

Unix/Linux toepassingen

CURSUSMATERIAAL VOOR

Operating systems 9

UNIX 3

VOOR DE OPLEIDING

Systeembeheer dual

Instituut voor Innovatie, Industrie en Informatica

Faculteit Natuur en Techniek

Hogeschool van Utrecht

Martijn P. Rijkeboer

30 december 2004

Voorwoord

Dit dictaat is geschreven als onderdeel van mijn afstudeerproject voor de opleiding Hogere Informatica aan de Hogeschool van Utrecht, Faculteit Natuur en Techniek. De opdracht van dit afstudeerproject is het ontwikkelen van cursusmateriaal voor het vak “Unix 3” van de duale opleiding Systeembeheer van de studierichting Informatica aan dezelfde faculteit.

Lezers die vooral geïnteresseerd zijn in de theoretische onderdelen worden verwezen naar de eerste paragrafen van ieder hoofdstuk. De practicum opdrachten komen na deze theoretische paragrafen aan bod. Het zwaartepunt van deze cursus zal liggen in het opdoen van praktijkervaring.

De auteur is veel dank verschuldigd aan Dhr. H. Karssenberg docent aan de Hogeschool van Utrecht en Dhr. drs. L.J.M. van Moergestel eveneens docent aan de Hogeschool van Utrecht voor hun adviezen en begeleiding tijdens dit afstudeerproject.

Martijn P. Rijkeboer
Utrecht, mei 2004

Inhoudsopgave

1	Inleiding	1
1.1	Uitgangssituatie	1
1.2	Doelstellingen	1
1.3	Licentie	2
2	De Unix besturingssystemen	3
2.1	Inleiding	3
2.2	Installatie overzicht	4
2.2.1	Kiezen installatie medium	5
2.2.2	Beginnen met installatie	5
2.2.3	Configureren harddisk indeling	5
2.2.4	Globale systeem configuratie	5
2.2.5	Installeren sets	6
2.2.6	Afronden installatie	6
2.2.7	Starten vanaf harddisk	6
2.3	Netwerk opzetten	6
2.3.1	myname	7
2.3.2	mygate	7
2.3.3	resolv.conf	7
2.3.4	hostname.if	7
2.4	Practicum installeren NetBSD	8
2.4.1	Kiezen installatie medium	8
2.4.2	Beginnen met installatie	8
2.4.3	Selecteren sets	8
2.4.4	Configureren harddisk indeling	9
2.4.5	Installeren sets	9
2.4.6	Afronden installatie	9
2.4.7	Starten vanaf harddisk	9
2.5	Practicum installeren OpenBSD	10
2.5.1	Kiezen installatie medium	10
2.5.2	Beginnen met installatie	10
2.5.3	Configureren harddisk indeling	10
2.5.4	Globale systeem configuratie	11
2.5.5	Installeren sets	11
2.5.6	Afronden installatie	11
2.5.7	Starten vanaf harddisk	12

2.6	Practicum configureren netwerk	12
3	Unix onderhouden	13
3.1	Inleiding	13
3.2	Binary patches	13
3.3	Source patches	14
3.4	Practicum patchen source tree	14
3.5	Practicum updaten source tree	15
3.6	Practicum kernel compileren	15
4	Extra applicaties	17
4.1	Inleiding	17
4.2	Voorgecompileerde applicaties	17
4.3	Applicaties in source-vorm	18
4.4	Practicum installeren applicaties	19
5	Webservices	21
5.1	Inleiding	21
5.2	HTTP servers	21
5.2.1	Aparte user en group	21
5.2.2	Chroot	22
5.2.3	Uitgeklede configuratie	22
5.2.4	Modules	26
5.2.5	Access control	26
5.2.6	Virtuele hosts	26
5.2.7	PHP	27
5.2.8	HTTPS	28
5.3	DNS servers	29
5.3.1	Authoritative Name Server	29
5.3.2	Caching Name Server	29
5.3.3	Combinatie	30
5.3.4	Beveiliging	30
5.3.5	Configuratie	30
5.4	Practicum opzetten DNS server	31
5.5	Practicum opzetten webserver	31
6	Fileservices	32
6.1	Inleiding	32
6.2	Anonieme FTP	32
6.2.1	Active mode	32
6.2.2	Passive mode	33
6.2.3	Beveiliging	33
6.3	Unix filesharing	34
6.4	MS Windows filesharing	34
6.5	Practicum anonieme FTP	35
6.6	Practicum NFS	35
6.7	Practicum SAMBA	36

7	Beveiliging	37
7.1	Inleiding	37
7.2	Firewall	37
7.2.1	Packet Filtering	38
7.2.2	Network Address Translation	39
7.2.3	Port Forwarding	39
7.2.4	Packet Normalization	40
7.3	Host hardening	41
7.3.1	Toegangsbeperking	41
7.3.2	Uitschakelen ongebruikte services	41
7.3.3	Verwijder ongebruikte applicaties	41
7.3.4	Bestandssysteem permissies	42
7.3.5	Kernel aanpassen	42
7.4	Policies	42
7.4.1	Gebruikspolicies	42
7.4.2	Root account	43
7.5	Practicum beveiligen OpenBSD	43
8	Overige onderwerpen	44
8.1	Backups	44
8.1.1	Inleiding	44
8.1.2	Utilities	44
8.1.3	Voorbeeld backupscript	46
8.1.4	Backup media	47
8.2	Systrace	48
8.3	Practicum maken encryptedbackup script	49
8.4	Practicum opzetten systraced sftp shell	49
9	Conclusie	50
A	Cheatsheet vi	51
B	Versies	53
	Bibliografie	54

Hoofdstuk 1

Inleiding

1.1 Uitgangssituatie

Dit dictaat is geschreven voor het vak “Unix 3” van de opleiding Systeembeheer aan de Hogeschool van Utrecht, Faculteit Natuur en Techniek. Dit vak is het negende vak binnen het thema “Operating Systems” en het derde vak in deze groep met het onderwerp Unix/Linux.

Gezien de voorkennis van de student wordt er niet ingegaan op basiskennis, maar wordt verondersteld dat deze inmiddels aanwezig is. Met name de kennis van de twee voorgaande Unix vakken is van belang.

Ook wordt er vanuit gegaan dat de student in staat is om manpages (manual-pages) te lezen. Deze pagina’s kunnen op het systeem zelf worden gelezen met behulp van het commando `man`. Daarnaast zijn de meeste manpages ook beschikbaar op de websites van FreeBSD [6], NetBSD [14] en OpenBSD [20].

Manpages worden, in het algemeen, onderverdeeld in verschillende secties waarbij iedere sectie wordt aangeduid met een cijfer. In dit document worden verwijzingen naar manpages voorzien van het cijfer van de sectie die wordt bedoeld. Een voorbeeld hiervan is `tar(1)`, in dit geval moet er `man 1 tar` worden uitgevoerd.

1.2 Doelstellingen

De doelstelling van dit vak is de student kennis bij te brengen over hoe deze Unix/Linux, als toekomstig systeembeheerder, kan inzetten binnen het bedrijfsleven. Er zal dan ook vooral aandacht worden besteed aan toepassingen die veel worden ingezet in het bedrijfsleven.

Om de student toepasbare kennis bij te brengen is er voor gekozen een groot deel van dit vak als zogenaamde “hands-on” cursus te maken. Dit houdt in dat de student vooral veel zelf achter de computers bezig is met de stof.

Alle practicum opdrachten die behoren bij deze cursus zullen worden uitgevoerd op NetBSD [11] en/of OpenBSD [18]. Er is gekozen voor deze twee Unix-achtige systemen, omdat ze hoog staan aangeschreven wat betreft beveiliging

en stabiliteit. Daarnaast is het van belang dat de systeembeheerder in spé niet alleen ervaring krijgt met Linux maar ook met andere Unix-achtigen.

Andere belangrijke punten bij de keuze van de te gebruiken besturingssystemen zijn dat ze vrij beschikbaar moeten zijn voor de student en dat ze moeten draaien op de meeste computers, zodat de student er ook thuis mee kan oefenen.

1.3 Licentie

Copyright (c) 2004, Martijn P. Rijkeboer
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of the authors nor the names of their contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Hoofdstuk 2

De Unix besturingssystemen

2.1 Inleiding

Vanaf het moment dat de broncode van de oorspronkelijke versie van UNIX, ontwikkeld door AT&T's Bell Laboratories, vrij beschikbaar kwam zijn er veel verschillende versies van UNIX ontstaan. De huidige afstammelingen van de oorspronkelijke versie van UNIX zijn globaal in twee groepen te verdelen, namelijk de "BSD" en de "System V" groep. De grootste verschillen zullen verderop in dit document besproken worden.

Voorbeelden uit de BSD groep zijn:

- BSDi
- FreeBSD
- NetBSD
- OpenBSD

Voorbeelden uit de System V groep zijn:

- AIX
- HP-UX
- Linux (meeste distributies)
- Unixware

Linux is echter op twee manieren een vreemde eend in de bijt. Ten eerste is het geen afstammeling van de originele UNIX, maar een kloon. Ten tweede is Linux geen besturingssysteem op zich, maar zijn er zogenaamde distributies die de Linux kernel combineren met allerlei extra applicaties om er een besturingssysteem van te maken. De meeste distributies behoren tot de System V groep. Voorbeelden uit deze groep zijn:

- Debian GNU/Linux
- Mandrake Linux
- Red Hat Linux
- Suse Linux

Het meest zichtbare verschil tussen de BSD en System V groep is de manier van opstarten. Generaliserend maakt de BSD groep gebruik van een paar scripts voor alle programma's en de System V groep één script per programma. Er zijn nog meer verschillen, maar die zijn niet zo duidelijk zichtbaar voor de beheerder. Zoals in de inleiding al is vermeld, is er voor dit vak gekozen om gebruik te maken van NetBSD en OpenBSD als besturingssystemen. Deze twee besturingssystemen zijn op een aantal punten anders dan het besturingssysteem (Linux) waar in de vorige vakken mee is gewerkt, namelijk:

1. Van zowel NetBSD als OpenBSD is er maar één versie, terwijl er van Linux zeer veel verschillende versies/distributies zijn. Dit komt doordat Linux in principe alleen een kernel is en de distributies er allerlei programma's aan toevoegen om er een compleet besturingssysteem van te maken. Dit laatste wil er nog wel eens voor zorgen dat de samenhang en als gevolg daarvan, de stabiliteit, minder is dan bij de complete systemen.
2. NetBSD en OpenBSD maken, in tegenstelling tot Linux, gebruik van disklabels. Disklabels maken het mogelijk om verschillende partities in één "harddisk" partitie te creëren. Deze harddisk partitie wordt voor de duidelijkheid ook wel een "slice" genoemd.
3. NetBSD en OpenBSD moeten, in tegenstelling tot Linux, altijd op een primaire partitie geïnstalleerd worden. Dit kan tot problemen leiden wanneer alle vier primaire partities al in gebruik zijn.
4. In sommige programma's, onder NetBSD en OpenBSD, zitten andere opties dan onder Linux. Dit komt vooral doordat Linux gebruik maakt van de GNU versies en de andere twee gebruik maken van de originele BSD versies.

2.2 Installatie overzicht

De installaties van NetBSD en OpenBSD lijken heel veel op elkaar en zijn globaal in een aantal onderdelen te verdelen. De hieronder genoemde onderdelen staan in de volgorde die gebruikt wordt door OpenBSD. In NetBSD is de volgorde iets anders, maar dit maakt niet veel uit.

- Kiezen installatie medium;
- beginnen met installatie;

- configureren harddisk indeling;
- globale systeem configuratie;
- installeren sets;
- afronden installatie;
- herstarten vanaf harddisk.

2.2.1 Kiezen installatie medium

Tijdens de practicumopdrachten wordt er gebruik gemaakt van installatie cd-roms. Er is gekozen voor deze methode, omdat er weinig mee fout kan gaan en het een erg snelle methode is. Naast het installeren vanaf cd-rom ondersteunen beide besturingssystemen ook het installeren vanaf diskette en netwerk.

2.2.2 Beginnen met installatie

Wanneer er opgestart is vanaf de cd-rom verschijnt het installatie programma. Dit programma doorloopt, aan de hand van vragen, het installatieproces. Het eerste deel van dit programma stelt algemene vragen die betrekking hebben op het installatie proces.

2.2.3 Configureren harddisk indeling

Dit onderdeel van het installatieproces is het meest risicovolle onderdeel, aangezien er aanpassingen gemaakt worden aan de indeling van de partities op de harddisk. Tijdens de practicumopdrachten wordt altijd de hele harddisk gebruikt en er hoeven dan ook geen extra partities aangemaakt te worden.

Binnen deze aangemaakte partitie zal met behulp van disklabel(8) een aantal "subpartities" aangemaakt worden. Om verwarring te voorkomen worden de hoofdpartities "slices" en de subpartities "partities" genoemd. Deze conventie zal in de rest van dit document gebruikt worden.

Verder is het van belang om de "bootcode" in de "master boot record" te zetten. Dit is echter niet aan te raden wanneer er meerdere besturingssystemen op dezelfde harddisk moeten komen.

