DNS COMO

Nicolai Langfeldt janl@math.uio.no Traducción: Pedro Pablo Fábrega Martínez pfabrega@arrakis.es
Julio 1997 v1.3.2, 3 Junio 1997

COMO ser un administrador DNS en poco tiempo.

Índice General

1 Preámbulo

Claves:

DNS, bind, named, dialup, ppp, slip, Internet, domain, name, hosts, resolving

1.1 Rollo Legal

1.2 Créditos y solicitud de ayuda.

Quiero dar las gracias a Arnt Gulbrandsen, quien leyó los borradores para este trabajo dedicándole mucho de su tiempo y proporcionando muchas y muy útiles sugerencias. También quiero dar las gracias a aquellas personas que me enviaron sugerencias por email y gracias por sus notas. Gracias, me ayudasteis a seguir con esto.

Esto nunca será un documento acabado, por favor mandadme correo con vuestros problemas y éxitos, esto hará mejorar este COMO. Así que por favor mandad dinero, comentarios y/o preguntas a `janl@math.uio.no`. Si envía un E-mail, por favor asegúrese de que su dirección de remitente es correcta, recibo un montón de E-mail. Asimismo, por favor lea la sección de las PUF (8) antes de enviarme un mail.

Si quiere traducir este COMO, por favor notifíquemelo para que yo pueda mantener la pista de las lenguas en las que he sido publicado `:-)`.

1.3 Dedicatoria

Este COMO está dedicado a Anne Line Norheim. Aunque ella probablemente nunca lo leerá porque no es de esa clase de chicas.

2 Introducción.

Lo que es esto y lo que no es.

Para los que comienzan, **DNS** es el **Domain Name System (Sistema de Nombres de Dominio)**, las reglas de nomenclatura de las máquinas y el software que mapea los nombres a números IP. Este documento COMO trata de cómo definir tales conversiones usando un sistema **Linux**. Una conversión es simplemente una asociación entre dos cosas, en este caso un nombre de máquina, como `ftp.linux.org` y el número IP de la máquina, `199.249.150.4`.

El DNS es, para los no iniciados (Vd. `;-)`, una de las áreas mas opacas de la administración de una red. Este COMO tratará de aclarar algunas cosas.

Este documento describe cómo configurar un servidor de nombres DNS simple. Comenzaremos con un servidor *caching only server*[1], y continuaremos con la configuración de un servidor DNS primario para un dominio. Para configuraciones más complejas puede consultar la sección de PUF (8) de este documento. Si lo que busca no está descrito allí, necesitará *Documentación Real*. Volveremos a lo que es la *Documentación Real* en el último capítulo.

Antes de empezar, debe configurar su sistema convenientemente, de forma que pueda hacer `telnet` desde y hacia su máquina, efectuando satisfactoriamente toda clase de conexiones de red, especialmente `telnet 127.0.0.1` entrando en su propia máquina (compruébelo ahora). También necesita que los archivos `/etc/host.conf` (o `/etc/nsswitch.conf`), `/etc/resolv.conf` y `/etc/hosts` sean correctos como punto de partida, ya que no explicaré sus funciones aquí. Si NO tiene aun esta configuración y no funciona en red, el *NET-2 HOWTO* explica como hacerlo. Léalo.

Si está usando SLIP o PPP necesitará que funcionen correctamente. Lea el *PPP-COMO* si no es así.

Cuando digo *"su máquina"* quiero decir la máquina en la que está intentando configurar DNS. No cualquier otra máquina que pueda tener en su red.

Supongo que **no está detrás de cualquier clase de cortafuegos** (*firewall*) que bloquee peticiones de nombres. Si necesita una configuración especial, vea la sección PUF (8).

El servicio de nombres en Unix es llevado a cabo por un programa, llamado `named`.

Éste forma parte del paquete `bind`, que es coordinado por Paul Vixie para *The Internet Software Consortium*. `named` está incluido en la mayoría de las distribuciones de Linux y generalmente se instala como `/usr/sbin/named`. Si tiene el fichero `named` probablemente pueda usarlo; si no lo tiene, puede obtener el binario en un ftp de Linux, o conseguir los últimos y más voluminosos fuentes en `ftp://ftp.vix.com/pub/bind`, bien de los subdirectorios de versión actual, o de prueba, lo que mejor se adapte a su estilo de vida.

DNS es una base de datos cuyo ámbito es la Red. Tenga cuidado con lo que pone en ella. Si pone incongruencias, Vd. y los demás obtendrán incongruencias de ella. Mantenga su DNS limpia y consistente y conseguirá un buen servicio de ella. Aprenda a usarla, administrarla, depurarla y será otro buen administrador, salvando a la red de caer sobre sus rodillas sobrecargada por falta de mantenimiento.

En este documento expongo de forma llana varias cosas que no son completamente verdad (son al menos medias verdades). Todo esto lo hago en aras de la simplicidad. Todas funcionarán (probablemente `;-)` si cree en lo que digo.

Aviso:

> Haga una copia de seguridad de todos los archivos que le indico que cambie si ya los tiene, y así si después si no funciona podrá volver al principio.

3 Servidor de nombres de "sólo cacheo".

Un primer ataque a la configuración DNS, muy útil para los usuarios de conexiones telefónicas.

[1] Servidor que se limita a guardar en una caché las IPs de los nombres de máquina más solicitados, obteniéndolas de servidores externos.

Un servidor de nombres de “sólo cacheo” (*caching only nameserver*) obtendrá la respuesta a las solicitudes de nombre provenientes de su red preguntando a servidores externos, recordando la respuesta para la próxima vez que lo necesite.

Lo primero que necesita es el archivo llamado `/etc/named.boot`. Este archivo es leído cuando se inicia `named`. Por ahora contendrá simplemente:

```
; Archivo boot de servidor de nombres de solo cacheo:
;
directory /var/named
;
; tipo          dominio                         fichero o maquina fuente
cache           .                               root.cache
primary         0.0.127.in-addr.arpa            pz/127.0.0
```

MUY IMPORTANTE:

En algunas versiones de este documento, en el contenido de los archivos que aquí aparecen hay un par de espacios o tabuladores antes del primer carácter no blanco. Se supone que estos caracteres **NO** están en el archivo. Borre cualquier espacio inicial de los archivos que corte y pegue de este COMO.

La línea `directory` indica a `named` dónde buscar los archivos. Todos los archivos indicados a continuación serán relativos a este directorio. `/var/named` es el directorio correcto de acuerdo con el *LFS, Linux File system Standard*. Así, `pz` es un directorio bajo `/var/named`, esto es, `/var/named/pz`.

3.1 /var/named/root.cache

Vamos a describir el archivo llamado `/var/named/root.cache` nombrado en el archivo `boot.named`.

`/var/named/root.cache` debería contener esto:

```
 .       518400  NS      D.ROOT-SERVERS.NET.
 .       518400  NS      E.ROOT-SERVERS.NET.
 .       518400  NS      I.ROOT-SERVERS.NET.
 .       518400  NS      F.ROOT-SERVERS.NET.
 .       518400  NS      G.ROOT-SERVERS.NET.
 .       518400  NS      A.ROOT-SERVERS.NET.
 .       518400  NS      H.ROOT-SERVERS.NET.
 .       518400  NS      B.ROOT-SERVERS.NET.
 .       518400  NS      C.ROOT-SERVERS.NET.
;
D.ROOT-SERVERS.NET.     3600000 A       128.8.10.90
E.ROOT-SERVERS.NET.     3600000 A       192.203.230.10
I.ROOT-SERVERS.NET.     3600000 A       192.36.148.17
F.ROOT-SERVERS.NET.     3600000 A       192.5.5.241
G.ROOT-SERVERS.NET.     3600000 A       192.112.36.4
A.ROOT-SERVERS.NET.     3600000 A       198.41.0.4
H.ROOT-SERVERS.NET.     3600000 A       128.63.2.53
B.ROOT-SERVERS.NET.     3600000 A       128.9.0.107
C.ROOT-SERVERS.NET.     3600000 A       192.33.4.12
```

¡Recuerde lo que dije sobre los espacios iniciales!

