvnFToAp4N+2Knzm2CxdfUROPmuzVAIf
xsR4/vE8VMr762llsnzakzA4xzEoTQ4xtKksxV+738lKZnwH8/umne/GpLPSzH/v
OohpLwYVxbO3M1b+D1lmFdflkoTVNmPcjGh986VlGkRHQdOxymy72uGNYnKllcfs
+sbNz77Km03ov5qF4mZO6IaWwZtIbSZZIttgEWKxl0iusOUULQmaAVtAxI+STQY5
Cul2wFt1RbgANSecC0ts9HiEeWnT/v0sUFG06vmRYjhnQT9YP+/N3u8xHcgNGHwN
ilPTnS87qVIsf8fiWHMZq2V53UjDm3dCIljocF5YGq2ySTEXjjkYiQI8BBgBCgAm
AhsMFiEEC+MydddMlTx5+BEH2a0qGAV0dMsFAmluhHIFCQmhpDsACgkQ2a0qGAV0
dMs5ag//V7mahuI9VikX1lG599GSjm1cXeDrZmGtamz9hYE7EzGlyxf5WQn83Sxu
2kNe9UbaLcIpKzZ9CWcb7NfC6odUIzJsO21qSKNuoP7FV9472JTNFMhaj1+o1P8b
v1nGOU4H9IH/y0Wqkuteef3t4le0iCtePMU3vXgfUFAmm/VF/tgmJQb2hLkXFNCQ
QwK//jjm9QHajaU2wEpvskarsRGfGBintBbUqUUxyPgirqAC8dFAWrjefmGePWEW
JfRLql39II+xbjEzADBB8yqnxLuaUQfVF7t2oAOfp09dP++bCua7FO1WfuFoa1Yw
um84/XszGgkxDRFkBxabpMCurPWgyhkGto4TqKdcWTMl4S1rTxt4Bm/6fqMn+04z
/AiXgU/iIbEKhvQUxsTxyNBLPFIRD7VmrqrOr3wkHgGHJgP0656ABq5J+T4mtFhX
uL0U9nGSm5lV9Hl1ana+Kcs4DqzgQO3QJk/SoulBDZ6at0pK2Y016JrvY51fYpx6
a/IOaXE2JRi8itVlpciDTjSOJs4eNlUx0C3b/LdzU5LZqHtBea1xWxlKPn0VFjYr
YNML6o5B6AL05AcySZj8kNZTWE2ZfOR2usLdiB8b+u2F7WGXZBiQYSnr75WsN94S
nYwO46v5SlzGqTwEN/UBfTkKPklVNgeOMn7fgjziEtMmtBlb7ws=
=ijsc
-----END PGP PUBLIC KEY BLOCK-----
```

D.1.2. Secretaris van het Core Team <core-secretary@FreeBSD.org>

```
pub   rsa4096/4D632518C3546B05 2024-02-17 [SC] [expires: 2028-02-08]
      Key fingerprint = 1A23 6A92 528D 00DD 7965  76FE 4D63 2518 C354 6B05
```

