

Domain Name Service (DNS)

Sejak dimulainya proyek ARPAnet sistem internet memiliki nama dan juga alamat IP. Pada sistem UNIX, sebagaimana juga copy dari implementasi BSD, menggunakan file */etc/hosts* untuk konversi nama host dan pengalamatan. File ini berisi daftar dari alamat IP dan berhubungan dengan nama host, satu nama per baris.

Jelas tidak mungkin file */etc/hosts* digunakan untuk memetakan internet dengan lengkap. Walaupun anda memiliki ruang disk yang besar, sejumlah update akan membebani jaringan anda. Solusinya adalah database terdistribusi, Domain Name System, atau DNS. Implementasi DNS umumnya menggunakan BIND, Berkeley Internet Name Domain. Bind merupakan bagian dari BSD. Daemon yang mengemban resolusi tersebut di sebut named (di ucapkan “name-dee”).

DNS memberikan informasi yang diperlukan untuk menghubungkan sistem remote yang berada di Resource Records, atau RRs.

- *A (Address) records* mentranslasi nama host ke alamat IP. Contohnya, satu catatan A mengatakan bahwa www.FreeBSD.org memiliki alamat IP 216.136.204.117. Nama yang dispesifikasikan di catatan A disebut canonical name dari interface, dan harusnya merupakan acuan bagi catatan PTR.
- *PTR (Pointer) records* mentranslasikan alamat IP ke nama. Proses ini disebut reverse lookup.
- *MX (Mail Exchange) records* menspesifikasikan alamat IP dari mail server suatu domain.
- *SOA (Start Of Authority) records* mendefinisikan zona-zona,
- *NS (Name Server) records* mendeskripsikan name servers untuk suatu zona.
- *HINFO (Hardware Information) records* mendeskripsikan hardware dan software yang sedang berjalan pada sistem.
- *CNAME (Canonical Name) records* mendeskripsikan nama alternatif untuk suatu sistem.

FreeBSD mengizinkan anda untuk menggunakan file */etc/hosts* dan DNS. Salah satu alasannya adalah keharusan untuk memiliki resolusi nama dari localhost pada saat waktu startup: adanya masalah ayam-dan-telur dengan mounting file sistem NFS sebelum daemon named berjalan.

Domains dan zones

Di internet, sebuah domain adalah suatu group dari nama-nama yang diakhiri dengan nama domain yang akurat. Seperti nama file, ada dua macam nama-nama domain:

- *fully qualified domain name (FQDN)* diakhiri dengan titik (.). Nama domain ini berhubungan ke domain induk . (titik tunggal).
- Relative domain name berhubungan ke domain sekarang. Anda akan melihatnya pada file konfigurasi.

Ketika anda menulis nama domain, anda bermaksud menjadikannya *fully qualified*. Namun jika anda menulisnya tanpa akhiran titik, DNS acapkali menambahkannya sendiri. Contohnya, jika anda mennspesikasikan nama seperti *freebie.example.org*, DNS tidak akan menemukan nama yang *fully qualified*: salah pengejaan dari *freebie.example.org*.. Sebagai hasilnya, DNS akan mencari nama *freebie.example.org.example.org*. Tentu saja tidak akan ditemukan, namun bisa saja ditemukan dengan waktu percobaan yang lama. Jadi ketika menulis file konfigurasi DNS selalu akhiri dengan titik (full stop) pada akhir nama domain yang *fully qualified*.

Zones

Suatu zona merupakan hal yang sama dengan domain: subset dari nama DNS yang dipelihara oleh spesifik name server, pada istilah DNS, name server adalah otoritatif untuk sebuah zona. Perbedaannya adalah pada cara penggunaannya. Bagaimanapun ada satu pengecualian, biasanya suatu subdomain akan memiliki name server yang berbeda. Subdomain ini bagian dari sebuah domain, namun tidak pada suatu zona.

Contohnya, pada referensi jaringan kita, name server *freebie* dan *presto* adalah otoritatif untuk *example.org*. Pemilik domain mungkin saja memberikan izin untuk seseorang, mungkin di negara lain, untuk menjalankan subdomain *china.example.org*, dengan name server *beijing.china.example.org* dan *xianggang.china.example.org*. Karena adanya name server yang berbeda, ada dua zona: *freebie.example.org* akan di otoritatif untuk zona *example.org*, namun tidak untuk *china.example.org*, *beijing.china.example.org* dan *xianggang.china.example.org* akan di otoritatif untuk zona *china.example.org*, namun tidak untuk *example.org*.

