Het opzetten van een DNS-Server met behulp van bind-4.9.7

Patrick van Zweden Mei 1999, Versie 1.1

Het opzetten van een DNS-server onder linux met behulp van bind-4.9.7.

Inhoudsopgave

1 Disclaimer

Copyright (c) 1999 Patrick van Zweden *pvzweden@caiw.nl* <mailto:pvzweden@caiw.nl>

Dit document valt onder de termen zoals besproken in versie 2 (of latere versies) van de GNU General Public License zoals gepubliceerd door de Free Software Foundation.

Dit document is gedistribueert in de hoop iets toe te voegen aan de al bestaande documentatie, zei het zonder ENIGE AANSPRAKELIJKHEID. Zie de GNU General Public License voor meer details.

De auteur van dit document moedigt het wijd verspreiden van dit document van harte aan, zei het dat dit gebeurt zonder dat er wijzigingen worden aangebracht in deze disclaimer en zonder dat het document incompleet wordt verspreid. Er is geen expliciete toestemming van de auteur nodig om dit document te vermenigvuldigen op enigerlei manier, zei het digitaal danwel via een ander medium.

De auteur stelt het zeer op prijs dat revisies en aanvullingen aan hem bekent worden gemaakt. Hij kan er dan voor zorgen dat deze worden opgenomen in het hele document en dit weer beschikbaar stellen aan de linux gemeenschap.

Als u dit document op enigerlei manier commercieel wilt verspreiden of u heeft er plannen voor dan zou de auteur graag hiervan op de hoogte worden gebracht. Hij kan er dan voor zorgen dat u voorzien wordt van de juiste en complete versie van het document.

Als u dit document op enigerlei manier commercieel wilt verspreiden of in boekvorm wilt uitbrengen dan zijn donaties of een gedrukte versie van dit document van harte welkom.

Mocht u nog vragen en/of opmerkingen hebben dan kunt u altijd contact opnemen met de auteur op *pvzweden@caiw.nl* <mailto:pvzweden@caiw.nl>.

2 Inleiding - Verklaring van de stijlen

Hier staan in het kort de stijlen beschreven die je tegen kunt komen in deze handleiding.

In deze handleiding wordt gebruik gemaakt van bepaalde stijlen om speciale dingen aan te geven. Hieronder vind je een overzicht van de stylen die worden gebruikt.

vet - Wordt gebruikt om waarschuwingen, weer te geven.

italics - Wordt gebruikt om de nadruk op bepaalde woorden te leggen.

`Typewriter` - Wordt gebruikt een schermafdruk weer te geven.

3 Inleiding tot DNS

Allereerst beginnen we met een kleine inleiding over hoe DNS nu precies in zijn werk gaat en hoe het ontstaan is.

3.1 Historie

Al in een vroeg stadium van het bestaan van internet realiseerde men zich dat de nummers (ip-adressen) die werden gebruikt om de computers te adresseren wel makkelijk in het gebruik waren voor computers onderling, maar dat het niet het ideale systeem was voor mensen. Mensen geven namelijk de voorkeur aan het gebruik van namen als ze het ergens over hebben.

Je zal niet gauw in een gesprek tegenkomen dat men tegen je zegt : "Ik had gisteren nog contact met 194.178.9.133 en die had een aantal leuke berichten voor me". Bovendien zijn de adressen nogal moeilijk om te onthouden. De naam mailserv.caiw.nl is namelijk veel makkelijker te onthouden dan het nummer 194.178.9.133.

Daarom bedacht men het domein namensystem, of kortweg DNS. DNS zorgt er voor dat een naam naar een ip-adres (nummer) vertaald kan worden. Maar bij het invoeren van dit systeem liep men tegen een aantal problemen aan, namelijk dat er gegarandeerd moest worden dat er geen twee computers op het netwerk dezelfde naam mochten hebben.

Ook moest er goed bedacht worden hoe het vertalen in z'n werk zou gaan, want de computer moet toch ergens in een lijstje kunnen zoeken.

3.2 Structuur van het domein-namensysteem

Het domein-namen systeem is een systeem om namen toe te kennen in domeinen. Hierbij hebben verschillende groepen de verantwoordelijkheid voor de verschillende onderdelen van de naam. Ieder niveau in het systeem heet een domein. De domeinen worden van elkaar gescheiden door een punt.
Bijvoorbeeld :

mailserv.caiw.nl
news.kabelfoon.nl
newsserv.caiw.nl

Het aantal domeinen in een naam kan verschillen maar het zullen er zelden meer dan vijf zijn. Als je een naam van links naar rechts bekijkt is elk opvolgend domein groter als het voorgaande. In de naam mailserv.caiw.nl is mailserv de daadwerkelijke naam van een computer. Deze naam is bedacht door de beheerders van het caiw-domein (Cai Westland). De afdeling caiw maakt weer deel uit van het nl-domein. Dit domein beheert alle namen in het nl gebied. Dus eigenlijk kun je het zien als dat het nl-domein alle computers in Nederland bevat, het caiw domein alle computers binnen het caiw-gebied en dat mailserv een daadwerkelijke machine is. Iedere groep kan binnen zijn gebied namen aanmaken, dus de beheerders van het caiw domein kunnen er heel makkelijk een aantal namen bijmaken zonder dat daar toestemming van bijvoorbeeld de beheerder van het nl-domein voor nodig is. Willen ze een nieuwe naam hebben in het nl-domein, bijvoorbeeld kabelfoon.nl dan zullen ze daar toestemming voor moeten vragen. Als ze die krijgen dan wordt door de beheerder van het nl domein een nieuwe naam aangemaakt en mogen ze zonder daarvoor iemand om toestemming te vragen namen aanmaken in het kabelfoon domein.

