Leerboek Linux deel 3

Sander van Vugt

Inleiding Open Source Versie Leerboek Linux deel 3

Dit boek wordt u aangeboden door Sander van Vugt als Open Source boek. U mag het lezen, afdrukken en verspreiden naar wens. Er zijn echter een paar beperkingen waar u rekening mee moet houden. Het gaat hier om het volgende:

* U mag dit boek alleen gratis verspreiden en er onder geen omstandigheden geld voor vragen. Deze beperking geldt ook nadrukkelijk voor gebruik van dit boek in syllabus vorm. U mag het dus wel drukken, u mag het alleen niet verkopen. Wel gratis weggeven. Uitgever SDU is de enige partij die het recht heeft dit boek in gedrukte vorm door te verkopen. Niets weerhoudt u ervan de gedrukte versie van dit boek te kopen via de boekhandel.
* De auteur van dit boek blijft de enige rechthebbende op het boek. Alle wijzigingen die u aan wilt brengen, moet u teruggeven naar de auteur Sander van Vugt. U kunt hem bereiken op sander.van.vugt@xs4all.nl. Ik sta open voor alle correcties, maar blijf per gesuggereerde aanpassing de enige die zeggenschap heeft over het al dan niet doorvoeren van de aanpassing.
* De versie die hier aangeboden wordt kent een paar kleine beperkingen. Als eerste gaat het hier voorlopig om ruwe tekst. Dit betekent dat er typefouten en meer in voorkomen. Als ik tijd heb haal ik die wellicht er ooit nog uit, maar tijd is momenteel een schaars goed. Ook zijn er geen afbeeldingen in het document opgenomen. Dit is om ervoor te zorgen dat het boek zo licht mogelijk blijft.
* Voor zover niet tegenstrijdig met de bepalingen die in het bovenstaande zijn opgenomen, zijn de bepalingen uit de Open Publication License van toepassing. Bij strijdigheden met de bepalingen uit het voorgaande, prevaleren de voorgaande bepalingen.

Voorwoord

Voor u ligt Leerboek Linux deel 3. Zoals de titel aangeeft, is dit het derde boek in een reeks leerboeken over Linux. Nu kan het zo zijn dat u op het moment dat u dit leest deel 2 nog niet gezien heeft. Dat klopt. Vanwege de grote vraag naar een goed leerboek Linux waarin netwerk onderwerpen behandeld worden, hebben wij ervoor gekozen eerst deel 3 uit te brengen, dan een bijgewerkte versie van deel 1 en tot slot deel 2. Hopelijk zal leerboek Linux deel 2 uiterlijk in het tweede kwartaal van 2006 beschikbaar zijn. U kunt echter als tijdelijke vervanger van deel 2 gebruik maken van Leerboek Linux deel II, geavanceerd systeembeheer zoals uitgegeven in 2001. In dit boek wordt een redelijk aantal onderwerpen behandeld dat ook in het nieuwe deel 2 aan de orde zal komen, houdt toch het nieuwe deel 2 in de gaten want het zal veel uitgebreider zijn en het zal meer zijn gericht op het LPI 201 examen dat nodig is voor het LPI 2 certificaat..

Deze drie leerboeken zijn geschreven met LPI-certificering in het achterhoofd. LPI is momenteel namelijk de enige zinnige certificering die er op het gebied van Linux bestaat. De inhoud van dit boek, sluit aan op de eisen van LPIC 202. Dit boek is echter geen LPI-studieboek. Dit zit als volgt: in een LPI studieboek, staan de specificaties van het LPI examen centraal. De auteur van dit boek heeft gemeend dat de specificaties van LPI 202 weliswaar nuttig zijn, maar niet altijd even praktisch. Waarom zou u bijvoorbeeld in een boek over netwerken geen informatie opnemen over de configuratie van de Samba-server? Om die reden vindt u in dit boek meer dan wat u moet weten om LPI 202 succesvol af te leggen, het kan ook zijn dat op het LPI examen naar details gevraagd wordt die in dit boek niet aan de orde zijn gekomen. De wegen van het LPI zijn soms ondoorgrondelijk en het is vrijwel onmogelijk een boek te schrijven dat 100% dekt van alle vragen die op het LPI examen gesteld worden. Wij hebben echter wel ons best gedaan het dekkingspercentage zo hoog mogelijk te houden. In ieder geval zo hoog dat u het examen gewoon haalt wanneer u dit boek bestudeerd hebt en zijn inhoud begrepen hebt. Buiten voor te bereiden op LPI 202, willen wij u vooral ook een praktisch boek bieden waar u ook wat aan hebt als u het LPI examen niet af gaat leggen. In appendici bij dit boek zijn een proefexamen opgenomen alsmede de laatste examenspecificaties.

Dit boek is geschreven als leerboek. Dat betekent dat het boek uitstekend gebruikt kan worden als cursusboek. Daarnaast echter is dit boek ook geschreven als naslagwerk, wij zijn van mening dat het boek ook een nuttige aanvulling moet zijn voor beheerders van Linux systemen in het Nederlandse taalgebied. Om het boek bruikbaar te maken voor onderwijssituaties, is aan het eind van elk onderwerp een opdracht opgenomen. Deze opdracht stelt u in staat om gericht het geleerde in praktijk te brengen. Wij geloven daarbij niet in het type opdracht waar stap voor stap beschreven staat welk commando u moet typen of op welke knop u moet drukken. Om de opdracht uit te kunnen voeren, mag u zelf op zoek naar de commando's die ingevoerd moeten worden, opdrachten waar deze informatie standaard al gegeven wordt zijn niet leerzaam omdat ze u niet helpen het geleerde te verwerken maar slechts vragen om in te voeren wat u leest. In de opdrachten in dit boek wordt waar mogelijk de procedure uiteengezet, de specifieke invullingen van deze procedures mag u zelf doen. Er zijn daarom ook geen goede antwoorden en er is daarom ook geen uitwerking van de praktijkopdrachten opgenomen, alles wat nodig is om de praktijkopdrachten uit te kunnen voeren, vindt u in de tekst van de hoofdstukken zelf. Het goede antwoord op een praktijkopdracht is een werkende service. Mocht blijken dat bepaalde opdrachten onwerkbaar zijn, dan kunt u mij dat laten weten. Doe dit bij voorkeur per mail, het mailadres dat u hiervoor kunt gebruiken vindt u op mijn website www.sandervanvugt.nl. Als dat nodig blijkt

te zijn, zal ik op deze website ook errata of andere aanvullingen op het boek presenteren. Als er dus iets is waar u niet tevreden mee bent (of juist wel), laat het mij weten, dan doe ik daar wat aan. Zo wil ik er voor zorgen dat u uiteindelijk toch nog tevreden bent met dit boek.

Als laatste onderwijskundig element zijn aan het eind van elk hoofdstuk een aantal herhalingsvragen opgenomen. Deze vragen helpen u te onthouden wat er in het hoofdstuk besproken is. Deze vragen zijn ongeveer even vervelend als de vragen die gesteld worden op het LPI 202 examen, antwoorden op al deze vragen vindt u op mijnwebsite.

Rest mij niets dan u veel plezier te wensen met het werken met dit boek!

Roosendaal, lente 2005,

Sander van Vugt
www.sandervanvugt.nl

Hoofdstuk 1:Configuratie van de netwerkkaart

Inleiding
De configuratie van elke server begint met de configuratie van de netwerkkaart. In dit hoofdstuk leest u hoe u hiervoor te werk gaat. Om te beginnen wordt uw kennis van TCP/IP wat opgefrist aangezien dit het belangrijkste protocol is dat binnen een Linux omgeving gebruikt wordt. Vervolgens leert u hoe u een netwerkkaart configureert voor gebruik van een vast IP-adres. Daarna leest u hoe u zelf een PPP-interface aan kunt maken. Vervolgens leert u hoe u de configuratie af kunt maken door ook routing en DNS name resolving te regelen. Tot besluit van dit hoofdstuk maakt u kennis met een paar handige opdrachten die gebruikt kunnen worden om te testen of de netwerkverbinding naar behoren werkt.

Leerdoelen
* Basiskennis TCP/IP
* Configuratie van een vaste netwerkkaart
* Configuratie van een PPP-interface
* Routing en name resolving
* Testen of het werkt.

1.1 Basiskennis TCP/IP

Om op een netwerk te kunnen communiceren, zijn protocollen nodig. In de loop der jaren zijn er aardig wat verschillende netwerkprotocollen ontworpen, slechts één daarvan is vandaag de dag nog relevant en dat is TCP/IP. Wanneer we het hebben over TCP/IP, hebben we het over een hele verzameling van protocollen die er samen voor zorgen dat netwerkfunctionaliteit beschikbaar is. Zo is er bijvoorbeeld IP dat ervoor zorgt dat elke netwerkkaart voorzien wordt van een uniek IP-adres, maar ook NNTP dat gebruikt kan worden om nieuwsberichten te versturen op een netwerk. Om een netwerkkaart te kunnen configureren, hebt u kennis nodig van de wijze waarop IP-adressen gebruikt worden. Voordat we bespreken hoe u de netwerkkaart zelf in kunt richten, zetten we eerst de meest relevante feiten over dit protocol uiteen.

1.1.1 Werken met IP-adressen

Het Internet Protocol (IP) is in 1974 ontworpen om ervoor te zorgen dat alle computers op de wereld voorzien kunnen worden van een uniek IP-adres. In die tijd had niemand het idee dat internet zou uitgroeien tot het gigantische netwerk dat het vandaag de dag is! Het vervelende gevolg daarvan, is dat bij de specificatie van IP nooit rekening gehouden is met de aantallen computers die momenteel verbonden zijn aan internet en allemaal hun eigen IP-adres nodig hebben met als gevolg dat de IP-adressen die er zijn op dit moment bijna op zijn. Hiervoor zijn echter wel weer oplossingen beschikbaar zoals Network Address Translation (NAT) waarbij een heel netwerk met één geregistreerd IP-adres het netwerk op kan en IP versie 6 waarmee vele miljarden apparaten voorzien kunnen worden van een IP-adres. Van deze oplossingen is NAT heel populair en wordt IP versie 6 momenteel nog niet op grote schaal toegepast.

Bij het ontwerp van het Internet Protocol (IP) is rekening gehouden met een aantal zaken.

- Elke computer op de wereld moet over een uniek adres beschikken.
- Groepen computers die op hetzelfde fysieke netwerk voorkomen, kunnen voorzien worden van een logisch netwerk adres.

- Communicatie tussen computers op verschillende netwerken kan tot stand gebracht worden. Deze functionaliteit wordt routering genoemd.

Een van de belangrijkste zaken die door IP geregeld wordt, is adressering van computers. Voordat je kunt communiceren, moet u immers eerst op basis van een adres contact tot stand kunnen brengen. Om IP-adressen op de juiste wijze toe te kunnen passen, is het van belang dat u weet hoe een IP-adres is opgebouwd..

1.1.2 Opbouw van een IP-adres

Een IP-adres bestaat uit vier bytes. Elk van deze bytes heeft een waarde die minimaal gelijk is aan 0 en maximaal 255. Het kleinst mogelijke getal dat u met vier bytes kunt vormen is dus 0.0.0.0; het grootst mogelijke getal dat u ermee kunt maken is 255.255.255.255 en in totaal zijn er met deze vier bytes ruim vier miljard combinatiemogelijkheden te maken. Voor de liefhebbers: 4294967296 om precies te zijn. Dat lijkt voldoende om elke computer op de wereld van een adres te voorzien toch? Toch is dat niet het geval.

IP adressen bestaan uit twee delen. Als eerste wordt het gebruikt om het netwerk waar een computer op voorkomt te adresseren, daarnaast wordt het gebruikt om de computer – of beter "node", want er zijn meer dingen dan computers alleen die een adres kunnen hebben – te voorzien van een adres. U kunt de adressen dus niet alleen gebruiken om computers te adresseren, u moet er ook alle netwerken mee van een adres voorzien. Het lastige daarbij, is dat de informatie over het te gebruiken netwerk en de te gebruiken node voorkomen in één hetzelfde IP-adres. Ergens is er dus een systeem nodig om onderscheid te maken tussen het netwerkdeel en het node-deel in het IP-adres 172.16.11.12.

Wat nu precies het netwerk- en wat het node-gedeelte is van een adres, wordt bepaald door het subnetmasker dat bij het IP-adres gebruikt wordt. Om het eenvoudig te maken, is er een afspraak gemaakt waardoor snel inzichtelijk wordt welk deel van een IP-adres verwijst naar het netwerk en welk deel verwijst naar de node. IP-adressen worden namelijk verdeeld in klassen en elke klasse heeft zijn eigen standaard subnetmasker. Daarbij wordt altijd het eerste deel van het adres gebruikt voor adressering van het netwerk en het laatste deel voor adressering van nodes. Om te bepalen wat nu waar voor gebruikt wordt, moet u kijken naar het eerste byte. In de volgende tabel wordt een overzicht gegeven.

Eerste byte	**Netwerkbytes**	**Klasse adres**	**Subnetmasker**
0-127	1e	A	255.0.0.0
128-191	Eerste twee	B	255.255.0.0
192-223	Eerste drie	C	255.255.255.0
224-255	NVT	D en E	NVT

Om u er alvast aan te laten wennen, zijn in de tabel gelijk ook maar de subnetmaskers meegegeven. Achter adressen die beginnen met 224 en hoger staat de aanduiding NVT; u kunt deze namelijk niet gebruiken om een computer van een adres te voorzien. Daarnaast hebben we er bij gezet om welke klasse adressen het gaat. Op basis van het getal waarmee een adres begint, wordt het namelijk in een bepaalde klasse onder gebracht; alle adressen in een adresklasse hebben met elkaar gemeen dat ze dezelfde bytes gebruiken voor adressering van netwerken dan wel nodes.

Wellicht vraagt u zich af hoe deze adresklassen nu in de praktijk gebruikt worden? Welnu, dat is eenvoudig. Op het moment dat u met een of ander programma bijvoorbeeld het IP-adres

10.0.0.1 instelt op een netwerkkaart, wordt op basis van de standaard klasse-indeling van IP-adressen bepaald dat dit een klasse A-adres is. Om die reden wordt het IP-adres automatisch voorzien van het subnetmasker 255.0.0.0. Dat is handig, want zo hoeft u hier als beheerder niet verder over na te denken. Aan de andere kant blijft het mogelijk om van de standaardafspraak af te wijken: als u het nodig vindt in plaats van het standaard subnetmasker bijvoorbeeld het subnetmasker 255.255.255.0 te gebruiken bij het IP-adres 10.0.0.1, dan is dat geen probleem en geeft u dat gewoon softwarematig op.

Volgens de standaard afspraak komen dus de computers met de klasse A-adressen 21.128.3.254 en 21.1.87.199 allebei voor in hetzelfde netwerk, namelijk het netwerk 21. De eerste byte begint met een getal tussen 0 en 127, alleen de eerste byte wordt dus gebruikt voor adressering van netwerken. Het is trouwens gebruikelijk dat dit netwerk 21 genoteerd wordt als 21.0.0.0.Als u praat over een netwerk als geheel, zet u de node bytes op nul. Kunt u nu zelf vertellen op welke netwerken de computers met adressen 212.128.67.3 en 212.34.199.56 voorkomen?

1.1.3 De noodzaak te registreren

Als u met IP-adressen aan het werk gaat, is het belangrijk te weten welke adressen u voor het netwerk kunt gebruiken. Om deze vraag te beantwoorden, moet u weten of de computer direct aan internet verbonden is en dus voorzien moet worden van een uniek adres, of dat de computer niet direct aan internet verbonden is en dus voorzien kan worden van een willekeurig adres.

In het eerste geval waarin computers direct aan internet verbonden worden, moet u ervoor zorgen dat elke computer een uniek adres heeft. Neem contact op met uw internetaanbieder, deze kan ervoor zorgen dat u een reeks adressen ter beschikking krijgt waarmee u aan het werk kunt.

Als computers niet direct aan internet verbonden worden, kunt u gebruikmaken van een willekeurig adres. Het is echter niet verstandig om computers zo maar te voorzien van een lukraak gekozen adres. Stel u maar eens voor dat de computers later wel aan internet verbonden worden, u moet dan alle adressen opnieuw uitdelen. Daarnaast is het bijzonder vervelend wanneer u een adres gebruikt dat door een server op internet ook al in gebruik is. Hoe zou immers het verschil bepaald kunnen worden tussen uw server en de server op internet? Om computers op een privé-netwerk van een adres te voorzien, is een aantal adressen gereserveerd. U kunt deze adressen naar hartelust gebruiken, ze worden op internet namelijk niet verder gerouteerd. Het gaat om de volgende adressen:

- Het netwerk 10.0.0.0
- De netwerken 172.16.0.0 tot en met 172.31.0.0
- De netwerken 192.168.0.0 tot en met 192.168.255.0

1.1.4 Speciale adressen

Naast de hierboven genoemde adressen, is er nog een aantal adressen dat voor speciale doelen gereserveerd is:

- 0.0.0.0. Dit adres verwijst naar de standaardgateway. Dit is de computer die u nodig hebt om te communiceren met andere netwerken.
- Het netwerk 127.0.0.0. Dit adres wordt gebruikt voor loopback. Dit is een mechanisme dat op een computer gebruikt wordt zodat verschillende processen op een computer

met elkaar kunnen communiceren. Daarnaast is de computer zelf altijd bereikbaar op het loopback adres 127.0.0.1

- De adressen 224.0.0.0 tot en met 239.255.255.255. Deze adressen zijn gereserveerd voor multicast. U kunt deze multicast-adressen beschouwen als groepsadressen. Met name speciale apparaten op het netwerk, zoals routers, maken er gebruik van om met elkaar te kunnen communiceren.
- De adressen 240.0.0.0 en hoger. Deze adressen zijn gereserveerd voor experimentele doeleinden zoals IP versie 6.
- Het adres 255.255.255.255. Dit is het zogenaamde “local broadcast” adres; het wordt gebruikt om alle computer op dit netwerk te adresseren.
- Adressen waarbij het host-gedeelte de waarde 255 heeft, zoals 11.255.255.255 en 192.168.1.255. Dit zijn zogenaamde “directed broadcast” adressen. Ze worden gebruikt om alle computers op het betreffende netwerk te adresseren.

1.1.5 Genoeg adressen

In de loop der tijd zijn er aardig wat computers aan het Internet gekoppeld en al deze computers hebben een uniek adres nodig. Daarnaast zijn hele reeksen adressen gewoon niet beschikbaar omdat ze al in eigendom zijn van een bedrijf. Dit heeft er toe geleid dat er niet zo heel veel adressen meer over zijn en dat leidt soms tot problemen. Het komt bijvoorbeeld voor dat een bedrijf een klasse B adres wil om een paar duizend computers te adresseren, maar dat zo’n klasse B adres gewoon niet meer beschikbaar is. Er zijn dus andere oplossingen nodig.

Naast computers die allemaal een uniek adres willen, komt nog eens dat er een hele generatie apparaten op de markt gebracht wordt dat ook voorzien moet worden van een uniek IP-adres. Denk hierbij bijvoorbeeld aan mobiele telefoons, maar ook aan bijvoorbeeld koelkasten waarvan over een internetverbinding bijgehouden wordt of er nog wel voldoende blikjes cola in staan. Zoals u begrijpt, maken deze ontwikkelingen het tekort alleen nog maar nijpender.

Om de problemen die wegens het tekort aan IP-adressen te verwachten zijn te voorkomen, is een aantal jaar geleden begonnen met de ontwikkeling van een geheel nieuwe, nog robuustere versie van dit protocol. In de eerste ontwikkelfase werd het IpnG; wat staat voor IP next generation genoemd, tegenwoordig staat het bekend als IP versie 6. Een van de belangrijkste vernieuwingen van IP versie 6 bestaat er uit dat niet langer 32, maar 128 bits gebruikt worden voor adressering. Afgerond levert dat $3{,}4^{e}37$, ofwel 34 gevolgd door 37 nullen op. Ruim voldoende om elke vierkante meter aardoppervlak meerdere adressen te geven!

In de zomer van 1999 is de eerste Internet provider in Japan begonnen deze adressen uit te delen. Het zal echter nog wel een tijdje duren voordat het hele internet over is, het is namelijk een gigantische klus om alle computers van een IP versie 6 adres te voorzien.

1.1.6 Subnetten in detail

Nog steeds veel te veel problemen op netwerken worden veroorzaakt doordat computers op een verkeerde wijze van IP-adressen voorzien worden, ondanks dat dit steeds vaker geautomatiseerd wordt door gebruik te maken van een DHCP-server. Een belangrijk aandeel in deze fout wordt genomen door het verkeerd gebruik van subnetmaskers. Om u te helpen dergelijke problemen te voorkomen, vindt u in deze paragraaf een overzicht van de achtergrondkennis die nodig is om op een succesvolle wijze netwerkadressen uit te delen en te gebruiken.

Tip! Wilt u het LPI examen gaan doen? Zorg dan dat u kunt toveren met subnetmaskers. Er komen in het LPI-examen namelijk nogal wat vragen over dit onderwerp voor.

Als netwerkbeheerder zorgt u ervoor dat alle computers op uw netwerk elkaar kunnen bereiken. Hiervoor moet gebruikgemaakt worden van een of andere vorm van adressering. Als meerdere netwerken met elkaar verbonden worden, moet uit deze adressering minimaal het volgende blijken:

- Op welk netwerk komt een computer voor?
- Wat is het unieke adres van de computer op dat netwerk?

Zoals u in het voorgaande hebt gelezen, wordt voor dit doel gebruikgemaakt van subnetmaskers. Om subnetmaskers goed te kunnen begrijpen, moet u weten hoe een subnetmasker is opgebouwd uit verschillende bytes die op hun beurt weer opgebouwd zijn uit bits. We gaan dus eerst gezellig een potje binair rekenen.

Een IP-adres bestaat uit vier bytes; een byte bestaat uit acht bits. Een bit kan twee waarden hebben. Vergelijk het maar met een schakelaar, hij staat aan of uit. Binair gezien heeft hij de waarde 1 of de waarde 0. Als u niet weet wat binair is, dat is het getalstelsel waarin elk getal twee waarden, namelijk 0 of 1 kan hebben. Zo is decimaal het getalstelsel waarin elk getal 10 waarden, namelijk 0 tot 9 kan hebben. Als u twee bits met elkaar combineert, kunnen deze twee bits gezamenlijk vier verschillende waarden vormen (2 tot de macht 2 om het wiskundig te zeggen). Dit zijn de volgende binaire waarden

00
01
10
11

U kan deze vier binaire waarden ook decimaal opschrijven, de waarden worden dan 0, 1, 2 en 3. Bij elk volgende bit dat wordt toegevoegd, wordt het aantal waarden dat erdoor gerepresenteerd wordt verdubbeld; zo leveren drie bits (2 x 2 x 2) de volgende waarden:

000 (0)
001 (1)
010 (2)
011 (3)
100 (4)
101 (5)
110 (6)
111 (7)

Voor het gemak hebben we er gelijk de decimale waarde maar achter gezet. Het eerste binaire getal uit dit rijtje heeft de decimale waarde 0. Dat zal iedereen volkomen duidelijk zijn, drie keer nul is immers nog steeds nul. Vervolgens heeft het tweede binaire getal de decimale waarde 1. Zie het zo: als het achterste bit aan staat (gelijk is aan 1), levert dat de decimale waarde 1 op. Zo levert het op-een-na achterste bit een decimale waarde 2 op als het aan staat; het uit twee bits bestaande binaire getal 10 is immers gelijk aan het decimale getal 2. Binair gezien is 1+1 dus inderdaad gelijk aan 3.

Om nu te bepalen welke decimale waarde gerepresenteerd wordt door een binair getal, is het zaak dat u weet welke decimale waarde een bit heeft als het aan staat. De regel daarbij is dat het laatste bit in een getal een waarde 1 heeft als het aanstaat en elk bit daarvoor het dubbele van die waarde. Dat klinkt cryptisch, laten we het daarom eens in een overzichtje plaatsen door de waarden van alle bits uit een byte te geven:

1	1	1	1	1	1	1	1
128	64	32	16	8	4	2	1

In de bovenstaande tabel hebben we voor het gemak elk bit even “aan” gezet, de waarde van elk bit moet dus geteld worden om de decimale waarde te kunnen bepalen. Als u nu een binair getal ziet, hou het gewoon even naast deze tabel en tel alle waarden die horen bij de bits die aan staan bij elkaar op. In de volgende tabel ziet u hoe dit op een willekeurig binair getal toegepast kan worden:

1	0	1	1	1	0	0	1
128	-	32	16	8	-	-	1

Voor het gemak zijn in deze tabel alleen de waarden weergegeven van de bits die aan staan. De bits die uit staan zijn niet relevant en hoeven we dus niet mee te tellen. Om nu te achterhalen wat de decimale waarde is van dit getal, tellen we al deze waarden bij elkaar op. In dit geval dus 128 + 32 + 16 + 8 + 1. Dat maakt een totaal van 185.

Subnetmaskers

Zoals gezegd, in een IP-adres wordt op een flexibele getal bepaald welk deel van het adres gebruikt wordt om het netwerk te adresseren en welk deel gebruikt wordt om computers, ook wel “hosts” of “nodes” genoemd, te adresseren. De factor die uiteindelijk bepaalt wat nu waar voor gebruikt wordt, is het subnetmasker. Dit is een getal dat net als een IP-adres uit vier bytes bestaat. Elk bit dat in het IP-adres gebruikt wordt voor de adressering van het netwerk, heeft de binaire waarde “1” in het subnetmasker; elk bit dat in een subnetmasker gebruikt wordt voor de adressering van hosts, heeft de waarde “0” in het subnetmasker. Als dus in het IP-adres 150.100.19.9 de eerste twee bytes gebruikt worden om het netwerk te adresseren en de laatste twee bytes dus gebruikt worden om nodes te adresseren, wordt het subnetmasker dat hier bij hoort dus 255.255.0.0.

Om het leven wat eenvoudiger te maken, is ooit besloten dat elk IP-adres een standaard subnetmasker krijgt. Dit zijn de afspraken waarin de verschillende adresklassen gedefinieerd worden. U hebt hier al over gelezen, maar we zetten het nog even op een rij. Deze keer proberen we de standaard adresklassen ook binair te verklaren.

- Als het eerste bit van het eerste byte van een IP-adres uit staat, dus de waarde 0 heeft, wordt alleen het eerste byte van het IP-adres gebruikt om netwerken te adresseren, de rest van het adres wordt gebruikt om nodes te adresseren. Het subnetmasker wordt dan dus 255.0.0.0. We noemen dit een “klasse A”-adres.
- Als het eerste bit van het eerste byte van een IP-adres de waarde 1 heeft en de tweede bit van het eerste byte heeft de waarde 0, worden de eerste twee bytes van het adres gebruikt om netwerken te adresseren. Het subnetmasker wordt dan dus 255.255.0.0 en we noemen het een klasse B-adres.
- Als de eerste twee bits van het eerste byte van een adres de waarde 1 hebben en het derde bit heeft de waarde 0, worden de eerste drie bytes van het adres gebruikt om

netwerken te adresseren. Het subnetmasker wordt dan dus 255.255.255.0 en we noemen het een klasse C-adres.

- Alle andere adressen hebben een speciale betekenis, ze kunnen niet gebruikt worden om computers mee te adresseren.

Laten we voor alle duidelijkheid het bovenstaande nog even weergeven in een tabel:

Binaire waarde van het eerste byte (x=maakt niet uit)	**Decimale waarde van het eerste byte (min-max)**	**Bijbehorend subnetmask**	**Klasse**
0xxxxxxx	0-127	255.0.0.0	A
10xxxxxx	128-191	255.255.0.0	B
110xxxxx	192-223	255.255.255.0	C

Uit het bovenstaande blijkt, dat er dus een beperkt aantal netwerken is waarin heel erg veel nodes geadresseerd kunnen worden, namelijk de klasse A-netwerken. Hierin zijn immers standaard drie bytes gereserveerd voor de adressering van nodes, er kunnen dus 256 x 256 x 256 –2 = 16777214 computers geadresseerd worden (we zullen het later over de reden van de "-2" hebben. Verder is er een wat groter aantal netwerken waarin iets minder computers geadresseerd kunnen worden, namelijk de klasse B adressen. Tot slot is er een behoorlijk aantal netwerken waarin maar 254 computers geadresseerd kunnen worden.

Subnetten en geregistreerde IP-adressen
Omdat IP-adressen schaars zijn, is het soms noodzakelijk dat u er op een slimme manier gebruik van maakt. Om dit te doen, kan het handig zijn het standaard subnetmasker, dat bij elk IP-adres gegeven wordt, aan te passen. Stel u bijvoorbeeld voor dat u een klasse B-adres gekregen hebt, bijvoorbeeld 150.100.0.0 , maar dat je daarmee 20 netwerken moet adresseren waar in elk netwerk 30 computers voorkomen. U hebt dan het probleem dat elk netwerk een uniek adres moet hebben. Onoplosbaar? Absoluut niet!

Met één klasse B-adres kunt u in totaal 65534 computers adresseren. Het is echter fysiek onmogelijk om zoveel computers op één netwerk te hebben. Dat hoeft ook niet, want u hebt 20 netwerken met 30 computers per netwerk. Hoe u dat oplost? Gebruik in dit geval ook de derde byte om netwerken te adresseren. U houdt dan nog de hele vierde byte over om computers te adresseren. Het komt er hier dus op neer dat u afwijkt van de standaard afspraak voor een klasse C-subnetmasker. U krijgt dan dus de volgende netwerken met bijbehorende subnetmaskers (we noemen alleen de eersten, het mag duidelijk zijn hoe de rest er uit ziet).

150.100.1.0 (255.255.255.0)
150.100.2.0 (255.255.255.0)
150.100.3.0 (255.255.255.0)
150.100.4.0 (255.255.255.0)

Maar wat is hier nu het voordeel van? Neem bijvoorbeeld twee computers: één heeft het adres 150.100.1.1, de andere heeft het adres 150.100.2.2. Als deze computers allebei het subnetmasker 255.255.0.0 gebruiken, komen de computers voor in hetzelfde netwerk; namelijk het netwerk 150.100.0.0. (Voor alle duidelijkheid; als alle node-bits in een IP-adres de waarde 0 hebben, betekent dit dat we het over een netwerk adres hebben.) Als we nu dan het subnetmasker van deze computers gaan wijzigen in 255.255.255.0, horen deze computers ineens thuis in verschillende netwerken. Het aardige is dat het daarbij niet uitmaakt of de

computers inderdaad op verschillende fysieke netwerken voorkomen of niet, ze zullen elkaar gewoon niet meer kunnen vinden omdat ze volgens het subnetmasker voorkomen in verschillende netwerken. De enige manier waarop ze nog met elkaar kunnen communiceren, is als er een router tussen hangt die de pakketjes van het ene netwerkadres overzet naar het andere netwerkadres.

//NOOT VOOR OPMAAK//
Onderstaande afbeeldingen graag naast elkaar plaatsen met één onderschrift ***zelfdenetwerk en ***andernetwerk Door het subnetmasker aan te passen, kunnen computers zich logisch gezien ineens in verschillende netwerken bevinden.

Dit voorbeeld was redelijk eenvoudig; als u wel voldoende bits hebt om nodes te adresseren, maar onvoldoende bits om netwerken te adresseren, leent u gewoon host-bits om netwerken te adresseren door het subnetmasker aan te passen. Binnen het kader van het geregistreerde netwerkadres dat u hebt, bent u volkomen vrij om dit te doen. Als u een klasse B-adres hebt, is dit niet moeilijk, u gebruikt dan gewoon de volledige derde byte. Als u echter een klasse C-adres hebt, kunt u dit niet doen. U hebt dan immers maar één byte om nodes te adresseren. U zult in dat geval genoegen moeten nemen met een aantal bits dat u van dat ene byte leent.

Eén klasse C-adres, meer dan een netwerk
Zoals gezegd, de kans dat u beschikt over een klasse B adres dat u vervolgens kunt gebruiken om meerdere netwerken te adresseren, is vrij klein. In de meeste gevallen hebt u hooguit een klasse C-adres. Toch is het mogelijk hiermee meerdere netwerken te bedienen, u mag alleen niet te veel computers per netwerk hebben. De oplossing hiervoor blijft in principe hetzelfde als in de vorige paragraaf; u moet bits van de host-adressen lenen om netwerken mee te adresseren. U kunt nu alleen niet een heel byte nemen, u zult zuinig precies het aantal bits moeten nemen dat u ook daadwerkelijk nodig hebt.

Laten we eens uitgaan van een voorbeeld. Stel, u hebt beschikking over het volledige netwerkadres 200.100.100.0. Hiermee moet u een totaal van vijf netwerken van geldige IP-adressen voorzien. Hoeveel bits hebt u daar dan voor nodig? Het antwoord op deze vraag is hetzelfde als het antwoord op de vraag hoeveel bits u nodig hebt om een totaal van 5 combinatiemogelijkheden te maken. Is 1 bit genoeg? Absoluut niet, daarmee kunt u immers maar 2 combinatiemogelijkheden maken. Zijn 2 bits genoeg? Bijna, maar net niet. Met 2 bits kunt u immers maar 4 combinatiemogelijkheden maken. U hebt dus een totaal van 3 bits nodig. Hiermee kunt 8 combinatiemogelijkheden maken dus maximaal 8 netwerken adresseren. Drie meer dan dat u nodig hebt, maar dat moet u maar voor lief nemen.

U weet dat het altijd het eerste deel van een IP-adres is dat gebruikt wordt voor de adressering van netwerken, het zijn dus ook de eerste bits in het subnetmasker die gebruikt worden om aan te geven dat het een netwerk wordt. Binair gezien wordt het subnetmasker voor het vierde byte in dit voorbeeld dus 11100000. Kunnen we dat ook decimaal opschrijven? Natuurlijk! 128 + 64 + 32 = 224. De eerste drie bytes van het subnetmasker blijven gewoon wat ze zijn; we hebben het hier immers over een geregistreerd klasse C-adres. We hebben met andere woorden niets te zeggen over die eerste drie bytes. Het subnetmasker wordt dus 255.255.255.224.

Nu is het natuurlijk de hamvraag welke netwerken dit subnetmasker oplevert. Ook dit kunt u weer het gemakkelijkst binair bekijken. Een subnetmasker 224 (we hebben het nu alleen even over het relevante vierde byte), betekent dat u de eerste drie bits van deze byte kunt gebruiken

om netwerken te adresseren. De laatste vijf bits moet u in alle gevallen van af blijven, die mag u immers alleen gebruiken om nodes mee te adresseren. Binair gezien mag u dus met dat vierde byte de volgende netwerken definiëren:

00000000 (0)
00100000 (32)
01000000 (64)
01100000 (96)
10000000 (128)
10100000 (160)
11000000 (192)
11100000 (224)

We hebben er voor het gemak gelijk maar even de decimale adressen achter gezet. Wellicht dat u zich nu afvraagt: "Ja maar leuk, maar waarmee moet ik nu mijn nodes adresseren?" Deze vraag is eenvoudig te beantwoorden. Per netwerk hebt u in principe nog 5 bytes over om nodes mee te adresseren. Maar let er wel op dat al deze vijf bytes niet de waarde 0 mogen hebben en ze mogen ook niet alle vijf de waarde 1 hebben. Alle host-bits de waarde nul, wordt immers gebruikt om te verwijzen naar een netwerk-adres, alle host bits de waarde 1 is een speciaal adres dat het "broadcast adres" genoemd wordt. Dit adres wordt gebruikt om alle computers die op een bepaald netwerk voorkomen te adresseren. Maar alle combinaties van 00001 tot en met 11110 zijn dus toegestaan.

Om dit maar even concreet te maken: zo zijn dus in het netwerk 200.100.100.32, dat vergezeld gaat van het subnetmasker 255.255.255.224 de adressen 200.100.100.33 tot en met 200.100.100.62 geldige adressen waarmee computers geadresseerd mogen worden.

Uit het bovenstaande heb je kunnen afleiden, dat hoe meer bits u gebruikt om netwerken te adresseren, hoe minder bits er uiteindelijk over blijven om nodes te adresseren. U hebt er immers maar acht in het totaal. Hieronder wordt dit voor alle duidelijkheid nog eens samengevat in een tabel weergegeven. In de tabel wordt er van uit gegaan dat het subnetten wordt toegepast op het vierde byte van een IP-adres:

Aantal bits gebruikt voor netwerken	**Aantal te adresseren netwerken**	**Aantal nodes dat in een netwerk voor mag komen**
0	0	254
1	2	126
2	4	62
3	8	30
4	16	14
5	32	6
6	64	2
7	128	0
8	256	0

Voordat u subnetmaskers toe gaat passen, moet u dus altijd eerst even rekenen. Als u bijvoorbeeld 6 netwerken hebt, met 35 nodes per netwerken, hebt u aan één geregistreerd klasse C-adres niet genoeg om alle nodes te adresseren. U zult dan meer adressen aan moeten vragen.

1.1.7 CIDR-notatie

U hebt tot dusver aardig wat kunnen lezen over het werken met subnetmaskers. In de voorbeelden die tot zover gehanteerd zijn, zijn de subnetmaskers steeds volledig uitgeschreven; bijvoorbeeld 255.255.255.0. Er is echter nog een andere manier waarop u een subnetmasker kunt schrijven; de zogenaamde CIDR (Classless Inter Domain Routing) methode. In deze methode schrijft u niet het volledige subnetmasker, maar alleen het aantal bits dat in het subnetmasker gebruikt wordt. In plaats van het volledige subnetmasker 255.255.255.0, kunt u dus de CIDR-notatie /24 gebruiken. U doet dit door de /24 direct achter het te gebruiken IP-adres te plaatsen; bijvoorbeeld 192.168.0.12/24.

Oefenvragen

1. Welk subnetmasker moet u gebruiken als u een klasse C-adres hebt waarmee u acht verschillende netwerken moet adresseren?
2. Wat is de CIDR notatie voor het subnetmasker 255.255.240.0?
3. Wat is het meest efficiente subnetmasker dat gebruikt kan worden als u drie netwerken moet kunnen adresseren en zelf beschikking hebt over het klasse A netwerkadres 10.0.0.0?
4. Wat is het broadcastadres dat hoort bij het netwerkadres 137.65.0.0. Motiveer uw antwoord.
5. Op welk netwerk komt de computer met IP-adres 137.65.1.89/27 voor?

1.2 Configuratie van een vaste netwerkkaart

Elke distributie heeft tegenwoordig zijn eigen manier om een netwerkkaart te configureren. Hetzelfde geldt overigens voor configuratie van modemverbindingen en DSL. Over het algemeen moeten u blij zijn met deze min of meer automatische configuratieprogramma's, ze maken het leven namelijk een stuk eenvoudiger. De configuratie van een modemverbinding bijvoorbeeld was een jaar of vijf geleden namelijk nog een regelrechte crime waarbij verschillende configuratiebestanden bewerkt moesten worden. In paragraaf 1.3 leest u hier in meer detail over. Vandaag de dag is de configuratie van modem, ISDN, netwerkkaart of DSL nauwelijks nog ingewikkeld. Als beheerder echter is het belangrijk dat u weet hoe u een netwerkkaart handmatig moet instellen. Een grafische interface is immers heel leuk, maar ook wanneer de grafische interface om welke reden dan ook niet beschikbaar is, moet u weten hoe u een netwerkkaart kunt configureren.

1.2.1 De netwerkkaart configureren met ifconfig.

Kort samengevat moeten er twee dingen gebeuren om een netwerkkaart te activeren. De netwerkkaart moet beschikbaar zijn en als hij dan beschikbaar is, moet hij aangestuurd worden. Voor wat betreft de beschikbaarheid van de netwerkkaart kunnen we kort zijn: hiervoor moet de netwerkkaart als kernelmodule geladen kunnen worden. Als de netwerkkaart dan in de kernel als module aanstaat, moet het bestand /etc/modules.conf nog bewerkt worden zodat de netwerkkaart op een eenvoudige wijze aangeroepen kan worden. In /etc/modules.conf moet een alias aangemaakt worden die ervoor zorgt dat uw specifieke netwerkkaart onder een algemene naam benaderd kan worden. U ziet hoe dit in onderstaand voorbeeld voor een 3c59x netwerkkaart gebeurt. Tevens ziet u een regel waarmee een tweede ethernet netwerkkaart aangestuurd kan worden en een regel waarmee een token ring netwerkkaart aangestuurd kan worden. Dezen staan echter beiden op "off" omdat deze netwerkkaarten niet in het systeem aanwezig zijn.

```
#
# Aliassen - om gebruik van hardware te vereenvoudigen
#
```

alias eth0 3c59x
alias eth1 off
alias tr0 off

Om in modules.conf een dergelijke verwijzing aan te kunnen maken, moet de betreffende kernelmodule wel op het systeem beschikbaar zijn. Controleer dit door de module eerst handmatig te laden met modprobe. Lukt dit niet? Pas dan met **make menuconfig** of **make xconfig** de configuratie van uw kernel aan.

Als de netwerkkaart als module geladen kan worden, moet er vervolgens voor gezorgd worden dat hij ook aangestuurd wordt. Dit gebeurt met het commando **ifconfig**. Tijdens de opstartprocedure zal er normaliter voor gezorgd worden dat dit commando automatisch uitgevoerd wordt, u kunt het commando ook handmatig gebruiken. Dit is niet moeilijk: geef het commando **ifconfig**, vervolgens de naam van de interface en tot slot het ip-adres dat gebruikt moet worden. Zo kan bijvoorbeeld met het volgende commando een netwerkkaart worden aangestuurd:

ifconfig eth0 192.168.0.10

Het commando ifconfig gaat vervolgens eth0 aansturen. Door het alias in /etc/modules.conf kan de kernel achterhalen dat eth0 gelijk is aan de 3c59x netwerkkaart. Dit zorgt ervoor dat de bijbehorende module op dat moment dynamisch geladen kan worden. Voor alle overige instellingen wordt de standaardwaarde gebruikt. OP basis van de standaard adresklasse, wordt in het voorgaande voorbeeld bijvoorbeeld gebruikgemaakt van het standaard subnetmasker 255.255.255.0.

Als u om welke reden dan ook de netwerkkaart daarna weer down wilt brengen, gebruikt u het volgende commando:

ifconfig eth0 down

1.2.2 Geavanceerde opties van ifconfig.

Op basis van het voorgaande kunt u een netwerkkaart op uw systeem activeren. Hierbij worden alle standaardinstellingen gebruikt. Leuk natuurlijk, maar soms wilt u meer dan de standaardinstellingen. Maak in dat geval gebruik van een van de vele opties die bij de opdracht **ifconfig** gebruikt kunnen worden. Ook is het mogelijk met **ifconfig** een tweede IP-adres aan een interface te verbinden. Dit laatste is bijvoorbeeld nuttig wanneer uw computer binnen een testnetwerk moet kunnen communiceren, maar gelijktijdig ook voorzien moet zijn van een IP-adres waarmee hij op internet kan communiceren. Ook als er op uw server een service draait die op een eigen IP-adres bereikt moet kunnen worden, is de optie te werken met secundaire IP-adressen erg handig. Ook hierbij maakt u gewoon gebruik van de opdracht ifconfig; u gebruikt alleen een speciale aanduiding bij de verwijzing naar de netwerkkaart om duidelijk te maken dat het hier een tweede netwerkkaart betreft. Waar u in het voorgaande de opdracht **ifconfig eth0 192.168.0.10** gebruikt hebt om uw netwerkkaart te voorzien van een IP-adres, gebruikt u bijvoorbeeld **ifconfig eth0:0 192.168.0.11** om diezelfde computer van een tweede adres te voorzien. Hebt u nog meer adressen nodig op dezelfde netwerkkaart? Geen probleem, dan herhaalt u gewoon de voorgaande procedure. het derde netwerkadres koppelt u aan eth0:1, het vierde adres aan eth0:2 enzovoorts.

***secundair Met ifconfig voegt u op eenvoudige wijze een secundair IP-adres toe aan een netwerkkaart.

Naast de mogelijkheid te werken met een secundair IP-adres, biedt ifconfig nog veel meer mogelijkheden. Twee van deze mogelijkheden zult u met name wel eens tegenkomen. Om te beginnen is dat de optie **netmask**. Als u namelijk een ander dan het standaard netmask wilt gebruiken, geeft u het te gebruiken netmask aan met de optie netmask, bijvoorbeeld in **ifconfig eth0 10.0.0.1 netmask 255.255.255.0**. Er is echter een probleem: bij het instellen van een afwijkend subnetmasker, wordt niet automatisch ook het broadcastadres aangepast. Dit blijkt wanneer u na de opdracht **ifconfig eth0 10.0.0.1 netmask 255.255.255.0** de instellingen van uw netwerkkaart bekijkt met het commando **ifconfig eth0**: het broadcastadres staat gewoon nog ingesteld op het broadcastadres van een standaard klasse A-netwerk. Om ervoor te zorgen dat ook deze optie goed wordt ingesteld, gebruikt u eveneens de optie **broadcast**. De volledige opdracht wordt dus **ifconfig eth0 10.0.0.1 netmask 255.255.255.0 broadcast 10.0.0.255**. Let er trouwens even op dat de opdracht ifconfig veel kan, maar dat hij niet gebruikt kan worden om de standaardgateway in te stellen. Daarvoor hebt u de opdracht **route** nodig (**route add default gw *adres-van-uw-gateway*** om precies te zijn). Later in dit hoofdstuk leest u meer over route.

1.2.3 Linux als DHCP-client

Met de opdracht ifconfig configureert u zelf handmatig met welk IP-adres uw computer aan het werk moet gaan. Uiteraard is het ook mogelijk een Linux-computer in te richten als DHCP-client. Hiervoor wordt gebruikgemaakt van het commando **dhclient**. Met behulp van deze opdracht zorgt u ervoor dat uw netwerk automatisch voorzien wordt van IP-configuratie. U kunt deze opdracht natuurlijk handmatig uitvoeren, in de meeste gevallen echter zal het vanuit de opstartroutine van de computer geactiveerd worden. De naam van deze opdracht wil overigens tussen de verschillende distributies nog wel eens afwijken: soms wordt dhclient gebruikt, het kan ook voorkomen dat de opdracht anders heet. Gebruik indien nodig de opdracht **man –k dhcp** om een lijst te tonen van alle opdrachten waarbij DHCP gebruikt wordt, u ziet dan in een handomdraai wat de juiste opdracht is om uw IP-configuratie op te halen bij een DHCP-server.

***dhclient Met de opdracht dhclient kunt u vanaf de meeste distributies IP-configuratie ophalen van een DHCP-server.

1.2.4 Andere opdrachten om IP-configuratie te beheren.

In het voorgaande hebt u gelezen over afzonderlijke opdrachten die gebruikt kunnen worden om Linux te voorzien van een “hard” IP-adres dat is ingesteld met ifconfig, of een IP-adres dat dynamisch wordt opgevraagd van een DHCP-server met behulp van de opdracht dhclient. In beide gevallen zal het niet voorkomen dat u deze opdrachten zelf in moet voeren; ze zullen vanuit de opstartprocedure automatisch geactiveerd worden op basis van configuratie die hiervoor is opgeslagen in configuratiebestanden onder de directory /etc/sysconfig. Als u met ifconfig een netwerkinterface down brengt, moet u er op het moment dat hij weer geactiveerd wordt, rekening mee houden dat alle configuratie van de netwerkkaart opnieuw moet worden ingevoerd. Dat is natuurlijk niet echt handig. Om die reden is het handiger gebruik te maken van de opdrachten **ifup** en **ifdown**. Wilt u uw eth0 uitzetten? Gebruik dan **ifdown eth0**. Wilt u eth0 weer activeren met zijn gangbare configuratie? Gebruik dan **ifup eth0**. De huidige configuratie wordt vervolgens ingelezen en u kunt direct weer aan het werk.
De opdrachten ifup en ifdown zijn overigens niet de enige andere methoden om een netwerkkaart aan het werk te krijgen. U kunt hiervoor ook gebruikmaken van het zeer

veelzijdige commando **ip** Dit commando kan alles wat **ifconfig** ook kan, en daarnaast nog wat extra's ook. Omdat het commando zo veelzijdig is, ligt het voor de hand dat het op den duur het bestaan van ifconfig overbodig zal maken. Omdat het echter zo complex is, wordt het momenteel nog niet al te veel gebruikt. We geven hier een korte introductie in het gebruik van dit commando.

Tip! Eigenlijk is er maar één manier die echt werkt om de volledige netwerkconfiguratie van alle netwerkkaarten in uw computer uit te zetten of weer aan te zetten en dat is met behulp van het script /etc/init.d/network. Het grote voordeel van gebruik van dit script, is dat u er niet slechts één netwerkkaart mee onderhanden neemt, maar gewoon alle netwerkkaarten op uw computer. Om één netwerkkaart aan of uit te zetten, zijn ifup en ifdown echter uitstekende alternatieven.

Om te beginnen is er de algemene syntaxis:

ip [opties] ***object*** **[commando [parameters]]**.

Het eerste wat u door moet hebben, is dat u met **ip** verschillende zaken aan kunt passen. Dit zijn de zogenaamde objecten. Drie verschillende objecten kunnen gebruikt worden:

* **link** Hiermee stelt u waarden in voor een netwerkkaart
* **address** Wordt gebruikt om een netwerkadres in te stellen
* **route** Kan gebruikt worden om een route te configureren.

De opties en parameters die u kunt gebruiken, zijn afhankelijk van het type object waarmee u werkt. Houdt er tevens rekening mee dat er gewerkt kan worden met afkortingen zolang deze tenminste ondubbelzinnig zijn. Zo kunt u in plaats van de opdracht **ip address show** ook gewoon gebruikmaken van **ip a s**.

***ipas Met de opdracht **ip address show** toont u een overzicht van de huidige configuratie van uw netwerkinterfaces.

Tip! De opdracht **ip** is zeer veelzijdig en krachtig. Hij heeft echter ook een nadeel: als u uw computer weer uit zet, bent u de volledige configuratie kwijt. Om die reden zou u kunnen overwegen dit lastige commando gewoon te vergeten en met ifconfig te werken als u dat tenminste gewend bent. Moet u definitief een wijziging aanbrengen in de netwerkconfiguratie van uw computer? Gebruik dan de configuratietool die voor dit doel met uw distributie geleverd is, zoals SUSE's YaST.

Om u te laten wennen aan de werking van de opdracht **ip**, tot besluit van deze paragraaf nog een korte samenvatting. Om te beginnen is er de parameter **link**. Deze gebruikt u om eigenschappen van een interface (link) te bekijken of in te stellen. De onderstaande opdrachten zijn hier voorbeelden van:

ip link show Toont de huidige configuratie voor alle netwerkkaarten in het systeem
ip link set eth0 down Zet eth0 uit.
ip link set eth0 mtu 1492 Stelt een maximum transfer unit (maximale pakketgrootte) in van 1492 bytes voor eth0.

Vervolgens kunt u **ip** ook gebruiken om adressen toe te kennen. De volgende opdrachten zijn hier voorbeelden van:

ip address add 10.0.0.1/24 dev eth0 broadcast+. Stelt een netwerkadres in met bijbehorend subnetmasker voor interface eth0. U moet een subnetmasker opgeven zoals hier gebeurt met de CIDR-notatie /24, als u dit vergeet wordt namelijk het standaard subnetmasker 255.255.255.255 ingesteld.
ip address add 192.168.0.1/24 broadcast+ dev eth0 label eth0:1. Dit commando toont een kleine variattie op het voorgaande: als extra wordt gebruikgemaakt van de toevoeging "label". In dit voorbeeld wordt deze toevoeging gebruikt om een netwerkkaart te voorzien van een extra IP-adres.

Tot slot lopen we even vooruit op de volgende paragraaf en tonen we u hoe u de opdracht **ip** kunt gebruiken om een standaardroute in te stellen:
ip route delete default. Als er al een standaardroute ingesteld was, wordt deze met behulp van deze opdracht verwijderd.
ip route add default via 192.168.0.1. Voegt de standaardroute toe. Let op de aanduiding **via**, deze is echt noodzakelijk om aan te geven via welke router de standaardroute loopt.
ip route add 192.168.1.0/24 via 192.168.0.1. Een variant op het voorgaande commando: hiermee voegt u een route toe voor een specifiek netwerk. Pakketjes naar dit netwerk moeten via 192.168.0.1 gerouteerd worden.
ip route show. Toont de opbouw van de routing tabel zoals die op dit moment is.

1.2.5 Configureren van een draadloze netwerkkaart

Als u boft, is uw draadloze netwerkkaart voorzien van een stuurprogramma voor Linux en kunt u hem zonder problemen installeren. Vaak is dit niet het geval en zult u zelf aan het werk moeten om de netwerkkaart te kunnen gebruiken. In veel gevallen is het zelfs niet eens mogelijk gebruik te maken van de draadloze netwerkkaart omdat de fabrikant van de netwerkkaart geen code vrijgeeft waarmee uw netwerkkaart aangestuurd kan worden. In dat geval bent u aangewezen op software die ervoor zorgt dat u de netwerkkaart onder Linux kunt gebruiken met behulp van het Windows-stuurprogramma dat voor de netwerkkaart verkrijgbaar is. Hiervoor maakt u gebruik van het open source project NdisWrapper of de commerciële software van linuxant. De eerste vindt u soms als onderdeel van uw distributie en anders op ndiswrapper.sourceforge.net, de laatste kunt u downloaden van www.linuxant.com. Aangezien de Linuxant software het op dit moment beter doet dat de NdisWrapper, bespreken we hier hoe u door gebruik te maken van deze software uw draadloze netwerkkaart aan het werk kunt krijgen. Let wel even op: het betreft hier commerciële software die u voor 20 dollar moet kopen. Voordat u het ook daadwerkelijk koopt, kunt u de software eerst dertig dagen uitproberen.

Tip! Probeer altijd eerst of uw netwerkkaart het ook doet met de software van uw distributie. Vooral in recente distributies is de ondersteuning van draadloze netwerkkaarten sterk verbeterd en als het op die manier lukt, kunt u toch eenvoudig twintig dollar besparen.

1. Ga naar www.linuxant.com, schrijf uzelf in en haal de gratis trial-versie van het programma binnen. Let erop dat u de software voor de juiste kernel-versie gebruikt, u kunt de versie van de kernel achterhalen door in een console-venster de opdracht **uname –r** te geven.
2. Geef de opdracht **unzip driverloaderversienummer** en vervolgens **rpm –i driverloaderversienummer** om de driver loader te installeren.

3. Activeer nu uw browser en ga naar http://127.0.0.1:18020; dit is de webpagina waarop u de Linuxant driverloader verder kunt configureren.
4. Zorg er nu voor dat de Windows-stuurprogramma's voor uw draadloze netwerkkaart ergens op het systeem beschikbaar zijn en klik op Upload Windows Driver om de Windows-stuurprogramma's voor uw netwerkkaart te uploaden.
5. Blader nu naar de locatie waar zich het Windows-stuurprogramma voor uw netwerkkaart bevindt. U herkent dit stuurprogramma aan zijn naam: het is een bestand met de extensie .inf dat zich op het installatiemedium van uw netwerkkaart bevindt. Let even op: het gaat niet om het bestand autorun.inf, maar om een ander bestand dat zich meestal in een subdirectory op de installatie-cd bevindt. Volg de aanwijzingen om het bestand te downloaden naar uw computer. Het kan zijn dat er meer bestanden nodig zijn. Volg de aanwijzingen om ook deze binnen te halen.
6. Als alles goed gegaan is, wordt uw netwerkkaart nu herkend. U zult hem echter nog niet kunnen gebruiken omdat er nog geen licentie beschikbaar is. Klik in de web-pagina op Settings om een licentie in te kunnen voeren. Volg hiervoor de aanwijzingen die vanuit de webpagina gegeven worden. Op deze webpagina kunt u ook aangeven dat u gebruik wilt maken van de gratis trial-licentie.
7. Nadat u hebt aangegeven van wat voor type licentie u gebruik wilt maken, moet u een license-token genereren. Klik hiervoor op de betreffende link in de webpagina. De licentie is nu geïnstalleerd. U komt nu weer terug in een venster waar u het MAC-adres van uw netwerkkaart ziet staan. Klik hier op Proceed om verder te gaan met installatie van de netwerkkaart.
8. Klik nu tenslotte op Save om de licentie te bewaren en de netwerkkaart in gebruik te nemen. Het kan nodig zijn in het laatste scherm dat u ziet nog even op Update te klikken voordat u het stuurprogramma in gebruik kunt nemen.

Uw netwerkkaart is nu aan uw computer toegevoegd als wlan0. Aangezien uw computer tot nu toe gewend was gebruik te maken van een bekabelde netwerkkaart, wordt de draadloze netwerkkaart niet automatisch gestart. Nu is het zaak de configuratie om te draaien en ervoor te zorgen dat de draadloze netwerkkaart automatisch gestart wordt en de bekabelde netwerkkaart alleen handmatig gestart wordt. Voordat u dit definitief in het systeem gaat doorvoeren, is het aan te raden eerst te controleren of een en ander handmatig aan de praat te krijgen is. In de onderstaande procedure leest u hoe u de bekabelde netwerkkaart handmatig uit kunt zetten terwijl u de draadloze netwerkkaart handmatig gaat configureren.

1. Geef de opdracht **ifconfig eth0 down**. Hiermee zet u de bekabelde netwerkkaart uit.
2. Geef nu de opdracht **ifconfig wlan0 192.168.0.100 netmask 255.255.255.0**. Uiteraard zorgt u ervoor dat het IP-adres dat u gebruikt geldig is op uw netwerk.
3. Gebruik nu de opdracht **ping** om te kijken of u met de router op het netwerk kunt communiceren. Gebruik bijvoorbeeld de opdracht **ping 192.168.0.1**, maar let er wel even op dat u het juiste IP-adres gebruikt. Krijgt u antwoord? Dan is uw draadloze netwerkkaart in de lucht.

Wanneer de bovenstaande test geslaagd is, wordt het tijd er voor te zorgen dat voortaan uw bekabelde netwerkkaart alleen handmatig gestart kan worden en de draadloze netwerkkaart automatisch in de lucht komt. Gebruik hiervoor het configuratieprogramma dat met uw distributie wordt meegeleverd. Onder SUSE Linux vindt u de relevante instellingen bijvoorbeeld wanneer u in YaST2 de netwerkkaart selecteert, de eigenschappen activeert en vervolgens onder de optie **Geavanceerd** kiest voor **Gedetailleerde instellingen**. Start uw

computer opnieuw op wanneer u hiermee klaar bent en controleer of de draadloze netwerkkaart het ook daarna het nog doet.

Gefeliciteerd als u tot hier gekomen bent, dat betekent dat uw draadloze netwerkkaart gebruikt kan worden. U bent er echter nog niet: de netwerkkaart wordt nu namelijk zonder beveiliging gebruikt. Voor een veilig netwerk is het aan te raden gebruik te maken van WEP-encryptie, wellicht dat tegen de tijd dat u dit leest ook de veel veiliger manier van encryptie WPA gebruikt kan worden.
Deze en andere geavanceerde opties kunt u instellen vanuit het beheersprogramma van uw distributie. In SUSE's YaST vindt u deze instellingen wanneer u de eigenschappen van de draadloze netwerkkaart selecteert. Selecteer vervolgens onder **Geavanceerd** de optie **Hardware Details** en klik daarna op **Draadloze instellingen**. U ziet nu een scherm waarin alle benodigde draadloze instellingen gedaan kunnen worden. Vooral belangrijk zijn de instellingen voor de netwerknaam en de encryptie-sleutel. Wanneer u de draadloze communicatie door middel van het WEP-protocol wilt beveiligen, moet u hier een sleutel invoeren die overeenkomt met de sleutel die op het wireless Access Point gebruikt wordt. Als dat nodig is, vindt u onder Expert Instellingen nog een aantal opties waarmee u de draadloze verbinding verder kunt configureren. Hier regelt u onder andere het draadloze kanaal dat gebruikt moet worden en de Bit-rate waarmee de gegevens verzonden moeten worden. Gebruik deze opties alleen wanneer u er zeker van bent dat u ze nodig hebt. Als u zich afvraagt waarvoor ze dienen, hebt u ze niet nodig en kunt u ze gewoon links laten liggen.

1.2.6 Instellen van de standaardroute en DNS-server

Op basis van het voorgaande kunt u nu gebruikmaken van uw netwerkverbinding. Eén ding moet echter nog geregeld worden: u kunt weliswaar op uw lokale netwerk communiceren, maar u bent nog niet in staat te communiceren met nodes op andere netwerken. Ook kunt u nog geen gebruik maken van namen zoals ze door middel van DNS beschikbaar gesteld worden. Om dit te regelen, gebruikt u eerst het commando **route** en wijzigt u vervolgens de inhoud van het configuratiebestand /etc/resolv.conf.
Om ervoor te zorgen dat uw computer de weg naar buiten kent, moet u de standaardroute definiëren. Gebruik hiervoor de opdracht **route add default gw *ip-adres***, bijvoorbeeld **route add default gw 192.168.0.1**. Vervolgens kunnen ook hosts op andere netwerken benaderd worden. Wel even opletten: de instellingen die u met route doet worden niet onthouden, om ervoor te zorgen gebruikt u het configuratieprogramma dat met uw distributie geleverd wordt.

**netdistro Gebruik het configuratieprogramma dat met uw distributie geleverd wordt om te zorgen dat de instellingen die u doet ook onthouden worden.

Nadat u de default route hebt ingesteld, zorgt u ervoor dat computers op het netwerk ook op basis van namen bereikt kunnen worden. Natuurlijk kunt u het bestand /etc/hosts aanpassen om daar IP-adressen en bijbehorende namen in op te nemen van computers waarmee u vaak contact hebt. Dit is vooral handig voor computers op uw eigen netwerk waarvoor geen entry in de DNS-database bestaat. Om echter te zorgen dat u ook computers op internet op basis van hun naam kunt bereiken, moet u zorgen dat er een DNS-server teruggevonden kan worden. Dit regelt u in het bestand /etc/resolv.conf.

***etchosts In /etc/hosts neemt u IP-adressen en namen op van computers op het lokale netwerk waarmee u regelmatig wilt communiceren.

De inhoud van resolv.conf hoeft helemaal niet ingewikkeld te zijn. In zijn meest eenvoudige vorm komt er maar één regel in voor:

```
nameserver     1.2.3.4
```

Met deze regel wordt het IP-adres gegeven van de DNS-server die u wilt gebruiken. Uiteraard zorgt u ervoor dat 1.2.3.4 vervangen wordt door het werkelijke IP-adres van uw DNS-naamserver. Dit is overigens een instelling die u niet beslist hoeft te doen met het configuratieprogramma van uw distributie. Omdat de waarde wordt opgeslagen in een configuratiebestand, is hij ook nadat de computer opnieuw gestart wordt gewoon nog beschikbaar.

1.3 Configuratie van een PPP-interface

Net als netwerkkaarten kunnen ook modems op twee manieren geconfigureerd worden: door de nodige configuratiebestanden handmatig te bewerken of met behulp van grafische configuratieprogramma's. In het ingevoegde kader kunt u alles lezen over handmatige modemconfiguratie waarbij de nodige scripts geconfigureerd moeten worden, in de meeste gevallen zult u echter voldoende hebben aan de (grafische) hulpmiddelen die door uw distributie beschikbaar gesteld worden. Als u studeert voor het LPI-2 examen, is het aan te raden het kader een keer goed door te lezen. U hoeft niet alles wat er staat letterlijk uit uw hoofd te weten, maar de handmatige configuratie van een modem maakt wel nog steeds deel uit van de LPI-specificaties.

<<AANMAKEN KADER>>
OPMERKING VOOR REDACTIE: ik weet dat het een lang stuk is, maar het moet toch als kader opgemaakt worden. Hierdoor weet de docent dat het optionele stof is die niet beslist behandeld hoeft te worden.

Handmatige modemconfiguratie.

Voordat u begint met uw pogingen om een modem aan te sturen, is het belangrijk even bij het volgende stil te staan. Vanuit de Linux optiek zijn er twee soorten modems. Als eerste zijn er de echte modems, daarnaast bestaan er ook apparaten die doen alsof ze een modem zijn. Deze laatste categorie modems hebben om te kunnen werken onderdelen van het Microsoft Windows besturingssysteem nodig en zullen het in veel gevallen niet doen onder Linux. Helaas horen veel PCI-modems in deze categorie thuis. Concreet betekent dit dus gewoon dat er een aantal modems is dat niet bruikbaar is onder Linux. Als u er zeker van wilt zijn dat uw modem wel met Linux te gebruiken is, schaf dan een extern modem aan. U kunt tevens lijsten met ondersteunde modems vinden op http://www.linux.org/hardware/components.html

De softwarematige configuratie van de modem bestaat uit twee stappen: de seriële poort moet geconfigureerd worden en vervolgens moet u er door middel van speciale modemcommando's voor zorgen dat de modem op de juiste manier aangestuurd kan worden.

Configuratie van de seriële poort

Vaak wordt al tijdens de installatie de vraag gesteld of u een modem wilt configureren. Als u hierop met 'yes' antwoordt, wordt er een bestand /dev/modem aangemaakt. Dit bestand wordt vervolgens gelinkt naar de communicatiepoort waarop uw modem aangesloten is, doorgaans ttyS0, ttyS1, ttyS2 en ttyS3, die overeenkomen met respectievelijk com1, com2, com3 en com4 onder Windows. U kunt de huidige instelling van de seriële poorten bekijken met het

commando setserial. Op het moment dat dit commando toont dat er een UART-chip gebruikt wordt op een van de aanwezige seriële poorten, is dit de poort waarop zich de modem bevindt. U ziet hiervan een voorbeeld in de onderstaande listing.

```
# setserial -g /dev/ttyS*
/dev/ttyS0, UART: 16550A,  Port: 0x03f8, IRQ: 4
/dev/ttyS1, UART: unknown, Port: 0x02f8, IRQ: 3
/dev/ttyS2, UART: 16550A, Port: 0x03e8, IRQ: 4
/dev/ttyS3, UART: unknown, Port 0x02f8, IRQ: 3
```

Configuratie van de modem
Als de modem eenmaal fysiek geïnstalleerd is, dient deze verder geconfigureerd te worden, zodat hij op de juiste manier aangestuurd kan worden. Deze configuratie houdt in dat u de juiste instellingen koppelt aan de modem. Voor deze instellingen worden initialisatiecommando's gebruikt die afkomstig zijn uit de AT-commandoset. Deze commandoset voorziet in een redelijk universele taal voor configuratie van modems.

De configuratie van de modem kan gebeuren met een klassiek communicatieprogramma zoals Minicom, Seyon of Kermit. Het is aan te raden om na de configuratie eerst door middel van het communicatieprogramma te proberen contact te maken met uw internetaanbieder. In de volgende paragrafen wordt uiteengezet hoe u dat kunt doen met behulp van het communicatieprogramma Minicom.

Om de modem te configureren opent u in het communicatieprogramma het menu waarmee u de seriële poort kunt configureren. In Minicom gebruikt u hiervoor de toetsencombinatie Ctrl-A, Z. Eerst stelt u de snelheid (bit-rate) van de seriële poort in. Maak deze altijd hoger dan de feitelijke snelheid van de modem; hiermee voorkomt u dat de seriële poort de vertragende factor wordt. Er is geen bezwaar de seriële poort in te stellen op een snelheid van 115.200 bps (bits per seconde). Daarnaast stelt u voor modems sneller dan 9600 baud (alle moderne modems dus) hardware-flowcontrol in met de optie RTS/CTS. Hierdoor kan de modem parameters uitwisselen met de andere modem over hoe de communicatie zo goed mogelijk tot stand kan worden gebracht.

Vervolgens dient u te controleren of in de initialisatiestring van de modem de volgende instellingen uit de AT-commandoset aan staan. (Houd er rekening mee dat de instellingen per modem verschillend kunnen zijn; raadpleeg de handleiding van uw modem voor de juiste instellingen.)

&K3 Zet hardware-flowcontrol aan. Dit is nodig voor alle modems die sneller zijn dan 9600 baud.
E1 Echo on. Zonder deze instelling kan het programma chat, dat u nodig hebt voor de PPP-configuratie, niet werken.
Q0 Laat resultaat codes zien. Ook deze instelling is essentieel voor het programma chat.
S0=0 Zorgt ervoor dat de modem niet automatisch antwoord geeft op binnenkomende telefoontjes.
&S0 Data Set Ready: altijd aan. Dankzij deze instelling kan de modem aan de remote modem laten weten dat hij klaar is voor verzenden en ontvangen van gegevens.

***minicom Om een modem te configureren, maakt u gebruik van een algemeen communicatieprogramma zoals minicom.

Als u de juiste instellingen hebt ingevoerd, kunt u deze wegschrijven naar de processor die voorkomt op de modem, of u kunt ze opnemen in een initialisatiebestand dat steeds wanneer het communicatieprogramma wordt opgestart, wordt geactiveerd. Als u meerdere besturingssystemen op uw computer gebruikt, heeft de laatste optie de voorkeur. De instellingen die voor Linux wenselijk zijn, kunnen immers voor gebruik onder een ander besturingssysteem onwenselijk zijn.

U kunt het communicatieprogramma weer opstarten wanneer u de bovenstaande instellingen hebt aangebracht. Met de commando's uit de AT-commandoset kunt u kijken of de modem werkt. Als eerste geeft u het commando AT om te kijken of de modem actief is. Hierop hoort de modem te antwoorden met OK. Als dit het geval is, kunt u een telefoonnummer opgeven om te bellen. Mocht de modem niet reageren op het commando AT, dan is de kans aanwezig dat het verkeerde initialisatiecommando is gegeven. U dient dan instellingen voor de initialisatiestring voor uw modem op te zoeken in de handleiding en deze string in te geven in het communicatieprogramma. Let er in elk geval op dat ook de hierboven vermelde instellingen in de initialisatiestring voorkomen. Met Minicom kunt u deze string instellen in het menu Configure Minicom. Dit menu roept u aan met de toetsencombinatie Ctrl-A, O. Uitgebreide informatie over de configuratie van modem en seriële poort vindt u in de Serial-HOWTO en de Modem-HOWTO.

Test of het werkt
Voordat u verder gaat, moet u eerst controleren of de modem werkt. U kunt nu proberen uw internetaanbieder te bellen. U gebruikt hiervoor in het communicatieprogramma het commando ATDT123456789 waarin 123456789 het telefoonnummer is van de internetaanbieder. Als het opbouwen van een connectie lukt, moet u exact noteren welke meldingen op uw beeldscherm verschijnen en welke antwoorden u moet geven om in te loggen. U hebt deze informatie in een later stadium nodig.

Er zijn twee zaken die u moet achterhalen: gebruikt de ISP PAP of CHAP en wordt PPP automatisch opgestart als u bent aangemeld bij de ISP, of moet u daar zelf nog iets voor doen? Of een server wel of niet PAP of CHAP gebruikt, kunt u zien wanneer u contact hebt gemaakt. Als de server wel PAP/CHAP gebruikt, krijgt u namelijk geen login-prompt te zien, maar machinecode. Het is dan zaak dat u een bestandje opstuurt naar de server waarin uw gebruikersnaam en wachtwoord staan. Als u een login-prompt te zien krijgt, mag u ervan uitgaan dat de server geen gebruik maakt van PAP/CHAP.

Daarnaast moet u kijken of de server automatisch PPP start of niet. Als de PPP-server automatisch wordt opgestart, krijgt u op uw beeldscherm iets te zien wat er uitziet als

~y}#.!}!}!} }9}!}$}%U}Ó}&}Ó} } }0 } } }%}& ...}'}Ó}(}Ó} . ~~y}
Krijgt u niet vanzelf iets dergelijks te zien, dan is het mogelijk dat de server wacht totdat u er een teken naartoe stuurt. Vaak is dat een willekeurig teken, soms moet u op Enter drukken. Als u op uw beeldscherm ziet dat de server PPP start, kunt u de modem afsluiten. Hiervoor gebruikt u de string +++ gevolgd door het commando ATH0.

Protocolconfiguratie
Als het bovenstaande allemaal goed is gegaan, werkt uw modem zoals het hoort. Nu kunt u beginnen met het aanmaken van een paar bestanden die nodig zijn voor het juist functioneren van DNS en PPP: /etc/resolv.conf en een paar aan PPP gerelateerde bestanden.

/etc/resolv.conf
Het eerste bestand dat u moet aanmaken is /etc/resolv.conf. Hiermee zorgt u ervoor dat de DNS-resolver contact kan maken met een DNS-server. Dit bestand krijgt een inhoud die eruit komt te zien als:

```
nameserver    192.168.191.1
nameserver    172.17.180.12
```

De permissies van dit bestand worden ingesteld op -rw-rw-rw-, zodat iedereen ze kan gebruiken. De gebruiker en groep van dit bestand zijn de gebruiker en de groep root.

PPP
Om ook iets te kunnen doen met uw modem, hebt u PPP nodig. Voordat u aan het werk kunt met PPP, moet u controleren of PPP-functionaliteit wel door de kernel wordt ondersteund. Dit doet u door het commando pppd te geven. Als PPP-functionaliteit niet door de kernel wordt ondersteund, krijgt u daar een duidelijke melding van. Als PPP-functionaliteit beschikbaar is, verschijnen op uw scherm tekenreeksen die eruit zien als:

```
~ij}#A!}!}!} }4}Ó}&} } } } }%}&MQE}'}Ó}(}
```

Vrijwel elke distrubitie heeft standaard ondersteuning voor ppp. Ook moet u ervoor zorgen dat iedereen gebruik kan maken van het programma pppd. Normaliter mag namelijk alleen de gebruiker root dit programma activeren. De handigste manier om dit te veranderen is door het User-ID bit aan het programma te koppelen. Gebruik hiervoor het commando **chmod u+s /usr/sbin/pppd**.
Vervolgens moet een algemeen instellingenbestand voor PPP worden aangemaakt. Hiervoor wordt het bestand /etc/ppp/options gebruikt. De inhoud ervan is afhankelijk van de vraag of uw ISP al dan niet PAP of CHAP gebruikt voor validatie van gebruikers. De laatste regel in het onderstaande voorbeeldbestand hoeft alleen te worden toegevoegd wanneer de ISP gebruik maakt van PAP/CHAP.

```
#voorbeeld /etc/ppp/options
#Maak van pppd geen background proces:
-detach
#Gebruik de modeminstellingen:
modem
#Laat pppd de communicatiepoort exclusief gebruiken door
#er een lock op te plaatsen. De lock komt voor als het bestand
#/var/lock/LCK..ttySx:
lock
#Gebruik hardware flowcontrol voor data die binnenkomt
#op de seriële poort. Alleen doen als uw modem sneller is
#dan 9600 baud:
crtscts
#voeg voor de duur van de verbinding de remote host
#toe aan de routing table als default router zodat alle
#IP-verkeer doorgestuurd wordt naar het Internet:
defaultroute
#Maak geen gebruik van “escaped” control-tekens:
```

```
asyncmap 0
#Stel de maximale grootte van uitgaande pakketjes in op 552:
mtu 552
#Stel de maximum receive unit (mru) in op 552 bytes. Hoe
#langzamer de verbinding, hoe lager de waarde die voor mtu
#en mru staat ingesteld moet zijn. Bij een langzame verbinding
#dient de waarde voor mru en mtu staan ingesteld op 296, de
#richtlijn voor een niet al te langzaam modem is 542. De waarde
#1500 dient alleen voor vaste netwerkverbindingen gebruikt te
#worden.
mru 552
#Alleen bij gebruik van PAP/CHAP: zorg ervoor dat PPPD uw
#ISP-gebruikersnaam als hostnaam gebruikt:
name <uw gebruikersnaam>
```

De algemene instellingen in het bovenstaande voorbeeldbestand kunnen voor de meeste computers die niet in een netwerk opgenomen zijn, worden gebruikt.

Alleen als uw internetaanbieder tijdens de inlogprocedure gebruik maakt van PAP of CHAP moet u nog wat extra werk doen. U moet dan een secrets-file aanmaken: een bestand waarin de informatie staat die u nodig hebt om u aan te melden door PAP of CHAP. Het bestand heet /etc/ppp/pap-secrets; het moet als eigenaar de gebruiker en groep root hebben en als permissie-modus 740. Als het echter de bedoeling is dat iedereen gebruik kan maken van dit bestand, moeten de permissies worden ingesteld op 744.

Het bestand /etc/ppp/pap-secrets komt eruit te zien als:

/etc/ppp/pap-secrets:

```
#Secrets for authentication using PAP
#client        server        secret        acceptable_local_IP_addresses
caroline              *              geheim
```

In pap-secrets geeft u eerst aan onder welke naam u aangemeld wilt worden bij de internetaanbieder. In het bovenstaande voorbeeldbestand is de gebruikersnaam ingesteld op caroline. Het tweede veld is bedoeld om de naam van de server op te geven. Aangezien u over het algemeen maar één internetaanbieder zult gebruiken, kunt u hier een asterisk neerzetten. Het derde veld wordt gebruikt om uw wachtwoord op te geven. In het vierde veld kunt u opgeven welke IP-adressen voor u acceptabel zijn. Dit veld kunt u over het algemeen leeg laten.

Behalve van PAP kan het PPP-protocol ook nog gebruik maken van CHAP voor authenticatie. Als dit het geval is, moet een bestand genaamd chap-secrets worden aangemaakt. Aangezien CHAP ervan uitgaat dat er aan wederzijdse validatie van gebruikersgegevens wordt gedaan, moet in chap-secrets behalve de naam van de lokale computer ook de naam van de computer waarop wordt ingelogd, worden opgenomen. Daarbij moet van beide computers het wachtwoord vermeldt worden.

Als de lokale computer de naam ‘Azlan’ heeft en het wachtwoord ‘training’, en de computer waarop wordt ingelogd ‘Akam’ en ‘Education’, komt het bestand /etc/ppp/chap-secrets er als volgt uit te zien:

/etc/ppp/chap-secrets

```
#Secrets for authentication using CHAP
#client server          secret          acceptable local IP addresses
Azlan          Akam          training
Akam           Azlan         education
```

Dit bestand maakt het mogelijk dat niet alleen de computer ‘Azlan’ zich aanmeldt op de computer ‘Akam’, maar dat ook het omgekeerde kan gebeuren. Dit is mogelijk omdat in /etc/ppp/chap-secrets ook de remote computer bekend wordt gemaakt. Omgekeerd dient ook op de remote computer een bestand /etc/ppp/chap-secrets te worden aangemaakt, waarin de computers die inbellen bekend worden gemaakt. Dit zal door de ISP geregeld worden.

Handmatig tot stand brengen van de PPP-connectie
Nu is het tijd om te kijken of u alles goed hebt gedaan: u gaat handmatig contact maken met de ISP en vervolgens PPP opstarten. Om dit succesvol te kunnen doen, is het belangrijk dat u uw communicatieprogramma kunt afsluiten zonder de modem daarbij te resetten. In Minicom doet u dit met de toetsencombinatie Ctrl-A, Q.

U begint met de modem te laten bellen naar de internetaanbieder. Nadat u succesvol bent aangemeld op diens server en aldaar een PPP-sessie voor u is opgestart, kunt u het communicatieprogramma afsluiten en het volgende commando geven:

pppd -d -detach /dev/ttySx 115200 &

Let erop dat u hierbij niet /dev/ttySx, maar ttyS0, ttyS1 typt, overeenkomstig de poort waaraan de modem geconfigureerd is. U kunt nu kijken of het werkt; gebruik hiervoor het commando ifconfig. Dit commando zou u een ppp-device moeten laten zien. De uitvoer van ifconfig moet een resultaat geven dat lijkt op

```
ppp0    Link encap:Point-to-Point Protocol
                inet addr:192.168.191.11 P-t-P:192.168.191.3 Mask...
                UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1524
                ...
```

Nadat u met een opdracht als **ping** gecontroleerd hebt dat de verbinding goed werkt, mag u ervan uitgaan dat alles wat u tot nu toe geconfigureerd hebt klopt. Start nu uw communicatieprogramma weer op en zorg ervoor dat de modem wordt ‘opgehangen’. Als hiervoor geen menuoptie beschikbaar is, kunt u hiervoor het commando +++ gevolgd door ATH0 (nul) gebruiken. Om het hierna nog een keer te proberen, dient u ook de lock-file, die is aangemaakt in /var/lock, te verwijderen. Deze lock-file heeft een naam als LCK..ttyS1.

Automatiseren van de connectie met scripts
Als het bovenstaande is gelukt, kunt u nu het tot stand brengen van de verbinding automatiseren. Dit doet u door een script aan te maken waarin een aantal omgevingsvariabelen wordt ingesteld en pppd wordt opgestart, waarbij automatisch het

programma chat geactiveerd wordt om de remote host te bellen. In de volgende paragrafen is uitgewerkt welke bestanden u hiervoor dient aan te maken. Het eerste bestand is /etc/ppp-on, dat zaken als login-naam, wachtwoord en telefoonnummer kan instellen. Tevens wordt in dit bestand pppd opgestart.

Het tweede bestand dat wordt aangemaakt, is een algemeen instellingenbestand waarin de parameters die door chat gebruikt worden, staan gespecificeerd. Dit bestand wordt geplaatst op een locatie waar iedereen er bij kan. In het volgende voorbeeld wordt het ppp-on-dialer genoemd.

Het derde bestand moet ervoor zorgen dat PPP ook weer wordt afgesloten. Ook dit bestand kan geplaatst worden in een directory waar iedereen bij kan. In het volgende voorbeeld heet het ppp-off. Elk van deze scripts moet in elk geval zijn voorzien van de execute-permissie voor de gebruiker die eigenaar is en voor ieder ander die er gebruik van moet kunnen maken.

ppp-on
Het bestand ppp-on moet worden aangemaakt in de homedirectory van elke gebruiker die het recht heeft om de modem aan te sturen of in een directory waar iedereen die er gebruik van mag maken het kan benaderen om het te starten. De inhoud is als volgt:

```
#!/bin/sh
TELEPHONE=0991111111
ACCOUNT=ppp:uwnaam@uwisp
PASSWORD=hiergewoonuittelezen
LOCAL_IP=0.0.0.0
REMOTE_IP=0.0.0.0

export TELEPHONE ACCOUNT PASSWORD

DIALER_SCRIPT=/etc/ppp/ppp-on-dialer
exec /usr/sbin/pppd debug /dev/ttyS1 115200 \
                        $LOCAL_IP:$REMOTE_IP \
                        connect $DIALER_SCRIPT
```

Het meest interessante in bovenstaand script is dat een aantal omgevingsvariabelen wordt gedefinieerd. Deze variabelen worden vervolgens met het bash-commando export ook buiten het script beschikbaar gesteld. Daarna wordt, ook weer door middel van een omgevingsvariabele, aangegeven op welke locatie het script met chat-parameters zich bevindt. Als laatste wordt pppd opgestart, waarbij de seriële poort waaraan de modem zich bevindt, wordt aangegeven en een verwijzing wordt gegeven naar het script met chat-parameters.

Ook worden in het bovenstaande script, indien bekend, de IP-adressen gegeven. Het is gebruikelijk dat het adres van de andere computer hier op 0.0.0.0 staat, tenzij u absoluut zeker weet dat u altijd aan dezelfde computer bent verbonden. Meestal is dit niet het geval. Tevens kan het lokale IP-adres – als het bekend is – worden gespecificeerd. Als gebruik wordt gemaakt van IP-adressen die dynamisch door de remote host worden uitgedeeld, dient hier 0.0.0.0 te staan.

Als uw internetaanbieder gebruikmaakt van PAP of CHAP voor validatie van gebruikersgegevens, hoeft in het script ppp-on geen informatie opgenomen te worden over gebruikersnaam en wachtwoord. Deze informatie staat dan immers gespecificeerd in /etc/ppp/pap-secrets of /etc/ppp/chap-secrets.

ppp-on-dialer
Het volgende script dat geschreven moet worden, is dat waarin de chat-parameters staan gespecificeerd. Dit wordt ook wel het chat-script genoemd. Hierin staat aangegeven hoe de communicatie plaatsvindt en wat er gecommuniceerd wordt. Het chat-script bestaat uit strings waarvan het eerste deel aangeeft wat verwacht wordt dat binnenkomt, en het tweede deel wat daarop geantwoord moet worden. Een regel uit een chat-script zou eruit kunnen zien als:

ogin: mijnloginnaam ssword: zeergeheim

In bovenstaande regel wordt gespecificeerd dat chat een string ogin verwacht. Als deze string binnenkomt, wordt daarop geantwoord met de string mijnloginnaam. Vervolgens wacht chat op een string ssword, waarop geantwoord wordt met zeergeheim. In de strings die verwacht worden, zijn in bovenstaande regel de eerste letters weggelaten. Dit om ervoor te zorgen dat niet direct een fout gegenereerd wordt als het eerste teken niet goed binnenkomt.
De status van het inbellen wordt vaak door een modem gerapporteerd door middel van een string als CONNECTED, NO CARRIER of BUSY. Wanneer er bij dergelijke berichten van de modem actie moet worden ondernomen, kan het keyword ABORT opgenomen worden. In het script ziet dat eruit als:

ABORT ‘BUSY’ ABORT ‘NO CARRIER’ ‘‘ ATZ OK ATDT111234
CONNECT

Hier wordt de modem eerst opnieuw ingesteld met het commando ATZ. Als de modem daarop antwoordt met OK gaat dit script bellen door het commando ATDT. Als hierop een BUSY of NO CARRIER volgt, wordt het script afgebroken. Wanneer een CONNECT volgt, wordt de rest van het script uitgevoerd.

Met het keyword ECHO kan op het beeldscherm getoond worden wat er gaande is. Met ECHO ON wordt alles wat er in het script gebeurt, getoond. Met ECHO OFF wordt het script afgehandeld zonder dat resultaten op het beeldscherm getoond worden. Het keyword TIMEOUT kan in een script gebruikt worden om op te geven hoe lang maximaal op een bepaalde string gewacht wordt. Zo kunt u het script onderbreken wanneer u te lang op een bepaalde string wacht.

Het statement REPORT CONNECT ten slotte kan ervoor zorgen dat de snelheid van de connectie wordt weergegeven nadat deze tot stand is gekomen. Om dit statement te kunnen gebruiken, is het wel noodzakelijk dat u het commando chat activeert met de optie -r.

Naast de hierboven genoemde keywords kunnen ook enkele Escape-reeksen in een script gegeven worden:
‘‘ Verwacht of stuur een string zonder inhoud. Hierop volgt als antwoord het Enter-teken.
\b Backspace.
\c Onderdruk het newline-teken dat aan het eind van elke string wordt meegestuurd.
\d Wacht een seconde.

\K Voeg een BREAK toe.
\n Stuur een newline-teken.
\r Stuur of verwacht een Carriage-Return.
\s Staat voor een spatie.
\t Staat voor een tab.
\\ Stuur of verwacht een backslash.

Het chat-script komt er als volgt uit te zien:

```
#!/bin/sh
/usr/sbin/chat -v -r                    \
                        TIMEOUT     3                       \
                        ABORT               ‘\nBUSY\r’                  \
                        ABORT               ‘\nNO ANSWER\r’             \
                        ABORT               ‘\nRINGING\r\n\r\nRINGING\r’    \
                        ‘‘                  \rAT            \
                        ‘OK-+++     \c-OK’          ATH0            \
                        TIMEOUT     30                      \
                        OK                  ATDT$TELEPHONE              \
                        CONNECT     ‘‘                      \
                        ogin:Ñogin:                 $ACCOUNT  \
                        assword:    \q$PASSWORD     \
                        ‘‘                  ‘‘                          \
                        REPORT CONNECT
```

In het bovenstaande script wordt allereerst aangegeven dat de shell /bin/sh gebruikt moet worden om voorkomende scriptcommando’s uit te voeren. Vervolgens wordt op de tweede regel het programma chat geactiveerd. De optie -v zorgt ervoor dat alle meldingen die door het script worden gegenereerd, worden opgeslagen in de standaard system-logfile (meestal /var/log/messages). Deze regel wordt afgesloten met een backslash, om aan chat te laten weten dat op de volgende regel de specificatie van argumenten gewoon doorgaat.

Nu wordt een time-out gegeven. Hierdoor blijft het systeem niet eindeloos wachten totdat er iets gebeurt, maar kan het script bij een voorkomende fout worden afgesloten. De time-out wordt ingesteld op drie seconden. De volgende drie regels specificeren condities op basis waarvan het script kan worden afgebroken. Dit kan gebeuren als de lijn bezet is (BUSY), als de remote host niet antwoordt (NO ANSWER) of als u gebeld wordt (RINGING).
Als geen van deze drie foutcondities voorkomt, kan overgegaan worden tot het echte werk. Dit wordt gestart met ‘‘, wat betekent dat niets van de modem verwacht wordt. Het systeem begint vervolgens de conversatie door het commando AT naar de modem te sturen. Het antwoord dat hierop wordt verwacht, is OK-+++\c-OK. Dit houdt in dat de modem OK terug dient te sturen. Als de modem dat niet doet, wordt de string +++ naar de modem gestuurd. De modem dient hierop te antwoorden met OK. Ten slotte wordt het Hang-up-signaal naar de modem gezonden. De modem is daardoor klaargezet om te gaan bellen.

Nu wordt een nieuwe time-out-waarde ingesteld. Hierdoor mag het langer duren voordat op de volgende activiteiten een reactie volgt. Dit is handig, omdat in de volgende regels contact wordt opgebouwd met de ISP. Het systeem communiceert daarbij niet meer met de modem die lokaal is aangesloten, maar over een telefoonverbinding met de ISP. Daar kan wat meer tijd overheen gaan.

Nadat het signaal OK van de modem is ontvangen, stuurt het script de waarde van de omgevingsvariabele TELEPHONE naar de modem. Deze variabele is ingesteld in het vorige script (ppp-on). De modem moet hierop reageren met het signaal CONNECT. Als dit signaal is ontvangen, stuurt de modem als antwoord een leeg pakketje terug (‘ ‘). Vervolgens wordt de string ogin: verwacht, waarop de inhoud van de variabele ACCOUNT wordt gestuurd. Nu wordt de string assword: verwacht, waarop de inhoud van de variabele PASSWORD wordt gezonden. De \q die hieraan voorafgaat, zorgt ervoor dat het wachtwoord niet wordt afgedrukt in de log-files van de lokale computer. Daarna wordt gewacht op een leeg pakketje dat door de remote computer verstuurd wordt, waarop geantwoord wordt met een pakketje zonder inhoud. Deze regel is nodig voor remote computers die wachten op een toetsaanslag voordat PPP op de remote machine wordt opgestart. Als laatste wordt de connectiesnelheid op het beeldscherm afgedrukt.

Als de ISP gebruik maakt van PAP of CHAP, wordt het chat-script veel eenvoudiger. Alles na de regels

```
OK      ATDT$TELEPHONE          \
CONNECT     ‘‘
```

kan dan namelijk weggelaten worden. Authenticatie wordt immers afgehandeld door het bestand /etc/ppp/pap-secrets of /etc/ppp/chap-secrets.

ppp-off
Het laatste script dat wordt gemaakt, moet de PPP-verbinding weer netjes afsluiten. Hiervoor wordt een vrij complex shell-script gebruikt, dat twee dingen doet. In de eerste plaats gaat het op zoek naar een bestand ppp0.pid. De deamon pppd zorgt ervoor dat dit bestand aanwezig is terwijl het proces actief is. In dit bestand staat het PID van de pppd-daemon. Dit bestand achterhaalt de PID van pppd, waardoor het proces opgeruimd kan worden met het commando kill. Daarnaast worden de locks die op de communicatiepoort staan, weggegooid. Hieronder volgt een voorbeeld van een ppp-off.

```
#!/bin/sh
if [ “$1” = “” ]; then
                        DEVICE=ppp0
else
                        DEVICE=$1
fi

if [ -r /var/run/$DEVICE.pid ]; then
                        kill -INT ‘cat /var/run/$DEVICE.pid’
                        rm -f /var/lock/LCK..*

                        if [ ! “$?”= “0”]; then
                                                rm -f /var/run/$DEVICE.pid
                                                echo “ERROR: Removed stale pid-file”
                                                exit 1
                        fi

                        echo “PPP link to $DEVICE terminated”
```

```
                exit 0
fi

echo "ERROR: PPP link is not active on $DEVICE"
exit 1
```

Testen of het werkt
Als de scripts allemaal zijn aangemaakt, kunt u kijken of het werkt. U dient dan eerst de logfile van het systeem op één terminal te openen met het commando tail -f /var/log/messages. Hierdoor kunt u nagaan wat er allemaal gebeurt. Op een andere terminal geeft u het commando **ppp-on &**. Nu gaat u terug naar het scherm waar de laatste regels van het bestand messages getoond worden. U ziet hier iets als het volgende:

```
Jul 2 13:20:12 laetitia pppd[211]: pppd 2.3.5 started by root
Jul 2 13:20:12 laetitia chat[216]: abort on (BUSY)
Jul 2 13:20:12 laetitia chat[216]: abort on (NO CARRIER)
Jul 2 13:20:12 laetitia chat[216]: expect ( )
Jul 2 13:20:12 laetitia chat[216]: warning: read() on stdin returned 0
Jul 2 13:20:12 laetitia chat[216]: Failed
Jul 2 13:20:13 laetitia chat[216]: Can't restore terminal parameters: Input/output error
```

In het logbestand is te zien waar en in welk proces er iets fout ging. In bovenstaand logfile treedt een foutmelding op nadat twee abort-condities wel goed zijn gegaan. Om de fout op te lossen, dient dus gekeken te worden in het script met chat-parameters en wel op de regel die volgt op de definitie Abort on NO CARRIER. Doet alles het naar behoren? Dan kunt u voortaan uw inbelverbinding activeren met behulp van de opdracht **wvdial**.
<EINDE KADER>

1.4 Testen of het werkt

Als u denkt dat u de netwerkverbinding naar behoren hebt ingericht, wordt het tijd voor de laatste fase: testen dat het werkt. Hiervoor staan u verschillende opdrachten ter beschikking:

* Gebruik ping om een verbinding tussen twee computers te testen
* Gebruik traceroute om te controleren of alle routers naar behoren werken
* Gebruik netstat om de status van alle netwerkverbindingen te controleren

1.4.1 Een netwerkverbinding testen met ping

Een van de belangrijkste en tevens een van de eenvoudigste tools die u in kunt zetten om een verbinding tussen twee computers te testen, is het commando ping. U gebruikt deze opdracht om de verbinding tussen twee computers te testen en daarbij maakt het niets uit waar deze computers precies voorkomen. Hebt u twijfel of de IP-stack op een computer naar behoren is geconfigureerd? Dan is ping de oplossing. U gebruikt deze opdracht echter niet alleen om de bereikbaarheid van computers op basis van hun IP-adres te testen, u kunt ook de DNS-naam van een computer als argument opgeven.

Wanneer u met ping een pakketje verstuurt, wordt er een zogenaamd ICMP-datagram verstuurt. ICMP is een hulpprotocol dat zich qua niveau naast IP bevindt: dit betekent dat er geen UDP of TCP voor nodig is om een ping-pakketje te versturen. Dit heeft een belangrijk voordeel: ping blijft hierdoor heel eenvoudig en is prima in staat om problemen op het netwerk op te lossen. Bij het versturen van een eenvoudig ping-pakketje naar een andere computer, wordt veel informatie duidelijk. Als u bijvoorbeeld met de opdracht **ping**

www.novell.com de bereikbaarheid van de genoemde host wilt testen, gaat ping onafgebroken pakketjes sturen naar deze host. Dit vesturen van pakketjes houdt op wanneer u het met de toetscombinatie Ctrl-C onderbreekt. Uit het resultaat van ping, worden verschillende zaken duidelijk.

***ping Met het versturen van ping-pakketjes wordt niet alleen duidelijk dat een host bereikbaar is, maar ook hoe deze host dan bereikbaar is.

Als eerste ziet u een aanduiding van het volgnummer van het pakketje dat u verstuurt. In de afbeelding wordt dit weergegeven met de aanduiding **icmp_seq**. Deze informatie is niet zo bar interessant, buiten dat u ermee kunt zien hoeveel ping-pakketjes er tot zover verstuurd zijn. Vervolgens ziet u iets wat wel interessant is: de Time To Live-waarde (ttl). Hieraan kunt u zien hoeveel tussenliggende routers er gepasseerd zijn om dit ping-pakketje te versturen. Het principe van een TTL is eenvoudig: op Linux wordt de standaard meestal ingesteld op 64 en elke router waardoor een pakketje geroutereerd wordt, haalt daar 1 van af. Als er dus tien routers tussen de verzender en de uiteindelijke bestemming voorkomen, houdt u een TTL van 54 over. Tot slot ziet u hoe lang het duurt om de betreffende host te bereiken. In de afbeelding wordt een pakketje verstuurd naar het andere eind van de wereld, in dat geval is een snelheid van gemiddeld 200 milliseconden niet echt slecht, zeker niet als u zich realiseert dat in die 200 ms het pakketje heen en teruggestuurd is. Dit heen en terug sturen wordt overigens de round trip genoemd. Als u pakketjes verstuurt naar een host die voorkomt op hetzelfde netwerk, zou u echter een betere doorvoersnelheid moeten krijgen. Tot slot ziet u ook nog een statistische samenvatting van het totale ping-verkeer tussen uw computer en de bestemming.

Houdt er overigens rekening mee dat de opdracht ping niet zaligmakend is. Omdat een host die een ping-pakketje ontvangt, daar ook op moet antwoorden, betekent het versturen van ping-pakketjes ook dat daarmee de host waarnaar u pingt belast wordt. Door gelijktijdig heel veel ping-pakketjes te versturen, zou u dus een leuke Denial of Service (DoS) aanval uit kunnen voeren. Om die reden komen er op internet aardig wat hosts voor die niet antwoorden op ping pakketjes.

Wanneer het versturen van een ping pakketje niet lukt, kunnen er verschillende foutmeldingen optreden.

* Er gebeurt niets. Probeer bijvoorbeeld maar eens de host www.microsoft.com te pingen. Er verschijnt in dat geval geen foutmelding, u krijgt gewoon geen antwoord. In de meeste gevallen is dit een teken dat er op de host waarnaar u pingt gefilterd wordt.
* U krijgt de melding Destination Host Unreachable. Dit betekent dat de host waarmee u contact probeert te maken, niet bereikt kan worden. Meestal komt het er op neer dat de standaardgateway plat ligt of dat de host in kwestie gewoon echt niet bereikt kan worden. Het kan ook zijn dat een slimme netwerkbeheerder zijn firewall zo geconfigureerd heeft dat dit antwoord gegeven wordt op elke host die u probeert te pingen, terwijl deze gewoon wel bereikt kan worden.
* U krijgt de melding network unreachable. Dit komt voor wanneer het netwerk in kwestie niet bereikt kan worden. Dit betekent vrijwel altijd dat er een probleem is met uw default gateway, het kan ook zijn dat er ergens op de route op internet tussen uw computer en die van de bestemming een netwerkprobleem is.
* U ping op naam en krijgt de melding unkown host. In dit geval kan het IP-adres dat bij de betreffende host hoort niet achterhaald worden. Dit duidt meestal op een probleem met uw DNS-configuratie.

* Naast deze redelijk vaak voorkomende foutmeldingen, kunnen nog andere foutmeldingen gegenereerd worden, bijvoorbeeld wanneer u probeert te pingen met een pakketje dat te groot is. Deze situaties doen zich minder vaak voor en worden hier om die reden niet verder behandeld.

***pingfout Bij het versturen van pakketjes met ping, kunnen verschillende soorten foutmeldingen gegenereerd worden.

Met het commando ping kan ook een aantal opties gegeven worden om het gedrag van ping verder te bepalen. Hieronder volgt een overzicht van een aantal nuttige opties.

