DNS HOWTO

Nicolai Langfeldt (janl@linpro.no), Jamie Norrish and others Version 3.1, 2001-01-18

Kako postati mali upravitelj DNS. V slovenščino prevedel Andrej Lajovic andrej.lajovic@guest.arnes.si, 7. avgusta 2000. Prevod posodobil in dopolnil Rok Papež rok.papez@lugos.si, 8. aprila 2002.

Kazalo

1 Predgovor

Ključni izrazi: DNS, BIND, BIND 4, BIND 8, named, povezava na klic, PPP, slip, ISDN, internet, domena, ime, ločljivost, računalniki, predpomnenje.

Ta dokument je del projekta Linux Documentation Project.

1.1 Pravne zadevščine

V slovenščino prevedel Andrej Lajovic `andrej.lajovic@guest.arnes.si`, 7. avgusta 2000. Za razširjanje slovenskega prevoda velja enako kakor za izvirnik.

Prevod malce posodobil in dopolnil Rok Papež `rok.papez@lugos.si`, 8. aprila 2002.

1.2 Zahvala in prošnja za pomoč

Rad bi se zahvalil Arntu Gulbrandsenu, ki se je pregrizel skozi vse delovne različice tega dokumenta in mi pomagal s številnimi uporabnimi predlogi. Rad bi se zahvalil tudi mnogim, ki so mi predloge in opozorila poslali po elektronski pošti.

To, kar prebirate, je dokument v trajnem razvoju, zato vas prosim, da me obveščate o morebitnih uspehih in težavah. Tako mi boste pomagali izboljševati ta HOWTO. Komentarje in/ali vprašanja ter denarne prispevke pošiljajte na janl@math.uio.no. Lahko pa kupite tudi mojo knjigo o DNS. O njej si preberite v bibliografskem razdelku. Preden mi pošljete sporočilo, na katerega želite odgovor, preverite, ali je vaš naslov, naveden v pismu, *zagotovo* pravilen in delujoč. Preden mi pišete, si preberite poglavje 10 (Vprašanja in odgovori). Pa še tole: razumem le norveško in angleško.

To je HOWTO. Vzdržujem ga kot del LDP že od leta 1995. Leta 2000 sem napisal knjigo na isto temo. Rad bi poudaril, da ta HOWTO, čeprav je v marsičem podoben knjigi, *ni* le razvodenela različica knjige, ki naj bi promovirala knjižno izdajo. Knjigo najdete navedeno v bibliografskem razdelku na koncu HOWTO. Bralci tega HOWTO so mi pomagali uvideti, kaj je pri DNS težko doumljivo. To mi je pomagalo pri pisanju knjige, pri tem početju pa sem obenem uvidel, kaj bi bilo nujno za ta HOWTO. HOWTO je torej pomagal oblikovati knjigo, knjiga pa je sooblikovala različico 3 tega HOWTO. Zahvalo dolgujem tudi založniku knjige, Que, da se je sploh spustil v tveganje z menoj :-)

1.3 Posvetilo

Ta HOWTO posvečam Anne Line Norheim Langfeldt, čeprav ga verjetno ne bo nikoli prebrala - pač ni take vrste dekle.

2 Uvod

Kaj to je in kaj ni

DNS (Domain Name System) preslikuje imena računalnikov v številke IP, ki jih imajo vsi računalniki v internetu. Preslikuje imena v naslove, naslove v imena in še nekaj drugih stvari. Ta HOWTO opisuje, kako ustvariti tak preslikovalni imenik v sistemu Unix z nekaterimi značilnostmi Linuxa.

Preslikava je preprosta povezava med dvema rečema, v tem primeru med imenom računalnika, na primer `ftp.linux.org`, in njegovo številko IP (oz. naslovom) `199.249.150.4`. DNS zmore tudi obratne preslikave, iz številke IP v ime računalnika; temu se reče angleško "reverse mapping".

DNS je za nepoučene (vas ;-) eno bolj nejasnih področij upravljanja omrežij. Na srečo pa ni tako zelo zapleten. Ta HOWTO poskuša pojasniti nekatere stvari. Opisuje, kako postaviti *preprost* imenski strežnik DNS. Začeli bomo s predpomnilnim imenskim strežnikom in prešli na postavitev primarnega strežnika DNS. Kar zadeva bolj zapletene postavitve, si lahko ogledate poglavje 10 (Vprašanja in odgovori). Če niso opisane tam, boste morali *prebrati* nekaj prave dokumentacije. Kaj sestavlja pravo dokumentacijo, sem navedel v poglavju 11 (Zadnje poglavje).

Preden začnete, naj bo vaš računalnik nastavljen tako, da se lahko telnetate vanj in iz njega ter uspešno izvajate različne povezave v omrežje. Še posebej pomembno je, da lahko izvedete ukaz `telnet 127.0.0.1` in dobite svoj računalnik (preizkusite zdaj!). Za začetek potrebujete tudi ustrezno nastavljene datoteke `/etc/nsswitch.conf`, `/etc/resolv.conf` in `/etc/hosts`, ker njihovih funkcij tu ne bom razlagal. Če računalnika še nimate ustrezno nastavljenega, si oglejte Networking-HOWTO in/ali Networking-Overview-HOWTO, kjer je razloženo, kako se vse to vzpostavi. Preberite si to.

Ko rečem 'vaš računalnik', mislim računalnik, na katerem postavljate DNS, in ne kak drug računalnik, ki sodeluje pri vaših omrežnih podvigih.

Predpostavljam, da niste za takim ali drugačnim požarnim zidom, ki bi preprečeval imenske poizvedbe. Če ste, boste potrebovali posebno konfiguracijo — poglejte v poglavje 10 (Vprašanja in odgovori).

Strežnik za imenske poizvedbe je v Unixu program `named`. Je del paketa “BIND”, ki ga vzdržuje The Internet Software Consortium. `Named` je del večje distribucije Linuxa in je najpogosteje nameščen v sklopu paketa `BIND` kot `/usr/sbin/named`.

Če imate named, ga verjetno lahko uporabite; drugače pa lahko prevedeni program dobite z Linuxove strani ftp ali si pretočite zadnjo različico izvirne kode z `<ftp://ftp.isc.org/isc/bind/src/>`. Ta HOWTO govori o različici bind 8. Starejša različica tega HOWTO, ki velja za bind 4, je še vedno dosegljiva na `<http://www.math.uio.no/`

~janl/DNS/>. Če vaš priročnik named (man) govori (na koncu, v razdelku FILES) o `named.conf`, imate bind 8, če govori o `named.boot`, pa imate bind 4. Če imate različico 4 in se vam zdi varnost pomembna, bi morali vsekakor nadgraditi na različico 8.

DNS je čez ves internet segajoča zbirka. Pazite, kaj vpišete vanjo. Če boste namreč vanjo vpisovali neumnosti, boste (in to velja tudi za druge) iz nje tudi dobili neumnosti. Vzdržujte svoj DNS urejen in pregleden, da vam bo dobro služil. Naučite se ga uporabljati, upravljati, odpravljati napake in bodite še en v vrsti dobrih upraviteljev, ki skrbijo, da internet zgledno deluje.

Nasvet: Naredite si rezervne kopije vseh datotek, ki jih boste spreminjali - če kaj ne bo delovalo, lahko sistem še vedno vrnete v staro, delujoče stanje.

3 Predpomnilni imenski strežnik

Prvi korak pri nastavljanju DNS, zelo uporaben za uporabnike klicnega dostopa

V Red Hat in sorodnih distribucijah lahko dosežete enak učinek kakor z napotki v prvih odsekih tega HOWTO z namestitvijo paketov `bind`, `bind-utils` in `caching-nameserver`. Če uporabljate Debian, preprosto namestite paketa `bind` in `bind-doc`. Seveda se z namestitvijo teh paketov ne boste naučili toliko kakor ob branju tega HOWTO, zato namestite pakete in ob prebiranju tega razdelka preverite nameščene datoteke.

Predpomnilni imenski strežnik poišče odgovor na imensko poizvedbo in si jo zapomni za naslednjič. S tem se znatno skrajša čakalni čas ob prihodnji poizvedbi, še posebej, če imate počasno povezavo.

Najprej potrebujete datoteko `/etc/named.conf` (Debian: `/etc/bind/named.conf`). Ta se prebere, ko se named zažene, in naj za zdaj vsebuje le:

```
// Konfiguracijska datoteka za predpomnilni imenski strežnik

options {
        directory "/var/named";

        // Odkomentiranje tega lahko pomaga, če ste za požarnim zidom in
        // stvari ne delujejo, kakor bi morale. Kljub temu se pogovorite
        // s svojim upraviteljem požarnega zidu.

        // query-source port 53;
};

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

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

Paketi distribucij Linuxa lahko uporabijo drugačna imena datotek; kljub temu imajo približno enake nastavitve.

Vrstica ‘`directory`’ pove named, kje naj pogleda za datoteke. Vse datoteke, navedene zatem, so spravljene relativno na to mesto. Imenik `pz` je v `/var/named` in je torej v resnici `/var/named/pz`. `/var/named` je pravilni imenik glede na datotečni hierarhični standard *Linux File system Standard*.

V `named.conf` je omenjena tudi datoteka `/var/named/root.hints`. Ta naj vsebuje: (*Pri kopiranju vsebine datoteke prek odlagališča iz elektronske različice tega dokumenta preverite, da v datoteki* **ni** *začetnih presledkov, tj. vse vrstice se morajo začeti z nepraznim znakom. Nekatero programje za urejanje besedil vnese na začetke vrstic presledke in povzroči zmedo. V tem primeru odstranite začetne presledke.*)

