elbauldelprogramador.com

Cómo configurar un servidor DNS

Alejandro Alcalde

7-8 minutos

> **¿Has visto algún error?**: Por favor, ayúdame a corregirlo contactando conmigo.

- Cómo configurar un servidor DNS - Parte 1 (Introducción)
- Cómo configurar un servidor DNS - Parte 2 (La Zona Primaria)
- Cómo configurar un servidor DNS - Parte 3 (Zona Inversa y DNS secundario)

> En esta serie de artículos, intentaré explicar lo mejor posible el funcionamiento de los servidores DNS, y cómo configurar el tuyo propio. Habrá una parte más teórica sobre el funcionamiento del sistema, que es una traducción de un artículo en howtoforge.
>
> Ya que los artículos están basados en distintas fuentes de información que he ido recopilando, no sé de cuantas partes estará formada esta serie, así que la lista de arriba irá cambiando hasta que estén completos todos los artículos.

Debo reconocer que el tema de los DNS me ha dado

muchos problemas, es algo que para mí ha sido dificil de entender. A base de leer y releer muchos artículos por internet, aprendí a configurar un servidor DNS manualmente. Hoy voy a explicar cómo.

En Linux, **BIND** es el encargado de gestionar los DNS, como su página de ayuda indica (*bind - bind a name to a socket*), asocia un nombre a un socket. Es importante que antes de continuar compruebes que la versión de **BIND** es superior a la 4. Lo ideal sería tener la 8 o 9. Puedes comprobarlo con el siguiente comando:

```
$ nslookup -type=txt -class=chaos
version.bind servidor
Server:     servidor
Address:    IP#53

version.bind   text = "VERSION"
```

BIND tiene tres componentes, el primero es llamado *named* o *name-dee*, es un demonio que ejecuta el lado servidor del DNS.

El segundo componente es llamado *resolver library* o biblioteca de resolución, encargada de realizar peticiones a servidores DNS para intentar traducir un nombre a una dirección IP. El archivo de configuración para este componente es **resolv.conf**.

El tercer y último componente de **BIND** proporciona herramientas para probar el servidor DNS. Entre estas herramientas se encuentran comandos como **dig**, que se verá más adelante.

DNS es una base de datos distribuida. Cuando pagas por un dominio, debes configurar dos servidores de nombres, y éstos deben ser registrados en el sistema DNS.

La base de datos del sistema DNS tiene tres niveles. Al primer grupo de servidores se les llama "**servidores root**". Al segundo, "**Top Level Domains (TLDs)**" o dominios de primer nivel. Cuando se necesita conocer la dirección de una web, el segundo componente de **BIND** (resolver library) realiza una petición, (De aquí en adelante lo llamaremos *resolver*).

Por ejemplo, supongamos que quieres encontrar a **google.com**. Tu resolver pide a los servidores root que identifique la IP de google.com. El servidor root responde, "*No lo sé, pero sí sé donde puedes encontrar la respuesta, comienza con los servidores TLD para COM*".

Así, el servidor root envia la petición a un servidor COM. Éste último servidor dice: "*No tengo esa información, pero sé de un servidor de nombres que sí, tiene dirección 173.194.34.6 y nombre ns1.google.com. Dirígete a esa dirección y te dirá la dirección del sitio web google.com.*"

En la figura de arriba, la parte superior izquierda representa los servidores root. En la jerga DNS, éstos servidores reprensentan el comienzo del camino en el sistema DNS. Suelen representarse con un punto ("."). En los archivos de configuración, el mapeo entre IP y nombre acabará en un punto. A lo largo de esta series de artículos quedará más claro este concepto.

Los servidores root son los principales de la base de datos distribuida. Poseen información sobre los **Top Level Domains (TLDs)** o dominios de primer nivel. En los TLDs se incluyen *com, net, org, mil, gov, edu etc*. Al contratar un nombre de dominio, es necesario elegir qué TLD se desea, este blog se encuentra en el espacio de nombres COM y se llama *elbauldelprogramador.com*.

En este punto es donde **BIND** entra en acción. El primer componente que mencionamos, **named**; está presente en todos los servidores de nombres y es el encargado de responder a las peticiones de los resolvers. Lee sus datos del archivo de configuración *named.conf*. Éste fichero obtiene su información de unos ficheros a los que se les suele llamar *zone files* ó *ficheros de zona*. Existen multidud de ellos, pero un archivo de zona en particular mantiene una base de datos de registros que proporciona named con la mayoría de sus respuestas.

En la figura 2, *named* ha recibido una petición. Busca en su fichero de configuración *named.conf*, que busca en el archivo de zona primaria y pasa la información solicitada al resolver desde el exterior.

Figura 2 - Respondiendo a una petición

El proceso *named* escucha en el puerto 53 en los sitemas Linux. Al recibir una petición para una dirección, busca en el primer archivo de configuración que conoce, *named.conf*. Tal y como se aprecia en la figura 2.

El fichero tiene la siguiente estructura:

```
// This is the primary configuration file
for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9
/README.Debian.gz for information on the
// structure of BIND configuration files in
Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do
that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-
zones";
```

Veamos el contenido de los tres archivos que incluye:

named.conf.options

```
options {
    directory "/var/cache/bind";
};
```

Aquí se definen el directorio por defecto para named.

named.conf.default-zones

```
zone "." {
   type hint;
  file "/etc/bind/db.root";
};
```

```
zone "localhost" {
   type master;
    file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
   type master;
    file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
   type master;
    file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
   type master;
    file "/etc/bind/db.255";
};
```

zone “**.**” contiene los nombres y direcciones de los servidores root. Como se mencionó arriba, éstos servidores saben en qué servidores autorizados existe tu dominio — Siendo el primero los TLD (com, org, net etc) y el segundo el servidor autorizado para tu dominio.

zone “**localhost**”. Cada servidor de nombres administra su propio dominio loopback (127.0.0.1). El motivo de crear una zona para localhost es reducir tráfico y permitir que el

mismo software funcione en el sistema como lo hace en internet.

El resto de las zonas son archivos de zonas inversas. Es una copia invertida de la base de datos definida en los otros archivos. Es decir, asocia una IP a un nombre, al contrario. Se pueden indentificar por la extensión **in-addr.arpa**.

En el siguiente artículo ser verá en detalle la estructura del archivo **named.conf.local**, en el que se definen nuevas zonas que corresponden a dominios que resolverá el servidor DNS. Así como a los archivos *pri.nombredominio.com* asociados a cada zona.

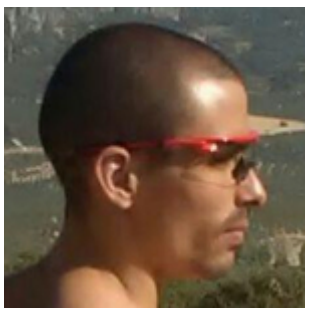
- *Traditional DNS Howto* »» Visitar sitio
- Cómo configurar un servidor DNS - Parte 2 (La Zona Primaria)
- Cómo configurar un servidor DNS - Parte 3 (Zona Inversa y DNS secundario)
- Dig - Chuleta básica de comandos
- Cómo instalar debian desde un USB
- Cómo ocultar la versión de BIND (Y cómo averiguarla)

elbauldelprogramador.com

Cómo configurar un servidor DNS

Alejandro Alcalde

6-7 minutos



> **¿Has visto algún error?**: Por favor, ayúdame a corregirlo contactando conmigo.

- Cómo configurar un servidor DNS - Parte 1 (Introducción)

- Cómo configurar un servidor DNS - Parte 2 (La Zona Primaria)
- Cómo configurar un servidor DNS - Parte 3 (Zona Inversa y DNS secundario)

  Siguiendo con los artículos de cómo configurar un servidor DNS. En el anterior artículo dejamos pendiente echar un vistazo al archivo **named.conf.local**, que contiene información sobre los dominios que serán resueltos por el servidor DNS. Veamos el contenido:

```
zone "elbauldelprogramador.com" {
        type master;
        allow-transfer {DNS_SECUNDARIO;};
        file "/etc/bind
/pri.elbauldelprogramador.com";
};
```

  El contenido de /**etc/bind/pri.elbauldelprogramador.com**:

```
$TTL        3600
@       IN      SOA
ks3277174.kimsufi.com.
correo.electronico.com. (
                        2013011703      ;
serial, todays date + todays serial #
                        7200             ;
refresh, seconds
                        540             ;
retry, seconds
                        604800
; expire, seconds
```

```
                            86400 )            ;
minimum, seconds
;

elbauldelprogramador.com. 3600 A
5.39.89.44
elbauldelprogramador.com. 3600      MX    10
mail.elbauldelprogramador.com.
elbauldelprogramador.com. 3600      NS
ks3277174.kimsufi.com.
elbauldelprogramador.com. 3600      NS
ns.kimsufi.com.
mail 3600 A         5.39.89.44
www 3600 A        5.39.89.44
```

SOA es el acrónimo para *"Start Authority"*. Si recuerdas la figura 1 del artículo anterior, recordarás que DNS es una base de datos distrubuida. Comenzando en los root servers, las peticiones se van desplazando hasta llegar a su destino, en este caso, hasta llegar al servidor DNS que estamos configurando. Por esa razón, en el fichero de zona es necesario indicar dónde comienza su autoridad(*authority*). Ésta autoridad comienza precisamente en el fichero de zona. Los servidores **TLD** (*Top Level Domain ó Dominios de primer nivel*) esperan del servidor DNS que realice su parte del trabajo.

El registro **SOA** consta de varios campos. Es necesario proporcionar datos a esos campos para que otros servidores en internet puedan llevar a cabo sus peticiones.

Los campos son:

Define el nombre principal de la zona. El *@* es una abreviatura a la zona actual, es decir, para */pri.elbauldelprogramador.com* en este ejemplo. El nombre del servidor maestro para esta zona es ks3277174.kimsufi.com. Esto significa que en el archivo *named.conf* existe una entrada que apunta y este archivo vuelve apunta a su vez a la entrada en el archivo de configuración.

Existen varios tipos de clases DNS. En nuestro caso solo se usará la clase *IN* o *Internet*, usadas para definir el mapeo entre la dirección IP y *BIND*.

El tipo de registro para el recurso DNS, en el ejemplo de arriba, el tipo es *SOA*.

Nombre completo del servidor de nombres primario. Debe acabar en un punto.

Dirección de correo de la persona responsable del dominio. Nota cómo se sustituye el símbolo @ por un punto.

Normalmente tiene el formato *YYYYMMDD* con dos dígitios más al final que indican el número de serie del día. El número de serie es útil para indicarle a servidor DNS secundario cuando debe actualizarse. Si el servidor esclavo al comprobar el número de serie ha cambiado, realizará una trasnferencia de zona (**zone transfer**).

En este campo indica al servidor DNS esclavo o secundario con qué frecuencia debería comprobar el estado del

maestro. El valor está representado en segundos. En cada ciclo de refresco, el esclavo realiza la comprobación para saber cuando es necesaria una transferencia de zona (**zone transfer**). En el ejemplo el valor es 7200

Frecuencia con la que el esclavo debería conectarse al maestro en caso de una conexión fallida.

Cantidad total de tiempo en la que el esclavo debería reintentar ponerse en contacto con el maestro antes de que expiren los datos que contiene. Referencias futuras serán dirigidas a los servidores root.

Este campo define el tiempo de vida (*Time To Live*) para el dominio en segundos. Sirve para responder a peticiones de subdominios que no existen en los registros. Cuando esté configurado, el servidor DNS responderá con una respuesta del tipo **no domain** o **NXDOMAIN**. Dicha respuesta será cacheada. El TTL establece la duración del cacheo para la respuesta.

Después de estos campos, se especifican los servidores de nombres para el dominio. **NS** es el acrónimo de **Name Server**. Como se ha visto un poco más arriba, el servidor de nombres principal del ejemplo es *ks3277174.kimsufi.com*. También se define el sevidor DNS secundario o esclavo, en este caso *ns.kimsufi.com*.

Además de los registros *NS*, se definen los registros **MX**, que identifican el servidor de correo para el dominio, el número 10 define la prioridad del servidor de correo. Así como el registro de tipo **A**, que asocia un nombre a una dirección ip.

En el ejemplo existe un único registro **MX**, pero puede haber más. Por ejemplo:

```
MX 10 mail.elbauldelprogramador.com.
MX 20 mail.otrodominio.com.
```

Si se envia un email al dominio, el servidor de correo que envía el email intenta conectarse a *mail.elbauldelprogramador.com* ya que tiene prioridad 10, si no puede establecer conexión, lo intentará con *mail.otrodominio.com*.

El último tipo de registro que vamos a ver es el de tipo **CNAME** (*Canonical Name*). Se suele referir a ellos como registros alias del tipo **A**. Por ejemplo:

significa que *ftp.elbauldelprogramador.com* es un alias de *www.elbauldelprogramador.com*. Es decir, *ftp.elbauldelprogramador.com* apunta al mismo servidor que *www.elbauldelprogramador.com*. Un registro **CNAME** debe apuntar a un registro de tipo **A** y solo de tipo **A**.

En el siguiente artículo se verá el archivo de zona inversa y la configuración del servidor DNS secundario, así como el uso del comando *dig*.

- *Traditional DNS Howto* »» Visitar sitio
- Cómo configurar un servidor DNS - Parte 3 (Zona Inversa y DNS secundario)
- Cómo configurar un servidor DNS - Parte 1 (Introducción)
- Dig - Chuleta básica de comandos
- Cómo instalar debian desde un USB

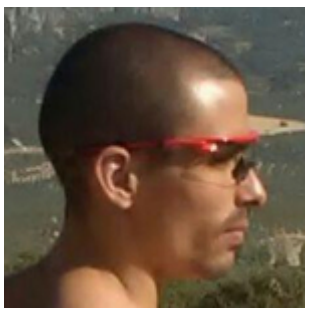
- Cómo ocultar la versión de BIND (Y cómo averiguarla)

elbauldelprogramador.com

Cómo configurar un servidor DNS

Alejandro Alcalde

5-6 minutos



> **¿Has visto algún error?**: Por favor, ayúdame a corregirlo contactando conmigo.

- Cómo configurar un servidor DNS - Parte 1 (Introducción)

- Cómo configurar un servidor DNS - Parte 2 (La Zona Primaria)
- Cómo configurar un servidor DNS - Parte 3 (Zona Inversa y DNS secundario)

  Ya se ha visto que existe una base de datos centralizada que asocia nombres de dominios a direccines IP, también se mencionó el caso inverso, una copia inversa de dicha base de datos, que asocia IP's a nombres de dominios. Ésta búsqueda inversa es usada por muchos programas, que rechazarán establecer una conexión si la búsqueda inversa y la búsqueda directa (*Dominio»IP*) no coinciden. Muchos proveedores de correo usan la búsqueda inversa para clasificar correos como spam.

  Con objetivo de que los emails enviados desde el dominio que se está configurando no sean clasificados como spam, es necesario crear la zona inversa en el archivo **named.conf.local**:

```
zone "89.39.5.in-addr.arpa" {
 type master;
    file "pri.89.39.5.in-addr.arpa";
};
```

  Los números son la dirección ip del servidor escritos en orden inverso. Es decir, la ip es **5.39.89.x**, así pues, la zona ha de llamarse *89.39.5.in-addr.arpa*.

  Es necesario crear el archivo de zone inversa también, *pri.89.39.5.in-addr.arpa*. Este archivo es necesario crearlo en el mismo directorio en el que se encuentra el archivo de

zona primario (*pri.elbauldelprogramador.com*).

El principio de este archivo es exáctamente igual que *pri.elbauldelprogramador.com*:

```
@       IN      SOA
ks3277174.kimsufi.com.
contacto.elbauldelprogramador.com. (
                        2013021001      ;
serial, todays date + todays serial #
                        7200             ;
refresh, seconds
                        540             ;
retry, seconds
                        604800
; expire, seconds
                        86400 )           ;
minimum, seconds
;
  NS      ks3277174.kimsufi.com.
  NS      ns.kimsufi.com.
```

A continuación, es necesario añadir un registro del tipo **PTR**. Los registros **PTR** son punteros. Apuntan a un nombre de dominio. Quedaría así:

```
44   PTR elbauldelprogramador.com.
```

El 44 es el último valor de la dirección IP del servidor.

Eso es todo, en este punto usaremos el comando **dig** para comprobar la configuración.

```
$ dig elbauldelprogramador.com

; <<>> DiG 9.8.4-P1 <<>>
elbauldelprogramador.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status:
NOERROR, id: 10156
;; flags: qr rd ra; QUERY: 1, ANSWER: 1,
AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;elbauldelprogramador.com.    IN  A

;; ANSWER SECTION:
elbauldelprogramador.com. 532    IN  A
5.39.89.44

;; Query time: 50 msec
;; SERVER: 80.58.61.250#53(80.58.61.250)
;; WHEN: Mon Feb 11 21:09:28 2013
;; MSG SIZE  rcvd: 58
```

Así, estamos buscando la ip del dominio. Como se aprecia, devuelve el valor correcto en la sección **ANSWER SECTION**.

```
$ dig -x 5.39.89.44

; <<>> DiG 9.8.4-P1 <<>> -x 5.39.89.44
```

```
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status:
NOERROR, id: 50347
;; flags: qr rd ra; QUERY: 1, ANSWER: 1,
AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;44.89.39.5.in-addr.arpa.    IN  PTR

;; ANSWER SECTION:
44.89.39.5.in-addr.arpa. 84513 IN  PTR
elbauldelprogramador.com.

;; Query time: 52 msec
;; SERVER: 80.58.61.250#53(80.58.61.250)
;; WHEN: Mon Feb 11 21:10:09 2013
;; MSG SIZE  rcvd: 76
```

Esta vez, se está realizando la petición inversa, preguntamos por el dominio.

En caso de disponer de otro servidor DSN propio, para configurarlo de modo que haga las veces de servidor DNS secundario es necesario añadir otra zona al archivo **named.conf.local** en el servidor **secundario**

```
zone "DOMINIO" {
     type slave;
     file "sec.DOMINIO.COM";
     masters { DIRECCION IP SERVIDOR
```

```
PRIMARIO; };
};
```

Esta vez, se declara la zona como **slave** o esclava y se especifica la dirección IP del servidor maestro. En el fichero indicado en **file** se almacenarán los datos de la zona esclava. Basta con reiniciar **named** y dicho fichero será creado al ponserse en contacto con el servidor primario y habiendo realizado una transferencia de zona.

Por último, por razones de seguridad es recomendable agregar una línea adicional en archivo de zona del servidor **principal** que únicamente permita al servidor secundario realizar la transferencia de zona:

```
zone "elbauldelprogramador.com" {
        type master;
        allow-transfer {IP SERVIDOR DNS
SECUNDARIO;};
        file "/etc/bind
/pri.elbauldelprogramador.com";
};
```

En mi caso, la ip corresponde al servidor DNS secundario que proporciona la compañia en la que tengo contratado el servidor.

Con éste último artículo doy por terminada este conjunto de artículos que pretendían dar a conocer al lector el funcionamiento de DNS. Habiendo adquirido este conocimiento, será mucho más fácil usar las distintas herramientas que proporcionan los paneles de administración web para configurar los DNS.

Para finalizar, reiterar que todos los artículos están basados en un How to de la web que se menciona en las referencias.

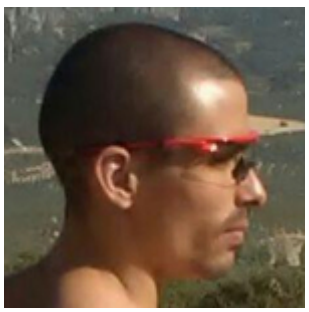
- *Traditional DNS Howto* »» Visitar sitio
- Cómo configurar un servidor DNS - Parte 2 (La Zona Primaria)
- Cómo configurar un servidor DNS - Parte 1 (Introducción)
- Dig - Chuleta básica de comandos
- Cómo instalar debian desde un USB
- Cómo ocultar la versión de BIND (Y cómo averiguarla)

elbauldelprogramador.com

Dig - Chuleta básica de comandos

Alejandro Alcalde

3-4 minutos



¿Has visto algún error?: Por favor, ayúdame a corregirlo contactando conmigo.

Dig (Domain Information Groper) es un comando de gran utilidad para realizar consultas a registros DNS. Se usa mucho — entre otras cosas — para el diagnóstico de Servidores DNS. Desde que administro yo mismo el servidor del blog lo he usado bastante, y hoy voy a dar unas nociones básicas de cómo usarlo.

En un comentario hecho en la página de Google+ del blog un usuario indicó que es posible obtener una salida simplificada de todos los comandos de abajo con la opción *+short*.