Parameter	**Beschrijving**
-c *getal*	Geeft aan hoeveel ping-pakketjes er verstuurd moeten worden.
-I *adres*	Hiermee kunt u op een computer met meerdere netwerkkaarten een ping-pakketje via een specifieke netwerkkaart versturen.
-i *seconden*	Gebruik deze optie om tussen het versturen van ping pakketjes een aantal seconden te wachten. Dit kan handig zijn om de instellingen van bepaalde firewalls te omzielen.
-f	Alleen root kan deze optie gebruiken om een ping-flood te starten. Deze optie zorgt ervoor dat er minimaal honderd pakketjes per seconde verstuurd worden.
-n	Vertaal in het antwoord van een host het binnenkomende IP-adres niet in een DNS-naam. Met behulp van deze optie kunt u de belasting van de opdracht ping minimaliseren.
-t *ttl*	Geeft aan welke waarde voor de TTL gebruikt moet worden.
-b	Maakt het mogelijk ping-pakketjes te versturen naar het Broadcast-adres van een netwerk.

1.4.2 Functionaliteit van routers testen met traceroute

Als er een probleem is met de routering van pakketjes, kunt u de opdracht traceroute inzetten. Met behulp van deze opdracht analyseert u de route die gevolgd wordt wanneer een pakketje naar een bepaalde bestemming op internet verstuurd wordt. Dit is een zeer nuttige opdracht om meer te leren over hoe het internet in elkaar zit en daarnaast kan hij ook nog gebruikt worden om problemen op te sporen. Houdt er echter wel rekening mee dat op sommige sites een firewall actief is die de werking van traceroute tegen houdt; u ziet in dat geval maar een beperkt gedeelte van de weg die daadwerkelijk wordt afgelegd.

Wanneer u het commando **traceroute** gebruikt, verstuurt dit commando een paar pakketjes met een TTL-waarde van 1 in de richting van de host die u wilt tracen. Aangezien elke router 1 aftrekt van de huidige TTL, komt de TTL bij de eerste router op 0 te staan. Een pakketje met een waarde van 0 wordt als onbestelbaar geïnterpreteerd en daarom krijgt de verzender van het pakketje (u dus) een foutmelding dat het pakketje niet verstuurd kon worden. Dat is mooi, want zo hebt u wel inmiddels geleerd wat de eerste router is die u op weg naar de bestemming tegenkomt. Vervolgens wordt het proces herhaald met een TTL-waarde van 2 waarop de tweede router zal reageren en dit herhaalt zich net zo lang totdat de uiteindelijke bestemming bereikt is. Het resultaat hiervan is dat u een mooi lijstje ziet van alle routers tussen uw computer en de uiteindelijke bestemming. Traceroute heeft echter wel een nadeel: als er te actief gefilterd wordt (bijvoorbeeld door een firewall op uw eigen computer) ziet u helemaal niets.

1.4.3 De status van netwerkverbindingen controleren met netstat

Een van de meest veelzijdige opties die u kunt gebruiken om problemen op uw netwerkverbinding op te lossen, is **netstat**. Het commando is zelfs zo veelzijdig dat door zijn veelzijdigheid veel beheerders niet goed weten hoe ze het nu precies moeten gebruiken. Kort samengevat gebruikt u netstat om de status van alle netwerkverbindingen te bekijken. Het gaat dan om openstaande connecties (handig om te detecteren of iemand op uw systeem aan het werk is), de routingtable en netwerk interfaces. Het verwarrende van netstat echter is dat u niet alleen informatie te zien krijgt over openstaande netwerkverbindingen, maar over sockets die in gebruik zijn. Dit betekent dat netstat wanneer u het zonder argumenten aanroept ook heel veel informatie laat zien over interne processen die actief zijn.

***netstat Als u het zonder argumenten aanroept, geeft netstat informatie over letterlijk alle netwerkverbindingen die open staan.

Netstat wordt echter wel een heel erg handig commando wanneer u een aantal opties gebruikt om te bepalen welke informatie precies getoond moet worden. Met name de opties –t (toont TCP-connecties) en –u (toont UDP-sockets) zijn erg handig om te achterhalen welke netwerkverbindingen momenteel open staan. Daarnaast kunt u de optie –r gebruiken om de routingtabel uit te lezen, maar die informatie kunt u natuurlijk ook op andere manieren boven water krijgen.

***netstats.tif Met de optie –t toont netstat u in een handomdraai welke verbindingen er naar uw computer open staan.

Samenvatting
In dit hoofdstuk hebt u geleerd hoe u een netwerkverbinding op moet zetten. Hierbij is aandacht besteed aan de configuratie van een netwerkkaart met een IP-adres en de wijze waarop u er voor zorgt dat uw host ook op basis van een DNS-naam kan communiceren met hosts op andere netwerken. Tevens hebt u geleerd hoe nu eigenlijk IP-adressen en subnetmaskers gebruikt moeten worden. Vervolgens hebt u kunnen lezen over de wijze waarop handmatig een PPP-verbinding tot stand gebracht moet worden. Tot slot hebt u kennisgemaakt met een aantal van de drie belangrijkste hulpmiddelen die ingezet kunnen worden wanneer een verbinding niet naar wens tot stand gebracht kon worden.

Oefening 1.1
Zorg ervoor dat uw computer voorzien wordt van een IP-adres. Uw docent zal u vertellen welk IP-adres hiervoor gebruikt kan worden. Bestudeert u dit boek zelfstandig? Gebruik dan een van de adressen uit de private address range zodat u anderen niet in de weg zit. Zorg er in elk geval voor dat dit adres in hetzelfde netwerk voorkomt als de standaardgateway die u nodig hebt om naar buiten te komen. Voorzie uw computer vervolgens van een tweede IP-adres die wel aan dezelfde netwerkkaart verbonden wordt. Controleer met het commando ping dat beide IP-adressen werken. Gebruik vervolgens het juiste commando om in één keer te testen of u contact kunt maken met de rest van de wereld (dit kan natuurlijk alleen maar wanneer u vanuit uw netwerk contact kunt maken met internet). Als u alles handmatig werkend hebt kunnen krijgen, gebruik dan het configuratieprogramma van uw distributie om ervoor te zorgen dat na herstart van uw computer deze configuratie toch nog gebruikt kan worden. Controleer vervolgens welke netwerkverbindingen er allemaal open staan op uw computer.

Tip: hebt u problemen om andere computers te bereiken? Dan kan het zijn dat er een firewall aan staat op uw computer. Gebruik de opdracht iptables --flush om de huidige firewall

configuratie tijdelijk weg te gooien zodat u zonder belemmeringen de voorgaande oefening uit kunt voeren.

Oefenvragen.

1. Welke opdracht gebruikt u om handmatig een PPP-verbinding te initialiseren?
2. Met welke opdracht toont u een lijst van alle openstaande TCP en UDP-connecties?
3. Wat is er waarschijnlijk aan de hand wanneer ping de foutmelding host unreachable geeft?
4. Wat is het kortst mogelijke commando om alle netwerkkaarten op uw systeem down te brengen?
5. In welk bestand regelt u de verwijzing naar uw DNS-servers?
6. Welk commando gebruikt u om 192.168.0.1 in te stellen als default gateway?
7. Hoe ziet het volledige commando eruit als u een computer in wilt stellen met IP-adres 193.173.192.99 en subnetmasker 255.255.255.224?
8. Hoe kunt u het subnetmasker 255.255.224.0 ook schrijven?
9. Welke opdracht kan als alternatief voor ifconfig gebruikt worden?
10. In welk bestand regelt u dat de juiste driver voor uw netwerkkaart automatisch geladen wordt?

Hoofdstuk 2 Berichten versturen met Linux

Inleiding
Een zeer populaire taak waar Linux ook voor gebruikt wordt, is als mailserver. Nu is er wat betreft mailservers aan klassieker waarvan u volgende de LPI-specificaties ook goed op de hoogte moet zijn en dat is sendmail. In dit hoofdstuk leert u in dit hoofdstuk hoe u Sendmail moet configureren om mail te versturen aangezien het toch een zeer veel gebruikt programma is. Na het gedeelte over Sendmail, leert u hoe u met Majordomo een mailing list kunt configureren zodat u automatisch periodiek berichten kunt sturen naar iedereen die op de mailinglist geabonneerd is. Als laatste onderdeel van dit hoofdstuk leest u hoe u onder Linux een nieuwsserver kunt configureren.

Leerdoelen
* Werking van mailservers
* Aanpassen van een sendmail configuratie
* Mail versturen met sendmail
* Inkomende mail sorteren met Procmail
* Configuratie van de POP-daemon qpopper
* Configuratie van een Majordomo mailinglist
* Configuratie van Nieuwsservers op Linux.

2.1 Linux als Sendmail mailserver.
Als er een gebied is waarop voor het Linux-platform zeer verschillende implementaties beschikbaar zijn, is dat wel voor mail. Wellicht heeft dat er ook mee te maken dat Linux op internet nog steeds het meest gebruikte mailplatform is. In dit hoofdstuk wordt een introductie gegeven in Linux mail. Om te beginnen leest u wat er überhaupt moet gebeuren om mail te kunnen versturen en ontvangen. Bij het configureren van een mailserver komt namelijk meer kijken dan alleen een programma te configureren. Na de algemene introductie van mail op een netwerk, leert u hoe u de sendmail mailserver aan kunt passen aan eigen behoeften. Vervolgens leest u hoe u een server kunt configureren voor de afhandeling van POP-mail op uw netwerk.

2.1.1 Introductie in het werken met mailservers
Voordat u begint met de configuratie en installatie van een Linux-mailserver, is het goed even op een rij te zetten wat er precies bij komt kijken om een mailserver in het leven te roepen. In deze inleiding kunt u lezen uit welke componenten elke mailoplossing bestaat, ongeacht of deze nu geïnstalleerd is op een Linux of op een Windows server.

Het eerste onderdeel dat nodig is in een mailserver, is het onderdeel dat ervoor zorgt dat de mailberichten verstuurd kunnen worden. Hiervoor wordt in de meeste gevallen gebruikgemaakt van het Simple Message Transfer Protocol (SMTP). Een SMTP-server is een server die het mailadres van de geadresseerde leest en op basis van dit mailadres het bericht bij de beoogde ontvanger aflevert. Hiervoor kan een apart geconfigureerde server gebruikt worden, het is ook mogelijk voor dit doel gebruik te maken van een proces dat op uw eigen computer actief is.

Om ervoor te zorgen dat een mailbericht inderdaad bij de ontvanger kan worden afgeleverd, moet de mailserver in staat zijn de naam van de mailserver van de geadresseerde te achterhalen. Als u bijvoorbeeld een bericht wilt versturen naar mail@uwdomein.nl, moet de sendmail-mailserver weten op welke computer hij dit bericht moet afleveren. Hierbij komt

DNS om de hoek kijken. In het volgende hoofdstuk van dit boek leest u meer over de configuratie van DNS. Om wat preciezer te zijn: in het geval dat een bericht verstuurd moet worden naar een gebruiker in het domein sandervanvugt.nl, moet het MX-record waarin de informatie staat welke server verantwoordelijk is voor mailafhandeling op het betreffende domein voor het betreffende domein achterhaald worden. Op basis van dit MX-record wordt bekend wat de verantwoordelijke mailserver van het domein in kwestie is en hierdoor kan het bericht worden afgeleverd.

Wanneer u een mailbericht stuurt naar een gebruiker in een bepaalde organisatie, wordt dit bericht ontvangen door de mailserver van die organisatie. Die kan er in principe twee dingen mee doen. Om te beginnen kan hij het direct doorsturen naar de computer waarop de eindgebruiker in kwestie op dat moment is aangemeld. Hiervoor moet echter wel aan een belangrijke voorwaarde voldaan worden: de gebruiker in kwestie moet op dat moment ook echt zijn aangemeld. Aangezien dit in veel gevallen niet zo zal zijn, is het fenomeen van de ontvangende mailserver verzonnen. Dit is een mailserver die alle berichten voor zijn gebruikers ontvangt en bewaart. Wanneer de gebruiker daar vervolgens een keer zin in heeft, kan hij deze mailserver benaderen om zijn mail er van binnen te halen. Dit laatste kan algemeen gesproken op twee manieren: door middel van het Post Office Protocol (POP) of door middel van IMAP. Gebruik van IMAP heeft als voordeel dat de berichten niet eerst volledig binnengehaald hoeven worden voordat de eindgebruiker er iets mee kan doen. Vanwege zijn grotere compatibiliteit zult u echter wanneer gebruikgemaakt wordt van internet mail in de meeste gevallen te maken hebben met POP-mailservers.

2.1.2 Sendmail

Het gerucht gaat dat de maker van Sendmail een product heeft geprogrammeerd dat zó complex is, dat hij uiteindelijk in een gesticht terecht is gekomen. Hoewel dit gerucht absoluut niet waar is, is wel te begrijpen hoe het komt dat een dergelijke mythe ooit ontstaan is (waarschijnlijk omdat het product mening beheerder topt absolute wanhoop heeft gedreven). Het basis configuratiebestand van sendmail (sendmail.cf) is namelijk zo complex dat ook zeer ervaren beheerders er geen touw aan vast kunnen knopen.

***sendmailcf.tif De syntaxis van sendmail.cf is erg moeilijk te doorgronden.

Toch gaan wij een poging wagen u in te wijden op mogelijke configuraties van deze mailserver. Sendmail is nog steeds een van de meest populaire mailservers die op Linux gebruikt wordt. De meeste mailservers die op internet in gebruik zijn, maken ondanks de complexiteit van de configuratie toch gebruik van deze server. Vrijwel elke Linux distributie wordt geleverd met een versie van Sendmail. Deze kan ingezet worden om de berichtenstroom voor een volledig bedrijf af te handelen, er kan echter ook gebruik van gemaakt worden om alleen lokale mailberichten te versturen.

De vraag is gerechtigd waarom iemand zich zou verdiepen in een ingewikkelde mailserver als sendmail terwijl andere servers zo veel eenvoudiger zijn. Toch is er een goede reden voor Sendmail te kiezen. De voornaamste hiervan is dat Sendmail Open Source software is. Dit betekent dat de Sendmail mailserver heel eenvoudig aan te passen aan al uw eigen behoeften. Daarnaast vermijd u door het gebruik van Open Source software de dure licentiekosten die aan de meeste andere commerciële mailservers verbonden zijn. Daarnaast is een laatste goede reden gebruik te maken van Sendmail de performance van deze server. Vooral in omgevingen waar grote hoeveelheden mail verstuurd moeten worden presteert de Sendmail mailserver uitstekend. Er zijn andere mailservers die voor het Linux platform gebruikt kunnen worden;

in veel gevallen zijn deze andere mailservers zelfs ook nog veel meer gebruikersvriendelijk. Ze hebben echter wel een nadeel en dat is dat juist voor die gebruikersvriendelijke mailservers vaak behoorlijk betaald moet worden. Om die reden geven veel beheerders toch de voorkeur aan Sendmail.

2.1.3 Componenten van mailoplossing

Voor het realiseren van een totale infrastructuur om mailberichten te versturen is Sendmail alleen niet voldoende. In een mailsysteem wordt doorgaans gebruik gemaakt van meerdere componenten die met elkaar samen werken om berichten zo efficiënt mogelijk te verhandelen. Om te beginnen is er de MTA. Dit is de message transfer agent. Het doel van deze componenten is mailberichten tussen verschillende servers te versturen. Daarnaast is er de MUA, de mail user agent. Dit is het component waar gebruikers gebruik van maken om hun berichten binnen te halen. Tot slot is er de MDA, de mail delivery agent. Dit component zorgt ervoor dat mailberichten van de server naar het account van een gebruiker verstuurd worden zodat ze uiteindelijk in zijn inbox terecht komen. Voor een complete oplossing moeten al deze drie componenten voor een gebruiker beschikbaar zijn.

Component	Toepassing	Voorbeeld
MTA	Versturen van berichten tussen servers	Sendmail, postfix, eximl
MUA	Programma dat door eindgebruiker gebruikt wordt om berichten binnen te halen en versturen	Outlook, pine, elm
MDA	Zorgt ervoor dat ontvangen berichten afgeleverd worden in de mailbox van de gebruiker	Procmail

Tabel 1. Overzicht componenten mail-infrastructuur

Wanneer u met sendmail aan het werk gaat, moet u configuratiebestanden bewerken. Er zijn meerdere bestanden die u in de gaten moet houden, het grootste deel van de configuratie beperkt zich echter tot een drietal bestanden. Om te beginnen is er het hoofdbestand /etc/sendmail.cf. Daarnaast wordt gebruik gemaakt van het hulpbestand /etc/sendmail.cw. Hierin wordt een lijst bijgehouden van de computers in het domein waarvoor mail geaccepteerd wordt. Tot slot is er het bestand /etc/aliases. Hierin wordt een lijst van mailadressen van gebruikers bijgehouden. Ook kunnen in het aliasbestand mailgroepen gedefinieerd worden en kunnen andere namen voor gebruikers worden opgegeven.

2.1.4 Aanpassen van sendmail.cf

In de inleiding van deze paragraaf hebt u een voorbeeld kunnen zien hoe het sendmail configuratiebestand er normaliter uit ziet. Het aantal regels in dit bestand kan enorm oplopen: meer dan duizend is geen uitzondering! Er zijn echter maar weinig beheerders die de regels in dit bestand ook direct beheren. De meeste mensen kennen maar één regel uit dit bestand en dat is over het algemeen ook meer dan genoeg. In de “Smart Host” definitie wordt aangegeven welke host in staat is de berichten voor deze server af te handelen. Dit scenario doet zich voor wanneer u de mailberichten doorstuurt naar een internetaanbieder. Uw server zorgt er dan niet zelf voor dat de berichten afgeleverd worden, maar de internetaanbieder doet dit voor u. Om duidelijk te maken welke server voor dit doel gebruikt moet worden, gebruikt u de Smarthost regel. Deze ziet er ongeveer als volgt uit:

```
#       "Smart" relay host (may be null)
DSmailserver.domein.nl
```

U kunt overigens ook de definitie van deze smarthost op een andere manier regelen, hierover leest u verderop meer.

De eerste regel die u in het bovenstaande voorbeeld ziet is commentaar; het #-teken zorgt ervoor dat de regel niet geïnterpreteerd wordt. Dergelijke regels dienen puur voor het vergroten van de leesbaarheid voor uzelf. Op de tweede regel wordt in het voorgaande voorbeeld aangegeven welke computer verantwoordelijk is voor de afhandeling van mailberichten. De letters "DS" geven aan dat op deze regel een smart host gedefinieerd wordt, direct achter deze aanduiding volgt de volledig naam van de betreffende host. Houd er rekening mee dat deze naam niet altijd voorkomt, u gebruikt hem alleen wanneer u uw eigen server niet direct mail wilt laten versturen. Het is dus een configuratieregel die typisch voorkomt op de computer van een eindgebruiker.

Om deze regel en eventueel ook andere instellingen aan te passen, kunt u zich in de ondoorgrondelijke syntaxis van sendmail.cf verdiepen. Dit is echter niet gebruikelijk. Meestal wordt voor het aanpassen van Sendmail-parameters gebruikgemaakt van een speciaal macrobestand. In dit macrobestand worden alle instellingen in een min-of-meer leesbare vorm gedefinieerd. Deze macro wordt vervolgens met behulp van de opdracht **m4** omgezet tot het sendmail.cf bestand. In jargon staat dit commando overigens ook bekend als de "preprocessor". Het voordeel van deze werkwijze? U hoeft zich zelf helemaal niet meer te verdiepen in de complexe opbouw van sendmail.cf. Het macro-bestand dat door m4 verwerkt wordt, bestaat uit een aantal commando's dat is opgesteld volgens de m4-syntaxis. In deze bestanden wordt gebruik gemaakt van een aantal veelgebruikte m4-macro's. In de onderstaande tabel treft u hier een overzicht van.

Macro	**Functie**
Define	Definieert een macro met een bepaald argument
Undefine	Heft de definitie van een eerder gebruikte macro weer op
Include	Zorgt er voor dat het bestand als argument gegeven wordt ook als macro-bestand verwerkt wordt.
Dnl	Commentaar teken; zorgt ervoor dat alle overige tekens op de regel niet geïnterpreteerd worden. U kunt hier gebruik van maken om de leesbaarheid van een bestand te vergroten.
Divert	Beheert uitvoer
Feature	Maakt het gebruik van andere features mogelijk

In het volgende ziet u een voorbeeld hoe dit bestand er uit zou kunnen zien. De meeste distributies hebben hun eigen oplossing voor de locatie van dit bestand, ze komen bijvoorbeeld regelmatig voor in /usr/share/sendmail-cf, maar dit kan per distributie afwijkend zijn. Houd er ook rekening mee dat er door middel van de include-macro regelmatig verwezen wordt naar een macro op een andere locatie die ook verwerkt moet worden.

```
divert(-1)
dnl This is the macro config file used to generate the /etc/sendmail.cf
dnl file. If you modify the file you will have to regenerate the
dnl /etc/sendmail.cf by running this macro config through the m4 preprocessor
```

```
dnl
dnl                m4 /etc/sendmail.mc > /etc/sendmail.cf
dnl
dnl You will need to have the sendmail-cf package installed for this to
dnl work
include(`../m4/cf.m4')
define(`confDEF_USER_ID'. ``8:12'')
OSTYPE(`linux')
undefine(`UUCP_RELAY')
undefined(`BITNET_RELAY')
define(`confAUTO_REBUILD')
define(`confTO_CONNECT', `1m')
define(`confTRY_NULL_MX_LIST',true)
define(`confDONT_PROBE_INTERFACES',true)
define(`PROCMAIL_MAILER_PATH',`/usr/bin/procmail')
FEATURE(`smrsh',`/usr/sbin/smrsh')
FEATURE(`mailertable',`hash -o /etc/mail/mailertable')
FEATURE(`virtusertable',`hash -o /etc/mail/virtusertable')
FEATURE(redirect)
FEATURE(always_add_domain)
FEATURE(use_cs_file)
FEATURE(local_procmail)
MAILER(procmail)
MAILER(smtp)
FEATURE(`access_db')
FEATURE(`blacklist_recipients')
dnl We strongly recommend to comment this one out if you want to protect
dnl yourself from spam. However, the laptop and users on that do
dnl not have 24x7 DNS do need this.
FEATURE(`accept_unresolved_domains')
dnl FEATURE(`relay_based_on_MX')
```

Dit bovenstaande macro-bestand dat ontleend is aan een standaard Fedora-configuratie, is met behulp van **m4** omgezet in een werkend sendmail.cf bestand dat uit een totaal van meer dan 1200 regels bestaat. Dit is gebeurd met de opdracht **m4 redhat.mc > /etc/sendmail.cf**
We zullen nu de belangrijkste regels uit dit bestand bespreken.
Om te beginnen zorgt de opdracht divert (-l) er voor dat alle overbodige informatie uit het resulterende configuratiebestand verwijderd wordt. Vervolgens wordt er een configuratiebestand met extra instellingen aangeroepen met behulp van de regel include(`..m4/cf.m4'). Let er uiteraard wel op wanneer u het bovenstaande bestand ooit uit wilt proberen dat het bestand waarnaar verwezen wordt, wel moet bestaan. Het is overigens goed gebruik dat gebruik gemaakt wordt van veel meer dan één m4-bestand om de uiteindelijke sendmail.cf te genereren. Dit heeft het voordeel dat de configuratie die gebruikt wordt heel mooi modulair opgeslagen kan worden. Helaas heeft het ook een nadeel: het wordt er niet echt leesbaarder op wanneer op deze wijze gewerkt wordt.

Wanneer dit inleidende werk gebeurd is, moet aangegeven worden welk gebruikersaccount Sendmail moet gaan gebruiken. Dit is een belangrijke verwijzing: zorg ervoor dat hiervoor een gebruikersaccount met niet al te veel rechten gebruikt wordt! Anders zou via een foutje in de sendmail-software (en die komen nogal eens voor) iemand ongeoorloofd toegang kunnen

krijgen tot belangrijke bestanden. In de verwijzing naar het gebruikersaccount wordt gebruikgemaakt van een UID en bijbehorend GID zoals dat in de respectievelijke bestanden /etc/passwd en /etc/group gedefinieerd is. De regel define (`confDEF_USER_ID',``8:12'') zorgt hiervoor. Daarna wordt met OSTYPE(`linux') aangegeven op welk besturingssysteem dit bestand is uitgevoerd, sendmail kan namelijk op veel meer platforms dan alleen Linux gebruikt worden.. Heel veel andere instellingen kunnen namelijk op basis van deze instelling automatisch ingevuld worden. Niet zelden wordt er vervolgens een extra bestand aangeroepen met de naam Linux.mc waarin de instellingen die voor Linux specifiek zijn gedefinieerd worden.

Na deze instellingen volgt een aantal algemene instellingen die het functioneren van de mailserver verder definiëren. Om te beginnen wordt met undefine(`UUCP_RELAY') aangegeven dat deze host niet gebruikt mag worden voor relaying van UUCP-mail. Vervolgens gebeurt met undefine(`BITNET_RELAY') hetzelfde voor Bitnet-mail. Daarna wordt geboden dat aliassen automatisch opnieuw opgebouwd moeten worden door de mailserver. Hiervoor wordt gebruikgemaakt van het instellingenbestand /etc/alias. De regel define(`confAUTO_REBUILD') draagt hier zorg voor. Dan wordt de time-out voor het opbouwen van een connectie ingesteld op een minuut met de regel define(`confTO_CONNECT', `1m')

Vervolgens wordt bepaald hoe er gecommuniceerd moet worden met een host die uw mailserver heeft ingesteld als beste mailhandler. De mailhandler is de server die door middel van het MX-record in de DNS-database staat ingesteld als mailserver die voor een bepaald domein gebruikt moet worden. Met behulp van de regel define(`confTRY_NULL_MX_LIST',true)  wordt bepaald dat berichten voor dergelijke hosts gewoon doorgestuurd mogen worden door uw mailserver. Daarna wordt door de regel define(`confDONT_PROBE_INTERFACES',true) bepaald dat namen en adressen van lokale interfaces niet doorgegeven worden door deze server. Daarna wordt aangegeven waar het programma procmail zich bevindt dat gebruikt wordt om berichten af te leveren in de mailbox van een gebruiker. Dit gebeurt met behulp van de instelling define(`PROCMAIL_MAILER_PATH',`/usr/bin/procmail').

Daarna wordt opgegeven welke Shell sendmail moet gebruiken wanneer er mailberichten doorgestuurd moeten worden naar bepaalde programma's. Als hier niets staat ingesteld, probeert de opdracht contact te maken met behulp van /bin/sh. Dit is echter een onveilige Shell en in het ergste geval zou een hacker met behulp van deze Shell toegang kunnen krijgen tot de rest van het systeem. Daarom wordt de regel FEATURE(`smrsh',`/usr/sbin/smrsh') gebruikt om aan te geven dat gebruikgemaakt moet worden van de speciale sendmail restricted Shell. Dit is een Shell waaruit wel de voor sendmail noodzakelijke commando's uitgevoerd kunnen worden, maar die geen toegang geeft tot de rest van de computer.

Vervolgens komt er een tweetal verwijzingen naar bestanden waarin extra instellingen gedaan worden. Om te beginnen is dat de regel FEATURE(`mailertable'.`hash -o /etc/mail/mailertable'). Hiermee verwijst u naar een bestand met de naam mailertable dat gebruikt kan worden om mailberichten op een afwijkende manier te routeren. Daarna wordt met de regel FEATURE(`virtusertable',`hash -o /etc/mail/virtusertable') verwezen naar een bestand waarin u virtuele domeinen aan kunt maken. Dit is handig wanneer uw mailserver meerdere maildomeinen moet ondersteunen. Deze maildomeinen kunnen dan in /etc/mail/virtusertable gedefinieerd worden.

Vervolgens zorgt het redirect-feature er voor dat een foutmelding verstuurd wordt wanneer mail naar een bepaald adres verstuurd wordt. De regel die hiervoor gebruikt wordt is FEATURE(redirect). Daarna wordt bepaald dat bij het versturen van mail altijd het maildomein wordt toegevoegd. Het is vanzelfsprekend dat dit handig is wanneer er buiten uw domein gemaild wordt; anders kan de ontvanger van uw mail immers niet zien van wie een bericht afkomstig is. Met de regel FEATURE(always_add_domain) zorgt u ervoor dat dit ook gebeurt wanneer er binnen het lokale domein berichten verstuurd worden.

Daarna wordt een aantal opties ingesteld waarmee u bepaalt hoe berichten lokaal afgehandeld wordt. Om te beginnen wordt met FEATURE(use_cw_file) aangegeven dat gebruikgemaakt kan worden van een lokaal bestand met de naam sendmail.cw. Hiermee kunnen instellingen in het algemene sendmail.cf overschreven worden. Vervolgens wordt met de regel MAILER(procmail) bepaald dat procmail gebruikt moet worden om mailberichten in de inbox van de gebruiker af te leveren. Daarna wordt met de regel MAILER(smtp) aangegeven dat SMTP-mailserver functionaliteit gebruikt moet worden.

De laatste regels in dit configuratiebestand hebben allen te maken met beveiliging van uw systeem en het voorkomen van misbruik ervan. Om te beginnen is er de regel FEATURE(`access_db'). Hiermee kan verwezen worden naar een database waarin een lijst met domeinen bijgehouden wordt waarvoor u mail wilt accepteren of juist wilt verwerpen. Iets soortgelijks kunt u voor elkaar krijgen met de regel FEATURE(`blacklist_recipients'). Hiermee kunt u mail tegenhouden op basis van gebruikersnaam, hostnaam of adres van de afzender. Vervolgens is er de regel waarmee u aan kunt geven of u mail wilt accepteren van hosts waarvan de domein naam niet door middel van DNS achterhaald kan worden. In principe is het niet verstandig dit te doen, wanneer een hostnaam namelijk niet door middel van DNS achterhaald kan worden, is de kans groot dat u te maken hebt met een hacker. Mocht u dit toch willen, dan kunt u het statement FEATURE(`accept_unresolvable_domains') op kunnen nemen.

2.1.5 Client en server configuraties

In het bovenstaande is een standaard sendmail configuratiebestand besproken zoals dat op Red Hat Linux voorkomt. In dit bestand wordt niet echt duidelijk welke taken nu op welk type computer uitgevoerd moeten worden. Daarom geven we hier nog een voorbeeld. Om te beginnen leest u het configuratiebestand zoals dat op een werkstation voorkomt. Het belangrijkste dat hier moet gebeuren, is dat de mail doorgestuurd wordt naar een mailserver. Dit gebeurt door middel van de regel define(`SMART_HOST`,`poffice.mydomain.com.`)

Dit zorgt ervoor dat alles wat niet expliciet een lokale bestemming heeft, wordt doorgestuurd naar de mailserver. We nemen ook de andere configuratieregels op, zodat u een compleet voorbeeld te zien krijgt.

```
divert(-1)
divert(0)
VERSIONID(`Config file for Red Hat Linux`)
OSTYPE(`Linux`)
FEATURE(`smrsh`)
define(`PROCMAIL_MAILER_PATH`,`/usr/bin/procmail`)
FEATURE(`local_procmail`)
define(`SMART_HOST`,`poffice.mydomain.com. `)
define(`STATUS_FILE`,`/var/log/mail.stats`)
```

MAILER(`smtp`)
MAILER(`procmail`)

We hebben hierin te maken met macro's, zo definieert OSTYPE een macro die er op zijn beurt voor zorgt dat een Linux specifiek configuratiebestand wordt aangeroepen. Deze bestanden komen voor in de directory ../ostype. In de eerste FEATURE-macro wordt gedefinieerd welk Shell-programma door sendmail gebruikt wordt; zoals gebruikelijk wordt gebruikgemaakt van de restricted Sendmail Shell (smrsh). Vervolgens wordt aangegeven dat procmail de local procmail delivery-agent is en waar dit programma gevonden kan worden. Dan krijgen we de definitie van de outgoing mail hub voor dit domein gevolgd door de locatie van het logbestand. Tot slot wordt aangegeven dat de SMTP en procmail delivery-agents op dit systeem in gebruik zijn. Een erg handige macro die gebruikt zou kunnen worden is DOMAIN(`generic'). Deze wordt gebruikt om te verwijzen naar een algemeen te gebruiken configuratiebestand zodat alle machines in een netwerk dezelfde configuratie hebben.

Bovenstaande is alleen maar een lokaal bestand voor eindnodes, de configuratie op de centrale mailserver komt er iets anders uit te zien.

FEATURE(`use_cw_file')
Dnl Send out all mail as username@mydomain.com
MASQUERADE_AS(`mydomain.com')
FEATURE(`masquerade_envelope')
FEATURE(`allmasquerade')
MAILER(`SMTP')
MAILER(`local')

We beginnen met een verwijzing naar het zogenaamde cw-file: /etc/mail/local-host-names . In dit bestand wordt verwezen naar hosts die van deze hub gebruik mogen maken. Let er op dat in dit bestand slechts één hostnaam per regel opgenomen mag worden. De regel die begint met "dnl" is een commentaar regel. Daarna zorgen de drie regels masquerade ervoor dat alle locale gebruikersnamen zoals user@host vertaald worden in meer gangbare namen als user@domain.com. Pleunie@localhost wordt in zo'n geval dus Pleunie@qwerty.com. Daarnaast zou u trouwens ook ALWAYS_ADD_DOMAIN kunnen gebruiken om een bepaald domein automatisch aan gebruikersnamen toe te voegen, in dit geval zorgt dat ervoor dat ook mailberichten met een lokale bestemming een internet domain suffix krijgen. In de laatste regel tenslotte worden de lokale delivery agents gespecificeerd.

In de MAILER-regels begint u altijd met de preferred mailer, daarna definieert u wat u in geval van problemen wilt gebruiken. Wanneer er MAILER(local) staat, betekent dat op Linux meestal dat gebruik gemaakt wordt van procmail. Deze zorgt ervoor dat de berichten worden afgeleverd in de mailboxen van gebruikers die op het systeem voorkomen. In principe zijn dat de directories die voorkomen onder /var/spool/mail.

2.1.6 Aliassen en doorsturen van mail

Standaard is er maar één naam verbonden aan een mailbox. Het kan soms echter handig zijn om meerdere namen aan één mailbox te verbinden. Zo kunt u er bijvoorbeeld voor zorgen dat alle mail die gericht is aan "verkoop@uwbedrijf.com" direct wordt doorgestuurd naar een medewerker van de betreffende afdeling. Als u het echt complex wilt maken, kunnen aliassen ook recursief worden toegepast. Dit betekent dat u de ene alias laat verwijzen naar een andere alias. Tevens is het mogelijk om aliassen aan te maken waarmee verwezen wordt naar een

programma dat uitgevoerd moet worden. Wanneer u wijzigingen aanbrengt in het alias-bestand, moet u zich realiseren dat deze wijzigingen niet meteen ook doorgevoerd worden. Om nieuwe aliassen te activeren, moet u eerst de opdracht **newalias** gebruiken.

Naast de mogelijkheden die de gebruiker root heeft om mail die gericht is aan een bepaald account door te sturen naar een ander account, hebben gebruikers de mogelijkheid zelf hun eigen alias bestand aan te maken. Dit bestand heeft dan de naam .forward en bevindt zich in de homedirectory van de gebruiker. Hierin kunnen gebruikers in een syntaxis die in grote lijnen gelijk is aan die van het bestand aliases aangeven dat mail doorgestuurd moet worden naar een bepaald account. Zo kunt u er bijvoorbeeld voor zorgen dat mail die u tijdens uw vakantie ontvangt, wordt doorgestuurd naar uw privé account. Als extra beveiligingscontrole controleert sendmail of dit bestand alleen beschrijfbaar is door de eigenaar ervan. Dit om misbruik te voorkomen en er voor te zorgen dat andere niet ongewenst uw mail door kunnen laten sturen.

In het onderstaande ziet u een voorbeeld van een aantal regels uit /etc/aliases. Dit voorbeeld is ontleend aan SUSE Linux.

```
# De volgende regel zorgt ervoor dat mail die gericht is aan de gebruiker root
# wordt doorgestuurd naar een gewone gebruiker. De constructie \root zorgt
# er voor dat de berichten echter ook doorgestuurd worden naar de gebruiker root.
root:           joe, \root

# nu komen er twee regels die altijd aanwezig moeten zijn
postmaster:     root
mailer-daemon:  root

# amavis
virusalert:     root

#redirection voor verschillende systeem accounts
bin:            root
daemon:         root
games:          root
ingres:         root
nobody:         root
system:         root
toor:           root
uucp:           root

#gebruikers waar mail voor root naar doorgestuurd wordt:
root:           linda, melissa

#doorsturen naar een ander domein:
Alex:           alex@xyqw.nl

#groepen: let op de spaties!
Boekhouders:    ad, paul, eric, marco
```

2.1.7 Starten en stoppen van sendmail

Er zijn twee verschillende manieren waarop sendmail beheerd kan worden. Oorspronkelijk werd deze service vanuit het inetd-mechanisme gestart. Dit betekent dat inetd als daemon luisterde naar calls die het sendmail proces aanspreken en vervolgens sendmail gaat starten. Daarbij kan bovendien nog gebruikgemaakt worden van de beveiliging door middel van instellingen in de bestanden hosts.allow en hosts.deny. Moderne distributies gaan er echter steeds vaker toe over om sendmail te laten starten door middel van een opstartbestand dat zich bevindt in de directory /etc/init.d (soms /etc/rc.d/init.d). Het script waarmee u sendmail kunt starten heet in dat geval dus /etc/init.d/sendmail. Het voordeel van het werken met zo'n script, is dat u meer controle hebt over het sendmail-proces. Dit script kan namelijk altijd gestart worden met de argumenten start, stop, restart en status.

Dank zij deze argumenten hebt u meer zeggenschap over de wijze waarop het sendmail-proces zich gedraagt. Bovendien kunt u er voor zorgen dat sendmail automatisch geactiveerd wordt bij het betreden van het standaard runlevel. Wanneer u een kijkje neemt in dit script, wordt duidelijk op welke wijze sendmail gestart wordt. Op de meeste Linux systemen is dat in de "stand-alone" modus. Het commando **sendmail -bd -q30m** zorgt ervoor dat sendmail als background daemon (bd) gestart wordt en dat de mailqueue elke dertig minuten gecontroleerd wordt om te kijken of er uitgaande mail klaar staat.

Een volgende taak die als beheerder van sendmail handig is om periodiek uit te voeren, is het opvragen van het versienummer dat op dit moment gebruikt wordt. Geef hiervoor de opdracht **echo | sendmail -bt -d0**. Als alternatief kan u ook gewoon telnetten naar de sendmail-poort, ook hiermee kan u het gebruikte versienummer opvragen.

***seninfo.tif Met de optie –bt –d0 kunt u informatie opvragen over de versie van Sendmail die gebruikt wordt en de modules waarmee Sendmail gecompileerd is.

Een laatste mogelijkheid die we hier willen noemen om sendmail te beheren, is de opdracht **mailq**. Hiermee kunt u naar de inhoud van de mail queue kijken. Dit is handig wanneer u bijvoorbeeld om wat voor reden dan ook de mail queue van een gebruiker op moet schonen. De output van dit commando toont u om te beginnen de interne identificatie van het mailbericht. In het onderstaande voorbeeld ziet dit er uit als SAA04789. Daarna wordt de grootte van het mailbericht getoond. Dit wordt gevolgd door de datum waarop het bericht in de queue geplaatst is en tot slot ziet u de naam van de gebruiker die het bericht in de queue geplaatst heeft. Vindt u dat de opdracht met zijn standaard output nog niet voldoende laat zien? Geef dan de optie -v (verbose) mee voor meer informatie.

```
Linux # mailq
                    Mail queue (2 requests)
--Q-ID-- --Size-- --Priority-- --Q-Time-- ----- -----Sender/Recipiënt----- -----
SAA04789     5        30026  Sep 19 19:58  root
                                            sander
SAA04790     9        30025  Sep 19 19:53  root
                                         sander
```

2.1.8 Voorbeeld gebruik Sendmail

Wij kunnen ons voorstellen dat u door de bomen het bos niet meer ziet. Dat is jammer, want als u uitgaat van een eenvoudige sendmail-configuratie, is het eigenlijk extreem eenvoudig ervoor te zorgen dat u met deze mailserver aan het werk kunt. Aan de andere kant is het ook heel kenmerkend voor Sendmail dat u verdwaalt in alle mogelijkheden, dat gebeurt ervaren

beheerders ook nog regelmatig. De eenvoud van Sendmail wordt gedemonstreerd door een eenvoudig voorbeeld met behulp van de aloude mailclient die op elk Linux systeem aanwezig is: de opdracht **mail**. We laten u zien hoe u met dit commando berichten kunt versturen zonder dat er ook maar iets aan uw sendmail mailserver geconfigureerd is. Het mooie van Sendmail is namelijk dat hij na opstarten spontaan begint te doen waar hij voor geschreven is: mail verzenden. Ook zult u zien dat elk ander mailprogramma dat met uw distributie geleverd wordt heel eenvoudig in staat is deze mail binnen te halen. Tot slot leggen we uit dat het versturen van mailberichten naar internet ook een fluitje van een cent is waarvoor alles eigenlijk al geregeld is. Het enige dat niet geregeld is, is het ontvangen van mail. Hiervoor moet u namelijk actie ondernemen en ervoor zorgen dat een ontvangende mailserver voor internetmail in uw domein bekend is.

De eerste stap om te kijken of sendmail het goed doet, is eerst eens een telnet-sessie te openen naar poort 25 op uw server. Hiermee kunt u controleren of sendmail actief is en voor u staat te wachten om berichten te gaan versturen. Is dit niet het geval? Start dan sendmail met de opdracht **sendmail –bd –q3m**.

***telnet25.tifDoor te telnetten op een bepaalde poort, kunt u snel achterhalen of op die poort een service actief is.

Draait sendmail? Dan wordt het tijd voor een eenvoudige test. In dit voorbeeld gaan we ervan uit dat er een gebruiker sander is die een bericht wil versturen aan Linda. Om dit voor elkaar te krijgen, moeten uiteraard wel accounts bestaan voor beide gebruikers. Om deze eenvoudige test te beginnen, moet sander eerst inloggen. Vervolgens opent hij een console-venster en geeft daar de opdracht **mail Linda**. Dit zorgt ervoor dat een mail-prompt geopend wordt waar sander alle mail-commando's in moet typen. Om te beginnen geeft hij aan wat het onderwerp van de mail is, daarna kan hij de body van het bericht typen. Als dit gebeurd is, kan het verzenden van het mailbericht worden afgesloten door op een afzonderlijke regel een punt te geven. In de onderstaande listing ziet u precies wat er allemaal moet gebeuren.

```
Linux: $ mail Linda
Subject: Hallo Linda
Hoi Linda
Dit is een testbericht.
.
EOT
Linux: $
```

Wilt u als beheerder nog controleren dat alles goed gegaan is? Dan kunt u nu de opdracht **mailq** gebruiken om de inhoud van de mailqueue te bekijken. Let op, het is heel goed mogelijk dat u helemaal geen inhoud te zien krijgt; dat betekent dan namelijk dat sendmail de mailqueue al leeggehaald heeft en het bericht reeds verzonden heeft. Over het algemeen gebeurt dit namelijk vrij snel. Eigenlijk mag u er van uit gaan dat het een goed teken is wanneer u niets in de mailqueue ziet staan.
Nu heeft Sander zijn werk gedaan en is het bericht als het goed is aangekomen bij Linda. Deze kan het bericht nu lezen door gebruik te maken van het traditionele UNIX-mailcommando **mail**. Wanneer zij dit commando zonder enige argumenten geeft, wordt meteen de inhoud van haar mailbox getoond. Deze mailbox bestaat overigens op het Linux-systeem in de vorm van een bestand met de naam /var/mail/Linda. Dit is een ASCII-bestand waar keurig alle mailberichten op een rij staan, maar waar Linda de enige is die voldoende

permissies heeft om het uit te lezen. Het is immers niet de bedoeling dat iedereen zomaar bij de mailbox van elke willekeurige gebruiker kan komen. Om een van de berichten in haar mailbox te lezen, geeft Linda het nummer van het bericht in kwestie, gevolgd door Enter. Om te stoppen met lezen van de berichten, gebruikt ze de opdracht q die ervoor zorgt dat de mailbox weer gesloten wordt.

***mailbox Elke gebruiker kan met de opdracht mail zijn binnengekomen berichten bekijken.

Het commando **mail** is natuurlijk het meest eenvoudige commando om snel als gebruiker je mailbox leeg te halen, er kan voor dit doel echter ook gebruikgemaakt worden van een ander programma. Veel hedendaagse gebruikers zullen namelijk niet overweg kunnen met de eenvoudige karakter-georiënteerde interface van mail. Denk als alternatief aan fraaie grafische mailers zoals Kmail. Om dit programma voor het ontvangen van mailberichten te configureren, moet in het menu Instellingen gekozen worden voor de optie KMail instellen. Selecteer nu in het kader links de optie Identiteiten en klik op Nieuw om een nieuw mailaccount aan te maken. Geef het nieuwe mailaccount een naam en druk vervolgens op OK. Het echt belangrijke werk gebeurt vervolgens onder de optie Netwerk. Hier specificeert u namelijk op welke wijze het mailaccount berichten moet verzenden en ontvangen. Als standaardoptie staat hier sendmail al genoemd als de manier om berichten te versturen. Dat kan ook wanneer de gebruiker in kwestie gebruik wil maken van de lokale sendmail-daemon. Als alternatief zou hier de optie SMTP gebruikt kunnen worden om te verwijzen naar een mailserverproces dat op een andere computer wordt aangeboden.

***sendmail20 Bij het aanmaken van een gebruikersaccount in KMail wordt er standaard al vanuit gegaan dat gebruikgemaakt wordt van de lokale sendmail-daemon.

Klik nu op het tabblad Ontvangen. U ziet dan dat ook hier de instellingen gewoon al goed staan. Er wordt gebruikgemaakt van de lokale mailserver om het bericht binnen te halen en aangezien onze gebruiker lokaal op dit systeem inlogt en aanwezig is, is dat geen probleem. Zou de gebruiker daarentegen op een andere computer werken, dan is het wel een probleem en moet u ervoor zorgen dat er een pop-server aanwezig is die het mogelijk maakt dat de gebruiker het POP-protocol gebruikt om zijn mailbox leeg te halen.
Leuk natuurlijk dat u er nu voor gezorgd hebt dat gebruikers die op hetzelfde systeem werken met elkaar kunnen communiceren, maar wellicht wilt u er ook voor zorgen dat de gebruikers van uw server kunnen communiceren met andere gebruikers op internet. Hieraan zijn twee aspecten verbonden. U wilt er waarschijnlijk voor zorgen dat gebruikers berichten kunnen versturen en daarnaast wilt u er ook voor zorgen dat gebruikers berichten kunnen ontvangen. Voor het eerste is eigenlijk maar één ding nodig. U moet regelen dat u gebruikmaakt van een valide DNS-account. Veel mailservers op internet doen namelijk een controle of de berichten die zij binnenkrijgen afkomstig zijn van een valide internet mailserver. Deze controle vindt plaats door te kijken of het DNS-domein waaruit het bericht afkomstig is wel bestaat. Als dit niet het geval is, krijgt u van de mailserver van de geadresseerde een vriendelijke foutmelding.

***weldns20 Wanneer het mailbericht verzonden wordt vanuit een niet-bestaand DNS-domein, wordt het door de meeste mailservers op internet geweigerd.

Als volgende wilt u er nu voor zorgen dat u ook mail kunt ontvangen. Eigenlijk is dit heel eenvoudig: u moet ervoor zorgen dat sendmail vanaf het internet bereikt kan worden.

Hiervoor moet u twee dingen doen. Om te beginnen zorgt u ervoor dat de router toegang geeft tot de sendmail-poort, daarnaast moet u ervoor zorgen dat het internet weet waar het moet zijn om berichten bij u af te leveren. Dit laatste doet u door een MX-record aan te maken in de DNS-database. U kunt over dit onderwerp meer lezen in het volgende hoofdstuk.

U hebt nu een werkende mailserver, gefeliciteerd. Voor een echt fraaie oplossing moeten er echter nog wel een paar dingen gebeuren. Om te beginnen kunt u ervoor zorgen dat de binnenkomende mail netjes gefilterd wordt zodat bijvoorbeeld spam-berichten automatisch doorgestuurd worden naar het null-device. Daarnaast is het in een netwerkomgeving aardig wanneer u een POP of IMAP-mailserver in het leven roept die de berichten bewaart voor gebruikers die op dat moment niet zijn aangemeld. Hierover leest u in het vervolg van dit hoofdstuk.

<<AANMAKEN KADER>>
Sendmail en Sendmail
Wellicht dat u na het lezen van het voorgaande denkt dat Sendmail handmatig configureren zo ongeveer het ergste is dat je als beheerder van een server kan overkomen. Dan hebt u nog gelijk ook! Gelukkig bestaan er genoeg handige hulpprogramma's waarmee Sendmail in een groot professioneel netwerk ingericht kan worden. Het Open Source product Sendmail heeft namelijk gewoon ook een commerciële broer. Meer informatie over deze commerciële broer vindt u op www.sendmail.com. Overweegt u om professionele mailtoepassingen te bouwen op basis van Sendmail? Dan raden wij u van harte aan om eens te kijken op www.sendmail.com naar de reeks producten die daar wordt aangeboden om het beheer van de Sendmail-mailomgeving een stuk dragelijker te maken.
<<EINDE KADER>>

2.1.9 Verwerken van inkomende mail met procmail

Wanneer u intensief gebruikmaakt van een mailsysteem, kan het de moeite waard zijn de binnenkomende mailberichten automatisch te sorteren. Zo zorgt u er bijvoorbeeld voor dat berichten van een bepaalde afzender automatisch in een speciaal voor deze afzender aangemaakt mapje geplaatst worden. Ook kunt u ervoor zorgen dat spam-berichten linea-recta doorgestuurd worden naar /dev/null zodat u ze in uw inbox nooit aan zult treffen. Het antwoord op dergelijke vragen is procmail. Met behulp van procmail kunt u zonder al te veel moeite een eenvoudig mailfilter definiëren. De werking van procmail is eenvoudig: het programma kijkt of er een bestand met de naam .procmailrc bestaat in de homedirectory van de betreffende gebruiker. In dit bestand kunnen namelijk regels gedefinieerd worden op basis waarvan inkomende mail verwerkt wordt.

Aan de hand van het onderstaande voorbeeld van de inhoud van het bestand .procmailrc kunt u kennismaken met de werking van procmail-filters:

```
:0
* ^TO_news@sandervanvugt\.nl
InteressanteBerichten

:0
* ^From:.*@bloodynerds\.nl
/dev/null
```

In dit voorbeeld worden twee regels gedefinieerd voor binnenkomende mail. Elk van deze regels begint met de constructie :0; hieraan kan het procmail-proces herkennen dat er een nieuwe definitie begint. In de eerste regel wordt gekeken naar berichten die verzonden zijn aan het mailadres news@sandervanvugt.nl. Let overigens op de manier waarop de punt in de DNS-naam vooraf gegaan wordt door een escape-teken (het dakje), dit wordt vereist door de Procmail syntaxis. Deze eerste regel is typisch voor lidmaatschap van een mailinglist; hierbij worden berichten verstuurd naar het adres van de mailinglist. Wanneer u een van deze berichten binnenkrijgt, wordt het automatisch doorgestuurd naar de mail "InteressanteBerichten" in uw mailbox.

De tweede regel in ons voorbeeld .procmailrc vervolgens is kenmerkend voor de afhandeling van ongewenste mailberichten. Deze regel gaat er van uit dat alle ongewenste berichten altijd afkomstig zijn vanuit het DNS-domein bloodynerds.nl. De oplossing is erg resoluut: al deze berichten worden regelrecht doorgestuurd naar het NULL-device. Dit zorgt ervoor dat de eindgebruiker er nooit meer iets van te zien krijgt. Uiteraard is een dergelijke configuratie alleen zinnig wanneer u echt zeker weet dat uit dit domein alleen maar ongewenste berichten komen, u ziet er namelijk nooit meer iets van terug.

2.1.10 Herdistributie van mail met qpopper

De POP-daemon qpopper wordt op de meeste Linux-distributies standaard geïnstalleerd. Daar is dan ook een goede reden voor, het is namelijk de meest gebruikte Linux POP-server. Ook de werking van een POP-server is relatief eenvoudig. Om gebruik te kunnen maken van een POP-server op uw systeem hebt u om te beginnen voor elke POP-gebruiker een geldig gebruikersaccount nodig. In principe is dat een Shell-account dat ook door gebruikers gebruikt kan worden om in te loggen op uw mailserver. Als u dit een onprettig idee vindt, kunt u in /etc/passwd de standaard-shell /bin/bash vervangen door bijvoorbeeld de opdracht /bin/passwd. Gebruikers kunnen dan niet langer inloggen, maar zijn wel in staat hun wachtwoord te wijzigen.

Wanneer u ervoor gezorgd hebt dat de gebruikersconfiguratie op de juiste wijze geregeld is, moet u regelen dat de POP-server automatisch gestart wordt. Dit kan door de server als daemon te starten met de opdracht /usr/sbin/popper, het is echter aan te raden in plaats daarvan gebruik te maken van het xinetd-mechanisme. In de meeste gevallen vindt u hier een kant-en-klaar qpopper configuratiebestand waarin u alleen nog maar hoeft aan te geven dat de qpopper server gestart moet worden. Doe dit door het bestand als in het onderstaande voorbeeld te wijzigen:

```
#
# qpopper – pop3 mail daemon
#
service pop3
{
        disable                 = no
        socket_type             = stream
        protocol                = tcp
        wait                    = no
        user                    = root
        server          = /usr/sbin/popper
        server_args             = -s
        flags                   = IPv4
```

}

Het laatste onderdeel van de qpopper-configuratie bestaat er tot slot uit ervoor te zorgen dat xinetd actief is, dit is namelijk niet altijd ook automatisch het geval. Controleer dit met de opdracht **ps aux | grep xinetd**. Als het niet actief is, kunt u het starten vanuit het System-V opstartmechanisme van uw server. Op SUSE Linux kunt u er bijvoorbeeld met de opdracht **chkconfig xinetd 2345** voor zorgen dat xinetd voortaan automatisch gestart wordt, op Fedora Linux kunt u hetzelfde doen met de Runlevel Editor. Wanneer dit geregeld is, luister qpopper voortaan naar binnenkomende connecties op de standaard POP3-poort 110. U kunt nu vanaf elke computer die verbinding kan maken met uw mailserver uw berichten binnenhalen.

Oefening 2.1
Zorg ervoor dat sendmail automatisch gestart wordt wanneer u uw computer aan zet zodat mail verstuurd kan worden. Controleer vervolgens met het lokale commando mail of u in staat bent om mail te versturen tussen twee gebruikers. Bekijk of u het mailbericht ook in de mailqueue ziet verschijnen voordat het verzonden wordt. Als dit niet het geval is, start Sendmail dan zo op dat de mailqueue minder vaak geleegd wordt. Als het verzenden van lokale mail met Sendmail lukt, configureer dan de POP-daemon qpopper. Test vanaf een client computer dat dit werkt door een POP-mailaccount te definiëren voor een gebruiker die op uw computer gedefinieerd is. Omschrijf wat er precies voor deze gebruiker moet gebeuren om een POP-account werkzaam te krijgen.

2.2 Mail distributielijsten aanmaken met Majordomo

Majordomo is een programma waarmee het beheer van mailinglists op internet vereenvoudigd wordt. Het mooie van Majordomo is dat alle communicatie met het programma plaats vindt door middel van mail. Eindgebruikers kunnen bijvoorbeeld met behulp van email aangeven dat ze zich in willen schrijven op een Majordomo maillijst en kunnen er ook met behulp van email voor zorgen dat ze niet langer berichten ontvangen van het Majordomo proces. Majordomo verstuurt zelf geen berichten, daarvoor maakt het gebruik van standaard mailers zoals Sendmail. Het enige dat Majordomo doet, is onderhouden van mailinglijsten: het vertelt aan Sendmail dat er een bericht verstuurd moet worden aan alle mensen die zijn ingeschreven op de Majordomo maillijst. Uiteraard kunt u Majordomo volledig vanaf de commandoregel configureren. Er is echter ook een goede web-interface voor Majordomo, namelijk het programma MajorCool. U kunt dit programma downloaden vanaf www.siliconexus.com/MajorCool/ .

2.2.1 Installatie van Majordomo

Veel software wordt standaard meegeleverd met alle distributies, dit is niet het geval voor Majordomo. Dit betekent dat u het programma handmatig zult moeten installeren voordat u het kunt gebruiken om mailinglists aan te maken en te beheren. De volgende procedure geeft een overzicht van de stappen die doorlopen moeten worden:

1. Ga naar www.greatcircle.com/majordomo/ en download daar de huidige versie van Majordomo. U vindt deze versie hier in verschillende formaten, selecteer **Source distribution (gzip'd)** en sla het bestand op in uw homedirectory.
2. Creëer nu een gebruiker majordomo en maak deze gebruiker lid van de groep daemon. Geef /usr/local/majordomo op als homedirectory van deze gebruiker en specificeer /bin/false als Shell zodat deze gebruiker niet in kan loggen. Gebruik hiervoor bijvoorbeeld de opdracht **useradd –u 45 –g daemon –d /usr/local/majordomo –s /bin/false majordomo**.

3. Pak nu het bestand dat u zojuist hebt opgehaald uit met de opdracht **tar zxvf majord*.tar.gz**. Activeer vervolgens de directory die met het uitpakken van dit bestand is aangemaakt.
4. Open vanuit de Majordomo installatiedirectory het bestand Makefile met een editor en controleer dat alle instellingen juist staan. Houdt er rekening mee dat u werkt met een programma dat niet specifiek voor gebruik op Linux geschreven is en dat in vrijwel alle gevallen de nodige aanpassingen gedaan moeten worden. Lees daarom het hele bestand aandachtig door en zorg ervoor dat in elk geval de volgende zaken naar de juiste plaats verwijzen
   - locatie van perl
   - Naam en locatie van de C-compiler
   - homedirectory van de gebruiker majordomo
   - UID en GID.

In het onderstaande voorbeeld ziet u de standaardwaarden die nodig zijn op ons Fedora testsysteem:

PERL = /usr/bin/perl
CC = gcc
W_HOME = /usr/local/majordomo
W_USER = 45
W_GROUP = 2

***majormake.tif Voordat u Majordomo kunt installeren, moet u de juiste instellingen voor uw systeem definiëren in het Makefile.

5. Nu moet u het bestand sample.cf in de Majordomo installatiedirectory handmatig aanpassen zodat de juiste verwijzingen erin gemaakt worden. Pas in elk geval de variabele $whereami aan. Daarnaast moet u ook de variable $sendmail_command aanpassen, op de meeste Linux distributies moet deze staan ingesteld als /usr/sbin/sendmail. , de overige variabelen kunnen in de meeste situaties gewoon zo blijven staan. Als u klaar bent met het maken van aanpassingen, kopieert u het bestand sample.cf naar majordomo.cf.

***samplecf.tif In het configuratiebestand sample.cf past u alle instellingen aan naar de waarden die op uw systeem gebruikt moeten worden.

6. Geef als root nu de opdracht **make wrapper**. Hiermee wordt het programma met de naam wrapper gecompileerd. Geef vervolgens ook de opdrachten **make install** en **make install-wrapper**. Hiermee wordt ook het programma majordomo gemaakt en worden beide programmabestanden op de juiste locatie neergezet. Let tijdens de uitvoering van deze opdrachten op foutmeldingen, als u namelijk in de eerdere stappen een fout gemaakt wordt, kan de compilatie van beide programmabestanden mislukken. Als alles goed gegaan is geeft u de opdracht **cd /usr/local/majordomo; ./wrapper config-test** om te kijken of alles in orde is. Als dit het geval is, krijgt u daarvan melding en kunt u beginnen te werken met Majordomo. U krijgt ook gelijk de gelegenheid om uw versie van Majordomo te registreren, als u dat doet, wordt u automatisch op de hoogte gesteld wanneer er patches voor het programma zijn uitgekomen.

Tip! Als u sendmail geconfigureerd hebt om te werken met de restricted Shell **smrsh**, moet u in de smrsh-directory een link aanmaken naar het **wrapper** programmabestand, anders is majordomo niet in staat om de Sendmail mailserver te gebruiken.

2.2.2 Een mailinglist aanmaken

Om een mailinglist te definiëren, moet u als root twee dingen doen:

* Sendmail aliassen aanmaken
* De benodigde bestanden aanmaken voor Majordomo.

In het volgende voorbeeld gaan we een mailinglist aanmaken met de naam roddellijst.

1. Om te beginnen moet u Sendmail aliassen aanmaken. Gebruik hiervoor het Sendmail aliases bestand of indien gewenst het speciale majordomo aliasbestand majordomo.aliases. Vervang de naam roddellijst in de onderstaande regels door de naam van de mailinglist die u wilt maken:

```
roddellijst:              "|/usr/local/majordomo/wrapper resend –l \
                          roddellijst roddellijst-list"
roddellijst-list::include:/usr/local/majordomo/lists/roddellijst
owner-roddellijst:        mail@uwdomein.nl
roddellijst-owner:        mail@uwdomein.nl
roddellijst-request: "|/usr/local/majordomo/wrapper majordomo \
                          -l roddellijst"
roddellijst-approval:     mail@uwdomein.nl
```

In deze aliassen wordt feitelijk alles geregeld dat nodig is voor gebruik van de Majordomo mailinglist. Om te beginnen wordt gedefinieerd hoe de lijst bediend moet worden, vervolgens wordt de eigenaar van de lijst gedefinieerd en tot slot wordt aangegeven hoe gebruikers zich in kunnen schrijven op de mailinglist. Door middel van de regel roddellijst-request in dit voorbeeld, kan een gebruiker een mailbericht sturen aan majordomo met de tekst **subscribe** in de body van het bericht. Dit wordt uitgevoerd als opdracht die ervoor zorgt dat de gebruiker inderdaad wordt ingeschreven. Vergeet niet na het aanmaken van de aliassen de opdracht **newaliases** uit te voeren om ervoor te zorgen dat de aliassen ook bekend worden gemaakt bij het mailprogramma.

2. Nu de nodige aliassen zijn aangemaakt, moet de mailinglist configuratie zelf gemaakt worden. Dit gebeurt door een tekstbestand aan te maken en daar een boodschap in te zetten. Voer hiervoor achter elkaar de volgende opdrachten uit:

```
cd /usr/local/majordomo/lists
touch roddellijst
echo "nieuwtjes die niemand iets interesseren" > roddellijst.info
chown majordomo.daemon svv*
chmod 664 svv*
echo "config roddellijst roddellijst.admin" | mail majordomo@sandervanvugt.nl
```

Met de voorgaande opdrachten wordt het majordomo configuratiebestand aangemaakt. Vooral de laatste opdracht is interessant: met behulp van deze opdracht wordt namelijk per mail het aanmaken van het configuratiebestand voltooid.

2.2.3 Onderhoud van de mailinglist

Nadat op de hiervoor beschreven wijze de mailinglist is aangemaakt, kan hij onderhouden worden. Dit gebeurt – daar is het majordomo voor – door opdrachten op te nemen in de body van de tekst. Let even op dat u de opdrachten niet per ongeluk op de subject-regel zet, deze wordt door majordomo namelijk niet gelezen. De volgende opdrachten kunnen gebruikt worden:

* **list** Toont een overzicht van beschikbare Majordomo lijsten op deze server

* **info <list>** Vraag de algemene informatie op voor de gespecificeerde lijst.
* **subscribe <list> | <address>** Schrijf uzelf in op de mailinglist. Ook is het mogelijk als extraatje een adres op te geven van iemand anders die u in wilt schrijven. Als de majordomo server goed geconfigureerd is, zal de eigenaar van het betreffende mailadres wel eerst een bevestiging moeten sturen dat hij ook inderdaad ingeschreven wil worden.
* **unsubscribe <list> | <address>** Verwijder uzelf of de eigenaar van het aangegeven mailadres van de majordomo lijst.
* **intro <list>** Vraag het bestand op waarin beschreven staat hoe de lijst gebruikt moet worden.
* **newintro <list>** Schrijf een nieuwe introductie voor de lijst. Alles na de opdracht newintro wordt in dat geval als de nieuwe introductie beschouwd. Om het introductiebericht af te sluiten, geeft u de tekenreeks EOF op een nieuwe regel. Afhankelijk van de configuratie van de lijst, kan het nodig zijn hierbij ook een wachtwoord op te geven.
* **passwd <list> <oudwachtwoord> <nieuwwachtwoord>** Stel een nieuw wachtwoord dat nodig is voor beheer van de lisjt.

Op basis van het bovenstaande kunt u een Majordomo mailinglijst maken. Naat u de aliassen en de configuratie hebt aangemaakt, is de mailinglijst meteen operationeel. Voor meer informatie over het gebruik van Majordomo, kunt u de Majordomo faq lezen. U vindt deze op www.greatcircle.com/majordomo/majordomo-faq.html. In deze FAQ wordt een antwoord gegeven op veel praktische vragen waar u als beheerder van een majordomo mailinglijst mee te maken krijgt.

Oefening 2.2
Installeer Majordomo. Definieer een mailinglist met de naam “informatie”. Koppel een informatiebericht aan deze mailinglist zodat nieuwe gebruikers zien wat het doel is van de lijst. Schrijf u in als twee lokale gebruikers op deze mailinglist (u kunt DNS-namen dan even verder vergeten om het eenvoudig te houden). Stuur een bericht naar de mailinglist en controleer dat dit bericht inderdaad doorgestuurd wordt naar alle gebruikers die lid zijn van de mailinglist.

2.3 Usenet en stand-alone nieuwsservers

Wanner u het hebt over nieuwsservers op internet, hebt u het al snel over Usenet. Maar wat is Usenet nu eigenlijk? We zouden het heel bot kunnen definiëren als een aantal nieuwsservers die op een of andere manier berichten met elkaar uitwisselen. Kenmerkend is dat er helemaal niets centraal geregeld wordt: het enige wat u nodig hebt om deel te worden van usenet, is een ‘feed’. Dit is een andere nieuwsserver op Usenet die bereid is u te voorzien van een update wanneer er nieuwe artikelen binnenkomen. Zorg voor voldoende capaciteit als u een Usenet nieuwsserver in wilt richten, want er gaan behoorlijk wat berichten rond op Usenet. Voor dat u het weet zorgen de verschillende feeds ervoor dat uw volledige schijfruimte gevuld is met nieuwsberichten. Een Usenet-nieuwsserver zou dan ook altijd een aparte server moeten zijn die speciaal voor dit doel is ingericht. Naast de mogelijkheid om een nieuwsserver in te richten en lid te maken van de Usenet-hiërarchie, is het ook mogelijk te werken met stand-alone nieuwsservers. Dit zijn losstaande servers waarop mensen kunnen intekenen om berichten te ontvangen en die verder helemaal niets synchroniseren met andere servers.

2.3.1 Structuur van Usenet

De basis van een nieuwsserver is een artikel. Dit is een bericht dat voorzien is van een header, net zoals dat het geval is voor mailberichten. Deze artikels worden in nieuwsgroepen geplaatst. Eenvoudig bekeken is een nieuwsgroep niet meer of minder dan een directory

waarin nieuwsberichten geplaatst kunnen worden. Net zoals in een bestandssysteem zijn de namen van de nieuwsgroepen hiërarchisch opgebouwd zodat het eenvoudig wordt om een bericht terug te vinden. De naam begint met de hoofdcategorie, vervolgens komen de subcategorieën, net zolang totdat u uit bent gekomen bij de uiteindelijke nieuwsgroep waar u gebruik van wilt maken. U kunt zich hier de meest uiteenlopende onderwerpen bij voorstellen: het varieert van serieuze zaken als comp.os.windows.bugs tot meer frivole zaken zoals alt.agriculture.beef. Houd er overigens rekening mee dat nieuwsservers publiekelijk en anoniem zijn: dit betekent dat er veel vervuiling optreedt en dat u in veel nieuwsgroepen nogal wat rommel tegenkomt.

***pannews Om vanaf een Linux-systeem nieuwsberichten te kunnen lezen, hebt u een nieuwslezer zoals PAN nodig.

Wanneer er een relatie wordt opgezet tussen twee nieuwsservers, kan daarbij per nieuwsgroep aangegeven worden of deze gesynchroniseerd moet worden of niet. Dit is prettig, want als het alles of niets zou zijn, zou elke nieuwsserver meerdere petabytes aan opslagruimte nodig hebben. U zou er dus voor kunnen kiezen op uw nieuwsserver alleen maar alle nieuwsgroepen onder comp te synchroniseren.

Om nieuwsberichten tussen sites te kunnen synchroniseren, doet elke nieuwsserver aan flooding. Dit betekent dat een nieuw artikel doorgestuurd wordt naar andere sites waarmee een relatie bestaat. Om dit proces goed te laten verlopen, wordt elk artikel voorzien van een duidelijk herkenbare header. Hierin bevindt zich de site-ID en een serienummer. Op basis daarvan kan heel snel achterhaald worden of een bericht al op een andere nieuwsserver voorkomt of niet. Zo kan de synchronisatiebeslissing snel genomen worden. In het meest primitieve denkbare scenario zou elk nieuwsbericht afzonderlijk verstuurd moeten worden. U kunt zich voorstellen dat dit niet handig is en daarom gebeurt dit ook niet.
Als de nieuwsserver gebruikmaakt van het archaïsche UUCP-protocol (een protocol dat er puur op gericht is om bestanden tussen servers heen en weer te schuiven), worden alle berichten verzameld in een batch. Wanneer er een redelijke verzameling nieuwsberichten is, wordt die pas verstuurd naar andere nieuwsservers. Moderne nieuwsservers die werken op basis van het NNTP-protocol gaan echter veel slimmer te werk. Hier wordt gewerkt met ihave/sendme lijsten. Nieuwsservers wisselen daarbij lijsten met elkaar uit waarin wordt samengevat welke berichten er allemaal op een bepaalde server voorkomt. Die lijsten worden met elkaar vergeleken en op basis daarvan kan een lijst worden samengesteld van alle andere berichten die nog ontvangen moeten worden van een andere server. Dit is de zogenaamde sendme-lijst. In feite is dit een verlanglijstje waardoor de ene nieuwsserver de andere server heel precies kan laten weten wat hij nog nodig heeft.

2.3.2 De rol van NNTP

Of het nu gaat om communicatie van Usenet nieuwsservers met elkaar of van een eindgebruiker met een nieuwsserver, in beide gevallen wordt gebruikgemaakt van het Network News Transfer Protocol (NNTP). In dit protocol wordt zowel gedefinieerd hoe eindgebruikers nieuwsberichten kunnen ophalen bij een nieuwsserver als hoe nieuwsservers onderling met elkaar kunnen synchroniseren.

NNTP definieert een drietal manieren om nieuwsberichten uit te wisselen:
* pushing
* pulling
* interactief lezen.

Bij pushing wordt gebruik gemaakt van het ihave/sendme mechanisme dat hierboven beschreven is. Pulling is de typische techniek die gebruikt wordt door gebruikers die bij de nieuwsserver een lijst opvragen van berichten die sinds een bepaalde datum zijn verschenen. De derde techniek wordt door moderne nieuwsclients gebruikt; dit is de techniek van interactieve nieuwslezers waarin u per nieuwsgroep kunt aangeven welke berichten van de server opgehaald moeten worden.

Om ervoor te zorgen dat een nieuwsserver zijn werk kan doen, moet de server waarop de berichten ontvangen worden hierop ingericht worden. Op een Linux nieuwsserver wordt in alle gevallen gebruikgemaakt van de directory /var/spool/news. Dit is de verzamelplaats waar alle berichten opgeslagen worden. Onder deze directory wordt een directorystructuur aangemaakt die overeenkomstig is aan de hiërarchie van nieuwsgroepen zoals die op Usenet gebruikt wordt. In welke subdirectory van /var/spool/news u deze hiërarchie terugvindt, verschilt per Linux distributie. Vaak wordt voor dit doel gebruikgemaakt van de directory articles, het kan zijn dat uw distributie een andere oplossing gekozen heeft. De nieuwsberichten worden hierin overigens niet eeuwig bewaard: de nieuwsserver zou in dat geval immers snel dichtslibben. In alle gevallen wordt gebruikgemaakt van het verjaren (expiry) van nieuwsberichten. Wanneer een bericht een door de beheerder ingestelde uiterste houdbaarheidsdatum bereikt heeft, wordt het automatisch van de server verwijderd.

2.3.3 Stand alone nieuwsservers

In feite is er geen technisch verschil tussen een nieuwsserver die deel uitmaakt van Usenet en een stand-alone nieuwsserver. Het enige verschil is dat op de Usenet-nieuwsserver een of meerdere feeds gedefinieerd zijn terwijl dat voor de stand-alone nieuwsserver niet het geval is. Op de achtergrond wordt gewoon gebruikgemaakt van dezelfde programmatuur. In het vervolg van dit hoofdstuk kunt u lezen over de inrichting van beiden.

2.3.4 Een eenvoudige nieuwsserver inrichten met Leafnode

De beste manier om bekend te worden met het fenomeen nieuwsserver, is door eenvoudig te beginnen en een stand-alone nieuwsserver in te richten. Dat kan bijvoorbeeld door gebruik te maken van de nieuwsserver leafnode. U kunt deze server downloaden van leafnode.sourceforge.net. Houd wel even in de gaten dat u hier te maken hebt met een zeer eenvoudige nieuwsserver. Leafnode is ontworpen voor gebruik in thuisnetwerken of kleine bedrijven. Het is ontworpen met de volgende gebruiksdoelen in het achterhoofd:

* Nieuwsberichten lezen zonder dat er verbinding is met internet;
* Berichten binnenhalen van meerdere servers zodat ze aangeboden kunnen worden aan een nieuwslezer die maar contact kan hebben met één server tegelijk;
* Voorkomen dat berichten in een klein netwerk meerdere keren binnengehaald moeten worden

Eigenlijk kan een Leafnode nieuwsserver dus beschouwd worden als een server waarop berichten gecached worden.

Een van de belangrijkste voordelen van het gebruik van Leafnode is dat er geen configuratie nodig is op de nieuwsservers waarvan berichten binnengehaald moeten worden. Dit komt omdat een leafnode-nieuwsserver zich in de optiek van deze nieuwsservers als een client gedraagt. Leafnode haalt nieuwsberichten binnen van deze servers en doet dat met enige intelligentie. Zo wordt er niet langer binnengehaald vanuit een nieuwsgroep waaruit de berichten niet gelezen worden. Ook is het mogelijk te werken met eenvoudige filters zodat

artikelen op basis van informatie in de header of grootte tegengehouden kunnen worden. Ook is er een optie om eerst alleen headers te downloaden zodat het volledige artikel pas wordt binnengehaald wanneer daar ook echt behoefte aan is.

Installatie van Leafnode
Voordat u met Leafnode aan het werk kunt, moet u het eerst downloaden en installeren; de software wordt namelijk met de meeste distributies niet standaard meegeleverd. Op sourceforge.leafnode.net is een RPM-package beschikbaar; u kunt dit na downloaden installeren met de opdracht **rpm –i leafnode-versienummer.rpm**. Dit commando zorgt ervoor dat alle bestanden op de juiste plaats worden neergezet. U hoeft vervolgens de leafnode-server alleen nog maar verder af te configureren door de verschillende configuratiebestanden te bewerken.

Configuratie van Leafnode
Om Leafnode aan het werk te krijgen, moet u een paar stappen doorlopen:
1. Bewerk het configuratiebestand /etc/config/leafnode. In dit configuratiebestand geeft u aan hoe de Leafnode-server nieuwsberichten moet ophalen van andere servers.
2. Zorg ervoor dat de server ook automatisch gestart wordt wanneer u de computer aanzet. Maak hiervoor gebruik van het xinetd-mechanisme.
3. Zorg ervoor dat de nodige nieuwsberichten binnengehaald worden. In het onderstaande worden deze stappen in detail beschreven.

Stap 1: /etc/leafnode/config
Het eerste deel van de leafnode-configuratie bestaat eruit dat u het lokale configuratiebestand /etc/leafnode/config moet aanpassen. In dit bestand bepaalt u in grote lijnen het gedrag van de leafnode-server. U geeft er onder andere in aan bij welke nieuwsserver berichten opgehaald moeten worden en tevens specificeert u hoe lang lokale berichten bewaard moeten blijven. Deze laatste optie is erg belangrijk, zo zorgt u er namelijk voor dat uw server niet volloopt met oude berichten. Daarnaast zijn er nog de nodige andere opties. In feite behoeft het bestand weinig toelichting; met de leafnode-software komt een voorbeeldbestand dat rijkelijk voorzien is van commentaar waardoor snel duidelijk wordt wat precies de bedoeling is.

***leafnodeconf Het grootste deel van de leafnode-configuratie wordt geregeld in een goed gedocumenteerd configuratiebestand

In feite zijn er maar twee instellingen die echt verplicht zijn in het leafnode-configuratiebestand. Om te beginnen is dat de parameter expire = waarmee u aangeeft hoelang nieuwsberichten bewaard moeten blijven. Als standaard wordt er gewerkt met expire = 20 zodat berichten na twintig dagen automatisch weer verwijderd worden; uiteraard kunt u dit geheel naar eigen behoefte aanpassen. Vervolgens moet er een verwijzing gegeven worden naar de nieuwsserver waar de berichten vandaan gehaald moeten worden. Hiervoor kunt u de nieuwsserver van uw internetaanbieder gebruiken, raadpleeg uw internetaanbieder voor de exacte naam van deze server. Vul deze naam vervolgens in achter de parameter server =; bijvoorbeeld server = news.mijninternetaanbieder.nl. Wanneer uw internetaanbieder alleen abonnees toelaat op zijn mailserver (wat meestal het geval is), hebt u naast de server-instelling ook nog de instellingen username = en password = nodig. Met behulp van deze instellingen kunt u authenticeren op de nieuwsserver van de internetaanbieder. Houdt er natuurlijk wel rekening mee dat het niet echt veilig is om uw gebruikersnaam en wachtwoord zomaar in leesbare vorm in een tekstbestand te plaatsen.

In veel gevallen zult u in het configuratieprogramma slechts naar één nieuwsserver verwijzen. Wanneer deze ene server echter niet alle nieuwsgroepen aanbiedt die u nodig hebt, kan het nodig zijn nieuwsberichten binnen te halen van meerdere nieuwsservers. Herhaal hiervoor de parameters server = en indien nodig ook username = en password =.

server = mijnserver.mijninternetaanbieder.nl
username = handigehenk
password = geheim

server = eenandereserver.eenanderdomein.nl
username = vatsigevincent
password = geheim

Naast de twee verplichte parameters is er ook nog een aantal instellingen dat optioneel is, maar in sommige gevallen zeer aan te raden:

* **initialfetch = 100** Zonder deze parameter zullen alle berichten uit een nieuwsgroep opgehaald worden. Dit kan leiden tot een aanzienlijke belasting aangezien er nieuwsgroepen zijn waarin per dag honderden berichten geplaatst worden. Met deze instelling zorgt u ervoor dat er maximaal honderd berichten worden binnengehaald.
* **timeout = 60** Wanneer een nieuwsserver niet bereikt kan worden, geeft leafnode het normaal na tien seconden op. Als echter bekend is dat de server in kwestie langzaam is, is het de moeite waard deze timeout-waarde te verhogen. Met behulp van de instelling timeout = 60 wordt de timeout ingesteld op 60 seconden.

Buiten de twee instellingen die hier genoemd zijn, kan er nog veel meer geregeld worden in het leafnode-configuratiebestand. Wij gaan daar verder niet op in, u kunt voor meer informatie het goed gedocumenteerde configuratiebestand op uw distributie bekijken.

Stap 2: xinetd configuratie
Het tweede bestand dat aangepast moet worden voordat u met leafnode aan het werk kunt, is het xinetd-configuratiebestand. Zoals vrijwel elke service die met xinetd gestart wordt, moet u er om te beginnen voor zorgen dat de service aangezet wordt. Dit doet u door de regel disable = no in het bestand op te nemen. Controleer tevens of de standaard gebruiker news op uw systeem bestaat en maak deze aan indien nodig; meestal zal dit echter niet nodig zijn. Wanneer verder alles naar wens geregeld is, kunt u xinetd (opnieuw) opstarten. De leafnode nieuwsserver is nu klaar voor gebruik.

Stap 3: Berichten binnenhalen met fetchmail
Wanneer uw leafnode-nieuwsserver volgens de bovenstaande procedure geconfigureerd is, kunt u aan het werk. Dit betekent dat u nu via leafnode contact kunt opnemen met de nieuwsserver van uw internetaanbieder om daar een lijst van nieuwsgroepen van te downloaden. Hiervoor gebruikt u de opdracht **fetchmail**. Houd er rekening mee dat deze opdracht enige tijd nodig kan hebben om zijn werk te doen. Hoeveel tijd precies nodig is, is uiteindelijk afhankelijk van de snelheid van uw internetverbinding. U moet fetchmail als root of als de speciale gebruiker news uitvoeren. Het is aan te raden de optie –vv mee te geven, dit zorgt ervoor dat u ook te zien krijgt wat er precies gebeurt. Zonder deze parameters wordt namelijk geen activiteit getoond; zeker wanneer u op een niet al te snelle verbinding werkt, kan dit met zich meebrengen dat het onduidelijk is of er überhaupt wel wat gebeurt.

Als fetchmail klaar is met zijn werk, heeft dat als resultaat dat een bestand is aangemaakt met de naam /var/spool/news/leaf.node. Hierin vindt u een lijst van alle nieuwsgroepen die de nieuwsserver van uw internetaanbieder in de aanbieding heeft. Als u hiermee klaar bent, kunt u met uw favoriete nieuwslezer contact maken met uw leafnode-nieuwsserver. Haal berichten binnen, teken in op nieuwsgroepen en doe alles wat u met uw nieuwsserver wilt doen. Deze informatie wordt doorgegeven aan de leafnode-nieuwsserver. U zult echter wel merken dat u nog niet direct nieuwsberichten van de uiteindelijke nieuwsservers binnen kunt halen. In uw nieuwslezer ziet u alleen een bericht dat u gebruik maakt van leafnode en dat iemand op de leafnode-server de opdracht fetchmail uit moet voeren om ervoor te zorgen dat ook de nieuwe nieuwsgroepen met hun inhoud binnengehaald worden. Om het proces te voltooien, moet u nog een keer de opdracht fetchmail gebruiken. Dit zorgt ervoor dat alle nieuwsgroepen waarop u bent ingeschreven automatisch door leafnode worden binnengehaald. Wordt uw nieuwsserver té actief gebruikt om elke keer op deze manier te werk te gaan? Dan kunt u overwegen om een cronjob aan te maken waardoor de opdracht fetchnews regelmatig wordt uitgevoerd. Ook wanneer u dat doet, zal er echter altijd een periode verstrijken tussen het moment waarop een gebruiker berichten in een nieuwsgroep wil lezen en het moment dat die berichten ook daadwerkelijk binnengehaald worden.

***outlookexpressnieuws Voordat u via leafnode mailberichten in een nieuwslezer kunt bekijken, moet u eerst op de nieuwsserver nog een keer de opdracht fetchmail uitvoeren.

Oude berichten wissen

U hebt nu een werkende leafnode-mailserver die ervoor zorgt dat automatisch alle berichten worden binnengehaald waar u interesse hebt. Er moet echter nog een zaak geregeld worden voordat alles klaar is: u moet ervoor zorgen dat oude berichten automatisch opgeruimd worden. De leafnode-server zelf heeft niet voldoende intelligentie om dit automatisch te doen, dat betekent dat u iets zult moeten regelen. De opdracht die u hiervoor nodig hebt, is **texpire**. Wanneer deze opdracht wordt uitgevoerd, kijkt het of er berichten zijn die in aanmerking komen om opgeruimd te worden. Het probleem is alleen dat de opdracht niet vanzelf actief wordt, maar dit is natuurlijk een koud kunstje wanneer u ervoor zorgt dat het automatisch gestart wordt door het Cron-mechanisme. Geef als root de opdracht **crontab –u news –e**. Dit zorgt ervoor dat vi geopend wordt als editor voor crontab. Druk nu op de toets i om de Vi-insert modus te activeren en voer de regel 0 19 * * * /usr/sbin/texpire in. Sluit Vi door eerst op de Escape-toets te drukken en vervolgens de opdracht **:wq!** te geven. Vanaf dat moment worden voortaan dagelijks om zeven uur `s avonds alle oude nieuwsberichten automatisch opgeruimd.

2.3.5 Internet News

Leafnode is leuk, het heeft echter ook een belangrijk nadeel. U kunt het namelijk niet gebruiken om er lokale nieuwsgroepen mee aan te maken. Eigenlijk is het geen echte nieuwsserver, maar een programma dat berichten van echte nieuwsserver in een cache bewaart. Dit probleem wordt opgelost in Leadnode versie 2, maar aangezien deze versie nog niet beschikbaar was op het moment dat dit geschreven werd, moet u zich voorlopig behelpen met een ander programma. Het meest gangbare programma dat voor dit doel gebruikt wordt, is de Internet News Daemon (INN). We geven hier een inleiding in de werking van dit programma.

Een volledige nieuwsserver zoals INN vervult drie functies:

* Hij accepteert artikelen van andere servers en slaat die op op schijf;
* Hij verstuurt artikelen die hij ontvangen heeft naar andere servers;

* Hij biedt artikelen aan aan eindgebruikers.

Daarnaast moet er intern nog wat beheer uitgevoerd worden door berichten te verwijderen die de uiterste houdbaarheidsdatum hebben overschreden.

Om te voorzien in deze functies, maakt INN gebruik van een aantal componenten. Het proces innd zorgt ervoor dat nieuwsberichten van andere servers worden binnengehaald. Als het nodig is berichten te versturen naar andere servers, wordt voor dat doel gebruik gemaakt van **innfeed**, **nntpsend** of **innxmit**. Welke oplossing uiteindelijk gebruikt wordt, is afhankelijk van de onderliggende protocollen. Tot slot is er **nnrpd** wat ervoor zorgt dat nieuwsclients contact kunnen opnemen met de nieuwsserver om zo berichten binnen te halen.

INN configuratie

Om met een INN-nieuwsserver aan het werk te kunnen gaan, moet een aantal zaken geregeld worden:

1. De server moet gestart worden;
2. Het hoofdconfiguratiebestand moet bewerkt worden;
3. Er moeten nieuwsgroepen aangemaakt worden;
4. U moet machtigingen op de nieuwsserver instellen;
5. De nieuwsfeeds moeten gedefinieerd worden.

In de volgende paragrafen leest u hoe deze taken uitgevoerd moeten worden.

Stap 1: De server starten

Om de INN-server te starten moet u ervoor zorgen dat het proces **innd** actief wordt. Hiervoor wordt doorgaans gebruikgemaakt van een speciaal script dat ervoor zorgt dat de daemon geactiveerd wordt en ook de benodigde environment-instellingen gedaan worden. U vindt dit script onder de naam **inndstart**; de exacte locatie van dit bestand verschilt per distributie. Zowel op SUSE als op Fedora Linux wordt bijvoorbeeld gebruikgemaakt van /usr/lib/news/bin/inndstart. Uiteraard kunt u er ook voor zorgen dat INN automatisch geactiveerd wordt wanneer u uw computer aan zet. Neem hiervoor een link op die verwijst naar het inndstart-script in de directory /etc/init.d.

Stap 2: Aanpassen van de configuratie

De configuratie van INN wordt gedaan door het ASCII-tekstbestand /etc/news/inn.conf te bewerken. Zoals vaker het geval is met configuratiebestanden op Linux, is dit programma rijkelijk voorzien van commentaar. U doet er dan ook goed aan het bestand gewoon te openen en eens door te nemen. U zult echter merken dat in veel gevallen de standaardinstellingen uitstekend voldoen, zeker wanneer u gebruik maakt van een versie van INN die standaard met uw distributie mee geïnstalleerd is.

***innconf De configuratie van de INN-nieuwsserver wordt gedaan in het configuratiebestand inn.conf.

Het totale aantal parameters dat in inn.conf kan worden opgenomen is behoorlijk groot, om die reden vindt u hier geen volledige opsomming van alle mogelijkheden. In de meeste gevallen zijn er maar vijf parameters die in elk geval ingesteld moeten worden:

mta: "/usr/sbin/sendmail –oi –oem %s"
organization: "Sander's INN server"
ovmethod: tradindexed
hismethod: hisv6

pathhost: localhost
pathnews: /usr/lib/news

* **mta** Hiermee geeft u aan welke message transfer agent gebruikt moet worden om nieuwe artikelen te posten op de nieuwsserver. De standaardinstelling is vaak dat deze verwijst naar de lokale sendmail-server. Deze regel moet gewijzigd worden wanneer u een andere mailserver gebruikt.
* **organization** Met behulp van deze instelling kunt u automatisch een naam laten invullen voor de organisatie waarvoor uw gebruikers werken. Let erop dat het aan te raden is hier zeker de standaardnaam te wijzigen, als u dit niet doet krijgen alle nieuwsberichten van de gebruikers van uw INN-server de ondertekening “A poorly configured INN server”.
* **ovmethod** Om de toegang tot uw mailserver te vereenvoudigen, staat standaard de optie enableoverview aan. Deze optie zorgt ervoor dat gebruikers snel een beeld kunnen krijgen van wat er aan berichten in een bepaalde nieuwsgroep voorkomt. Om dit goed te laten verlopen, moet echter met behulp van de instelling ovmethod worden aangegeven welke opslagmethode hiervoor gebruikt moet worden. De instelling tradindex zorgt ervoor dat de nieuwslezers de gegevens met een optimale snelheid kunnen benaderen, het nadeel is dat schrijfacties enigszins vertraagd worden.
* **histmethod** Met behulp van deze parameter wordt bepaald welke methode er gebruikt moet worden voor het history-mechanisme. Dit mechanisme bepaalt hoe bijgehouden kan worden welke berichten nog met andere servers gesynchroniseerd moeten worden. Het curieuze is dat er hier maar één instelling mogelijk is, namelijk hisv6. Vergeet deze instelling echter niet, hij is namelijk verplicht.
* **pathhost** Hiermee wordt verwezen naar de nieuwsserver waarop de nieuwsberichten geplaatst moeten worden.
* **pathnews** Met behulp van deze instelling wordt aangegeven waar alle programmabestanden zich bevinden die met de werking van de INN-nieuwsserver te maken hebben.

Stap 3: Nieuwsgroepen aanmaken
Een van de meest interessante verschillen tussen leafnode en INN is dat INN de mogelijkheid biedt lokale nieuwsgroepen aan te maken, terwijl leafnode in feite alleen maar een cache is van groepen die van een of meerdere nieuwsservers worden binnengehaald. INN kan om die reden ook uitstekend als stand-alone nieuwsserver functioneren. Om dit voor elkaar te krijgen, moet u het bestand /var/lib/news/active bewerken. In theorie is het mogelijk dit bestand met de hand te bewerken, het is echter niet verstandig omdat de syntaxis nogal strikt is. U kunt daarom beter gebruik maken van de opdracht **ctlinnd** om er nieuwsgroepen in aan te maken. Er is echter wel een probleem: u kunt **ctlinnd** alleen gebruiken wanneer INN actief is en INN kan alleen geactiveerd worden als er een minimale configuratie voorkomt in het nieuwsgroepenbestand; een mooi voorbeeld van het klassieke probleem van de kip en het ei dus. Bij de meeste standaardconfiguraties bestaat er een configuratiebestand met een minimale inhoud waarmee u kunt beginnen, als dit niet het geval is zorgt u er handmatig voor dat de volgende regels in het bestand worden opgenomen. Hiermee worden twee standaardnieuwsgroepen gedefinieerd. Vaak komen deze regels standaard voor in het bestand.

control 00000000000 0000000001 y
junk 00000000000 0000000001 y

Deze twee standaardgroepen zorgen ervoor dat alle berichten die niet duidelijk thuisgebracht kunnen worden in een bepaalde nieuwsgroep toch ergens terecht komen. Als u het bestand

active handmatig aanmaakt, moet u er overigens voor zorgen dat in dezelfde directory ook een bestand met de naam active.times wordt aangemaakt. Beide bestanden moeten zowel de gebruiker als de groep news als eigenaar hebben.

Als u ervoor gezorgd hebt dat er een minimale configuratie aanwezig is in het bestand active, kunt u **ctlinnd** gebruiken om nieuwsgroepen aan te maken. Dit commando werkt als **ctlinnd newgroup <groepsnaam> <flags> <aanmaker>**. Met het eerste argument van ctlinnd geeft u aan dat u een nieuwe groep wilt maken. U zou als alternatief voor het argument newgroup bijvoorbeeld ook **ctlinnd changegroup** kunnen gebruiken om de instellingen van een bestaande groep te wijzigen. Vervolgens specificeert u de naam van de groep die u aan wilt maken. Let erop dat u hier de volledige naam specificeert, dus niet alleen “suse”, maar comp.os.linux.suse. De naam van de groep wordt gevolgd door eventuele flags die u wilt instellen. Als laatste kunt u als u dat leuk vindt de naam specificeren van de gebruiker die de betreffende groep heeft aangemaakt.

Wanneer u wilt werken met flags, kunt u een keuze maken uit een van de volgende mogelijkheden:

* **y** Gebruikers mogen rechtstreeks berichten in deze nieuwsgroep plaatsen;
* **n** Gebruikers mogen niet rechtsreeks berichten plaatsen in deze nieuwsgroep. Dit betekent dat nieuwe berichten alleen geaccepteerd worden door middel van een feed van een andere nieuwsserver;
* **m** Er is een moderator actief. Alle berichten die in deze nieuwsgroep geplaatst worden, worden eerst doorgestuurd naar de moderator die het bericht goed moet keuren voordat het in de nieuwsgroep geplaatst wordt;
* **j** Artikelen in deze groep worden niet bewaard maar direct doorgestuurd. Het kan dus niet via deze nieuwsserver benaderd worden, maar wel via de nieuwsservers waarmee deze nieuwsserver zijn nieuwsgroepen synchroniseert;
* **x** Er kunnen geen artikelen in deze groep geplaatst worden.

U zou bijvoorbeeld een nieuwsgroep kunnen aanmaken met de opdracht **ctlinnd newgroup local.test y sander**. Het resultaat daarvan ziet er als volgt uit:

```
Linux:/var/lib/news # cat active
control 0000000000 0000000000 y
control.cancel 00000000000 00000000000 y
junk 0000000000 00000000000 y
local.test 00000000000 0000000001 y
```

In dit bestand ziet u eerst de naam van de nieuwsgroep. Vervolgens volgen de zogenaamde himark en lomark. Dit zijn volgnummers die aan de berichten in de nieuwsgroepen worden meegegeven. Deze nummers zijn belangrijk voor de synchronisatie met andere nieuwsservers: op basis van deze nummers kan een andere server namelijk zien of er een update uitgevoerd moet worden of niet. Wanneer u een nieuwsgroep aanmaakt, betekent dat niet alleen dat er een regel in dit configuratiebestand geplaatst wordt. Om de berichten in de nieuwsgroep op te kunnen slaan wordt er in de directory /var/spool/news een directorystructuur aangemaakt waarin de berichten fysiek opgeslagen worden.

Naast het bestand active dat ook een belangrijke rol speelt in het uitvoeren van newsfeeds, kan er nog een configuratiebestand gebruikt worden. Dit is het bestand

/var/lib/news/newsgroups. Hier worden de namen van alle nieuwsgroepen onder elkaar gezet met daarachter een beschrijving van de betreffende nieuwsgroep. Hierdoor kunt u het de eindgebruiker wat eenvoudiger maken een keuze te maken voor de nieuwsgroep waarop hij zich in wil schrijven.

***localtest Voordat u echt online gaat, is het verstandig eerst eens een lokale test uit te voeren van uw nieuwsserver

Stap 4: Machtigingen op de nieuwsserver instellen
U bent nu zover dat u vanaf uw lokale systeem toegang krijgt tot uw nieuwsserver. Grote kans dat u vanaf elk ander systeem alleen maar een foutmelding krijgt: 502 You have no permission. Aangezien er in het verleden nogal eens wat mis is gegaan met de beveiliging van de INN-nieuwsserver, wordt tegenwoordig namelijk de toegang ontzegd aan iedereen behalve gebruikers die contact proberen te maken vanaf de lokale server. Om hier iets aan te doen, moet u het bestand /etc/news/readers.conf bewerken. Dit is het configuratiebestand dat gebruikt wordt door **nnrpd**, het proces dat toegang verschaft aan nieuwslezers die vanaf een ander werkstation contact willen maken.

In readers.conf kunt u complexe toegangsfilters definiëren om te bepalen wie er toegang krijgen tot de nieuwsservers. Hierbij wordt onderscheid gemaakt in gebruikers die alleen berichten kunnen lezen en gebruikers die ook in staat zijn berichten te posten. Houd er bij de configuratie van dit bestand rekening mee dat de permissies die het laatst gevonden worden ook daadwerkelijk worden toegepast. Dit betekent dat u de meest algemene instellingen eerst moet doen en zeer specifieke instellingen als laatste.

In readers.conf wordt gewerkt met blokjes permissies die er als volgt uitzien:

```
access “localhost” {
        users: “<localhost>”
        newsgroups: “*”
        access: RP
}
```

In de eerste regel van het bovenstaande voorbeeld wordt bepaald wat er mogelijk gemaakt moet worden. U hebt hier de keuze uit de opties access en auth. Met auth verleent u gebruikers permissies om te authenticeren, met access maakt u het mogelijk bepaalde nieuwsgroepen te benaderen. Voordat een gebruiker toegang kan krijgen op basis van een access-regel, moet hij altijd eerst toegang hebben gekregen door middel van een auth-regel. De aanduiding die volgt op auth of access is een naam die gegeven wordt aan het groepje instellingen. In dit geval is deze naam dus ingesteld op “localhost”.

Vervolgens wordt in het gedeelte tussen de blokhaken bepaald wat de betreffende gebruikers precies mogen. Eerst wordt in dit voorbeeld bepaald voor welke gebruikers de toegang geldt. Vervolgens wordt aangegeven tot welke nieuwsgroepen deze gebruikers toegang krijgen en als laatste wordt in dit voorbeeld aangegeven welke permissies de gebruikers krijgen. In dit voorbeeld mag de gebruiker berichten lezen (R) en posten (P). Let overigens even op dat niet alle mogelijke parameters in dit voorbeeld voorkomen, voor een volledig overzicht kunt u de man-pagina van readers.conf raadplegen.

Met behulp van readers.conf is het mogelijk op een zeer gedetailleerde wijze te bepalen wie toegang krijgt tot welke nieuwsgroepen. Het is hier niet het doel u dit mechanisme volledig duidelijk te maken. We laten u wel zien hoe u ervoor zorgt dat gebruikers vanaf elke willekeurige computer toegang krijgen om berichten op de nieuwsserver te posten en te lezen. Realiseer u wel dat dit voorbeeld bijzonder onveilig is en niet bepaald is aan te raden voor een nieuwsserver die vanaf internet te bereiken is, al zijn er natuurlijk ook nieuwsservers die juist speciaal hiervoor worden ingericht.

```
auth “iedereen” {
        hosts: “*”
        default: “<iedereen>”
}

access “iedereen” {
        users: “<iedereen>”
        newsgroups: “*”
        access: RP
}
```

In het bovenstaande wordt om te beginnen een authenticatiegroep “iedereen” gedefinieerd. Aan deze groep worden alle computers die verbinding kunnen maken met deze nieuwsserver toegewezen. Vervolgens wordt met de parameter default: een variabele <iedereen> gedefinieerd. Deze variabele wordt vervolgens gebruikt in de access-regel waar alle gebruikers zowel het read als het post-recht tot alle nieuwsgroepen krijgen.

Als u de beveiliging van uw nieuwsserver serieus neemt, is het wellicht aan te raden iets conservatiever te werk te gaan. Zo zorgen de volgende regels ervoor dat alleen gebruikers die voorkomen op het lokale netwerk het recht hebben iets in de nieuwsgroepen te doen. Als er verder geen regels zijn die hiermee tegenstrijdig zijn, krijgt verder niemand anders toegang.

```
auth “lokaalnet” {
        hosts: “192.168.”
        default: “<lokaalnet>”
}

access “lokaalnet”{
        users: “<lokaalnet>”
        newsgroups: “*”
        access: RP
}
```

Wij raden u van harte aan dit laatste voorbeeld te gebruiken in de configuratie van uw nieuwsserver en het voorgaande voorbeeld waarmee de nieuwsserver voor iedereen wordt opengesteld alleen te gebruiken wanneer u de implicaties hiervan volledig kunt overzien.