```
;
; Če to datoteko že imate, so lahko tu komentarji.
; Če je še nimate, naj vas to ne skrbi.
;
.                        6D IN NS       M.ROOT-SERVERS.NET.
.                        6D IN NS       I.ROOT-SERVERS.NET.
.                        6D IN NS       E.ROOT-SERVERS.NET.
.                        6D IN NS       D.ROOT-SERVERS.NET.
.                        6D IN NS       A.ROOT-SERVERS.NET.
.                        6D IN NS       H.ROOT-SERVERS.NET.
.                        6D IN NS       C.ROOT-SERVERS.NET.
.                        6D IN NS       G.ROOT-SERVERS.NET.
.                        6D IN NS       F.ROOT-SERVERS.NET.
.                        6D IN NS       B.ROOT-SERVERS.NET.
.                        6D IN NS       J.ROOT-SERVERS.NET.
.                        6D IN NS       K.ROOT-SERVERS.NET.
.                        6D IN NS       L.ROOT-SERVERS.NET.
;
M.ROOT-SERVERS.NET.      6D IN A        202.12.27.33
I.ROOT-SERVERS.NET.      6D IN A        192.36.148.17
E.ROOT-SERVERS.NET.      6D IN A        192.203.230.10
D.ROOT-SERVERS.NET.      6D IN A        128.8.10.90
A.ROOT-SERVERS.NET.      6D IN A        198.41.0.4
H.ROOT-SERVERS.NET.      6D IN A        128.63.2.53
C.ROOT-SERVERS.NET.      6D IN A        192.33.4.12
G.ROOT-SERVERS.NET.      6D IN A        192.112.36.4
F.ROOT-SERVERS.NET.      6D IN A        192.5.5.241
B.ROOT-SERVERS.NET.      6D IN A        128.9.0.107
J.ROOT-SERVERS.NET.      6D IN A        198.41.0.10
K.ROOT-SERVERS.NET.      6D IN A        193.0.14.129
L.ROOT-SERVERS.NET.      6D IN A        198.32.64.12
```

Datoteka opisuje korenske imenske strežnike po svetu. Ti podatki se sčasoma spreminjajo, zato jih je treba tu in tam posodobiti. V poglavju 8 (Vzdrževanje) si preberite navodila, kako to storiti.

Naslednji razdelek v `named.conf` je zadnji vnos `zone`. Njegov namen in uporabo bom razložil v naslednjem poglavju, za zdaj samo ustvarite datoteko `127.0.0` v podimeniku `pz`: (*Spet po potrebi odstranite začetne presledke*)

```
$TTL 3D
@               IN      SOA     ns.linux.izmislek. hostmaster.linux.izmislek. (
                                1       ; Serial
                                8H      ; Refresh
                                2H      ; Retry
                                4W      ; Expire
                                1D)     ; Minimum TTL
                        NS      ns.linux.izmislek.
1                       PTR     localhost.
```

Vaš `/etc/resolv.conf` mora biti videti približno takole: (*Spet odstranite presledke!*)

```
search poddomena.vasa-domena.edu vasa-domena.edu
nameserver 127.0.0.1
```

Vrstica '`search`' pove, katere domene je treba preiskati za vsako ime računalnika, na katerega se želite priključiti. '`nameserver`' določa naslov vašega imenskega strežnika, v tem primeru vašega lastnega računalnika, na katerem bo named tekel (127.0.0.1 je v redu, četudi ima vaš računalnik še drug naslov). Če želite navesti več imenskih strežnikov, vpišite za vsakega po eno '`nameserver`' vrstico. (Napotek: named te datoteke ne bo nikoli prebral, potreboval jo bo le del sistema, ki bo uporabljal named. Napotek 2: V nekaterih datotekah resolv.conf je vrstica "domain". To je v redu, samo ne uporabite šearch"in "domain", ti dve nastavitvi se izključujeta).

Za ponazoritev, kaj ta datoteka počne: če poskuša program opraviti poizvedbo za `foo`, bo najprej poskusil `foo.poddomena.vasa-domena.edu`, nato `foo.vasa-domena.edu` in šele na koncu `foo`. Pametno je, da v to vrstico ne vnesete preveč domen, ker iskanje po vseh vzame kar nekaj časa.

Zgled predpostavlja, da spadate v domeno `poddomena.vasa-domena.edu` in je vaš računalnik potemtakem `vas-racunalnik.poddomena.vasa-domena.edu`. V vrstici šearch"naj ne bo vaše vrhnje domene (ang.: TLD, Top Level Domain), v tem primeru '`edu`'. Če se pogosto priključujete na računalnike v drugi domeni, lahko dodaste tudi vrstico: (*Odstranite presledke, če je treba*)

```
search poddomena.vasa-domena.edu vasa-domena.edu druga-domena.com
```

in tako naprej. Vsekakor morate tu navedene zglede nadomestiti s pravimi domenami. Zapomnite si, da na koncu domen ni pik. To je zelo pomembno; zapišite si za uho, da na koncu domen ni pik.

3.1 Zagon named

Po vsem tem je čas, da zaženemo named. Če uporabljate povezavo na klic, jo vzpostavite. Napišite '`ndc start`' in pritisnite enter. Če to ne deluje, poskusite '`/usr/sbin/ndc start`'. Če vam tudi tu spodleti, si oglejte poglavje 10 (Vprašanja in odgovori). V datoteki, v katero vaš syslog piše sporočila (navadno `/var/adm/messages`, lahko tudi v imeniku `/var/log`, morda tudi v datoteki `syslog`), se mora med zagonom named (naredite `tail -f /var/log/messages`) izpisati nekaj takega:

(vrstice, ki se končajo s \, se nadaljujejo v naslednji vrstici)

```
Dec 15 23:53:29 localhost named[3768]: starting.  named 8.2.2-P7 \
                Fri Nov 10 04:50:23 EST 2000 ^Iprospector@porky.\
```

```
                devel.redhat.com:/usr/src/bs/BUILD/bind-8.2.2_P7/\
                src/bin/named
Dec 15 23:53:29 localhost named[3768]: hint zone "" (IN) loaded\
                (serial 0)
Dec 15 23:53:29 localhost named[3768]: Zone "0.0.127.in-addr.arpa"\
                (file pz/127.0.0): No default TTL set using SOA\
                minimum instead
Dec 15 23:53:29 localhost named[3768]: master zone\
                "0.0.127.in-addr.arpa" (IN) loaded (serial 1)
Dec 15 23:53:29 localhost named[3768]: listening on [127.0.0.1].53 (lo)
Dec 15 23:53:29 localhost named[3768]: listening on [10.0.0.129].53\
                (wvlan0)
Dec 15 23:53:29 localhost named[3768]: Forwarding source address is\
                [0.0.0.0].1034
Dec 15 23:53:29 localhost named[3769]: Ready to answer queries.
```

Če zagledate tako ali drugačno sporočilo o napaki, ste se verjetno nekje zmotili. Named vam bo povedal, v kateri datoteki je napaka. Odpravite napako in izvedite "ndc restart".

Zdaj lahko preverite svoje nastavitve. Tradicionalno je bil za to uporabljen program `nslookup`. Dandanes se priporoča naslednje:

```
$ dig -x 127.0.0.1

; <<>> DiG 8.2 <<>> -x
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0
;; QUERY SECTION:
;;      1.0.0.127.in-addr.arpa, type = ANY, class = IN

;; ANSWER SECTION:
1.0.0.127.in-addr.arpa.  1D IN PTR  localhost.

;; AUTHORITY SECTION:
0.0.127.in-addr.arpa.   1D IN NS        ns.penguin.bv.

;; Total query time: 30 msec
;; FROM: lookfar to SERVER: default -- 127.0.0.1
;; WHEN: Sat Dec 16 00:16:12 2000
;; MSG SIZE  sent: 40  rcvd: 110
```

Če dobite to, vse deluje, kot je treba. Vsaj upamo tako. Če ne, se vrnite in še enkrat vse preglejte. Vsakič, ko spremenite datoteko `named.conf`, morate znova zagnati named. To lahko storite z ukazom `ndc restart`.

Zdaj lahko izvedete poizvedbo. Poskusite poizvedeti po kakem računalniku, ki je blizu vas. V moji bližini je na primer `pat.uio.no` - univerza v Oslu.

```
$ dig pat.uio.no
```

```
; <<>> DiG 8.2 <<>> pat.uio.no
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 3
;; QUERY SECTION:
;;      pat.uio.no, type = A, class = IN

;; ANSWER SECTION:
pat.uio.no.             1D IN A         129.240.130.16

;; AUTHORITY SECTION:
uio.no.                 1D IN NS        nissen.uio.no.
uio.no.                 1D IN NS        ifi.uio.no.
uio.no.                 1D IN NS        nn.uninett.no.

;; ADDITIONAL SECTION:
nissen.uio.no.          1D IN A         129.240.2.3
ifi.uio.no.             1H IN A         129.240.64.2
nn.uninett.no.          1D IN A         158.38.0.181

;; Total query time: 112 msec
;; FROM: lookfar to SERVER: default -- 127.0.0.1
;; WHEN: Sat Dec 16 00:23:07 2000
;; MSG SIZE  sent: 28  rcvd: 162
```

Tokrat je dig povprašal named za poizvedbo o `pat.uio.no`. Ta je nato stopil v stik z enim od imenskih strežnikov, navedenih v datoteki `root.hints`, in iskal od tam naprej. Poizvedovanje utegne trajati nekaj časa, ker je treba preiskati vse domene v `/etc/resolv.conf`. Naj vas opozorim na zastavico "aa"v vrstici "flags:". Oznanja, da je odgovor prišel neposredno od strežnika, ki je odgovoren za domeno, o kateri smo poizvedovali (odgovor "authoritive"). Pozneje bomo povedali več o odgovorih "authoritive".

Če še enkrat izvedete isto poizvedbo, dobite naslednje:

```
$ dig pat.uio.no

; <<>> DiG 8.2 <<>> pat.uio.no
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 3
;; QUERY SECTION:
;;      pat.uio.no, type = A, class = IN

;; ANSWER SECTION:
pat.uio.no.             23h59m58s IN A  129.240.130.16

;; AUTHORITY SECTION:
UIO.NO.                 23h59m58s IN NS  nissen.UIO.NO.
UIO.NO.                 23h59m58s IN NS  ifi.UIO.NO.
UIO.NO.                 23h59m58s IN NS  nn.uninett.NO.
```

```
;; ADDITIONAL SECTION:
nissen.UIO.NO.          23h59m58s IN A  129.240.2.3
ifi.UIO.NO.             1d23h59m58s IN A  129.240.64.2
nn.uninett.NO.          1d23h59m58s IN A  158.38.0.181

;; Total query time: 4 msec
;; FROM: lookfar to SERVER: default -- 127.0.0.1
;; WHEN: Sat Dec 16 00:23:09 2000
;; MSG SIZE  sent: 28  rcvd: 162
```

Bodite pozorni, da tokrat manjka zastavica "aa". To pomeni, da named ni šel še enkrat izvajat iste poizvedbe po internetu, saj jo ima že v svojem predpomnilniku. *Lahko* pa se zgodi, da je informacija v tem času zastarela. Na to (zelo malo verjetno) možnost ste opozorjeni s tem, da ni zastavice "aa". Zdaj veste, da predpomnenje poizvedb deluje.

3.2 Poizvedovalci (ang. Resolvers)

Vsi OS imajo standardizirane klice C API gethostbyname in gethostbyaddr, ki lahko prejemajo podatke iz več virov. Ti viri so določeni v `/etc/nsswitch.conf` v Linuxu (in nekaterih drugih Unixih). To je dolga datoteka, ki določa, iz katere datoteke ali zbirke podatkov se dobi različne podatke. Ponavadi so pri vrhu uporabne opombe, ki bi jih bilo nemara pametno prebrati. Najdite vrstico, ki se začne z '`hosts:`'; bila naj bi:

```
hosts:      files dns
```

(*Saj se še spomnite nasvetov o začetnih presledkih? Ne bom jih namreč več ponavljal.*)

Če ni vrstice, ki bi se začela s '`hosts:`', jo sami vstavite, tako kot je navedeno zgoraj. Vrstica pravi, naj se najprej preveri datoteka `/etc/hosts` in šele nato DNS, tako je navedeno v `resolv.conf`.

3.3 Čestitam

Zdaj znate postaviti named s predpomnilnikom. Natočite si piva, mleka ali tistega, kar pač najraje žulite, in zažurajte.

4 Posredovanje (ang. forwarding)

V velikih, dobro organiziranih akademskih omrežjih ali omrežjih ponudnikov interneta boste včasih ugotovili, da ima omrežje nastavljeno posredovalno hierarhijo strežnikov DNS, kar pomaga razbremeniti tako strežnike v lastnem omrežju kakor tudi zunanje strežnike. Pogosto je težko ugotoviti, ali ste v takem omrežju ali ne. To pravzaprav sploh ni pomembno, če nastavite strežnik DNS vašega ponudnika interneta kot "posredovalnik" in tako opravite poizvedbe hitreje ter zmanjšate obremenitev vašega omrežja. Če uporabljate dostop prek modema, je to lahko kar precejšnja pridobitev. Za ta zgled bomo predpostavili, da ima vaš ponudnik interneta dva imenska strežnika, ki ju želite uporabljati, njuni številki IP pa sta `10.0.0.1` in `10.1.0.1`. V svoji datoteki `named.conf` torej vstavite v razdelek "options" naslednje vrstice:

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

Za računalnike s povezavo na klic, ki uporabljajo posredovalce, je na voljo imenitna zvijača. Opisana je v razdelku 10 (Vprašanja in odgovori).

Znova zaženite svoj imenski strežnik in ga preverite z dig. To naj bi delovalo.

5 *Preprosta* domena

Kako postaviti svojo domeno

5.1 Najprej kanček teorije

Najprej: Saj ste prebrali vse do tu, kajne? To je namreč nujno.

Preden se *zares* lotimo tega poglavja, vam bom postregel z nekaj teorije in zgledi o tem, kako deluje DNS. In vi boste to prebrali, ker vam bo koristilo. Če se vam tega zares ne ljubi brati, pa bi bilo dobro, ko bi vsaj na hitro preleteli. Branje upočasnite, ko pridete do tistega, kar mora iti v vašo datoteko `named.conf`.

DNS je hierarhičen, drevesno strukturiran sistem. Vrh se imenuje '`.`' in izgovarja 'koren' (ang.: root), kot je značilno za drevesne podatkovne strukture. Pod `.` je veliko število vrhnjih domen (ang.: TLD, Top Level Domain), od katerih so najbolj znane `ORG`, `COM`, `EDU` in `NET`, seveda pa jih je še veliko več. Struktura je prav taka kakor drevo - ima korenino in je razvejena. Če imate vsaj nekaj računalniškega znanja, boste v DNS prepoznali iskalno drevo, našli boste stičišča, liste in robove. Pike so stičišča in robovi so na imenih.

Ko iščemo določen računalnik, gre poizvedba rekurzivno po hierarhiji, začenši pri vrhu. Če hočete izvedeti naslov `prep.ai.mit.edu`, mora vaš imenski strežnik nekje začeti. Začne s svojim predpomnilnikom. Če je v predpomnilniku odgovor, bo odgovoril, kakor smo videli v prejšnjem razdelku. Če odgovora ne pozna, bo začel odstranjevati dele imena, začenši na levi strani. Pogledal bo, ali ve, kaj o `ai.mit.edu.`, potem o `mit.edu.`, potem o `edu.`. Vedno pa pozna `.`, saj je v datoteki `root.hints`. Potem bo povprašal `.` strežnik o `prep.ai.mit.edu`. Ta `.` strežnik ne bo poznal odgovora, bo pa pomagal s preusmeritivjo (ang. refferal), s katero bo nakazal, kje naj se išče odgovor. Prek preusmeritev bo naš strežnik prišel do strežnika, ki bo imel pravi odgovor. To bom zdaj ponazoril. `+norec` pomeni, naj dig sprašuje nerekurzivno, tako da moramo sami delati rekurzijo. Z drugimi stikali zmanjšamo obseg podatkov, ki jih izpisuje dig:

```
$ dig +norec +noH +noques +nostats +nocmd prep.ai.mit.edu.
;; res options: init defnam dnsrch
;; got answer:
; flags: qr ra; QUERY: 1, ANSWER: 0, AUTHORITY: 13, ADDITIONAL: 13
;; AUTHORITY SECTION:
.                       5d23h48m47s IN NS  I.ROOT-SERVERS.NET.
.                       5d23h48m47s IN NS  E.ROOT-SERVERS.NET.
.                       5d23h48m47s IN NS  D.ROOT-SERVERS.NET.
.                       5d23h48m47s IN NS  A.ROOT-SERVERS.NET.
```

```
.                        5d23h48m47s IN NS  H.ROOT-SERVERS.NET.
.                        5d23h48m47s IN NS  C.ROOT-SERVERS.NET.
.                        5d23h48m47s IN NS  G.ROOT-SERVERS.NET.
.                        5d23h48m47s IN NS  F.ROOT-SERVERS.NET.
.                        5d23h48m47s IN NS  B.ROOT-SERVERS.NET.
.                        5d23h48m47s IN NS  J.ROOT-SERVERS.NET.
.                        5d23h48m47s IN NS  K.ROOT-SERVERS.NET.
.                        5d23h48m47s IN NS  L.ROOT-SERVERS.NET.
.                        5d23h48m47s IN NS  M.ROOT-SERVERS.NET.

;; ADDITIONAL SECTION:
I.ROOT-SERVERS.NET.      6d23h48m47s IN A  192.36.148.17
E.ROOT-SERVERS.NET.      6d23h48m47s IN A  192.203.230.10
D.ROOT-SERVERS.NET.      6d23h48m47s IN A  128.8.10.90
A.ROOT-SERVERS.NET.      6d23h48m47s IN A  198.41.0.4
H.ROOT-SERVERS.NET.      6d23h48m47s IN A  128.63.2.53
C.ROOT-SERVERS.NET.      6d23h48m47s IN A  192.33.4.12
G.ROOT-SERVERS.NET.      6d23h48m47s IN A  192.112.36.4
F.ROOT-SERVERS.NET.      6d23h48m47s IN A  192.5.5.241
B.ROOT-SERVERS.NET.      6d23h48m47s IN A  128.9.0.107
J.ROOT-SERVERS.NET.      6d23h48m47s IN A  198.41.0.10
K.ROOT-SERVERS.NET.      6d23h48m47s IN A  193.0.14.129
L.ROOT-SERVERS.NET.      6d23h48m47s IN A  198.32.64.12
M.ROOT-SERVERS.NET.      6d23h48m47s IN A  202.12.27.33
```

To je preusmeritev. Dobili samo samo "Authority section", brez "Answer section". Naš lastni imenski strežnik nas preusmeri. Izberimo naključno enega:

```
$ dig +norec +noH +noques +nostats +nocmd prep.ai.mit.edu. @H.ROOT-SERVERS.NET.
; (1 server found)
;; res options: init defnam dnsrch
;; got answer:
; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 3, ADDITIONAL: 3
;; AUTHORITY SECTION:
MIT.EDU.                 2D IN NS        BITSY.MIT.EDU.
MIT.EDU.                 2D IN NS        STRAWB.MIT.EDU.
MIT.EDU.                 2D IN NS        W20NS.MIT.EDU.

;; ADDITIONAL SECTION:
BITSY.MIT.EDU.           2D IN A         18.72.0.3
STRAWB.MIT.EDU.          2D IN A         18.71.0.151
W20NS.MIT.EDU.           2D IN A         18.70.0.160
```

Takoj nas preusmeri na strežnik MIT.EDU. Znova naključno izberemo neki strežnik:

```
$ dig +norec +noH +noques +nostats +nocmd prep.ai.mit.edu. @bitsy.mit.edu
; (1 server found)
;; res options: init defnam dnsrch
;; got answer:
; flags: qr ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 4
```

```
;; ANSWER SECTION:
prep.ai.mit.edu.        3h50m7s IN A    198.186.203.18

;; AUTHORITY SECTION:
AI.MIT.EDU.             6H IN NS        FEDEX.AI.MIT.EDU.
AI.MIT.EDU.             6H IN NS        LIFE.AI.MIT.EDU.
AI.MIT.EDU.             6H IN NS        ALPHA-BITS.AI.MIT.EDU.
AI.MIT.EDU.             6H IN NS        BEET-CHEX.AI.MIT.EDU.

;; ADDITIONAL SECTION:
FEDEX.AI.MIT.EDU.       6H IN A         192.148.252.43
LIFE.AI.MIT.EDU.        6H IN A         128.52.32.80
ALPHA-BITS.AI.MIT.EDU.  6H IN A         128.52.32.5
BEET-CHEX.AI.MIT.EDU.   6H IN A         128.52.32.22
```

Tokrat smo dobili tudi "ANSWER SECTION"in odgovor na naše vprašanje. V razdelku "AUTHORITY SECTION"je podatek, katere strežnike naj naslednjič povprašamo o `ai.mit.edu`. Tako lahko neposredno njih vprašamo, kadar bomo spraševali po imenih iz `ai.mit.edu`.

Začenši s `.`, smo našli imenski strežnik za vsako stopnjo v imenu domene s preusmeritvijo. Če bi uporabili svoj strežnik DNS namesto vseh drugih strežnikov, bi se vsi tako zbrani podatki shranili v predpomnilnik in kar nekaj časa mu ne bi bilo treba ponoviti že opravljenih poizvedb.

V drevesni prispodobi je vsaka “`.`” v imenu vejišče. Vsak del med dvema “`.`” je ime posameznih vej v drevesu. Po drevesu se gre tako, da se vzame želeno ime (`prep.ai.mit.edu`) in poizvemo pri korenu (`.`) ali pri kateremkoli že strežniku od korena proti prep.ai.mit.edu, ki ga imamo v predpomilniku. Šele ko v predpomnilniku ni več iskanih zapisov, se rekurzivne poizvedbe opravijo pri zunanjih strežnikih in se sledi preusmeritvam (ang. refferal) vse dlje proti imenu.

Domena, o kateri se manj govori, a je vseeno zelo pomembna, je `in-addr.arpa`. Prav tako kakor ‘navadne’ domene je ugnezdena, vendar nam omogoča ravno nasprotno - da izvemo imena računalnikov iz njihovih naslovov. Pomembno si je zapomniti, da so naslovi IP v domeni `in-addr.arpa` zapisani v nasprotnem vrstnem redu. Če je naslov računalnika `192.128.52.43`, je postopek, ki ga ubere named, prav tak kakor za `prep.ai.mit.edu`: najdi strežnik za `arpa.`, najdi strežnik `in-addr.arpa.`, najdi strežnik za `192.in-addr.arpa.`, najdi strežnik za `128.192.in-addr.arpa.`, najdi strežnik za `52.128.192.in-addr.arpa.` in končno najdi zapise za `43.52.128.192.in-addr.arpa.` Bistro, kaj? (Recite ‘da’.) Prav dejstvo, da so številke obrnjene, lahko povzroči veliko zmešnjavo.

5.2 Naša lastna domena

Zdaj bomo definirali svojo domeno - `linux.izmislek` - in v njej definirali računalnike. Uporabil sem popolnoma izmišljeno domeno; tako smo lahko gotovi, da ne bomo komu skočili v zelje.

Še nekaj, preden začnemo: v imenih računalnikov ne sme biti določenih znakov – omejeni smo na črke angleške abecede: a-z, na številke:0-9 in na znak ’-’ (pomišljaj). Držite se jih. DNS ne razlikuje različnih velikosti črk, zato je `pat.uio.no` zanj isto kakor `Pat.UiO.No`.

Delo smo začeli že s tem razdelkom v `named.conf`:

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