3.3 Het opzoeken van domeinnamen

Nu we weten hoe de namen in elkaar zitten en hoe ze worden toegekend zullen we eens kijken hoe dit systeem nu werkt. Zodra je een internet adres in een applicatie intikt, bijvoorbeeld Netscape, zal deze applicatie de hulp in gaan roepen van een DNS-server. Hij begint dan met het meest rechtse stuk en zal de naam verder naar links afwerken. Eerst zal hij de lokale DNS-server verzoeken om het adres op te zoeken, er zijn dan drie mogelijkheden mogelijk :

- De lokale server kent het adres omdat het zich in het gedeelte bevint dat hij beheert.
- De lokale Server kent het adres omdat iemand anders het kortgeleden al een keer heeft opgevraagd. De DNS-server houd de informatie namelijk een tijdje vast voor het geval er meer verzoeken komen voor hetzelfde adres.
- De lokale Server weet het adres niet maar weet wel hoe hij aan het adres kan komen.

Hoe zoekt de DNS-server nu een adres op dat hij niet kent ? De software op de server kan contact zoeken met een zogenoemde root-server. Dit is een server die de namen van de DNS servers voor het hoogste deel van de naam kent, dus bijvoorbeeld de DNS-server voor het .nl-domein. Vervolgens gaat hij aan die server het adres vragen van de DNS-server die het gebied wat als één na laatste in de naam staat, bijvoorbeeld kabelfoon.

Als hij daar het adres van heeft gekregen dan stuurt hij een verzoek naar die server voor het adres wat weer een positie naar links staat op, bijvoorbeeld kgg. Als hij hier het adres van heeft gekregen dan is zijn zoekactie voltooid en kan hij het adres melden voor de server die luistert naar de naam kgg.kabelfoon.nl. Hoe nu zo'n server wordt opgezet waarmee je dus namen op kunt laten zoeken en/of beheren wordt in de volgende hoofdstukken besproken.

4 Het begin

In dit hoofdstuk bespreken we welke software we nodig hebben en waar we die software kunnen halen. Ook bespreken we hoe we ervoor zorgen dat alles correct wordt geïnstalleerd.

4.1 De benodigde software

Allereerst hebben we natuurlijk een draaiend linux systeem nodig, de linux versie die in deze handleiding wordt gebruikt is de distributie Slackware versie 3.5 met kernel 2.0.36. Op zich zijn er niet heel erg veel verschillen tussen andere distributies. Een van de dingen die wel anders is met bijvoorbeeld een distributie als Red Hat is de plek van de opstart bestanden. Als we het hebben over de directory `/etc/rc.d` dan moet je daar de directory invullen waar jouw configuratie bestanden staan.

Het volgende wat we nodig hebben is de software voor de DNS server. De software die in deze handleiding daarvoor wordt gebruikt is het pakket Bind. Bind is te verkrijgen bij de publieke ftp-server van ISC. Het adres van deze server is ftp.isc.org, de laatste sources van de software bevinden zich dan in de directory `/isc/bind/src`.

Er zijn op dit moment twee versies van Bind in omloop, namelijk 4.9 (meest recente 4.9 patchlevel 7) en 8.1 (meest recente 8.1 patchlevel 2). Deze handleiding bespreekt versie 4.9 patchlevel 7. De versie die op dit moment verder wordt ontwikkeld is versie 8. Versie 4 wordt alleen geupdate als er beveiligingslekken in zitten.

4.2 Het uitpakken

Nadat we het benodigde archief hebben opgehaald van de ftp-server dienen we het uit te pakken. Dit doen we door middel van het commando tar aan te roepen. Allereerst maken we een directory aan met de naam : `bind-4.9.7`, in deze directory zetten we straks de sources neer. Bind maakt bij het uitpakken, in tegenstelling tot veel *nix programmatuur, geen directory aan. We doen dus het volgende :

pinky:˜$ mkdir bind-4.9.7
pinky:˜$ tar -xvzf bind-4.9.7-REL.tar.gz -C bind-4.9.7

We hebben nu de source van bind uitgepakt in de directory `bind-4.9.7`, laten we eens kijken wat voor structuur we hebben gecreëerd. We hebben als het goed is nu de volgende directory structuur staan in de directory `bind-4.9.7`:

```
drwxr-xr-x   9 pvzweden users       1024 Jun 24  1997 BSD
-r--r--r--   1 pvzweden users     116408 May 11  1998 CHANGES
-r--r--r--   1 pvzweden users       4190 Jun 24  1997 INSTALL
-r--r--r--   1 pvzweden users      26150 May 11  1998 Makefile
-r--r--r--   1 pvzweden users      21567 May 21  1996 OPTIONS
-r--r--r--   1 pvzweden users      11631 Apr  7  1998 README
-r--r--r--   1 pvzweden users      10156 Jun 19  1995 TODO
drwxr-xr-x   2 pvzweden users       1024 Dec 15  1994 bin
drwxr-xr-x   5 pvzweden users       1024 Dec 23  1995 compat
drwxr-xr-x   3 pvzweden users       1024 Jun  1  1997 conf
drwxr-xr-x  28 pvzweden users       1024 Dec  2  1996 contrib
drwxr-xr-x   6 pvzweden users       1024 Jun 24  1997 doc
drwxr-xr-x   3 pvzweden users       1024 Jun  1  1997 include
drwxr-xr-x   2 pvzweden users       1024 Apr  7  1998 man
drwxr-xr-x   2 pvzweden users       1024 May 11  1998 named
drwxr-xr-x   2 pvzweden users       1024 May 11  1998 res
drwxr-xr-x   7 pvzweden users       1024 Sep 28  1996 shres
drwxr-xr-x   3 pvzweden users       1024 Apr  7  1998 tools
```

```
INSTALL   -    Dit is een bestand met informatie over het installeren van bind
README    -    Wat tips voor het compileren en installeren van bind en wat
               notities die meekwamen met de release van bind
OPTIONS   -    De opties die aan- en uitgezet kunnen worden
RUNSON    -    Een lijst compilers/platforms waar bind op draait
TODO      -    Wat er nog moet gebeuren, als je tijd en zin hebt om mee te
               ontwikkelen aan bind dan kan je hier in vinden wat nog nodig
               is.
BSD/*     -    Hoe Bind geïntegreerd dient te worden in BSD
contrib/* -    Software die door ontwikkelaars toegevoegd is aan de bind
               distributie. Dit zijn vooral handige utility's.
doc/info/*-    Platform-specifieke tips voor het compileren van bind
doc/*     -    RFC'S (Request for comments) en overige informatie
```