2.2.4 Globale systeem configuratie

Binnen dit onderdeel van het installatieproces worden een aantal fundamentele onderdelen van het besturingssysteem geconfigureerd, zoals bijvoorbeeld de hostname, netwerksetup en het root password.

2.2.5 Installeren sets

Binnen dit onderdeel van het installatieproces kan er worden geselecteerd welke onderdelen van het besturingssysteem moeten worden geïnstalleerd, deze onderdelen worden over het algemeen “sets” genoemd. Deze sets kunnen worden gezien als soort modules. Om een werkend systeem te krijgen moeten er minimaal drie sets geïnstalleerd worden namelijk de kernelset, de basisset en de configuratieset. Voorbeelden van andere sets zijn de compilerset en de documentatieset.

Om de sets te kunnen installeren dient eerst opgegeven te worden waar ze te vinden zijn. Tijdens de practicumopdrachten zal altijd gebruik worden gemaakt van de sets op de cd-roms.

2.2.6 Afronden installatie

Tijdens de laatste fase van het installatieproces moeten er nog een paar kleine dingen worden ingesteld, zoals de tijdzone en of er een ssh(1) server op het systeem moet draaien.

2.2.7 Starten vanaf harddisk

Hierna kan de cd-rom worden verwijderd en de computer opnieuw worden gestart. Als alles goed is verlopen staat er nu een werkend besturingssysteem op de harddisk.

2.3 Netwerk opzetten

Tijdens de installatie is het mogelijk om een simpele configuratie voor het netwerk te maken. Wanneer er tijdens de installatie geen netwerk setup is gedaan, dient deze naderhand te worden gemaakt. Binnen NetBSD en OpenBSD zijn er twee manieren om het netwerk te configureren.

Ten eerste kan het netwerk run-time worden geconfigureerd met de programma's ifconfig(8) en route(8). Een nadeel van deze methode is dat de configuratie na een herstart verdwenen is en dus weer opnieuw moet worden aangemaakt.

Daarnaast kan het netwerk door middel van een aantal bestanden worden geconfigureerd, hierbij zijn de onderstaande bestanden van belang:

- /etc/myname, zie myname(5);
- /etc/mygate, zie mygate(5);
- /etc/resolv.conf, zie resolv.conf(5);
- /etc/hostname.if (OpenBSD), zie hostname.if(5);
- /etc/ifconfig.if (NetBSD): zelfde informatie als /etc/hostname.if.

2.3.1 myname

In het bestand `/etc/myname` moet de volledige hostnaam van de machine staan. Een voorbeeld `myname` bestand ziet er als volgt uit:

```
openbsd.fnt.hvu.nl
```

2.3.2 mygate

In het bestand `/etc/mygate` staat het ipadres van de default-gateway. Het ipadres van de default-gateway wordt gebruikt om te bepalen wat de default route is. Een voorbeeld `mygate` bestand ziet er als volgt uit:

```
192.168.0.1
```

2.3.3 resolv.conf

In het bestand `/etc/resolv.conf` staan de ipadressen van de DNS servers en eventueel het standaard domein. Een voorbeeld `resolv.conf` bestand ziet er zo uit:

```
search fnt.hvu.nl
lookup file bind
nameserver 145.89.38.3
nameserver 62.251.0.6
```

In dit bestand geeft de eerste regel aan wat het standaard domein voor deze machine is. De tweede regel geeft aan dat er eerst in het bestand `/etc/hosts` moet worden gekeken en wanneer de hostname daar niet in voorkomt er dan pas een DNS lookup moet worden gedaan. De laatste twee regels bevatten de ipadressen van de DNS servers. Er kunnen maximaal drie DNS servers worden opgegeven.

2.3.4 hostname.if

Het bestand `/etc/hostname.if` bestaat niet echt, het “if” gedeelte moet namelijk vervangen worden door de interface naam waar deze configuratie voor bedoeld is. Bijvoorbeeld `/etc/hostname.xl0` is de configuratie voor de interface “xl0”. Een voorbeeld van een `hostname.if` bestand ziet er als volgt uit:

```
inet 192.168.0.5 255.255.255.0 NONE
```

Deze regel is als volgt opgebouwd:

1. `inet`: de adresfamilie, in dit geval IPv4;
2. `192.168.0.5`: het ipadres van deze interface;

3. 255.255.255.0: de netmask van deze interface;
4. NONE: de extra opties voor deze interface.

Note: Bij NetBSD moet er tussen het ipadres en de netmask het woord “netmask” worden gezet.

Het voordeel van deze manier van configureren is dat de informatie ook blijft bestaan na een herstart van de machine. Wanneer er gebruik wordt gemaakt van DHCP op het netwerk, zijn geen van bovengenoemde methoden nodig. Er kan dan worden volstaan met het uitvoeren van het programma `dhclient(8)` met als optie de interface waarvoor de configuratie moet gebeuren.

2.4 Practicum installeren NetBSD

In dit practicum gaat de student een NetBSD installatie doen. Hieronder zullen een aantal tips worden gegeven waarop moet worden gelet tijdens de installatie. Voor een volledig installatievoorbeeld wordt verwezen naar het installatie onderdeel van de NetBSD Guide [12] (<http://www.netbsd.org/guide/en/chap-inst.html> en <http://www.netbsd.org/guide/en/chap-exinst.html>).

2.4.1 Kiezen installatie medium

Zoals al eerder vermeld wordt NetBSD geïnstalleerd vanaf een cd-rom. Deze cd-rom zal voor de duur van het practicum aan de student uitgeleend worden.

2.4.2 Beginnen met installatie

- Start op vanaf de cd-rom;
- kies voor de optie om NetBSD op de harddisk te installeren;
- kies voor een “custom” instalatie;

2.4.3 Selecteren sets

Selecteer de onderstaande sets:

- Kernel (Generic)
- Base
- System (/etc)
- Compiler Tools
- Game
- Online Manual Pages

- Miscellaneous
- Text Processing Tools

Alle andere sets moeten niet geselecteerd zijn.

2.4.4 Configureren harddisk indeling

- Selecteer de optie “Use the entire disk”;
- overschrijf de harddisk met NetBSD;
- vervang de bootcode;
- maak de volgende partitie indeling in de NetBSD slice:
 - / (100 MB)
 - swap (1024 MB)
 - /usr (5500 MB)
 - /var (500 MB)
 - /home (rest)
- laat de partities op de harddisk aanmaken;
- selecteer de optie “Use BIOS console”.

2.4.5 Installeren sets

- Selecteer als voortgangsindicator de “progressbar”;
- selecteer als medium cd-rom.

2.4.6 Afronden installatie

- Selecteer als tijdzone “Europa/Amsterdam”;
- selecteer als password cipher “MD5”;
- stel het root password in;
- selecteer als shell “/bin/ksh”.

2.4.7 Starten vanaf harddisk

- Verwijder de cd-rom;
- herstart de computer;
- er draait nu een werkend NetBSD systeem.

Laat de bovenstaande opdracht door de docent aftekenen.

2.5 Practicum installeren OpenBSD

In dit practicum gaat de student een OpenBSD installatie doen. Hieronder zullen een aantal tips worden gegeven waarop moet worden gelet tijdens de installatie. Voor een volledig installatievoorbeeld wordt verwezen naar het installatieonderdeel van de OpenBSD FAQ [\[19\]](#)

2.5.1 Kiezen installatie medium

Zoals al eerder vermeld wordt OpenBSD geïnstalleerd vanaf een cd-rom. Deze cd-rom zal voor de duur van het practicum aan de student uitgeleend worden.

2.5.2 Beginnen met installatie

- Start op vanaf de cd-rom;
- kies voor de optie “Install”;
- accepteer het standaard terminaltype (vt220)
- accepteer de standaard keyboard encoding

2.5.3 Configureren harddisk indeling

- Selecteer de eerste disk;
- selecteer de volledige disk;
- maak de volgende partities aan volgens de hieronder gegeven instructies;
 - / (a, 100M)
 - swap (b, 1G)
 - /var (d, 500M)
 - /usr (e, 5G)
 - /home (f, rest)

In OpenBSD worden partities in een slice aangemaakt met het commando `disklabel(8)`. Hieronder staan de commando's die nodig zijn om partities aan te kunnen maken:

1. maak de partitietabel leeg met het commando ‘z’;
2. maak een partitie aan met het commando ‘a’, bijvoorbeeld ‘a a’ of ‘a b’;
3. accepteer de opgegeven offset;
4. geef de grote van de partitie op, bijvoorbeeld 100M of 1G;

5. accepteer het opgegeven filesystem;
6. geef het mountpoint op, bijvoorbeeld / of /usr;
7. herhaal stap 2 tot en met 6 totdat alle partities aangemaakt zijn;
8. sla de partitie indeling op met het commando 'w';
9. sluit disklabel(8) af met het commando 'q';

Voor meer informatie kan het commando 'h' in disklabel(8) gegeven worden.

- Geef de mount-points nogmaals op en sluit af met "done";
- geef toestemming om de filesystems aan te maken.

2.5.4 Globale systeem configuratie

- Geef de hostname op;
- configureer het netwerk niet, dit komt namelijk aan bod in de volgende opdracht;
- stel het root password in.

2.5.5 Installeren sets

- Geef op dat de sets vanaf de cd-rom geïnstalleerd gaan worden;
- accepteer de standaard padnaam;
- selecteer de volgende sets:
 - bsd
 - baseXX.tgz
 - etcXX.tgz
 - miscXX.tgz
 - compXX.tgz
 - manXX.tgz
 - gameXX.tgz
- type "done" om verder te gaan.

2.5.6 Afronden installatie

- Er moet een ssh server aangezet worden;
- er moet geen X Window System geconfigureerd worden;
- selecteer voor de tijdzone "Europe/Amsterdam".

2.5.7 Starten vanaf harddisk

- Verwijder de cd-rom;
- herstart de computer;
- er draait nu een werkend OpenBSD systeem.

Laat de opdracht door de docent aftekenen.

2.6 Practicum configureren netwerk

In dit practicum wordt het netwerk van de, in de vorige opdracht geïnstalleerde, versie van OpenBSD geconfigureerd. Hiervoor kan in de manpages `myname(5)`, `mygate(5)`, `resolv.conf(5)` en `hostname.if(5)` informatie gevonden worden.

Configureer het netwerk statisch aan de hand van onderstaande gegevens:

- hostnaam: *.hvu.nl;
- ipadres: 145.89.182.*;
- netmask: 255.255.255.0;
- gateway: 145.89.182.1;
- nameserver: 145.89.38.3;
- domein: hvu.nl

Laat de werking, door middel van een ping naar `www.google.com`, zien dat het netwerk werkt.

Om de nodige bestanden aan te kunnen passen kan gebruik worden gemaakt van de teksteditor `vi`. Voor mensen die nog nooit met `vi` hebben gewerkt zit er in de bijlagen een lijst met veel gebruikte commando's.

Hoofdstuk 3

Unix onderhouden

3.1 Inleiding

Wanneer een besturingssysteem wordt uitgebracht is er uiteraard de nodige aandacht besteed aan het testen op fouten, het blijkt echter, dat er toch vaak fouten over het hoofd worden gezien. Deze fouten kunnen in sommige gevallen worden gebruikt om in te breken in het besturingssysteem of om het besturingssysteem te laten crashen.

Dit inbreken kan risico's opleveren voor het bedrijf, aangezien er vitale bedrijfsgegevens of privacy-gevoelige informatie op straat kan komen te liggen. Daarnaast kan het ook zijn dat het bedrijf, door het crashen van het besturingssysteem van een belangrijke server, een periode geen of minder zaken kan doen.

De hierboven genoemde punten maken het dan ook uitermate belangrijk om, wanneer er fouten worden gevonden, het besturingssysteem zo snel mogelijk bij te werken met de patches van de leverancier. Patches zijn in principe niets anders dan kleine wijzigingen aan de software om de fouten te repareren. Er zijn globaal twee soorten patches, namelijk binary patches en source patches. Welke vorm wordt gebruikt verschilt per besturingssysteem.

3.2 Binary patches

Bij deze vorm van patches worden de aanpassingen in binaire vorm verspreid. Dit heeft als voordeel dat de bestanden op de harddisk gewoon kunnen worden vervangen door de aangepaste versies.

Dit kan echter lastig zijn wanneer er een kleine fout wordt gevonden in een erg groot programma, aangezien in dit geval de patch het hele programma moet bevatten. Voorbeelden van besturingssystemen die binary patches gebruiken zijn:

- Debian GNU/Linux

- FreeBSD
- Microsoft Windows
- Red Hat Linux
- Slackware Linux

3.3 Source patches

Bij deze vorm van patches worden de aanpassingen in source vorm verspreid. In de meeste gevallen worden alleen de wijzigingen, die aan de verschillende source bestanden moeten worden gedaan, verspreid. Een voordeel van deze methode is dat de patches over het algemeen erg klein zijn. Daarnaast is het makkelijk om te zien wat er aan de code is aangepast om het lek te dichten.