Authenticatie op basis van gebruikersnaam

Het bovenstaande is natuurlijk heel aardig, maar biedt nog geen mogelijkheid om aan te melden op basis van gebruikersnaam en wachtwoord. Op een veilige nieuwsserver wilt u er waarschijnlijk voor zorgen dat ook deze optie tot de mogelijkheden behoort. Om dit te configureren hebt u twee opties: u kunt gebruik maken van een ASCII-authenticatiebestand

waarin gebruikersnamen en wachtwoorden worden opgenomen of u kunt gebruikmaken van externe Directories zoals een LDAP-server. We bespreken hier hoe u de authenticatie kunt regelen op basis van een afzonderlijk configuratiebestand met gebruikersnamen en wachtwoorden.

Ook voor het aanmaken van een INN-gebruikersdatabase bestaan verschillende mogelijkheden. Wij gaan hier uit van een van de meest eenvoudige opties. Hierbij wordt gebruik gemaakt van de opdracht **ckpasswd**. Dit is het mechanisme om aan te melden op basis van een wachtwoord wanneer gebruik gemaakt wordt van **nnrpd**. Deze opdracht kijkt in principe naar een databasebestand waarin gebruikersnamen en wachtwoorden staan opgeslagen. De wachtwoorden dienen wel versleuteld te zijn door middel van het crypt-algoritme. Dit is hetzelfde algoritme als dat gebruikt wordt om versleutelde wachtwoorden op te slaan in het bestand /etc/shadow; u zou dus de versleutelde wachtwoorden vanuit shadow kunnen kopiëren naar het wachtwoordbestand dat u voor INN wilt gebruiken. Het maakt vervolgens niet zo heel veel uit waar u deze gebruikersnamen en wachtwoorden opslaat; hier wordt namelijk naar verwezen door middel van de optie –f bij de opdracht **ckpasswd**. Een en ander moet worden aangeroepen vanuit het bestand readers.conf. De configuratie die hiervoor nodig is kan er als volgt uitzien:

```
auth all {
        auth “ckpasswd –f /etc/news/authdb”
}

access full {
        users: *
        newsgroups: *
}
```

Met deze regels wordt volledige toegang gegeven aan alle gebruikers die succesvol aangemeld zijn op het wachtwoordmechanisme. Natuurlijk moet u er wel voor zorgen dat het bestand /etc/news/authdb. De structuur van dit bestand is eenvoudig. Het begint met de naam van de gebruiker, gevolgd door het met crypt versleutelde wachtwoord.

Als alternatief voor het bestand waarin u zelf wachtwoorden mag plaatsen die door middel van crypt versleuteld zijn, kunt u ook authenticeren op basis van een database waarin gebruikersnamen en wachtwoorden voorkomen. Dit vereist echter de nodige programmeerkennis: om die reden wordt het hier niet verder behandeld.

Stap 5: Configureer newsfeeds

Om als INN-server echt mee te kunnen spelen in de wereld der groten, moet u ook een configuratie hebben om nieuwsfeeds te doen. Hiervoor wordt gebruikgemaakt van twee verschillende bestanden. In het bestand /etc/news/incoming.conf geeft u aan wat u van een andere nieuwsserver allemaal wilt ontvangen; in /etc/news/newsfeeds specificeert u welke nieuwsgroepen u allemaal met remote nieuwsservers wilt synchroniseren. De inhoud van /etc/news/incoming.conf kan er als volgt uitzien:

```
streaming:          true
max-connections:    8

peer ME {
```

```
        hostname:       “localhost, 127.0.0.1”
}

peer isp {
        hostname:       news.myisp.nl
        patterns:       *
}
```

De meest interessante instellingen in het bovenstaande bestand zijn de instellingen die betrekking hebben op de peer isp. Hierin wordt gedefinieerd met welke andere nieuwsserver u wilt synchroniseren. Vervolgens wordt achter patterns aangegeven welke nieuwsgroepen u wilt synchroniseren. De aanduiding * betekent dat ze gewoon allemaal binnengehaald moeten worden. Voor testdoeleinden is het echter verstandiger eerst wat minder nieuwsgroepen te synchroniseren. Gebruik bijvoorbeeld alt.fr.* om alleen de Franstalige nieuwsgroepen onder alt binnen te halen.

Uiteraard wilt u ook dat er de andere kant op iets gebeurt. Om aan te geven dat de inhoud van uw nieuwsserver gesynchroniseerd moet worden met een andere nieuwsserver, configureert u het bestand /etc/news/newsfeeds. De regels die in dit bestand voorkomen zien er als volgt uit:

```
news.myisp.nl:alt.fr.*:Tf:news.myisp.nl
```

In het eerste deel van de regel wordt aangegeven wat er met wie gesynchroniseerd moet worden. Vervolgens wordt met de parameter Tf bepaald dat van elke nieuws-feed een regel moet worden weggeschreven in een logbestand met de naam news.myisp.nl. Wanneer u echter alleen maar dit soort regels opneemt in het bestand newsfeeds, gebeurt er nog niets. Het werkelijke werk wordt verzet door het afzonderlijke programma **innfeed**. Om ervoor te zorgen dat dit programma zijn werk goed kan doen, moet u ook nog een configuratiebestand met de naam innfeed.conf aanmaken. In dit configuratiebestand geeft u aan met welke peer of peers er gecommuniceerd moet worden. De inhoud van dit bestand kan er bijvoorbeeld als volgt uitzien:

```
peer isp {
        hostname:       nieuws.mijnisp.nl
}
```

Oefening 2.3
Configureer INN als standalone nieuwsserver. U hoeft dus geen nieuwsfeeds te definiëren om te communiceren met andere servers op internet. Maak één nieuwsgroep aan op deze server met een naam naar keuze. Controleer vervolgens of u deze nieuwsgroep vanuit een willekeurige nieuwsclient kunt benaderen. Beperk vervolgens de toegang tot de nieuwsserver door ervoor te zorgen dat alleen gebruikers vanaf het lokale netwerk er gebruik van kunnen maken.

Samenvatting
In dit hoofdstuk hebt u geleerd hoe u mail en nieuws kunt configureren op een Linux-server. om te beginnen hebt u enig inzicht verworven in de structuur en werking van de Sendmail mailserver. Vervolgens is besproken hoe u met fetchmail een mailfilter kunt definiëren om te filteren op verschillende criteria en hoe u Qpopper in kunt zetten om ook te voorzien in een POP-proces. In het tweede deel van dit hoofsdtuk hebt u kennis gemaakt met een toepassing

die gebruik kan maken van de Sendmail mailserver: Majordomo. Met behulp van Majordomo kunt u mailinglijsten definniëren die door gebruikers gebruikt kunnen worden om geautomatiseerd berichten met elkaar uit te wisselen. Tot slot is gesproken over de mogelijkheid te werken met nieuwsservers. In dit gedeelte is gebleken dat er eigenlijk maar één nieuwsserver is die echt de moeite waard is: de INN nieuwsserver. U hebt geleerd hoe u deze kunt configureren.

Oefenvragen

1. Wat is de preprocessor?
2. Wat is het belangrijkste configuratiebestand dat bewerkt moet worden voordat u Majordomo succesvol kunt installeren?
3. Welke opdracht wordt gebruikt om het sendmail.cf bestand te genereren?
4. Wat is Usenet
5. Welk onderdeel hebt u naast de Sendmail mailserver nodig om een volledige mailserver te kunnen implementeren?
6. Hoe heeft het bestand waarin geconfigureerd wordt welke nieuwsservers in een Usenet netwerkopstelling met elkaar moeten communiceren?
7. Hoe schrijft u zich als eindgebruiker in op een Majordomo mailinglist?
8. Welke opdracht gebruikt u als beheerder om de inhoud van een mailqueue te monitoren. Waarom gaat dit regelmatig niet goed?
9. Wat moet u doen nadat u nieuwe sendmail aliasen gedefinieerd hebt?
10. Welke opdracht gebruikt u om een nieuwe INN nieuwsgroep te maken?

Hoofdstuk 3 Domain Name System

Inleiding
Op internet wordt gebruik gemaakt van IP-adressen. Aangezien deze adressen voor de meeste mensen vrij lastig te onthouden zijn, zijn er verschillende manieren beschikbaar om namen te verbinden aan IP-adressen. Www.sandervanvugt.nl is nu eenmaal een stuk eenvoudiger te onthouden als 199.201.13.89. De belangrijkste van alle manieren om namen aan IP-adressen te verbinden, is het Domain Name System (DNS). DNS zorgt ervoor dat het volledige internet verdeeld is in meerdere zones waarbij elke computer zijn eigen naam krijgt toegewezen. In dit hoofdstuk leert u hoe u een Linux-server als DNS-server in kunt richten.

Leerdoelen
* Kennis van de werking en componenten van het DNS-protocol
* Configuratie van een cache-only naamserver
* Configuratie van een DNS zone
* Configuratie van een slave server
* Delegatie van een zone
* Analyseren van de werking van een DNS-server.

3.1 Het DNS-protocol

Op internet komen zeer veel computers voor. Sommige computers verlenen diensten die zo eenvoudig mogelijk door anderen gebruikt moeten kunnen worden. Het is dus nodig dat deze computers op een eenvoudige wijze teruggevonden kunnen worden. Om hierin te voorzien, kan gebruik gemaakt worden van het Domain Name System (DNS), zodat de host niet op basis van zijn IP-adres, maar op basis van een logische naam benaderd kan worden.

3.1.1 De DNS namespace

Om alle computers op internet op een efficiënte manier van een naam te voorzien, wordt gebruik gemaakt van DNS-domeinen. Deze zorgen ervoor dat de computers op een logische manier gegroepeerd kunnen worden. Als eerste wordt gebruik gemaakt van een aantal top level-domeinen. Dit zijn een aantal vastliggende domeinnamen; het is niet mogelijk hier zomaar een domeinnaam aan toe te voegen. Om hier domeinnamen aan toe te voegen, wordt in de meeste gevallen langdurig vergaderd door de Internet Assigned Numbers Authority (IANA) die daarover gaat. Op de website van deze organisatie, www.iana.net, vindt u veel informatie over de wereldwijde toepassing van DNS. Momenteel zijn naast domeinnamen voor de verschillende landen op aarde, de volgende algemene domeinen in gebruik:

***iana Op de website van het IANA, vindt u veel informatie over de wijze waarop DNS gedefinieerd is en werkt.

* **com** Commerciële organisaties. Kan door bedrijven van over de hele wereld gebruikt worden.
* **edu** Voor Amerikaanse onderwijsinstellingen.
* **gov** Voor Amerikaanse overheidsinstellingen
* **mil** Amerikaanse militaire organisaties
* **org** Non-profit organisaties. Heel veel bedrijven die zich bezig houden met open source software hebben een domein onder het org domain, denk bijvoorbeeld aan www.apache.org.

* **int** Internationale organisaties die ontstaan zijn als gevolg van verdragen tussen verschillende nationale overheden. Denk bijvoorbeeld aan www.un.int, de website van de verenigde naties.
* **net** Organisaties die iets te maken hebben met de werking en het gebruik van internet. Denk bijvoorbeeld aan internetaanbieders.
* **eu** Europese organisaties. Deze recente toevoeging aan de DNS-hiërarchie kan gebruikt worden door allerhande Europese bedrijven.

Onder deze top level-domeinen kunnen organisaties en personen hun eigen domein in gebruik nemen. Om dit te doen, is het wel nodig dat dit domein geregistreerd wordt bij de beheerder van het top level-domein. U kunt voor deze dienst echter ook terecht bij uw internetaanbieder. Er zijn in Nederland bijvoorbeeld aardig wat organisaties waar u terecht kunt wanneer u uw eigen DNS-domeinnaam wilt registreren, neem bijvoorbeeld de internetaanbieder www.xs4all.nl Eventueel kunnen onder deze domeinen van organisaties weer nieuwe subdomeinen aangemaakt worden. Of dit het geval is, is afhankelijk van de schaal van het bedrijf dat zo'n internetsite nodig heeft: alleen grote bedrijven werken regelmatig met meerdere subdomeinen. In zo'n domein worden dan tenslotte de computers en andere apparaten die door middel van een DNS naam bereikbaar moeten zijn (printers bijvoorbeeld) ondergebracht.

Dit alles kan er toe leiden dat er bijvoorbeeld een computer is die de naam "www" heeft. Deze computer komt voor in het domein sandervanvugt en dit domein sandervanvugt komt voor onder het topleveldomein nl. Alles achter elkaar wordt dan de volledige naam, ook wel fully qualified domain name (FQDN) van deze computer www.sandervanvugt.nl.

3.1.2 Delegatie van het beheer van de DNS namespace

De DNS namespace wordt wereldwijd niet alleen door IANA beheerd. IANA heeft veel beheerstaken gedelegeerd naar locale instanties. In Europa is bijvoorbeeld de organisatie RIPE (Réseau Internet Pan Européenne) de verantwoordelijke organisatie. Dat betekent dat Europese internetaanbieders hun zaken met RIPE moeten regelen om opgenomen te worden in de wereldwijde DNS hiërarchie. Voor Nederland zit er zelfs nog een organisatie tussen: de Stichting Internet Domeinregistratie Nederland (www.domain-registry.nl). Ook andere landen hebben een dergelijke tussenlooag: zo wordt in België bijvoorbeeld gebruikgemaakt van www.dns.be voor registreren van DNS-namen. Voor eindgebruikers is het aardig te weten dat dergelijke instanties ook een whois-database onderhouden. Dit is voor netwerkbeheerders belangrijke informatie; op basis van deze informatie kunt u namelijk achterhalen wie er voor een bepaald domein verantwoordelijk is. Krijgt u regelmatig spam uit een bepaald Nederlands domein? Raadpleeg dan de whois functie op www.domain-registry.nl om de beheerder van het domein te vragen maatregelen te nemen. Voor België is deze functionaliteit beschikbaar via www.dns.be. In het volgende voorbeeld leert u hoe u de Nederlandse whois-database kunt raadplegen:

1. Activeer de link www.domain-registry.nl in uw browser.
2. Klik op het sysmbool >> dat u ziet naast de functie **.nl domeinnaam nog vrij** en klik de optie **Uitgebreid** aan.
3. Voer het domein in waarvoor u meer informatie wilt, en klik op **zoek** om te starten met zoeken. Na een klein moment ziet u een venster met uitgebreide informatie over het betreffende domein.

Tip! Buiten de diensten die u op internet kunt benaderen om informatie over een bepaald domein te achterhalen, is er ook gewoon op Linux een whois-service aanwezig. Maak

hiervoor gebruik van de opdracht **whois**; met **whois novell.com** achterhaalt u bijvoorbeeld alle informatie over het domein novell.com. De opdracht **whois** werkt erg eenvoudig: u krijgt automatisch contact met de aangewezen whois-server van het domein waarover u meer wilt weten.

***whois Met de opdracht **whois** kunt u gewoon vanaf de Linux commandoregel meer informatie over een DNS-domein achterhalen.

3.1.3 De rol van de naamserver

Om er voor te zorgen dat informatie over computers in een domein kan worden opgevraagd, is het noodzakelijk dat er naamservers zijn. Dit zijn speciale servers die een database bijhouden met daarin alle relevante DNS informatie, zoals onder andere de namen van computers binnen een domein en de IP-adressen die daar bij horen. Deze gegevens worden opgeslagen in Resource Records. Dit betekent dat voor elke computer die aan de hiërarchie toegevoegd moet worden, een nieuw Resource Record aangemaakt moet worden. Een DNS-naamserver kan trouwens gegevens over meerdere domeinen tegelijk bijhouden. Denk daarbij bijvoorbeeld aan het domein van een bedrijf en twee subdomeinen die daaronder voorkomen; bijvoorbeeld het domein azlan.com en de subdomeinen training.azlan.com en distributie.azlan.com. Het totaal van aaneengeschakelde domeinen dat door een naamserver bediend wordt, wordt een DNS-zone genoemd.

Tenslotte moet over de naamserver nog verteld worden dat het mogelijk is meerdere naamservers te installeren. Dit is handig om te regelen, want als er onverhoopt een naamserver uitvalt, kunnen de gegevens altijd nog via de andere server achterhaald worden. Daarbij wordt onderscheid gemaakt tussen de master-naamserver en de slave-naamserver. Als alternatief wordt ook wel gesproken over de primary en de secondary naamserver. De master naamserver is de server die verantwoordelijk is voor inhoud van de DNS-database. Slave-naamservers worden ingezet voor fouttolerantie en om de bereikbaarheid van de DNS-gegevens te vergroten. Slaves zijn naamservers die een kopie hebben van de database van de master. Zij zorgen ervoor dat hun database periodiek bijgewerkt wordt met de laatste wijzigingen. Dit proces wordt zone transfer genoemd. Het aanbrengen van wijzigingen is overigens een éénrichtingsverkeer: wijzigingen worden aangebracht op de master-naamserver en van daaruit gesynchroniseerd naar alle slave-naamservers. Het is niet mogelijk wijzigingen aan te brengen in de database van een slave naamserver.

3.1.4 Delegatie

Zoals uit het bovenstaande blijkt, bestaat de DNS-hiërarchie uit een grote verzameling domeinen die allemaal bediend worden door middel van naamservers. Daarnaast zijn deze naamservers ook nog eens aan elkaar verbonden. Dat wil zeggen dat elke naamserver op de hoogte is van de naamservers die onderliggende domeinen bedienen. In jargon heet het dat subzones gedelegeerd zijn. In principe bedient een naamserver immers alles onder een bepaald domein. Het is echter mogelijk om het beheer van een onderliggend domein verder te delegeren naar een naamserver van een onderliggen domein, zodat elk bedrijf uiteindelijk in elk geval de mogelijkheid heeft zijn eigen deel van de hiërarchie te beheren. In veel gevallen kan echter ook het volledige beheer van een DNS-server uitbesteed worden aan de internetaanbieder. Zeker wanneer u maar een paar servers in uw domein hebt, kan dat een handige oplossing zijn, om die reden komt er bij de meeste kleinere bedrijven dan ook helemaal geen DNS naamserver voor.

3.1.5 De werking van DNS

Als vanaf een werkstation op basis van een DNS-naam iets verstuurd moet worden, wordt de DNS-resolver gebruikt. Dit is de client-component waarmee verwezen wordt naar de DNS-server(s) die gebruikt moeten worden. Op een Linux systeem wordt deze functie vervuld door het configuratiebestand /et/resolv.conf. Hierin staat minstens één IP-adres van een DNS-server die in staat is het IP adres wat bij de gegeven naam hoort te achterhalen. Deze naam, bijvoorbeeld www.sandervanvugt.nl, wordt doorgestuurd naar de naamserver die in de DNS-resolver van de client is opgegeven. Vervolgens kijkt de naamserver in zijn database of hij informatie over de betreffende naam heeft. Als dat niet het geval is, wordt het pakketje in principe doorgestuurd naar een naamserver van het root-domein. Elke DNS-naamserver heeft namelijk wetenschap van een aantal naamservers die het root-domein bedienen. Dit zijn de naamservers die op hun beurt in staat zijn om de naamservers van alle top level-domeinen terug te vinden.

***resolver18 Om contact te kunnen maken met een DNS-server, is een DNS-resolver nodig.

***resolver218 De resolver kan ook vanuit grafische hulpprogramma's geconfigureerd worden.

De naamserver van het root domein zal niet in staat zijn ons voorbeeldpakketje, dat afgeleverd moet worden op www.sandervanvugt.nl, door te sturen. Hij is immers niet verantwoordelijk voor het sandervanvugt.nl-domein, maar als naamserver van het root-domein is hij er alleen verantwoordelijk voor dat het nl-domein teruggevonden kan worden. Hij zal daarom aan de naamserver van de client die het verzoek gedaan had een pakketje terug sturen, waarin vermeld wordt dat hij het bij de naamserver van het nl-domein moet proberen. Ook deze is waarschijnlijk niet in staat om direct het IP-adres van www.sandervanvugt.nl door te geven en zal dus aan de oorspronkelijke naamserver doorgeven dat hij het beter kan proberen bij de naamserver van het domein sandervanvugt.nl. Als het goed is, is deze uiteindelijk op basis van de inhoud van zijn database in staat een pakketje met daarin het gevraagde IP-adres terug te sturen naar de naamserver van de client, die het op zijn beurt weer doorgeeft aan de client zelf.
Het antwoord wordt overigens gelijk een tijdje bewaard op de naamserver van de client; zo wordt voorkomen dat het hele proces herhaald moet worden als er een paar minuten later door een andere gebruiker in hetzelfde netwerk een pakketje verstuurd moet worden naar dezelfde server.

Opmerking: DNS is een complex protocol. Er zijn zeer veel verschillende wijzen waarop DNS-servers geconfigureerd kunnen worden. In dit hoofdstuk is het niet de bedoeling volledig te zijn over alle mogelijkheden, we willen slechts een globaal beeld scheppen waardoor u kunt begrijpen hoe een DNS-naamserver werkt. Raadpleeg voor meer details een van de vele gespecialiseerde boeken die over dit onderwerp geschreven zijn.

3.1.6 Omgekeerde DNS

Naast het hiervoor beschreven proces, waardoor clients IP-adressen kunnen achterhalen die bij bepaalde DNS-namen horen, is het omgekeerde ook mogelijk. Dat wil zeggen dat dus de naam achterhaald kan worden die bij een zeker IP-adres hoort. Dit is vooral voor sommige beveiligingssystemen noodzakelijk. Denk daarbij aan lijstjes waarin staat dat alle hosts uit het domein *.microsoft.com geen toegang hebben. Een ander voorbeeld is de verificatie die meestal plaatsvindt bij het versturen van mail waarbij gecontroleerd wordt of een bericht uit een geldig DNS-domein afkomstig is. Er moet dan natuurlijk wel een manier zijn om te achterhalen of een bepaalde hostnaam inderdaad thuishoort binnen dat domein. Dit gebeurt

doordat in de in-addr.arpa-hierarchie bijgehouden wordt welke IP-adressen gebruik maken van welke hostnamen. Ook deze hiërarchie bestaat uit een verwijzing van naamservers. Om een volledige DNS-server te kunnen configureren, is het van belang dat u ook deze omgekeerde naamresolutie regelt op uw DNS-server.

Dit werkt als volgt: stel dat achterhaald moet worden welke naam gebruikt wordt door host 194.95.236.39. Onder de in-addr.arpa-hiërarchie komt een DNS-structuur voor waarin elk mogelijk IP-adres een domein is. Om dit te bereiken, komen de eerste bytes van het IP-adres direct onder in-addr.arpa voor. Dat wil zeggen, dat in dit geval eerst achterhaald moet worden welke naamserver informatie heeft over namen die gebruikt worden door alle computers waarvan het adres met 194 begint. Vervolgens moeten de naamservers achterhaald worden die informatie hebben over alle adressen die met 194.95 beginnen. Dan moet de naamserver achterhaald worden die informatie heeft over alle adressen die met 194.95.236 beginnen. Deze naamserver heeft uiteindelijk een database, waarin voor elke computer een zogenaamd ptr-record voorkomt. In dit ptr-record wordt weergegeven welke naam door elke node in dat netwerk gebruikt wordt.

Als u hier gebruik van wilt maken, houdt dan rekening met een belangrijke beperking; in-addr.arpa werkt namelijk alleen met volledige netwerkadressen. Dit betekent dat er een in-addr.arpa naamserver kan zijn voor het netwerk 194.95.236.0, maar dat het niet mogelijk is een in-addr.arpa-server in het leven te roepen voor het netwerk 194.95.236.32 waarbij gebruik gemaakt wordt van het niet-standaard subnetmasker 255.255.255.240.

3.1.7 De inhoud van de DNS-database

Zoals u uit het voorgaande af hebt kunnen leiden, is het uiteindelijk de DNS-database waardoor bepaald wordt of bepaalde gegevens wel of niet achterhaald kunnen worden. In deze database komen verschillende soorten resource records voor. We geven hier een overzicht van de belangrijkste daarvan. Verderop leest u hoe u deze resource records op een DNS-server op Linux kunt gebruiken.

resource record	Toepassing
a (address)	Wordt gebruikt om IP-adressen aan een bepaalde naam te verbinden. U moet voor elke server waarvan de naam door middel van DNS achterhaald kan worden een A-record aanmaken. Het A resource record is het meest belangrijke resource record omdat het gebruikt wordt om namen aan adressen te koppelen.
Cname	Wordt gebruikt om aliassen (alternatieve namen) voor bepaalde computers te gebruiken. Dit is handig wanneer één server meerdere rollen vervult. De afkorting staat overigens voor ‘canonical name’.
prt (pointer)	Wordt gebruikt om namen aan bepaalde IP-adressen te verbinden. Toegepast in in-addr.arpa. Hierdoor kan achterhaald worden welke naam bij een bepaald IP-adres hoort.
ns (naamserver)	Toegepast om andere naamservers voor hetzelfde domein te identificeren, of om naamservers voor onderliggende domeinen te identificeren.
Mx	Mail-exchanger; ter identificatie van mailservers. Dit is een essentieel resource-record wanneer u wilt dat anderen mail kunnen versturen op basis van uw eigen domeinnaam.
soa (start of authority)	Toegepast om aan te geven welke naamserver verantwoordelijk is voor de betreffende zone.

Srv	Een relatief nieuw record waarmee services gelokaliseerd kunnen worden in een netwerk.
Txt	Dit record wordt niet echt veel gebruikt, maar kan op een handige wijze worden ingezet om andere dan standaard informatie toe te voegen aan de DNS-database.

Tabel 3.1 Overzicht van de belangrijkste resource records.

3.1.8 Beheer van de DNS database

Het beheer van een DNS database, komt er in wezen op neer dat deze database handmatig bijgewerkt moet worden. Als er een nieuwe computer aan een domein wordt toegevoegd, moet er dus handmatig een nieuwe resource record voor die computer worden aangemaakt. Gelukkig is dit niet de enige optie. Als alternatief kan gebruikgemaakt worden van dynamic DNS. Hierbij werken de DHCP-server en de DNS server samen zodat de DNS database automatisch bijgewerkt kan worden wanneer een nieuw IP-adres aan een bepaalde host wordt toegekend.

3.2 Configuratie van DNS

Voordat we beginnen, eerst iets over versies. Er worden vrij algemeen nog twee verschillende versies van DNS gebruikt. De configuratie van deze verschillende versies is totaal verschillend. Beide versies hebben echter met elkaar gemeen dat ze ontleend zijn aan de oer-DNS, de Berkeley Internet Name Daemon. De meest recente en momenteel ook waarschijnlijk meest gebruikte versie is BIND 9. Bind 9 is in feite een doorontwikkeling van BIND 8. Deze versie van de Berkeley Internet Name Daemon werkt namelijk op een heel andere wijze als zijn voorganger BIND 4. De meeste Linux distributies installeren standaard BIND 9. Dit hoofdstuk gaat dan ook over de configuratie van BIND 9. U kunt eenvoudig controleren of u hier gebruik van maakt; als op uw systeem een bestand met de naam named.conf bestaat, wordt BIND 8 gebruikt, als dit bestand niet bestaat, maar wel een bestand genaamd “named.boot”, gebruikt u BIND 4, de kans dat dit het geval is, is zeer klein. Als alternatief kunt u ook de opdracht **named –v** gebruiken om te achterhalen welke versie van BIND op uw systeem gebruikt wordt.

Tip! Het kan zijn dat uw server nog gebruikmaakt van BIND 4. Is dit het geval? Overweeg dan toch over te gaan naar een meer recente versie van BIND. BIND 4 kent namelijk een aantal beveiligingsproblemen die in latere versies zijn opgelost. Voor een zo veilig mogelijke omgeving wordt stellig aangeraden gebruik te maken van de meeste recente BIND versie.

U kunt nu lezen hoe u zelf een DNS-server kunt configureren. We bespreken hiervoor twee verschillende manieren. Om te beginnen leest u over de configuratie van een cache-only naamserver. Een dergelijk type naamserver kan ingezet worden om het DNS-proces te versnellen: uw DNS-server wordt dan als lokale cache in het netwerk gebruikt zodat voor namen die regelmatig opgevraagd worden niet steeds de DNS-hiërarchie op internet benaderd hoeft te worden. Vervolgens kunt u lezen over het echte werk: de configuratie van een volledig werkende DNS-server.

3.2.1 Configuratie van een cache-only naamserver

Als voor elk DNS-verzoek een computer op internet benaderd moet worden, kost dat tijd. Dit zal zeker een relevant probleem zijn als u gebruik maakt van een langzamere verbinding zoals een ISDN-verbinding. Hier kan op vrij eenvoudige wijze wat aan gedaan worden door een cache-only naamserver te configureren. Het principe hiervan is vrij eenvoudig; u zorgt er voor dat alle name-resolving verzoeken naar uw eigen server gestuurd worden; hierop draait

natuurlijk de DNS-server named. Dit proces gaat vervolgens met behulp van de informatie over de DNS-rootservers die op de machine voorkomt op zoek naar het IP-adres dat bij de naam hoort die achterhaald moet worden. Als de server dat heeft uitgevonden, bewaart hij dat IP-adres in zijn geheugen, zodat het de volgende keer veer sneller aan de client doorgegeven kan worden. U kunt zich voorstellen dat dit voordelen heeft wanneer meerdere gebruikers in uw netwerk vaak naar dezelfde server gaan. Het gebruik van een cache-only naamserver kan tevens goed gecombineerd worden met het gebruik van een Proxy-server.

De basis van de configuratie van uw DNS-server is het bestand /etc/named.conf. Hierin kan aangegeven worden welke DNS-zones door de server bediend worden. Voor een cache-only naamserver ziet dit bestand er meestal eenvoudig uit:

```
#/etc/named.conf
options {
        directory "/var/named";
        };

zone "." {
        type hint;
        file "root.hints";
};

zone "localhost"in {
        type master;
        file "localhost.zone";
};

zone "0.0.127.in-addr.arpa"{
        type master;
        file "127.0.0";
};
```

Het eerste dat in dit bestand wordt ingesteld, is de naam van de directory waarin de DNS-configuratie bewaard wordt; in dit geval dus /var/named. Dit is vrijwel altijd de locatie waar overige DNS-configuratiebestanden gevonden worden, maar u bent natuurlijk vrij van deze standaard af te wijken als u dat handig vindt. Alle andere namen van directory's en bestanden die gedaan worden, zijn aan deze directory gerelateerd.

Vervolgens wordt opgegeven hoe informatie over de zone ".", of wel de root van de DNS-hiërarchie achterhaald kan worden. Hiervoor wordt gebruikgemaakt van het bestand met de naam "root.hints"; dit is overigens een vrij algemeen voorkomende naam die voor een bestand met deze functie gebruikt wordt. In dit bestand staan de namen en adressen van naamservers die het root-domein bedienen. Als dit bestand op uw systeem niet voorkomt, geen probleem, u kunt het aanmaken met de opdracht dig; zo zorgt **dig @e.root-servers.net . ns > root.hints** ervoor dat in de huidige directory een bestand "root.hints" wordt aangemaakt. Met de opdracht dig wordt in dit geval als het ware de database van een internet naamserver leeggezogen. Dit bestand wordt gevuld met de informatie die verkregen is van de root-naamserver met de naam e.root-servers.net. (U hebt natuurlijk wel een goed geconfigureerde DNS-resolver nodig om dit te kunnen doen, raadpleeg hoofdstuk 1 voor meer informatie daarover).

De inhoud van root.hints is ongeveer als volgt en zorgt ervoor dat naamservers van het rootdomein teruggevonden kunnen worden:

```
.                       6D IN NS    J.ROOT-SERVERS.NET.
.                       6D IN NS    K.ROOT-SERVERS.NET.
.                       6D IN NS    L.ROOT-SERVERS.NET.
.                       6D IN NS    M.ROOT-SERVERS.NET
(…)
J.ROOT-SERVERS.NET.   5w6d16h     IN A   198.41.0.10
K.ROOT-SERVERS.NET.   5w6d16h IN A       193.0.14.129
L.ROOT-SERVERS.NET.   5w6d16h     IN A   198.32.64.12
M.ROOT-SERVERS.NET    5w6d16h     IN A   202.12.27.33
```

Houd er trouwens rekening mee dat de root-servers op internet soms aan verandering onderhevig zijn; u doet er goed aan de opdracht **dig** zo af en toe te herhalen om ervan verzekerd te zijn dat u nog met goede informatie werkt. Om ervoor te zorgen dat u altijd automatisch volledig bijgewerkt bent, zou u de opdracht bijvoorbeeld uit kunnen laten voeren als een cron-job.

De volgende stap bestaat eruit dat u in de directory /var/named/ een bestand met de naam 127.0.0 plaatst. In dit bestand zorgt u ervoor dat omgekeerde naamresolving geregeld wordt zodat IP-adressen vertaald kunnen worden in een DNS-naam. Het is in dit geval heel eenvoudig: alleen naam resolving voor IP-adres 127.0.0.1 wordt namelijk geregeld, maar dit is wel noodzakelijk voor goede functionaliteit. Dit bestand krijgt voor het domein sandervanvugt.nl de volgende inhoud:

```
@               IN              SOA   ns.sandervanvugt.nl.  root.sandervanvugt.nl. (
; Serial
8H      ; Refresh
2H      ; Retry
1W      ; Expire
1D)     ; Minimum TTL

NS      ns.sandervanvugt.nl.
1                               PTR   localhost
```

Als dat gebeurd is, moet u er voor zorgen dat in /etc/resolv.conf een verwijzing staat naar de juiste naamserver; uw eigen naamserver dus. Hiermee zorgt u ervoor dat uw DNS-server ook DNS-client van zichzelf wordt. Dit klinkt onlogisch, maar is toch volkomen logisch omdat u er op deze wijze voor zorgt dat alle DNS-aanvragen via de DNS-server verlopen, anders zou het immers weinig zin hebben een cache-only nameserver in het leven te roepen. U regelt dit door een regel

```
nameserver     127.0.0.1
```

in dit bestand op te nemen. U kunt in /etc/resolv.conf ook nog andere informatie opnemen, zoals een regel “search” waarmee u aangeeft in welke domeinen standaard gezocht moet worden, maar dit is niet noodzakelijk. Naast /etc/resolv.conf moet u ook even controleren of in /etc/nsswitch.conf een regel voorkomt met de inhoud

hosts: files dns

Deze regel zorgt ervoor dat altijd eerst het locale bestand /etc/hosts en dan pas de DNS hiërarchie doorzocht wordt. Op deze wijze kunt u ervoor zorgen dat veelgebruikte computernamen ook achterhaald kunnen worden zonder dat de DNS geraadpleegd is. Het verdient echter aanbeveling dit niet te vaak te doen en vooral de naam resolving te regelen door middel van DNS. Als dit allemaal gebeurd is, bent u klaar om de DNS daemon named te starten. Als u het fraai wilt doen, gebruikt u hiervoor het opstartbestand dat voorkomt in de directory /etc/init.d; u geeft dan dus de opdracht **/etc/init.d/named start**. Controleer wel even dat dit inderdaad ook de juiste syntaxis is die op uw systeem gebruikt moet worden.

Om te kijken of het werkt, kunt u nu de opdracht **nslookup** geven. Ondanks dat dit commando "deprecated" is, kunt u het voorlopig nog steeds gebruiken om te kijken of uw DNS-server goed functioneert. Geef bijvoorbeeld de opdracht nslookup www.novell.com om te achterhalen op welke wijze deze naam door DNS achterhaald wordt. In het resultaat van deze opdracht ziet u welke server als naamserver gebruikt wordt; hier zou nu 127.0.0.1 moeten staan. U kunt trouwens de prompt die door nslookup gestart wordt weer afsluiten met de opdracht "exit".

U kunt dit alles trouwens nog aanzienlijk versnellen. Op dit moment maakt u namelijk gebruik van de naamservers van het root-domein. Hier valt aanzienlijke winst te behalen als u de aanvraag eerst doorstuurt naar een andere naamserver, bijvoorbeeld van uw internetaanbieder. Die heeft namelijk al een aanzienlijke cache waaruit aanvragen vertaald kunnen worden en zo wordt de weg verkort die op internet afgelegd moet worden om een naam te kunnen resolven. De naamserver van de internetaanbieder wordt dan als zogenaamde "DNS-forwarder" gebruikt. U kunt dit bewerkstelligen door in /etc/named.conf in de sectie "options" de volgende regels op te nemen:

```
forward first;
forwarders {
        10.0.0.1;
        10.1.0.1;
};
```

Let erop dat u voor de verwijzing voor de forwarder wel gebruik maakt van valide IP-adressen. Het is handig hier de IP-adressen van de DNS-servers van uw internetaanbieder te plaatsen. Denk eraan het proces named opnieuw te starten voordat u gaat testen of dit werkt. De beste manier om dit te doen, is met behulp van de opdracht **killall –HUP named**, hiermee forceert u named namelijk om opnieuw zijn configuratie uit te lezen.

Tip! Leuk natuurlijk zo'n naamserver, maar wat wanneer er zich corrupte entries bevinden in de cache van de naamserver? In dat geval doet u er goed aan de naamserver opnieuw te starten. gebruik hiervoor bijvoorbeeld **/etc/init.d/named restart** of **killall –HUP named**. Gebruikt u BIND 9.2 of later? Dan kunt u gebruikmaken van de opdracht **rndc restart** om de cache van de naamserver snel te legen. Ook een heel aardige mogelijkheid is dat u de inhoud van de cache gewoon kunt bekijken. Gebruik hiervoor de opdracht **rnnc dumpdb** (BIND 9) of **ndc dumpdb** (BIND 8).

3.2.2 Configuratie van een SOA

Het voorgaande is leuk, maar wellicht hebt u binnen uw netwerk ook computers waarvan u de adressen wilt kunnen achterhalen door gebruik te maken van DNS. Dan is het handig zelf een DNS-domein in het leven te roepen. Om dit te doen, moet u een SOA (Start Of Authority) definiëren. Deze procedure wordt in deze paragraaf uitgelegd. We zullen hiervoor een DNS-server configureren voor het domein sandervanvugt.nl. Uiteraard kunt u het voorbeeld dat hier gegeven wordt gewoon volgen. Denk er dan alleen wel even aan een andere naam te gebruiken als sandervanvugt.nl. Deze naam is namelijk al in gebruik en geregistreerd door de auteur van dit boek. Wilt u uw DNS-naam ook registreren? Raadpleeg dan uw internetaanbieder, deze kan u hier meer informatie over geven.

Het begin van de configuratie die nodig is om een eigen naamserver in te richten, is in de voorgaande paragraaf al gedaan. Het gaat daarbij om de regels

```
zone    "0.0.127.in-adddr.arpa" {
        type master;
        file "127.0.0";
};
```

in het bestand /etc/named.conf. Hiermee wordt de in-addr.arpa configuratie voor het lokale domein geregeld. Zoals u eerder hebt kunnen lezen, wordt in-addr.arpa gebruikt om omgekeerde DNS-resolving te doen; dat betekent dat hierdoor de naam die bij een IP-adres hoort achterhaald kan worden. Hou er trouwens rekening mee dat de in-addr.arpa-zones die daarvoor nodig zijn gebruik maken van omgekeerde IP-adressen. Daarom wordt de zone gedefinieerd als 0.0.127.in-addr.arpa en niet als 127.0.0.in-addr-arpa. Er wordt trouwens ook weer gebruik gemaakt van het bestand 127.0.0 dat in de vorige paragraaf is aangemaakt. Laten we nog eens kijken naar de inhoud van dit bestand:

```
@       IN      SOA     ns.sandervanvugt.nl   root.sandervanvugt.nl. (
                                1       ;       Serial
                                8H      ;       Refresh
                                2H      ;       Retry
                                1W      ;       Expire
                                1D)     ;       Minimum TTL

                                NS      ns.sandervanvugt.nl.
1                               PTR     localhost
```

In dit bestand komen drie resource records voor; een voor de SOA, een voor de NS en een PTR-resource record. Als allereerste is wordt de Start of Authority (SOA) gedefinieerd. Hiermee wordt aangegeven welke naamserver uiteindelijk verantwoordelijk is voor de inhoud van de database, ofwel welke de master naamserver is. In het bestand /etc/named.conf werd overigens bij de definitie van deze zone met de aanduiding "type master" ook al geregeld dat deze server master is voor de betreffende zone. De definitie van de SOA gebeurt met de regel

```
@               IN      SOA     ns.sandervanvugt.nl.   root.sandervanvugt.nl. (
```

De "@" kan hier geïnterpreteerd worden als "de huidige zone"; deze regel betekent dus eigenlijk dat ns.sandervanvugt.nl naamserver is voor 0.0.127.in-addr.arpa. Let ook even op de manier waarop de naam van de naamserver wordt opgeschreven: "ns.sandervanvugt.nl.", met een punt aan het einde. De punt aan het einde geeft aan dat het hier een absolute naamgeving,

dat wil zeggen gerelateerd aan de root van de DNS-hierarchie, betreft. Het gebruik van deze punt is belangrijk. Als in de naamgeving namelijk geen punt aan het einde gegeven wordt, wordt de naam vaak gerelateerd aan de huidige domeinnaam. De naam ns.sandervanvugt.nl in dit voorbeeld wordt als u deze fout maakt dus geïnterpreteerd als ns.sandervanvugt.nl.sandervanvugt.nl. Let er dus goed op of u al dan niet punten aan het einde van namen moet gebruiken!
Wat verder opvalt is dat in de SOA-regel het mailadres van de beheerder van deze naamserver gegeven wordt; waarschijnlijk had u echter root@sandervanvugt.nl. verwacht. Wen er maar aan; in plaats van de @ wordt gewoon een punt gebruikt. Het DNS-protocol weet namelijk geen raad met de @-tekens die vanuit de mail-wereld afkomstig zijn. Dit komt omdat DNS gespecificeerd is in een tijd dat mail nog niet zo wijdverbreid was als dat nu het geval is.

Vervolgens is er een rijtje met instellingen die bepalen hoe lang entry's in de database houdbaar zijn en hoe vaak slave- en master-naamservers met elkaar moeten synchroniseren. Als eerste is er de regel "1 ; serial". Dit is het versienummer van dit zone-bestand. Als gegevens in de database worden aangepast, moet dit versienummer aangepast worden; hieraan kunnen slave-naamservers zien dat er wijzigingen geweest zijn. U zult er zelf voor moeten zorgen dat dit versienummer aangepast wordt, alleen bij het gebruik van een goede grafische front-end, wordt het door deze front-end geregeld.

Daarna wordt in de regel "8H ; Refresh" aangegeven hoe vaak de slave-naamserver contact op moet nemen met de master om te kijken of er wijzigingen zijn. De waarde die hier voor staat ingesteld is 8H, ofwel 8 uur, hetgeen vrij vaak is. In de meeste netwerken is de DNS-database immers niet zo heel erg dynamisch. De regel die daar op volgt bepaalt wanneer de slave-naamserver het weer moet proberen als het de eerste keer niet lukt; na twee uur dus.

Vervolgens is er de maximale houdbaarheid van een record op de slave-naamserver. Het kan namelijk voorkomen dat de master gedurende een hele tijd niet benaderd kan worden om te kijken of alle gegevens nog correct zijn. In dat geval bepaalt deze waarde hoe lang de gegevens op de slave bewaard blijven. In dit voorbeeld is dat dus een week. Deze instelling is vooral interessant voor internetaanbieders die de rol van slave vervullen voor de master-server van hun klant. Wanneer de klant failliet gaat en de server dus verdwijnt, verdwijnen automatisch na een week alle gerelateerde entries uit de database bij de internetaanbieder. Als laatste tijdsinstelling is er dan tenslotte de "Minimum TTL". Dit is de waarde dat een entry op een naamserver in de cache bewaard mag blijven zonder dat gekeken wordt of er iets gewijzigd is. Het zal u niet verbazen dat "1D" verwijst naar een dag.

Dan wordt de Name Server (NS) gedefinieerd, ook weer voor het huidige domein. Eigenlijk staat er dus volledig "0.0.127.in-addr.arpa. IN NS ns.sandervanvugt.nl". Als er meerdere naamservers zijn, kunnen hier ook de namen van die andere naamservers gegeven worden. Let er dan wel op dat er ook een adresveld in de DNS-database moet voorkomen op basis waarvan die naamservers benaderd kunnen worden!

Het verschil tussen de SOA en de NS is dat de SOA gebruikt wordt om te definiëren wie er verantwoordelijk is voor het betreffende domein; ofwel wie is de master-naamserver, wat is het adres van de beheerder van deze server voor het geval dat er iets mis gaat enzovoorts. Daarnaast worden bij de SOA ook een paar parameters gegeven die bepalen hoe caching plaatsvindt en hoe synchronisatie met slave-naamservers gebeurt. De NS-entry geeft aan welke servers er zijn die iets doen met de inhoud van de database.

De laatste relevante regel definieert het pointer-record; dit is het algemene record waarmee wordt aangegeven welk IP-adres welke hostnaam heeft. De regel

```
1                       PTR                     localhost.
```

betekent dus dat in het netwerk 127.0.0 de host met het adres 1 bekend staat als “localhost”.

Als deze wijzigingen zijn toegevoegd, kunt u met nslookup controleren dat het werkt. Geef op de prompt het adres van de locale computer (127.0.0.1), nslookup zou terug moeten komen met de naam “localhost”.

Toevoegen van een zone deel 1
Als al het voorwerk gedaan is, kan een nieuwe zone toegevoegd worden. Het werk hiervoor begint in het hoofdconfiguratiebestand /etc/named.conf. De zone sandervanvugt.nl kan als volgt toegevoegd worden:

```
zone “sandervanvugt.nl” {
        notify no;
        allow update { 192.168.0.1; };
        type master;
        file “sandervanvugt.nl”;
};
```

Deze regels definiëren dus een nieuwe zone, geven aan dat de huidige server daar master voor is en specificeren verder dat de data voor deze zone voorkomen in het bestand /var/named/sandervanvugt.nl. Dit laatste bestand zal overigens handmatig aangemaakt moeten worden. Daarnaast wordt met “notify no” aangeduid dat er geen slave nameservers zijn die automatisch van wijzigingen op de hoogte gesteld moeten worden. De parameter notify no heeft zin als u netwerkverkeer tot een minimum wilt reduceren. In andere gevallen is het echter aan te raden gebruik te maken van de optie notify yes;. Hiermee zorgt u ervoor dat alle servers die voor een zone bekend zijn als nameservers automatisch op de hoogte gesteld worden wanneer er wijzigingen zijn geweest. Dit is handig omdat een slave server in dat geval eerder op de hoogte gesteld wordt van een wijziging, deze optie moet echter wel door de software op de slave server ondersteund worden. Als laatste speciale optie wordt in dit voorbeeld gebruikgemaakt van de optie allow update. Deze optie is nodig voor ondersteuning van dynamic DNS waarbij de database van de DNS-server door een DHCP-server bijgewerkt kan worden. Als u van deze mogelijkheid gebruik wilt maken, moet u als argument het IP-adres van de betreffende DHCP-server opgeven.

In het voorgaande stukje voorbeeldcode uit named.conf, wordt een zonefile sandervanvugt.nl aangeroepen. Dit bestand kan bijvoorbeeld de volgende inhoud hebben:

```
@               IN      SOA     ns.sandervanvugt.nl. root.sandervanvugt.nl (
                                200407271       ; serial, todays date + serial #
                                8H              ; refresh, seconds
                                2H              ; retry, seconds
                                1W              ; expire, seconds
                                1D )    ; minimum, seconds
;
```

```
                        NS      ns                  ; Inet Address of name server
                        MX      10 mail.sandervanvugt.nl.       ; Primary Mailsrvr
                        MX      20 mail.arianus.nl. ; Secondary Mailsrvr
;
localhost       A       127.0.0.1
linux           A       192.168.0.10
Julia           A       192.168.0.50
oes-linux       A       192.168.0.100
oes-netware     A       192.168.0.110
squid           A       192.168.0.200
squid           A       192.168.0.201
ns              CNAME       linux
mail            CNAME       linux
```

Ook in dit bestand wordt eerst de SOA gedefinieerd, met het mailadres van de beheerder. Dit kan elk willekeurige mailadres zijn, het hoeft dus niet beslist voor te komen in hetzelfde domein. Hou er trouwens ook rekening mee dat in de eerste regel de naam van de SOA gegeven moet worden, verderop in dit zelfde bestand wordt vervolgens een adres-record aangemaakt voor de SOA.

Nieuw zijn hier de Resource Records "MX", waarmee verwezen wordt naar de mailserver die in dit domein gebruikt wordt. Als er dus een mailbericht binnenkomt voor nadja@sandervanvugt.nl, moet dat bericht doorgezonden worden naar mail.sandervanvugt.nl. Dit is de primaire mailserver, omdat er een lager getal (10) bij opgegeven staat. Daarnaast is er ook een secondaire mailserver, die benaderd kan worden als het met de eerste server om welke reden dan ook niet lukt. Het bestand eindigt tenslotte met de adresvelden voor de computers die voorkomen in dit domein.

In het voorgaande ziet u trouwens ook een aardig voorbeeld van wat een primitieve vorm van load balancing genoemd kan worden. De host squid wordt namelijk tweemaal gedefinieerd: een keer met IP-adres 192.168.0.200 en een keer met IP-adres 192.168.0.201. Om te bepalen welk IP-adres een client nu te horen krijgt, wordt het principe van round robin toegepast: de ene keer wordt de eerste gebruikt, de volgende keer de tweede en zo benadert dus de ene helft van de clients host squid op het ene IP-adres en de andere host benadert hem op het andere. Dit is echter niet te beschouwen als volwaardige load balancing omdat er bij deze oplossing totaal niet gekeken wordt hoe druk een bepaalde server het heeft, het is echter een heel aardig begin.

Nu u de configuratie van uw zone geregeld hebt, kunt u de named server forceren zijn configuratie opnieuw te lezen door **killall –HUP named** te gebruiken. Vervolgens kijkt u of het werkt door een van de computers waarvan de naam gegeven is door middel van ping te benaderen; als alternatief kunt u de inhoud van de database uitvragen door in nslookup de volgende commando's te geven:

```
nslookup
>       set q=any
>       www.novell.com
```

Grote kans dat u helemaal geen fraaie melding krijgt over DNS-naamservers die hun werk goed doen, maar alleen maar een foutmelding:

*** Can't find server name for address 192.168.0.10: Non-existent domain

Deze foutmelding is eenvoudig te verklaren. Een goede DNS-server regels namelijk niet alleen dat namen in IP-adressen vertaald kunnen worden, maar ook dat IP-adressen in namen kunnen worden omgezet door middel van de in-addr.arpa configuratie.

Toevoegen van een nieuwe zone deel 2
Veel services maken gebruik van lijstjes met daarin de namen van computers om te bepalen of een computer wel of geen gebruik mag maken van de service. Een van die services is de DNS-server zelf. Hiermee bestaat echter een probleem; aan binnenkomende pakketjes is niet direct te zien wat de naam van de zendende computer is; u kunt alleen maar het IP-adres zien. Om ervoor te zorgen dat hier de naam bij gevonden kan worden, gebruikt u de DNS-zone in-addr.arpa. in-addr.arpa heeft echter een tekortkoming waarmee u rekening moet houden; er bestaat geen mogelijkheid om aan te geven dat een ander dan het standaard subnetmasker gebruikt wordt. In-addr.arpa gaat dus altijd uit van het standaard klasse A, B of C adres en het is niet mogelijk op te geven dat het alleen maar de adressen 192.168.192.32/27 mag bedienen. Om in-addr.arpa te activeren voor netwerk 192.168.0.0, moeten om te beginnen de volgende regels in named.conf opgenomen worden:

```
zone  "0.168.192.in-addr.arpa" {
        notify no;
        type master;
        file "192.168.1";
};
```

We hebben iets dergelijks al eerder gezien, namelijk toen het bestand 127.0.0 aangemaakt werd. De inhoud van het bestand 192.168.1 lijkt hier dan ook veel op.

```
$TTL 1W
@           IN      SOA     ns.sandervanvugt.nl. root.sandervanvugt.nl. (
                            200407271 ; serial
                            8H      ; Refresh
                            2H      ; Retry
                            1W      ; Expire
                            1D )    ; Minimum TTL

            IN      NS      ns.sandervanvugt.nl.

10          IN      PTR     linux.sandervanvugt.nl.
50          IN      PTR     Julia.sandervanvugt.nl.
100         IN      PTR     oes-linux.sandervanvugt.nl.
110         IN      PTR     oes-netware.sandervanvugt.nl.
200         IN      PTR     squid.sandervanvugt.nl.
201         IN      PTR     squid.sandervanvugt.nl.
```

Wanneer het in-addr.arpa bestand naar bovenstaand model is aangemaakt, kan de naamserver-daemon weer opnieuw gestart worden en kunt u met nslookup controleren of alles er nu goed uitziet. Als het goed is, ziet u nu geen foutmeldingen meer:

```
>       192.168.0.10
Server: ns.sandervanvugt.nl
Address:        192.168.0.10

Name: ns.sandervanvugt.nl
Address:        192.168.0.10
>
```

3.2.3 Delegatie van een subzone

Zonder dat de internetaanbieder ook maar ergens van op de hoogte is, hebt u nu een volledig werkende interne DNS-server geconfigureerd. Met behulp van deze DNS-server kunt u servers op uw eigen netwerk benaderen op basis van hun naam en daarnaast kunt u ook nog alle servers op internet benaderen. Er ontbreekt echter nog wel iets aan. Het zou ook heel leuk zijn wanneer de gebruikers op internet nu ook uw server kunnen benaderen. Wellicht is in het bovenstaande domein een server met de naam www.sandervanvugt.nl waarop de meest geweldige informatie voorkomt, of moet er mail verstuurd kunnen worden naar alex@sandervanvugt.nl of franck@sandervanvugt.nl. Dat werkt allemaal alleen wanneer de internetaanbieder ook weet dat u bestaat.

Het eerste dat hiervoor nodig is, is een geregistreerde DNS domeinnaam. Elke internetaanbieder kan u hier aan helpen. Vervolgens moet er een “delegation of subzone authority” geregeld worden. Dit betekent dat op de DNS-server van de internetaanbieder een verwijzing gemaakt wordt naar uw domein en het IP-adres van uw naamserver. Veel internetproviders bieden overigens aan hun gebruikers een mogelijkheid om dit vanaf de service-pagina’s helemaal zelf te regelen. Daar is dus nog geen telefoontje voor nodig.

Een zelfde soort verhaal doet zich voor wanneer u binnen uw DNS-domein een subzone aan wilt maken. Zo zou het bestand waarin de subzone “franck” bekend gemaakt wordt er op de naamserver van het domein sandervanvugt als volgt uit kunnen zien:

```
@       IN      SOA         ns.sandervanvugt.nl.  root.sandervanvugt.nl. (
                            200401271   ;       serial
                            8H          ;       refresh
                            2H          ;       retry
                            1W          ;       expire
                            1D)         ;       minimum

                            NS      ns
        franck          NS      ns.franck.sandervanvugt.nl.
;
localhost               A       127.0.0.1
ns                      A       10.0.0.1
ns.franck               A       192.168.192.2
…
```

Als eerste gaat het hier natuurlijk om de regel “franck NS
ns.franck.sandervanvugt.nl.”. Hiermee wordt aangegeven dat er een domein “franck” onder dit domein voorkomt en dat de volledige naam van de naamserver van dit domein “ns.franck.sandervanvugt.nl” is. Let er trouwens wel op dat een paar regels verderop het adres van deze ns.franck.sandervanvugt.nl gegeven wordt. Het is van essentieel belang dat dit gebeurt; in de zone sandervanvugt.nl moet namelijk wel bekend zijn hoe de subzone “franck” bereikt moet worden.

Ook als er op de naamserver van de bovenliggende zone geen delegatie gedaan wordt kunt u nog wel een naamserver voor een eigen zone inrichten. Zo kunnen clients eerst naar uw DNS-server gaan en vervolgens via uw DNS-server naar internet. Daarbij zorgt uw DNS-server nog voor een stuk extra snelheid ook doordat hij aan caching doet. Het nadeel is wel dat niemand van internet uw DNS-server kan benaderen om informatie te vragen over de hosts die voorkomen binnen uw netwerk. Maar misschien wilt u dat ook wel helemaal niet?

3.2.4 Configuratie van een slave-server

Het is natuurlijk leuk als u een master-naamserver hebt, maar als deze ontploft, kan niemand nog de IP-adressen van de computers in uw domein achterhalen. Daarom is het erg de moeite waard ook een slave-server in het leven te roepen. Dit is een server die ook benaderd kan worden om dezelfde gegevens te achterhalen. Normaliter haalt deze server door middel van een proces dat zone transfer genoemd wordt, af en toe de gegevens op bij de master-server. We zullen nu bekijken wat er moet gebeuren om een slave-server in het leven te roepen.

Het eerste wat u op de slave-server moet doen, is het algemene configuratiebestand named.conf bewerken. Hierin kunnen bijvoorbeeld de volgende regels voorkomen:

```
zone    “sanderdervanvugt.nl” in {
        type slave;
        file “slave/sandervanvugt.nl”;
        masters { 192.168.0.10; };
};
```

Hiermee wordt aangegeven dat deze server slave is voor het domein “sandervanvugt.nl”, dat hij daarvoor een bestand met de naam “slave/sandervanvugt.nl” bijhoudt en dat hij gebruik maakt van de master-server die bereikbaar is op 192.168.0.10. U kunt hier trouwens ook de naam opgeven van een andere slave-server, dat maakt niet uit. Een slave-server kan namelijk overal zijn database van ophalen als er op de master tenminste geen beveiligingsoptie is die dit verbiedt.

Vervolgens moet u ervoor zorgen dat de benodigde gegevens op de slave aanwezig zijn. Als eerste is er het bestand 127.0.0, waarmee de in-addr.arpa zone voor het locale domein bediend wordt. Deze kan net zo goed lokaal op de slave aangemaakt worden; deze gegevens veranderen toch nooit. Hetzelfde geld voor het bestand waarin de rootservers gespecificeerd worden. U kunt deze bestanden gewoon letterlijk vanaf de master-server kopiëren.

Als dit gebeurd is, kunt u de naamserver starten. Hij neemt dan zelf contact op met de master-server en haalt daar de gegevens van binnen. Het lijkt misschien weinig, maar meer dan dit is er voor de configuratie van een slave-server niet nodig. Het kan natuurlijk ook eens voorkomen dat u een zonetransfer wilt forceren. Dat kan: de odpracht die u hiervoor gebruikt is afhankelijk van de BIND versie die u gebruikt. Maakt u gebruik van BIND 8? Gebruik dan

ndc reload naam-van-de-zone, als u gebruikmaakt van BIND 9 (wat meestal het geval zal zijn) gebruikt u de optie **rdnc refresh naam-van-de-zone**. Voor de voorbeeldzone sandervanvugt.nl die in dit hoofdstuk gebruikt wordt, voert u dus de opdracht **rdnc refresh sandervanvugt.nl** uit.

3.2.5 Werken met forwarders

Standaard zit het DNS-protocol zo in elkaar, dat verzoeken voor namen die lokaal niet achterhaald kunnen worden, doorgestuurd worden naar de DNS rootservers. In veel gevallen is dit helemaal niet handig. Denk bijvoorbeeld aan het geval waarin 90% van alle verzoeken die in uw netwerk gedaan worden, gericht wordt aan hosts in het NL domein. In zo'n geval is het nuttig gebruik te maken van een forwarder. Een DNS forwarder is een DNS-server waarnaar verzoeken gestuurd worden die lokaal niet afgehandeld kunnen worden. Er zijn twee manieren om DNS-forwarders te definiëren. Als eerste kunt u een forwarder definiëren die gewoon alle verzoeken afhandelt die lokaal niet gehonoreerd kunnen worden. Daarnaast is het mogelijk een forwarder te definiëren die verzoeken voor één bepaald domein voor zijn rekening neemt. Verzoeken die niet gericht zijn aan hosts in dat bepaalde domein, worden gewoon doorgestuurd naar naamservers van het rootdomein.

Om een algemene forwarder te definiëren, zorgt u ervoor dat de volgende code wordt opgenomen in named.conf. Let er wel even op dat u het juiste adres gebruikt van de server die u als forwarder gebruikt:

```
options {
        directory "/var/named";
        forwarders { 10.0.0.1; };
};
```

Als u niet gewoon alles door wilt sturen naar die ene naamserver die hier genoemd is, maar alleen specifieke requests, kunt u met het volgende stukje code een forwarder definiëren die zijn werk doet voor één bepaalde zone. In dit voorbeeld doen we dat voor het NL-domein.

```
zone "nl"{
        type forward;
        forwarders ( 10.0.0.2; };
};
```

3.3 Beveiliging van DNS

Er zijn verschillende aspecten die te maken hebben met de beveiliging van een naamserver. Het eerste aspect is de beveiliging van de naamserver zelf: hiermee zorgt u ervoor dat de software die op uw server draait op een veilige wijze is ingericht zodat een hacker die door de DNS-server heen breekt niet ineens root permissies heeft op het bestandssysteem van uw server. Het tweede aspect van beveiliging van een naamserver bestaat eruit dat uw naamserver niet zomaar met iedereen die dat wil een zone transfer uit gaat voeren, maar alleen met hosts aan wie dat ook wordt toevertrouwd. In deze paragraaf maakt u kennis met een aantal van de meest belangrijke manieren om uw DNS-server te beveiligen.

3.3.1 Named starten in een chroot-gevangenis.

Wanneer u uw named-server laat draaien in een chroot jail, betekent dat dat u voor de server een speciale directory maakt waarin alle configuratie staat die de naamserver nodig heeft.

Buiten deze specifieke configuratie bevinden zich in deze directory geen andere bestanden. Vervolgens zorgt u ervoor dat de naamserver gestart wordt met de optie –t gevolgd door de naam van de chroot directory en u bent klaar: de named server chroot jail is een feit! Het enige punt bestaat eruit dat de directory die u als chroot jail gebruikt voorzien moet zijn van de juiste bestanden. Als u bind 9 gebruikt, moeten in deze directory de volgende componenten voorkomen:

* Een subdirectory etc met daarin de bestanden named.conf en locatime.
* Een directory var/run waarin het PID-bestand van de naamserver voorkomt
* Een subdirectory met de naam dev waarin de devices log, random en zero voorkomen.
* Daarnaast moeten onder de chroot subdirectory var ook alle bestanden voorkomen waarin de betreffende zones gedefinieerd worden: een verwijzing naar directory /var/named in named.conf, wordt nu immers geínterpreteerd als een verwijzing naar een directory met deze naam die voorkomt onder de chroot directory.

U kunt er met de volgende opdrachten voor zorgen dat deze componenten worden aangemaakt:

 # mkdir /var/named/namedroot
 # cd /var/named/namedroot
 # mkdir –p dev var/run var/named
 # cp /etc/localtime etc
 # mknod dev/random c 1 8
 # mknod dev/zero c 1 5
 # cp /etc/named.conf etc/named.conf
 # cp /var/named/* var/named

Nadat u op deze wijze de chroot-directory hebt aangemaakt, kunt u de naamserver starten met de optie –t: aansluitend op het voorgaande voorbeeld, zou u in dit geval de opdracht **named –t /var/named/namedroot** gebruiken. Als dit probleemloos gaat, zorgt u er vervolgens voor dat de startup scripts aangepast worden zodat de naamserver voortaan elke keer gestart wordt in deze speciaal beveiligde directory. Zowel op SUSE als op Fedora Linux bestaat in de directory /etc/sysconfig een speciaal configuratiebestand voor named. Hierin kunt u opties specificeren die door named moeten worden uitgevoerd tijdens opstarten. Fedora doet dit door middel van de variable ROOTDIR=/var/name/chroot. Deze staat standaard ingesteld zodat uw DNS server standaard al chroot gestart wordt.

***chroot.tif Op Fedora wordt named automatisch al in een chroot omgeving gestart.

3.3.2 Bepalen met welke permissies named gestart wordt.

Leuk natuurlijk als u named start in een chroot omgeving, maar wanneer de DNS naamserver vervolgens nog wel als root gestart wordt, hebt u hier niets aan. Dit is omdat de gebruiker root zonder probleem uit kan breken uit een chroot omgeving: hij hoeft alleen maar de odpracht **exit** te gebruiken. Om te voorkomen dat een inbreker zo toch nog kwaad kan doen door middel van slecht geconfigureerde software, zorgt u ervoor dat named gebruikt wordt met andere permissies als die van de gebruiker root. Het probleem hierbij is echter dat named altijd wel als gebruiker root gestart moet worden. Dit komt omdat de server gebruikmaakt van poort 53 en dit is een well-known poort. Zoals u weet, mag alleen root gebruikmaken van deze well known poorten. Om ervoor te zorgen dat named nadat het gestart is gebruikmaakt van permissies van een ander gebruikersaccount, gebruikt u bij het opstarten van named de optie –u gevolgd door de naam van het gebruikersaccount dat u voor dit doel wilt gebruiken. U zult merken dat voor dit doel meestal al een gebruikersaccount bestaat dat de naam named heeft. Op Fedora Linux ziet de definitie van deze gebruiker in /etc/passwd er als volgt uit:

named:x:25:25:Named:/var/named:/sbin/nologin.

Als op uw distributie nog geen speciale gebruiker bestaat voor gebruik van de DNS-server, dan maakt u deze handmatig aan. Let er in dat geval wel op dat de gebruiker de juiste permissies heeft op alle relevante bestanden en directories. Hierbij gaat het om het volgende:

* Alle rechten in de working directory van named (meestal /var/named)
* Read en write op alle bestanden waarin de naamserver gecofnigureerd wordt
* Write op de directory waarin het PID-bestand van de naamserver staat opgeslagen.

Tip! Heel veel servers maken gebruik van een PID-bestand. Hierin wordt opgeslagen van welk PID de betreffende service gebruikmaakt. Als een server netjes wordt afgesloten, worden deze PID-bestanden automatisch opgeruimd om vervolgens weer automatisch aangemaakt te worden wanneer de server weer gestart wordt. Als een server niet netjes afgesloten wordt maar crasht, worden de PID-bestanden niet netjes opgeruimd. Dit kan problemen opleveren wanneer de service die bij zo'n PID-bestand hoort weer gestart wordt: hij kan namelijk weigeren te starten omdat er al een PID-bestand voor die service bestaat. Dit probleem lost u op door de betreffende PID-bestanden handmatig te verwijderen.

3.3.3 Beperken van zone transfers

In een standaardsituatie kan iedereen die dat wil een zone transfer starten. Dit kan bijvoorbeeld al met behulp van de opdracht **dig @naamserver domein.nl axfr**. Als u dat leuk vindt, kunt u dus zelf een zonetransfer initiëren voor het domein sandervanvugt.nl met de opdracht **dig @212.100.231.90 sandervanvugt.nl axfr**. Alle informatie over het betreffende domein wordt vervolgens op uw scherm getoond.

***axfr.tif U kunt gewoon met de opdracht dig een zonetransfer starten.

Wanneer u uw naamserver extra wilt beveiligen, kunt u overwegen om deze functie uit te schakelen. Buiten de slave naamserver is het immers voor geen enkel ander nodig een zone transfer te starten naar uw domein. Om dit voor elkaar te krijgen, moet named.conf zowel op de master server als op de slave servers aangepast worden. U doet dat door in de definitie van de zone de aanduiding allow-transfer op te nemen:

```
allow-transfer {
        lijst-van-slave-servers;
};
```

Aangezien een slave server maar een slave server is en hier eigenlijk helemaal geen zone transfer nodig is, gebruikt u hetzelfde statement op de slave server. Het enige verschil is dat u nu aangeeft dat niemand een zone transfer mag starten: dat is voor slave servers immers helemaal niet nodig. Dit krijgt u voor elkaar met behulp van de volgende regels die voorkomen onder de definitie van de zone:

```
allow-transfer {
        none;
};
```

Naast het beperken van de zone transfers dat u op deze wijze kunt doen, is het ook mogelijk om te beperken wie een query uit mag voeren. Dit doet u door gebruik te maken van het

statement allow-query in named.conf. In het volgende voorbeeld ziet u hoe eerst een ACL gedefinieerd wordt waarmee een naam gegeven wordt aan een groep hosts en vervolgens de naam van deze ACL gebruikt wordt in het allow-queries statement:

```
acls “mijnnet”{
        192.168.0.0/24;
};

// …..

zone “sandervanvugt.nl” IN {
        // …

        allow-queries {
                mijnnet;
        };
};
```

3.3.4 Communicatie beveiligen met behulp van keys.

Leuk natuurlijk dat u met allow-transfer en allow-queries kunt bepalen wie er allemaal in staat zijn om gegevens uit te wisselen met uw DNS-server. Er is echter een zwak punt in deze hele configuratie: hoe weet u zeker dat een host die u denkt te vertrouwen inderdaad is wie hij zegt te zijn? Dit probleem lost u op door gebruik te maken van cryptografie waarmee de communicatie nog eens extra beveiligd wordt. In BIND 8 maakt u hiervoor gebruik van de opdracht **dnskeygen**, in een BIND 9 omgeving, maakt u gebruik van de opdracht **dnssec-keygen.** Om ervoor te zorgen dat de communicatie tussen een master en slave extra beveiligd wordt, kunt u het bestand named.conf uitbreiden met een verwijzing naar een secret key. Om zo’n secret key te generen, gebruikt u bijvoorbeeld het onderstaande commando:

```
dnssec-keygen –a HMAC-MD5 –b 512 –n HOST dnskey.sandervanvugt.nl
```

In dit commando specificeert de optie –a HMAC-MD5 het algoritme dat u wilt gebruiken. De optie –b 512 geeft aan hoeveel bits voor de key gebruikt moeten worden: hierbij geldt hoe meer bits hoe beter, -n HOST geeft aan dat het een host-key is die gebruikt wordt voor veilige communicatie tussen twee computers en tot slot wordt een naam gegeven aan de key. Om nu deze key ook te gebruiken, neemt u in named.conf op alle servers die veilig met elkaar moeten communiceren de volgende regels op:

```
key dnskey.sandervanvugt.nl {
        algorithm hmac-md5;
        secret “öWv61InvUG2X…….c9zvxg”;
};
```

Let er hierbij vooral op dat u de juiste key’s gebruikt. U kunt deze key letterlijk overnemen uit het bestand dat door de opdracht **dnssec-keygen** is aangemaakt. Dit bestand heeft de als naam de naam die bij het commando gespecificeerd is als argument (in dit geval dus dnskey.sandervanvugt.nl) gevolgd door een + teken en wat cijfers en de extensie .key.

3.4 DNS utilities

Er is een aantal commando's dat gebruikt kan worden om te kijken of uw DNS-server het naar behoren doet. Als eerste is dat de opdracht dig: hiermee kunt u uitgebreid bekijken of een DNS-server naar behoren is opgebouwd. Vervolgens is er het eenvoudige commando host waarmee gegevens uit de database opgevraagd kunnen worden. Als laatste noemen we de opdracht nslookup. In het verleden werd deze opdracht gebruikt voor functionaliteit die nu door dig geboden worden. Gebruik van deze opdracht is dus momenteel niet langer nodig.

3.4.1 Dig

De opdracht dig kan worden gebruikt om eens uitgebreid met uw DNS-server te communicren. U kunt er de status van een entry mee uitvragen, bijvoorbeeld door de opdracht **dig hostnaam.domein.nl** te geven. U krijgt vervolgens niet alleen te zien op welk IP-adres deze host te bereiken is, maar ook wat de DNS-namen zijn van de naamservers die verantwoordelijk zijn voor dat domein en wat de IP-adressen van die naamservers zijn.

***dig De opdracht dig is in staat uitgebreid informatie te geven over een ondervraagde host.

Zoals veel opdrachten die heel veel kunnen, heeft ook dig soms de neiging onduidelijk te worden. Als u bijvoorbeeld informatie opvraagt voor een host die niet bestaat in een domein dat wel bestaat, wordt dat niet met zoveel woorden gezegd. In plaats daarvan ziet u een verwijzing naar de SOA van het betreffende domein, met daarbij ook gelijk het adres waarop de beheerder van de SOA te vinden is. Dit lijkt misschien onhandig, maar hier zit een filosofie achter: u kunt op basis van de gegevens die in dat geval verstrekt worden namelijk contact opnemen met de beheerder van de SOA van het betreffende domein om te vragen wat er aan de hand is.

***digfout In plaats van een foutmelding, vertelt dig u waar u bepaalde informatie wél op kunt vragen.

Wat ook heel aardig is aan dig, is dat u er een specifieke DNS-server mee kunt ondervragen. Als u niets doet, gaat dig naar de standaard DNS-server zoals die via uw DNS-resolver teruggevonden wordt. Wilt u dat niet, maar wilt u informatie opvragen bij een specifieke server? Geef dan als eerste argument welke server u wilt ondervragen en vervolgens de host waarover u informatie wilt hebben. Een voorbeeld hiervan is de opdracht **dig @10.0.0.1 x-tina.sandervanvugt.nl**. Dig heeft nog een aantal interessante opties, raadpleeg de man pagina voor meer informatie.

3.4.2 Host

Het belangrijkste voordeel van het commando **host**, is vooral dat het eenvoudig is. U gebruikt deze opdracht voor twee verschillende zaken: geeft u de naam van een host als argument, dan wordt het bijbehorende IP-adres getoond. Vooral ook heel aardig is de mogelijkheid een IP-adres als argument te geven. Door middel van het in-addr.arpa mechanisme wordt dan namelijk achterhaald welke naam bij dat adres hoort. Daarnaast zijn er nog wat geavanceerde opties; gebruik bijvoorbeeld –C om te achterhalen wat de SOA is voor een bepaalde zone waarvan de naam als argument gegeven wordt. Vindt u de opdracht host niet uitgebreid genoeg in zijn resultaat? Gebruik dan de optie –v om ervoor te zorgen dat meer informatie getoond wordt.

***host De opdracht host is vooral handig om te achterhalen welke naam bij een bepaald IP-adres hoort.

3.4.3 Nslookup

Het klassieke hulpmiddel om een DNS-server te ondervragen, is **nslookup**. Met behulp van deze opdracht kan vrijwel alle informatie die door een bepaalde naamserver aangeboden wordt achterhaald worden. Een kenmerkende eigenschap van nslookup, is dat u ermee kunt werken vanuit een eigen specifieke prompt. Op deze prompt maakt u gebruik van een van de vele beschikbare opdrachten van nslookup. Gebruik bijvoorbeeld de opdracht **server** om een bepaalde naamserver te ondervragen en vervolgens **host** gevolgd door de naam van een specifieke computer om meer informatie over deze host te krijgen. De opdracht nslookup is echter een oude opdracht, tegenwoordig wordt veel van de functinoaliteit van nslookup uitgevoerd met het commando **dig**.

Oefening

Om deze oefening uit te voeren, hebt u twee servers nodig. De ene server wordt geconfigureerd als master naamserver, de ander als slave.
Configureer een DNS master naamserver. Zorg ervoor dat alle noodzakelijke resource records aangemaakt worden die nodig zijn in uw netwerk. U maakt in elk geval een resource record aan voor de master en slave server, maar als er meerdere nodes in het netwerk aanwezig zijn, zorgt u dat ook deze bereikbaar worden op hun DNS-naam. Maak een alias aan die ervoor zorgt dat uw master-server ook bereikbaar is op de naam “meester” en de slave-server op de naam “slaaf”. Zorg ook dat de verwijzing naar de mailserver die gebruikt moet worden voor het domein goed geregeld is. Vergeet bij de configuratie de reversed-DNS niet zodat niet alleen namen in adressen omgezet kunnen worden, maar ook adressen in namen kunnen worden vertaald. Ook moet uw server in staat zijn verzoeken door te sturen naar servers van het DNS-root domein op internet. Wanneer de master-server naar behoren geconfigureerd is, zorgt u dat de slave-server zo wordt ingericht dat deze een volledige kopie van de database bevat. Start vervolgens ook direct een zone-transfer van de master naar de slave. Regel dat alleen de slave server in staat is een zone tranfer uit te voeren op de master en verder niemand. Is er nu nog een manier om met de opdracht dig een zone transfer uit te voeren op een van uw DNS-servers? Verklaar dit.

Samenvatting

In dit hoofdstuk hebt u kennisgemaakt met DNS. Om te beginnen is de werking van het protocol uiteen gezet. U hebt geleerd hoe er wereldwijd een DNS-hiërarchie wordt toegepast die ervoor zorgt dat alle computers op internet op basis van hun naam teruggevonden kunnen worden. Vervolgens hebt u geleerd hoe u zelf een DNS-server kunt bouwen. Hierbij is eerst aandacht besteed aan de cache-only server, vervolgens aan de configuratie van een master server. Daarna hebt u geleerd hoe de database op de master-server gerepliceerd kan worden naar een salve-server en hoe de authoriteit over zones binnen de hiërarchie gedelegeerd kan worden. Daarna hebt u gelezen over een aantal mogelijke problemen in de beveiliging van uw DNS-server en hoe u deze problemen op kunt lossen. Tot slot hebt u kennisgemaakt met een drietal programma’s die ingezet kunnen worden om informatie op te vragen bij een DNS-naamserver.

Oefenvragen

1. Is het mogelijk om één DNS-server meerdere zones te laten bedienen? Verklaar uw antwoord.
2. Welke opdracht gebruikt u om op BIND 9 een zone transfer te initiëren?
3. Welke opdracht gebruikt u om sleutels te genereren waarmee communicatie tussen twee DNS-servers beveiligd kan worden?
4. Hoe heet de organisatie die wereldwijd verantwoordelijk is voor DNS?

5. Hoe heet het bestand waarin u op een Linux-systeem aangeeft welke naamserver u wilt gebruiken?
6. Wat is het doel van het bestand dat meestal voorkomt met de naam root.hints en hoe zorgt u dat dit bestand inhoud krijgt?
7. Welke regel neemt u op in named.conf om een subzone te definiëren?
8. Hoe regelt u dat één naam verbonden wordt aan twee adressen?
9. Wat is het verschil tussen SOA en NS?
10. Welke opdracht gebruikt u om de werking van een DNS-server te controleren?

Hoofdstuk 4 Web services

Wanneer u contact maakt met een willekeurige webserver ergens op het Internet, is er een grote kans dat dit een Apache-webserver is. Apache is namelijk de meest gebruikte webserver op deze wereld. Ondanks dat Apache inmiddels voor elk mogelijk platform beschikbaar is, wordt deze webserver toch nog het meest ingezet op UNIX en Linux servers. De reden hiervoor is eenvoudig: Apache levert op deze platforms namelijk de beste prestaties. In dit hoofdstuk leert u hoe u deze veelzijdige webserver in uw netwerk kunt inzetten

Leerdoelen:
In dit hoofdstuk leert u het volgende:

* Functionaliteit van een webserver
* Installatie en setup van de Apache webserver
* Structuur en configuratiebestanden van de Apache webserver
* Inrichting van httpd.conf
* Configuratie van virtuele hosts
* Beperken van toegang tot de webserver
* Apache beveiligen met OpenSSL
* Configuratie van een Squid Proxy server

4.1 Inleiding

Wanneer u zich gaat verdiepen in webservers, zijn er twee manieren waarop u dit kunt doen. De eerste manier is de aanpak van de web-designer. Dit is een persoon dat geïnteresseerd is in hoe een web pagina zo efficiënt mogelijk kan worden opgezet. Hij houdt zich daarvoor bezig met HTML, Perl, CGI, PHP en meer van dit soort fraais. Dit is niet de aanpak die wij in dit hoofdstuk willen volgen. De aanpak die wij wel willen volgen, is de aanpak van de systeembeheerder die als taak heeft het platform te leveren waarop de web designer zijn werk kan doen. In dit hoofdstuk wordt dus besproken hoe u ervoor zorgt dat een Apache-webserver zijn diensten aan kan bieden aan gebruikers van deze server. Aangezien Apache zeer uitgebreide configuratiemogelijkheden biedt, kunnen wij niet verder gaan dan een algemene inleiding. Zo wordt er bijvoorbeeld niet ingegaan in de koppeling naar verschillende scripttalen als Perl en PHP die mogelijk is en blijven ook veel andere zaken liggen. Voor meer details hierover raden wij u aan een boek te kopen dat in dit onderwerp gespecialiseerd is.

4.2 Functionaliteit van een webserver

Hoewel tegenwoordig iedereen wel een vaag idee heeft wat een webserver nu eigenlijk is, besteden we voordat we beginnen toch nog even aandacht aan wat nu eigenlijk het fenomeen webserver is. Een webserver is een applicatie die gegevens beschikbaar stelt die door een browser benaderd kunnen worden. Deze gegevens worden aangeleverd in een standaardformaat. Het formaat dat het uitgangspunt vormt is HyperText Markup Language (HTML), maar door identificatie van andere typen gegevensformaten kunnen met behulp van plug-ins ook heel veel andere gegevens ingelezen worden. Denk bijvoorbeeld aan afbeeldingen, filmpjes, flash-animaties of muziek. De enige voorwaarde voor het gebruik van deze formaten, is dat in de browser ondersteuning aanwezig is voor het formaattype door middel van een plug-in en dat de server in staat is aan te geven welk type formaat er nu eigenlijk gebruikt wordt. Het laatste onderdeel dat nodig is voor een webserver, is een protocol dat definieert hoe de browser en de webserver met elkaar communiceren. Hiervoor wordt gebruikgemaakt van het HyperText Transfer Protocol (http).
Een webserver is niet alleen maar een suffe server die in staat is gegevens te verplaatsen naar een client-programma, er wordt ook wat extra intelligentie geleverd. Denk daarbij

bijvoorbeeld aan het beperken van toegang tot bepaalde gebruikers, of het versleutelen van gegevens die tussen de webserver en de client verstuurd worden.

4.3 Installatie en setup van de Apache webserver

Vaak is de Apache webserver standaard al aanwezig nadat u Linux geïnstalleerd hebt. Als dit niet het geval is, is het een koud kunstje om hem alsnog te installeren. De installatie van een webserver behelst echter meer dan alleen maar ervoor te zorgen dat een toepassing gestart wordt: u moet er ook voor zorgen dat de webserver automatisch gestart wordt wanneer u de computer moet herstarten en u moet duidelijk maken waar de webserver zijn documenten terug kan vinden.

4.3.1 Controleren of er al een Apache server draait.

De Apache webserver wordt op de meeste Linux systemen standaard geïnstalleerd. Het kan alleen zijn dat de server niet automatisch gestart wordt; dit kunt u echter snel genoeg achterhalen door het commando **ps axl | grep httpd** te geven. Wanneer Apache gestart is, luister namelijk het proces httpd naar binnenkomende verzoeken. Toont de opdracht **ps axl | grep httpd** geen resultaat? Dan kunt u kijken in de directory /etc/init.d of er een opstartscript is waarmee u de webserver handmatig kunt activeren. Vaak zult u hier een bestand vinden met de naam apache of apache2. Het bestand dat u nodig hebt kan overigens ook gewoon httpd heten. Met behulp van dit bestand kunt u de Apache webserver starten. Activeer om te beginnen de directory /etc/init.d en geef vervolgens de opdracht **./apache start**. Dit zorgt ervoor dat het webserver proces het configuratiebestand httpd.conf leest en aan de hand daarvan zijn werk gaat doen.

Tip! Op SUSE Linux kunt u versie 2 van de Apache webserver starten met de opdracht **rcapache2 start**. Om de server weer te deactiveren, geeft u de opdracht **rcapache2 stop**. Wilt u ervoor zorgen dat de apache webserver altijd automatisch geactiveerd wordt wanneer u uw computer aanzet, geef dan de opdracht **insserv apache2** om hem definitief toe te voegen aan de opstartroutines van uw server.

***apastart Wanneer httpd niet actief is, kunt u de Apache webserver starten vanuit de directory /etc/init.d

Wanneer Apache gestart is, kunt u direct beginnen met gebruik ervan. Open uw browser en geef de URL http://localhost. U komt nu op een standaard welkomstpagina van de Apache webserver waaruit blijkt dat de server klaar is voor gebruik.

***localhost Nadat de webserver gestart is, kunt u hem direct met uw browser benaderen op http://localhost.

4.3.2 Apache installeren

De wijze waarop u te werk gaat om Apache te installeren, hangt af van de distributie die u gebruikt. Main-stream distributies zoals Fedora en SUSE leveren kant en klare RPM's waarmee u de Apache server in een handomdraai geïnstalleerd hebt. Als u gebruikmaakt van een distributie die Apache RPM's levert, raden wij u aan om van deze methode gebruik te maken. Hiermee voorkomt u immers dat uw installatiepoging voortijdig strandt omdat er een aantal dependencies niet aanwezig is. Alleen wanneer u geen kant en klare installatiebestanden voorhanden hebt, of natuurlijk wanneer het voor u noodzakelijk is de allerlaatste versie van de Apache webserver te gebruiken, kunt u een tarball downloaden van www.apache.org en de bestanden hieruit installeren.

Houdt er bij het werken met Apache rekening mee dat er twee versies zijn die beiden nog vrij algemeen gebruikt worden. Om te beginnen is er versie 1 waarvan subversie 1.3 het laatste onderdeel is. Deze versie is lange tijd op grote schaal ingezet. Al enige tijd echter is ook versie 2 van de Apache webserver beschikbaar. Veel distributies leveren beiden naast elkaar. De reden hiervoor is dat veel organisaties nog steeds behoefte hebben aan Apache versie 1.3 vanwege achterwaarde compatibilitiet. Als toepassingen ontwikkeld zijn samen te werken met Apache 1.3, is het lastig om zo maar even over te gaan naar versie 2. In dit boek echter besteden we uitsluitend aandacht aan Apache versie 2.

Nadat u de Apache webserver geïnstalleerd hebt, kunt u hem verder configureren. Om te beginnen doet u dat door aan te geven welke directory als documentroot gebruikt moet worden. De documentroot is de directory waarin u alle HTML-documenten plaatst die door de Apache server aangeboden moeten worden. De standaarddirectory die hiervoor gebruikt wordt, verschilt per distributie. Op SUSE Linux wordt de directory /srv/www/htdocs voor dit doel gebruikt. Na installatie van de Apache webserver, vindt u in deze directory alleen nog maar een paar voorbeeldbestanden. Een van deze bestanden is het bestand index.html, dit is het bestand dat standaard aangeboden wordt door de Apache webserver wanneer hij door een gebruiker benaderd wordt. U kunt overigens in het Apache configuratiebestand aangeven dat voor dit doel een ander bestand gebruikt moet worden als u dat wenst.
Als beheerder van de Apache webserver moet u ervoor zorgen dat hier ook andere bestanden geplaatst kunnen worden. Als u ooit gebruik zou willen maken van een andere dan de standaard documentroot, zorg er dan in elk geval voor dat de gebruiker wwwrun rechten heeft in deze directory; dit is namelijk de gebruikers ID waarvan de Apache webserver standaard gebruikmaakt. Ook kunt u in deze directory subdirectories aanmaken. Deze subdirectories kunnen vervolgens door gebruikers in de URL gespecificeerd worden. Om bijvoorbeeld het bestand index.html te benaderen dat op uw webserver is opgeslagen in /srv/www/htdocs/Sales, geeft de gebruiker in zijn browser de URL http://uwserver/sales.

4.4 Structuur en configuratiebestanden van de Apache webserver

Versie 1 van de Apache webserver maakte gebruik van één configuratiebestand van waaruit alles geregeld werd: het bestand httpd.conf. Daarnaast kon ook gebruikgemaakt worden van de configuratiebestanden access.conf en smr.conf voor wat additionele configuratie, maar veel beheerders gaven er de voorkeur aan om alles vanuit httpd.conf te regelen. Voor Apache 2 is de situatie iets anders. Het bestand httpd.conf speelt nog steeds een hoofdrol, maar vanuit dit bestand wordt door middel van het **include** statement een groot aantal secundaire configuratiebestanden aangeroepen. Op SUSE Linux vindt u alle Apache configuratiebestanden in de subdirectory /etc/apache2, op Fedora komen ze voor onder /etc/httpd/conf. Op SUSE Linux komt u in deze directory onder andere de volgende bestanden tegen. Houdt er rekening mee dat dit een kenmerkende SUSE-configuratie is, op andere Linux distributies ziet de structuur van configuratiebestanden er vaak heel anders uit:

* https.conf. Dit is het meest belangrijke configuratiebestand. Vanuit dit configuratiebestand worden alle andere configuratiebestanden aangeroepen.
* default-server.conf. Hier vindt u de standaard setup van de webserver. De inhoud van dit bestand kan echter overschreven worden door andere configuratiebestanden.
* vhost.d/ In deze directory wordt de configuratie voor virtuele hosts opgeslagen. Later in dit hoofdstuk leert u meer over het werken met virtuele hosts.
* uid.conf In dit bestand wordt bepaald van welke UID en GID Apache gebruikmaakt. Na een standaardinstallatie zijn dit de gebruiker wwwrun en de groep www.

* listen.conf. Hierin wordt aangegeven op welke poort de Apache websesrver luistert. Standaard luistert de Apache webserver op alle aanwezige netwerkkaarten naar poort 80.
* server-tuning.conf. In dit bestand wordt de werking van de Apache webserver geoptimaliseerd. In de meeste gevallen voldoen de standaardwaarden uitstekend, alleen voor zwaar belaste webservers kan het de moeite zijn parameters in dit bestand aan te passen.
* error.conf. Hier wordt bepaald wat de Apache webserver moet doen wanneer zich een fout voordoet.
* ssl-global.conf. Dit configuratiebestand bevat de configuratie die gebruikt wordt wanneer u gebruik wilt maken van SSL-encryptie.

In welke bestanden de Apache configuratie standaard geregeld wordt, staat overigens niet vast. De beheerder van de server kan dit allemaal zelf naar wens inrichten door in /etc/apache2/httpd.conf met Include regels te werken. De regels waarin dit gebeurd, worden Directives genoemd. Let er overigens op dat deze directives hoofdlettergevoelig zijn: als u Include bedoelt, moet u dus niet include schrijven.
Door een Include directive op te nemen, wordt vanuit het hoofdconfiguratiebestand een extra bestand met configuratie aangeroepen. Zo wordt bijvoorbeeld een bestand aangeroepen waarin wordt aangegeven welke modules er bij het laden van de Apache server mee geladen moeten worden:

```
# generated from APACHE_MODULES in /etc/sysconfig/apache2
Include /etc/apache2/sysconfig.d/loadmodule.conf
```

Bij het werken met directives, komt het vaak voor dat deze gegroepeerd worden. Door directives te groeperen, kunt u ze toepassen voor een specifieke set parameters. Zo ziet u in het onderstaande een voorbeeld waarin een directive gedefinieerd wordt dat alleen instellingen doet voor de directory /srv/www/htdocs:

```
<Directory “/srv/www/htdocs”>
        Options None
        AllowOverride None
        Order allow,deny
        Allow from all
</Directory>
```

Let even op de wijze waarop het groeperen plaatsvindt: de groepering begint met het keyword Directory en wordt afgesloten met hetzelfde keyword, maar dan wordt dit keyword voorafgegaan door een slash.
Een ander algemeen element dat in de Apache configuratiebestanden opgenomen kan worden, is het commentaarteken. Het zal u niet verbazen dat ook de Apache server hiervoor de # gebruikt.
Waarschuwing! De Apache webserver is niet zo slim dat hij er automatisch achter komt wanneer u iets gewijzigd hebt. U moet dit zelf duidelijk maken door de server opnieuw te starten. Geef hiervoor de opdracht /etc/init.d/apache2 restart.

Tip!Wilt u controleren of u het Apache configuratiebestand op de juiste wijze hebt aangemaakt? Gebruik dan de opdracht **apache2 ctltest**. Deze opdracht controleert uw Apache configuratie op eventueel voorkomende fouten in de syntaxis. Als er een probleem is, wordt daar duidelijk melding van gegeven en weet u ook precies waar u het probleem op moet lossen. Is alles in orde, dan wordt de boodschap Syntax OK gegeven.

4.5 Inrichting van httpd.conf

Het bestand httpd.conf is de aangewezen locatie waar de configuratie van een Apache webserver gedaan wordt. Aangezien de Apache webserver ook heel flexibel is, is dit echter geen vaststaand gegeven: zo maakt SUSE Linux bijvoorbeeld gebruik van het hoofdbenstand default-server.conf. Welk bestand er precies gebruikt wordt, maakt niet zo heel veel uit, als u uiteindelijk maar weet hoe deze bestanden aangepast moeten worden. Hiervoor maakt u gebruik van een aantal standaard directives. In het onderstaande vindt u een overzicht van veel gebruikte directives en andere constructies die u in deze bestanden tegen kunt komen:

Directive	**Toepassing**
DocumentRoot	Geeft aan in welke directory alle documenten van de webserver geplaatst worden.
Directory “naam” /Directory	Wordt gebruikt om een uitzondering te definiëren voor een bepaalde directory.
Include	Roept een bestand aan waarin additionele configuratie gedaan wordt.
Options	Wordt gebruikt om opties toe te voegen aan een uitzondering die gedefinieerd is, zoals een speciale directory.
AllowOverride	Bepaalt of het is toegestaan dat instellingen in een directive overschreven worden door een lokaal configuratiebestand met de naam .htaccess.
Alias “naamvandealias” “echtenaam”	Maakt een alias naar een directory met een bepaalde naam
ScriptAlias	Maakt een alias naar een directory waarin scripts voorkomen die ervoor zorgen dat webpagina’s dynamisch gegenereerd worden.

Dit is slechts een beperkt overzicht van beschikbare directives. Een volledig overzicht kunt u vinden in de documentatie van de Apache server op http://www.apache.rog/docs-2.0/mod/directives.html.

4.6 Configuratie van virtuele hosts

Nadat een standaard installatie van een Apache webserver is utigevoerd, kan deze bereikt worden op het adres van de server waarop hij geïnstalleerd is door dit adres aan te roepen vanuit een browser die begint met de aanduiding http:// gevolgd door het serveradres. Nadat de gebruiker op deze wijze zijn server benadert, worden de bestanden getoond die in de directory staan die als DocumentRoot gedefinieerd is in de Apache configuratiebestanden. Dit systeem werkt prima, zolang er per server maar één Apache webserver geïnstalleerd is. Wanneer een bedrijf de behoefte heeft meerdere Apache webservers in te richten, worden de nadelen van dit systeem zichtbaar: er zou dan immers voor elke afzonderlijke webserver een afzonderlijke computer nodig zijn. Dit probleem wordt opgelost door gebruik te maken van virtuele hosts.
Als gebruikgemaakt wordt van virtuele hosts, wordt een entry opgenomen in de DNS-database voor elke virtuele apache webserver. Op basis van de DNS-naam die een gebruiker invoert, wordt hij vervolgens doorgestuurd naar de juiste virtuele webserver. Het contact met een virtuele host wordt op de volgende wijze tot stand gebracht:

1. Vanuit de webbrowser wordt een URL opgegeven. De DNS-naam die in deze URL gebruikt wordt, wordt vervolgens omgezet in een IP-adres door contact op te nemen met de DNS-server.
2. Op basis van dit IP-adres, kan de browser contact opnemen met de juiste fysieke machine.
3. Er wordt nu een http-pakketje naar de server verstuurd waarop de virtuele host actief is. In dit http-pakketje wordt verwezen naar de naam van de virtuele host.
4. Op basis van deze hostnaam die gebruikt wordt, is de Apache webserver in staat het verzoek door te sturen naar de juiste virtuele host.

De wijze waarop virtuele hosts gedefinieerd worden, verschilt per gebruikte Linux distributie. In het onderstaande ziet u hoe de configuratie van virtuele hosts in het algemene configuratiebestand httpd.conf geregeld kan worden. Dit is bijvoorbeeld de wijze waarop het op Fedora Linux gebeurt:

```
BindAddress   *
Listen  192.168.1.1:8080
<VirtualHost “www.mijnwebserver.nl”>
        Documentroot /var/apache/mijnwebserver
        ServerName www.mijnwebserver.nl
        ServerAdmin Webmaster@mijnwebserver.nl
        ErrorLog logs/mijnwebserver.nl-error-log
        CustomLog logs/mijnwebserver.nl-custom-log
</VirtualHost>
<VirtualHost “www.jouwwebserver.nl”>
        Documentroot /var/apache/jouwwebserver
        Servername www.jouwwebserver.nl
        ServerAdmin Webmaster@jouwwebserver.nl
        ErrorLog logs/jouwwebserver.nl-error-log
        CustomLog logs/jouwwebserver.nl-custom-log

</VirtualHost>
```

In de voorgaande regels wordt eerst gedefinieerd dat de webserver benaderd kan worden via elke netwerkkaart in het systeem. Dit hoeft overigens niet apart vermeld te worden, de standaardwaarde zorgt er namelijk ook al voor dat dit mogelijk is maar het is leuk dat u zo deze parameter ook eens tegenkomt. Vervolgens wordt geregeld dat de webserver naast de standaardpoort 80 ook luistert naar verzoeken die binnenkomen op poort 8080, maar dat geldt alleen voor de host die draait op IP-adres 192.168.1.1.

Daarna worden twee virtuele hosts gedefinieerd. De regels waarin dit gebeurt zullen duidelijk zijn, per virtuele host is er een extra blokje met configuratieregels. U ziet dat specifieke parameters zoals het mailadres van de beheerder van de server en de locatie van logbestanden ook per virtuele host gedefinieerd worden. Waar u vooral goed op moet letten, is dat u in DNS ook een voorziening geregeld hebt voor deze virtuele hostnamen. Om te kunnen werken met virtuele hosts, moet de naam van die host door middel van een CNAME record in DNS gedefinieerd zijn.

Als u gebruikmaakt van SUSE Linux, moet u een iets andere werkwijze betrachten om virtuele hosts te kunnen definiëren. U maakt in dit geval namelijk voor elke virtuele host die u

wilt definiëren een apart configuratiebestand aan in de directory /etc/apache2/vhosts.d. Hoe dit configuratiebestand heet maakt niet uit, als de naam ervan maar eindigt op .conf. Om het werken met een dergelijk configuratiebestand iets te vereenvoudigen, vindt u in /etc/apache2/vhosts.d een voorbeeldbestand met de naam vhost.template. In het configuratiebestand voor een virtuele host kunnen de volgende directives worden opgenomen:

Directive	**Betekenis**
ServerAdmin	Het mailadres waarop de beheerder van de virtuele host bereikt kan worden.
ServerName	De naam van de virtuele host. Deze naam moet op dezelfde wijze in DNS geconfigureerd zijn.
DocumentRoot	De documentroot die gebruikt wordt door de betreffende virtuele host. Deze directory moet in elk geval leesbaar zijn voor de gebruiker wwwrun.
ErrorLog	Het bestand dat gebruikt moet worden voor het loggen van foutmeldingen. Dit bestand moet schrijfbaar zijn voor de gebruiker wwwrun.
CustomLog	De naam van het algemene logbestand. Dit bestand moet beschreven kunnen worden door de gebruiker wwwrun.
ScriptAlias	Deze directive kan gebruikt worden om te verwijzen naar een directory waarin scripts voorkomen die gebruikt worden om de inhoud van webpagina's dynamisch te genereren. Als geen gebruik gemaakt wordt van scripts, hebt u deze directive niet nodig.
<Directory "scriptdirectory">	Dit blok gegevens kan gedefinieerd worden indien gebruikgemaakt wordt van een aparte script directory. In dat geval plaatst u hier de instellingen die voor die directory van toepassing zijn.

Nadat u de instellingen voor virtuele hosts hebt aangepast, moet de Apache webserver opnieuw geladen worden. Vergeet ook niet te controleren dat de DNS-configuratie op uw systeem in orde is zodat de virtuele host herkend wordt.

4.7 Beperken van toegang tot de webserver

Onder normale omstandigheden krijgen alle computers die dat willen toegang tot de bestanden die door uw webserver worden aangeboden. U kunt hier echter wat tegen doen door toegang voor bepaalde gebruikers tegen te houden. Dit kan op basis van IP-adres, het kan ook op basis van namen van gebruikers die u gedefinieerd hebt.