Setting name server

DNS service di suplay oleh daemon name, disebut *named*. *Named* dapat dijalankan dalam beberapa mode. Pada bab ini, kita akan konsentrasi untuk setting jaringan referensi tadi. Jika anda ingin mengetahui lebih jauh, silahkan lihat dokumen berikut:

- BIND Online Documentation, distribusi sumber pada direktori */usr/src/contrib/bind/doc/html/index.html*.
- TCP/IP Network Administration, oleh Craig Hunt (O'Reilly).
- DNS and BIND, oleh Paul Albitz dan Cricket Liu (O'Reilly).

Penggunaan Pasif DNS

Tidak setiap sistem perlu menjalankan daemon name'nya sendiri. Jika anda memiliki mesin lain dalam jaringan yang sama, anda dapat mengirim permintaan ke mesin tersebut. Contohnya, dalam jaringan referensi, *freebie* dan *presto* mungkin menjalankan name server. Namun tidak untuk mesin lainnya yaitu *bumble* dan *wait*, keduanya tampaknya mesin yang lambat. Sehingga anda dapat mengkonfigurasi kedua mesin tadi untuk menggunakan name server pada dua mesin lainnya.

Untuk melakukan hal tersebut, pastikan *named* tidak tercantum pada file */etc/rc.conf*, dan buatlah file */etc/resolv.conf* dengan isi berikut :

```
domain example.org
nameserver 223.147.37.1      #freebie
nameserver 223.147.37.2      #presto
```

Spesifikasikan alamat IP, bukan nama host nameserver. Ini juga merupakan masalah klasik *chicken-and-egg*: anda tidak dapat mengakses nameserver sampai anda tahu alamat IP'nya.

Maka mesin ini akan mengirim seluruh permintaan name server ke *freebie* atau *presto*.

Name server pada sistem standalone

Jika anda memiliki mesin tunggal yang terhubung ke internet, dan mesin anda merupakan bagian dari zona ISP, anda dapat menggunakan metoda *resolv.conf*. Ini kondisi yang khas jika anda menggunakan koneksi PPP atau DSL. Bagaimanapun hal tersebut bukan merupakan ide yang bagus. Setiap lookup memerlukan hubungan ke luar, sehingga anda melakukan lookup yang sama terus menerus. DNS menjawab masalah tersebut: menyimpan informasi secara lokal. Anda dapat melakukan ini dengan *caching-only name server*. Seperti namanya *caching-only name server* tidak memiliki informasi sendiri, namun menyimpan hasil dari permintaan ke sistem lain, jadi jika program meminta hal yang sama- dimana sering terjadi-akan memberikan hasil yang lebih cepat. Setup *caching-only* seperti berikut ini :

- rename atau hapus */etc/resolv.conf* dan buat baru lagi dengan isi

```
nameserver 127.0.0.1      local name server
```
- tambahkan baris ini di file */etc/rc.conf*:

```
named_enable="YES"      #jalankan named, server DNS
```

- buat file `/etc/namedb/localhost.rev` yang berisi:


```
$TTL 1d
@ IN SOA @host@. root.@host@. (
@date@ ; Serial
1h ; Refresh
5m ; Retry
100d ; Expire
1h ) ; Negative cache
IN NS @host@.
1 IN PTR localhost.@domain@.
```

Kita akan membahas artinya pada seksi berikut. Untuk membuatnya, anda dapat memulainya dengan file `/etc/namedb/PROTO.localhost.rev`, dimana berisi template untuk file ini. Ganti `@host@` dengan FQDN host anda (pada contoh ini `freebie.example.org`), `@date@` (nomor serial) dengan form waktu `yyymmddxx`, dimana `xx` adalah integer kecil seperti 01, dan `@domain@` dengan `example.org`. Pastikan FQDN diakhiri titik. Cara alternatif anda dapat menjalankan script `/etc/namedb/make-localhost`.