Este archivo describe los servidores de nombres raíz en el mundo. Este archivo cambiará a lo largo del tiempo y tiene que ser mantenido y actualizado con una cierta regularidad. Vea la sección de mantenimiento (6) para saber cómo mantenerlo actualizado. Este archivo está descrito en la página `man` de `named`, pero esto es, IMHO[2], más apropiado para gente que ya comprende `named`.

La siguiente línea de `named.boot` es la línea `primary`. Explicaré su uso en un capítulo posterior: Por ahora, cree un archivo llamado `127.0.0` en el subdirectorio `pz`:

```
@               IN      SOA     ns.linux.bogus. hostmaster.linux.bogus. (
                                1       ; Numero de Serie
                                28800   ; Tasa de Refresco
                                7200    ; Tasa de Reintento
                                604800  ; Caducidad para secundario
                                86400)  ; Validez para Clientes
                        NS      ns.linux.bogus.
1                       PTR     localhost.
```

A continuación necesita el archivo `/etc/resolv.conf`, que será algo similar a este:

```
search subdominio.su-dominio.edu su-dominio.edu
nameserver 127.0.0.1
```

La línea ‘`search`’ especifica en qué dominios se buscaría para cualquier nombre de máquina a la que quiera conectar. La línea ‘`nameserver`’ especifica la dirección de su servidor de nombres, en este caso su propia máquina, ya que es ahí donde `named` se estará ejecutando. Si quiere una lista de varios servidores ponga una línea `nameserver` para cada uno. (Nota: `named` nunca lee este archivo, lo hace el resolutor que usa `named`).

Para ilustrar lo que hace este archivo:
Si un cliente intenta buscar a `fulano`, `fulano.subdominio.su-dominio.edu` se probará primero, a continuación `fulano.su-dominio.edu`, y finalmente `fulano`. Si un cliente intenta buscar `sunsite.unc.edu`, `sunsite.unc.edu.subdominio.su-dominio.edu` se prueba primero (sí, es tonto, pero es así como tiene que ser), después `sunsite.unc.edu.su-dominio.edu`, y finalmente `sunsite.unc.edu`. Puede que no quiera poner demasiados dominios en la línea `search`, lleva su tiempo el efectuar las búsquedas.

El ejemplo supone que pertenece al dominio `subdominio.su-dominio.edu`, su máquina probablemente se llame `su-maquina.subdominio.su-dominio.edu`. La línea `search` no debería contener su *TLD* (*Top Level Domain* o *Dominio de Nivel Superior*, ‘`edu`’ en este caso). Si necesita conectar frecuentemente con máquinas de otro dominio, puede añadir ese dominio a la línea `search` como sigue:

```
search subdominio.su-dominio.edu su-dominio.edu otro-dominio.com
```

y así sucesivamente. Obviamente necesita poner un dominio real en su lugar. Por favor, dése cuenta de la falta de puntos al final de estos nombres de dominio.

Lo siguiente, dependiendo de su versión de la librería `libc` puede necesitar arreglar `/etc/nsswitch.conf` o `/etc/host.conf`. Si ya tiene `nsswitch.conf` corregiremos éste, en otro caso arreglaremos `host.conf`.

3.2 `/etc/nsswitch.conf`

Se trata de un extenso archivo donde se especifica de dónde obtener las diferentes clases de tipos de datos, y de cuál archivo o base de datos. Generalmente contiene comentarios útiles al comienzo, que por cierto debería considerar leer ahora. Después busque la línea que comienza por `hosts:`, debe leerse:

```
hosts:      files dns
```

Si no hay una línea que comience por `‘hosts:’` póngala. Eso indica que los programas deben mirar primero en el fichero `/etc/hosts` , y después comprobar DNS de acuerdo con `resolv.conf`.

[2] *In My Honest Opinion*, EMMO o *En Mi Modesta Opinión* en castellano.

3.3 /etc/host.conf

Probablemente contiene varias líneas, una de ellas debería comenzar con `order` y tendría que parecerse a lo siguiente:

```
order hosts,bind
```

Si no hay una línea `order` tiene que incluirla. Esto le indica a las rutinas de resolución de nombres que busquen primero en `/etc/hosts`, y pregunte luego al servidor de nombres (que dijo en `resolv.conf` que está en `127.0.0.1`). Estos dos últimos archivos están documentados en la página de manual `resolv (8)` (haciendo `man 8 resolv`) en la mayoría de las distribuciones **Linux**. Esta página de manual es de obligada lectura *IMHO*, y todos, especialmente los administradores DNS, deberían leerla. Hágalo ahora, si se dice a sí mismo *“lo haré más tarde”* entonces nunca lo hará.

3.4 Arranque de named

Después de todo esto, ya es hora de iniciar `named`. Si está utilizando una conexión telefónica, conéctese primero. Teclee `ndc start` y presione `return`, sin opciones. Si tiene problemas intente `/usr/sbin/ndc start` en su lugar. Si el problema persiste vea la sección PUF (8). Ahora ya puede comprobar su configuración. Si mira en el archivo de mensajes de `syslog` (generalmente llamado `/var/adm/messages`, o en el directorio `/var/log`) mientras inicia `named`, (haga `tail -f /var/adm/messages`) debería ver algo como esto:

```
Jun 30 21:50:55 roke named[2258]: starting.  named 4.9.4-REL Sun Jun 30 21:29:03 MET DST 1996
                            janl@roke.slip.ifi.uio.no:/var/tmp/bind/named
Jun 30 21:50:55 roke named[2258]: cache zone "" loaded (serial 0)
Jun 30 21:50:55 roke named[2258]: primary zone "0.0.127.in-addr.arpa" loaded (serial 1)
```

Si hay cualquier mensaje de error se deberá a alguna equivocación. `named` determinará el archivo que ocasiona el error (de `named.boot` o `root.cache` espero :-). Mate a `named` y vuelva a comprobar el archivo.

Ahora es el momento de iniciar `nslookup` para examinar su trabajo:

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

>
```

Si es eso lo que obtiene entonces está funcionando. Eso espero. En cualquier otro caso, vuelva atrás y compruébelo todo. Cada vez que cambie el archivo `named.boot` tiene que reinicializar `named` usando el comando `ndc restart`.

Ahora puede introducir una consulta. Intente buscar alguna máquina cercana a la suya. `pat.uio.no` está cerca de mí, en la Universidad de Oslo:

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

Name:    pat.uio.no
Address:  129.240.2.50
```

`nslookup` ahora solicita a `named` que busque la máquina `pat.uio.no`. Contactará con alguna de las máquinas servidoras de nombres nombradas en el archivo `root.cache`, y preguntará allí. Puede tardar un poco antes de conseguir el resultado ya que busca todos los dominios indicados en `/etc/resolv.conf`.