```
};
```

Prosim, zapomnite si, da na koncu domen v tej datoteki ni pike `.`. Ta razdelek pove, da bomo zdaj definirali `0.0.127.in-addr.arpa`, da smo za to domeno glavni (angl. master) strežnik in da so podatki zanjo shranjeni v datoteki `pz/127.0.0`. Datoteko smo že prej ustvarili, v njej pa je:

```
$TTL 3D
@               IN      SOA     ns.linux.izmislek. hostmaster.linux.izmislek. (
                                1       ; Serial
                                8H      ; Refresh
                                2H      ; Retry
                                4W      ; Expire
                                1D)     ; Minimum TTL
                        NS      ns.linux.izmislek.
1                       PTR     localhost.
```

Prosim, zapomnite si pike `.` na koncu vseh polnih domen v tej datoteki kot nasprotje datoteki `named.conf`. Nekateri začnejo vsako območno datoteko z ukazom `$ORIGIN`, vendar je to popolnoma odveč. Izvor (kam v hierarhiji DNS spada) območne datoteke je namreč naveden v `named.conf`, v tem primeru je to `0.0.127.in-addr.arpa`.

V tej območni datoteki so tri zapisi virov (ang.: RRs, resource records): SOA, NS in PTR. SOA pomeni začetek pristojnosti (ang.: Start Of Authority). '@' je oznaka za izvor in ker je v stolpcu 'domena' za to datoteko navedena 0.0.127.in-addr.arpa, prva vrstica v resnici pomeni

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

NS je zapis vira imenskega strežnika (ang.: Name Server RR). Na začetku te vrstice ni oznake '@', ker se to ohranja še iz prejšnje vrstice - to prihrani veliko tipkanja. Zapis vira NS bi torej lahko zapisali tudi kot

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

DNS pove, kateri računalnik je strežnik za domeno `0.0.127.in-addr.arpa`, in to je `ns.linux.izmislek`. 'ns' je standardno ime za imenske strežnike, kot je `www.`*nekaj* standardno ime za spletne strežnike, vendar to ni pravilo - ime je lahko karkoli.

Zapis vira PTR (ang.: Domain Name Pointer) pravi, da se računalniku na naslovu 1 v podomrežju `0.0.127.in-addr.arpa`, torej 127.0.0.1, imenuje `localhost`.

Zapis SOA (Start Of Authority) je glava za *vse* območne datoteke, v katerih mora biti natanko en zapis SOA. Ta zapis obsega podatke o domeni, od kod prihaja (računalnik z imenom `ns.linux.izmislek`), kdo je odgovoren za njeno vsebino (`hostmaster@linux.izmislek` - tu vstavite svoj e-naslov), katera različica območne datoteke je to (serijska številka: 1) in druge podatke, ki imajo opraviti s strežniki DNS s predpomnenjem ter s sekundarnimi strežniki DNS. Za vsa druga polja (osveževanje - refresh, vnovični poskus - retry, razveljavljenje - expire in minimum) se zanesite na številke, ki so navedene v tem HOWTO, in vse bi moralo ustrezno delovati. Pred zapisom SOA pride obvezna vrstica `$TTL 3D`. Naj bo v vseh vaših območnih datotekah.

Zdaj spet zaženite named (ukaz je `ndc restart`) in uporabite dig, če si želite ogledati svoje delo. `-x` opravi obratno poizvedbo (ang.: reverse query):

```
$ dig -x 127.0.0.1

; <<>> DiG 8.2 <<>> -x
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0
;; QUERY SECTION:
;;      1.0.0.127.in-addr.arpa, type = ANY, class = IN

;; ANSWER SECTION:
1.0.0.127.in-addr.arpa.  1D IN PTR  localhost.

;; AUTHORITY SECTION:
0.0.127.in-addr.arpa.   1D IN NS        ns.penguin.bv.

;; Total query time: 5 msec
;; FROM: lookfar to SERVER: default -- 127.0.0.1
;; WHEN: Sat Dec 16 01:13:48 2000
;; MSG SIZE  sent: 40  rcvd: 110
```

Torej uspe preslikati 127.0.0.1 v `localhost`. Odlično. Zdaj se lahko lotimo našega poglavitnega opravila, domene `linux.izmislek`. Najprej vstavimo nov območni (ang.: zone) razdelek v `named.conf`:

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

Spet si zapomnite, da na koncu domen v `named.conf` ni pik '`.`'.

V območno datoteko `linux.izmislek` bomo napisali nekaj popolnoma izmišljenih podatkov:

```
;
; Območna datoteka za linux.izmislek
;
; Polna območna datoteka
;
$TTL 3D
@       IN      SOA     ns.linux.izmislek. hostmaster.linux.izmislek. (
                        199802151       ; serial, današnji datum + današnja serijska številka
                        8H              ; refresh, sekund
                        2H              ; retry, sekund
                        4W              ; expire, sekund
                        1D )            ; minimum, sekund
;
                NS      ns              ; internetni naslov imenskega strežnika
                MX      10 mail.linux.izmislek          ; Primarni poštni strežnik
                MX      20 mail.prijatelj.izmislek.     ; Sekundarni poštni strežnik
```

```
;
localhost       A       127.0.0.1
ns              A       192.168.196.2
mail            A       192.168.196.4
```

O zapisu SOA si morate zapomniti dvoje. `ns.linux.izmislek` *mora* biti resničen računalnik z zapisom A. Prepovedano je imeti zapis CNAME za računalnik, naveden v zapisu SOA. Vsekakor ni pomembno, da se imenuje 'ns', lahko ima kakršnokoli veljavno ime. Drugo, hostmaster.linux.izmislek se bere kot hostmaster@linux.izmislek in mora biti veljaven naslov ali preusmeritev naslova, na katerega bo tisti, ki vzdržuje DNS, prejemal pošto (ta človek naj pošto tudi redno pregleduje). Ni nujno, da je to ravno 'hostmaster', lahko je vaš navaden e-naslov, vendar se pogosto pričakuje tudi naslov 'hostmaster'.

V tej datoteki je nov zapis vira, in sicer MX, kar pomeni poštni strežnik (ang.: Mail eXchanger). Ta zapis pove sistemom, kam naj pošljejo pošto, ki je naslovljena na `nekdo@linux.izmislek`, v tem primeru na `mail.linux.izmislek` ali `mail.prijatelj.izmislek`. Številka pred imeni računalnikov pomeni prednost. Če je le mogoče, se pošta pošlje strežniku z najmanjšo številko (10), drugače pa naslednjemu z najmanjšo številko, v tem primeru `mail.prijatelj.izmislek`, ki ima prednost 20.

Znova zaženite named z ukazom `ndc restart` in opazujte rezultate z digom:

```
$ dig any linux.izmislek +pfmin
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 23499
;; QUERY: 1, ANSWER: 4, AUTHORITY: 1, ADDITIONAL: 1
;; QUERY SECTION:
;;      linux.izmislek, type = ANY, class = IN

;; ANSWER SECTION:
linux.izmislek.           3D IN MX        10 mail.linux.izmislek.linux.izmislek.
linux.izmislek.           3D IN MX        20 mail.prijatelj.izmislek.
linux.izmislek.           3D IN NS        ns.linux.izmislek.
linux.izmislek.           3D IN SOA       ns.linux.izmislek. hostmaster.linux.izmislek. (
                                       199802151       ; serial
                                       8H              ; refresh
                                       2H              ; retry
                                       4W              ; expiry
                                       1D )            ; minimum
```

Po pazljivem opazovanju boste odkrili napako. Vrstica

```
linux.izmislek.           3D IN MX        10 mail.linux.izmislek.linux.izmislek.
```

je popolnoma napačna. Morala bi biti

```
linux.izmislek.           3D IN MX        10 mail.linux.izmislek.
```

Namenoma sem naredil napako, da se boste lahko iz nje učili :-) Če pogledate v območno datoteko, boste ugotovili, da v vrstici:

```
            MX      10 mail.linux.izmislek          ; Primarni poštni strežnik
```

manjka pika. Ali, drugače povedano, ima 'linux.izmislek' preveč. Če se v območni datoteki ime računalnika ne konča s piko, se mu doda izvor in v tem primeru povzroči dvojni `linux.izmislek.linux.izmislek`. Torej je

```
                MX      10 mail.linux.izmislek.         ; Primarni poštni strežnik
```

ali

```
                MX      10 mail                 ; Primarni poštni strežnik
```

pravilno. Priporočam uporabo druge možnosti, ker je treba manj tipkati. Nekateri strokovnjaki za BIND se s tem ne bodo strinjali, nekateri drugi pa se bodo. V območni datoteki je lahko domena napisana in se konča s '`.`' ali pa ni napisana in se namesto nje privzame izvirna domena.

Poudariti moram, da v named.conf *ne sme* biti pike na koncu domen. Še sanja se vam ne, kolikokrat je '`.`' tu povzročila pošteno zmešnjavo.

Zdaj, ko vse to veste, je pred vami nova, popravljena območna datoteka s še nekaj dodatnimi informacijami:

```
;
; Območna datoteka za linux.izmislek
;
; Polna območna datoteka
;
$TTL 3D
@       IN      SOA     ns.linux.izmislek. hostmaster.linux.izmislek. (
                        199802151       ; serial, današnji datum + današnja serijska številka
                        8H              ; refresh, sekund
                        2H              ; retry, sekund
                        4W              ; expire, sekund
                        1D )            ; minimum, sekund
;
                TXT     "Linux.Izmislek, vaš svetovalec za DNS"
                NS      ns              ; internetni naslov imenskega strežnika
                NS      ns.prijatelj.izmislek.
                MX      10 mail         ; Primarni poštni strežnik
                MX      20 mail.prijatelj.izmislek.     ; Sekundarni poštni strežnik
;
localhost       A       127.0.0.1

gw              A       192.168.196.1
                HINFO   "Cisco" "IOS"
                TXT     "Usmerjevalnik"

ns              A       192.168.196.2
                MX      10 mail
                MX      20 mail.prijatelj.izmislek.
```