El ejemplo más básico es:

La sintaxis es:

```
dig @servidor-dns ejemplo.com tipo-de-registro
```

Ejemplo de uso:

```
dig @208.67.222.222 google.com A

;; QUESTION SECTION:
;google.com.          IN  A

;; ANSWER SECTION:
google.com.      300 IN  A   173.194.34.231
google.com.       300 IN  A   173.194.34.226
google.com.       300 IN  A   173.194.34.230
google.com.       300 IN  A   173.194.34.228
google.com.       300 IN  A   173.194.34.238
google.com.       300 IN  A   173.194.34.229
google.com.       300 IN  A   173.194.34.227
google.com.       300 IN  A   173.194.34.225
google.com.       300 IN  A   173.194.34.232
google.com.       300 IN  A   173.194.34.224
google.com.       300 IN  A   173.194.34.233
```

```
dig @208.67.222.222 google.com NS

;; QUESTION SECTION:
;google.com.            IN  NS

;; ANSWER SECTION:
google.com.     172749  IN  NS
```

```
ns2.google.com.
google.com.      172749  IN  NS
ns1.google.com.
google.com.      172749  IN  NS
ns3.google.com.
google.com.      172749  IN  NS
ns4.google.com.
```

```
dig @208.67.222.222 google.com MX

;; QUESTION SECTION:
;google.com.            IN  MX

;; ANSWER SECTION:
google.com.     469 IN  MX  20
alt1.aspmx.l.google.com.
google.com.      469 IN  MX  10
aspmx.l.google.com.
google.com.       469 IN  MX  40
alt3.aspmx.l.google.com.
google.com.      469 IN  MX  50
alt4.aspmx.l.google.com.
google.com.      469 IN  MX  30
alt2.aspmx.l.google.com.
```

```
dig @208.67.222.222 google.com TXT

;; QUESTION SECTION:
;google.com.          IN  TXT
```

```
;; ANSWER SECTION:
google.com.        3600    IN  TXT "v=spf1
include:_spf.google.com ip4:216.73.93.70/31
ip4:216.73.93.72/31 ~all"
```

```
dig any google.com

;; QUESTION SECTION:
;google.com.           IN  ANY

;; ANSWER SECTION:
google.com.        407 IN  MX  40
alt3.aspmx.l.google.com.
google.com.      407 IN  MX  30
alt2.aspmx.l.google.com.
google.com.      407 IN  MX  20
alt1.aspmx.l.google.com.
google.com.      407 IN  MX  10
aspmx.l.google.com.
google.com.       407 IN  MX  50
alt4.aspmx.l.google.com.
google.com.      172781  IN  NS
ns2.google.com.
google.com.      172781  IN  NS
ns1.google.com.
google.com.      172781  IN  NS
ns3.google.com.
google.com.      172781  IN  NS
ns4.google.com.
```

```
dig -x 173.194.34.233

;; QUESTION SECTION:
;233.34.194.173.in-addr.arpa.  IN  PTR

;; ANSWER SECTION:
233.34.194.173.in-addr.arpa. 83265 IN  PTR
mad01s09-in-f9.1e100.net.
```

Puedes encontrar una descripción un poco más extensa sobre el funcionamiento de la consulta inversa en la tercera parte del artículo *Cómo configurar un servidor DNS*

- *dig – Linux DNS Lookup utility cheat sheet* »» linuxaria
- Cómo configurar un servidor DNS - Parte 2 (La Zona Primaria)
- Cómo configurar un servidor DNS - Parte 1 (Introducción)
- Chuleta de comandos para GPG
- Cómo configurar un servidor DNS - Parte 3 (Zona Inversa y DNS secundario)
- Git: Mini Tutorial y chuleta de comandos

eltallerdelbit.com

Crear y configurar un Servidor DNS Bind

6-8 minutos

El Servidor DNS Bind nos permitirá crear nuestro propio servidor DNS en Linux.

En este artículo veremos:

En este caso lo haremos desde la terminal de Ubuntu Server .

1. INSTALACIÓN DE BIND

Para descargar el servidor ***Bind*** podeis visitar el Sitio oficial de Bind.

Aunque lo más normal desde Linux es descargarlo e instalarlo así:

```
apt-get install bind9
```

2. LOS ARCHIVOS MÁS IMPORTANTES DE BIND AL CREAR UN SERVIDOR DNS CON *Bind*

- */etc/bind/**named.conf***: Archivo de configuración principal; permite definir qué archivos serán llamados. Por defecto contiene:

  ```
  include "/etc/bind/named.conf.options";
  include "/etc/bind/named.conf.local";
  include "/etc/bind/named.conf.default-
  zones";
  ```

  Así que *named.conf* permite definir que se aplicarán las opciones que se configuren en el archivo *named.con.options*, se encargará de generar las zonas definidas en *named.conf.local*, y contiene también una llamada al archivo de las zonas por defecto (que serán las de localhost, directa e inversa, la ROOT, y la broadcast).
- */etc/bind/**named.conf.local***: Archivo en el que definimos las zonas del server DNS
- */etc/bind/**named.conf.options***: algunas opciones interesantes para nuestro server DNS (por ejemplo los *forwarders* o reenviadores, que son otros servidores a los que reenviamos las consultas que nuestro server DNS desconoce).
- Archivos de Zona Directa y archivos de Zona Inversa: los llamaremos db.zona_lo_que_sea. En ellos definimos los registros DNS de las zonas definidas en *named.conf.local*, de las cuales seremos *Master* o *Slave*.
- */etc/**resolv.conf***: En este archivo ha de constar la IP del servidor DNS que deseamos resuelva en nuestra máquina.

Introduciremos el servidor DNS al que deseamos hacer peticiones DNS de esta forma (en este caso ponemos nuestro propio server DNS, para probar que funciona):

```
nameserver 192.168.1.10
```

A continuación ,suponiendo que ya tenemos correctamente configuradas las interfaces de red, tendremos que crear las nuevas zonas DNS en Bind (en Linux).

3. CREACIÓN DE ZONAS DNS en named.conf.local

En el archivo ***named.conf.local*** definiremos las nuevas zonas DNS que vamos a crear:

Así que vamos al archivo */etc/bind/named.conf.local* y definimos en él las zonas que vamos a crear, la **zona de busqueda directa DNS** y la **zona de búsqueda inversa** (Abrimos el documento, vamos al final del todo y añadimos las nuevas zonas.

También podemos configurar:

– Si la zona será *Master* o *Slave*

```
type master;
o
type slave;
```

– Si se permite la transferencia de la zona, y a qué servidores

```
allow-transfer { 192.168.1.1; 192.168.1.142; 192.168.1.143 ;};
```

– Si se envía notificación cada vez que cambie la zona.

```
notify yes;
o
notify no;
```

VEAMOS LAS NUEVAS ZONAS DNS EN EL ARCHIVO *named.conf.local*:

```
//
// Do any local configuration here
//
// Consider adding the 1918 zones here, if they are not used in your
// organization

//include "/etc/bind/zones.rfc1918";


zone "midominio.com"{
type master;
file "/etc/bind/db.midominio.com";
allow-transfer { 192.168.1.1; 192.168.1.142; 192.168.1.143 ;};
notify yes;
```

```
};


zone "1.168.192.in-addr.arpa" {
type master;
file "/etc/bind/db.1.168.192";
};
```

4. CREACIÓN Y CONFIGURACIÓN DE ARCHIVOS DE ZONAS DNS DIRECTA/INVERSA

Ahora creamos el archivo de **resolución directa** de la zona *midominio.com*.

Podemos copiar el archivo de zona *db.local*, que hay en el directorio */etc/bind*, y transformarlo para crear la nueva zona DNS.

Como podéis ver, se han creado más registros además del dominio www1 (www2, www3, www4 …). Los usaremos para siguientes ejercicios:

```
;BIND data file for local loopback interface

$TTL    604800
@       IN      SOA     midominio.com.
root.midominio.com. (
```

```
                                2         ;
Serial
                           604800         ;
Refresh
                            86400         ;
Retry
                          2419200         ;
Expire
                           604800 )       ;
Negative Cache TTL
;
@       IN      NS      midominio.com.
@       IN      A       192.168.1.10
www1    IN      A       192.168.1.10
www2    IN      A       192.168.1.10
www3    IN      A       192.168.1.30
www4    IN      A       192.168.1.40
www5    IN      A       192.168.1.40
```

A continuación creamos la **zona de Resolución inversa**:

```
GNU nano 2.2.2           Archivo:
db.1.168.192

;

; BIND data file for local loopback
interface

;
```

```
$TTL    604800

@       IN      SOA     midominio.com.
root.midominio.com. (

2         ; Serial

604800         ; Refresh

86400         ; Retry

2419200         ; Expire

604800 )       ; Negative Cache TTL

;

@       IN      NS      midominio.com.

10      IN      PTR     midominio.com.

10      IN      PTR     www1.midominio.com.

10      IN      PTR     www2.midominio.com.

30      IN      PTR     www3.midominio.com.

40      IN      PTR     www4.midominio.com.
```

```
40      IN      PTR     www5.midominio.com.
```

4.1 Sintáxis y definición de los archivos de zona y sus registros DNS.

La arroba @ significa el propio servidor
El registro NS define el servidor primario

El registro A define la IP del servidor

El registro MX define el servidor de correo

```
@       IN      NS      midominio.com.

@       IN      MX 10       ns1

@       IN      MX 10       mail

@       IN      A       192.168.1.10
```

Después definimos las IP´s de los equipos que componen el dominio. Ejemplo de configuración de archivo de zona:

```
ns1         IN      A       192.168.1.10

mail        IN      A       192.168.1.10
```

4.2 Herramientas para comprobar la sintaxis correcta de los archivos de configuración de Bind

- Tenemos el comando ***named-checkconf named.conf.local***

  Este comando comprobará si hay errores en la sintaxis de las zonas creadas en named.conf.local

- Otro comando muy interesante es ***named-checkzone***

  ```
  named-checkzone midominio.com db.midominio.com
  ```

  ```
  named-checkzone nombre_de_la_zona nombre_del_archivo_de_la_zona
  ```

  ```
  named-checkzone zonename filename
  ```

5. RECARGAR CONFIGURACIÓN DE BIND Y REINICIAR EL SERVICIO PARA APLICAR LOS CAMBIOS

Después de terminar las configuraciones, recargamos la configuración de bind y reiniciamos **el servicio bind** con :

```
/etc/init.d/bind9 reload
/etc/init.d/bind9 restart
```

También podemos ejecutar:

```
rndc reload
```

6. PRUEBAS DE FUNCIONAMIENTO DE LAS NUEVAS ZONAS DNS CREADAS

Resolución del Dominio **DNS** desde un cliente con el comando DIG:

Recordemos que para resolver correctamente los DNS, el cliente ha de tener configurado como resolvedor **DNS** al servidor que acabamos de crear.

Así que nos aseguramos de que en el archivo */etc/resolv.conf* contenga lo siguiente en este caso:

```
nameserver 192.168.1.10
```

Si necesitáis algo más de ayuda, podéis miraros los Conceptos y comandos sobre **DNS**.

Tampoco olvidéis visitar la entrada sobre **Ejercicios de**

Apache y DNS.

eltallerdelbit.com

Comando nslookup | Opciones

5-6 minutos

---

nslookup es "*name server lookup*" (búsqueda de servidores de nombre), y es una herramienta que puede realizar diversos tipos de consultas a servidores de nombres DNS.

Podemos realizar **consultas directas** (nslookup *nombredominio.com*) o consultas reversas (nslookup *IP*).

Respuesta Autoritativa (*Authoritative Answer*) y respuesta no Autoritativa (*Non-Authoritative Answer*)

En las consultas DNS veremos frecentemente estos dos términos.

- *Authoritative Answer* significa que la respuesta DNS se ha producido desde el servidor DNS que tiene todo el archivo de información disponible para esa zona.
- *Non Authoritative Answer* significa que la respuesta DNS se ha producido desde un servidor DNS que tiene en caché una copia de las consultas realizadas para esa zona, al servidor que tiene la Autoridad para responder (el que tiene el archivo de zona). Por esto veremos muy a menudo la

respuesta desde servidores que son *Non Authoritative*.

nslookup en Linux

1. nslookup sin argumentos

Muestra el servidor que realiza la consulta

```
Server: 192.168.130.202
```

que en este caso resulta ser un servidor DNS local (aquí podría aparecer el router de nuestra casa contestando a las consultas DNS).

Y después muestra un servidor DNS (No es el DNS principal, es un DNS no autoritario o *Non Authoritative*) del dominio consultado, que responde a nuestra consulta DNS:

```
Non-Authoritative Answer:

Name: google.es

Address: 216.58.201.131
```

2. nslookup realizando consultas sobre el tipo de registro del que deseamos info.

Con el argumento: *-type=*

- **2.1 -type=any –> muestra todos los registros NS disponibles del dominio.**

es decir:

```
nslookup -type=any google.es
```

Vemos que muestra respuesta de servidores *Non-Authoritative* y de servidores *Authoritative*:

- **2.2 -type=ns –> muestra todos los registros NS disponibles del dominio y una lista de los servidores *Authoritative* de nombre si es que están disponibles (normalmente estarán ocultos y no aparecerán, solo aparecerán los *Non-authoritative*).**

- **2.3 -type=mx –> muestra todos los registros MX (mail) del dominio.**

```
nslookup -type=mx unizar.es
```

Vemos que los registros MX de unizar son:

```
Non-authoritative answer:

mx01.puc.rediris.es

mx02.puc.rediris.es
```

- **2.4 -type=soa –> muestra todos los registros** SOA (Start of Authority) **del dominio.**

  ```
  nslookup -type=soa google.es
  ```

  Mostrará el servidor con autoridad para responder sobre ese dominio. También mostrará los parámetros del registro *SOA*: el TTL (tiempo de respuesta), *refresh*, *retry*, *expire*, *minimum* …

3. nslookup reverso (reverse nslookup)

Podemos realizar una consulta dns inversa utilizando la IP como un argumento para el comando nslookup; y nos mostrará los datos de resolución inversa de la zona

4. nslookup utilizando un servidor específico

En lugar de utilizar los servidores DNS por defecto para la consulta, podemos especificar qué servidor deseamos que resuelva la consulta.

5. nslookup en modo debug

El modo debug de nslookup muestra info sobre los paquetes durante la consulta dns.

6. nslookup en modo interactivo

El modo interactivo nos permite efectuar consultas como las mencionadas hasta ahora, pero de un modo diferente, introduciendo los comandos uno por uno.

Por ej:

```
nslookup
```

y se abrirá la terminal de nslookup para comenzar a introducir comandos:

```
>
```

entonces podemos introducir el nombre de dominio que queremos consultar:

```
google.es
```

y nos mostrará la respuesta.

Otro comando que podemos introducir en modo interactivo, es el de type= , que nos permite elegir qué tipo de registros deseamos que nos muestre la consulta.

lo haremos así:

```
>set type=ns

>google.es
```

Y obtendremos en este caso la consulta de los registros ns del dominio google.es :

7. nslookup en Windows

Básicamente el comando funciona de forma parecida, aunque con alguna pequeña diferencia en los comandos respecto a los utilizados en Linux.

- nslookup directo

- nslookup reverso

- nslookup modo interactivo

  Si ejecutamos simplemente el comando nslookup, lo primero que mostrará es el servidor DNS predeterminado y su IP, y se abrirá el modo interactivo de nslookup:

Después nos permite comenzar a teclear parámetros para ejecutar diversos tipos de consultas DNS, que veremos a continuación.

- Siguiendo con lo anterior, vamos a realizar una consulta con nslookup, y utilizando después algunos parámetros:

  nslookup
  server 8.8.8.8 (le decimos qué servidor queremos que conteste)

  set q=MX (queremos averiguar los registros MX)

  google.es (le decimos de qué NS queremos saber la info)

- Otra opción es la de hacer una consulta con nslookup sobre el tipo de registro DNS del que deseamos info.

  ```
  nslookup -type=ns unizar.es
  ```

Acabamos de hacer un interesante repaso al **comando nslookup**, sus opciones y funcionamiento, tanto en Linux como en Windows. Realmente el comando nslookup ha sido relegado a un segundo plano, sobre todo en Linux (donde ha sido desplazado por el comando *dig*), pero aún así viene bien conocer su funcionamiento. Para más info sobre nslookup, podemos consultar el artículo de Microsoft sobre nslookup

eltallerdelbit.com

dig | Linux (Resolución DNS)

15-18 minutos

Dig es un comando de Linux que nos permite realizar consultas DNS.

Los comandos *nslookup* y *host* son de la misma familia (aunque nslookup está en desuso).

Las siglas de dig significan *Domain Information Groper* (algo así como "*tanteador*" de información de Dominio)

Definición de **dig** desde la página de *man dig* (manual del comando dig en Linux):

dig (domain information groper) is a flexible tool for interrogating DNS name servers.

DIG es una herramienta flexible que muestra los resultados de forma clara. Es parte del paquete **Bind** (Berkeley Internet Name Domain). Otras herramientas de búsqueda DNS no suelen tener tantas funcionalidades como dig.

1. dig sin argumentos

dig a secas, realiza una consulta de los servidores NS raíz

```
root@debian:/etc# dig

; <<>> DiG 9.9.5-9+deb8u10-Debian <<>>
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR,
id: 5298
;; flags: qr rd ra; QUERY: 1, ANSWER: 13,
AUTHORITY: 0, ADDITIONAL: 3

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;.                  IN    NS
```

```
;; ANSWER SECTION:
.            254899    IN    NS    k.root-
servers.net.
.            254899    IN    NS    a.root-
servers.net.
.            254899    IN    NS    j.root-
servers.net.
.            254899    IN    NS    b.root-
servers.net.
.            254899    IN    NS    e.root-
servers.net.
.            254899    IN    NS    h.root-
servers.net.
.            254899    IN    NS    c.root-
servers.net.
.            254899    IN    NS    g.root-
servers.net.
.            254899    IN    NS    m.root-
servers.net.
.            254899    IN    NS    d.root-
servers.net.
.            254899    IN    NS    i.root-
servers.net.
.            254899    IN    NS    f.root-
servers.net.
.            254899    IN    NS    l.root-
servers.net.

;; ADDITIONAL SECTION:
a.root-servers.net.    79533    IN    A
198.41.0.4
g.root-servers.net.    94186    IN    A
```

```
192.112.36.4

;; Query time: 3 msec
;; SERVER: 192.168.130.202#53(192.168.130.202)
;; WHEN: Tue Jun 06 12:45:15 CEST 2017
;; MSG SIZE  rcvd: 271
```

A continuación podemos ver que la consulta ***dig*** muestra los servidores raíz.

Podemos comprobar los servidores raíz desde este enlace y desde root-servers.org.

2. El uso más normal de *dig*: *dig dominio* como argumento

```
ej: dig google.com
```

Con este tipo de consulta, dig mostrará el registro A

```
root@debian:/etc# dig google.com

; <<>> DiG 9.9.5-9+deb8u10-Debian <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR,
id: 21115
;; flags: qr rd ra; QUERY: 1, ANSWER: 1,
AUTHORITY: 13, ADDITIONAL: 16

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;google.com.            IN    A

;; ANSWER SECTION:
google.com.        103    IN    A
216.58.201.142

;; AUTHORITY SECTION:
com.            171958    IN    NS    f.gtld-
servers.net.
com.            171958    IN    NS    j.gtld-
servers.net.
com.            171958    IN    NS    c.gtld-
servers.net.
com.            171958    IN    NS    a.gtld-
servers.net.
com.            171958    IN    NS    h.gtld-
servers.net.
com.            171958    IN    NS    i.gtld-
servers.net.
```

```
com.            171958    IN    NS    e.gtld-
servers.net.
com.            171958    IN    NS    k.gtld-
servers.net.
com.            171958    IN    NS    b.gtld-
servers.net.
com.            171958    IN    NS    l.gtld-
servers.net.
com.            171958    IN    NS    d.gtld-
servers.net.
com.            171958    IN    NS    m.gtld-
servers.net.
com.            171958    IN    NS    g.gtld-
servers.net.

;; ADDITIONAL SECTION:
a.gtld-servers.net.    171958    IN    A
192.5.6.30
a.gtld-servers.net.    171958    IN    AAAA
2001:503:a83e::2:30
b.gtld-servers.net.    171958    IN    A
192.33.14.30
b.gtld-servers.net.    171958    IN    AAAA
2001:503:231d::2:30
c.gtld-servers.net.    171958    IN    A
192.26.92.30
d.gtld-servers.net.    171958    IN    A
192.31.80.30
e.gtld-servers.net.    171958    IN    A
192.12.94.30
f.gtld-servers.net.    171958    IN    A
192.35.51.30
g.gtld-servers.net.    171958    IN    A
```

```
192.42.93.30
h.gtld-servers.net.    171958    IN    A
192.54.112.30
i.gtld-servers.net.    171958    IN    A
192.43.172.30
j.gtld-servers.net.    171958    IN    A
192.48.79.30
k.gtld-servers.net.    171958    IN    A
192.52.178.30
l.gtld-servers.net.    171958    IN    A
192.41.162.30
m.gtld-servers.net.    171958    IN    A
192.55.83.30

;; Query time: 7 msec
;; SERVER: 192.168.130.202#53(192.168.130.202)
;; WHEN: Tue Jun 06 12:55:16 CEST 2017
;; MSG SIZE  rcvd: 543
```

Consulta: *dig dominio*

Será la sección *ANSWER SECTION* la que mostrará la info que precisamos.

De esta forma dig mostrará:

- si la resolución ha sido correcta, mostrará status: *NOERROR*
- Mostrará la sección que ha respondido a la interrogación sobre el DNS
- Muestra el tiempo de respuesta de la consulta: *Query time: 8 msec*
- Y el servidor que responde a esa consulta *DNS: SERVER: 192.168.130.202#53* (se trata de un servidor DNS local, configurado como servidor DNS caché con Bind, que ejecuta reenvíos de consultas al servidor DNS primario de la red, el cual le devuelve el resultado de esas consultas y le permite guardarlos

para mostrarlos y guardarlos en caché).

2.1 dig a un dominio, usando como argumento un NS concreto (nameserver)

Es interesante poder utilizar el **comando dig** con otros parámetros y argumentos, como el argumento *NS*, que nos permite realizar una consulta dig de un dominio, pero utilizando un *NS* (*Name Server*) concreto, cuya IP pondremos en la consulta:

```
dig @8.8.8.8 google.com
```

De esta forma dig consulta los DNS de google.com, al servidor 8.8.8.8

Y vemos cómo el NS que ha respondido a la petición es el server 8.8.8.8

```
root@debian:/etc# dig @8.8.8.8 google.com

; <<>> DiG 9.9.5-9+deb8u10-Debian <<>> @8.8.8.8
google.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR,
id: 3070
```

```
;; flags: qr rd ra; QUERY: 1, ANSWER: 3,
AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;google.com.            IN    A

;; ANSWER SECTION:
google.com.        299    IN    A
216.58.201.142
google.com.        299    IN    A
216.58.201.142
google.com.        299    IN    A
216.58.201.142

;; Query time: 41 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Tue Jun 06 13:06:14 CEST 2017
;; MSG SIZE  rcvd: 87
```

2.2 Con dig también podemos especificar el tipo de registro que deseamos consultar (*ANY*, *A*, *MX*, *NS* …)

Vamos a ver un ejemplo buscando los registros MX (correo).

```
dig MX @8.8.8.8 google.com
```

```
root@debian:/etc# dig MX @8.8.8.8 google.es

; <<>> DiG 9.9.5-9+deb8u10-Debian <<>> MX
@8.8.8.8 google.es
; (1 server found)
;; global options: +cmd
;; Got answer:
```

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR,
id: 53401
;; flags: qr rd ra; QUERY: 1, ANSWER: 5,
AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;google.es.             IN    MX

;; ANSWER SECTION:
google.es.         599    IN    MX    30
alt2.aspmx.l.google.com.
google.es.         599    IN    MX    10
aspmx.l.google.com.
google.es.         599    IN    MX    50
alt4.aspmx.l.google.com.
google.es.         599    IN    MX    20
alt1.aspmx.l.google.com.
google.es.         599    IN    MX    40
alt3.aspmx.l.google.com.

