DNS HOWTO

Nicolai Langfeldt *janl@math.uio.no*,
Vertaald door Maarten van Steenbergen *maarten@nl.linux.org* v2.2, 11 februari 1999

Deze HOWTO is een inleiding in het configureren van een nameserver met BIND.

Inhoudsopgave

1 Inleiding

Keywords: DNS, bind, bind-4, bind-8, named, dialup, ppp, slip, isdn, Internet, domain, name, hosts, resolving, caching.

Dit document is een onderdeel van het Linux Documentation Project.

1.1 Copyright

1.2 Dank aan...

Ik wil Arnt Gulbrandsen bedanken die de proefversies van dit document heeft doorgelezen en verbeterd, en enkele handige suggesties heeft gegeven. Ik wil ook de talrijke mensen bedanken die suggesties en tips mailden.

Dit document is nooit af. Als je fouten ziet of een verbetering weet mail dan even, dan kan de HOWTO beter worden. Zend commentaar en/of vragen of geld naar *janl@math.uio.no*. Als je email verzendt en je wilt een antwoord, zorg er dan voor dat het antwoordadres er duidelijk bijstaat en het ook doet. Anders moet je even de 8 (QnA) sectie lezen voor je me mailt. Ik spreek Noors en Engels, dus mail in één van die twee talen.

Mail me als je deze HOWTO wilt vertalen. Ten eerste kan ik dan bijhouden in welke talen dit document is verschenen. Ten tweede kan ik je dan op de hoogte houden van nieuwe versies.

1.3 Opgedragen aan...

Deze HOWTO is opgedragen aan Anne Line Norheim Langfeldt. Ze zal het echter nooit lezen, omdat dit niet haar interessegebied is.

2 Introductie

Wat is DNS?

DNS staat voor 'Domain Name System'. DNS vertaalt namen van computers naar een IP adres. Elke computer op een netwerk heeft een uniek IP adres. DNS converteert namen naar nummers en andersom. Deze conversie is niets anders dan een "tabel"met een kolom namen (zoals `ftp.linux.org`) en een kolom adressen (zoals `199.249.150.4`).

DNS is voor de beginner (jij ;-) één van de minder doorzichtige aspecten van systeembeheer. Dit document tracht een aantal zaken op te helderen. Het beschrijft het opzetten van een *simpele* DNS server. Het begint met het configureren van een 'caching-only' DNS server, en behandelt vervolgens een 'primary' DNS server voor een domein. Voor complexere configuraties kan je bij het 8 (QnA) gedeelte terecht. Als het daar nog niet wordt besproken zal je de Echte Documentatie moeten lezen. In 9 (het laatste hoofdstuk) wordt naar deze Echte Documentatie verwezen.

Voordat je begint met DNS en BIND zal je eerst je machine zodanig moeten configureren dat je er vanuit en naartoe kan telnetten. In elk geval moet `telnet 127.0.0.1` werken (test het nu!). Ook zal je reeds goede `/etc/host.conf`, `/etc/hosts.conf`, en `/etc/resolv.conf` moeten hebben; de werking van deze bestanden wordt niet in deze HOWTO uitgelegd. Als deze basis van je netwerkconfiguratie nog niet goed is ingesteld, lees dan eerst de NET-3/4-HOWTO.

Als in deze HOWTO verwezen wordt naar 'je machine', dan wordt de machine bedoeld waarop je DNS aan het configureren bent. Dus niet een andere machine die op het netwerk aanwezig is.

Ik neem aan dat je niet achter een firewall zit die DNS verkeer tegenhoudt. Als dat wel het geval is, kijk dan in de 8 (QnA) sectie voor een oplossing.

De meest gebruikte nameserver software is `named`. Dit programma is onderdeel van "BIND", dat gemaakt wordt door het Internet Software Consortium (`www.isc.org`). Named is onderdeel van bijna elke Linux distributie en staat gewoonlijk in `/usr/sbin/named`. Als je het programma nog niet hebt, kan je het downloaden van *ftp.isc.org:/isc/bind/src/cur/bind-8/*. Deze HOWTO gaat over BIND versie 8. De oude HOWTO, over BIND versie 4, is nog steeds beschikbaar op *http://www.math.uio.no/~janl/DNS/*. Het versienummer is te controleren in de named man pagina. Als onderaan bij de FILES sectie het bestand `named.conf` genoemd staat, dan is het versie 8. Als gesproken wordt over `named.boot`, dan is het versie 4. Als je versie 4 draait, is het onder meer om beveiligingsredenen aan te raden naar versie 8 te upgraden.

DNS is een gedistribueerde database. Let dus op wat je erin zet. Als je er onzin in zet, dan krijg je er onzin uit en zullen ook anderen er last van hebben. Houd je DNS configuratie schoon en consistent, dan zal DNS erg nuttig zijn. Leer het te gebruiken, beheer het, debug het, en je zal bijdragen aan het voorkomen van mismanagement van het Internet.

In dit document wordt ter vereenvoudiging niet altijd de volledige waarheid verteld. Alles zal echter (waarschijnlijk ;) werken, als je doet wat in dit document beschreven staat.

Tip: Maak backups van alle bestanden die je wijzigt tijdens het testen van de DNS configuratie. Mocht het misgaan, dan kan je altijd nog de oude bestanden terugzetten.

3 Een 'caching only' nameserver

Een eerste probeersel met DNS, erg bruikbaar voor inbelverbindingen

Een 'caching only' nameserver zoekt adressen bij andere nameservers. De volgende keer dat om het adres voor dezelfde machinenaam gevraagd wordt, heeft de nameserver het antwoord al in het geheugen staan. Dit betekent een behoorlijke tijdswinst, zeker op langzame verbindingen.

Eerst moet een bestand `/etc/named.conf` aangemaakt worden. Dit bestand wordt gelezen als de nameserver start. Voorlopig bevat dit bestand alleen het volgende:

```
// Config file for caching only name server

options {
directory "/var/named";

// Uncommenting this might help if you have to go through a
// firewall and things are not working out:

// query-source port 53;
```

```
};

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

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

Named kan aan de `directory` regel zien waar de configuratiebestanden gezocht moeten worden. Elk bestand dat hierna in `named.conf` voorkomt, staat onder de directory die je opgeeft. In dit voorbeeld is `pz` een subdirectory van `/var/named`, het is dus de directory `/var/named/pz`.

In het bovenstaande configuratiebestand wordt `/var/named/root.hints` genoemd. `/var/named/root.hints` moet het volgende bevatten:

```
;
; There might be opening comments here if you already have this file.
; If not don't worry.
;
.                       6D IN NS        G.ROOT-SERVERS.NET.
.                       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.
.                       6D IN NS        A.ROOT-SERVERS.NET.
.                       6D IN NS        H.ROOT-SERVERS.NET.
.                       6D IN NS        B.ROOT-SERVERS.NET.
.                       6D IN NS        C.ROOT-SERVERS.NET.
.                       6D IN NS        D.ROOT-SERVERS.NET.
.                       6D IN NS        E.ROOT-SERVERS.NET.
.                       6D IN NS        I.ROOT-SERVERS.NET.
.                       6D IN NS        F.ROOT-SERVERS.NET.

G.ROOT-SERVERS.NET.     5w6d16h IN A    192.112.36.4
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
A.ROOT-SERVERS.NET.     5w6d16h IN A    198.41.0.4
H.ROOT-SERVERS.NET.     5w6d16h IN A    128.63.2.53
B.ROOT-SERVERS.NET.     5w6d16h IN A    128.9.0.107
C.ROOT-SERVERS.NET.     5w6d16h IN A    192.33.4.12
D.ROOT-SERVERS.NET.     5w6d16h IN A    128.8.10.90
E.ROOT-SERVERS.NET.     5w6d16h IN A    192.203.230.10
I.ROOT-SERVERS.NET.     5w6d16h IN A    192.36.148.17
F.ROOT-SERVERS.NET.     5w6d16h IN A    192.5.5.241
```

Dit bestand levert informatie over de root nameservers op deze wereld. Deze informatie kan mettertijd veranderen, en moet dus worden bijgehouden. Zie de sectie 6 (Onderhoud) voor informatie over het bijhouden van dit bestand.

De volgende sectie van `named.conf` is de laatste `zone`. Het gebruik hiervan wordt later uitgelegd. Voorlopig is het voldoende het volgende bestand genaamd `127.0.0` in de subdirectory `pz` te plaatsen:

```
@               IN      SOA     ns.linux.bogus. hostmaster.linux.bogus. (
                        1       ; Serial
                        8H      ; Refresh
                        2H      ; Retry
                        1W      ; Expire
                        1D)     ; Minimum TTL
                NS      ns.linux.bogus.
1                       PTR     localhost.
```

Vervolgens heb je ook nog een `/etc/resolv.conf` nodig, dat er ongeveer uitziet als:

```
search subdomain.your-domain.edu your-domain.edu
nameserver 127.0.0.1
```

De '`search`' regel geeft aan welke domeinen afgezocht moeten worden naar machinenamen die je zoekt. De `nameserver` regel bevat het IP adres van de DNS server die gebruikt moet worden. In dit geval is dat je eigen machine. De lokale machine zit altijd op IP adres `127.0.0.1`. Als je meerdere nameservers wilt gebruiken, geef dan een aparte `nameserver` regel op voor elke DNS server die je wilt gebruiken. Named leest dit bestand overigens nooit in. Dit bestand wordt alleen gebruikt door applicaties die een IP adres opzoeken.

Een uitleg van dit bestand: Als een programma de naam `foo` opzoekt, dan wordt eerst `foo.subdomain.your-domain.edu` geprobeerd, daarna `foo.your-domain.edu`, en tenslotte `foo`. Als een programma het IP adres van `www.linux.org` zoekt dan wordt eerst `www.linux.org.subdomain.your-domain.edu` geprobeerd, vervolgens `www.linux.org.your-domain.edu`, en tenslotte `www.linux.org`. Het is vreemd, maar zo werkt het nu eenmaal. Het opzoeken van elke naam kost tijd. Je moet dus niet al te veel domeinen in de `search` regel opnemen.

Dit voorbeeld gaat ervan uit dat het domein `subdomain.your-domain.edu` bestaat. Je eigen machine heet dan waarschijnlijk `your-machine.subdomain.your-domain.edu`. De `search` regel moet nooit een TLD (Top Level Domain, `edu` in dit geval) bevatten. Als je vaak machines in een ander domein bezoekt, kan je de volgende regel opnemen:

```
search subdomain.your-domain.edu your-domain.edu other-domain.com
```

Natuurlijk moet je deze regel aan je eigen situatie aanpassen. Merk op dat aan het eind van deze domains geen punt staat. Dit is belangrijk!

Vervolgens zal je, afhankelijk van je libc versie, `/etc/nsswitch.conf` of `/etc/host.conf` aan moeten passen. Als je een `/etc/nsswitch.conf` bestand hebt, dan passen we dat aan. Anders passen we `/etc/host.conf` aan.

/etc/nsswitch.conf

`/etc/nsswitch.conf` is een lang bestand waarin beschreven wordt waar data vandaan gehaald moet worden. Bovenin staat bruikbaar commentaar wat je zou moeten lezen. Zoek daarna de regel die begint met `hosts:`. Daar moet staan:

```
hosts:      files dns
```

Als er geen `hosts:` regel in staat, voeg dan bovenstaande regel toe. De regel zorgt ervoor dat programma's eerst in `/etc/hosts` kijken, en dan pas de nameserver om een IP adres vragen.

/etc/host.conf

Dit bestand bevat waarschijnlijk meerdere regels. Eén van die regels moet beginnen met '`order`', en moet er ongeveer als volgt uitzien:

```
order hosts,bind
```

Als deze regel er niet staat, maak er dan één aan. De regel zorgt ervoor dat eerst in `/etc/hosts` gekeken wordt, en dan pas de nameserver gebruikt wordt.

3.1 Starten van Named

Nu is het tijd om named te starten. Als je een inbelverbinding gebruikt, bel dan eerst in. Type '`ndc start`' en tik enter. Als dat niet werkt, tik dan `/usr/sbin/ndc start`. Als het dan nog niet werkt, raadpleeg dan de 8 (QnA) sectie. Als je het bestand `/var/log/messages` bekijkt tijdens het opstarten van named (bijvoorbeeld met het commando `"tail -f /var/log/messages"), dan zie je iets als:`

`(regels die eindigen op "\"lopen door op de volgende regel)`

```
Feb 15 01:26:17 roke named[6091]: starting.  named 8.1.1 Sat Feb 14 \
  00:18:20 MET 1998 ^Ijanl@roke.uio.no:/var/tmp/bind-8.1.1/src/bin/named
Feb 15 01:26:17 roke named[6091]: cache zone "" (IN) loaded (serial 0)
Feb 15 01:26:17 roke named[6091]: master zone "0.0.127.in-addr.arpa" \
  (IN) loaded (serial 1)
Feb 15 01:26:17 roke named[6091]: listening [127.0.0.1].53 (lo)
Feb 15 01:26:17 roke named[6091]: listening [129.240.230.92].53 (ippp0)
Feb 15 01:26:17 roke named[6091]: Forwarding source address is [0.0.0.0].1040
Feb 15 01:26:17 roke named[6092]: Ready to answer queries.
```

`Als er een foutmelding verschijnt, geeft named aan in welk bestand de fout zit (named.conf of root.hints hoop ik ;-). Stop named, en controleer de bestanden.`

`Nu is het zaakje klaar om getest te worden. Start nslookup om de configuratie te testen.`

```
$ nslookup
Default Server:  localhost
Address:  127.0.0.1

>
```

`Als dit op het scherm verschijnt, dan werkt het. Hopen we. Als het nog niet werkt, controleer alles dan nog eens. Altijd als named.conf veranderd wordt, moet je named herstarten met ndc restart.`

`Nu kan je proberen een IP adres op te zoeken. Gebruik een computer die dicht in de buurt staat. In mijn geval is dat par.uio.no, op de Universiteit van Oslo:`

```
> pat.uio.no
Server:  localhost
```

```
Address:  127.0.0.1

Name:    pat.uio.no
Address:  129.240.130.16
```

nslookup gaf aan named de opdracht par.uio.no op te zoeken. Het ondervraagt vervolgens één van de nameservers die in root.hints genoemd staan. Het kan even duren voordat je antwoord krijgt, omdat eerst gezocht wordt in elk domein dat in /etc/resolv.conf genoemd staat.

Als je nogmaals dezelfde machine opzoekt krijg je dit:

```
> pat.uio.no
Server:  localhost
Address:  127.0.0.1

Non-authoritative answer:
Name:    pat.uio.no
Address:  129.240.2.50
```

Let op de regel 'Non-authoritative answer'. Dat betekent dat named dit keer niet het netwerk op is gegaan om een antwoord te zoeken, maar dat het antwoord uit de cache komt. Deze melding wordt gegeven zodat je er rekening mee kan houden dat de informatie inmiddels incorrect kan zijn. Dat je dit antwoord gekregen hebt betekent in elk geval dat je caching-only nameserver werkt. nslookup kan je verlaten door het commando exit te tikken.

3.2 Verbeteren van de configuratie

In grote netwerken wordt vaak gebruik gemaakt van "forwarders". Deze machines verzorgen het opzoeken van een adres voor een andere machine. Dit vermindert de belasting op het interne en externe netwerk. Door gebruik te maken van zo'n forwarder worden de DNS lookups sneller, en ze genereren minder netwerkverkeer. Vooral als je een modemverbinding hebt kan dit behoorlijk wat schelen. Voor het voorbeeld nemen we aan dat je provider 2 nameservers heeft, met de IP nummers 10.0.0.1 en 10.1.0.1. In het named.conf bestand moet bij de eerste sectie, öptions", worden toegevoegd:

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

Herstart named en test het met nslookup. Dit zou zonder problemen moeten werken.

3.3 Gefeliciteerd

Nu weet je hoe je een caching nameserver opzet. Neem een biertje, een glas melk, of wat dan ook om het te vieren!

4 Een voorbeeld van een *simpel* domein

Het opzetten van je eigen domein

4.1 Maar eerst wat droge theorie

Voordat we *echt* met deze sectie aan de slag gaan, behandel ik eerst wat droge theorie met voorbeelden. Lees dit onderdeel, of neem het tenminste vluchtig door. Als je het vluchtig doorneemt, pak de draad dan weer op bij het gedeelte waar named.conf aangepast wordt.

DNS is een hierarchisch systeem met een boom-structuur. De 'top' heet . en dit wordt 'root' genoemd. Onder . zijn een aantal Top Level Domains (TLD), waarvan de meest bekende COM, NET, ORG, en EDU zijn. Net als een boom begint de stuctuur ergens, en het breidt zich steeds verder uit. Als je een informatica-achtergrond hebt, dan herken je in DNS een 'search tree' met knooppunten, eindpunten, en begrenzingen.

Als je een machinenaam opvraagt, begint de zoektocht in de top van de hierarchie. Als je bijvoorbeeld prep.ai.mit.edu zoekt dan moet je nameserver de server vinden die het edu domein verzorgt. Daarvoor ondervraagt het een . server. Van deze servers heeft de nameserver al het adres, dat staat namelijk in het root.hints bestand. De . server geeft een lijst van edu servers:

```
$ nslookup
Default Server:  localhost
Address:  127.0.0.1
```

Ondervraag een root server:

```
> server c.root-servers.net.
Default Server:  c.root-servers.net
Address:  192.33.4.12
```

Zet het 'query type' op NS (Name Server gegevens):

```
> set q=ns
```

Vraag naar edu:

```
> edu.
```

De . achter edu is erg belangrijk. Het vertelt nslookup dat edu onder . zit, en niet onder een van onze 'search' domeinen. Dit versnelt het zoeken.