```
                HINFO   "Pentium" "Linux 2.0"
www             CNAME   ns

donald          A       192.168.196.3
                MX      10 mail
                MX      20 mail.prijatelj.izmislek.
                HINFO   "i486"  "Linux 2.0"
                TXT     "DEK"

mail            A       192.168.196.4
                MX      10 mail
                MX      20 mail.prijatelj.izmislek.
                HINFO   "386sx" "Linux 1.2"

ftp             A       192.168.196.5
                MX      10 mail
                MX      20 mail.prijatelj.izmislek.
                HINFO   "P6" "Linux 2.1.86"
```

Tu smo spoznali še cel kup novih zapisov: HINFO, informacija o računalniku (ang.: Host INFOrmation), ima dva dela in k dobrim navadam sodi, da ju izpolnite. Prvi del je strojna oprema ali procesor računalnika, drugi pa operacijski sistem. Računalnik z imenom 'ns' ima procesor Pentium in poganja Linux 2.0. CNAME, kanonično ime (ang.: Canonical NAME) je način za poimenovanje istega računalnika z več različnimi imeni. V našem primeru je www alternativno ime za ns.

Uporaba zapisa CNAME je nekoliko sporna, vendar se je popolnoma varno ravnati po pravilu, da se zapisi MX, CNAME in SOA ne smejo *nikoli* nanašati na zapis CNAME, temveč samo na nekaj z zapisom A. Ni, denimo, priporočljivo imeti

```
foobar          CNAME   www                     ; NE!
```

pravilno pa je takole

```
foobar          CNAME   ns                      ; Da!
```

Prav tako se je varno zanašati na to, da CNAME ni dovoljeno ime računalnika za e-naslov: `webmaster@www.linux.izmislek` na primer ni dovoljen naslov glede na zgoraj navedene nastavitve. Pričakujete lahko, da bo kar nekaj upraviteljev hotelo, da to pravilo upoštevate, čeprav pri vas morda deluje. Težavam se izognete z uporabo zapisov A (in še nekaterih drugih, na primer MX):

```
www             A       192.168.196.2
```

Nekaj super strokovnjakov za BIND celo priporoča, naj se zapisi CNAME *sploh* ne uporabljajo, vendar razprava o tem, 'zakaj da' in 'zakaj ne', ne sodi v ta HOWTO.

Kakor ste že opazili, ta HOWTO in številni strežniki ne upoštevajo tega pravila.

Naložite novo zbirko podatkov z ukazom `ndc reload`, ki pove named, naj še enkrat prebere svoje datoteke.

```
$ dig linux.izmislek axfr

; <<>> DiG 8.2 <<>> linux.bogus axfr
$ORIGIN linux.izmislek.
@                       3D IN SOA       ns hostmaster (
                                        199802151       ; serial
                                        8H              ; refresh
                                        2H              ; retry
                                        4W              ; expiry
                                        1D )            ; minimum

                        3D IN NS        ns
                        3D IN NS        ns.prijatelj.izmislek.
                        3D IN MX        10 mail
                        3D IN MX        20 mail.prijatelj.izmislek.
                        3D IN TXT       "Linux.Izmislek, vaš svetovalec za DNS"
gw                      3D IN TXT       "Usmerjevalnik"
                        3D IN HINFO     "Cisco" "IOS"
                        3D IN A         192.168.196.1
localhost               3D IN A         127.0.0.1
mail                    3D IN HINFO     "386sx" "Linux 1.2"
                        3D IN MX        10 mail
                        3D IN MX        20 mail.prijatelj.izmislek.
                        3D IN A         192.168.196.4
www                     3D IN CNAME     ns
donald                  3D IN TXT       "DEK"
                        3D IN HINFO     "i486" "Linux 2.0"
                        3D IN MX        10 mail
                        3D IN MX        20 mail.prijatelj.izmislek.
                        3D IN A         192.168.196.3
ns                      3D IN HINFO     "Pentium" "Linux 2.0"
                        3D IN MX        10 mail
                        3D IN MX        20 mail.prijatelj.izmislek.
                        3D IN A         192.168.196.2
ftp                     3D IN HINFO     "P6" "Linux 2.1.86"
                        3D IN MX        10 mail
                        3D IN MX        20 mail.prijatelj.izmislek.
                        3D IN A         192.168.196.5
@                       3D IN SOA       ns hostmaster (
                                        199802151       ; serial
                                        8H              ; refresh
                                        2H              ; retry
                                        4W              ; expiry
                                        1D )            ; minimum

;; Received 29 answers (29 records).
;; FROM: lookfar to SERVER: 127.0.0.1
;; WHEN: Sat Dec 16 01:35:05 2000
```

Tole je v redu. Kakor vidite, je izpis zelo podoben območni datoteki. Poglejmo, kaj pravi samo za `www`:

```
$ dig www.linux.izmislek +pfmin
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27345
;; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 1
;; QUERY SECTION:
;;      www.linux.izmislek, type = A, class = IN

;; ANSWER SECTION:
www.linux.izmislek.        3D IN CNAME    ns.linux.izmislek.
ns.linux.izmislek.         3D IN A        192.168.196.2
```

Z drugimi besedami, pravo ime `www.linux.izmislek` je `ns.linux.izmislek`, poleg tega pa vam poda še nekaj drugih informacij o ns, dovolj, da bi se lahko priključili nanj, če bi bili program.

Zdaj smo na polovici poti.

5.3 Obratni vnosi (ang. reverse zone)

Zdaj lahko programi pretvorijo imena v domeni linux.izmislek v naslove, na katere se lahko priključijo, potrebujemo pa še obratne vnose, ki bodo omogočili DNS, da pretvarja naslove v imena. Imena so pomembna informacija za številne strežnike (FTP, IRC, WWW in druge), saj se na podlagi tega odločajo, ali se bodo sploh pogovarjali z vami ali ne in kakšno prednost vam bodo dali, če se bodo. Za popoln dostop do vseh storitev v internetu torej potrebujete obratne vnose.

Dodajte tole v `named.conf`:

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

Stvar je natanko taka kakor pri `0.0.127.in-addr.arpa`, pa tudi vsebina je podobna:

```
$TTL 3D
@       IN      SOA     ns.linux.izmislek. hostmaster.linux.izmislek. (
                        199802151 ; Serial, todays date + todays serial
                        8H      ; Refresh
                        2H      ; Retry
                        4W      ; Expire
                        1D)     ; Minimum TTL
                NS      ns.linux.izmislek.

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

Znova zaženite named (`ndc restart`) in spet preglejte svoje delo z dig:

```
$ dig -x 192.168.196.4 +pfmin
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 8764
;; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
;; QUERY SECTION:
;;      4.196.168.192.in-addr.arpa, type = ANY, class = IN

;; ANSWER SECTION:
4.196.168.192.in-addr.arpa.  3D IN PTR  mail.linux.izmislek.
```

Zaenkrat je videti v redu, preglejmo še vse naenkrat:

```
dig -x 192.168.196 AXFR

; <<>> DiG 8.2 <<>> -x AXFR
$ORIGIN 196.168.192.in-addr.arpa.
@                       3D IN SOA       ns.linux.izmislek. hostmaster.linux.izmislek. (
                                        199802151       ; serial
                                        8H              ; refresh
                                        2H              ; retry
                                        4W              ; expiry
                                        1D )            ; minimum

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

;; Received 8 answers (8 records).
;; FROM: lookfar to SERVER: 127.0.0.1
;; WHEN: Sat Dec 16 01:44:03 2000
```

Odlično! Če vaš izpis ni videti takole, poglejte v syslog za sporočila o napakah, kot sem razložil na začetku tega poglavja, pod naslovom 3.1 (Zagon named).

5.4 Opozorila

Rad bi še nekaj dodal. Številke IP, ki sem jih uporabil v zgledih, navedenih tu, so vzete iz blokov 'zasebnih omrežij', kar pomeni, da se jih v internetu ne sme uporabljati za prave naslove. So pa zato varne za uporabo v zgledih. Drugo, na kar vas moram opozoriti, je vrstica `notify no;`, ki pove named, naj svojih sekundarnih strežnikov ne opozarja na spremembe območnih datotek. V BIND-8 lahko named opozori druge strežnike, navedene v zapisih NS, kadar se katera od teh spremeni. To je sicer pripravno pri resnični uporabi, za zasebne poskuse pa mora biti izključeno - saj vendar nočemo, da bi naši poskusi onesnažili internet, kajne da ne?

In seveda, ta domena je popolnoma izmišljena, kot so izmišljeni tudi naslovi v njej. Zgled prave domene si oglejte v naslednjem poglavju.

5.5 Zakaj obratne poizvedbe ne delujejo.

Pri postavljanju obratnih vnosov preži na nas nekaj "pasti", ki jih navadno rešimo z imenskimi poizvedbami. Preden greste naprej, potrebujete delujoče obratne vnose - če vam ne delujejo, se vrnite in jih popravite.

Razložil bom dve možnosti, zakaj obratni vnosi ne delujejo, kot so videti zunaj vašega omrežja.

5.5.1 Obratni vnosi niso pristojni

Ko dobite od ponudnika internetnih storitev svoj del omrežja in domeno, je domena navadno pristojna za ta del omrežja. Pristojnost je zapis NS, lepilo, ki vam omogoča, da pridete od enega imenskega strežnika do drugega, kot sem pojasnil v poglavju o teoriji. Saj ste ga prebrali? Če vaši obratni vnosi ne delujejo, se vrnite in si preberite poglavje. Takoj zdaj.

Tudi obratni vnosi morajo biti pristojni za določen del omrežja. Če ste dobili omrežje `192.168.192` z domeno `linux.izmislek`, mora ponudnik v svoje območne datoteke zapisati tako vnos NS za imenske poizvedbe kakor tudi vnos NS za obratne poizvedbe. Če sledite verigi od `in-addr.arpa.` do vas, boste verjetno našli prekinitev - najverjetneje pri svojem ponudniku. V tem primeru stopite v stik z njim in ga prosite, naj odpravi napako.

5.5.2 Imate brezrazredno podomrežje

To je dokaj zapletena tema, vendar so brezrazredna podomrežja dandanes zelo pogosta in če niste vsaj srednje veliko podjetje, ste najverjetneje tudi vi v takem podomrežju.

Brezrazredna podomrežja danes ohranjajo internet pri življenju. Nekaj let nazaj se je veliko razpravljalo o pomanjkanju številk IP. Bistre glave v IETF (Internet Engineering Task Force, organizacija, ki skrbi za internet) so se staknile in iznašle rešitev. Za določeno ceno. Ta cena je, da dobite manj kakor podomrežje "C", poleg tega pa se lahko zgodi, da nekatere stvari ne bodo delovale. Oglejte si *Ask Mr. DNS at* <`http://www.acmebw.com/askmrdns/00007.htm`>, če si želite dobre razlage tega in napotkov, kako se s tem spopasti.