;; Query time: 35 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Tue Jun 06 13:45:29 CEST 2017
;; MSG SIZE  rcvd: 156
```

2.3 Consultando TODOS los tipos de registros (*ANY*)

Con la opción *ANY* consultaremos TODOS los registros de un dominio.

```
dig ANY google.es
```

```
root@debian:/etc# dig ANY google.es

; <<>> DiG 9.9.5-9+deb8u10-Debian <<>> ANY
google.es
;; global options: +cmd
;; Got answer:
```

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR,
id: 4952
;; flags: qr rd ra; QUERY: 1, ANSWER: 13,
AUTHORITY: 4, ADDITIONAL: 5

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;google.es.            IN    ANY

;; ANSWER SECTION:
google.es.        293    IN    AAAA
2a00:1450:4003:807::2003
google.es.        293    IN    TXT    “v=spf1
-all”
google.es.        593    IN    MX    20
alt1.aspmx.l.google.com.
google.es.        593    IN    MX    10
aspmx.l.google.com.
google.es.        593    IN    MX    50
alt4.aspmx.l.google.com.
google.es.        593    IN    MX    40
alt3.aspmx.l.google.com.
google.es.        593    IN    MX    30
alt2.aspmx.l.google.com.
google.es.        53    IN    SOA
ns4.google.com. dns-admin.google.com. 158122735
900 900 1800 60
google.es.        233    IN    A
216.58.210.131
google.es.        72196    IN    NS
ns4.google.com.
google.es.        72196    IN    NS
```

```
ns1.google.com.
google.es.        72196    IN    NS
ns3.google.com.
google.es.        72196    IN    NS
ns2.google.com.

;; AUTHORITY SECTION:
google.es.        72196    IN    NS
ns2.google.com.
google.es.        72196    IN    NS
ns3.google.com.
google.es.        72196    IN    NS
ns4.google.com.
google.es.        72196    IN    NS
ns1.google.com.

;; ADDITIONAL SECTION:
ns1.google.com.        69377    IN    A
216.239.32.10
ns2.google.com.        84802    IN    A
216.239.34.10
ns3.google.com.        170862    IN    A
216.239.36.10
ns4.google.com.        170862    IN    A
216.239.38.10

;; Query time: 8 msec
;; SERVER: 192.168.130.202#53(192.168.130.202)
;; WHEN: Tue Jun 06 14:12:21 CEST 2017
;; MSG SIZE  rcvd: 462
```

4. RESOLUCIÓN INVERSA: Resuelve de IP a nombre

Con el parámetro -x podremos realizar la **resolución inversa de un dominio**. Esta resolución inversa nos mostrará por tanto la IP del dominio. Si no añadimos más argumentos, la resolución inversa será del registro A.

ejemplo:

```
dig -x 8.8.8.8
```

```
root@debian:/home/# dig -x 8.8.8.8

; <<>> DiG 9.9.5-9+deb8u11-Debian <<>> -x 8.8.8.8
;; global options: +cmd
```

```
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR,
id: 23544
;; flags: qr rd ra; QUERY: 1, ANSWER: 1,
AUTHORITY: 13, ADDITIONAL: 14

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;8.8.8.8.in-addr.arpa. IN PTR

;; ANSWER SECTION:
8.8.8.8.in-addr.arpa. 75915 IN PTR google-public-
dns-a.google.com.

;; AUTHORITY SECTION:
. 246702 IN NS i.root-servers.net.
. 246702 IN NS l.root-servers.net.
. 246702 IN NS e.root-servers.net.
. 246702 IN NS f.root-servers.net.
. 246702 IN NS k.root-servers.net.
. 246702 IN NS j.root-servers.net.
. 246702 IN NS h.root-servers.net.
. 246702 IN NS c.root-servers.net.
. 246702 IN NS g.root-servers.net.
. 246702 IN NS a.root-servers.net.
. 246702 IN NS d.root-servers.net.
. 246702 IN NS m.root-servers.net.
. 246702 IN NS b.root-servers.net.

;; ADDITIONAL SECTION:
a.root-servers.net. 83834 IN A 198.41.0.4
b.root-servers.net. 83907 IN A 192.228.79.201
```

```
c.root-servers.net. 83948 IN A 192.33.4.12
d.root-servers.net. 83989 IN A 199.7.91.13
e.root-servers.net. 84030 IN A 192.203.230.10
f.root-servers.net. 84135 IN A 192.5.5.241
g.root-servers.net. 84176 IN A 192.112.36.4
h.root-servers.net. 84217 IN A 198.97.190.53
i.root-servers.net. 84258 IN A 192.36.148.17
j.root-servers.net. 84427 IN A 192.58.128.30
k.root-servers.net. 84468 IN A 193.0.14.129
l.root-servers.net. 84509 IN A 199.7.83.42
m.root-servers.net. 83793 IN A 202.12.27.33

;; Query time: 2 msec
;; SERVER: 192.168.130.202#53(192.168.130.202)
;; WHEN: Thu Jul 13 13:23:23 CEST 2017
;; MSG SIZE rcvd: 512
```

- Vamos a ver también el ejemplo con el dominio *Linkedin.com*, primero resolución directa y luego realizaremos la resolución inversa:

  ```
  dig linkedin.com
  ```

  Ahora realizamos la resolución inversa de la IP que nos ha aparecido:

5. RESOLUCIÓN DE ERRORES DNS: DEBUG DNS (*dig +trace*)

Podemos realizar un rastreo en la ruta de búsqueda DNS con la opción *+ trace*.

Como se muestra a continuación al consultar *google.es* podemos ver lo que realmente sucede:
– Primero se muestran los servidores de nombres raíz ‘.’

– Después se rastrean los **servidores de nombres para el dominio .es** y, finalmente, se devuelven los **servidores de nombres** de *google.es*, seguidos de los **registros DNS** de la misma.

```
dig +trace google.es
```

```
root@debian:/etc# dig +trace google.es

; <<>> DiG 9.9.5-9+deb8u10-Debian <<>> +trace
```

```
google.es
;; global options: +cmd
.            249366    IN    NS    j.root-
servers.net.
.            249366    IN    NS    i.root-
servers.net.
.            249366    IN    NS    d.root-
servers.net.
.            249366    IN    NS    f.root-
servers.net.
.            249366    IN    NS    l.root-
servers.net.
.            249366    IN    NS    a.root-
servers.net.
.            249366    IN    NS    e.root-
servers.net.
.            249366    IN    NS    b.root-
servers.net.
.            249366    IN    NS    g.root-
servers.net.
.            249366    IN    NS    k.root-
servers.net.
.            249366    IN    NS    c.root-
servers.net.
.            249366    IN    NS    m.root-
servers.net.
.            249366    IN    NS    h.root-
servers.net.
;; Received 271 bytes from
192.168.130.202#53(192.168.130.202) in 70 ms

es.            172800    IN    NS    sns-
pb.isc.org.
```

```
es.             172800    IN    NS    ns3.nic.fr.
es.             172800    IN    NS    f.nic.es.
es.             172800    IN    NS    ns-
ext.nic.cl.
es.             172800    IN    NS    g.nic.es.
es.             172800    IN    NS    a.nic.es.
es.             172800    IN    NS
ns1.cesca.es.
es.             86400    IN    DS    44290 8 1
7711F564D55B41C8CE7DFAF4DD323C5B271F86CD
es.             86400    IN    DS    44290 8 2
562EF35E7065588A7178A4BD0155C8527F029C82AA455DD359C84908
B2A7FE17
es.             86400    IN    RRSIG    DS 8 1
86400 20170619050000 20170606040000 14796 .
Z2dBpJNwru3b15TSXSDBDDvK9oSUW11YEWxBHpY6CMVlWNs66gICRhMv
uFkMqvje0wL+7t7v0lJPhGaLx3L1hybn/3tPhkPGCUClzrnTpOiXGULi
ydvUcB8xCt1FxvxMUJ4NiIxvpJs51xYMIxTLoihJ8s2wXhm+tL8joQ3l
WE42jfBIXfVw6PKCfoHlxgiQ/ZTbUKBSUzTdSMKzhlL1zGFclVXNdNKv
jZgNMRwN7G5cn/4Dv1IAnQMXJ12S/Cr4sIowe5yE
/u7XNB4hHIiDSGeQ
nxAjRpZOFZ0tzCGWnwc8mGMsQoKp2i3My1s5S7poX+Ut/GvOGgVjgOHi
WILoEg==
;; Received 844 bytes from 192.5.5.241#53(f.root-
servers.net) in 71 ms

google.es.        86400    IN    NS
ns2.google.com.
google.es.        86400    IN    NS
ns1.google.com.
vhgig769k8fqo5v5ig9q7a6un45t1hhj.es. 86400 IN
NSEC3 1 1 5 7F6C8C66C2BC205D
VHOEO0370MQNG4CA8NT2VHR8REPI91SU NS SOA RRSIG
```

```
DNSKEY NSEC3PARAM
vhgig769k8fqo5v5ig9q7a6un45t1hhj.es. 86400 IN
RRSIG NSEC3 8 2 86400 20170611025013
20170528075535 61810 es.
kSDEA9hbIoynY757uNfRzHVfdoaFeiGRu5u1Ph79Y1RaqOmnfum5SBc1
HRE+nztM7B4A7jfCa5kPNtjQeLwYR0N8L6hivDveHrqg6d/SQLLGZTHC
nmMgt1PbM0siEStFcntKK9n5JaCSI0tabiwXDKWK1S1WA50WGQvFvPxE
4GQ=
bat3rrbtibfqoh1t1ottbvdk2q30b3am.es. 86400 IN
NSEC3 1 1 5 7F6C8C66C2BC205D
BBAI9G16OK3893I7B3QMCL4TRS6A9ID4 NS DS RRSIG
bat3rrbtibfqoh1t1ottbvdk2q30b3am.es. 86400 IN
RRSIG NSEC3 8 2 86400 20170611152235
20170528195912 61810 es.
DcnxJPHRI7Ztgwbnorn58BWbGhxLd5vpAgut4onDZpZIP/YbmwvJM7/F
HydAkFyGL1WVIlebVVyEMRbdsLxOL0+Ly0JWVnsBk1YLQwckyd3Gg0b\
dQh4K2pq2RpBgfKmkGFCQPAcsJuN4nSWLbDFJb+iAHlIRKCkMUNygFau
l2o=
;; Received 583 bytes from 192.5.4.1#53(sns-
pb.isc.org) in 53 ms

google.es.        300    IN    A
216.58.210.163
;; Received 43 bytes from
216.239.32.10#53(ns1.google.com) in 33 ms
```

Todas estas opciones y argumentos del comando dig nos serán muy útiles para ejecutar diferentes tipos de consultas DNS.

- Otros ejemplos del uso del comando dig
- Cómo usar el comando dig

eltallerdelbit.com

Cómo Configurar un Servidor Maestro paso a paso.

5-6 minutos

El servidor maestro DNS es aquel que tiene una copia del archivo de configuración de zona.

En este caso empezaremos por lo más simple y crearemos un **Servidor Maestro** paso a paso.

Si hay aspectos del DNS que necesitais reforzar , no dudéis en echarle un vistazo a este Resúmen del DNS. Conceptos y Definiciones básicas.

DEFINIR LA ZONA DEL SERVIDOR MAESTRO

Lo primero que debemos hacer es definir la zona maestra llamada “pruebas.ElTallerDelBit.com” y la zona inversa de la zona maestra (llamada “1.168.192.in-addr.arpa”) , en el archivo named.conf.local.

```
zone "pruebas.ElTallerDelBit.com" {
type master;
file "/etc/bind/db.ElTallerDelBit.com";
};
```

```
zone "1.168.192.in-addr.arpa" {
type master;
file "/etc/bind/db.1.168.192";
};
```

Como véis, hemos declarado la zona,

– primero la directa (pruebas.ElTallerDelBit.com)

– y luego la inversa (1.168.192.in-addr.arpa)
– luego hemos declarado el tipo de servidor que será en este caso “master”,

– y hemos introducido la ruta dónde se encuentra el fichero en el que se define esa zona.

En algunas distribuciones de Linux es posible que no necesitéis poner la ruta absoluta sino la relativa. Si teneis problemas con el Bind podeis probarlo.

Creamos el archivo “/etc/bind/db.ElTallerDelBit.com” (el nombre de archivo puede ser cualquiera, pero que quede bien reflejado en el named.conf.local) ,

con el que crearemos y definiremos la zona “pruebas.ElTallerDelbit.com” (la zona primaria maestra, con nombre de dominio “pruebas.ElTallerDelbit.com”).

Nuestro servidor será maestro de esa zona:

```
GNU nano 2.2.2         Archivo:
db.ElTallerDelBit.com

;

; BIND data file for local loopback
interface

;

$TTL    604800

@       IN      SOA
pruebas.ElTallerDelBit.com.
```

```
root.pruebas.ElTallerDelBit.com. (

2012020101      ; Serial

604800          ; Refresh

86400           ; Retry

2419200         ; Expire

604800 )        ; Negative Cache TTL

;

@       IN      NS
pruebas.ElTallerDelBit.com.

@       IN      A       192.168.1.254
```

Nuestro Servidor responderá con la dirección local 192.168.1.254.

ARCHIVO DE ZONA INVERSA

Lo mismo haremos con la zona inversa.
Crearemos un archivo para la zona inversa, que llamaremos db.1.168.192 , en el que definiremos lo siguiente:

```
; BIND data file for local loopback
interface
```

```
$TTL    604800

@       IN      SOA
pruebas.ElTallerDelBit.com.
root.pruebas.ElTallerDelBit.com.(

4         ; Serial

604800    ; Refresh

86400     ; Retry

2419200   ; Expire

604800 )  ; Negative Cache TTL

;

@       IN      NS
pruebas.ElTallerDelBit.com.

254     IN      PTR
pruebas.ElTallerDelBit.com.
```

– El símbolo @ se refiere al propio servidor,

– ” NS ” es el nombre del servidor DNS, que se traduce con el registro siguiente tipo “A”, que lo asocia con una IP que

muestra 254. Se refiere a la ip 192.168.1.254.

– ” SOA ” se refiere al Registro de “Start of Authority”, es decir la autoridad para un dominio, y almacena varios parametros de configuración de la zona de la que es autoritativo nuestro servidor.

– ” PTR ” se usa para asociar nombres del dominio inverso (in-addr.arpa) con sus nombres. Se usará para obtener una traducción a partir de una IP.

Antes de nada debemos reiniciar el Bind con el comando

```
/etc/init.d/bind9 restart
```

y si tenemos errores , podemos utilizar los comandos :

named-checkconf: Para revisar la sintáxis del **named.conf.local**

named-checkzone: Para revisar la sintáxis de los archivos de zona.

Una vez que hayamos reiniciado el bind y no tengamos errores, nos dispondremos a comprobar qe hemos configurado bien nuestro dominio dns y que responde correctamente.

RESULTADO RESOLUCIÓN DIRECTA (dig *nombre_dominio*)

RESULTADO RESOLUCIÓN INVERSA:

```
RESULTADO RESOLUCION INVERSA:
; <<>> DiG 9.7.0-P1 <<>> ; -x 192.168.1.254
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status:
NOERROR, id: 414
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1,
AUTHORITY: 1, ADDITIONAL: 1
;; QUESTION SECTION:
;254.1.168.192.in-addr.arpa.    IN      PTR

;; ANSWER SECTION:
254.1.168.192.in-addr.arpa. 604800 IN
```

```
PTR     pruebas.ElTallerDelBit.com.

;; AUTHORITY SECTION:
1.168.192.in-addr.arpa. 604800  IN
NS      pruebas.ElTallerDelBit.com.

;; ADDITIONAL SECTION:
pruebas.ElTallerDelBit.com. 604800 IN
A       192.168.1.254

;; Query time: 0 msec
;; SERVER: 192.168.1.254#53(192.168.1.254)
;; WHEN: Sun Feb  5 13:51:28 2012
;; MSG SIZE  rcvd: 114
```

Si tienes más dudas, puedes adquirir el ebook "Aprende Apache y DNS, con ejercicios resueltos 100%", con el que podrás realizar varias prácticas DNS y Apache explicadas y resueltas paso a paso

Si deseas aprender por ti mismo puedes echarle un vistazo a la página web oficial de Bind

eltallerdelbit.com

Servidor DNS Caché

6-7 minutos

Un servidor DNS Caché no tiene autoridad sobre ninguna Zona. Se dedica a reenviar la consulta y luego memorizar la respuesta para no tener que volver a preguntarla.

Lo primero que hemos de hacer es instalar el servicio Bind (en LINUX). Existen otros servidores de nombres, pero el bind es el más famoso.

Lo instalaremos así:

```
sudo apt - get install bind9
```

Como ya sabemos, el fichero de configuración de ***Bind DNS*** es el */etc/bind/named.conf.local*.

En este archivo es donde configuraremos las opciones principales del servidor DNS, como las zonas (caché, primaria, secundaria o de reenvío) .

En este caso, como hemos dicho, vamos a configurar un servidor DNS caché con Bind.

Un **servidor DNS caché** es una zona, que indica que actúa como caché. Cuando recibe una **consulta DNS** , reenvía la consulta al servidor con autoridad para responder sobre la zona solicitada, y guarda la información recibida acerca de las **consultas DNS** para utilizarla en posteriores consultas.

Es fundamental que para reenviar las consultas DNS, en el archivo */etc/bind/named.conf.options*, debe constar la IP del servidor DNS al que se reenviarán todas las consultas (***forwarder***) que posteriormente el **Server DNS Caché** guardará:

```
forwarders {
192.168.0.1;
};
```

En este punto hemos de recordar que **el servidor DNS al que se van a reenviar las consultas, ha de permitir las consultas del servidor DNS caché** (igual ocurre cuando configuramos un servidor esclavo o secundario).

Es importante recordar que debemos configurar un ***nameserver*** *(o servidor de nombres)* en el archivo */etc/**resolv.conf***, que es el archivo dónde configuraremos los Servidores DNS que nos proporcionarán la información de las consultas.

Así que el servidor que actuará de *forwarder* debe contener en su *named.conf.options* la línea:

```
allow-query { 192.168.0.0/24; };
```

En este caso incluimos la red 192.168.0.0/24 que casualmente es la red en la que se encuentran los dos servidores, pero hemos de poner aquí la red donde se encuentra el **servidor DNS Caché**.

Como en este caso nuestro Servidor DNS caché va a ser el servidor DNS que responderá las consultas (reenviando al *forwarder* las consultas que no conoce), el nameserver que aparecerá en el archivo /etc/resolv.conf, será el propio servidor, con su interfaz loopback:

```
nameserver 127.0.0.1
```

Pero también hemos de configurar la **zona caché**, así que entramos al archivo */etc/bind/named.conf.local*

```
nano named.conf.local
```

Y hemos de añadir esto en el archivo:

```
zone "." {

type hint;

file "/etc/bind/db.root";
};
```

Y ahora mostraré un ejemplo del archivo de dicha zona "." que se menciona.

Este archivo (***db.root***) contiene la información en los servidores de nombres raíz necesarios para inicializar la **caché de los servidores** de nombres de dominio.

Aquí tenemos un ejemplo:

```
;       This file is made available by
InterNIC

;       under anonymous FTP as

;           file                /domain
/named.root

;           on server
FTP.INTERNIC.NET

;       -OR-
RS.INTERNIC.NET
```

```
;

;       last update:    Dec 12, 2008

;       related version of root zone:
2008121200

; formerly NS.INTERNIC.NET

;

.                       3600000  IN  NS
A.ROOT-SERVERS.NET.

A.ROOT-SERVERS.NET.     3600000      A
198.41.0.4

A.ROOT-SERVERS.NET.     3600000      AAAA
2001:503:BA3E::2:30

;

; FORMERLY NS1.ISI.EDU

;

.                       3600000      NS
B.ROOT-SERVERS.NET.
```

```
B.ROOT-SERVERS.NET.      3600000      A
192.228.79.201

;

; FORMERLY C.PSI.NET

;

.                        3600000      NS
C.ROOT-SERVERS.NET.

C.ROOT-SERVERS.NET.      3600000      A
192.33.4.12

;

; FORMERLY TERP.UMD.EDU

;

.                        3600000      NS
D.ROOT-SERVERS.NET.

D.ROOT-SERVERS.NET.      3600000      A
128.8.10.90

;
```

```
; FORMERLY NS.NASA.GOV

;

.                        3600000      NS
E.ROOT-SERVERS.NET.

E.ROOT-SERVERS.NET.      3600000      A
192.203.230.10

;

; FORMERLY NS.ISC.ORG

;

.                        3600000      NS
F.ROOT-SERVERS.NET.

F.ROOT-SERVERS.NET.      3600000      A
192.5.5.241

F.ROOT-SERVERS.NET.      3600000      AAAA
2001:500:2F::F

;

; FORMERLY NS.NIC.DDN.MIL
```

```
;

.                        3600000      NS
G.ROOT-SERVERS.NET.

G.ROOT-SERVERS.NET.      3600000      A
192.112.36.4

; FORMERLY AOS.ARL.ARMY.MIL

;

.                        3600000      NS
H.ROOT-SERVERS.NET.

H.ROOT-SERVERS.NET.      3600000      A
128.63.2.53

H.ROOT-SERVERS.NET.      3600000      AAAA
2001:500:1::803F:235

;

; FORMERLY NIC.NORDU.NET

;

.                        3600000      NS
I.ROOT-SERVERS.NET.
```

```
I.ROOT-SERVERS.NET.      3600000      A
192.36.148.17

;

; OPERATED BY VERISIGN, INC.
```

COMPROBAR EL FUNCIONAMIENTO DEL SERVIDOR CACHÉ

Para comprobar nuestro **servidor caché**, utilizaremos el propio equipo como cliente; Nosotros mismos seremos nuestro servidor DNS.