```
edu     nameserver = A.ROOT-SERVERS.NET
edu     nameserver = H.ROOT-SERVERS.NET
edu     nameserver = B.ROOT-SERVERS.NET
edu     nameserver = C.ROOT-SERVERS.NET
edu     nameserver = D.ROOT-SERVERS.NET
edu     nameserver = E.ROOT-SERVERS.NET
edu     nameserver = I.ROOT-SERVERS.NET
edu     nameserver = F.ROOT-SERVERS.NET
```

```
edu     nameserver = G.ROOT-SERVERS.NET
A.ROOT-SERVERS.NET      internet address = 198.41.0.4
H.ROOT-SERVERS.NET      internet address = 128.63.2.53
B.ROOT-SERVERS.NET      internet address = 128.9.0.107
C.ROOT-SERVERS.NET      internet address = 192.33.4.12
D.ROOT-SERVERS.NET      internet address = 128.8.10.90
E.ROOT-SERVERS.NET      internet address = 192.203.230.10
I.ROOT-SERVERS.NET      internet address = 192.36.148.17
F.ROOT-SERVERS.NET      internet address = 192.5.5.241
G.ROOT-SERVERS.NET      internet address = 192.112.36.4
```

Dit vertelt ons dat alle ROOT-SERVERS.NET vragen beantwoorden over EDU, dus we kunnen op dezelfde server blijven doorvragen. We blijven de server C ondervragen. Nu willen we weten wat de server is voor mit.edu:

```
> mit.edu.
Server:  c.root-servers.net
Address:  192.33.4.12

Non-authoritative answer:
mit.edu nameserver = W20NS.mit.edu
mit.edu nameserver = BITSY.mit.edu
mit.edu nameserver = STRAWB.mit.edu

Authoritative answers can be found from:
W20NS.mit.edu   internet address = 18.70.0.160
BITSY.mit.edu   internet address = 18.72.0.3
STRAWB.mit.edu  internet address = 18.71.0.151
```

strawb, w20ns en bitsy zijn allemaal nameservers voor mit.edu. We kiezen hieruit een nameserver en gaan weer een stap verder, naar ai.mit.edu:

```
> server W20NS.mit.edu.
```

Het maakt niet uit of je hoofd- of kleine letters gebruikt in de servernaam. Maar ik gebruik m'n muis om te knippen en plakken zodat ik geen spelfouten maak.

```
Server:  W20NS.mit.edu
Address:  18.70.0.160

> ai.mit.edu.
Server:  W20NS.mit.edu
Address:  18.70.0.160

Non-authoritative answer:
ai.mit.edu      nameserver = ALPHA-BITS.AI.MIT.EDU
ai.mit.edu      nameserver = GRAPE-NUTS.AI.MIT.EDU
ai.mit.edu      nameserver = TRIX.AI.MIT.EDU
ai.mit.edu      nameserver = MUESLI.AI.MIT.EDU
ai.mit.edu      nameserver = LIFE.AI.MIT.EDU
ai.mit.edu      nameserver = BEET-CHEX.AI.MIT.EDU
ai.mit.edu      nameserver = MINI-WHEATS.AI.MIT.EDU
ai.mit.edu      nameserver = COUNT-CHOCULA.AI.MIT.EDU
ai.mit.edu      nameserver = MINTAKA.LCS.MIT.EDU
```

```
Authoritative answers can be found from:
AI.MIT.EDU      nameserver = ALPHA-BITS.AI.MIT.EDU
AI.MIT.EDU      nameserver = GRAPE-NUTS.AI.MIT.EDU
AI.MIT.EDU      nameserver = TRIX.AI.MIT.EDU
AI.MIT.EDU      nameserver = MUESLI.AI.MIT.EDU
AI.MIT.EDU      nameserver = LIFE.AI.MIT.EDU
AI.MIT.EDU      nameserver = BEET-CHEX.AI.MIT.EDU
AI.MIT.EDU      nameserver = MINI-WHEATS.AI.MIT.EDU
AI.MIT.EDU      nameserver = COUNT-CHOCULA.AI.MIT.EDU
AI.MIT.EDU      nameserver = MINTAKA.LCS.MIT.EDU
ALPHA-BITS.AI.MIT.EDU   internet address = 128.52.32.5
GRAPE-NUTS.AI.MIT.EDU   internet address = 128.52.36.4
TRIX.AI.MIT.EDU internet address = 128.52.37.6
MUESLI.AI.MIT.EDU       internet address = 128.52.39.7
LIFE.AI.MIT.EDU internet address = 128.52.32.80
BEET-CHEX.AI.MIT.EDU    internet address = 128.52.32.22
MINI-WHEATS.AI.MIT.EDU  internet address = 128.52.54.11
COUNT-CHOCULA.AI.MIT.EDU        internet address = 128.52.38.22
MINTAKA.LCS.MIT.EDU     internet address = 18.26.0.36
```

muesli.ai.mit.edu is dus een nameserver voor ai.mit.edu:

```
> server MUESLI.AI.MIT.EDU
Default Server:  MUESLI.AI.MIT.EDU
Address:  128.52.39.7
```

Nu verander ik het query type. We hebben de nameserver gevonden, dus kunnen nu kijken wat muesli weet over prep.ai.mit.edu.

```
> set q=any
> prep.ai.mit.edu.
Server:  MUESLI.AI.MIT.EDU
Address:  128.52.39.7

prep.ai.mit.edu CPU = dec/decstation-5000.25    OS = unix
prep.ai.mit.edu
        inet address = 18.159.0.42, protocol = tcp
          ftp  telnet  smtp  finger
prep.ai.mit.edu preference = 1, mail exchanger = gnu-life.ai.mit.edu
prep.ai.mit.edu internet address = 18.159.0.42
ai.mit.edu      nameserver = beet-chex.ai.mit.edu
ai.mit.edu      nameserver = alpha-bits.ai.mit.edu
ai.mit.edu      nameserver = mini-wheats.ai.mit.edu
ai.mit.edu      nameserver = trix.ai.mit.edu
ai.mit.edu      nameserver = muesli.ai.mit.edu
ai.mit.edu      nameserver = count-chocula.ai.mit.edu
ai.mit.edu      nameserver = mintaka.lcs.mit.edu
ai.mit.edu      nameserver = life.ai.mit.edu
gnu-life.ai.mit.edu     internet address = 128.52.32.60
beet-chex.ai.mit.edu    internet address = 128.52.32.22
alpha-bits.ai.mit.edu   internet address = 128.52.32.5
mini-wheats.ai.mit.edu  internet address = 128.52.54.11
trix.ai.mit.edu internet address = 128.52.37.6
```

```
muesli.ai.mit.edu       internet address = 128.52.39.7
count-chocula.ai.mit.edu        internet address = 128.52.38.22
mintaka.lcs.mit.edu     internet address = 18.26.0.36
life.ai.mit.edu internet address = 128.52.32.80
```

We zijn gestart op . en vonden nameservers voor elke laag van de machinenaam. Als je eigen DNS server zo'n zoektocht zou ondernemen, zou het gecached worden en voorlopig niet meer opgezocht hoeven worden.

In de boomstructuur is elke . in de machinenaam een vertakking. En elk woord dat tussen twee punten staat is een "takïn de boomstructuur.

We zoeken steeds hoger in de boom, op zoek naar prep.ai.mit.edu. We beginnen bij . en zoeken naar de eerstvolgende tak. In dit geval edu. Als we die hebben gevonden gaan we een stapje hoger, en komen bij de edu server. Daar kunnen we vragen naar het domein mit. Als we die tak hebben gevonden gaan we weer een stapje hoger en komen op mit.edu. Als we nog een stapje hoger zitten, komen we bij ai.mit.edu en dan zijn we bij de laatste nameserver beland. De laatste stap is het zoeken naar prep.ai.mit.edu. In informaticatermen zou je prep een 'leaf', blad, of eindpunt van de boom kunnen noemen.

Een domein waar minder over gesproken wordt, maar wat minstens net zo belangrijk is, is het in-addr.arpa domein. Dit domein heeft net als de normale domeinen een boomstructuur. in-addr.arpa maakt het mogelijk dat een machinenaam opgezocht wordt, als je reeds het IP adres weet. Het is belangrijk op te merken dat IP adressen in omgekeerde volgorde worden genoteerd binnen het in-addr.arpa domein. Als je het adres van een machine hebt, bijvoorbeeld 192.128.52.43, dan werkt named op dezelfde manier als bij het prep.ai.mit.edu voorbeeld: zoek de arpa. servers. Zoek vervolgens de in-addr.arpa. servers, zoek 192.in-addr.arpa., zoek 128.192.in-addr.arpa., zoek 52.128.192.in-addr.arpa., zoek de benodigde informatie voor 43.52.128.192.in-addr.arpa. Handig, toch? (zeg 'ja'). Het omdraaien van de nummers kan echter verwarrend zijn.

Wat ik net verteld heb is niet helemaal waar. DNS werkt niet precies zoals hierboven beschreven staat. Maar het komt aardig dicht in de buurt.

4.2 Je eigen domein

Nu ga je een eigen domein maken. We maken het domein linux.bogus, en zetten daar machines in. We gebruiken expres een domein dat niet bestaat, zodat het geen conflict oplevert met echte domeinen.

Nog één ding voordat we beginnen: in machinenamen zijn niet alle tekens toegestaan. We mogen alleen letters (a-z), cijfers (0-9) en het minteken ('-') gebruiken. Gebruik dus alleen deze tekens. Hoofd- en kleine letters zijn voor DNS hetzelfde, dus pat.uio.no is hetzelfde als Pat.UiO.No.

We zijn eigenlijk al met dit onderdeel begonnen door de volgende regel in named.conf te plaatsen:

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

Merk op dat er geen . staat aan het eind van de domeinnamen in dit bestand. Dit zorgt ervoor dat, nu we de zone 0.0.127.in-addr.arpa configureren, we de master server voor deze zone zijn en dat het in een bestand staat genaamd pz/127.0.0. We hebben dit bestand al geconfigureerd, en er staat in:

```
@               IN      SOA     ns.linux.bogus. hostmaster.linux.bogus. (
                                1       ; Serial
                                8H      ; Refresh
                                2H      ; Retry
                                1W      ; Expire
                                1D)     ; Minimum TTL
                        NS      ns.linux.bogus.
1                       PTR     localhost.
```

Let op de . aan het eind van de volledige domeinnamen in dit bestand, in tegenstelling tot het named.conf bestand hierboven. Sommige mensen beginnen elk zone bestand graag met een $ORIGIN regel, maar dat is overbodig. De oorsprong van een zone bestand (de plaats die het inneemt in de DNS hierarchie) wordt gespecificeerd in het zone gedeelte van het named.conf bestand, in dit geval is het 0.0.127.in-addr.arpa.

Dit zone bestand bevat drie 'Resource Records' (RR's). Een SOA RR, een NS RR, en een PRT RR. SOA betekent Start Of Authority. De '@' is een notatie voor 'oorsprong' en omdat in named.conf staat dat het domein voor dit bestand 0.0.127.in-addr.arpa is, staat er op de eerste regel eigenlijk:

```
0.0.127.in-addr.arpa.   IN      SOA ...
```

NS is de Name Server RR. Er staat geen '@' aan het begin van deze regel. Omdat de bovenstaande regel een '@' heeft staan, wordt verondersteld dat datzelfde ook bij deze regel hoort. Op de NS regel zou dus net zo goed kunnen staan:

```
0.0.127.in-addr.arpa.   IN      NS      ns.linux.bogus
```

Deze regel vertelt DNS wat de nameserver is voor het domein 0.0.127.in-addr.arpa, het is ns.linux.bogus. 'ns' is een veelgebruikte naam voor nameservers, net als 'www' dat is voor webservers. Maar net zoals bij webservers mag je van de naam maken wat je wilt.

Tenslotte zegt het PTR record dat de machine op adres 1 in het subnet 0.0.127.in-addr.arpa, ofwel 127.0.0.1, localhost heet.

Alle zone bestanden beginnen met een SOA record, en er mag er hoogstens 1 zijn in een zone bestand. Het beschrijft de zone, zegt waar het vandaan komt (een machine genaamd ns.linux.bogus), wie verantwoorlijk is voor de inhoud (hostmaster@linux.bogus, vul je eigen email adres hier in), welk versienummer van het zonebestand het is (hier: serienummer 1), en dingen die te maken hebben met cachen en secundaire DNS servers. Gebruik voor de andere velden (refresh, retry, expire, en minimum) de getallen die in deze HOWTO gebruikt worden, dan zit je goed.

Herstart nu named met "ndc restartën gebruik nslookup om te kijken wat je hebt gedaan:

```
$ nslookup

Default Server:  localhost
```

```
Address:  127.0.0.1

> 127.0.0.1
Server:  localhost
Address:  127.0.0.1

Name:    localhost
Address:  127.0.0.1
```

named begrijpt dus dat 127.0.0.1 localhost heet, goed zo. Nu het belangrijke werk, het linux.bogus domein. Voeg een zone sectie aan named.conf toe:

```
zone "linux.bogus" {
        notify no;
        type master;
        file "pz/linux.bogus";
};
```

Let weer op het ontbreken van . aan het einde van de domein naam in het named.conf bestand.

In het linux.bogus zone bestand zetten we wat zelfbedachte gegevens:

```
;
; Zone file for linux.bogus
;
; The full zone file
;
@       IN      SOA     ns.linux.bogus. hostmaster.linux.bogus. (
                        199802151       ; serial, todays date + todays serial #
                        8H              ; refresh, seconds
                        2H              ; retry, seconds
                        1W              ; expire, seconds
                        1D )            ; minimum, seconds
;
                NS      ns              ; Inet Address of name server
                MX      10 mail.linux.bogus     ; Primary Mail Exchanger
                MX      20 mail.friend.bogus.   ; Secondary Mail Exchanger
;
localhost       A       127.0.0.1
ns              A       192.168.196.2
mail            A       192.168.196.4
```

We maken nog twee opmerkingen over het SOA record. ns.linux.bogus moet een bestaande machinenaam zijn met een A record. Het is niet voldoende om een CNAME te hebben voor de machine die in het SOA record genoemd wordt. De naam hoeft niet 'ns' te zijn, je mag elke naam gebruiken die je wilt. Vervolgens, lees hostmaster.linux.bogus als 'hostmaster@linux.bogus'. Dit moet het email adres zijn van de persoon die verantwoordelijk is voor de DNS configuratie. Je mag zelf weten welk email adres je hier plaatst, maar hostmaster is een adres dat veel gebruikt wordt.

Er is een nieuw type 'resource record' in dit bestand, het MX record of Mail eXchanger RR. Mail servers kunnen hieraan zien dat mail voor iemand@linux.bogus naar mail.linux.bogus of mail.friend.bogus gestuurd moet worden. Het nummer voor elke machinenaam is de prioriteit van de MX records. De RR met het laagste nummer (10) is de machine waar mail heen gestuurd moet worden, indien mogelijk. Als dat niet lukt wordt de mail gestuurd naar een machine met een hoger nummer, in dit geval mail.friend.bogus, met prioriteit 20.

Herstart named weer door ndc restart te draaien, en controleer het resultaat met nslookup:

```
$ nslookup
> set q=any
> linux.bogus
Server:  localhost
Address:  127.0.0.1

linux.bogus
        origin = ns.linux.bogus
        mail addr = hostmaster.linux.bogus
        serial = 199802151
        refresh = 28800 (8 hours)
        retry   = 7200 (2 hours)
        expire  = 604800 (7 days)
        minimum ttl = 86400 (1 day)
linux.bogus     nameserver = ns.linux.bogus
linux.bogus     preference = 10, mail exchanger = mail.linux.bogus.linux.bogus
linux.bogus     preference = 20, mail exchanger = mail.friend.bogus
linux.bogus     nameserver = ns.linux.bogus
ns.linux.bogus  internet address = 192.168.196.2
mail.linux.bogus        internet address = 192.168.196.4
```

Als je goed kijkt zie je een bug. De regel:

```
linux.bogus     preference = 10, mail exchanger = mail.linux.bogus.linux.bogus
```

is verkeerd. Er zou moeten staan:

```
linux.bogus     preference = 10, mail exchanger = mail.linux.bogus
```

Ik heb expres deze fout gemaakt, zodat je ervan kan leren ;-). Als je in het zone bestand kijkt zie je dat de regel

```
                MX      10 mail.linux.bogus     ; Primary Mail Exchanger
```

een punt mist. Of er staat een ".linux.bogus"te veel. Als een machinenaam niet met een punt eindigt, dan wordt de oorsprong eraan toegevoegd. In dit geval wordt dat dus linux.bogus.linux.bogus. De regel moet zijn:

```
                MX      10 mail.linux.bogus.    ; Primary Mail Exchanger
```

of:

```
                MX      10 mail                 ; Primary Mail Exchanger
```

Ik geef de voorkeur aan de laatste regel, het is minder tikken. De meningen verschillen over wat beter zou zijn. In een zone bestand moet het domein voluit geschreven worden met een . erachter, of het domein moet er niet bijstaan. In het laatste geval wordt de oorsprong automatisch het domein.

Ik zeg hier nogmaals dat in het named.conf bestand *geen* punten achter domeinnamen moeten staan. Je hebt geen idee hoe vaak een simpele punt voor een hoop problemen heeft gezorgd.

Hier volgt het nieuwe zone bestand, nu met wat extra informatie:

```
;
; Zone file for linux.bogus
;
; The full zone file
;
@       IN      SOA     ns.linux.bogus. hostmaster.linux.bogus. (
                        199802151       ; serial, todays date + todays serial #
                        8H              ; refresh, seconds
                        2H              ; retry, seconds
                        1W              ; expire, seconds
                        1D )            ; minimum, seconds
;
                TXT     "Linux.Bogus, your DNS consultants"
                NS      ns              ; Inet Address of name server
                NS      ns.friend.bogus.
                MX      10 mail         ; Primary Mail Exchanger
                MX      20 mail.friend.bogus. ; Secondary Mail Exchanger

localhost       A       127.0.0.1

gw              A       192.168.196.1
                HINFO   "Cisco" "IOS"
                TXT     "The router"

ns              A       192.168.196.2
                MX      10 mail
                MX      20 mail.friend.bogus.
                HINFO   "Pentium" "Linux 2.0"
www             CNAME   ns

donald          A       192.168.196.3
                MX      10 mail
                MX      20 mail.friend.bogus.
                HINFO   "i486"  "Linux 2.0"
                TXT     "DEK"

mail            A       192.168.196.4
                MX      10 mail
                MX      20 mail.friend.bogus.
                HINFO   "386sx" "Linux 1.2"
```

```
ftp             A       192.168.196.5
                MX      10 mail
                MX      20 mail.friend.bogus.
                HINFO   "P6" "Linux 2.1.86"
```

Er zijn een aantal nieuwe RR's te zien. Het HINFO (Host INFOrmation) heeft twee gedeeltes. Het is een goede gewoonte om aanhalingstekens om beide gedeeltes te zetten. Het eerste deel beschrijft de hardware van de machine, en het tweede deel beschrijft de software. De machine 'ns' heeft een pentium en draait Linux 2.0. De tweede nieuwe RR is CNAME (Canonical NAME). Met CNAME kan je een alias voor een machine maken. In dit voorbeeld is www een alias voor ns.

Het gebruik van het CNAME record is een beetje controversieel. Hou je in elk geval aan de regel dat een MX, CNAME, of SOA record *nooit* naar een CNAME record mag verwijzen, het moet altijd naar een machine met een A record verwijzen. De volgende regel is dus fout:

```
foobar          CNAME   www                     ; NO!
```

Maar de volgende regel is goed:

```
foobar          CNAME   ns                      ; Yes!
```

Je kan er ook van uit gaan dat CNAME geen geldige machinenaam is voor een email adres, webmaster@www.linux.bogus is dus fout in dit voorbeeld. Als je dit probleem wilt voorkomen, gebruik dan een A record, zoals:

```
www             A       192.168.196.2
```

Sommige BIND experts adviseren het CNAME record helemaal niet te gebruiken. Maar het zou te ver gaan daar in deze HOWTO verder op in te gaan.

Maar zoals je ziet houdt deze HOWTO zich, net als een hoop domeinen, zich niet aan die regel.

Laat named z'n configuratiebestanden opnieuw inlezen met het commando ndc reload

```
$ nslookup
Default Server:  localhost
Address:  127.0.0.1

> ls -d linux.bogus
```

Bovenstaand commando zegt dat alle records weergegeven moeten worden. Dit resulteert in:

```
[localhost]
$ORIGIN linux.bogus.
@                       1D IN SOA       ns hostmaster (
                                        199802151       ; serial
                                        8H              ; refresh
                                        2H              ; retry
                                        1W              ; expiry
                                        1D )            ; minimum
```

```
                        1D IN NS        ns
                        1D IN NS        ns.friend.bogus.
                        1D IN TXT       "Linux.Bogus, your DNS consultants"
                        1D IN MX        10 mail
                        1D IN MX        20 mail.friend.bogus.
gw                      1D IN A         192.168.196.1
                        1D IN HINFO     "Cisco" "IOS"
                        1D IN TXT       "The router"
mail                    1D IN A         192.168.196.4
                        1D IN MX        10 mail
                        1D IN MX        20 mail.friend.bogus.
                        1D IN HINFO     "386sx" "Linux 1.0.9"
localhost               1D IN A         127.0.0.1
www                     1D IN CNAME     ns
donald                  1D IN A         192.168.196.3
                        1D IN MX        10 mail
                        1D IN MX        20 mail.friend.bogus.
                        1D IN HINFO     "i486" "Linux 1.2"
                        1D IN TXT       "DEK"
ftp                     1D IN A         192.168.196.5
                        1D IN MX        10 mail
                        1D IN MX        20 mail.friend.bogus.
                        1D IN HINFO     "P6" "Linux 1.3.59"
ns                      1D IN A         192.168.196.2
                        1D IN MX        10 mail
                        1D IN MX        20 mail.friend.bogus.
                        1D IN HINFO     "Pentium" "Linux 1.2"
```

Dat ziet er goed uit. Je ziet dat het lijkt op het zone bestand zelf. Laten we nu eens kijken wat er over www alleen gezegd wordt:

```
> set q=any
> www.linux.bogus.
Server:  localhost
Address:  127.0.0.1

www.linux.bogus canonical name = ns.linux.bogus
linux.bogus     nameserver = ns.linux.bogus
linux.bogus     nameserver = ns.friend.bogus
ns.linux.bogus  internet address = 192.168.196.2
```

Hierin zie je dat www.linux.bogus een alias is voor ns.linux.bogus, en het geeft voor programma's die een IP adres zoeken voldoende informatie over ns.linux.bogus om een verbinding met deze machine op te bouwen.

We zijn nu al halverwege!

4.3 De reverse zone

Nu kunnen programma's de namen in linux.bogus vertalen naar IP nummers waarmee ze een verbinding kunnen opbouwen. Maar wat ook nodig is, is een 'reverse zone', die DNS in staat stelt van een IP adres naar een machinenaam te converteren. Deze naam wordt door veel servers (WWW, FTP, IRC) gebruikt om te besluiten of je een verbinding mag opbouwen en welke prioriteit je krijgt. Als je volledig van het Internet gebruik wilt maken heb je een reverse zone nodig.

Plaats het volgende in named.conf:

```
zone "196.168.192.in-addr.arpa" {
        notify no;
        type master;
        file "pz/192.168.196";
};
```

Dit is precies hetzelfde als met 0.0.127.in-addr.arpa, en de inhoud lijkt er ook op:

```
@       IN      SOA     ns.linux.bogus. hostmaster.linux.bogus. (
                        199802151 ; Serial, todays date + todays serial
                        8H      ; Refresh
                        2H      ; Retry
                        1W      ; Expire
                        1D)     ; Minimum TTL
                NS      ns.linux.bogus.

1               PTR     gw.linux.bogus.
2               PTR     ns.linux.bogus.
3               PTR     donald.linux.bogus.
4               PTR     mail.linux.bogus.
5               PTR     ftp.linux.bogus.
```

Herstart named weer met ndc restart en controleer het resultaat met nslookup:

```
> 192.168.196.4
Server:  localhost
Address:  127.0.0.1

Name:    mail.linux.bogus
Address:  192.168.196.4
```

Het ziet er goed uit. Nu kunnen we alle informatie over het domein bekijken:

```
> ls -d 196.168.192.in-addr.arpa
[localhost]
$ORIGIN 196.168.192.in-addr.arpa.
@                       1D IN SOA       ns.linux.bogus. hostmaster.linux.bogus. (
                                        199802151       ; serial
                                        8H              ; refresh
                                        2H              ; retry
                                        1W              ; expiry
                                        1D )            ; minimum

                        1D IN NS        ns.linux.bogus.
1                       1D IN PTR       gw.linux.bogus.
```