Si intenta de nuevo obtendrá esto:

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

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

Nota a la línea "`Non-authoritative answer:`" : le dedicaremos un poco de tiempo. Esto significa que `named` no sale de la red para preguntar esta vez, en su lugar mira en su caché y lo encuentra allí. Pero la información de la caché puede no estar actualizada. Entonces informa de este peligro (de modo un tanto eufemístico) con `Non-authorative answer:`. Cuando `nslookup` dice esto la segunda vez que pregunta por una máquina, es un signo seguro de que `named` almacena la información en la caché y que está funcionando. Ahora puede salir de `nslookup` usando el comando `exit`.

Si es un usuario de conexiones telefónicas, (ppp, slip) por favor lea la sección sobre conexiones telefónicas (7), hay algunas advertencias para Vd.

Ahora ya sabe cómo configurar un servidor de nombres de "sólo cacheo". Tómese una cerveza, un vaso de leche o cuaquier otra cosa que prefiera para celebrarlo.

4 Un dominio simple.

Como configurar su propio dominio.

4.1 Pero primero algo de teoría a secas.

Antes de comenzar realmente con esta sección, voy a dar un poco de teoría sobre cómo funciona DNS. Y lo va a leer porque será mejor para Vd. Si no quiere, al menos debería echar un vistazo rápido. Deje el repaso cuando sepa lo que debe incluir en su archivo `named.boot`.

El DNS es un sistema jerárquico. La raíz se escribe como '`.`' y se denomina *'root'*. Debajo hay cierto número de Dominios de Nivel Superior (*Top Level Domains*, *TLD*s), los más conocidos son `ORG, COM, EDU` y `NET`, pero hay muchos más.

Cuando se busca una máquina, la pregunta procede recursivamente en la jerarquía comenzando desde arriba. Si quiere localizar la dirección de `prep.ai.mit.edu`, su servidor de nombres ha de encontrar primero un servidor de nombres que sirva a `edu`. Pregunta al servidor `.` (ya conoce los servidores `.`, es para lo que se utiliza el `archivo root.cache`), y el servidor `.` proporcionará una lista de servidores `edu`:

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

Comienza preguntando a un servidor raíz.

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

Pone el tipo de petición (*Query*) a `NS` (*Name Server records*).

```
> set q=ns
```

Pregunta por `edu`.

```
> edu.
```

El punto (".") final aquí es significativo, indica al servidor que le pedimos un `edu` que está justo debajo de ".", y esto reduce la búsqueda un poco.

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

Esto nos dice que `*.root-servers.net` sirve a `edu.`, y así podemos seguir preguntando a `C`. Ahora queremos saber quién sirve el siguiente nivel del nombre de dominio: `mit.edu.`

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

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

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

`steawb, w20ns` y `bitsy` sirven a `mit`, selecciona uno y pregunta por `ai.mit.edu`:

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

Los nombres de máquina no son sensibles a mayúsculas/minúsculas, pero como yo uso el ratón para cortar y pegar, obtengo una copia tal y como aparece en la pantalla.

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

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

Non-authoritative answer:
ai.mit.edu      nameserver = WHEATIES.AI.MIT.EDU
ai.mit.edu      nameserver = ALPHA-BITS.AI.MIT.EDU
ai.mit.edu      nameserver = GRAPE-NUTS.AI.MIT.EDU
ai.mit.edu      nameserver = TRIX.AI.MIT.EDU
ai.mit.edu      nameserver = MUESLI.AI.MIT.EDU

Authoritative answers can be found from:
AI.MIT.EDU      nameserver = WHEATIES.AI.MIT.EDU
AI.MIT.EDU      nameserver = ALPHA-BITS.AI.MIT.EDU
AI.MIT.EDU      nameserver = GRAPE-NUTS.AI.MIT.EDU
AI.MIT.EDU      nameserver = TRIX.AI.MIT.EDU
AI.MIT.EDU      nameserver = MUESLI.AI.MIT.EDU
WHEATIES.AI.MIT.EDU     internet address = 128.52.32.13
WHEATIES.AI.MIT.EDU     internet address = 128.52.35.13
ALPHA-BITS.AI.MIT.EDU   internet address = 128.52.32.5
ALPHA-BITS.AI.MIT.EDU   internet address = 128.52.37.5
GRAPE-NUTS.AI.MIT.EDU   internet address = 128.52.32.4
GRAPE-NUTS.AI.MIT.EDU   internet address = 128.52.36.4
TRIX.AI.MIT.EDU internet address = 128.52.32.6
TRIX.AI.MIT.EDU internet address = 128.52.38.6
MUESLI.AI.MIT.EDU       internet address = 128.52.32.7
MUESLI.AI.MIT.EDU       internet address = 128.52.39.7
```

Entonces `weaties.ai.mit.edu` es un servidor de nombres para `ai.mit.edu`:

```
> server WHEATIES.AI.MIT.EDU.
Default Server:  WHEATIES.AI.MIT.EDU
Addresses:  128.52.32.13, 128.52.35.13
```

Ahora cambia el tipo de solicitud; ha encontrado el servidor de nombres y va a preguntar todo lo que queremos saber sobre `prep.ai.mit.edu`.

```
> set q=any
> prep.ai.mit.edu.
Server:  WHEATIES.AI.MIT.EDU
Addresses:  128.52.32.13, 128.52.35.13

prep.ai.mit.edu CPU = dec/decstation-5000.25    OS = unix
prep.ai.mit.edu
        inet address = 18.159.0.42, protocol = tcp
         #21 #23 #25 #79
prep.ai.mit.edu preference = 1, mail exchanger = life.ai.mit.edu
prep.ai.mit.edu internet address = 18.159.0.42
ai.mit.edu      nameserver = alpha-bits.ai.mit.edu
ai.mit.edu      nameserver = wheaties.ai.mit.edu
ai.mit.edu      nameserver = grape-nuts.ai.mit.edu
ai.mit.edu      nameserver = mini-wheats.ai.mit.edu
ai.mit.edu      nameserver = trix.ai.mit.edu
```

```
ai.mit.edu      nameserver = muesli.ai.mit.edu
ai.mit.edu      nameserver = count-chocula.ai.mit.edu
ai.mit.edu      nameserver = life.ai.mit.edu
ai.mit.edu      nameserver = mintaka.lcs.mit.edu
life.ai.mit.edu internet address = 128.52.32.80
alpha-bits.ai.mit.edu   internet address = 128.52.32.5
wheaties.ai.mit.edu     internet address = 128.52.35.13
wheaties.ai.mit.edu     internet address = 128.52.32.13
grape-nuts.ai.mit.edu   internet address = 128.52.36.4
grape-nuts.ai.mit.edu   internet address = 128.52.32.4
mini-wheats.ai.mit.edu  internet address = 128.52.32.11
mini-wheats.ai.mit.edu  internet address = 128.52.54.11
mintaka.lcs.mit.edu     internet address = 18.26.0.36
```

De esta forma comenzando en `.` ha encontrado los sucesivos servidores de nombre para el siguiente nivel en el nombre de dominio. Si ha usado su propio servidor DNS en lugar de usar todos esos otros servidores, su `named`, desde luego, habrá almacenado en el caché toda la información que haya encontrado mientras profundizaba en la búsqueda, y en consecuencia no tendrá que preguntar de nuevo durante un tiempo.

Se habla mucho menos sobre él, pero un dominio importante es `in-addr.arpa`. También está anidado como los dominios *'normales'*. `in-addr.arpa` nos permite determinar el nombre de la máquina cuando conocemos su dirección IP. Una cosa importante aquí es observar que que las direcciones IP están escritas en orden inverso en el dominio `in-addr.arpa`. Si tiene la dirección de máquina `192.128.52.43`, `named` procede como para el ejemplo de `prep.ai.mit.edu`: Busca los servidores `arpa.`. Busca los servidores `in-addr.arpa.`, los servidores `192.in-addr.arpa.`, los servidores `128.192.in-addr.arpa.` , y los servidores `52.128.192.in-addr.arpa.` y finalmente, los registros necesarios para `43.52.128.192.in-addr.arpa.` ¿Inteligente? (Diga 'sí'). La inversión de números puede ser confusa los 2 primeros años.

He contado una mentira. DNS no funciona como he dicho de forma literal. Pero es bastante parecido.

4.2 Nuestro propio dominio

Ahora vamos a definir nuestro propio dominio. Vamos a crear el dominio `linux.bogus` y definir máquinas en él. Uso un nombre de dominio totalmente falso para estar seguro de que no molestamos a nadie de fuera.

Ya hemos comenzado esta parte con la siguiente línea en `named.boot`:

```
primary         0.0.127.in-addr.arpa            pz/127.0.0
```

Por favor tome nota de la ausencia de '.' al final de los nombres de dominio en este archivo. La primera línea nombra al archivo `pz/127.0.0` como definición de `0.0.127.in-addr.arpa`. Ya hemos configurado este archivo, en él podremos leer:

```
@               IN      SOA     ns.linux.bogus. hostmaster.linux.bogus. (
                                1       ; Numero de Serie
                                28800   ; Tasa de Refresco
                                7200    ; Tasa de Reintento
                                604800  ; Caducidad para secundario
                                86400)  ; Tiempo de Validez para Clientes
                        NS      ns.linux.bogus.
1                       PTR     localhost.
```

Por favor observe los '`.`' al final de los nombres de dominio completo en contraste con el archivo `named.boot` anterior. A algunas personas les gusta iniciar cada zona del archivo con una directiva

`$ORIGIN`, pero esto es superfluo. El origen (lugar de la jerarquía DNS a donde pertenece) de un fichero de zona se especifica en la columna `dominio` del archivo `named.boot`; en este caso es `0.0.127.in-addr.arpa.`

Este "fichero de zona" contiene tres *registros de recursos* (RRs): Un `RR SOA`, Un `RR NS` y un `RR PTR`. `SOA` es una abreviatura de *Start Of Authority*. La '`@`' es una notación especial que simboliza el origen, y como la columna `dominio` para este archivo indica `0.0.127.in-addr.arpa`. La primera línea realmente significa:

```
0.0.127.IN-ADDR.ARPA. IN      SOA ...
```

`NS` es el `RR Name Server` (Servidor de Nombres), e indica a DNS qué máquina es el servidor de nombres del dominio. Y finalmente el registro `PTR` tiene valor `1` (igual a `1.0.0.127.IN-ADDR.ARPA`, esto es, `127.0.0.1`) que es el `localhost` de `named`.

El registro `SOA` es el preámbulo de todos los archivos de zona y debe haber uno exactamente en cada archivo de zona, como primer registro de todos. El registro `SOA` describe la zona, de dónde proviene (una máquina llamada `linux.bogus`), quién es el responsable de su contenido (`hostmaster@linux.bogus`), qué versión del archivo de zona es (`Numero de Serie`, `1`), y otras cosas que tienen que ver con el caché y los servidores secundarios DNS. Para el resto de los campos (`Tasa de Refresco`, `Tasa de Reintento`, `Caducidad para secundario` y `Tiempo de Validez para Clientes`) use los valores que aparecen aquí para mayor seguridad.

El registro `NS` nos indica quién efectúa el servicio DNS para `0.0.127.in-addr.arpa`, que es `ns.linux.bogus`. El registro `PTR` nos dice que `1.0.0.127.in-addr.arpa` (aka `127.0.0.1`) es conocido como `localhost`.

Ahora reiniciamos `named` (el comando es `ndc restart`) y usamos `nslookup` para examinar lo que ha hecho:

```
$ nslookup

Default Server:  localhost
Address:  127.0.0.1

> 127.0.0.1
Server:  localhost
Address:  127.0.0.1

Name:    localhost
Address:  127.0.0.1
```

así obtiene `localhost` de `127.0.0.1`, bien. Ahora para nuestra tarea principal, el dominio `linux.bogus`, inserte una nueva línea, `primary`, en `named.boot`:

```
primary                 linux.bogus                     pz/linux.bogus
```

Observe que continúa la ausencia de "`.`" final en el nombre de dominio del archivo `named.boot`.

En el archivo de zona de `linux.bogus` pondremos algunos datos totalmente falsos[3]:

```
;
; Fichero de zona para linux.bogus
;
; Minimo indspensable para tener funcionando un dominio
;
@       IN      SOA     ns.linux.bogus. hostmaster.linux.bogus. (
```

[3]N del T
Por si no lo ha notado todavía, *bogus* en inglés significa precisamente *falso*.

```
                        199511301       ; Numero de serie, fecha de hoy + n. de serie de hoy
                        28800           ; Tasa de Refresco, en segundos
                        7200            ; Tasa de Reintento, en segundos
                        3600000         ; Caducidad para secundario, en segundos
                        86400 )         ; Tiempo de Validez para Clientes, en segundos
                NS      ns.linux.bogus.
                NS      ns.friend.bogus.
                MX      10 mail.linux.bogus   ; Intercambiador de Correo Primario
                MX      20 mail.friend.bogus. ; Intercambiador de Correo Secundario

localhost       A       127.0.0.1
ns              A       127.0.0.2
mail            A       127.0.0.4
```

Deben de observarse dos cosas sobre los registros `SOA`. **`ns.linux.bogus`** debe ser una máquina actual con un registro `A`. No es legal tener un registro `CNAME` para la máquina mencionada en el registro `SOA`. Su nombre no necesita ser **`ns`**, podría ser cualquier nombre legal de máquina. A continuación, en **`hostmaster.linux.bogus`** deberá aparecer algo como **`hostmaster@linux.bogus`**; esto sería un alias de email, o una cuenta de correo, donde la(s) persona(s) que realizan el mantenimiento de DNS deberían leer con frecuencia el correo. Cualquier email respecto del dominio será mandado a la dirección aquí indicada. El nombre no tiene por que ser **`hostmaster`**, puede ser cualquier dirección email legal, pero la dirección email **`hostmaster`** funcionará bien.

Hay un nuevo tipo de `RR` en este archivo, el `MX`, o *Mail eXchanger*. Este indica el sistema de correo a donde mandar el correo dirigido a **`alguien@linux.bogus`**, pudiendo ser también **`mail.linux.bogus`** o **`mail.friend.bogus`**. El número que precede a cada nombre de máquina es la prioridad del `RR MX`. El `RR` con el número más bajo (10) es aquel al que el correo será enviado primero. Si este falla, puede ser mandado a otro con un número más alto, que será gestor secundario de correo, como **`mail.friend.bogus`** que tiene una prioridad 20 aquí.

Reinicie **`named`** ejecutando **`ndc restart`**. Examine los resultados con **`nslookup`**:

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

linux.bogus
        origin = linux.bogus
        mail addr = hostmaster.linux.bogus
        serial = 199511301
        refresh = 28800 (8 hours)
        retry   = 7200 (2 hours)
        expire  = 604800 (7 days)
        minimum ttl = 86400 (1 day)
linux.bogus     nameserver = ns.linux.bogus
linux.bogus     nameserver = ns.friend.bogus
linux.bogus     preference = 10, mail exchanger = mail.linux.bogus.linux.bogus
linux.bogus     preference = 20, mail exchanger = mail.friend.bogus
linux.bogus     nameserver = ns.linux.bogus
linux.bogus     nameserver = ns.friend.bogus
ns.linux.bogus  internet address = 127.0.0.2
mail.linux.bogus        internet address = 127.0.0.4
```