4.7.1 Allow en Deny

Een zeer elementaire wijze om te bepalen wij toegang krijgen tot een bepaalde directory, is door middel van de Allow en Deny directives. Deze directives kunnen gebruikt worden om de toegang aan computers toe te staan of juist te ontzeggen.

```
<Directory "/var/www/htmldocs">
Order deny,allow
Deny from all
Allow from mijnwebserver.nl
</Directory>
```

Door middel van de directive “Order” bepaalt u om te beginnen in welke volgorde de toegangsregels bekeken moeten worden. U hebt hiervoor de volgende opties:

* **Deny,Allow**. Eerst wordt naar de deny directives gekeken, dan pas naar de allow directives. De standaardinstelling is dat geen toegang verleend wordt. Alleen gebruikers voor wie een Allow directive gedefinieerd is, krijgen toegang tot de server.
* **Allow,Deny**. Er wordt eerst gekeken naar allow directives en dan pas naar deny directives. De standaardinstelling is dat toegang verleend wordt. Alleen gebruikers waarvoor een Deny-directive gedefinieerd is, wordt toegang tot de server ontzegd. Deze werkwijze is een stuk minder veilig als de werkwijze waarbij Order Deny,Allow gebruikt wordt.
* **Mutual-failure**. Alleen hosts die voorkomen in de Allow-lijst en niet voorkomen in de deny-lijst krijgen toegang. Dit heeft hetzelfde effect als Order Allow,Deny. Er wordt voorkeur gegeven aan het gebruik van Order Allow,Deny in plaats van deze optie.

U kunt bovenstaande statements op verschillende manieren definiëren: met behulp van gedeeltelijke of volledige computernamen of op basis van volledige of gedeeltelijke IP-adressen. Daarnaast is het ook mogelijk een verwijzing naar “all” op te nemen om toegang uit te sluiten voor iedereen. Hieronder ziet u een voorbeeld hoe deze regels in de praktijk toegepast kunnen worden:

```
Order deny,allow
Deny from all
Allow from somedomain.com
Allow from 192.168.0.0/255.255.0.0
```

4.7.1 Toegang beperken op basis van gebruikersnamen

Naast authenticatie op basis van IP-adressen, is het ook mogelijk te authenticeren op basis van gebruikersnamen. Als u gebruikmaakt van deze mogelijkheid, moeten gebruikers zichzelf met hun gebruikersnaam bekend maken voordat ze toegang tot een directory krijgen waarop deze vorm van authenticatie van toepassing is. Voordat u echter gebruik kunt maken van authenticatie op basis van gebruikersnamen, moet u ervoor zorgen dat er gebruikersaccounts worden aangemaakt voor uw webserver. Op Fedora gebruikt u hiervoor de opdracht **htpasswd**, op SUSE Linux gebruikt u **htpasswd2**. De werking van de opdracht is verder voor beide systemen gelijk. In het volgende voorbeeld wordt een gebruikersaccount aangemaakt voor gebruiker Betty. In dit voorbeeld is Betty de eerste gebruiker in de wachtwoordendatabase, daarom wordt met de optie –c /etc/apache2/htpasswd aangegeven met welke naam en op welke locatie het bestand aangemaakt moet worden. Houd er rekening mee dat de locatie die hier vermeld is specifiek is voor SUSE Linux, op andere distributies zal gebruikgemaakt worden van een andere locatie zoals /etc/httpd.:

htpasswd2 –c /etc/apache2/htpasswd Betty.

Direct nadat op deze wijze een gebruikeraccount voor Betty is aangemaakt, verschijnt een prompt waarop een wachtwoord voor deze gebruiker ingevoerd moet worden. Dit wachtwoord wordt opgeslagen in het bestand htpasswd. Nadat de database met wachtwoorden eenmaal is aangemaakt met behulp van de parameter –c, hoeft deze parameter volgende keren niet meer gebruikt te worden. Om een volgende keer een gebruiker Franck toe te voegen aan het bestand, gebruikt u dus gewoon **htpasswd2 /etc/apache2/htpasswd Franck**. Het is ook mogelijk gebruikers vanuit deze database te verwijderen: maak in dat geval gebruik van de optie –D: gebruik bijvoorbeeld **htpasswd2 –D /etc/apache2/htpasswd bas** om gebruiker bas uit de database te halen zodat hij niet langer in kan loggen op beveiligde directories.

Nadat u een database hebt aangemaakt waarin gebruikersaccounts worden opgeslagen, kunt u daar vervolgens gebruik van maken wanneer u toegang verleent tot bepaalde directories. In het onderstaande ziet u een voorbeeld waarin dat gebeurt:

```
AuthType Basic
AuthName “Boekhouders”
AuthUserFile /etc/apache2/htpasswd
Require user Betty Alex
```

In dit voorbeeld wordt gewerkt met de authenticatiemethode “Basic”. Er is ook nog een geavanceerde methode maar die laten we hier buiten beschouwing. Vervolgens wordt met AuthName “Boekhouders” een naam gegeven aan de directory waartoe de toegang beperkt moet worden. Daarna wordt aangegeven welk bestand gebruikt wordt als gebruikersdatabase en tot slot wordt aangegeven welke gebruikers specifiek toegang krijgen tot de directrory in kwestie. In dit geval geldt dat dus voor gebruikers Betty en Alex. Let even op de wijze waarop gebruikersnamen in deze lijst van elkaar onderscheiden worden: u zet alleen een spatie en verder helemaal niets tussen de namen van de gebruikers. Als alternatief is het overigens ook mogelijk om alle gebruikers die een account hebben in het bestand htpasswd toegang te geven. Neem in dat geval de regel **Require user valid-user** op.

4.8 Apache beveiligen met OpenSSL

Standaard communicatie tussen een webserver en een browser is niet beveiligd. Dit betekent dat het verkeer onderschept kan worden door iemand die gebruikmaakt van een sniffer en dat uit de onderschepte pakketjes de verzonden data gereconstrueerd kunnen worden. Als het gaat om onschuldige data, is dat geen probleem. Zodra echter gegevens verzonden worden waarvoor het van belang is dat ze goed beveiligd zijn, is het van belang dat deze gegevens tijdens het transport versleuteld worden. Hiervoor kan gebruikgemaakt worden van SSL. De meest voor de hand liggende keuze voor het Linux-platform is OpenSSL.

4.8.1 De werking van Secure Sockets Layer

Bij communicatie op een netwerk wordt vaak gebruikgemaakt van een public/private-key paar. Deze sleutels kunnen voor verschillende doelen worden ingezet:

* Encryptie
* Bewijzen van identiteit
* Bewijzen dat een bericht tijdens transport niet gewijzigd is.

Basis voor al deze drie vormen van encryptie, zijn een publieke sleutel en een privé sleutel, we zullen het verder hebben over de public en de private key. Beide sleutels zijn wiskundig aan elkaar gerelateerd: ze hebben elkaar nodig om gebruikt te kunnen worden. De private key is alleen bekend bij de eigenaar van het sleutelpaar: deze sleutel wordt gebruikt om de vertrouwelijkheid van verschillende soorten boodschappen te kunnen garanderen, dus het is van het grootste belang dat deze sleutel ook geheim blijft. De public key daarentegen wordt aan de hele wereld beschikbaar gesteld.
De voornaamste toepassing van public/private keys is encryptie. Als Franck bijvoorbeeld een bericht wil sturen aan Audrey en dat wil versleutelen, heeft Franck de publieke sleutel van Audrey nodig. Het bericht wordt vervolgens versleuteld met deze public key en naar Audrey gestuurd. Zij kan het bericht op haar beurt lezen door het te ontcijferen met haar private key. Het mooie van het systeem is dat iedereen kan beschikken over de public key van Audrey en haar zo versleutelde berichten kan sturen, maar dat om het bericht te ontcijferen de private key nodig is: Audrey kan dat alleen zelf doen.

Een andere belangrijke toepassing van public/private-keys, is het bewijzen van identiteit. Hierbij gebruikt een gebruiker zijn private key om te bewijzen dat een bericht ook echt van hem afkomstig is. Dit doet de gebruiker door een signature aan een bericht toe te voegen en deze te versleutelen met zijn private key. Als deez signature op basis van de public key van diezelfde gebruiker ontcijferd kan worden, wordt daarmee bewezen dat het bericht inderdaad van die gebruiker afkomstig is.
Nu is er met beide bovenstaande technieken een probleem: hoe weet iemand zeker dat hij inderdaad communiceert met de partij waarmee hij denkt te communiceren? Om daar een garantie voor te kunnen geven, wordt in netwerken gebruikgemaakt van de Certificate Authority. Een gebruiker stelt zijn public key beschikbaar in een PKI-certificaat. Op dit PKI-certificaat wordt vervolgens door de Certificate Authority (CA) een bewijs van echtheid geplaatst: de CA doet dit door het bewijs van echtheid te versleutelen met zijn private key. Wanneer een gebruiker die de public key van de CA heeft deze ondertekening kan ontcijferen, is daarmee bewezen dat het PKI-certificaat door de CA ondertekend is, en is het in hoge mate aannemelijk dat het bericht ook inderdaad van de beoogde gebruiker afkomstig is. In de meeste gevallen wordt gebruikgemaakt van PKI-certificaten die ondertekend zijn door een CA die algemeen bekend is, zoals bijvoorbeeld Verisign. Deze techniek wordt onder andere ook ingezet wanneer de communicatie met een webserver beveiligd wordt door middel van SSL. Dit gaat als volgt:
1. De browser maakt contact met de webserver op de HTTPS-poort 443 in plaats van de standaardpoort 80.
2. De browser vraagt aan de server om zijn public key en deze wordt direct daarop door de server verzonden.
3. De browser controleert of het PKI-certificaat dat van de server ontvangen is, ondertekend is door een geldige CA. Voor dit doel zijn browsers geconfigureerd met een lijst van CA's die wereldwijd algemeen bekend zijn.
4. Aks de sleutel geldig is, wordt een veilige verbinding tussen de browser en de server tot stand gebracht. Is de sleutel die van de server ontvangen is niet geldig, dan ziet de gebruiker een waarschuwing waarin gemeld wordt dat er problemen waren met de public key van de server. De gebruiker mag vervolgens zelf bepalen of hij de server in kwestie voldoende vertrouwt om er mee verder te gaan of niet.

4.8.2 Een PKI-certificaat aanmaken

Om veilig te kunnen communiceren met een webserver, is dus een PKI-certificaat nodig. Er zijn verschillende mogelijkheden om zo'n certificaat aan te maken:
* U neemt contact op met een commerciële Certificate Authority en dient een verzoek in om een PKI-certificaat te maken. Hieraan zijn de nodige kosten verbonden.
* U installeert in uw netwerk een CA en gebruikt deze om PKI-certificaten te ondertekenen.
* U maakt een testcertificaat om een en ander te configureren.

Houdt er rekening mee dat er voor een werkelijk veilige oplossing maar één echte mogelijkheid is en dat is gebruik te maken van de eerste optie. U klanten zullen immers snel afhaken wanneer ze op uw website vertrouwelijke gegevens achter moeten laten maar het PKI-certificaat dat door de server gebruikt wordt niet valide is. Als u daarentegen alleen te maken hebt met internet klanten en u de gelegenheid hebt ervoor te zorgen dat de interne CA op elk werkstation bekend is, is er niets op tegen gebruik te maken van een CA die in het eigen netwerk geïnstalleerd is. Omdat het voor dit hoofdstuk te ver gaat om zelf een CA in het leven te roepen, leest u nu hoe u zelf een testcertificaat aan kan maken. Voor commerciële toepassingen is deze werkwijze niet aanbevolen, maar u kunt hem uitstekend gebruiken om te testen dat het systeem werkt.

1. Om een sleutelpaar te kunnen maken, hebt u eerst een bestand nodig waarin zoveel mogelijk willekeurige getallen staan. Om zo'n bestand te genereren, maakt u gebruik van de random nummer generator in /dev/random. Gebruik de opdracht **cat /dev/randmom > /tmp/random** om een bestand te maken dat gevuld is met willekeurige getallen. Als u niets doet, gaat het aanmaken van dit bestand door tot in het oneindige: gebruik na een paar seconden de toetscombinatie Ctrl-C om te stoppen met aanmaken van het bestand. Langer is niet nodig, want in het volgende commando worden toch maar 1024 bits uit dit bestand met random nummers gebruikt.
2. Op basis van het bestand met de willekeurige getallen, kunt u nu een server key aanmaken. Gebruik hiervoor de opdracht **openssl genrsa –des3 –out server.key –rand /tmp/random 1024**. Het duurt enige momenten voordat dit commando voltooid is. Er zijn verschillende parameters die gebruikt kunnen worden om dit bestand aan te maken raadpleeg eventueel de man-agina van openssl voor meer details. Tijdens het aanmaken van de sleutels wordt u om een passphrase (wachtwoord) gevraagd. Dit wachtwoord wordt gebruikt om de private key van dit bestand te beveiligen. Houd er rekening mee dat deze passphrase elke keer wanneer de Apache server gestart wordt ook ingevoerd moet worden! Beide sleutels worden vervolgens opgeslagen in het bestand server.key. Dit bestand wordt aangemaakt in de huidige directory.

***random Het is uitstekend mogelijk om vanaf de console zelf een public/private-key paar voor gebruik met SSL aan te maken.

3. Nu moet een certificaat worden aangemaakt. Omdat er geen CA voorhanden is om de ondertekening van het certificaat te regelen, wordt in deze procedure een zogenaamd self-signed certificaat gemaakt. Om de signing van het certificaat te regelen, moet een behoorlijk aantal vragen beantwoord worden. Deze vragen worden gesteld zodat de identiteit van degene die het PKI-certificaat achterhaald kan worden. Dit is belangrijk voor gebruikers van het certificaat die meer informatie willen over de echtheid ervan. Als u dit certificaat voor uw webserver wilt gaan gebruiken, is het aan te raden in deze vragen dusdanige antwoorden te gebruiken dat het mogelijk wordt voor gebruikers om u te traceren. Gebruik de opdracht **openssl req –new –x509 –key server.key –out server.crt** om het certificaat te genereren.

***vragen Om met succes een certificaat aan te kunnen maken, moet aantwoord gegeven worden op een behoorlijk aantal vragen.

4. Als gevolg van de voorgaande procedure zijn er nu twee bestanden aangemaakt: het bestand server.crt waarin het certificaat van de server is opgeslagen en het bestand server.key met daarin de private key van de server. Om beide bestanden bruikbaar te maken voor uw Apache webserver, moet u ze nu kopiëren naar de directory waarin de configuratiebestanden van de Apache server zijn opgeslagen: /etc/apache2 op SUSE Linux en /etc/httpd op Fedora Linux. Eigenlijk maakt de exacte locatie waar u deze bestanden naartoe kopieert niet bijzonder veel uit; u verwijst namelijk in het Apache configuratiebestand toch naar de exacte locatie van deze bestanden.

4.8.3 Apache configureren voor gebruik van SSL

De procedure die nu gevolgd moet worden, hangt af van de Linux distributie waarvan u gebruikmaakt. We bespreken eerst welke werkwijze voor SUSE Linux gevolgd moet worden en vervolgens hoe u te werk moet gaan op Fedora Linux.

Apache 2 op SUSE configureren voor gebruik van SSL
Om te beginnen moet u op SUSE Linux het algemene instellingenbestand /etc/sysconfig/apache2 bewerken. Hierin regelt u als eerste dat het opstarten van de Apache server vertraagd wordt, zodat u in gelegenheid bent de passphrase in te voeren tijdens het opstarten. Dit doet u door ervoor te zorgen dat de volgende regel in het configuratiebestand is opgenomen: APACHE_START_TIMEOUT=”15”. Vervolgens zorgt u ervoor dat een aantal variabelen in de overige configuratiebestanden automatisch goed gezet worden met behulp van de parameter APACHE_SERVER_FLAGS=”SSL”. Als dit gebeurt is, zorg u er in het hoofdconfiguratiebestand voor dat de Apache server gebruikmaakt van de juiste parameters. Op SUSE Linux, bewerkt u hiervoor het bestand /etc/apache2/default-server.conf. Zorg dat in elk geval de volgende parameters in dit bestand zijn opgenomen. Deze regels zorgen ervoor dat uw server weet dat van SSL-encryptie gebruikgemaakt moet worden en waar de encryptiesleutels teruggevonden kunnen worden:

```
SSLEngine on
SSLCipherSuite
ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL
SSLCertificateFile /etc/apache2/server.crt
SSLCertificateKeyFile /etc/apache2/server.key
```

Tip ! Als u werkt met virtual hosts, hebt u ook de bovenstaande regels nodig. Neem ze dan op in de definitie van uw virtual host en zorg ervoor dat de virtual host gedefinieerd wordt met een directive als <VirtualHost uwhostnaam:443>

Nadat u deze wijzigingen hebt aangebracht, moet u de Apache server opnieuw starten. Tijdens het opstarten wordt gevraagd de passphrase voor het server key bestand in te voeren. Als de Apache server automatisch geactiveerd wordt tijdens het opstarten van uw server, wordt ook dan om de passphrase gevraagd. Dit betekent dat u altijd paraat moet staan wanneer uw server opnieuw gestart wordt (eigenlijk mag dat natuurlijk geen probleem zijn: Linux servers worden namelijk helemaal niet vaak opnieuw gestart!). Als u dit een probleem vindt, kunt u overwegen ervoor te zorgen dat de Apache server niet langer automatisch gestart wordt door hem uit de runlevelconfiguratie te verwijderen.

Apache2 configureren voor gebruik van SSL op andere Linux distributies.
Als u geen gebruikmaakt van SUSE, maar van een andere Linux distributie, volgt u een werkwijze die in grote mate lijkt op de werkwijze die in de bovenstaande procedure beschreven is. Toch zijn er een paar afwijkingen. De belangrijkste zit hem natuurlijk in de locatie en naam van de configuratiebestanden: op de meeste distributies zorgt u ervoor dat de onderstaande directives opgenomen worden in het bestand /etc/httpd/httpd.conf:

```
Listen 443
SSLEngine on
SSLCipherSuite
ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL
SSLCertificateFile /etc/apache2/server.crt
SSLCertificateKeyFile /etc/apache2/server.key
```

Let op de eerste regel uit de bovenstaande voorbeeldconfiguratie : hierin wordt aangegeven dat de Apache webserver ook op poort 443 moet luisteren. Op SUSE Linux is het niet nodig

deze regel op te nemen, hier wordt dit namelijk geregeld vanuit het apache configuratiebestand in de directory /etc/sysconfig.

4.9 Werken met Apache modules

Na installatie van een Apache webserver, hebt u aardig wat functionaliteit standaard al voorhanden. Heel veel mogelijkheden zijn echter ook niet standaard aanwezig. Denk bijvoorbeeld aan de mogelijkheid gebruik te maken van geavanceerde scripttalen als Perl en PHP. Daarnaast zijn er nog veel meer optionele zaken die voor de Apache server geregeld kunnen worden. Al deze functionaliteit is niet standaard aanwezig in de Apache webserver, maar wordt geregeld door bepaalde modules aan te zetten. Het spreekt voor zich dat hiervoor wel de juiste module geïnstalleerd moet zijn, maar dat gebeurt in veel gevallen automatisch. Als u bijvoorbeeld PHP installeert vanuit SUSE's YaST, hebt u daarbij gelegenheid om gelijk ook de bijbehorende Apache module weg te schrijven. Als dit gebeurt is, moet u er vervolgens zelf nog voor zorgen dat de module in kwestie ook wordt aangeroepen vanuit de Apache configuratiebestanden.

***apamoduleModules waarmee de functionaliteit van Apache uitgebreid kan worden, kunnen gewoon als RPM op uw server geïnstalleerd worden.

Wanneer u ervan verzekerd bent dat een module op uw systeem geïnstalleerd is, kunt u hem vanuit het Apache configuratiebestand aanroepen. Maak hiervoor gebruik van het statement **LoadModule**. Gebruik bijvoorbeeld **LoadModule php4** om tijdens het laden van de Apache webserver de betreffende module ook te activeren.

Opdracht 4.1
Configuratie van de Apache webserver
In deze oefening configureert u de apache webserver. Hierbij wordt gebruikgemaakt van alle besproken features. De exacte invulling van de stappen van de oefening kunt u in de voorgaande tekst terugvinden. Omdat er verschillen bestaan in de werkwijze tussen de distrubuties, worden de uit te voeren stappen in de oefening niet stuk voor stuk beschreven.

1. Controleer of Apache2 op uw server geïnstalleerd is. Gebruik hiervoor de opdracht **rpm –qf /usr/sbin/httpd**. Volgt op deze opdracht geen melding van een of andere apache package? Gebruik dan de package manager van uw distributie om de Apache packages handmatig te installeren.
2. Controleer dat u de standaard webpagina van de zojuist geconfigureerde Apache server kunt benaderen op http://localhost. Lukt dit niet? Analyseer dan het configuratiebestand van uw Apache webserver om te kijken welke directory staat ingesteld als documentroot en controleer dat zich in deze directory een bestand met de naam index.html bevindt.
3. Zorg er nu voor dat er een virtual host gemaakt wordt met de naam Sales. Let daarbij op de volgende zaken:
- U moet voor deze virtuele host een aparte subdirectory maken
- Maak het gebruikersaccount dat door de apache server gebruikt wordt eigenaar van deze directory
- Zorg ervoor dat in de betreffende directory een bestand bestaat met de naam index.html. U kunt zelf een bestand aanmaken, u kunt ook het bestaande bestand vanuit de documentroot naar deze directory kopiëren en aanpassen zodat u kunt zien dat u inderdaad op de virtuele host aanwezig bent. Vergeet niet ervoor te zorgen dat het account dat door de Apache server gebruikt wordt ook eigenaar is van dit bestand!

- Maak een entry aan voor de virtuele host in de apache configuratie. Geef de virtuele host de naam Sales.mijndomein.com. Zorg ervoor dat deze naam op zijn minst via /etc/hosts achterhaald kan worden.
- Start de Apache server opnieuw op en kijk of u de virtuele host op http://sales.mijndomein.com kunt benaderen.

4. Gebruik de opdracht htpasswd of htpasswd2 om een gebruikersdatabase voor de Apache webserver te maken. Neem vervolgens een directory block op dat u alleen toegankelijk maakt voor alle gebruikers die in dit bestand gedefinieerd zijn. Vergeet niet om de directory waarnaar u verwijst ook daadwerkelijk aan te maken, er een voorbeeld HTML-bestand neer te zetten en de Apache server opnieuw te starten nadat u deze directory hebt aangemaakt.
5. Maak een sleutelpaar aan dat door SSL gebruikt kan worden en zorg ervoor dat dit sleutelpaar wordt ingezet zodat uw Apache server ook op de beveiligde poort 443 benaderd kan worden.

4.10 Configuratie van een Squid Proxy server

In het voorgaande hebt u gelezen hoe u een Apache webserver op uw server kunt installeren en configureren. Een andere aan http-gerelateerde server die veelvuldig wordt ingezet, is de Squid Proxy server. De Squid Proxy wordt ingezet als web-cache server. Hierbij neemt een gebruiker contact op met de Proxy server, de Proxy server neemt op zijn beurt contact op met de server op internet die benaderd moet worden. Een dergelijke constructie biedt meerdere voordelen:

* Er is geen rechtstreeks contact tussen de eindgebruiker en het internet
* Op de Proxy kan gewerkt worden met gebruikersauthenticatie
* De Proxy functioneert als cache: alle pagina's die recentelijk zijn opgevraagd worden in het geheugen van de Proxy bewaard zodat ze een volgende keer niet meer helemaal bij de originele site op internet binnengehaald hoeven worden.

4.10.1 Inzetten van een proxy

Proxy servers kunnen op verschillende manieren worden ingezet. Het meest belangrijke criterium zit hem echter in de protocollen die door de Proxy ondersteund worden. Traiditionele proxy's zijn puur ontworpen voor het werken met http, Squid kan echter ook andere protocollen aan zoals FTP en zelfs het archaïsche Gopher wordt nog steeds ondersteund. Daarnaast worden ook DNS-aanvragen ondersteund en wordt er een cache bijgehouden van negatieve antwoorden. Voor elk van deze verschillende opties kan ingesteld worden hoe de optie zich precies moet gedragen.

Squid kan voor verschillende doelen ingezet worden:

* bandbreedtebesparing
* load balancing
* tijd besparen
* seurity

Het belangrijkste voordeel van een goed geconfigureerde Squid server, zit hem in de snelheidswinst: een pagina die gecached is door de Squid Proxy hoeft niet meer helemaal bij de originele site opgehaald te worden, maar kan gewoon worden ingelezen vanuit de Proxy. Op deze wijze kan een goed geconfigureerde Proxy ook functioneren als http-accelerator: hierbij wordt de Proxy puur ingezet met als doel een webpagina sneller beschikbaar te stellen voor de eindgebruikers die hem nodig hebben. Een http-accelerator kan natuurlijk worden ingezet op de locatie van de eindgebruiker, het is ook mogelijk dit type Proxy neer te zetten op de locatie waar de website staat om de toegang tot zwaar belaste websites te verlichten.

4.10.2 Installatie van Squid

De squid installatiebestanden worden meegeleverd met alle gangbare distributies. U kunt ze dus vanaf de installatiemedia van uw distributie installeren. Ook is het mogelijk de meest recente versie te downloaden van de squid website: ga hiervoor naar www.squid-cache.org. Wij raden u echter aan gebruik te maken van de bestanden die geleverd worden bij uw distributie: over het algemeen kunt u zo veel problemen voorkomen omdat deze bestanden het beste zijn afgestemd op de rest van uw distributie.
Voor wat betreft de Squid configuratie bestaan er weinig verschillen tussen SUSE en Fedora: er er wordt gebruikgemaakt van dezelfde belangrijke bestanden en u vindt deze bestanden ook nog eens terug op dezelfde locatie. Het enige verschil is dat SUSE gebruikmaakt van het bestand /etc/sysconfig/Proxy voor het instellen van een aantal algemene variabelen, terwijl Fedora gebruikmaakt van het instellingenbestand /etc/sysconfig/squid voor het doen van een aantal variabelen die net een klein beetje afwijken. Beide distributies regelen in elk geval het grootste deel van de configuratie in het bestand /etc/squid/squid.conf. Dit bestand wordt uitgelezen door het squid proces /usr/sbin/squid dat natuurlijk tijdens het starten van een server automatisch geactiveerd kan worden.

4.10.3 Definitie van de Squid configuratie

In het uitermate goed gedocumenteerde bestand /etc/squid/suid.conf definieert u het grootste deel van de werking van de Squid Proxy. In dit omvangrijke bestand (meer dan 3000 regels) wordt met behulp van verschillende parameters precies aangegeven wat er moet gebeuren wanneer Squid gestart wordt. We bespreken een aantal van de meest gangbare tags die in dit bestand gedefinieerd worden:

* **http_port**: hiermee definieert u de poort waarop de Squid Proxy luistert naar binnenkomend verkeer. De standaard poort is 3128, veel eindgebruikers zullen u echter dankbaar zijn wanneer u deze poort instelt op 8080 omdat veel andere proxy's (met name ook die van een niet nader te noemen leverancier) standaard op dze poort bereikbaar zijn.
* **cache_dir**: ook dit is een heel belangrijke parameter. Hiermee geeft u namelijk aan in welke directory bestanden opgeslagen moeten worden die door Squid gecached worden. De performance van de Squid-proxy is in veel gevallen recht evenredig aan de prestaties die door deze directory geleverd worden. Op zware servers die speciaal als Squid-proxy worden ingezet, is het niet ongebruikelijk meerdere schijven toe te wijzen aan deze paramter zodat snel zeer veel bestanden gecached kunnen worden. In deze directory wordt door Squid een groot aantal subdirectories aangemaakt waarin de te cachen bestanden worden opgeslagen. De reden hiervoor is dat het sneller is om veel bestanden op te slaan in veel kleine directories, dan een groot aantal bestanden in één grote directory. Met de definitie van de parameter cache_dir kunt u aangeven welke directory voor dit doel gebruikt moet worden, hoe groot deze directory maximaal mag worden en hoeveel subdirectories er maximaal in deze directory aangemaakt mogen worden. De standaardinstelling staat op **cache_dir ufs /var/cache/squid 100 16 256**: te cachen bestanden worden opgeslagen in de directory /var/cache/squid, deze directory mag maximaal 100 MB groot worden; direct onder deze directory mogen 16 subdirectories worden aangemaakt en daar weer onder mogen nog eens 256 subdirectories worden aangemaakt voor een mooie gedistribueerde directory tree. Houd er rekening mee dat de maximale grootte van 100 MB voor zeer actieve proxy's aan de krappe kant is.
* **http_access** en **acl**: deze parameters worden samen gebruikt om aan te geven wat er allemaal is toegestaan vanaf de Squid Proxy. Om te beginnen wordt met **acl** een netwerk gedefinieerd; bijvoorbeeld **acl private src 192.168.0.0/255.255.0.0**: alle IP-adressen die beginnen met 192.168 worden beschouwd als IP-adressen in het netwerk "private". Vervolgens wordt gedefinieerd op welke wijze computers vanaf dat netwerk toegang krigen,

bijvoorbeeld **http_access allow private**. Met deze twee eenvoudige regels wordt toegang gegeven aan alle computers op het netwerk private. Houdt er rekening mee dat hiermee voor alle andere computers het omgekeerde geïmpliceerd wordt; dat is een standaardinstelling voor een Squid-proxy. Computers waarvan het IP-adres begint met iets anders als 192.168, krijgen dus geen toegang. Het definiëren van access regels kan een zeer complexe taak zijn, u leest hier verderop in dit hoofdstuk meer over.

* **authenticate_program**. Squid kan ook gebruikmaken van gebruikersauthenticatie. Als dat gebeurt, moet natuurlijk wel duidelijk gemaakt worden welk programma hiervoor wordt ingezet. Dit doet u met de parameter **authenticat_program**.

Naast deze vier standaardparameters kunnen nog veel meer verschillende instellingen gebruikt worden, wij raden u aan de documentatie van squid.conf te lezen voor meer informatie. Als u wijzigingen hebt aangebracht in het algemene configuratiebestand, moeten deze wijzigingen vervolgens geactiveerd worden. Gebruik hiervoor de opdracht **squid –k reconfigure**; de wijzigingen worden vervolgens direct geactiveerd zonder dat het proces helemaal opnieuw gestart hoeft te worden.

4.10.4 Authenticatie

Voor gebruik van de Squid-proxy, is het mogelijk te werken met authenticatie. Dit kan plaatsvinden op verschillende niveau's: u kunt authenticeren op basis van netwerk, maar het is ook mogelijk te werken met authenticatie van individuele gebruikers. Als authenticatie nodig is, zorgt Squid er automatisch voor dat de gebruiker een bericht ziet waarin hij gevraagd wordt zichzelf bekend te maken.
Wanneer gebruikersauthenticatie door de Squid Proxy wordt afgehandeld, bestaan daarvoor twee mogelijkheden. Om te beginnen is er de digest mode. Hierbij wordt een wachtwoord locaal gecodeerd en authenticeert de gebruiker op basis van het gecodeerde wachtwoord. Het voordeel van deze werkwijze is dat het wachtwoord nooit in clear-text over de kabel verstuurd kan worden. Deze methode is dan ook aan te raden, maar wordt niet door elke versie van de Squid-proxy ondersteund. In basic mode worden de gebruikersnaam en wachtwoord gecodeerd met base64. Dit is geen echte versleuteling, u moet een wachtwoord dat op deze wijze verstuurd wordt dan ook als clear-text en uitermate onveilig beschouwen.

Om duidelijk te maken welk programma gebruikt wordt voor authenticatie, gebruikt u in squid.conf de parameter **auth_param basic program**. Zo kunt u bijvoorbeeld verwijzen naar de MSNT-module om authenticatie af te handelen op een Windows NT server. In een Linux omgeving is het toch handiger om voor dit doel gebruik te maken van PAM. Het grote voordeel van deze werkwijze is namelijk dat u via het PAM-configuratiebestand verder af kunt handelen welk authenticatiemechanisme gebruikt moet worden en in de Squid-configuratie verder niets geregeld hoeft te worden U regelt dit door het PAM-mechanisme aan te roepen met de regel **auth_param basic program /usr/sbin/pam_auth**. Elke keer wanneer er geauthenticeerd moet worden, gebeurt dat nu volgens de instellingen in het PAM-configuratiebestand /etc/pam.d/squid:

```
#%PAM-1.0
auth    required        pam_unix2.so
account         required        pam_unix2.so
```

In dit voorbeeldbestand dat ontleend is aan SUSE Linux, wordt voor authenticatie gebruikgemaakt van het traditionele UNIX loginmechanisme dat gedefinieerd is in de PAM-module pam_unix2.so. Elke andere methode om in te loggen die door PAM ondersteund

wordt, kan in plaats daarvan gebruikt worden. Of een gebruiker geauthenticeerd is, kan vervolgens worden afgevangen in het algemene mechanisme waarbij op basis van ACL's gedefinieerd wordt wat wel mag en wat niet mag.

4.10.5 Werken met ACL's

Een ACL is een lijst van obejcten die toegang krijgen. Er zijn verschillende manieren om deze ACL's samen te stellen. De meest eenvoudige wijze is om zelf de ACL te definiëren op basis van een IP-adres. Er zijn echter meer mogelijkheden om te werken met andere criteria. Hieronder ziet u een aantal voorbeelden van manieren waarop een ACL samengesteld kan worden:

* **acl localhost**: een standaard ACL voor alle gebruikers op de locale computer
* **acl all**: ook deze ACL is standaard aanwezig en wordt gebruikt voor alle computers op het netwerk.
* **acl badpeople srcdomain .microsoft.com**: hierbij wordt de ACL ingericht op basis van de domeinnaam waar een gebruiker vandaan komt. Om dit te achterhalen, is het wel een vereiste dat reverse DNS geïmplementeerd is.
* **acl private src 192.168.0.0/255.255/0.0**: u hebt deze eerder voorbij zien komen: alle gebruikers waarvan het IP-adres begint met 192.168 zijn goed en krijgen toegang tot het systeem.
* **acl toolate time MTWHF 20:00-23:59**: hierbij wordt op basis van het keyword time gekeken naar de huidige tijd.
* **acl dirty dstdomain playboy.com sex.com fuck.com**: in deze ACL wordt niet gekeken naar waar de afzender van afkomstig is, maar waar een gebruiker naar toe wilt. Met behulp van de optie **dstdomain** kan een lijstje domeins gedefinieerd worden waarvan u liever niet hebt dat uw gebruiker zich daar naartoe begeeft.
* **acl password proxy_auth REQUIRED**: en tot slot wordt in deze ACL gekeken of de gebruiker in kwestie geauthenticeerd is en als dat niet het geval is, vindt de authenticatie alsnog plaats volgens het mechanimse dat gedefinieerd is met de optie **auth_param basic program**.

Nadat de ACL's gedefinieerd zijn, kunt u er vervolgens iets mee doen. Dit doet u door te werken met de parameter **http_access**. Met behulp van deze parameter definieert u wie er in staat gesteld wordt de Proxy te gebruiken en wat die gebruikers dan wel op het internet mogen. Deze parameter verwijst altijd naar ACL's die eerder in het configuratiebestand gedefinieerd zijn en kan dus niet op zich staand gebruikt worden. Er kan een aantal http_access regels achter elkaar gedefinieerd worden. Wanneer een gebruiker gebruik wil maken van de Proxy, worden al deze regels stuk voor stuk doorlopen. Op het moment dat er een treffer optreedt is het afgelopen: er wordt niet verder gekeken omdat er immers een hit ontstaan is. In het onderstaande voorbeeld ziet u een eenvoudig voorbeeld hoe de http_access regel gebruikt kan worden:

```
http_access deny dirty
http_access allow localhost
http_access allow password
http_access deny all
```

In dit voorbeeld wordt uitgegaan van de ACL's die op de voorgaande pagina gedefinieerd zijn. In de eerste regel wordt de toegang ontzegd tot alles wat niet netjes is. Gebruikers die naar www.fuck.com willen, worden dus direct tegengehouden en er wordt niet meer verder

gekeken. Vervolgens wordt toegang gegeven aan iedereen die vanaf de Proxy-server zelf een verbinding tot stand gebracht heeft en in de regel daarna wordt toegang verleend aan iedereen die zich netjes geauthenticeerd heeft. Voor ieder ander wordt de toegang ontzegd door middel van de regel http_access deny all.

Oefening 4.2
Voor uitvoering van deze oefening hebt u twee computers nodig. Een van deze computers wordt ingezet als Squid Proxyserver, de andere computer wordt ingezet als client. Regel in elk geval de volgende zaken:
- Alleen gebruikers die geauthenticeerd zijn, krijgen toegang. Regel dit via het PAM-mechanisme. Als u tijd over hebt, gebruikt u het PAM mechanisme om authenticatie af te handelen op een LDAP server, maar regel eerst dat normale users in /etc/passwd kunnen authenticeren.
- Alleen gebruikers die afkomstig zijn van het lokale netwerk mogen de Proxy gebruiken.
- Op doordeweekse dagen tussen 10 en 12 mag de Proxy door niemand gebruikt worden.
- De directory die door Squid gebruikt wordt om cache bestanden op te slaan, mag niet groter worden dan 500 MB.
- De Proxy moet benaderd kunnen worden op standaard poort 8080.

Samenvatting
In dit hoofdstuk hebt u een introductie gehad in het werken met de Apache webserver en de Squid-proxy. U hebt geleerd hoe u beide services voor elementair gebruik in uw netwerk in kunt zetten. Voor de Apache webserver weet u hoe u deze server gebruiksklaar kunt maken, inclusief de configuratie van virtuele webservers en beveiligde SSL-communicatie. Ook hebt u geleerd hoe u het gebruik van internet voor gebruikers in uw netwerk kunt reguleren en versnellen.

Oefenvragen.
1. Wat is er mis met de volgende access controls?
http_access deny all
http_access allow dirty
http_access allow password
http_access allow localhost
2. Hoe heet het hoofdbestand waarin op SUSE Linux de Apache configuratie geregeld wordt?
3. Welke device wordt gebruikt om een bestand met daarin heel veel willekeurige tekens samen te stellen?
4. Hoe komt de PAM-configuratie eruit te zien om iedereen die kan valideren op de LDAP-server toegang te geven tot de Squid-proxy?
5. Wat moet u doen om te zorgen dat wijzigingen in httpd.conf geactiveerd worden?
6. wat is het belangrijkste verschil tussen order Deny,Allow en order Allow,Deny?
7. In welke directory wordt de Apache configuratie opgeslagen op een Fedora Linux systeem?
8. Welk commando wordt gebruikt om een nieuw wachtwoordenbestand aan te maken waarin gebruiker Pleunie gedefinieerd is?
9. Waarvoor wordt op SUSE Linux het configuratiebestand uid.conf gebruikt?
10. In welke directory moeten Apache SSL-certificaten worden opgeslagen?

Hoofdstuk 5 Netwerk client beheer

Er zijn veel verschillende zaken die u zich voor zou kunnen stellen bij het beheer van de computers van een gebruikers in een netwerk. Om te beginnen moet u weten hoe deze computers automatisch voorzien kunnen worden van een IP-adres door de DHCP-server in het netwerk. Vervolgens is het van belang dat het inloggen centraal geregeld kan worden. Voordat we dit onderwerp serieus kunnen behandelen, is kennis nodig van het achtergrondmechanisme dat gebruikt wordt bij het inloggen van gebruikers, te weten Pluggable Authentication Modules. Dit systeem speelt namelijk een belangrijke rol wanneer u tijdens het inloggen gebruik wilt maken van een ander systeem dan de standaard gebruikersdatabase in /etc/passwd en /etc/shadow. Als laatste deel van dit hoofdstuk worden twee systemen besproken die gebruikt kunnen worden om beheer van clients in een netwerk te centraliseren. Als eerste is dat het oeroude Network Information Services (NIS) wat ook vandaag de dag nog steeds gebruikt wordt, daarnaast is dat het Lightweight Directory Access Protocol (LDAP) dat steeds vaker wordt ingezet als vervanging voor NIS.

Leerdoelen
* Clients automatisch voorzien van IP-gerelateerde informatie met behulp van de DHCP-server
* Inzicht in de werking van het PAM-systeem dat gebruikt wordt bij het afhandelen van authenticatie
* Configuratie en implementatie van NIS
* Configuratie en implementatie van LDAP

4.1 DHCP

Vrijwel elke router is vandaag de dag uitgerust met een ingebouwde DHCP-server. Toch kan het ook de moeite waard zijn om Linux voor dit doel in te richten, al is het alleen al om het grotere aantal opties dat wordt geboden. DHCP-servers die in routers zijn geïntegreerd, bieden vaak veel minder mogelijkheden In dit hoofdstuk leest u hoe u Linux kunt inzetten als DHCP-server. Ook leest u hoe u gebruik kunt maken van geavanceerde client opties wanneer u Linux gebruikt als DHCP-client.

4.1.1 DHCP en Bootp

Het Dynamic Host Configuration Protocol (DHCP) en het Bootstrap Protocol (BOOTP) zijn protocollen waarmee computers op het netwerk automatisch van protocolinformatie voorzien kunnen worden. BOOTP is lang geleden ontworpen om er voor te zorgen dat een diskless workstation voorzien kan worden van een IP-adres, DHCP is hier de qua functionaliteit sterk uitgebreide opvolger van.
Met behulp van DHCP kan veel meer informatie dan het IP-adres alleen naar de client gebracht worden. Zo kan bijvoorbeeld informatie over de standaardgateway, de DNS-nameservers en nog veel meer automatisch op het werkstation worden ingesteld. DHCP is volledig achterwaards compatible met BOOTP. U kunt met hedendaagse DHCP-servers gewoon nog BOOTP-parameters beheren en doorgeven aan werkstations die dat nodig hebben. Omdat vrijwel elk modern werkstation ook gewoon gebruik kan maken van DHCP, zult u deze mogelijkheid echter maar weinig gebruiken.

4.1.2 Werking van DHCP

Het DHCP-protocol maakt gebruik van twee componenten; de DHCP-server en de DHCP-client. De client verstuurt tijdens het opstarten een DHCP-verzoek (de DHCP-request) in de vorm van een broadcast naar de server. Omdat het een Broadcast is, gaat het pakketje dus niet

verder dan de lokale router en kan een DHCP-server op een ander netwerksegment niet bereikt worden. De DHCP-server stuurt als reactie op de DHCP-request een pakketje terug met daarin de benodigde informatie. Dit pakketje wordt ook wel de DHCP-offer genoemd. De client antwoordt hierop met een DHCP-ack om aan te geven dat hij met deze DHCP-server in zee wil gaan.De client werkt daarbij heel direct: hij gaat in zee met de eerste server die hem een configuratie aanbiedt.

Tip! U kunt informatie die door de DHCP-server gegeven wordt ook combineren met informatie die u lokaal instelt. Als u bijvoorbeeld op elk netwerk waar u uw pc gebruikt. gebruik wilt maken van één specifieke DNS-server, is het gewoon mogelijk het adres van die DNS-server hard in te stellen op de client. Het adres wat dan door de DHCP-server wordt aangeboden, wordt door de client genegeerd.

Om een DHCP-verzoek succesvol te laten verlopen, is het nodig dat de DHCP-server op hetzelfde netwerksegment als de client voorkomt; het verzoek wordt immers verstuurd in de vorm van een broadcast. Als de DHCP-server op een ander segment voorkomt, kan op de router gebruik gemaakt worden van de DHCP-relay-agent. Dit is een stukje software dat in staat is een DHCP-request door te sturen naar een DHCP-server op een ander netwerk. DHCP-relay-agents zijn beschikbaar voor vrijwel elke router. Hardware routers hebben hiervoor een speciale module, wanneer u een software-router zoals uw Linux server voor dit doel gebruikt, moet u een proces laden dat ervoor zorgt dat deze functionaliteit beschikbaar is.

***dhcprelay De DHCP-relay-agent zorgt ervoor dat een DHCP-verzoek wordt doorgestuurd naar een DHCP-server/

Een DHCP-server is in staat op twee manieren adressen toe te kennen:
* Als BOOTP-server
* Door middel van dynamische allocatie

Als de DHCP-server als BOOTP-server wordt ingezet, betekent dat dat een lijst van MAC-adressen met bijbehorende IP-adressen gegenereerd moet worden. Hierbij moet een tabel gemaakt worden, waarin voor elke node bepaald wordt welk IP-adres wordt uitgegeven. Dit is een bewerkelijke methode, maar hij heeft wel als voordeel dat elke node elke keer gegarandeerd weer hetzelfde IP-adres krijgt. Als u het niet op deze manier doet, zal het proces er voor proberen te zorgen dat een node weer zijn laatst gebruikte adres kan gebruiken, maar dat kan niet gegarandeerd worden.

Bij dynamische allocatie krijgt een node alleen voor een bepaalde periode de beschikking over een IP-adres. De duur van deze periode wordt gedefinieerd in de zogenaamde lease. Hierbij moet een reeks adressen die uitgedeeld kunnen worden (de range) gedefinieerd worden. Deze reeks bestaat uit een begin- en een eindadres. Elk adres dat tussen het begin en het eindadres voorkomt kan uitgedeeld worden. Sommige DHCP-servers staan bovendien toe dat binnen de reeks uitzonderingen gedefinieerd worden. Nadat de lease-periode verstreken is, moet de client opnieuw vragen of hij het adres nog een tijdje kan gebruiken. Het eerste verzoek de lease te verlengen wordt normaliter op de helft van de lease-periode gedaan. Hoe lang de lease periode precies duurt en welke parameters er in ingesteld worden, wordt bepaald in het algemene configuratiebestand dhcpd.conf.

4.1.3 Configuratie van de server: dhcpd.conf

Om een DHCP-server te configureren, moet u het configuratiebestand /etc/dhcpd.conf bewerken. Zoals vrijwel alle configuratiebestanden die voor Linux servers gebruikt worden, bestaat in dit bestand een groot aantal mogelijkheden om het gedrag van uw server te bepalen. We geven eerst een eenvoudig voorbeeld waarmee u snel een werkende server kunt maken, vervolgens vindt u een overzicht van de meest gebruikte opties die u in dhcpd.conf in kunt zetten. U kunt het onderstaande voorbeeldbestand overnemen, naar eigen wens aanpassen en uitproberen. Om het te activeren, hoeft u alleen nog maar het DHCP-serverproces dhcpd te activeren.

```
option domain-name "sandervanvugt.nl";
option domain-name-servers 192.168.0.10;

ddns-update-style ad-hoc;

default-lease-time 600;
max-lease-time 7200;

subnet 192.168.0.0 netmask 255.255.255.0 {
        range 192.168.0.100 192.168.0.200;
        option broadcast-address 192.168.0.255;
        option routers 192.168.0.1;
}
```

De eerste regel in het bovenstaande bestand definieert het standaard DNS-domein voor de werkstations. De regel daaronder geeft aan welke server als DNS-naamserver gebruikt moet worden. Let goed op het gebruik van de puntkomma aan het einde van elke regel, zonder de puntkomma begrijpt de DHCP-server niet dat het einde van een regel bereikt is. Als u hem vergeet, of een andere fout maakt in de syntaxis, zult u een foutmelding zien wanneer u de server probeert te starten. Als derde wordt de manier gedefinieerd waarop de server met Dynamic DNS-verzoeken om moet gaan. De enige reden waarom deze regel hier voorkomt, is omdat de DHCP-server anders weigert op te starten, daarnaast is het natuurlijk best handig dat deze regel bestaat wanneer u gebruik wilt maken van Dynamic DNS.

Vervolgens komt er een tweetal regels waarin gedefinieerd wordt hoe lang een DHCP-lease van kracht blijft. De waarden "600" en "7200" die hier gegeven zijn, geven de periode in minuten aan: een standaard lease duurt dus tien uur en bij communicatieproblemen waardoor de DHCP-server en client geen gegevens met elkaar uit kunnen wisselen, kan deze leaseperiode verlengd worden naar 5 dagen.

In het laatste blokje tenslotte worden de specificaties voor het netwerk gegeven. Als eerste zijn het betreffende netwerkadres en het bijbehorende subnetmasker gespecificeerd. Alle opties die voor dit netwerk ingesteld moeten worden, zijn daarna tussen accolades gegeven. Ook hierbij wordt weer elke regel afgesloten met een puntkomma. Op deze wijze kunnen in dhcpd.conf heel eenvoudig meerdere netwerken gedefinieerd worden door voor elk netwerk de bijbehorende opties op te nemen tussen accolades.

De eerste optie in de netwerkspecificatie is de verplichte optie "range". Hiermee worden de adressen bepaald die door de server uitgedeeld mogen worden. Dan volgen nog twee opties die niet verplicht zijn en die beiden voorafgegaan worden door de aanduiding "option". Allereerst is dat het te gebruiken broadcast-adres, daarnaast wordt het adres van de

standaardgateway gegeven. Vooral die laatste zult u in de praktijk altijd willen gebruiken om aan de clients duidelijk te maken welke weg naar buiten er bewandeld moet worden.

Het bovenstaande voorbeeldbestand kan gebruikt worden op een eenvoudige configuratie waar de DHCP-server adressen uitdeelt aan één subnet. Wanneer het DHCP-proces dhcpd geactiveerd wordt op een server die aan meerdere netwerken verbonden is, kan het nodig zijn dat voor elk van deze netwerken een definitie is opgenomen. Oudere versies zullen niet starten wanneer er netwerken bekend zijn die niet in dhcpd.conf gedefinieerd zijn, de nieuwere versies van dhcpd hebben hier echter geen moeite mee.

Wanneer u /etc/dhcpd.conf naar behoren hebt aangemaakt, kunt u de server starten. Dit kan met behulp van /usr/sbin/dhcpd, als alternatief kan ook gebruikgemaakt worden van het configuratiebestand dat voor dit doel meestal is opgenomen in /etc/init.d. Start hiermee bijvoorbeeld de DHCP-server door de opdracht /etc/init.d/dhcpd start te geven. Hiermee kunt u er ook voor zorgen dat de DHCP-server automatisch gestart wordt wanneer u de server aanzet.

4.1.4 Veel gebruikte configuratieopties

In dhcpd.conf kan gebruik gemaakt worden van verschillende opties, die vervolgens weer logisch bij elkaar gegroepeerd kunnen worden. Let bij het toepassen van deze opties even op wat u er precies mee wilt. Sommige opties zijn voor het hele netwerk bedoeld: plaats deze opties helemaal aan het begin van dhcpd.conf. Wanneer een optie bestemd is om alleen op één specifiek subnet gebruikt te worden (denk bijvoorbeeld aan de standaard-router), dan neemt u hem op in de definitie van het betreffende subnet. Daarnaast is het ook mogelijk zelfs voor individuele computers opties op te nemen in de DHCP-configuratie.

Het algemene doel van de opties is er voor te zorgen dat bepaalde informatie op de client-computer aanwezig is. Het DHCP-protocol voorziet hiervoor in meer dan tachtig verschillende mogelijkheden waaraan door sommige producenten zelfs nog specifieke opties worden toegevoegd. Zo zorgt bijvoorbeeld de optie "LPR server" er voor dat de client over een lijst van lpr-printservers beschikt. Vervolgens kan een toepassing ontwikkeld worden die gebruik maakt van deze informatie. Aangezien niet elke optie een toepassing heeft die er gebruik van maakt, is niet elke optie even zinnig. Zo zal misschien een Linux-werkstation nog wel iets kunnen met de instelling van een lpr-server, maar zal een Windows computer hier geen weg mee weten en de optie gewoon negeren.

Hieronder volgt een overzicht van de meest interessante opties. Voor een volledig overzicht kunt u de opdracht man dhcp-options gebruiken. Hou er rekening mee dat niet alle opties door alle client-programma's ondersteund worden. Het heeft dus geen zin naar een cookie-server te verwijzen als de client software niet begrijpt waar dit over gaat.

* **option subnet-mask ip-address;** Per netwerk dient een subnetmasker gegeven te worden. Als dit niet gebeurt, gebruikt dhcpd hetzelfde subnetmasker als dat wat gebruikt wordt op het netwerk waar het adres uitgedeeld wordt. In de meeste gevallen zal dit overigens gewoon goed gaan, definieer echter om problemen te voorkomen de optie subnet-mask.
* **option routers ip-adres[, ip-adres...];** Hiermee worden de adressen gegeven van routers die op het netwerk van de client voorkomen. Deze lijst wordt op volgorde afgewerkt. Dit betekent dat de belangrijkste router vooraan in de lijst moet staan. Voor alle duidelijkheid, deze optie moet u gebruiken om de Windows client-optie standaardgateway een waarde te geven.

* **option domain-name-servers ip-adres[, ip-adres...];** Specificeert de DNS-naamservers die door de clients gebruikt kunnen worden. De eerste server die hier genoemd wordt, wordt als eerste gebruikt, alleen als deze onbereikbaar is, wordt de volgende server in de lijst benaderd.
* **option domain-name naam;** Hiermee kunt u aangeven welke DNS-domeinnaam door de client gebruikt moet worden als deze contact maakt met de DNS-hiërarchie.
* **option ip-forwarding boolean;** Deze optie wordt gebruikt om aan te geven of een client al dan niet aan IP-packet forwarding doet. Deze optie bepaalt dus of een computer zich als router gedraagt of niet. Om deze eigenschap aan te zetten, moet de optie "1" gegeven worden. Deze optie heeft alleen zin om op te nemen voor afzonderlijke computers die als router worden ingezet.
* **option default-ip-ttl getal;** Hiermee kan aangegeven worden welke Time To Live (TTL) clients aan gegenereerde IP-packets mee moeten geven. Als u deze optie niet gebruikt, wordt gebruikgemaakt van de standaard optie die het betreffende besturingssysteem hiervoor gebruikt. Vanuit beveiligingsoptiek kan het handig zijn een andere waarde voor deze optie in te stellen, zodat een bepaald besturingssysteem door hackers herkend wordt als een ander besturingssysteem: bij het proces van fingerprinting waarbij achterhaald wordt met welk type besturingssysteem men te maken heeft, wordt namelijk vaak gekeken naar dit soort besturingssysteem specifieke instellingen.
* **option interface-mtu getal;** Hiermee kan in bytes de grootte van de maximum transfer unit (MTU) die door een interface gebruikt moet worden opgegeven worden. Gebruik van deze optie kan nuttig zijn wanneer er in uw netwerk routers voorkomen die niet goed overweg kunnen met te grote pakketjes. Op moderne werkstations zult u deze optie echter zelden nodig hebben.
* **option broadcast-address ip-adres;** Met deze optie kan het broadcast-adres dat op een subnet gebruikt moet worden gespecificeerd worden. Gebruik van deze optie is vooral aan te raden wanneer gebruikgemaakt wordt van een ander dan het standaard subnetmasker voor het betreffende netwerk.
* **option perform-mask-discovery boolean;** U kunt deze optie gebruiken om aan te geven of een client zelf door middel van ICMP het te gebruiken subnetmasker op het netwerk mag opzoeken. Deze optie is aan te bevelen wanneer u in een testomgeving werkt waarin het nog wel eens voor kan komen dat het subnetmasker op de router gewijzigd wordt maar is in andere situaties overbodig.
* **option router-discovery boolean;** Hiermee wordt aangegeven of een client door middel van ICMP router-discovery zoals gedefinieerd in RFC 1256 zelf routers mag gaan opzoeken.
* **option static-routes ip-adres ip-adres [,...];** Met deze optie kan een statische lijst van routes gedefinieerd worden die door de client in zijn werkgeheugen geladen moeten worden. Deze optie kan niet gebruikt worden om een standaardroute te definiëren; hiervoor is de optie "routers" nodig.
* **option nis-servers ip-adres[, ip-adres ...];** Met deze optie kan een lijst NIS-servers meegegeven worden. Meer informatie over NIS vindt u later in dit hoofdstuk.
* **option ntp-servers ip-adres[, ip-adres ...];** Deze optie wordt gebruikt om, in te gebruiken volgorde, een lijst van NTP tijdservers te geven.
* **option netbios-name-servers ip-adres [,ip-adres ...];** U kunt deze optie gebruiken om te specificeren welke NetBIOS (WINS) naamservers gebruikt moeten worden.
* **option netbios-node-type getal;** Met deze optie kunt u aangeven hoe computers op het NetBIOS netwerk teruggevonden moeten worden. De volgende types zijn beschikbaar:

1 B-node: (Broadcast) maakt gebruik van broadcast en geen WINS
2 P-node: (Peer) maakt alleen gebruik van WINS

4 M-node: (Mixed) maakt eerst gebruik van broadcast en dan van WINS
8 H-node (Hybride) maakt eerst gebruik van WINS en dan van broadcast.
* **option smtp-server ip-adres[, ip-adres ...];** Met deze optie wordt een lijst van SMTP-servers die beschikbaar zijn voor de client gegeven.
* **option pop-server ip-adres[, ip-adres ...];** Hiermee wordt een lijst gegeven van POP3-servers die voor de client beschikbaar zijn.
* **option nntp-server ip-adres[, ip-adres ...];** Geeft aan welke nieuwsservers gebruikt kunnen worden.
* **option irc-server ip-adres[, ip-adres ...];** Hiermee kunt u aangeven welke irc-servers voor clients beschikbaar zijn. Een IRC-server is een server die gebruikt kan worden voor IRC-chat, een oude vorm van instant messaging.
* **option www-server ip-address [, ip-address...];** Gebruik deze optie om aan te geven welke webserver voor een client ingesteld moet worden als standaard webserver.

De bovenstaande opties kunnen op verschillende manieren toegepast worden:
* per host;
* per groep hosts;
* per subnet;
* per gedeeld netwerk;
* globaal.

Zoals u straks zult zien, kunnen parameters voor afzonderlijke hosts ingesteld worden. Dit is met name handig als u wilt verzekeren dat voor een computer een aantal vaststaande gegevens is ingesteld. Als meerdere computers dezelfde gegevens met elkaar delen, is het de moeite daarvoor een host-group te maken. De parameters die hiervoor gedefinieerd worden, zijn geldig voor alle computers die in de groep voorkomen. U definieert in dat geval eerst de host group en neemt de volledige verzameling opties voor die host group op tussen accolades.

Weer een stapje hoger, kunnen parameters aan een subnet gekoppeld worden. De nodes in een subnet zijn alle hosts die hetzelfde IP-netwerkadres met elkaar gemeen hebben. Als het netwerkadres 193.173.100.0 met het subnetmasker 255.255.255.224 gebruikt wordt, komen dus 193.173.100.97 en 193.173.100.153 niet in hetzelfde subnet voor. Het is echter wel mogelijk om parameters te koppelen aan hosts die voorkomen binnen een zelfde netwerk, ongeacht het subnetmasker dat daarbij gebruikt wordt; u moet ze dan verbinden aan een "shared network".
De laatste mogelijkheid bestaat eruit om de parameters helemaal aan het begin van dhcpd.conf te definiëren; deze parameters worden dan als globaal beschouwd en gelden aan alle nodes waaraan de DHCP-server opties uitdeelt. Vooral algemene instellingen zoals die van de te gebruiken DNS-server lenen zich hier goed voor. Hieronder ziet u een voorbeeld waarin een en ander is toegepast in dhcpd.conf.

```
option domain-name "sandervanvugt.nl";

max-lease-time 1200;
default-lease-time 1200;

shared-network Zoetermeer {
        option domain-name-servers 192.168.237.239;
        subnet 192.168.193.0 netmask 255.255.255.224 {
                option routers 192.168.193.1;
```

```
        range 192.168.193.10 192.168.193.20;
    }
    subnet 192.168.193.32 netmask 255.255.255.224 {
        option routers 192.168.193.33;
        range 192.168.193.40 192.168.193.50;
    }
}

group {
    max-lease-time 120;
    default-lease-time 120;
    host ceylan.azlan.nl {
        hardware ethernet 08:00:0b:ad:ca:fe;
        fixed-addres 192.168.193.21;
    }
    host caroline.azlan.nl {
        hardware ethernet 08:00:2f:ff:f3:49
        fixed-address 192.168.193.39
    }
}
```

In het bovenstaande voorbeeld ziet u tevens een aantal afzonderlijke computers met bijbehorende parameters gedefinieerd.

4.1.5 Configuratie van Dynamisch DNS

Als gebruik gemaakt wordt van dhcpd versie 3 of later (wat voor alle moderne Linux-distributies het geval is), kunt u deze instrueren gebruik te maken van Dynamisch DNS. Dit zorgt ervoor dat clients die met dynamische IP-adressen werken toch ook bekend gemaakt worden bij de DNS-server. Toepassing van DDNS kan bijvoorbeeld handig zijn in een omgeving waar printers door een DHCP-server van een adres worden voorzien. Hierdoor kunnen eindgebruikers die op basis van DNS-naam een bepaalde client willen benaderen daar altijd contact mee opnemen, ongeacht of het IP-adres van de client de laatste tijd nog gewijzigd is of niet; de DNS-naam verandert immers nooit als gebruikgemaakt wordt van dynamic DNS..

Dynamisch DNS kan plaatsvinden via de DHCP-server, maar het is ook mogelijk dat een DHCP-client zelf met de DNS-server gaat onderhandelen. Het heet echter de voorkeur om de DHCP-server in dit proces te laten bemiddelen. Het proces vindt dan als volgt plaats:

1. De DHCP-clients stuurt de hostnaam die hij wil gebruiken mee met het DHCP-request;
2. DHCP stuurt een bevestiging (acknowledgement) terug;
3. De DHCP-server neemt contact op met de DNS-server named en vraagt deze zijn zone bij te werken;
4. De DNS-server past zijn zone aan;
5. Wanneer de leaseperiode verstreken is, wordt de dynamische entry uit de DNS-zone automatisch weer verwijderd. Een volgende keer dat een DDNS client IP-configuratie aan de DHCP-server vraagt, herhaalt het proces zich weer.

Om met de DNS-server te communiceren, moet er gebruikgemaakt worden van een of andere vorm van beveiliging. Voor dit doel wordt met de opdracht **genDDNSkey** een public key aangemaakt die vervolgens door de DHCP-server gebruikt kan worden. De identiteit van zowel de DNS-server als van de DHCP-server kan dan wederzijds bevestigd worden. Om een en ander te configureren, voert u de volgende procedure uit.

1. Geef de opdracht **genDDNSkey** om de sleutel te genereren. Deze wordt vervolgens geplaatste in het bestand /etc/named.keys.
2. Pas /etc/dhcpd.conf aan. Hier moeten in elk geval de volgende regels in opgenomen worden:

```
ddns-update-style interim;
ignore client-updates;

include “/etc/named.keys”;
…
subnet 192.168.0.0 netmask 255.255.255.0 {
        range dynamic-bootp 192.168.0.201 192.168.0.219;

        zone uwdnszone. { primary 127.0.0.1; key DHCP_UPDATER; }
        zone 0.168.192.in-addr.arpa. { primary 127.0.0.1; key DHCP_UPDATER; }
}
```

Let erop dat u het bovenstaande configuratie aan moet passen aan uw eigen situatie. Om te beginnen wordt verwezen naar een DNS-server die op dezelfde server actief is als de DHCP-server. Daarnaast moet u uwdsnzone. vervangen door de werkelijke naam van de zone waarin de dynamische entries aangemaakt moeten worden.

3. Pas vervolgens de DNS-configuratie in /etc/named.conf aan zodat de dynamische updates ook geaccepteerd worden; in hoofdstuk 3 van dit boek kunt u meer lezen over de werking en configuratie van DNS:

```
include “/etc/named.keys”;

zone “uwdnszone” in {
        type master;
        file “dyn/uwdnszone.zone”;
        allow-update { key DHCP_UPDATER; };
        };
zone “0.168.192.in-addr.arpa” in {
        type master;
        file “dyn/0.168.192.zone”;
        allow-update { key DHCP_UPDATER; };
};
```

Let er op SUSE-Linux vervolgens ook op dat de variabelen in /etc/sysconfig aangepast moeten worden. In /etc/sysconfig/named vindt u de waarde NAMED_CONF_INCLUDE_FILES; geef deze parameter de waarde /etc/named.keys. U hebt dit nodig om ervoor te zorgen dat de DNS-server vanuit zijn chroot-omgeving toch bij dit configuratiebestand kan komen.

4.1.5 De DHCP-server

Voor het functioneren van DHCP, is het belangrijk dat een client, zolang de lease nog niet verlopen is, gebruik kan blijven maken van eenzelfde IP-adres. Daarbij is het niet nodig dat de client voortdurend contact onderhoudt met de DHCP-server; integendeel, als de server down is en de client is nog steeds gerechtigd om het IP-adres te gebruiken, hoeft hij niet opnieuw contact op te nemen met de server. Wanneer echter de DHCP-server down is en de client wordt opnieuw opgestart, is er wel een probleem. Wanneer een client namelijk opnieuw opstart, zal deze weer om een IP-configuratie vragen aan de DHCP-server. Wanneer deze server op dat moment niet beschikbaar is, is er dus voor de client ook geen configuratie beschikbaar en kan deze dus niet op het netwerk communiceren. Als de client gebruik maakt van het besturingssysteem Windows, zal hij op dat moment spontaan met een adres uit de 169.254-reeks gaan werken.

Om er voor te zorgen dat de server geen adressen uitdeelt die al uitgedeeld zijn, houdt de server een lijst bij van adressen die in gebruik zijn. Deze lijst wordt weggeschreven in het ASCII-bestand /etc/dhcpd.leases. Door elk uitgedeelde adres hier in weg te schrijven, kan verzekerd worden dat adressen nog achterhaald kunnen worden als de server onverhoopt down gaat. De server weet dan met wie het contact hersteld moet worden op het moment dat hij weer up komt.

De dhcp-daemon is niet in staat zelf te ontdekken of er wijzigingen zijn in het configuratiebestand dhcpd.conf; het is dus nodig de daemon opnieuw te starten nadat wijzigingen zijn aangebracht. Gebruik hiervoor de opdracht **killall –HUP dhcpd**. Als gebruik gemaakt wordt van een system-V style initialisatiebestand, dat bijvoorbeeld "dhcp" als naam heeft, kan de service opnieuw gestart worden door vanuit de runlevel-directory /etc/init.d de opdracht dhcp restart te geven.

4.1.6 De DHCP-client

Het inrichten van een DHCP-server heeft natuurlijk alleen maar nut wanneer er ook gebruikers zijn die er gebruik van kunnen maken. Elk besturingssysteem dat IP ondersteunt, heeft een implementatie van een DHCP-client. Bij sommige besturingssystemen zijn er wat eigenaardigheden die u in de gaten moet houden.

De Linux DHCP-client

Welke DHCP-client op uw distributie gebruikt wordt, is een beetje afhankelijk van de distributie die in gebruik is. In de meeste gevallen wordt gebruikgemaakt van dhcpcd, soms echter wordt gebruikgemaakt van het programma dhclient. Doorgaans zal dit tijdens opstarten vanuit de rc-scripts geactiveerd worden. De DHCP-client kan gebruik maken van twee configuratiebestanden, namelijk dhclient.conf en dhclient.leases. Hierin kan bepaald worden hoe de client zich moet gedragen wanneer deze gestart wordt.

In het bestand dhclient.conf kan bepaald worden welke informatie bij een DHCP-server opgevraagd moet worden. Daarnaast kan een lijst IP-adressen en andere informatie opgenomen worden, voor het geval dat als de client geactiveerd wordt geen DHCP-server beschikbaar is. Hieronder ziet u als voorbeeld het standaard voorbeeldconfiguratiebestand

```
send host-name "andare.fugue.com";
send dhcp-client-identifier 1:0:a0:24:ab:fb:9c;
send dhcp-lease-time 3600;
supersede domain-name "fugue.com home.vix.com";
```

```
prepend domain-name-servers 127.0.0.1;
requestsubnet-mask, broadcast-address, time-offset, routers, domain-name, domain-name-
servers, host-name;
require subnet-mask, domain-name-servers;
timeout 60;
retry 60;
reboot 10;
select-timeout 5;
initial-interval 2;
script "/etc/dhclient-script";
media "-link0 -link1 -link2", "link0, link1";
reject 192.33.137.209;

alias {
        interface "ep0";
        fixed-address 192.5.5.213;
        option subnet-mask 255.255.255.255;
}

lease {
        interface "ep0";
        fixed-address 192.33.137.200;
        medium "link0 link1";
        option host-name "andare.swiftmedia.com";
        option subnet-mask 255.255.255.0;
        option broadcast-address 192.33.137.255;
        option routers 192.33.137.250;
        option domain-name-servers 127.0.0.1;
        renew 2 2000/1/12 00:00:01;
        rebind 2 2000/1/12 00:00:01;
        expire 2 200/1/12 00:00:01;
}
```

U ziet dat het gedrag van een DHCP-client eigenlijk bij voorbaat volledig bepaald kan worden. Hieronder wordt een aantal parameters uit het voorgaande bestand toegelicht. De parameters die hier niet besproken worden, hebben een vergelijkbare betekenis als de overeenkomstige parameters in dhcpd.conf. Gebruik eventueel man dhclient.conf om de betekenis van ontbrekende instellingen te achterhalen.

* **timeout seconden**; Hiermee wordt de tijd opgegeven dat een DHCP-client maximaal wacht op antwoord van de DHCP-server. Als deze tijd verstreken is, besluit de client dat de server dus niet bereikbaar is. Vervolgens zal de client eerst kijken of er statische leases gedefinieerd zijn in dhclient.conf. Als dat het geval is, maakt de client hiervan gebruik. Is dit niet het geval, dan maakt de client gebruik van eventuele vorige, nog niet verlopen leases die voorkomen in dhclient.leases. Dit bestand wordt namelijk door elke Linux DHCP-client bijgehouden om informatie over de vorige lease te onthouden. Als deze er ook niet zijn, wordt na de retry-parameter opnieuw een DHCP-verzoek naar de server verstuurd.
* **retry seconden**; Dit is de tijd dat een client standaard wacht nadat hij geen DHCP-server gevonden heeft, voordat hij het opnieuw gaat proberen.

* **select-timeout seconden**; Als er meerdere DHCP-servers op een netwerk voorkomen, krijgt een client meerdere leases aangeboden. Daarbij kan het de voorkeur hebben dat de client zijn vorige adres weer opnieuw gaat gebruiken. Als u niets doet, gaat de client in zee met de eerste DHCP-server die zijn diensten beschikbaar stelt. Door de parameter select-timeout op een paar seconden in te stellen, wacht de client dat aantal seconden op andere servers die een aanbod doen. Als deze er zijn en als daaronder een server voorkomt die het vorige IP-adres van de client weer aanbiedt, zal de client daar vervolgens gebruik van maken.
* **reboot time seconden**; Als een client opnieuw opstart, probeert hij, als hij nog aan hetzelfde netwerk verbonden is, eerst zijn oude IP-adres terug te krijgen. De reboot-time geeft aan hoe lang de client hier maximaal op wacht.
* **request [optie] [, ... optie]**; Door middel van de parameter "request" kan de client specificeren welke opties door de server gegeven moeten worden.
* **require [optie] [, ... optie]**; De opties die als argument van "require" gegeven zijn, zijn verplichte opties. DHCP-offer-pakketjes waarin niet al deze opties voorkomen, worden niet geaccepteerd.
* **send [optie] [, ... optie]**; Met de parameter "send" kan de client een verzoek doen aan de server om een bepaalde parameter met een zekere waarde te geven
* **default [optie] [, ... optie]**; Middels de parameter “default” kan een standaardwaarde meegegeven worden voor bepaalde opties. Deze waarde wordt alleen ingesteld als de server de betreffende optie niet meelevert; zo kan voorkomen worden dat een voor de client kritische optie geen waarde heeft.
* **supersede [optie] [, ... optie]**; Om ervoor te zorgen dat opties van de server genegeerd worden en dat in plaats daarvan een lokale waarde gebruikt wordt, kan de optie "supersede" (vervangen) gebruikt worden. De parameter die hier gedefinieerd wordt, heeft voorrang boven de parameter zoals die door de server wordt aangeleverd.
* **prepend [optie] [, ... optie]**; Sommige parameters kunnen meerdere argumenten hebben. Het kan dan wenselijk zijn eerst een eigen, altijd gelijke waarde te gebruiken en daarna de waarden die door een server gegeven worden. Dit kan geregeld worden met het statement "prepend".
* **append [optie] [, ... optie]**; Als voor een bepaalde parameter eerst de waarden van de server gebruikt moeten worden en daarna de waarden die op de client staan ingesteld, kan dat geregeld worden met de optie "append". De parameters die hierbij als argument gegeven worden, worden achter in de lijst geplaatst.

Windows DHCP-clients

Gebruik van een DHCP-client op een Windows-computer is meestal erg eenvoudig; als het TCP-IP protocol geactiveerd wordt, zal Windows automatisch proberen een adres van een DHCP-server te krijgen. Als een computer geconfigureerd is met een vast IP-adres, kunt u onder de eigenschappen van het TCP-IP protocol instellen dat een adres van een DHCP-server verkregen moet worden.

Wanneer een Windows-client eenmaal gebruik maakt van een IP-adres dat door een DHCP-server is toegekend, kunt u deze adresinstelling met de opdracht **ipconfig** op de Windows-computer beheren (gebruik **winipcfg** op WIndows 9x en Me). Een veelvoorkomende taak is dat de huidige IP-configuratie door een Windows-computer losgelaten moet worden waarop opnieuw contact gezocht kan worden met de DHCP-server om een nieuwe configuratie te verkrijgen. Gebruik hiervoor achtereenvolgens de commando’s **ipconfig /release** en **ipconfig /renew**.

***windhcp Een Windows client is zo geconfigureerd dat hij altijd probeert een IP-adres te krijgen van een DHCP-server.

5.1.7 De DHCP-relay-agent

In de inleiding van dit hoofdstuk is besproken dat een dhcp-client zijn DHCP-pakketjes stuurt in de vorm van een broadcast. Aangezien broadcasts niet door routers worden doorgelaten, betekent dit dat de DHCP-server op hetzelfde netwerk als de client voor moet komen. Dit zou betekenen dat op elk netwerksegment waar DHCP-services nodig zijn, een DHCP-server geïnstalleerd moet worden, of dat de DHCP-server als module op de router actief moet zijn. Veel routers bieden voor dit laatste overigens inderdaad gewoon een mogelijkheid.

Om er voor te zorgen dat toch gebruik gemaakt kan worden van één DHCP-server voor meerdere netwerksegmenten, kan de DHCP relay-agent ingezet worden. Dit is een proces dat actief is op de router en er voor zorgt dat DHCP-pakketjes worden opgevangen en doorgestuurd naar de betreffende DHCP-server. In de meest eenvoudige manier, wordt de opdracht toegepast als **dhcrelay serveradres**. Dit betekent dat alle DHCP-pakketjes die door de relay-agent ontvangen worden, naar de gespecificeerde server worden doorgestuurd. De relay-agent doet dit voor verzoeken vanaf alle segmenten. Aangezien het soms niet wenselijk is dat de relay-agent op alle segmenten werkt, kunnen pakketjes die op een bepaalde interface binnenkomen genegeerd worden. Het is bijvoorbeeld handig om op deze wijze de WAN-interface uit te sluiten. Hiervoor wordt de parameter -i gebruikt; als argument van de optie –i worden de interfaces gegeven die wel geconfigureerd moeten worden. Alle andere interfaces worden dus automatisch niet geconfigureerd.

Gebruik bijvoorbeeld de opdracht **dhcrelay -i eth2 192.168.191.99** om ervoor te zorgen dat alleen DHCP-pakketjes die binnenkomen op interface eth2 worden doorgestuurd naar de DHCP-server die bereikbaar is op adres 192.168.191.99.

Houd er overigens rekening mee dat de DHCP-relay-agent ook gebruikt kan worden om fouttolerantie in te bouwen in een situatie waar gebruikgemaakt wordt van een DHCP-server. Clients doen namelijk altijd eerst een verzoek naar een DHCP-server, pas als dat verzoek niet gehonoreerd kan worden, probeert de client contact te maken met een relay agent. In een configuratie waarin twee netwerken gebruikt worden die door middel van een router aan elkaar verbonden zijn, zou u dus op beide netwerken een DHCP-server neer kunnen zetten. Deze DHCP-servers moeten dan geconfigureerd zijn met adresgegevens voor beide netwerken. Daarnaast configureert u een DHCP-relay-agent die verzoeken van het ene netwerk doorstuurt naar het andere netwerk als de DHCP-server daar om een of andere reden niet toe in staat was. Zo weet u zeker dat er toch nog een adres uitgedeeld kan worden waneer een DHCP-server tijdelijk even niet beschikbaar is.

Oefening 5.1
Configureer een DHCP-server voor gebruik in uw netwerk. Zorg ervoor dat in elk geval de volgende opties met de juiste waarden worden doorgegeven:
- standaardgateway
- DNS-server
- Te gebruiken subnetmasker

5.2 Pluggable Authentication Modules

Wanneer u op Linux iets met gebruikersauthenticatie te maken hebt, kunt u ervan verzekerd zijn dat op de achtergrond een rol gespeeld wordt door Pluggable Authentication Modules

(PAM). PAM is het systeem dat er op een moderne Linux-distributie voor zorgt dat gebruikers zich op Linux op een zeer flexibele wijze kunnen aanmelden (authenticeren). Aangezien deze service een fundamentele rol speelt bij het gebruik en beheer van veel netwerk-services, leest u in dit hoofdstuk hoe de service werkt. In andere hoofdstukken in dit boek zult u kennismaken met services die van PAM gebruikmaken.

5.2.1 Inleiding

Als we het hebben over gebruikers, dan hebben we het in feite over authenticatie, ofwel validatie van de gegevens waarmee de gebruiker zich op het netwerk bekend maakt om toegang te krijgen tot netwerkresources. Denk als voorbeeld aan de mogelijkheid van een gebruiker te authenticeren wanneer hij op de normale manier inlogt op zijn computer. Als u niets doet, wordt hierbij gebruikgemaakt van de bestanden waarin gebruikers gedefinieerd zijn: /etc/passwd en /etc/shadow. Wanneer u echter in een netwerkomgeving tientallen of honderden Linux-werkstations moet beheren, is het onhandig wanneer u op elk van deze computers lokale gebruikers aan moet maken. In zo'n geval is het handig als de authenticatie afgehandeld kan worden door een centrale Directory-service, zoals OpenLDAP waarover u later in dit hoofdstuk meer leest. Hierbij doet zich het probleem voor dat op dat moment de opdracht **login** wel moet weten dat het eerst in de Directory-service moet kijken of de gegevens waarmee de gebruiker zich bekend maakt (credentials) daarin geverifieerd kunnen worden. Om dit probleem op te lossen, is PAM ontworpen.

Dankzij PAM wordt de validatie van gebruikers dynamisch configureerbaar gemaakt. Dit betekent dat een systeembeheerder zelf kan bepalen van welke soort validatie gebruikgemaakt moet worden. Hiervoor wordt gebruikgemaakt van verschillende configuratiebestanden die aangemaakt zijn in de directory /etc/pam.d. In deze configuratiebestanden wordt een verband gelegd tussen de applicaties (services) enerzijds en anderzijds de Pluggable Authentication Modules (PAM's) die de daadwerkelijke taak van validatie van gebruikersgegevens uitvoeren. Hieronder ziet u een voorbeeld van zo'n bestand. In het voorbeeld ziet u het bestand /etc/pam.d/login dat is aangepast om tijdens de authenticatie eerst te kijken of ingelogd kan worden op de service die gedefinieerd is in de PAM-module pam_nam (met behulp van deze module kan aangemeld worden op een Novell's eDirectory, een krachtige Directory service die door middel van LDAP benaderd kan worden). Pas wanneer dat niet het geval is, gaat het loginproces kijken of er op de normale wijze via het configuratiebestand pam_unix2 ingelogd kan worden op /etc/passwd en /etc/shadow. Verderop in dit hoofdstuk leest u precies wat er allemaal in de verschillende PAM-modules gedefinieerd is.

```
auth    sufficient      /lib/security/pam_nam.so.0
account         sufficient      /lib/security/pam_nam.so.0
password        sufficient      /lib/security/pam_nam.so.0
session optional        /lib/security/pam_mkhomedir.so skel=/etc/skel umask =0022
session sufficient      /lib/security/pam_nam.so.0
auth    required                /lib/security/pam_unix2.so.0 nullok
account         required        /lib/security/pam_unix2.so.0
password        required        /lib/security/pam_pwcheck.so nullok
password        required        /lib/security/pam_unix2.so nullok use_first_pass use_authtok
session required        /lib/security/pam_unix2.so
```

5.2.2 De voordelen van PAM

Voorheen had elke toepassing zijn eigen database met wachtwoorden. Vanuit de beheersoptiek is dat niet handig. Dit betekent immers dat gebruikers geconfronteerd worden

met verschillende wachtwoorden en dat er voor de beheerder geen enkele manier is om in bepaalde gevallen verhoogde authenticatie-eisen te stellen. Denk bijvoorbeeld aan de mogelijkheid tijdens het inloggen niet alleen een gebruikersnaam en wachtwoord in te voeren, maar ook een fingerprint-reader. PAM maakt het mogelijk hier iets aan te doen. U kunt namelijk - zolang u tenminste over de broncode van het betreffende programma beschikt - elk programma opdragen de authenticatie af te handelen door gebruik te maken van PAM-modules. In andere gevallen kan een leverancier er zelf voor zorgen dat een PAM-module beschikbaar gesteld wordt. Welke modules dit zijn, kunt u regelen door op de juiste wijze een configuratiebestand te maken. Een aantal modules wordt bij elke Linux-distributie meegeleverd, andere modules worden geïnstalleerd bij de installatie van bepaalde toepassingen die iets met Authenticatie doen op uw Linux-server. Op deze manier wordt het mogelijk om op verschillende manieren een configuratie samen te stellen, van een configuratie waar beveiligde toegangsmethodes nagenoeg helemaal gedeactiveerd zijn, tot configuraties waar eerst een iris-scan-analyse gedaan wordt, gevolgd door een wachtwoord. De mogelijkheden zijn onbegrensd, zolang er maar een PAM-module beschikbaar is. Als deze module niet beschikbaar is, hoeft dat overigens ook geen probleem te zijn. Iedereen met voldoende kennis van zaken is namelijk in staat zijn eigen PAM-module te maken, u moet daarvoor echter wel kunnen programmeren.

De manier waarop PAM voor u als systeembeheerder interessant is, is tweeledig. Enerzijds is het mogelijk door middel van PAM alle applicaties op gelijke wijze hun authenticatie af te laten handelen. Als u het handig vindt dat alle authenticatie gebeurt op een Windows 2000 server, dan kan dat. Het voordeel hiervan mag duidelijk zijn; u hebt nog maar één punt in het netwerk waarop gebruikersgegevens bijgehouden moeten worden. Het tweede voordeel voor de beheerder is dat het mogelijk wordt om voor bepaalde toepassingen verhoogde eisen te stellen. Bij het werken met verschillende netwerk-services op Linux, zult u merken dat PAM heel vaak een rol speelt. Kort samengevat biedt PAM de volgende voordelen:

* Het authenticatiemechanisme wordt modulair en daardoor veel meer flexibel
* Het is heel eenvoudig mogelijk authenticatie centraal af te handelen
* Het wordt op eenvoudige wijze mogelijk gebruik te maken van krachtiger mechanismen voor authenticatie op het netwerk.

Om in de praktijk met PAM te kunnen werken, is er een aantal zaken waarvan u als beheerder op de hoogte moet zijn. Ten eerste is dat of een toepassing wel met PAM om kan gaan. Daarnaast moet u goed weten wat een module precies doet en wat de mogelijkheden zijn van een bepaalde module. Zo zijn er bijvoorbeeld PAM-modules die werken met opties die meegegeven moeten worden in het PAM-configuratiebestand, er zijn echter ook modules die gebruik maken van een apart configuratiebestand dat speciaal voor die module ontworpen is.

Zoals gezegd, in principe elke toepassing bevat de mogelijkheid om met PAM te werken, zolang deze optie maar meegeprogrammeerd is in de toepassing. Voor de beheerder is het interessant te weten óf een toepassing gebruik maakt van PAM-modules. Indien dit het geval is, kunt u dit zichtbaar maken met de opdracht **ldd programmanaam**. Geef bijvoorbeeld de opdracht **ldd /bin/login** om te achterhalen van welke modules de odpracht login gebruik maakt. Als "libpam" en "libpam_misc" niet in de lijst voorkomen, werkt het programma niet met PAM.

*** lddlogin Met de odpracht ldd kunt u achterhalen of een commando of service geprogrammeerd is om gebruik te maken van PAM.

Nadat bevestigd is dat een toepassing met PAM werkt, moet de beheerder ervoor zorgen dat in de directory /etc/pam.d een bestand wordt aangemaakt waarin wordt opgegeven welke PAM-modules gebruikt moeten worden. Dit bestand heeft doorgaans dezelfde naam als het programmabestand waarmee de toepassing gestart wordt. Zo heet het PAM-configuratiebestand voor /bin/login ook gewoon /etc/pam.d/login en wordt de PAM-configuratie voor de Proxy Squid geregeld in het bestand squid.

Of een programma daadwerkelijk goed met PAM werkt, is te testen door in het configuratiebestand te verwijzen naar de modules pam_permit.so en pam_warn.so. De eerste module zorgt ervoor dat aan iedereen toegang verleend wordt, de tweede module zorgt ervoor dat een bericht geschreven wordt naar syslog. Niet echt handig voor een operationele omgeving, maar wel een ideale wijze om te testen of de PAM-omgeving voor dat programma functioneel is. Voor de fictieve service "pamprog" komt het bijbehorende PAM-configuratiebestand er dus uit te zien als:

```
# /etc/pam.d/pamprog
auth    required        pam_permit.so
auth    required        pam_warn.so
```

Als u nu probeert deze service te gebruiken, zou u daar in de logbestanden van uw systeem iets van terug moeten vinden. In de meeste gevallen wordt voor dit doel gebruik gemaakt van het algemene logbestand /var/log/messages.

5.2.3 De werking van PAM

De aanvraag van een gebruiker om een bepaalde service te mogen gebruiken, wordt door PAM in vier onderdelen onderscheiden; u vindt deze onderdelen ook terug in /etc/pam.conf of de bestanden in /etc/pam.d. Het gaat hier om account management, authentication management, password management en session management. Deze vier onderdelen hebben betrekking op de volgende zaken:

* **account management**: Verificatie van het account, zoals: is het wachtwoord nog wel geldig en heeft de gebruiker rechten op de betreffende service.
* **authentication**: Het bevestigen van de identiteit van de gebruiker. Het kan hier gaan om challenge-response methodes zoals het invoeren van een wachtwoord, maar ook andere methodes behoren tot de mogelijkheid; bijvoorbeeld smart-cards en biometrische apparatuur, zodat u in kunt loggen waarbij gevalideerd wordt op basis van een vingerafdruk of iets dergelijks.
* **password**: Hierbij gaat het om de technieken die gebruikt worden om het wachtwoord te vernieuwen wanneer bijvoorbeeld de houdbaarheid verstreken is.
* **session**: Zaken die moeten gebeuren voordat de gebruiker toegang krijgt tot de service en nadat hij de toegang tot de service beëindigt. Denk aan het activeren van een auditing mechanisme, of het mounten van een home-directory.

Voor elke service wordt in de directory /etc/pam.d dus een apart configuratiebestand aangemaakt. In dit bestand wordt in een aantal regels alles gedefinieerd dat voor de service van toepassing is. Het gaat hier om de hierboven genoemde types account, auth, password en session. Voor elk van deze vier verschillende facetten kan een aantal modules aangeroepen worden. Per module kan worden aangegeven hoe belangrijk het is dat aan de criteria die door die module gesteld worden voldaan wordt:

* **requisite**: bij voorkomende problemen wordt het authenticatie-proces onmiddellijk afgebroken en wordt er niet eerst verder gekeken naar de andere modules.
* **required**: uiteindelijk wordt een fout gegenereerd, maar dit gebeurt pas nadat alle andere modules die bij deze service horen doorlopen zijn.
* **sufficient**: succesvolle verwerking van deze module betekent dat de hele authenticatieprocedure geslaagd is. De overige regels van hetzelfde type hoeven in dit geval niet langer doorlopen te worden.
* **optional**: succesvolle verwerking van deze module is alleen belangrijk als het de enige module is die verbonden is met deze service en dit type service.

Als laatste onderdeel van de PAM-configuratieregels wordt de bestandsnaam gegeven van de module die gebruikt moet worden. Standaard wordt naar deze modules gekeken in de directory /lib/security, het is echter mogelijk door een volledige bestandsnaam te gebruiken te verwijzen naar een module die op een andere plaats voorkomt. Tot slot worden argumenten gegeven die eventueel bij deze module gebruikt kunnen worden. Of er überhaupt argumenten gebruikt kunnen worden, wordt per afzonderlijke module bepaald. In het onderstaande voorbeeld kunt u zien hoe de PAM-configuratie voor de login-service er op een standaard Fedora installatie uitziet. Het is aardig om dit bestand snel te vergelijken met het configuratiebestand dat u in het begin van dit hoofdstuk hebt kunnen zien: het zal u opvallen dat het onderstaande Fedora-configuratiebestand gebruik maakt van totaal andere modules als het eerdere configuratiebestand dat aan SUSE-Linux ontleend is. De ontwikkelaar van een distributie is in deze keuze namelijk volkomen vrij.

```
#%PAM-1.0
auth            required        pam_securetty.so
auth            required        pam_stack.so service=system-auth
auth            required        pam_nologin.so
account                 required                pam_stack.so service=system-auth
password        required        pam_stack.so service=system-auth
session         required        pam_selinux.so multiple
session         required        pam_stack.so service-system-auth
session         optional        pam_console.so
```

5.2.4 Hoe veilig wilt u het?

Op een standaard Linux-systeem zal netjes voor elke afzonderlijke service die authenticatie behoeft een PAM-oplossing getroffen zijn. Hierdoor kunnen deze services op een veilige wijze gebruikt worden. Uiteindelijk wordt dus de toegang tot de services verleend door de PAM-configuratiebestanden. Dit betekent ook dat u geen toegang meer krijgt tot het systeem als deze bestanden per ongeluk verwijderd worden. Aangezien dit iedereen die met PAM dingen gaat uitproberen kan overkomen, zullen we hier eerst de procedure beschrijven die u toe kunt passen wanneer u onverhoopt niets meer kan met uw computer omdat u een fout gemaakt hebt in de PAM-configuratie. We gaan er daarbij van uit dat het ergst mogelijke gebeurd is: alle PAM-configuratiebestanden zijn per ongeluk verwijderd.

1. Start Linux opnieuw op in Single-user modus of start een rescue-systeem door uw distributie vanaf de installatie-cd te starten..
2. Kopieer eventueel bewaard gebleven bestanden en maak een nieuwe directory aan:

```
cd /etc
mv pam.conf pam.conf.orig
```

```
mv pam.d pam.d.orig
mkdir pam.d
cd pam.d
```

3. Maak vervolgens met een editor een bestand genaamd "other" aan in /etc/pam.d. Alle pam-toepassingen die zelf geen configuratiebestand hebben, kunnen hiervan gebruikmaken. In dit bestand komen de volgende regels:

```
auth            required        pam_unix_auth.so
account         required        pam_unix_acct.so
password        required        pam_unix_passwd.so
sessionrequired         pam_unix_session.so
```

Als u geen typefouten gemaakt hebt, kunt u nu weer op de normale manier inloggen. Is dit niet het geval, kijk dan in /var/log/messages of er aanwijzingen in zijn weggeschreven waaruit blijkt wat er aan de hand is.

4. Installeer in elk geval alle PAM-packages en "pwdb" opnieuw. Gebruik hiervoor een commando als **rpm -Uvh pam* pwdb*** vanuit de directory waarin zich de betreffende PAM-packages bevinden. Uiteraard kunt u ook gebruik maken van het gereedschap dat door uw distributie wordt aangeboden om software opnieuw te installeren. Vervolgens moeten in elk geval de laatste versies van libc, util-linux, wuftp en NetKit opnieuw geïnstalleerd worden.

5.2.5 Het bestand other

In de bovenstaande uitwerking is gebruikgemaakt van de optie "other". Door de instellingen in dit bestand helemaal open te zetten, kon op eenvoudige wijze toegang verleend worden aan iedereen. Op veel systemen is een zwakke opbouw van dit bestand echter een oorzaak van veel problemen. U zou ervoor moeten zorgen dat de inhoud ervan veilig mogelijk is en dat niemand buiten de gebruiker root rechten heeft de inhoud van het bestand aan te passen. U kunt dit bewerkstelligen door het bestand de volgende inhoud te geven. Op deze wijze zorgt u er namelijk voor dat alle toegang ontzegd wordt en een waarschuwing wordt weggeschreven naar het syslog-mechanisme op het moment dat iemand probeert via het bestand other contact te maken.

```
auth            required        pam_deny.so
auth            required        pam_warn.so
account                 required        pam_deny.so
account                 required        pam_warn.so
password        required        pam_deny.so
password        required        pam_warn.so
session         required        pam_deny.so
session         required        pam_warn.so
```

5.2.6 Beschikbare modules

Het zal u duidelijk zijn dat een succesvolle PAM-toepassing staat of valt met de modules die gebruikt worden. Hieronder treft u een overzicht van een aantal van de meest gebruikte PAM-modules. Dit overzicht is niet volledig en kan ook niet volledig zijn. De reden hiervan is dat door er nogal wat modules in omloop zijn die door verschillende leveranciers zelf ontwikkeld zijn en niet met standaard distributies meegeleverd worden. Ook zijn er tussen de distributies onderling nogal wat verschillen.

* **pam_access** Deze module zorgt ervoor dat op basis van de instellingen die hiervoor gedaan zijn in /etc/security/access.conf toegang wel of niet verleend wordt. Hiertoe moeten in access.conf namen van gebruikers en/of systemen die wel of geen toegang hebben worden opgenomen. Deze module werkt alleen op systemen waar niet gekeken wordt naar een .rhosts-bestand.
* **pam_chroot** Met deze module kunt u ervoor zorgen dat een gebruiker na authenticatie in een nep rootdirectory terecht komt; / is dan eigenlijk /ergens/anders. Het voordeel van deze werkwijze is dat u gastgebruikers die Shell-toegang krijgen tot uw systeem kunt beperken zodat ze niet op locaties terecht kunnen komen waar ze niets te zoeken hebben.
* **m_cracklib** Deze module zorgt ervoor dat wordt nagegaan of een nieuw wachtwoord wel veilig genoeg is. Hiervoor is het nodig dat de system library libcrack en het woordenboek /usr/lib/cracklib_dict op het systeem geïnstalleerd zijn. Elk nieuw in te voeren wachtwoord wordt dan eerst vergeleken met woorden in dit woordenboek. Als het nieuwe wachtwoord door deze test heen komt, wordt gekeken of het nieuwe wachtwoord niet te veel lijkt op het vorige wachtwoord. Zorg ervoor dat de pam_cracklib module altijd aangeroepen wordt vóór dat pam_pwdb wordt aangeroepen.
* **pam_deny** Deze module zorgt ervoor dat toegang ontzegd wordt, waarbij een foutmelding naar de betreffende toepassing wordt doorgegeven. Deze module kan in alle vier de componenten van PAM worden aangeroepen; bij aanroep bij het account-component wordt het onmogelijk in te loggen, indien aangeroepen bij het authentication-component wordt toegang tot de standaardtoepassingen ontzegd, bij toepassing op het password-component, is het niet langer toegestaan het wachtwoord te wijzigen en indien toegepast op het session-component wordt het een applicatie onmogelijk gemaakt een sessie te starten op een host-computer.
* **pam_ftp** Om als anonymous-gebruiker toegang te krijgen tot een FTP-server, kunt u deze module inzetten. De module vraagt tijdens de authenticatie van een anonymous-gebruiker op een FTP-server een mailadres van deze gebruiker dat als wachtwoord gebruikt wordt. Dit gebeurt alleen als een gebruiker met de naam "anonymous" of "ftp" zich aanmeldt bij het systeem.
* **pam_group** Deze module zorgt ervoor dat een gebruiker automatisch lid wordt van een bepaalde groep. Welke groepen hiervoor voor welke gebruikers gebruikt worden, wordt geregeld in /etc/security/group.conf. Groepen waarvan op deze manier lidmaatschap wordt toegewezen, worden toegevoegd aan de groepen waar de gebruiker al lid van is op basis van /etc/group.
* **pam_issue** Deze module zorgt ervoor dat het bestand /etc/issue getoond wordt voordat de login-prompt gegenereerd wordt. In /etc/issue kan een tekst worden opgenomen. Denk hierbij aan mededelingen die u aan de gebruikers wilt tonen voordat ze op uw server in kunnen loggen.
* **pam_limits** Op basis van deze module is het mogelijk om door gebruik te maken van het configuratiebestand /etc/security/limits.conf beperkingen in te stellen op resources die door gebruikers of groepen gebruikt kunnen worden. Deze limits gelden dan voor iedereen, met uitzondering de gebruikers waarvan het UID gelijk is aan 0. De instellingen die hiermee gedaan kunnen worden, doen denken aan de beperkingen die door middel van quota kunnen worden ingesteld.
* **pam_mail** Door middel van deze module kan na inloggen aangegeven worden of een gebruiker nieuwe mail heeft. De module kijkt hiervoor in de mail-directory van de gebruiker.
* **pam_mkhomedir** Als een gebruiker tijdens het inloggen geen homedirectory blijkt te hebben, zorgt deze module er voor dat er automatisch een wordt aangemaakt. Deze module kan met name nuttig gebruikt worden als een gebruiker door gebruik te maken van NIS of

LDAP inlogt op verschillende systemen. Het is dan niet nodig dat de systeembeheerder op elk van deze systemen handmatig een homedirectory aanmaakt. De module kan als volgt gebruikt worden: session required pam_mkhomedir.so skel=/etc/skel umask=0022. Hierbij wordt verwezen naar de skeleton directory. Dit is een directory waarin algemene bestanden geplaatst kunnen worden die bij het aanmaken van een gebruiker automatisch naar diens home-directory gekopieerd moeten worden.