4.3 Voorbereiding voor compileren

Voordat we bind gaan compileren zullen we het pakket aan onze wensen aan moeten passen. Bind komt al ingesteld uit het archief dus we zouden in principe zo kunnen gaan compileren. De opties die op het moment van uitpakken namelijk ingesteld zijn moeten voldoende zijn voor een willekeurige site. Toch lopen we hier even door alle opties heen die aan en/of uitgezet kunnen worden. Deze opties bevinden zich in het bestand `options.h` dat zich in de directory conf bevindt. Als je bind wilt installeren met de meest gangbare opties kan je doorspringen naar hoofdstuk 5 (Het configureren van bind). De beschrijving van deze opties is voor als je wilt weten wat ze doen of als je bind helemaal naar je hand wilt gaan zetten.

DEBUG

Met deze optie, de naam zegt het al kun je debugging aanzetten. Hierdoor kan je wat makkelijker fouten opsporen als er iets mis gaat.

ALLOW_T_UNSPEC

Dit is een optie voor oude "Athena" software die het rr type gebruikt. Je hoeft deze optie naar alle waarschijnlijkheid niet te gebruiken omdat de software bijna niet meer gebruikt wordt. De optie staat in de standaard configuratie ook uit.

ALLOW_UPDATES

Deze optie zorgt ervoor dat dynamische updates mogelijk zijn, je kan dan (als je de juiste client hebt) de database die de nameserver bij houdt "on the fly" updaten. Deze optie bied totaal geen beveiliging omtrent het kijken of je wel mag updaten, iedereen die je name-server kan bereiken mag de namen database updaten. Deze optie wordt dus niet aangeraden.

INVQ

Deze optie is om ervoor te zorgen dat oude versies van nslookup een inverse query kunnen doen. Als je een oude versie van nslookup rond hebt slingeren op je netwerk kan je misschien deze optie aanzetten, maar het wordt aangeraden om liever de optie fake-iquery te gebruiken.

DSTORAGE

Deze optie is er ook weer om ervoor te zorgen dat debugging wat makkelijker gaat. Hij zorgt ervoor dat bij het adresseren van stukken geheugen wordt gekeken of er geen "overruns" plaats vinden. Deze optie heb je niet echt nodig, tenzij je van plan bent om serieus te gaan debuggen. Bij de standaard configuratie staat deze optie ook uit.

DMALLOC

Deze optie zorgt ervoor dat er weer een debug optie wordt aangezet, namelijk om bij te houden hoeveel stukken geheugen gealoceerd worden en wanneer ze worden gebruikt cq vrijgemaakt. Deze optie is normaliter niet nodig tenzij je van plan bent om bind serieus te gaan debuggen.

XFRNETS

Met deze optie kan je ervoor kiezen om in het bestand named.boot een aantal regels op te nemen met daarin de ip-nummers van computers of netwerken die "zone bestanden" mogen kopiëren. Dit is vooral handig als je ervoor wilt zorgen dat niet iedereen zomaar je hele netwerk door kan vlooien. En om de name server niet steeds druk bezig te laten zijn met "zone transfers" die niet mogen te proberen uit te voeren. In de standaard configuratie staat deze optie aan.

PID_FIX

Deze optie zorgt ervoor dat als je bind probeert op te starten en er al een name server draait dat bind dan het bestandje /etc/named.pid of /var/run/named.pid weer terugzet zoals hij het aantrof. In dit bestandje houdt de name server namelijk bij onder welk process-id hij draait. In de standaard configuratie staat deze optie aan.

FWD_LOOP

Deze optie zorgt ervoor dat als de name server in zijn boot-file zieti dat je een van je eigen ip-adressen er neer hebt gezet hij er een opmerking over maakt. Deze optie staat in de standaard configuratie aan.

NO_GLUE

Deze optie zorgt ervoor dat "rotzooi" die meekomt met het zone-bestand van een andere name server tijdens het uitwisselen van zone bestanden wordt uitgekomentaard, zodat het niet in de database terecht komt. Deze optie staat in de standaard configuratie aan.

BOGUSNS

Deze optie zorgt ervoor dat je in het bestand named.boot een regel bogusns kan opnemen. Hier meldt je dan alle adressen van name servers die je wilt negeren. Dit kan een laatste redmiddel zijn als "iemand" zich onterecht loopt te profileren als root-server. Deze optie staat in de standaard configuratie aan.

QRYLOG

Deze optie zorgt ervoor dat een SIGWINCH ervoor zorgt dat de name server alle aanvragen logt. Ook kan je dan de name server opstarten met een -q switch zodat het loggen aangezet wordt. Deze optie staat in de standaard configuratie aan.

LOGFAC

Als je de name server opstart met de -q switch dan produceert hij nogal wat log-data. Deze data wil je waarschijnlijk niet in je syslog hebben waar het normaal gesproken in verdwijnt. Met deze optie kan je ervoor zorgen dat de data met het kenmerk LOC_LOCALn (0 <= n <= 7) Je kan deze optie (LOC_LOCAL) ook defineren. Als je dan in `/etc/syslogd.conf` ook de settings goed veranderd kan je ervoor zorgen dat alles in een apart bestand terecht komt. Deze optie staat in de standaard configuratie aan.

YPKLUDGE

Als je op je netwerk NIS/YP draait kan het voorkomen dat de versie die je draait de namen die niet op de YP-servers zijn te vinden probeert op te zoeken via DNS. Sommige versies van NIS/YP "zien" niet het verschil tussen een deel van een domain-naam en een domain-naam. Als resultaat proberen ze dus een stukje van de domein-naam op te vragen alszijnde een domeinnaam. Als je DNS-server hier niets mee doet zullen deze requests doorgestuurd worden naar andere DNS-servers op het netwerk wat onnodig verkeer tot gevolg heeft. Deze optie zorgt ervoor dat jouw DNS-server op deze requests reageert en gelijk een foutmelding terugstuurt dat de domein-naam niet te vinden is. Deze optie staat in de standaard configuratie uit.