- Edit file `/etc/namedb/named.conf` dengan isi:


```
options {
    directory "/etc/namedb";
    forwarders {
        139.130.237.3; 139.130.237.17;
    };
    zone "0.0.127.in-addr.arpa" {
        type master;
        file "localhost.rev";
    };
};
```

file `/etc/namedb/named.conf` seharusnya telah siap digunakan pada sistem anda. Alamat 139.130.237.3 dan 139.130.237.17 adalah alamat IP nameserver dari ISP. Baris `forwarders` dapat berisi hingga 10 alamat nameserver.

- Jalankan `named`:

```
#ndc start
```

Name server pada jaringan pemakai-akhir

Tentunya, caching-only name server yang sederhana tidak akan bekerja jika anda memiliki domain sendiri. Faktanya, kebanyakan yang mengautoritaskan alokasi domain names tidak akan membiarkan pendaftaran domain internet kecuali jika anda menspesifikasikan dua fungsional nameserver, dan mereka akan memeriksa sebelum pendaftaran di proses. Pada bagian ini, kita akan melihat apa yang anda lakukan untuk menjalankan name server yang “real”. Hal pertama yang perlu kita lakukan adalah membuat file zona untuk zona `example.org`. Kita simpan seluruh file zona tersebut dan lainnya tersebut pada direktori `/etc/namedb` dan `/etc/namedb/db.example.org`.

SOA record

Hal pertama yang kita perlukan adalah mencatat deskripsi *Start of Authority*. Ini mendefinisikan zona baru. Tulis :

```
$TTL 1d
example.org. IN SOA freebie.example.org. grog.example.org. (
    2003031801 ; Serial (date, 2 digits version of day)
    1d ; refresh
    2h ; retry
    100d ; expire
    1h ) ; negative cache expiry
```

Baris pertama, \$TTL 1d, menspesifikasikan lamanya waktu remote name server akan di cache dari zone ini. Selama waktu tersebut DNS tidak melakukan lookup lainnya. Pada versi BIND lama, nilai ini disimpan pada bagian akhir dari SOA berikut.

Nama pada bagian sisi kiri adalah nama zona. Kata kunci IN berarti Internet, dengan kata lain Internet Protocols. Software BIND mendukung beberapa tipe jaringan, dimana kini sudah dilupakan. Kata kunci SOA mendefinisikan tipe catatan. freebie.example.org adalah master name server.

Bagian lainnya, *grog.example.org* adalah alamat mail dari administrator DNS. Serial number mengidentifikasi versi dari konfigurasi zona. Remote name server awalnya mengambil catatan SOA dan memeriksa apakah serial number telah bertambah sebelum memutuskan apakah mengakses zona, dimana mungkin saja telah menjadi besar. Pastikan anda menambah bagian ini setiap saat mengedit file tersebut. Jika tidak hasil update anda tidak akan propagasi ke name server lainnya. Ide yang baik untuk menggunakan format berupa tanggal, format memberikan empat digit untuk tahun, dua digit untuk bulan, dua digit untuk hari, dan dua digit untuk nomor modifikasi pada tanggal tertentu. Serial number pada contoh ini menunjukkan modifikasi kedua untuk konfigurasi zona pada 18 Maret 2003.

Parameter sisanya menjelaskan karakteristik timeout dari suatu zona. Gunakan nilai pada contoh hingga anda mempunyai alasan yang bagus untuk mengubahnya. Format data untuk catatannya memerlukan semua waktu untuk di spesifikasikan dalam detik dan pada versi BIND sebelumnya, ini hanya pilihan yang anda punya. Pada versi BIND sekarang, anda dapat menggunakan faktor skala seperti d untuk hari dan h untuk jam pada file konfigurasi. Named mengubahnya ke detik sebelum transmisi dilakukan.

- *Refresh time* adalah waktu setelah remote name server akan memeriksa apakah konfigurasi zona telah berubah. 1 day adalah cukup beralasan sampai anda mengubah konfigurasi beberapa kali per hari.

- *Retry time* adalah waktu untuk menunggu jika terjadi kegagalan dalam suatu zona.
- *expire time* adalah waktu setelah slave name server akan meng'drop informasi mengenai suatu zona jika tidak dapat di muat dari master name server. Mungkin anda ingin membuat nilai tersebut besar.
- Pada versi BIND sebelumnya, bagian akhir adalah *minimum time to live*. Kini parameter \$TTL yang mengatur nilai tersebut, dan parameter terakhir menspesifikasikan waktu negative caching. Jika otoritatif name server melaporkan bahwa record nya tak ada, dan mengembalikan SOA record untuk mengindikasikan berasal dari otoritatifnya. Lokal name server menjaga informasi ini untuk perioda waktu tertentu yang ditentukan oleh bagian dari SOA record yang dikembalikan dan tidak akan mencoba lagi hingga waktu tersebut berakhir. Jika remote hostmaster merubah konfigurasi DNS, sangat beralasan untuk menjaga waktu negative cache sekitar satu jam.