Es decir, como decíamos antes, en el archivo */etc/resolv.conf*, ha de constar la línea:

```
nameserver 127.0.0.1
```

De forma que nuestra interfaz loopback resolverá las peticiones DNS.

Hecho esto , ya podemos utilizar la herramienta dig para resolver los nombres de Dominio .
Así que ejecutamos un dig a google y observamos que el tiempo de respuesta es de 366 milisegundos.

En este momento, tras mostrar los resultados, la caché del servidor DNS Caché ha comenzado a funcionar y ha guardado la consulta que acabamos de hacer, por si se necesita de nuevo:

Así que vamos a ver si es verdad; hacemos una nueva petición de resolución de nombre para google.es:

```
dig google.es
```

Y comprobamos que el tiempo de respuesta ha disminuido hasta 0 milisegundos .

Cómo podéis ver, es bastante recomendable **instalar un servidor cachéDNS** en en nuestra red siempre que nos sea posible.

Los que utilicéis Linux lo tenéis facil, simplemente tenéis que instalar el Bind y seguir las indicaciones de este artículo.

Los que utiliceis Windows, podéis descargar e instalar una de estas dos aplicaciones :

Acrylic DNS

Dns Speeder

Espero que os haya gustado.
Sigue El Taller del Bit y suscríbete para recibir los artículos directamente.

eltallerdelbit.com

Servicio DNS | Conceptos y Comandos

6-7 minutos

Los **servidores DNS** contienen información sobre nombres en la red, y también sobre otros resolvedores de nombres, de tal forma que esa información la van pasando a otros resolvedores que le puedan consultar.

El sistema DNS adopta una forma de base de datos distribuida y jerárquica, que almacena info acerca de computadoras que forman parte de una red.

(eso significa que está dividido en jerarquías o estamentos de diferente y gradual importancia, como una pirámide) y además utiliza el conocido modelo cliente-servidor. Así , por su jerarquía, sabemos que hay varios niveles de Dominios.

En la siguiente imágen podemos ver un ejemplo de la jerarquía del **DNS**.

El ICANN es el responsable de los dominios superiores y del dominio raíz .

El ICANN posee 13 servidores de nombres distribuidos por el mundo, son los TNS (Top Name Servers).

Podemos consultarlos en http://root-servers.org/.

Todos ellos tienen la misma info replicada: las zonas con los nombres de dominio de segundo nivel de todo el mundo.

En este punto surge otro término importante:

***FQDN*, Fully Qualified Domain Name – Nombre de dominio completamente cualificado.**

El FQDN es formado por el nombre de la computadora y el nombre de dominio asociado.
Por ej, si tenemos un servidor que se llama “Servidor 1” , y un nombre de dominio como “ejemplo7.com”, el FQDN será “Servidor1.ejemplo7.com”

¿Cómo se Resuelve un nombre ?

Cuando intenta resolver un nombre, el cliente DNS dirige una petición al **servidor DNS** que tenga configurado. Si éste conoce la información , generará una respuesta; sino pasará la petición a otro servidor DNS configurado para el dominio superior.

Las peticiones sobre nombres de dominio pueden ser Recursivas o Iterativas.

– Consulta recursiva:

El servidor recibe una consulta y la retransmite, y así determina la info buscada y luego la devuelve al cliente.

– Consulta iterativa:

El servidor de nombres devuelve la información de la que dispone , y mostrará además de una lista de servidores adicionales con los que el cliente puede contactar para completar su consulta.

Resolución Inversa:

La resolución Inversa es el proceso de averiguar el nombre de dominio asociado a una dirección IP (en lugar de la resolución normal que averigua la ip asociada a un dominio).

Para efectuar esta resolución se dispone del dominio especial : “in-addr.arpa”, también llamado el dominio inverso.

Dentro de este dominio se deben especificar las cifras de la IP al revés, empezando por el final ,y añadiendo “in-addr.arpa” al final del dominio.

Por ej, para la IP “181.73.45.20”, tendríamos “20.45.73.181.in-addr.arpa”

La resolución inversa es muy útil para ahorrar tiempo, por ej, si tuvieramos que resolver esta ip sin la resolución inversa, debería buscarse la IP en todas las zonas de todos los dominios de la red(y eso sería muy costoso).

Las Zonas

Una Zona es la parte de un dominio que es gestionada por una organización. Una zona y su dominio asociado no tienen por qué coincidir.

Tipos de Servidores DNS

Primario

Se encarga de cargar la info de una zona, y tiene autoridad sobre ella.

Secundario

Un servidor secundario tiene autoridad sobre una zona , pero obtiene la información de esa zona de un servidor primario utilizando un proceso llamado transferencia de zona.

Para mantenerse sincronizados, los servidores de nombres secundarios consultan a los primarios cada cierto tiempo, y vuelven a ejecutar la transferencia de zona si el primario ha sido actualizado

Un servidor primario o secundario realiza todas las funciones de un servidor caché.

Caché

No tienen autoridad sobre ninguna zona.
Cuando se le hace una consulta, la reenvía a los servidores que saben su respuesta.
Después almacena la respuesta en memoria y ya no tiene

que preguntarla.

Reenvío

No poseen autoridad sobre las zonas que resuelven.

Responden las peticiones reenvíandolas a los servidores que tienen configurados ,y esperan su respuesta.

Comandos para resolución DNS

dig

Es una herramienta de resolución de nombres muy útil en entornos Linux.

por ej:

dig google.es

En este caso estamos realizando una consulta resursiva del dominio google.es, por lo que el servidor reenviará la consulta, y luego recoge la información y la mostrará.

Aquí tenéis un ejemplo:

Se pueden apreciar las direcciones de los servidores a los que se ha recurrido para consultar la información deseada acerca del nombre de dominio “google.es” .

Y al final se aprecia también la dirección del servidor DNS que responde con la información conseguida después del reenvío de la consulta (192.168.1.94).

dig -x 192.168.1.220

Con esto lo que conseguimos es la resolución inversa. Es decir, resolveremos el nombre de dominio a partir de la IP.

Recordamos que las consultas son por defecto recursivas (el servidor pregunta al DNS y devuelve la información deseada). Pero con este comando :

-dig google.es +trace
Podemos observar un ejemplo de consulta iterativa, es decir, nos devuelve la información disponible sobre nuestra consulta y una lista de servidores que pueden completar la

información deseada.

nslookup

También permite realizar consultas sobre DNS.

Tiene 2 modos: Modo interactivo y modo no interactivo.

El modo no interactivose ejecuta de esta forma: nslookup nombre_de_dominio.com

por ej: nslookup google.es
nslookup eltallerdelbit.com

El resultado de una consulta será algo como :

El modo interactivo consiste en ejecutar :

nslookup

y a continuación el cursor cambiará y podremos introducir nombres de dominio e ip´s una tras otra.

Espero que os haya sido útil este resúmen sobre el

servicio DNS y algunas herramientas de resolución de nombres de dominio.

tiendalinux.com

Berkeley Internet Name Domain (BIND)

7-8 minutos

Actualmente, Internet y todas las redes locales dependen de un *Servicio de nombres de dominio(Domain Name Service, DNS)* eficaz y fiable que se usa para asociar los nombres de sistemas a las direcciones IP y viceversa.

Para poder obtener un DNS en la red, se necesita un *servidor de nombres* el cual traduce las direcciones IP necesarias para sus resoectvas conexiones. Además, un servidor de nombres puede efectuar la traducción en el nombre del sistema, lo que se llama a menudo *reverse lookup*, o resolución inversa.

Este capítulo describe BIND, la estructura de sus ficheros de configuración y la manera en la que se puede administrar localmente o a distancia.

Para mayor información sobre la configuración de BIND usando la herramienta **BIND Configuration Tool** con la interfaz Gŕfica, consulte el *Manual oficial de personalización de Red Hat Linux*. Observe que si usa la herramienta **BIND Configuration Tool**, no es necesario modificar manualmente los ficheros de configuración de BIND ya que

la herramienta los anula.

Los sistemas que usan las redes IP tienen que conocer la dirección IP de una máquina para poder conectarse. La mayor parte de los usuario prefieren usar nombres de máquinas como el nombre de un host o de un *fully qualified domain name (FQDN)*, para especificar el sistema en el momento de la conexión. Además, muchos programas de usan nombres de dominio en sus ficheros de configuración cuando hacen referncia a un sistema distante con el fin de permitir el cambio de las direcciones IP sin tener que modificar el nombre del sistema, entre otras razones. El servicio que lo facilita se llama DNS y normalmente lo arrancan servidores centralizados que tienen la autorización de ciertos dominios y que se refieren a otros servidores DNS para obtener información que ya poseen.

El DNS funciona gracias a los deminios de servidores de nombres que efectuan una traducción del nombre IP. Un aplicación cliente requiere información del servidor de nombres conectándose normalmente al puerto 53. El servidor de nombres intenta solucionar el FQDN con la librerín de soluciones que puede contener información y que tienen la autrización sobre el host que se ha pedido o sobre los datos en caché del nobre que se ha pedido anteriormente. Si el servidor de nombres no encuentra la solución en su librería, se dirige a otros servidores que se llaman *root nameservers*, o servidores de nombres raíz, para el FQDN en cuestión. Con esta información, realiza una búqueda en los servidores que tienen la autorización

para determinar el nombre de la dirección IP. Si efectua la operación en sentido contrario (reverse lookup) el procedimiento es el mismo salvo que en este caso se desconoce la dirección IP en vez de un nombre.

Zonas

En Internet, el FQDN de un host se puede analizar en diversas secciones y estas secciones se analizan a su vez por orden jerárquico, como en un árbol el tronco, las ramas primarias, las ramas secundarias, etc. Por ejemplo:

Figura 17-1. Ejemplo de FQDN (fully qualified domain name)

Cuando miramos un FQDN para encontrar la dirección IP de un determinado sistema, hay que leer el nombre de derecha a izquierda; los niveles jerárquicos están separados por un punto (`.`). En nuestro ejemplo, `com` define el *dominio superior* para este FQDN. El nombre del `domain` es un subdominio de `com`, con `sales` como subdominio de `domain`. El nombre que se encuentra más a la derecha es un FQDN de un host que identifica una determinada máquina.

Aparte del nombre del dominio, cada sección se llama *zona*, la cual define un espacio de nombre particular (namespace). Un *namespace*, o espacio de nombre, controla los nombres de los subdominios de la izquierda. Aunque en el ejemplo solamente hay dos subdominios, un FQDN tiene que contener al menos un subdominio pero puede incluir muchos más; depende de la organización del

espacio de nombres elegido.

Las zonas las defienen servidores de nombres que hacen de autoridad con la utilización de *ficheros de zona*, que describen el espacio de nombres de esa zona, los servidores de correo que un determinado subdomnio tiene que usar. Los ficheros de zona se almacenan en los *servidores de nombres primarios*) ,que son los que tienen la autorida y donde se realizan los cambios de los ficheros, y en los *servidores de nombres esclavos* (que se llaman también *servidores de nombres secundarios*), que reciben los ficheros de zona de los servidores de nombres maestros o primarios. Todos los servidores de nombres puden ser maestros o esclavos para cada una de las diferentes zonas al mismo tiempo. Todo depende de la configuración de cada servidor de nombres.

Tipos de servidores de nombres

Existen cuatro tipos de servidores de nombres:

- *Maestros* — Almacena los registros de las zonas originales y tienen la autoridad de un cierto espacio de nombres donde buscan respuestas concernientes a dicho espacio de nombres.
- *Esclavo* — Responde también a las peticiones que provienen de otros servidores de nombres y que se refieren a los espacios de nombres sobre los que tiene autoridad. Los servidores esclavos obtienen la información de espacios de nombres de servidores de nombres maestros a través de una *zona de transfeencia*, en la que el esclavo

manda al servidor maestro un petición que se llama `NOTIFY` para una determinada zona y el maestro responde csi el esclavo está autorizado para recibir la tranferencia.

- *Caching-only* — Ofrece servicios de resolución de nombres a direcciones IP pero no tiene ninguna autoridad sobre ninguna zona. Las respuestas en general se pone en un caché que se encuentra en la base de datos almacenada en la memoria durante un periodo fijo, la cual está especificada por la zona importada y así obtener una resolución más rápida para otros clientes DNS después de la primera resolución.
- *Forwarding* — Hace que determinados servidores de nombres llevan a cabo la resolución. Si alguno de estos servidores no puede efectuar la resolución, el preceso se para y la resolución se anula.

  Un servidor de nombres puede ser de varios tipos. Por ejemplo, puede ser servidor de nombres maestro para determinadas zonas, esclavo para otras o incluso ofrecer solamente la transmisión de una resolución.

BIND como servidor de nombres

Red Hat Linux incluye BIND, que es un servidor de nombres open source potente y muy famoso. BIND utiliza el demonio `named` para los servicios de resolución de nombres. Toda la información sobre la configuración se almacena en el fichero `/etc/named.conf` y los ficheros de zona se encuentran en `/var/named`. Para mayor información sobre estos ficheros, consulte la [la sección de

nombre *Ficheros de configuración BIND*.

La versión 9 de BIND incluye una utilidad que se llama `rndc` y que permite la administración del demonio `named`. Para mayor información, sobre el comando consulte la la sección de nombre *Utilización del comando `rndc`*.

tiendalinux.com

Ficheros de configuración BIND

27-34 minutos

El demonio named del servidor de nombres BIND utiliza el fochero /etc/named.conf para la configuración. Todos los ficheros de zona se encuentran en /var/named.

⚠Advertencia

No modificque manualmente el fichero /etc/named.conf o cualquier otro fichero del /var/named si está usando al herramienta **BIND Configuration Tool**. Todos los cambios que se realicen manualmente en este fichero o en cualquier otros fichero de éste, se borrarán cuando se use la próxima vez la herramienta **BIND Configuration Tool**.

El fichero /etc/named.conf no puede tener errres para podr arrancar el demonio named. Aunque existen algunas opciones erróneas en algunas declaraciones que no son tan importantes como para bloquear el servidor, todos los errores que se encuentren en las declaraciones impedirán que arranque el demonio named.

/etc/named.conf

El fichero /etc/named.conf es un conjunto de

declaraciones que usa las opciones que se encuentran en los corchetes { }. He aquí cómo está organizado un fichero /etc/named.conf Figura 17-2.

```
<statement-1> ["<statement-1-name>"]
[<statement-1-class>] {
   <option-1>;
   <option-2>;
   <option-N>;
};

<statement-2> ["<statement-2-name>"]
[<statement-2-class>] {
   <option-1>;
   <option-2>;
   <option-N>;
};

<statement-N> ["<statement-N-name>"]
[<statement-N-class>] {
   <option-1>;
   <option-2>;
   <option-N>;
};
```

Figura 17-2. Organización común de /etc/named.conf

La opción "*<statement-name>*" es necesaria solamente con las declaraciones acl, include, server, view y zone. La opción *<statement-N-class>* aparece sólo con la declaración zone.

Se pueden ecribir comentarios en el fichero `/etc/named` usando los caracteres de C `/* */` o después de `//` y `#`.

Se pueden usar las siguientes declaraciones en el fichero `/etc/named.conf`:

- `acl <acl-name>` — Configura una lista de contrl del acceso de las direcciones IP a las que se autorizarán o se denegarán ciertos serviciosnamed. En general, las direcciones IP individuales o la nota de IP (como `10.0.1.0/24` sirve para identificar las direcciones IP correctas.

  Ya se han definido algunas listas de control de acceso, por lo tanto no tiene que configruar una declaración `acl` para definirlas:

- `any` — Corresponde a todas las direcciones IP.
- `localhost` — Corresponde a todas las direcciones IP utilizadas para un sistema local.
- `localnets` — Corresponde a todas las direccines IP de una red a la que se conecta el sistema local con los dispositivos.
- `none` — No corresponde a ninguna dirección IP.

  Cuando de utilizan con otras declaraciones `/etc/named.conf` y sus opciones, las declaraciones `acl` pueden ser muy útiles para usar el servidor de nombres BIND. Observe el ejemplo siguiente en Figura 17-3:

```
acl black-hats {
   10.0.2.0/24;
```

```
    192.168.0.0/24;
};

acl red-hats {
    10.0.1.0/24;
};

options {
    blackhole { black-hats; };
    allow-query { red-hats; };
    allow-recursion { red-hats; };
}
```

Figura 17-3. Ejemplo de la utilización de la declaración `acl`

`named.conf` contiene dos listas de control de acceso (`black-hats` y `red-hats`.

- `controls` — Configura diversas restricciones de seguridad necesarias para la utilización del comando `rndc` y así poder usar el demonio `named`.

  Consulte la la sección de nombre */etc/named.conf* para mayor información sobre la declaración `controls`.

- `include "<file-name>"` — Incluye el fichero especificado en el fichero de configuración que se está usando y que permite así situar los datos de configuración sensibles (como `keys`) en un fichero separado con los permiso que impiden a los usuarios sin privilegios leerlos.

- `key "<key-name>"` — Define una clave particular. Estas

claves sirven para autenticar dversas acciones como la actualización de la seguridad o la utilización del comando `rndc`. Se usan dos opciones con `key`:

- `algorithm <algorithm-name>` — El tipo de algoritmo utilizado, como `dsa` o `hmac-md5`.
- `secret "<key-value>"` — La clave encriptada.

  Consulte Figura 17-22 el ejemplo de la declaración n `key`.

- `logging` — Permite usar varios tipos de logs que se llaman *channels o canales*. Usando la opción `channel` en la declaración `logging`, se puede construr un tipo de log personalizado con el nombre del fichero (`file`), con el tamaño(`size`), la versión (`version`) y el nivel de importancia (`severity`. Una vez que se ha definido el canal personalizado, se usa la opción `category` para calificar el canal y comenzar la conexión mientras que se arranca `named`.

  Por defecto, `named` envía mensajes de log a los estándares del demonio `syslog`, que les sitúa en `/var/log/messages` por defecto. Esto se debe a que varios canales estándares se encuentran en BIND junto con varios niveles de importancia como el que trata los mensajes de registros informativos (`default_syslog`) y otro que trata específicamente la corrección de errores (`default_debug`). La categoría predeterminada `default`, utiliza los canales de BIND para llevar a cabo la conexión normal sin ninguna configuración especial.

  La personalización del proceso de conexión es un proceso

que requiere una explicación muy detallada y no es el objetivo de este capítulo. Para mayor información, consulte el *Manual de referencia del administrador de BIND 9*.

- `options` — asigna valores a muchas opciones entrelazadas, incluidos los comandos que se usan para situar un fichero de funcionamiento de `named`, el nombre de ficheros y otros.

  Las siguientes opciones son las más utilizadas:

- `allow-query` — Especifica los hosts que se usarán para establecer las peticiones en el servidor de nombres. Por defecto, todos los hosts están autorizaodos a presentar peticiones. Se puede usar una lista de control de acceso o una colección de direcciones IP para no autorizar a un determiando número de servidores de nombres.
- `allow-recursion` — Parecida a la opción `allow-query`, salvo que se aplica a las peticiones recursivas. Por defecto, todos los hosts están autorizados a presentar peticiones en eu servidor de nombres.
- `directory` — Reemplaza el fichero de funcionamiento de `named` en vez del fichero predeterminado `/var/named`.
- `forward` — Controla cómo se lleva a cabo el forwarding, si la opción `forwarders` contiene direcciones IP válidas que designen dónde enviar las peticiones.

  Si se usa la opción `first`, los servidores de nombres esecificados en la opción `forwarders` son los primeros en recibir las peticiones y si no pueden solucionarlas el demonio `named` intenta la resolución.

Si se usa la opción `only`, `named` no intentará la resolución si los servidores de nombres han fallado.

- `forwarders` — Especifica una lista de servidores de nombres a los que hay que mandar las peticiones para obtener la resolución.
- `listen-on` — Especifica el dispositivo de red que `named` va a utilizar para recibir las peticiones. Por defectom se usan todos los dispositivos.

  Esta opción es útil si dispone de más de un dispositivo de red y desea limitar los sistemas que puedan efectuar las peticiones gateway y servidor de nombres y desea bloquear todas las peticiones excepto las que provienen de su red privada, la opción en su servidor de nombres. Por ejemplo, si tiene una máquina sirve como gateway y como servidor de nombres la opción `listen-on` se parece a la Figura 17-4.

  ```
  options {
    listen-on { 10.0.1.1; };
  };
  ```

  Figura 17-4. Ejemplo de la opción `listen-on`

  De esta manera, solamente se aceptan las peticiones que provienen del dispositivo de red de la red privada (`10.0.1.1`).

- `notify` — Determina si `named` envía notificaciones a los servidores esclavos cuando una zona se actualiza. Por defecto, se usa la opción `yes`, pero se puede usar también la opción `no`, para evitar que se manden notificaciones a

los servidores esclavo y así solamente mandar las notificaciones a los servidores de la lista `also-notify`.