Een nadeel van deze methode is dat iedere keer dat er een patch is, er een deel van het besturingssysteem opnieuw moet worden gecompileerd. Dit kan op langzame machines veel tijd kosten. Voorbeelden van besturingssystemen die source patches gebruiken zijn:

- Gentoo Linux
- NetBSD
- OpenBSD

3.4 Practicum patchen source tree

Zoals hierboven beschreven vallen beide, voor dit practicum gekozen, besturingssystemen in de groep die de patches in source vorm aanbieden. Om ervaring op te doen met deze manier van werken zal in dit practicum de source tree van OpenBSD gepatched worden.

Opdracht:

Bekijk hoe OpenBSD omgaat met beveiligingslekken en pas de beveiligings updates toe op de geïnstalleerde versie van OpenBSD. Om dit te doen moeten globaal de volgende stappen worden uitgevoerd:

1. Installeer OpenBSD (configureer ook het netwerk).
2. Download, met behulp van het programma `ftp(1)`, de kernel source (`sys.tar.gz`) en de userland source (`src.tar.gz`) van <ftp://ftp.nluug.nl/pub/OpenBSD/<versie>/>.
3. Pak deze bestanden uit in de directory `/usr/src`. Maak hiervoor gebruik van bijvoorbeeld `tar(1)` of `pax(1)`.
4. Download de patches, voor de gebruikte versie van OpenBSD, vanaf <http://www.openbsd.org/errata.html>

5. Patch het systeem aan de hand van de instructies in de patch bestanden. Let er op dat de patches in de juiste volgorde moeten worden toegepast.

Voor een uitgebreidere beschrijving zie de OpenBSD FAQ [19].

Laat de opdracht aftekenen door de docent.

3.5 Practicum updaten source tree

Wanneer de source tree van NetBSD of OpenBSD aangepast moeten worden, dient er rekening mee te worden gehouden dat er van deze besturingssystemen drie versies beschikbaar zijn, namelijk:

- -release: De versie die verkrijgbaar is in binaire vorm op de cd-roms en op de mirrors.
- -stable: De “-release” versie plus beveiligings patches *en* betrouwbaarheids patches.
- -current: De ontwikkelings versie, deze versie zal na goed testen de nieuwe “-release” worden.

Wanneer men een machine installeert, gebruikt men meestal een cd-rom of de binaire sets van één van de mirrors. Deze versie is nagenoeg altijd de “-release” versie van het besturingssysteem. Het is daarom belangrijk dat deze versie zo snel mogelijk wordt bijgewerkt naar de “-stable” versie. De makkelijkste manier om dit te doen is de kernel en userland source vanaf een mirror te downloaden en vervolgens met behulp van cvs(1) bij te werken naar de “-stable” versie.

Opdracht:

Lees de release(8) manpage door en gebruik de daar opgedane kennis om de, in de vorige practicumopdracht geïnstalleerde, source code bij te werken naar de “-stable” versie.

3.6 Practicum kernel compileren

In de bovenstaande onderdelen over het aanpassen van de source code wordt er vooral vanuit gegaan dat de beveiligingslekken zich bevinden in de userland code. Dit zal in veel gevallen ook zo zijn, aangezien de hoeveelheid userland code veel groter is dan de hoeveelheid kernel code.

Mochten er toch beveiligingslekken zitten in de kernel code dan is het uitermate belangrijk dat deze lekken zo snel mogelijk worden gepatched. Dit is nodig aangezien grote delen van de kernel worden uitgevoerd met dezelfde rechten als de gebruiker “root”. Wanneer een inbreker op deze manier het systeem kan binnendringen heeft hij direct volledige controle over het systeem.

Naast dit beveiligingsoogpunt kan het ook erg handig zijn om zelf een aangepaste kernel voor een machine te maken. Op deze manier kan er namelijk voor

gezorgd worden dat er alleen ondersteuning in de kernel zit voor hardware die daadwerkelijk in de machine aanwezig is. Daarnaast kan op deze manier ook een aantal systeeminstellingen geoptimaliseerd worden voor het doel van de machine.

Opdracht:

Lees het onderdeel van de OpenBSD FAQ door waarin wordt besproken hoe een eigen kernel kan worden gemaakt. Maak hierna zelf een aangepaste kernel op basis van de “GENERIC” kernel waarin alle SCSI onderdelen zijn verwijderd.

Vervang vervolgens de huidige kernel door de zelf gemaakte kernel en herstart de machine. Kijk of de machine inderdaad nog werkt met de zelf gemaakte kernel. Is dit het geval laat dan de opdracht door de docent aftekenen.

Hoofdstuk 4

Extra applicaties

4.1 Inleiding

Wanneer een besturingssysteem op een machine wordt geïnstalleerd bevat dit besturingssysteem meestal een beperkte hoeveelheid applicaties. Dit wordt gedaan om te zorgen dat er geen onnodige of niet gebruikte applicaties op de machine aanwezig zijn.

De machine in kwestie zal dus in de meeste gevallen niet genoeg hebben aan het besturingssysteem alleen. Vandaar dat bijna alle besturingssystemen het mogelijk maken om naast het besturingssysteem ook extra applicaties te installeren.

De manier waarop extra applicatie worden geïnstalleerd verschilt echter per besturingssysteem. Globaal zijn er twee groepen te onderscheiden, namelijk de voorgecompileerde, ook wel binaire applicaties genoemd, en de applicaties in source-vorm. Hieronder zal een verdere uitleg worden gegeven van beide groepen.

4.2 Voorgecompileerde applicaties

Bij voorgecompileerde applicaties gaat het meestal om een archief dat bestaat uit de programmacode en een aantal scripts om de configuratie van het programma af te handelen. Wanneer een dergelijk archief geïnstalleerd wordt, worden deze programma bestanden naar de juiste locatie op de hardeschijf gekopieerd. Vervolgens kunnen de scripts gebruikt worden om een aantal instellingen te configureren, hierna is het programma meestal klaar voor gebruik.

Het voordeel van deze methode is dat het een erg snelle methode is. Wanneer de installatie archieven zijn uitgepakt en de configuratie gedaan is, kan het programma meestal direct worden gebruikt. Voor deze methode is dus weinig tijd en processorkracht nodig.

Een ander voordeel van deze methode is dat het voorgecompileerde programma door iedereen kan worden gebruikt, zonder dat deze mensen toegang hoeven te

hebben tot de source-code van het programma. Dit voordeel kan echter ook een nadeel zijn, aangezien het op deze manier mogelijk is, voor de leverancier, om allerlei vreemde code in het programma te stoppen, zonder dat iemand dit kan controleren. Voorbeelden van vreemde code zijn spyware en backdoors.

Voorbeelden van besturingssystemen die gebruik maken van voorgecompileerde applicaties:

- Debian GNU/Linux (deb)
- Mac OS X (hqx)
- Microsoft Windows (exe)
- Red Hat Linux (rpm)
- Slackware Linux (tgz)
- Suse Linux (rpm)

Het is niet helemaal correct om de Linux versies in deze groep te plaatsen, aangezien daar meestal naast de voorgecompileerde applicaties ook de source-code beschikbaar is. Deze Linux versies horen echter meer in deze categorie dan in de andere, omdat de meeste gebruikers de voorgecompileerde applicaties gebruiken in plaats van ze zelf te compileren.

4.3 Applicaties in source-vorm

Bij applicaties in source-vorm wordt, zoals de naam al doet suggereren, de applicatie in source-code vorm aangeboden. De source-code wordt meestal aangeboden in de vorm van een archief. Dit archief moet op de harddisk worden uitgepakt en vervolgens geconfigureerd en gecompileerd. Wanneer dit klaar is staan alle programma bestanden in de directory en zullen ze moeten worden verplaatst naar de juiste locaties op de hardeschijf. Ten slotte zal er eventueel nog wat moeten worden aangepast aan de instellingen voordat het programma kan worden gebruikt.

Een voordeel van deze methode is dat het systeem een grote flexibiliteit met zich mee brengt. Het is bijvoorbeeld mogelijk om een aantal ongebruikte onderdelen van het programma uit te schakelen voordat het wordt gecompileerd, zodat het uiteindelijke uitvoerbare programma kleiner is.

Daarnaast zullen er minder problemen optreden met betrekking tot gedeelde bibliotheken, aangezien er tijdens het compileren dezelfde versies worden gebruikt als tijdens het draaien van het programma.

Een ander voordeel is dat de source-code aanwezig is en dus kan worden aangepast naar de wensen van de gebruiker (mits de gebruiker kan programmeren natuurlijk) en kan worden gecontroleerd op vreemde code.

Een nadeel van deze methode is dat het vaak veel tijd, processorkracht, geheugen en harddisk ruimte kost om een programma te compileren.

Voorbeelden van besturingssystemen die gebruik maken van applicaties in source-vorm:

- FreeBSD (ports)
- Gentoo (portage)
- NetBSD (pkgsrc)
- OpenBSD (ports)

Gebruikers met minder snelle computers zullen niet altijd blij zijn met applicaties in source-vorm, aangezien het veel tijd en processorkracht kan kosten om deze applicaties te compileren. Daarom bieden alle vier de projecten de meest gebruikte applicaties ook in voorgecompileerde vorm aan. Deze voorgecompileerde applicaties zijn over het algemeen in dezelfde vorm als de Slackware Linux tgz's.

Om het installeren van source-vorm applicaties te vergemakkelijken bieden alle vier de projecten een framework aan waarin alle beschikbare applicaties zijn opgenomen. Wanneer een gebruiker een bepaalde applicatie wil installeren zal dit framework de source-code voor de gebruiker ophalen en compileren. Mocht de applicatie in kwestie nog andere applicaties nodig hebben dan zal de source-code van deze applicaties tevens worden opgehaald en gecompileerd.

4.4 Practicum installeren applicaties

In dit practicum zal de student applicaties uit beide groepen gaan installeren. Om dit te kunnen doen moet eerste een besturingssysteem op de computer worden gezet. In dit practicum zal gebruik worden gemaakt van NetBSD. Om het practicum uit te kunnen voeren moet er een werkende netwerk verbinding zijn.

Opdracht 1:

Lees het onderdeel van de NetBSD Guide [12] door dat over de “package collection” gaat (<http://www.netbsd.org/guide/en/chap-pack.html>).

Opdracht 2:

Zoek uit hoe de, in het onderdeel “voorgecompileerde applicaties”, genoemde archiefformaten globaal zijn opgebouwd. Maak een lijst met voor- en nadelen van iedere soort.

Opdracht3 :

Installeer de voorgecompileerde Samba 3 applicatie op het systeem en schrijf op waar alle verschillende onderdelen worden neergezet.

Opdracht 4:

Download de NetBSD pkgsrc van een van de mirrors en installeer deze. Compileer en installeer vervolgens het programma Wget.

Opdracht 5:

Maak zelf een voorgecompileerde applicatie van de applicatie Icecast.

Laat de opdrachten aftekenen door de docent.

Hoofdstuk 5

Webservices

5.1 Inleiding

Met het in hoog tempo toenemen van het aantal mensen met een breedband internet verbinding groeit ook de markt voor webgebaseerde oplossingen. Een groot aantal van deze webgebaseerde oplossingen maken gebruik van een webserver, daarom zal in deze les aandacht worden besteed hoe een webserver op een veilige manier kan worden opgezet.

5.2 HTTP servers

Er is in deze cursus gekozen om gebruik te maken van de Apache HTTP Server [1]. Deze keuze is gemaakt aangezien het een webserver is die op veel platformen te gebruiken is en het is de meest gebruikte webserver op dit moment [15]. In dit practicum zal gebruik worden gemaakt van de 1.3 versie, aangezien deze versie standaard wordt geïnstalleerd met OpenBSD en er een aantal onderdelen zijn aangepast die de veiligheid ten goede komen.

Hieronder zullen een aantal van de belangrijke onderwerpen worden toegelicht. Deze informatie is bedoeld om de gebruiker een beeld te geven van de mogelijkheden, voor de volledige informatie wordt verwezen naar de officiële documentatie van Apache [2].

5.2.1 Aparte user en group

Iedere applicatie die wordt geschreven bevat een aantal fouten. In sommige gevallen kunnen deze fouten leiden tot toegang tot het besturingssysteem. Om de schade van deze toegang te beperken is het verstandig om publiek toegankelijke services niet met de rechten van de user “root”, maar onder een andere user en group te laten draaien.

Wanneer een aanvaller gebruik maakt van een fout in de applicatie en toegang tot het systeem krijgt, zal hij deze toegang krijgen met dezelfde rechten als de applicatie. In het geval van een applicatie die draait met “root” rechten heeft

de aanvaller direct volledige controle over het systeem. Wanneer de applicatie echter onder een andere user en group draait beperkt dit de mogelijkheden van de aanvaller.

5.2.2 Chroot

Met het draaien van services onder een andere user en group kunnen veel problemen worden voorkomen, maar mocht de aanvaller het systeem binnen zijn gekomen, dan kan hij nog steeds bij heel veel gegevens. Om dit te voorkomen is het mogelijk om sommige applicaties op te sluiten in een directory. Dit opsluiten van een applicatie in een directory kan worden uitgevoerd met de systemcall `chroot()`. Vanaf het moment dat deze systemcall is aangeroepen ziet de applicatie de directory waarin hij is opgesloten als zijn “/” (root) directory en kan dus niet meer naar de bovenliggende directories.