A records

Hal mutlak yang perlu diketahui adalah alamat IP dari sistem pada jaringan. Pada zona example.org, definisikan A records seperti ini:

```
localhost    IN    A    127.0.0.1          local machine, via loopback interface
freebie      IN    A    223.147.37.1
presto       IN    A    223.147.37.2
bumble       IN    A    223.147.37.3
wait         IN    A    223.147.37.4
gw           IN    A    223.147.37.5
```

Kita taruh A records lebih jauh pada akhir baris, karena mereka biasanya paling banyak. Hal itu membuat file tersebut lebih mudah dibaca jika kita menyimpan setelah entri group terpendek.

NS records

DNS menggunakan tipe record khusus untuk mengatakan dimana name server berada. Pada kasus kita, kita menjalankan name server pada freebie dan presto. Kita tulis :

```
IN NS freebie.example.org.
IN NS presto.example.org.
```

Konfigurasi sederhana tersebut seharusnya sudah dapat berjalan dengan baik, namun kita akan melakukan sedikit perubahan pada seksi selanjutnya.

Nicknames

Kita menjalankan banyak layanan pada jaringan referensi, biasanya web server dan ftp server. Suatu mesin web server disebut www, ftp server disebut ftp, dan name server disebut ns. Namun keduanya berjalan pada mesin dengan nama berbeda. Jadi apa yang kita lakukan ? Kita beri mereka nickname mesin:

```
www IN CNAME freebie
ftp  IN CNAME presto
```

Kita tidak dapat melakukannya terhadap NS, karena DNS akan complain jika anda membuat ns sebuah CNAME. Jika anda menggunakan CNAME records untuk mendefinisikan name server anda, remote system harus melakukan dua lookup untuk menemukan alamat dari name server, satu untuk me'retrieve CNAME dan satu lagi untuk mendapatkan korespon A record untuk CNAME. Definisikan A record baru untuk mereka:

```
                IN NS ns
                IN NS ns1
ns      IN      A      223.147.37.1
ns1    IN      A      223.147.37.2
```

Kita menggunakan relatif domain name pada contoh tadi.

MX records

Anda dapat mengirim surat ke host yang terdaftar dalam A record, namun hal itu bukan ide yang bagus. Bahkan anda harus memiliki setidaknya 2 MX records untuk mengatakan SMTP apa yang harus dengan mail untuk domain anda. Metoda ini mempunyai keuntungan tambahan: mengizinkan anda untuk mengganti nama individual mesin tanpa harus mengubah ID pengguna mail. Kita akan mengasumsikan bahwa seluruh mail dikirim ke user@example.org. Sebagai tambahan, kita gunakan mail server ISP mail.example.net sebagai backup jika terjadi down pada mail server. Hasil dari MX record seperti berikut:

```
IN MX 50 bumble.example.org.
IN MX 100 mail.example.net.
```

Nomor 50 dan 100 disebut preferences. Secara teori anda dapat membuatnya 1 dan 2, kecuali . Suatu agen mail transfer mengirim mail ke sistem dengan preferences terendah kecuali jika tidak merespon-maka akan digunakan MX record dengan preference terendah selanjutnya, dan begitu seterusnya.

HINFO records

Akhirnya, anda mungkin ingin mengatakan di internet tentang hardware anda dan ini merupakan sistem operasi handal yang anda jalankan. Anda dapat melakukannya dengan menggunakan HINFO record:

```
freebie      IN HINFO "Pentium/133" "FreeBSD 4.0-CURRENT (4.4BSD)"
presto       IN HINFO "Pentium II /233" "FreeBSD 3.2 (4.4BSD)"
bumble       IN HINFO "Pentium/133" "SCO OpenServer"
wait         IN HINFO "Pentium Pro 266" "Microsoft Windows 95%"
gw           IN HINFO "486/33" "FreeBSD 3.2 (4.4BSD)"
```

Tentunya dengan memberikan informasi yang detail tentang hardware anda juga akan membantu *crackers* untuk memilih tools yang akan mereka gunakan untuk menjebol sistem anda. Jika anda khawatir resiko tersebut, jangan gunakan HINFO. Melihat HINFO masih merupakan pengecualian.