TRACEROOT

Deze optie zorgt ervoor dat er gecheckt wordt op zogenaamde bogus root nameservers. Dit zijn servers die zich voordoen als root server maar het niet zijn of die helemaal niet bestaan. Deze optie staat in de standaard configuratie aan.

LOCALDOM

Deze optie zorgt ervoor dat de DNS-server twee keer kijkt voor een domein- naam. De eerste keer zal hij zoeken naar de naam zoals de client die wil hebben, de tweede keer echter zal hij een standaard domein-naam achter de naam plakken die de client wil hebben. Deze standaard domein-naam kan je specificeren in named.boot configuratie bestand. Deze optie wordt afgeraden omdat een DNS-server eigenlijk nooit in mag gaan schatten wat de client precies zou willen door er gewoon maar een domein-naam aan vast te plakken. Deze optie staat in de standaard configuratie uit.

SLAVE_FORWARD

Deze optie zorgt ervoor dat als er slave servers zijn gedefinieerd dat deze zich verschrikkelijk "goed" gedragen. En query naar 1 van de slave servers zal dan een timeout hebben van 60/N waar N het aantal forwarders is. Deze optie is experimenteel in deze versie van bind, maar wordt ten sterkste aangeraden om te gebruiken. De optie staat in de standaard configuratie aan.

FORCED_RELOAD
Deze optie zorgt ervoor dat als de server een HUP signaal ontvangt hij zijn secondary zones gaat vergelijken met de primary zones ook al hebben deze nog niet hun refresh timeout bereikt. Deze optie staat in de standaard configuratie aan.

WANT_PIDFILE
Deze optie zorgt ervoor dat een bestand genaamd `named.pid` wordt gecreëerd in de directory `/etc` of `/var/run`. Deze optie wordt aangeraden om aan te laten. Deze optie staat in de standaard configuratie aan.

4.4 Het compileren

We zijn nu op het punt aangeland waar we ervoor gaan zorgen dat alle programma-bestanden voor bind worden gecompileerd. Het enigste dat we nog moeten doen is het aanpassen van de zogenaamde Makefile. Hier stellen we nog een aantal opties in voor het systeem waar we bind op willen gaan draaien. Dit is eigenlijk heel simpel, we hoeven enkel bij een paar regels het # (hekje) weg te halen en dan het commando te geven om bind te gaan "maken". Hieronder staat een stukje uit de Makefile afgebeeld, dit is het stuk wat je voor linux moet hebben. Als je door de Makefile heen bladert zie je vanzelf staan waar het stuk voor linux staat.

```
#(Linux - on modern systems, all you need to do is rename or
#remove compat/include/sys/cdefs.h.  See doc/info/Linux for more
#information.)
CC = gcc $(CPPFLAGS)
CDEBUG = -g
CPPFLAGS = -DSYSV
LEX=flex -8 -I
INSTALL_COMPAT = install-compat
LIBS = -lfl
DESTEXEC = /usr/sbin
DESTMAN = /usr/man
MANDIR = man
MANROFF = cat
DESTHELP = /usr/lib
CATEXT = $$$$N
PS = ps -p
IOT = IOT
```

Je zult ook nog iets tegen komen over shared libraries, in dit document wordt daar geen aandacht aan besteed, dus dat kan je zo laten staan met hekjes ervoor. Nu we de Makefile hebben aangepast moeten we nog één ding doen voordat we de opdracht geven tot het compileren van bind. We moeten namelijk een bestand in de directory-tree van bind verwijderen of hernoemen. Dit bestand wat staat in de directory compat/include/sys en wat de naam cdefs.h heeft moeten we verwijderen of hernoemen. Als we dat namelijk niet doen zal bind niet willen compileren. Er zullen dan een aantal fouten optreden en het hele compilatie proces zal afbreken. Het is eigenlijk het beste om het bestand te hernoemen, dit doen we door het volgende commando te geven in de directory `compat/include/sys`:

pinky:~/bind-4.9.7/compat/include/sys$ mv cdefs.h cdefs.h.old

Nadat we dit hebben gedaan kunnen we weer terug gaan naar de root van de bind directory-tree en de opdracht gaan geven om het hele zaakje te laten compileren. We doen dit doormiddel van het commando make. We gaan dus naar de directory toe waar we bind hebben uitgepakt en typen dan :

pinky:~/bind-4.9.7$ make

Hierna zal de computer eventjes bezig zijn met het compileren van bind. Als alles goed gaat is hij er na een tijdje mee klaar en heeft hij geen foutmeldingen gegeven. Je krijgt dan de prompt weer terug.
Noot: Warnings worden niet beschouwd als foutmelding.

Nu komt eigenlijk enkel nog het geven van de opdracht om alles te installeren in de directory's waar het thuishoort. Wat je misschien al gemerkt zal hebben is dat alles tot nu toe als normale user is gedaan. We zullen om de boel in de juiste directory's neer te zetten eerst moeten opwaarderen naar root. Dit gebeurt door het commando su (Set UserID). Allereerst een waarschuwing vooraf, als de installatie van linux al een DNS-server heeft geïnstalleerd en ook al bestanden ervoor neer heeft gezet kan je beter die bestanden even backuppen. Zodra je make install geeft worden die bestanden namelijk zonder pardon vervangen. Wil je eerst kijken wat er vervangen wordt dan kan dat ook, je typt dan het volgende commando :

pinky:~/bind-4.9.7$ make -n install

Die -n zorgt ervoor dat make alleen maar laat zien wat er gaat gebeuren, er wordt dan dus nog niks vervangen. Maar goed laten we maar met de installatie beginnen. We moeten eerst root worden, dus we typen eerst :

pinky:~/bind-4.9.7$ su -

Er zal nu om het root paswoord gevraagd worden, zodat je dat hebt gegeven krijg je de prompt weer terug, alleen ditmaal als root. Je zult nu naar de directory moeten gaan waar je daarstraks, voordat je `su` deed, stond. Hier geef je dan het commando om bind te installeren :

pinky:/home/pvzweden/bind-4.9.7# make install

Na een tijdje krijg je de prompt weer terug en is bind op het systeem geïstalleerd, het enigste wat ons nu nog rest is het maken van de zone bestanden en dan de DNS-server op te starten.