Con un examen cuidadoso podrá descubrir un error. La línea

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

está equivocada. Debería ser

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

Cometí el error de forma deliberada para que aprenda de él `:-)` Mirando en el archivo de zona podemos ver que la línea

```
@               MX      10 mail.linux.bogus     ; Intercambiador de Correo Primario
```

no tiene punto. O tiene demasiados `linux.bogus`. Si un nombre de máquina no termina en punto en un archivo de zona, el origen es añadido a su final. Así,

```
@               MX      10 mail.linux.bogus.    ; Intercambiador de Correo Primario
```

o

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

serán correctos. Yo prefiero la última forma, hay que escribir menos. En un archivo de zona el dominio debería ser escrito y terminado con un punto, o no debe ser incluido, en cuyo caso se referirá al origen por defecto. Debo hacer hincapié que en el archivo `named.boot` no debería haber puntos después de los nombres de dominio. No tiene ni idea de cuantas veces un '.' por estar o por no estar ha hecho fallar toda una configuración y ha confundido horrorosamente a la gente...

Una vez hecha esta *puntualización*, he aquí el nuevo archivo de zona, con algo de información extra también:

```
;
; Archivo de zona para  linux.bogus
;
; minimo indispensable para hacer funcionar un dominio
;
@       IN      SOA     ns.linux.bogus. hostmaster.linux.bogus. (
                        199511301       ; Numero de Serie, fecha de hoy + n. de serie de hoy
                        28800           ; Tasa de Refresco, en segundos
                        7200            ; Tasa de Reintento, en segundos
                        604800          ; Caducidad para secundario, en segundos
                        86400 )         ; Validez para Clientes, en segundos

                NS      ns              ; Direccion de Internet del servidor de nombres
                NS      ns.friend.bogus.
                MX      10 mail         ; Intercambiador de Correo Primario
                MX      20 mail.friend.bogus. ; Intercambiador de Correo Secundario

localhost       A       127.0.0.1
ns              A       127.0.0.2
mail            A       127.0.0.4
;
; Extras
;
@               TXT     "Linux.Bogus, your DNS consultants"

ns              MX      10 mail
```

```
                MX      20 mail.friend.bogus.
                HINFO   "Pentium" "Linux 1.2"
                TXT     "RMS"
richard         CNAME   ns
www             CNAME   ns

donald          A       127.0.0.3
                MX      10 mail
                MX      20 mail.friend.bogus.
                HINFO   "i486"  "Linux 1.2"
                TXT     "DEK"

mail            MX      10 mail
                MX      20 mail.friend.bogus.
                HINFO   "386sx" "Linux 1.0.9"

ftp             A       127.0.0.5
                MX      10 mail
                MX      20 mail.friend.bogus.
                HINFO   "P6" "Linux 1.3.59"
```

Puede que quiera desplazar los tres primeros registros tipo `A` (`localhost, ns` y `mail`) junto con los otros registros de su mismo tipo (`donald`, `mail`, y `ftp`), en vez de colocarlos separados al principio como aquí.

Hay varios registros nuevos aquí: `HINFO` (*Host INFOrmation*), tiene dos partes, es una buena costumbre poner comillas a cada uno. La primera parte es el hardware o CPU de la máquina, y la segunda parte corresponde al software o Sistema Operativo de la misma. `ns` tiene una CPU Pentium con Linux 1.2. El registro `TXT` es un texto en formato libre que puede usar para cualquier cosa que le interese. `CNAME` (*Canonical NAME*) es una forma de dar a cada máquina varios nombres. Por tanto `richard` y `www` son alias para `ns`. Es importante observar que los registros `A`, `MX`, `CNAME` y `SOA` **nunca deben hacer referencia al registro** `CNAME`, sólo pueden referirse a registros `A`.

```
fulanito  CNAME   richard                ; ¡¡¡NO!!!
```

siendo correcto tener

```
fulanito  CNAME   ns                     ; ¡¡¡SI!!!
```

También es importante observar que `CNAME` no es un nombre de máquina legal para direcciones de correo: `webmaster@www.linux.bogus` es una dirección email ilegal dada en la configuración anterior. Encontrará muy pocos administradores de correo de Ahí Afuera que recomienden esta regla, incluso si a Vd. le funciona. La forma de evitar esto es usar un registro `A` (y quizás algunos otros también, como un registro `MX`) en su lugar:

```
www     A       127.0.0.2
```

Paul Vixie, el principal gurú de `named` recomienda no usar CNAME. Por tanto considere el no utilizarlo seriamente.

Cargue la nueva base de datos ejecutando `ndc reload`, esto provoca la lectura de sus archivos de nuevo.

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

> ls -d linux.bogus
```

Esto haría que todos los registros fuesen listados.

```
[localhost]
 linux.bogus.       SOA   ns.linux.bogus hostmaster.linux.bogus. (199511301 28800 7200 604800 86400)
 linux.bogus.       NS    ns.linux.bogus
 linux.bogus.       NS    ns.friend.bogus
 linux.bogus.       MX    10   mail.linux.bogus
 linux.bogus.       MX    20   mail.friend.bogus
 linux.bogus.       TXT   "Linux.Bogus, your DNS consultants"
 localhost          A     127.0.0.1
 mail               A     127.0.0.4
 mail               MX    10   mail.linux.bogus
 mail               MX    20   mail.friend.bogus
 mail               HINFO 386sx       Linux 1.0.9
 donald             A     127.0.0.3
 donald             MX    10   mail.linux.bogus
 donald             MX    20   mail.friend.bogus
 donald             HINFO i486        Linux 1.2
 donald             TXT   "DEK"
 www                CNAME ns.linux.bogus
 richard            CNAME ns.linux.bogus
 ftp                A     127.0.0.5
 ftp                MX    10   mail.linux.bogus
 ftp                MX    20   mail.friend.bogus
 ftp                HINFO P6          Linux 1.3.59
 ns                 A     127.0.0.2
 ns                 MX    10   mail.linux.bogus
 ns                 MX    20   mail.friend.bogus
 ns                 HINFO Pentium     Linux 1.2
 ns                 TXT   "RMS"
 linux.bogus.       SOA   ns.linux.bogus hostmaster.linux.bogus. (199511301 28800 7200 604800 86400)
```

Esto está bien. Comprobemos qué dice para `www` sólo:

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

www.linux.bogus canonical name = ns.linux.bogus
```

...En otras palabras, el nombre real de `www.linux.bogus` es `ns.linux.bogus`

```
linux.bogus     nameserver = ns.linux.bogus
linux.bogus     nameserver = ns.friend.bogus
ns.linux.bogus  internet address = 127.0.0.2
```

y `ns.linux.bogus` tiene la dirección `127.0.0.2`. Parece correcto también.

4.3 Relajémonos

Desde luego, este dominio es falso, y como tal son todas sus direcciones, y quizás, desafortunadamente sea un poco confuso. Para un ejemplo real de dominio vea la siguiente sección.

5 Un ejemplo de dominio real

Donde describiremos algunos archivos de zona reales.

Los usuarios han sugerido que incluya un ejemplo real de dominio que esté en funcionamiento como explicación de las diferencias entre un dominio en funcionamiento y el ejemplo falso que no era del todo muy claro.