Menggabungkan semuanya

File konfigurasi */etc/namedb/db.example.org* seperti berikut:

```
; Definition of zone example.org
$TTL 1d
example.org. IN SOA freebie.example.org. grog.example.org. (
                2003031801 ; Serial (date, 2 digits version of day)
                1d ; refresh
                2h ; retry
                100d ; expire
                1h ) ; negative cache expiry

; name servers
                IN NS ns
                IN NS ns1

; MX records
                IN MX 50 bumble.example.org.
                IN MX 100 mail.example.net.
ns      IN      A 223.147.37.1
ns1     IN      A 223.147.37.2

; Hosts
localhost IN      A 127.0.0.1
freebie   IN      A 223.147.37.1
presto    IN      A 223.147.37.2
bumble    IN      A 223.147.37.3
wait      IN      A 223.147.37.4
gw        IN      A 223.147.37.5

; nicknames
www IN CNAME freebie
ftp IN CNAME presto

; System information
freebie   IN      HINFO "Pentium/133" "FreeBSD 4.0-CURRENT (4.4BSD)"
```

```

presto      IN      HINFO "Pentium II/233" "FreeBSD 3.2 (4.4BSD)"
bumble      IN      HINFO "Pentium/133" "SCO OpenServer"
wait        IN      HINFO "Pentium Pro 266" "Microsoft Windows 95%"
gw          IN      HINFO "486/33" "FreeBSD 3.2 (4.4BSD)"

```

Baris komentar di mulai dengan tanda ; , dan jangan dengan #. Juga kita dapat mengatur MX records dan A records untuk name server. Jika kita tempatkan MX records dibawah A records untuk name servers, mereka akan mengacu ke *ns1.example.org*.

Itulah seluruh informasi yang kita perlukan untuk zona kita example.org.

Reverse lookup

Reverse lookup diperlukan untuk mencari nama yang berhubungan dengan alamat IP tertentu. Untuk me'resolve nama, kita memerlukan zone lain. Artinya ada tambahan file baru, yang kita beri nama */etc/namedb/example-reverse*, dan tidak serumit file forward:

```

$TTL 1d
@ IN SOA freebie.example.org. grog.example.org. (
        2003022601 ; Serial (date, 2 digits version of day)
        1d ; refresh
        2h ; retry
        100d ; expire
        2h ) ; negative cache
    IN NS ns.example.org.
    IN NS ns1.example.org.

1 IN PTR freebie.example.org.
2 IN PTR presto.example.org.
3 IN PTR bumble.example.org.
4 IN PTR wait.example.org.
5 IN PTR gw.example.org.

```

Pada kasus ini, catatan SOA identik pada file */etc/namedb/db.example.org*, dengan dua pengecualian: nama zone pada awal baris, kita memiliki simbol @ dan nomor serial yang berbeda -- normalnya anda tidak perlu meng'update domain reverse lookup. Simbol @ menyatakan nama dari zona, pada *37.147.223.in-addr.arpa..* Nanti kita akan lihat bagaimana kerjanya ketika membuat file */etc/named/named.root* berikut. Kita juga menggunakan entri name server yang sama.

Catatan PTR (*reverse lookup*) hanya menspesifikasikan digit terakhir (bagian host) dari alamat IP. Nama host pada akhir baris merupakan fully qualified form, karena termasuk dalam zona lain. Sebagai contoh :

```

4.37.147.223.in-addr.arpa. IN PTR wait.example.org.