Dit opsluiten van een applicatie in een directory heeft als voordeel dat, wanneer een aanvaller toegang heeft gekregen tot het systeem, deze alleen de bestanden in de chrooted directory kan zien en niet de rest van de bestanden op het systeem. Standaard wordt Apache onder OpenBSD gechrooted naar `/var/www`.

Een nadeel van deze methode is dat een aantal extra modules en CGI scripts problemen kunnen opleveren, wanneer Apache is gechrooted. Dit wordt veroorzaakt doordat deze programma's externe programma's of bibliotheken nodig hebben die niet aanwezig zijn in de chrooted directory.

5.2.3 Uitgeklede configuratie

Wanneer Apache wordt geïnstalleerd op een systeem, wordt er ook een standaard configuratie van Apache gedaan. Deze standaard configuratie is vooral gericht op gebruiksvriendelijkheid en is daardoor niet echt veilig. Het is daarom nodig om, voor een publiek toegankelijke webserver, zelf een aangepaste configuratie te maken die zo veilig mogelijk is.

Bij het maken van een aangepaste configuratie is het van belang dat er zo min mogelijk onderdelen en functies worden aangezet. Hieronder zal een uitgeklede configuratie per onderdeel worden besproken.

```
# =====  
# Basic settings  
# =====  
ServerType standalone  
ServerRoot "/var/www"  
PidFile logs/httpd.pid
```

In dit onderdeel worden de basis instellingen van de webserver geconfigureerd. In dit geval gaat het om een “standalone” server die als homedirectory `/var/www` heeft. De “PidFile” bevat de pid van de server en is te vinden in `/var/www/logs/`.

```
# =====
# Performance settings
# =====
Timeout 300
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15
MinSpareServers 5
MaxSpareServers 10
StartServers 10
MaxClients 150
MaxRequestsPerChild 0
```

In dit onderdeel worden een aantal van de opties ten behoeve van de performance van webserver geconfigureerd. De hierboven weergegeven waarden zijn proefondervindelijk vastgesteld op een server (de webserver van de auteur), maar zullen als goede uitgangswaarden voor andere situaties kunnen worden gebruikt.

```
# =====
# Apache's modules
# =====
```

In dit onderdeel kunnen verschillende modules worden geconfigureerd. Om alleen statische contents te kunnen aanbieden zijn geen extra modules nodig. Verdere informatie met betrekking tot modules zal verderop in dit hoofdstuk worden gegeven.

```
# =====
# General settings
# =====
Port 80
User www
Group www
ServerAdmin webmaster@example.com

DocumentRoot "/var/www/html"
UserDir disabled

<IfModule mod_dir.c>
    DirectoryIndex index.html
</IfModule>

AccessFileName .htaccess
<Files .htaccess>
    Order allow,deny
    Deny from all
</Files>

UseCanonicalName Off
```

```
HostnameLookups Off
ServerSignature Off
ServerTokens Prod
```

In dit onderdeel wordt de globale configuratie van de webserver gedaan. In het eerste stuk wordt gedefiniëerd op welke poort de server moet draaien en onder welke user en group. Daarnaast wordt ingesteld wat het emailadres van de beheerder is.

In het tweede deel wordt ingesteld waar de webpagina's te vinden zijn. Daarnaast wordt de optie, die het mogelijk maakt voor lokale gebruikers om webpagina's in hun homedirectory te zetten, uitgeschakeld.

In het derde deel wordt gedefiniëerd welke bestanden moeten worden weergegeven wanneer er naar een directory wordt gevraagd. In dit geval wordt het bestand "index.html" weergegeven wanneer men de directory opvraagt.

In het vierde deel wordt aangegeven wat de naam is van de AccessFiles. Om de snelheid en beveiliging te verbeteren is het gebruik van AccessFiles in deze configuratie uitgeschakeld. Wanneer er namelijk wel van deze optie gebruik wordt gemaakt moet de server in iedere directory gaan zoeken naar een AccessFile.

Tenslotte worden er een aantal opties uitgeschakeld die, onnodig veel, informatie over de webserver aan de buitenwereld kunnen geven. Deze opties zijn in principe niet echt gevaarlijk, maar ze kunnen een aanvaller wel van waardevolle informatie voorzien.

```
# =====
# Access control
# =====
<Directory />
    Options None
    AllowOverride None
    Order deny,allow
    Deny from all
</Directory>
```

Het onderdeel "Access control" is een van de belangrijkste onderdelen van de configuratie, aangezien hiermee kan worden ingesteld waar de clients kunnen komen. Dit onderdeel zal verderop in dit hoofdstuk worden besproken.

```
# =====
# MIME encoding
# =====
<IfModule mod_mime.c>
    TypesConfig conf/mime.types
</IfModule>
<IfModule mod_mime.c>
    AddEncoding x-compress Z
    AddEncoding x-gzip gz
</IfModule>
```

DefaultType text/plain

In dit onderdeel wordt weergegeven waar informatie over de MIME encoding van de webbestanden gevonden kan worden. Daarnaast worden er twee extenties toegevoegd aan de MIME bibliotheek (.Z en .gz). Tenslotte wordt de standaard MIME encoding op “text/plain” gezet.

```
# =====  
# Log settings  
# =====  
LogLevel warn  
  
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""  
    combined  
LogFormat "%h %l %u %t \"%r\" %>s %b" common  
LogFormat "%{Referer}i -> %U" referer  
LogFormat "%{User-agent}i" agent  
  
ErrorLog logs/error_log  
CustomLog logs/access_log common
```

In dit onderdeel worden de configuraties met betrekking tot het loggen ingesteld. Als eerste wordt het logniveau ingesteld door middel van de optie “LogLevel”. Vervolgens wordt er geconfigureerd hoe de te loggen regels er uit moeten zien. Tenslotte worden de namen van de logbestanden geconfigureerd.

```
# =====  
# Browser settings  
# =====  
BrowserMatch "Mozilla/2" nokeepalive  
BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0  
BrowserMatch "RealPlayer 4\.0" force-response-1.0  
BrowserMatch "Java/1\.0" force-response-1.0  
BrowserMatch "JDK/1\.0" force-response-1.0
```

In dit onderdeel kunnen per browsertype een aantal instellingen worden ver-richt. Deze instellingen worden gedaan om te zorgen dat, een groot aantal verschillende browsers van de aangeboden diensten gebruik kunnen maken.

```
# =====  
# VirtualHosts  
# =====  
  
#EOF
```

In dit onderdeel kunnen de virtuele hosts worden opgenomen. Dit onderdeel zal echter verderop in dit hoofdstuk worden besproken.

5.2.4 Modules

Er bestaan zeer veel verschillende modules voor Apache die allemaal extra functionaliteit aan de webserver kunnen toevoegen. Wanneer een webserver alleen maar statisch contents hoeft aan te bieden zijn er in principe geen extra modules nodig. Zorg er voor dat er zo min mogelijk modules worden geladen en zoek uit wat de risico's zijn van de modules die wel worden geladen. Enkele voorbeelden van modules zijn:

- `mod_fastcgi`: FastCGI module voor Apache
- `mod_python`: Python module voor Apache
- `mod_ssl`: SSL (SecureSocketLayer) module voor Apache
- `php4_module`: PHP4 module voor Apache

5.2.5 Access control

Een zeer belangrijk onderdeel van de configuratie is het onderdeel Access control. In dit onderdeel kan worden ingesteld welke onderdelen van het filesysteem toegankelijk zijn voor de clients. In de hierboven genoemde uitgekilde configuratie wordt de toegang tot alles ontzegd. Dit is uiteraard niet echt praktisch aangezien er toch webpagina's door de clients moeten kunnen worden bekeken. Om te zorgen dat de bestanden in de map `/var/www/htdocs` kunnen worden bekeken moet het onderstaande gedeelte worden toegevoegd.

```
<Directory "/htdocs">
    Order deny,allow
    Allow from all
</Directory>
```

5.2.6 Virtuele hosts

Om te zorgen dat er meerdere websites op dezelfde webserver kunnen draaien heeft Apache de mogelijkheid om virtuele hosts aan te maken. Er zijn op dit moment twee manieren om virtuele hosts aan te maken, namelijk ipadres gebaseerd en naam gebaseerd.

Bij ipadres gebaseerde virtuele hosts moet de webserver voor iedere website een uniek ipadres hebben. Dit heeft tot gevolg dat een webserver met veel websites evenveel verschillende ipadressen moet hebben en dus niet echt praktisch is.

Bij naam gebaseerde virtuele hosts wordt er aan de hand van de gevraagde URL gekeken om welke website het gaat. In dit geval kan de webserver volstaan met één ipadres, deze methode wordt het meest toegepast. Om dit te laten werken is een goed geconfigureerde DNS server nodig. De configuratie van een DNS server zal verderop in dit hoofdstuk worden besproken.

Om de op naam gebaseerde virtuele hosts `www.bar.com` en `www.foo.com` te kunnen gebruiken moet het onderstaande fragment worden toegevoegd aan de webserver configuratie.

```
NameVirtualHost *
<VirtualHost *>
    ServerAdmin webmaster@bar.com
    DocumentRoot /var/www/htdocs/bar.com/
    ServerName www.bar.com
</VirtualHost>
<VirtualHost *>
    ServerAdmin webmaster@foo.com
    DocumentRoot /var/www/htdocs/foo.com/
    ServerName www.foo.com
</VirtualHost>
```

5.2.7 PHP

In de configuratie, zoals hierboven besproken, wordt er vanuit gegaan dat er alleen maar statische contents moeten worden aangeboden. Deze statische contents is echter in veel gevallen niet afdoende. In deze gevallen kan er gebruik worden gemaakt van externe programma's die het mogelijk maken contents dynamisch te genereren. Een van deze applicaties is PHP [22].

In deze cursus zal worden behandeld hoe de basis PHP functionaliteit geïntegreerd kan worden in Apache. Naast deze basis zijn er nog veel meer PHP modules beschikbaar, maar deze zullen niet worden besproken.

Om PHP te kunnen gebruiken zullen een aantal stappen moeten worden doorlopen. Ten eerste moet de PHP-core applicatie worden geïnstalleerd. Ten tweede moeten er een aantal wijzigingen aan de Apache configuratie worden gedaan. Tenslotte moet Apache opnieuw worden opgestart. Let op dat Apache echt opnieuw moet worden opgestart, aangezien alleen een “reload” niet goed werkt.

De volgende drie stappen zullen moeten worden uitgevoerd om de nodige aanpassingen aan de Apache configuratie te maken.

Stap 1:

Voeg de PHP-module toe aan de Apache configuratie:

```
# =====
# Apache's modules
# =====
LoadModule php4_module          /usr/local/lib/php/libphp4.so
```

Wanneer de module niet in de hierboven aangegeven directory aanwezig is, kan deze worden gevonden door gebruik te maken van het commando `find`. Voor meer informatie met betrekking tot `find` wordt verwezen naar de manpage van `find`.

Stap 2:

Voeg `index.php` toe aan de “DirectoryIndex”, zodat deze er als volgt uitziet:

```
<IfModule mod_dir.c>
    DirectoryIndex index.html index.php
</IfModule>
```

Stap 3:

Voeg het bestandstype .php toe aan de webserver configuratie:

```
AddType application/x-httpd-php .php
```

5.2.8 HTTPS

In sommige gevallen is het nodig dat de communicatie tussen de client en de webserver onleesbaar is voor andere partijen, een voorbeeld hiervan is het internetbankieren. Wanneer de communicatie onleesbaar moet zijn zal er gebruik moeten worden gemaakt van een vorm van encryptie. Op dit moment wordt voor deze encryptie meestal gebruik gemaakt van de “Secure Socket Layer” (SSL) laag. De combinatie tussen de “Secure Socket Layer” en het HTTP protocol wordt vaak aangeduid met HTTPS.