Una cosa sobre este ejemplo: ¡NO lo introduzca en su servidor de nombres!. Úselo sólo como lectura de referencia. Si quiere experimentar, hágalo con el ejemplo falso. Yo uso este ejemplo con permiso de David Bullock y LAND-5. Estos archivos eran los usados el 24 de Septiembre de 1996, y podrían diferir de los que encuentre si pregunta ahora al servidor de nombres LAND-5. También tenga en mente eliminar los espacios iniciales ;-).

5.1 `/etc/named.boot` (o `/var/named/named.boot`)

Aquí encontramos la líneas `primary` para las dos zonas que necesitamos: la red `127.0.0.0` y también la subred `206.6.177` de `LAND-5`. Una línea `primary` para la zona de redirección (`forward`) `land-5.com` de `land-5`. Observe también que en lugar de situar los archivos en un directorio llamado `pz`, como hago en este COMO, él los sitúa en un directorio llamado `zone`.

```
;  Fichero de arranque para el servidor de nombres LAND-5
;
directory /var/named
;
; tipo          dominio                         fichero o maquina origen
cache           .                               root.cache
primary         0.0.127.in-addr.arpa            zone/127.0.0
primary         177.6.206.in-addr.arpa          zone/206.6.177
primary         land-5.com                      zone/land-5.com
```

5.2 `/var/named/root.cache`

Tenga en cuenta que este archivo varía con mucha frecuencia, y que el listado de aquí es viejo. Mejor utilice uno producido ahora.

```
; <<>> DiG 2.1 <<>>
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6
;; flags: qr rd ra; Ques: 1, Ans: 9, Auth: 0, Addit: 9
;; QUESTIONS:
;;      ., type = NS, class = IN

;; ANSWERS:
 .       518357  NS      H.ROOT-SERVERS.NET.
 .       518357  NS      B.ROOT-SERVERS.NET.
 .       518357  NS      C.ROOT-SERVERS.NET.
 .       518357  NS      D.ROOT-SERVERS.NET.
 .       518357  NS      E.ROOT-SERVERS.NET.
 .       518357  NS      I.ROOT-SERVERS.NET.
 .       518357  NS      F.ROOT-SERVERS.NET.
 .       518357  NS      G.ROOT-SERVERS.NET.
 .       518357  NS      A.ROOT-SERVERS.NET.
```

```
;; ADDITIONAL RECORDS:
H.ROOT-SERVERS.NET.     165593  A       128.63.2.53
B.ROOT-SERVERS.NET.     165593  A       128.9.0.107
C.ROOT-SERVERS.NET.     222766  A       192.33.4.12
D.ROOT-SERVERS.NET.     165593  A       128.8.10.90
E.ROOT-SERVERS.NET.     165593  A       192.203.230.10
I.ROOT-SERVERS.NET.     165593  A       192.36.148.17
F.ROOT-SERVERS.NET.     299616  A       192.5.5.241
G.ROOT-SERVERS.NET.     165593  A       192.112.36.4
A.ROOT-SERVERS.NET.     165593  A       198.41.0.4

;; Total query time: 250 msec
;; FROM: land-5 to SERVER: default -- 127.0.0.1
;; WHEN: Fri Sep 20 10:11:22 1996
;; MSG SIZE  sent: 17  rcvd: 312
```

5.3 /var/named/zone/127.0.0

Lo básico, el registro obligatorio `SOA`, y el registro que mapea `127.0.0.1` a `localhost`. Se requieren ambos. No debería haber ninguno más en este fichero. Probablemente nunca se necesitará actualizarlo, salvo que cambien su servidor de nombres o la dirección del `hostmaster`.

```
@               IN      SOA     land-5.com. root.land-5.com. (
                                199609203   ; Numero de Serie
                                28800   ; Tasa de Refresco
                                7200    ; Tasa de Reintento
                                604800  ; Caducidad para secundario
                                86400)  ; Validez para clientes
                        NS      land-5.com.

1                       PTR     localhost.
```

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

Aquí vemos el registro `SOA` y los registros `NS` necesarios. Podemos observar que dispone de un servidor de nombres secundario `ns2.psi.net`. Esto es como debe ser, tenga siempre un servidor secundario de seguridad. También podemos ver que tiene una máquina principal llamada `land-5` que se encarga de todos los diferentes servicios, y que se ha hecho usando `CNAME` (una alternativa al uso de los registros `A`).

Como puede ver en el registro `SOA`, el origen del archivo de zona es `land-5.com`, la persona de contacto es `root@land-5.com.` `hostmaster` es otro uso frecuente para la persona de contacto. El número de serie en el formato habitual *yyyymmdd* con el número de serie de hoy añadido; esta es probablemente la sexta versión del archivo de zona del 20 de Septiembre de 1996. Recuerde que el número de serie debe incrementerse monótonamente, aquí hay sólo un dígito para las series de hoy, así que después de 9 ediciones tendrá que esperar hasta mañana antes de poder editar el el archivo de nuevo. Considere el uso de dos dígitos.

```
@       IN      SOA     land-5.com. root.land-5.com. (
                        199609206       ; Numero de Serie, fecha de hoy + numero de serie de hoy
                        10800           ; Tasa de Refresco, en segundos
                        7200            ; Tasa de Reintento, en segundos
                        10800           ; Caducidad para secundario, en segundos
                        86400 )         ; Validez para Clientes, en segundos
                NS      land-5.com.
                NS      ns2.psi.net.
                MX      10 land-5.com.  ; Intercambiador Primario de Correo
```

```
localhost       A       127.0.0.1

router          A       206.6.177.1

land-5.com.     A       206.6.177.2
ns              CNAME   land-5.com.
ftp             CNAME   land-5.com.
www             CNAME   land-5.com.
mail            CNAME   land-5.com.
news            CNAME   land-5.com.

funn            A       206.6.177.3
illusions       CNAME   funn.land-5.com.
@               TXT     "LAND-5 Corporation"

;
;       Estaciones de Trabajo
;
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
; {Muchas definiciones repetitivas borradas}
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
```

Otra cosa a tener en cuenta es que las estaciones de trabajo no tienen nombres propios, sino un prefijo seguido por las dos últimas porciones de los números IP. Usar tal convención puede simplificar el mantenimiento significativamente, pero puede resultar un poquito impersonal.

5.5 /var/named/zone/206.6.177

Comentaré este archivo después.

```
@               IN      SOA     land-5.com. root.land-5.com. (
                                199609206  ; Numero de Serie
                                28800    ; Tasa de Refresco
                                7200     ; Tasa de Reintento
                                604800   ; Caducidad para secundario
```

```
                                86400)  ; Validez para Clientes
                        NS      land-5.com.
                        NS      ns2.psi.net.
;
;       Servidores
;
1       PTR     router.land-5.com.
2       PTR     land-5.com.
3       PTR     funn.land-5.com.
;
;       Estaciones de Trabajo
;
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.
; {Eliminadas muchas definiciones repetitivas}
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.
```

La **zona de resolución inversa** es la parte de la configuración que parece crear más dolores de cabeza. **Se usa para encontrar el nombre de la máquina a partir de su dirección IP**. Ejemplo: suponga que está en un servidor `irc` y acepta conexiones de clientes `irc`. El servidor `irc` es noruego y sólo quiere aceptar conexiones de clientes de Noruega y otros países escandinavos. Cuando se produce una conexión de un cliente, la librería *C* es capaz de indicar el número IP de la máquina conectada porque el número IP del cliente está contenido en todos los paquetes que se pasan a través de la red. Ahora puede llamar a una función llamada `gethostbyaddr` que busca el nombre de la máquina dada su dirección IP.