```
2                       1D IN PTR       ns.linux.bogus.
3                       1D IN PTR       donald.linux.bogus.
4                       1D IN PTR       mail.linux.bogus.
5                       1D IN PTR       ftp.linux.bogus.
@                       1D IN SOA       ns.linux.bogus. hostmaster.linux.bogus. (
                                        199802151       ; serial
                                        8H              ; refresh
                                        2H              ; retry
                                        1W              ; expiry
                                        1D )            ; minimum
```

Ziet er ook goed uit! Als je andere informatie op je scherm krijgt, zoek dan naar foutmeldingen in de syslog. Hoe dat gaat staat uitgelegd in het begin van dit hoofdstuk.

4.4 Dingen waar je op moet letten

Er zijn een paar dingen waar je goed op moet letten. Ten eerste, de IP nummers die in de voorbeelden gebruikt worden vallen binnen de verzameling "prive adressen"; deze adressen mogen niet op het Internet gebruikt worden, en zijn derhalve uitermate geschikt om te gebruiken in de voorbeelden in deze HOWTO. De tweede opmerking gaat over de notify no; regel. Deze regel vertelt named dat de secondary name servers niet op de hoogte gesteld moeten worden van de veranderingen. Als we dat wel zouden doen bij deze experimenten, dan zouden we daarmee alleen maar het Internet vervuilen.

En natuurlijk is het door ons gemaakte domein nep, net als de IP adressen. Voor een voorbeeld van een echt domein, zie het volgende hoofdstuk.

4.5 Waarom reverse lookups niet werken

Er zijn een aantal dingen die bij normale machinenaam lookups wel werken, en bij reverse lookups niet. Hieronder staat een uitleg en oplossing van deze zaken, maar eerst moet je zorgen dat de reverse lookups op je eigen machine werken. Als dat nog niet zo is, repareer dat en lees daarna verder.

Er zullen twee problemen met reverse lokups worden besproken:

4.5.1 De reverse zone is niet gemachtigd

Als je van een ISP een reeks IP adressen krijgt en een domein naam, dan is dat domein gewoonlijk gemachtigd nameserver informatie te verstekken aan computers die daarom vragen. Die machtiging bestaat uit een NS record dat vertelt dat de nameserver informatie over jouw domein op jouw nameserver gevonden kan worden. Deze methode van recursief machinenamen opzoeken staat hierboven beschreven, in de sectie "droge theorie".

De reverse zone moet ook gemachtigd worden. Als je de 192.168.196 netwerkadressen en het linux.bogus domein hebt gekregen, dan moet je provider ook NS records hebben voor je reverse zone en je forward zone. Als je de boom langsloopt van in-addr.arpa en bij je eigen domein probeert te komen, bestaat de kans dat er ergens iets niet werkt. Dat zal dan waarschijnlijk bij je ISP zijn. Neem in dat geval contact met je ISP op zodat ze een goed NS record aan kunnen maken.

4.5.2 Subnetten die niet in de A, B, of C class vallen.

Door gebrek aan IP nummers worden er regelmatig subnets uitgedeeld die kleiner zijn dan een class C netwerk (256 adressen). Zo'n netwerk heet een classless netwerk. Deze kleine subnets maken het mogelijk dat het Internet nog steeds draait. Bij subnets kleiner dan een class C netwerk, kunnen er problemen optreden. Deze problemen met de oplossingen staan beschreven bij *Ask Mr. DNS, op http://www.acmebw.com/askmrdns/00007.htm.*

Het probleem ligt iets te ingewikkeld om hier te behandelen, dus lees de 'Mr. DNS' pagina goed door, en zorg dat je het begrijpt. Het kan gebeuren dat je ISP het probleem niet begrijpt. Dan zal je het dus moeten uitleggen, zodat ze toch een goede configuratie kunnen neerzetten die je vervolgens met nslookup kan testen.

De Mr. DNS uitleg van het probleem bespreekt o.a. een CNAME truuk en een truuk met het zone bestand. Niet elke resolver begrijpt deze CNAME truuk echter. Als je daar problemen mee krijgt, vraag je provider dan een PTR record rechtstreeks in het getruukte zone bestand te zetten in plaats van de CNAME truuk te gebruiken. Sommige ISP's hebben andere oplossingen voor dit probleem zoals een websysteem voor het bewerken van 'reverse mappings'.

5 Een voorbeeld van een echt domein

In dit hoofdstuk laten we *eindelijk* wat echte zone bestanden zien.

Gebruikers van deze HOWTO vroegen om voorbeelden van een echt zone bestand. Het bestand dat hier gebruikt wordt is gemaakt door David Bullock van LAND-5, met wat aanpassingen van mezelf. Wat je hier ziet kan dus iets verschillen van de informatie die je krijgt als je met nslookup de LAND-5 nameservers ondervraagt.

5.1 /etc/named.conf (of /var/named/named.conf)

Hieronder staat de zone informatie voor LAND-5's subnet, 206.6.177, en de 'reverse zones'. Merk op dat de bestanden niet in een directory pz geplaatst worden zoals dat in deze HOWTO gebeurt, maar dat de directory zone heet.

```
// Boot file for LAND-5 name server

options {
        directory "/var/named";
};

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

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

```
zone "land-5.com" {
        type master;
        file "zone/land-5.com";
};

zone "177.6.206.in-addr.arpa" {
        type master;
        file "zone/206.6.177";
};
```

Als je dit in je named.conf bestand plaats, zet als *alsjeblieft* ‘notify no;’ in de zone secties voor de twee LAND-5 zones om ongelukken te voorkomen.

5.2 /var/named/root.hints

Het bestand hieronder is inmiddels verouderd. Je zal dus zelf een nieuwe versie aan moeten maken. Het aanmaken van een root.hints bestand wordt verderop in deze HOWTO beschreven.

```
; <<>> DiG 8.1 <<>> @A.ROOT-SERVERS.NET.
; (1 server found)
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10
;; flags: qr aa rd; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 13
;; QUERY SECTION:
;;      ., type = NS, class = IN

;; ANSWER SECTION:
.                       6D IN NS        G.ROOT-SERVERS.NET.
.                       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.
.                       6D IN NS        A.ROOT-SERVERS.NET.
.                       6D IN NS        H.ROOT-SERVERS.NET.
.                       6D IN NS        B.ROOT-SERVERS.NET.
.                       6D IN NS        C.ROOT-SERVERS.NET.
.                       6D IN NS        D.ROOT-SERVERS.NET.
.                       6D IN NS        E.ROOT-SERVERS.NET.
.                       6D IN NS        I.ROOT-SERVERS.NET.
.                       6D IN NS        F.ROOT-SERVERS.NET.

;; ADDITIONAL SECTION:
G.ROOT-SERVERS.NET.     5w6d16h IN A    192.112.36.4
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
```

```
A.ROOT-SERVERS.NET.     5w6d16h IN A    198.41.0.4
H.ROOT-SERVERS.NET.     5w6d16h IN A    128.63.2.53
B.ROOT-SERVERS.NET.     5w6d16h IN A    128.9.0.107
C.ROOT-SERVERS.NET.     5w6d16h IN A    192.33.4.12
D.ROOT-SERVERS.NET.     5w6d16h IN A    128.8.10.90
E.ROOT-SERVERS.NET.     5w6d16h IN A    192.203.230.10
I.ROOT-SERVERS.NET.     5w6d16h IN A    192.36.148.17
F.ROOT-SERVERS.NET.     5w6d16h IN A    192.5.5.241

;; Total query time: 215 msec
;; FROM: roke.uio.no to SERVER: A.ROOT-SERVERS.NET.  198.41.0.4
;; WHEN: Sun Feb 15 01:22:51 1998
;; MSG SIZE  sent: 17  rcvd: 436
```

5.3 /var/named/zone/127.0.0

In dit bestand staat niet bijzonders, alleen het verplichte SOA record en een record dat 127.0.0.1 laat verwijzen naar de lokale machine. Ook dit laatste is verplicht. In dit bestand hoeft verder niets te staan. Ook zal het nooit aangepast hoeven worden, tenzij het adres van je nameserver of hostmaster verandert.

```
@               IN      SOA     land-5.com. root.land-5.com. (
                                199609203       ; Serial
                                28800   ; Refresh
                                7200    ; Retry
                                604800  ; Expire
                                86400)  ; Minimum TTL
                        NS      land-5.com.

1                       PTR     localhost.
```

5.4 /var/named/zone/land-5.com

In dit bestand zien we het verplichte SOA record en de benodigde NS records. We kunnen zien dat er een secundaire nameserver aanwezig is op ns2.psi.net. Dit is ook zoals het zou moeten zijn; neem altijd een secundaire nameserver in een ander netwerk. Als je netwerk platligt is je secundaire DNS server nog steeds beschikbaar. We zien ook dat er een centrale server is genaamd land-5, die verschillende Internet servers draait. Naar deze machine wordt verwezen met CNAME records, dit zou je ook met A records kunnen doen.

Zoals je in het SOA record kan zien, is de oorsprong van dit zone bestand land-5.com en is de contactpersoon root@land-5.com. In plaats van root wordt ook het adres hostmaster vaak hiervoor gebruikt. Het serienummer is in het JJJJMMDDS formaat: de datum en het serienummer van de dag. Dit bestand is waarschijnlijk de zesde versie van 9 september 1996. Onthoud dat het serienummer altijd hoger moet zijn dat het vorige serienummer. Hier wordt slechts 1 cijfer gebruikt voor het serienummer van de dag. Als er op een dag 9 versies worden gemaakt, moet hij weer wachten tot de volgende dag voordat nieuwe versies kunnen worden gemaakt. Om dit te voorkomen zou je twee getallen kunnen gebruiken voor het serienummer.

```
@       IN      SOA     land-5.com. root.land-5.com. (
                        199609206       ; serial, todays date + todays serial #
                        8H              ; refresh, seconds
                        2H              ; retry, seconds
                        1W              ; expire, seconds
                        1D )            ; minimum, seconds
                NS      land-5.com.
                NS      ns2.psi.net.
                MX      10 land-5.com.  ; Primary Mail Exchanger
                TXT     "LAND-5 Corporation"

localhost       A       127.0.0.1

router          A       206.6.177.1

land-5.com.     A       206.6.177.2
ns              A       206.6.177.3
www             A       207.159.141.192

ftp             CNAME   land-5.com.
mail            CNAME   land-5.com.
news            CNAME   land-5.com.

funn            A       206.6.177.2

;
;       Workstations
;
ws-177200       A       206.6.177.200
                MX      10 land-5.com.   ; Primary Mail Host
ws-177201       A       206.6.177.201
                MX      10 land-5.com.   ; Primary Mail Host
ws-177202       A       206.6.177.202
                MX      10 land-5.com.   ; Primary Mail Host
ws-177203       A       206.6.177.203
                MX      10 land-5.com.   ; Primary Mail Host
ws-177204       A       206.6.177.204
                MX      10 land-5.com.   ; Primary Mail Host
ws-177205       A       206.6.177.205
                MX      10 land-5.com.   ; Primary Mail Host
; {Many repetitive definitions deleted - SNIP}
ws-177250       A       206.6.177.250
                MX      10 land-5.com.   ; Primary Mail Host
ws-177251       A       206.6.177.251
                MX      10 land-5.com.   ; Primary Mail Host
ws-177252       A       206.6.177.252
                MX      10 land-5.com.   ; Primary Mail Host
ws-177253       A       206.6.177.253
                MX      10 land-5.com.   ; Primary Mail Host
ws-177254       A       206.6.177.254
```

```
                MX      10 land-5.com.   ; Primary Mail Host
```

Als je goed naar de land-5 nameserver kijkt, zie je dat de namen van de vorm ws_*nummer* zijn. In nieuwere versies mag het underscoreteken niet gebruikt worden, daarom is dat hier vervangen door een minteken.