Hieronder staat het deel dat moet worden toegevoegd aan de Apache configuratie om HTTPS te kunnen gebruiken (Let er wel op dat er geen virtuele hosts zijn geconfigureerd, aangezien deze problemen opleveren in combinatie met een globale configuratie van HTTPS):

```
<IfDefine SSL>
    Listen 80
    Listen 443
    AddType application/x-x509-ca-cert .crt
    AddType application/x-pkcs7-crl .crl
</IfDefine>
<IfModule mod_ssl.c>
    SSLPassPhraseDialog builtin
    SSLSessionCache dbm:logs/ssl_scache
    SSLSessionCacheTimeout 300
    SSLMutex sem
    SSLRandomSeed startup builtin
    SSLRandomSeed connect builtin
    SSLRandomSeed startup file:/dev/arandom 512
    SSLLog logs/ssl_engine_log
    SSLLogLevel info
    SSLEnable
    SSLCertificateFile /etc/ssl/server.crt
    SSLCertificateKeyFile /etc/ssl/private/server.key
</IfModule>
```

Het is echter mogelijk om naam gebaseerde virtuele hosts te combineren met HTTPS al kan er dan maar maximaal één van deze virtuele hosts gebruik maken van HTTPS. Om dit te doen moet het onderstaand fragment worden toegevoegd aan de VirtualHosts sectie. Daarnaast moeten er uit het bovenstaande fragment drie regels worden verwijderd namelijk, SSLEnable, SSLCertificateFile en SSLCertificateKeyFile.

```

<IfDefine SSL>
<VirtualHost *:443>
    SSLEnable
    ServerAdmin webmaster@foo.com
    DocumentRoot /var/www/htdocs/foo.com/secure/
    ServerName secure.foo.com
    SSLCertificateFile /etc/ssl/server.crt
    SSLCertificateKeyFile /etc/ssl/private/server.key
</VirtualHost>
<VirtualHost *>
    redirect 302 / https://secure.foo.com/
    ServerName secure.foo.com
</VirtualHost>
</IfDefine>

```

5.3 DNS servers

Om te zorgen dat de verschillende webservices op het internet en op de lokale netwerken met behulp van namen in plaats van ipadressen te bereiken zijn, is het “Domain Name System” (DNS) ontwikkeld. Dit systeem zorgt er voor dat de ipadressen van webservices gekoppeld worden aan, voor mensen, beter te onthouden namen.

Er is voor gekozen om in deze les gebruik te maken van de DNS server “BIND 9”. Deze keuze is gemaakt aangezien dit de nieuwste versie van de meest voorkomende DNS server is. BIND 9 wordt standaard meegeleverd met OpenBSD.

Een DNS server kan globaal op drie manieren worden ingezet. Ten eerste als een “Authoritative Name Server”. Ten tweede als een “Caching Name Server” Tenslotte als een combinatie van beide opties.

5.3.1 Authoritative Name Server

Een Authoritative Name Server is een server die alle data heeft van de zone waar deze authoritative voor is. Iedere DNS zone heeft dan ook tenminste één Authoritative Name Server nodig. Om netwerk problemen op te kunnen vangen zijn er meestal meerdere Authoritative Name Servers voor één DNS zone.

Binnen deze groep zal er één zijn die de master is en de rest van de groep zijn slave servers. De master server is de enige die de DNS data op de hardeschijf heeft staan. De slave servers halen hun informatie van de master server.

5.3.2 Caching Name Server

Aangezien de meeste applicaties niet in staat zijn om zelf een volledig DNS verzoek te doen bij een van de Authoritative Name Servers wordt dit uitbesteed aan een ander programma. Dit programma wordt ook wel een Recursive Name Server genoemd.

Om de performance van een Recursive Name Server te verbeteren slaan deze servers meestal hun informatie op in een cache. Hieruit is de naam Caching Name Server ontstaan. Een Caching Name Server is dus zelf geen Authoritative Name Server.

5.3.3 Combinatie

Wanneer een organisatie niet groot genoeg is om voor ieder van de hierboven genoemde functies een aparte server neer te zetten is het mogelijk om deze functies te combineren. In dit geval moet er echter wel extra aandacht worden besteed aan de beveiliging.

5.3.4 Beveiliging

Aangezien veel DNS servers publiek toegankelijk zijn, is het nodig deze servers goed te beveiligen. De beveiliging van BIND 9 heeft veel overeenkomsten met de beveiliging van Apache. Ten eerste is het aan te raden om de server te draaien onder een andere user en group dan root. Ten tweede is het mogelijk om de DNS server te chrooten naar een directory. Tenslotte kunnen er access control lists worden gemaakt die de toegang kunnen beperken. Voor meer informatie zie de BIND 9 documentatie [9].

5.3.5 Configuratie

De configuratie bestanden van BIND 9 staan onder OpenBSD in de directory /var/named. Om het configureren van een domein makkelijker te maken staat hieronder een voorbeeld van een domein en de reverse voor dat domein.

```
; hvu.foo domain database (/var/named/master/hvu.foo)
$TTL 6h
@      IN      SOA      server.hvu.foo. admin.hvu.foo. (
                                200403161      ; Serial
                                3600            ; Refresh
                                900            ; Retry
                                3600000       ; Expire
                                3600 )         ; Minimum
      IN      NS       server.hvu.foo.
      IN      MX       10 mail.hvu.foo.

; Addresses
localhost.hvu.foo.  IN      A       127.0.0.1
server.hvu.foo.     IN      A       192.168.0.1
mail.hvu.foo.       IN      A       192.168.0.2
host1.hvu.foo.     IN      A       192.168.0.3
host2.hvu.foo.     IN      A       192.168.0.4
host3.hvu.foo.     IN      A       192.168.0.5
```

```

; hvu.foo domain reverse database (/var/named/master/hvu.foo.rev)
$TTL 6h
@      IN      SOA      server.hvu.foo. admin.hvu.foo. (
                                200403161      ; Serial
                                3600            ; Refresh
                                900             ; Retry
                                3600000        ; Expire
                                3600 )         ; Minimum
0.168.192.in-addr.arpa. IN      NS          server.hvu.foo.

; Addresses
1.0.168.192.in-addr.arpa.  IN      PTR      server.hvu.foo.
2.0.168.192.in-addr.arpa.  IN      PTR      mail.hvu.foo.
3.0.168.192.in-addr.arpa.  IN      PTR      host1.hvu.foo.
4.0.168.192.in-addr.arpa.  IN      PTR      host2.hvu.foo.

```

5.4 Practicum opzetten DNS server

Om dit practicum te kunnen doen moet er een werkende installatie van OpenBSD op de computer staan, het netwerk geconfigureerd zijn en een DNS server draaien. Zie Hoofdstuk 10 van de OpenBSD FAQ [19] over het opstarten van services.

Opdracht:

Zet een Authoritative Name Server op voor een niet bestaand domein. Let er hierbij op dat de DNS server zo veilig mogelijk is opgezet.

5.5 Practicum opzetten webserver

Om dit practicum te kunnen doen moet er een werkende installatie van OpenBSD op de computer staan, het netwerk geconfigureerd zijn, een DNS server geconfigureerd zijn en een webserver draaien.

Opdracht 1:

Maak een veilige webserver configuratie en test deze.

Opdracht 2:

Pas de webserver configuratie zodanig aan dat deze minimaal twee naam gebaseerde virtuele hosts bevat. Deze hostnamen moeten ook aanwezig zijn in de DNS server.

Opdracht 3:

Installeer de PHP-core applicatie en pas de webserver configuratie zodanig aan dat er gebruik kan worden gemaakt van PHP pagina's. Schrijf een kleine PHP webpagina, maak hiervoor gebruik van de documentatie op de PHP website [23].

Opdracht 4:

Zorg dat een van de naam gebaseerde virtuele hosts gebruik kan maken van een beveiligde verbinding op basis van HTTPS. Zie voor meer informatie Hoofdstuk 10 van de OpenBSD FAQ [19].

Laat bovenstaande opdrachten aftekenen door de docent.

Hoofdstuk 6

Fileservices

6.1 Inleiding

Het werken met een computer bestaat voor het overgrote deel uit bewerkingen in en van bestanden. Deze bestanden kunnen op een aantal methoden ter beschikking worden gesteld. Welke methode het meest geschikt is hangt uiteraard af van de toepassing. Hieronder zullen een aantal methoden, om bestanden ter beschikking te stellen, worden belicht. Het gaat hierbij om methoden die het mogelijk maken de bestanden op een centrale plaats op te slaan. Er wordt dus niet gesproken over lokale opslag.

6.2 Anonieme FTP

Om bestanden met iedereen op het internet te kunnen delen wordt er meestal gebruik gemaakt van het “File Transport Protocol”. De anonieme versie van dit protocol zorgt er voor dat iedereen met een FTP-client de bestanden kan downloaden. Het is over het algemeen niet toegestaan bij deze versie om bestanden op de server te zetten.

FTP is een telnet gebaseerd protocol waarbij de communicatie tussen de client en de server als leesbare tekst over het internet wordt verstuurd. Het is daarom ook af te raden om FTP als niet anonieme services aan te bieden, aangezien het password om in te loggen bij deze methode als leesbare tekst wordt verstuurd en dus makkelijk is af te luisteren.

Er zijn twee manieren waarop FTP servers en clients met elkaar kunnen communiceren namelijk, active mode en passive mode. Deze twee manieren hebben allebei hun voor- en nadelen en zullen hierna verder worden toegelicht.

6.2.1 Active mode

Bij deze mode verbindt de client vanaf een random unprivileged poort N (> 1024) met de control poort (21) van de server. Op hetzelfde moment gaat de client luisteren op de poort $N+1$ en geeft dit door aan de server. Vervolgens

opent de server een verbinding vanaf zijn data poort (20) met de client poort N+1. Wanneer deze laatste verbinding tot stand is gekomen is de connectie klaar voor gebruik.

Het voordeel van deze mode is dat er aan de server kant alleen maar verbindingen naar poort 21 moeten worden doorgelaten om alles te kunnen laten werken. Een nadeel van deze mode is dat deze problemen op kan leveren wanneer de client achter een firewall zit, aangezien er een connectie van buiten naar een random poort op de client gemaakt moet worden en dit meestal niet wordt toegestaan door de firewall.

6.2.2 Passive mode

Bij deze mode verbindt de client vanaf een random unprivileged poort N (> 1024) met de control poort (21) van de server. In plaats van het gaan luisteren op de poort N+1 stuurt de client een “PASV” commando naar de server. De server gaat hierop luisteren op een random unprivileged poort N en stuurt dit poortnummer terug naar de client. Vervolgens zet de client een verbinding op vanaf de poort N+1 naar de overeengekomen poort van de server. Wanneer deze laatste verbinding tot stand is gekomen is de connectie klaar voor gebruik.

Het voordeel van deze methode is dat clients achter een firewall met de passive mode wel kunnen connecten met een FTP server, aangezien zij beide verbindingen zelf opzetten. Een nadeel van deze methode is dat de server naast poort 21 ook alle poorten > 1024 open moet hebben staan. De problemen van de firewall worden met deze mode dus eigenlijk verplaatst van de client naar de server.

6.2.3 Beveiliging

Wanneer een FTP server publiek toegankelijk is, moet er voldoende aandacht worden besteed aan de beveiliging van deze server. Zoals al eerder genoemd is het af te raden om de server op een andere manier dan als anonieme FTP server te gebruiken.

Daarnaast zal de FTP gebruiker moeten worden gechooted (zie 5.2.2) naar een speciaal daarvoor bestemd deel van de harddisk. Dit om te voorkomen dat configuratie en password bestanden van de server gedownload kunnen worden.

Ten slotte is het ook verstandig om de gebruikers alleen “read-only” toegang te geven tot de data om ongeautoriseerde wijzigingen te voorkomen. Wanneer het toch mogelijk moet zijn om bestanden te kunnen uploaden, is het verstandig om dit naar een speciaal daarvoor bestemde directory te doen. Bij voorkeur moet er voor gezorgd zijn dat er geen listing, van de inhoud van deze directory, gemaakt kan worden. Dit gebeurt meestal om te voorkomen dat er bestanden uit deze directory kunnen worden gedownload.

De hierboven genoemde manier van het beschikbaar stellen van bestanden is handig om veel mensen toegang te geven tot een aantal publieke bestanden. Het

is echter veel minder geschikt om te worden ingezet als services voor het centraal opslaan van de gebruikersdirectories. De hieronder genoemde oplossingen zijn hier juist op toegespitst.

6.3 Unix filesharing

Om gebruikersdirectories te kunnen delen tussen verschillende Unix machines zijn er een aantal protocollen beschikbaar. Het meest bekende en toegepaste protocol uit deze groep is NFS. Naast dit “Network FileSystem” zijn er inmiddels andere protocollen ontwikkeld, zoals Coda [3] en OpenAFS [17], om een aantal van de limitaties van NFS te overkomen. In deze les zal echter alleen worden ingegaan op NFS.

NFS is oorspronkelijk ontwikkeld door Sun Microsystems en biedt transparante toegang tot bestanden en directories op andere machines. Het protocol is gebaseerd op “Remote Procedure Call” primitieven en daardoor behoorlijk makkelijk te implementeren op verschillende besturingssystemen.

Het protocol is opgebouwd uit een client en een server die via RPC met elkaar communiceren. De server “exporteert” de verschillende directories en de client “mount” deze directories alsof het lokale partities zijn. De gebruiker ziet echter geen verschil tussen lokale partities en directories die via NFS zijn gemount.

NFS heeft als voordelen dat het een erg simpele opzet en configuratie heeft en het op bijna alle Unix platformen beschikbaar is. Een nadeel van NFS is dat het niet echt veilig is, aangezien alles als leesbare tekst over het netwerk gaat. Daarnaast is het niet echt goed schaalbaar en het kan niet goed omgaan met verbroken verbindingen.