```
uid                             FreeBSD Core Team Secretary <core-
secretary@FreeBSD.org>
sub   rsa4096/CABFDE12CA516ED2 2024-02-17 [E] [expires: 2028-02-08]
```

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGXQ1o8BEAC+Rcg8cmVxuP17Vu+q5KgCx/XiulQuqKXAqqBlYCH2jqk6DINP
yFrREGBhzd/qNmlAYEahQ4Zgl0bUZNTrZVDyzicOvPP0jH+KSTQwRs7NOawEdlVO
cyHrwDCPEqf5ZzD4NhfTriEOw+j0pEH/onitUGvoQRtx15xWyaJQxDEBMTYMLewE
86D1bltwnTNczE3UZb7oQLJXkAX5hcLtou70XJGgZITvJkK+kp/xot2eFjnqRz/u
WeXnKhYAmC07EKwZ1uw047eHKwMMRBYqzApLwoQtfE430Kxf2q8de64x8zDbi6YM
1J4r8OAxOtHVyfJ0j7Q23DEZz0VVb4b1Tx5OG2Re/KSNvqI0awJO4TcRmOR88OyY
dzyXgnX6Sa7GVQY1FXvn7vtFuDAt7egZOzeomSHL9bdX07LTQ4UtM88EV9wm3q4q
smoatV9jsvPQ1zxCU3aQD/5eWTJH2/kz1LIuBL/Qi5XQpJn91lBtUWJrCgkHWPGu
f//rnnXmsG7DAChW+yZ7cFO8lfNa8sFhPqSxCYphWmJTrvadyQtDngB8JakWdnmK
pfGS6y5lel+181vw38ZZKt04AKM+nDY8051lBM7Q9Q6kTLI33UZeImndx5xYukVD
kV6aQ31HYfEark15c7iEz+OAcwFnM2ntXMt7kKGd40CqzusiPcQkPqPbAQARAQAB
tDhGcmVlQlNEIENvcmUgVGVhbSBTZWNyZXRhcnkgPGNvcmUtc2VjcmV0YXJ5QEZy
ZWVCU0Qub3JnPokCVwQTAQoAQQIbAwgLCQ0IDAcLAwUVCgkICwUWAwIBAAIeBQIX
gBYhBBojapJSjQDdeWV2/k1jJRjDVGsFBQJnp8PiBQkHeofTAAoJEE1jJRjDVGsF
GMIQALhj+mNpH8OmTFeihQ6t9P8un3llz6Wmqe/Q+ULWeqJvV/uC1J5T9fnoGhwF
MgECuguXYJtoYQ16KXnsSOs1tcqMOK9GtEtFJTGe2DtfLiBednwUvu9j3HmTlwLN
M+7rqkiC0HCg2qSjcMjxbVbA5BSgNkgfyTSO2YdjfaZ+ceiHwo/qa5xWE6i2dMR1
PLGMMHTeELdtdSH6VR9/3h0qt3qzwdMBRCVAQHbim7CqwjUH9jg+IOySXl81jNoB
xuVZR3pKshyY4Oxo1dK+W86Uiff/+c4jCAxokGWIR7C4MkZZWUQqV7920gkZFC/5
qzpT1A1/sFUg8HfFTr0vCoPSVFWn2+Tto3vr78DICVaAf02aYAFlyKK18BMCoHkS
hDDO+/JQZmvHOIgEYK+T2WN1c9gm9IDJzGZuH0X2C/vkREnJKkccJfB4pXuN10wt
fqyP9fn8h6+/t+sv3qNLm4d8fkmLXofulG3WB4i/F+Hip2rjPvvBCF7zl4xy6cJ2
xVY5HUOBTqmIwVhYwUpXaqNoWa/qJBLTuot3z7ciKmKX6Lq+Dze5dzhrPNl/CalC
HBf3miHRK3TZbYLooG3bcEWgxU2BnBi3vl5NpCoUOKkPYRlALXi2TyHmPxO7oT4e
mIzS6BgnXOqO+AXvbKKfayuSePdBqkNMK/SMC4Dylkf6Xj25iQIzBBABCgAdFiEE
EBpxaxYrAOVb7eoFrbv4YQo3ibcFAmbu7x4ACgkQrbv4YQo3ibcr5A//TIcbD/EW
YJz0zrUQdc9xG3UNfU6uHmQzAuUy5ginevyqv0TSso5qvSkOHvdxbi41rfMiB2RJ
V3q0n0PSvHFld89fOTZUMZXaPvozCiBYWScrt+KA/2pq3K8mUumHS+IFtpHLL2Tu
+gI8XCHUFxO9HUThM/rFlIyEdzoctgmqQ7IG4uZG+o2J3w09llhDAUe0vraJK1On