- `pid-file` — Permite especificar la localización del fichero del proceso ID creado por `named` cuando arranca.
- `statistics-file` — Permite especificar la localización del fichero de estadística que se ha creado. Por defecto, las estadísticas de `named` se encuentran en `/var/named/named.stats`.

  Existen numerosas opciones disponibles, muchas de ellas dependen unas de otras para poder funcionar correctamente. Consulte el *Manual de referencia del administrador BIND 9* para mayor información.
- `server` — Define opciones particulares que afectan a la manera en la que `named` reacciona ante los servidores de nombres distantes y particularmente conciernen a las notificaciones y las transferencias de zona.

  La opción `transfer-format` determina si se ha enviado el record de los recursos con cada mensaje (`one-answer`) o la grabación de recursos múltiples con cada mensaje (`many-answers`). Aunque la opción `many-answers` es más eficaz, solamente los últimos servidores de nombres BIND la entienden.
- `trusted-keys` — Contiene las calves públicas que usa DNSSEC. Para mayor información sobre la seguridad de BIND, consulte la la sección de nombre *Seguridad*.
- `view "<view-name>"` — Visualizaciones especiales que responden a un tipo de información particular dependiendo

del host que contacta el servidor de nombres. Esto permite a determinados hosts recibi una respuesta que se refiere a una zona particular mientras que otros hosts reciben información completamente diferente. Alternativamente, ciertos hosts pueden estar autorizados para acceder a determinadas zonas mientra que otros menos autorizados continuan a efectuar peticiones a otros zonas.

Se pueden usar múltiples visualizaciones, las cuales tienen un solo nombre. La opción `match-clients` especifica las direcciones IP que se aplican a una visualización determinada. Se pueden usar todas las declaraciones `option` en un a visualización, aunque tienen prioridad las opciones globales ya configuradas para `named`. La mayor parte de las declaraciones `view` contienen múltiples declaraciones `zone` que se aplican a la lista `match-clients`. El orden de aparición en la lista de las declaraciones `view` es importante porque la primera declaración `view` corresponde a la dirección IP de un cliente particular.

Consulte la la sección de nombre *Visualizaciones múltiples* para mayor información sobre la declaración `view`.

- `zone "<zone-name>"` — Especifica zonas particulares para las que está autorizado es servidor de nombres. La declaración `zone` se usa sobre todo para especificar el fichero que contiene la configuración de la zona y transmite ciertas opciones de esa zona a `named` que tendrán prioridad sobre todas las otras declaraciones `option` del fichero `/etc/named.conf`.

El nombre de la zona es importante porque representa el valor por defecto asignado a la directiva `$ORIGIN` que se usa en el fichero de zona y que está relacionado con los el no-FQDN. Por ejemplo, si estas declaración `zone` define el espacio de nombre para `domain.com`, hay que usar `domain.com` y no *<zone-name>* para que se situe al final de los nombres de los hosts que se usan en ese fichero de zona.

Las opciones más usadas de la declaración `zone` son las siguientes:

- `allow-query` — Especifica los clientes que se autorizan para pedir información sobre una zona. Por defecto todas las peticiones de información sin autorizadas.
- `allow-transfer` — Especifica los servidores esclavos que están autorizados para pedir una transferencia de información de la zona. Por defecto, todas las peticiones se autorizan.
- `allow-update` — Especifica los hosts que están autorizados para actualizar dinámicamente la informción de la zona. Por defecto, no se autoriza la actualización de la información.

Advertencia

Tenga cuidado cuando autorice a los hosts para actualizar la información de la zona. No habilite esta opción si no tiene confianza en el host que vaya a usar. Es mejor que el administrador actualice manualmente los records de zona y que

recargue el servicio named, si es posible.

- file — Especifica el nombre del fichero que contiene los datos de configuración de la zona en el fichero de funcionamiento named (por defecto /var/named).
- masters — Se utiliza si la zona se define como type esclava. La opción masters indica al named de un esclavo la/las direcciones en las que se puede pedir información de las zona en la que se tiene autoridad.
- notify — Es parecida a la opción notify que se usa con la declaración option.
- type — Define el tipo de zona. Se pueden usar los siguientes tipos:
- forward — Dice al servidor de nombres que lleve a cabo todas las peticiones de información de la zona en cuestión hacia otros servidores de nombres.
- hint — Tipo especial de zona que se usa para orientar hacia los servidores de nombres root que sirven para resolver peticiones de una zona que no se conoce. Normalmente, no tendrá que configurar una zona que está situada fuera del /etc/named.conf.
- master — Designa el servidor de nombres actual que tiene la autoridad en esa zona. Una zona se puede configurar como tipo master si tiene ficheros de configuración de la zona en el sistema actual.
- slave — Designa el servidor de nombres actual que es servidor esclavo para dicha zona y le dice a named que

pida los ficheros de configuración de la zona de las direcciones IP al servidor de nombres master.

- `zone-statistics` — Dice a `named` que conserve las estadísticas que conciernen a esa zona escribiéndolas bien en la localización por defecto de (`/var/named/named.stats`), o en la localización designada por la opción `statistics-file` en la declaración `server`, si existe.

Ejemplos de declaraciones de zona

La mayor parte de los cambios del fichero `/etc/named.conf` de un servidor de nombres maestro o esclavo se refieren a añdir. modificar o suprimir declaraciones de `zone`. Aunque estas declaraciones pueden contener muchas opciones, la mayor parte de los servidores de nombres usan pocas. Las declaraciones de `zone` siguientes son ejemplos básicos que se pueden usar en una relación de servidores de nombre maestro/esclavo.

Una declaración de `zone` en un servidor de nombres primario del dominio `domain.com` se parece a la Figura 17-5.

```
zone "domain.com" IN {
  type master;
  file "domain.com.zone";
  allow-update { none; };
};
```

Figura 17-5. Ejemplo de una declaración de zone

maestra simple

Esta declaración de `zone` nombra la zona `domain.com`, define la opción`type` como `maître`, dice a `named` que lea el fichero `/var/named/domain.com.zone` para configurar la zona e impide que se actualice otro host.

La declaración de `zone` para `domain.com` se parece a Figura 17-6.

```
zone "domain.com" {
  type slave;
  file "domain.com.zone";
  masters { 192.168.0.1; };
};
```

Figura 17-6. Ejemplo de declaración de zone esclava simple

Esta declaración de `zone` dice a `named` en el servidor esclavo que busque el servidor maestro `192.168.0.1` para encontrar la información de configuración para la zona `domain.com`. La información que el servidor esclavo recibe se registran en el fichero `/var/named /domain.com.zone`.

Ficheros de zona

Los *ficheros de zona* que contienen información sobre el espacio de nombre particular se almacenan en el fichero de funcionamiento de `named`, que por defecto es `/var/named`. Cada fichero de zona se nombra según los datos de las opciones de `file` en la declaración `zone`,

generalmente de tal manera que se refiere al dominio en cuestión e identifica el fichero que contiene los datos de zona como `example.com.zone`.

Cada fichero de zona puede contener directivas y registros de recursos. Las *directivas* dicen al servidor de nombres que efectue una determinada acción o que aplique una configuración especial a la zona. Los *registros de recursos* definen los parámetros de la zona asiganando una identidad a sistemas particulares en el interior del espacio de nombre de la zona. Las directivas no son obligatorias pero los registros de recursos son necesarios para ofrecer un servicio de nombres a dicha zona. Todas las directivas y registros de recursos tienen que estar en la línea correspondiente.

Se pueden escribir comentarios en los ficheros de zona después de los puntos y comas (`;`).

Directivas de los ficheros de zona

Las directivas se identifican por el caracter $ que se sitúa delante del ombre de la directiva y generalmente en la parte de arriba del fichero de zona.

A continuación le mostramos las directivas más usadas:

- `$INCLUDE` — Dice a `named` que incluya otro fichero de zona en el fichero de zona donde se usa la directiva. Así se pueden almacenar configuraciones de zona suplementarias que dependen del fichero de zona principal.
- `$ORIGIN` — Determina el nombre del registro no

cualificado, como por ejemplo los que especifican solamente el host.

Por ejemplo, un fichero de zona puede contener la línea siguiente:

Todos los nombres que se usan en los registros de recursos y qu eno acaban por un punto (`.`) se añaden al nombre de dominio. Es decir, cuando el servidor lee el registro de zona, la primera línea de arriba se interpretará como segunda línea:

```
ftp                IN   CNAME   server1
ftp.domain.com.    IN   CNAME
server1.domain.com.
```

Nota

La utilización de la directiva `$ORIGIN` no es necesaria si nombramos la zona en `/etc/named.conf` con el valor de la opción `$ORIGIN`. El nombre de la zona se usa por defecto como valor de la directiva `$ORIGIN`.

- $TTL — Ajusta el valor *Time to Live (TTL)* predeterminado para la zona. Es el nombre, en segundos, que se da a los servidores de nombres para determinar cuánto tiempo los registros de recursos de la zona serán válidos. Un registro de recursos puede contener su propio valor TTL, que tendrá prioridad sobre la directiva presente.

  Cuando se decide aumentar este valor la esta directiva dice a los servidores de nombres que metan en caché esta

información de zona durante máa tiempo. Esto reduce el número de peticiones de la zona pero aumenta el tiempo necesario para aumentar la cantidad de registros de recursos.

Registros de recursos de ficheros de zona

Los regisitros de recursos de ficheros de zona contienen columnas de datos, separadas por un espacio, que definen estos registros. Todos los registros de recursos de ficheros de zona están asociados a un tipo particular que designa el motivo del registro. A continuación le mostramos los tipos de registros más frecuentes:

- A — Registro de dirección que especifica una dirección IP que se debe asignar a un nombre.

Figura 17-7. Ejemplo de configuración del registro A

Si se omite el valor *<host>*, el registro A designa una dirección IP predeterminada para la parte superior del nombre. Este sistema es la base de todas las peticiones que no sean FQDN.

Consideremos los siguiente ejemplos del registro A:

```
             IN     A       10.0.1.3
server1      IN     A       10.0.1.5
```

Figura 17-8. Ejemplos de registros de A

Las peticiones para el dominio `domain.com` se dirigen al valor 10.0.1.3, mientra que las del dominio `server1.domain.com` se dirigen al valor 10.0.1.5.

- CNAME — Registro del nombre canónico que dice al servidor de nombres qu e todos los nombres son conocidos.

```
<alias-name>      IN      CNAME
<real-name>
```

Figura 17-9. Ejemplo de configuración del registro CNAME

En Figura 17-9, todas las peticiones enviadas a *<alias-name>* se dirigerána al host *<real-name>*. Los registros CNAME son los más utilizados para orientar servicios que usan un procedimiento común para dar nombres a los hosts.

Examinemos el ejemplo de Figura 17-10, donde el registro A fija un nombre de un host particular a una dorección IP y un registro CNAME que orienta los nombres de los hosts más utilizados www.

```
server1         IN      A       10.0.1.5
www             IN      CNAME   server1
```

Figura 17-10. Ejemplo de registro CNAME

- MX — Registro Mail eXchange, que dice dónde se tiene que dirigir el correo enviado a un nombre de espacio particular controlado para esa zona.

```
        IN      MX      <preference-value>
<email-server-name>
```

Figura 17-11. Ejemplo de registro MX configuration

En Figura 17-11, la opción *<preference-value>* le permite listar numéricamente los servidores de correo que seleccione para recibir mensajes para ese espacio de nombres, dando preferencia a ciertos sistemas de correo. El registro de recursos MX tiene el valor más bajo de la opción *<preference-value>*,pero puede ajustar varios servidores de correo con el mismo valor para distribuir el tráfico de correo entre ellos.

La opción *<email-server-name>* puede ser un nombre de host o u FQDN, siempre y cuando se oriente hacia el sistema adecuado.

```
        IN      MX      10
mail.domain.com.
        IN      MX      20
mail2.domain.com.
```

Figura 17-12. Ejemplo de registro MX

En este ejemplo, el primer servidor de correo `mail.domain.com` se prefiere al servidor `mail2.domain.com` para recibir los correos destinados al dominio `domain.com`.

- NS — egistro de servidor de nombres (NameServer) que indica a los servidores los servidores de nombres que tienen la autoridad de una determinada zona.

  Figura 17-13. Ejemplo de configuración del registro NS

  La opción *<nameserver-name>* tiene que ser un FQDN.

  En la Figura 17-14, aparecen dos servidores de nombres

que tienen la autoridad de un dominio. El hecho de que dos servidores sean esclavos o maestros no es importante. Los dos tienen la autoridad en esa zona.

```
        IN      NS      dns1.domain.com.
        IN      NS      dns2.domain.com.
```

Figura 17-14. Ejemplo de registro NS

- PTR — Registro PoinTeR creado para orientar hacia otra parte del espacio de nombres.

  Los registros PTR sirven, sobre todo,para la resolución inversa de nombres ya que reorientan las direcciones IP hacia un mobre determinado. Consulte la la sección de nombre *Ficheros de resolución de nombres inversa* para ver maás ejemplos de esta opción.

- SOA — Registro "Start Of Authority", que proclama información importante sobre la autoridad de determinados servidores sobre determinados espacios de nombres.

  Está situado detrás de las directivas y es el primer registro en un fichero de zona.

```
@       IN      SOA     <primary-name-
server>     <hostmaster-email> (
                        <serial-number>
                        <time-to-refresh>
                        <time-to-retry>
                        <time-to-expire>
                        <minimum-TTL> )
```

Figura 17-15. Ejemplo de configuración del registro SOA

El símbolo @ sitúa la directiva `$ORIGIN` (o el nombre de zona, si la directiva `$ORIGIN` no está instalada) mientras que el espacio de nombre esta7aacute; definido por el registro presente de recursos `SOA`. El servidor de nombres primario que tiene la autoridad para este dominio la usa la opción *<primary-name-server>*, y la dirección de correo electrónico de la persona con la que hay que contactar se reemplaza con *<hostmaster-email>*.

La opción *<serial-number>* aumenta cada vez que cambia el fichero de zona con el fin de que `named` sepa que debe recargar esta zona. *<time-to-refresh>* dice a todos los servidores esclavos cuánto tiempo tienen que esperar antes de pedir información al servidor de nombres maestro si se han realizado cambios en la zona. El valor *<serial-number>* lo utiliza el servidor esclavo para determinar si está usando datos caducados o si tiene que actualizarlos.

La opción *<time-to-retry>* informa al servidor de nombres esclavo sobre el intervalo de tiempo que teiene que esperar antes de emitir una petición de actualización de datos en caso de que el servidor de nombres maestro no le responda. Si el servidor maestro no ha respondido a una petición de actualización de datos antes que se acabe el intervalo de tiempo *<time-to-expire>*, reponde el servidor esclavo presentándose como servidor con la autoridad en este espacio de nombres.

La opción *<minimum-TTL>* pide a los otros servidores de nombres que situen en caché la información de esa zona

durante al menos ese periodo.

En BIND, el tiempo se mide en segundos. Sin embargo, puede usar las abreviaciones de las otras unidades de tiempo como minutos (`M`), horas (`H`), días (`D`), y semanas (`W`). La Tabla 17-1 le muestra la cantidad de teiempo en segundos y el periodo equivalente en otro formato.

Tabla 17-1. Equivalencia en segundos de las otras unidades de tiempo

Segundos	Otras unidades de tiempo.
60	1M
1800	30M
3600	1H
10800	3H
21600	6H
43200	12H
86400	1D
259200	3D
604800	1W

El ejemplo siguiente le muestra a qué se parece el registro de recursos básico SOA.

```
@     IN     SOA    dns1.domain.com.
hostmaster.domain.com. (
                     2001062501 ; serial
                     21600      ;
```

```
refresh after 6 hours
                        3600       ; retry
after 1 hour
                        604800     ;
expires after 1 week
                        86400 )    ;
minimum TTL of 1 day
```

Figura 17-16. Ejemplos de registros SOA

Ejemplos de ficheros de zona

Si obervamos las directivas y los registros de recursos por separado, quizás tengamos algunas dificultades en comprenderlos. Sin embargo, si los estudiamos en su conjunto en un fichero no resulta tan difícil.

He aquí un fichero de zona clásico en la Figura 17-17

```
$ORIGIN domain.com
$TTL 86400
@     IN     SOA    dns1.domain.com.
hostmaster.domain.com. (
                        2001062501 ; serial
                        21600      ; refresh
after 6 hours
                        3600       ; retry
after 1 hour
                        604800     ; expires
after 1 week
                        86400 )    ; minimum
TTL of 1 day
```

```
        IN      NS      dns1.domain.com.
        IN      NS      dns2.domain.com.

        IN      MX      10      mail.domain.com.
        IN      MX      20
mail2.domain.com.

                IN      A       10.0.1.5

server1         IN      A       10.0.1.5
server2         IN      A       10.0.1.7
dns1            IN      A       10.0.1.2
dns2            IN      A       10.0.1.3

ftp             IN      CNAME   server1
mail            IN      CNAME   server1
mail2           IN      CNAME   server2
www             IN      CNAME   server2
```

Figura 17-17. Ejemplo de fichero de zona básico

En este ejemplo se usan las directivas estándar y los valores SOA. Se ve que los servidores de nombres que tendrán la autoridad son `dns1.domain.com` y `dns2.domain.com`, que tienen como registros A que les conecta respectivamente a `10.0.1.2` y `10.0.1.3`.

Los servidores de correo configruados por los registros MX s eorientan hacia los servidores `server1` y `server2` gracias a los registros CNAME. Debido a que los nombres

de los servidores `server1` y `server2` no acaban con un punto (`.`), el dominio `$ORIGIN` se situa a continuación, extendiendo los dominios `server1.domain.com` y `server2.domain.com`. Gracias a los registros de recursos A se peuden determinar las direcciones IP.

Los servicios ftp y de web, disponibles en `ftp.domain.com` y `www.domain.com`, están orientados a máuinas que ofrecen los servicios apropiados para dichos nombres usando el registro `CNAME`.

Ficheros de resolución de nombres inversa

Una resolución de nombres inversa sirve para traducir una dirección IP en un espacio de nombres particular en un FQDN. Esto se parece mucho a un fichero de zona estándar, salvo que los registros de recursos PTR sirven para relacionar las direcciones IP con los nombres de un sistema determinado.

La escritura de un registro PTR se hace de la misma manera que en la Figura 17-18.

```
<last-IP-digit>  IN  PTR  <FQDN-of-system>
```

Figura 17-18. Ejemplo de configuración del registro PTR

<last-IP-digit> hace referencia al último nombre en una dirección IP que tiene que orientar el FQDN de un determinado sistema.

En la Figura 17-19, las direcciones IP de `10.0.1.20` a `10.0.1.25` se orientan a los FQDNs correspondientes.

```
$ORIGIN 1.0.10.in-addr.arpa
$TTL 86400
@     IN     SOA    dns1.domain.com.
hostmaster.domain.com. (
                     2001062501 ; serial
                     21600      ; refresh
after 6 hours
                     3600       ; retry
after 1 hour
                     604800     ; expire
after 1 week
                     86400 )    ; minimum
TTL of 1 day

      IN     NS     dns1.domain.com.
      IN     NS     dns2.domain.com.

20    IN     PTR    alice.domain.com.
21    IN     PTR    betty.domain.com.
22    IN     PTR    charlie.domain.com.
23    IN     PTR    doug.domain.com.
24    IN     PTR    ernest.domain.com.
25    IN     PTR    fanny.domain.com.
```

Figura 17-19. Ejemplo de un fichero básico de resolución inversa de zona

Este fichero de zona se tiene que poner en funcionamiento con una declaración `zone` en el fichero `/etc/named.conf`, parecido a la Figura 17-20.

```
zone "1.0.10.in-addr.arpa" IN {
  type master;
  file "domain.com.rr.zone";
  allow-update { none; };
};
```

Figura 17-20. Ejemplo de declaración zone de resolución inversa

Existe algunas pequeñas diferencias ente este ejemplo y una declaración `zone` estándar, salvo en la manera de nombrar el host. Observe que uns zona de resolución de nombres inversa necesita que los tres primeros bloques de la dirección IP sean invertidos y que "`.in-addr.arpa`" esté a su vez incluido. Esto permite asociar correctamente a la zona el bloque único de nombres que se usa en el fichero de zona de resolución de nombres inversa.

Keywords: linux colombia tienda debian redhat red hat informática virtual compra comprar venta hosting cd libre gpl software hardware freeware slackware SuSE caldera mandrake stampede linux libro libros cd-rom cdrom nuevo ultimo último descarga administrador dns precio staroffice business card manual manuales release versión version documentacion documentación docs documentación info información distribución distribucion distribuciones oficial openlinux open aplicaciones kilyx comercial productos servicios catalogo barato bajo costo noticias recursos comentarios comercio electronico electrónico desarrollo tiendas virtuales linea online on-line soporte bogota nfs ftp http interchange minivend seguridad freebsd winlinux

instalación instalacion configuracion configuración corel carro carrito descripcion descripción mercado mercados gratis gnu iso image unix windows download 2000 2001 5.2 8.1 7.1 7.0 6.1 6.2 6.3 7.2 7.3 8.0 8.1 8.2 9.0 9.1 9.2 1.0

tiendalinux.com

Utilización del comando rndc

7-9 minutos

BIND incluye la utilidad `rndc`, que permite admnistrar localmente o a distancia, el demonio `named` gracias a las declaraciones de las líneas de comandos. El programa `rndc` utiliza el fichero `/etc/rndc.conf` para las opciones de configuración que serán sobrescritas por las opciones de las líneas de comandos.

Para evitar que los usuarios no autorizados a controlar BIND en su sitema, se utiliza el método de claves secretas compartidas para dar privilegios a determinados hosts. Para que `rndc` emita comandos hacia cualquier `named`, incluso hacia la máquina local, las claves utilizadas en los ficheros `/etc/named.conf` y `/etc/rndc.conf` se tienen que corresponder.