`gethostbyaddr` interrogará a un servidor DNS el cual efectuará una búsqueda DNS para la máquina. Suponiendo que la conexión cliente viene de `ws_177200.land-5.com`, la dirección IP que la librería *C* proporciona al servidor irc será `206.6.177.200`. Para encontrar el nombre de la máquina necesitamos encontrar `200.177.6.206.in-addr.arpa`. El servidor DNS primero encuentra los servidores `arpa.`, después los servidores `in-addr.arpa.`, a continuación sigue por `206`, `6` y al final busca el servidor para la zona `177.6.206.in-addr.arpa` en `land-5`. Aquí obtendrá finalmente que para `200.177.6.206.in-addr.arpa` tenemos un registro '`PTR ws_177200.land-5.com`', que significa que el nombre que va con `206.6.177.200` es `ws_177200.land-5.com`. Como con la explicación de cómo buscar `prep.ai.mit.edu`, esto es ligeramente ficticio.

Volviendo al ejemplo del servidor irc. El servidor irc sólo acepta conexiones de los países escandinavos, osea, `*.no`, `*.se`, y `*.dk`; el nombre `ws_177200.land-5.com` claramente no se ajusta a ninguno de ellos, y el servidor denegará la conexión. Si no hubiese habido resolución inversa de `206.2.177.200` mediante la zona `in-addr.arpa` el servidor habría sido incapaz de de encontrar el nombre y habría tenido que comparar `206.2.177.200` con `*.no`, `*.se` y `*.dk`, es decir, cifras con nombres, ninguna de las cuales concordaría.

Algunas personas le dirán que la resolución inversa sólo es importante para los servidores, o que no tienen importancia. No es así; muchos servidores de ftp, news, irc e incluso algunos servidores http (WWW) NO aceptarán conexiones de máquinas de las cuales no son capaces de resolver el nombre. Por tanto el mapeo inverso de máquinas es de hecho obligatorio.

6 Mantenimiento

Manteniéndolo en funcionamiento.

Hay una tarea de mantenimiento que tiene que realizar con `named`, además de mantenerlo en funcionamiento. Esta tarea es mantener el archivo `root.cache` actualizado. La forma más fácil es usar `dig`, primero ejecute `dig` sin argumentos, conseguirá `root.cache` de acuerdo con su propio servidor. Entonces pregunte a alguno de los servidores raíz listados con

```
dig @rootserver
```

Podrá observar que la salida se parece terriblemente al archivo `root.cache` excepto por un par de números extras. Esos números no ocasionan problemas. Guárdelo en un archivo

```
dig @rootserver . ns > root.cache.new
```

y sustituya el antiguo `root.cache` con él.

Recuerde reiniciar `named` después de sustituir el archivo caché.

Al Longyear me envió este script que puede ser ejecutado automáticamente para actualizar `root.cache`, instale una entrada en el `crontab` para ejecutarlo una vez al mes y olvídese. El script supone que trabaja con correo y que el alias de mail `hostmaster` está definido. Debe editarlo para ajustarlo a su configuración.

```
#!/bin/sh
#
# Actualizacion del cache del servidor de nombres una vez al mes.
# Esto es ejecutado automaticamente mediante una entrada de cron
#
(
 echo "To: hostmaster <hostmaster>"
 echo "From: system <root>"
 echo "Subject: Actualizacion automatica del fichero named.boot"
 echo

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

 dig @rs.internic.net . ns >root.cache.new

 echo "El fichero named.boot ha sido actualizado para contener la
siguiente informacion:"
 echo
 cat root.cache.new

 chown root.root root.cache.new
 chmod 444 root.cache.new
 rm -f root.cache.old
 mv root.cache root.cache.old
 mv root.cache.new root.cache
 ndc restart
 echo
 echo "El servidor de nombres ha sido rearrancado a fin de asegurar que la
actualizacion es completa."
 echo "El anterior fichero root.cache se ha renombrado a /var/named/root.cache.old."
) 2>&1 | /usr/lib/sendmail -t
exit 0
```

Alguno de ustedes puede haber observado que el archivo `root.cache` está también disponible mediante ftp en *Internic*. Por favor NO utilice ftp para actualizar `root.cache`, el método anterior es mucho más amistoso con la red.

7 Configuración de Conexiones Automáticas vía telefónica .

Esta sección explica cómo he dispuesto las cosas para automatizarlo todo. Mi método puede que no se adapte completamente al suyo, pero puede obtener ideas de algunas de las cosas que he hecho. También, uso `ppp` para marcar, mientras que mucha gente usa `slip` o `cslip` y por tanto casi toda su configuración puede ser distinta a la mía. Pero el programa de slip `dip` debería poder hacer muchas de las cosas que yo hago.

Normalmente, cuando no estoy conectado a la red tengo un archivo `resolv.conf` que simplemente contiene la línea

```
domain uio.no
```

Eso me asegura que no tengo que esperar a que la librería de resolución de nombres del sistema intente conectar con un servidor de nombres que no puede ayudarme. Pero cuando me conecto quiero arrancar mi `named` y tener un `resolv.conf` parecido a los descritos anteriormente. He resuelto esto teniendo dos archivos `resolv.conf` llamados `resolv.conf.local` y `resolv.conf.connected`. El último se parece al `resolv.conf` descrito anteriormente en este documento.

Para conectarme automáticamente a la red ejecuto un script llamado `ppp-on`:

```
#!/bin/sh
echo llamando...
pppd
```

`pppd` tiene un archivo llamado `options` que indica las características de la conexión. Una vez que mi conexión ppp está activa `pppd` llama a un *script* llamado `ip-up` (este está descrito en la página `pppd (8)` de man). He aquí una parte del *script*:

```
#!/bin/sh
interface="$1"
device="$2"
speed="$3"
myip="$4"
upip="$5"
 ...
cp -v /etc/resolv.conf.connected /etc/resolv.conf
 ...
/usr/sbin/named
```

Es decir, arranco el `named` desde aquí. Cuando se corta la conexión ppp, `pppd` ejecuta un script llamado `ip-down`:

```
#!/bin/sh
cp /etc/resolv.conf.local /etc/resolv.conf
read namedpid < /var/run/named.pid
kill $namedpid
```

Así configuramos las cosas de una forma cuando estamos conectados y las desconfiguramos cuando nos desconectamos.

Algunos programas, `irc` y `talk` me vienen a la mente, hacen algunas suposiciones, y para que en `irc` el comportamiento de las capacidades `dcc`, y `talk` funcionen bien tiene que modificar su archivo `hosts`. Yo he insertado en mi script `ip-up` lo siguiente:

```
cp /etc/hosts.ppp /etc/hosts
echo $myip      roke >>/etc/hosts
```

`hosts.ppp` simplemente contiene

```
127.0.0.1       localhost
```

y `echo` inserta la dirección IP que he recibido para mi nombre de host (`roke`). Vd. deberá usar en su lugar el nombre de su máquina. Este nombre se puede saber con el comando `hostname`.

Probablemente no sea inteligente ejecutar `named` cuando no esté conectado a la red, esto es porque `named` intentará enviar solicitudes a la red y eso consume tiempo, y Vd. tendrá que esperar este tiempo cada vez que algún programa intente resolver un nombre. Si está usando conexiones telefónicas debería iniciar `named` cuando se conecte y matarlo cuando se desconecte. Pero por favor lea la sección de PUF (8) para los trucos.