6.4 MS Windows filesharing

Om gebruikersdirectories te kunnen delen tussen verschillende Windows machines is het CIFS protocol ontwikkeld. Dit “Common Internet File System” is een doorontwikkelde versie van het “Server Message Block”. Om Windows clients met een Unix fileservers te kunnen laten communiceren is het nodig dat de Unix fileservers dit protocol kan spreken.

Een implementatie van het CIFS protocol voor Unix wordt geleverd door de applicatie SAMBA [25]. Wanneer deze applicatie op de fileservers wordt geïnstalleerd is het mogelijk om onderdelen van het bestandssysteem beschikbaar te maken voor de Windows clients.

De CIFS implementatie van SAMBA kan niet alleen gebruikt worden om bestanden en printers met Windows clients te delen, maar kan ook dienst doen als Primary Domain Controller (PDC). Wanneer SAMBA deze rol heeft, moeten alle gebruikers zich authentifieren bij de SAMBA server. Naast de functionaliteit van PDC is het ook mogelijk om SAMBA in te zetten als een member server.

De SAMBA configuratie bevat een enorme hoeveelheid configuratieopties hetgeen kan leiden tot een zekere onoverzichtelijkheid. Een simpele SAMBA configuratie kan er zo uit zien:

```
[global]
  workgroup      = WORKGROUP
  server string  = Samba %v (%h)
  security       = user
  hosts allow    = 192.168.0. 127.
  load printers  = yes
  printing       = bsd
  socket options = TCP_NODELAY
  os level       = 33

[homes]
  comment        = Home Directories
  browseable     = no
  writable       = yes

[printers]
  comment        = All Printers
  path           = /var/spool/lpd/samba
  browseable     = no
  guest ok       = no
  writable       = no
  printable      = yes
```

Voor verdere informatie en configuratie mogelijkheden wordt verwezen naar de SAMBA documentatie [\[26\]](#).

6.5 Practicum anonieme FTP

Om dit practicum te kunnen doen moet er een werkende versie van NetBSD op de computer worden geïnstalleerd en moet het netwerk worden geconfigureerd.

Opdracht:

Configureer de NetBSD FTPD zodanig dat deze anonieme FTP verbindingen accepteert en dat de anonieme gebruikers worden gecrooted naar `/home/ftp`.

Lees voor dat er begonnen wordt, met het opzetten van de FTP server, eerst de manual pages `ftpd(8)` en `ftpd.conf(5)` deze bevatten belangrijke informatie. Let er op dat de NetBSD FTPD gestart moet worden vanuit `inetd(8)`!

Laat de opdracht aftekenen door de docent.

6.6 Practicum NFS

Om dit practicum te kunnen doen moet er een werkende versie van NetBSD op de computer geïnstalleerd zijn en moet het netwerk geconfigureerd zijn.

Voor deze opdracht kan de, in de vorige opdracht gemaakte, installatie gebruikt worden.

Opdracht:

Configureer een NFS server die de /home directory exporteert naar een aantal machines. Lees voor informatie over hoe dit moet worden opgezet de volgende informatiebronnen:

- `rpcbind(8)`
- `nfsd(8)`
- `exports(5)`
- <http://www.netbsd.org/guide/en/chap-net.html#nfs>
- <http://www.netbsd.org/guide/en/chap-rc.html>

Laat de opdracht aftekenen door de docent.

6.7 Practicum SAMBA

Om dit practicum te kunnen doen moet er een werkende versie van NetBSD op de computer geïnstalleerd zijn en moet het netwerk geconfigureerd zijn. Voor deze opdracht kan de, in de eerste opdracht gemaakte, installatie gebruikt worden.

Opdracht:

Installeer SAMBA 3 en configureer deze zodanig dat Windows clients toegang hebben tot de gebruikersdirectories op deze machine. Zorg er ook voor dat er een publieke directory is waar alle gebruikers kunnen schrijven.

Laat de opdracht aftekenen door de docent.

Hoofdstuk 7

Beveiliging

7.1 Inleiding

De configuratie van een goed beveiligd systeem bestaat uit een meerlaagse beveiliging strategie. Binnen deze strategie vervuld iedere laag zijn eigen taak en zorgen de verschillende lagen er voor dat, wanneer de ene laag faalt, dit door de volgende laag wordt opgevangen.

Het configureren van al deze lagen ten behoeve van de beveiliging van het systeem is een uitgebreid en tamelijk complex proces. De reikwijdte van deze cursus staat het derhalve dan ook niet toe om hier erg diep op in te gaan. Er zullen hieronder echter wel een aantal technieken worden besproken die de beveiliging van iedere machine positief kunnen beïnvloeden.

7.2 Firewall

Een veel gebruikte methode, om toegang tot services op een computer of het lokale netwerk te voorkomen, is het inzetten van een firewall. Een firewall is een applicatie die aan de hand van een gemaakte configuratie bepaalt of een gegeven connectie wel of niet mag worden doorgelaten. Met behulp van een firewall is het bijvoorbeeld mogelijk dat de machines in het lokale netwerk niet toegankelijk zijn vanaf het internet, maar wel omgekeerd.

Er bestaan zeer veel verschillende firewallapplicaties en al deze applicaties hebben hun voor- en nadelen. Om een firewall op een goede manier te configureren is voldoende kennis nodig van de werking van de lokale machine en het lokale netwerk. Wanneer de firewall moet worden ingezet voor complexe toepassing, wordt de configuratie meestal zeer ingewikkeld.

In deze cursus is gekozen om gebruik te maken van de firewall van OpenBSD, genaamd "PF". Deze keuze is gemaakt aangezien PF een firewall is met veel mogelijkheden en een tamelijk eenvoudige configuratie. Deze eigenschappen maken het een goede keuze voor onderwijsdoeleinden. Daarnaast is OpenBSD's PF sinds kort ook beschikbaar in FreeBSD [5] en is het NetBSD project bezig om OpenBSD's PF zondanig aan te passen dat deze ook werkt onder NetBSD.

Ten slotte wordt PF door veel beveiligingsexperts gezien als een van de beste, vrij beschikbare, firewalls van dit moment.

Hieronder zullen een aantal functies, waarover de meeste firewalls beschikken, worden besproken aan de hand van OpenBSD's PF. De termen zijn, voor de consistentie met andere documentatie, in het engels gehouden.

7.2.1 Packet Filtering

Packet Filtering is de meest basale functionaliteit van een firewall. In deze basale uitvoering wordt ieder binnenkomend pakket vergeleken met een set regels en op basis daarvan wordt beslist of het pakket weggegooid moet worden of worden doorgelaten. Een nadeel van deze methode is dat er extra moeite moet worden gedaan om het verschil te zien tussen, verkeer dat een verbinding opzet van buiten naar binnen en verkeer dat als antwoord op verkeer van binnen naar buiten komt. Deze extra moeite zorgt er vaak voor dat de configuratie complexer wordt.

Om dit nadeel te voorkomen is er "Stateful Packet Filtering" ontwikkeld. Bij Stateful Packet Filtering bestaat er onderscheid tussen verkeer dat bij een bestaande verbinding hoort en verkeer dat bij een nieuwe verbinding hoort. Op deze manier is het mogelijk om alleen verkeer van binnen naar buiten toe te laten en verkeer van buiten naar binnen te blokkeren. Het antwoord op het verkeer van binnen naar buiten wordt namelijk tot de bestaande verbinding gerekend en deze is op basis van de regels toegestaan. Deze manier van werken maakt het configureren van de firewall eenvoudiger en duidelijker.

OpenBSD's PF is een voorbeeld van een Stateful Packet Filtering Firewall. Andere voorbeelden van firewalls die stateful verbindingen ondersteunen zijn Netfilter [16] (standaard in Linux vanaf 2.4) en IPFilter [24].

Een voorbeeld van een simpele firewall configuratie voor OpenBSD's PF kan er zo uit zien:

```
ExtIF="xl0"      # Externe netwerkinterface
IntIF="fxp0"     # Interne netwerkinterface

set loginterface $ExtIF  # Log hoeveelheid verkeer

# Blokkeer al het verkeer tenzij anders is geconfigureerd
block all

# Laat al het verkeer over de loopback interface door
pass quick on lo0

# Laat al het uitgaande verkeer op de externe interface door
pass out quick on $ExtIF all keep state

# Laat verkeer van het lokale netwerk door
pass in quick on $IntIf any keep state
```

De hierboven getoonde configuratie laat verkeer door dat, vanaf de firewall naar buiten gaat, verkeer dat vanaf het lokale netwerk naar de firewall gaat en verkeer over de loopback interface. Er wordt echter geen verkeer doorgelaten vanaf buiten naar binnen en vanaf de firewall naar het lokale netwerk, tenzij het gaat om antwoord verkeer van verbindingen die wel zijn toegestaan.

7.2.2 Network Address Translation

Wanneer een organisatie of persoon minder ipadressen krijgt dan dat er machines zijn, die verbinding moeten hebben met het internet, is het mogelijk om door middel van Network Address Translation een heel netwerk achter één of meer externe ipadressen te koppelen. Dit heeft als voordeel dat deze organisatie of persoon intern gebruik kan maken van één van de, in RFC 1918 beschreven, gereserveerde netwerkblokken.

Deze blokken komen uit de volgende netwerkranges:

- 10.0.0.0/8
- 172.16.0.0/12
- 192.168.0.0/16

Een voorbeeld van een simpele NAT configuratie voor OpenBSD's PF kan er als volgt uit zien:

```
ExtIF="xl0"  
IntIF="fxp0"
```

```
# Network Address Translation  
nat on $ExtIF from $IntIF:network to any -> $ExtIF
```

Deze configuratie zorgt er voor dat de ipadressen van de machines in het lokale netwerk, zodra zij verbinding maken met het internet, worden vervangen door het externe ipadres van de firewall. Hierdoor kan er binnen het lokale netwerk gebruik gemaakt worden van *niet* routeerbare ipadres blokken. Om dit te laten werken is het wel nodig om het doorsturen van ip pakketten in de kernel van de firewall / gateway aan te zetten (IP forwarding).

7.2.3 Port Forwarding

Wanneer er Network Address Translation wordt gebruikt, om machines in het lokale netwerk te laten verbinden met het internet, kan dit problemen opleveren wanneer een aantal van deze machines services aanbieden die bereikbaar moeten zijn vanaf het internet. Dit probleem kan worden opgelost door gebruik te maken van Port Forwarding.

Port Forwarding maakt het mogelijk om een poort of een groep poorten, op de Network Address Translation Gateway, door te laten wijzen naar een poort of een groep poorten op een machine in het lokale netwerk.

Zo is het bijvoorbeeld mogelijk om op de firewall (NAT Gateway) poort 80 (HTTP) door te laten wijzen naar een machine in het lokale netwerk waarop een webserver draait. Vanaf het internet gezien lijkt het alsof de Firewall ook webserver is, ondanks dat deze service door een andere machine draait. Deze techniek wordt vaak toegepast om een zogenaamde “DeMilitarized Zone” (DMZ) op te zetten voor de publieke services van de organisatie.

Een demilitarized zone is een netwerk dat publiek toegankelijke services aanbiedt aan gebruikers op het internet, maar wel beschermd wordt door een firewall. Dit netwerk van services wordt apart gehouden van het lokale netwerk om te voorkomen dat derden toegang kunnen krijgen tot het lokale netwerk.

Een voorbeeld van een simpele Port Forwarding in OpenBSD’s PF kan er als volgt uit zien:

```
ExtIF="xl0"  
HTTPD="10.0.0.100"
```

```
# Port Forwarding  
rdr pass on $ExtIF proto tcp from any to any port http -> $HTTPD
```

Deze configuratie zorgt er voor dat verbindingen naar poort 80 (http), op de firewall, worden doorgestuurd naar poort 80 op de machine met ipadres 10.0.0.100. Uiteraard is het ook mogelijk om een verbinding naar een andere poort door de sturen, bijvoorbeeld poort 80 naar poort 8080. Let er bij Port Forwarding op dat de terugkerende pakketten van de server, waarna geforward wordt, via de firewall lopen.

7.2.4 Packet Normalization

Wanneer netwerkpakketten binnenkomen bij de firewall kan er van alles mis zijn met deze pakketten. De pakketten kunnen bijvoorbeeld zijn gefragmenteerd of op andere wijze corrupt zijn. Dit corrupt zijn kan opzettelijk zijn, om bijvoorbeeld de firewall te omzeilen, of per ongeluk door problemen met het netwerk.

Wanneer de netwerkpakketten niet in orde zijn wordt het lastig om deze pakketten aan de regels van de firewall te toetsen. Om te voorkomen dat sommige opzettelijk verminkte pakketten worden doorgelaten bieden een aantal firewalls de mogelijkheid om de pakketten eerst in orde te maken en dan pas te filteren. Deze optie wordt over het algemeen Packet Normalization genoemd. OpenBSD’s PF bevat een dergelijke optie en het wordt aangeraden om, uit beveiligings oogpunt, deze optie te gebruiken.