Ficheros de configuración

Antes de usar el comando `rndc`, verifique que las líneas de configuración adecuadas se encuentren en un los ficheros necesarios. Es probable que los ficheros de configuración no estén instalados como debería ser si después de que lanza el comando aparece el siguiente mensaje `rndc`:

```
rndc: connect: connection refused
```

/etc/named.conf

Para que el comando rndc se pueda conectar al servicio named, tiene que tener una declaración controls en el fichero /etc/named.conf cuando arranque named. El ejemplo de declaración controls que se muestra en la Figura 17-21 le permite ejecutar los comandos rndc localmente.

```
controls {
  inet 127.0.0.1 allow { localhost; } keys
{ <key-name>; };
};
```

Figura 17-21. Ejemplo de declaración de controls dans /etc/named.conf

Esta declaracoón dic a named que tiene que escuchar en el puerto TCP 953 por defecto de la dirección inversa y que tiene que autorizar los comandos rndc que provienen del host local, si existe la clave adecuada. *<key-name>* hace referencia a la declaración key, que también se encuentra en el fichero /etc/named.conf. Puede ver un ejemplo de este tipo de declaración en Figura 17-22.

```
key "<key-name>" {
  algorithm hmac-md5;
  secret "<key-value>";
};
```

Figura 17-22. Ejemplo de declaración key en `/etc/named.conf`

En este caso, *<key-value>* es una clave HMAC-MD5. Puede crear sus propias claves HMAC-MD5 con el siguente comando.

```
dnssec-keygen -a hmac-md5 -b <bit-length>
-n HOST <key-file-name>
```

Una calve de al menos 256 bits de longitud es una buena elección. La clave creada tiene que situarse en la zona *<key-value>* que se encuentra en *<key-file-name>*.

El nombre de la clave utilizada en `/etc/named.conf` tiene que ser diferente de `key`.

`/etc/rndc.conf`

Para configurar `rndc` para que utilice automáticamente la clave creada en el fichero `/etc/named.conf` para el host local, tiene que tener tres declaraciones. La declaración `options` le permite ajustar el servidor y la clave por defecto para usar el comando `rndc`, como se muestra en la Figura 17-23.

```
options {
  default-server  localhost;
  default-key     "<key-name>";
};
```

Figura 17-23. Ejemplo de declaración `options` en `/etc/rndc.conf`

Alernativamente, se puede decir al comando `rndc` que use una clave por defecto cuando acceda a un servidor determinado, como se muestra en la Figura 17-24.

```
server localhost {
  key  "<key-name>";
};
```

Figura 17-24. Ejemplo de declaración `server` en `/etc/rndc.conf`

Sin embargo, esta declaración `server` es realmente útil si se quiere conectar a varios servidores con el comando `rndc`.

La declaración `key` es la más importante del fichero `/etc/rndc.conf`.

```
key "<key-name>" {
  algorithm hmac-md5;
  secret "<key-value>";
};
```

Figura 17-25. Ejemplo de declaración `key` en `/etc/rndc.conf`

<key-name> y *<key-value>* tienen que ser completamente iguales a la configuración en el fichero `/etc/named.conf`.

Para probar todas las configuraciones, pruebe el comando `rndc reload`. Verá algo parecido a lo siguiente:

```
rndc: reload command successful
```

Si el comando no funciona, examine los ficheros `/etc/named.conf` y `/etc/rndc.conf` para buscar los errores.

Advertencia

Hay que asegurarse que los usuarios no autorizados no puedan leer ni escribir en el fichero `/etc/rndc.conf`.

Opciones de la línea de comandos

Un comando `rndc` tiene la siguiente forma:

```
rndc <options> <command> <command-options>
```

Figura 17-26. Estructura del comando `rndc`

La zona *<options>* no es necesaria ni tampoco es necesario que use *<command-options>* salvo si lo requiere el comando.

Cuando ejecuta el comando `rndc` en un host local, se encuentran disponibles los siguientes comandos:

- `halt` — Para inmediatamente el servicio `named`.
- `querylog` — Ejecuta la conexión para todas las peticiones efectuadas por los clientes hacia el servidor de nombres.
- `refresh` — Actualiza la base de datos del servidor de nombres.
- `reload` — Dice al servidor de nombres que recargue los ficheros de zona paro que conserve todas las respuestas precedentes situadas en caché. Esto le permite realizar cambios en los ficheros de zona y de ponerlos en práctica

en los servidores maestros y esclavos sin perder las resoluciones de nombres almacenadas.

Si los cambios no afectan a una zona determinada, puede decir al comando named que recargue esa zona. Escriba el nombre de la zona después del comando `reload`.

- `stats` — Pasa las estadísticas del comando named al fiochero `/var/named/named.stats`.
- `stop` — Para el servidor salvando todas las actualizaciones dinámicas y los datos IXFR antes de parar el servidor completamente.

  Se pueden sobrescribir los parámetros predeterminados del fichero `/etc/rndc.conf`. Existen varias posiblidades:

- `-c <configuration-file>` — Dice al comamdo t à `rndc` que use otro fichero de configuración diferente del fichero predeterminado `/etc/rndc.conf`.
- `-p <port-number>` — Especifica la utilización de un número de puerto diferente del predeterminado 953 para la conexión del comando `rndc`.
- `-s <server>` — Dice a `rndc` que envie comandos a otro servidor distinto del servidor que designa la opción `default-server` en el fichero `/etc/rndc.conf`.

  Para que se lleve a cabo esta tarea, tiene que aver configurado el servicio named para que acepte los comandos del host y que tenga la clave para este servicio de nombres.

- `-y <key-name>` — Le permite especificar una clave

distinta de la opción `default-key` en el fichero `/etc/rndc.conf`.

Para mayor información sobre estas opciones, consulte la página man del comando `rndc`.

Keywords: linux colombia tienda debian redhat red hat informática virtual compra comprar venta hosting cd libre gpl software hardware freeware slackware SuSE caldera mandrake stampede linux libro libros cd-rom cdrom nuevo ultimo último descarga administrador dns precio staroffice business card manual manuales release versión version documentacion documentación docs documentación info información distribución distribucion distribuciones oficial openlinux open aplicaciones kilyx comercial productos servicios catalogo barato bajo costo noticias recursos comentarios comercio electronico electrónico desarrollo tiendas virtuales linea online on-line soporte bogota nfs ftp http interchange minivend seguridad freebsd winlinux instalación instalacion configuracion configuración corel carro carrito descripcion descripción mercado mercados gratis gnu iso image unix windows download 2000 2001 5.2 8.1 7.1 7.0 6.1 6.2 6.3 7.2 7.3 8.0 8.1 8.2 9.0 9.1 9.2 1.0

tiendalinux.com

Propidades avanzadas de BIND

5-6 minutos

La mayor parte de las acciones de BIND usan named solamente para ofrecer un servicio de resolución de nombres o para obtener la autoridad de un dominio o subdominio particular. Sin embargo, la versión 9 de BIND posee un determinado número de propiedades avanzadas que cuando se configuran y se usan de manera adecuada, permiten ofrecer un servicio DNS eficaz y seguro.

 **Advertencia**

Algunas de estas propiedades avanzadas como DNSSEC, TSIG y IXFR, solamente se pueden usar en los entornos de red que tengan servidores de nombres que soporten estas propiedades. Si su entorno de red incluye servidores de nombres no-BIND o versiones anteriores de BIND verifique si la propiedad avanzada está soportada.

No se presupone que otro servidor de nombres soporte estas propiedades porque en general no lo hacen.

Todas las propiedades citadas aquí se describen en el *Manual de referencia del administrador de BIND 9*. Consulte la la sección de nombre *Recursos adicionales*

para mayor información.

Mejoras del protocolo DNS

BIND soporta las *Incremental Zone Transfers,(IXFR)*, en las que el servidor de nombres esclavo descarga solamente las porciones de actualizaciones de una zona modificada en un servidor de nombres maestro. El proceso de tranferencia AXFR estándar necesita que la zona entera se transfiera al servidor de nombres esclavo incluso si se hacen pequeños cambios. Para los dominios más famosos con ficheros de zona muy largos, IXFR hace que la notificación y los procesos de actualización sean menos exigentes en recursos.

Observe que IXFR solamente estáa disponible si usa al mismo tiempo la *actualización dinámica* para realizar los cambios en los registros de zona maestra. Si cambia los ficheros de zona manualmente para dichos cambios, tiene que usar AXFR. Encontrará más información en el *Manual de referencia del administrador* .

Visualizaciones múltiples

BIND le permite con la opción `view` en `/etc/named.conf`, configurar un servidor de nombres para responder a las peticiones de determinados clientes en una manera diferente de otros.

Esto es útil sobre todo si desea que clientes externos a su red no puedan ejecutar un servicio DNS particular o que accedan a una determinada información , siempre y

cuando se autoricen a los clientes internos.

La declaración `view` utiliza la opción `match-clients` para hacer corresponder las direcciones IP o las redes enteras y atribuirlas opciones y datos de zonas específicas.

Seguridad

BIND soporta varios métodos diferentes para proteger la actualización y la transferencia de zonas al mismo tiempo que a los srvidores de nombres maestros y esclavos:

- *DNSSEC* — Abreviación de *DNS SECurity*, esta propiedad permite firmar con caracteres criptográficos zonas con una *clave de zona (zone key)*.

  De esta manera, puede verificar que la información de una zona provenga de un servidor de nombres que la ha firmado con caracteres crptográficos con una clave privada desde el momento en que el receptor posee la clave pública de ese servidor de nombres.

  La versión 9 de BIND soporta también el método de la llave públoca/privada SIG(0) para la autentificación de mensajes.

- *TSIG* — AbreviaciOacute;n de *Transaction SIGnatures*, instala una clave secreta compartida en el servidor maestro y el servidor esclavo y verifica que la transferencia del servidor maestro al esclavo esté autorizada.

  Esta propiedad refuerza la autorización de transferencias basada en la dirección IP estándar. Un agresor no puede acceder a la dirección IP para transferir la zona porque necesita conocer la clave secreta.

La versión 9 de BIND soporta también *TKEY*, que es otro método de clave secreta compartida para autorizar las transferencias de zona.

IP versión 6

La versión 9 de BIND ofrece un servicio de nombres en la versión 6 (IPv6), gracias a los registros de zona A6.

Si su entorno de red incluye al mismo tiempo hosts IPv4 y IPv6, necesita usar el demonio de resolución ligero `lwresd` en los clientes de red. Este demonio es muy eficaz y funciona solamente en caché, que soporta los nuevos registros A6 y DNAME que funcionan con IPv6. Consulte la página man `lwresd` para mayor información.

Keywords: linux colombia tienda debian redhat red hat informática virtual compra comprar venta hosting cd libre gpl software hardware freeware slackware SuSE caldera mandrake stampede linux libro libros cd-rom cdrom nuevo ultimo último descarga administrador dns precio staroffice business card manual manuales release versión version documentacion documentación docs documentación info información distribución distribucion distribuciones oficial openlinux open aplicaciones kilyx comercial productos servicios catalogo barato bajo costo noticias recursos comentarios comercio electronico electrónico desarrollo tiendas virtuales linea online on-line soporte bogota nfs ftp http interchange minivend seguridad freebsd winlinux instalación instalacion configuracion configuración corel carro carrito descripcion descripción mercado mercados

gratis gnu iso image unix windows download 2000 2001 5.2 8.1 7.1 7.0 6.1 6.2 6.3 7.2 7.3 8.0 8.1 8.2 9.0 9.1 9.2 1.0

tiendalinux.com

Errores frecuentes que hay que evitar

2-3 minutos

Es normal que los principiantes cometan errores modificando los ficheros de configuración BIND o que encuentren dificultades a la hora de activar el comando named. Evite los siguientes errores:

- *Asegúrese que aumenta el número de serie cuando modifique un fichero de zona.*

  Si el número de serie no ha aumentado, puede ser que el servidor de nombres maestro posea información nueva y correcta pero que los servidores esclavos no hayan recibido ninguna notificación de los cambios o que no intenten actualizar los datos de esa zona. Después de todo, el número de serie corresponde al que se encuentra en el servidor maestro aunque los datos de esta zona sean completamente diferentes a los del servidor maestro.

- *Preste atención a la utilización de los corchetes y de los puntos y comas en el fichero `/etc/named.conf`.*

  La omisión de un punto y coma o de un corchete impide que arranque named.

- *Acuérdese de situar los puntos (`.`) en los ficheros de zona*

después de los FQDN y de evitarlos en los nombres de hosts.

El punto significa que el nombre está completo. Si se omite el punto, `named` situará el nombre de la zona o el valor `$ORIGIN` después del nombre para completarlo.

- *Si el firewall bloquea las conexiones con el comando `named` con otros servidores de nombres, tendrá que decir a `named` que use el puerto 53 para las peticiones acumulativas.*

  La versión 9 de BIND utiliza puertos al azar que superan el 1024 para pedir a otros servidores de nombres la resolución de nombres, como lo haría un cliente DNS conectándose al puerto 53 del servidor de nombrs distante. Existe, sin embargo, firewalls que exigen que los servidores de nombrs se comuniquen entre ellos usando ambos el puerto 53. Puede añadir la línea siguiente en la declaración `options` para forzar a `named` a que envíe peticiones desde le puerto 53:

  ```
  query-source address * port 53;
  ```

tiendalinux.com

Recursos adicionales

2 minutos

BIND propone toda una gama de documentación instalada que cubre numerosos aspectos, cada uno se encuentra en un fichero distinto:

- `/usr/share/doc/bind-<version-number>` — Contiene un fichero README con la lista de las propiedades más recientes.
- `/usr/share/doc/bind-<version-number>/arm` — Contiene las versiones en HTML y SGML del *Manual de referncia del administrados BIND 9*, que describe en detalle los recursos necesarios para BIND, la manera de configurar los diferentes tipos de servidores de nombres, cómo establecer un equilibrio de cargas y otros temas de interés. Es el mejor punto de partida para los principiantes.
- `/usr/share/doc/bind-<version-number>/draft` — Contiene los documentos técnicos que se refieren al servicio DNS y a algunos métodos propuestos para responder a las resoluciones.
- `/usr/share/doc/bind-<version-number>/misc` — Contiene documentos para cuestiones avanzadas específicas. Los usuarios de la versión 8 de BIND tendrían

que consultar el documento `migration` en lo que se refiere a los cambios importantes cuando se pasa a la versión 9 de BIND. El fichero `options` lista todas las opciones disponibles en BIND 9 que se usan en el fichero `/etc/named.conf`.

- `/usr/share/doc/bind-<version-number>/rfc` — Documento RFC sobre BIND.

Capítulo 12. Berkeley Internet Name Domain (BIND)

En la mayoría de las redes modernas, incluyendo la Internet, los usuarios localizan otras máquinas por su nombre. Esto libera a los usuarios de la pesada tarea de recordar la dirección numérica de los recursos de red. La forma más efectiva de configurar una red para permitir tales conexiones basadas en nombres es configurando un *Domain Name Service* (*DNS*) o *servidor de nombres*, el cual resuelve los nombres de hosts en la red a direcciones numéricas y viceversa.

Este capítulo revisa el servidor de nombres incluido con Red Hat Enterprise Linux, el servidor DNS *Berkeley Internet Name Domain* (*BIND*), con énfasis en la estructura de sus archivos de configuración y en cómo deberían ser administrados localmente y remótamente.

Para instrucciones sobre la configuración de BIND usando la **Herramienta de configuración del Servicio de Nombres de Dominio** (`redhat-config-bind`) gráfica, consulte el capítulo llamado ***Configuración de BIND*** en el ***Manual de administración del sistema de Red Hat Enterprise Linux***.

Aviso

Si utiliza la **Herramienta de configuración del Servicio de Nombres de Dominio**, no edite manualmente ningún archivo de configuración BIND pues todos los cambios serán sobreescritos la próxima vez que utilice la **Herramienta de configuración del Servicio de Nombres de Dominio**.

12.1. Introducción a DNS

Cuando hosts en una red se conectan a través de sus nombres de máquinas, también llamado *nombre de dominio completamente cualificado (FQDN)*, DNS es usado para asociar los nombres de las máquinas a las direcciones IP para el host.

El uso de nombres de un dominio completamente cualificado y DNS tiene ventajas para los administradores del sistema, éstos dan a los administradores flexibilidad a la hora de cambiar las direcciones IP para máquinas individuales sin realizar preguntas sobre el nombre en las máquinas. Por otro lado, los administradóres pueden revolver cuáles máquinas manejan consultas basadas en nombre .

DNS es normalmente implementado usando servidores centralizados que autorizan algunos dominios y se refieren a otros servidores DNS para otros dominios.

Cuando un host cliente solicita información desde un servidor de nombres, usualmente se conecta al puerto 53. El nombre de servidor luego intenta resolver el

FQDN basado en su librería de resolución, la cual puede contener información de autorización sobre el host solicitado o datos en caché de una consulta anterior. Si el nombre del servidor no tiene la respuesta en su librería de resolución, consultará otros nombres de servidores, llamados *servidores de nombres de root*, para determinar cuáles servidores de nombres son fidedignos para el FQDN en cuestión. Luego, con esa información, consulta los servidores de nombres autoritarios para determinar la dirección IP del host solicitado. Si se está realizando una búsqueda inversa, se usa el mismo procedimiento, excepto que la consulta es realizada con una dirección IP desconocida en vez de un nombre.

12.1.1. Zonas de servidores de nombres

En Internet, el FQDN de un host se puede dividir en diversas secciones. Estas secciones son organizadas en orden jerárquico (como en un árbol), con un tronco, ramas principales, ramas secundarias, etc. Por ejemplo, considere el siguiente FQDN:

```
bob.sales.example.com
```

Cuando miramos cómo un FQDN es resuelto para encontrar la dirección IP que se relaciona a un sistema particular, lea el nombre de derecha a izquierda, con cada nivel de la jerarquía dividido por puntos (`.`). En nuestro ejemplo, `com` define el *dominio de nivel superior* para este FQDN. El nombre `example` es un subdominio bajo `com`, mientras que `sales` es un subdominio bajo `example`. El nombre más hacia la izquierda, `bob`, identifica el nombre de una máquina específica.

Aparte del nombre del dominio, cada sección se llama *zona*, la cual define un *espacio de nombre* particular. Un espacio de nombre, controla los nombres de los subdominios de la izquierda. Aunque en el ejemplo solamente hay dos subdominios, un FQDN tiene que contener al menos un subdominio pero puede incluir muchos más; depende de la organización del espacio de nombres elegido.

Las zonas son definidas en servidores de nombres autorizados a través del uso de *archivos de zona*, lo cual describen el espacio de nombres de esa zona, los servidores de correo a ser utilizados por un dominio particular o sub-dominio, y más. Los archivos de zona son almancenados en *servidores de nombres primarios* (también llamados *servidores de nombres maestro*), los cuales son verdaderamente autorizados y donde los cambios se hacen a los archivos, y *servidores de nombres secundarios* (también llamados *servidores de nombres esclavos*), que reciben sus archivos de zona desde los servidores de nombres primarios. Cualquier servidor de nombres puede ser un servidor primario y secundario para zonas diferentes al mismo tiempo, y también pueden ser considerados autoritarios para múltiples zonas. Todo depende de cómo se configure el servidor de nombres.

12.1.2. Tipos de servidores de nombres

Existen cuatro tipos de configuración de servidores de nombres primarios:

- *maestro* — Almacena los registros de las zonas originales y de autoridad para un cierto espacio de nombres y responde a consultas sobre el espacio de nombres de otros servidores de nombres.
- *esclavo* — Responde a las peticiones que provienen de otros servidores de nombres y que se refieren a los espacios de nombres sobre los que tiene autoridad. Sin embargo, los servidores esclavos obtienen la información de sus espacios de nombres desde los servidores maestros.
- *sólo caché* — ofrece servicios de resolución de nombres a direcciones IP pero no tiene ninguna autoridad sobre ninguna zona. Las respuestas en general se introducen en un caché por un período de tiempo fijo, la cual es especificada por el registro de zona recuperado.
- *reenvío* — Reenvía las peticiones a una lista específica de servidores de nombres para la resolución de nombres. Si ninguno de los servidores de nombres especificados puede resolver los nombres, la resolución falla.

Un servidor de nombres puede ser uno o más de estos tipos. Por ejemplo, un servidor de nombres puede ser un maestro para algunas zonas, un esclavo para otras y sólo ofrecer el reenvío de resoluciones para otras.

12.1.3. BIND como un servidor de nombres

BIND realiza la resolución de nombres a través del demonio `/usr/sbin/named`. BIND también incluye una utilidad de administración llamada `/usr/sbin/rndc`. Se puede encontrar más información sobre `rndc` en Sección 12.4.

BIND almacena sus archivos de configuración en los siguientes lugares:

- `/etc/named.conf` — El archivo de configuración para el demonio `named`.
- directorio `/var/named/` — El directorio de trabajo `named` el cual almacena zonas, estadísticas y archivos caché.

Las próximas secciones revisan los archivos de configuración de BIND en más detalle.

12.2. /etc/named.conf

El archivo `named.conf` es una colección de declaraciones usando opciones anidadas rodeadas por carácteres de llaves, `{ }`. Los administradores deben tener mucho cuidado cuando estén modificando `named.conf` para evitar errores sintácticos puesto que hasta el error más pequeño puede impedir que el servicio `named` arranque.

> **Aviso**
>
> *No* modifique manualmente el archivo `/etc/named.conf` o cualquier archivo en el directorio `/var/named/` si está usando la **Herramienta de configuración del Servicio de Nombres de Dominio**. Cualquier cambio manual a esos archivos serán sobreescritos la próxima vez que se use **Herramienta de configuración del Servicio de Nombres de Dominio**.