p9yFACnRrhqvl41JeUWcv9MH9JcwHUqtUo9WLTCIb+hkByTOyRfHBYpYw//bdXdf
6rkCwKVWpyMDbk61zq2VsS1kqbP9IH/A8CsBnA6mg+zPq2i7HIFw8Swj4OGJcIvG
a9ubUYJRjDhX/vBpNrtncANZ88FQmA+Maq0vu0LS5IIGyIKkvd1fKIsyvBDDa9kE
nfCW0XMkJA0Gf+kxdB+elXQHBwKO2sr5BiKnJT5lJq3Yu9fxxGBnf93yiN4E9bmF
gG7cZxpyb1Bp76TJhLcANyybOTjCtiNRrgqeaSx9/6hSPfPigGXIne0H2lmJO6oq
jUrsYmFiWvU9sc7AcPVw/eHG8FgW35TuwKX71z8w994iaahUPNcSVyXOUD+QNROv
HhGUXrc81t613rivh22N0NZpNubVatq43KV7+/bnPyWBIi1Awh3vIFsNNSQSrYxF
lUuQaAHQXTeZMZ/7npE4t86seMt0T7BGn765Ag0EZdDWjwEQAL3VwFifpnRCYzQ4
VZ1dAjp8w542iRprqeA+C3tvNbk2OvKpN3DIc49l2bgNZ/VI7/T58lEKrFsgLK4w
AWtv18OV7xuh0AuOblmq5MvcPrUelkPj5HA7M6Ng/rAubuHfwdP/IjIrzzY6+XDh
fP9N5KIv9VRJYT23BbvLPeit4J4tQEB/NpxL3L6zI75qq4R3T/EkHP6Y9Buup9Lr
isJlcxkSK+CyORCgTpEz6fWsXiTDgS7cTaQ969XCygBpj4wRQzwkBdokxo1wsiFt
4zLt07/PrPYjeHltTDkrF9XDNhLNJ5GliHnd01oHg1j5/n+9Osvh1maoFwuBxXTN
nTZ2P+7RKLBaVQVSkUa5KJbXoM3v7bMbXM5aLs2XjglrAzrZOV+Y7zOu5UYQdpXd
7x8XAEtEozRJzt7dosQIhKx9h4LlltFFulDUpf5VCvcUEoAzMbiX0aju9n5Rwsql
```

```
DHatU9Mm3W8OdFXj1foIXZD+VX2Jp9DgxPLoAJ0CWhPXJ8f+WFSJZCjWoPJEJKWY
0EOxiXyAuvAyniAZAC/eKZkfGckXmu7edRgYbRTTwPmZ/axa/k9aHsHLwmbxuZo/
xxQXzqU70EleHZ31A3mOqsC1epjN6dn4AFKgvVesP/K/fbvSsfsfiABS2A/68ne4
zJG73Tnk4L6J79vrXc2iMJbmdg3ZABEBAAGJAjwEGAEKACYCGwwWIQQaI2qSUo0A
3Xlldv5NYyUYw1RrBQUCZ6fEFQUJB3qIBgAKCRBNYyUYw1RrBQd9EACMVckxzy4w
aUGlERguJ+kslS8MkMjNqnfDPLRDVxbUxNvbbhw7/u9blM65DCGCeNLc2n4oiu5C
E3I095AKmvq/0dOa81mEEdZkC1CVc64bXWbEyz5AtSHUpgdxRso6C+YopndSiz1T
WcIagQRXfWaw3FBWPooA87gmibmSmCegCtqx+uyc5QxX2eFI8mRK7vlfnGpYKHs0
D1/yUSGQOwoNRJ5FYm+ynfDE3FzHEQL7lv1vpv1k3xNKfBziMMg4IMEBKNHV4VKN
qJpa8UCodeTGSWQdNnCeCWPsxz5oQjCcVH9Z3e8sMpWLHhRcBZzSwXUOws2GbRMH
xHwrfrpHJcrBhe64pgfGOvZLUJ9BDs+8egTHsqRFacipbtTR+hhVhuJEHDaQQWuM
8IRHj9HIuTAczET8JTDHTMIoo8DZdOtiW/YgqCDwYghkI77d12oNQqYoeJ2HiqbK
cGzwCpsR0A+p/iOAxJG13tsxqZV8TQ8iTokWG6ACtZ7sfeWEhxqMbUKUMgogZn0I
3n1kV+tUZC6BQrIYI7TiKg95wLZsIydeoIsQoNZwvyKAXfVmQ62YjIX8njZwN+07
8/ipUPJxCYa8zL8BZyDmoFJqa3y9z+11+vtiZ9t+aTwGvjpHDwyeCJco7go9cU3m
GRFZYciqIoG4n3tl8Pob15vFlVqk47rRqg==
=8TzT
-----END PGP PUBLIC KEY BLOCK-----
```