```

File named.conf

file named.conf akan mengarahkan name server untuk menggunakan file apa saja, telah tersedia kerangka file named.conf di `/etc/namedb/named.conf`. Dengan komentar dihapus akan terlihat seperti berikut :

```
options {
    directory "/etc/namedb";
    forwarders {
        127.0.0.1;
    };
};

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

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

zone "domain.com" {
    type slave;
    file "s/domain.com.bak";
    masters {
        192.168.1.1;
    };
};

zone "0.168.192.in-addr.arpa" {
    type slave;
    file "s/0.168.192.in-addr.arpa.bak";
    masters {
        192.168.1.1;
    };
};
```

Tiap entri terdiri dari keyword yang diikuti oleh tanda kurung ({}). Entri tersebut mempunyai arti :

- `directory` mengatakan dimana named mencari file-file konfigurasi.
- Zone pertama adalah top-level domain, `.` mengarahkan named untuk mencari `named.root` pada direktori konfigurasi. `named.root` berisi alamat IP dari 13 top-level name server.
- Entri `0.0.127.IN-ADDR.ARPA` merupakan reverse lookup untuk alamat `localhost`.
- Entri zona untuk `domain.com` dan `0.168.192.in-addr.arpa` sebagai slave name server. Suatu slave name server akan mengarahkan seluruh permintaan ke master name server. Pada versi DNS sebelumnya, slave name server di sebut sebagai secondary name server, dan master name server disebut sebagai

primary name server.

```
zone "example.org" {
    type master;
    file "db.example.org";
};

zone "37.147.223.in-addr.arpa" {
    type master;
    file "example-reverse";
};
```

Jika telah selesai, kita dapat menjalankan named dengan perintah `ndc`:

```
#ndc start
new pid is 86183

#ndc reload
Reload initiated.
```

Restarting atau starting name server tidak berarti named akan berjalan dengan lancar. Jika anda membuat kesalahan pada file konfigurasi, kemungkinan named tidak akan berjalan dengan semestinya. Mungkin saja named berjalan, namun akan menolak untuk memuat zona tertentu. Sebaiknya setelah menjalankan named, anda harus melihat file log di `/var/log/messages`.

Slave name servers

Apa yang akan terjadi jika satu name server karena suatu hal tidak berfungsi, tentunya dapat menyebabkan masalah besar. Ini merupakan salah satu alasan mengapa registrar domain mengharuskan pendfatar domain sekurangnyanya memiliki dua name server sebelum mereka mendaftarkan suatu domain.

Jika anda menjalankan beberapa name server, tidak menjadi masalah hasil jawaban berasal dari mana. Jadi apa perbedaan antara master dan slave name server? Ini murni : suatu name server memuat datanya dari file konfigurasi yang anda buat tadi. Slave name server memuat informasi datanya dari master name server yang sedang aktif. Ini akan lebih memudahkan pengertian untuk menkonfigurasi slave name server, segala yang kita perlukan ada di `/etc/namedb/named.conf`:

```
zone "." {
    type hint;
    file "named.root";
};

zone "example.org" {
    type slave;
    file "backup.example.org";
    masters {
```

```

        223.147.37.1;
    };
};

zone "37.147.223.in-addr.arpa" {
    type slave;
    file "backup.example-reverse";
    masters {
        223.147.37.1;
    };
};

zone "0.0.127.in-addr.arpa" {
    type master;
    file "localhost.rev";
};

```

Walaupun slave name server, namun tidak untuk pemetaan localhost reverse, entri terakhir masih merupakan master.

Alamat numerik untuk freebie.example.org, name server dimana zona dimuat. Kita menggunakan alamat numerik karena name server memerlukan alamat sebelum melakukan resolusi. Anda dapat menspesifikasikan beberapa name server jika anda inginkan. Backup file adalah nama file dimana informasi zona disimpan, diperlukan jika suatu saat name server di restart namun master name server tidak dapat di akses.

Delegasi Zones

Pada contoh sebelumnya, kita menkonfigurasi name server untuk zona tunggal dengan no subzones. Pada bagian ini kita akan membuat file-file konfigurasi untuk subzone ini dan melihat bagaimana menghubungkannya ke zona induk, prosesnya di sebut delegasi.

China.example.org

Untuk subdomain *china.example.org*, pertimbangan yang sama di terapkan seperti pada contoh sebelumnya: kita memiliki domain tanpa subdomain. Hanya nama dan alamat yang berubah.