Iets anders dat het opmerken waard is, is dat de werkstations geen individuele namen hebben, maar dat de laatste twee getallen van het IP adres gebruikt worden om unieke namen te maken. Dit vereenvoudigt de administratie, maar is een beetje onpersoonlijk. Het kan zijn dat gebruikers dit soort namen niet op prijs stellen.

We zien ook dat funn.land-5.com een alias is voor land-5.com, maar hier wordt gebruik gemaakt van een A record in plaats van een CNAME record. Zoals eerder opgemerkt is, is dat een goed gebruik.

5.5 /var/named/zone/206.6.177

Onderstaand bestand zal in deze sectie besproken worden.

```
@               IN      SOA     land-5.com. root.land-5.com. (
                                199609206       ; Serial
                                28800   ; Refresh
                                7200    ; Retry
                                604800  ; Expire
                                86400)  ; Minimum TTL
                        NS      land-5.com.
                        NS      ns2.psi.net.
;
;       Servers
;
1       PTR     router.land-5.com.
2       PTR     land-5.com.
2       PTR     funn.land-5.com.
;
;       Workstations
;
200     PTR     ws-177200.land-5.com.
201     PTR     ws-177201.land-5.com.
202     PTR     ws-177202.land-5.com.
203     PTR     ws-177203.land-5.com.
204     PTR     ws-177204.land-5.com.
205     PTR     ws-177205.land-5.com.
; {Many repetitive definitions deleted - SNIP}
250     PTR     ws-177250.land-5.com.
251     PTR     ws-177251.land-5.com.
252     PTR     ws-177252.land-5.com.
253     PTR     ws-177253.land-5.com.
254     PTR     ws-177254.land-5.com.
```

De reverse zone is het deel van de DNS configuratie dat het meeste problemen oplevert. Het wordt gebruikt om erachter te komen welke machinenaam bij een IP adres hoort.

Voorbeeld: Stel dat een Noorse IRC server alleen verbindingen wil accepteren van Noorse IRC clients. De irc server kan er makkelijk achter komen wat het IP adres van de client is. Dit IP adres zit namelijk in elk IP pakket dat de client stuurt. Nu wil de server weten welke machinenaam bij dit IP adres hoort. De server roept dan de functie gethostbyaddr aan. Deze functie zoekt vervolgens 200.177.6.206.in-addr.arpa op door te beginnen bij een . server, en steeds een niveau verder te gaan. Deze methode staat beschreven in de sectie "droge theorie". De nameserver voor 177.6.206.in-addr.arpa heeft een 'PTR ws-177200.land-5.com' record voor 200.177.6.206.in-addr.arpa, wat betekent dat bij 206.6.177.200 de machinenaam ws-177200.land-5.com hoort. Zoals eerder gezegd klopt deze uitleg niet helemaal, maar het komt aardig in de buurt van de werkelijkheid.

Even terug naar het voorbeeld van de IRC server. De IRC server accepteert alleen connecties van Noorse clients. Door de reverse lookup weet de server dat het verzoek afkomstig is van ws-177200.land-5.com. Omdat alleen verbindingen vanaf .no machines zijn toegestaan, zal de verbinding in dit geval niet opgebouwd worden. Als de reverse lookup zou falen, zou er helemaal geen hostname gevonden worden, en wordt er ook geen verbinding gemaakt.

Sommige mensen zeggen dat reverse lookups alleen belangrijk zijn voor servers, of dat ze helemaal niet belangrijk zijn. Dat is niet waar. Sommige ftp, news, IRC of zelfs HTTP servers accepteren geen connecties van een machine als er geen machinenaam gevonden kan worden. Reverse lookups zijn dus zelfs verplicht.

6 Onderhoud

Alles aan de praat houden

Er is een onderhoudstaak aan DNS servers, namelijk het up-to-date houden van het root.hints bestand. De makkelijkste manier om dat te doen is met het programma dig. Draai dig zonder opties, en je krijgt het root.hints bestand van je eigen server. Als je een rootserver ondervraagt met dig, is het resultaat te gebruiken als root.hints bestand. Het commando "dig @e.root-servers.net . ns >root.hints.new)"levert een nieuw hints bestand. Hernoem het bestand root.hints.new naar root.hints en laat named het bestand opnieuw inlezen met ndc reload.

Al Longyear heeft me onderstaand script gemaild, dat gebruikt kan worden om root.hints automatisch te laten updaten. Het script gaat er van uit dat mail lokaal werkt en dat het adres hostmaster op de lokale machine bestaat. Pas het script eventueel aan zodat het werkt met de configuratie van jouw machine.

```
#!/bin/sh
#
# Update the nameserver cache information file once per month.
# This is run automatically by a cron entry.
#
# Original by Al Longyear
# Updated for bind 8 by Nicolai Langfeldt
# Miscelanious error-conditions reported by David A. Ranch
# Ping test suggested by Martin Foster
#
(
 echo "To: hostmaster <hostmaster>"
```

```
 echo "From: system <root>"
 echo "Subject: Automatic update of the root.hints file"
 echo

 PATH=/sbin:/usr/sbin:/bin:/usr/bin:
 export PATH
 cd /var/named

 # Are we online?  Ping a server at your ISP
 case `ping -qnc 10 some.machine.net` in
   *'100% packet loss'*)
        echo "The network is DOWN. root.hints NOT updated"
        echo
        exit 0
        ;;
 esac

 dig @rs.internic.net . ns >root.hints.new 2>&1

 case `cat root.hints.new` in
   *NOERROR*)
        # It worked
        :;;
   *)
        echo "The root.hints file update has FAILED."
        echo "This is the dig output reported:"
        echo
        cat root.hints.new
        exit 0
        ;;
 esac

 echo "The root.hints file has been updated to contain the following information:"
 echo
 cat root.hints.new

 chown root.root root.hints.new
 chmod 444 root.hints.new
 rm -f root.hints.old
 mv root.hints root.hints.old
 mv root.hints.new root.hints
 ndc restart
 echo
 echo "The nameserver has been restarted to ensure that the update is complete."
 echo "The previous root.hints file is now called
/var/named/root.hints.old."
) 2>&1 | /usr/lib/sendmail -t
exit 0
```

Misschien heb je al gezien dat het root.hints bestand ook via ftp beschikbaar is via Internic. Gebruik geen ftp om dit bestand op te halen. Bovenstaand script is beter voor

Internic en voor het Internet als geheel.

7 Converteren van BIND 4 naar BIND 8

Oorspronkelijk was dit een sectie over het gebruik van BIND 8, door David E. Smith (dave@bureau42.ml.org). Het is iets aangepast, zodat de sectie beter bij bovenstaande titel past.

Eigenlijk is het heel eenvoudig. Behalve het gebruik van named.conf in plaats van named.boot is er weinig veranderd. Bij BIND 8 zit een perl script dat de oude naar de nieuwe configuratiebestanden converteert. Een voorbeeld van een named.boot (van BIND 4) bestand voor een caching-only nameserver:

```
directory /var/named
cache   .                                                root.hints
primary 0.0.127.IN-ADDR.ARPA                             127.0.0.zone
primary localhost                                        localhost.zone
```

Ergens op je systeem staat het named-bootconf.pl script. In de BIND8 source distributie staat dit in: bind8/src/bin/named. Om het bestand te converteren, tik:

```
./named-bootconf.pl < named.boot > named.conf
```

Hiermee wordt named.conf aangemaakt:

```
// generated by named-bootconf.pl

options {
        directory "/var/named";
};

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

zone "0.0.127.IN-ADDR.ARPA" {
        type master;
        file "127.0.0.zone";
};

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

Het script converteert alles dat in een named.boot bestand kan staan, maar voegt er niet alle verbeteringen aan toe die in BIND8 te gebruiken zijn. Hieronder volgt een complete named.conf die hetzelfde doet, maar wat efficiënter werkt:

```
// This is a configuration file for named (from BIND 8.1 or later).
// It would normally be installed as /etc/named.conf.
// The only change made from the ‘stock’ named.conf (aside from this
// comment :) is that the directory line was uncommented, since I
// already had the zone files in /var/named.

options {
        directory "/var/named";
        datasize 20M;
};

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

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

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

Dit bestand kan je vinden in bind8/src/bin/named/test van de BIND8 distributie, samen met zone bestanden die voor de meeste mensen zonder aanpassing te gebruiken zijn.