Ste prebrali? Tega ne bom razlagal, zato si, prosim, preberite.

Prvi del problema je, da mora vaš ponudnik interneta razumeti tehniko, ki jo opisuje g. DNS. Nekateri majhni ponudniki je ne razumejo popolnoma, zato jim jo boste morda morali razložiti. Prej se prepričajte, ali jo sploh sami obvladate ;-) Ponudnik bo nato postavil obratne vnose v njihovem strežniku. To lahko preverite z dig.

Drugi in zadnji del problema pa je, da morate tudi vi razumeti tehniko. Če o tem niste prepričani, se vrnite in si znova preberite, nato pa lahko postavite svoje brezrazredne obratne vnose, kot vam opisuje g. DNS.

Je pa še ena težava. Stari programi *ne bodo* sposobni slediti zvijači `CNAME` v verigi poizvedovanja in zato ne bodo sposobni opraviti obratnih poizvedb za vaš računalnik. To lahko pomeni, da vas bo strežnik razvrstil v napačen razred, vam prepovedal dostop ali storil kaj podobnega. Če res potrebujete njegove storitve, je edina možnost (ki jo jaz poznam), da poprosite svojega ponudnika, naj vstavi vaš zapis PTR v njihovo brezrazredno območno datoteko namesto v zvijačo z vnosom CNAME.

Nekateri ponudniki interneta vam bodo ponudili tudi drugačne rešitve, kot so spletni forumi za obratne vnose in podobni avtomagični sistemi.

5.6 Sekundarni (ang. slave) strežniki

Po tem, ko ste pravilno vzpostavili svoje območje v primarnem strežniku, morate vzpostaviti vsaj še en sekundarni strežnik. Ti so potrebni zaradi zanesljivosti. Če vaš strežnik pade, bodo uporabniki še vedno dobili potrebne podatke za vašo domeno s pomočjo sekundarnega strežnika. Sekundarni strežnik naj bo čim dlje od vas in imejta čim manj skupnega z naslednjim: električno napajanje, povezava LAN, ponudnik internetnih storitev, mesto in dežela. Če so vsi našteti atributi za primarni in sekundarni strežnik različni, imate zelo dober sekundarni strežnik.

Sekundarni strežnik je imenski strežnik, ki prekopira vse območne datoteke iz primarnega strežnika. Nastavite ga takole:

```
zone "linux.izmislek" {
        type slave;
        file "sz/linux.izmislek";

        masters { 192.168.196.2; };
};
```

Uporabi se mehanizem, imenovan prenos območja (ang. zone-transfer). Tega nadzira vaš SOA.

```
@       IN      SOA     ns.linux.izmislek. hostmaster.linux.izmislek. (
                        199802151       ; serial, todays date + todays serial #
                        8H              ; refresh, seconds
                        2H              ; retry, seconds
                        4W              ; expire, seconds
                        1D )            ; minimum, seconds
```

Območje se prenese le, če je serijska številka na primarnem strežniku večja od tiste na sekundarnem. Na vsak osveževalni interval (ang. refresh) sekundarni preveri, ali ima primarni novejši zapis. Če zaradi nedosegljivosti ni mogoče preveriti zapisa, bo poskusil vsak interval, kot je določeno v polju vnovični poskus (ang. retry). Če bo primarni nedosegljiv ves čas do izteka razveljavitnega (ang. expire) intervala, sekundarni odstrani to območje iz datotečnega sistema in ne bo več posredoval njegovih podatkov.

6 Temeljne varnostne nastavitve

Avtor Jamie Norrish

Nastavitve, s katerimi zmanjšate možnosti za težave

Nekaj preprostih korakov, ki povečajo varnost in potencialno zmanjšajo obremenitev. Tu predstavljena snov je le začetek; če vas skrbi varnost (in ta bi vas morala skrbeti), si oglejte še preostalo gradivo v internetu (glej 11 (Zadnje poglavje)).

Naslednje nastavitve se pokažejo v `named.conf`. Če se nastavitev pokaže v razdelku `options`, zadeva vsa območja v tisti datoteki. Če se pokaže znotraj vnosa `zone`, zadeva le tista območja. Zapis v vnosu `zone` je močnejši od tistega v razdelku `options`.

6.1 Omejevanje prenosa območja

Da bi lahko sekundarni strežniki odgovarjali na poizvedbe, morajo najprej prenesti območje iz vašega primarnega strežnika. Zelo malo drugih ima to potrebo. Zato omejite prenose območja z `allow-transfer` nastavitvijo. Privzemimo, da je 192.168.1.4 številka IP strežnika ns.friend.izmislek in dodajte še sebe za razhroščevalne namene:

```
zone "linux.izmislek" {
      allow-transfer { 192.168.1.4; localhost; };
};
```

S to omejitvijo so ljudem na voljo le podatki, ki jih neposredno zahtevajo. Nihče ne more dobiti natančnih podatkov o vaši postavitvi.

6.2 Zaščita pred prikrivanjem naslova

Najprej onemogočite poizvedbe za domene, ki jih nimate v lasti, razen iz vaših notranjih/krajevnih računalnikov. S tem ne le onemogočite zlonamerno uporabo vašega strežnika, ampak tudi omejite nepotrebno uporabo tega strežnika.

```
options {
      allow-query { 192.168.196.0/24; localhost; };
};

zone "linux.izmislek" {
      allow-query { any; };
};

zone "196.168.192.in-addr.arpa" {
      allow-query { any; };
};
```

Zatem onemogočite rekurzivne poizvedbe, razen iz vaših notranjih/krajevnih računalnikov. S tem se izognete napadom z okvarjenimi podatki (ang. cache poisoning attack - napačni podatki se podtaknejo vašemu strežniku).

```
options {
        allow-recursion { 192.168.196.0/24; localhost; };
};
```

6.3 Uporaba named kot ne-root

Zelo dobro je, da vaš strežnik named teče pod drugim uporabnikom, kot je root. Tako morebitni vlomilec (ang.: cracker) nima na voljo vseh pravic. Najprej morate ustvariti uporabnika in skupino, pod katero naj teče named, nato pa popravite zagonsko (ang.: init) skripto named. Namedu nastavite uporabniško ime in skupino s stikaloma -u in -g.

Na primer, v Debian GNU/Linux 2.2 spremenite skripto `/etc/init.d/bind` tako, da je v njej vrstica:

```
start-stop-daemon --start --quiet --exec /usr/sbin/named -- -u named -g named
```

Enako je mogoče narediti pri Red Hat in drugih distribucijah. Dave Lugo je opisal varno dvojno postavitev z aječo (ang.: chroot) `<http://www.etherboy.com/dns/chrootdns.html>`, ki bi utegnila biti zanimiva za branje.

7 Zgled prave domene

Tu je nekaj *pravih* datotek 'zone'

Uporabniki so predlagali, da poleg učnih zgledov dodam tudi datoteke prave in delujoče domene.

Te zglede uporabljam z dovoljenjem Davida Bullocka z LAND-5. Datoteke so bile posnete 24. septembra 1996 in sem jih nato nekoliko popravil, da delujejo z BIND-8, zato se to, kar vidite tu, nekoliko razlikuje od rezultatov, ki jih dobite, če poizvedujete pri imenskih strežnikih LAND-5.

7.1 /etc/named.conf (ali /var/named/named.conf)

Tu najdemo poglavitne razdelke 'zone' za dvoje obratnih vnosov: omrežje 127.0.0 in omrežje LAND-5 `206.6.177`. Tu je tudi primarni razdelek za `land-5.com`. Bodite pozorni na to, da datoteke niso shranjene v mapi `pz`, kot v tem HOWTO, temveč v mapi `zone`.

```
// 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";
};
```

Če boste to datoteko prekopirali v svoj named.conf, da bi se igrali, **PROSIM**, vpišite še vrstico “`notify no;`” v razdelke ‘zone’ za obe `land-5` in tako preprečite nesrečo.

7.2 /var/named/root.hints

Zapomnite si, da je ta datoteka dinamična in zato tale ni več veljavna. Svojo lahko naredite z digom, kot sem razložil.

```
; <<>> 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
```

7.3 /var/named/zone/127.0.0

Samo osnove; nujen zapis SOA in zapis, ki preslika 127.0.0.1 v `localhost`. Oba sta potrebna. Ni treba, da je v tej datoteki še kaj. Verjetno je ne bo nikoli treba popravljati, razen če se spremeni imenski strežnika ali e-naslov upravitelja DNS.

```
@               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.
```

Če si ogledate naključno namestitev BIND, boste najbrž ugotovili, da manjka `$TTL`, tako kakor tu. Prej ni bila uporabljena in šele različica BIND 8.2 je začela opozarjati na to, da je ni. Svetujem, da v vsako območno datoteko vstavite vrstico `$TTL`.

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

Tu vidimo zapis SOA in potrebni zapis NS. Kot lahko opazite, ima LAND-5 tudi sekundarni imenski strežnik na `ns2.psi.net`. Tako tudi mora biti: *vedno* mora biti zunaj določene strani še sekundarni imenski strežnik, za vsak primer. Vidimo lahko, da je glavni računalnik, `land-5`, ki skrbi za veliko različnih internetnih storitev in ima tudi nekaj alternativnih imen, narejenih s CNAME (druga možnost je uporaba zapisov A).

Iz zapisa SOA je razvidno, da domena izvira iz `land-5.com`, kontaktna oseba je `root@land-5.com`. Tudi `hostmaster` je pogosto uporabljeni naslov za kontaktno osebo. Serijska številka je v formatu yyyymmdd (leto, mesec, dan) s pripeto dnevno serijsko številko; to je verjetno šesta različica te datoteke na dan 20. septembra 1996. Zapomnite si, da se serijska številka *mora* povečevati monotono, in ker je današnja serijska številka omejena na *eno* cifro, mora po 9 popravkih počakati do naslednjega dne, da lahko spet popravlja datoteko. Premislite, ali ne bi uporabili dveh številk.

```
@       IN      SOA     land-5.com. root.land-5.com. (
                        199609206       ; serial, todays date + todays serial #
                        8H              ; refresh, seconds
                        2H              ; retry, seconds
                        4W              ; 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
```

Če si ogledate imenski strežnik land-5, boste ugotovili, da so imena računalnikov v obliki `ws_`*številka*. Po BIND-4 je named namreč začel zahtevati, da se točno držite znakov, ki se smejo uporabiti v imenih. BIND-8 s tem sploh ne bi deloval več, zato sem zamenjal '_' s pomišljajem (-).

Zanimivo je tudi to, da delovne postaje nimajo individualnih imen, temveč so le-ta sestavljena iz predpone in zadnjih delov številk IP. Uporaba takega sistema lahko zelo olajša vzdrževanje, vendar je precej neosebna in utegne vznejevoljiti vaše stranke.