5 Het configureren van bind

In dit hoofdstuk gaan we behandelen hoe we bind nu configureren nadat we de software hebben geïnstalleerd, dit houdt in de zone bestanden maken en de diverse andere bestanden creëren.

5.1 Het aanmaken van named.boot

Nu we de software hebben geïnstalleerd moeten we enkel nog de bestanden aanmaken die de software configureren. Een daarvan is het bestand `named.boot` wat zich in de directory `/etc` hoort te bevinden. In dit bestand staan een aantal dingen, namelijk o.a waar de zone bestanden zich bevinden en ook welk(e) domein(en) de DNS-server beheert. We zullen aan de hand van een voorbeeld van een niet bestaande domein- naam een `named.boot` bestand in elkaar zetten. De domein-naam die we hiervoor gaan gebruiken is linux.bogus, dus waar in het voorbeeld linux.bogus staat genoemd kan je simpelweg je eigen domeinnaam invullen. Allereerst zullen we moeten inloggen als root of ervoor zorgen dat we root zijn, de bestanden moeten namelijk in directory's gemaakt worden waar meestal alleen root schrijfrechten heeft. Daarna gaan we naar de directory `/etc` waar we het bestand gaan creëren, dit doen we met de editor `vi`. (Voor wie geen ervaring heeft met `vi` staat er achterin dit document een quick reference guide afgedrukt)

We geven nu de volgende opdracht :

pinky:/etc# vi named.boot

We zullen waarschijnlijk nu een leeg bestand voor ons hebben (herkenbaar aan een scherm met alleen maar tildes). We kunnen hier nu de configuratie voor de DNS-server in aanmaken, commentaar geven we aan door een punt-comma (;) te typen voor een regel. We beginnen om een directory statement toe te voegen, hetgeen wat we hierbij specificeren is de directory waar de rest van de configuratiebestanden staat. Als we dit niet doen zal er naar alle bestanden worden gezocht in de `/etc` directory. In deze handleiding zetten we alles in de directory `/var/named`. Als deze directory nog niet bestaat moet je hem even aanmaken, we gebruiken ook nog een aantal subdirectory's van deze directory. Als je een andere directory wil gebruiken kan dat natuurlijk ook, je moet dan gewoon de directory die je zelf kiest gebruiken op de plaats waar wordt gerefereerd aan de directory `/var/named`.

We zetten dus het volgende in de `named.boot` :

```
;

; Named.boot bestand voor de DNS-server

;Directory waar we de andere bestanden willen zetten
 directory /var/named
```

We hebben nu de directory gespecificeerd waar we dus de andere bestanden gaan neerzetten. We maken van de DNS-server ook een zogenaamde "caching DNS-server". Dit houdt in dat je de server ook kunt gebruiken voor het opzoeken van internet domein-namen. Als je dat niet wilt moet je dit stuk even overslaan. Om de aanvragen te kunnen "cachen" moet de server natuurlijk wel weten waar hij de antwoorden op de aanvragen kan opzoeken. We specificeren de servers waar hij kan zoeken in het bestand `root.cache`. Hierin noemen we de zogenaamde "root" servers. We nemen hiervoor de volgende regel op in `named.boot` :

```
;Opzoeken van de aanvragen
cache     .                root.cache
```

Let op ! De stukken wit-ruimte die je ziet zijn in dit geval tabs en geen spaties.

Nu we ook dit hebben toegevoegd komt eigenlijk alleen nog het allerbelangrijkste, het instellen van de domeinen die de DNS-server beheert. We geven dit aan met zogenaamde "primary" regels, dit houdt in dat de DNS-server deze namen als primair heeft en dus ook de "baas" ervan is. Hiernaast heb je ook nog "secondary" regels, deze zijn voor domeinen die de server mede-beheert, hij is dus samen met een andere DNS-server verantwoordelijk voor het domein. Onze server wordt verantwoordelijk voor het verzonnen domein linux.bogus. We voegen daarom de volgende regel toe aan onze `named.boot`:

```
;Type     domein              host-bestand
;Primair domein dat de server beheerd
primary   linux.bogus              pz/linux.bogus
```

Ook hier zijn de ruimtes tussen de keywords tabs in plaats van spaties, je moet hier goed op letten, bind is niet zo dol op spaties. We moeten nog één regel toevoegen, namelijk de reverse lookup regel voor onze eigen machine. Deze reverse lookup is een geval apart. Het bestand wat we hier aan gaan hangen dient namelijk om de namen van de machines in op te zoeken als alleen het ip-adres bekent is. In dit geval bevat het bestand waar we de regel naar laten wijzen alleen een pointer naar de localhost, de machine waar de DNS-server op draait en waar dus de namen opgezocht kunnen worden. De regel de we moeten toevoegen ziet er wat anders uit als de andere regels, dit omdat we een reverse lookup aangeven, hierdoor staat het ip-adres omgekeerd geschreven. Dus voor een ip-adres als 192.168.0.1 zou het geschreven worden als 1.0.168.192. We voegen de volgende regel dus toe :

```
;Primair domein voor reverse lookup
primary   0.0.127.IN-ADDR.ARPA          pz/127.0.0
```

Nu we ook deze regel hebben toegevoegd zijn we klaar met het maken van het `named.boot` bestand. We moeten nu nog de zone bestanden maken en een bestand maken waarin wordt verwezen naar de "root-servers" waar onze servers de adressen op kan zoeken die hij niet weet. Als laatste staat hier nog even een afdruk van het `named.boot` bestand zoals je het nu op je scherm hebt staan.

```
;
; Named.boot bestand voor de DNS-server

;Directory waar we de andere bestanden willen zetten
 directory /var/named

;Opzoeken van de aanvragen
cache     .                             root.cache

;Type     domein                   host-bestand
;Primair domein dat de server beheerd
primary   linux.bogus              pz/linux.bogus

;Primair domein voor reverse lookup
primary   0.0.127.IN-ADDR.ARPA     pz/127.0.0
```

Nu we dit bestand af hebben gaan we verder met de andere bestanden, te beginnen bij het bestand waar in staat gespecificeerd waar de "root-servers" zich bevinden.