Het formaat van zone bestanden en het root.hints bestand zijn identiek, net als de commando’s die gebruikt worden om deze bestanden bij te werken.

8 Vragen en antwoorden (Q & A)

Lees deze sectie voordat je me een mailtje stuurt.

1. Mijn named vraagt om een named.boot bestand.

   Je leest de verkeerde HOWTO. Lees de DNS howto die over BIND 4 gaat, en te vinden is op *http://www.math.uio.no/˜janl/DNS/*

2. Hoe gebruik ik DNS binnen een firewall?

   Een hint: gebruik forward only;, je hebt waarschijnlijk ook de regel:

```
query-source port 53;
```

   nodig in het öptions"gedeelte van je named.conf bestand, zoals beschreven staat in de sectie ’3 (caching)’.

3. Hoe zorg ik ervoor dat DNS verschillende nummers voor dezelfde machinenaam (bijvoorbeeld www.drukke.site) uitdeelt, zodat ik de load over meerdere machines kan uitspreiden?

   Maak meerdere A records voor www.drukke.site en gebruik BIND 4.9.3 of een nieuwere versie. BIND zal dan de adressen "roteren". Deze truuk werkt niet met eerdere versies van BIND.

4. Ik wil DNS gebruiken op een (gesloten) intranet. Hoe doe ik dat?

   Maak geen gebruik van het root.hints bestand, en gebruik alleen de zone bestanden. Op deze manier hoef je ook niet steeds nieuwe root.hints bestanden op te halen.

5. Hoe configureer ik een secundaire nameserver?

   Als de primaire nameserver adres 127.0.0.1 heeft, zet dan het volgende in het named.conf bestand van je secundaire nameserver:

```
zone "linux.bogus" {
      type slave;
      file "sz/linux.bogus";
      masters { 127.0.0.1; };
};
```

   Eventueel kan je meerdere machines opgeven in de masters regel. De adressen van de machines moeten gescheiden worden met een ';'.

6. Ik wil BIND kunnen gebruiken zonder Internet verbinding

   Er zijn drie antwoorden op deze vraag:

7. Ik heb onderstaand mailtje ontvangen van Ian Clark <ic@deakin.edu.au>, waarin hij uitlegt hoe hij dat doet:

```
Ik draai zelf named op m'n masquerading machine. Ik heb twee <tt/root.hints/
bestanden, <tt/root.hints.real/ met de echte rootnameserver informatie en
<tt/root.hints.fake/ met de volgende inhoud:

----
; root.hints.fake
; this file contains no information
----

Als ik de internet verbinding verbreek zet ik de inhoud van
<tt/root.hints.fake/ in <tt/root.hints/ en herstart ik named. Als ik weer
verbinding maak, zet ik de inhoud van <tt/root.hints.real/ in <tt/root.hints/
en herstart ik named. Deze commando's staan respectievelijk in m'n
<tt/ip-down/ en <tt/ip-up/ scripts.

De eerste keer dat ik een IP adres opzoek als ik offline gegaan ben, komt de
volgende regel in /var/log/messages te staan:

Jan 28 20:10:11 hazchem named[10147]: No root nameserver for class IN

En daar valt mee te leven.
```

```
Deze methode werkt goed bij mij. Ik kan de nameserver voor lokale adressen
gebruiken als ik off-line ben zonder de lange timeouts voor adressen buiten
het lokale netwerk. En als ik online ben kunnen adressen buiten het lokale
netwerk gewoon gevonden worden.
```

8. Ik heb ook een mailtje gekregen van Karl-Max Wanger over hoe BIND samenwerkt met NFS en de portmapper op een voornamelijk offline machine:

```
Ik draai BIND op al m'n machines die alleen af en toe verbonden zijn met het
Internet via een modem. De nameserver werkt alleen als cache, het bevat zelf
geen gegevens die niet ergens anders vandaan gehaald zijn. Zoals gebruikelijk
in Slackware startte named voor nfsd en mountd.

Met één van m'n machines had ik het probleem dat ik hem soms kon
mounten vanaf een ander systeem, maar meestal niet. Het maakte geen verschil
of ik nu via PLIP, ethernet, of SLIP verbinding had gemaakt.

Na een tijdje experimenteren kwam ik erachter dat named een conflict
veroorzaakte met het communicatieproces tussen nsfd en de portmapper en tussen
mountd en de portmapper. Als ik named na nfsd en mountd startte, was er geen
probleem.

Omdat deze volgorde geen nadelen heeft, raad ik iedereen aan in deze volgorde
te booten, om zo problemen te voorkomen.
```

9. Tenslotte meld ik dat er een HOWTO is over dit onderwerp, bij *Ask Mr. DNS op http://www.acmebw.com/askmrdns/#linux-ns*. Het gaat echter wel over BIND 4, dus je zal op sommige plekken de informatie moeten aanpassen voor BIND 8.

10. Waar plaatst een caching nameserver z'n cache? En kan ik de grootte van de cache aanpassen?

    De cache staat volledig in het geheugen, het wordt nooit naar disk geschreven. Elke keer dat je named stopt gaat de cache verloren. De grootte van de cache is niet aan te passen. Als je dit toch wilt, zal je named moeten hacken. Dit is niet aan te raden.

11. Hoe krijg ik een domein? Als ik bijvoorbeeld het domein linux-rules.net wil hebben, hoe zorg ik dan dat het aan mij toegewezen wordt?

    Neem contact op met je ISP, zij zullen je hiermee kunnen helpen. Hou er wel rekening mee dat je zal moeten betalen voor het domein.

9 Hoe wordt ik een nog betere DNS beheerder

Documentatie en tools

Er bestaat ook Echte Documentatie over dit onderwerp, online en in boeken. Je zal dit zeker moeten lezen wil je meer verstand krijgen van DNS. Het standaard boek over DNS is *DNS and BIND* door C. Liu and P. Albitz van de uitgeverij O'Reilly & Associates. ISBN 0-937175-82-X. Ik heb het gelezen, en het is erg goed. Het is gebaseerd op BIND4, maar dat is niet echt een probleem. Er is ook een sectie over DNS in het boek *TCP/IP Network Administration*, ISBN 0-937175-82-X, van Craig Hunt.

Online kan je informatie vinden op <http://www.dns.net/dnsrd/> (DNS Resources Directory) en <http://www.isc.org/bind.html>; een FAQ, een naslagwerk, definities van protocollen en DNS hacks (deze zijn ook, samen met de relevante protocollen die hieronder genoemd staan bij de BIND distributie ingesloten). Ik heb het grootste deel hiervan niet gelezen, maar ik ben dan ook geen ervaren DNS beheerder. Arnt Gulbrandsen is dat wel, en hij vindt ze fantastisch ;-). De nieuwsgroep *comp.protocols.tcp-ip.domains* gaat over DNS. Verder zijn er een aantal RFC's over DNS, waarvan de belangrijkste zijn:

RFC 2052

A. Gulbrandsen, P. Vixie, *A DNS RR for specifying the location of services (DNS SRV)*, October 1996

RFC 1918

Y. Rekhter, R. Moskowitz, D. Karrenberg, G. de Groot, E. Lear, *Address Allocation for Private Internets*, 02/29/1996.

RFC 1912

D. Barr, *Common DNS Operational and Configuration Errors*, 02/28/1996.

RFC 1912 Errors

B. Barr *Errors in RFC 1912*, this is available at <http://www.cis.ohio-state.edu/~barr/rfc1912-errors.html>

RFC 1713

A. Romao, *Tools for DNS debugging*, 11/03/1994.

RFC 1712

C. Farrell, M. Schulze, S. Pleitner, D. Baldoni, *DNS Encoding of Geographical Location*, 11/01/1994.

RFC 1183

R. Ullmann, P. Mockapetris, L. Mamakos, C. Everhart, *New DNS RR Definitions*, 10/08/1990.

RFC 1035

P. Mockapetris, *Domain names - implementation and specification*, 11/01/1987.

RFC 1034

P. Mockapetris, *Domain names - concepts and facilities*, 11/01/1987.

RFC 1033

M. Lottor, *Domain administrators operations guide*, 11/01/1987.

RFC 1032

M. Stahl, *Domain administrators guide*, 11/01/1987.

RFC 974

C. Partridge, *Mail routing and the domain system*, 01/01/1986.