* **pam_motd** Deze module zorgt ervoor dat het bestand /etc/motd automatisch wordt weergegeven nadat een gebruiker succesvol is aangemeld. De module kan als volgt gebruikt worden: login session pam_motd.so motd=/etc/motd. Let overigens op het verschil in gebruik met pam_issue dat er voor zorgt dat een melding gegenereerd wordt voordat de gebruiker een login-prompt te zien krijgt.
* **pam_nologin** Deze module zorgt ervoor dat het nologin-mechanisme in werking gezet wordt. Wanneer u gebruikmaakt van deze module, kunt u inloggen voor alle gebruikers met uitzondering van de gebruiker root onmogelijk maken door een bestand met de naam nologin aan te maken in de directory /etc op uw server. Alle andere gebruikers krijgen de inhoud van /etc/nologin te zien, zorg er dus voor dat in dit bestand een duidelijke melding gegeven wordt waarom u inloggen (tijdelijk) onmogelijk hebt gemaakt. Als dit bestand niet bestaat, doet pam_nologin niets. Om zinnig gebruik te maken van deze module, moeten alle logins er door beveiligd worden. Dit betekent dat het attribuut "required" er aan meegegeven moet worden.
* **pam_permit** Dit is wellicht de meest onveilige pam-module; hij zorgt er namelijk voor dat altijd en overal toegang wordt verleend. Het is hiervoor wel nodig dat een gebruikersnaam gegeven wordt en de meeste toepassingen willen ook nog dat dit een valide gebruikersnaam is, sommige toepassingen nemen echter ook genoegen met een ongeldige gebruikersnaam. Gebruik van deze module zet een systeem dus absoluut voor iedereen open, u moet hem dan ook uitsluitend voor testdoeleinden gebruiken. De module kan als volgt aangeroepen worden: login account required pam_permit.so.
* **pam_pwcheck** Deze module zorgt ervoor dat op basis van de instellingen die gedaan kunnen worden in /etc/login.defs gekeken wordt of een nieuw wachtwoord wel aan gestelde minimumeisen voldoet. Daarnaast wordt het nieuwe wachtwoord vergeleken met de cracklib-library, die dientengevolge ook geïnstalleerd moet zijn.
* **pam_pwdb** Deze module wordt gebruikt als vervanging voor de oudere pam_unix-modules. Indien aangeroepen vanuit het account-component, zorgt deze module ervoor dat gecontroleerd wordt of het account en bijbehorende wachtwoord nog geldig zijn. Om te achterhalen welk wachtwoord daar dan exact voor gebruikt moet worden, wordt gebruikgemaakt van het instellingenbestand /etc/pwdb.conf, waarin bijgehouden wordt in welke bestanden naar een wachtwoord gekeken wordt wanneer een gebruiker zich aanmeldt. Dit bestand /etc/pwdb.conf is ook het bestand dat u aan moet passen wanneer u er voor zou willen zorgen dat tijdens het inloggen niet langer gekeken wordt naar de inhoud van het bestand /etc/shadow. Laat overigens duidelijk zijn dat het af te raden is dit te doen, maar het is aardig te weten waar het geregeld wordt.
* **pam_rhosts_auth** Deze module bepaalt op welke wijze authenticatie bij een andere computer moet plaatsvinden als deze binnenkomt door een van de onveilige r-commando's (rlogin, rcp en rshell). De standaard methode hiervoor is dat eerst gebruik gemaakt wordt van de instellingen in /etc/hosts.equiv en vervolgens gekeken wordt in ~/.rhosts. het exacte gedrag kan echter door middel van argumenten beïnvloedt worden; zo bestaat er bijvoorbeeld de optie "no_rhosts", die ervoor zorgt dat de inhoud van het bestand ~/.rhosts niet bekeken wordt.
* **pam_rootok** Deze module zorgt ervoor dat de gebruiker gebruik kan maken van een service zonder dat hij daarvoor een wachtwoord hoeft in te voeren. Het klassieke voorbeeld is

dat de module gebruikt kan worden om "root" toegang tot services te geven door gebruik te maken van de odpracht su. Hiervoor zijn de volgende twee regels nodig:

```
su      auth    sufficient      pam_rootok.so
su      auth    required        pam_unix_auth.so
```

* **pam_securetty** Met behulp van deze module kunt u ervoor zorgen dat op basis van de terminals (tty's) die genoemd zijn in/etc/securetty bepaald wordt of de superuser (de gebruiker root) wel of geen toegang krijgt tot het systeem. Aangezien het vanuit de optiek van beveiliging absolute noodzaak is dat eerst dit bestand gecheckt wordt, voordat toegang verleend wordt, moet deze module met de flag "required" worden aangeroepen voordat enig andere module met de flag "sufficient" wordt aangeroepen. Zo kunt u er bijvoorbeeld voor zorgen dat de gebruiker root niet via een onveilige telnet-sessie binnen kan komen.
* **pam_stack** Dit is een algemene module die door Red Hat ontworpen is en in deze distributie en Fedora wordt toegepast. Pam_stack geldt als de algemene module die specifieke modules aanroept die voor bepaalde services relevant zijn. Deze specifieke modules zijn ook altijd modules die voorkomen in de directory /etc/pam.d.
* **pam_time** Deze module zorgt ervoor dat door gebruik te maken van de instellingen in het configuratiebestand /etc/security/time.conf de toegang tot bepaalde services verboden kan worden. Op elke regel in time.conf staan achtereenvolgens de services;ttys;gebruikers;tijden gespecificeerd. Zo kan bijvoorbeeld aan gebruiker "root" de toegang via pseudo-terminals tot de toepassing "xsh" ontzegd worden in het weekend en op maandag door in time.conf de regel xsh;ttyp*;root;!WdMo0000-2400 op te nemen. De belangrijkste beperking voor deze module is dat er momenteel niets is dat er voor zorgt dat een sessie wordt afgebroken op het moment dat een verboden tijdstip is aangebroken, wel maakt de service het onmogelijk na het aanbreken van het tijdstip de betreffende service nog te gebruiken.
* **pam_unix** Dit is de standaard Unix-authenticatie module die er voor zorgt dat op basis van /etc/passwd, shadow of NIS ingelogd kan worden. Deze module is inmiddels opgevolgd door pam_unix2.
* **pam_warn** Deze module zorgt ervoor dat waarschuwingen weggeschreven worden naar syslog. Op deze manier zorgt u ervoor dat wanneer er iets mis gaat u daar een melding van terug kunt vinden in uw logbestanden.

Oefening 5.2
Zorg ervoor dat het inloggen op uw computer aan banden wordt gelegd. Regel om te beginnen dat alleen op werkdagen tussen 6 uur 's ochtends en 11 uur 's avonds ingelogd mag worden. Zorg er ook voor dat elke keer dat iemand inlogt er een melding geschreven wordt naar syslog. Daarnaast mag de gebruiker root alleen inloggen vanaf een beveiligde sessie. Een normale login op de console moet als veilig beschouwd worden, een SSH-sessie is dat ook. Bedenk u echter goed welke bestanden u aan moet passen om op deze wijze een beveiligde sessie te regelen!

5.3 NIS

Op Linux-computers komen verschillende soorten bestanden voor waarin namen van het een of ander worden bijgehouden. Denk hierbij aan bijvoorbeeld namen van gebruikers, groepen en computers. Soms zijn deze bestanden uniek voor één computer omdat het computerspecifieke configuraties betreft. In andere gevallen is het wenselijk dat de inhoud van dergelijke bestanden beschikbaar gesteld wordt voor verschillende computers in een netwerk. Een van de klassieke manieren om hiervoor te zorgen, is door gebruik te maken van Network Information Services (NIS). In dit hoofdstuk leest u hoe u in uw eigen netwerkomgeving een NIS-structuur kunt installeren waarbij gebruik gemaakt wordt van NIS-

master en –slaveservers. Realiseer u voordat u aan het werk gaat om een NIS-server in te richten dat het hier om een oude werkwijze gaat die vooral in de jaren negentig populair was bij beheerders die meerdere UNIX-computers onder hun hoede hadden. Op moderne servers wordt in plaats van NIS vaak gebruikgemaakt van LDAP-servers. Toch is de kans nog steeds aanwezig dat u NIS tegen zult komen in een netwerk. Daarom moet u ervoor zorgen dat u op zijn minst weet hoe het systeem in elkaar zit.

5.3.1 Inleiding

Een van de belangrijke toepassingen van NIS is te voorzien in een gebruikersdatabase (/etc/passwd) die overal op het netwerk gebruikt kan worden. Hierdoor is het namelijk niet nodig een gebruiker meer dan eens te definiëren op een heel netwerk. Een andere veel voorkomende toepassing is dat NIS gebruikt wordt om mappings tussen computernamen en IP-adressen die op het netwerk gebruikt worden in onder te brengen. In dat geval wordt dus het bestand /etc/hosts ondergebracht in de NIS database. Het is echter ook mogelijk andere bestanden onder het beheer van NIS te brengen, zodat ze op een plaats beheerd worden maar wel vanaf elke computer in het netwerk toegankelijk zijn. In de onderstaande tabel treft u een overzicht van de bestanden die door NIS gedistribueerd kunnen worden:

Bestand	**Toepassing**
/etc/ethers	Mappings van MAC- naar IP-adressen
/etc/group	Definitie van groepen
/etc/hosts	Namen van computers en de bijbehorende IP-adressen
/etc/netmasks	Overzicht van subnetmaskers
/etc/passwd	Gebruikersdatabase. Voor de wachtwoorden in /etc/shadow wordt geen aparte map aangemaakt, maar de informatie uit passwd en shadow wordt samengevoegd in één map
/etc/protocols	Netwerkprotocollen en bijbehorende protocol-ID’s
/etc/rpc	RPC-adressen
/etc/services	Services en bijbehorende poort-adressen

Tabel 5.1 Bestanden die door NIS beheerd kunnen worden.

5.3.2 Remote Procedure Calls

Veel functionaliteit van NIS is ontleend aan de tools en libraries die geboden worden door Remote Procedure Calls (RPC). Naast NIS maakt ook het Network File System (NFS) gebruik van deze RPC’s. NIS wordt ook vaak gebruikt om een NFS-configuratie te ondersteunen. Een RPC-omgeving bestaat uit een RPC-server en een RPC-client. Een RPC-server levert een aantal verschillende procedures waarvan een client gebruik kan maken. Deze functionaliteit wordt beschikbaar gesteld door aparte programma’s op te starten. De namen van deze programma’s zijn eenvoudig te herkennen, ze beginnen meestal met de letters rpc. Zo levert bijvoorbeeld het programma rpc.nfsd de functionaliteit van een NFS-server. Voor u als beheerder is het om problemen met NIS op te kunnen lossen belangrijk dat u enig inzicht hebt in de werking van RPC.

Sinds de eerste versie van RPC die beschikbaar is gesteld, zijn er meerdere verbeteringen geweest. Het probleem bij deze verbeteringen is dat een verbeterde versie vaak niet

uitwisselbaar is met de voorgaande versie. Om toch communicatie mogelijk te maken, worden vaak meerdere versies van een RPC-programma geactiveerd. Hierdoor kan elke client in zijn eigen versie calls doen naar de RPC-server.

RPC-nummers
Elke RPC-service maakt gebruik van een eigen RPC-programmanummer. Vergelijk dit met poortadressen zoals die door TCP en UDP gebruikt worden, maar dan specifiek voor een RPC-omgeving. Deze programmanummers worden gedefinieerd in het bestand /etc/rpc. In dit bestand wordt eerst de naam van de service gegeven, vervolgens het bijbehorende RPC-nummer en als laatste eventueel een alias waaronder de service ook bereikbaar is. Hieronder ziet u een voorbeeld van een deel van dit bestand.

```
portmapper     100000          portmap sunrpc rpcbind
rstatd   100001          rstat rup perfmeter rstat_svc
ruserd  100002          rusers
nfs              100003          nfsprog
ypserv 100004          ypprog
mountd          100005          mount showmount
ypbind 100007
walld            100008          rwall shutdown
yppasswdd      100009          yppasswd
etherstatd       100010          etherstat
rquotad          100011          rquotaprog quota rquota
sprayd 100012          spray
3270_mapper  100013
rje_mapper     100014
selection_svc  100015          selnsvc
database_svc   100016
rexd             100017          rex
```

Portmap
Om gebruik te kunnen maken van RPC-programma's die allemaal hun eigen nummer hebben, moet de portmap-daemon actief zijn. Dit komt omdat elk RPC-programma zijn eigen, bij RPC geregistreerde RPC-nummer heeft. Omdat deze RPC-nummers niets betekenen voor de protocollen TCP en UDP, die zelf met poortnummers werken, moeten ze vertaald worden in een algemeen TCP/UDP poortadres. Portmap zorgt hiervoor.

Als een RPC-server gestart wordt, vertelt hij aan portmap naar welk poortadres hij luistert en voor welke RPC-programma's services geboden worden. Als een clientprogramma een RPC-call wil maken naar een bepaald programmanummer, zal het eerst contact opnemen met portmap op de server om te achterhalen naar welke TCP-poort de RPC-pakketjes gestuurd moeten worden. Het is derhalve noodzakelijk dat portmap gestart wordt voordat RPC-servers gestart worden.

RPCinfo
Om te kunnen achterhalen welke RPC-programma's op een server actief zijn, kan gebruik gemaakt worden van het commando rpcinfo. Het commando rpcinfo -p hostnaam geeft bijvoorbeeld een overzicht van alle RPC-services die op een host actief zijn. Dit commando wordt vooral gebruikt om problemen op te lossen. Zo kan als een service niet bereikbaar is, achterhaald worden of de benodigde RPC-programma's wel actief zijn.

***rpcinfo10 Met het commando rpcinfo kunt u achterhalen of noodzakelijke RPC-services op een computer actief zijn.

5.3.3 Installatie van NIS domeinen

Door computers te groeperen in een NIS-domein, wordt het mogelijk gemaakt om de configuratiebestanden voor een netwerk centraal te beheren. Dit gebeurt door middel van een centrale computer waarop al deze databases voorkomen; de NIS-master. Hierop wordt, op basis van de bovenvermelde configuratiebestanden een centrale database aangemaakt. Andere computers kunnen dan informatie opvragen uit deze centrale database. Vanuit het oogpunt van fout-tolerantie is het ook zeer aan te raden binnen een NIS-domein ook minimaal één NIS-slave te installeren. Dit is een computer waar de informatie die door NIS beheerd wordt ook vandaan gehaald kan worden. De NIS-slave is een kopie van de NIS-master. Het voordeel van het werken met een NIS-slave is dat u altijd door kunt werken wanneer de NIS-master tijdelijk niet beschikbaar is. Als u alleen een NIS-master hebt, is dat een single point of failure dat ervoor kan zorgen dat het totale netwerk onbereikbaar is.

***nisnetwerkIn een NIS-netwerkopstelling wordt gebruikgemaakt van verschillende servers waarbij de NIS-clients gegevens op kunnen halen.

Het ontwerp van NIS-domeinen staat los van andere domeinen, zoals ze bijvoorbeeld in DNS of Windows 2000 gebruikt worden. Vaak worden echter wel overeenkomende namen gebruikt, maar dit is niet noodzakelijk. Het bereik van een NIS-domein is vaak namelijk ook kleiner dan het bereik van een DNS-domein, NIS is typisch een service voor intern gebruik binnen een bedrijf. DNS-domeinen worden gebruikt om alle computers in een bedrijf onder te brengen in een hiërarchie die het mogelijk maakt dat alle computers op internet door middel van een unieke naam bereikt worden, NIS-domeinen worden gebruikt om administratieve informatie die bijgehouden wordt op een centrale computer in een netwerk te delen.

Om een NIS-domein werkend te krijgen moeten de volgende zaken gebeuren:

1. Er moet een naam voor het domein bepaald worden;
2. De NIS-master en slaves moeten geïdentificeerd en geconfigureerd worden;
3. De NIS-clients moeten in het domein ondergebracht worden;
4. Er moet bepaald worden in welke volgorde de clients naar informatie zoeken.

Voorbereidend werk

Om ervoor te zorgen dat het NIS-domein tot stand komt, moet op elke computer in het netwerk aangegeven worden in welk NIS-domein de computer voorkomt. Dit gebeurt door middel van de opdracht **domainname**. Het argument van dit commando is de naam van het domein waarin de computer voorkomt. Als u het op deze manier doet, zal het commando effectief zijn tot op het moment dat de computer opnieuw opgestart wordt. U moet er dus voor zorgen dat het in een van de opstartbestanden van de computer voorkomt.

***domainname Met de opdracht domainname kunt u instellen of bekijken bij welk NIS-domein een computer hoort.

Naast de specificatie van de juiste domeinnaam, moet ervoor gezorgd worden dat de rpc.portmapper actief is. Zoals gezegd zorgt deze service ervoor dat RPC-adressen vertaald

kunnen worden in TCP/IP-poortadressen. U kunt de RPC-poortmapper starten met behulp van het script portmap dat voor dit doel is aangemaakt in /etc/init.d.

Configuratie
Om een NIS-domein in te richten, moeten drie soorten computers geconfigureerd worden. Allereerst is er de NIS-master. Dit is de computer waarop de NIS-databases voorkomen. Eventuele wijzigingen in de gegevens die door NIS beheerd worden, moeten altijd gebeuren vanaf de NIS-master. Om een drukke NIS-master enigszins te ontlasten, kan ook gebruik gemaakt worden van NIS-slaves. Dit zijn computers waarop een kopie voorkomt van de NIS-database. Deze computers kunnen ook benaderd worden door de clients, maar er kunnen geen wijzigingen in gegevens aangebracht worden vanaf de NIS slaves. Als laatste zijn er de NIS-clients. Dit zijn de computers die gegevens opvragen van een NIS-master of slave. De NIS-client kan een NIS-master bereiken door middel van een broadcast, of door middel van een naam die is opgegeven in een configuratiebestand, u leest hier later meer over.

Configuratie van NIS master
De configuratie van een NIS-master begint met een controle van de bestanden die opgenomen moeten worden in de NIS-database. Zo moet u bijvoorbeeld het bestand /etc/passwd controleren op accounts die niet gebruikt worden of accounts die geen wachtwoord hebben. Dit zijn namelijk gaten in de beveiliging die hersteld moeten worden voordat de NIS-master daadwerkelijk in de lucht gebracht wordt; een dergelijk account kan namelijk binnen NIS gebruikt worden om illegaal toegang te verkrijgen tot andere computers in het netwerk. Houd er overigens rekening mee dat niet alle gebruikers uit /etc/passwd ook automatisch worden opgenomen in de NIS-configuratie; alleen gebruikers waarvan de UID hoger is dan 500 krijgen een NIS-account. U zult de systeemaccounts die op uw computer gedefinieerd zijn er dus niet terugvinden.

De tweede stap bestaat er uit dat de NIS-server **ypserv** geladen moet worden; ook hiervoor bestaat een script in /etc/init.d. Vervolgens moet de NIS-database gegenereerd worden. De configuratiebestanden die opgenomen moeten worden in de NIS-configuratie, moeten namelijk verwerkt worden tot een binaire database. Hiervoor gebruikt u het commando **ypinit -m**. Let er op dat dit commando meestal niet in het zoekpad staat; gebruik **locate** of **find** om de exacte locatie te achterhalen. De optie -m geeft aan dat hier de database van een NIS-master gegenereerd moet worden. Indien dit commando gegeven wordt op een Red Hat-Linux systeem, wordt de volgende output gegeven:

at this point, we have to construct a list of the hosts which will run NIS servers.
Laksmi.azlan.nl is in the list of NIS server hosts. Please continue to add the names for the other hosts, one per line. When you are done with the list, type a <control D>.
Next host to add: laksmi.azlan.nl
next host to add:

Het is hier de bedoeling dat u de namen van servers invoert die NIS-slave worden. Het gaat hier dus om de servers die u naast de NIS-master kunt benaderen om de benodigde informatie op te vragen. Deze servers moeten wel vanuit /etc/hosts geïdentificeerd kunnen worden. Als u dit gedaan hebt, wordt een Makefile verwerkt. Dit bestand komt standaard voor in de directory /var/yp. Hierdoor vindt de configuratie van de server plaats. In het Makefile komen een aantal instellingen voor, die soms handmatig aangepast moeten worden:

* **NOPUSH=** Het keyword “NOPUSH” bepaalt of de database die op de master-server wordt aangemaakt moet worden doorgestuurd naar de slave-servers. Als u geen gebruik maakt van slave-servers, heeft deze parameter de waarde “true”, anders moet u hier “false” neerzetten. Als gebruik gemaakt wordt van slave-servers, moeten de namen van de slave-servers ingevoerd worden in het bestand /var/yp/ypservers.
* **MINUID=, MINGID=** De parameters MINUID en MINGID geven aan wat de minimale ID is die aan gebruikers en groepen meegegeven wordt. Aangezien de ID’s beneden 500 regelmatig gebruikt worden voor gebruikersnamen die bij bepaalde services horen of op een andere manier gereserveerd zijn, is het gebruikelijk dat de minimale UID staat ingesteld op 500.
* **MERGE_PASSWD=, MERGE_GROUP=** Deze parameters bepalen of bij de samenstelling van de NIS-database wel of geen rekening gehouden moet worden met de shadow-bestanden. Het is gebruikelijk de informatie uit /etc/passwd en /etc/group samen te voegen in één NIS-map. Houdt er rekening mee dat dit niet bijzonder veilig is, maar er is nu eenmaal nooit volledige ondersteuning voor shadow-files in NIS geïmplementeerd.

De eigenlijke NIS-database wordt aangemaakt onder een subdirectory die dezelfde naam heeft als het NIS-domein. Afhankelijk van de bestanden die verwerkt zijn in de NIS-database, komt hier voor de meeste inputbestanden een tweetal indexbestanden voor. Zo bestaan er voor gebruikersinformatie de bestanden “passwd.byname.db” en “passwd.byuid.db”. Hierdoor kan de NIS-server op basis van een bepaalde index-sleutel snel informatie opvragen. In het ene bestand kan de server namelijk zoeken op een alfabetische lijst met gebruikersnamen, in het andere bestand wordt gezocht op UID.

De bestanden die als input voor de NIS-database gebruikt worden, staan gespecificeerd in het Makefile. Daarnaast wordt in het Makefile ook aangegeven welke NIS-mappen aangemaakt moeten worden. Als u bijvoorbeeld geen map met aliassen aan wilt maken, dient de bijbehorende regel uit het Makefile verwijderd te worden. Hieronder ziet u hoe een en ander geregeld wordt.

```
# These are the files from which the NIS databases are built. You
# may edit these to taste in the event that you wish to keep
# your NIS source files seperate from your NIS server’s actual
# configuration files.
#
GROUP        = $(YPPWDDIR)/group
PASSWD       = $(YPPWDDIR)/passwd
SHADOW       = $(YPPWDDIR)/shadow
GSHADOW      = $(YPPWDDIR)/gshadow
ADJUNCT      = $(YPPWDDIR)/passwd.adjunct
#ALIASES     = $(YPSRCDIR)/aliases
ALIASES      = /etc/aliases
ETHERS       = $(YPSRCDIR)/ethers
BOOTPARAMS= $(YPSRCDIR)/bootparams
HOSTS        = $(YPSRCDIR)/hosts
NETWORKS = $(YPSRCDIR)/networks
PROTOCOLS = $(YPSRCDIR)/protocols
PUBLICKEYS= $(YPSRCDIR)/publickey
RPC    = $(YPSRCDIR)/rpc
SERVICES     = $(YPSRCDIR)/services
```

NETGROUP = $(YPSRCDIR)/netgroup
NETID   = $(YPSRCDIR)/netid
AMD_HOME= $(YPSRCDIR)/amd.home
AUTO_MASTER=$(YPSRCDIR)/auto.master
AUTO_HOME = $(YPSRCDIR)/auto.home

YPSERVERS = $(YPDIR)/ypservers

Nadat de NIS-server geactiveerd is, moet deze nog geconfigureerd worden als client van zichzelf. De manier om dit te doen is eenvoudig, gebruik hiervoor het commando ypbind. Als alles goed gegaan is, toont het commando ypwhich nu als resultaat de naam van de server waaraan u als client verbonden bent.

De overige parameters die voorkomen in het NIS-Makefile worden gebruikt om te definiëren van welke compilers gebruik gemaakt moet worden en op welke locaties configuratiebestanden teruggevonden kunnen worden. In de meeste gevallen zal het niet nodig zijn hier veranderingen in aan te brengen.

Soms gaat er iets mis bij de uitvoering van het Makefile. Dit kan verschillende oorzaken hebben. Allereerst moet u ervoor zorgen dat de juiste producten geïnstalleerd zijn. Om het Makefile uit te voeren, hebt u natuurlijk het package “Make” nodig. Daarnaast wordt door het Makefile gebruik gemaakt van gawk; zorg er dus voor dat ook deze software geïnstalleerd is. Een laatste probleem dat voor kan komen, is dat u een foutmelding krijgt bij het genereren van de NIS-mappen. In veel gevallen worden deze foutmeldingen veroorzaakt doordat het Makefile probeert mappen te genereren op basis van bestanden die niet op uw systeem voorkomen, dit zijn geen kritische fouten. U hoeft zich dus niets van deze foutmeldingen aan te trekken.

***welkefiles10 U kunt foutmeldingen bij het genereren van de NIS-database voorkomen door nauwkeurig aan te geven welke bestanden in NIS-mappen geplaatst moeten worden.

Nadat door gebruik van het commando **ypinit** de NIS-database succesvol is aangemaakt, kan de NIS-server gestart worden. Hiervoor wordt het commando **ypserv** gebruikt, gebruik bij voorkeur het script dat voor dit doel is aangemaakt in de directory /etc/init.d. Om vervolgens te controleren of de NIS-server ook daadwerkelijk geactiveerd is, kan het commando **rpcinfo -u localhost ypserv** op de server geven worden. Het resultaat van dit commando kan er als volgt uit zien:

program 100004 version 1 ready and waiting
program 100004 version 2 ready and waiting

Nadat u een NIS-server voor gebruik op het netwerk geconfigureerd hebt, kunt u overgaan tot de configuratie van de client.

Kort samengevat moeten de volgende stappen uitgevoerd worden om een NIS server te activeren:

1. **domainname naam** om het lokale NIS domein een naam te geven;
2. **ypserv** om de NIS-service te starten;

3. **ypinit –m** om de NIS-database op de NIS master te genereren;;
4. **rpcinfo –u localhost ypserv** om te controleren of de NIS-server inderdaad gestart is;
5. **ypbind** om de NIS server te configureren als client van zichzelf;
6. **ypwhich** om te controleren dat de NIS server inderdaad client is van zichzelf.

***yastnis Op SUSE-Linux kunt u gebruikmaken van YaST om een NIS-server te installeren en configureren.

Als u ook staat te popelen om een werkstation als client voor een NIS-server te configureren, voer dan de volgende procedure uit:
1. **domainname nisdomainnaam** om aan te duiden in welk NIS-domein de client voor moet komen.
2. **ypbind –broadcast** om de client zelf te laten zoeken naar een NIS-master voor het domein waarvan hij lid is. Als alternatief kunt u de naam van een NIS-server opnemen in het bestand /etc/yp.conf.
3. **ypwhich** om te controleren of de NIS-client inderdaad als client aan een domain is toegevoegd. Later in dit hoofdstuk vindt u meer informatie over de configuratie van een NIS-client.

5.3.4 Structuur van de database

Nadat NIS op de NIS-master geïnstalleerd is, zijn op deze computer de NIS-bestanden (mappen) aangemaakt. Deze NIS-bestanden worden door clients geraadpleegd om gegevens op te vragen. Daarnaast wordt een aantal configuratiebestanden aangemaakt. Deze configuratie wordt geplaatst in de directory /var/yp.

***nisdb10 De NIS-databasebestanden worden standaard weggeschreven in de directory /var/yp.

Het bestand ypservers in de NIS-database is een ASCII-bestand waarin de namen van alle NIS-servers voorkomen. Het is belangrijk dat dus zowel de naam van de NIS-master, als de namen van de NIS-slaves voorkomen. Normaliter wordt dit bestand automatisch aangemaakt als de NIS-database gegenereerd wordt.

5.3.5 Beveiliging

NIS is nooit ontworpen als een bijzonder veilige netwerkservice. Houdt daar altijd rekening mee. Toch is er een aantal maatregelen dat u kunt treffen om de onveiligheid van NIS te verminderen:
* Geef in het bestand securenets aan wie toegang heeft tot een NIS-server
* Gebruik bij het starten van de NIS-server de optie –p om hem op een niet-standaard poort te starten.
* Gebruik het bestand nicknames om te verwijzen naar de verschillende NIS-mappen die op de server gebruikt kunnen worden en een naam te koppelen aan die mappen.

Het bestand “securenets” wordt gebruikt om aan te geven wie toegang heeft tot de NIS-server. De inhoud bestaat uit paren waarin eerst het subnetmasker en vervolgens het netwerkadres genoemd worden van computers die toegang hebben tot de NIS-server. Direct na aanmaken heeft dit bestand, naast wat commentaar, de volgende inhoud:

255.0.0.0 127.0.0.0
0.0.0.0 0.0.0.0

Hiermee wordt toegang verleend aan elke computer die gegevens uit de database wil halen. Om ervoor te zorgen dat alleen computers die voorkomen op het netwerk 192.168.0.0 toegang hebben, verandert u de laatste regel in 255.255.255.0 192.168.0.0. Let even op: u geeft dus op een regel eerst het subnetmasker en vervolgens het netwerkadres dat gebruikt moet worden.

Een andere manier om enige beveiliging aan de NIS-server te verbinden, is door het commando **ypserv** te geven met de optie **–p**. Deze optie maakt het namelijk mogelijk als argument een poort te geven waarvan de NIS server gebruik maakt; hierdoor wordt het mogelijk toegang tot de NIS-server op een router tegen te houden door toegang tot de betreffende port uit te filteren middels een packet filter. Zo maakt u toegang vanaf andere netwerken sowieso onmogelijk.

In het bestand “nicknames” wordt aangegeven onder welke namen bepaalde NIS-mappen ook bereikbaar zijn. Dit zorgt ervoor dat op een eenvoudiger manier verwezen kan worden naar een NIS-map. Dit bestand kan bijvoorbeeld de volgende inhoud hebben:

passwd	passwd.byname
group	group.byname
networks	networks.byaddr
hosts	hosts.byname
protocols	protocols.bynumber
services	services.byname
aliases	mail.aliases
ethers	ethers.byname

Door dit bestand weet de NIS-server dat als verwezen wordt naar “passwd”, eigenlijk informatie opgevraagd moet worden uit de NIS-map passwd.byname.

<<KADER>>
Waarschuwing! NIS is een leuke manier om bijvoorbeeld een centrale login-server te definiëren op uw netwerk. Houd echter rekening met een belangrijke onvolkomenheid: NIS kan niet overweg met /etc/shadow. U moet er dus bij het genereren van de NIS-mappen rekening mee houden dat wachtwoorden vanuit de shadow-files niet in de NIS-mappen terecht komen.
<<EINDE KADER>>

5.3.6 Configuratie van de NIS-slave

Als de NIS-master bereikbaar is, met het commando **ypinit -s naamvandemaster** een slave-server geconfigureerd worden. Er wordt dan een kopie van de database van de master op de slave geplaatst. Om ervoor te zorgen dat gewijzigde gegevens van de NIS-master gepropageerd worden naar alle NIS-slaves, moet op de master het commando **yppush** gebruikt worden. Dit commando zorgt ervoor dat wijzigingen doorgestuurd worden naar alle slave-servers in het NIS-domein. Het is gebruikelijk dat dit commando automatisch wordt uitgevoerd door middel van het NIS-makefile /var/yp/Makefile. U dient er dan echter wel voor te zorgen dat de regel “NOPUSH=True", die standaard aan staat, door middel van een commentaar-teken gedeactiveerd wordt. Houd er rekening mee dat er geen mechanisme is op de master dat ervoor zorgt dat wijzigingen automatisch doorgestuurd worden. Om ervoor te zorgen dat wijzigingen toch periodiek worden doorgegeven, is het aan te raden om **yppush** door middel van Cron automatisch op gezette tijden uit te voeren.

Naast de mogelijkheid met behulp van **yppush** wijzigingen vanaf de master te propageren naar alle NIS-slaves, is het ook mogelijk om vanaf een NIS-slave wijzigingen vanaf de master-server binnen te halen. Hiervoor wordt het commando **ypxfr** gebruikt. U kunt deze opdracht alleen gebruiken nadat de database met **yppush** al een keer naar de slaves gepropageerd is. Yppush kopieert namelijk de hele database over het netwerk terwijl ypxfr alleen wijzigingen synchroniseert.

5.3.7 Configuratie van NIS clients

Als de NIS-server geactiveerd is, kunt u overgaan tot de configuratie van de client. De belangrijkste taak die moet gebeuren vanaf de client, is dat gespecificeerd moet worden hoe de client een NIS-server terugvindt. Hiervoor bestaand twee mogelijkheden; de client kan proberen de NIS-server te vinden door middel van een broadcast en de client kan gebruik maken van het configuratiebestand /etc/yp.conf waarin staat aangegeven welke NIS-server benaderd moet worden. In dit bestand kunt u aangeven op welke wijze de NIS-server teruggevonden kan worden. Het is voldoende om hier een regel in op te nemen waardoor de naam van de NIS-server gegeven wordt, bijvoorbeeld:

ypserver xtina

identificeert de NIS-server voor dit domein als xtina. Uiteraard moet het IP-adres van deze server teruggevonden kunnen worden door middel van /etc/hosts of DNS.

Als het NIS-configuratiebestand is aangemaakt, kan de NIS-clientsoftware gestart worden. Dit doet u door middel van het commando **ypbind**. Als u geen foutmelding krijgt, is **ypbind** naar alle waarschijnlijkheid succesvol gestart. U kunt dit met twee opdrachten verifiëren. Het commando **rpcinfo -p localhost** toont alle rpc-services die op de locale computer actief zijn, het commando **rcpinfo -u ypbind** geeft informatie over de versie ypbind die gebruikt wordt. Deze versie moet natuurlijk overeenkomen met de versie ypserv op de server. Hieronder ziet u een mogelijk resultaat dat door deze commando's gegeven wordt:

```
# rpcinfo -p localhost
program   vers   proto   port
100000    2      tcp     111    portmapper
100000    2      udp     111    portmapper
100029    1      udp     734    keyserv
100029    2      udp     734    keyserv
100007    2      udp     749    ypbind
100007    1      udp     749    ypbind
100007    2      tcp     752    ypbind
100007    1      udp     752    ypbind
100005    1      udp     759    mountd
100005    2      udp     759    mountd
100005    1      tcp     762    mountd
100005    2      tcp     762    mountd
100003    2      udp     2049   nfs
100003    2      tcp     2049   nfs
# rpcinfo -u localhost ypbind
program 100007 version 1 ready and waiting
program 100007 version 2 ready and waiting
```

Als u de bovenstaande resultaten te zien krijgt, is ook de client juist geconfigureerd. U kunt dit controleren met het commando **ypcat**; hiermee kan de inhoud van een van de NIS-mappen opgevraagd worden. Zo moet het commando **ypcat hosts** de inhoud van het hosts-bestand op de server laten zien.

***ypcat Op de NIS-client kunt u met behulp van de opdracht **ypcat** de inhoud van een NIS-map bekijken.

Bepalen van de zoekvolgorde op de client
Op een Linux-systeem bestaan meerdere mogelijkheden om informatie te verkrijgen over diverse gegevens, zoal gebruikersnamen en IP-adressen die bij computernamen horen. Voor het laatste wordt meestal gebruik gemaakt van een van de volgende systemen:

DNS
NIS
Configuratiebestanden

De volgorde waarin gezocht wordt op een systeem, zal per systeem bepaald moeten worden. Om te bepalen hoe gezocht moet worden op uw computer, maakt u gebruik van het configuratiebestand /etc/nsswitch.conf Ook kan gebruik gemaakt worden van de Pluggable Authentication Modules (PAM), maar dat is meer bedoeld voor validatie van gebruikersgegevens.

Het bestand /etc/nsswitch.conf biedt uitgebreide mogelijkheden om te configureren in welke volgorde verschillende databronnen doorzocht moeten worden. In nsswitch.conf wordt per datasoort aangegeven wat de zoekvolgorde is. De volgende datasoorten kunnen beschreven worden:

aliases
ethers
group
hosts
netgroup
network
passwd
protocols
publickey
rpc
services
shadow

Hieronder ziet u een vrij algemeen voorbeeld van een configuratie van nsswitch.conf

passwd: compat
group: compat
shadow: compat

hosts: dns [!UNAVAIL=return] files
networks: nis [NOTFOUND=return] files

ethers: nis [NOTFOUND=return] files
protocols: nis [NOTFOUND=return] files
rpc: nis [NOTFOUND=return] files
services: nis [NOTFOUND=return] files

De eerste kolom is de datasoort waarnaar gezocht moet worden, op de rest van de regel wordt gespecificeerd hoe gezocht moet worden. De services waarin gezocht kan worden zijn “files” (de ASCII-configuratiebestanden), “nis”, “nisplus”, “dns” (alleen voor hosts) en “compat” (alleen voor passwd en group). Het is overigens ook mogelijk hier zaken aan toe te voegen, denk bijvoorbeeld aan de aanduiding ldap die gebruikt kan worden om te zoeken in de LDAP-server die geconfigureerd is.

Er bestaat ook een mogelijkheid om meer controle te geven over het lookup-proces. Dit gebeurt door het specificeren van bepaalde acties. Hiervoor bestaan de volgende mogelijkheden:

STATUS => success | notfound | unavail | tryagain
ACTION => return | continue

De bedoeling hiervan is dat bij het optreden van een bepaalde status een zekere actie uitgevoerd kan worden, bijvoorbeeld direct te stoppen wanneer een belangrijk bestand niet aanwezig is.

De volgende statussen zijn beschikbaar:

* **success** Er is geen fout opgetreden en de gewenste entry is gevonden. De standaardactie die hierbij hoort is “return”; met andere woorden er wordt niet verder gezocht.
* **notfound** Het lookup-proces werkt zoals het hoort, maar de waarde is niet gevonden. De actie die hierbij hoort is “continue”, wat betekent dat verder gezocht moet worden door gebruik te maken van de volgende gedefinieerde zoekmethode.
* **unavail** De betreffende service is niet beschikbaar. Dit kan betekenen dat een configuratiebestand niet bestaat, maar ook dat een service niet actief is. De standaardwaarde die hier bij hoort is “continue”.
* **tryagain** De service is tijdelijk niet beschikbaar, bijvoorbeeld omdat het maximum aantal connecties op een server bereikt is. De standaard actie die daarbij hoort is “continue”.

5.3.8 Toevoegen van gegevens aan de NIS-database

Om gegevens toe te voegen aan de NIS-database, kunnen gewoon de inputbestanden op de master-server bewerkt worden. Dit betekent dat gebruikers nog steeds aangemaakt kunnen worden door bijvoorbeeld het commando useradd te gebruiken om /etc/passwd te bewerken. Hetzelfde geldt voor de andere configuratiebestanden. Er zouden echter problemen kunnen ontstaan als in het NIS-makefile verwezen wordt naar andere sourcefiles. Verzeker u er dus eerst van dat inderdaad de beoogde bestanden gebruikt worden om wijzigingen aan te brengen. Zoek hiervoor in het makefile naar de tekst waarmee de sourcedirectory’s en sourcefiles gedefinieerd worden. Vaak zal dit gebeuren door middel van regels als

YPSRCDIR = /etc
YPPWDDIR = /etc

GROUP = $(YPPWDDIR)/group

PASSWD = $(YPPWDDIR)/usr

Als op uw systeem naar sourcefiles op een andere locatie verwezen wordt, moeten de bestanden op die locatie aangepast worden.

Als dan de benodigde wijzigingen in de sourcefiles zijn aangebracht, moet de NIS-database opnieuw gegenereerd worden. Hiervoor wordt in de directory /var/yp het commando **make** gegeven. Dit commando zorgt ervoor dat de inhoud van de sourcefiles wordt overgebracht naar de NIS-database. Als u met NIS wilt gaan werken, is het handig een en ander te automatiseren door een speciaal script aan te maken op uw server waarmee u nieuwe gebruikers aan het systeem toevoegt.

***nieuweuser Voordat nieuwe gegevens in de NIS-mappen verwerkt kunnen worden, moet u de opdracht **make** geven in de directory /var/yp om de NIS-mappen opnieuw te genereren.

Naast deze manier om wijzigingen in de NIS-databases door te voeren, kan het ook op een andere manier. Er is namelijk een aantal YP-opdrachten beschikbaar waarmee u direct in de NIS-database kunt schrijven. Zo kunnen gebruikers bijvoorbeeld hun wachtwoord wijzigen door in plaats van het commando **passwd** de opdracht **yppasswd** te gebruiken.

Oefening 5.3
Om deze opdracht uit te kunnen voeren, hebt u minimaal drie computers nodig. Configureer één van deze computers als NIS-master. Let op: alleen gebruikersaccounts met een UID groter dan 1000 mogen in de NIS-mappen opgenomen worden. Specificeer tijdens het genereren van de mappen op de NIS-master meteen welke NIS-slave er gebruikt moet worden. Zorg er na het aanmaken van de NIS master voor dat alle mappen op de NIS master naar deze slave gesynchroniseerd worden. Maak veervolgens op de slave zelf een cron-job waarmee u ervoor zorgt dat nieuwe wijzigingen op de NIS-master elk kwartier worden binnengehaald. Controleer dat deze informatie werkt, bijvoorbeeld door gebruik te maken van de opdracht **ypcat**. Maak nu vanaf de NIS-master een nieuwe gebruiker aan met de naam pleunie. Tot slot configureert u de client-computer. Hierop regelt u met behulp van nsswitch dat ingelogd wordt op de NIS-database en niet langer op de lokale gebruikersdatabase. Kijk vervolgens of u er ook voor kunt zorgen dat door middel van PAM eerst op de NIS-database gekeken wordt en alleen als dat niet lukt op de lokale gebruikersdatabase.

5.4 LDAP

Er zijn nogal wat manieren waarop u er voor kunt zorgen dat meerdere Linux-systemen centraal beheerd worden. Wanneer het gaat om het centrale beheer van gebruikers zijn NIS en het Lightweight Directory Access Protocol (LDAP) de belangrijkste mogelijkheden. Vooral de populariteit van LDAP is de afgelopen jaren sterk toegenomen. Wanneer u tegenwoordig bijvoorbeeld een installatie van SUSE Linux Enterprise Server uitvoert, wordt er zelfs al standaard een LDAP-server geïnstalleerd en geconfigureerd. In dit hoofdstuk leest u hoe u zelf op basis van OpenLDA{P (www.openldap.org) zo'n server in het leven kunt roepen en kunt beheren.

5.4.1 Introductie

Net als op andere platformen, heeft het Lightweight Directory Access Protocol (LDAP) ook op Linux twee hoofdtoepassingen. De eerste toepassing is de algemene toepassing als adresboek. Daarbij kunt u bijvoorbeeld denken aan de gebruiker die vanaf internet de LDAP-

server benadert om daar het e-mailadres en andere gegevens van een gebruiker vanaf te halen. In die hoedanigheid gedraagt LDAP zich eigenlijk als een willekeurige toepassing die op Linux draait. Daarnaast kan LDAP gebruikt worden als service waarmee gebruikers gevalideerd worden. Dit betekent dat in de LDAP-Directory alle gegevens moeten voorkomen die de gebruiker nodig heeft om gebruik te maken van resources op het Linux-werkstation. Vooral in deze hoedanigheid neemt het gebruik van LDAP grote vlucht. Het voordeel van het gebruik van een centrale Directory-server is duidelijk; gebruikers kunnen op een centrale plaats in het netwerk beheerd worden in plaats van op elke afzonderlijke node in het netwerk. Tegelijkertijd kan het onveilige NIS uitgefaseerd worden
Om in te kunnen loggen op een LDAP-server, moet tijdens de loginprocedure gebruikgemaakt worden van de algemene module genaamd pam_ldap. Deze module kan geïntegreerd worden in het mechanisme van Pluggable Authentication Modules (PAM) dat door verschillende programma's op Linux gebruikt wordt om gebruikers te valideren.

<<KADER>>
Directory en directory Om duidelijk te maken waar we het over hebben, wordt in dit hoofdstuk een onderscheid gemaakt tussen de Directory met een hoofdletter D en een directory met een kleine letter d. Wanneer we het hebben over een directory, dan wordt daarmee verwezen naar een map op het bestandssysteem waarin bestanden worden opgeslagen. Hebben we het daarentegen over een Directory, dan verwijzen we daarmee naar een database die gebruikt wordt om gegevens over het netwerk in te bewaren. LDAP maakt dus gebruik van een Directory en deze Directory is ergens op het bestandssysteem opgeslagen in een directory.
<<EINDE KADER>>

5.4.2 Terminologie

LDAP is het Lightweight Directory Access Protocol dat toegang biedt tot de Directory. Deze Directory is een hiërarchisch opgebouwde database waarin gegevens van zeer verschillende aard bijgehouden kunnen worden. De meest gebruikte gegevens zijn waarschijnlijk de combinaties van gebruikersnamen en e-mailadressen, maar er kan ook heel andere informatie in voorkomen, zoals bijvoorbeeld gegevens over de printers die op het netwerk voorkomen. De hiërarchie van de Directory wordt gevormd met behulp van verschillende containerobjecten. Dit worden Directory Components (DC) genoemd. U kunt deze DC's vergelijken met soortgelijke objecten binnen de DNS-hiërarchie; denk daarbij aan namen zoals www.sandervanvugt.nl. Ook zijn deze directory's qua functionaliteit te vergelijken met directory's die op een bestandssysteem gebruikt worden om bestanden op een logische wijze op te slaan.

In de Directory komen entry's voor. Deze entry's worden ook wel objecten of classes genoemd. Zo bestaat er bijvoorbeeld voor elke gebruiker waarvan de gegevens in de Directory zijn opgenomen een gebruikersobject. Deze objecten zijn de bouwstenen waaruit de Directory bestaat. Elk van deze classes heeft zijn eigen unieke naam, die Distinguished Name (DN) genoemd wordt. Deze Distinguished Name bestaat uit de objectnaam (Common Name = CN) van het object met daaraan toegevoegd de namen van de containers waarin dit object voorkomt. Zo is bijvoorbeeld de DN van gebruikster LindaV die bij Azlan werkt cn=LindaV,dc=azlan,dc=com. Alle objecten hebben attributen. De attributen zijn de stukjes informatie die met het object verbonden zijn. Denk daarbij aan een gebruikersnaam, een e-mailadres en een wachtwoord.
Om te kunnen valideren in de LDAP-database, is het belangrijk dat alle Linux-attributen die daarvoor nodig zijn een waarde hebben. Zo kan bijvoorbeeld niet ingelogd worden op een

Linux-systeem als het attribuut uidnumber geen waarde heeft. Welke attributen precies door welk object gebruikt kunnen en moeten worden, wordt gedefinieerd in het LDAP-schema. Dit gebeurt door middel van een definitie van zogeheten objectclasses. Op Linux wordt het schema gedefinieerd in verschillende bestanden. U vindt deze bestanden in de directory /etc/openldap/schema. Even opletten: een standaard LDAP-installatie maakt slechts van een eenvoudig schema gebruik. Wilt u ondersteuning voor meer dan alleen de basis? Dan moet u ervoor zorgen dat het standaard schema uitgebreid wordt door de hiervoor noodzakelijke modules te laden.

***schema Het LDAP-schema bestaat uit verschillende bestanden die allemaal aangeroepen moeten worden vanuit het configuratiebestand /etc/openldap/slapd.conf

Het algemene formaat dat door LDAP gebruikt wordt om te werken met gegevens, is het LDAP Data Interchange Format (LDIF). Dit ASCII-formaat wordt bijvoorbeeld gebruikt om gegevens toe te voegen aan de Directory. Daarbij is het belangrijk dat voor elke class in elk geval de verplichte attributen gespecificeerd worden. Als dat niet gebeurt, kunnen er vervelende foutmeldingen volgen en wordt het object niet gemaakt.

5.4.3 OpenLDAP

De LDAP-implementatie die vooral op het Linux-platform gebruikt wordt, is OpenLDAP (http://www.openldap.org). Na installatie van dit product, dat op de meeste Linux-systemen deel uitmaakt van een custom-installatie, wordt op uw systeem een aantal configuratiebestanden, hulpprogramma's en programmabestanden weggezet. Voordat we ingaan op de precieze configuratie, zullen we hier eerst een overzicht geven van de verschillende componenten die met OpenLDAP geïnstalleerd worden.

Het belangrijkste programmabestand dat deel uitmaakt van de OpenLDAP-software is de daemon slapd. Dit is de stand-alone LDAP-daemon. Dit proces moet geactiveerd worden om over LDAP-functionaliteit te kunnen beschikken; slapd is dus de LDAP-server. Als u meer dan één LDAP-server binnen een netwerk gebruikt, kunt u daarnaast ook gebruikmaken van slurpd. Als er meerdere servers zijn die eenzelfde database bedienen, moeten de gegevens immers up-to-date gehouden worden tussen de verschillende kopieën van de Directory die op deze servers voorkomen. Slurpd is het proces dat er voor zorgt dat replicatie van gegevens plaatsvindt. Hiertoe verstuurt slurpd de gegevens van de master-server naar alle aanwezige slave-servers.

De configuratie van LDAP vindt plaats doordat een aantal configuratiebestanden wordt bewerkt. Deze configuratiebestanden komen doorgaans voor in de directory /etc/openldap, maar let op: sommige distributies plaatsen ook een exemplaar van de betreffende configuratiebestanden in een andere directory, zoals bijvoorbeeld direct onder /etc. Zorg er altijd voor dat u zeker weet dat u het juiste bestand bewerkt voordat u er mee aan het werk gaat.

Het belangrijkste configuratiebestand is slapd.conf. Hierin vindt vrijwel de gehele configuratie van de LDAP-daemon slapd plaats. Dit bestand moet in elk geval aangepast worden voordat u het proces slapd opstart. Naast slapd.conf bestaat er een aantal bestanden waarin het schema van de LDAP-Directory gedefinieerd is. Dit schema bepaalt welke classes waar in de LDAP-tree kunnen voorkomen en welke attributen met deze classes verbonden zijn. Het schema bestaat uit een aantal bestanden dat voorkomt in de directory /etc/openldap/schema.

Als laatste is er een aantal opdrachten waarmee gegevens aan de database toegevoegd, eruit verwijderd en erin bewerkt kunnen worden. Het betreft hier **ldapadd** waarmee gegevens toegevoegd worden, **ldapmodify** om gegevens te wijzigen, **ldapdelete** om gegevens te verwijderen en **ldapsearch** om te zoeken naar gegevens. Daarnaast is er **ldif2ldbm**, waarmee gegevens vanuit het algemene LDIF-formaat vertaald kunnen worden in het door LDAP gerbuikte LDBM databaseformaat.

Daarnaast komen er meestal nog twee modules voor die door de clientsoftware gebruikt kunnen worden. U mag er echter niet van uitgaan dat deze modules daadwerkelijk voorkomen; ze maken namelijk deel uit van andere software-packages. Als eerste is er de LDAP-module die door de nameservice switch nsswithc.conf gebruikt kan worden; deze heeft de naam nss_ldap. U gebruikt hem in de algemene procedure waarmee ingesteld wordt welke configuratiebestanden gebruikt moeten worden om informatie over bijvoorbeeld gebruikers, groepen, servers enzovoort te achterhalen. U regelt de configuratie hiervan met behulp van het instellingenbestand /etc/nsswitch.conf.

Een andere belangrijke module die door de clients gebruikt kan worden is pam_ldap. Dit is de module die er voor zorgt dat het standaardmechanisme van Pluggable Authentication Modules (PAM) dat op Linux gebruikt wordt, ook gebruik kan maken van LDAP om gebruikers te laten inloggen op basis van gegevens in de LDAP-database.

5.4.4 Overzicht van de configuratie

Om een LDAP-server te configureren moeten in elk geval de volgende stappen doorlopen worden:

1. Zorg ervoor dat de Open LDAP software geïnstalleerd is. Haal deze eventueel op van http://www.openldap.org. Gebruik bij voorkeur de RPM's die bij uw distributie geleverd worden.
2. Configureer het bestand slapd.conf.
3. Start slapd.
4. Voeg gegevens aan de database toe door met **ldapadd** gegevens uit een LDIF-bestand te importeren.
5. Kijk met **ldapsearch** of dit goed gelukt is.

Door deze vijf stappen uit te voeren, zorgt u voor een basis installatie van LDAP. U kunt nu met een LDAP-client gegevens uit uw LDAP-directory halen. Als u echter in staat wilt zijn meer geavanceerde zaken te doen, zoals bijvoorbeeld inloggen op de LDAP-database, dan moeten er nog een paar extra taken uitgevoerd worden.

Stap 1. Installatie van de software

In veel gevallen kan de LDAP-server tijdens de installatie van de distributie gekopieerd worden; op de meeste moderne distributies zijn hier niet eens extra opties voor nodig. Als dit niet het geval is, kunt u de meest recente software zelf ophalen en uitpakken. We gaan er in dit hoofdstuk van uit dat u voor de softwareinstallatie gebruikmaakt van de RPM-bestanden die met uw distributie geleverd zijn.

Stap 2. Configuratie van de server: slapd.conf

Na installatie van de software wordt een standaardvoorbeeldbestand /etc/openldap/slapd.conf gemaakt. Dit moet bewerkt worden, zodat het aan de behoeften van uw organisatie voldoet.

We bespreken nu een voorbeeld van dit bestand. In een eenvoudige vorm kan dit bestand er als volgt uitzien:

```
include/etc/openldap/schema/core.schema
include/etc/openldap/schema/cosine.schema
include/etc/openldap/schema/inetorgperson.schema
include/etc/openldap/schema/rfc2307bis.schema
include/etc/openldap/schema/yast.schema

pidfile /usr/local/var/slapd.pid
argsfile        /usr/local/var/slapd.args

# load dynamic backend modules:
modulepath    /usr/lib/openldap/modules

access to dn.base=""
        by * read

access to dn.base="cn=Subschema"
        by * read

access to attr=userPassword,userPKCS12
        by self write
        by * auth

access to attr=shadowLastChange
        by self write
        by * read

access to *
        by * read

loglevel 0
TLSCertificateFile /etc/ssl/servercerts/servercert.pem
TLSCACertificatePath /etc/ssl/certs/
TLSCertificateKeyFile /etc/ssl/servercerts/serverkey.pem

database bdb
suffix "dc=sandervanvugt, dc=nl"
rootdn "cn=Administrator,dc=sandervanvugt, dc=nl"
rootpw "{ssha}sBnN0CZ1MWsg0720IKLzP0B9SmVaTFZCVg=="
directory /var/lib/ldap
checkpoint 1024 5
cachesize 10000
index objectClass,uidNumber,gidNumber eq
index member,mail eq,pres
index cn,displayname,uid,sn,givenname sub,eq,pres
```

We zullen nu bespreken wat er in het voorgaande bestand gebeurt.

include/etc/openldap/schema/core.schema
include/etc/openldap/schema/cosine.schema
include/etc/openldap/schema/inetorgperson.schema
include/etc/openldap/schema/rfc2307bis.schema
include/etc/openldap/schema/yast.schema

In de eerste regels van dit bestand wordt een aantal extra configuratiebestanden aangeroepen. Deze bestanden vormen met elkaar het LDAP-schema. Hierin wordt aangegeven welke objecten in de LDAP-Directory kunnen worden gemaakt, waar in de tree (structuur) deze objecten mogen voorkomen en welke attributen met deze objecten verbonden kunnen worden. LDAP maakt gebruik van een schema dat uitbreidbaar is. Elke LDAP-server heeft om te beginnen een basisschema. In dit geval wordt het basisschema gedefinieerd in het bestand core.schema. In dit basisschema bestaan alle objecten die altijd sowieso moeten voorkomen. Dit basisschema wordt in deze configuratie uitgebreid met een aantal extra schemamodules. Als u zelf schemamodules hebt die aan de jusite syntaxis voldoen, is het geen probleem deze toe te voegen aan het bestaande schema dat u gebruikt.

Vervolgens zijn er twee regels waarin het beheer van de LDAP-server wordt bijgehouden. Om te beginnen gebeurt dat in het bestand slapd.pid, waarin het procesidentificatienummer van de server bewaard wordt. Daarnaast is er slapd.args, waarin de argumenten bewaard worden waarmee de LDAP-server gestart is.

In het volgende deel van het configuratiebestand kan een aantal LDAP-modules geladen worden. U ziet hier alleen de definitie van het pad waarin deze modules horen voor te komen; deze regel wordt doorgaans gevolgd door een aantal regels waarop de namen staan van de modules die geladen moeten worden. Met behulp van deze modules wordt ook weer de functionaliteit van de LDAP-server uitgebreid. Om een willekeurig aantal modules toe te voegen, kan bijvoorbeeld gebruikgemaakt worden van de volgende regels:

```
modulepath    /usr/lib/openldap/modules
moduleload    back_ldap.la
moduleload    back_meta.la
moduleload    back_monitor.la
moduleload    back_perl.la
```

Nadat de werking van additionele LDAP-modules ingesteld is, wordt aangegeven wie rechten heeft voor welke delen van de LDAP-Directory. U ziet dat op de grootste delen van de LDAP-server leesmachtigingen gegeven worden aan iedereen. Voor sommige belangrijke attributen zoals het wachtwoord van gebruikers hebben gebruikers zelf rechten om te schrijven. Dit is handig; zo is namelijk een gebruiker in staat zijn eigen wachtwoord te wijzigen.

Als laatste gedeelte van het algemene configuratiebestand volgt een aantal algemene parameters dat bepaalt hoe de LDAP-server zich moet gedragen. Om te beginnen moet ervoor gezorgd worden dat een beveiligde verbinding mogelijk is. Hiervoor zijn de drie regels waarin wordt aangegeven waar de TLS-certificaten teruggevonden kunnen worden. Het gebruik van deze certificaten is zeer belangrijk, u hebt ze namelijk nodig om een SSL-verbinding met de LDAP-server tot stand te kunnen brengen. Verderop in dit hoofdstuk leest u meer over de wijze waarop de LDAP-server met certificaten beveiligd kan worden. Vervolgens wordt

aangegeven van welk type database gebruikgemaakt wordt. Hiervoor zijn verschillende mogelijkheden; in dit geval is gekozen voor het type bdb.

Dan komt er een aantal instellingen waarmee meer geregeld wordt voor de database zelf. Als eerste is dat de standaardsuffix, die hier staat ingesteld op sandervanvugt,nl. Deze suffix bepaalt op welke plaats in de database nieuwe objecten standaard worden neergezet. Vervolgens wordt gedefinieerd wie de beheerder is van de database. Houd er rekening mee dat deze beheerder oppermachtig is en toegang krijgt tot werkelijk alle aspecten van de database.

Er zijn verschillende manieren om het wachtwoord van de beheerder in te voeren. De meest eenvoudige manier is om het gewoon in ASCII-tekst in het configuratiebestand slapd.conf te plaatsen. Dit heeft echter nadelen voor de beveiliging van uw server. Om ervoor te zorgen dat het wachtwoord in een onleesbaar formaat wordt opgeslagen, maakt u gebruik van de opdracht slappasswd. Deze opdracht wordt speciaal gebruikt voor het invoeren van het wachtwoord van de beheerder.

Vervolgens wordt met directory /var/lib/ldap bepaald in welke directory op het lokale bestandssysteem de LDAP-database moet worden weggeschreven. Tot slot is er een aantal regels dat de prestaties van de LDAP-database beïnvloedt. De parameter cachesize bepaalt hoeveel objecten in de database cache bewaard kunnen worden en de drie indexregels tot slot zorgen er voor dat een aantal indexen automatisch wordt aangemaakt. Het gebruik van deze indexen zorgt er voor dat in de database sneller gezocht kan worden naar objecten en attributen die vaak opgevraagd worden.

Stap 3. start slapd
Als slapd.conf eenmaal op de juiste wijze is gemaakt, moet de server gestart worden. Hiervoor kunt u natuurlijk handmatig het programmabestand **/usr/sbin/slapd** activeren. Als u het op deze manier doet, kan het vooral in de testfase interessant zijn de optie -dx mee te geven bij het starten van deze server. Wanneer u in plaats van de x in deze optie een getal opgeeft, bepaalt u hiermee het debug-level waarmee de server gestart wordt. Hoe hoger dit getal is, hoe meer details u terugvindt in de logboekbestanden van uw server. Naast de mogelijkheid de LDAP-server handmatig te starten, kunt u als alternatief gebruikmaken van het script dat meegeleverd wordt om de server automatisch te starten in de System-V-opstartroutine van uw server. Zo start u bijvoorbeeld op SUSE Linux de LDAP server met de opdracht /etc/init.d/ldap start.

Stap 4. Voeg gegevens toe aan de directory
U bent nu klaar om gegevens toe te voegen aan de directory. De meest gebruikelijke manier om dit te doen, is door middel van LDIF. U maakt een bestand waarin gegevens in het LDIF-formaat voorkomen en de inhoud van dit bestand voegt u vervolgens met een opdracht als ldapadd toe aan de Directory. Dit bestand kan bijvoorbeeld de volgende eenvoudige inhoud hebben:

```
dn: dc=azlan, dc=com
dc: azlan
o: azlan
objectclass: organization
objectclass: dcObject
```

```
dn: cn=Manager, dc=azlan, dc=com
cn: Manager
sn: Manager
objectclass: person

dn: cn=Linda, dc=azlan, dc=com
objectClass: person
cn: Linda
sn: Verkaik
userPassword: geheim

dn: cn=LindaT, dc=azlan, dc=com
objectClass: person
cn: Linda
sn: Thomassen
userPassword: geheim

dn: cn=Marije, dc=azlan, dc=com
objectClass: person
cn: Marije
sn: Ronteltap
userPassword: geheim
```

Containers eerst! Als u met behulp van een LDIF-bestand gegevens wilt toevoegen aan een LDAP-directory, is het belangrijk er op te letten dat u dit in de goede volgorde doet. Als u bijvoorbeeld een gebruiker cn=LindaT, dc=azlan, dc=com toe wilt voegen, lukt dat alleen als u er eerst voor gezorgd hebt dat de container dc=azlan, dc=com bestaat. In het voorgaande voorbeeld ziet u dat prima aan deze voorwaarde wordt voldaan.

Als dit bestand de naam users.ldif heeft, kunt u het toevoegen aan de Directory met de opdracht ldapadd –x –D “cn=Manager, dc=azlan, dc=com” –W < users.ldif. Als het goed is, krijgt u als antwoord op deze opdracht Enter LDAP Password: te zien. Geef hier het wachtwoord van de account op die u als manager gespecificeerd hebt. De naam en het wachtwoord van deze manager komen overigens overeen met de naam en wachtwoord die u in slapd.conf hebt opgenomen voor de account rootdn. Controleer trouwens goed of er niet per ongeluk aan het begin van een regel ergens een spatie te veel staat. Dit kan er toe leiden dat u met een foutmelding geconfronteerd wordt, waardoor het onmogelijk is de gegevens aan de database toe te voegen.

Stap 5. Bewerken van ldap.conf
Als alles goed gegaan is, hebt u nu de LDAP-database gevuld met gebruikersgegevens. Voordat u echter ook de LDAP-database kunt gebruiken vanaf een werkstation, moet u ervoor zorgen dat uw werkstation deze database ook terug kan vinden. Op een Linux-computer doet u dit door het configuratiebestand /etc/openldap/ldap.conf te bewerken. In dit bestand moeten in elk geval twee parameters voorkomen: de parameter HOST waarmee u verwijst naar het IP-adres van de LDAP-server en de parameter BASE waarmee u verwijst naar de container waarin gezocht moet worden. De inhoud van dit bestand zou er bijvoorbeeld als volgt uit kunnen zien:

```
HOST 192.168.0.10
```

BASE dc=sandervanvugt, dc=nl

Kijk of het werkt
Voordat u nu ook maar iets kunt doen, moet u eerst kijken of het allemaal werkt. Gebruik hiervoor de opdracht **ldapsearch**, bijvoorbeeld ldapsearch –x –L –b "dc=azlan, dc=com" –W "(objectclass=*)". Hierna kunt u overgaan naar fase 2, het configureren van een clienttoepassing.

5.4.5 Werken met certificaten

In het voorgaande hebt u gelezen hoe u op redelijk eenvoudige wijze een LDAP-server kunt configureren. Om het eenvoudig te houden, is daarbij niet gesproken over het werken met SSL-certificaten. Dat betekent dat de bovenstaande procedure weliswaar werkt, maar dat alle gegevens als leesbare tekst verstuurd worden. Dat is natuurlijk niet wenselijk, om die reden moet u de LDAP-server configureren om gebruik te maken van SSL-keys. In wezen is. dit een eenvoudige procedure. Om te beginnen moet u de juiste sleutels genereren. Als dat gebeurd is, moet u ervoor zorgen dat de LDAP-server die sleutels ook gaat gebruiken.

Om certificaten aan te maken die gebruikt kunnen worden door de LDAP-server, zorgt u er in een professionele omgeving voor dat u een certificaat krijgt van een algemeen erkende certificate server zoals Verisign. In dit hoofdstuk houden we het echter eenvoudig en wordt gebruikgemaakt van certificaten die u zelf hebt aangemaakt. Het nadeel van deze werkwijze is dat de echtheid van deze certificaten door de gebruiker niet gecontroleerd kan worden, ze komen daardoor als minder betrouwbaar over. Werken met dergelijke certificaten is om die reden niet aanbevolen wanneer u werkt met externe partijen. Als het er echter om gaat binnen uw netwerk een veilige SSL-verbinding met de LDAP-server op te zetten, is er niets op tegen te werken met certificaten die u zelf gemaakt hebt.
De eerste stap in het aanmaken van certificate, bestaat eruit dat u deze certificaten moet genereren. Als de SSL-programmabestanden op uw computer geïnstalleerd zijn, bevindt zich op uw pc een bestand met de naam CA.pl dat voor dit doel gebruikt kan worden. U vindt dit bestand op /usr/share/ssl/misc/CA.pl. Gebruik de opdracht **CA.pl –newcert** om met behulp van dit commando SSL-certificaten te gereneren. U moet vervolgens verschillende stappen doorlopen om het certificaat aan te kunnen maken. In het onderstaande wordt besproken wat er in deze verschillende stappen gebeurt:

```
#/usr/share/ssl/misc/CA.pl –newcert
Generating a 1024 bit RSA private key
..+++++++
……++++++++
writing new private key to 'newreq.pem'
Enter PEM pass phrase:
```

U moet nu een wachtzin invoeren. Dit is net zoiets als een wachtwoord, u gebruikt deze wachtzin om toegang tot de private key te beperken. Voer twee maal achter elkaar dezelfde wachtzin in

```
Verifying – Enter PEM pass phrase:
------
…
Contry Name (2 letter code) [AU]:
```

Bovenstaande is de eerste van een aantal vragen waarmee u aan moet geven van wie de sleutel afkomstig is. De sleutels worden weggeschreven in een PKI-certificaat en de bedoeling van het antwoord op deze vragen, is dat de gebruiker van het certificaat u terug weet te vinden als er iets mis is met het certificaat. Vul uw countrycode in en druk op Enter om verder te gaan. Nadat u uw locatie hebt ingevoerd, moet u vervolgens ook uw eigen naam en mailadres invoeren. De gebruiker van het certificaat moet immers in staat zijn u terug te kunnen vinden. Als de volledige procedure doorlopen is, is in de huidige directory een bestand aangemaakt dat newreq.pem heet. Even opletten dat dit bestand in de huidige directory is aangemaakt, het is aan te raden het te verplaatsen naar een lokatie waarvan u zeker weet dat deze veilig is. In dit bestand vindt u de private key die beschermd is met een wachtwoord en het certificaat zelf. Het is niet handig dat die private key beschermd is met een wachtwoord: dit zou namelijk betekenen dat u het wachtwoord elke keer in moet voeren wanneer u de LDAP-server wilt starten. Gebruik daarom de volgende opdrachten om het wachtwoord van de private key te verwijderen en de resulterende sleutel weg te schrijven naar een nieuw bestand:

```
# openssl rsa –in newreq.pem –out newkey.pem
load RSA key
Enter PEM pass phrase:
writing RSA key
```

Nu hebt u een bestand met de naam newkey.pem. Voordat u hiermee aan het werk kunt, moet er nog wel wat gebeuren: in dit bestand staan namelijk zowel de public key als de private key gescheiden. Voordat u deze keys kunt gebruiken, moet u ze uit dit bestand kopiëren en opslaan in afzonderlijke bestanden. Dit kunt u gewoon doen door te knippen en plakken vanuit uw editor. We gaan er vanuit dat u deze procedure hebt uitgevoerd en dat als resultaat de public key is opgeslagen in het bestand ldap-pub.pem en de public key in ldap-priv.pem. Het enige wat u nu nog hoeft te doen, is naar deze sleutels verwijzen in het configuratiebestand slapd.conf:

```
## TLS options for slapd
TLSCertificateFile /etc/ssl/servercerts/ldap-pub.pem
TLSCACertificatePath /etc/ssl/certs/
TLSCertificateKeyFile /etc/ssl/servercerts/ldap-priv.pem
```

Dit zorgt ervoor dat de LDAP server tijdens het starten de juiste keys kan gebruiken en er een veilige SSL-verbinding met de server opgezet kan worden.

5.4.6 De LDAP-client

Als de server eenmaal draait, wat meestal het grootste probleem niet is, kunt u uw Linux-werkstation configureren om gebruikersnaam en wachtwoord te valideren in de LDAP-database. Zorg er om te beginnen voor dat op elke clientmachine de volgende RPM-packages geïnstalleerd zijn: openldap, auth_ldap en nss_ldap. U kunt met de opdracht rpm –q packagenaam verifiëren of het betreffende package geïnstalleerd is; kijk gelijk ook even met rpm –V packagenaam of alle bestanden uit het package nog intact zijn. Wanneer de juiste packages op de LDAP-client aanwezig zijn, moet u er voor zorgen dat de gebruikers voortaan het lokale wachtwoordmechanisme kunnen omzeilen en kunnen inloggen op de LDAP-server. Voer hiervoor de volgende stappen uit.

1. Pas ldap.conf aan.

2. Pas /etc/nsswitch.conf aan.
3. Zorg ervoor dat PAM gaat gebruikmaken van LDAP.
4. Zorg ervoor dat de gebruikers met de juiste attributen in de database geplaatst worden.

Stap 1. Pas ldap.conf aan
Op uw systeem bestaan soms twee bestanden met de naam ldap.conf. Het bestand /etc/ldap.conf wordt gebruikt door de modules nss_ldap en pam_ldap om te bepalen waar de informatie vandaan gehaald moet worden; het bestand /etc/openldap/ldap.conf wordt gebruikt door clienttoepassingen zoals **ldapsearch** en **ldapadd**., maar ook om ervoor te zorgen dat uw computer de LDAP-server waarop aangemeld moet worden terug kan vinden. Beide bestanden moeten aangepast worden om gebruikt te kunnen worden. Om verwarring te voorkomen, doet u er verstandig aan een van deze bestanden te vervangen door een symbolic link naar het andere bestand. In dit bestand moet in elk geval aangegeven worden vanaf welke host de informatie gehaald moet worden en in welke container standaard naar informatie gezocht moet worden. Er kan meer geconfigureerd worden; dit is echter in eerste instantie niet noodzakelijk.

***ldapconf19 In het configuratiebestand ldap.conf wordt gespecificeerd met welke LDAP-server contact gemaakt moet worden.

Stap 2. Pas nsswitch.conf aan
In /etc/nsswitch.conf wordt bepaald voor welke functionaliteit gebruikgemaakt moet worden van welke configuratiebestanden en/of databases. Zorg er voor dat in elk geval de volgende regels in dit bestand komen te staan:

passwd: files ldap
shadow: files ldap
group: files ldap

Hierdoor kan behalve naar gegevens in de normale bestanden, zoals /etc/passwd, /etc/shadow en /etc/group, ook in de LDAP-database gekeken worden naar informatie om gebruikers te valideren. U neemt deze maatregel om er zeker van te zijn dat elke toepassing in staat is contact te maken met de LDAP-server. De meeste toepassingen echter zullen deze regels niet nodig hebben en door middel van PAM contact maken met de LDAP-server.

Stap 3. Pas PAM aan
De laatste en wellicht ook meest complexe stap in de configuratie is dat u PAM moet aanpassen, zodat gebruikgemaakt wordt van LDAP. Het uiteindelijke configuratiebestand komt er dan als volgt uit te zien:

```
auth        required        /lib/security/pam_securetty.so
auth        sufficient      /lib/security/pam_ldap.so
auth        required        /lib/security/pam_unix.so shadow    nullok
auth        required        /lib/security/pam_nologin.so
account     sufficient      /lib/security/pam_ldap.so
account     required        /lib/security/pam_unix.so
password    required        /lib/security/pam_cracklib.so
password    sufficient      /lib/security/pam_ldap.so
password    required        /lib/security/pam_unix.so shadow nullok use_authtok
sessionsufficient       /lib/security/pam_ldap.so
```

sessionrequired /lib/security/pam_unix.so

Stap 4. Maken van gebruikers met de juiste attributen
Als dit allemaal gebeurd is, bent u er bijna. U moet er nu alleen nog voor zorgen dat gebruikers met de benodigde attributen worden gemaakt. Dit betekent dat alle gebruikersinformatie die normaal in /etc/passwd en /etc/shadow voorkomt, nu in de LDAP-database geplaatst moet worden. U kunt er op twee manieren voor zorgen dat dit gebeurt. De eerste manier is dat u door middel van een aantal perl-scripts de gegevens uit de gebruikersdatabases in de LDAP-database importeert. Maak hiervoor gebruik van de migratiescripts die geïnstalleerd zijn in /usr/share/openldap/migration. Met deze scripts kunt u alle netwerkinformatie die waar dan ook gebruikt moet worden, importeren; het gaat dus niet alleen om gebruikers, maar bijvoorbeeld ook om computernamen.

U kunt er zelf voor zorgen dat de gebruikers met de juiste attributen worden gemaakt; hiervoor moet u gebruikmaken van een LDIF-bestand waarvan dan later de inhoud met ldapadd in de database wordt toegevoegd. Zorg er in dat geval wel voor dat de container waarin u de gebruikers wilt toevoegen bestaat voordat u begint met importeren van gebruikers! Hieronder ziet u een voorbeeld van zo'n bestand:

```
dn: cn=kees,dc=azlan,dc=com
uid: kees
cn: kees
objectclass: account
objectclass: posixAccount
objectclass: top
objectclass: shadowAccount
userpassword: {crypt}S1SpW0yoDDY$yPGD1cJfqb28exnrVyNcy
shadowlastchange: 11354
shadowmax: 99999
shadowwarning: 7
shadowinactive: -1
shadowexpire: -1
shadowflag: -1073744532
loginshell: /bin/bash
uidnumber: 511
gidnumber: 100
homedirectory: /home/kees
```

Houd wel rekening met twee tekortkomingen die nog aan deze werkwijze verbonden zijn. Als eerste is dat de homedirectory; die wordt niet automatisch gemaakt als dit bestand geïmporteerd wordt. Dit probleem kunt u echter oplossen door gebruik te maken van het PAM-configuratiebestand pam_mkhomedir. Vervolgens is dit het wachtwoord. U wilt natuurlijk de wachtwoorden in een versleutelde vorm in de database hebben. De eenvoudigste wijze om dit correct te doen, is door eerst de gebruiker zonder wachtwoord met behulp van een LDIF-bestand in de directory te importeren. Vervolgens kunt u de gebruiker met behulp van de opdracht ldappasswd achteraf een wachtwoord geven.

Werken met de migratiescripts
Het overzetten van gegevens van uw locale computer naar de LDAP-server, is een zeer bewerkelijke taak. U moet er immers voor zorgen dat alle gegevens die u wilt migreren in het

juiste formaat in een LDIF-bestand gezet worden. Wij raden u aan dit niet zelf te doen, maar gebruik te maken van de Migration Tools. U kunt deze downloaden van www.padl.com/OSS/MigrationTools.html. In de volgende procedure wordt besproken hoe u deze handige Perl-scripts kunt downloaden en installeren en hoe u ze vervolgens gebruikt om gegevens uit locale bestanden op uw computer te migreren naar de LDAP-database.

1. Download vanaf www.padl.com/OSS/MigrationTools.html het bestand MigrationTools.tgz en sla dit bestand op in uw homedirectory.
2. Installeer de MigrationTools door in de directory waar u het bestand hebt opgeslagen de opdracht tar –zxvf MigrationTools.tgz te geven. Het resultaat van deze opdracht is dat er een directory wordt aangemaakt waaronder u alle migration tools terugvindt. Er zijn er meerdere, voor elk belangrijk configuratiebestand op het netwerk is er een afzonderlijk bestand. Zo is er bijvoorbeeld het bestand migrate_passwd.pl waarmee u informatie uit uw locale passwd-bestand kunt migreren.

***migtools.tif Met behulp van de migration tools kunt u alle belangrijke informatie van uw computer omzetten naar een geldig LDIF-formaat.

3. Open het bestand migrate_common.ph met een editor. In dit bestand worden verschillende variabelen met de naam NAMINGCONTEXT gedefinieerd. Verzeker u ervan dat de containernamen waarnaar deze variabelen verwijzen ook daadwerkelijk in de LDAP-Directory bestaan voordat u de Migration Tools gaat gebruiken om informatie in de LDAP-Directory te importeren. Hebt u geen zin om voor elke afzonderlijke informatiecategorie een aparte container aan te maken? Zorg er dan voor dat de waarde van al deze variabelen leeg wordt. Dit doet u bijvoorbeeld door een regel als $NAMINGCONTEXT{‘passwd’} “ou=People” te wijzigen in $NAMINGCONTEXT{‘passwd’} “”.
4. Zoek ook in het bestand migrate_common.ph de regel $DEFAULT_BASE = “dc=padl,dc=com”. Op deze regel wordt aangegeven in welke container de LDAP-objecten aangemaakt moeten worden. Wijzig deze regel zodat verwezen wordt naar de container die u wilt gebruiken, bijvoorbeeld $DEFAULT_BASE = “dc=sandervanvugt,dc=nl”.

***migrate-common In het bestand migrate_common.ph geeft u aan naar welke containers informatie van uw locale computer gemigreerd moet worden.

5. Gebruik nu de opdracht **./migrate_passwd.pl /etc/passwd > newusers.ldif**. Met deze opdracht zorgt u ervoor dat alle gebruikers die nu voorkomen in /etc/passwd in een LDIF-bestand gezet worden.
6. Gebruik tot slot de opdracht **ldapadd –D “cn=Manager, dc=sandervanvugt, dc=nl” –W < newusers.ldif** (let op de namen van de LDAP-manager en het LDIF-bestand dat u wilt gebruiken) om de gebruikers die u zojuist uit /etc/passwd geëxporteerd hebt in uw LDAP-database te importeren.

***migrate-passwd.tif Met het migrate-passwd.pl-script exporteert u alle gegevens uit /etc/passwd naar een geldig LDIF-formaat.

U bent nu klaar met importeren van gebruikers uit /etc/passwd en /etc/shadow. Om te bewijzen dat het ook inderdaad werkt, verwijdert u nu een willekeurige gebruikersnaam uit deze twee bestanden. U zult zien dat u toch nog als deze gebruiker in kunt loggen.

Oefening 5.4

Deze oefening kan per computer uitgevoerd worden. Het is niet nodig samen te werken met andere gebruikers. Maak een LDAP-server. Zorg ervoor dat deze server de container dc=*uwnaam*,dc=com gaat bedienen en migreer vervolgens alle gewone gebruikers vanuit /etc/passwd naar deze container. Zorg er tevens voor dat de LDAP-server door middel van SSL beveiligd wordt. Voeg vervolgens een gebruiker met de naam Nadja toe en zorg ervoor dat deze gebruiker alle properties heeft die nodig zijn om op een Linux-werkstation in te kunnen loggen. Pas daarna de PAM-procedure die door het login-commando gebruikt wordt dusdanig aan dat eerst gekeken wordt of geauthenticeerd kan worden op de LDAP-server en alleen als dat niet lukt dat lokaal geauthenticeerd kan worden. Controleer vervolgens of dit werkt.

Samenvatting

In dit hoofdstuk hebt u geleerd hoe u in een netwerk met meerdere Linux computers bepaalde beheerstaken kunt vereenvoudigen. Als eerste hebt u gelezen hoe u er met een DHCP-server voor zorgt dat alle nodes in het netwerk voorzien worden van de benodigde IP-configuratie. Vervolgens hebt u gelezen ewlke rol PAM speelt op het moment dat authenticatie gecentraliseerd geregeld moet worden. Daarna hebt u gelezen hoe het klassieke NIS gebruikt kan worden om de configuratie van uw netwerk op één centrale plaats in het netwerk bij te houden. Tot slot hebt u kunnen lezen op welke wijze een LDAP-server ingezet kan worden als eenentwintigste eeuwse opvolger van de NIS-server.

Oefenvragen

1. Welke opdracht gebruikt u om gebruikers toe te voegen aan een NIS-database?
2. Welke optie gebruikt u in het DHCP configuratiebestand om een default gateway voor werkstations te specificeren?
3. Welke opdracht gebruikt u om de mappen op de NIS-server te genereren?
4. Welke opdracht geeft u op een Linux systeem dat als router functioneert om DHCP-requests door te sturen naar een netwerk waarop een DHCP-server aanwezig is?
5. Hoe geeft u in een PAM-configuratiebestand aan dat het voldoende is als een gebruiker op de betreffende service kan authenticeren en andere statements in het configuratiebestand niet meer geëvalueerd hoeven worden?
6. Welke PAM-module kan gebruikt worden om inloggen alleen toe te staan op bepaalde tijdstippen?
7. Met welke opdracht genereert u nieuwe SSL-keys voor de LDAP-server?
8. Welke opdracht gebruikt u om te achterhalen welke libraries gebruikt worden door een gegeven programmabestand?
9. Hoe heet het algemene configuratiebestand dat door de DHCP-daemon dhcpd gebruikt wordt?
10. Hoe heet het algemene configuratiebestand dat door de LDAP-server slapd gebruikt wordt?

Hoofdstuk 6: Bestanden delen met Linux

Inleiding
Het onderwerp van dit hoofdstuk valt buiten de leerdoelen van LPI 202. Toch vinden wij dat een boek over Linux netwerken niet compleet is zonder een hoofdstuk over de wijze waarop bestanden gedeeld moeten worden door een Linux server. Om die reden maakt u in dit hoofdstuk kennis met de twee belangrijkste systemen die voor dit doel ingezet worden: het Network File System en de Samba fileserver. Omdat deze onderwerpen ook aan bod komen in Leerboek Linux deel 2, worden ze hier niet uitputtend behandeld. Toch willen we u voorzien van voldoende informatie om uw Linux server als NFS of Samba-server in te richten.

Leerdoelen
* Inrichten van Linux als NFS-server
* Inrichten van Linux als Samba fileserver

6.1 Linux als NFS server

NFS (Network File System) is de traditionele manier waarop UNIX-computers bestanden met elkaar delen. De NFS-server moet er voor zorgen dat directories geëxporteerd worden, op de client kan de geëxporteerde directory vervolgens worden binnengehaald met behulp van het commando **mount –t nfs**. Zowel voor gebruik van NFS op een server als op een client moet ondersteuning voor NFS in de kernel gecompileerd worden. Of dit op uw systeem al gebeurt is, kunt u nagaan door het bestand /proc/filesystems te raadplegen. NFS is echter zo algemeen, dat deze functionaliteit vrijwel altijd in de kernel aanwezig is.

Om een computer in te zetten als NFS-server, hebt u een aantal ingrediënten nodig. Om te beginnen is dat een aantal daemons dat ervoor zorgt dat de NFS-service op uw computer beschikbaar gesteld wordt. Daarnaast moet u het configurtiebestand /etc/export aanmaken en daarin de namen opnemen van de directories die u met NFS aan anderen beschikbaar wilt stellen.

6.1.1 De NFS-daemons

Om NFS-services aan andere computers beschikbaar te stellen, zijn de daemons rpc.nfsd en rpc.mountd nodig. Deze services registreren zich vervolgens bij de rpc-portmapper. Dit is een service die de speciale RPC-adressen die door NFS gebruikt worden omzet in poortadressen. Op de meeste distributies wordt de RPC portmapper gestart met behulp van een eigen opstartscript in de directory /etc/init.d. Vaak is het niet nodig om dit proces apart te starten, maar wordt het automatisch gestart wanneer u de NFS-server op uw computer activeert. Op Fedora doet u dit met het script /etc/init.d/nfs, op SUSE Linux wordt hiervoor het script /etc/init.d/nfsserver gebruikt.

6.1.2 Definitie van de shares

Tijdens het opstarten en daarna niet meer, kijkt de NFS-daemon in het configuratiebestand /etc/exports welke directories op de NFS-server voor anderen beschikbaar gesteld worden. Dat het NFS-proces dit alleen tijdens het opstarten doet, betekent dus ook dat u de NFS-service opnieuw moet starten nadat u wijzigingen in dit bestand hebt aangebracht, dit is handig te doen door het commando **killall –HUP nfsd** te gebruiken, maar u kunt natuurlijk ook het init-script voor NFS gebruiken hiervoor.

De structuur van het bestand /etc/exports is vrij eenvoudig; u geeft op welke directories toegankelijk gemaakt worden, vervolgens geeft u eventueel aan voor wie ze toegankelijk zijn

en daarbij specificeert u door middel van opties op welke manier ze gebruikt mogen worden. De inhoud van dit bestand kan er bijvoorbeeld als volgt uitzien:

/share 192.168.0.0/255.255.0.0(rw,no_root_squash,insecure)

In dit voorbeeld wordt een directory met de naam /share geëxporteerd. Om ervoor te zorgen dat alle computers op alle lokale netwerken bestanden in deze directory mogen benaderen, wordt gebruikgemaakt van de aanduiding 192.168.0.0/255.255.0.0. Hiermee wordt verwezen naar alle computers waarvan het IP-adres begint met 192.168. In plaats van een IP-adres en bijbehorend subnetmasker, kunt u hier trouwens ook computernamen of gedeeltelijke domeinnamen opgeven. Ook is het mogelijk om de server gewoon helemaal open te zetten door hier een * te specificeren. Tussen haakjes staan vervolgens de opties die bepalen hoe de computers toegang hebben tot deze directory. De opties die in dit voorbeeld gebruikt worden, zorgen ervoor dat er met weinig restricties contact gezocht kan worden met deze server.

Onder andere de volgende opties zijn beschikbaar.

* **secure** Deze standaardinstelling bepaalt dat aanvragen afkomstig moeten zijn van poorten waarvan het nummer lager is dan 1024. Dit zijn gereserveerde (well-known) poortadressen. Deze optie is met name bedoeld om misbruik door crackers te voorkomen.
* **insecure** Het contact met deze directory mag vanaf elk poortadres opgebouwd worden.
* **ro** De directory wordt read-only gemount. Gebruikers kunnen dus wel bestanden lezen, maar niet schrijven.
* **rw** De directory wordt read/write gemount.
* **noaccess** Hiermee wordt alles beneden de gespecificeerde directory ontoegankelijk gemaakt voor de client. Door gebruik van deze parameter kunt u toegang tot subdirectories van een geëxporteerde directory ontzeggen.
* **link_relative** Symbolic links waarvan de verwijzing absoluut is ingesteld, worden ingesteld op een relatieve verwijzing; een verwijzing naar bijvoorbeeld /etc/hosts wordt dan zoiets als ../hosts. Deze optie heeft alleen zin als het volledige bestandssysteem van een server geëxporteerd wordt, anders loopt u namelijk het risico dat de verwijzingen van de links niet meer kloppen.
* **link_absolute** Symbolic links blijven zoals ze zijn. Dit is het standaardgedrag.
* **root_squash** De gebruiker met het UID 0 en de groep met het GID 0 (de gebruiker en de groep root) komen er niet in als deze optie aanstaat. Alle verzoeken die van hun afkomstig zijn, worden uitgevoerd met het UID 65534 op de server. Deze UID wordt standaard gebruikt door de gebruiker ‘nobody’, als root via een andere machine over een NFS-mount binnen komt, heeft hij dus minimale rechten als deze optie gebruikt is.
* **no_root_squash** Deze standaardoptie zorgt ervoor dat ook de gebruiker root gewoon toegang heeft tot de server. Dit is de standaardinstelling.
* **squash_uids, squash_gids** Met deze opties kunnen UID's en GID's opgegeven worden waarvan aanvragen ook doorgestuurd moeten worden naar de gebruiker anonymous. De syntaxis hiervoor is squash_uids=0-10,13,18,100-1000. Indien van deze optie geen gebruikgemaakt wordt, worden alle client-UID’s doorgestuurd naar gebruikers met dezelfde UID op de NFS-server.
* **all_squash** Deze optie zorgt ervoor dat alle aanvragen geïnterpreteerd worden als afkomstig van de gebruiker "anonymous".
* **map_daemon** Hiermee wordt ervoor gezorgd dat elke gebruikers en groups-ID op de client vertaald wordt in een valide naam die voorkomt op de server. Om deze optie te kunnen

gebruiken, moet op de client de daemon rpc.ugidd actief zijn. De standaard instelling is "map_identity", die alle ID's met rust laat.

* **map_identity** Deze optie specificeert dat een UID van een gebruiker op een client-computer geïnterpreteerd wordt als dezelfde UID op de server en vice versa. Dit is het standaardgedrag. Om dit systeem goed te laten werken, is het handig gebruik te maken van een onderliggend mechanisme waardoor alle gebruikersnamen gedeeld worden. Hiervoor zou gebruikgemaakt kunnen worden van bijvoorbeeld NIS of OpenLDAP.
* **map_static** Met deze optie kan verwezen worden naar een tekstbestand waarin statische mappings tussen users voorkomen. U specificeert de optie als bijvoorbeeld "map_static=/etc/nfs/guid.map". De inhoud van dit bestand ziet er uit als

```
#             remote local
uid     0-99          -           #deze ID's worden gemapt naar anonymous
uid     100-500       1000  #deze ID's worden gemapt naar 1000-1400
gid     0-49          -           #ook deze ID's worden gemapt naar anonymous
gid     50-100 700          #GID 50-100 worden gemapt naar 700-750
```

* **map_nis** De mappings worden overgelaten aan de NIS-server; het NIS-domein van de client moet hierbij als argument gegeven worden.
* **anonuid, anongid** Met deze opties kan worden ingesteld welke UID en GID gebruikt moeten worden voor het account "anonymous".

In het volgende ziet u een voorbeeld van hoe een NFS-configuratie waarin deze opties gebruikt worden eruit kan zien:

```
#Voorbeeld van /etc/exports
/       laksmi(rw) laetitia(rw,no_root_squash)
/pub    (ro,insecure,all_squash)
/pub/private    (noaccess)
/home  @local(ro,all_squash,anonuid=0,anongid=100)
```

Wellicht is het bovenstaande bestand niet echt een voorbeeld van een slimme configuratie (hopelijk ziet u zelf ook waarom), maar de betekenis is als volgt. Op de eerste regel wordt Read/write toegang gegeven aan de computers laksmi en laetitia. Hierbij wordt gebruiker root van computer laetitia niet automatisch gemapt naar de gebruiker anonymous. Dit betekent dat de gebruiker root gewoon als root toegang heeft tot het andere systeem. Lekker makkelijk, maar ook redelijk onveilig. Vervolgens krijgt iedereen toegang tot de directory pub. Er staat hier geen naam van een server gespecificeerd, dus de regel heeft betrekking op iedereen. Mensen die binnen komen, mogen vanaf elke poort een aanvraag sturen en ze krijgen de privileges die de gebruiker "anonymous" op de server heeft. De subdirectory "private" echter is geheim, vandaar dat niemand daar toegang in heeft. Tot slot mogen alle gebruikers uit de netgroup "local" de bestanden onder /home benaderen. Dit gebeurt echter wel met read-only toegang, en alle UIDs en GIDs worden vertaald, respectievelijk in UID 0 en GID 100.

6.1.3 Client authenticatie

Wanneer het gaat om de rechten die u als NFS-gebruiker hebt op een NFS-server, gaat NFS uit van een erg eenvoudig principe. Bij het aanmelden op de NFS-server neemt u uw UID mee naar deze server. Dit betekent dat de server u probeert aan te melden op basis van hetzelfde UID. In het voorgaande overzicht hebt u echter gelezen dat er een aantal opties is dat u op kunt nemen in /etc/exports en waarmee u dit gedrag kunt beïnvloeden.

Wanneer u in uw NFS-omgeving uitgaat van een gebruikersconfiguratie die geheel wordt bijgehouden in de locale wachtwoordbestanden, kan dat tot verwarrende resultaten leiden. Het systeem functioneert echter het best wanneer gebruikgemaakt wordt van een centraal mechanisme waarin gebruikersnamen worden bijgehouden. Denk daarbij aan NIS, LDAP of eventueel een directory-service van een derde partij zoals Novell's eDirectory. Over deze services kunt u elders in dit boek lezen. Ook bij gebruik van dergelijke oplossingen is het nog steeds mogelijk uitzonderingen te definiëren. Hierover hebt u in het voorgaande kunnen lezen.

6.1.4 Locking

In het oorspronkelijke ontwerp, wist de NFS-server niets van locken van bestanden. Dit betekent dat het voor kon komen dat twee gebruikers gelijktijdig hetzelfde bestand geopend hadden en daar wijzigingen in aanbrachten die niet met elkaar in overeenstemming waren. Omdat dit onvoorspelbare en ongewenste gevolgen kan hebben, wordt tegenwoordig automatisch met het starten van een NFS-server de locking op bestanden geregeld. Dit zorgt ervoor dat slechts één gebruiker tegelijk een bepaald bestand geopend mag hebben. Een volgende gebruiker die probeert toegang te krijgen tot hetzelfde bestand, zal er niet inkomen. Hiermee kan op voorhand veel ellende worden voorkomen.

6.1.5 Linux als NFS client

Als de NFS-server eenmaal goed geconfigureerd is, is het een koud kunstje om vervolgens vanaf een andere computer een mount uit te voeren; u gebruikt hiervoor het commando mount en specificeert daarbij het juiste type van de mount, de naam van de server en te mounten directory en tot slot de naam van het punt waarop de directory gemount moet worden. Een voorbeeld hiervan ziet u in het commando mount -t nfs damayanti:/home /home. Hiermee wordt de directory /home vanaf de computer damayanti gekoppeld aan de directory /home op het locale systeem. Deze werkwijze heeft echter wel één nadeel: u moet root zijn om de mount uit te kunnen voeren. Alleen de gebruiker root heeft namelijk permissies om op een systeem een mount uit te voeren. Dit zorgt ervoor dat het er in de praktijk meestal op neerkomt dat NFS-mounts automatisch worden uitgevoerd vanuit /etc/fstab. Als alternatief zou u ervoor kunnen kiezen een sudo-configuratie aan te maken zodat ook normale gebruikers directories mogen mounten.

Bij het mounten van een met NFS-geëxporteerde directory, kunt u een aantal opties specificeren. Met name de volgende opties zijn interessant:

* **rsize=n, wsize=n** Hiermee geeft u op hoe groot de NFS-pakketjes zijn die verstuurd worden. De standaardwaarde is 1024 bytes, voor betere prestaties kunt u beiden instellen op 8192. Alleen wanneer u contact maakt met een versie NFS-server die ouder is dan versie 2, moet u ervoor zorgen dat beide opties staan ingesteld op de waarde 1024. Dit bereikt u door de parameters rsize en wsize gewoon niet te gebruiken.
* **timeo=n** Met deze optie wordt gespecificeerd welke timeout er aan een NFS-request verbonden is. De waarde hiervan wordt opgegeven in tienden van seconden; de standaardwaarde is 7. Wat er gebeurt wanneer deze waarde overschreden wordt, hangt er van af of u de parameter "hard" of "soft" hebt opgegeven.
* **hard** Deze standaard optie zorgt ervoor dat een server een melding geeft als er een timeout optreedt en vervolgens oneindig door gaat met proberen de mount uit te voeren. Dit kan erg vervelend zijn, want het betekent dat u in het ergste geval ook oneindig staat te wachten totdat de mount voltooid is. Dit kunt u echter voorkomen door gebruik te maken van

de optie bg (background) waarmee het verzoek de mount uit te voeren op de achtergrond wordt uitgevoerd.
* **soft** Deze optie zorgt ervoor dat niet oneindig geprobeerd wordt een mount uit te voeren, maar dat een I/O-error gegenereerd wordt wanneer er na 60 seconden nog steeds geen contact gemaakt is.
* **intr** Als deze optie aan staat, is het toegestaan een NFS-request te onderbreken met bijvoorbeeld de toetscombinatie Ctrl-C.

De opties kunnen zowel direct bij het commando mount gebruikt worden, als in /etc/fstab. In beide gevallen worden verschillende opties met een komma van elkaar onderscheiden; zo kan bijvoorbeeld een mount worden uitgevoerd met het commando **mount –t nfs -o rsize=8192,wsize=8192,intr laetitia:/home /home**. Een dergelijke oplossing leent zich er goed voor om uw homedirectory die op een andere server voorkomt lokaal te mounten.

Als u snel een overzicht wilt hebben van mounts die door een NFS-server zijn uitgevoerd, kunt u hiervoor op die server het commando **showmount** gebruiken. Als het commando zonder argumenten gegeven wordt, wordt een lijst gegeven van alle computers die een directory op de NFS-server in gebruik hebben. U kunt het ook gebruiken vanaf een client om aan een server op te vragen wie wat in gebruik heeft, zo kunt u met het commando **showmount -a laetitia** een lijst opvragen van alle computers die een directory van NFS-server laetitia in gebruik hebben.

6.1.6 NFS en beveiliging

Enerzijds is NFS een erg handig protocol om even snel een directory met een andere computer te delen. Daar staat echter tegenover dat NFS anderzijds ook een onveilig protocol is. Deze onveiligheid bestaat eruit dat het standaardprotocol niet voorziet in een werkwijze om de identiteit van een andere computer te garanderen. Dit betekent dat het voor een hacker erg eenvoudig is deze te spoofen. Om die reden is NFS in hedendaagse configuraties in afnemende mate populair. Om er toch voor te zorgen dat NFS op een veilige wijze gebruikt wordt, kunt u het gebruiken in combinatie met een Secure Shell (SSH)-tunnel. Hiermee zorgt u ervoor dat het NFS-verkeer over een veilige versleutelde verbinding verstuurd wordt. Om hiermee een veilige variant van het NFS-protocol te kunnen implementeren, hebt u echter ook extra software nodig. Raadpleeg voor meer informatie onder andere http://www.math.ualberta.ca/imaging/snfs/ of voer in een zoekmachine op internet de string “secure NFS” in.

Opdracht 6.1
Om deze opdracht uit te voeren, hebt u twee computers nodig. De ene computer wordt ingericht als NFS-server en wordt hier aangeduid als NFS-server, de andere computer functioneert als NFS-client.
Configureer NFS-server met een tweetal directories die door middel van NFS gedeeld moeten worden. Als eerste moet de directory home gedeeld worden en bereikbaar zijn voor alle gebruikers die op het lokale systeem bekend zijn. Daarnaast moet de directory /usr gedeeld worden. Hierop mogen alleen gebruikers komen die afkomstig zijn van een computer die voorkomt in hetzelfde netwerk als de NFS-server. Daarnaast moet de gebruiker root op deze share volledige toegang hebben, ook als hij binnenkomt vanaf een andere computer.
Controleer dat de NFS-server naar behoren werkt en zorg ervoor dat de shares direct na opstarten actief zijn. Als dit het geval is, configureert u nu de NFS-client op een dusdanige wijze dat tijdens het opstarten automatisch een mount wordt aangemaakt naar de directory /home die door de NFS-server gedeeld wordt.

6.2 Linux als Samba server

Server Message Blocks (SMB) is het protocol dat in een Microsoft omgeving gebruikt wordt om bestanden te delen tussen clients onderling en tussen servers en werkstations. De gestandaardiseerde versie van dit protocol staat bekend als CIFS (Common Internet File System). Al een behoorlijke tijd is er een open source implementatie van dit protocol: de Samba-server. U leert nu u hoe u deze server kunt configureren om uw Linux-server in te zetten als fileserver in een Windows-omgeving. De belangrijkste aspecten van deze server worden uiteengezet. Zo leest u hoe u deze server kunt inzetten als normale bestandsserver, maar ook hoe u er hem in kunt zetten op een soortgelijke wijze als de Windows NT Primary Domain Controller (PDC). Tevens leest u hoe u een Samba-server kunt integreren in Active Directory.

6.2.1 Componenten van de Samba-server

Voordat we in detail kunnen bekijken op welke wijze een Samba-server in een netwerk ingezet kan worden, zetten we eerst uiteen uit welke componenten de Samba-server bestaat. Om de Samba server naar behoren te kunnen beheren, moet u immers wel weten welke processen en configuratiebestanden daarvoor beheerd moeten worden. Om deze vraag te kunnen beantwoorden, is het overigens van belang te weten met welke versie van de Samba server u werkt. In de meeste gevallen zal dit versie 3.x zijn. Oudere distributies maken gebruik van versie 2.2.x, in dit boek echter besteden we geen aandacht aan de configuratie van 2.2.x servers.

Samba-processen

Om gebruik te kunnen maken van de Samba-server, wordt gebruikgemaakt van een drietal processen. Ten eerste is er smbd. Dit is het proces dat ervoor zorgt dat bestanden gedeeld kunnen worden met behulp van het protocol CIFS (Common Internet File System). Smbd is dus het meest belangrijke proces dat u nodig hebt om te kunnen werken met de Samba-server. Naast smbd is er ook nog het proces nmbd. Dit is de NetBIOS naamserver die nodig is om door middel van NetBIOS over TCP/IP (NBT) de Samba-service bekend te maken. Deze service is niet echt noodzakelijk, maar vervult toch een heel handige rol: dankzij nmbd wordt uw Samba-server namelijk ook zichtbaar in de netwerkomgeving op andere computers. Als derde is er winbdinbdd. Deze service is nieuw in Samba versie 3.x en zorgt ervoor dat de Samba server kan communiceren met een Active Directory omgeving. Later in dit hoofdstuk leert u meer over dit proces.