5.2 Het bestand root.cache

In dit bestand staan de verschillende "root-servers" vermeld, dit zijn de servers die elk een apart "root" domein beheren. De zogenaamde root domeinen zijn de stukjes in de domein-naam die helemaal rechts staan, dus oa : `.com`, `.edu`, `.nl`, enz. Dit bestand is niet moeilijk aan te maken, we kunnen dit namelijk automatisch laten doen met behulp van het commando dig. Voorwaarde hiervoor is wel dat je een internet verbinding hebt waarmee je door dig de gewenste gegevens kunt laten opzoeken. Heb je geen internet verbinding dan kun je gewoon de schermafdruk van het bestand overnemen. Je hebt trouwens niet echt heel veel aan een caching DNS-server als je niet de internet verbinding via dat ding hebt lopen. Voor een klein netwerk heb je meestal toch maar een DNS- server en als je niet bent verbonden met andere netwerken is dit niet echt nodig. Maar hoe laten we `dig` nu dat bestand maken ? Heel simpel : we gaan in de directory `/var/named` staan en typen het volgende commando :

pinky:/var/named# dig . @rs.internic.net >root.cache

Dat is alles, we hebben nu een perfecte `root.cache` gemaakt. Als we het bestand openen zien we iets in de trend van :

```
; <<>> DiG 2.2 <<>> . @rs.internic.net
; (1 server found)
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10
;; flags: qr rd; Ques: 1, Ans: 0, Auth: 13, Addit: 13
;; QUESTIONS:
;;  ., type = A, class = IN

;; AUTHORITY RECORDS:
.   463656 NS F.ROOT-ServerS.NET.
.   463656 NS I.ROOT-ServerS.NET.
.   463656 NS E.ROOT-ServerS.NET.
.   463656 NS D.ROOT-ServerS.NET.
.   463656 NS C.ROOT-ServerS.NET.
.   463656 NS B.ROOT-ServerS.NET.
.   463656 NS H.ROOT-ServerS.NET.
.   463656 NS A.ROOT-ServerS.NET.
.   463656 NS M.ROOT-ServerS.NET.
.   463656 NS L.ROOT-ServerS.NET.
.   463656 NS K.ROOT-ServerS.NET.
.   463656 NS J.ROOT-ServerS.NET.
.   463656 NS G.ROOT-ServerS.NET.

;; ADDITIONAL RECORDS:
F.ROOT-ServerS.NET.     3600000   A    192.5.5.241
I.ROOT-ServerS.NET.     3600000   A    192.36.148.17
E.ROOT-ServerS.NET.     3600000   A    192.203.230.10
D.ROOT-ServerS.NET.     3600000   A    128.8.10.90
C.ROOT-ServerS.NET.     3600000   A    192.33.4.12
B.ROOT-ServerS.NET.     3600000   A    128.9.0.107
H.ROOT-ServerS.NET.     3600000   A    128.63.2.53
A.ROOT-ServerS.NET.     3600000   A    198.41.0.4
M.ROOT-ServerS.NET.     3600000   A    202.12.27.33
L.ROOT-ServerS.NET.     3600000   A    198.32.64.12
K.ROOT-ServerS.NET.     3600000   A    193.0.14.129
J.ROOT-ServerS.NET.     3600000   A    198.41.0.10
G.ROOT-ServerS.NET.     3600000   A    192.112.36.4

;; Total query time: 16016 msec
;; FROM: pinky to Server: rs.internic.net  198.41.0.6
;; WHEN: Sun Feb 28 13:43:33 1999
;; MSG SIZE  sent: 17  rcvd: 436
```

Zoals je ziet heeft `dig` nogal wat output gegenereerd, als je het `root.cache` bestand met de hand moet maken is alleen de informatie zonder punt-komma's relevant. We gaan nu verder met het maken van het bestand voor reverse lookup, dit bestand is nodig, anders kan de DNS-server straks niet gevonden worden.

6 Het bestand voor reverse lookup van de lokale host

Het bestand wat we nu gaan maken is nodig om straks de lokale host zichzelf als DNS-server te laten vinden. Het bestand bevindt zich in de directory `/var/named/pz`, als deze directory nog niet bestaat zul je hem even aan moeten maken met een mkdir. Het bestand draagt de naam `127.0.0.` Dit bestand ziet er als volgt uit :

```
@        IN   SOA  linux.bogus. hostmaster.linux.bogus. (
              1    ; Serial
              28800     ; Refresh
              7200 ; Retry
              604800    ; Expire
              86400 )   ; Minimum TTL
         NS   ns.linux.bogus.
1        PTR  localhost.
```

Zoals je ziet bestaat het bestand uit verschillende onderdelen, zogenaamde resource records (RR) genaamd. Het eerste stuk is het zogenaamde Start Of Authority (SOA). Dit bevat de volgende gegevens :

De naam van de machine waar het zone bestand vandaan komt, in dit geval een machine genaamd `linux.bogus`. Degene die verantwoordelijk is voor het zone bestand, in dit geval `hostmaster.linux.bogus`. (hostmaster@linux.bogus) De verschillende zone parameters, zoals het nummer, de refresh tijd, enz.