A algunas personas le gusta usar la directiva `forwarders` para conexiones de escasa velocidad. Si su proveedor de Internet tiene servidores DNS en `1.2.3.4` y `1.2.3.5` puede insertar la línea

```
forwarders 1.2.3.4 1.2.3.5
```

en el archivo `named.boot`. Deje también vacío el archivo `root.cache`. Esto disminuirá el tráfico IP que origina su máquina. Esto es especialmente importante si paga por cada byte que circule por el cable. Tiene el valor añadido de evitarle el deber del mantenimiento; no tiene porqué actualizar un archivo `root.cache` vacío.

8 PUFs *Preguntas de Uso Frecuente (FAQ)*

En esta sección incluyo algunas de las preguntas más frecuentes realizadas relativas a DNS y este COMO. Y las respuestas `:-)` Por favor, lea esta sección antes de enviarme correo electrónico.

8.1 ¿Cómo uso DNS desde dentro de un cortafuegos (*firewall*)?

Unas cuantas pistas: '`forwarders`', '`slave`', y echar un vistazo a la literatura que hay al final de este COMO.

8.2 ¿Cómo hago que DNS rote entre las direcciones disponibles para un servicio, por ejemplo para `www.siempre.ocupado` para obtener balanceo de carga o similar?

Haga varios registros `A` para `www.siempre.ocupado` y use `bind 4.9.3` o posterior. `bind` hará una rotación tipo *round-robin* de las respuestas. Esto no funcionará con versiones anteriores de `bind`.

8.3 Quiero configurar DNS en una intranet (cerrada) ¿qué hago?

Elimine el archivo de caché y haga los archivos de zona. Eso también significa que nunca tendrá que actualizar el archivo de caché.

8.4 Mi sistema no tiene el programa `ndc`. ¿Qué hago?

El *bind* instalado en su sistema es viejo y de alguna forma obsoleto. Si la seguridad es importante para Vd.: actualice *bind* inmediatamente. Si no, de todas formas podrá vivir con ello. En lugar de ejecutar `ndc start` ejecute `named`; `ndc reload` será `named.reload` y `ndc restart` será `named.restart`. Esos programas probablemente estarán en `/usr/sbin`.

8.5 ¿Cómo configuro un servidor de nombres secundario?

Si el servidor primario tiene la dirección `127.0.0.1`, ponga la siguiente línea en el archivo `named.boot` de su secundario:

```
secondary     linux.bogus             127.0.0.1       sz/linux.bogus
```

8.6 Quiero que *bind* se ejecute cuando me desconecto de la red.

He recibido este mail de Ian Clark, `ic@deakin.edu.au` donde explica la forma de hacer esto:

"Ejecuto named en la máquina que hace *masquerading* aquí. Tengo dos archivos `root.cache`, uno llamado `root.cache.real` que contiene el servidor de nombres raíz real y el otro llamado `root.cache.falso` que contiene...

```
; root.cache.falso
; este archivo no contiene informacion
```

Cuando dejo de estar conectado copio el archivo `root.cache.falso` en `root.cache` y reinicio `named`.

Cuando me conecto copio `root.cache.real` en `root.cache` y reinicio `named`.

Esto se hace desde `ip-down` & `ip-up` respectivamente.

La primera vez que efectúo una consulta *off line* sobre un nombre de dominio del cual `named` no tiene detalles, éste pone una entrada como esta en `messages`...

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

con la cual puedo convivir sin problemas.

Esto ciertamente parece funcionarme. Puedo usar el servidor de nombres para máquinas locales mientras no estoy conectado sin el retraso con nombres de dominio externos, y cuando sí estoy conectado, funciona de forma normal con dominios externos."

8.7 ¿Dónde almacena su caché el servidor de nombres? ¿Hay alguna forma de controlar el tamaño del caché?

El caché se almacena en memoria completamente. No se escribe en disco en ningún momento. Cada vez que mata a `named` se pierde el caché. El caché no es controlable de ninguna forma, `named` lo maneja de acuerdo con unas reglas simples. No puede controlar ni el caché ni su tamaño de ninguna forma ni por ningún motivo. Si quiere, puede cambiar esto tocando los fuentes de `named`, lo cual no es recomendable.

8.8 ¿Salva `named` el caché entre reinicios? ¿Puedo guardarlo?

No, `named` no salva el caché cuando muere. Esto significa que el caché se debe reconstruir de nuevo cada vez que mate y reinicie `named`. No hay forma de hacer que `named` salve el caché en un archivo. Si quiere, puede cambiar esto tocando los fuentes de `named`, lo cual no es recomendable.

9 Cómo hacerse un gran *admin* DNS.

Documentación y herramientas.

Existe *Documentación Real*. En línea e impresa. Se requiere la lectura de esta documentación para seguir los pasos de pequeño a gran *admin* DNS. En formato impreso el libro estándar es *DNS and BIND* de C. Liu y P. Albitz de *O'Reilly & Associates*, Sebastopol, CA, ISBN 0-937175-82-X. Lo leí, es excelente. Hay también una sección sobre DNS en *TCP/IP Network Administration*, de Craig Hunt de *O'Reilly...*, ISBN 0-937175-82-X. Otros libros necesarios para un Buenos Administradores DNS (o bueno para cualquier cosa de la materia) es *Zen and the Art of Motorcycle Maintenance* de Robert M. Prisig `:-)` Disponible con ISBN 0688052304 y otros.

Puede encontrar material en línea en `http://www.dns.net/dnsrd/`, `http://www.vix.com/isc/bind/`; una *PUF*, un manual de referencia (*BOG*; *Bind Operations Guide*) así como *papers* y definición de protocolos y diversos retoques o *hacks* de DNS (éstos y la mayoría, si no todas las referencias mencionadas arriba, están también contenidas en la distribución de *bind*). No he leído la mayoría, pero tampoco soy un gran admin DNS. Arnt Gulbrandsen, por otra parte ha leído el *BOG* y está extasiado con él `:-)`. El grupo de noticias `comp.protocols.tcp-ip.domains` es sobre DNS. En suma, hay un gran número de `RFC`s sobre DNS, las más importantes son probablemente las siguientes:

RFC 2052

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

RFC 1918

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

RFC 1912

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

RFC 1713

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

RFC 1712

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

RFC 1183

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

RFC 1035

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

RFC 1034

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

RFC 1033

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

RFC 1032

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

RFC 974

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

10 Traducción

Este documento ha sido traducido por

Pedro Pablo Fábrega Martínez, `pfabrega@arrakis.es`

Si encontráis mejoras, añadidos o fallos, de cualquier tipo, indicádmelo para mejorar el documento.

11 Anexo: El INSFLUG

El *INSFLUG* forma parte del grupo internacional *Linux Documentation Project*, encargándose de las traducciones al castellano de los Howtos (Comos), así como la producción de documentos originales en aquellos casos en los que no existe análogo en inglés.

En el **INSFLUG** se orienta preferentemente a la traducción de documentos breves, como los *COMOs* y *PUFs* (**P**reguntas de **U**so **F**recuente, las *FAQs*. `:)`), etc.

Diríjase a la sede del INSFLUG para más información al respecto.

En la sede del INSFLUG encontrará siempre las **últimas** versiones de las traducciones: `www.insflug.org`. Asegúrese de comprobar cuál es la última versión disponible en el Insflug antes de bajar un documento de un servidor réplica.

Se proporciona también una lista de los servidores réplica (*mirror*) del Insflug más cercanos a Vd., e información relativa a otros recursos en castellano.

Francisco José Montilla, `pacopepe@insflug.org`.