Om Samba te starten, wordt gebruikgemaakt van scripts in /etc/init.d. Op Fedora gebeurt het eenvoudig en zijn er twee scripts die alles regelen: smb zorgt ervoor dat de Samba-server en alle noodzakelijke componenten (denk bijvoorbeeld aan nmbd) geladen worden, daarnaast is er op Fedora het script winbind dat de communicatie met Active Directory regelt als dat nodig is. SUSE daarentegen maakt gebruik van meerdere scripts:

* smb zorgt ervoor dat de Samba-server gestart wordt.
* nmb wordt gebruikt om nmbd te activeren
* smbfs kan tijdens het booten van de computer worden aangeroepen om mounts naar een Samba-server uit te voeren.

Het configuratiebestand smb.conf

De daemon-processen van de Samba-server maken gebruik van één configuratiebestand. Dit bestand heeft de naam smb.conf en komt doorgaans op de standaardlocatie /etc/samba voor. In dit configuratiebestand wordt de volledige Samba-server gedefinieerd. U kunt er in totaal

een kleine vierhonderd verschillende parameters in opnemen. Deze parameters zijn op uitstekende wijze gedocumenteerd in de man-pagina van smb.conf.

Smb.conf is opgedeeld in twee hoofdonderdelen. Deze worden ‘secties’ genoemd. Om te beginnen is er de sectie [global]. Hierin vindt u algemene parameters die ervoor zorgen dat de Samba-server zijn werk kan doen. Denk hierbij aan algemene zaken zoals de naam van de server die gedefinieerd moet worden. In principe is er niet zo veel nodig in deze sectie om een werkende server te krijgen. Aangezien alle opties een standaardwaarde hebben die standaard goed staat, is het voldoende om hier alleen de naamgeving van de server te regelen. Naast de sectie [global] zijn er secties waarin de shares gedefinieerd kunnen worden. Dit zijn de namen van de gedeelde netwerkbronnen. U kunt hier zowel printers als bestanden mee delen. Om u gelijk maar kennis te laten maken met de mogelijkheden van de Samba-server, ziet u in het onderstaande een minimaal voorbeeld waarmee een werkende Samba-server gedefinieerd kan worden. Op basis van dit voorbeeld kunt u een werkende Samba-server maken, probeer het maar eens uit op uw computer, u zult zien dat u gelijk kunt beginnen met delen van bestanden door gebruik te maken van de directory /tmp. Op een operationele server is het overigens niet verstandig om een share aan te maken op /tmp, maar om snel een Samba server uit te proberen is het een handig voorbeeld: u hoeft namelijk niets aan rechten te doen op het lokale bestandssysteem. Verderop in dit hoofdstuk kunt u lezen waar de verschillende opties in dit voorbeeld precies voor gebruikt worden.

```
[global]
        workgroup = workgroup
        printing = cups
        printcap name = cups
        security = user
        encrypt passwords = yes
        server string = Samba-server
[share]
        comment = Algemene datashare
        path = /tmp
        writeable = Yes
        inherit permissions = Yes
        browseable = yes
        guest ok = no
        printable = no
```

Additionele configuratiebestanden
Naast de drie componenten die in het voorgaande genoemd zijn, kan ook gebruikgemaakt worden van extra configuratiebestanden. Of dit ook inderdaad in uw situatie het geval is, hangt af van de specifieke configuratie die gebruikt wordt. We zullen kort de overige mogelijke configuratiebestanden configureren.

lmhosts Dit bestand kan door de Samba-server gebruikt worden om NetBIOS namen in IP-adressen te vertalen. De opbouw van dit bestand heeft veel weg van de opbouw van het bestand /etc/hosts. Door gebruik te maken van een lmhosts-bestand, kunt u ervoor zorgen dat er geen netwerkverkeer gegenereerd wordt om te achterhalen welk IP-adres bij een bepaalde computernaam hoort. Gebruik van dit bestand is optioneel. Hieronder ziet u een voorbeeld van hoe de inhoud van dit bestand eruit kan zien.

127.0.0.1 localhost
192.168.0.50 xtina
192.168.0.51 laetitia

smbfstab In sommige gevallen wilt u ervoor zorgen dat tijdens het opstarten van uw Linux computer automatisch een aantal Samba-shares gemount worden. Als u gebruikmaakt van het bestand /etc/fstab, heeft dit tot gevolg dat gewonen gebruikers de definities van deze shares kunnen uitlezen. Dit is vervelend, aangezien hier ook gebruikersnamen en wachtwoorden in voor kunnen komen. Om die reden kan voor dit doel gebruikgemaakt worden van een afzonderlijk configuratiebestand. Houd er overigens rekening mee dat dit bestand niet automatisch ook afdoende beveiligd is. In veel gevallen wordt namelijk gewoon nog gebruikgemaakt van de permissiemodus 644.

smbpasswd Dit bestand wordt onder bepaalde omstandigheden gebruikt om de namen van Samba-gebruikers en de bijbehorende versleutelde wachtwoorden bij te houden. In versie 3 van de Samba-server is de functionaliteit van dit bestand overgenomen door de zogenaamde Trivial Database (TDB), het configuratiebestand smbpasswd kan echter nog steeds gebruikt worden.

smbusers Soms is het nodig dat de naam van een gebruiker op het werkstation vertaald wordt in de naam van een gebruiker op de Linux-server. Denk bijvoorbeeld aan een mapping van de gebruiker administrator naar de Linux-gebruiker root. Om dit soort mappings automatisch plaats te laten vinden, kunt u gebruikmaken van het bestand smbusers.

6.2.2 Samba als fileserver

We zullen nu een aantal mogelijkheden bespreken om de Samba-server te configureren. Voordat we bespreken wat er moet gebeuren om met succes een Samba-server te installeren, zullen we eerst duidelijk maken welke verschillende componenten met elkaar moeten communiceren om een werkende Samba-server te krijgen. Er moet namelijk op Linux het een en andere geconfigureerd worden en daarnaast ook op Windows. Vervolgens moet u ervoor zorgen dat beide werelden elkaar ook kunnen begrijpen.

Een Samba-server is een Linux server die aan Windows-gebruikers toegang geeft tot Linux-bestanden. Dit betekent dat er ergens een vertaalslag plaats moet vinden. De toegang tot de Linux-bestanden wordt namelijk geregeld door middel van Linux-permissies die uitgedeeld worden aan Linux-gebruikers. Dit zit hem in het Linux-bestandssysteem en ook al benadert u dat met Samba, het is en blijft een Linux-bestandssysteem. U kunt dit vergelijken met de configuratie van een Windows 2003 netwerkomgeving. Als u daar immers alleen de share definieert maar de NTFS-permissies niet regelt, hebben gebruikers namelijk ook geen toegang. Om de hele procedure compleet te maken, beginnen we bij de configuratie van de Linux permissies.

6.2.2.1 Permissies, owners en andere Linux eigenaardigheden

Onder Linux zijn er drie belangrijke permissies: Read, Write en Execute. Deze permissies worden uitgedeeld op bestanden en op directories. In de onderstaande tabel kunt u lezen wat de betekenissen zijn van de permissies.

<<TABEL MAKEN>>

	Bestanden	Directories
Read	Bekijken van de inhoud	Zien van bestanden in de directory

Write	Wijzigen van de inhoud	Aanmaken van bestanden
	Verwijderen van het bestand	Verplaatsen van bestanden
		Verwijderen van bestanden
Execute	Opstarten van programma's	Subdirectories activeren

<<EINDE TABEL>>

Waar op netwerkbesturingssystemen als Netware of Windows op een uitgebreide manier met groepen gewerkt wordt, ligt de situatie onder Linux anders. Om te beginnen heeft elk bestand onder Linux een eigenaar. De eigenaar van het bestand is in principe de gebruiker die dit bestand heeft aangemaakt, maar dit kan gewijzigd worden. Wanneer u de opdracht **ls -l** geeft, krijgt u in de tweede kolom een overzicht te zien van gebruikers die eigenaar zijn van bestanden in de directory die op dat moment actief is.

*** owners Met de opdracht ls -l kunt u precies zien wie er eigenaar is van de bestanden in de huidige directory.

Naast het feit dat elk bestand een eigenaar heeft, is ook aan elk bestand een groep als eigenaar verbonden. Ook deze groep is relevant voor de permissies. Nu bent u van andere netwerk besturingssystemen misschien gewend dat een gebruiker lid kan zijn van meerdere groepen tegelijk. Onder Linux ligt dit een stuk lastiger. Elke gebruiker heeft één primaire groep waar hij lid van is. U kunt gebruikers wel lid maken van meerdere groepen, maar dit is zo omslachtig dat vrijwel iedereen het probeert te vermijden, we gaan er hier dan ook niet verder op in hoe u dit zou moeten regelen. Naast de gebruiker en groep die eigenaar zijn van een bestand, is er de rest van de wereld, lees: alle andere gebruikers die op het systeem gedefinieerd zijn. Ook aan de rest van de wereld worden permissies gegeven. De standaardinstelling is in de meeste gevallen dat de rest van de wereld hooguit de bestanden mag lezen. U zou zich echter af moeten vragen of zelfs zo'n minimale instelling al wenselijk is.

Wanneer een gebruiker een bestand benadert op een Linux-computer, moet er altijd bepaald worden welke permissies deze gebruiker op het betreffende bestand heeft. Hiervoor worden achtereenvolgens de volgende vragen gesteld:

1. Is de gebruiker eigenaar van het bestand?
2. Is de gebruiker lid van de groep die eigenaar is van het bestand?
3. Zo niet: dan behoort de gebruiker dus bij de rest van de wereld.

Overeenkomstig deze drie entiteiten worden permissies op Linux bestanden uitgedeeld. Wanneer u de opdracht **ls -l** geeft in een willekeurige directory, wordt als resultaat een aantal kolommen getoond. De permissies worden weergegeven aan het begin van elke regel. Dit ziet er uit als -rwx-r-xr-x. Het eerste liggende streepje wordt gebruikt om aan te geven wat voor type bestand het is. Als hier bijvoorbeeld de letter d staat, is het een directory. Daarna worden de volgende drie posities gereserveerd voor de permissies van de eigenaar van het bestand. In dit voorbeeld is dat rwx Het tweede groepje van drie wordt gebruikt om aan te duiden wat de permissies zijn van de groep. Als laatste worden de permissies van alle andere gebruikers getoond.

6.2.2.2. Groepslidmaatschap van Linux gebruikers

Zoals eerder gezegd, op een Windows of Netware systeem is het gebruikelijk dat een gebruiker lid is van meerdere groepen tegelijk. Op Linux is dit niet gebruikelijk. Een

gebruiker is meestal slechts lid van één groep en deze groepslidmaatschap is geregeld in de gebruikersdatabase /etc/passwd. We noemen dit overigens de primaire groep.

*** passwdgroep In /etc/passwd wordt gedefinieerd van welke groep een gebruiker lid is.

In de afbeelding ziet u een voorbeeld van een de gebruikersdatabase /etc/passwd. Op de laatste regel wordt gebruiker pleunie gedefinieerd. Het tweede veld bevat alleen een kruisje dat aangeeft dat het wachtwoord van Pleunie in /etc/shadow versleuteld staat opgeslagen. In het derde veld treffen we de gebruikers ID (UID) van pleunie. Dat is het unieke nummer waarmee pleunie intern bij het systeem bekend is. Vervolgens is in het vierde veld de group ID (GID) vermeld. Het is handig wanneer dit voor alle gebruikers gelijk is, houdt er echter rekening mee dat Fedora niet uitgaat van dit principe en voor elke gebruiker een groep aanmaakt met de naam van de gebruiker. Alleen de gebruiker zelf is lid van die groep. Vaak wordt voor dit doel gebruikgemaakt van de groep met GID 100. Welke groep dat precies is, wordt geregeld in het groepsbestand /etc/group.

*** group In /etc/group wordt gedefinieerd welke groepen er op een systeem voorkomen.

Om te kijken van welke groep alle gebruikers lid zijn, moet dus het bestand /etc/group geopend worden. Hierin lezen we in dit voorbeeld dat het om de groep “users” gaat. Alle gebruikers zijn dus lid van de zelfde groep.

6.2.2.3 Van Linux naar Samba

Wanneer u aan de hand van het bovenstaande ervoor gezorgd hebt dat de Linux server op het gebied van rechten goed is afgeschermd, kunt u de Samba-server inrichten. Hiervoor moeten twee dingen gebeuren. Als eerste moeten de gebruikers van de Samba-server gedefinieerd worden. De Windows-gebruikers die binnenkomen met hun Windows-gebruikersnaam, kunnen namelijk niet rechtstreeks authenticeren op de Linux-gebruikersdatabase. Dit komt omdat Windows van een totaal ander authenticatiemechanisme gebruik maakt als Linux. Om die reden moet met de opdracht **smbpasswd** een database aangemaakt worden waarin de Samba gebruikers gedefinieerd zijn. Op een Samba-server bestaat elke gebruiker dus twee keer: eerst een keer als Linux gebruiker, daarna een keer als Samba gebruiker. Dit kan overigens ook op andere manieren geregeld worden, daarover leest u verderop in dit hoofdstuk meer.

Wanneer ervoor gezorgd is dat de Samba-gebruikers gedefinieerd zijn, moeten vervolgens de Samba-shares aangemaakt worden. Dit zijn de gedeelde bronnen die de gebruikers in hun netwerkomgeving te zien krijgen. U regelt dit in het configuratiebestand smb.conf. U kunt dit bestand op de meeste distributies terugvinden in de directory /etc/samba. Wanneer u ervoor gezorgd hebt dat dit configuratiebestand is aangemaakt, moet ervoor gezorgd worden dat de Samba processen gestart worden.

6.2.3 Configuratie van de Samba-server: een praktijkvoorbeeld

Nu u op de hoogte bent van wat er allemaal moet gebeuren om een Samba-server aan het werk te krijgen, zullen we een praktisch scenario uitwerken waarin een Samba-server wordt ingericht. Alle noodzakelijke stappen worden op een rij gezet zodat u direct aan het werk kunt. Verderop in dit hoofdstuk leest u in meer detail hoe de Samba-server geconfigureerd kan worden.

In dit scenario wordt uitgegaan van een klein bedrijf waarin twee gebruikers op de administratie zitten en er drie mensen op de werkvloer staan. Beide groepen hebben een privé directory die niet toegankelijk is voor mensen uit de andere groepen. Gebruikers Linda en Frans zijn lid van de groep Administratie, Kees, Eric en Sander zijn lid van de groep Delivery. Leden van de beide groepen moeten vanaf hun Windows werkplek toegang krijgen tot hun home-directory en daarnaast ook tot hun gedeelde groepsdirectory. Zorg ervoor dat u ingelogd bent als root om de onderstaande procedure uit te kunnen werken.

1. Aanmaken van groepen

Het is handig om als eerste de benodigde Linux groepen aan te maken. U kunt er dan namelijk bij het aanmaken van de gebruikers voor zorgen dat deze direct aan de juiste groep toegevoegd kunnen worden. Maak hiervoor gebruik van de opdracht groupadd.

groupadd delivery
groupadd administratie

2. Aanmaken van gebruikers

Nu de groepen zijn aangemaakt, kunnen als tweede stap de gebruikers aangemaakt worden. Hiervoor maken we gebruik van de opdracht **useradd**. Door bij dit commando de optie -g te gebruiken, kan direct aangegeven worden van welke groepen de gebruikers lid gemaakt moeten worden. Op sommige distributies moet u ook de optie -m gebruiken om ervoor te zorgen dat de home-directories aangemaakt worden. Nadat de gebruikers zijn aangemaakt, kunt u ze met de opdracht **passwd** van een wachtwoord voorzien.

useradd -g administratie –m linda
passwd linda
useradd -g administratie –m frans
passwd frans
useradd -g delivery –m kees
passwd kees
useradd -g delivery –m eric
passwd eric
useradd -g delivery –m sander
passwd sander

3. Aanmaken van de groepsdirectories

Nu de gebruikers en groepen gecreëerd zijn, moet u ervoor zorgen dat de gedeelde groepsdirectories waarvan deze gebruikers gebruik gaan maken ook gemaakt worden. Dit doet u met de opdracht **mkdir**. De optie –p in het onderstaande voorbeeld zorgt er trouwens voor dat bij het aanmaken van de directory /data/administratie ook gelijk de directory /data wordt aangemaakt als deze nog niet bestaat.

mkdir –p /data/administratie
mkdir –p /data/delivery

4. Aanpassen van permissies

Alle directories zijn op dit moment aangemaakt, alleen zijn nog niet de juiste gebruikers en groepen lid van deze directories. Om hier verandering in aan te brengen, maakt u gebruik van de opdracht **chown** wat staat voor Change Owner. Met dit commando kunt u zowel de gebruiker als de groep die eigenaar moet worden aanpassen. De namen van de gebruiker en de

groep die lid moeten worden, worden in dat geval met een punt (een dubbele punt mag ook) van elkaar gescheiden.

chown root.administratie /data/administratie
chown root.delivery /data/delivery

5. Alle Linux permissies staan nu goed ingesteld. In de volgende stap maken we de Samba-gebruikers aan. De Windows-gebruiker meldt zich immers vanaf zijn pc aan op de share door middel van zijn Windows-credentials. Deze Windows-credentials moeten vervolgens via de Samba-gebruiker doorgegeven worden aan het Linux-systeem. Om de Samba-gebruikers aan te maken, gebruikt u de opdracht **smbpasswd**. Direct na het aanmaken van de Samba-gebruiker, vraagt het systeem om een wachtwoord. Het is handig maar niet verplicht hier gebruik te maken van hetzelfde wachtwoord als dat de Linux-gebruiker in gebruik heeft.

smbpasswd -a linda
smbpasswd -a frans
smbpasswd -a kees
smbpasswd -a eric
smbpasswd -a sander

6. Aanpassen van de shares in smb.conf.
Tot nu toe hebben we ons alleen nog maar beziggehouden met het voorbereidende werk. De werkelijke configuratie van de Samba-server bestaat eruit dat het configuratiebestand smb.conf wordt aangepast. We laten u hier een redelijk minimalistisch voorbeeld van dit bestand zien, verderop in dit hoofdstuk kunt u kennismaken met uitgebreidere voorbeelden.

```
[global]
        workgroup=Samba
        netbios name = samba
        time server = yes
        encrypt passwords = yes
        socket options = SO_KEEPALIVE IPTOS_LOWDELAY TCP_NODELAY
        wins support = no
        veto files = /*.eml/*.nws/riched20.dll/*.{*}/
        security = user
[homes]
        comment = Home Directories
        valid users = %S
        browseable = no
        writeable = yes
        create mask = 0640
        directory mask = 0750
[administratie]
        comment = administratie
        valid users = linda frans
        browseable = no
        writeable = yes
        create mask = 0660
        directory mask = 0770
```

```
[delivery]
        comment = delivery
        valid users = kees eric sander
        browseable = no
writeable = yes
        create mask = 0660
        directory mask = 0770
```

We zetten kort de betekenis van de verschillende opties uit het bovenstaande voorbeeld naast elkaar.

workgroup=samba
Hiermee wordt de naam van de werkgroep aangegeven zoals die in de netwerkomgeving zichtbaar zal zijn.

netbios name = samba
Met behulp van deze instelling bepaalt u hoe de Samba-server zichzelf op het netwerk bekend gaat maken.

time server = yes
Deze waarde bepaalt dat de Samba-server zich als tijdbron adverteert voor de Windows clients in het netwerk. Dat is handig omdat dan de klokken van de werkstations in uw netwerk gelijk staan met de klok van uw server.

encrypt passwords = yes
Deze parameter zorgt ervoor dat de Samba-server in staat is Windows versleutelde wachtwoorden te ontcijferen. U moet deze parameter op yes hebben staan wanneer u gebruikmaakt van Windows 95b of Windows NT 4.0 SP 3 of hoger.

socket options = SO_KEEPALIVE IPTOS_LOWDELAY TCP_NODELAY
Deze parameter zorgt ervoor dat de Samba-server zo efficiënt mogelijk op het netwerk communiceert.

wins support = no
Hiermee wordt aangegeven dat de Samba-server op het netwerk niet de rol van WINS-server vervult.

veto files = /*.eml/*.nws/riched20.dll/*.{*}/
Dit is een lijst met bestanden die nooit in een Samba-share zichtbaar zullen worden. Deze bestanden kunnen dus ook niet gebruikt worden. De verschillende bestandsnamen worden met behulp van een / van elkaar onderscheiden.

security = user
Er zijn verschillende manieren waarop u de authenticatie op uw Samba-server kunt regelen. Al deze manieren worden gedefinieerd met de regel security =. Het meest gebruikelijk is dat er gebruikgemaakt wordt van de regel security = user. Hiermee wordt bepaald dat op de Samba-server een aparte gebruikersdatabase moet worden aangemaakt waarop de Windows gebruikers kunnen authenticeren. Als alternatief kan ook gebruikgemaakt worden van andere instellingen. Hierover leest u verderop in dit hoofdstuk meer.

[homes]
Door een willekeurige naam tussen blokhaken te zetten en volledig links uit te lijnen, wordt een nieuwe share gedefinieerd. Vergelijk dit bijvoorbeeld ook met de shares [administratie] en [delivery].

comment = Home Directories
Dit is de aanduiding voor de share die in de netwerkomgeving zichtbaar wordt.

valid users = %S
Met de parameter valid users kan aangegeven worden welke gebruikers toegang hebben tot een share. De waarde %S staat voor alle gebruikers die een account hebben. Als alternatief kunnen hier ook namen van gebruikers gespecificeerd worden. Wanneer de permissies op Linux niveau goed geregeld zijn, is gebruik van deze parameter in feite overbodig.

browseable = no
Door in de share de optie browseable op no te zetten, wordt bepaald dat de share niet getoond wordt in het overzicht van beschikbare shares. Dit is vooral nuttig wanneer het gaat om de share die toegang geeft tot de home-directories, zo voorkomt u namelijk dat gebruikers naar elkaars home-directory gaan browsen. De gebruikers zullen overigens wel hun eigen gedeelde home-directory terugvinden in hun netwerkomgeving.

writeable = yes
Hiermee wordt bepaald dat gebruikers gewoon read/write toegang hebben tot de bestanden als hun Linux permissies dat ook toestaan. Ze mogen dus bestanden in de share aanmaken en bestanden die in de share voorkomen kunnen gelezen worden.

create mask = 0660
Met deze parameter worden de standaardpermissies voor nieuw aan te maken bestanden ingesteld. Hierbij wordt de permissie read vertegenwoordigd door de waarde 4, write is 2 en execute is 1. Het eerste getal kan verwaarloosd worden. Het tweede getal verwijst naar de permissies van de owner, het derde getal naar de permissies van de groep en het laatste getal naar de permissies van de rest van de wereld. Let vooral even op het gebruik van deze instelling bij de gedeelde directories: door hier 0660 te specificeren, wordt bepaald dat iedereen die lid is van de betreffende groep alles mag met de bestanden in de directory. Voor gedeelde groepsdirectories is dat waarschijnlijk ook de bedoeling, wanneer u echter deze instelling gebruikt op de home-directories van de gebruikers, loopt u het risico dat deze daar iets minder blij mee zijn.

directory mask = 0750
Zelfde functie als het create mask, maar dan toegepast op directories die worden aangemaakt.

Wanneer u handmatig met behulp van een editor een Samba-configuratiebestand aanmaakt, loopt u natuurlijk het risico dat dit bestand door een typefout niet gebruikt kan worden. Het is daarom handig om even een controle uit te voeren op het bestand voordat u de Samba-server start. Gebruik hiervoor de opdracht **testparm**. Dit commando voert een controle uit of de syntaxis van het bestand in orde is.

***testparm Met behulp van de opdracht testparm kunt u controleren of er geen syntaxis-fouten voorkomen in het Samba-configuratiebestand.

7. Opstarten van de daemons

Wanneer u de bovenstaande stappen succesvol hebt uitgevoerd, hoeven alleen nog de daemons opgestart te worden. Gebruik hiervoor de scripts in de directory /etc/init.d van uw server.

6.2.4 Samba als PDC

Tot nu toe hebt u kunnen lezen over de Samba-server in een omgeving met werkgroepen. Het werkgroepmodel is echter voor professionele omgevingen niet de meest ideale oplossing. In dat soort omgevingen is het aan te raden uw Samba-server te installeren als Domain Controller. Dit biedt onder andere het voordeel dat u de werkomgeving van de gebruiker veel beter kunt vormgeven. Door middel van login-scripts zorgt u ervoor dat bepaalde zaken automatisch geregeld worden wanneer de gebruiker zich aanmeldt, daarnaast kunt u gebruikmaken van Roaming Profiles die ervoor zorgen dat de voorkeursinstellingen van de gebruiker altijd beschikbaar zijn, waar op het netwerk hij zich ook aanmeldt. U kunt hier lezen hoe u de Samba-server aan kunt maken als Windows NT 4-stijl Domain Controller. Voor integratie van de Samba-server in een omgeving waarin Active Directory gebruikt wordt, kunt u de volgende paragraaf lezen. U kunt er overigens meteen alvast rekening mee houden dat het niet mogelijk is een Samba-server in te zetten als Domain Controller in een Active Directory omgeving.

6.2.4.1 De rol van de Domain Controller

Wanneer u Samba inzet als Domain Controller, maakt u gebruik van een configuratiemethode die door Microsoft werd toegepast in Windows NT 4.0. Hierbij is de Domain Controller de server waarop alle gegevens worden bijgehouden die nodig zijn om aan te kunnen melden op een netwerk. Windows NT 4.0 maakt hiervoor gebruik van twee verschillende soorten servers: de PDC en de BDC.

De Primary Domain Controller (PDC) is de server in het netwerk die de hoofdrol speelt. Wanneer u gegevens van gebruikers wilt kunnen aanpassen, hebt u altijd deze PDC nodig. Vanuit de optiek van fouttolerantie wordt naast de PDC ook gebruikgemaakt van een Backup Domain Controller (BDC). Deze vervult twee taken. Als eerste maakt de BDC het mogelijk om gegevens over het netwerk te verspreiden. Hierdoor kan het voor een gebruiker eenvoudiger gemaakt worden om te authenticeren omdat de netwerkgegevens zich dichter bij hem bevinden. Daarnaast functioneert de BDC als reservekopie van de PDC. Dit betekent dat de BDC in staat is het werk van de PDC over te nemen. Dit gaat echter niet zondermeer, om een BDC te verheffen tot PDC moet u hem ‘promoveren’. In een Windows-omgeving maakt u hiervoor gebruik van de opdracht **dcpromo**, dit commando bestaat echter niet onder Linux.

Wanneer u Samba als Domain Controller inricht, moet u er rekening mee houden dat de Samba-server niet opgenomen kan worden in een bestaand Windows NT 4-netwerk waarin al Domain Controllers voorkomen. Samba kan namelijk niet communiceren met Windows NT Domain Controllers. Wel is het mogelijk voor een Samba PDC om te communiceren met Samba BDC’s. Voor meer informatie hierover verwijzen we naar de Samba-BDC-HOWTO die u onder andere op www.tldp.org kunt raadplegen.

6.2.4.2 De configuratie van de Samba PDC

Uiteraard moet u om een Samba-PDC te maken het configuratiebestand smb.conf bewerken. Houd er rekening mee dat de wijze waarop u de PDC configureert afhankelijk is van het type werkstation dat u gebruikt. In een omgeving met alleen Windows 9x hebt u een andere configuratie nodig als in een omgeving met Windows NT, 2000 en XP werkstations. In het

onderstaande voorbeeld vindt u een configuratie die geoptimaliseerd is voor een omgeving waarin gebruikgemaakt wordt van Windows 2000/XP werkstations.

```
[global]
netbios name = samba
workgroup = MYDOMAIN
os level = 99
encrypt passwords = yes
log file = /var/log/samba/%m
debug level = 1
socket options = SO_KEEPALIVE IPTOS_LOWDELAY TCP_NODELAY
wins support = yes
character set = ISO8859-15
client code page = 850
veto files = /*.eml/*.nws/riched20.dll/*.{*}/
time server = yes

domain master = yes
local master = yes
preferred master = yes
domain admin group = root

domain logons = yes
security = user

logon script = %U.bat
logon path = \\%L\profiles\%u\%m
logon home = \\%L\%u\.win_profile\%m
logon script = logon.bat
logon drive = H:

add user script = /usr/sbin/useradd –d /dev/null –g 100 –s /bin/false %u


[netlogon]
comment = Network Logon Service
path = /usr/local/netlogon
writable = no
browsable = no

[profiles]
path = /home/sambaprofiles
browsable = no
writable = yes
create mask = 0600
directory mask = 0700

[homes]
read only = no
browsable = no
```

guest ok = no
map archive = yes

De bovenstaande configuratie zorgt ervoor dat u een volledig werkende Samba-server kunt instellen als PDC. We zullen in het onderstaande niet alle parameters bespreken, maar ons vooral richten op die instellingen die relevant zijn voor een Samba-PDC. Veel van deze instellingen hebt u al eerder in dit hoofdstuk gezien.

De configuratie bestaat in dit geval naast de sectie [global] uit drie andere secties. Ten eerste is dat [netlogon] waarin geregeld wordt dat gebruikers toegang krijgen tot hun logon-script. Met dit script kunt u ervoor zorgen dat bepaalde instellingen automatisch worden uitgevoerd wanneer de gebruiker zich aanmeldt. Vervolgens wordt in [profiles] de toegang tot de profielen van de gebruikers geregeld en tot slot is er de sectie [homes] waarin geregeld wordt dat gebruikers ook toegang krijgen tot hun home-directories. Afhankelijk van de werkwijze die u wilt gebruiken, zou u ook zonder deze laatste drie secties kunnen werken. Pas dit dus aan naar eigen behoefte.

De eerste parameter die opvalt in de sectie global is netbios name = samba. Hiermee wordt de naam ingesteld waarmee u de Samba-server in de netwerkomgeving van de client-computers terug zult vinden. Vervolgens is er een aantal instellingen waarmee bepaald wordt dat uw Samba-server ‘de baas’ is in het netwerk. Om te beginnen moet de Samba-server onder alle omstandigheden de verkiezing van de master browser winnen. De parameter os level = 99 zorgt hiervoor. Het effect hiervan is dat de Samba-server het browse-proces in uw netwerk beheert en er dus voor zorgt dat andere computers zichtbaar kunnen worden in uw netwerkomgeving. Het is belangrijk deze taak over te laten aan de Samba-server, als namelijk een normale Windows-computer de verkiezing van Master Browser zou winnen, bestaat het risico dat u de Samba-server nooit in de netwerkomgeving van uw computers terugvindt. Een volgende parameter die opvalt is time server = yes. U raadt het waarschijnlijk al: met behulp van deze instelling zorgt u ervoor dat de Samba-server de tijd op uw netwerk bepaalt. De werkstations halen de tijd op bij uw Samba-server en wanneer uw Samba-server vervolgens door middel van NTP de tijd ophaalt bij een Internet-tijdserver, zorgt dit ervoor dat overal op het netwerk dezelfde tijd gebruikt wordt. Dit is zeer aan te raden, op deze manier kunnen ook toepassingen die op een of andere manier afhankelijk zijn van de juiste netwerktijd hun werk goed doen.

Dan komt er weer een aantal instellingen dat ermee te maken heeft dat de Samba-server de browser-verkiezingen op het netwerk wint. De parameter domain master = yes zorgt ervoor dat de Samba-server Domain Master Browser wordt. Dit betekent dat de Samba-server ook in een domein dat uit verschillende fysieke netwerken bestaat, de browser-services kan verschaffen. Met de parameter local master = yes wordt aangegeven dat de Samba-server meedoet in de browser-verkiezingen en ook in staat is deze te winnen. De regel set preferred master zorgt er in combinatie met het os-level voor dat de Samba-server de verkiezing van de master-browser ook wint. Interessant om te weten: door middel van set preferred master = yes wordt ervoor gezorgd dat er direct na het starten van de Samba-server een browser-verkiezing geforceerd wordt die de Samba-server vervolgens ook gaat winnen.

Vervolgens zijn er twee regels die ervoor zorgen dat de Samba-server zijn werk als PDC kan doen:

security = user

domain logons = yes

De regel security = user komt misschien als een verrassing aangezien er ook een optie is om de security in te stellen op domain. Er is echter een belangrijk verschil tussen beiden. Met de regel security = domain zorgt u ervoor dat een andere PDC gebruikt wordt als domain controller. Deze instelling is dus uitstekend wanneer u de authenticatie van gebruikers af wilt laten handelen door een andere Samba-server of een Windows-PDC. Als u er echter voor wilt zorgen dat uw Samba-server zich als PDC gaat gedragen, gebruikt u altijd de instelling security = user. Met de regel domain logons = yes vervolgens vertelt u aan deze Samba-server dat hij het werkstations mogelijk moet maken in te loggen op het domain.

Vervolgens komt er een aantal instellingen waarin geregeld wordt dat de gebruikersomgeving op de juiste wijze gedefinieerd wordt:

logon script = logon.bat
logon path = \\%L\profiles\%u\%m
logon home = \\%L\%u\.win_profile\%m
logon drive = H:

Om te beginnen wordt het logon-script logon.bat aangeroepen. Dit is een MS-DOS batchfile waarin commando's staan die ervoor zorgen dat de gebruikersomgeving wordt vormgegeven. Hierbij kunt u onder andere denken aan een aantal netwerkverbindingen die automatisch worden aangemaakt. De regels waarin het logon path en de logon home gedefinieerd worden, zorgen er vervolgens voor dat op de juiste wijze verwezen wordt naar de locatie waar de roaming profiles met daarin de instellingen van de gebruiker bewaard worden. De regel komt in twee varianten voor: dit is nodig om zowel aan Windows 9x gebruikers (logon home) als aan Windows NT en later gebruikers (logon path) de juiste ondersteuning te geven. Door middel van variabelen wordt verwezen naar de precieze directories die verderop in het configuratiebestand gedefinieerd worden. Bovendien zorgt de variabele %u ervoor dat voor iedere gebruiker naar een aparte directory verwezen wordt. In deze directory wordt vervolgens het profile van de gebruiker neergezet. Dit profile heeft voor Windows NT en later standaard de naam NTprofile.dat en moet beschrijfbaar zijn voor gebruikers. Mocht het nodig zijn om de gebruikers iedere keer met vaste instellingen te confronteren, dan kunt u als beheerder de naam van dit bestand wijzigen naar NTprofile.man. U maakt er zo een mandatory profile van dat door gebruikers niet gewijzigd kan worden.

Tot slot is er in deze verzameling instellingen de aanduiding waarmee de logon-drive voor gebruikers geregeld wordt. Hiermee zorgt u ervoor dat de gebruiker vanuit de Windows verkenner automatisch een driveletter H: ziet die verwijst naar de home-directory van de gebruiker. Deze parameter is optioneel, maar voor een gebruiker is het wel erg prettig als hij bestaat.

Als laatste in de sectie [global] is er de parameter add user script. Hiermee zorgt u ervoor dat op het lokale Linux-systeem automatisch een gebruiker aangemaakt wordt wanneer deze nog niet bestaat. Wanneer u gebruikmaakt van deze parameter, moet u er zeker van zijn dat de authenticatie op de Samba-server betrouwbaar is. Wanneer het voorkomt dat een gebruiker wél op de Samba-server kan authenticeren, maar niet bestaat in de lokale gebruikersconfiguratiebestanden op de Linux machine, treedt dit script in werking. In het voorbeeld add user script = /usr/bin/useradd –d /dev/null –g 100 –s /bin/flase %u wordt op de Linux-server een lokale gebruiker aangemaakt die daar verder ook helemaal niets kan. Met de

optie –d /dev/null wordt zijn home-directory doorgelust naar het null-device, hij krijgt als Shell de niet-shell /bin/false toegewezen. Dit laatste zorgt ervoor dat de gebruiker niet in staat is lokaal in te loggen op de Linux-machine. Het enige dat wel geregeld wordt, is dat de gebruiker automatisch lid gemaakt wordt van de groep met GID 100. In de meeste gevallen wordt deze GID gebruikt voor een algemene groep waarvan alle gebruikers lid zijn. Uiteraard kunt u de opdracht waarmee een gebruiker wordt aangemaakt ook zo instellen dat de gebruiker op de lokale Linux-computer ook wat kan, u dient zich echter af te vragen of dit wenselijk is. Wij denken dat het nuttig kan zijn een gebruiker een eigen home-directory te geven, maar dat er voor de meeste Windows-gebruikers geen enkele reden is in te kunnen loggen met een Shell-account op de Linux-server.

Als tweede onderdeel moet een aantal shares aangemaakt worden. Om te beginnen is dat de share [netlogon]. De gebruiker doet hier verder praktisch weinig mee, maar de share moet bestaan om een succesvolle domain-logon te kunnen doen. Daarnaast kan deze share gebruikt worden om logon-scripts en systeem policy-bestanden te plaatsen. Let er wel op dat de share niet toegankelijk en zeker niet beschrijfbaar hoeft te zijn voor gebruikers, om die reden hebben zowel de parameter writable als browsable de waarde no.

De tweede share die wordt aangemaakt, is de locatie waar de profielen van de gebruikers geplaatst kunnen worden. Dit zijn de zogenaamde roaming profiles die ervoor zorgen dat de gebruiker op elke pc waarop hij zich aanmeldt gebruik kan maken van dezelfde desktopinstellingen. Deze share is nodig voor gebruikers die vanaf een Windows NT, 2000 of XP werkstatiohn binnenkomen. De share hoeft niet browsable te zijn, maar hij moet wel beschrijfbaar zijn voor de gebruikers. Anders zouden gebruikers immers niet in staat zijn om wijzigingen in hun profiel op te slaan. De instellingen voor het create mask en het directory mask zorgen ervoor dat de gebruiker als enige toegang heeft tot zijn profiel.

Als laatste wordt er een share gedefinieerd voor de home-directory van de gebruiker. Deze share is nodig voor de verwijzingen die gedaan worden door de parameters logon drive en logon path. Houd er rekening mee dat gebruikers ook daadwerkelijk op het lokale Linux-bestandssysteem een home-directory moeten hebben, anders werkt het nog niet. Dat betekent dat het voorbeeld dat eerder gegeven is voor de parameter add user script in dit geval niet handig is. De optie –d /dev/null zorgde er immers voor dat het null-device als home-directory gebruikt werd. Om te kunnen werken met profielen, is het om die reden aan te raden gebruik te maken van een add user script dat wel een home-directory aanmaakt. De betreffende regel zou er dan uit kunnen komen te zien als add user script = /usr/sbin/useradd –m –g 100 –s /bin/false %u. Hierbij zorgt de optie –m ervoor dat op de standaard locatie een home-directory wordt aangemaakt die dezelfde naam heeft als de gebruiker zelf.

Wanneer u alle hierboven beschreven wijzigingen hebt aangebracht, kunt u met de opdracht testparm controleren of er geen fouten voorkomen in smb.conf. Als dit het geval is, kunt u de server opnieuw starten. Gebruik hiervoor als root de opdracht /etc/rc.d/init.d/smb restart.

Tip! Wanneer u de opdracht /etc/init.d/smb restart uitvoert, weet u zeker dat de nieuwe configuratie meteen wordt geactiveerd. Het kan ook eenvoudiger: gewoon een minuutje wachten, dat zorgt Samba er in versie 3 voor dat de wijzigingen vanzelf geactiveerd worden.

6.2.4.3 Aanpassen van de configuratie van de werkstations

Op basis van het voorgaande hebt u ervoor gezorgd dat uw Samba-server zich als PDC gedraagt. U bent er echter nog niet helemaal. Om gebruik te maken van de Samba-PDC moet

namelijk voor elke computer een computeraccount worden aangemaakt. Hiervoor zijn drie stappen nodig. Als eerste moet u in /etc/passwd een account aanmaken voor elke computer die in uw netwerk voorkomt. Vervolgens moeten deze computer-account met de opdracht **smbclient** worden toegevoegd aan de database waarvan de Samba-server gebruik maakt. Als dat gedaan is, moet tot slot op de Windows-computer een aantal instellingen gewijzigd worden om ervoor te zorgen dat deze aanmeldt op het domein dat u op de Samba-PDC gedefinieerd hebt.

Stap 1: De computeraccounts aanmaken
Om computeraccounts aan te kunnen maken, hebt u een domain-administrator nodig. Het is niet voldoende dat u op uw lokale Linux-systeem al een gebruiker hebt met de naam root, u moet op uw Samba-server een soortgelijke gebruiker hebben. Voordat u computers aan kunt maken, moet u eerst deze gebruiker definiëren. Dit doet u met de opdracht **smbpasswd –a root**. Als dit gebeurd is, kunt u de computeraccounts aanmaken in de lokale gebruikersdatabase op uw Linux server. Hiervoor maakt u gebruik van de NetBIOS-namen zoals deze bekend zijn op uw Windows-werkstations. U kunt deze naam als volgt achterhalen:

1. Klik met de rechter muisknop op Deze Computer;
2. Selecteer de optie Eigenschappen;
3. Activeer nu het tabblad Computernaam. Deze naam hebt u nodig.

Nu moet u deze computernaam toevoegen aan de gebruikersdatabase op uw Samba-server. Houd daarbij rekening met de volgende zaken:

* Elke computer moet een unieke UID hebben. U kunt dit bereiken door de computeraccounts toe te voegen met een opdracht als **useradd**, deze opdracht zorgt er namelijk voor dat UID's automatisch gegenereerd worden.
* Ook computers moeten lid zijn van een groep. Het is aan te raden voor dit doel een aparte groep aan te maken in /etc/group.
* Zorg ervoor dat de computers geen wachtwoord, home-directory of Shell hebben. Als dit wel het geval is, zou een hacker er namelijk misbruik van kunnen maken.
* De naam van elke computer moet eindigen op een dollar-teken. Als dit niet het geval is, kan het account niet als computeraccount herkend worden.

Om bijvoorbeeld de computer x-tina een computeraccount te geven op de Samba-server, zou u de opdracht **useradd –d /dev/null –g 400 –s /bin/false x-tina$** kunnen gebruiken. Hierbij wordt er overigens wel vanuit gegaan dat op dat moment reeds een groep met GID 400 op het systeem aanwezig is. Gebruik hiervoor eventueel de opdracht **groupadd**.

Stap 2: Voeg de computeraccounts toe aan de Samba-gebruikersdatabase
U herinnert zich wellicht dat het niet voldoende is om een lokale gebruiker aan te maken op een Linux-systeem als u met de Samba-server wilt werken. Windows computers en gebruikers melden zich namelijk aan op een manier die door het Linux login-mechanisme niet begrepen wordt. In jargon heet het dat Windows gebruikmaakt van andere credentials. Om die reden moet er wanneer op de Samba-server de parameter security=user gebruikt wordt, een aparte gebruikersdatabase gedefinieerd worden op de Samba-server. In deze gebruikersdatabase moeten ook de Windows-computeraccounts voorkomen. Om dit te doen, maakt u gebruik van de opdracht **smbclient**. Om de computer uit het voorgaande voorbeeld toe te voegen aan deze database, gebruikt u bijvoorbeeld de opdracht **smbclient –m –s x-tina**. In dit commando geeft de optie –m aan dat het om een computeraccount gaat, de optie –s zorgt ervoor dat eventuele

wachtwoorden vanaf de console ingevoerd kunnen worden. Let erop dat in dit geval de naam van het computeraccount niet beëindigd hoeft te worden met een dollar-teken, u hebt immers –m gebruikt om aan te geven dat u een computeraccount wilt aanmaken.

Stap 3: Windows aanpassen voor aanmelden op het domein
De laatste stap tot slot bestaat eruit Windows aan te passen om aan te laten melden op het domein. Let even op: het is hierbij belangrijk welke Windows versie u gebruikt. De werkwijze is namelijk per Windows versie verschillend. Ook moet u er rekening mee houden dat Windows XP Home Editie geen mogelijkheid heeft aan te melden op een domein. Alle andere Windows-versies sinds Windows 98 hebben deze mogelijkheid wel. Als op een of meerdere computers in het netwerk gebruikgemaakt wordt van Windows XP Home Editie, zou u dus kunnen overwegen hier een ander besturingssysteem te installeren. Gebruik bijvoorbeeld een Linux desktop besturingssysteem want Linux kan uitstekend als Samba-client geconfigureerd worden. U kunt hier lezen hoe u Windows 98 en Windows XP Professional aan kunt laten loggen op een domein.

Windows 98
1. Open het Windows Configuratiescherm en open hier de optie Netwerk.
2. Klik de optie Client voor Microsoft-netwerken aan en klik vervolgens op Eigenschappen.
3. Selecteer nu de optie Aanmelden bij Windows NT-domein. Vervolgens moet u in het kader Windows NT-domein de naam opgeven van het domein waarop u aan wilt melden. Dit is de naam die u in smb.conf gespecificeerd hebt bij de optie workgroup.
4. Start Windows 98 opnieuw op. Er verschijnt nu een logon-venster waarin u aan kunt geven bij welk domein u aan wilt melden.

Windows XP Professional
1. Klik op Start en selecteer vervolgens het configuratiescherm. Zorg ervoor dat hier de klassieke weergave actief is.
2. Dubbelklik op Systeem om de eigenschappen van uw computer te openen.
3. Activeer nu het tabblad Computernaam en klik hier op Network ID. De Network Identification Wizard wordt nu gestart. Klik op Volgende om te beginnen.
4. Selecteer nu de optie Deze computer maakt deel uit van een zakelijk netwerk en klik op Volgende.
5. Nu kiest u de optie Mijn computer gebruikt een netwerk met een domein. Lees het overzichtsscherm en klik op Volgende.
6. Er verschijnt nu een scherm waarin u zichzelf als gebruiker bekend kunt maken. Hier moet u eenmalig inloggen als domain administrator. Dit is de gebruiker root die u in het begin van de procedure hebt opgegeven. Voer het wachtwoord van deze gebruiker in en de naam van het domein waarop u aan wilt maken.
7. Start nu de computer opnieuw op. Nadat dit gebeurt is, verschijnt een domein-loginprompt waarmee u zich kunt aanmelden op het domein.

6.2.5 Een Samba-server integreren in Active Directory
Samba kan ingericht worden als PDC of BDC. Over die eerste mogelijkheid hebt u in het voorgaande kunnen lezen. Sinds versie 3.0 van de Samba-server kan een Samba-server ook als Member-server opgenomen worden in een Windows netwerk waarin Active Directory gebruikt wordt. Let wel even goed op voordat u overenthousiast aan het werk gaat: u hebt nog steeds Windows 2000 servers nodig die de rol van Domain Controller vervullen. Het enige wat u met Samba kunt doen, is de Samba-server lid maken van Active Directory. Dit betekent

dat gebruikers kunnen authenticeren op Active Directory en vervolgens toegang kunnen krijgen tot gedeelde resources op de Samba-server. Dit is vooral handig wanneer Samba in een netwerk naast Windows-servers gebruikt wordt. Om een Samba server in te haken op een Active Directory omgeving, moet u hem eerst lid maken van het domein.

6.2.5.1 Samba lid maken van een Active Directory Domain

Om Samba te integreren met Active Directory, moet u de Samba-server lid maken van het domein. In feite is dit een simpele procedure die slechts uit vier stappen bestaat. Aangezien heel veel al in het voorgaande besproken is, kunt u in het onderstaande bondig lezen hoe u te werk moet gaan.

1. Maak een Active Directory gebruiker op de Windows 2000 server aan. Wat we uiteindelijk willen, is immers als Active Directory gebruiker toegang krijgen tot een share die op een Samba-server gedefinieerd is. Dit betekent dat u alle Samba-gebruikers in Active Directory aan moet maken. AD is het primaire mechanisme waarop gevalideerd wordt.
2. Zorg ervoor dat de Samba-gebruiker als Linux-gebruiker bestaat en regel de permissies. Zoals onder alle omstandigheden geldt, moet u er ook in dit scenario voor zorgen dat op de Samba-server een gebruiker bestaat die permissies heeft op de share die u wilt delen. Dit betekent dat u hem in /etc/passwd aan moet maken en dat hij ook permissies moet krijgen. U kunt als alternatief ook gebruikmaken van een van de opties om dit account automatisch aan te laten maken, in kleine omgevingen echter is het zinnig dit handmatig te doen omdat u er dan meer controle over hebt.
3. Pas smb.conf aan. Uiteraard moeten ook in dit geval een paar regels toegevoegd worden aan smb.conf. Dit bestand is immers de enige locatie waar u de totale configuratie van de Samba-server regelt. Het gaat hier om de volgende regels:

```
[global]
        workgroup = NWTRADERS
        password server = IPadres van je 2000 server
        security = ADS
```

Met de eerste regel maakt u de Samba-server lid van het Active Directory Domain dat in dit geval NWTRADERS heet. Vervolgens verwijst u met de regel password server naar de server waarop de authenticatie moet worden afgehandeld. Gebruik hiervoor het IP-adres van uw Windows 2000 Domain Controller. Tot slot geeft u aan dat nu eens voor de verandering gebruik moet worden gemaakt van ADS (Active Directory Services) om de beveiliging af te handelen.

4. Voeg uw Samba-server toe aan Active Directory. Net als in het geval waar u Samba als PDC configureert, hebt u ook bij het gebruik van Active Directory een computeraccount nodig. Dit computeraccount moet vanaf de Samba-server worden aangemaakt in Active Directory. Hiervoor maakt u gebruik van het zeer veelzijdige commando **net**. Met dit commando kunt u toveren vanaf een Samba-server, als u zich eens verveelt is het absoluut de moeite waard hier eens de man-pagina van door te bladeren. In dit geval gebruikt u de opdracht **net ads join -U Administrator%password**. Zoals te raden valt, maakt u hiermee een computer-account aan voor uw server in AD. Let erop dat achter de optie –U naam en wachtwoord gebruikt worden van een beheerdersaccount in Active Directory dat voldoende permissies heeft om daar een computeraccount aan te maken. Het resultaat van dit alles is dat u in Active Directory een computer account voor uw Samba-server ziet verschijnen.

Uw Samba-server is nu lid van Active Directory. Het resultaat hiervan is dat u met uw credentials uit de Windows-omgeving shares kunt benaderen die door de Samba-server worden aangeboden.

6.2.5.2 Samba met Kerberos authenticeren op het domein

In het voorgaande hebt u gebruikgemaakt van normale wachtwoord-authenticatie om de Samba-server toe te voegen aan het Active Directory domain. Het kan ook beter: wanneer u gebruik maakt van Kerberos, vindt de authenticatie op een veel slimmere manier plaats. Deze werkwijze is als eerste slimmer omdat gebruikgemaakt wordt van encryptie en Kerberos-tickets ter identificatie. Het voordeel hiervan is dat er geen wachtwoorden uitgelezen kunnen worden wanneer deze over het netwerk verstuurd worden. Kerberos-authenticatie zorgt er namelijk voor dat er helemaal geen wachtwoorden verstuurd hoeven worden. Om dit voor elkaar te krijgen voert u drie stappen uit.

1. Pas smb.conf aan. Deze keer gaat het om de volgende regels. Het belangrijke verschil met het voorgaande is dat er een Kerberos-realm toegevoegd worden dat nodig is voor het uitwisselen van Kerberos-tickets.

```
[global]
        workgroup = NWTRADERS
        realm = NWTRADERS.MSFT
        ads server = IPadres van je 2000 server
        security = ads
```

2. Haal de initiële Kerberos tickets op. Wanneer smb.conf op de juiste wijze is aangemaakt, moet u ervoor zorgen dat de benodigde Kerberos-tickets op de Samba-server bekend zijn. Dit doet u met behulp van de opdracht **kinit administrator@NWTRADERS.MSFT**.
3. Voeg de Samba-server toe aan Active Directory. Hiervoor gebruikt u de opdracht **net ads join**. Dit zorgt ervoor dat uw server wordt toegevoegd aan AD. Het voordeel van dit alles? Op de achtergrond kan nu gebruikt worden van het geavanceerde protocol Kerberos en het is niet langer nodig dat voortdurend wachtwoorden heen en weer gestuurd worden. Op deze wijze wordt er veel veiliger geauthenticeerd.

6.2.5.3 De wereld op zijn kop: Linux gebruikers loggen in op Active Directory

Het gaat er in dit hoofdstuk om dat Windows-gebruikers gebruik kunnen maken van gedeelde bronnen op een Samba-server. Dit betekent dat u op uw Linux-server een voorziening moet treffen die het voor de Windows-gebruikers mogelijk maakt daarop in te loggen. Dit scenario leent zich vooral in een omgeving waar op de servers voornamelijk gebruikgemaakt wordt van Linux als besturingssysteem. Soms echter is het tegenovergestelde het geval en komt er binnen een omgeving die vooral uit Windows-servers bestaat een Linux-server voor. In zo'n omgeving zullen alle gebruikersaccounts bijgehouden worden in Active Directory. In principe moet u voor gebruikers die ook iets moeten doen op de Linux-computer een apart account aanmaken op deze computer. Niet echt handig, want dat geeft u als beheerder dubbel werk. Gelukkig is het met behulp van winbind mogelijk om dit dubbele werk te voorkomen. U maakt dus in Active Directory een gebruiker aan en zorgt ervoor dat u vanaf een Linux-machine als deze gebruiker aan kunt melden zonder dat de gebruiker ook daadwerkelijk op de Linux-server bestaat. Voer hiervoor de onderstaande stappen uit.

1. Maak de gebruiker aan in Active Directory

2. Pas nsswitch.conf aan. In het configuratiebestand /etc/nsswitch.conf wordt bepaald in welke volgorde belangrijke configuratiebestanden gebruikt moeten worden. Dit configuratiebestand stampt uit het tijdperk dat nog niet zo intensief gebruikgemaakt werd van PAM. Voor services die niet met PAM overweg kunnen, moet u er in dit bestand voor zorgen dat tijdens de authenticatie eerst gekeken wordt naar het winbind-mechanisme en pas daarna van de traditionele UNIX-gebruikersbestanden. U kunt dit aangeven door de onderstaande regels op te nemen in nsswitch.conf

```
Passwd:     winbind compat
Group:      winbind compat
```

3. Wijzig de PAM-configuratie. De meeste moderne services maken niet langer gebruik van nsswitch.conf, maar van het PAM-mechanisme waarover u eerder in dit boek hebt kunnen lezen. Om ervoor te zorgen dat gebruikers die middels PAM inloggen zich aan kunnen melden op Active Directory, neemt u in het begin van het configuratiebestand een regel op die ervoor zorgt dat iedereen die zich aanmeldt op Active Directory ook welkom is op het Linux-systeem. Gebruik hiervoor de volgende regel:

```
auth    sufficient      pam_winbind.so
```

De bovenstaande regel moet als allereerste regel worden opgenomen in het bestand /etc/pam.d/login. Vergeet ook niet te controleren of de PAM-module pam_winbind.so wel op uw systeem voorkomt.

4. Pas smb.conf aan. U maakt straks gebruik van de Winbind-service. Deze service is gerelateerde aan de Samba-server. Dit betekent dat de totale configuratie van deze service ook geregeld wordt in het Samba-configuratiebestand smb.conf. U regelt dit met de onderstaande parameters. Deze parameters zorgen ervoor dat voor elke gebruiker die via het Winbind-mechanisme binnenkomt, automatisch de juiste credentials op het systeem worden aangemaakt. U kunt de regels uit het voorbeeld hieronder gewoon letterlijk overnemen. Voor meer uitleg over de exacte betekenis van deze regels kunt u de man-pagina van smb.conf raadplegen.

```
[global]
        winbind separator = +
        winbind cache time = 15
        template shell = /bin/bash
        idmap uid = 10000-20000
        idmap gid = 10000-20000
        template homedir = /home/%U
        winbind uid = 1000-20000
        winbind gid = 1000-20000
        workgroup = NWTRADERS
```

5. Start winbindd. Wanneer alle noodzakelijke voorzieningen getroffen zijn, moet u ervoor zorgen dat Winbind actief wordt op uw Linux-server. Hiervoor maakt u gebruik van de Winbind-daemon (winbindd). U kunt deze service starten vanuit /usr/sbin, u kunt hem ook activeren met behulp van het script dat voorkomt in de directory /etc/init.d. Wanneer dit gebeurd is, voldoet u aan alle voorwaarden en kunt u op de Linux-prompt inloggen op Active Directory.

6. Controleer de werking. Als u alle bovenstaande stappen doorlopen hebt, kunt u uitproberen of het werk. Dit doet u door op de Linux loginprompt in te loggen als een Active Directory-gebruiker. Let even op de syntaxis die daarvoor nodig is: u geeft als eerste de naam van het domein waarop u inlogt, vervolgens een + en tot slot de naam van de gebruiker als wie u wilt inloggen. Dat betekent dat gebruiker melissa uit het domein NWTRADERS zich aanmeldt als NWTRADERS+melissa. Vindt u het +-teken niet elegant? Geen nood: dit is heel eenvoudig te wijzigen. In smb.conf wordt met behulp van de parameter winbind separator bepaald welk teken gebruikt moet worden om de domeinnaam en gebruikersnaam van elkaar te scheiden. Wilt u hiervoor een uitroepteken, &-teken of wat dan ook gebruiken, dan regelt u dat in een handomdraai met deze parameter.
7. Problemen? Lukt het niet op de manier die hiervoor beschreven is? Verzeker u er dan van dat de Linux-server waarop u inlogt wel een computeraccount heeft in Active Directory. Is dit niet het geval? Geef dan op de Linux-server de opdracht **net ads join** om hem alsnog lid te maken van AD.

6.2.6 Authenticatie op een andere server

In het voorgaande hebt u gelezen over de werkwijzen die u kunt volgen om te authenticeren op iets anders als de locale Samba-gebruikersdatabase. Deze authenticatiemethode wordt geregeld met behulp van de parameter security = in de sectie [global] van smb.conf. Er is nog een werkwijze die we niet besproken hebben. Het is namelijk ook mogelijk te authenticeren op een andere server. Hiervoor wordt gebruikgemaakt van de parameter security = server. Met behulp van deze instelling kunt u zich bijvoorbeeld aanmelden bij een andere Samba-server, of bij een Windows NT-server. Het nut daarvan ligt voor de hand. Wanneer in uw netwerk gebruikgemaakt wordt van meerdere Samba-servers, voorkomt u met deze parameter dat u op elke Samba-server een aparte Samba-gebruikersdatabase moet aanmaken. Naast de parameter security = server, moet u natuurlijk ook aangeven op welke server er dan geauthenticeerd moet worden. Gebruik hiervoor de optie password server =. In het onderstaande ziet u een voorbeeld van twee regels die voor dit doel opgenomen kunnen worden in smb.conf.

```
[global]
        security = server
        password server = laksmi damayanti
```

Houd er rekening mee dat u met behulp van deze instelling voor Samba-gebruikersnamen gaat informeren op een andere server. Het is echter nog steeds noodzakelijk dat op elk van die servers wel een Linux gebruiker bestaat met de benodigde permissies op het Linux bestandssysteem. Als u geen zin hebt deze Linux-gebruikers overal handmatig aan te maken, zou u gebruik kunnen maken van een service als NIS of LDAP. Hierover leest u elders in dit boek meer.

6.2.7 Grafische hulpprogramma's voor configuratie

In het voorgaande hebt u kunnen lezen over de manier waarop Samba geconfigureerd kan worden. Hierbij hebben we vrijwel uitsluitend gekeken naar de beheersmogelijkheden die geboden worden door het configuratiebestand smb.conf direct aan te passen. Dit is echter niet de enige mogelijkheid. Als onderdeel van de Samba-server wordt standaard ook de component SWAT (Samba Web Administration Tool) meegeleverd. Daarnaast worden door de verschillende distributies ook de nodige mogelijkheden geboden. Deze tools staan echter in geen verhouding tot de mogelijkheden die SWAT biedt. Waar u met SWAT gewoon alle instellingen vanuit een grafische interface kunt regelen, beperken zowel Fedora als SUSE zich tot grafische hulpprogramma's waarmee alleen het hoogstnodige kan.

Voordat u SWAT kunt gebruiken, zult u het eerst aan moeten zetten. De mogelijkheid de Samba-server met behulp van SWAT te beheren, wordt geleverd vanuit inetd of xinetd. Dit betekent dat u ervoor moet zorgen dat de service vanuit dit mechanisme ook beschikbaar is. U moet in elk geval het bestand /etc/services aanpassen om de SWAT-service bekend te maken. De volgende stap is afhankelijk van de configuratie die u gebruikt: ofwel u definieert in inetd.conf een regel om de service te starten, of u zorgt ervoor dat er een SWAT-configuratiescript aanwezig is voor xinetd. Wanneer dit gebeurt is, moet u inetd of xinetd opnieuw starten. Vervolgens is SWAT klaar voor gebruik.

1. /etc/services aanpassen

Om SWAT te kunnen starten, moet de service bekend zijn. Pas hiervoor het configuratiebestand /etc/services aan en neem er de volgende regel in op:

```
swat    901/tcp
```

2a. Pas inetd.conf aan

Wanneer u gebruik maakt van inetd, moet u het configuratiebestand /etc/inetd.conf aanpassen om ervoor te zorgen dat SWAT benaderd kan worden. Neem hiervoor de volgende regel op in dit bestand:

```
swat    stream tcp      nowait root     /usr/sbin/swat swat
```

In de meeste gevallen zal deze regel uitstekend zijn werk doen, het kan echter soms nodig zijn om de locatie van het SWAP-programmabestand aan te passen. Nadat u inetd.conf aangepast hebt, moet de service inetd opnieuw gestart worden. Gebruik hiervoor de opdracht killall –HUP inetd. SWAT is nu beschikbaar.

2b. Pas xinetd aan

Het verschil tussen inetd en xinetd is dat bij xinetd elke service zijn eigen configuratiebestand krijgt. Dit bestand wordt geplaatst in de directory /etc/xinetd.d. Op SUSE Linux ziet het configuratiebestand voor SWAT er standaard als volgt uit:

```
service swat
{
        socket_type     =       stream
        protocol                =       tcp
        wait            =       no
        user            =       root
        server          =       /usr/sbin/swat
        only_from       =       127.0.0.1
        log_on_failure+=        USERID
        disable         =       yes
}
```

Let in dit voorbeeld vooral even op de laatste regel: deze zorgt ervoor dat SWAT standaard uit staat. U moet er op zijn minst voor zorgen dat deze regel gewijzigd wordt in disable = no. Ook de regel only_from is interessant. Deze zorgt er standaard voor dat SWAT alleen vanaf de locale computer gebruikt mag worden. Wilt u SWAT ook vanaf andere locaties op het netwerk kunnen gebruiken, wijzig deze regel dan bijvoorbeeld in only_from = 0.0.0.0.

Hiermee maakt u toegang vanaf elke andere computer mogelijk. Nadat u het SWAT-configuratiebestand voor xinetd gemaakt of aangepast hebt, moet ook xinetd opnieuw gestart worden. Gebruik hiervoor de opdracht **killall –HUP xinetd**, dit forceert het proces xinetd om zijn configuratie opnieuw in te lezen.

Als de bovenstaande stappen met succes zijn uitgevoerd, kunt u SWAT gebruiken. Voordat u toegang krijgt, ziet u eerst een loginscherm waarin u zich bekend moet maken. U kunt hier inloggen als Linux-gebruiker root. Nadat dit gebeurd is, komt u in de SWAT-interface terecht. U ziet hier in één grafisch programma alles wat u voor de Samba-server in kunt stellen. Veel van de informatie die u hier kunt regelen zal u op basis van de informatie in dit hoofdstuk bekend voorkomen, er zijn echter ook nog heel veel opties die in dit hoofdstuk niet besproken zijn. Wij raden u aan vooral uitgebreid rond te kijken in de mogelijkheden die geboden worden, SWAT is overzichtelijk en biedt veel informatie over alles wat mogelijk is. Houd er overigens rekening mee dat SWAT werkt met twee verschillende weergaven. U komt standaard in de basis-weergave waarin u niet alle opties ziet. Om wel alle mogelijkheden met SWAT aan te kunnen passen, maakt u gebruik van de geavanceerde weergave.

*** swat Met behulp van SWAT kunt u alle instellingen van de Samba-server vanuit een overzichtelijke Web-interface aanpassen.

Oefening
Om deze oefening uit te kunnen voeren, hebt u twee computers nodig. De ene server wordt uitgerust met een Samba-share, de andere server wordt voorzien van de Samba gebruikersdatabase.
Maak op uw server een directory met de naam /share. Zorg ervoor dat deze directory gedeeld wordt en na delen toegankelijk is voor alle gebruikers waarvoor een account bestaat op uw lokale Linux-computer. U onderhoudt echter de Samba-gebruikersaccounts niet lokaal, maar op uw andere server. Test met behulp van het commando mount dat uw Samba-server toegankelijk is.

Samenvatting
In dit hoofdstuk hebt u gelezen over de wijze waarop u bestanden op uw Linux-server beschikbaar kunt stellen aan andere gebruikers. Hierbij is aandacht besteed aan twee belangrijke systemen. Allereerst hebt u gelezen over de wijze waarop met NFS bestanden gedeeld kunnen worden, vervolgens hebt u gelezen hoe u een Samba-server in kunt zetten om bestanden toegankelijk te maken voor Windows, Linux en Apple gebruikers. U hebt kennisgemaakt met de belangrijkste methodes die gebruikt kunnen worden om met Samba bestanden te delen: Samba als stand-alone server, Samba als PDC en Samba als deel van Active Directory.

Oefenvragen
1. Hoe heet het proces dat u nodig hebt om als Linux gebruiker in te loggen op Active Directory?
2. Welke processen zijn nodig om een NFS-share te kunnen maken?
3. Hoe ziet de regel eruit waarmee u een NFS-share aanmaakt die toegang geeft aan alle computers op de hele wereld.
4. Met welk commando maakt u een Samba-server member van een domain?
5. Hoe heet het bestand waarin Samba gebruikers worden opgeslagen?
6. Met welke opdracht maakt u een Samba-mount naar de share //server/share vanaf uw lokale computer?

7. Met welke regel in smb.conf zorgt u ervoor dat authenticatie plaatsvindt op een PDC?
8. Met welke regel in smb.conf zorgt u ervoor dat gebruikers alleen mogen lezen in een directory?
9. Hoe heet het bestand waarin lokale Samba-gebruikers worden opgeslagen?
10. Hoe zorgt u ervoor dat Windows gebruikers de tijd ophalen bij de Samba server in het netwerk?

Hoofdstuk 7: Linux Server beveiliging

Ooit was er een periode waarin u zich als beheerder van een server over beveiliging nauwelijks zorgen behoefte te maken. Die periode is zeer lang geleden; op het moment dat de eerste server aan internet verbonden werd (we hebben het dan over 1969!) werd beveiliging ook gelijk een issue.
De beveiliging van een Linux systeem kent vele facetten. Om te beginnen zorgt u ervoor dat alleen die services actief zijn die ook actief moeten zijn en hebt u bij het starten van die services gelijk de mogelijkheid een elementaire beveiliging te regelen. Vervolgens zorgt u ervoor dat er veilig gecommuniceerd kan worden, door gebruik te maken van Secure Shell, VPN of Kerberos. Dan zorgt u ervoor dat uw server voorzien is van een firewall waarin op pakket-niveau bepaald wordt wat wel mag en wat niet mag. Alsof dat nog niet genoeg is, zorgt u er daarna voor dat uw server op een veilige manier log-berichten wegschrijft, bij voorkeur naar een andere server. Als dat dan allemaal draait en uw systeem in principe veilig zou moeten zijn, bent u er nog niet: op dat moment moet u namelijk nadenken over pro-actief beheer. In dit geval houdt dat in dat u actief als beheerder in de gaten houdt wat er op uw netwerk, maar ook op internet aan beveiligingsrisico's voorkomt en dat u actie onderneemt wanneer dat nodig is.

Leerdoelen:
* Beveiligign van services door middel van xinetd en tcpd
* Onveilige services beveiligen
* Veilig log-bestanden wegschrijven op remote servers
* Uw server beveiligen met een firewall
* Veilig communiceren met uw server
* Pro-actief beheer

7.1 TCP wrappers

Er zijn voor Linux twee "superservices" beschikbaar. Dit zijn programma's die diensten aan het netwerk beschikbaar stellen waar vervolgens weer andere programma's gebruik van kunnen maken. Als eerste is dat xinetd, de daemon die er voor zorgt dat een aantal andere services gestart kunnen worden. Soms wordt gebruikgemaakt van een oude voorloper van xinetd genaamd inetd. De tweede superservice is tcpd, die er voor zorgt dat een zekere mate van beveiliging kan worden toegepast; er kan algemeen worden aangegeven of hosts al dan niet toegang krijgen en het is mogelijk hetzelfde per individuele service te doen. Stel u daarbij overigens niet al te veel voor, serieuze beveiliging van een server word namelijk op een andere wijze geregeld. Hierover leest u later in dit hoofdstuk meer.

7.1.1 inetd

Het programma inetd wordt ook wel aangeduid als de Internet Superserver. Deze service wordt opgestart vanuit de opstartscripts van uw server. Inetd luistert naar verzoeken die binnenkomen om gebruik te maken van een bepaalde service. Hiervoor worden poortnummers die bij bepaalde services horen afgevangen. Vervolgens wordt het pakketje doorgestuurd naar de betreffende service.
De configuratie van inetd wordt gedaan in het bestand /etc/inetd.conf. In dit bestand worden services gekoppeld aan bepaalde programma's.

*** inetd In inetd.conf wordt bepaald welke services allemaal via dit mechanisme benaderd kunnen worden.

Een deel van de beveiliging van een systeem bestaat er uit het configuratiebestand /etc/inetd.conf op de juiste wijze te bewerken. Op veel systemen zorgt dit bestand er namelijk voor dat services door de computer beschikbaar gesteld worden, terwijl deze soms helemaal niet nodig zijn. U kunt uw systeem veiliger maken door in inetd alleen die services te activeren die ook echt nodig zijn. Accepteer dus nooit de standaardinstellingen maar onderwerp het configuratiebestand inetd.conf eens aan een kritische blik, vooral als u van plan bent de server aan internet te verbinden.

/etc/inetd.conf
Het bestand /etc/inetd.conf levert de configuratie voor inetd. In dit bestand wordt gedefinieerd welke programma's gestart moeten worden om een service te activeren. Hieronder ziet u een aantal voorbeelden van regels waarmee dit gebeurt.

```
ftp     stream tcp nowait root /usr/sbin/tcpd in.ftpd
telnet  stream tcp nowait root /usr/sbin/tcpd in.telnetd
shell   stream tcp nowait root /usr/sbin/tcpd in.rshd -L
login   stream tcp nowait root /usr/sbin/tcpd in.rlogind
talk    dgram udp wait root /usr/sbin/tcpd in.talkd
ntalk   dgram udp wait root /usr/sbin/tcpd in.talkd
pop3    stream tcp nowait root /usr/sbin/tcpd /usr/sbin/popper -s
```

De regels in inetd.conf beginnen met de naam van de service die gestart moet worden. Vervolgens wordt in het "socket" veld beschreven welke soort pakketje het betreft. De waarde van dit veld wordt bepaald door het type service dat aangeboden wordt. In veel gevallen zal hier de aanduiding `stream' staan. Daarna wordt beschreven of het pakketje door middel van TCP of door middel van UDP verstuurd moet worden. Dan wordt aangegeven of de betreffende service moet wachten totdat er een reactie van een gebruiker gegeven wordt. Daarna wordt gespecificeerd van welke gebruikers- of groepspermissies de service gebruik maakt en als laatste wordt de opdracht gegeven waarmee het programma gestart kan worden. In het bovenstaande voorbeeld ziet u dat de programma's in de meeste gevallen gestart worden door de naam van het programmabestand op te geven als argument van tcpd. Hierdoor kan namelijk direct gebruikgemaakt worden van de beveiliging die door tcpd geboden wordt; hierover kunt u later in dit hoofdstuk meer lezen.

Het werk van de beheerder bestaat eruit ervoor te zorgen dat alle services die nodig zijn hier geactiveerd worden, maar dat services die een mogelijk risico vormen voor de beveiliging van de computer hier uitgezet worden. Als u geen idee hebt welke services wel en welke niet nodig zijn, doet u er goed aan alle regels in dit bestand aan het begin te markeren met een commentaarteken (#). Dit zorgt ervoor dat de service niet beschikbaar gesteld wordt. Pas wanneer u zeker weet dat u een service ook nodig hebt, verwijdert u het commentaar teken zodat de betreffende service weer bereikt kan worden..

7.1.2 xinetd

De daemon xinetd is een door veel distributies gebruikte variant op inetd zoals besproken is in de vorige paragraaf. Het verschil bestaat eruit dat xinetd de enige service is die gestart wordt tijdens het opstarten en luistert naar inkomende pakketjes voor alle poorten waarvoor het geconfigureerd is. Inetd daarentegen start de services alvast maar houdt ze slapend op de achtergrond; xinetd gaat dus efficiënter om met het beschikbare werkgeheugen. Daarnaast wordt de configuratie om services te starten bij xinetd op een andere wijze geregeld.

Inetd had als voornaamste doel om kostbaar werkgeheugen te sparen door services die even niet nodig zijn niet continue actief te laten zijn, xinetd heeft als doel om services flexibel beschikbaar te stellen waarbij gebruik van de service weggeschreven kan worden in log-bestanden en waarbij gedefinieerd kan worden welke computers wel en welke computers geen toegang krijgen tot de service. Daarnaast wordt bij inetd gebruikgemaakt van één algemeen configuratiebestand, terwijl bij xinetd voor elke service een afzonderlijk configuratiebestand beschikbaar is. In deze configuratiebestanden kan veel nauwkeuriger worden aangegeven hoe een service gestart moet worden. Dit gebeurt door te werken met verschillende opties.

Er worden door xinetd twee soorten services gestart. De multi-threaded services zijn services waarvan er voor elke inkomende connectie een gestart wordt; dit betekent dus dat als er tien connecties zijn naar service X, dat deze service ook daadwerkelijk tien keer gestart wordt. De rol van xinetd is dat het er verantwoordelijk voor is dat deze services daadwerkelijk gestart worden. Xinetd is dus eindverantwoordelijke voor de connecties. Daarnaast zijn er single-threaded services. Hiervoor geldt dat er slechts één instantie van de service gestart wordt en dat deze service zelf verantwoordelijk is voor alle connecties. Dit verschil tussen multi-threaded en single-threaded services komt overigens ook voor bij services die door middel van een aparte daemon gestart worden. Het kan zelfs voorkomen dat een bepaalde service op de ene distributie als multi-threaded behandeld wordt terwijl er op een andere distributie maar een enkele versie van gestart wordt. Hoe een bepaalde service uiteindelijk gestart wordt, wordt door de distributie die u gebruikt bepaald. Of in xinetd-service multithreader is of niet, wordt bepaald door het configuratiebestand dat ervoor zorgt dat de betreffende service gestart wordt. De configuratiebestanden waarvan xinetd-services gebruikmaken, bevinden zich in de directory /etc/xinetd.d.

De bestaansreden van een superservice was altijd dat op deze wijze voorkomen kon worden dat kostbare systeembronnen continue geclaimd werden door services die meestal toch niet in gebruik zijn. Inetd.conf loste dit probleem op door services pas te activeren op het moment dat ze daadwerkelijk nodig waren. Met de huidige systeemconfiguraties vervalt deze noodzaak echter voor een groot deel, want wie heeft er nu nog last van een ftp-daemon die nog geen 100 KB werkgeheugen in gebruik heeft? Het gebruik van xinetd heeft echter nog steeds voordelen boven het steeds automatisch starten van een daemon-proces:

* Informatie over gebruik van services kan gelogd worden;
* Er kan een zekere mate van beveiliging aan verbonden worden;
* Het is mogelijk om ook services die niet in /etc/services gedefinieerd zijn te gebruiken;
* Services kunnen op een uniforme wijze geactiveerd worden.

7.1.3 Configuratie van xinetd-services

De configuratie van xinetd kan op twee manieren plaatsvinden. Om te beginnen is het mogelijk één algemeen configuratiebestand aan te maken met de naam /etc/xinetd.conf. Vanuit dit configuratiebestand kunnen alle afzonderlijke services beheerd worden. Deze werkwijze is echter hoogst ongebruikelijk. Een tweede mogelijkheid is per service een apart configuratiebestand aan te maken en dit te plaatsen in de directory /etc/xinetd.d. Hieronder ziet u hoe /etc/xinetd.conf en een willekeurig bestand uit /etc/xinetd.d er uit kunnen zien.

```
#/etc/xinetd.conf
include /etc/xinetd.d/
```

```
defaults {      instances       =       60
                log_type        =       SYSLOG authpriv
                log_on_success          =       HOST PID
                log_on_failure=         HOST RECORD }

includedir /etc/xinetd.d
```

Dit configuratiebestand is niet echt schokkend. Er wordt door middel van de regels "include…" en "includedir" geregeld dat gebruik gemaakt kan worden van extra configuratiebestanden die voorkomen in de directory /etc/xinetd.d. Verder wordt bepaald op welke wijze meldingen in logbestanden weggeschreven moeten worden. Ook aardig is de regel "instances = 60"; hiermee wordt bepaald hoe vaak de service maximaal gestart kan worden.

Zoals gezegd, naast de algemene xinetd.conf zijn er configuratiebestanden voor de specifieke toepassingen die door xinetd beheerd worden. Deze komen standaard voor in /etc/xinetd.d. Dit betekent dat voor elke toepassing die u op wilt kunnen starten, een dergelijk bestand gemaakt moet worden. Hieronder ziet u hoe een dergelijk bestand er uit kan zien; weergegeven wordt het bestand waarmee toegang tot de telnet-service geregeld kan worden. (Houd er overigens rekening mee dat telnet onveilig is en daarom beter niet gebruikt kan worden)

```
service telnet
{
        flags           =       REUSE
        socket_type     =       stream
        wait                    =       no
        user                    =       root
        server          =       /usr/sbin/in.telnetd
        log_on_failure+=        USERID
}
```

De meeste van de bovenstaande parameters zouden u bekend voor moeten komen op basis van wat u al weet over inetd.conf, wat in de vorige paragraaf besproken is. Nieuw is de wijze waarop een waarde wordt gegeven aan bepaalde attributen. Zo is er bijvoorbeeld de definitie "socket_type = stream" waarmee het type socket wordt vastgesteld op de waarde "stream". Daarnaast is het ook mogelijk te werken met += en -=; daarbij wordt met += een waarde aan een lijst toegevoegd en wordt deze waarde er met -= weer van verwijderd. Op die manier is het mogelijk meerdere waarden aan een bepaalde optie te verbinden. Houdt er verder rekening mee dat de opties die beschikbaar zijn, per service anders zijn.
Een optie die wel algemeen is bij het gebruik van xinetd, is de optie "disable = yes". Het is gebruikelijk dat deze optie als standaard gebruikt wordt: bij de installatie van een service die van xinetd afhankelijk is, wordt gelijktijdig een script weggeschreven waarmee deze service gestart kan worden. In dit script staat echter standaard de optie "disable = yes" aan; dit zorgt ervoor dat de service niet automatisch ook direct gebruikt kan worden. Om gebruik van de betreffende service mogelijk te maken, moet u deze regel veranderen in "disable = no".

***disable De standaard instelling is dat alle services die vanuit xinetd gestart worden uit staan. Om ze te kunnen gebruiken, moet u ze eerst aan zetten.

7.1.4 Beveiliging

Zoals gezegd, met xinetd komt ook een mogelijkheid om iets aan beveiliging te doen. Hiertoe dienen twee attributen in de configuratiebestanden gedefinieerd te worden, namelijk only_from en no_access. Door middel van only_from kan aangegeven worden vanaf welke computers toegang tot de service toegestaan is, door middel van no_access wordt aangegeven vanaf welke computers toegang juist verboden is. De verwijzing naar computers gebeurt door de IP-adressen van de computer op te geven. Daarbij kan gewerkt worden met adressen als 220.0.0.0 om aan te geven dat toegang ontzegd wordt aan alle computers waarvan het IP-adres begint met 220. Een andere en meer algemene manier om de beveiliging van bepaalde services te regelen, is door gebruik te maken van tcpd. Tcpd controleert aan de hand van de instellingen in de configuratiebestanden /etc/hosts.allow en /etc/hosts.deny of een verzoek tot toegang tot een service positief of negatief afgehandeld moet worden. Daarnaast kunt u alle verzoeken die via tcpd binnenkomen laten loggen met het syslog-mechanisme.

Als een aanvraag voor een bepaalde service binnenkomt, ontvangt inetd of xinetd deze. Deze service zorgt er vervolgens voor dat tcpd gestart wordt (als dit tenminste in de configuratie voor de specifieke service geregeld is). Tcpd controleert aan de hand van de configuratiebestanden /etc/hosts.allow en /etc/hosts.deny of alles in orde is en pas als dat het geval is, start het de service in kwestie op. In principe kan tcpd dat voor alle services op uw systeem regelen. Om duidelijk te maken hoe services opgestart kunnen worden door tcpd, volgt nu een voorbeeld. Stel dat de service finger normaliter vanuit inetd wordt gestart met de regel

finger stream tcp nowait nobody /usr/sbin/in.fingerd in.fingerd

in het bestand /etc/inetd. Om ervoor te zorgen dat deze service in het vervolg gestart wordt door tcpd, verandert u deze regel in

finger stream tcp nowait nobody /usr/sbin/tcpd in.fingerd.

U ziet dat in deze werkwijze niet langer de daemon van het betreffende programma direct wordt aangeroepen, maar als argument van de wrapper-daemon tcpd.

Beperken van toegang tot tcpd-services

Naast de mogelijkheid om toegang tot de services te loggen met het syslog-mechanisme, bestaat het andere voordeel van gebruik van tcpd eruit dat de toegang tot de services beperkt kan worden door gebruik te maken van de bestanden /etc/hosts.allow en /etc/hosts.deny.

Om in deze bestanden toegang tot services te beperken, werkt u met namen of adressen van computers of groepen computers. Ook kan gebruikgemaakt worden van het statement ALL om toegang te verlenen of ontzeggen aan alle computers die contact proberen te maken. Deze computernamen en adressen kunnen op hun beurt weer aan de naam van een service verbonden worden. Als een treffer voorkomt in /etc/hosts.allow, wordt toegang verleend. De volgende basisregels worden gebruikt bij het doorlopen van beide bestanden:

* Als geen treffer voorkomt in /etc/hosts.allow, maar wel in /etc/hosts.deny, wordt toegang geweigerd.
* Als noch in /etc/hosts.allow, noch in /etc/hosts.deny een treffer voorkomt, wordt toegang verleend.

Hieruit blijkt dat de standaardsituatie waarin beide bestanden leeg zijn er dus voor zorgt dat iedereen toegang kan krijgen tot elke service. Op elke regel wordt eerst gedefinieerd om welke service het gaat, vervolgens wordt na de dubbele punt aangegeven welke computers hier toegang toe krijgen. Op basis van deze informatie, zou u het volgende voorbeeld moeten begrijpen. De eerste twee regels uit het voorbeeld hebben betrekking op /etc/hosts.allow, de volgende twee regels komen uit /etc/hosts.deny.

```
#/etc/hosts.allow:
ALL:LOCALHOST

#/etc/hosts.deny
ALL:ALL
```

In het bovenstaande voorbeeld wordt eerst toegang verleend aan `localhost' (alle services op uw eigen computer) tot alle services. Vervolgens wordt de toegang tot alle services voor alle andere clients geweigerd. Dit is een zeer eenvoudige manier om ervoor te zorgen dat toegang tot alle services die via het tcpd-mechanisme gestart worden, wordt tegengehouden. Houdt er echter rekening mee dat services die door middel van hun eigen configuratiebestand gestart worden gewoon bereikbaar blijven, tcpd heeft hier namelijk niets over te zeggen.

Zowel /etc/hosts.allow als /etc/hosts.deny kunnen bestaan uit meerdere regels tekst. Deze regels worden doorzocht in de volgorde waarin ze voorkomen. Als een van de regels een treffer oplevert, wordt niet meer verder gezocht. Zoals gezegd, komt op elke regel de naam van de betreffende service en de specificatie van de werkstations die toegang hebben voor. Daarnaast kunnen ook shell-commando's gespecificeerd worden. Hierdoor kunnen bijvoorbeeld informatieve berichten weggeschreven worden naar log-bestanden wanneer een regel een treffer oplevert. Dit is vooral interessant wanneer het een service betreft waar toegang toe verboden wordt. U ziet hier in de onderstaande regels een voorbeeld van:

```
#/etc/hosts.deny
in.telnetd: ALL EXCEPT .sandervanvugt.nl :\
        if [ %h != ".sandervanvugt.nl:" ]; then \
        echo "aanvraag van %d@%h: >> /var/log/boeven.log; \
        finger -s %c >> /var/log/boeven.log \
fi
```

Bovenstaande code verbiedt iedereen toegang tot de telnet service, behalve computers uit het domein sandervanvugt.nl. Als er toch iemand toegang vraagt tot de telnet-service, wordt door middel van de shell-code gekeken of dat dan iemand is uit het DNS-domein sandervanvugt.nl. Als dat niet zo is, worden meldingen weggeschreven naar het bestand /var/log/boeven. In deze meldingen wordt gebruik gemaakt van variabelen die verwijzen naar de naam van de gebruiker, de service en de host waar de geweigerde poging van heeft plaatsgevonden.

In de onderstaande tabel ziet u welke variabelen in hosts.allow en hosts.deny gebruikt kunnen worden en wat door deze variabelen weergegeven wordt.

%a	Het adres van de client
%A	het adres van de server
%c	zo uitvoerig mogelijke informatie over de client
%d	de naam van het proces

%h	de hostnaam van de client; indien niet beschikbaar het adres
%H	de hostnaam van de server; indien niet beschikbaar het adres
%n	de hostnaam van de client, indien niet beschikbaar de tekst “unknown”
%N	de hostnaam van de server; indien niet aanwezig de tekst “unknown”
%p	het PID van de betreffende daemon
%s	serverinformatie in de vorm daemon@host
%u	de gebruikersnaam van de client

Naast variabelen kunnen, zoals ook blijkt uit het bovenstaande voorbeeld, patronen gebruikt worden om te verwijzen naar groepen clients en servers. De volgende mogelijkheden worden hiertoe geboden:

* Een string die begint met een punt. Een DNS-naam levert een treffer op, als de laatste delen van de naam overeenkomen met het patroon. Als het patroon .sandervanvugt.nl gebruikt wordt, levert games.franck.sandervanvugt.nl dus een treffer op.
* Een tekenreeks die eindigt met een punt. Dit wordt gebruikt om te verwijzen naar elk IP-adres dat begint met deze tekenreeks. Zo levert 192.168.96.100 een treffer op als een verwijzing naar 192.168. gegeven wordt.
* Een tekenreeks die begint met een @ wordt geïnterpreteerd als verwijzing naar een NIS-netgroup (zie hoofdstuk 5). Als een computer lid is van deze netgroup, levert dat dus een treffer op.
* Een patroon in de vorm n.n.n.n/m.m.m.m wordt gebruikt om te verwijzen naar computers die voorkomen in een bepaald netwerk. 192.168.96.32/255.255.255.224 heeft dus betrekking op de IP-adressen 192.168.96.33 tot en met 192.168.96.62.

Naast de hierboven vermelde patronen, kan ook gebruik gemaakt worden van wildcards:

ALL Heeft betrekking op alles. Levert dus ook altijd een treffer op.
LOCAL Levert een treffer op voor elke computer waarvan de naam niet met een punt begint; dit worden namelijk geacht computers te zijn die voorkomen op het locale netwerk.
UNKNOWN Levert een treffer op voor elke gebruiker en host waarvan de naam of het adres niet achterhaald kon worden.
KNOWN Levert een treffer op voor elke gebruiker of computer waarvan naam of adres achterhaald kon worden.
PARANOID Levert een treffer op voor elke host waarvan de naam niet met het opgegeven adres overeenkomt.

Als laatste is het ook mogelijk om gebruik te maken van de operator EXCEPT; hiermee kan een uitzondering gemaakt worden op een bepaalde regel. U hebt hiervan een voorbeeld gezien in het voorgaande voorbeeld waarin de toegang tot iedereen geweigerd werd, behalve tot computers uit het domein sandervanvugt.nl.

U hebt in deze paragraaf kunnen lezen over het tcpd-mechanisme. Door middel van tcpd kunt u een beperkte mate van beveiliging toevoegen aan uw computer. Houdt er echter rekening mee dat deze wijze van beveiliging onvolledig is: het heeft namelijk geen betrekking op alle services die buiten tcpd om gestart worden. Toch is het de moeite deze beveiliging mee te nemen in uw lokale beveiligingsstrategie, een aantal services kan er immers veiliger mee gemaakt worden.

Oefening 7.1

Deze oefening kan op elke computer afzonderlijk uitgevoerd worden.
Zorg ervoor dat alleen computers uit uw lokale netwerk toegang krijgen via de tcpd-daemon.
Bepaal vervolgens dat alleen die services die u daadwerkelijk nodig hebt door xinetd gestart worden.

7.2 Remtoe Logging

Het ligt misschien niet direct voor de hand, maar ook het installeren en onderhouden van een goed log-systeem kan de beveiliging van een netwerk aanzienlijk ten goede komen. Het eerste belang van iemand die ongeoorloofd toegang heeft verworven tot uw computer, is zijn sporen uit te wissen. Dit kan hij doen door logbestanden waarin zijn activiteit gelogd staat te verwijderen. U hebt dan als beheerder van het systeem in kwestie wel door dat er iets gebeurd is, maar het is onmogelijk nog te achterhalen wie de boosdoender is geweest. U kunt dit oplossen door gebruik te maken van de mogelijkheid de logbestanden op een cetrale en goed beveiligde log-server op te slaan. Dit regelt u door het syslog-mechanisme op de juiste wijze te beveiligen.

7.2.1 Introductie in syslog

Het centrale proces dat in een Linux-omgeving gebruikt wordt om logging mogelijk te maken, is sysklogd. Dit proces voorziet in twee systemen die gebruikt worden om meldingen weg te schrijven in verschillende bestanden die voor dit doel worden aangemaakt in de directory /var/log:

* syslogd
* klogd

Van deze twee zorgt klogd ervoor dat berichten die door de kernel gegenereerd worden weggeschreven kunnen worden in logbestanden. Klogd kan dat als afzonderlijk proces doen, maar meestal wordt het niet op deze manier gebruikt, maar functioneert klogd als client van syslogd. Het algemene proces syslogd zorgt er in dat laatste geval voor dat de totale logging van een systeem afgehandeld wordt. Hiervoor wordt gebruikgemaakt van instellingen die gedaan worden in het configuratiebestand syslog.conf. In dit bestand wordt door middel van verschillende regels gedefinieerd wat er moet gebeuren wanneer een evenement van een bepaalde ernst zich voordoet. Hiervoor wordt gebruikgemaakt van verschillende facilities en priorities. Een facility definieert een aantal categorieën waarbij iets mis kan gaan, een priority definieert de ernst van het feit dat zich voordoet.

De volgende facilities kunnen gebruikt worden:

* auth Algemene zaken die met authenticatie te maken hebben.
* authpriv Wordt gebruikt door alle services die iets te maken hebben met systeem beveiliging of authenticatie. Wordt onder andere gebruikt voor alle aan PAM gerelateerde berichten en de SSH-daemon.
* cron Ontvangt berichten van de processen cron en at.
* daemon Wordt gebruikt door verschillende daemons die niet hun eigen facility hebben.
* kern Alle kernel berichten. Voor deze categorie wordt samengewerkt met klogd.
* lpr Berichten van het printersysteem.
* mail Voor berichten die iets te maken hebben met het mailsysteem. Houd er rekening mee dat een slecht functionerend mailsysteem ervoor kan zorgen dat in deze categorie heel snel heel veel berichten binnen kunnen komen.
* mark Kan gebruikt worden om af en toe een markeringssignaal weg te schrijven naar syslog. Dit maakt de resulterende logbestanden gemakkelijker te lezen.

* news Voor berichten van het nieuwssysteem. Ook in deze categorie kunnen bij problemen zeer snel zeer veel meldingen binnenkomen.
* security (zelfde als auth) Wordt niet langer gebruikt. De functionaliteit wordt nu overgenomen door auth.
* syslog Voor intern gebruik door het syslog proces.
* user Algemene categorie voor zaken die iets met gebruikers te maken hebben. Wordt onder andere gebruikt door het loginproces om mislukte loginpogingen te bemerken.
* uucp Berichten die gegenereerd worden door het UUCP systeem.
* local0 tot en met local7 Beschikbaar voor eigen gebruik. Door van deze categorie gebruik te maken, kunt u nieuwe categorieën definiëren die door bepaalde specifieke programma's en processen gebruikt worden.

Voor elke facility kan aangegeven worden in welk geval er gelogd moet worden. De afzonderlijke gevallen worden aangeduid door middel van priorities. Met deze priorities wordt de urgentie van een bericht aangegeven. Hieronder vindt u een qua urgentie oplopende lijst van de verschillende priorities. Als u in de syslog configuratie een bepaalde priority aangeeft, betekent dat dat niet alleen berichten met die specifieke priority gelogd worden, maar ook alle berichten met een hogere prioriteit.

* debug Gebruik deze priority alleen voor debugging doeleinden. Deze priority zorgt ervoor dat werkelijk alles gelogd wordt en dat kan er op zijn beurt weer voor zorgen dat uw logbestanden zeer snel worden volgeschreven.
* info Deze prioriteit zorgt ervoor dat informatieve berichten en alles wat hoger is gelogd worden.
* notice Dit heeft betrekking op berichten die een normale status van het systeem beschrijven.
* warning Gebruik deze categorie om afwijkingen van de normale situatie te loggen. In veel gevallen is het zinnig gebruik te maken van deze priority.
* err Voor foutmeldingen
* crit Om kritische situaties met betrekking tot een bepaald programma te loggen.
* alert Onmiddelijke actie is vereist om het systeem draaiend te houden.
* emerg Wordt gebruikt om te melden dat het systeem niet langer bruikbaar is.

Onder normale omstandigheden wordt bij de aanduiding van een bepaalde prioriteit gelogd wanneer deze prioriteit zich voordoet en voor alle hogere prioriteiten. Dit is meestal ook wel zo handig, maar er zijn een aantal mogelijkheden om hier flexibel mee om te gaan:
* **Gebruik een = teken**. Hiermee wordt aangegeven dat alleen actie moet worden ondernomen bij het voorkomen van de betreffende prioriteit en niet ook voor alle hogere prioriteiten die daar op volgen.
* **Gebruik een ! teken**. Hiermee geeft u aan dat bij de betreffende prioriteit én alle hogerliggende prioriteiten juist geen actie moet worden ondernomen.
* **Gebruik een ***. Op deze wijze geeft u aan dat u het hebt over alle categorieën of alle prioriteiten.
* **Gebruik none**. Hiermee kunt u een bepaalde categorie volledig uitsluiten van het log mechanisme.

Nadat bepaald is wat er dan gelogd moet worden en bij welke ernst van de situatie er tot logging overgegaan moet worden, moet vervolgens bepaald worden wat er in die gevallen moet gebeuren. Hiervoor wordt het tweede veld in de regels in syslog.conf gebruikt. De volgende opties zijn beschikbaar:

* **Wegschrijven naar een bestand**. Geef de naam van het bestand waarnaar weggeschreven moet worden. Om er voor te zorgen dat het bestand niet onmiddellijk bijgewerkt wordt na elke gebeurtenis, laat u de naam van dit bestand voorafgaan door een -. Dit gebeurt in het volgende voorbeeld: news.* -/var/log/news
* **Specificeer een devicenaam**. Door de devicenaam van een of ander apparaat te specificeren, zorgt u ervoor dat de melding naar het betreffende apparaat wordt doorgestuurd. Zo is het bijvoorbeeld redelijk gangbaar dat ernstige meldingen weggeschreven worden naar /dev/tty10 met de regel kern.warn;*.err;authpriv.none /dev/tty10.
* **Maak gebruik van een named pipe**. Hiermee zorgt u ervoor dat alle meldingen doorgestuurd worden naar een FIFO-bestand om daar vervolgens verder verwerkt te worden. Dit gebeurt bijvoorbeeld in de regel kern.warn;*.err;authpriv.nonel/dev/xconsole.
* **Specificeer een lijst gebruikersnamen.** Als het terminaltype waarop de gebruiker is aangemeld dit ondersteund, zorgt u er op deze wijze voor dat een bericht verstuurd wordt naar gebruikers met de overeenkomstige naam als die op dat moment zijn aangemeld. Om er bijvoorbeeld voor te zorgen dat in alle kritische omstandigheden een bericht verstuurd wordt naar de gebruikers root en Linda, gebruikt u de regel *.alert root,Linda.
* **Geef de naam van een computer**. U kunt er op deze manier voor zorgen dat meldingen doorgestuurd worden naar een andere computer waarop de syslog-daemon actief is. Let er wel op dat de naam van die computer voorafgegaan moet worden door een apestaart. Vanuit de optiek van beveiliging is dit een belangrijke optie, als gelogd wordt naar een andere computer, kan de eventuele inbreker er immers niet meer bij komen. Met de volgende regel wordt alles doorgestuurd naar computer xtina: *.* @xtina.sandervanvugt.nl
* **Gebruik een asterisk**. Hiermee zorgt u ervoor dat alle gebruikers die op dat moment ingelogd zijn een melding op hun beeldscherm krijgen. Gebruik deze optie alleen bij services die werkelijk van zeer kritisch belang zijn, als in *.crit *.

Op basis van de voorgaande informatie moet u in staat zijn een werkende syslog configuratie aan te maken in het configuratiebestand /etc/syslog.conf. Toch willen we aan de hand van een concreet voorbeeld een en ander graag illustreren. Dit voorbeeld spreekt voor zich, het onderstaande is letterlijk overgenomen van een Fedora systeem.

```
# Log all kernel messages to the console
# Logging much else clutters up the screen
#kern.*                         /dev/console.

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;news.none;authpriv.none;cron.none      /var/log/messages

# The authpriv file has restricted access.
authpriv.*                      /var/log/secure

# Log all the mail messages in one place.
mail.*                          -/var/log/maillog

# Log cron stuff
cron.*                          /var/log/cron

# Everybody gets emergency messages
```

```
*.emerg                 *

# Save news error of level crit and higher in a special file.
uucp,news.crit          /var/log/spooler

# Save boot messages also to boot.log
local7.*                /var/log/boot.log

#
# INN
#
news.=crit              /var/log/news/news.crit
news.=err               /var/log/news/news.err
news.notice             /var/log/news/news.notice
```

Zoals gezegd, het voorgaande voorbeeld zal weinig verwondering wekken. Toch is er een aantal zaken dat interessant is, let vooral op de wijze waarop verschillende criteria met elkaar verbonden worden, zoals in uucp,news.crit, waarin actie ondernomen wordt bij een severity van crit of hoger voor zowel uucp als news.
Wanneer u ervoor gezorgd hebt dat het configuratiebestand /etc/syslog.conf op deze wijze voor u op bevredigende manier is aangemaakt, kunt u de wijzigingen activeren. Dit doet u door de syslog-daemon opnieuw op te starten. Gebruik hiervoor de opdracht **/etc/init.d/syslogd restart**.

Tip! Leuk natuurlijk dat u gebruik kunt maken van allemaal vooraf gedefinieerde instellingen om te loggen. Soms echter zou u zélf willen bepalen wanneer er iets weggeschreven wordt in een logbestand. Bijvoorbeeld wanneer een script succesvol uitgevoerd wordt. Voor dat soort gevallen is er **logger**. Met behulp van deze opdracht schrijft u rechtstreeks in de verschillende logbestanden die door syslog onderhouden worden; uw melding komt terecht in het instellingenbestand /var/log/messages. Gebruik bijvoorbeeld **logger hallo wereld** om deze uiterst zinnige melding naar /var/log/messages te schrijven.