Un archivo típico de `named.conf` está organizado de forma similar al ejemplo siguiente:

```
<statement-1> ["<statement-1-name>"] [<statement-1-class>] {
   <option-1>;
   <option-2>;
   <option-N>;
};

<statement-2> ["<statement-2-name>"] [<statement-2-class>] {
   <option-1>;
   <option-2>;
   <option-N>;
};

<statement-N> ["<statement-N-name>"] [<statement-N-class>] {
   <option-1>;
   <option-2>;
   <option-N>;
};
```

12.2.1. Tipos de declaraciones comúnes

Los siguientes tipos de sentencias son usados a menudo en `/etc/named.conf`:

12.2.1.1. Declaración `acl`

La sentencia `acl` (o sentencia de control de acceso) define grupos de hosts a los que se les puede permitir o negar el acceso al servidor de nombres.

Una declaración `acl` tiene la siguiente forma:

```
acl <acl-name> {
    <match-element>;
    [<match-element>; ...]
};
```

En esta declaración, sustituya *<acl-name>* con el nombre de la lista de control de acceso y reemplace *<match-element>* con una lista de direcciones IP separada por puntos y comas. La mayoría de las veces, una dirección IP individual o notación de red IP (tal como `10.0.1.0/24`) es usada para identificar las direcciones IP dentro de la declaración `acl`.

La siguiente lista de control de acceso ya están definidas como palabras claves para simplificar la configuración:

- `any` — Hace coincidir todas las direcciones IP.
- `localhost` — Hace coincidir cualquier dirección IP que se use el sistema local.
- `localnets` — Hace coincidir cualquier dirección IP en cualquier red en la que el sistema local está conectado.
- `none` — No concuerda ninguna dirección IP.

Cuando lo utilice con otras pautas (tales como declaraciones `options`), las declaraciones `acl` pueden ser muy útiles al

asegurar el uso correcto de su servidor de nombres BIND.

El ejemplo siguiente define dos listas de control de acceso y utiliza una declaración `options` para definir cómo son tratadas en el servidor de nombres:

```
 acl black-hats {
    10.0.2.0/24;
    192.168.0.0/24;
 };

acl red-hats {
    10.0.1.0/24;
 };

options {
    blackhole { black-hats; };
    allow-query { red-hats; };
    allow-recursion { red-hats; };
 }
```

Este ejemplo contiene dos listas de control de acceso, `black-hats` y `red-hats`. Los hosts en la lista `black-hats` se les niega el acceso al servidor de nombres, mientras que a los hosts en la lista `red-hats` se les dá acceso normal.

12.2.1.2. Declaración `include`

La declaración `include` permite incluir archivos en un `named.conf`. De esta forma los datos de configuración confidenciales (tales como `llaves`) se pueden colocar en un archivo separado con permisos restringidos.

Una declaración `include` tiene la forma siguiente:

```
include  "<file-name>"
```

En esta declaración, *<file-name>* es reemplazado con una ruta absoluta a un archivo.

12.2.1.3. Declaración `options`

La declaracion `options` define opciones de configuración de servidor globales y configura otras declaraciones por defecto. Puede ser usado para especificar la ubicación del directorio de trabajo `named`, los tipos de consulta permitidos y mucho más.

La declaración `options` toma la forma siguiente:

```
options {
        <option>;
        [<option>; ...]
};
```

En esta declaración, las directivas *<option>* son reemplazadas con una opción válida.

Las siguientes son opciones usadas a menudo:

- `allow-query` — Especifica cuáles hosts tienen permitido consultar este servidor de nombres. Por defecto, todos los hosts tienen derecho a consultar. Una lista de control de acceso, o una colección de direcciones IP o redes se puede usar aquí para sólo permitir a hosts particulares hacer consultas al servidor de nombres.
- `allow-recursion` — Parecida a la opción `allow-query`, salvo que se aplica a las peticiones recursivas. Por defecto, todos los hosts están autorizados a presentar peticiones recursivas en un servidor de nombres.
- `blackhole` — Especifica cuáles hosts no tienen permitido consultar al servidor de nombres.
- `directory` — Especifica el directorio de trabajo `named` si es diferente del valor predeterminado `/var/named`.
- `forward` — Especifica el comportamiento de reenvío de una directiva `forwarders`.

  Se aceptan las siguientes opciones:

  - `first` — Indica que los servidores de nombres especificados en la directiva `forwarders` sean consultados antes de que `named` intente resolver el nombre el mismo.
  - `only` — Especifica que `named` no intente la resolución de nombres él mismo en el evento de que fallen las

consultas a los servidores de nombres especificados en la directriz `forwarders`.

- `forwarders` — Especifica una lista de direcciones IP válidas para los servidores de nombres donde las peticiones se pueden reenviar para ser resueltas.
- `listen-on` — Especifica la interfaz de red en la cual `named` escucha por solicitudes. Por defecto, todas las interfaces son usadas.

  Usando esta directiva en un servidor DNS que también actúa como una gateway, BIND se puede configurar para sólo contestar solicitudes que se originan desde algunas de las redes.

  Una directiva `listen-on` se parece al ejemplo siguiente:

```
options {
   listen-on { 10.0.1.1; };
};
```

  En este ejemplo, solamente son aceptadas las peticiones que llegan desde la interfaz de red sirviendo a la red privada (`10.0.1.1`).
- `notify` — Controla si `named` notifica a los servidores esclavos cuando una zona es actualizada. Acepta las opciones siguientes:
  - `yes` — Notifica a los servidores esclavos.
  - `no` — No notifica a los servidores esclavos.
  - `explicit` — Solamente notifica a los servidores esclavos especificados en una lista de `also-notify` dentro de la declaración de una zona.
- `pid-file` — Especifica la ubicación del archivo del proceso ID creado por `named`.
- `root-delegation-only` — Activa la implementación de las propiedades de delegación en dominios de nivel superior (TLDs) y zonas raíz con una lista opcional de exclusión. La *delegación* es el proceso de separar una zona sencilla en múltiples zonas. Para poder crear una zona delegada, se utilizan elementos conocidos como *registros NS*. Los registros de servidor de nombres (registros de delegación) anuncian los servidores de nombres autorizados para una zona particular.

  El siguiente ejemplo de `root-delegation-only` especifica una lista excluyente de los TDLs desde los que se esperan respuestas no delegadas:

```
options {
   root-delegation-only exclude { "ad"; "ar"; "biz"; "cr"; "cu"; "de"; "dm"; "id;
                                  "lu"; "lv"; "md"; "ms"; "museum"; "name"; "no"; "pa";
                                  "pf"; "se"; "sr"; "to"; "tw"; "us"; "uy"; };
};
```

- `statistics-file` — Permite especificar la localización alternativa de los archivos de estadísticas. Por defecto, las estadísticas de `named` son guardadas al archivo `/var/named/named.stats`.

Existen numerosas opciones disponibles, muchas de ellas dependen unas de otras para poder funcionar correctamente. Consulte el ***Manual de referencia para el administrador de BIND 9*** en Sección 12.7.1 y la página man para `bind.conf` para más detalles.

12.2.1.4. Declaración `zone`

Una declaración `zone` define las características de una zona tal como la ubicación de su archivo de configuración y opciones específicas de la zona. Esta declaración puede ser usada para ignorar las declaraciones globales `options`.

Una declaración `zone` tiene la forma siguiente:

```
zone <zone-name> <zone-class> {
     <zone-options>;
     [<zone-options>; ...]
};
```

En esta declaración, *<zone-name>* es el nombre de la zona, *<zone-class>* es la clase opcional de la zona, y *<zone-options>* es una lista de opciones que caracterizan la zona.

El atributo *<zone-name>* para la declaración de zona es particularmente importante, pues es el valor por defecto asignado

para la directriz `$ORIGIN` usada dentro del archivo de zona correspondiente localizado en el directorio `/var/named/`. El demonio `named` anexa el nombre de la zona a cualquier nombre de dominio que no esté completamente cualificado listado en el archivo de zona.

Por ejemplo, si una declaración `zone` define el espacio de nombres para `example.com`, utilice `example.com` como el *<zone-name>* para que sea colocado al final de los nombres de hosts dentro del archivo de zona `example.com`.

Para más información sobre los archivos de zona, consulte Sección 12.3.

Las opciones más comunes para la declaración `zone` incluyen lo siguiente:

- `allow-query` — Especifica los clientes que se autorizan para pedir información sobre una zona. Por defecto, todas las peticiones de información son autorizadas.
- `allow-transfer` — Especifica los servidores esclavos que están autorizados para pedir una transferencia de información de la zona. Por defecto, todas las peticiones se autorizan.
- `allow-update` — Especifica los hosts que están autorizados para actualizar dinámicamente la información en sus zonas. Por defecto, no se autoriza la actualización de la información dinámicamente.

  Tenga cuidado cuando autorice a los hosts para actualizar la información de su zona. No habilite esta opción si no tiene confianza en el host que vaya a usar. Es mejor que el administrador actualice manualmente los registros de zona y que vuelva a cargar el servicio `named`.
- `file` — Especifica el nombre del archivo en el directorio de trabajo `named` que contiene los datos de configuración de zona.
- `masters` — Especifica las direcciones IP desde las cuales solicitar información autorizada. Solamente se usa si la zona está definida como `type slave`.
- `notify` — Controla si `named` notifica a los servidores esclavos cuando una zona es actualizada. Esta directiva sólo acepta las opciones siguientes:
  - `yes` — Notifica a los servidores esclavos.
  - `no` — No notifica a los servidores esclavos.
  - `explicit` — Solamente notifica a los servidores esclavos especificados en una lista de `also-notify` dentro de la declaración de una zona.
- `type` — Define el tipo de zona.

  Abajo se muestra una lista de las opciones válidas:
  - `delegation-only` — Refuerza el estado de delegación de las zonas de infraestructura tales como COM, NET u ORG. Cualquier respuesta recibida sin una delegación explícita o implícita es tratada como `NXDOMAIN`. Esta opción solamente es aplicable en TLDs o en archivos raíz de zona en implementaciones recursivas o de caché.
  - `forward` — Dice al servidor de nombres que lleve a cabo todas las peticiones de información de la zona en cuestión hacia otros servidores de nombres.
  - `hint` — Tipo especial de zona que se usa para orientar hacia los servidores de nombres root que sirven para resolver peticiones de una zona que no se conoce. No se requiere mayor configuración que la establecida por defecto con una zona `hint`.
  - `master` — Designa el servidor de nombres actual como el que tiene la autoridad para esa zona. Una zona se puede configurar como tipo `master` si los archivos de configuración de la zona residen en el sistema.
  - `slave` — Designa el servidor de nombres como un servidor esclavo para esa zona. También especifica la dirección IP del servidor de nombres maestro para la zona.
- `zone-statistics` — Configura `named` para mantener estadísticas concerniente a esta zona, escribiéndola a su ubicación por defecto (`/var/named/named.stats`) o al archivo listado en la opción `statistics-file` en la declaración `server`. Consulte la Sección 12.2.2 para más información sobre la declaración `server`.

12.2.1.5. Ejemplo de declaraciones de `zone`

La mayoría de los cambios al archivo `/etc/named.conf` de un servidor de nombres maestro o esclavo envuelven agregar, modificar o borrar declaraciones `zone`. Mientras que estas declaraciones `zone` pueden contener muchas opciones, la mayoría de los servidores de nombres requieren sólo un pequeño subconjunto para funcionar efectivamente. Las siguientes declaraciones `zone` son ejemplos muy básicos que ilustran la relación de servidores de nombres maestro-esclavo.

A continuación se muestra un ejemplo de una declaración de `zone` para un servidor de nombres primario hospedando `example.com` (`192.168.0.1`):

```
zone "example.com" IN {
  type master;
  file "example.com.zone";
  allow-update { none; };
};
```

En la declaración, la zona es identificada como `example.com`, el tipo es configurado a `master` y el servicio `named` se instruye para leer el archivo `/var/named/example.com.zone`. También le dice a `named` que no permita a ningún otro host que realice actualizaciones.

Una declaración `zone` de servidor esclavo para `example.com` se ve un poco diferente comparado con el ejemplo anterior. Para un servidor esclavo, el tipo se coloca a `slave` y en lugar de la línea `allow-update` está una directiva diciéndole a `named` la dirección IP del servidor maestro.

A continuación se muestra un ejemplo de una declaración `zone` para un servidor esclavo para la zona `example.com`:

```
zone "example.com" {
  type slave;
  file "example.com.zone";
  masters { 192.168.0.1; };
};
```

Esta declaración `zone` configura `named` en el servidor esclavo a que busque por el servidor maestro en la dirección IP `192.168.0.1` por información sobre la zona `example.com`. La información que el servidor esclavo recibe desde el servidor maestro es guardada al archivo `/var/named/example.com.zone`.

12.2.2. Otros tipos de declaraciones

La lista siguiente muestra tipos de declaraciones usadas con menos frecuencia disponibles dentro de `named.conf`:

- `controls` — Configura varios requerimientos de seguridad necesarios para usar el comando `rndc` para administrar el servicio `named`.

  Consulte la Sección 12.4.1 para conocer más sobre la estructura de la declaración `controls` y de las opciones que están disponibles.

- `key "`*`<key-name>`*`"` — Define una llave particular por nombre. Las claves son usadas para autenticar varias acciones, tales como actualizaciones seguras o el uso del comando `rndc`. Se usan dos opciones con `key`:

  - `algorithm` *`<algorithm-name>`* — El tipo de algoritma usado, tal como `dsa` o `hmac-md5`.
  - `secret "`*`<key-value>`*`"` — La clave encriptada.

  Consulte la Sección 12.4.2 para instrucciones sobre cómo escribir una declaración `key`.

- `logging` — Permite el uso de múltiples tipos de registro, llamados *channels*. Usando la opción `channel` dentro de la declaración `logging`, se puede construir un tipo registro personalizado, con su propio nombre de archivo (`file`), tamaño límite (`size`), versión (`version`), y nivel de importancia (`severity`). Una vez que se haya definido el canal personalizado, se usa una opción `category` para clasificar el canal y comenzar a conectar cuando se reinicie `named`.

  Por defecto, `named` registra mensajes estándar al demonio `syslog`, que les sitúa en `/var/log/messages`. Esto se debe a que varios canales estándares se encuentran incorporados a BIND junto con varios niveles de severidad, tales como uno que maneja la información de mensajes de registros (`default_syslog`) y otro que maneja mensajes de depuración (`default_debug`). Una categoría por defecto, llamada `default`, usa los canales incorporados para hacer conexiones normales sin ninguna configuración especial.

  La personalización del proceso de conexión es un proceso con muchos detalles y que está más allá del objetivo de este capítulo. Para información sobre la creación de registros personalizados BIND, consulte el ***Manual de referencia del administrador de BIND 9*** mencionado en la Sección 12.7.1.

- `server` — Define opciones particulares que afectan como `named` debería actuar con respecto a servidores de nombres remotos, especialmente en lo que respecta a las notificaciones y transferencias de zonas.

  La opción `transfer-format` controla si un registro de recursos es enviado con cada mensaje (`one-answer`) o si registros de múltiples recursos son enviados con cada mensaje (`many-answers`). Mientras que `many-answers` es más eficiente, sólo los nuevos servidores de nombres BIND lo entienden.

- `trusted-keys` — Contiene llaves públicas utilizadas por DNS seguro, DNSSEC. Para mayor información sobre la seguridad de BIND, consulte la Sección 12.5.3.

- `view "<view-name>"` — Crea vistas especiales dependiendo de en qué red esté el host que contacta el servidor de nombres. Esto permite a determinados hosts recibir una respuesta que se refiere a una zona particular mientras que otros hosts reciben información completamente diferente. Alternativamente, algunas zonas puede que sólo estén disponibles para ciertos hosts de confianza mientras que otros hosts menos autorizados, sólo pueden hacer peticiones a otras zonas.

  Se pueden usar múltiples visualizaciones, siempre y cuando sus nombres sean únicos. La opción `match-clients` especifica las direcciones IP que aplican a una vista particular. Cualquier declaración de `options` puede también ser usada dentro de una vista, ignorando las opciones globales ya configuradas por `named`. La mayoría de las sentencias `view` contienen múltiples declaraciones `zone` que aplican a la lista `match-clients`. El orden en que las sentencias `view` son listadas es importante, pues la primera sentencia `view` que coincida con una dirección IP de cliente particular es usada.

  Consulte la Sección 12.5.2 para más información sobre la declaración `view`.

12.2.3. Etiquetas de comentarios

La siguiente es una lista de las etiquetas de comentarios válidas usadas dentro de `named.conf`:

- `//` — Cuando se coloca al comienzo de una línea, esa línea es ignorada por `named`.
- `#` — Cuando se coloca al comienzo de una línea, esa línea es ignorada por `named`.
- `/*` y `*/` — Cuando el texto se coloca entre estas etiquetas, se ignora el bloque de texto por `named`.

12.3. Archivos de zona

Los *Archivos de zona* contienen información sobre un espacio de nombres particular y son almacenados en el directorio de trabajo `named`, por defecto `/var/named/`. Cada archivo de zona es nombrado de acuerdo a la opción `file` en la declaración `zone`, usualmente en una forma que relaciona al dominio en cuestión e identifica el archivo como conteniendo datos de zona, tal como `example.com.zone`.

Cada archivo de zona contiene *directivas* y *registros de recursos*. Las directivas le dicen al servidor de nombres que realice tareas o aplique configuraciones especiales a la zona. Los registros de recursos define los parámetros de la zona y asignan identidades a hosts individuales. Las directivas son opcionales, pero los registros de recursos se requieren para proporcionar servicios de nombres a la zona.

Todas las directivas y registros de recursos deberían ir en sus propias líneas individuales.

Los comentarios se pueden colocar después de los punto y comas (`;`) en archivos de zona.

12.3.1. Directivas de archivos de zona

Las directivas comienzan con el símbolo de dollar (`$`) seguido del nombre de la directiva. Usualmente aparecen en la parte superior del archivo de zona.

Lo siguiente son directivas usadas a menudo:

- `$INCLUDE` — Dice a `named` que incluya otro archivo de zona en el archivo de zona donde se usa la directiva. Así se pueden almacenar configuraciones de zona suplementarias aparte del archivo de zona principal.
- `$ORIGIN` — Anexa el nombre del dominio a registros no cualificados, tales como aquellos con el nombre de host solamente.

  Por ejemplo, un archivo de zona puede contener la línea siguiente:

  ```
  $ORIGIN example.com.
  ```

  Cualquier nombre utilizado en registros de recursos que no terminen en un punto (`.`) tendrán `example.com` anexado.

  Nota

  El uso de la directiva `$ORIGIN` no es necesario si la zona es especificada en `/etc/named.conf` porque la zona es usada como el valor de la directiva `$ORIGIN` por defecto.

- `$TTL` — Ajusta el valor *Time to Live (TTL)* predeterminado para la zona. Este es el tiempo, en segundos, que un registro de recurso de zona es válido. Cada recurso puede contener su propio valor TTL, el cual ignora esta directiva.

  Cuando se decide aumentar este valor, permite a los servidores de nombres remotos hacer

caché a la información de zona para un período más largo de tiempo, reduciendo el número de consultas para la zona y alargando la cantidad de tiempo requerido para proliferar cambios de registros de recursos.

12.3.2. Registros de recursos de archivos de zona

El componente principal de un archivo de zona es su registro de recursos.

Hay muchos tipos de registros de recursos de archivos de zona. A continuación le mostramos los tipos de registros más frecuentes:

- `A` — Registro de dirección que especifica una dirección IP que se debe asignar a un nombre, como en el ejemplo:

```
<host>      IN      A      <IP-address>
```

  Si el valor *<host>* es omitido, el registro `A` apunta a una dirección IP por defecto para la parte superior del espacio de nombres. Este sistema es el objetivo para todas las peticiones no FQDN.

  Considere el siguiente ejemplo de registro `A` para el archivo de zona `example.com`:

```
              IN      A       10.0.1.3
server1       IN      A       10.0.1.5
```

  Las peticiones para `example.com` son apuntadas a 10.0.1.3, mientras que las solicitudes para `server1.example.com` son dirigidas a 10.0.1.5.

- `CNAME` — Registro del nombre canónico, que enlaza un nombre con otro: también conocido como un alias.

  El próximo ejemplo indica a `named` que cualquier petición enviada a *<alias-name>* apuntará al host, *<real-name>*. Los registros `CNAME` son usados normalmente para apuntar a servicios que usan un esquema de nombres común, tal como `www` para servidores Web.

```
<alias-name>      IN      CNAME       <real-name>
```

  En el ejemplo siguiente, un registro `A` vincula un nombre de host a una dirección IP, mientras que un registro `CNAME` apunta al nombre host comúnmente usado `www` para este.

```
server1       IN      A       10.0.1.5
www           IN      CNAME   server1
```

- `MX` — Registro de Mail eXchange, el cual indica dónde debería de ir el correo enviado a un espacio de nombres particular controlado por esta zona.

```
      IN      MX      <preference-value>  <email-server-name>
```

En este ejemplo, *<preference-value>* permite una clasificación numérica de los servidores de correo para un espacio de nombres, dando preferencia a algunos sistemas de correo sobre otros. El registro de recursos `MX` con el valor más bajo *<preference-value>* es preferido sobre los otros. Sin embargo, múltiples servidores de correo pueden tener el mismo valor para distribuir el tráfico de forma pareja entre ellos.

El *<email-server-name>* puede ser un nombre de servidor o FQDN.

```
        IN      MX      10      mail.example.com.
        IN      MX      20      mail2.example.com.
```

En este ejemplo, el primer servidor de correo `mail.example.com` es preferido al servidor de correo `mail2.example.com` cuando se recibe correo destinado para el dominio `example.com`.

- `NS` — Registro NameServer, el cual anuncia los nombres de servidores con autoridad para una zona particular.

  Este es un ejemplo de un registro `NS`:

  ```
        IN      NS      <nameserver-name>
  ```

  El *<nameserver-name>* debería ser un FQDN.