D.1.3. Secretaris van het Ports Beheerteam <portmgr-secretary@FreeBSD.org>

```
pub   ed25519/E3C401F60D709D59 2023-03-06 [SC] [expires: 2027-03-05]
      Key fingerprint = BED4 A1D3 6555 B681 2E9F  ABDA E3C4 01F6 0D70 9D59
uid                           FreeBSD Ports Management Team Secretary <portmgr-
secretary@FreeBSD.org>
sub   cv25519/2C92B55E27A641C3 2023-03-06 [E] [expires: 2027-03-05]
```

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mDMEZAXJvxYJKwYBBAHaRw8BAQdASFAC20WL3R1T6uNyGMZbfJCxDkcP4C5vi3Op
tcZ2fbq0R0ZyZWVCU0QgUG9ydHMgTWFuYWdlbWVudCBUZWFtIFNlY3JldGFyeSA8
cG9ydG1nci1zZWNyZXRhcnlARnJlZUJTRC5vcmc+iJYEExYKAD4WIQS+1KHTZVW2
gS6fq9rjxAH2DXCdWQUCZAXJvwIbAwUJB4TOAAULCQgHAwUVCgkICwUWAwIBAAIe
BQIXgAAKCRDjxAH2DXCdWYN1AP43TjyfZtZ3DLYT++g0+SuPsoO/3yWVybA+UmFL
zb8MngEA+LLNUfvEwCuXS/soh+ww5bpfmi3UUmeGiQEAXug3iA+JATMEEAEKAB0W
IQT7N0XIbxXo7ayBMvzYKU7Du8TX1QUCZAXLkwAKCRDYKU7Du8TX1XHMB/9R1MX4
6zMgpKqPPt76GOI+eGEdBK6bY8aJZjQGdqTh9f6VtXVoTGIG7cvhc9X8tDBoB0PT
2KZWheF51AV1+NHU4HwLAQ1BMebrFvWSfkw4xg4fBGwDhz9/GN85No+Js772V5ey
8lRiL6meRVWxMlLyWcxGd8JjcC5yX/iAUQ3SBGCLqW7unWjjg7CTd+AMBwcqPGrv
ax8q6eFVguJcHJAjMnKf6HAy4cpK3s+uMoUBCGnszSN12B3ysKfyC4pNO/pix5tA
Q5v8aRqTeFPh5zmNhWo0KGPzplTPqRQSHDl7GDQC8Ru3MhzFkeWzHsexjZVwS6W2
DPcYpuuAsA0XOZIZiQIzBBABCgAdFiEEEEBpxaxYrAOVb7eoFrbv4YQo3ibcFAmQF
0u0ACgkQrbv4YQo3ibccwg/9F2Xuic3nhKxRbB3mJeDo6SYQETa/Gh1qQ34+8zlt
8UMazOx67gnYQfy+pXjro6eQ2up0a4eUYezcNOudqAQD21nRz3HA6EQVNcE/TzEA
xl5CJntTaLOt7S+EDXFW5BuQIvhhoMGgm8+WNVgA0EJ7tfL0OcYBSvr19fqwChEn
9c14cSk6mgHSsleP5NvskYN053pxHwy0LTSb8YBBv52th37t/CRFC1363rS5q+D7
JixFopd1O5pKpA5ipvE4gGgRjPtwjx0SjjepwK/3fuhEJQQyKzTIKlMfu2Dj/iR2
Li1Sfccau5LQXOj9fUITU3u1YG7yrm8VGzT7ao4d+KRwgMLjd2pLqiGIbbJwGBiP
```

```
FRmtilWQoeIlmSlFX4obAA517DOK0pW1mH8+eEn4EJd3SekT3yzFyKTASv0J48Z8
3F928xg+eZvHxVC0t1J+J5IG0gt3EEncuWKIPQGR7PiQbti6R3FQVTz6WfMWOebP
Qi0E9F/Aqakr6Vj2sKGrDq+ebpaF5G8Yw1YrUl2IDiPzkCegp3ZbI0wh11Xvzhi8
LXPQGK4jBQas4G8cegfitzmtdGRHYrbMv0R9I4mvaL+WlOuD2AvyVG28lguqVhnN
AZP+ohdquYyX2CNCVvbKWAtXo6Ur0vWG8BL8m6defAtEkIwVBALaOHQOSI3aNUz4
lwy4OARkBcm/EgorBgEEAZdVAQUBAQdAsefmSfxEOdOr02+K/6noYCuJ1FeAWVz6
jFYQ+9w6jggDAQgHiH4EGBYKACYWIQS+1KHTZVW2gS6fq9rjxAH2DXCdWQUCZAXJ
vwIbDAUJB4TOAAAKCRDjxAH2DXCdWRl4AP9h5ot212BK29S6ZcMBhHvmtF5PG1oD
c7LnZycSRmbFiwEAndCMpAGOhDW8iVgDd0wLQq/ZMPe+xccfG1b3zFH2EgE=
=iiAT
-----END PGP PUBLIC KEY BLOCK-----
```

D.1.4. <doceng-secretary@FreeBSD.org>