7.2.2 Aanzetten van remote logging

U weet inmiddels al de helft van wat er nodig is om te loggen op een remote computer: geef in het tweede veld van de syslog-configuratieregels de naam van de betreffende computer voorafgegaan door een apenstaart. Leuk natuurlijk dat dat zo kan, maar de remote computer in kwestie moet binnenkomende meldingen van andere computers wel accepteren. Standaard zal uw server geen logberichten van andere servers accepteren en dat is natuurlijk om begrijpelijke redenen: het zou zo immers wel erg eenvoudig worden om een andere computer vol te schrijven met logberichten.
Om uw server in staat te stellen logberichten van andere servers te ontvangen, moet het proces syslogd gestart worden met de optie –r. Zowel Fedora als SUSE Linux regelen deze instellingen in het configuratiebestand /etc/sysconfig/syslog door aan een variabele mee te geven met welke opties syslog gestart moet worden. In Fedora gebruikt u voor dit doel de variabele SYSLOGD_OPTIONS, in SUSE Linux maakt u gebruik van SYSLOGD_PARAMS.

Tip! Als u gebruikmaakt van de mogelijkheid remote te loggen, kan het handig zijn om alle logs dubbel uit te voeren. Definieer eerst een regel waarmee een bepaald evenement lokaal

gelogd wordt, geef in de volgende regel die er verder exact hetzelfde uitziet aan dat het evenement in kwestie ook remote gelogd moet worden.

***sysconfig Om aan te geven dat syslogd gestart moet worden zodat remote hosts op de lokale host berichten kunnen loggen, moet een variabele in /etc/sysconfig/syslog gespecificeerd worden.

7.2.3 Roteren van logbestanden

Logging is een mooi iets, maar theoretisch is er ook een schaduwkantje aan verbonden. Een hacker die door heeft dat u in sommige gevallen net iets té enthousiaste logging geconfigureerd hebt, zou door middel van logging de harde schijf van uw computer volledig vol kunnen schrijven. Om te zorgen dat dit onmogelijk is, worden logbestanden geroteerd. Dit gebeurt door het programma **logrotate** dat dagelijks kijkt of de logbestanden nog niet te groot zijn geworden. Hiervoor wordt logrotate aangeroepen door de cron-daemon die de bijbehorende configuratie vindt in /etc/cron.daily/logrotate. Als beheerder bepaalt u wat er met een logbestand moet gebeuren wanneer aan het gestelde criterium wordt voldaan. U kunt het bestand laten verwijderen, u kunt het comprimeren en een nieuwe aanmaken en nog veel meer. Wat er precies moet gebeuren, regelt u in het configuratiebestand /etc/logrotate.conf. Hieronder ziet u de inhoud van dit bestand zoals het gebruikt wordt op Fedora Linux:

```
# see “man logrotate” for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you wnt your log files compressed
#compress

# RPM packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own wtmp  -- we’ll rotate them here
/var/log/wtmp {
        monthly
        create0664 root utmp
        rotate 1
}
```

In logrotate configuratiebestanden kan gebruikgemaakt worden van een groot aantal opties. Een aantal van de belangrijkste opties vindt u in de onderstaande tabel.

NOOT VOOR REDACTIE: KAN HET ONDERSTAANDE VOOR DE AFWISSELING IN EEN TABEL WORDEN OPGENOMEN?

Optie Beschrijving

weekly Geeft aan dat eens per week actie ondernomen moet worden

rotate 4 Hiermee wordt bepaald dat de laatste vier versies van het bestand bewaard moeten worden. Als de optie rotate niet gebruikt wordt, wordt bij het aanmaken van een nieuw logbestand het oude logbestand verwijderd.

create Het oude bestand wordt onder een nieuwe naam aangemaakt en een nieuw leeg logbestand wordt gemaakt

compress Bij het aanmaken van een nieuw bestand, wordt het oude bestand gecomprimeerd opgeslagen.

maxage 365 Wanneer een gecomprimeerd logbestand 365 dagen lang bewaard is, wordt het automatisch verwijderd.

notifempty Als een log-bestand leeg is, wordt het logrotate mechanisme er niet op toegepast.

create 664 root root Maak na rotatie een nieuw bestand waarvan de modus wordt ingesteld op 640 en gebruiker en groep root eigenaar worden gemaakt.

postrotate … endscript Soms is het na succesvolle rotatie nodig dat een service opnieuw gestart wordt. Deze en andere zaken kunnen geregeld worden met de optie postrotate.

Als laatste komt u in dit bestand een verwijzing tegen naar bestanden in de directory /etc/logrotate.d. In deze directory kunnen door verschillende RPM's op een systeem op maat gemaakte bestanden worden neergezet die door logrotate verwerkt moeten worden. In deze bestanden vindt u opdrachten die dor logrotate geïnterpreteerd worden. Houd deze bestanden goed in de gaten, want hier gebeurt de werkelijke configuratie van de wijze waarop rotatie van de logbestanden geregeld moet worden. In deze bestanden wordt eveneens gebruikgemaakt van opties die beschreven zijn in de voorgaande tabel.

Oefening 7.2

Om deze oefening uit te kunnen voeren, zijn twee verschillende computers nodig. In deze oefening wordt een computer aangeduid als de logserver, de andere als de client. Bepaal zelf welke computer u in wilt zetten als logserver.

Zorg ervoor dat alle logging centraal geregeld wordt op uw logserver. Neem in de syslog configuratie op de client alle hiervoor benodigde verwijzingen op en zorg ervoor dat de logserver alle meldingen van de client accepteert, ook nadat de logserver opnieuw gestart wordt. Regel alleen dat op de client een melding verstuurd wordt naar alle gebruikers die daar zijn ingelogd wanneer de gebruiker root zich aanmeldt. Regel ook dat er een melding naar syslog verstuurd wordt op het moment dat de DHCP-server op uw systeem opnieuw gestart wordt.

7.3 Beveiligen van FTP-servers

Er zijn nogal wat FTP-servers die in een Linux omgeving gebruikt kunnen worden. Deze staan echter niet bekend om het feit dat ze zo enorm veilig zijn. Er zijn echter uitzonderingen: in sommige FTP-servers kan beveiliging wel gewoon goed geregeld worden. De Very Secure ftp server vsftpd is zo'n server. In deze paragraag lees u hoe u vsftpd kunt configureren als veilige FTP-server in uw netwerk.

7.3.1 Installatie

Er zijn twee manieren om met vsftpd aan het werk te gaan. U kunt de noodzakelijke software downloaden van http://vsftpd.beasts.org, als alternatief kan ook gebruik gemaakt worden van een distributie waarop deze meegeleverd wordt zoals SUSE Linux Professional of Fedora

Linux. Als vsftpd op uw distributie niet automatisch meegeleverd wordt kunt u de volgende procedure volgen om hem te installeren.

1. Download de software van vsftpd.beasts.org.
2. Pak het bestand dat u opgehaald hebt uit met de opdracht **tar -zxvf vsftpd-versienummer.tar.gz**.
3. Activeer na het uitpakken de directory die door het uitpakken is aangemaakt.
4. Geef in deze directory de opdracht **make** om vanuit de bronbestanden een uitvoerbaar programmabestand te bouwen. Gebruik vervolgens vanuit dezelfde directory de opdracht **make install** om de programmabestanden naar de juiste locaties te kopiëren.

De bovenstaande procedure heeft er toe geleid dat er een programmabestand bestaat met de naam /usr/local/sbin/vsftpd. Ook andere bestanden zijn op de juiste locatie neergezet. Dit programmabestand heeft als eigenaar de gebruiker die de opdracht **make** gegeven heeft om de bronbestanden te compileren. U kunt dit controleren met de opdracht **ls -l /usr/local/sbin/vsftpd**. Wanneer vsftpd met uw Linux distributie meegeleverd werd, kan het overigens zo zijn dat het bestand op een andere locatie staat. Op SUSE komt het bijvoorbeeld voor in /usr/sbin/vsftpd. Gebruik indien nodig de opdracht **locate vsftpd** om de exacte locatie van het programmabestand op uw distributie te achterhalen.

Nadat u het programmabestand geïnstalleerd hebt, moet u er voor zorgen dat er een gebruiker nobody bestaat. Dit wordt een gebruiker die verder geen machtigingen heeft op het systeem. De FTP-server gaat gebruikmaken van deze gebruikersaccount om op een optimaal veilige wijze te kunnen werken. Doordat de FTP-server als proces van de gebruiker nobody gestart wordt, kan er geen misbruik van gemaakt worden wanneer de server onverhoopt gehackt wordt. Of de gebruiker nobody bestaat, kunt u controleren met de opdracht **grep nobody /etc/passwd**. Als deze opdracht een resultaat geeft, bestaat de gebruiker nobody al, als u geen resultaat ziet, kunt u deze gebruiker toevoegen met de opdracht **useradd nobody**. Vervolgens moet u er voor zorgen dat er een lege directory bestaat die later door het ftp-proces gebruikt kan worden. U kunt deze directory aanmaken met de opdracht **mkdir /usr/share/empty**. Als dit commando de melding terug geeft dat de betreffende directory al bestaat, is dat geen probleem, u kunt dan gewoon de bestaande directory gebruiken.

Vervolgens moet u de keuze maken of u de FTP-server alleen wilt gebruiken voor gevalideerde gebruikers, of dat u ook anonymous-FTP mogelijk wilt maken. In het geval dat u ook gebruik wilt kunnen maken van anonymous-FTP, moet u een gebruiker met de naam ftp aanmaken. U moet ervoor zorgen dat deze gebruiker een homedirectory krijgt in de directory /var en tot slot is het nodig dat de gebruiker ftp zelf geen eigenaar is van deze directory. U kunt hiervoor de volgende opdrachten gebruiken:

```
# mkdir /var/ftp/
# useradd -d /var/ftp ftp
# chown root.root /var/ftp
# chmod og-w /var/ftp
```

Tot slot kunt u het standaardconfiguratiebestand vsftpd.conf dat met de vsftpd bronbestanden wordt meegeleverd kopiëren naar de directory /etc. Gebruik om de exacte locatie van dit bestand te achterhalen, indien nodig de opdracht locate.

7.3.2 Testen of het werkt

Nadat u de bovenstaande procedure hebt uitgevoerd, bent u klaar om de FTP-server te testen. U doet dit om te controleren of alles goed gegaan is. Het optimaliseren van de FTP-server is nu nog even niet aan de orde, dat volgt later. Onder normale omstandigheden zult u vsftpd waarschijnlijk willen laten activeren door middel van xinetd omdat dit wat extra mogelijkheden biedt op het gebied van beveiliging. Aangezien we nu eerst gaan testen of hij het überhaupt doet, kunt u hem eerst handmatig opstarten zonder xinetd te gebruiken. Voordat u dit doet, moet u er echter wel voor zorgen dat /etc/vsftpd.conf op de juiste manier is ingesteld: zorg ervoor dat aan het eind van dit bestand de regel listen=YES is opgenomen. Deze regel zorgt er namelijk voor dat u vsftpd als standalone-server zonder inetd kunt activeren. Wanneer dit gebeurd is, kunt u vsftpd activeren. Onthoud dat u om vsftpd vanuit xinetd te kunnen activeren deze regel later weer uit moet zetten.

1. Controleer dat er geen andere FTP-servers actief zijn. Vsftpd maakt namelijk gebruik van de standaard FTP-poort 21 en kan niet opstarten wanneer deze al in gebruik is. U kunt controleren of er al een FTP-server actief is door als root de opdracht **netstat -nl | grep 21** te geven. Als er al een FTP-server in gebruik is, zal deze opdracht u dat laten zien. U kunt dan deze FTP-server deactiveren met de opdracht killall naam-van-de-ftpserver.
2. Voer nu eveneens als root de opdracht **/usr/sbin/vsftpd &** uit. Pas de naam van deze opdracht aan als vsftpd op uw distributie op een andere locatie voorkomt. Wanneer deze opdracht als resultaat een regel als [1] 8511 laat zien (het exacte getal achter de [1] is waarschijnlijk anders), wil dat zeggen dat vsftpd met succes gestart is. U kunt nu contact maken met deze server door op uw server de opdracht **ftp localhost** te geven. Hierbij wordt gebruikgemaakt van de standaard FTP-client. Wanneer gevraagd wordt om een gebruikersnaam, gebruikt u ftp, als het goed is hoeft u geen wachtwoord in te voeren om binnen te komen. Aanmelden als een reguliere gebruiker gaat op dit moment nog niet, daarvoor zijn namelijk nog niet de nodige voorzieningen getroffen.

***vsftptest Voer eerst een eenvoudige test uit om op de FTP-server in te loggen als de gebruiker ftp.

7.3.3 Automatisch opstarten

Om te testen of hij werkt, hebt u zojuist de vsftp-server handmatig opgestart. Onder normale omstandigheden wilt u dit waarschijnlijk liever met behulp van xinetd doen. De reden hiervoor is eenvoudig: wanneer de service op deze wijze gestart wordt, wordt hij ten eerste alleen geactiveerd wanneer dat ook echt nodig is. Dat kost u minder systeembronnen en is bovendien ook wat veiliger. Xinetd start de server namelijk pas op het moment dat er een gebruiker is die een verzoek naar de server stuurt. Zolang dat niet gebeurt, wordt de server niet gestart en kost u dat dus geen resources. Een andere reden waarom het beter is te werken met xinetd, is beveiliging. Alle services die met xinetd gestart worden, maken namelijk gebruik van de bestanden hosts.allow en hosts.deny waarin geregeld kan worden dat computers die aan bepaalde voorwaarden voldoen sowieso geen toegang krijgen.

Als op uw systeem gebruikgemaakt wordt van xinetd, moet u in de directory /etc/xinet.d een bestand aanmaken. Geef dit bestand de naam vsftpd en zorg ervoor dat de inhoud er ongeveer als volgt uit ziet:

```
service ftp
{
        disable         = no
        socket_type             = stream
```

```
        wait                    = no
        user                    = root
        server          = /usr/local/sbin/vsftpd
        per_source              = 5
        instances               = 200
        no_access               = 192.168.0.1
        banner_fail             = /etc/vsftpd.busy_banner
        log_on_success          += PID HOST DURATION
        log_on_failure+= HOST
}
```

Zoals u waarschijnlijk zult zien, wordt in het voorgaande bestand gelijk al een groot deel van het gedrag van de FTP-server bepaald. Met name de onderste vijf regels zijn interessant. Om te beginnen wordt met instances = 200 aangegeven hoe vaak de vsftpd server gestart mag worden. Aangezien voor elke nieuwe verbinding een vsftpd-proces gestart wordt, kan in deze configuratie een maximum aantal van 200 connecties gelijktijdig open staan. Dit is een behoorlijk aantal waarmee u een aardige FTP-server op internet draaiend moet kunnen houden. Daaronder wordt met behulp van de regel no_access = 192.168.0.1 het IP-adres gespecificeerd van een host die onder geen enkele voorwaarde toegang krijgt tot de FTP-server. Tot slot wordt met de log_on_succes en log_on_failure regels bepaald op welke wijze er gelogd moet worden. De bovenste regel is echter het meest belangrijk: hiermee wordt gespecificeerd dat vsftpd ook daadwerkelijk gebruikt mag worden. Zorg ervoor dat u ook ergens in het xinetd-bestand voor vsftpd de regel disable = no hebt staan, standaard staat er namelijk disable = yes en kunt u de server dus niet gebruiken.

7.3.4 Toegang voor geautoriseerde gebruikers

Tot nu toe kan alleen de gebruiker anonymous gebruikmaken van de vsftpd-server. Als u ervoor wilt zorgen dat ook gebruikers met een valide gebruikersaccount zich op uw FTP-server aan kunnen melden, moet u in de directory /etc/pam.d een bestand aanmaken met de naam vsftp. Dit bestand kan op verschillende manieren in elkaar gezet worden. Indien u de volgende twee regels opneemt in dit bestand, wordt gewerkt met zogenaamde virtuele gebruikers. Dit zijn gebruikersaccounts die alleen bruikbaar zijn voor toegang tot de FTP-server.

```
auth required /lib/security/pam_userdb.so db=/etc/vsftpd_login
account required /lib/security/pam_userdb.so db=/etc/vsftpd_login
```

*** vsftpusers Om als gebruiker bij vsftpd te kunnen authenticeren, moet een special PAM-configuratiebestand bestaan.

Virtuele gebruikers

Het werken met virtuele gebruikers kan een extra beveiligingsmaatregel zijn. Dergelijke gebruikers worden typisch ingezet wanneer u het niet vertrouwt om normale gebruikersaccounts open te zetten voor de FTP-server, maar een gebruiker toch iets meer vrijheid wilt geven als geboden wordt door het werken met anonymous users. Voordat u echter gebruik kunt maken van een virtuele gebruikersdatabase, moet deze wel eerst gegenereerd worden. Hiervoor kan de volgende procedure uitgevoerd worden:

1. Maak een bestand met de naam logins.txt. In dit bestand definieert u op elke regel een nieuwe gebruiker en op de regel daaronder het bijbehorende wachtwoord. De inhoud van dit bestand kan er bijvoorbeeld als volgt uit zien:

Pleunie
geheim
Linda
geheim
Melissa
geheim
Sanne
geheim

2. Voer nu als root de opdracht **db_load -T -t hash -f login.txt /etc/vsftpd_login.db** uit. Dit zorgt ervoor dat de gegevens uit het ASCII-tekstbestand weggeschreven worden naar een database waarin de wachtwoorden versleuteld bewaard worden. Om dit te kunnen doen, moet echter wel het db-softwarepakket geïnstalleerd zijn. Of dit het geval is, kunt u achterhalen met de opdracht **rpm -q db**. Installeer dit pakket indien nodig eerst voordat u verder gaat.

Virtuele gebruikers en beveiliging
Normaal krijgen virtuele gebruikers alleen toegang tot bestanden die leesbaar zijn voor iedereen (o=r). Het is echter mogelijk per virtuele gebruiker aan te geven wat de mogelijkheden zijn. Om dit te kunnen doen, moet gebruikgemaakt worden van de mogelijkheid per gebruiker een configuratiebestand aan te maken. Laten we er van uitgaan dat gebruiker Pleunie die we in het voorgaande hebben aangemaakt alleen bestanden moet kunnen downloaden, terwijl gebruiker Melissa naast het downloaden van bestanden ook in gelegenheid moet zijn bestanden te uploaden.

1. Voeg de regel user_config_dir=/etc/vsftpd_user_conf toe aan het algemene configuratiebestand /etc/vsftpd.conf. Hiermee verwijst u naar een aan te maken directory waarin configuratiebestanden voor afzonderlijke gebruikers opgeslagen kunnen worden.
2. Maak de directory aan waarnaar in het voorgaande verwezen is met de opdracht mkdir /etc/vsftpd_user_conf.
3. Om er voor te zorgen dat gebruiker Melissa meer rechten krijgt, moet voor haar een configuratiebestand aangemaakt worden. In dit configuratiebestand kan bijvoorbeeld staan dat Melissa ook toegang krijgt tot bestanden die niet voor iedereen leesbaar (o=r) zijn. Een eenvoudige manier om dit voor elkaar te krijgen, is met de opdracht echo "anon_world_readable_only=NO" > /etc/vsftpd_user_conf/melissa.
4. Nu willen we er natuurlijk voor zorgen dat Melissa ook bestanden kan uploaden. Voer hiervoor de volgende twee commando's uit:
echo "write_enable=YES" >> /etc/vsftpd_user_conf/melissa
echo "anon_upload_enable=YES" >> /etc/vsftpd_user_conf/melissa

U zult zien dat wanneer u nu inlogt als Pleunie, u alleen bestanden ziet die leesbaar zijn voor iedereen. Logt u daarentegen in als Melissa, dan ziet u ook alle andere bestanden. Bovendien bent u in staat om bestanden te uploaden, voor zover dat geen conflicten oplevert met bestaande bestanden.

Normale gebruikersaccounts

In het voorgaande hebt u gezien hoe u gebruik kunt maken van virtuele gebruikers om in te loggen op de FTP-server. Als alternatief is het echter ook mogelijk om de gebruikers gewoon op hun account in /etc/passwd te laten authenticeren. Om dit voor elkaar te krijgen, moet u in het configuratiebestand /etc/vsftpd.conf de volgende twee regels opnemen:

```
anonymous_enable=NO
local_enable=YES
```

Met deze regels zorgt u er eerst voor dat anonieme gebruikers niet langer in kunnen loggen, terwijl locale gebruikers dat wel kunnen. Verderop leest u in een uitgewerkt voorbeeld hoe u een FTP-server in kunt richten voor lokaal gebruik.

7.3.5 Verdere configuratie met vsftpd.conf

Naast de mogelijkheid gebruik te maken van het PAM-mechanisme om gebruikers te authenticeren, is er nog een groot aantal andere opties dat gedefinieerd kan worden om het gedrag van vsftpd te bepalen. Deze opties worden ingesteld in het configuratiebestand /etc/vsftpd.conf. Voor uitgebreide informatie over de opties die u in dit bestand kunt gebruiken, kunt u de man pagina van vsftpd.conf opvragen. Wij zullen echter een aantal van de meest interessante opties de revue laten passeren. Dit gebeurt aan de hand van een voorbeeld configuratiebestand waarin de verschillende opties worden toegelicht.

Vsftpd als internet FTP server

De opties die u in het configuratiebestand wilt gebruiken, verschillen per wijze waarop u de server in wilt zetten. De onderstaande configuratie is gebaseerd op het gebruik van vsftpd als FTP-server op internet.

```
#access rights
anonymous_enable=YES
local_enable=NO
write_enable=NO
anon_upload_enable=NO
anon_mkdir_write_enable=NO
anon_other_write_enable=NO
```

Door middel van deze instellingen zorgt u er voor dat de FTP-server alleen door anonieme gebruikers benaderd kan worden. De eerste regel zorgt ervoor dat anonieme gebruikers worden toegestaan, de tweede regel bewerkstelligt dat inloggen als locale gebruiker verboden is. Let er op dat deze instelling vóór alle andere instellingen gaat, met deze instelling actief wordt bijvoorbeeld niet eens gekeken naar de instellingen die op dit moment gelden voor de PAM-configuratie. De volgende vier regels zorgen er voor dat de anonieme gebruiker op geen enkele wijze in de gelegenheid is wijzigingen aan te brengen in het bestandssysteem van de FTP-server.

```
#security
anon_world_readable_only=YES
connect_from_port_20=YES
hide_ids=YES
pasv_min_port=50000
pasv_max_port=60000
```

Met deze instellingen zorgt u er als eerste voor dat alleen directory's en bestanden die de read-machtiging voor others ingesteld hebben gezien kunnen worden. Bestanden waarop (ook) andere machtigingen zijn ingesteld, kunnen dus niet door anonieme gebruikers binnengehaald worden. Vervolgens zorgt de instelling connect_from_port_20 er voor dat alleen connecties vanaf de bekende en veilige poort 20 worden toegestaan. Daarna bewerkstelligt de instelling hide_ids dat als eigenaar van alle bestanden de gebruiker ftp getoond wordt, ook al is dit in werkelijkheid niet het geval. Dit is niet alleen veilig, maar levert ook nog een betere performance omdat niet steeds via de inode in het passwd bestand gekeken hoeft te worden welke gebruikersnaam bij een bepaalde UID hoort. De laatste instelling tenslotte definieert de reeks poorten die als passieve FTP-poorten gebruikt kunnen worden. Deze definitie is vooral handig wanneer de FTP-server achter een firewall staat, de beheerder van de firewall weet dankzij deze definitie exact welke reeks poorten niet uitgefilterd mag worden.

```
#features
xferlog_enable=YES
ls_recurse_enable=NO
ascii_download_enable=NO
async_abor_enable=YES
```

Met dit blokje opties worden vier verschillende instellingen gedaan. Als eerste wordt er een transfer log aangemaakt. In dit bestand worden alle transfers bijgehouden. Als logbestand wordt gebruikgemaakt van /var/log/vsftpd.log. Vervolgens dienen de regels waarin ASCII-downloads en de opdracht ls -R uitgezet worden als beveiliging. Beiden kunnen namelijk als Denial of Service attack gebruikt worden. Om die reden is het voor een internet FTP-server aan te raden geen gebruik te maken van deze opties. Tot slot zorgt de laatste optie ervoor dat oudere clients in staat zijn een transfer die al is gestart weer af te breken.

```
#performance
one_process_model=YES
idle_session_timeout=120
data_connection_timeout=300
accept_timeout=60
connect_timeout=60
anon_max_rate=50000
```

Met behulp van deze laatste reeks instellingen wordt de performance door gebruik van anonieme gebruikers geoptimaliseerd. Om te beginnen wordt ervoor gezorgd dat voor elke connectie een apart proces gestart wordt. Deze parameter wordt in verband gebracht met de parameter instances = 200 in het configuratiebestand xinetd die het maximale aantal vsftpd-processen beperkt tot 200. Samen zorgen beide opties er dus voor dat er maximaal 200 gebruikers gelijktijdig op de FTP-server actief kunnen zijn. Vervolgens is er een viertal opties waarmee allerlei soorten niet-actieve connecties afgebroken worden. Vooral voor een druk bezochte FTP-server is deze instelling cruciaal, een niet actieve gebruiker houdt immers wel een verbinding bezet waardoor anderen wellicht geen contact kunnen maken met de FTP-server. Als laatste wordt de maximale doorvoersnelheid voor anonieme gebruikers ingesteld op een maximum van 50 kilobytes per seconde.

Met een vsftpd.conf die er als dusdanig uitziet, hebt u uw FTP-server geoptimaliseerd voor gebruik als internet service. Vergeet niet dat u na alle wijzigingen die u aanbrengt in het configuratiebestand de FTP-server moet herstarten.

Configuratie van vsftpd voor gebruik op een LAN
Wanneer een FTP-server ingezet wordt voor gebruik in een LAN of intranet-omgeving, gelden er doorgaans heel andere wensen dan bij gebruik als internet-server. Een van de voornaamste eigenschappen is dat gebruikers toegang willen tot hun homedirectory's en daar ook in staat willen zijn bestanden weg te schrijven, directory's aan te maken en veel meer. Realiseer u echter wel dat vervulling van deze wens in een verminderde beveiliging resulteert. Alle gegevens worden namelijk zonder dat ze door encryptie beveiligd worden over de kabel verstuurd. In een dergelijke setting zou het configuratiebestand er als volgt uit kunnen zien.

```
#general settings
dirmessage_enable=YES
ftpd_banner="Welkom op deze mooie FTP-server"
pam_service_name=vsftpd
listen=YES
```

Om te beginnen een aantal algemene instellingen. De eerste twee regels zorgen ervoor dat gebruikers hier en daar wat berichten te zien krijgen. Als eerste indien een bijbehorend berichtenbestand bestaat, in elke directory die geactiveerd wordt, daarnaast is er een algemeen bericht dat getoond wordt wanneer de gebruiker contact maakt met de FTP-server. Vervolgens wordt gespecificeerd wat de naam is van het PAM-bestand dat aangemaakt moet worden in de directory /etc/pam.d. Let er even op dat dit bestand in een behoorlijk aantal gevallen standaard gewoon ftp heet en dat dus of de naam van dit bestand, of de waarde van deze instelling gewijzigd moeten worden.

```
#permissies en beveiliging
local_enable=YES
write_enable=YES
local_umask=027
chroot_local_user=yes
chroot_list_enable=YES
chroot_list_file=/etc/vsftpd.chroot_list
chown_uploads=YES
chown_username=ftp
```

Vervolgens moet de FTP-server open gezet worden voor lokale gebruikers. Daarna wordt in de tweede regel aangegeven dat deze gebruikers ook bestanden weg mogen schrijven. Als dit gebeurt moet dat natuurlijk wel met een bepaalde umask-instelling gebeuren zodat de machtigingen die standaard op het bestand worden ingesteld goed geregeld zijn. Wanneer u niets doet, wordt het standaard umask 077 gebruikt, door middel van deze instelling kunt u gebruik maken van een afwijkend umask. De drie chroot-regels die daarna gedefinieerd worden, zorgen er voor dat de lokale gebruikers in een 'chroot jail' terecht komen. Dit betekent dat ze hun homedirectory zien als de root van het bestandssysteem en dat het dus onmogelijk is naar andere directory's te bladeren omdat ze die gewoon niet zien. Dit is een belangrijk beveiligingsmechanisme waarmee de risico's van de FTP-server aardig beperkt worden. De tweede chroot-regel definieert dat gebruikgemaakt wordt van een lijst met gebruikersnamen voor wie de chroot-instelling geldt. Het verwarrende is dat als deze

instelling op zich gebruikt wordt, de gebruikers in het chroot-bestand de gebruikers zijn waarvoor de chroot van toepassing is. Aangezien echter ervoor al de parameter chroot_local_user gebruikt is, is de chroot-lijst in dit geval een lijst van gebruikers waarvoor de chroot jail niet van toepassing is. De laatste twee regels ten slotte zorgen ervoor dat de eigenaar van elk bestand dat geupload wordt gewijzigd wordt. Deze instelling is vooral zinnig wanneer meerdere gebruikers in een zelfde directory bestanden weg mogen schrijven.

```
#anonymous toegang
anonymous_enable=YES
anon_upload_enable=YES
anon_umaks=022
anon_mkdir_write_enable=yes
anon_other_write_enabe=yes
```

In deze configuratie hebben ook anonieme gebruikers behoorlijk wat mogelijkheden. U moet zich wel bedenken dat dergelijk open instellingen voor anonieme gebruikers echt alleen kunnen wanneer de FTP-server zich achter een firewall bevindt en dus niet door echte onbevoegden benaderd kan worden. Zelfs dan echter kunt u zich nog afvragen of het wenselijk is dat anonieme gebruikers bestanden weg mogen schrijven op uw server.

```
#features
nopriv_user=ftpuser
ls_recursive_enable=YES
deny_email_enable=YES
banned_email_file=/etc/vsftpd.banned_emails
```

Aangezien u toch uw FTP-server wilt beveiligen, wordt met de eerste optie in dit blokje een naam van een gebruiker gespecificeerd waarvan de FTP-server gebruik kan maken. Om een goede beveiliging te kunnen krijgen, is dit een gebruiker die geen enkele machtiging heeft op het lokale systeem. Vervolgens wordt het mogelijk gemaakt de opdracht ls -R uit te voeren, zodat gebruikers ook de inhoud van subdirectory's te zien kunnen krijgen. Tot slot is er nog een aardig maar niet bepaald waterdichte mogelijkheid om te werken met een lijst van mailadressen van anonieme gebruikers die tegengehouden worden. In het bestand /etc/vsftpd.banned_emails worden emailadressen genoteerd van gebruikers aan wie u de toegang wilt ontzeggen. Deze beveiliging is echter bepaald zwak te noemen, anonieme gebruikers kunnen namelijk elk willekeurig emailadres invullen dat ze maar willen en dit wordt op geen enkele wijze gecontroleerd.

```
#performance
Local_max_rate=0
Anon_max_rate=500
```

Tot slot nog twee aardige performance opties. In de eerste regel wordt er voor gezorgd dat de bandbreedte die aan lokale (geauthenticeerde) gebruikers beschikbaar gesteld wordt onbeperkt is. Voor anonieme gebruikers echter wordt deze bandbreedte beperkt tot 500 bytes per seconde.

Werken met Virtual hosts

Wellicht moeten er binnen uw organisatie twee verschillende FTP-servers gebruikt worden: een voor lokale geauthenticeerde gebruikers en een voor anonieme internet-gebruikers. In dat

geval kunt u natuurlijk twee machines neerzetten en op elke afzonderlijke machine een FTP-server installeren. Het is echter evengoed mogelijk te werken met virtuele FTP-servers. Dit betekent dat u uw machine voorziet van meerdere IP-adressen en vervolgens aan elk van deze IP-adressen een aparte FTP-server toewijst. Het leuke hiervan is natuurlijk dat u deze aparte FTP-servers vervolgens ook kunt voorzien van een aparte configuratie. Dit werken met meerdere IP-adressen kan overigens op twee manieren. Als uw machine slechts van één netwerkkaart voorzien is, kunt u deze ene netwerkkaart zonder probleem van een tweede IP-adres voorzien. Het is echter ook mogelijk om gewoon te werken met een machine met twee (of meer) afzonderlijke netwerkkaarten waarvan elke netwerkkaart een afzonderlijk IP-adres heeft.

Nadat de verschillende IP-adressen op een of andere manier gecreëerd zijn, is de rest relatief eenvoudig. U moet er nu voor zorgen dat een tweede configuratiebestand gemaakt wordt en dit tweede configuratiebestand via een apart xinetd-bestand aan een nieuwe virtuele server gekoppeld wordt. In de onderstaande procedure kunt u lezen wat daar precies voor moet gebeuren.

1. Voorzie uw netwerkkaart van een tweede IP-adres. Dit kunt u doen met behulp van de opdracht ifconfig eth0:1 192.168.0.10. De :1 achter de aanduiding van de netwerkkaart geeft aan dat het hier om een tweede IP-adres gaat. We gaan er even voor het gemak van uit dat de routering verder al geregeld is.
2. Maak een gebruiker voor de nieuwe virtuele FTP-site:

 # useradd -d /var/ftp2 ftp2
#chown root.root /var/ftp2
#chmod +rx /var/ftp2
#mkdir /var/ftp2/pub
#touch /var/ftp2/pub/blah

3. Pas het xinetd-bestand waarmee uw primaire FTP-server gestart wordt aan zodat het gekoppeld wordt aan het primaire IP-adres van uw machine. U kunt dit doen door de regel bind = 192.168.0.9 in dit bestand op te nemen. (Uiteraard vervangt u het hier genoemde IP-adres door het adres van uw server.)
4. Maak nu een xinetd-bestand waarmee uw nieuwe site gestart kan worden. In dit xinetd-bestand moet u ervoor zorgen dat de site aan het secondaire IP-adres gebonden wordt. Tevens moet u verwijzen naar het configuratiebestand in de directory /etc waarmee het gedrag van uw tweede FTP-server bepaald wordt. Kopieer eerst uw originele xinetd bestand met de opdracht cp /etc/xinetd.d/vsftpd /etc/xinetd.d/vsftpd2. Voeg nu de volgende twee regels aan vsftpd2 toe:

server_args = /etc/vsftpd2.conf
bind = 192.168.0.10

5. Zorg er nu voor dat het configuratiebestand in /etc naar wens wordt aangepast. Kopieer hiervoor eerst het originele configuratiebestand met de opdracht cp /etc/vsftpd.conf /etc/vsftpd2.conf. Voeg in vsftpd2.conf de volgende twee regels toe:

ftp_username=ftp2
ftpd_banner=Dit is site 2

Pas voor de rest het bestand /etc/vsftpd.conf zo aan dat site 2 exact aan uw wensen voldoet.
6. Gebruik de opdracht /etc/init.d/xinetd restart om xinetd opnieuw te starten en u bent klaar. Er zijn nu twee FTP-servers op uw machine actief.

Oefening 7.3
Om deze oefening uit te kunnen voeren, hebt u aan één computer genoeg. Deze computer kan gebruikt worden als FTP-server en als FTP-client tegelijk.
Neem vsftpd in gebruik als FTP-server. Zorg ervoor dat twee virtuele servers beschikbaar gesteld worden: een minder beveiligde server voor gebruik op een LAN en een goed beveiligde server voor gebruik op internet. De internet server mag alleen voor download benaderd worden door anonieme gebruikers. Op de LAN-server mogen anonieme gebruikers bestanden downloaden en moeten geautoriseerde gebruikers in staat zijn om ook bestanden te uploaden. Welke wijze van beveiliging kiest u en waarom?

Extra opdracht
Voer deze opdracht alleen uit wanneer u tijd over hebt. De opdracht is redelijk complex van aard omdat gegevens uit verschillende hoofdstukken in dit boek gecombineerd moeten worden.
Pas de configuratie die u in het eerste deel van deze oefening gedaan hebt dusdanig aan dat lokale gebruikers nog steeds bestanden kunnen uploaden. Zorg er echter voor dat de authenticatie van deze lokale gebruikers afgehandeld wordt door middel van de LDAP-server op uw netwerk.

7.4 Netfilter en de beveiliging van routers

Als het goed is, wordt in een netwerkomgeving de beveiliging op twee locaties geregeld. Als eerste is er het punt waarop het netwerk met de buitenwereld verbonden wordt. Hier staat in de meeste omstandigheden een goed geconfigureerde firewall die ervoor zorgt dat geen ongeoorloofd verkeer van de buitenwereld uw netwerk op kan en dat het ook onmogelijk is dat bepaalde soorten verkeer vanaf uw eigen netwerk naar buiten gestuurd worden. Dit type beveiliging wordt in de meeste netwerken afgehandeld door een hardware firewall die voor dit doel geoptimaliseerd is. Wist u overigens dat aardig wat gerenommeerde merken van firewall leveranciers voor dit doel gebruikmaken van Linux als besturingssysteem dat op de firewall draait?
De tweede laag van beveiliging gebeurt binnen het netwerk zelf. U mag helemaal zelf weten hoe u dit inricht. In veel situaties worden er binnen het LAN extra services ingezet die voor extra beveiliging zorgen. Denk bijvoorbeeld aan de firewall die er op het netwerk van een school voor zorgt dat het netwerk van de studenten gescheiden wordt van het netwerk van de docenten. Soms echter komen dergelijke interne firewalls niet voor. Wat wel altijd gebruikt zou moeten worden, is een firewall op uw servers zelf. Deze firewall wordt ingezet als tweede laag van defensie en zorgt dat misbruikers binnen het eigen netwerk tegengehouden worden. De meeste Linux-distributies zorgen ervoor dat standaard al tijdens de installatie zo'n firewall wordt aangezet. Voor deze firewalls wordt gebruikgemaakt van de Linux firewallfunctionaliteit die in de kernel is meegeleverd door netfilter. Dit netfilter firewall systeem wordt beheerd met behulp van de opdracht **iptables**.

Tip! Netfilter is de naam van de firewall oplossing die door Linux geboden wordt. Om deze firewalloplossing te beheren, maakt u gebruik van de opdracht iptables.

7.4.1 Tables en Chains

Het uitgangspunt van de **iptables**, is dat de regels waaruit een firewall bestaat zijn ingedeeld in drie tabellen; de zogenaamde tables:

* filter
* nat
* mangle

In elk van deze tabellen vindt u een aantal chains. Deze chains zijn de verzamelingen regels waaruit de firewall is opgebouwd. Welke chains u kunt gebruiken, is afhankelijk van het type table dat u gebruikt. Verderop in dit hoofdstuk vindt u hier meer informatie over. Naast de hoofdverdeling in drie verschillende tabellen biedt Netfilter nog verschillende soorten 'kleinere' functionaliteit:

* stateful packet filtering
* port forwarding
* packet filtering op basis van TCP flags
* filtering op basis van inkomend MAC-adres
* filtering van uitgaande pakketjes op basis van user ID
* anti-DoS features
* logging

Ook bevat netfilter uitstekende mogelijkheden te werken met regels die gemaakt zijn in het 2.2. kernel firewall mechanisme ipchains. Netfilter bevat hiervoor een geavanceerd vertalingsmechanisme.

Tip! Er is in de laatste Linux kernelversies nogal gesleuteld aan het firewall mechanisme. In 2.0 kernels, werd gebruikgemaakt van ipfwdadmin. Met behulp van deze opdracht kon een heel behoorlijke firewall worden aangemaakt. Vervolgens werd in 2.2 kernels ipchains geïntroduceerd. Als u ooit met dit systeem in aanmerking komt, zult u merken dat het grote overeenkomsten vertoont met iptables dat in kernel versie 2.4 geïntroduceerd werd. Tussen de 2.4 kernels en de 2.6 kernels is geen belangrijke wijziging opgetreden in het mechanisme dat voor firewalling gebruikt wordt.

De filter table
De filter table bestaat uit drie ingebouwde chains. Let overigens op de notatie van de chains, die is altijd in hoofdletters:

* INPUT
* OUTPUT
* FORWARD

De INPUT-chain wordt gebruikt om binnenkomende pakketjes te filteren, de OUTPUT-chain wordt gebruikt voor het verwerken van pakketjes die op de lokale machine gedefinieerd zijn en naar buiten verstuurd moeten worden en de FORWARD-chain verwerkt alle pakketjes die door de firewall gerouteerd worden.

De nat table
Ook de nat table bestaat uit drie chains:

PREROUTING

OUTPUT
POSTROUTING

Door middel van de nat table worden twee soorten NAT ondersteund. Om te beginnen is dit het veel gebruikte masquerading. Dit is de vorm van NAT waarin hosts op het privé-netwerk internet op kunnen door gebruik te maken van het IP-adres van de firewall. Computers op internet zien in dat geval niets anders als de firewall en zijn zich er niet van bewust dat er achter die firewall een volledig netwerk voorkomt. Daarnaast is het ook mogelijk het omgekeerde te doen. Dit betekent dat u nodes op het interne netwerk met behulp van de nat table bereikbaar kunt maken voor de buitenwereld door er een geregistreerd IP-adres aan toe te wijzen. Dit verschijnsel wordt ook aangeduid als statisch NAT. Dit betekent dat u aan de interface van de router een tweede IP-adres toekent en er door middel van de nat table voor zorgt dat alles wat op dat secundaire IP-adres binnenkomt wordt doorgestuurd naar de betreffende host op het privénetwerk.

Van de drie regels wordt de PREROUTING chain gebruikt om pakketjes te behandelen zodra ze binnenkomen; dat wil zeggen nog voordat het routing proces er mee aan de werk kan. De OUTPUT-chain wordt gebruikt om pakketjes te behandelen die door het lokale systeem gegenereerd worden maar nog voordat het routing proces er mee aan het werk gaat en de POSTROUTING chain tot slot wordt gebruikt om pakketjes te behandelen voordat ze het systeem verlaten, maar nadat het routing proces er mee aan het werk is geweest.

***vuurmuur SUSE Linux biedt vanuit YaST2 een mogelijkheid op een eenvoudige maar weinig gedetailleerde wijze een firewall in elkaar te klikken.

De mangle table
De mangle table tot slot bestaat uit twee ingebouwde chains:

PREROUTING
OUTPUT

De PREROUTING chain behandelt alle inkomende pakketjes voordat ze gerouteerd worden, de OUTPUT chain behandelt uitgaande pakketjes voordat ze gerouteerd worden. De mangle table wordt alleen in speciale gevallen gebruikt, bijvoorbeeld wanneer speciale services zoals Quality of Service (QoS) gebruikt worden om pakketjes een hogere prioriteit te geven. Om die reden wordt deze table hier niet uitputtend behandeld.

7.4.2 Werken met tables en chains

De opdracht **iptables** biedt alles wat nodig is om chains te definiëren. Wanneer u dit doet, moet u als eerste aangeven in welke table een chain thuishoort. Als u dit niet doet, wordt een nieuw gedefinieerde table automatisch toegewezen aan de default table “filter”. Om een chain wel aan een specifieke table toe te wijzen, maakt u gebruik van de optie -t.

Rules
Bij het definiëren van een chain wordt gebruik gemaakt van verschillende rules. Het is belangrijk dat in deze rules de juiste volgorde gebruikt wordt. In normale gevallen gaat een pakketje door alle regels in de chain. Dit is niet het geval als er een DENY of REJECT in de rule voorkomt. In dat geval wordt het pakketje direct tegengehouden. In alle andere gevallen worden dus alle regels verwerkt. Als geen enkele regel een match oplevert met het betreffende pakketje, wordt de laatste regel in het verhaal toegepast. Deze regel staat ook bekend als de

policy. In het Nederlands mag u dit gerust vertalen in “standaardbeleid”. Dit is dus de standaardregel die wordt toegepast voor alle pakketjes waarvoor geen specifieke rule gevonden kon worden.

Flags, extensions en actions
Wanneer u voor het eerst kennis maakt met netfilter, komt het systeem behoorlijk ingewikkeld over. Als u echter eenmaal door hebt hoe de opdracht iptables waarmee u netfilter instelt in elkaar zit, valt het best mee. Het commando **iptables** kent een paar standaard soorten parameters die altijd terugkomen.

* Om te beginnen zijn er de flags. Hiermee vertelt u wat voor soort actie er uitgevoerd moet worden. Vervolgens moet verwezen worden naar de chain waarin u aan het werk wilt. *
Als tweede zijn er de opties. Deze worden gebruikt om verdere specificatie aan te brengen. Een veelgebruikte optie die bijvoorbeeld toegepast kan worden, is die waarin u de source en destination adressen aangeeft.
* Bij deze opties kunnen vervolgens extensions aangegeven worden. Hiermee kan het gedrag van een optie verder gespecificeerd worden. Denk bijvoorbeeld aan source en destination poortadressen die gebruikt moeten worden.
* Tot slot zijn er de actions. Deze worden altijd voorafgegaan door de schakeloptie -j en bepalen wat er met een pakketje moet gebeuren. De algemene syntaxis van iptables komt er dus samengevat als volgt uit te zien:

Iptables [-flags] [chain] [options [extensions]] [ACTION]

Op de volgende pagina’s krijgt u een overzicht van wat er aan flags, options, extensions en actions gebruikt kan worden. We beperken ons tot de meest belangrijke opties, voor een volledig overzicht kunt u de man pagina van iptables raadplegen.

Flag	**Beschrijving**
-t table	Specificeert in welke table een rule toegepast moet worden. Als deze flag niet gebruikt wordt, wordt de standaard table filter gebruikt.
-A, --append	Voegt een of meer rules toe aan de gespecificeerde table. Als argument moet aangegeven worden op welke chain dit commando uitgevoerd moet worden en welke regel er aan toegevoegd moet worden.
-D, --delete	Verwijdert een of meer rules uit de aangegeven table. Als argument moet aangegeven worden in welke chain gewerkt wordt en welke regel uit de chain bewerkt moet worden. U kunt verwijzen naar de betreffende regel door middel van een regelnummer (gebruik de optie -L voor een overzicht van alle regelnummers)
-F, --flush	Verwijdert alle regels uit een chain.
-I, --insert	Voeg een regel toe in een chain. De syntaxis hiervoor is iptables -I chain regelnummer regel.
-L, --list	Toont alle regels. Bij deze optie kan aangegeven worden dat alleen regels uit één bepaalde chain getoond moeten worden. Als dit niet gebeurd, worden alle regels uit alle chains getoond.
-N, --new-chain	Definieer een nieuwe chain. U mag zelf bepalen welke naam die chain moet krijgen. Uiteraard moet deze naam wel uniek

	zijn.
-P, --policy	Wordt gebruikt om de policy te definiëren. De policy is de standaardregel die in een van de standaard-chains gebruikt wordt. Er kunnen geen policies ingesteld worden voor custom chains. Een policy heeft altijd ofwel de DROP, ofwel de ACCEPT action.
-R, --replace	Gebruik deze optie om een regel te vervangen. De syntaxis van deze optie is -R chain regelnummer regelspecificatie.
-X, --delete-chain	Verwijdert een custom chain. Dit kan alleen wanneer er geen regels meer in de chain voorkomen.
-Z, --zero	Zet alle counters waarin bijgehouden wordt hoeveel pakketjes / bytes door een bepaalde regel verwerkt zijn op nul.
-E, --rename-chain	Geef een andere naam aan een van de custom chains. Kan niet worden toegepast op een standaard chain.

In het voorgaande hebt u een overzicht gevonden van de flags die gebruikt kunnen worden. Deze flags definiëren alleen nog maar op welke positie in de chain iets moet gebeuren. De werkelijke actie wordt uitgevoerd door de options. Hier wordt namelijk heel concreet aangegeven aan welke eigenschappen een pakketje moet voldoen om voor verwerking door een regel in aanmerking te komen. In de onderstaande tabel vindt u een overzicht van alle opties die gebruikt kunnen worden.

Optie	Beschrijving
-d [!] adres[/mask], --destination	Hiermee wordt aangegeven wat het adres is waar een pakketje naartoe gestuurd moet worden. Als adres mag een IP-adres van een node gegeven worden. Het is ook mogelijk een IP-netwerkadres te gebruiken, maar in dat geval moet ook het subnetmasker ingevoerd worden; bijvoorbeeld 193.173.100.0/24 om te verwijzen naar het netwerk 193.173.100.0. Het is ook mogelijk te verwijzen naar namen, maar die moeten dan wel via het standaard name resolving mechanisme achterhaald kunnen worden. Als u een regel wilt maken waarin verwezen wordt naar alle IP-adressen, gebruikt u de aanduiding 0/0 of 0.0.0.0/0. Door gebruik te maken van het uitroepteken, is het mogelijk te verwijzen naar adressen die juist niet behandeld moeten worden.
[!] -f, --fragment	Soms worden pakketjes in meerdere delen, de zogenaamde fragments verstuurd. Als dit het geval is, komt in deze fragments geen adresinformatie meer voor. Als u in een regel wilt verwijzen naar deze fragments, gebruikt u de optie –f, zo weet u zeker dat alle fragmenten waaruit een pakketje bestaan worden meegenomen.
-h	Geeft hulp over het gebruik.
-i [!] interface, --in-interface	Specificeert de interface waarop de pakketjes binnenkomen. Deze optie is alleen nuttig voor de INPUT, PREROUTING en FORWARD chains aangezien alleen deze chains betrekking hebben op de inkomende interface. Als argument moet de naam van de interface gegeven worden: eth0, ppp0, lo etc. In deze naamgeving kan gebruik gemaakt worden van een + als wildcard; eth+ verwijst dus naar alle eth interfaces. Let er

	vooral goed op dat het weglaten van deze interface betekent dat de regel van toepassing is op elke interface; dit zou eenvoudig tot ongewenst gedrag kunnen leiden.
-o [!] interface, --out-interface	Met deze optie wordt verwezen naar de naam van de interface waarop de pakketjes naar buiten gestuurd worden. Vanwege de aard van de chains heeft deze optie alleen nut in de OUTPUT, POSTROUTING en FORWARD chains.
-j target, --jump	Met deze zeer belangrijke optie wordt aangegeven wat er moet gebeuren op het moment dat een rule een treffer oplevert. U verwijst hiermee naar een van de targets (ACCEPT, DROP, REJECT etc.) die later in dit hoofdstuk beschreven worden. Als u deze optie vergeet, gebeurt er niets met een pakketje dat voldoet aan de voorwaarden van een regel, alleen de teller dat er een treffer is geweest wordt opgehoogd. Vergeet deze optie dus niet!
-n, --numeric	Zorgt ervoor dat iptables niet probeert IP-adressen in namen te vertalen wanneer het een overzicht geeft van alle aanwezige regels. Dit komt de snelheid aanzienlijk ten goede.
-p [!] protocol, --protocol	Wordt gebruikt om te verwijzen naar het protocol waarop de regel betrekking heeft. Elke naam of protocolnummer dat gespecificeerd is in /etc/protocols kan hier gebruikt worden. Als deze optie niet gebruikt wordt, geldt de regel voor alle protocollen op het systeem.
-s [1] address[/mask], --source	Deze optie wordt gebruikt om te verwijzen naar het source address dat in een pakketje gebruikt wordt.
-v, --verbose	Bij het tonen van een lijst van alle regels zorgt deze optie er voor dat er wat extra informatie over de regels getoond wordt.
-x, --exact	Toont het precieze en niet het afgeronde aantal pakketjes en bytes wanneer getoond wordt hoeveel pakketjes door een bepaalde regel afgehandeld zijn.
--line-numbers	Wordt in combinatie met de optie -L gebruikt om nummers voor elke regel te tonen.
-m, match_extension	Deze optie wordt gebruikt om te verwijzen naar een match extension. In de volgende paragraaf kunt u lezen wanneer u hier gebruik van wilt maken.

Iptables Extensions

In het voorgaande hebt u gelezen hoe u op een algemene wijze een regel kunt definiëren. Om ervoor te zorgen dat de regel verder verfijnd wordt, maakt u gebruik van match extensions. Match extensions zijn uitbreidingen op de bestaande functionaliteit van iptables. Elk van deze extensies wordt aangeroepen door een bepaalde kernel-module. Dit heeft een enorm stuk flexibiliteit tot gevolg, door een nieuwe kernel-module te schrijven, zou u er dus voor kunnen zorgen dat er nog meer match-extensions beschikbaar komen. Dat is gelijk ook de reden waarom de match-extensions gescheiden zijn van de options. Er zijn twee soorten match-extensions. Om te beginnen zijn er de match-extensions die gerelateerd zijn aan een protocol. Deze worden altijd aangeroepen met een protocol specificatie zoals -p tcp. Daarnaast zijn er de overige meer algemene match extensions. Deze worden aangeroepen met de speciale optie -m.

In de onderstaande tabel vindt u een overzicht van alle soorten match-extensions. In de eerste kolom ziet u de protocolspecificatie en het type match-extension. Deze kolom toont u hoe u de betreffende match-extension aan moet roepen. In de tweede kolom ziet u de exacte match-extension waar het om gaat en in de derde kolom wordt de betekenis verder uitgewerkt.

Specificatie	Match extension	Beschrijving
-p tcp	--sport [!] [port[:port]], --source-port	Wordt gebruikt om de source poort of reeks source poorten te specificeren. Om een reeks op te geven, gebruikt u bijvoorbeeld 1234:1240.
	--dport [!] [port[:port]], --destination-port	Wordt gebruikt om de destination poort of reeks destination poorten op te geven.
	--tcp-flags [!] mask comp	Hiermee kunt u opgeven welke TCP/IP flags bekeken moeten worden. Als mask specificeert u alle flags die bekeken moeten worden, als comp specificeert u die flags waarop gefilterd moet worden.
	[!] –syn	Staat gelijk aan --tcp-flags SYN,RST,ACK SYN waar gekeken wordt naar pakketjes waarin SYN aan staat en RST en ACK niet. Dergelijke pakketjes worden gebruikt om een TCP-connectie op te bouwen, ze kunnen echter ook misbruikt worden in een SYN-attack.
	--tcp-option [!] nummer	Kijkt naar een specifieke TCP-optie.
-p udp	--sport [!] [port[!port]], --source-port	Gelijk aan de overeenkomende optie bij -p tcp
	--dport [!] [port[:port]], --destination port	Gelijk aan de overeenkomende optie bij -p tcp
-p icmp	--icmp-type [!] type[/code]	Kan gebruikt worden om te filteren op een specifiek type ICMP pakketje. Gebruik de opdracht iptables -p icmp -h om een overzicht te krijgen van alle typen ICMP pakketjes waarop gefilterd kan worden
-m mac	--mac-source [!] adres	Hiermee kunt u filteren op inkomende MAC-pakketjes. Aangezien deze optie alleen werkt op inkomende pakketjes, heeft het dus alleen zin er gebruik van te maken in de PREROUTING en de INPUT-chains.
-m limit		Deze match-extension kan gebruikt worden om het aantal van een bepaald soort pakketje te beperken. Deze optie is bijvoorbeeld nuttig om er voor te zorgen dat DoS-attacks voorkomen worden, hiermee wordt een server immers bestookt met constant hetzelfde pakketje. Zonder verdere specificatie wordt het aantal pakketjes van een bepaalde soort met deze optie beperkt tot drie per uur.

	--limit aantal	Hiermee kunt u de match extension -m limit verder specificeren. De opties zijn n/s, n/m, n/h en n/d waarin n een getal is en /s, /m, /h en /d bepalen of het gaat om een aantal pakketjes per seconde, minuut, uur of dag.
	--limit-burst aantal	Stelt een maximaal aantal pakketjes in dat in een burst voor mag komen waarna de limit wordt toegepast. De standaardwaarde is 5.
-m multiport	--source-port port[,port[,…]]	Deze extensie module wordt gebruikt om tot een maximum van 15 source-poorten op te kunnen geven.
	--destination-port port[,port[,…]]	Zie voorgaande, maar dan voor destination-poorten.
-m mark	--mark waarde[/masker]	Kan alleen gebruikt worden in de MANGLE-table en heeft betrekking op iproute2. Deze optie heeft een beperkte toepasbaarheid.
-m state	--state state	Wordt gebruikt om de status van een pakketje te bekijken. Deze kan ingesteld staan op INVALID (het pakketje maakt geen deel uit van een connectie), ESTABLISHED (pakketje maakt deel uit van een connectie, NEW (pakketje poogt een nieuwe connectie op te bouwen) en RELATED (pakketje bouwt een nieuwe connectie op die gerelateerd is aan een reeds bestaande connectie)
-m unclean		Dit is een experimentele module waarin gekeken wordt naar ongebruikelijke pakketjes. Denk bijvoorbeeld aan TCP-pakketjes met een ongebruikelijke optie zoals FIN. Ook wordt gekeken naar andere ongeldige pakketjes.
-m tos	--tos tos	Wordt gebruikt om te kijken naar een waarde die in het TOS-veld is ingesteld. Gebruik de opdracht iptables -m tos -h voor een volledig overzicht van deze waarden.
-m owner	--uid-owner UID	Deze match extension zoekt naar de eigenaar van een pakketje. Deze eigenaar wordt bepaald op basis van de eigenaar van het proces dat het pakketje gegenereerd heeft en kan dientengevolge alleen gebruikt worden voor pakketjes die op de lokale machine gegenereerd zijn en via de OUTPUT-chain naar buiten gaan.
	--gid-owner gid	Zoekt naar de overeenkomstige GID van de owner.
	--pid-owner pid	Zoekt naar het process ID van het proces dat het pakketje gegenereerd heeft.
	--sid-owner sessionid	Zoekt naar de overeenkomstige session-ID die aan een pakketje verbonden is.

Iptables actions

Tot nu toe hebben we alleen nog gekeken naar de criteria waar in een pakketje naar gekeken moet worden. Waar het bij een firewall natuurlijk allemaal om gaat, is dat er wanneer een

match ontstaat, ook een bepaalde actie wordt uitgevoerd. Iptables biedt hiervoor een behoorlijk aantal opties. In de onderstaande tabel vindt u hier een volledig overzicht van.

Action	Beschrijving
ACCEPT	Laat het pakketje door
DROP	Laat het pakketje vallen zonder dat er een bericht teruggestuurd wordt naar de afzender van het pakketje.
REJECT [--reject-with option]	Laat het pakketje vallen, maar zorgt er voor dat er wel een ICMP-bericht terug gestuurd wordt naar de afzender van het pakketje. Standaard wordt het ICMP bericht icmp-port-unreachable terug gestuurd, wanneer u gebruik maakt van --reject-with, is het ook mogelijk zelf op te geven welke ICMP-foutmelding gegenereerd moet worden. Hiervoor kunt u kiezen uit: * icmp-net-unreachable * icmp-host-unreachable * icmp-port-unreachable * icmp-proto-unreachable * icmp-net-prohibited * tcp-reset * icmp-host_prohibited
MASQUERADE [--to-ports port [-port]]	Deze action kan alleen gebruikt worden in de nat/POSTROUTING chain. Daarnaast moet deze optie alleen gebruikt worden wanneer het masquerade adres dynamisch door middel van DHCP uitgedeeld wordt. Bij gebruik van statische adressen moet u gebruik maken van SNAT. Deze optie zorgt ervoor dat iptables zich als klassieke NAT-router gaat gedragen: alle pakketjes die naar buiten verstuurd worden, worden verstuurd met als afzender het geregistreerde IP-adres van de NAT-router.
REDIRECT [[to-ports port[-port]]	Ook deze action kan alleen gebruikt worden in de nat table en wel in de PREROUTING en de OUTPUT chains. Met deze action wordt ervoor gezorgd dat de betreffende pakketjes doorgestuurd worden naar de lokale machine. Eventueel kan hierbij opgegeven worden dat de redirection plaats moet vinden naar gespecificeerde poorten.
RETURN	Als deze actie voorkomt in een door u zelf aangemaakte custom chain, is het resultaat dat het pakketje doorgestuurd wordt naar de volgende regel in de betreffende chain. Wanneer de actie in een van de standaard chains voorkomt, zorgt deze actie ervoor dat de default policy wordt uitgevoerd.
QUEUE	Deze geavanceerde optie zorgt ervoor dat pakketjes doorgestuurd kunnen worden naar een specifieke toepassing. Om dit te kunnen doen, moet gebruik gemaakt worden van de module ip_queue.
LOG [--log-level level] [--log-prefix string] --log-tcp-sequence --log-tcp-options --log-ip-options	Deze optie zorgt ervoor dat het betreffende pakketje gelogd wordt naar de standard log functie, in de meeste gevallen syslogd. Met behulp van de optie --log-level wordt ingesteld welk loglevel hiervoor gebruikt moet worden. Dit verwijst naar standaard loglevels zoals die in het syslogd mechanisme gebruikt moeten worden. Om er voor te zorgen dat

	voorafgaand aan een gelogde regel een bepaalde string getoond wordt, kan gebruik gemakt worden van de optie --log-prefix. De overige drie specificaties kunnen gebruikt worden om respectievelijk tcp-sequence nummers, tcp-opties en ip-opties te loggen.
TOS [--set-tos tos]	Deze actie kan alleen gebruikt worden in de mangle tabel. Met behulp van deze actie kunt u het TOS-veld in de IP-header instellen op een bepaalde waarde. Zie voor meer informatie de opdracht iptables -j TOS –h
DNAT --to-destination ipaddr[ipaddr][:port:port]	Deze actie kan alleen gebruikt worden in de nat/PREROUTING en nat/OUTPUT-chains. Deze actie zorgt ervoor dat het destination IP-adres in een pakketje vertaald moet worden in een ander IP-adres, eventueel in een reeks IP-adressen. Ook is het mogelijk te vertalen naar een specifiek poort adres. Om dit laatste te bewerkstelligen moet in de regel echter wel gebruik gemaakt worden van de optie -p tcp of de optie -p udp om een match op een bepaalde poort te definiëren. De hier gebruikte techniek staat ook bekend als poort-forwarding.
SNAT --to-source ipaddr[-ipaddr][:port:port]	Deze regel kan alleen worden toegepast in nat/POSTROUTING. De actie specificeert dat het source IP-adres vertaald moet worden in een ander IP-adres. Eventueel kunnen hier ook poort adressen bij gebruikt worden.
MIRROR	Dit is een experimentele actie waarbij source en destination IP-adressen omgedraaid worden.
MARK [--set-mark mark]	Hiermee kan in de mangle table de mark waarde van een pakket worden ingesteld.
Custom-chain	Zorgt ervoor dat een zelf aangemaakte chain met de gespecificeerde naam wordt aangeroepen.

7.4.3 Iptables in de praktijk

In het voorgaande hebt u gelezen dat er zeer veel opties zijn om gebruik te maken van iptables. Omdat wij ons voor kunnen stellen dat u door de bomen even het bos niet meer ziet, zullen we nu bespreken hoe u een en ander het beste in de praktijk kunt toepassen.

Om te beginnen moet u er voor zorgen dat er een standaard policy wordt ingesteld voor een aantal chains. De policy is de standaardregel die het standaard gedrag voor iptables bepaalt. Alleen wanneer er in een eerdere regel een uitzondering gedefinieerd is, wordt er van deze standaard policy afgeweken, zo niet dan is altijd de policy op pakketjes van toepassing. Vooral wanneer u iptables gebruikt om te bepalen welke pakketjes wel en welke pakketjes niet doorgestuurd mogen worden, is het aan te raden de policy zo in te stellen dat alle verkeer wordt tegengehouden. De onderstaande regels zorgen er voor dat in de filter table het standaardgedrag voor elke chains wordt ingesteld op DROP. Hiermee maakt u een veilig systeem, namelijk een systeem dat volledig dicht zit. De kunst is hier vervolgens uitzonderingen op te definiëren.

Tip! U kunt naar hartelust experimenteren met iptables. Alle opdrachten die u geeft, zijn na een herstart van uw computer toch weer vergeten; u krijgt tijdens het opstarten gewoon de standaardinstellingen weer terug. Als u op een systeem met een bestaande configuratie met iptables wilt experimenteren, geef dan eerst de opdracht **iptables –L** om een overzicht te

geven van alle chains die op dit moment gedefinieerd zijn en de inhoud van die chains. Geef vervolgens het commando **iptables --flush** om alle bestaande regels weg te gooien en met een schone lei te beginnen. Vervolgens kunt u verder gaan met het instellen van de standaard policies.

iptables -P FORWARD DROP
iptables -P INPUT DROP
iptables –P OUTPUT DROP

Met de standaard-policy ingesteld op DROP, weet u in elk geval zeker dat u veilig bent. Dit is een goede basis om verder te bouwen aan de iptables firewall. In het volgende voorbeeld zullen we bespreken hoe u iptables zo in kunt stellen dat een router op basis van iptables alleen FTP-pakketjes van het interne netwerk doorlaat naar buiten. Voor de rest gebeurt er niets anders. Dit is misschien niet het meest realistische scenario, maar toont wel heel aardig hoe u een iptables firewall op moet zetten. Daarnaast is het aardig dat voor het gebruik van FTP twee poorten open gezet moeten worden, er is immers een FTP-commando en een FTP-data poort.

```
iptables -P FORWARD DROP
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -A FORWARD -s 192.168.1.0/24 -d 0/0 -m multiport -p tcp --dport ftp,ftp-data  -j ACCEPT
iptables -A FORWARD -s 0/0 -d 192.168.1.0/24 -p tcp --sport ftp-data –j ACCEPT
iptables -A FORWARD -s 0/0 -d 192.168.1.0/24 -p tcp --sport ftp ! --syn -j ACCEPT
iptables -A FORWARD –j LOG --log-prefix “iptables FORWARD:  “
```

De regels beginnen er mee dat de policy voor elke chain wordt ingesteld op DROP zodat een veilige situatie als uitgangspunt genomen wordt. Hierop wordt vervolgens een uitzondering gedefinieerd in de FORWARD-chain. In de eerste regel in de FORWARD-chain wordt bepaald dat alle ftp en ftp-data pakketjes die uit het source netwerk 192.168.1.0 komen, mogen worden doorgelaten. De volgende twee regels definiëren dat zowel ftp-data als ftp-pakketjes teruggestuurd mogen worden. Op de normale ftp-pakketjes die vanaf het externe netwerk terugkomen, wordt echter nog wel een uitzondering gemaakt: wanneer de TCP SYN-flag aan staat, wordt het pakketje niet doorgelaten. In dat geval is het namelijk een pakketje dat probeert een sessie te initiëren en dat is niet toegestaan, het gaat er immers niet om dat buitenstaanders uw FTP-servers kunnen gebruiken, maar dat uw gebruikers FTP-servers op internet kunnen gebruiken.. Vanaf het externe netwerk mogen alleen pakketjes teruggestuurd worden as antwoord op verzoeken die vanaf het lokale netwerk afkomstig zijn. Tot slot wordt voor alle overige pakketjes in de FORWARD chain een entry weggeschreven in het logbestand. De reden waarom deze laatste regel alleen geldt voor alle overige pakketjes is voor de hand liggend: wanneer een pakketje immers een regel tegenkomt waaraan voldaan wordt, wordt er niet verder gekeken of er nog meer regels zijn waaraan voldaan wordt. Dit betekent dat alle FTP-pakketjes al afgehandeld zijn op het moment dat een pakketje bij de laatste regel komt. Houd er rekening mee dat deze laatste regel er waarschijnlijk voor zorgt dat gigantisch veel informatie weggeschreven wordt naar de logbestanden! (Voor een hacker zou dit een leuke uitdaging zijn om dit als DoS-attack te gebruiken.)

De bovenstaande regels zorgen er voor dat iptables alle FTP-pakketjes van het interne netwerk doorlaat naar het externe netwerk en dat het ook mogelijk is dat er antwoord komt op

deze pakketjes. Deze regels hebben zin wanneer de machine waarop u ze gebruikt een router is. Maar hoe zit het nu met FTP-pakketjes die gegenereerd worden vanaf de lokale machine? Welnu, deze worden niet doorgelaten. Het verkeer op de lokale machine wordt immers bepaald door de INPUT en OUTPUT-chains in de filter table; de FORWARD-chain die in het bovenstaande gebruikt is definieert de werking van een router. Om ervoor te zorgen dat ook de lokale machine FTP-pakketjes mag versturen, moeten de regels nog wat uitgebreid worden. De nu volgende regels dienen als uitbreiding op het voorgaande voorbeeld. We gaan er daarbij overigens van uit dat de lokale machine op de externe interface gebruik maakt van IP-adres 10.0.0.1

```
iptables -A OUTPUT -o eth0 -s 10.0.0.1 -d 0/0 -m multiport -p tcp --dport ftp,ftp-data -j
ACCEPT
iptables -A OUTPUT -j LOG --log-prefix “iptables OUTPUT:  “
iptables -A INPUT -i eth0 -s 0/0 -d 10.0.0.1 -p tcp --sport ftp-data -j ACCEPT
iptables -A INPUT -I eth0 -s 0/0 -d 10.0.0.1 -p tcp --sport ftp ! --syn -j ACCEPT
iptables -A INPUT -j LOG --log-prefix “iptables INPUT:  “
```

U ziet dat de bovenstaande regels eigenlijk grotendeels gelijk zijn aan de regels in de FORWARD-chain. Het enige verschil bestaat eruit dat het nu om een intern proces gaat. Dat betekent dat er niet gewerkt wordt met de FORWARD-chain, maar met OUTPUT voor uitgaande pakketjes en met INPUT voor alles wat binnenkomt.

Alles bij elkaar zijn de bovenstaande regels efficiënt om er voor te zorgen dat er geen pakketjes verstuurd kunnen worden met uitzondering van FTP-pakketjes. Het kan echter veel eenvoudiger: vooral voor wat betreft de pakketjes die als antwoord weer terug komen. Wanneer gebruikgemaakt wordt van connection tracking door de twee hiervoor noodzakelijke kernelmodules te laden, kunt u gewoon kijken naar pakketjes met een status (--state) van ESTABLISHED of RELATED. Hoe dat er dan uit komt te zien, kunt u in de onderstaande regels zien. Let vooral op de eerste twee regels waarin de kernelmodules geladen worden die noodzakelijk zijn voor connection tracking:

```
modprobe ip_conntrack
modprobe ip_conntrack_ftp
iptables -P FORWARD DROP
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -m state --state NEW -s 192.168.1.0/24 -d 0/0 -p tcp --dport ftp -j
ACCEPT
iptables -A FORWARD -m limit -j LOG --log-prefix “iptables FORWARD:  “
iptables -A OUTPUT -o eth0 -s 10.0.0.1 -d 0/0 -m state --state ESTABLISHED,RELATED -j
ACCEPT
iptables -A OUTPUT -m state --state NEW -o eth0 -s 10.0.0.1 -d 0/0 -p tcp --dport ftp -j
ACCEPT
iptables -A OUTPUT -m limit -j LOG --log-prefix “iptables OUTPUT:  “
iptables -A INPUT -I eth0 -s 0/0 -d 10.0.0.1 -m state --state ESTABLISHED,RELATED -j
ACCEPT
iptables -A INPUT -m limit -j LOG --log-prefix “iptables INPUT:  “
```

Let er op dat naast de nieuwe functie van connectie tracking hier ook een limiet gesteld wordt aan het maximale aantal regels dat in de logbestanden zal verschijnen. Hiervoor wordt gebruik gemaakt van de match extension -m limit. Deze match extension zorgt ervoor dat er van elke pakketsoort maximaal drie logs per uur plaatsvinden, zo hebt u gelijk actie ondernomen tegen de hacker die er misbruik van wil maken dat hij anders uw logbestanden razendsnel vol kan schrijven (maar daar had u natuurlijk met behulp van logrotate al maatregelen tegen getroffen).

7.4.4 Iptables als Network Address Translator

U hebt tot nu toe gelezen over de wijze waarop u een eenvoudige router met iptables kunt beveiligen. In het dagelijks leven zult u echter vooral routers tegenkomen waarop gebruik gemaakt wordt van NAT. Dit zorgt ervoor dat op het privé-netwerk gebruik gemaakt wordt van adressen uit de private address range. Deze privé-adressen worden door de NAT-router vertaald zodat elke node uit het lokale netwerk internet op kan door gebruik te maken van het IP-adres van de NAT-router. NAT is echter meer dan dat. Wanneer u de beschikking hebt over meerdere geregistreerde IP-adressen, is het ook mogelijk om nodes die op het lokale netwerk voorkomen bereikbaar te maken op een van deze geregistreerde IP-adressen.

Tip! Adressen uit de private address range worden op internet niet gerouteerd. De volgende adressen kunnen voor dit doel gebruikt worden:

* Alle adressen uit het netwerk 10.0.0.0
* Alle adressen van 172.16.0.0 tot en met 172.31.255.255
* Alle adressen die beginnen met 192.168

***netwerk Dit netwerk moet geconfigureerd worden met iptables

In de bovenstaande afbeelding ziet u het voorbeeldnetwerk dat we gaan configureren met behulp van iptables. Wat hier opvalt, is dat het netwerk uit twee componenten bestaat. Om te beginnen is er een aantal nodes dat alleen maar contact op hoeft te kunnen nemen met nodes op internet, maar niet vanaf internet bereikbaar hoeft te zijn. Daarnaast zijn er twee servers op het interne netwerk die wel vanaf internet bereikt moeten kunnen worden. Om dit laatste voor elkaar te krijgen is het absoluut noodzakelijk dat u beschikt over meerdere geregistreerde IP-adressen. U kunt een dergelijk netwerk configureren met behulp van de onderstaande configuratie.

```
modprobe iptable_nat
modprobe ip_nat_ftp
modprobe ip_conntrack
modprobe ip_conntrack_ftp
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP
iptables -t nat -A PREROUTING -d 193.173.97.2 -j DNAT --to-destination 192.168.1.2
iptables -t nat -A PREROUTING -d 193.173.97.3 -j DNAT --to-destination 192.168.1.3
iptables -t nat -A POSTROUTING -s 192.168.1.2 -j SNAT --to-source 193.173.97.2
iptables -t nat -A POSTROUTING -s 192.168.1.3 -j SNAT --to-source 193.173.97.3
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -j SNAT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -s 192.168.1.0/24 -d 0/0 -m state --state NEW -j ACCEPT
```

```
iptables -A FORWARD -s 0/0 -d 192.168.1.2 -m state --state
NEW,ESTABLISHED,RELATED -p tcp --dport www -j ACCEPT
iptables -A FORWARD -s 0/0 -d 192.168.1.3 -m state --state
NEW,ESTABLISHED,RELATED -p tcp --dport ftp -j ACCEPT
iptables -A OUTPUT -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -I eth0 -s 192.168.1.0/24 -d 192.168.1.1 -m state --state NEW -j ACCEPT
```

Veel van de regels in het bovenstaande voorbeeld zult u herkennen. Nieuw zijn de regels waar gewerkt wordt in de nat tabel (-t nat). De eerste twee regels zorgen ervoor dat alle pakketjes die binnenkomen op IP-adres 193.173.97.2 en 3 vertaald worden naar de overeenkomstige private IP-adressen. Vervolgens zijn er twee regels (de POSTROUTING regels) waarin het omgekeerde gebeurt. Dit is noodzakelijk om er voor te zorgen dat pakketjes die afkomstig zijn van de WWW en FTP server op het private netwerk door de IP-tables router verstuurd worden met de juiste source-adressen. Als dit niet zou gebeuren, zouden ze immers met de source-adressen uit het netwerk 192.168.1.0 het internet op gaan en dientengevolge onbereikbaar zijn. Vervolgens is er als derde POSTROUTING-regel in de nat-table de regel die er voor zorgt dat alle pakketjes die afkomstig zijn van alle andere nodes op het private netwerk naar buiten gaan met het IP-adres van de iptables-router. Aangezien hier gebruik gemaakt wordt van een statisch IP-adres (het adres 193.173.97.1) wordt gebruikgemaakt van de action SNAT. Als er nu geen gebruikgemaakt zou zijn van een statisch IP-adres op de iptables router, maar een dynamisch adres dat verkregen is van een DHCP-server, zou hier in plaats van SNAT gebruikgemaakt moeten worden van de action MASQUERADE. Doordat naar de status van een pakketje gekeken wordt, is het niet nodig om nog aparte regels te definiëren die ervoor zorgen dat de antwoorden vanaf internet ook weer teruggestuurd mogen worden naar deze nodes, dit gebeurt automatisch door te kijken naar de status ESTABLISHED en RELATED.

Misschien denkt u dat u er nu bent: dat is niet helemaal het geval. De iptables-router is op dit moment namelijk nog niet geconfigureerd om te luisteren naar de IP-adressen 193.173.97.2 en 193.173.97.3. Dit is eenvoudig op te lossen door gebruik te maken van IP-aliasen. Dit betekent dat een tweede IP-adres aan een interface wordt toegewezen. U verwijst dan alleen niet naar eth1, maar naar eth1:0 en eth1:1 enzovoorts. Om het bovenstaande voorbeeld compleet te maken, moet u hiervoor nog de volgende commando's uitvoeren:

```
ifconfig eth1:0 193.173.97.2
ifconfig eth1:1 193.173.97.3
```

Deze commando's zorgen er voor dat eth1 niet langer naar slechts één IP-adres luistert, maar naar drie verschillende IP-adressen.

Tip! Zoals u waarschijnlijk gemerkt zult hebben, komen er in een beetje serieuze firewall al snel heel veel regels voor. U moet er natuurlijk niet aan denken na een vergissing al deze regels opnieuw in te moeten voeren! Dit is eenvoudig te voorkomen door een bakc-up van alle regels te maken met de opdracht **iptables-save**. Als u op deze wijze een back-up hebt weggeschreven naar een bestand, kunt u de regels uit dit bestand weer terugzetten met de opdracht **iptables-restore**. Gebruik daarbij de redirector om te verwijzen naar het bestand waarin de back-up opgeslagen is:

iptables-save > firewallrules

iptables-restore < firewallrules

Oefening 7.4
Om deze opdracht uit te kunnen voeren, hebt u genoeg aan één computer.
Op uw server draait inmiddels een behoorlijk aantal services die vanaf het netwerk bereikt moeten worden. U wilt er natuurlijk voor zorgen dat deze services toegankelijk zijn terwijl dat voor services die misschien per ongeluk ook aan staan niet het geval is. Zorg ervoor dat in elk geval de LDAP server, de webserver en de SSH-daemon op uw server bereikt kunnen worden. Verder mogen er vanaf andere computers geen sessies naar uw server geïnitieerd worden. Wel moet alle verkeer vanaf uw server zelf toegestaan zijn, zodat een beheerder geen belemmering heeft contact op te nemen met andere servers.

7.5 Beveiliging van communicatie

Ooit was telnet een degelijk commando dat gebruikt kon worden om op veilige wijze contact te maken met een server. In de eenentwintigste eeuw weten we echter beter en behoort telnet tot die zaken die in een beveiligde omgeving echt niet meer gebruikt mogen worden. Om te communiceren met een server waarbij belangrijke en gevoelige informatie verstuurd wordt, moet gebruikgemaakt worden van aanvullende maatregelen om het verkeer zo goed mogelijk te beveiligen. Hierbij gaat het om twee zaken: verkeer moet door middel van encryptie versleuteld worden en er moet voorzien worden in een methode waarbij de identiteit van de afzender vastgesteld kan worden. In dit hoofdstuk behandelen we drie technieken die hier allen mee te maken hebben: Kerberos, VPN en SSH.

7.5.1 Installatie en configuratie van Kerberos

Kerberos is een authenticatieprotocol voor client/server toepassingen. Wanneer u gebruikmaakt van Kerberos, behoren zowel strong authentication als data encryptie tot de mogelijkheden. Nadat twee partijen die met elkaar willen communiceren ervan overtuigd zijn dat ze inderdaad met de juiste partij te maken hebben aan de andere kant, kan versleutelde communicatie gebruikt worden zodat wachtwoorden en gegevens goed beschermd zijn.

7.5.1.1 Benodigdheden

De centrale faciliteit in een Kerberos netwerkomgeving, is de Key Distribution Center (KDC). Wanneer een client (dat kan een gebruiker zijn maar ook een toepassing) gebruik wil maken van Kerberos, verstuurt deze een verzoek om een ticket aan de KDC. De KDC maakt dan vervolgens een Ticket Granting Ticket (TGT) voor de client, versleutelt deze waarbij het wachtwoord van de gebruiker als sleutel gebruikt wordt en stuurt daarop de versleutelde TGT terug naar de client. Vervolgens ontcijfert de client de TGT met zijn wachtwoord en als dat succesvol gebeurt, is daarmee de identiteit van de client positief vastgesteld. Op basis van een gevalideerde TGT (die wel na een tijdje verloopt), krijgt de client toegang tot aanvullende tickets waarmee toegang tot specifieke services verkregen kan worden.

Voordat u Kerberos kunt installeren, moet u hebben nagedacht over een aantal zaken:
* De naam van het Kerberos Realm
* De wijze waarop hostnamen naar Kerberos Realms vertaald worden
* De wijze waarop master en slave KDC's met elkaar samenwerken.

Tip! Kerberos maakt gebruik van standaard poort 88 voor het KDC en poort 749 voor de admin server. Zorg er dus voor dat deze poorten niet uitgefilterd worden op uw firewall. U kunt deze poorten aanpassen als u daar een gegronde reden voor hebt. In dat geval regelt u dat

op de server die als KDC functioneert in /etc/services en /etc/kdc.conf en op alle andere servers in krb5.conf.

De naam van het realm
Het eerste wat u nodig hebt in een Kerberos configuratie, is de Kerberos Realm. Noem dit voor het gemak maar het domein dat door Kerberos gebruikt wordt. Het wordt aangeraden om de naam van het realm gelijk te houden aan uw DNS-naam. Mocht uw DNS-naam dus geregistreerd zijn als bloodynerds.nl, dan heet uw Kerberos realm ook bloodynerds.nl.
Nadat u bepaald hebt van welke realm u gebruik wilt maken, moeten computernamen gerelateerd worden aan de naam van een realm. Er zijn twee methodes om dit te regelen:

* Gebruik het configuratiebestand /etc/krb5.conf om de mapping in te specificeren
* Maak een TXT record in de DNS database om het te regelen.

Van deze twee methodes is het het meest gebruikelijk om de mappings te definiëren in het configuratiebestand krb5.conf. U kunt mappings maken voor een volledig domein, maar het is ook mogelijk mappings te maken voor individuele hosts. De methode om name resolutie te regelen via DNS is relatief nieuw. Hierbij wordt door de client gezocht naar een speciaal TXT record in DNS. Dit DNS-record wordt gelocaliseerd door de aanduiding _kerberos toe te voegen vóór de naam van de host in kwestie. Als dat niet lukt, wordt gezocht naar een DNS-record dat bestaat uit de aanduiding _kerberos gevolgd door de naam van het domein waarin de betreffende host voorkomt, als dat ook niet lukt, wordt een niveau hoger gekeken en dat gaat net zo lang door totdat het betreffende DNS-record wordt aangetroffen. Voor de host boston.engineering.foobar.com, wordt dus achtereenvolgens gezocht naar de volgende records:

_kerberos.boston.engineering.foobar.com
_kerberos.engineering.foobar.com
_kerberos.foobar.com
_kerberos.com

***krb5conf.tif Een belangrijk deel van de Kerberos configuratie wordt geregeld in het configuratiebestand /etc/krb5.conf.

Samenwerking tussen master en slave KDC’s
Om te kunnen authenticeren, moet een client kunnen beschikken over een KDC. Het is dus van groot belang dat het KDC altijd beschikbaar is. Om die reden is het aan te raden gebruik te maken van slave KDC’s die de taken van de master over kunnen nemen als deze tijdelijk niet beschikbaar is. Zorg ervoor dat u minimaal over één slave KDC kunt beschikken. Werkt u in een omgeving die uit meerdere sites bestaat? Dan is het sterk de moeite waard ervoor te zorgen dat op elke site een KDC beschikbar is.
Om master en slave Kerberos servers op het netwerk terug te kunnen vinden, moet u er voor zorgen dat een duidelijke en consistente naamgeving gebruikt wordt. Eén methode is bijvoorbeeld om in DNS CNAME records te definiëren zodat de kerberos server onder een duidelijk naam als “kerberos” teruggevonden kan worden en de slave servers namen hebben als “kerberos-1”, “kerberos-2” enzovoorts. In Kerberos versie 5 is hier een nieuwe methode aan toegevoegd: gebruik van het SRV record in DNS wordt nu namelijk ook ondersteund. In dit SRV-record wordt de naam van een service gekoppeld aan de naam van een DNS-domein en als waarde voor de SRV resource records wordt verwezen naar de hostnaam en het poortadres waarop deze service teruggevonden kan worden. Als u er dus voor zorgt dat de KDC beschikbaar is op de host met de naam mijnserver.mijndomein.com, verwijst het SRV-record kerberos.mijndomein.com naar de host KDC.mijndomein.com en het bijbehorende

poortadres 88. Dit voorbeeld maakt duidelijk hoe het systeem van SRV-records werkt, maar houdt er rekening mee dat in werkelijkheid een viertal Kerberos servicenamen gebruikt kan worden:

* **_kerberos._udp** Wordt gebruikt om contact op te nemen met de KDC. Dit is de meest gebruikte SRV resource record.
* **_kerberos-master._udp**. Hiermee wordt verwezen naar de master KDC server. Deze informatie is nodig om in te kunnen loggen wanneer het wachtwoord van een gebruiker recent gewijzigd is en deze wijziging nog niet is gepropageerd naar alle slave servers.
* **_kerberos-adm._tcp**. Wordt momenteel nog niet volledig ondersteund. Met behulp van deze instelling kan verwezen worden naar de poort op de master KDC waar het kadmin programma gebruik van kan maken. Voorlopig lost u dit nog op door in het bestand krb5.conf te verwijzen naar de admin_server.
* **_kpasswd._upd**. Verwijst naar de poort op de master KDC server die gebruikt wordt voor het wijzigen van wachtwoorden.

Zoals u merkt, wordt in een moderne Kerberos omgeving veel geregeld door middel van DNS. Om het u iets eenvoudiger te maken, vindt u hieronder een voorbeeld van de records die in uw DNS zonefile voor kunnen komen om het werken met Kerberos te vereenvoudigen. Houdt er rekening mee dat alle SRV records alleen mogen verwijzen naar de echte hostnamen van servers en niet naar aliassen die met behulp van een CNAME gedefinieerd zijn:

```
@ORIGIN foobar.com
_kerberos               TXT     “FOOBAR.COM”
kerberos                CNAME   daisy
kerberos-1              CNAME   use-the-force-luke
Kerberos-2              CNAME   bunny-rabbit
_kerberos._udp          SRV     0 0 88 daisy
                        SRV     0 0 88 use-the-force-luke
                        SRV     0 0 88 bunny-rabbit
_kerberos-master._udp   SRV     0 0 88 daisy
_kerberos-adm._tcp      SRV     0 0 749 daisy
_kpasswd._udp           SRV     0 0 464 daisy
```

Kerberos configuratie

Het centrale item in een Kerberos opstelling, is dus de KDC. Hiervan configureert u er minimaal twee op uw netwerk: één functioneert als master en de ander functioneert als slave. Het is aan te raden ervoor te zorgen dat een KDC als beiden kan functioneren, dat is handig want als dan bijvoorbeeld de master opeens ontploft, hebt u in een handomdraai de slave omgetoverd tot de nieuwe master. De master is essentieel in uw netwerk, alleen op deze server kan namelijk beheer worden uitgevoerd. Houdt daarbij in de gaten dat ook een eenvoudige taak als het wijzigen van een wachtwoord al als een beheerstaak wordt aangemerkt. Een slave server kan wel kerberos tickets uitdelen, maar er kan niets op gewijzigd worden. Een gebruiker heeft dus een probleem als hij een wachtwoord wil wijzigen terwijl de master server tijdelijk niet bereikbaar is. Bij het opzetten van een Kerberos omgeving, is het handig waar mogelijk de master en de slaves tegelijk te configureren. De procedure die hiervoor doorlopen moet worden, omvat de volgende stappen:

1. Installeer de software
2. Bewerk de configuratiebestanden
3. Genereer de database
4. Voeg beheerders toe aan de ACL

5. Voeg beheerders toe aan de Kerberos database
6. Maak de kadmind keytab
7. Start de daemons op de master KDC
8. Installeer de slave KDC's
9. Maak host keys voor de slave KDC's
10. Genereer host keytabs voor de KDC's
11. Regel het doorgeven van wijzigingen van de master naar de slave servers
12. Propageer de database vanaf de master naar de slave KDC's
13. Maak een stash file op de slave KDC
14. Start de krb5kdc daemon op alle KDC's
15. Voeg kerberos principals toe aan de database

Stap 1. Installeer de software
Voordat u aan het werk kunt, moet de software uiteraard geïnstalleerd worden. In verband met exportwetgeving die export van encryptietechnologie naar bepaalde landen verbiedt, wordt er geen Kerberos software meegeleverd met SUSE en Fedora. Om Kerberos toch in te kunnen zetten, moet u het dus eerst zelf downloaden en installeren. U kunt de source files downloaden van http://web.mit.edu/kerberos. Nadat u de tarball hebt uitgepakt, bestaat de installatie zelf uit een eenvoudige procedure: geef achtereenvolgende de opdrachten **./configure**, **make** en **make install** om de benodigde software te installeren.

Stap 2. Bewerk de configuratiebestanden
Pas de configuratiebestanden kdc.conf en krb5.con aan zodat ze voldoen aan uw huidige situatie. U vindt krb5.conf in de directory /etc, het bestand kdc.conf treft u meestal op een andere locatie. U moet er in deze bestanden voor zorgen dat een correcte verwijzing wordt opgenomen naar te gebruiken hostnamen en realmnamen. Het is ook aan te raden een sectie op te nemen waarin u er voor zorgt dat er log-informatie wordt weggeschreven naar logbestanden. Deze sectie zou er als volgt uit kunnen zien:

```
[logging]
        kdc= FILE:/var/log/krb5kdc.log
        admin_server = FILE:/var/log/kadmin.log
        default = FILE:/var/log/krb5lib.log
```

Stap 3. Genereer de database
Gebruik nu de opdracht **kdb5_util** op de master KDC om de Kerberos database en het optionele stash bestand te genereren. In dit bestand bevindt zich een kopie van de master key van de KDC, deze sleutel is nodig om de KDC te kunnen authenticeren voordat de processen kadmind en krb5kdc gestart kunnen worden. De opdracht kdb5_util vraagt tijdens het opstarten een master key in te voeren. Dit is een wachtwoord dat u moet gebruiken om de key zelf mee te beschermen, het is zaak hier een goed beveiligd wachtwoord voor in te zetten. Het juiste commando om de database aan te maken is **kdb5_util create –r UW.KERBEROS.REALM –s**. Dit zorgt ervoor dat de vijf databasebestanden worden aangemaakt in de directory /usr/local/var/krb5kdc.

Stap 4. Voeg beheerders toe aan de ACL
Nu moet u een Access Control List (acl) bestand maken. De standaardnaam voor dit bestand is kadm5.acl, let even op dat de verwijzing die u in het algemene configuratiebestand kdc.conf doet naar dit bestand klopt. In dit bestand neemt u regels op om aan te geven wie als kerberos beheerder gebruikt wordt en waarop deze beheerder dan wel rechten heeft. In de verwijzing

naar deze instances kunt u verwijzen naar gebruikersnamen, het is ook mogelijk gebruik te maken van jokertekens als * om te verwijzen naar hele groepen. Maak verwijzingen aan in het formaat gebruiker@REALM en specificeer als tweede veld op dezelfde regel de permissies die u aan deze gebruiker wilt geven. Het is mogelijk een keuze te maken uit een flink aantal permissies waarmee u heel gedetailleerd aan kunt geven wie wat mag, u kunt het als het gaat om beheerstaken ook eenvoudig houden en gewoon een * opnemen; hiermee deelt u alle beheerspermissies uit aan beheerders. Om bijvoorbeeld gebruiker Sanne alle beheersrechten te geven, neemt u de regel Sanne@REALM * op in het acl-bestand.

Stap 5. Voeg beheerders toe aan de Kerberos database
U hebt zojuist beheerders toegevoegd aan het ACL-bestand, voordat die beheerders ook echt iets kunnen, moet u er nu voor zorgen dat ze ook als gebruiker in de Kerberos database worden aangemaakt. Gebruik hiervoor de opdracht **kadmin.local**. Deze opdracht zorgt dat er een lokale prompt geopend wordt waarmee u de gespecificeerde gebruiker kunt aanmaken. Gebruik in deze lokale prompt vervolgens het commando **addprinc** om de gebruiker toe te voegen.Om bijvoorbeeld Sanne aan te maken met behulp van deze opdracht gebruikt u **kadmin.local** en vervolgens **addprinc Sanne@REALM** waarbij REALM uiteraard verwijst naar de volledige naam van uw Kerberos Realm.

Stap 6. Maak de kadmind keytab
Nu moet u een kadmind keytab aanmaken. Hierin wordt de sleutel bewaard die kadmind nodig heeft om Kerberos tickets van beheerders te ontcijferen. Op basis van deze keytab wordt bepaald of een beheerder al dan niet toegang krijgt tot de database. U moet deze keytabs in elk geval aanmaken voor de principals kadmin/changepw en kadmin/admin; dit zijn standaard principals die automatisch worden aangemaakt. Om de kadmin keytab aan te maken, gebruikt u het programma **kadmin.local** en vervolgens de opdracht **ktadd –k /usr/local/var/krb5kdc/kadm5.keytab kadmin/admin kadmin/changepw**.

Stap 7. Start de daemons op de master KDC
Nu bent u zover dat u de Kerberos processen op de Master KDC kunt starten. Zorg ervoor dat **krb5kdc** en **kadmind** hiervoor geactiveerd worden.

Stap 8. Installeer de slave KDC's
Nadat u de master KDC geïnstalleerd en actief hebt, kunt u de slave KDC's activeren. De onderstaande taken moeten zowel op de master KDC als op de slave KDC uitgevoerd worden.

Stap 9. Maak host keys voor de slave KDC's
De eerste taak bestaat eruit de host key's toe te voegen. Gebruik hiervoor vanuit de **kadmin** Shell de opdracht **addprinc**. Om bijvoorbeeld achtereenvolgens host principals aan te maken voor de hosts kerberos, kerberos-1 en kerberos-2, geeft u achtereenvolgens de volgende opdrachten:
kadmin
kadmin: addprinc –randkey host/kerberos.uwdomein
kadmin: addprinc –randkey host/kerberos-1.uwdomein
kadmin: addprinc –randkey host/kerberos-2.uwdomein

Stap 10. Genereer host keytabs voor de KDC's
Elke KDC heeft een keytab nodig om tickets te kunnen ontcijferen. Onder ideale omstandigheden is dit een procedure die per KDC wordt uitgevoerd. Om dit bijvoorbeeld te

doen op een KDC met de naam kerberos.uwdomein, geeft u vanuit de kadmin interface de opdracht **ktadd host/kerberos.uwdomein**.

Stap 11. Regel het doorgeven van wijzigingen van de master naar de slave servers
Wijzigingen in de Kerberos database worden doorgegeven vanaf de master naar de slave servers. Hiervoor wordt gebruikgemaakt van de **kpropd** daemon. Deze daemon doet zijn werk op basis van instellingen die gedaan worden in het bestand /usr/local/var/krb5kdc/kpropd.acl. Let erop dat dit bestand op elke KDC moet bestaan! In het onderstaande voorbeeld ziet u hoe dit bestand eruit moet zien als gebruikgemaakt wordt van de master kerberos.mit.edu in het realm ATHENA.MIT.EDU en de slaves kerberos-1.mit.edu en kerberos-2.mit.edu in hetzelfde realm:

host/kerberos.mit.edu@ATHENA.MIT.EDU
host/kerberos-1.mit.edu@ATHENA.MIT.EDU
host/kerberos-2.mit.edu@ATHENA.MIT.EDU

Vervolgens moeten ook de volgende regels worden opgenomen in de inetd-configuratie van elke KDC

krb5_prop stream tcp nowait root /usr/local/sbin/kpropd kpropd
eklogin stream tcp nowait root /usr/local/sbin/klogind
klogind –k –c –e

Daarnaast moet u er van verzekerd zijn dat de processen die nodig zijn ook gedefinieerd zijn in het bestand /etc/services op elke KDC. Neem hiervoor de volgende regels op in dit configuratiebestand:

kerberos	88/udp kdc	# Kerberos authentication (udp)
kerberos	88/tcp kdc	# Kerberos authentication (tcp)
krb5_prop	754/tcp	# Kerberos salve propagation
kerberos-adm	749/tcp	# Kerberos 5 admin/changepw (tcp)
kerberos-adm	749/udp	# Kerberos 5 admin/changepw (udp)
eklogin	2105/tcp	# Kerberos encrypted rlogin

Stap 12. Propageer de database vanaf de master naar de slave KDC’s
Door middel van de bovenstaande procedure hebt u alle slave KDC’s klaar gemaakt om wijzigingen vanaf de master te accepteren. Om dit te doen, moet eerst een dump worden aangemaakt van de database en moet vervolgens de database overgestuurd worden naar de slave KDC’s. De meest handige wijze om dit te regelen is door middel van een Shell script dat u periodiek door cron laat uitvoeren. Geef het shellscript hiervoor de volgende inhoud:

```
#!/bin/sh

kdclist = “kerberos-1.mit.edu.kerberos-2.mit.edu”
/usr/local/sbin/kdb5_util –R “dump /usr/local/var/krb5kdc/slave_datatrans”
for kdc in $kdclist
do
/usr/local/sbin/kprop –f /usr/local/var/krb5kdc/slave_datatrans $kdc
done
```

Stap 13. Maak een stash file op de slave KDC

Alle slave servers hebben nu een kopie van de Kerberos database. U kunt nu een stash-file aanmaken voor al deze servers door op elke slave KDC de opdracht **kdb5_util stash** te geven. Let er op dat bij het invoeren van dit stash file de Kerberos database master key ingevoerd moet worden. Dit is de sleutel die u ingevoerd hebt bij het aanmaken van het stash-file voor de master KDC.

Stap 14. Start de krb5kdc daemon op alle KDC's

Nu bent u zo ver dat de slave KDC's klaar zijn voor gebruik. U kunt nu de daemon krb5kdc activeren om ervoor te zorgen dat de daemons ook gebruikt kunnen worden.

Stap 15. Voeg kerberos principals toe aan de database

Nu de KDC's in de lucht zijn, kunt u met behulp van **kadmin** principals maken voor alle gebruikers, hosts en andere services die u in de Kerberos database wilt opnemen. U vindt een uitvoerige beschrijving van deze procedure in de Kerberos Administrator's Guide die deel uitmaakt van de download met Kerberos sourcefiles.

7.5.2 OpenSSH

Omdat er nogal wat beveiligingsproblemen zijn met de oude commando's rsh en aanverwante commando's kunt u beter gebruik maken van Secure Shell (ssh) om een veilige verbinding op te zetten met een andere server. Het doel van secure shell is gelijk aan het doel van remote shell, namelijk inloggen op een andere computer en uitvoeren van commando's op de andere computer. De meerwaarde van secure shell bestaat eruit dat de communicatie versleuteld wordt door gebruik van encryptie-technieken. Hierdoor levert ssh niet alleen een beter alternatief op voor de r-programma's maar kan het ook gebruikt worden als alternatief voor telnet. Zowel telnet als de verschillende r-commando's sturen immers wachtwoorden in leesbaar formaat over het netwerk en beide programma's worden immers gebruikt om commando's uit te voeren op een andere computer.
Naast Secure Shell, is er een aantal gerelateerde commando's dat deel uitmaakt van de SSH-suite. Dit zijn de commando's sftp en scp. Sftp wordt gebruikt om over een beveiligde verbinding contact te maken met een sftp-FTP-server. Scp heeft als doel het mogelijk te maken op een veilige wijze bestanden te kopiëren tussen twee computers.

Let er bij het werken met SSH op, dat er twee versies van het protocol in omloop zijn. De parameters die door beide versies gebruikt worden zijn verschillend. De meeste hedendaagse Linux-distributies maken gebruik van versie 2. Om die reden zullen we in dit hoofdstuk dan ook de nadruk leggen op deze versie.

7.5.2.1 Authenticatiemethoden

Als gebruikgemaakt wordt van SSH, kan op verschillende manieren geauthenticeerd worden. Welke manieren precies gebruikt worden, wordt bepaald door de versie van het SSH-protocol die gebruikt wordt. Bij beide versies is echter het meest essentiële onderdeel dat er gebruik gemaakt wordt van encryptie-sleutels die tijdens het inloggen met elkaar uitgewisseld moeten worden. In het onderstaande verhaal maken we duidelijk hoe dit werkt. De volgende methodes voor SSH authenticatie zijn beschikbaar:

* **Host-based authentication**. De authenticatie vindt plaats op basis van instellingen in de bestanden /etc/hosts.equiv, /etc/shosts.equiv en/of ~/.rhosts of ~/.shosts. Deze onveilige wijze van authenticatie werd door SSH versie 1 ondersteund om compatibiliteit met de onveilige r-services te beiden. In nieuwere versies wordt echter geen ondersteuning meer

geboden voor deze methode. U regelt deze vorm van authenticatie door middel van de optie RhostsAuthentication in het configuratiebestand sshd_config.