  Luego, dos nombres de servidores son listados como con autoridad para el dominio. No es importante si estos nombres de servidores son esclavos o si son maestros; ambos son todavía considerados con autoridad.

  ```
        IN      NS      dns1.example.com.
        IN      NS      dns2.example.com.
  ```

- `PTR` — Registro PoinTeR o puntero, diseñado para apuntar a otra parte del espacio de nombres.

  Los registros `PTR` son usados principalmente para la resolución inversa de nombres, pues ellos apuntan direcciones IP de vuelta a un nombre particular. Consulte la Sección 12.3.4 para más ejemplos de registros `PTR` en uso.

- `SOA` — Registro de recursos Start Of Authority, que declara información importante de autoridad relacionada con espacios de nombres al servidor nombres.

  Está situado detrás de las directivas, un registro `SOA` es el primer registro en un archivo de zona.

  El ejemplo siguiente muestra la estructura básica de un registro de recursos `SOA`:

  ```
  @       IN      SOA     <primary-name-server>     <hostmaster-email> (
                          <serial-number>
                          <time-to-refresh>
                          <time-to-retry>
                          <time-to-expire>
                          <minimum-TTL> )
  ```

El símbolo `@` coloca la directiva `$ORIGIN` (o el nombre de la zona, si la directiva `$ORIGIN` no está configurada) como el espacio de nombres que esta siendo definido por este registro de recursos `SOA`. El nombre del host del servidor de nombres que tiene autoridad para este dominio es la directiva *<primary-name-server>* y el correo electrónico de la persona a contactar sobre este espacio de nombres es la directiva *<hostmaster-email>*.

La directiva *<serial-number>* es un valor numérico que es incrementado cada vez que se cambia el archivo de zona para así indicar a `named` que debería recargar esta zona. La directiva *<time-to-refresh>* es el valor numérico que los servidores esclavos utilizan para determinar cuánto tiempo debe esperar antes de preguntar al servidor de nombres maestro si se han realizado cambios a la zona. El valor *<serial-number>* es usado por los servidores esclavos para determinar si esta usando datos de la zona desactualizados y si debería refrescarlos.

La directiva *<time-to-retry>* es un valor numérico usado por los servidores esclavo para determinar el intervalo de tiempo que tiene que esperar antes de emitir una petición de actualización de datos en caso de que el servidor de nombres maestro no responda. Si el servidor maestro no ha respondido a una petición de actualización de datos antes que se acabe el intervalo de tiempo *<time-to-expire>*, los servidores esclavo paran de responder como una autoridad por peticiones relacionadas a ese espacio de nombres.

La directiva *<minimum-TTL>* es la cantidad de tiempo que otros servidores de nombres guardan en caché la información de zona.

Cuando se configura BIND, todos los tiempos son siempre referenciados en segundos. Sin embargo, es posible usar abreviaciones cuando se especifiquen unidades de tiempo además de segundos, tales como minutos (`M`), horas (`H`), días (`D`) y semanas (`W`). La Tabla 12-1 le muestra la cantidad de tiempo en segundos y el tiempo equivalente en otro formato.

Segundos	Otras unidades de tiempo
60	1M
1800	30M
3600	1H
10800	3H
21600	6H
43200	12H
86400	1D
259200	3D
604800	1W
31536000	365D

Tabla 12-1. Segundos comparados a otras unidades de tiempo

El ejemplo siguiente ilustra la forma que un registro de recursos `SOA` puede tomar cuando es configurado con valores reales.

```
@     IN     SOA    dns1.example.com.     hostmaster.example.com. (
                    2001062501 ; serial
                    21600      ; refresh after 6 hours
                    3600       ; retry after 1 hour
                    604800     ; expire after 1 week
                    86400 )    ; minimum TTL of 1 day
```

12.3.3. Ejemplo de archivo de zonas

Vistos individualmente, las directivas y registros de recursos pueden ser difíciles de comprender. Sin embargo, cuando se colocan juntos en un mismo archivo, se vuelven más fáciles de entender.

El ejemplo siguiente muestra un archivo de zona muy básico.

```
$ORIGIN example.com.
$TTL 86400
@     IN     SOA    dns1.example.com.     hostmaster.example.com. (
                    2001062501 ; serial
                    21600      ; refresh after 6 hours
                    3600       ; retry after 1 hour
                    604800     ; expire after 1 week
                    86400 )    ; minimum TTL of 1 day

      IN     NS     dns1.example.com.
      IN     NS     dns2.example.com.

      IN     MX     10     mail.example.com.
      IN     MX     20     mail2.example.com.

             IN     A       10.0.1.5

server1      IN     A       10.0.1.5
server2      IN     A       10.0.1.7
dns1         IN     A       10.0.1.2
dns2         IN     A       10.0.1.3

ftp          IN     CNAME   server1
mail         IN     CNAME   server1
mail2        IN     CNAME   server2
www          IN     CNAME   server2
```

En este ejemplo, las directivas estándar y los valores `SOA` son usados. Los servidores de nombres con autoridad se configuran como `dns1.example.com` y `dns2.example.com`, que tiene archivos `A` que los juntan a `10.0.1.2` y a `10.0.1.3`, respectivamente.

Los servidores de correo configurados con los registros `MX` apuntan a `server1` y `server2` a través de registros `CNAME`. Puesto que los nombres `server1` y `server2` no terminan en un punto (`.`), el dominio `$ORIGIN` es colocado después de ellos, expandiéndolos a `server1.example.com` y a `server2.example.com`. A través de registros de recursos relacionados `A`, se puede determinar sus direcciones IP.

Los servicios FTP y Web, disponibles en los nombres estándar `ftp.example.com` y `www.example.com`, son apuntados a los servidores apropiados usando registros `CNAME`.

12.3.4. Archivos de zona de resolución de nombres inversa

Se usa un archivo de zona de resolución inversa de nombres para traducir una dirección IP en un espacio de nombres particular en un FQDN. Se vé muy similar a un archivo de zona estándar, excepto que se usan registros de recursos `PTR` para enlazar las direcciones IP a un nombre de dominio completamente cualificado.

Un registro `PTR` se vería similar a esto:

```
<last-IP-digit>      IN      PTR     <FQDN-of-system>
```

El valor *<last-IP-digit>* se refiere al último número en una dirección IP que apunta al FQDN de un sistema particular.

En el ejemplo siguiente, las direcciones IP de la `10.0.1.20` a la `10.0.1.25` apuntan a los FQDNs correspondientes.

```
$ORIGIN 1.0.10.in-addr.arpa.
$TTL 86400
@     IN     SOA    dns1.example.com.     hostmaster.example.com. (
                    2001062501 ; serial
                    21600      ; refresh after 6 hours
                    3600       ; retry after 1 hour
                    604800     ; expire after 1 week
                    86400 )    ; minimum TTL of 1 day

      IN     NS     dns1.example.com.
      IN     NS     dns2.example.com.

20    IN     PTR    alice.example.com.
21    IN     PTR    betty.example.com.
22    IN     PTR    charlie.example.com.
23    IN     PTR    doug.example.com.
24    IN     PTR    ernest.example.com.
25    IN     PTR    fanny.example.com.
```

Este archivo de zona se colocará en funcionamiento con una declaración `zone` en el archivo `named.conf` el cual se ve similar a lo siguiente:

```
zone "1.0.10.in-addr.arpa" IN {
  type master;
  file "example.com.rr.zone";
  allow-update { none; };
};
```

Hay muy poca diferencia entre este ejemplo y una declaración de `zone` estándar, excepto por el nombre de la zona. Observe que una zona de resolución de nombres inversa requiere que los primeros tres bloques de la dirección IP esten invertidos seguido por `.in-addr.arpa`. Esto permite asociar con la zona a un bloque único de números IP usados en el archivo de zona de

resolución de nombres inversa.

Anterior
/etc/named.conf
Inicio
Subir
Siguiente
Uso de rndc

12.4. Uso de `rndc`

BIND incluye una utilidad llamada `rndc` la cual permite la administración de línea de comandos del demonio `named` desde el host local o desde un host remoto.

Para prevenir el acceso no autorizado al demonio `named`, BIND utiliza un método de autenticación de llave secreta compartida para otorgar privilegios a hosts. Esto significa que una llave idéntica debe estar presente en los archivos de configuración `/etc/named.conf` y en el `rndc`, `/etc/rndc.conf`.

12.4.1. Configuración de `/etc/named.conf`

Para que `rndc` se pueda conectar a un servicio `named`, debe haber una declaración `controls` en el archivo de configuración del servidor BIND `/etc/named.conf`.

La declaración `controls` mostrada abajo en el ejemplo siguiente, permite a `rndc` conectarse desde un host local.

```
controls {
  inet 127.0.0.1 allow { localhost; } keys { <key-name>; };
};
```

Esta declaración le dice a `named` que escuche en el puerto por defecto TCP 953 de la dirección loopback y que permita comandos `rndc` provenientes del host local, si se proporciona la llave correcta. El valor *<key-name>* especifica un nombre en la declaración `key` dentro del archivo `/etc/named.conf`. El ejemplo siguiente ilustra la declaración `key`.

```
key "<key-name>" {
  algorithm hmac-md5;
  secret "<key-value>";
};
```

En este caso, el *<key-value>* utiliza el algoritmo HMAC-MD5. Utilice el comando siguiente para generar llaves usando el algoritmo HMAC-MD5:

```
dnssec-keygen -a hmac-md5 -b <bit-length> -n HOST <key-file-name>
```

Una llave con al menos un largo de 256-bit es una buena idea. La llave que debería ser colocada en el área *<key-value>* se puede encontrar en el archivo *<key-file-name>* generado por este comando.

Aviso

Debido a que `/etc/named.conf` está accesible a todo el mundo, es una buena idea colocar la declaración `key` en un archivo separado que sólo esté accesible por root y luego utilizar una declaración `include` para referenciarlo. Por ejemplo:

```
include "/etc/rndc.key";
```

12.4.2. Configuración de `/etc/rndc.conf`

La declaración `key` es la más importante en `/etc/rndc.conf`.

```
key "<key-name>" {
  algorithm hmac-md5;
  secret "<key-value>";
};
```

<key-name> y *<key-value>* deberían ser exáctamente los mismos que sus configuraciones en `/etc/named.conf`.

Para coincidir las claves especificadas en el archivo de configuración del servidor objetivo `/etc/named.conf`, agregue las líneas siguientes a `/etc/rndc.conf`.

```
options {
  default-server  localhost;
  default-key     "<key-name>";
};
```

Este directriz configura un valor de llave global por defecto. Sin embargo, el archivo de configuración `rndc` también puede usar llaves diferentes para servidores diferentes, como en el ejemplo siguiente:

```
server localhost {
  key  "<key-name>";
};
```

Atención

Asegúrese de que sólo el usuario root pueda leer y escribir al archivo `/etc/rndc.conf`.

Para más información sobre el archivo `/etc/rndc.conf`, vea la página man de `rndc.conf`.

12.4.3. Opciones de línea de comandos

Un comando `rndc` toma la forma siguiente:

```
rndc <options> <command> <command-options>
```

Cuando esté ejecutando `rndc` en una máquina local configurada de la forma correcta, los comandos siguientes están disponibles:

- `halt` — Para inmediatamente el servicio named.
- `querylog` — Registra todas las peticiones hechas a este servidor de nombres.
- `refresh` — Refresca la base de datos del servidor de nombres.
- `reload` — Recarga los archivos de zona pero mantiene todas las respuestas precedentes situadas en caché. Esto le permite realizar cambios en los archivos de zona sin perder todas las resoluciones de nombres almacenadas.

  Si los cambios sólo afectaron una zona específica, vuelva a cargar solamente esa una zona específica añadiendo el nombre de la zona después del comando `reload`.
- `stats` — Descarga las estadísticas actuales de `named` al archivo `/var/named/named.stats`.
- `stop` — Detiene al servidor salvando todas las actualizaciones dinámicas y los datos de las *Transferencias de zona incremental* (*IXFR*) antes de salir.

Ocasionalmente, puede ser necesario ignorar las configuraciones por defecto en el archivo `/etc/rndc.conf`. Estan disponibles las siguientes opciones:

- `-c <configuration-file>` — Especifica la ubicación alterna de un archivo de configuración.
- `-p <port-number>` — Especifica la utilización de un número de puerto diferente del predeterminado 953 para la conexión del comando `rndc`.
- `-s <server>` — Especifica un servidor diferente al `default-server` listado en `/etc/rndc.conf`.
- `-y <key-name>` — Le permite especificar una llave distinta de la opción `default-key` en el archivo `/etc/rndc.conf`.

Se puede encontrar información adicional sobre estas opciones en la página del manual de `rndc`.

Archivos de zona Subir Características avanzadas de BIND

12.5. Características avanzadas de BIND

La mayoría de las implementaciones BIND solamente utilizan `named` para proporcionar servicios de resolución de nombres o para actuar como una autoridad para un dominio particular o sub-dominio. Sin embargo, BIND versión 9 tiene un número de características avanzadas que permiten un servicio DNS más seguro y avanzado.

Atención

Algunas de estas propiedades avanzadas, tales como DNSSEC, TSIG e IXFR (las cuales se definen en la sección siguiente), solamente se deberían usar en los entornos de red que tengan servidores de nombres que soporten estas propiedades. Si su entorno de red incluye servidores de nombres no-BIND o versiones anteriores de BIND, verifique que cada característica avanzada sea soportada antes de intentar utilizarla.

Todas las propiedades citadas aquí se describen en detalle en el ***Manual de referencia para el administrador de BIND 9*** referenciado en la Sección 12.7.1.

12.5.1. Mejoras al protocolo DNS

BIND soporta Transferencias de zona incremental (Incremental Zone Transfers, IXFR), donde un servidor de nombres esclavo sólo descargará las porciones actualizadas de una zona modificada en un servidor de nombres maestro. El proceso de transferencia estándar requiere que la zona completa sea transferida a cada servidor de nombres esclavo hasta por el cambio más pequeño. Para dominios muy populares con archivos de zona muy largos y muchos servidores de nombres esclavos, IXFR hace que la notificación y los procesos de actualización sean menos exigentes en recursos.

Observe que IXFR solamente está disponible cuando utiliza la *actualización dinámica* para realizar los cambios en los registros de zona maestra. Si cambia los archivos de zona manualmente, tiene que usar AXFR (Automatic Zone Transfer). Encontrará más información sobre la actualización dinámica en el ***Manual de referencia para el administrador de BIND 9***. Consulte la Sección 12.7.1 para más información.

12.5.2. Vistas múltiples

A través del uso de la declaración `view` en `named.conf`, BIND puede presentar información diferente dependiendo desde cuál red se esté realizando la petición.

Esto es básicamente usado para negar entradas DNS confidenciales a clientes fuera

de la red local, mientras se permiten consultas desde clientes dentro de la red local.

La declaración `view` usa la opción `match-clients` para coincidir direcciones IP o redes completas y darles opciones especiales y datos de zona.

12.5.3. Seguridad

BIND soporta un número de métodos diferentes para proteger la actualización y zonas de transferencia, en los servidores de nombres maestro y esclavo:

- *DNSSEC* — Abreviación de *DNS SECurity*, esta propiedad permite firmar con caracteres criptográficos zonas con una *clave de zona*.

  De esta manera, puede verificar que la información de una zona provenga de un servidor de nombres que la ha firmado con caracteres criptográficos con una clave privada, siempre y cuando el recipiente tenga esa clave pública del servidor de nombres.

  BIND versión 9 también soporta el método SIG(0) de llave pública/privada de autenticación de mensajes.

- *TSIG* — Abreviación para *Transaction SIGnatures*, esta característica permite una transferencia desde el maestro al esclavo sólo después de verificar que una llave secreta compartida existe en ambos servidores maestro y en el esclavo.

  Esta característica fortalece el método estándar basado en direcciones IP de transferencia de autorización. Un intruso no solamente necesitará acceso a la dirección IP para transferir la zona, sino también necesitará conocer la clave secreta.

  BIND versión 9 también soporta *TKEY*, el cual es otro método de autorización de zonas de transferencia basado en clave secreta compartida.

12.5.4. IP versión 6

BIND versión 9 puede proporcionar servicios de nombres en ambientes IP versión 6 (IPv6) a través del uso de registros de zona `A6`.

Si el entorno de red incluye hosts IPv4 e IPv6, use el demonio ligero de resolución `lwresd` en todos los clientes de la red. Este demonio es muy eficiente, funciona solamente en caché y además entiende los nuevos registros `A6` y `DNAME` usados bajo IPv6. Consulte la página de manual para `lwresd` para más información.

Anterior
Uso de `rndc`

Inicio
Subir

Siguiente
Errores comunes que debe evitar

12.6. Errores comunes que debe evitar

Es normal que los principiantes cometan errores modificando los archivos de configuración BIND. Asegúrese de evitar los siguientes errores:

- *Asegúrese que aumenta el número de serie cuando esté modificando un archivo de zona.*

  Si no se incrementa el número de serial, el servidor de nombres maestro tendrá la información nueva correcta, pero los servidores esclavos nunca serán notificados del cambio ni intentarán actualizar sus datos de esa zona.

- *Preste atención a la utilización correcta de las llaves y de los puntos y comas en el archivo `/etc/named.conf`.*

  La omisión de un punto y coma o de una llave en una sección causará que `named` se niegue a arrancar.

- *Recuerde colocar puntos (`.`) en los archivos de zona después de todos los FQDNs y omítalos en los nombres de máquinas.*

  Un punto al final de un nombre de dominio denota un nombre de dominio completamente cualificado. Si el punto es omitido, entonces `named` añade el nombre de la zona o el valor `$ORIGIN` para completarlo.

- *Si un firewall está bloqueando las conexiones con el programa `named` a otros servidores de nombres, modifique su archivo de configuración.*

  Por defecto, la versión 9 de BIND usa los puertos aleatorios por encima de 1024 para consultar otros servidores de nombres. Algunos cortafuegos, sin embargo, esperan que todos los servidores de nombres se comuniquen usando solamente el puerto 53. Puede forzar `named` a que use el puerto 53 añadiendo la línea siguiente a la declaración `options` de `/etc/named.conf`:

```
query-source address * port 53;
```

12.7. Recursos adicionales

Las siguientes fuentes de información le proporcionarán recursos adicionales relacionados a BIND.

12.7.1. Documentación instalada

- BIND presenta un rango completo de documentación instalada cubriendo muchos tópicos diferentes, cada uno colocado en su propio directorio:
  - `/usr/share/doc/bind-<version-number>/` — Este directorio contiene una lista con las propiedades más recientes. Reemplace *<version-number>* con la versión de `bind` instalado en el sistema.
  - `/usr/share/doc/bind-<version-number>/arm/` — Este directorio contiene una versión en HTML y SGML del ***Manual de referencia para el administrador de BIND 9***, el cual detalla los requerimientos de recursos de BIND, cómo configurar diferentes tipos de servidores de nombres, balancear cargas y otros temas avanzados. Para la mayoría de los usuarios nuevos de BIND, este es el mejor lugar para comenzar. Reemplace *<version-number>* con la versión de `bind` instalado en el sistema.
  - `/usr/share/doc/bind-<version-number>/draft/` — Este directorio contiene documentos técnicos varios que revisan los problemas relacionados con el servicio DNS y algunos métodos propuestos para solucionarlos. Reemplace *<version-number>* con la versión de `bind` instalada en el sistema.
  - `/usr/share/doc/bind-<version-number>/misc/` — Directorio que contiene documentos diseñados para referenciar problemas avanzados. Los usuarios de la versión 8 de BIND deberían consultar el documento `migration` para cambios específicos que se deben hacer cuando se esté moviendo a BIND 9. El archivo `options` lista todas las opciones implementadas en BIND 9 que son usadas en el archivo `/etc/named.conf`. Reemplace *<version-number>* con la versión de `bind` instalada en el sistema.
  - `/usr/share/doc/bind-<version-number>/rfc/` — Este directorio suministra cada documento RFC relacionado a BIND. Reemplace *<version-number>* con la versión de `bind` instalada en el sistema.
- Las páginas man relacionadas a BIND — Hay un gran número de páginas man para las diferentes aplicaciones y archivos de configuración referentes a BIND. Lo siguiente es una lista de algunas de las páginas importantes.

  Aplicaciones administrativas

- La página `man rndc` — Explica las diferentes opciones disponibles cuando se utilice el comando `rndc` para controlar un servidor de nombres BIND.

Aplicaciones de servidor

- La página `man named` — Explora argumentos varios que se pueden usar para controlar el demonio de servidor de nombres BIND.
- `man lwresd` — Describe las opciones disponibles y el propósito para el demonio lightweight resolver.

Archivos de configuración

- La página `man named.conf` — Una lista completa de las opciones disponibles dentro del archivo de configuración `named`.
- La página `man rndc.conf` — Una lista completa de opciones disponibles dentro del archivo de configuración `rndc`.

12.7.2. Sitios web de utilidad

- http://www.isc.org/products/BIND/ — La página principal del proyecto BIND conteniendo información sobre los lanzamientos recientes así como también la versión PDF del ***Manual de referencia para el administrador de BIND 9***.
- http://www.redhat.com/mirrors/LDP/HOWTO/DNS-HOWTO.html — Cubre el uso de BIND como un servidor de nombres de caché y la configuración de varios archivos de zona necesarios para servir como el servidor de nombres principal para un dominio.

12.7.3. Libros relacionados

- ***Manual de administración del sistema de Red Hat Enterprise Linux*** — El capítulo ***Configuración de BIND*** explica cómo configurar un servidor DNS usando **Herramienta de configuración del Servicio de Nombres de Dominio**.
- ***DNS y BIND*** por Paul Albitz y Cricket Liu; O'Reilly & Associates — Una referencia popular que explica opciones de configuración comunes y esotéricas de BIND, así como también proporciona estrategias para asegurar su servidor DNS.
- ***The Concise Guide to DNS and BIND*** por Nicolai Langfeldt; Que — Hace referencia a la conexión entre servicios de red múltiples y BIND, haciendo énfasis en los tópicos técnicos orientados a tareas.