Het Apestaartje (@) in het SOA RR staat voor huidig, dus eigenlijk staat er op de plaats van het apestaartje staat eigenlijk : 0.0.127.IN-ADDR.ARPA. Het volgende RR bevat de verwijzing naar de DNS-server voor het domein, in dit geval de machine ns.linux.bogus. Het laatste RR bevat een pointer die verteld dat het adres 127.0.0.1 bekent is onder de naam localhost. Dit was eigenlijk alles wat we moesten invullen voor dit zone bestand, we gaan nu verder met een zone bestand voor een simpel domein, als je dit niet wilt en alleen een caching DNS server wilt kan je doorspringen naar hoofdstuk 4 waarin wordt uitgelegd welke bestanden we nog moeten aanpassen om de DNS-server te laten draaien.

6.1 Een simpel domein

Nu we eigenlijk bijna alles gehad hebben wat bind betreft zijn we nu op het punt aangeland waarin we een zone bestand gaan maken voor een eigen domein. Dit domein is linux.bogus en het bestand waarin het zone bestand voor dit domein zich bevind heeft de naam : linux.bogus. Dit bestand bevindt zich in de directory `/var/named/pz`, dus daar zullen we eerst heen moeten. Hier maken we het bestand `linux.bogus` met behulp van `vi`:

pinky:/var/named/pz# vi linux.bogus

Het bestand ziet er als volgt uit :

```
;Zone bestand voor domein linux.bogus
;
@  IN   SOA               linux.bogus. hostmaster.linux.bogus. (
                          1999280201   ; serial (datum + nummer)
                          28800        ; refresh in seconden
                          7200         ; retry in seconden
```

```
                    3600000 ; expire in seconden
                    86400 ) ; Minimale levensduur
               NS   ns.linux.bogus.
               MX   10 mail.linux.bogus. ; Primare mailserver
               MX   20 mail.ander.bogus. ; Secondaire mailserver

localhost      A    127.0.0.1
ns             A    192.168.0.2
mail        A    192.168.0.3
```

Let op ! Let er goed op dat je zodra je de volledige domein-naam neerzet dat je er een . (punt) achter zet. Anders loopt het heel leuk in de soep.

Zoals je iet is er een RR veld bij gekomen, namelijk het MX veld. Dit veld staat voor Mail Exchanger oftewel degene die de mail voor het domain afhandelt. Dit veld vertelt waar mailservers aan moeten kloppen als ze een mailtje te bezorgen hebben naar bijvoorbeeld jan.jansen@linux.bogus. Je kan meerdere MX velden opnemen in een zone bestand, je geeft ze dan een prioriteit mee. In dit voorbeeld heeft de mailserver mail.linux.bogus de hoogste prioriteit. Mocht deze mailserver niet te bereiken zijn dan kan de mail afgeleverd worden bij mail.ander.bogus. Nu we dit hebben gedaan zijn we eigenlijk klaar met het configureren van bind, we hebben nu een klein domein en we moeten nu enkel nog de bestanden van de lokale machine aanpassen zodat hij zijn eigen DNS- server gaat gebruiken.

7 Het configureren van de rest van het systeem

In dit hoofdstuk behandelen we de overige bestanden die veranderd moeten worden zodat de DNS-server wordt opgestart en door het lokale systeem wordt gebruikt.

7.1 Ervoor zorgen dat bind start bij het booten

Om ervoor te zorgen dat de DNS-server bereikbaar is voor het netwerk moet hij natuurlijk wel opgestart worden bij het starten van het systeem. Hiervoor passen we de opstart-scripts aan. Deze bevinden zich bij de Slackware distributie in de directory `/etc/rc.d`. Waar ze bij jouw staan is afhankelijk van de distributie die je gebruikt. Helaas is de naam van de opstartscripts niet gelijk. Ik zal ter informatie bij de scripts die we veranderen neerzetten in welk runlevel ze worden aangeroepen, misschien dat je dan zelf wel weet welk bestand je moet veranderen. Om ervoor te zorgen dat bind wordt gestart bij het starten van het systeem passen we het bestand `rc.inet2` aan. Dit opstartscript wordt aangeroepen zodra het systeem "multi-user" gaat. Dit is runlevel 3. Vanuit dit script worden alle services gestart, dus als je het bestand doorloopt hoef je eigenlijk alleen maar te kijken of er meerdere "deamons" in worden gestart.

We voegen aan dit bestand de volgende regel toe :

```
if [ -f /usr/sbin/named ]; then
    echo -n " named"
    /usr/sbin/named
fi
```

Deze regels zorgen ervoor dat er gekeken wordt of named (bind) bestaat en zo ja dan wordt hij gestart. Dit is alles, als het systeem nu onverhoopt herstart wordt dan zal bind keurig weer gestart worden.

8 Ervoor zorgen dat het lokale systeem de DNS-server gebruikt

Om ervoor te zorgen dat de server waar de DNS-server op draait zelf ook gebruik maakt van de DNS-server moeten we het bestand resolv.conf aanpassen. Dit bestand bevindt zich in de directory `/etc`. Om ervoor te zorgen dat het systeem dus de DNS-server gebruikt om ip- adressen voor domein-namen in het lokale domein op te zoeken zetten we in het bestand :

```
search linux.bogus
nameserver 127.0.0.1
```

We hebben nu de lokale server geconfigureerd voor het opzoeken van de domein-namen van het domein dat de DNS-server beheerd, hebben we echter ook een caching name server gemaakt dan kunnen we er nog een regeltje aan toevoegen, namelijk :

```
search .
nameserver 127.0.0.1
```

Zo zorgen we ervoor dat de server de lokale DNS-server gebruikt voor alle domeinnamen die hij moet opzoeken.

8.1 Het aanpassen van host.conf

In het bestand `host.conf` staat aangegeven hoe de lokale server de namen opzoekt, met andere woorden waar hij het eerst kijkt als hij op zoek is naar een bepaalde naam. In dit bestand moet een regel staan die er als volgt uit ziet :

```
order hosts, bind
```

Dit zorgt er voor dat de server eerst in het hosts bestand kijkt voordat hij het aan de DNS-server gaat vragen.

We zijn nu klaar met het instellen van de overige bestanden, nu komt de laatste fase, namelijk het starten van bind en kijken of alles goed draait.