* **Host-based RSA authenticatie**. Deze vorm van authenticatie is eigenlijk host-based authenticatie waarbij gebruikgemaakt wordt van RSA keys. Ook deze wijze van authenticatie is inmiddels door veiliger manieren vervangen. Public keys van clients worden op de server opgeslagen in de bestanden ~/.ssh/known_hosts en /etc/ssh/ssh_known_hosts. Nadat de public key op de server is opgeslagen, moet de client bewijzen dat hij de overeenkomstige private key heeft. Deze is opgeslagen in het bestand /etc/ssh/ssh_host_key dat alleen leesbaar is door de gebruiker root. U activeert deze vorm van authenticatie met behulp van de parameter RhostsRSAAuthentication in SSH1 en HostbasedAuthentication in SSH2.
* **Public key authenticatie**. In deze vorm van authenticatie wordt gebruikgemaakt van private/public key paren. Hierbij bewijst de gebruiker dat hij bezitter is van zijn private key door een challenge-response procedure die automatisch gestart wordt. Dit is de meest veilige wijze van communicatie. In SSH1 wordt voor deze wijze van authenticatie gebruikgemaakt van de parameter RSAAuthentication, SSH2 gebruikt de parameter PubkeyAuthentication.
* **Password authenticatie**. Bij deze methode van authenticatie wordt gebruikgemaakt van het wachtwoord van de gebruiker. Dit wachtwoord wordt beschermd door middel van encryptie verstuurd. Deze wijze van authenticatie is de enige manier die standaard werkt. U kunt hem aan of uit zetten met behulp van de parameter PasswordAuthentication.

7.5.2.2 De werking van SSH-authenticatie

Wanneer SSH versie 1 gebruikt wordt, vindt authenticatie als volgt plaats. Elke host heeft een eigen host-key. Deze wordt gebruikt ter identificatie van de betreffende host. Daarnaast heeft elke host een server RSA-key. Dit is een geheime sleutel die elk uur opnieuw gegenereerd wordt en nooit op schijf wordt opgeslagen. Wanneer een client contact probeert te maken, antwoordt het serverproces sshd hierop met zijn public host-key en zijn server-key. Deze public host-key wordt door de client vergeleken met een versie van deze sleutel die op de client bekend is. Het doel hiervan is de identiteit van de server onomstotelijk vast te stellen. Als dit om een of andere reden niet lukt, wordt er een foutmelding gegenereerd:

```
[root@fedora root]# ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
RSA key fingerprint is 12:da:36:3b:8f:40:8a:0e:37:4f:a1:54:ff:6c:30:1e
Are you sure you want to continue connecting (yes/no)?
```

Wanneer u er zeker van bent dat u inderdaad met de beoogde host verbinding hebt, kunt u deze vraag positief beantwoorden. Dit heeft als gevolg dat de betreffende host permanent wordt toegevoegd aan de lijst van bekende hosts in het bestand /etc/ssh/ssh_known_hosts of ~/.ssh/known_hosts.. Vervolgens kan overgegaan worden naar de tweede fase van de authenticatie waarbij de gebruiker in kwestie zich aanmeldt op de andere computer.

Wanneer de client verbinding heeft gemaakt, genereert deze een willekeurig getal van 256 bits. Dit getal wordt door de client versleuteld met zowel de host-key als de server-key en teruggestuurd naar de SSH-server. Wanneer dit versleutelde getal op de server bekend is, kan het door beide partijen gebruikt worden als session-key waarmee alle gegevens die over een weer verstuurd worden versteuteld worden.

Wanneer dit initiële contact tot stand gebracht is, wordt de authenticatie voortgezet. Op dit moment is er namelijk weliswaar een veilige communicatie mogelijk tussen client en server, maar de identiteit van de client is nog op geen enkele wijze vastgesteld. Hiervoor kan gebruik

gemaakt worden van een van de vier verschillende mechanismen die in het voorgaande besproken zijn::

* Host-based authenticatie;
* Host-based RSA authenticatie
* Public key authenticatie;
* Wachtwoord authenticatie.

De werkwijze die hiervoor uiteindelijk gebruikt wordt, wordt bepaald in het configuratiebestand /etc/ssh/sshd_config. Het eerste wat u in dit bestand doet, is ervoor zorgen dat alle op .rhosts gebaseerde authenticatie wordt uitgezet. Deze authenticatievorm maakt namelijk gebruik van een lijstje van trusted hosts. Aangezien het relatief eenvoudig is de naam of het IP-adres van een andere computer aan te nemen (te spoofen), is dit een zeer onveilige werkwijze die niet gebruikt moet worden.

Nadat de authenticatie succesvol is uitgevoerd, bestaat er een beveiligd kanaal tussen de client en de server. Dit beveiligde kanaal kan door beide partijen gebruikt worden om gegevens heen en weer te sturen. De client kan op dit moment een remote terminal op de server starten, TCP/IP-verbindingen doorsturen of gewoon een commando uitvoeren op de andere machine. Authenticatieinstellingen op de server vastleggen. Om ervoor te zorgen dat de client op de juiste wijze kan werken in de remote omgeving die door de server geboden wordt, wordt een tweetal omgevingsvariabelen ingesteld: TERM en DISPLAY. Daarbij worden X11 connecties en eventuele TCP-connecties omgeleid.

Om te bepalen welke wijze van authenticatie gebruikt moet worden voor clients die zich aan willen melden met behulp van SSH, moet u het configuratiebestand /etc/ssh/sshd_config op de server bewerken. In dit bestand wordt gebruik gemaakt van een aantal standaardwaarden. Wanneer u niets aanpast in sshd_config, worden alle standaardwaarden gebruikt zoals u ze in de regels kunt lezen die in het voorbeeldbestand gebruikt worden. Wilt u daarentegen andere dan de standaardwaarden gebruiken, dan moet u dit bestand bewerken. Hieronder vindt u een overzicht van enkele parameters die in dit bestand gebruikt kunnen worden en die gerelateerd zijn aan authenticatie:

* **ChallengeResponseAuthentication** Met behulp van deze instelling bepaalt u of authenticatie door middel van challenge-response is toegestaan. De standaardwaarde staat op “yes”.
* **HostbasedAuthentication** Hiermee kunt u aangeven of authenticatie op basis van het rhosts-mechanisme is toegestaan. Hieronder wordt ook inbegrepen de authenticatie die plaats vindt op basis van een lijstje met computers dat gedefinieerd is in het configuratiebestand /etc/hosts.equiv. Aangezien dit een onveilige werkwijze is, wordt het gebruik van deze optie afgeraden. In verband met achterwaardse compatibiliteit, kan het soms echter noodzakelijk zijn toch van deze werkwijze gebruik te maken. De standaardwaarde voor deze parameter is “no”.
* **IgnoreUserKnownHosts** Met behulp van deze parameter kunt u aangeven of eindgebruikers zelf in staat zijn een bestand aan te maken in hun homedirectory met de naam .ssh/known_hosts. Het nut van een dergelijk bestand is dat het hiermee mogelijk gemaakt kan worden dat de gebruiker zelf lijsten bijhoudt van hosts die vertrouwd worden. De standaardinstelling voor deze parameter is “yes”. Alleen wanneer u vanuit de optiek van een strikt beveiligingsbeleid zelf wilt kunnen bepalen met welke veilige hosts contact gemaakt kan worden, is het aan te raden deze optie uit te zetten.

* **KerberosAuthentication** en **KerberosOrLocalPasswd**`Met de parameter KerberosAuthentication wordt aangegeven of authenticatie met behulp van Kerberos tickets is toegestaan. De standaardwaarde voor deze parameter is “no”. De optie KerberosOrLocalPassword daarentegen wordt gebruikt om aan te geven dat ofwel gebruik gemaakt wordt van Kerberos, ofwel van het locale passwd-mechanisme waarbij gebruikers aanmelden op de traditionele UNIX-manier waarbij gebruik gemaakt wordt van de locale gebruikersdatabase. Deze waarde staat standaard op “yes”, wat het mogelijk maakt altijd in te loggen middels het passwd-mechanisme.
* **PasswordAuthentication** Om ervoor te zorgen dat gebruikers over de SSH-verbinding in kunnen loggen met hun normale wachtwoord zoals dat gedefinieerd is in /etc/passwd en /etc/shadow op de SSH-server, moet deze optie aan staan. Dit is standaard het geval. Voor een optimale beveiliging kan overwogen worden deze optie uit te zetten en uitsluitend PublicKeyAuthentication toe te staan.
* **PubKeyAuthentication** Deze parameter die standaard aan staat zorgt er in protocol versie 2 voor dat authenticatie kan plaatsvinden op basis van public keys.
* **RhostsAuthentication** Om er voor te zorgen dat op basis van het Rhosts-mechanisme geauthenticeerd wordt, kunt u in protocol versie 1 deze optie aan zetten. Standaard staat deze optie uit, hetgeen ook vanuit optiek van beveiliging absoluut is aan te raden.
* **RSAAuthentication** Hiermee geeft u aan of aanmelding met behulp van het RSA-protocol is toegetaan. Deze optie kan alleen in versie 1 gebruikt worden en staat daar standaard aan. In SSH versie 2 is gebruik van deze optie overbodig.

Waar u er in /etc/ssh/sshd_config voor zorgt dat bepaalde mogelijkheden op de server aan of uit gezet worden, moet u ook op de client regelen op welke wijze de authenticatie plaats moet vinden. De exacte wijze waarop u dit regelt, is afhankelijk van het programma dat u op de client gebruikt. Als de standaard SSH-client gebruikt wordt die met Linux meegeleverd wordt, kan de gebruiker gebruikmaken van een configuratiebestand waarin de nodige instellingen gedaan worden. Dit bestand kan aangemaakt worden als algemeen configuratiebestand met de naam /etc/ssh/ssh_config. Dit bestand is geldig voor iedere gebruiker, tenzij een gebruiker in zijn eigen homedirectory een bestand heeft met de naam .ssh/config waarin instellingen in het algemene configuratiebestand overschreven worden. De parameters met betrekking tot authenticatie zijn in dit bestand grotendeels gelijk aan de instellingen in het serverconfiguratiebestand sshd_config. Raadpleeg voor details de man-pagina.

Initiatie van een SSH-sessie.
Wanneer u de authenticatie-instellingen geregeld hebt, kunt u in principe direct aan het werk met SSH en aanverwante programma’s. Ook zonder dat u ook maar iets regelt, kunt u direct al gebruikmaken van authenticatie op basis van wachtwoorden.
Hiervoor kan gebruikgemaakt worden van de opdracht ssh. Als op de server de SSH-daemon actief is, kunnen clients met behulp van de opdracht **ssh hostnaam** contact maken met een remote computer. Uiteraard kan in plaats van de hostnaam ook gebruik gemaakt worden van een IP-adres.

ssh
Wanneer de opdracht ssh vanaf de prompt wordt gestart, kunnen hierbij ook de nodige parameters gegeven worden. Om te beginnen is dat natuurlijk de naam van de computer waarmee contact gemaakt moet worden, daarnaast kunt u in de vorm user@hostnaam ook uw gebruikersnaam op de betreffende host meegeven. Dit kunt u overigens ook doen met behulp van de parameter –l <gebruikersnaam>. Ook is het mogelijk met de opdracht ssh een

commando te specificeren dat direct uitgevoerd moet worden. Zo zorgt de opdracht **ssh melissa@server kill -9 sshd** er voor dat op de remote host wordt ingelogd en vervolgens na succesvolle authenticatie automatisch een niet zo heel erg zinnig commando wordt uitgevoerd.

Wanneer er niets anders geregeld is, wordt bij gebruik van SSH 1 als eerste geprobeerd in te loggen door gebruik te maken van RSA public/private-keys. Alleen als dit niet lukt, wordt gevraagd om een wachtwoord. Versie 2 van het protocol gaat op soortgelijke wijze tewerk, met als enige afwijking dat hier naast RSA-algoritmes ook gebruik gemaakt kan worden van het DSA-algoritme. U zult om succesvol tewerk te kunnen gaan, bij gebruik van versie 2 dan ook aan moeten geven welk type key u wilt gebruiken. Dit doet u door bij de opdracht **ssh-keygen** gebruik te maken van de optie –t, geef bijvoorbeeld de opdracht **ssh-keygen –t rsa** om een RSA-key te genereren. Dit doet u op de machine waarop u als gebruiker werkt. De sleutel die op deze wijze gegenereerd wordt, moet vervolgens gekopieerd worden naar het bestand authorized_keys dat voorkomt in de directory .ssh in de homedirectory van de gebruiker op de ssh-server. Verderop in dit hoofsdtuk vindt u de exacte procedure om authenticatie op basis van public keys mogelijk te maken stap voor stap uitgewerkt.

Wanneer u door middel van SSH gebruik maakt van een andere computer, kunt u er ook voor zorgen dat alle uitvoer van X-programma's wordt doorgestuurd naar uw eigen computer. Hiervoor moet gebruik gemaakt worden van de optie ForwardX11. Deze moet voor dit doel op "yes" zijn ingesteld. Ook moet u ervan verzekerd zijn dat er met de opdracht **xauth** voor gezorgd is dat de andere computer zijn X-schermuitvoer op uw computer mag tonen en dat de omgevingsvariabelen TERM en DISPLAY goed staan ingesteld.

Hieronder vindt u een aantal voorbeelden van de wijze waarop een gebruiker contact kan maken met een SSH-server.

alex@x-tina:~> ssh –l franck@linux.sandervanvugt.nl
Met de bovenstaande regel logt gebruikt alex die lokaal is aangemeld in op linux.sandervanvugt.nl. De optie –l zorgt ervoor dat hij gebruik kan maken van een andere loginnaam op de andere computer: franck in dit geval. Een andere mogelijkheid is dat een gebruiker van een afstandje inlogt op een computer en een commando op die computer uitvoert. Het volgende commando toont hoe gebruiker alex computer Linux.sandervanvugt.nl uit zet:

alex@x-tina:~> ssh root@linux.sandervanvugt.nl shutdown –h now
Ook erg handig is wanneer de weergave van grafische toepassingen meteen geregeld wordt. De volgende odpracht bijvoorbeeld zorgt ervoor dat alle grafische toepassingen die door alex gestart worden op Linux.sander.nl, kunnen worden weergegeven op x-tina:

alex@x-tina:~> ssh –X Linux.sandervanvugt.nl
Tot slot is het met SSL ook mogelijk om connecties die binnenkomen op bepaalde poorten door te sturen. Zo verzekert u zich ervan dat gegevens die door andere netwerkprotocollen verstuurd worden op een veilige wijze worden doorgegeven. Dit gebeurt bijvoorbeeld nadat het volgende commando gegeven wordt:

alex@x-tina:~> ssh –L 5609:linux.sandervanvugt.nl:110 alex@linux.sandervanvugt.nl

scp

Boven op het standaard SSH-mechanisme, kan gebruikgemaakt worden van de opdracht **scp**. Dit is een erg handig klein commando waarmee u snel een bestand kunt kopiëren van de ene computer naar de andere computer. De opdracht **scp** is de opvolger van het principieel onveilige commando rcp. Het belangrijkste verschil tussen beide commando's bestaat eruit dat scp wél vereist dat authenticatie plaatsvindt, terwijl dit voor rcp niet het geval is. Gebruik van scp is eenvoudig: u geeft als eerste argument een volledige verwijzing naar het bronbestand en als tweede argument een verwijzing naar het doelbestand. Daarbij kan ook gelijk de gebruikersnaam meegegeven worden van de gebruiker wiens credentials op beide servers gebruikt moeten worden. Zo kunt u bijvoorbeeld de opdracht **scp sander@x-tina:/etc/passwd root@laetitia:/etc/passwd** gebruiken om ervoor te zorgen dat de gebruikersdatabase op server laetitia vervangen wordt door de gebruikersdatabase van server x-tina. Wij raden u overigens aan heel goed na te denken of dit echt is wat u wilt voordat u dit voorbeeld in de praktijk gaat brengen. Wellicht is het veiliger gebruik te maken van een minder belangrijk bestand zoals motd.

sftp
Naast de klassieke r-commando's is er nog een gouwe ouwe die standaard behoorlijk onveilig is: de opdracht ftp. Net als vrijwel alle andere commando's die al lang in gebruik zijn, verstuurt ook dit commando zijn gegevens zonder dat er op enige wijze iets versleuteld wordt. Dit betekent dat gevoelige gegevens zoals wachtwoorden en gebruikersnamen uitgelezen kunnen worden wanneer iemand ze met een sniffer probeert te onderscheppen. Gelukkig is het redelijk eenvoudig hier wat aan te doen: configureer in plaats van een FTP-server een sftp-server. Deze is namelijk in staat te werken met beveiligde verbindingen. Om deze speciale beveiligde FTP-server te activeren, gebruikt u de opdracht **sshd command="sftp-server**" bij het opstarten van de SSH-daemon. U kunt deze service ook automatisch laten starten bij het activeren van de SSH-daemon. Bewerk hiervoor de parameter Subsystem in het configuratiebestand /etc/ssh/sshd_config. Zorg ervoor dat deze eruit ziet als Subsystem=sftp-server om de beveiligde FTP-server automatisch te activeren op het moment dat sshd geactiveerd wordt.

Wanneer uw SSH-service zo geconfigureerd is dat de sftp-server automatisch gestart wordt, is de rest kinderspel. U kunt de opdracht sftp gebruiken om bestanden op de FTP-server te plaatsen of er vanaf te downloaden. Dit commando werkt precies hetzelfde als de normale ftp-client.

7.5.2.3 Aan het werk met Public Key authenticatie
Standaard is OpenSSH geconfigureerd voor authenticatie op basis van wachtwoorden. Dit is echter niet de meest veilige manier en het is ook niet eens de meest handige werkwijze. Veel handiger is het wanneer op basis van public key's geauthenticeerd kan worden. Hieronder vindt u een beschrijving hoe u deze vorm van authenticatie kunt regelen. We gaan hierbij uit van SSH versie 2, dat is overigens sowieso de versie die u moet gebruiken omdat hij veel veiliger is dan versie 1.
Bij public key authenticatiew wordt de public key van een gebruiker opgeslagen op de server. In de meeste gevallen wordt hiervoor de homedirectory van de gebruiker gebruikt. De private key daarentegen is opgeslagen op de computer van de client zelf. Als dit een Linux-computer is, zal dat doorgaans ook in de homedirectory van de gebruiker zijn, als het gaat om een ander type bestuyringssysteem waarop een SSH client gebruikt wordt, wordt de pricate key ergens in de configuratie van de client opgeslagen. Als gebruikgemaakt wordt van public key authenticatie, vindt tijdens dit proces het volgende plaats:
1. De client laat aan de server weten welke public key gebruikt moet worden.

2. De server kijkt of deze sleutel beschikbaar is.
3. De server genereert een willekeurig getal op basis van de public key en stuurt dat naar de client toe.
4. De client ontcijfert dit willekeurige getal op basis van zijn private key.
5. De client stuurt de server een MD5-checksum die berekend is op basis van het random getal dat hij van de server ontvangen heeft.
6. De server berekent op basis van dit zelfde random getal ook een checksum. Als deze gelijk is aan de checksum van de client, is de client met succes geauthenticeerd.

U ziet dat er bij deze vorm van authenticatie helemaal geen gegbruikgemaakt wordt van een wachtwoord, het enige dat verstuurd wordt is een versleuteld willekeurig getal. De belangrijkste factor om in dit scenario te kunnen authenticeren, is de private key van de gebruiker. Wanneer iemand deze private key te pakken kan krijgen, kan hij succesvol authenticeren als die gebruiker. Om die reden moeten de rechten op de sleutel zo geregeld worden dat alleen de gebruiker zelf er gebruik van kan maken. Daarnaast is het aan te raden de private key te beschermen met een passphrase. Dit is een wachtwoord dat aan de key gekoppeld wordt en ingevoerd moet worden voordat de key gebruikt kan worden tijdens de authenticatie.

De sleutels aanmaken

Voordat u als gebruiker kunt authenticeren op basis van sleutels, moeten deze sleutels gegenereerd worden. Hiervoor wordt gebruikgemaakt van de opdracht **ssh-keygen**. Achter deze opdracht wordt met de parameter **–t** aangegeven welke encryptietechniek voor de sleutel in kwestie gebruikt moet worden: gebruik sshkeygen –t dsa om een DSA-key te genereren en sshkeygen –t rsa om een RSA key te genereren. De sleutels worden vervolgens opgeslagen in twee bestanden: de private key komt terecht in het bestand ~/.ssh/identity en de public key in ~/.ssh/identity.pub. In de onderstaande afbeelding ziet u wat er tijdens het aanmaken van het sleutelpaar allemaal gebeurt. Let vooral even op het moment waar een passphrase gevraagd wordt, hier kunt u natuurlijk een wachtzin invoeren om uw sleutels mee te beschermen. Om het gebruik van de sleutels te vereenvoudigen, kunt u er ook voor kiezen de wachtzin leeg te laten door op Enter te drukken wanneer er om gevraagd wordt.

***sshkeugen.tif Iedere gebruiker kan voor zichzelf een SSH sleutelpaar aanmaken.

De pubic key naar de server verplaatsen.

Nadat de gebruiker op zijn werkstation een sleutelpaar heeft aangemaakt, moet hij ervoor zorgen dat de public-key ook op de server terecht komt. Hier heeft de gebruiker in zijn homedirectory een bestand met de naam ~/.ssh/authorized_keys. In dit bestand komen alle public keys voor die die gebruiker mag gebruiken om zich aan te melden. Om de public key naar de server toe te kopiëren kan bijvoorbeeld gebruikgemaakt worden van **scp**, geef **scp .ssh/id_dsa.pub 192.168.0.6:fedorapubkey**. Dit maakt op de SSH-server een bestand aan waarin de public key voorkomt. Als dit de eerste keer is dat de gebruiker contact opneemt met de server, krijgt hij nu eerst eenmelding dat de identiteit van de server niet bevestigd kan worden. Geef op deze melding de tekst “yes” in om ervoor te zorgen dat de RSA key fingerprint van de server lokaal in de homedirectory van de client wordt opgeslagen, dit zorgt ervoor dat hij deze vraag de volgende keer niet meer krijgt. Voer vervolgens het wachtwoord in van de gebruiker zoals het op de SSH-server gedefinieerd is en het bestand kan naar de server in kwestie gekopieerd worden.

Als u de public key naar een nieuw bestandje op de SSH-server hebt gekopieerd, bent u er nog niet. U moet er namelijk voor zorgen dat de public key terecht komt in het bestand ~/.ssh/authorized_keys. Om de public key uit het bestand fedorapubkey naar dit

authorized_keys bestand te kopiëren, geeft de gebruiker nu vanuit zijn homedirectory op de server de opdracht **cat fedorapubkey >> ~/.ssh/authorized_keys**.
Als de public key op de server naar het juiste bestand is gekopieerd, bent u klaar. U kunt nu vanaf de SSH-clientcomputer testen of het werkt door een SSH-verbinding met de server op te bouwen. Hebt u ervoor gekozen uw SSH-key met een passphrase te beschermen? Dan moet u eerst deze passphrase invoeren. Als u op Enter hebt gedrukt toen gevraagd werd een passphrase aan te maken, kunt u zonder dat er iets gevraagd wordt contact maken met de server.

De passphrase opslaan
U hebt er nu voor gezorgd dat u op basis van public keys in kunt loggen. Om uw private key echter goed te beschermen, hebt u deze voorzien van een passphrase. Deze passphrase moet wel elke keer worden ingevoerd voordat u contact kunt maken met de SSH-server. Ook zijn er andere momenten waarop u nog gewoon een wachtwoord in moet voeren, bijvoorbeeld wanneer u met behulp van **scp** een bestand wilt kopiëren naar de server. Dit probleem lost u op met behulp van **ssh-agent**. Met behulp van dit programma kunt u wachtwoorden bewaren voor bepaalde omstandigheden. De onderstaande opdrachten laten u zien hoe u dit kunt doen vanaf de SSH-client zodat u nooit meer een passphrase hoeft in te voeren, maar wél gebruikmaakt van een passphrase zodat uw privte key goedbeschermd is.

ssh-agent bash
ssh-add .ssh/id_dsa
Enter passphrase for .ssh/id_dsa:
Identity added: .ssh/id_dsa (.ssh/id_dsa)

U zult nu merken dat de volgende keer dat u contact maakt met de SSH-server, geen passphrase meer ingevoerd hoeft te worden. U hebt nu echter met ssh-agent de passphrase verbonden aan uw bash-environment. Dat is leuk, maar in de echte wereld werken mensen niet vanuit een bash-shell maar vanuit een grafische omgeving. Om ervoor te zorgen dat ook dan de gecachede passphrase automatisch gebruikt kan worden, moet u het configuratiebestand aanpassen dat geladen wordt wanneer u de X display manager activeert om in te loggen. Op SUSE Linux wordt het bestand /etc/X11/xdm/sys.xsession voor dat doel gebruikt. Neem in dit bestand de regel **usessh="yes"** op, dit zorgt ervoor dat ssh-agent automatisch gestart wordt wanneer de grafische gebruikersomgeving geactiveerd wordt. U moet alleen nog wel éénmalig met de opdracht **ssh-add** de passphrase invoeren zodat deze op de juiste wijze gecached wordt.

7.5.3 Installatie en configuratie van VPN-verbindingen

In de meeste gevallen zullen de pakketjes die vanaf je netwerk verstuurd worden gewoon over het Internet verstuurd worden zonder dat er iets extra's gebeurd. Soms is dat echter gewoon niet goed genoeg. Denk daarbij aan de situaties waarin vertrouwelijke bedrijfsgegevens verstuurd moeten worden. Een VPN biedt in dergelijke gevallen een oplossing. Op Linux zijn er verschillende manieren waarop een VPN-verbinding tot stand gebracht kan worden.

Het begrip VPN is een zeer breed begrip. VPN wil namelijk niets meer of minder zeggen dan dat het gaat om een versleutelde beveiligde verbinding tussen twee nodes en/of netwerken. Er zijn verschillende manieren waarop een VPN-verbinding tot stand gebracht kan worden. Een populaire manier op het Linux platform is de methode waar gebruik gemaakt wordt van SSH en PPP. Hierbij wordt SSH gebruikt om de versleutelde tunnelverbinding tussen twee nodes tot stand te brengen en wordt vervolgens pppd gebruikt om het verkeer door die tunnel heen te

sturen. Aan de client zijde moet hiervoor gebruik gemaakt worden van pppd in combinatie met een speciale utility die er voor zorgt dat de ssh verbinding als een soort seriële lijn gebruikt wordt. Aan de server kant vervolgens draait ook pppd in de ssh sessie om de verbinding te voltooien. Wanneer dit systeem is opgezet, hoeft er vervolgens alleen nog maar gerouteerd te worden. In deze paragraaf leert u hoe u dit type VPN-verbinding kunt configureren.

Onafhankelijk van de protocollen die u gebruikt, is er nog een ander onderscheid bij het configureren van een VPN verbinding. Dit is het verschil tussen de VPN verbinding die twee netwerken aan elkaar verbindt en de VPN verbinding waarmee een gebruiker contact kan maken met een netwerk. Beide soorten worden namelijk op een iets andere wijze ingericht. Als u er van uitgaat dat een satellietvestiging aan een hoofdkantoor verbonden moet worden, hebt u daarvoor vanuit VPN-optiek twee routers nodig. Een van deze routers gedraagt zich als client router, de ander als server. De client is de router die zich op het bijkantoor bevindt. In de meeste gevallen wilt u op deze router niet alleen een beveiligde VPN verbinding vanaf de router naar het hoofdkantoor, maar ook nog een normale internetverbinding. Dit betekent dat u in elk geval op deze router twee routes aan zult moeten maken: een route voor al het VPN verkeer en een default route voor al het overige verkeer.

Naast de router op het bijkantoor, komt er ook een VPN-router voor op het hoofdkantoor. Deze router zal standaard zo geconfigureerd zijn dat alle verkeer vanaf de VPN verbinding binnengelaten mag worden, terwijl het verkeer dat binnenkomt via andere adressen geweerd moet worden. In veel gevallen zal dit ook een gespecialiseerde router zijn die niets anders doet als regelen van VPN-verkeer. We zullen deze centrale router aanduiden als de VPN-server.

7.5.3.1 Installatie van het ppp / ssh VPN op de server
Voordat u kunt beginnen met de daadwerkelijke installatie van het VPN op de server, moet u eerst de nodige beveiligingsmaatregelen nemen. Om te beginnen bestaan deze eruit zoveel mogelijk processen niet te draaien. De reden hiervoor is simpel: elk proces dat op uw server actief is, levert een mogelijk gevaar op. Aangezien de server aan de onveilige kant van de firewall staat en dus direct vanaf het internet te bereiken is, is het aan te raden er geen enkele daemon op te draaien behalve dat wat absoluut noodzakelijk is om het VPN tot stand te brengen.

Het tweede punt dat van belang is, heeft te maken met wachtwoorden van gebruikers. De regel hiervoor is eenvoudig: gebruikers die binnenkomen via het VPN mogen geen gebruik maken van wachtwoorden. Wachtwoorden kunnen immers gekraakt worden; om die reden kunnen gebruikers beter gebruik maken van SSH keys. In de voorgaande paragraaf hebt u uitgebreid kunnen lezen hoe u dit kunt configureren. Om ervoor te zorgen dat geen wachtwoorden gebruikt mogen worden, moet u op de positie van het wachtwoord veld in de gebruikersdatabase /etc/passwd een * neerzetten.

Ook is het handig ervoor te zorgen dat de default Shell van gebruikers wordt aangepast: u maakt geen gebruik van de standaard Shell /bin/bash waarmee u interactieve Shell logins toestaat, maar u specificeert in plaats daarvan /usr/sbin/pppd als Shell voor VPN gebruikers. Hiermee zorgt u ervoor dat geautoriseerde gebruikers niets anders kunnen doen dan een VPN verbinding tot stand brengen op basis van pppd. Zeker wanneer uw server speciaal is ingericht voor het opzetten van VPN verbindingen, hebben gebruikers er verder ook helemaal niets te zoeken en is er dus geen enkel bezwaar om de gebruiker met pppd als Shell te laten werken.

Het gaat immers uitsluitend om het tot stand brengen van een beveiligde remote netwerkverbinding en niets meer dan dat. We willen het gebruikers bijvoorbeeld niet toestaan ook daadwerkelijk in te loggen op de VPN-server.

Als voorbeeld hiervan ziet u hieronder een paar regels uit het gebruikersconfiguratiebestand /etc/passwd. In deze regels wordt eerst een tweetal normale gebruikers gedefinieerd (die dus op de server in mogen loggen). Vervolgens wordt een aantal gebruikers gedefinieerd dat alleen via de VPN verbinding naar binnen mag. Om een dergelijke configuratie tot stand te brengen is het overigens handig geen gebruik te maken van standaard gereedschap als useradd, maar even het passwd bestand handmatig te bewerken. Let daarbij goed op typefouten, u kunt namelijk door een tikfout uw hele password bestand ongeldig maken en dan maakt u het knap lastig achteraf nog in te loggen.

```
eric:x:501:100::/home/eric:/bin/bash
kees:x:502:100::/home/kees
bouke:*:503:101::/home/vpn-sers:/usr/sbin/pppd
alex:*:503:101::/home/vpn-users:/usr/sbin/pppd
```

Let er even op dat het aan te raden is alle gebruikers lid te maken van één groep die bijvoorbeeld de naam “vpnusers” kan heten. Deze groep kunt u gewoon met het commando **groupadd** aanmaken. In het voorgaande voorbeeld gaan we er van uit dat hiervoor de groep met GID 101 gebruikt wordt. Tevens moet u voor de VPN-users een homedirectory aanmaken. In deze homedirectory moet in elk geval een bestand .ssh/authorized_keys voorkomen. Hierin komen de SSH-keys voor die nodig zijn om met succes in te kunnen loggen. Let er tevens op dat dit een gedeelde homedirectory is. Dit is ook geen probleem, want gebruikers gaan als het goed is toch geen bestanden opslaan op de VPN-server.

Nadat u /etc/passwd op de juiste manier bewerkt hebt, moet u ervoor zorgen dat de gebruikers via ssh binnen kunnen komen. Dit regelt u met behulp van het sshd configuratiebestand /etc/sshd_config. Hierin moeten in elk geval de volgende parameters opgenomen zijn:

```
PermitRootLogin yes
IgnoreRhosts yes
StrictModes yes
QuietMode no
CheckMail no
IdleTimeout 3
X11Forwarding no
PrintMotd no
KeepAlive yes
RhostsAuthentication no
RhostsRSAAuthentication no
PubkeyAuthentication yes
PasswordAuthentication no
PermitEmptyPasswords no
UseLogin no
```

De belangrijke regels in het bovenstaande voorbeeld bestaan er uit dat gewone gebruikers niet in mogen loggen met behulp van het gebruikersnaam / wachtwoord mechanisme, maar dat er gebruik gemaakt moet worden van Public Key Authentication. De overige regels zorgen

ervoor dat het proces verder verfijnd wordt. Zo wordt er bijvoorbeeld zorg voor gedragen dat er geen Message Of The Day geprint wordt na het inloggen van gebruikers wat ook erg voor de hand ligt omdat dit geen interactieve Shell login in, maar een speciale login voor de PPP daemon.

Nadat u bepaald hebt dat de gebruikers alleen op een veilige wijze binnen mogen komen, moet u er voor zorgen dat de netwerkfunctionaliteit op de juiste wijze geregeld is. Dit betekent dat de VPN-server ingericht moet worden als router en dat de juiste firewall regels gedefinieerd moeten worden. Hiervoor moeten in elk geval de volgende kernel opties aan staan:

CONFIG_FIREWALL
CONFIG_IP_ADVANCED_ROUTER
CONFIG_IP_FIREWALL
CONFIG_IP_ROUTER
CONFIG_IP_MASQEURADE
CONFIG_IP_MASQUERADE_ICMP
CONFIG_PPP

Vervolgens moet u ervoor zorgen dat de firewall op de juiste manier wordt ingericht. De volgende regels zorgen er voor dat geen verkeer wordt toegestaan op de tunnel, behalve dat verkeer dat gericht is aan de andere zijde van de VPN-tunnel. We gaan hier voor het gemak even uit van het lokale IP-netwerkadres 192.168.1.0 en het remote IP-netwerkadres 172.16.1.0, pas deze regels aan zodat ze toegepast kunnen worden op de situatie van uw eigen netwerk.

iptables -P FORWARD DROP
iptables -A FORWARD -j ACCEPT -s 192.168.1.0/24 -d 172.16.1.0/24

Nadat u de firewall geconfigureerd hebt, moet u aangeven hoe het verkeer over de juiste verbinding gerouteerd moet worden. Als eth1 de netwerkkaart is waarover het VPN verkeer verstuurd moet worden, gebruikt u hiervoor het volgende commando:

route add -net 172.16.1.0 netmask 255.255.255.0 gw 172.16.1.254 eth1

Uiteraard zult u deze regels aan moeten passen zodat ze overeenkomen met de situatie in uw netwerk.

Wanneer het bovenstaande allemaal gebeurd is, kunt u de ppp daemon starten. Daaraan voorafgaand moet u wel eerst een PPP-configuratie definiëren. Doe dit met behulp van het bestand/etc/ppp/options. Het volstaat wanneer hier de volgende vier regels in voorkomen:

ipcp-accept-local
ipcp-accept-remote
proxyarp
noauth

De eerste twee regels in dit bestand zorgen er voor dat de server elk IP-adres accepteert waarvan de client gebruik wil maken. De derde regel zorgt dat de client voorkomt alsof hij op het lokale netwerk aanwezig is en de laatste regel tot slot vertelt pppd dat hij moet werken

zonder gebruikersnaam en wachtwoord. Bent u gewend dat dit bestand er normaal veel ingewikkelder uit ziet? Dat kan kloppen. Nu is dat echter niet nodig, er hoeft immers geen gebruik gemaakt te worden van gebruikersauthenticatie.

De client
Ook op de client moeten meerdere zaken geregeld worden. Om te beginnen is dat een netfilter dat er voor zorgt dat het juiste verkeer naar het VPN wordt doorgestuurd.

```
iptables -P FORWARD DROP
iptables -A FORWARD -j ACCEPT -s 192.168.10./24 -d 172.16.1.0/24
iptables -A FORWARD -s 0/0 -d 192.168.1.0/24 -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT
```

Deze regels zorgen ervoor dat het IP-verkeer twee kanten op door de firewall mag gaan; van de client naar de server en weer terug.

Wanneer u zo ver bent, kunt u verder gaan met de configuratie van pppd op de client. Het prettige is dat het hier niet nodig is een /etc/ppp/options aan te maken, u hebt namelijk geen opties nodig. Mocht pppd weigeren op te starten omdat dit bestand niet bestaat, dan kunt u een leeg bestand aanmaken met de naam /etc/ppp/options.

Wanneer u ervoor gezorgd hebt dat wat betreft pppd alles in orde is, moet u ervoor zorgen dat u ook gebruik kunt maken van ssh. Hiervoor moet u ssh keys aanmaken. Raadpleeg de voorgaande paragraaf voor uitgebreide informatie over de wijze waarop dit moet gebeuren.

Wanneer aan alle bovenstaande voorwaarden voldaan is, wordt het tijd de VPN-verbinding up te brengen. Voordat u dit kunt doen, hebt u het hulpprogramma pty-redir nodig. Met behulp van dit commando kan de output van het ssh-commando omgeleid worden naar een speciaal device dat zich gedraagt als een serieel device. Dit device is nodig om pppd te kunnen starten. Pty-redir maakt geen deel uit van de meeste distributies. U kunt het downloaden van http://www.shinythings.com/pty-redir. Met behulp van de volgende commando's kunt u de VPN-verbinding vanaf de client kant in de lucht krijgen. Let er even op dat gebruikgemaakt wordt van de juiste verwijzingen naar de bestanden die gebruikt moeten worden.

```
# ssh naamvandevpnserver
# /usr/sbin/pty-redir /usr/sbin/ssh -t -e none -o 'Batchmode yes' -c blowfish –i \
/root/.shh/identity.vpn -l bouke > /tmp/vnp-device
# sleep 10
# /usr/sbin/pppd `cat /tmp/vpn-device`
# sleep 15
# /sbin route add -net 172.16.1.0 gw naamvandevpnserver netmask 255.255.255.0
```

De bovenstaande commando's kunnen achter elkaar in een script geplaatst worden (denk er dan wel aan het # teken te verwijderen aan het begin van elke regel. Let er in de eerste regel even op dat hier contact gemaakt wordt als lokale gebruiker Bouke. Zoals u eerder hebt kunnen lezen, is deze in /etc/passwd op de VPN-server aangemaakt als speciale VPN-gebruiker. Wanneer u de bovenstaande procedure hebt uitgevoerd, moet alles werken.

Oefening 7.5

Om de onderstaande oefening uit te kunnen voeren hebt u twee computers nodig. Er wordt gevraagd naar wat informatie die u niet letterlijk uit de tekst kunt halen, met behulp van de juiste man-pagina moet u hier echter uit kunnen komen.
Configureer een SSH-server en een SSH-client. Zorg ervoor dat gebruiker Sanne vanaf de VPN-client in kan loggen op de SSH-server. Hiervoor wordt gebruikgemaakt van public-key authenticatie. Elke andere vorm van authenticatie moet expliciet onmogelijk gemaakt worden, zet dus bijvoorbeeld password authenticatie uit. Zorg er tevens voor dat alleen toegang wordt verleend aan clients die gebruikmaken van een programma dat SSH versie 2 aankan, versie 1 is immers onveilig. Genereer vervolgens RSA key's waarmee Sanne in kan loggen en bescherm de private key met het wachtwoord l1nu+. Zorg er tot slot voor dat Sanne nooit haar wachtwoord in hoeft te voeren, maar gewoon automatisch op basis van haar sleutels contact kan maken met de server. Test vervolgens of dit werkt door met behulp van scp een aantal bestanden vanaf de SSH-client te kopiërn naar Sanne's home directory op de SSH server.

7.6 Pro-actief beheer

Tot nu toe hebben we in dit hoofdstuk gesproken over manieren waarop u ervoor zorgt dat uw server beter beveiligd wordt. Door deze manieren toe te passen is de beveiliging van uw server aanzienlijk toegenomen, u bent er echter nog niet. Wat vandaag een veilige situatie lijkt, kan morgen namelijk blijken absoluut onveilig te zijn. Om ervoor te zorgen dat de beveiliging van uw netwerk werkelijk gegarandeerd is, moet u ook iets doen aan pro-actief beheer. Dit betekent dat u actief in de gaten moet houden welke beveiligingsproblemen er actueel zijn en dat u ook actief in de gaten moet houden door welke beveiligignsrisico's uw netwerk bedreigd wordt. Soms betekent dat dat u als een hacker moet proberen binnen te dringen op uw eigen netwerk. Aan de andere kant betekent het dat u door middel van het juiste gereedschap ervoor gaat zorgen dat anderen dat juist niet kunnen doen. In deze laatste paragraaf van dit hoofdstuk over beveiliging, maakt u kennis met enkele van de bekendste middelen die u hiervoor kunt gebruiken.

7.6.1 Ontvangen van security alerts

Het is natuurlijk niet te doen om zelf in de gaten te hoden wanneer er ergens serieuze problemen bekend worden in de software die u gebruikt. Gelukig is er een aantal manieren om ervoor te zorgen dat u bij blijft:

* **De Bugtraq mailinglist**. U vindt de Bugtraq mailinglist op www.securityfocus.com. Wanneer u zich hierop inschrijft, krijgt u bij elke zwakheid die ontdekt wordt een mailbericht. Daarbij wordt ook aangegeven wat u kunt doen om de zwakke plek zelf te repareren. Om u in te schrijven op deze mailinglist, stuurt u een bericht naar bugtraq-subscribe@securityfocus.com. U krijgt als reactie een mail waarin staat hoe u verder moet handelen. Volg deze aanwijzingen op om u in te schrijven.
* **CERT** Het CERT Coördinatie Centrum is een centrum dat zich toelegt op beveiliging in relatie tot het internet. Deze organisatie wordt hiervoor door de Amerikaanse overheid gesponsord. Als er iets mis is op het gebeid van beveiliging, vindt u dat terug op de website van het CERT: www.cert.org. Ook kunt u zich inschrijven op een mailinglist om altijd op de hoogte gesteld te worden van belangrijke bedreigingen op het gebeid van beveiliging.

***cert Via de website van CERT bent u altijd op de hoogte van de laatste bedreigingen op het gebied van beveiliging.

* **SANS** Een beroemdheid op het gebied van security, is het SANS. Deze organisatie is vooral bekend vanwege de SANS top 20, dit is een lijst met de 20 meest gevaarlijke bedreigingen die er momenteel zijn voor de beveiliging van uw netwerk. Tevens organiseert

SANS regelmatig seminars en cursussen op het gebeid van beveiliging. Meer informatie over SANS vindt u op www.sans.org.

7.6.2 Traceren van netwerkverkeer

Een onderdeel van het beveiligen van uw netwerk bestaat eruit te bekijken wat er aan verkeer verstuurd wordt op uw netwerk. Hiervoor kan een drietal tools ingezet worden: tcpdump is de meest basic packetsniffer die u in kunt zetten. Als u het commando activeert, zult u zien dat als resultaat alle pakketjes die op het netwerk voorbij komen op uw console gedumpt worden – het commando heet tenslotte niet voor niets tcpdump. Met behulp van de nodige parameters (het blijft Linux tenslotte) kunt u deze stroom van gegevens beïnvloeden. Niet de meest eenvoudige werkwijze, maar wel erg effectief. Een veel efficiëntere sniffer die ingezet kan worden is Ethereal: deze sniffer helpt u met een grafische interface uitgebreid pakketjes te sniffen en op te slaan om ze vervolgens op uw gemak te analyseren.

***tcpdump Het commando tcpdump zorgt ervoor dat de inhoud van alle pakketjes die op uw netwerk verstuurd worden op uw scherm gedumpt worden.

7.6.2.1 Ethereal

Voordat u met sniffers aan het werk gaat, is het zaak te begrijpen op basis van welk principe een sniffer werkt. U kunt dan namelijk ook inschatten wanneer uw sniffer het wel zal doen en onder welke omstandigheden hij het niet doet. De basis van een sniffer op een Ethernet netwerk, is dat op een dergelijk netwerk oorspronkelijk alle nodes alle pakketjes voorbij zagen komen. Een Ethernet node is echter zo geprogrammeerd dat alleen die pakketjes die aan hemzelf geadresseerd zijn ook binnengehaald worden. De node doet dit op basis van het MAC-adres dat hij herkent. Wanneer de netwerkkaart in “promiscuous mode” (letterlijk: overspelige modus) gezet wordt, doet hij het met iedereen en is hij in staat alle pakketjes die op het netwerk verzonden worden binnen te halen en te analyseren. Op een Linux-systeem is het hiervoor nodig dat de libpcap module geïnstalleerd is.

Sniffers die op zo’n manier werken zijn zoals gezegd gebaseerd op het principe dat elke node elk pakketje voorbij ziet komen. Dit principe echter was ook een van de meest zwakke punten op een Ethernet netwerk: het feit dat pakketjes die verstuurd worden door iedereen zichtbaar zijn, zorgt er namelijk ook voor dat er nogal eens pakketjes op elkaar botsen en er een zogenaamde collision optreedt. Die collisions zorgen er voor dat pakketjes opnieuw verstuurd worden en doen dus de performance van het netwerk drastisch afnemen. Om die reden wordt op moderne Ethernet netwerken niet langer gebruik gemaakt van een gedeeld medium, maar van switches die er voor zorgen dat een relay wordt omgezet zodat de zender en ontvanger rechtstreeks met elkaar communiceren. Dit is zeer gunstig voor het gebruik van de bandbreedte op uw netwerk, het heeft echter wel het nadeel dat niet langer alle pakketjes die op het netwerk verstuurd worden door elke node gezien worden. Dit betekent dat deze pakketjes ook niet langer gesnift kunnen worden. Wanneer het er echter om gaat het verkeer te analyseren dat door een bepaalde server ontvangen en verstuurd wordt, hebt u van dit probleem geen last.

Er zijn verschillende sniffers die gebruikt kunnen worden. Sommigen zoals SnifferPro zijn commercieel verkrijgbaar en ook behoorlijk prijzig, anderen zoals tcpdump zijn gratis. Een van de beste sniffers die op dit moment bestaan, is het gratis programma Ethereal. Deze sniffer wordt met veel distributies standaard mee geïnstalleerd, goed nieuws als u Windows gebruikt, er is namelijk zelfs ook een Windows-versie van. Nadat u het programma gestart hebt, kunt u er direct mee aan het werk.

1. Klik op het menu **Analyze** en selecteer de optie **Enabled Protocols**. Er verschijnt nu een venster waarin u aan kunt geven welke protocollen u allemaal wilt analyseren. Standaard staan alle protocollen aan, op een druk netwerk kan dit er echter toe leiden dat het resultaat van uw capture heel erg druk wordt.
2. Ga nu naar het menu **Capture** en selecteer de optie **Start**. Er verschijnt nu een venster waarin u aan kunt geven hoe de capture gestart moet worden. Zorg er voor dat u om te beginnen de juiste netwerk interface selecteert achter de optie **Interface**. Vervolgens moet u er ook voor zorgen dat de optie **Capture packets in promiscuous mode** aan staat. Zonder deze optie bent u namelijk alleen maar in staat pakketjes te analyseren die door de host zelf verzonden worden. De overige opties zijn optioneel. Wellicht dat u iets hebt aan de opties onder **Capture limits**. Hiermee kunt u er namelijk voor zorgen dat na een bepaalde periode de capture automatisch stop gezet wordt.
3. Klik nadat u alle gewenste opties hebt aangegeven op **OK** om te beginnen met binnenhalen van de pakketjes. Er verschijnt nu een venster waarin u ziet hoeveel pakketjes van welk type er binnengehaald zijn. Houdt er rekening mee dat deze buffer op een druk netwerk zeer snel vol kan lopen! Druk op **Stop** wanneer u wilt stoppen met binnenhalen van pakketjes.
4. Klik op **Stop** wanneer er zich voldoende pakketjes in de buffer bevinden. Het Ethereal scherm wordt nu gevuld met alle pakketjes die in de buffer zijn ingelezen. Deze pakketjes worden in chronologische volgorde getoond. Wilt u liever dat de pakketjes per pakketsoort geordend zijn? Klik dan op de tabel **Protocol** dit zorgt ervoor dat u alle pakketjes van hetzelfde protocoltype bij elkaar ziet staan.

U hebt nu Ethereal geconfigureerd en draaiend. Dat is leuk, maar het echte werk begint nu pas. De volgende stap bestaat eruit te bestuderen welke pakketjes er door uw server verstuurd worden en vooral te bekijken hoe leesbaar de informatie in deze pakketjes is. Dacht u dat u veilig aan het communiceren was met uw mailserver? Zoek dan bijvoorbeeld eens de POP en SMTP pakketjes die naar deze server verstuurd worden.

Wanneer u meer wilt weten over de manier waarop gecommuniceerd wordt, is het ook de moeite een pakketje aan te klikken. U kunt dan in de onderste twee delen van het scherm zien wat er precies gebeurt; in het middelste scherm is het mogelijk elk veld dat in de protocolheaders gebruikt wordt te bekijken. Wilt u in meer detail zien wat er gebeurt? Klik dan een veld aan. U ziet vervolgens in het onderste deel van het scherm hoe dit veld er in het totaal van het pakketje uitziet.

7.6.2.2 Ettercap

Zoals u in de voorgaande paragraag hebt kunnen lezen, is er een probleem met het sniffen van pakketjes vanaf een node die aangesloten is op een switch. De reden hiervoor is dat pakketjes die op een switch verstuurd worden niet langer voor alle nodes zichtbaar zijn, maar direct doorgestuurd worden naar de uiteindelijke bestemming. De switch doet dit door even een relay om te zetten en dit betekent dat de node waarop een sniffer draait de pakketjes niet voorbij ziet komen en dus ook niet in staat is deze pakketjes te analyseren.

Voor dit probleem is gelukkig een oplossing: het programma Ettercap. Dit programma zorgt ervoor dat alle verkeer wordt “afgetapt” door de node waarop de sniffer draait. De basis hiervoor is een trucje dat bekend staat als “ARP poisoning”. Voordat we kunnen bespreken hoe u Ettercap in uw netwerk in kunt zetten, zullen we eerst kort bespreken wat nu eigenlijk ARP-poisoning is.

Het protocol ARP is een belangrijk hulpmiddel dat nodig is voordat twee nodes contact met elkaar op kunnen nemen. Stel bijvoorbeeld dat u het commando **ping 193.173.97.45** geeft. Voordat het protocol ICMP daadwerkelijk zijn werk kan doen en pakketjes kan gaan versturen, moet eerst bekend zijn welk MAC-adres bij 193.173.97.45 hoort. Hiervoor wordt gebruik gemaakt van het protocol ARP, wat letterlijk staat voor Address Resolution Protocol; ARP verstuurt een broadcast met daarin "Who has 193.173.97.45? Tell 193.173.97.34". Hierin wordt de betreffende node verzocht even een pakketje terug te sturen. Dit gebeurt en nu weet 193.173.97.34 welk MAC-adres hij nodig heeft. Vervolgens kunnen pakketjes verstuurd worden.

Wanneer eenmaal bekend is welk MAC-adres bij een bepaald IP-adres hoort, wordt deze informatie opgeslagen in de ARP-cache. Dit zorgt ervoor dat uw pc niet elke keer wanneer er een pakketje verstuurd moet worden eerst ARP moet gebruiken om te achterhalen welk MAC-adres hiervoor nodig is.

De werking van Ettercap is gebaseerd op het om de tuin leiden van andere nodes op het netwerk door middel van ARP. Ettercap doet dit door middel van ARP poisoning, letterlijk: het vergiftigen van de ARP-cache. Dit komt er op neer dat uw node een connectie kaapt. Stel u bijvoorbeeld voor dat u wilt zien wat er allemaal gebeurt tussen 193.173.97.45 en de default gateway 193.173.97.62. Door middel van Ettercap kunt u er voor zorgen dat alle verkeer omgeleid wordt via uw eigen pc. Hoe dat gebeurt? Heel eenvoudig: Ettercap gaat de ARP-tabel van 193.173.97.45 vervalsen en zorgt ervoor dat niet het daadwerkelijke MAC-adres van de default gateway in die ARP-tabel komt te staan, maar het MAC-adres van uw pc. Het resultaat is dat de argeloze 193.173.97.45 alle pakketjes doorstuurt naar uw pc. Die stuurt ze op zijn beurt natuurlijk wel weer gewoon door naar de uiteindelijke bestemming, anders zou het grapje immers snel uitkomen.

Met Ettercap kaapt u dus als het ware een connectie. Dit zorgt ervoor dat alle pakketjes die een bepaalde host verstuurd, via uw netwerkkaart lopen. Het kan overigens ook uitgebreider dan gewoon één connectie te kapen, u kunt ook gebruik maken van een optie die er voor zorgt dat niet één maar alle connecties afgeluisterd kunnen worden. Verderop kunt u lezen wat hier voor moet gebeuren. Wanneer u gebruik maakt van Ettercap om alle verkeer om te leiden naar uw pc, is het overigens nog wel aan te raden om daarnaast gebruik te maken van een sniffer als Ethereal. Ettercap heeft weliswaar een eenvoudige sniffer in zich, deze is echter behoorlijk beperkt en stelt u niet in staat om een zorgvuldige analyse te doen van uw netwerkverkeer.

In de volgende procedure kunt u lezen hoe u Ettercap kunt gebruiken om uw netwerk te analyseren.

1. Start uw browser en ga naar ettercap.sourceforge.net. Gebruik de link download om Ettercap te downloaden op uw pc en volg de eenvoudige installatieprocedure om het programma op uw pc te installeren.
2. Na installatie geeft u de opdracht **ettercap** om het programma te installeren. Selecteer nu de netwerkkaart waarop Ettercap zijn werk moet gaan doen. Vervolgens zal het programma door middel van ARP-pakketjes achterhalen welke nodes er allemaal op het netwerk actief zijn. Het resultaat van deze actie wordt in een venster getoond.
3. Het is de moeite waard in het Ettercap venster op F1 te drukken. U krijgt nu een overzicht van alle opties die beschikbaar zijn. Houdt er overigens rekening mee dat deze hulp

context gevoelig is,u zult zien dat deze opties dus veranderen naar gelang u andere opties geselecteerd hebt.
4. Selecteer in de linker kolom van het Ettercap scherm een IP-adres en doe dit ook in de rechter kolom. Druk vervolgens op de toets **s** om de connectie tussen deze twee hosts te kapen. Wanneer u nu uw sniffer start, zult u precies zien wat er allemaal aan verkeer tussen beide hosts verstuurd wordt.

***gekaapt Nadat u een connectie gekaapt hebt, kunt u in uw sniffer precies zien wat er allemaal aan verkeer tussen twee hosts verstuurd wordt.

7. Druk nu op de toets **o**, Dit zorgt ervoor dat naast deze specifieke connectie ook alle andere verkeer dat over uw switch verstuurd wordt, wordt afgeluisterd.

7.6.3 Scannen voor openstaande poorten

Om te testen of de beveiliging op uw netwerk goed geregeld is, is het zeer de moeite waard om zo af en toe eens uw eigen netwerk te proberen te hacken. In dit geval betekent dat dat u gebruik kunt maken van de open source tool nmap om te bepalen welke computers up zijn in uw netwerk en welke diensten door deze computers worden aangeboden. De kracht van nmap is dat het een programma is dat door kwaadwillenden ook gebruikt kan worden om uw netwerk aan te vallen: u analyseert met nmap dus uw netwerk op dezelfde wijze als dat een aanvaller dat doet. Nmap biedt daarvoor ondersteuning voor een groot aantal scan-technieken. Het enige nadeel echter van het gebruik van nmap is dat u kennis van zaken nodig hebt. Aangezien nmap op verschillende manieren pakketjes kan verzenden om te analyseren op zwakheden, moet u goed weten hoe IP, ICMP, UDP en TCP pakketjes verstuurd worden over een netwerk. In de onderstaande tabel vindt u een overzicht van de belangrijkste scanmethodes die door nmap gebruikt kunnen worden.

7.6.3.1 Nmap scanmethodes

De onderstaande tabel toont een overzicht van alle manieren waarop u nmap in kunt zetten om het netwerk te scannen op zwakke plekken. In de tabel vindt u eerst de naam van het type scanmethode, vervolgens het argument dat u moet gebruiken bij het activeren van nmap en tot slot een beschrijving van de scanmethode.

<<OPMERKING VOOR REDACTIE: KAN DIT OPGEMAAKT WORDEN ALS TABEL MET 3 KOLOMMEN>>

Type	**Argument**	**Description**
Connect	-sT	Poortscan waarbij gebruikgemaakt wordt van een volledige TCP-connectie waarbij alle drie stappen van de threeway handshake zijn doorlopen. Dit is de enige manier waarop een gebruiker die niet root is een scan kan uitvoeren.
Stealth SYN scan	-sS	Hiermee wordt alleen het eerste pakketje uti de TCP threeway handshake verstuurd: het SYN-pakketje. Als hier antwoord op komt, weet de aanvaller dat de betreffende poort open staat. Hij bouwt echter geen connectie op waardoor zijn scanpoging op de aangevallen computer in de meeste gevallen onzichtbaar blijft.
FIN	-sF	Hierbij wordt een FIN pakketje verstuurd. Als een RST als antwoord volgt, staat de poort dicht. Komt er niets terug, dan staat de poort open. Deze wijze van scannen werkt niet altijd goed op Windows targets.
Xmas Tree	-sX	Variant op de FIN scan. In een Xmas Tree scan worden echter de URG en de PUSH flags ook aangezet om de kans op succes te vergroten.
Null	-sN	Weer een variant op de FIN scan waarbij geen gebruikgemaakt wordt van extra flags.

UDP -sU Bij deze methode wordt naar elke poort een UDP-pakketje van 0 bytes verstuurd. Als een ICMP port unreachable teruggestuurd wordt als reactie, betekent dat dat de poort dicht staat. In het andere geval kan dat een aanwijzing zijn dat de poort open staat. Het probleem met deze manier van scannen is dat er maar een maximaal aantal ICMP foutmeldingen per seconde verstuurd kan worden en hierdoor kan deze wijze van scannen erg langzaam uitpakken.
IP Protocol -sO Bepaalt welke IP-protocollen ondersteund worden. Ook hierbij worden pakketjes van het betreffende protocoltype verstuurd. Volgt hierop een ICMP unreachable foutmelding, dan wordt het protocoltype dus niet ondersteund.
ACK -sA Hiermee kan bepaald worden hoe een firewall geconfigureerd is. Nmap stuurt een ACK pakketje dat normaal verstuurd wordt als antwoord op een TCP SYN pakketje om te bevestigen dat een connectie tot stand gebracht kan worden. Er is echter helemaal geen connectieopbouw geweest door middel van een SYN pakketje, dus zou er een RST pakketje terug moeten komen om aan te geven dat iets niet in de haak is. Als de poort echter gefilterd is door de firewall, komt dit RST pakketje niet.
Window Size -sW Hiermee wordt bepaald op basis van onderhandelingen over TCP window size of een poort open staat of niet. Linux is niet gevoelig voor dit type scan, dus zal er niet op reageren.
RPC -sR Scant op beschikbare RPC poorten (zie hoofdstuk 5)
OS -O Probeert te achterhalen welk besturingssysteem gebruikt wordt.

De verschillende scan types kunnen ook nog ondersteund worden door middel van diverse argumenten. In de onderstaande table vindt u een overzicht van een aantal van de argumenten die voor dit doel kunnen worden ingezet. Er zijn veel meer mogelijkheden, lees de man pagina voor een uitgebreid overzicht:

Argument Beschrijving
-P0 Normaal stuurt nmap een ping pakketje voordat een host gescand wordt. Als op deze ping geen reactie komt, wordt geconcludeerd dat de host niet bereikbaar is. Het kan echter ook zijn dat ICMP ping pakketjes gewoon tegengehouden worden. Om die reden is het aan te raden altijd standaard gebruik te maken van de optie –P0 om aan te geven dat er niet gepingd mag worden.
-I Probeert nadat een connectie is opgebouwd met de target (deze scan werkt dus niet in combinatie met stealth scans) te achterhalen welke gebruikersnaam aan een bepaalde poort gekoppeld is.
-f Gebruik gefragmenteerde pakketjes. Deze optie kan ervoor zorgen dat pakketjes toch stiekum door een filter heen glippen.
-v Wees verbose; dit zorgt dat er veel informatie gegenereerd wordt tijdens het uitvoeren van de nmap-scan.
-vv Wees zeer verbose. Dit level van verbositeit geeft over het algemeen meer informatie dan dat de gemiddelde beheerder aan kan.
-D Maak gebruik van spoofing om pakketjes te versturen namens een aantal niet bestaande hosts. De essentie van deze optie is dat uw host op deze manier verstopt wordt tussen alle niet bestaande hosts.
-T Hiermee kan aangegeven worden hoe vaak een pakketje verstuurd wordt. Dit kan nuttig zijn omdat sommige scan detectors kijken naar een aantal pakketjes van een bepaald type dat binnen een bepaalde periode voorkomt.

7.6.3.2 Gebruik van het commando nmap

Met behulp van de voorgaande opties, kunt u met nmap al heel wat tests uitvoeren. We zullen nu een paar voorbeelden bespreken. Als u dit gaat uitproberen, moet u er overigens rekening mee houden dat het een tijdje kan duren voordat u resultaat te zien krijgt. Zeker bij scans waarbij een hele reeks poorten afgezocht moet worden, kan het voorkomen dat u een tijdje naar een blank scherm zit te kijken. Vindt u het prettig te zien dat er iets gebeurt? Gebruik dan de optie –v bij alle opdrachten om activiteit te laten zien. Om te beginnen een eenvoudige test: met de opdracht **nmap –sT fedoraserver** willen we bekijken welke TCP-poorten er allemaal open staan op de genoemde computer. Een slecht beveiligde machine zal u vervolgens een lijst tonen van alle openstaande poorten. Krijgt u niet direct reactie maar weet u wel zeker dat de host bereikbaar is? Grote kans dat uw poging dan stukloopt op het feit dat nmap eerst probeert hem te pingen terwijl ICMP verkeer geblokkeerd is. Probeer in dat geval **nmap –sT –P0 fedoraserver**.

***nmapst.tif Een slecht beveiligde host zal als reactie op de opdracht nmap –sT gelijk laten zien welke TCP-services hij in de aanbieding heeft.

Bovenstaande opdracht is vrij simpel en geeft niet bijzonder veel informatie. Het wordt al leuker wanneer u wat meer opties meegeeft. Wat dacht u bijvoorbeeld van **nmap –sT –sU –P0 –O linuxlaptop**? Hiermee kijkt u welke TCP poorten open staan, welke UDP poorten bereikbaar zijn, zorgt u ervoor dat er niet gepingd wordt voordat contact met de host gemaakt wordt en tot slot wordt ook nog eens achterhaald welk besturingssysteem op de betreffende host actief is. Houdt er echter rekening mee dat de optie –sU veel tijd nodig kan hebben in verband met het maximale aantal ICMP-foutmeldingen dat per minuut verstuurd kan worden. U doet er daarom goed aan deze opdracht te geven, hem rustig op de achtergrond te draaien en na een tijdje weer eens te kijken of het ook wat heeft opgeleverd. Ondertussen kan het natuurlijk leuk zijn om eens te kijken op welk besturingssysteem Novell nou haar webservers draait: geef hiervoor de opdracht **nmap –O www.novell.com**.

In het voorgaande is maar een zeer beperkt aantal mogelijkheden besproken van scans die u met nmap uit kunt voeren. Het gaat veel te ver om hier alles te bespreken wat mogelijk is. Toch willen we nog een paar opties laten zien. Om te beginnen is dat een gerichte scan waarmee u alleen maar voor een bepaald aantal poorten kijkt of deze open staan. Een dergelijke scan heeft een belangrijk voordeel: u bent namelijk veel eerder klaar als wanneer u stuk voor stuk alle poorten gaat testen. Geef bijvoorbeeld het commando **nmap –P0 –sS –sU –O –p 135,137,138,139,445,25,110** ***ipadresvantarget***. De genoemde poorten staan op de meeste Windows-computers gewoon open. Met behulp van deze test kunt u achterhalen of uw firewall ze netjes afschermt. Bang dat uw pogingen door een oplettend beheerder gedetecteerd worden? Dan is het zaak ervoor te zorgen dat al uw scans stealth worden uitgevoerd. Hiervoor staan u drie verschillende opties ter beschikking, u kunt elk van deze drie opties gewoon toevoegen aan een van de nmap-opdrachten die in het voorgaande besproken is. Het gaat om de opties **–sF** (FIN-scan) **–sX** (X-mas tree scan) of **–sN** (NULL scan). Zeker wanneer u in uw netwerk een IDS hebt (zie volgende paragraaf) is het aan te raden om deze opties eens te gebruiken. Houdt er wel rekening mee dat u maar één van deze drie tegelijk kunt gebruiken, als u ze alledrie gebruikt, werken ze elkaar namelijk tegen.

***vergissing Op een goed beschermde server kan het voorkomen dat nmap zich lelijk vergist. Kijk maar welk besturingssysteem gedetecteerd wordt voor www.novell.com!

Tip! Hebt u het niet zo op het werken vanaf een console prompt? Dan is het de moeite om de Nmap Front End utility nmapfe eens te installeren. U kunt deze utilty downloaden en installeren vanaf de thuisbasis van nmap: www.insecure.org/nmap/nmap_download.html.

7.6.4 Configuratie van een IDS

In het voorgaande hebt u kunnen lezen over de manieren die hackers kunnen gebruiken die het op uw kostbare systemen voorzien hebben. Gelukkig is het ook mogelijk het deze hackers zo moeilijk mogelijk te maken. Dit kan met behulp van een Intrusion Detection System (IDS). Er zijn er twee die vooral de moeite waard zijn: Snort en Tripwire. Snort wordt gebruikt om te kijken of er op het netwerk zaken gebeuren die niet geoorloofd zijn, met tripwire kunt u de integriteit van bestanden in de gaten houden. Dat is een mooie manier om te zeggen dat u met behulp van deze tool kunt zien of er wijzigingen zijn opgetreden in bestanden en als zo'n wijziging voorkomt, actie kunt laten ondernemen. Tussen beide tools zit een belangrijk verschil als het gaat om de toepassing ervan: Tripwire is eigenlijk pas nuttig wanneer het al te laat is. U kunt ermee kijken welke schade door inbrekers is aangericht. Veel interessanter is het natuurlijk om te voorkomen dat mensen überhaupt de kans krijgen dergelijke schade aan te richten. Voor dat doel maakt u gebruik van Snort.

7.6.4.1 Snort

Wanneer mensen het hebben over Snort, wordt vaak verteld dat snort een Intrusion Detection Systeem is. In goed Nederlands zou je het een inbrekersalarm noemen: Snort slaat alarm wanneer er iets onoirbaars op het netwerk gebeurt. Snort is echter veel meer dan dat. Het programma kan voor drie doelen worden ingezet:

* Als packet sniffer om pakketjes van het netwerk af te vangen en te analyseren;
* Als packet logger om pakketjes die gesnift zijn te bewaren op de lokale computer;
* Als Intrusion Detection systeem.

In dit hoofdstuk zullen we het vooral hebben over het laatste aspect van Snort.

Voordat u met Snort aan het werk kunt, moet u de programmabestanden downloaden en installeren, zowel op Fedora als op SUSE zijn ze namelijk niet standaard aanwezig. U kunt ze binnenhalen van www.snort.org. De installatieprocedure zelf is eenvoudig en vindt plaatst zoals deze meestal plaatsvindt wanneer u sourcefiles wilt installeren: geef achtereenvolgens de opdrachten **./configure**, **make** en **make install** en alle snort-bestanden zijn naar uw systeem gekopieerd. Na installatie is er in de meeste gevallen een directory /etc/snort aangemaakt. In deze directory vindt u alle configuratiebestanden die door Snort in gebruik zijn.Drie items in deze directory zijn vooral de moeite waard:

* Het bestand snort.conf; hierin regelt u de totale configuratie van Snort
* De directory rules; hierin komen alle regels voor op basis waarvan Snort zijn werk doet
* Het bestand classification.config; hierin wordt namelijk de ernst van bepaalde regels bepaald.

Voordat u ook maar iets gaat doen, zorgt u ervoor dat u de meest recente regels gebruikt. U kunt een bijgewerkte versie van de rules downloaden va www.snort.org/dl/signatures. Pak vervolgens het bestand snortrules.tar.gz uit in de directory /etc/snort/rules en u hebt de meest recente versie van alle regels geïnstalleerd.

Als alle Snort-componenten op hun juiste plaats geïnstalleerd zijn, moet u bepalen hoe u met Snort aan het werk wilt. Als u Snort als IDS wilt gebruiken, is het gebruikelijk om de meldingen weg te schrijven in een Snort database. Voor dit doel wordt meestal gebruikgemaakt van MySQL, maar het is ook mogelijk om andere toepassingen in te zetten

zoals bijvoorbeeld Postgres, Oracle of zelfs Microsoft SQL. Om een database aan te maken die door Snort gebruikt kan worden, gebruikt u de scripts die standaard met de distributie van de Snort sourcefiles worden meegeleverd; u vindt ze in de directory contrib en ze hebben een naam als create_mysql. Aangezien het installeren van een SQL database verder een wetenschap op zich is, wordt hier in dit boek niet verder op in gegaan: bij de Snort sourcefiles wordt goede documentatie geleverd van deze procedure.

Nadat u de basisinstallatie hebt afgerond, kunt u Snort in gebruik nemen. Hieronder wordt besproken hoe u snort kunt inzetten als sniffer, packet logger of IDS.

Snort als Sniffer

Om Snort als sniffer aan te roepen, volstaat het gebruik te maken van de opdracht **snort –dvi eth0**. Met de parameter –d geeft u aan dat Snort pakketjes moet decoderen, de optie –v vertelt dat dat Verbose (uitgebreid) moet gebeuren en de optie –i eth0 tot slot vertelt dat er geluisterd moet worden naar netwerk interface eth0. Vervolgens wordt u bedolven onder een lading informatie: hier is de nodige kennis van TCP/IP nodig om er iets zinnigs over te kunnen zeggen.

***snortdvi.tif Als u Snort gebruikt als packet sniffer, wordt er heel veel informatie op het scherm van uw computer getoond.

Houdt overigens wel rekening met de beperking van sniffers als u aangesloten bent op een switch: u zult standaard alleen maar verkeer zien dat door uzelf verstuurd en ontvangen wordt.

Als u Snort inzet als sniffer, is het mogelijk gebruik te maken van de zelfde filter opties (primitives in jargon) als waar **tcpdump** gebruik van maakt. Gebruik bijvoorbeeld **snort –dv host 192.168.0.1** om alleen verkeer van en naar host 192.168.0.1 af te tappen, of gebruik **snort –dv not port 80** om alle verkeer af te tappen, behalve verkeer van en naar poort 80.

Snort als pakcet logger

Als variant op het thema Snort als sniffer, kunt u Snort ook inzetten als packet logger. Hierbij worden alle pakketjes die door Snort gesnift zijn keurig opgeslagen in een directory die u van te voren voor dit doel hebt aangemaakt. Het leuke is dat in deze directory voor elke host in het gesnifte netwerk een subdirectory gemaakt wordt en in deze subdirectory worden vervolgens de pakketjes opgeslagen. Zo kunt u relatief eenvoudig bekijken welk verkeer door welke host gegenereerd is. Om snort bijvoorbeeld te laten loggen naar de directory /var/log/snort, geeft u de opdracht **snort –d –l /var/log/snort –h 192.168.0.0/24**. De optie –d zorgt ervoor dat de pakketjes gedecodeerd worden, met –l geeft u aan dat er gelogd moet worden en met –h geeft u vervolgens het adres van het netwerk waarvoor pakketjes gelogd moeten worden.

Snort als IDS

De werkelijke reden waarom beheerders gebruik maken van Snort, is om het in te zetten als Intrusion Detection System. Om Snort als IDS te starten, is echter meer nodig dan alleen maar een programmabestand opstarten. Om te beginnen moet u in het configuratiebestand snort.conf aangeven wat er precies moet gebeuren. In snort.conf regelt u vier zaken:

* **Definitie van variabelen**. Om het werken met de Snort configuratie te vereenvoudigen (althans, dat vinden sommigen) worden veel waarden die gebruikt moeten worden opgeslagen in een variabele. Zo zijn er bijvoorbeeld variabelen waarmee u aangeeft op welk netwerk Snort actief is, welke DNS-servers gebruikt moeten worden en nog veel meer. Voor een eenvoudige configuratie komt u al een heel eind als u de variabelen

HOME_NET en DNS_SERVERS instelt, als u Snort werkelijk in witl zetten in uw netwerk, is het aan te raden het configuratiebestand eens zorgvuldig te bekijken en elke variabele de benodigde waarde te geven.

* **Configuratie van de preprocessors**. De preprocessor is een belangrijk onderdeel in een Snort-configuratie. Om de preprocessor te kunnen begrijpen, moet u weten hoe Snort als IDS te werk gaat. Om te beginnen moeten pakketjes op het netwerk worden binnengehaald. Hiervoor wordt gebruikgemaakt van de Kibpcap library; deze library maakt capturen van pakketjes mogelijk. De Snort decoder zorgt er vervolgens voor dat deze pakketjes leesbaar worden. Het resultaat hiervan is een enorme hoop gegevens die geanalyseerd moeten worden. Om deze analyse te vereenvoudigen, wordt er door de pre-processors die als plug-ins geladen kunnen worden een schifting gemaakt. Uiteindelijk zorgt dit dat er voor de detection engine die moet bepalen of pakketjes kwalijke code bevatten of niet, minder werk over blijft en dat is goed. De werking van deze detection engine kan overigens vanuit snort.conf ook weer verder aangepast worden met behulp van plug-ins. Uiteindelijk levert dit proces uitvoer op en ook deze uitvoer kan met behulp van plug-ins bekeken worden. Al met al bestaat de taak van de beheerder er dus uit om de juiste plug-ins te configureren zodat Snort zo optimaal mogelijk zijn werk doet.
* **Configuratie van de output plugins**. Als Snort zijn gegevens wegschrijft naar tekstbestanden, is het niet nodig te werken met output plugins. Als u echter gebruikmaakt van een aparte database waarnaar uitvoer weggeschreven wordt, moet u opgeven hoe dit moet gebeuren. Hiervoor maakt u gebruik van een output plugin. Zo kunt u bijvoorbeeld de volgende regel opnemen om de gegevens op de juiste wijze naar de uitvoer database te sturen:

output database: log, mysql, user=root dbname=snort host=localhost

Tip! Als u dan toch gegevens wegschrijft naar een SQL daabase, kan het zeer zinnig zijn om die database op een remote host weg te schrijven. Mocht uw Snort IDS dan namelijk ooit gekraakt worden, dan hebt u de gegevens die gelogd zijn in elk geval ergens anders veilig weggeschreven.

* **Configuratie van regels**. Het is mogelijk om in snort.conf direct zelf regels te definiëren, maar dit is niet aan te raden. De Snort regels worden namelijk op de Snort website elke 30 minuten aangepast. Om die reden doet u er goed aan de regels ergens als aparte bestanden op te nemen en hiernaar te verwijzen door middel van include statements in snort.conf. Dat is dan ook precies wat er standaard al gebeurt. Om bijvoorbeeld te verwijzen naar het bestand waarin de bad-traffic rules gedefinieerd zijn, zorgt u ervoor dat in snort.conf de regel include /etc/snort/rules/bad-traffic.rules is opgenomen. Doe dit voor elk rules bestand dat u hebt opgeslagen in de directory /etc/snort/rules.
Bij de configuratie van Snort rules is het zaak om bijzonder goed na te denken welk doel u precies wilt bereiken. Hoe meer rules namelijk bekeken moeten worden, hoe zwaarder Snort het krijgt en hoe groter de kans wordt dat instellingen in bepaalde regels niet verwerkt kunnen worden bij top belasting. Bekijk de categorieën regels dus een keer serieus en zorg ervoor dat alle regels die u niet nodig hebt ook gewoon verwijderd worden. Bekijk ook de inhoud van de bestanden met rules, sommige regels staan namelijk standaard uit; u zult ze dus eerst aan moeten zetten als u er gebruik van wilt maken.

Snort als IDS starten
Nadat u de configuratie in het bestand snort.conf hebt aangepast, wordt het zaak om het IDS te starten. Het is echter verstandig om eerst eens te testen of uw huidige instellingen werkbaar zijn: gebruik hiervoor de opdracht **snort –T –c /etc/snort/snort.conf**. Snort bekijkt nu alle

regels in het configuratiebestand en zal netjes een fout genereren als er iets niet in orde is. Is alles in orde? Dan kunt u nu Snort starten met behulp van de opdracht **snort –Dd–c /etc/snort/snort.conf**. Houdt er rekening mee dat Snort geen foutmelding zal geven als er iets niet in orde is tijdens het opstarten; u kunt echter in /var/log/messages uitstekend zien of het opstarten van Snort al dan niet goed gegaan is.

***logmess.tif In /var/log/messages ziet u of het opstarten van Snort al dan niet goed gegaan is.

In deze opdracht worden de volgende parameters gebruikt:
-D Start snort als Daemon. Dit zorgt ervoor dat er geen meldingen weggeschreven worden naar het scherm, maar dat alle meldingen netjes gelogd worden in het gespecificeerde log-mechanisme.
-d Packet decode; zorgt ervoor dat pakketjes vertaald worden
-c Maakt duidelijk welk configuratiebestand gebruikt moet worden om instellingen in te lezen.

Nu u Snort draaiend hebt, wordt het natuurlijk zaak om het resultaat van het werk van Snort te bekijken. Snort schrijft standaard al zijn uitvoer weg naar de directory /var/log/snort. In deze directory wordt voor elke host in het gespecificeerde netwerk een aparte subdirectory aangemaakt, handig want zo is het eenvoudig om pakketjes die van een bepaalde host afkomstig zijn te analyseren. In tegenstelling tot snort in packet loggin mode, worden nu niet alle pakketjes in deze directories weggeschreven, maar alleen pakketjes waarmee iets aan de hand is zoals gedefinieerd is in de snort rules. Alle alerts worden standaard gelogd naar het bestand /var/log/snort/alert; headers van port scans worden gelogd naar het bestand /var/log/portscan.log.

***snortscan.tif Na een portscan vindt u in het bestand /var/log/snort/alert ongeveer de volgende inhoud.

Tip! Wellicht denkt u dat de configuratie van Snort erg ingewikkeld is. Wel, eigenlijk hebt u dan nog gelijk ook. Een zinnige implementatie van Snort in een netwerk is specialistenwerk waar heel veel bij komt kijken, de informatie die u in dit hoofdstuk gelezen hebt is niet meer dan een eerste aanzetje om dit werk te starten. Toch kan het ook heel envoudig zijn: als u gebruik wilt maken van alle standaardinstellingen, kunt u gewoon direct aan het werk. Geef het commando **snort –Dd –c /etc/snort/snort.conf** en kijk wat er allemaal gebeurt op uw netwerk. Houd er echter wel rekening mee dat Snort op deze manier niet bepaald geoptimaliseerd voor uw netwerk aan het werk is.

Als u dan eenmaal een werkende snort configuratie hebt, wordt het zaak om het resultaat ook in de gaten te houden. Als u ooit de inhoud van het snort logbestand bekeken hebt, weet u waarschijnlijk al dat dit niet echt iets is dat vanzelfsprekend is, zeker niet wanneer u probeert om met klassieke tools als less of tail –f bij te houden wat er allemaal gebeurt. U kunt voor dit doel gebruikmaken van een geavanceerde tool zoals Swatch; een generieke tool om logbestanden mee in de gaten te houden, maar de installatie en het onderhoud hiervan is niet echt super gebruikersvriendelijk. Een ander alternatief is gebruik te maken van SnortSnarf. Dit Perl-script houdt de Snort alertfile in de gaten en stuurt uitvoer vanuit dit bestand door naar HTML-formaat. In dit HTML-formaat kunt u vervolgens vanuit een browser de gegevens bekijken. U kunt SnortSnarf downloaden van

http://www.silicondefense.com/software/snortsnarf. Daarnaast zijn er nog veel andere tools beschikbaar. Hieronder vindt u een overzicht van een aantal van deze tools:

* Snort Alert Monitor: een Java-programma dat uw MySQL database in de gaten houdt op Snort alerts. De tool kan zo ingesteld worden dat er geluiden worden afgespeeld, mails worden verstuurd en wat al niet meer op het moment dat zich een kritische situatie voordoet. U kunt het downloaden van sourceforge.net/projects/snortalertmon.
* PigSentry: doet precies dat waar u een hekel aan hebt, hij houdt namelijk de Snort alert log in de gaten en verstuurt meldingen wanneer er een belangrijke nieuwe alert bij is gekomen. U kunt PigSentry downloaden van web.solv.com/tools/pigsentry.
* Snortalog Maakt rapporten op basis van events die door Snort gedetecteerd worden. Kan niet alleen met het formaat in het Snort alerts file overweg, maar is bijvoorbeeld ook in staat met syslog te communiceren. Het resultaat wordt vervolgens netjes weggeschreven naar het formaat dat u wilt: ASCII tekst, HTML, afbeelding, het is allemaal mogelijk.

Loggen naar MySQL

In het voorgaande hebt u geleert hoe u Snort kunt configureren om naar een tekstbestand te loggen. Dat is een leuk begin, maar om een echt druk netwerk in de gaten te houden is het natuurlijk geen doen. Wat u in een omvangrijk netwerk nodig hebt, is de mogelijkheid logmeldingen weg te schrijven naar een MySQL database. De belangrijkste reden hiervoor is dat heel veel tools die gebruikt worden om Snort alerts te beheren speciaal ontworpen zijn om samen te werken met zo'n MySAL database. MySQL wordt met vrijwel elke distributie meegeleverd, het is dus niet nodig om eerst de programmabestanden te downloaden en installeren, maar u kunt gewoon direct aan het werk. Een MySQL configuratie bestaat uit de server en de cliëntbestanden: verzeker u ervan dat beide componenten geïnstalleerd zijn voordat u aan het werk gaat. Zowel op Fedora als SUSE bent u in de gelegenheid beide componenten vanuit de standaard lijst met toepassingen te installeren. In de nu volgende procedure gaan we er van uit dat dat gebeurd is. U leest nu hoe u Snort gegevens weg kunt laten schrijven naar de MySQL database. We gaan er daarbij van uit dat MySQL actief is, zorg er indien nodig voor dat het vanuit uw runlevel configuratie automatisch gestart wordtvoordat u aan de onderstaande procedure begint.

1. Zoek in snort.conf de regel die begint met output database. Standaard ziet deze er uit als **output database: log, mysql, user=root password=test dbname=db host=localhost**.
2. Pas de bovenstaande regel aan zodat gebruikgemaakt wordt van de juiste instellingen. Om te beginnen wijzigt u de parameter user=root in user=snort, de standaard naam van uw Snort gebruikersaccount. Wijzig vervolgens dbname=db in dbname=snortdb. Deze database bestaat op dit moment nog niet, maar dat gaat u zo regelen. Pas ook het wachtwoord van snortuser aan in het wachtwoord dat u wilt gebruiken. De uiteindelijke regel komt er dus uit te zien als **output database: log, mysql, user=snortuser password=geheim dbname=snortdb host=localhost**.
3. Nu moet u ervoor zorgen dat de Mysql database ook klaar wordt gemaakt zodat Snort er gegevens in weg kan schrijven. Hiervoor moet u als gebruiker vanuit de MySQL client inloggen op de MySQL database en vervolgens een database aanmaken. In het onderstaande overzicht ziet u de opdrachten die hiervoor gegeven moeten worden en ook de output die daarbij door MySQL gegenereerd wordt. Voor alle duidelijkheid: we maken gebruik van het wachtwoord 'geheim', u wordt van harte aangeraden dit wachtwoord door een beter wachtwoord te vervangen:

 # /usr/bin/mysqladmin –u root password geheim

 # mysql –u root –p
Enter password:
Welcom to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 4 to server version: 3.23.58

Type 'help;' or '\h' for help. Type '\c'to clear the buffer
mysql> create database snortdb;
Query OK, 1 row affected (0.00 sec)

mysql> grant INSERT,SELECT on snortdb.* to snort@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> SET PASSWORD FOR snort@localhost=PASSWORD('geheim');
Query OK, 0 rows affected (0.00 sec)

mysql> grant CREATE,INSERT,SELECT,DELETE,UPDATE on snortdb.* to snort@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
mysql> exit
Bye

U hebt nu een database aangemaakt en op deze database heeft gebruiker snortuser voldoende permissies.
4. Nu moet u ervoor zorgen dat de Snort database op de juiste wijze wordt aangemaakt. Maak hiervoor gebruik van het script create_mysql dat met de Snort sourcefiles wordt meegeleverd. Als u dit script niet op uw computer hebt staan, kunt u het downloaden van cvs.sourceforge.net. Het is aan te raden even met Google te zoeken String "snort create_mysql download" en van daar uit het script te downloaden naar uw computer.
Om nu de Snort database in MySQL aan te maken, geeft u de opdracht **mysql –u root –p < create_mysql snortdb**. Vergeet niet een volledige padaanduiding te geven naar de lokatie waar het script create_mysql gevonden kan worden.
5. Gefeliciteerd! Als u de voorgaande procedure gevolgd hebt, bestaat de database snortdb nu in MySQL. U bent klaar om hem te gaan gebruiken. Indien gewenst kunt u controleren dat de database bestaat: geef dan achter elkaar de volgende opdrachten:

 # mysql –u root –p
mysql > SHOW DATABASES;
mysql > use snortdb
mysql > SHOW TABLES;
exit;

In de onderstaande afbeelding ziet u wat u dan als het goed is te zien krijgt.

***snortdb.tif Om te controleren of uw snortdb nu inderdaad bestaat, kunt u in MySQL een aantal opdrachten geven.

Werken met Snort output

Als u de voorgaande procedure met succes voltooid hebt, kunt u nu de uitvoer van Snort zoals weggeschreven in de MySQL database bekijken. Ook is het mogelijk op andere manieren met de snort output om te gaan. De werkwijze die u wilt hanteren, wordt in Snort geïmplementeerd met behulp van een output module. De wijze waarop deze modules aangeroepen worden, is afhankelijk van de manier waarop u Snort gestart hebt. Om in daemon modus te werken met output modules, worden hiervoor vaak opties opgenomen in het configuratiebestand snort.conf. Hieronder volgt een korte beschrijving van een aantal output modules die gebruikt kunnen worden:

* **alert_fast**. Deze module zorgt ervoor dat naar aanleiding van elk alert een korte regell wordt weggeschreven naar het uitvoerbestand waarmee u werkt. U roept deze regel aan door in snort.conf de regel **output alert_fast: snort.log** op te nemen
* **alert_full**. Waar **alert_fast** ervoor zorgt dat snel alleen de belangrijkste gegevens over een alert gelogd worden, zorgt alert_full ervoor dat ook de gedecodeerde packet headers gelogd worden. Dit kost veel werk en zal er voor zorgen dat pakketjes gedropt worden omdat Snort geen tijd heeft ze weg te schrijven. Om deze uitgebreide uitvoer weg te schrijven naar het bestand alert.full, neemt u de regel **output alert_full: alert.full** op in uw snort configuratiebestand.
* **alert_syslog**. Met behulp van deze module zorgt u dat alerts doorgestuurd worden naar syslog. De syntaxis is eenvoudig: **alert_syslog: <facility> <priority>**. U kunt gebruikmaken van de facilities LOG_AUTH, LOG_AUTHPRIV, LOG_DAEMON, LOG_USER of LOG_LOCAL0 tot en met LOG_LOCAL7. Alle standaard syslog priorities kunnen gebruikt worden, alleen moet u de naam van de priority wel vooraf laten gaan door de aanduiding LOG_, bijvoorbeeld LOG_ERR
* **alert_csv**. Wilt u later nog iets anders met de uitvoer kunnen doen? Dan kunt u hem doorsturen naar een CSV-bestand. In het CSV-bestand kunnen verschillende velden gedefinieerd worden, raadpleeg de man pagina voor meer informatie hierover. Om ervoor te zorgen dat alle informatie gewoon in standaard formaat gelogd wordt naar een bepaald bestand, neemt u de regel **output alert_CSV: <bestandsnaam> default** op in snort.conf. Specificeer in plaats van <bestandsnaam> wel de naam van het bestand waarnaar u de meldingen weg wilt schrijven.
* **log_tcpdump**. Wilt u vooral snel veel gegevens weg kunnen schrijven? Dan kunt u log_tcpdump gebruiken om de gegevens weg te schrijven naar een binair TCP-dump bestand. Gebruik hiervoor de regel **output log_tcpdump: snort.dump**.
* **database**. We hebben het er eerder al even over gehad: u kunt de Snort-data ook doorsturen naar een database. Een regel om dit te regelen hebt u in de voorgaande paragraaf al aangetroffen: gebruik bijvoorbeeld **output database: log, myssql, user=snort password=geheim dbname=snortdb host=localhost**.

7.6.4.2 Tripwire

U moet Snort inzetten om ervoor te zorgen dat ongenode gasten niet ongezien uw netwerk kunnen binnendringen. Ondanks alles wat u doet om dit te voorkomen, kan het natuurlijk een keer voorkomen dat iemand toch toegang krijgt. In dat geval wilt u beschikken over een programma dat u kan helpen te achterhalen waar en hoe veel schade er is aangericht. Tripwire is zo'n tool. Tripwire wordt ingezet om de integriteit van bestanden op uw systeem in de gaten te houden. Hiervoor wordt voor elk bestand een fingerprint gegenereerd, deze fingerprints worden vervolgens opgeslagen in een database. Wordt een bestand veranderd? Dan verandert automatisch ook de fingerprint die bij het betreffende bestand hoort. Met behulp van Tripwire is het heel eenvoudig fingerprints te vergelijken tussen twee runs van het programma en ziet u in een handomdraai welke bestanden er zijn gewijzigd. Bij het werken met tripwire moet een aantal zaken geregeld worden:

1. Genereer een sitekey die gebruikt kan worden voor de encryptie die door Tripwire wordt toegepast.
2. Definieer een standaardconfiguratie in het bestand /etc/tripwire/tw.cfg
3. Maak een policy file waarin wordt bijgehouden wat er allemaal door Tripwire moet worden bijgehouden.
4. Initialiseer de Tripwire database
5. Gebruik een Tripwire integrity check om te bepalen welke bestanden er gewijzigd zijn ten opzichte van de policy
6. Als er wijzigingen zijn die u wilt toestaan, accepteer deze dan.

Houd er rekening mee dat Tripwire niet automatisch mee wordt geïnstalleerd bij elke distributie. U zult het dus waarschijnlijk eerst moeten installeren. Maak hiervan gebruik van de RPM's of de tarball die u vindt op www.tripwire.org. De installatieprocedure kan uitgevoerd worden door het installatiescript install.sh uit te voeren. Nadat de software op uw systeem geïnstalleerd is, voert u de volgende stappen uit om Tripwire in gebruik te nemen:

Stap 1. Genereer een sitekey
Tijdens de installatie worden al een sitekey en een local key gegenereerd. Deze worden weggeschreven in de bestanden site.key en hostnaam-local.key. U vindt deze bestanden in de directory /etc/tripwire. Ook vindt u hier het bestand twcfg.txt waarin een verwijzing voorkomt naar de exacte locatie van die twee bestanden. Met behulp van de opdracht **twadmin** kunt u naderhand zonder problemen een nieuwe sitekey aanmaken. Gebruik hiervoor de opdracht **twadmin –m G –S site.key –Q *wachtwoord***. Op soortgelijke wijze is het mogelijk een local key aan te maken: gebruik hiervoor indien nodig de opdracht **twadmin –m G –L `hostname` -local.key –P *wachtwoord***.

Stap 2. Genereer een binair configuratiebestand.
in de voorgaande stap hebt u kennisgemaakt met het bestand dat als invoerbestand gebruikt wordt om de tripwire configuratie op poten te zetten: twcfg.txt. Nadat u wijzigingen hebt aangebracht in dit configuratiebestand, moet u deze wijzigingen compileren zodat ze opgenomen worden in het binaire configuratiebestand /etc/tripwire/tw.cfg. U moet deze compileeractie uitvoeren na elke wijziging die u hebt aangebracht. Als u bijvoorbeeld de variabele SYSLOGREPORTING =false wilt wijzigen in SYSLOGREPORTING =true zodat wijzigingen in gemonitorde bestanden weggeschreven worden naar syslog, moet u vervolgens de gewijzigde parameter compileren in het binaire configuratiebestand. Hiervoor gebruikt u de opdracht **twadmin –m F –S site.key –Q *wachtwoord* twcfg.txt**.

Stap 3. Maak een policyfile waarin u aangeeft welke bestanden u in de gaten wilt houden.
Het slechte nieuws over Tripwire policy files is dat ze complex zijn. In een policy file wordt gewerkt met variabelen die aangeven wat er precies moet gebeueren, vervolgens geeft u door middel van property masks aan welke eigenschappen van de bestanden die gemonitord worden in de gaten moeten houden. Als property wordt vaak gebruikgemaakt van de aanduiding pinugs, wat staat voor permissions, inode, number of hard links, user, group, size. Complexe policy files kunnen even ondoorgrondelijk zijn als het sendmail configuratiebestand, soms echter hebt u aan een eenvoudig policybestand ook genoeg. Een heel eenvoudig policybestand kan bijvoorbeeld de inhoud **/etc -> pinugs ;** hebben (vergeet de puntkomma niet): dit zorgt ervoor dat alarm geslagen wordt als een van de bestanden in de directory /etc wijzigt. In veel gevallen is dat ook meer dan genoeg en moet u zeker niet te enthousiast aan het werk gaan met het definiëren van policies. Als immers een beheerder te vaak geïnformeerd wordt over gewijzigde bestanden, zorgt dat er uiteindelijk voor dat hij het

tripwire verkeer als hinderlijk af zal doen. Nadat u een policy bestand hebt aangemaakt met ASCII-tekst als inhoud, moet dit bestand gegenereerd worden. Gebruik hiervoor de opdracht **twadmin –m P tw.pol**.

Stap 4. Initialiseer de Tripwire database
Nu u alle standaardwaarden hebt ingevuld, wordt het tijd om de Tripwire database te genereren. Gebruik hiervoor de opdracht **tripwire –init** en voer vervolgens de passphrase in die u gedefinieerd hebt voor de sitekey

Stap 5. Gebruik een Tripwire integrity check om te bepalen welke bestanden er gewijzigd zijn ten opzichte van de policy
In het voorgaande hebt u een database aangemaakt waarbij de huidige status van de bestanden in de directory /etc/ gemonitord wordt. Nu is het natuurlijk zaak om te kijken of het systeem inderdaad ook werkt. Zorg ervoor dat een bestand in /etc/gewijzigd wordt, bijvoorbeeld door een gebruiker aan te maken. Gebruik vervolgens de opdracht **tripwire –m c** om U zult nu zien dat Tripwire aangeeft dat verschillende bestanden in de directory /etc gewijzigd zijn. Ook wordt er een rapport aangemaakt; dit rapport wordt opgeslagen in de directory /var/lib/tripwire/report

Stap 6. Als er wijzigingen zijn die u wilt toestaan, accepteer deze dan.
Als beheerder van een tripwire systeem, is het zaak om ook vooral periodiek wijzigingen te accepteren, anders raakt u het overzicht al snel kwijt. Concludeert u na uw periodieke run van de Tripwire controle dat er wijzigingen zijn die u best in orde vindt? Gebruik dan de opdracht **tripwire –m u –V /bin/vi –r /var/lib/tripwire/report/*uwrapportnaam***. Deze opdracht zorgt ervoor dat het rapport getoond wordt in de editor vi. Sluit deze editor af met de opdracht **:q**, op basis van de wijzigingen in het rapport wordt er nu een nieuwe Tripwire database aangemaakt.

Oefening 7.6
Deze oefening bestaat uit verschillende stappen. Voor uitvoering van sommige stappen hebt u twee comptuer nodig. Deze computers worden aangeduid als server en werkstation. Welke computer u als server gebruikt en welke als werkstation, maakt voor de uitvoering van deze oefening niet uit.

1. Installeer een sniffer op uw systeem en zorg ervoor dat u in staat bent om alle netwerkverkeer dat op het netwerk verstuurd wordt te sniffen. Open vanaf uw eigen werkstation een http-sessie naar een willekeurige website. Kijk wat u van deze http-sessie in de sniffer terug kunt vinden.
2. Installeer Snort op de server. Zorg ervoor dat de gegevens gewoon gelogd worden naar tekstbestanden. Als u aan het einde van deze oefening tijd over hebt, zorgt u ervoor dat de gegevens eveneens gelogd worden naar een MySQL database.
3. Start nmap op het werkstation en voer een volledige poortscan uit op de server. Bekijk de meldingen die op de server gedaan worden over deze poortscan.
4. Installeer tripwire op beide computers. Zorg ervoor dat gescand wordt op wijzigingen in de directory /etc. Test of dit werkt.
5. Zoek op internet naar de meest gevaarlijke bedreiging voor het Linux platform op dit moment. Zorg er eventueel door het uitvoeren van een patch voor dat deze bedreiging op uw computer geen invloed heeft.

Samenvatting

In dit laatste hoofdstuk hebt u kennis kunnen maken met verschillende aspecten van beveiliging die op een Linux-systeem ingezet kunnen worden. Zoals u hebt gemerkt, is beveiliging een veelomvattend onderwerp; dit hoofdstuk vormt haast een boekje op zich. In dit hoofdstuk hebt u geleerd hoe u ervoor kunt zorgen dat toegang tot bepaalde services beperkt wordt door gebruik te maken van een firewall die op netfilter gebaseerd is, of van de 'gouwe ouwe' tcpd-funcitonaliteit. Ook hebt u geleerd hoe u veilig met uw server kunt communiceren. Voor deze beveiliging zijn verschillende technieken aan de orde gekomen, waarvan SSH de meest belangrijke is. Het laatste belangrijke onderdeel van dit hoofdstuk ging over pro-actief beheer. U hebt hier geleerd hoe u ervoor zorgt dat uw netwerk veilig blijft en hoe u inbraakpogingen op uw netwerk zelf kunt detecteren.

Oefenvragen

1. Met welke opdracht maakt u een public/private-key paar aan voor gebruik in SSH?
2. Wat is de belangrijkste taak voor de beheerder van een Snort-systeem?
3. Wat is de meest handige wijze om ervoor te zorgen dat gebruikers in kunnen loggen op een vsftp-server?
4. Waarom valt het gebruik van xinetd boven inetd te prefereren?
5. Hoe heet het policybestand waarin aangegeven wordt wat Tripwire in de gaten moet houden?
6. Hoe lost u het voornaamste probleem op dat zich voordoet wanneer u verkeer wilt sniffen op een geswitcht netwerk?
7. Hoe stelt u een standaard policy in op deny voor de INPUT chain?
8. Welke opdracht geeft u om een volledige UDP-scan uit te voeren waarbij niet eerst naar de host in kwestie gepingd wordt maar wel wordt achterhaald welk besturingssysteem in gebruik is?
9. Wat vindt u persoonlijk de meest handige manier om op de hoogte te blijven van nieuwe bedreigingen op internet?
10. Welke opdracht gebruikt u om ervoor te zorgen dat u nooit meer handmatig een passphrase hoeft in te voeren bij het tot stand brengen van een SSH-verbinding op basis van DSA-keys?