Vidimo tudi, da je `funn.land-5.com` alternativno ime za `land-5.com`, vendar z uporabo zapisa A in ne CNAME. To je dobra politika, kot sem povedal že prej.

7.5 /var/named/zone/206.6.177

Komentarji za to datoteko sledijo.

```
@               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.
```

Obratni vnosi so del nastavitev, ki povzroča največ težav. Uporabljajo se za iskanje imena računalnika, če imate njegovo številko IP. Zgled: ste strežnik IRC in sprejemate povezave odjemalcev IRC. A ker ste norveški strežnik IRC, želite sprejemati le povezave odjemalcev na Norveškem in v drugih skandinavskih državah. Ko se na vas priključi odjemalec, vam lahko vaša knjižnica C pove številko IP računalnika, ki se je priključil, ker je napisana v vsakem paketu podatkov, ki pride po internetu. Zdaj lahko pokličete funkcijo gethostbyaddr, ki preslika številko IP v ime računalnika. Gethostbyaddr bo vprašal strežnik DNS in ta bo prevzel iskanje imena. Predpostavimo, da se je poskusil priključiti odjemalec z ws-177200.land-5.com. Številka IP, ki vam jo pove knjižnica C, je 206.6.177.200. Da bi ugotovili ime računalnika, moramo najti `200.177.6.206.in-addr.arpa`. Strežnik DNS bo najprej našel strežnike za `arpa.`, nato strežnike za `in-addr.arpa.` in sledil po številkah 206 in 6 ter končno našel strežnik za `177.6.206.in-addr.arpa.` na LAND-5. Od njega bo dobil odgovor, da ima za `200.177.6.206.in-addr.arpa` zapis “`PTR ws-177200.land-5.com`”, kar pomeni, da k naslovu `206.6.177.200` sodi ime `ws-177200.land-5.com`. Tako kakor opis poizvedbe za `prep.ai.mit.edu` je tudi ta malce fiktiven.

Vrnimo se k zgledu strežnika IRC. Strežnik sprejema samo povezave iz skandinavskih dežel, to je `*.no`, `*.se` in `*.dk`. Ime `ws-177200.land-5.com` vsekakor ne ustreza nobeni od teh možnosti, zato bo strežnik zavrnil povezavo. Če *ne bi* bilo obratnih preslikav za `206.6.177.200` skozi `in-addr.arpa`, strežnik sploh ne bi mogel najti imena računalnika in vse, kar bi imel za primerjati z `*.no`, `*.se` in `*.dk`, bi bilo `206.2.177.200` - to pa se seveda ne bi ujemalo.

Rekli vam bodo, da so obratne poizvedbe pomembne samo za strežnike, oziroma da sploh niso pomembne. Ni res: številni strežniki ftp, novičarski strežniki, strežniki IRC in celo nekateri http (WWW) se vam *ne bodo* dovolili priključiti iz računalnika, katerega imena ne morejo izvedeti. Zato so obratne poizvedbe v resnici *obvezne*.

8 Vzdrževanje

Skrbite, da bo vse delovalo

Poleg tega, da skrbite, da bo vse delovalo, imate še eno nalogo - vzdržujte datoteko `root.hints` točno. To je najenostavneje z uporabo programa dig. Najprej ga zaženite brez argumentov in dobili boste `root.hints` glede na vaš strežnik, nato pa zaženite dig še z enim od korenskih strežnikov: `dig @korenski-strežnik`. Izhod, ki ga dobite, je nova datoteka root.hints. Shranite ga v datoteko (`dig @e.root-servers.net . ns >root.hints.nov`) in zamenjajte staro datoteko `root.hints` z njo.

Ne pozabite znova zagnati named po tem, ko ste zamenjali datoteko.

Al Longyear mi je poslal skript za vzdrževanje `root.hints`, ki ga lahko poganjate samodejno - v crontab vpišite vnos, ki ga bo pognal vsak mesec, nato pa lahko nanj pozabite. Skript predpostavlja, da imate delujočo elektronsko pošto in da je v vašem računalniku naslov ‘hostmaster’. Da bo ustrezala vašim nastavitvam, jo morate prirediti svojim potrebam.

```
#!/bin/sh
#
# Posodobimo predpomnilniško datoteko imenskega strežnika enkrat na mesec.
# To skripto samodejno zaganja vnos v crontabu.
#
# Izvirnik je napisal Al Longyear
# Za BIND 8 priredil Nicolai Langfeldt
```

```
# David A. Ranch je poročal o različnih možnih napakah
# Preizkus s pingom je predlagal Martin Foster
# Ali named deluje? - preizkus predlagal Erik Bryer.
#
#
(
 echo "To: hostmaster <hostmaster>"
 echo "From: system <root>"

 # Is named up? Check the status of named.
 case `ndc status 2>&1` in
    *'cannot connect to command channel'*)
        echo "named is DOWN. root.hints was NOT updated"
        echo
        exit 0
        ;;
 esac

 PATH=/sbin:/usr/sbin:/bin:/usr/bin:
 export PATH
 # NOTE: V /var/named smejo pisati le ta skripta in ustrezni uporabniki
 # ne bo odprl možnosti za zlonamerno prekinitev delovanja strežnika
 # ali možnosti za vlom v račun superuporabnika - root.
 cd /var/named 2>/dev/null || {
    echo "Subject: Cannot cd to /var/named, error $?"
    echo
    echo "The subject says it all"
    exit 1
 }

 # Smo povezani v internet? Pingajmo strežnik vašega ponudnika
 case `ping -qnc 1 some.machine.net 2>&1` in
   *'100% packet loss'*)
        echo "Subject: root.hints NI posodobljen.  Mreža NE DELUJE."
        echo
        echo "The subject says it all"
        exit 1
        ;;
 esac

 dig @e.root-servers.net . ns >root.hints.new 2> errors

 case `cat root.hints.new` in
   *NOERROR*)
        # Delovalo je
        :;;
   *)
```

```
        echo "Posodobitev datoteke root.hints NI USPELA."
        echo
        echo "Posodabljanje root.hints ni uspelo"
        echo "dig je sporočil naslednje:"
        echo
        cat root.hints.new errors
        exit 1
        ;;
 esac

 echo "Subject: Datoteka root.hints je posodobljena"
 echo
 echo "V datoteki root.hints so zdaj naslednji podatki:"
 echo
 cat root.hints.new

 chown root.root root.hints.new
 chmod 444 root.hints.new
 rm -f root.hints.old errors
 mv root.hints root.hints.old
 mv root.hints.new root.hints
 ndc restart
 echo
 echo "Imenski strežnik je bil vnovič zagnan in zdaj ve za morebitne spremembe."
 echo "Prejšnja datoteka root.hints se zdaj imenuje /var/named/root.hints.old."
) 2>&1 | /usr/lib/sendmail -t
exit 0
```

Morda ste izvedeli, da je datoteka `root.hints` na voljo tudi prek protokola ftp na Internic. Prosim vas, da za posodabljanje `root.hints` uporabljate raje dig, ker je ta možnost prijaznejša tako do interneta kakor do Internica.

9 Prehod z različice 4 na različico 8

To poglavje, ki ga je napisal David E. Smith (dave@bureau42.ml.org), je bilo v izvirniku poglavje o uporabi BIND 8. Da se sklada z novim naslovom, sem ga nekoliko popravil.

Za to temo ni treba veliko besed. Poleg tega, da se uporablja named.conf namesto named.boot, je vse popolnoma enako. BIND 8 ima tudi skripto v perlu, ki vam pretvori stare datoteke v nove. Zgled named.boot (starejša različica) za imenski strežnik s predpomnilnikom:

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

V mapi bind8/src/bin/named (*tu predvidevam, da imate pred seboj izvirno kodo. Če imate paket s prevedenim programom, je skripta verjetno prav tako nekje nameščena, vendar ne vem točno, kje. -ed*) v ukazno vrstico napišite:

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

To naredi named.conf:

```
// 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";
};
```

Deluje za vse, kar je lahko v `named.boot`, čeprav ne doda novih pridobitev in možnosti, ki jih omogoča BIND 8. Tule je bolj popoln `named.conf`, ki deluje enako, vendar nekoliko bolj učinkovito.

```
// To je nastavitvena datoteka za named (BIND 8.1 ali novejši).
// Navadno bo nameščena v /etc/named.conf
// Edina razlika med to in 'že pripravljeno' datoteko (poleg tega
// komentarja :) je to, da sem odkomentiral vrstico 'directory', ker
// sem območne datoteke že imel v /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";
};
```

V distribuciji BIND 8 lahko v mapi bind8/src/bin/named/test najdete to in nekaj izvodov območnih datotek, ki jih večina lahko prekopira in takoj uporabi.

Formati območnih datotek in `root.hints` so enaki, tako kakor tudi ukazi za njihovo posodobljenje.

10 Vprašanja in odgovori

Prosim, preberite to poglavje, preden mi pišete.

1. Moj named zahteva datoteko named.boot

   Berete napačen HOWTO. Oglejte si starejšo različico teg a HOWTO, ki pokriva bind 4, na `<http://www.math.uio.no/~janl/DNS/>`

2. Kako uporabljam DNS izza požarnega zidu?

   Namig: `forward only;`, Verjetno boste potrebovali tudi

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

   v razdelku "options" v datoteki `named.conf`, kot je napisano v zgledu 3 (Imenski strežnik s predpomnilnikom).

3. Kako naj prepričam DNS, da bo periodično obračal razpoložljive naslove za določeno storitev, na primer za www.zelo-zaseden.com, da se obremenitev porazdeli na več računalnikov?

   Naredite več **A** zapisov za www.zelo-zaseden.com in uporabite bind 4.9.3 ali novejšega. Potem bo bind samodejno spreminjal svoje odgovore. Pri prejšnjih različicah binda to *ne bo* delovalo.

4. Rad bi postavil DNS v (zaprtem) intranetu. Kaj moram storiti?

   Izpustite datoteko `root.hints` in napišite le območne datoteke. To tudi pomeni, da vam ne bo treba posodabljati datoteke root.hints.

5. Kako postavim sekundarni (pomožni) imenski strežnik?

   Če ima primarni/glavni strežnik naslov 127.0.0.1, dodajte tale razdelek v named.conf sekundarnega strežnika:

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

V polje `masters` lahko vpišete tudi več glavnih strežnikov, ločenih s ';' (podpičji).