9 Het starten en testen van bind

Nu we alles hebben ingesteld is het tijd om bind voor de eerste keer te starten en te kijken of hij goed werkt.

9.1 Het starten

Het starten van bind stelt eigenlijk niks voor, het is gewoon `/usr/sbin/named` intypen en voila hij draait. We kunnen nu in het syslog zien of de start succesvol is geweest. Als het goed is heb je namelijk iets in de trend van het onderstaande in je syslog staan : (Er zijn enters toegevoegd om het geheel leesbaar te maken)

```
Feb 28 17:04:59 pinky named[684]: starting.  named 4.9.7-REL
Sat Feb 20 22:04:27 GMT 1999
pvzweden@pinky:/home/pvzweden/tijdelijk/projekt/bind-4.9.7/named

Feb 28 17:04:59 pinky named[684]: cache zone "" loaded (serial 0)
```

```
Feb 28 17:04:59 pinky named[684]: primary zone
"0.0.127.IN-ADDR.ARPA" loaded (serial 1)

Feb 28 17:04:59 pinky named[684]: primary zone "linux.bogus" loaded
(serial 1999280201)

Feb 28 17:04:59 pinky named[685]: Ready to answer queries.
```

Ok, bind is nu gestart het wordt nu tijd om te kijken of hij het ook naar behoren doet.

9.2 Het testen

Nu bind is gestart gaan we even testen of alles goed is ingesteld, we doen dit met behulp van het programma nslookup. We typen eerst op de prompt :

pinky:˜# nslookup

We krijgen dan op het scherm eerst de informatie van nslookup te zien en daarna een prompt.

```
Default server: localhost
Address: 127.0.0.1

>
```

We kunnen nu de naam van een host intypen. Dat doen we dus ook, we typen in :

```
> ns.linux.bogus
```

We zien nu de volgende output van nslookup verschijnen :

```
Server: localhost
Address: 127.0.0.1

Name: ns.linux.bogus
Address: 192.168.0.2

>
```

Als je hetzelfde ziet verschijnen als wat hier staat afgedrukt dan werkt het. We verlaten nslookup door exit te typen.

9.3 Het laten herladen van de configuratie

Als je een van de configuratie bestanden van bind hebt veranderd, je hebt bijvoorbeeld een host toegevoegd of verwijderd dan zal je bind moeten vertellen dat hij zijn configuratie opnieuw moet inlezen. Met bind komt hiervoor een heel mooi programmaatje mee, namelijk `ndc`. Met `ndc` (Name Deamon Control) kan je een aantal opdrachten geven aan bind. Een van die opdrachten is het herladen van de configuratie. Om de configuratie te herladen typ je het volgende :

pinky:˜# ndc reload

Bind zal nu zijn database opnieuw inladen, ndc geeft ook keurig de melding Reloading Database terug. Met `ndc` kan je een aantal dingen doen, een overzicht :

- **status**
  Hiermee kan je zien of bind draait, als hij draait krijg je het PID (Process ID) van bind te zien.
- **dumpdb**
  Dit zorgt ervoor dat bind zijn complete database en cache naar het bestand named_dump.db wegschrijft in de directory /var/tmp.
- **reload**
  Deze optie hebben we net gebruikt, hij zorgt ervoor dat bind zijn configuratie opnieuw inleest.
- **stats**
  Deze optie zorgt ervoor dat bind zijn statistieken naar een bestand wegschrijft. Dit bestand heet named.stats en is te vinden in de directory `/var/tmp`.
- **trace**
  Deze optie zorgt ervoor dat bind zijn "tracing level" verhoogd. Dit houdt in dat hij meer informatie gaat loggen naar `/var/tmp/named.run`.
- **notrace**
  Deze optie zorgt ervoor dat het "tracing level" van vind op nul wordt gezet. Dit houdt in dat hij geen informatie meer logt naar `/var/tmp/named.run`.
- **querylog**
  Deze optie zorgt ervoor dat bind gaat loggen wie er allemaal een beroep op hem doen. Dit geeft een hoop log-info dus wees gewaarschuwd.
- **start**
  Met deze optie kan je opdracht geven om bind te starten als hij niet draait.
- **stop**
  Je raadt het al, deze optie stopt bind.
- **restart**
  Deze optie zorgt ervoor dat bind herstart wordt.

Dit was het, als het goed is heb je nu een configureerde DNS-server draaien. Als je meer wilt weten over DNS dan verwijs ik je naar internet. Er is behoorlijk wat informatie in omloop.

10 Quick Reference Vi

Hier volgen een aantal commando's waarmee je het editen in een bestand kan starten :

Toetsen die je kan geven vanuit de commando-modus

i
Hiermee start je de insert-mode van vi

o
Hiermee start je ook de insert-mode maar je opent nu eerst een nieuwe regel

Tot slot nog een aantal links:

The DNS Resources Directory
http://www.dns.net/dnsrd/ `<http://www.dns.net/dnsrd/>`

De officiële homepage v bind
http://www.isc.org/bind.html `<http://www.isc.org/bind.html>`

Common DNS Operational and Configuration Errors
http://www.pmg.lcs.mit.edu/cgi-bin/rfc/view?number=1912 `<http://www.pmg.lcs.mit.edu/cgi-bin/rfc/view?number=1912>`

Address Allocation for Private Internets
http://www.pmg.lcs.mit.edu/cgi-bin/rfc/view?number=1918 `<http://www.pmg.lcs.mit.edu/cgi-bin/rfc/view?number=1918>`

Domain Administrators Operations Guide
http://www.pmg.lcs.mit.edu/cgi-bin/rfc/view?number=1033 `<http://www.pmg.lcs.mit.edu/cgi-bin/rfc/view?number=1033>`

:w
Hiermee schrijf je het bestand weg waar je op waarmee je op dat moment bezig bent.

:q
Hiermee verlaat je vi

Toetsen die je kan geven vanuit de edit-modus

<esc>
Door op escape te drukken kom je in de commando-modus van vi terecht.