Pada contoh berikut, diasumsikan bahwa china.example.org mempunyai dua name server yaitu *beijing.china.example.org* dan *xianggang.china.example.org*. Isi File konfigurasi */etc/namedb/db.china.example.org*:

```

; Definition of zone china.example.org
$TTL 1d
@      IN SOA beijing.china.example.org. zhang.china.example.org. (
        2001061701 ; Serial (date, 2 digits version of day)
        1d ; refresh
        2h ; retry
        100d ; expire
        2h ) ; negative cache

; name servers
        IN NS ns
        IN NS ns1
ns     INA 223.169.23.1
ns1   INA 223.169.23.2
; MX records
        IN MX 50 xianggang.china.example.org.
        IN MX 70 bumble.example.org.
        IN MX 100 mail.example.net.

; Hosts
beijing     IN      A 223.169.23.1
xianggang  IN      A 223.169.23.2
shanghai   IN      A 223.169.23.3
guangzhou  IN      A 223.169.23.4
gw         IN      A 223.169.23.5
; nicknames
www        IN      CNAME shanghai
ftp       IN      CNAME shanghai

```

kemudian */etc/namedb/china-reverse:*

```

; Definition of zone china.example.org
@ IN SOA beijing.china.example.org. zhang.china.example.org. (
        1997090501 ; Serial (date, 2 digits version of day)
        86400 ; refresh (1 day)
        7200 ; retry (2 hours)
        8640000 ; expire (100 days)
        86400 ) ; minimum (1 day)

; name servers
        IN NS ns.china.example.org.
        IN NS ns1.china.example.org.

; Hosts
1 IN PTR beijing
2 IN PTR xianggang
3 IN PTR shanghai
4 IN PTR guangzhou
5 IN PTR gw

```

dan yang terakhir */etc/namedb/named.conf*:

```
zone "." {
    type hint;
    file "named.root";
};
zone "0.0.127.IN-ADDR.ARPA" {
    type master;
    file "localhost.rev";
};
zone "china.example.org" {
    type master;
    file "db.china.example.org";
};
zone "23.169.233.IN-ADDR.ARPA" {
    type master;
    file "china-reverse";
};
```

File-file ini terlihat hampir sama untuk *example.org*. Perbedaan akan terlihat pada konfigurasi untuk *example.org*, tidak untuk *china.example.org*.

Example.org dengan delegasi

Apa yang dilakukan name server *example.org* agar mengetahui *china.example.org*? Untuk remote name server mencari *china.example.org*, name server menuju ke *example.org*, sehingga domain parent harus menjaga informasi yang cukup untuk menemukan domain child. Proses ini disebut delegasi. Name server induk menjaga NS records (“*delegation records*”) dan A records (“*glue records*”) untuk child zone. Ide yang bagus jika name server untuk *example.org* menjaga secondary name server untuk *china*: sehingga kita bisa menyimpan lookup ke master name server untuk *china.example.org* tiap saat. Untuk melakukan hal tersebut, tambahkan baris ke */etc/namedb/named.conf*:

```
zone "china.example.org" {
    type slave;
    file "backup.china";
    masters {
        223.169.23.1;
        223.169.23.2;
    };
};

zone "23.169.233.in-addr.arpa" {
    type slave;
    file "backup.china-reverse";
    masters {
        223.169.23.1;
        223.169.23.2;
    };
};
```

Tambahkan informasi berikut ke `/etc/namedb/db.example.org`:

```
@      IN SOA      freebie.example.org. grog.example.org. (
      1997090501 ; Serial (date, 2 digits version of day)
      86400 ; refresh (1 day)
      7200 ; retry (2 hours)
      8640000 ; expire (100 days)
      86400 ) ; minimum (1 day)

china      IN      NS ns.china.example.org.
china      IN      NS ns1.china.example.org.
ns.china   IN      A 223.169.23.1
ns1.china  IN      A 223.169.23.2
```

Karena merubah informasi, kita juga rubah serial number dari SOA record sehingga secondary name server untuk example.org akan me'reload informasi terupdate.

Setelah merubah konfigurasi, restart name server:

```
#ndc reload
```

Periksa keluaran system log dengan menggunakan perintah `tail /var/log/messages`:

```
Mar 18 15:23:40 freebie named[69752]: reloading nameserver
Mar 18 15:23:40 freebie named[69752]: master zone "china.example.org" (IN)
loaded (serial 2001061701)
Mar 18 15:23:40 freebie named[69752]: Forwarding source address is
[0.0.0.0].4673
Mar 18 15:23:40 freebie named[69752]: Ready to answer queries.
```

Referensi :

- FreeBSD Handbook
- Arsip di Internet

Semoga Bermanfaat

Faiz

faiz@purwakarta.org

05:52:38 PM - 09/07/05