```
pub   rsa2048/E1C03580AEB45E58 2019-10-31 [SC] [expires: 2022-10-30]
      Key fingerprint = F24D 7B32 B864 625E 5541  A0E4 E1C0 3580 AEB4 5E58
uid                           FreeBSD Doceng Team Secretary <doceng-
secretary@freebsd.org>
sub   rsa2048/9EA8D713509472FC 2019-10-31 [E] [expires: 2022-10-30]
```

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQENBF27FFcBCADeoSsIgyQUY8vREwkTikwFFlNg31MVy5s/Nq1cNK1PRfRMnprS
yfB62KqbYuz16bmQKaA9zHN4FGfiTvR6tl66LVHm1s/5HPiLv8sP14GsruLro9zN
v72dO7a9i68bMw+jarPOnu9dGiDFEI0dACOkdCGEYKEUapQeNpmWRrQ46BeXyFwF
JcNx76bJJUkwk6fWC0W63D762e6lCEX6ndoaPjjLBnFvtx13heNGUc8RukBwe2mA
U5pSGHj47J05bdWiRSwZaXa8PcW+20zTWaP755w7zWe4h60GANY7OsT9nuOqsioJ
QonxTrJuZweKRV8fNQ1EfDws3HZr7/7iXvO3ABEBAAG0PEZyZWVCU0QgRG9jZW5n
IFRlYW0gU2VjcmV0YXJ5IDxkb2Nlbmctc2VjcmV0YXJ5QGZyZWVic2Qub3JnPokB
VAQTAQoAPhYhBPJNezK4ZGJeVUGg5OHANYCutF5YBQJduxRXAhsDBQkFo5qABQsJ
CAcDBRUKCQgLBRYDAgEAAh4BAheAAAoJEOHANYCutF5YB2IIALw+EPYmOz9qlqIn
oTFmk/5MrcdzC5iLEfxubbF6TopDWsWPiOh5mAuvfEmROSGf6ctvdYe9UtQV3VNY
KeeyskeFrIBOFo2KG/dFqKPAWef6IfhbW3HWDWo5uOBg01jHzQ/pB1n6SMKiXfsM
idL9wN+UQKxF3Y7S/bVrZTV0isRUolO9+8kQeSYT/NMojVM0H2fWrTP/TaNEW4fY
JBDAl5hsktzdl8sdbNqdC0GiX3xb4GvgVzGGQELagsxjfuXk6PfOyn6Wx2d+yRcI
FrKojmhihBp5VGFQkntBIXQkaW0xhW+WBGxwXdaAl0drQlZ3W+edgdOl705x73kf
Uw3Fh2a5AQ0EXbsUVwEIANEPAsltM4vFj2pi5xEuHEcZIrIX/ZJhoaBtZkqvkB+H
4pu3/eQHK5hg0Dw12ugffPMz8mi57iGNI9TXd8ZYMJxAdvEZSDHCKZTX9G+FcxWa
/AzKNiG25uSISzz7rMB/lV1gofCdGtpHFRFTiNxFcoacugTdlYDiscgJZMJSg/hC
GXBdEKXR5WRAgAGandcL8llCToOt1lZEOkd5vJM861w6evgDhAZ2HGhRuG8/NDxG
r4UtlnYGUCFof/Q4oPNbDJzmZXF+8OQyTNcEpVD3leEOWG1Uv5XWS2XKVHcHZZ++
ISo/B5Q6Oi3SJFCVV9f+g09YF+PgfP/mVMBgif2fT20AEQEAAYkBPAQYAQoAJhYh
BPJNezK4ZGJeVUGg5OHANYCutF5YBQJduxRXAhsMBQkFo5qAAAoJEOHANYCutF5Y
kecIAMTh2VHQqjXHTszQMsy3NjiTVVITI3z+pzY0u2EYmLytXQ2pZMzLHMcklmub
5po0X4EvL6bZiJcLMI2mSrOs0Gp8P3hyMI40IkqoLMp7VA2LFlPgIJ7K5W4oVwf8
khY6lw7qg2l69APm/MM3xAyiL4p6MU8tpvWg5AncZ6lxyy27rxVflzEtCrKQuG/a
oVaOlMjH3uxvOK6IIxlhvWD0nKs/e2h2HIAZ+ILE6ytS5ZEg2GXuigoQZdEnv71L
xyvE9JANwGZLkDxnS5pgN2ikfkQYlFpJEkrNTQleCOHIIIp8vgJngEaP51xOIbQM
```

```
CiG/y3cmKQ/ZfH7BBvlZVtZKQsI=
=MQKT
-----END PGP PUBLIC KEY BLOCK-----
```

Unresolved directive in book.adoc - include::glossary.adoc[leveloffset=+1]

Colofon

Dit boek is het gezamenlijke werk van honderden vrijwilligers aan het "FreeBSD Documentatie Project". De tekst is opgesteld in AsciiDoc.