Een voorbeeld van Packet Normalization in OpenBSD’s PF kan er als volgt uit zien:

```
# Packet Normalization  
scrub all
```

Met deze configuratie worden alle netwerkpakketten, die de machine in- of uitgaan op iedere netwerkkaart, gecontroleerd en zonodig in orde gemaakt. Dit is echter niet altijd de bedoeling en het is daarom mogelijk om aan te geven in welke richting en op welke netwerkkaart er moet worden gecontroleerd en genormaliseerd.

Hierboven is een klein aantal van de mogelijkheden van OpenBSD's PF aan de orde gekomen. Van deze besproken mogelijkheden is ook maar een zeer beperkt deel van de opties aan bod gekomen.

De auteur heeft echter bewust voor deze insteek gekozen, aangezien er uitstekende documentatie bestaat over OpenBSD's PF en het zou zonde zijn om deze documentatie hier te kopiëren. Voor uitgebreidere informatie wordt de lezer derhalve verwezen naar de OpenBSD's PF Guide [21] en de manual `pf.conf(5)`.

7.3 Host hardening

Behalve het installeren van een firewall is het verstandig om een machine die toegankelijk is vanaf een publiek netwerk, zoals het internet, te “harden” tegen aanvallen vanaf dat netwerk. Dit harden van de machine kan hele extreme vormen aannemen, maar in deze cursus zullen alleen een aantal tips gegeven worden die, op eenvoudige wijze, de beveiliging ten goede kunnen komen.

7.3.1 Toegangsbeperking

Zorg er voor dat er geen onnodige gebruikersaccounts aanwezig zijn op de machine. Ieder aanwezig account is, in principe, een potentiële beveiligingslek, aangezien het mogelijk kan zijn, voor een inbreker, om achter het password te komen en dus toegang tot het systeem te krijgen. Daarnaast wordt de controle, welke gebruiker wat heeft gedaan, een stuk lastiger wanneer er meer accounts zijn.

7.3.2 Uitschakelen ongebruikte services

Zorg er voor dat er op de machine geen services draaien die niet nodig zijn op de machine. Iedere draaiende service is in principe een potentiële beveiligingslek, aangezien het mogelijk is dat er voor die service een exploit is ontdekt, maar dat deze nog niet bekend is gemaakt of nog niet is verholpen.

7.3.3 Verwijder ongebruikte applicaties

Het is aan te raden om alle applicaties en bibliotheken, die voor het goed functioneren van de machine niet nodig zijn, te verwijderen. Dit lijkt misschien vreemd, maar door deze applicaties te verwijderen, wordt het een stuk lastiger voor de inbreker, om gebruik te maken van het systeem dat deze net tot de beschikking heeft gekregen. Een voorbeeld van applicaties die niet thuis horen op een productie machine zijn compilers.

7.3.4 Bestandssysteem permissies

Stel de bestandssysteem permissies zodanig in dat deze zo restrictief mogelijk zijn, zonder daarbij het goed functioneren van het systeem te belemmeren. In sommige gevallen is het mogelijk om bepaalde partities volledig readonly te maken, hetgeen een positieve uitwerking kan hebben op de beveiliging. Door alles zo restrictief mogelijk te configureren wordt het, bij een inbraak, de inbreker een stuk moeilijker gemaakt om zijn werk te doen.

7.3.5 Kernel aanpassen

Maak een kernel voor de machine, waar alleen de onderdelen in zitten die noodzakelijk zijn voor het goed functioneren van de machine. Het verwijderen van kernel onderdelen die niet gebruikt worden heeft twee voordelen. Ten eerste wordt de gebruikte geheugenruimte kleiner en ten tweede wordt de kans dat er een fout zit in de kernel kleiner, aangezien er minder in de kernel zit. Ten slotte zou er ook nog opgemerkt kunnen worden dat, het weghalen van devices die niet in de machine aanwezig zijn, de stabiliteit van het besturingssysteem ten goede kunnen komen. Hierover zijn de meningen echter verdeeld.

Over het algemeen komt het er bij host hardening op neer dat, alleen de onderdelen en opties die nodig zijn voor het goed functioneren van de machine, aanwezig moeten zijn op de machine.

7.4 Policies

Naast de bovengenoemde technische oplossingen is het, bij het beveiligen van een machine, ook zeer belangrijk dat er goede afspraken bestaan over *hoe* er met de machine moet worden omgegaan. Deze afspraken worden over het algemeen policies genoemd. Er zijn een groot aantal policies te bedenken, maar in deze cursus zal alleen aandacht worden besteed aan een aantal algemene policies.

7.4.1 Gebruikspolicies

Het is verstandig om, met de verschillende beheerders van de machine, overeen te komen dat de machine alleen mag worden gebruikt voor de taken waar deze voor bedoeld is. Deze afspraak voorkomt bijvoorbeeld dat iemand, op de firewall van het bedrijf, een SETI@home¹ client gaat draaien.

Daarnaast moet er ook worden afgesproken dat de beheerders geen, niet door de leidinggevende geautoriseerde, wijzigingen aan de machines mogen doen. Deze afspraak voorkomt dat er wijzigingen plaats vinden waar verder niemand van op de hoogte is.

¹<http://setiathome.ssl.berkeley.edu/>

7.4.2 Root account

Eén van de grootste problemen van gezamenlijk beheer van een machine, is het gebruik van het root account. Wanneer meerdere personen de beschikking hebben over dit account, is het onmogelijk om te controleren wie welke wijzigingen heeft gedaan. Deze vorm van anoniemiteit kan leiden tot grote problemen.

Dit probleem kan gedeeltelijk worden opgelost door een goede indeling van users en groups te maken, maar helaas werkt dit niet in alle gevallen. Om toch te zorgen dat iedereen de taken kan uitvoeren zonder dat het root account nodig is, is de applicatie `sudo(8)` ontwikkeld.

Deze applicatie stelt de gebruikers, die hier toegang toe hebben, in staat om root rechten te krijgen zonder daarvoor het root password te moeten weten. Een bijkomend voordeel van de applicatie `sudo(8)` is, dat er heel precies mee kan worden gedefiniëerd welke personen welke acties kunnen doen. Het is dus heel verstandig om een policy in te voeren die voorkomt dat gebruikers rechtstreeks toegang hebben tot het root account.

7.5 Practicum beveiligen OpenBSD

Om dit practicum te kunnen doen moet er een werkende versie van OpenBSD op de computer geïnstalleerd worden en moet het netwerk geconfigureerd zijn.

Opdracht 1:

Harden de OpenBSD installatie aan de hand van de hierboven genoemde mogelijkheden.

Opdracht 2:

Configureer een firewall die alle verkeer van binnen naar buiten toelaat en het verkeer van buiten naar binnen op poort 22 en 80 toelaat. Let er bij deze laatste groep op dat deze connecties worden beschermd tegen “syn floods”. Zorg er voor dat, verbindingen naar poort 25 (smtp), worden doorgestuurd naar een andere machine. Lees voor informatie de genoemde documentatie.

Opdracht 3:

Maak een beheerdersaccount aan en test de werking van `sudo(8)`. Configureer vervolgens dit account zodanig dat de beheerder, zonder een password te hoeven typen, alleen het commando `shutdown(8)` kan uitvoeren.

Laat de opdracht aftekenen door de docent.

Hoofdstuk 8

Overige onderwerpen

8.1 Backups

8.1.1 Inleiding

Het maken van backups is een zeer belangrijk onderdeel van het werk van een systeembeheerder. Het goed bijhouden van backups kost tijd en geld, maar het kan veel narigheid voorkomen. Daarnaast is het binnen een aantal bedrijfstakken verplicht om een goede backup strategie te hebben.

Er zijn heel veel verschillende oplossingen om het maken van backups te vergemakkelijken. Deze oplossingen variëren van het op een floppy zetten van bestanden tot het gebruik van een dedicated netwerkbackup server. Binnen deze cursus zal alleen aandacht worden besteed aan systemen die backups van de lokale harddisk maken.

8.1.2 Utilities

In veel gevallen worden de backup applicaties, door de systeembeheerder, zelf samengesteld uit een aantal standaard Unix utilities. Door deze utilities op een slimme manier te combineren is het mogelijk om zeer geavanceerde backup applicaties te maken. Hieronder zullen een aantal van de meest gebruikte utilities, op dit gebied, worden besproken.

bzip2

De utility `bzip2(1)` is een applicatie die bestanden kan comprimeren. Het comprimeeralgoritme van `bzip2` is zodanig dat het hoge compressieratio's kan halen, maar het is een tamelijk traag algoritme.

date

De utility `date(1)` is een applicatie die het mogelijk maakt om de datum en tijd, van de machine, op te vragen en eventueel aan te passen. Door bepaalde

opmaak opties mee te geven is het bijvoorbeeld mogelijk om de dag van de week op te vragen.

dump en restore

De utilities `dump(8)` en `restore(8)` behoren tot de oudste applicaties voor het maken en terugzetten van backups. Met behulp van de applicatie `dump(8)` is het mogelijk om een kopie van de volledige partitie naar een bestand te maken. Deze kopie kan vervolgens met de applicatie `restore(8)` weer worden teruggezet. Met deze applicaties is het ook mogelijk om incrementele backups te maken.

find

De utility `find(1)` is een applicatie die het mogelijk maakt om bestanden en directories te vinden. Door middel van een groot aantal opties is het mogelijk om criteria aan deze bestanden en directories te stellen.

Find wordt, in de context van backup applicaties, vooral gebruikt voor het samenstellen van lijsten van bestanden en directories die moeten worden geback-uped. Find kan, in combinatie met het commando `touch`, gebruikt worden om incrementale backups mogelijk te maken.

gpg

De utility `gpg(1)` maakt het mogelijk om bestanden te versleutelen en te ontsleutelen. Dit versleutelen en ontsleutelen kan zowel met symmetrische als met asymmetrische encryptie algoritmes. Wanneer er gebruik wordt gemaakt van asymmetrische encryptie algoritmes is het ook mogelijk om bestanden te ondertekenen.

De asymmetrische encryptie mogelijkheden van `gpg` zijn erg handig om toe te passen in scripts, aangezien deze het mogelijk maken om bestanden te versleutelen zonder tussenkomst van de gebruiker.

gzip

De utility `gzip(1)` is een applicatie die bestanden kan comprimeren. Het com-primeeralgoritme van `gzip` haalt minder hoge compressieratio's dan `bzip2`, maar is over het algemeen een stuk sneller dan `bzip2`. Deze applicatie wordt vaak gebruikt in combinatie met `tar` of `pax`.

md5

De utility `md5(1)` is een applicatie die een controlegetal berekent van een ander bestand. Door van de backuparchieven controlegetallen te bewaren kan er, naderhand, altijd worden gecontroleerd of een archief nog intact is.

pax

De utility `pax(1)` is gemaakt om verschillende archieven te kunnen lezen en schrijven. Pax kan gebruikt worden om complete directory structuren te back-uppen naar een archief. Het voordeel van pax is dat het verschillende archief formaten ondersteund waaronder `cpio`, `tar` en `ustar`.

scp

De utility `scp(1)` is een onderdeel van de `ssh` suite en bedoelt om, via een versleutelde verbinding, bestanden van de ene machine naar de andere machine te kopiëren. Scp ondersteund ook `publickey` authenticatie, waardoor het mogelijk wordt om bestanden te verplaatsen van de ene naar de andere machine zonder dat er een password moet worden ingegeven.

Scp kan gebruikt worden om via een veilig kanaal de backups van de machines naar een centrale server te kopiëren.

tar

De utility `tar(1)` is gemaakt om complete directory structuren te back-uppen naar een archief of naar een tape. Dit commando kan tevens gebruikt worden om de archieven uit te pakken.

touch

De utility `touch(1)` is gemaakt om de verschillende datums van een bestand of directory te kunnen aanpassen. Door dit commando te gebruiken in combinatie met `find` is het mogelijk om incrementele backups met bijvoorbeeld `tar` te maken.

8.1.3 Voorbeeld backupscript

Hieronder staat een voorbeeld van een simpel backup script dat de `/etc`, `/home` en `/root` directories naar verschillende archieven backuppert, Vervolgens worden deze archiven gecomprimeerd met `gzip`. De archieven worden opgeslagen in een directory met als naam de eerste drie letters van de week.

```
#!/bin/sh
#
# Simple backupscript

SHELL=/bin/sh
PATH=/bin:/sbin:/usr/bin:/usr/sbin

DAY_OF_WEEK='date +%a'
DEST_DIR=/mnt/backup/$DAY_OF_WEEK
LOG_FILE=$DEST_DIR/backup.log
```

```

# Mount backup disk
mount /dev/wd1h /mnt/backup > /dev/null 2>&1

# Remove old backuparchives
rm $DEST_DIR/*
echo "Starting Backup on: " `date` > $LOG_FILE 2>&1

# Backup /etc
cd /
tar -cpf $DEST_DIR/etc.tar etc >> $LOG_FILE 2>&1

# Backup /home
cd /
tar -cpf $DEST_DIR/home.tar home >> $LOG_FILE 2>&1

# Backup /root
cd /
tar -cpf $DEST_DIR/root.tar root >> $LOG_FILE 2>&1

# Compress tar archives.
cd $DEST_DIR
for x in `ls *.tar`
do
    /usr/bin/gzip ${x}
done

# Finish backup
echo "Finished Backup on: " `date` >> $LOG_FILE 2>&1

# Umount filesystem
cd /
umount /mnt/backup >/dev/null

# EOF

```

8.1.4 Backup media

Bij het maken van een backuppolicy is het van belang om na te denken op wat voor media deze backups opgeslagen gaan worden. Welke opslagmedia gebruikt kunnen worden hangt van een aantal factoren af, zoals de benodigde opslagcapaciteit, de benodigde betrouwbaarheid, de vereiste bewaarduur en het budget.