6. Rad bi poganjal bind tudi, ko nisem priključen v internet.

   V zvezi s tem so na voljo štiri rešitve:

   - Specifično za BIND 8, Adam L Rice mi je poslal naslednje e-sporočilo o tem, kako teče DNS brez težav v računalniku z dostopom na klic:

     ```
     Odkril sem novo različico BIND, pri kateri ni več potrebe po
     [prekladanju datotek -ed];.  Poznamo nastavitev "forward" poleg
     nastavitve "forwarders", ki nadzira, kako se uporabljajo. Privzeta
     nastavitev je "forward first", ki najprej povpraša vse strežnike,
     podane v nastavitvi "forwarders", in šele zatem poskuša opraviti
     normalen, dolgotrajen postopek.  Zato klic gethostbyname() traja tako
     dolgo časa, ko povezave ni. Toda če se nastavi "forward only", potem
     BIND obupa, ko ne dobi odgovora od posredovalcev. Tako se
     gethostbyname() hitro vrne. Zato se potem ni treba igrati s
     premeščanjem datotek v /etc in vnovičnim zagonom strežnika.

     V svojem zgledu sem le dodal vrstice

     forward only;
     forwarders { 193.133.58.5; };

     v options { } razdelku moje datoteke named.conf. Deluje zelo
     lepo. Edina pomanjkljivost je, da spremeni zelo napreden kos programja
     DNS v neumen predpomnilnik. Ponekod bi želel uporabljati samo neumen
     predpomnilnik DNS, toda ni videti, da bi bilo kaj takega na voljo za
     Linux.
     ```

   - To pismo sem prejel od Iana Clarka, <ic@deakin.edu.au>. V njem opisuje, kako je kos temu.

     ```
     Named tukaj poganjam v računalniku za 'Maškarado'. Imam dve datoteki
     root.hints, ena se imenuje root.hints.real in so v njej resnični podatki,
     druga pa je root.hints.fake in so v njej...

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

     Ko se odklopim iz interneta, prekopiram datoteko root.hints.false v root.hints in
     vnovič zaženem named.

     Ko pa se priključim v internet, prekopiram root.hints.real v root.hints in vnovič
     zaženem named.

     Oboje postorita skripti ip-down in ip-up.

     Prvič, ko opravim poizvedbo o domeni, named nima podatkov o njej in v 'messages'
     napiše...
     ```

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

A to me ne moti.

Kar se mene tiče, vsekakor odlično deluje. Imenski strežnik lahko uporabljam za
krajevne računalnike, ko niso priključeni v internet, in sicer brez poteka časa za
zunanje domene; ko sem priključen v internet, pa delujejo poizvedbe normalno.
```

Peter Denison je menil, da Ian ne gre dovolj daleč, in je napisal:

```
Ko povezan)     Ponudi predpomnjene (in LAN) vnose takoj.
                Nepredpomnjene vnose pa posreduj imenskemu strežniku ISP
Ko nepovezan)   Ponudi LAN vnose takoj
                Za druge poizvedbe **takoj** vrni napako.

Kombinacija spreminjanja datoteke root.cache in posredovanja poizvedb ne
deluje.

Zato sem vzpostavil (po debati s krajevno LUG - Linux Users Group) dva
Nameda, kot je dano:

named-povezan:  posreduje imenskemu strežniku ISP
                primarni za območje lokalnet
                primarni za obratno območje lokalnet (1.168.192.in-addr.arpa)
                primarni za 0.0.127.in-addr.arpa
                posluša na vratih 60053

named-zaprt:    brez posredovanja
                datoteka "nepravi" root.cache
                sekundarni za 3 območja localnet (primarni je 127.0.0.1:60053)
                posluša na vratih 61053

Skupaj s preusmerjanjem vrat, ki preusmeri vrata 53 na 61053, ko sem povezan,
in na 60053, ko sem nepovezan. (Uporabljam novi paket netfilter pod 2.3.18,
toda stari (ipchains) mehanizem bi tudi moral delovati.)

Opozorilo: to ne bo delovalo čisto takoj po priključitvi, ker hrošček v BIND
8.2, ki sem ga že posredoval razvijalcem, prepreči delovanje sekundarnega,
če je primarni na istem IP (četudi na drugih vratih). Gre za enostaven
popravek, ki bo moral biti kmalu dodan.
```

- Od Karla-Maxa Wagnerja sem prejel tudi informacijo, kako bind deluje vzajemno z NFS in portmapperjem v računalniku, ki večinoma ni priključen v internet:

```
Navadno poganjam lasten named v vseh računalnikih, ki so le občasno priključeni v
internet prek modema. Imenski strežnik se vede le kot predpomnilnik, nima območja
pristojnosti in za vse poizvedbe sprašuje strežnike iz datoteke root.hints. Kot je v
navadi za Slackware, se tudi zažene pred nfsd in mountd.

Z enim od mojih računalnikov (Libretto 30 notebook) sem imel težave - včasih sem ga
lahko dosegel iz drugega sistema, priključenega v moj LAN, večinoma pa to ni
delovalo. Ista težava se je pokazala ne glede na to, ali sem uporabljal PLIP,
```

```
omrežno kartico PCMCIA ali PPP po zaporedni povezavi.

Z nekaj ugibanja in poskušanja sem ugotovil, da je named nekako pokvaril postopek
prijave, ki ga nfsd in mountd izvedeta s portmapperjem pri zagonu (navadno takrat,
ko zaženem računalnik). Zagon named po nfsd in mountd je to težavo popolnoma
odpravil.

Ker tako spremenjena zagonska procedura nima stranskih učinkov, vam priporočam, da
jo popravite in tako preprečite morebitne težave.
```

- Zadnja možnost so informacije HOWTO o povezavah na klic *Ask Mr. DNS at* `<http://www.acmebw.com/askmrdns/\#linux-dialup>`, ki pa govorijo o bind 4 in jih morate zato prilagoditi bindu 8.

7. Kje ima named svoj predpomnilnik? Lahko kako vplivam na njegovo velikost?

   Predpomnilnik je v celoti shranjen v pomnilniku in ni *nikoli* zapisan na disk, zato vsakič, ko ubijete named, izgubite njegovo vsebino. Predpomnilnika *ni mogoče* upravljati. Named ga uredi po nekih preprostih pravilih in to je to. Njegove vsebine in velikosti ne morete nikakor spremeniti ali omejiti. Če želite, lahko to "popravite" s hekanjem named, vendar vam tega ne priporočam.

8. Ali named shranjuje predpomnilnik med vnovičnimi zagoni? Ga lahko pripravim do tega, da ga shrani?

   Named svojega predpomnilnika *ne* shranjuje, ko umre. To pomeni, da mora predpomnilnik zgraditi na novo vsakič, ko ga ubijete in vnovič zaženete. *Ni načina*, da bi named shranil svoj predpomnilnik v datoteko. Če želite, lahko to "popravite" s hekanjem named, vendar vam tega ne priporočam.

9. Kako lahko dobim domeno? Želel bi svojo domeno (na primer) `linux-rules.net`. Kako bi jo lahko pripisal nase?

   Povprašajte svojega ponudnika internetnih storitev, on vam bo zagotovo znal pomagati. Vendar si zapomnite, da je po svetu večinoma treba za domeno plačati.

10. Kako lahko zavarujem svoj DNS ? Kako uredim ločena DNS?

    Oboje sta napredni temi in si o njima lahko preberete v `<http://www.etherboy.com/dns/chrootdns.html>`. O tem tu ne bom več razglabljal.

11 Kako postati veliki upravitelj DNS

Dokumentacija in orodja

Prava dokumentacija je na voljo v internetu in v tiskani obliki. Nekaj je morate prebrati in naredili boste korak od majhnega k velikemu upravitelju. V tiskani obliki sem napisal *The Concise Guide to DNS and BIND* (avtor: Nicolai Langfeldt), založil Que (ISDN 0-7897-2273-9). Knjiga je podobno temu HOWTO, le da je v njej več podrobnosti in več vsega. Tipična knjiga pa je *DNS in BIND*, avtorja sta C. Liu in P. Albitz iz O'Reilly & Associates, Sebastopol, CA, ISBN 0-937175-82-X. Tudi ta je odlična. V 3. izdaji pokriva tako BIND8 kakor BIND 4. Poglavje o DNS je tudi v *Upravljanje omrežja TCP/IP*, katerega avtor je Craig Hunt iz O'Reilly..., ISBN 0-937175-82-X. Še eno nujno branje za dobro upravljanje DNS (oziroma česarkoli) je *Zen in umetnost vzdrževanja motornih koles* izpod peresa Roberta M. Pirsiga :-) Dosegljivo je kot ISBN 0688052304 in druge.

V internetu boste našli marsikaj dobrega na `<http://www.dns.net/dnsrd/>` (Mapa z zapisi o DNS), `<http://www.isc.org/bind.html>`; FAQ in v referenčnem priročniku (BOG; Bind Operations Guide), pa tudi v člankih in definicijah protokola in zvijač DNS (skoraj, če ne popolnoma vse RFC, navedene spodaj, najdete tudi v distribuciji binda). Večine jih nisem prebral, zato tudi nisem velik upravitelj DNS. Arnt Gulbrandsen pa je prebral BOG in je čisto vzhičen :-). Deluje tudi novičarska skupina za DNS, `<news:comp.protocols.tcp-ip.domains>`, poleg tega je v internetu tudi veliko število RFC o DNS. Najpomembnejši so verjetno spodaj nanizani. Tiste, ki imajo BCP (ang. Best Current Practice), vam *toplo priporočam.*

RFC 2671

P. Vixie, *Extension Mechanisms for DNS (EDNS0)* avgust 1999.

RFC 2317

, BCP 20, H. Eidnes et. al. *Classless IN-ADDR.ARPA delegation*, marec 1998. Govori o CIDR ali brezrazrednih obratnih poizvedbah.

RFC 2308

, M. Andrews, *Negative Caching of DNS Queries*, marech 1998. O negativnem predpomnenju in o nastavitvi $TTL območnih datotek.

RFC 2219

, BCP 17, M. Hamilton in R. Wright, *Use of DNS Aliases for Network Services*, oktober 1997. O uporabi CNAME.

RFC 2182

, BCP 16, R. Elz et. al., *Selection and Operation of Secondary DNS Servers*, julij 1997.

RFC 2052

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

RFC 1918

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

RFC 1912

D. Barr, *Common DNS Operational and Configuration Errors*, 28. februar 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*, 3. november 1994.

RFC 1712

C. Farrell, M. Schulze, S. Pleitner, D. Baldoni, *DNS Encoding of Geographical Location*, 1. november 1994.

RFC 1183

R. Ullmann, P. Mockapetris, L. Mamakos, C. Everhart, *New DNS RR Definitions*, 8. oktober 1990.

RFC 1035

P. Mockapetris, *Domain names - implementation and specification*, 1. november 1987.

RFC 1034

P. Mockapetris, *Domain names - concepts and facilities*, 1. november 1987.

RFC 1033

M. Lottor, *Domain administrators operations guide*, 1. november 1987.

RFC 1032

M. Stahl, *Domain administrators guide*, 1. november 1987.

RFC 974

C. Partridge, *Mail routing and the domain system*, 1. januar 1986.