De volgende media worden geregeld gebruikt voor backups (deze lijst is geen uitputtende lijst):

- floppy
- usb memorystick
- cd-rom

- cd-rw
- dvd+/-r
- dvd+/-rw
- ide harddisk
- scsi harddisk
- tape streamer
- off-site networkstorage

Welke media, van de lijst hierboven, het meest geschikt is voor het opslaan van backups, is erg afhankelijk van de eisen die er worden gesteld. Er is daarom ook geen universele oplossing.

8.2 Systrace

Wanneer een binaire applicatie wordt gedraaid is het soms interessant om te kunnen verifiëren welke systemcalls, aanroepen vanuit de applicatie naar de kernel, dit programma doet. Aan de hand van deze systemcalls kan er namelijk makkelijker worden gecontroleerd of een applicatie vreemde "dingen" doet op het systeem.

Om dit verifiëren van systemcalls makkelijker te maken is de applicatie systrace(1) ontwikkeld. Deze applicatie is in staat om, van een bepaalde applicatie, alle systeemcalls af te vangen en te monitoren.

Naast deze monitor mogelijkheden is het ook mogelijk om, aan de hand van een configuratiebestand, in te stellen welke systemcalls een bepaalde applicatie mag doen. Met deze functionaliteit is het mogelijk om applicaties te beveiligen of te monitoren tijdens het draaien.

Wanneer deze laatste configuratie mogelijkheid wordt toegepast op de inlog shells van de gebruikers, is het mogelijk om heel precies in te stellen wat deze gebruikers wel en niet kunnen. Op deze manier kan er bijvoorbeeld een shell-server gemaakt worden waarbij de gebruikers maar een paar applicaties kunnen gebruiken. In het practicum zal een soortgelijke oplossing voor sftp moeten worden geconfigureerd.

Systrace is op dit moment beschikbaar in NetBSD en OpenBSD, daarnaast zijn er een aantal Linux distributies die het als optie aanbieden. Om systrace goed te kunnen inzetten is het vaak nodig om zelf een configuratie te maken. Dit configureren vergt echter wel diepe kennis van de werking van de applicaties en het besturingssysteem. Het blijft dan ook een beveiligingsmechanisme dat alleen in te zetten is door goed opgeleide systeembeheerders.

8.3 Practicum maken encryptedbackup script

Om dit practicum te kunnen doen moet er een werkende versie van OpenBSD op de computer geïnstalleerd worden en moet het netwerk geconfigureerd worden.

Opdracht:

Maak een backupscript dat één keer per week een volledige backup maakt en de rest van de week incrementele backups. Dit script moet de backups versleutelen en vervolgens uploaden naar een andere machine via scp. Het script moet de volgende directories backuppen:

- /etc
- /home
- /root
- /var/log

Laat de opdracht aftekenen door de docent.

8.4 Practicum opzetten systraced sftp shell

Om dit practicum te kunnen doen moet er een werkende versie van OpenBSD op de computer geïnstalleerd worden, inclusief compiler! Ook moet het netwerk geconfigureerd worden. Voor dit practicum kan de installatie uit de vorige opdracht gebruikt worden.

Opdracht 1:

Maak een systraced sftp shell. Download hiervoor het archief vanaf http://www.gtd5.net/public/projects/systrace.sftp_jail.tar.gz en volg de instructies in de README.

Opdracht 2:

Maak een gebruiker aan en geef hem als shell het net gecompileerde programma. Test of de gebruiker, met sftp, kan inloggen en bestanden in zijn homedirectory kan plaatsen.

Laat de opdrachten aftekenen door de docent.

Hoofdstuk 9

Conclusie

Aan de hand van de opgedane kennis, tijdens deze en voorgaande Unix cursussen, zou de systeembeheerder in spé zich moeten kunnen redden in de meeste Unix omgevingen. Aangezien iedere Unix omgeving weer anders is, is het belangrijk dat de systeembeheerder in spé in staat is zich, door middel van het lezen van de systeemdokumentatie, aan te passen. Dit vermogen is getracht aan te leren door gebruik te maken van verschillende besturingssystemen tijdens de verschillende practica.

Natuurlijk zijn lang niet alle mogelijke onderwerpen aan bod gekomen tijdens deze cursussen. Dit is echter ook onmogelijk, daarom is er voor gekozen om de meest voorkomende onderwerpen te behandelen. Daarnaast is er van de behandelde onderwerpen telkens maar één voorbeeld gegeven en dit voorbeeld hoeft natuurlijk niet de beste oplossing voor iedere situatie te zijn.

Ondanks deze beperkingen denkt de auteur dat, de gegeven oplossingen, makkelijk moeten kunnen worden aangepast aan de specifieke eisen van de toekomstige omgevingen. Voor de systeembeheerders die zich verder willen ontwikkelen, op het gebied van Unix, zouden er voldoende aanknopingspunten gegeven moeten zijn in dit cursusmateriaal.

Ten slotte wil de auteur nog meegeven dat de beste manier, om ervaring op te doen met Unix, het geregeld gebruiken van een Unix besturingssysteem is.

Bijlage A

Cheatsheet vi

De teksteditor vi(1) is één van de oudste schermgebaseerde teksteditoren voor Unix. Helaas is het ook één van de minder gebruiksvriendelijke editors die er op dit moment beschikbaar zijn maar, doordat deze editor al zo lang bestaat, is deze op nagenoeg alle Unix besturingssystemen aanwezig. Dit laatste maakt het handig, voor een systeembeheerder, om toch met deze editor over weg te kunnen.

Een van de grootste problemen van vi is dat het twee verschillende modussen heeft, namelijk een “command mode” en een “insert mode”. Deze modussen kunnen voor veel verwarring zorgen tijdens het gebruik van de editor.

In de “command mode” is het mogelijk om de cursor te verplaatsen en allerhande commando’s op de tekst uit te voeren, zoals het verwijderen, zoeken en vervangen van tekst. Ook het opslaan en afsluiten van de editor gebeurt in deze mode. Om in “command mode” te komen maakt men gebruik van de toets ESC. Na het opstarten staat vi altijd in de “command mode”.

In de “insert mode” is het mogelijk om daadwerkelijk tekst te typen. In sommige versies van vi is het ook mogelijk om in deze mode te navigeren met behulp van de pijltjes-toetsen. Deze functionaliteit is echter niet in alle versies aanwezig. Om in “insert mode” te komen moet de letter i worden ingetyperd.

Hieronder zullen de meest gebruikte vi commando’s worden uitgelegd. Deze commando’s moeten altijd in “command mode” worden uitgevoerd.

Open, save en quit

vi <i>filename</i>	Open <i>filename</i> met vi
:w	Save file (write)
:q	Quit vi (geen wijzigingen gemaakt)
:q!	Quit vi (zonder de wijzigingen op te slaan)
:wq	Save file en quit vi

Navigatie toetsen

h	Verplaats de cursor naar links
j	Verplaats de cursor naar beneden

Navigatie toetsen

k	Verplaats de cursor naar boven
l	Verplaats de cursor naar rechts
w	Verplaats de cursor naar het volgende woord
e	Verplaats de cursor naar het einde van het woord
\$	Verplaats de cursor naar het einde van de regel
^	Verplaats de cursor naar het begin van de regel

Tekst invoegen

i	Voeg tekst voor de cursor toe (insert)
I	Voeg tekst voor de regel toe
a	Voeg tekst achter de cursor toe (append)
A	Voeg tekst na de regel toe

Tekst wijzigen

x	Verwijder karakter rechts van de cursor
<i>nx</i>	Verwijder <i>n</i> karakters rechts van de cursor
X	Verwijder karakter links van de cursor
<i>nX</i>	Verwijder <i>n</i> karakters links van de cursor
D	Verwijder alles tot aan het einde van de regel (delete)
dd	Verwijder huidige regel
<i>ndd</i>	Verwijder <i>n</i> regels vanaf de huidige regel

Kopieëren en plakken

yy	Kopieër de huidige regel naar de buffer (yank)
<i>nyy</i>	Kopieër <i>n</i> regels naar de buffer
p	Plak de inhoud van de buffer achter de cursor (paste)
P	Plak de inhoud van de buffer voor de cursor

Bijlage B

Versies

Dit cursusmateriaal is geschreven in de opmaaktaal \LaTeX . Het document is opgebouwd uit een aantal bronbestanden waarvan hieronder de versies zijn aangegeven.

Bestand	Versie	Laatste wijziging	
<code>cursusmateriaal.tex</code>	1.14	2004-06-15	07:56
<code>voorwoord.tex</code>	1.13	2004-06-15	07:56
<code>inleiding.tex</code>	1.12	2004-04-21	13:52
<code>unixsystemen.tex</code>	1.25	2004-11-25	18:07
<code>onderhoud.tex</code>	1.13	2004-06-15	07:56
<code>applicaties.tex</code>	1.14	2004-11-25	18:07
<code>webservices.tex</code>	1.18	2004-06-15	07:56
<code>fileservices.tex</code>	1.17	2004-11-09	11:29
<code>beveiliging.tex</code>	1.22	2004-12-30	12:36
<code>overig.tex</code>	1.14	2004-06-15	07:56
<code>conclusie.tex</code>	1.9	2004-06-15	07:56
<code>visheet.tex</code>	1.10	2004-06-15	07:56
<code>bronnen.tex</code>	1.12	2004-04-05	13:10

Bibliografie

- [1] Apache Project, *The Apache HTTP Server Project*, maart 2004.
<http://httpd.apache.org/>
- [2] Apache Project, *Apache HTTP Server Documentation*, maart 2004.
<http://httpd.apache.org/docs/>
- [3] Carnegie Mellon University, *Coda File System Website*, maart 2004.
<http://www.coda.cs.cmu.edu/>
- [4] R. Elling, B. Andeweg, J. de Jong en C. Swankhuisen, *Rapportage techniek*, Wolters Noordhoff, 2000.
- [5] FreeBSD Project, *FreeBSD website*, maart 2004.
<http://www.freebsd.org>
- [6] FreeBSD Project, *Hypertext Man Pages*, maart 2004.
<http://www.freebsd.org/cgi/man.cgi>
- [7] GNU Project, *The GNU Privacy Guard website*, maart 2004.
<http://www.gnupg.org/>
- [8] Internet Systems Consortium, *BIND website*, maart 2004.
<http://www.isc.org/index.pl?/sw/bind/>
- [9] Internet Software Consortium, *BIND 9 Administrator Reference Manual*, maart 2004.
<http://www.fifi.org/doc/bind9-doc/arm/Bv9ARM.html>
- [10] T. Lammle, *CCNA: Cisco Certified Network Associate, Study Guide*, Sybex, 2003.
- [11] NetBSD Project, *The NetBSD website*, maart 2004.
<http://www.netbsd.org/>
- [12] NetBSD Project, *FAQs and HOWTOs*, maart 2004.
<http://www.netbsd.org/guide/en/>
- [13] NetBSD Project, *NetBSD Documentation*, maart 2004.
<http://www.netbsd.org/Documentation/>
- [14] NetBSD Project, *NetBSD Manual Pages*, maart 2004.
<http://netbsd.gw.com/cgi-bin/man-cgi>

- [15] Netcraft, *Web Server Survey*, maart 2004.
http://news.netcraft.com/archives/web_server_survey.html
- [16] Netfilter Project, *Netfilter website*, maart 2004.
<http://www.netfilter.org/>
- [17] OpenAFS Project, *OpenAFS website*, maart 2004.
<http://www.openafs.org/>
- [18] OpenBSD Project, *The OpenBSD website*, maart 2004.
<http://www.openbsd.org/>
- [19] OpenBSD Project, *Documentation and FAQs*, maart 2004.
<http://www.openbsd.org/faq/>
- [20] OpenBSD Project, *OpenBSD Manual Pages*, maart 2004.
<http://www.openbsd.org/cgi-bin/man.cgi>
- [21] OpenBSD Project, *PF: The OpenBSD Packet Filter*, maart 2004.
<http://www.openbsd.org/faq/pf/>
- [22] PHP Project, *PHP website*, maart 2004.
<http://www.php.net/>
- [23] PHP Project, *PHP documentatie*, maart 2004.
<http://www.php.net/manual/en/>
- [24] Darren Reed, *IP Filter website*, maart 2004.
<http://coombs.anu.edu.au/ipfilter/>
- [25] SAMBA Project, *Samba website*, maart 2004.
<http://www.samba.org/>
- [26] SAMBA Project, *Samba Documentatie*, maart 2004.
<http://ftp.easynet.be/samba/docs/>