JUNTA DE ANDALUCIA

CONSEJERIA DE EDUCACION
CEP Lora del Río

DNS con Bind9

Alberto Molina Coballes, José Domingo Muñoz Rodríguez y José Luis Rodríguez Rodríguez.

20 de marzo de 2010

En este documento se describe de forma breva las características fundamentales del protocolo de resolución de nombres DNS y la configuración elemental de un servidor DNS con Bind9 en Debian GNU/Linux. Este documento se elaboró para el curso *Servicios en GNU/Linux. Portal Educativo*, organizado por el CEP de Lora del Río (Sevilla) en 2010.

Este trabajo es una obra derivada de la documentación del curso *Software Libre y Educacion: servicios de red, gestores de contenidos y seguridad* de José Angel Bernal, Fernando Gordillo, Hugo Santander y Paco Villegas.

Índice

1. Introducción

El sistema DNS es una base de datos distribuida. Presenta una jerarquía en la que su parte más alta es el *punto* o raíz y de él cuelgan los dominios de primer nivel (.com, .edu, .es, etc). Su lectura en el orden jerárquico se realiza de derecha a izquierda. Por ejemplo, para la

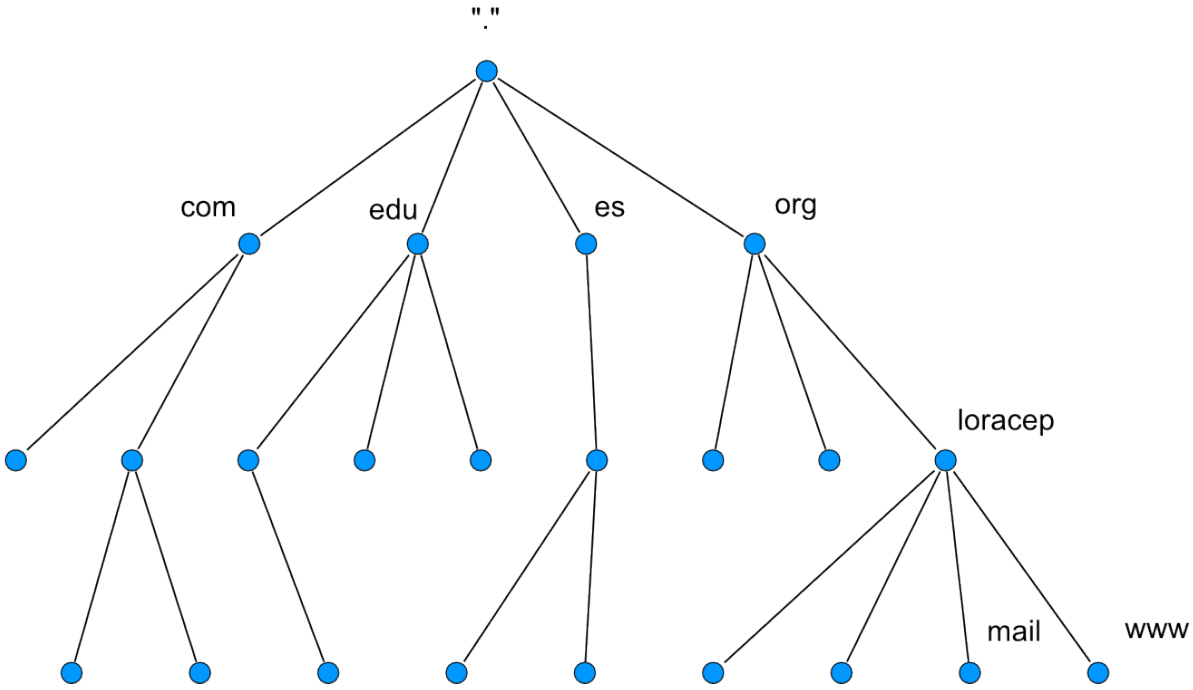

Figura 1: Distribución jerarquizada de los nombres de dominio.

máquina `www.loracep.org`, primero en la jerarquía se encuentra el dominio de primer nivel[1] (.org), luego va el subdominio o subdominios (en este caso, loracep) y por último el nombre de la máquina (www).

Los dominios genéricos de primer nivel eran inicialmente .com, .edu, .org, .gov, .mil y .net más los correspondientes a los países (.es, .it, .uk, .pt, ...). Posteriormente se ampliaron los dominios de primer nivel con .aero, .asia, .cat, etc. y hay algunas peticiones todavía no resueltas como los dominios de primer nivel .sex o .xxx.

Al principio, con pocas máquinas en Internet, bastaba para mantener este sistema con unos ficheros de nombre `HOSTS.TXT` o `/etc/hosts`, en los que se encontraban los nombres de las máquinas uno a uno. A medida que el sistema fue creciendo, se hacía necesario el soporte de un sistema más potente, que es el basado en *Servidores de Nombres*.

2. Recursos del Servidor de Nombres

Para ver qué nos ofrece un servidor de nombres utilizaremos la herramienta `dig`[2]. En su forma más simple, le preguntamos como argumento con un nombre de host para conocer la dirección que le corresponde.

```
avatar:~$ dig www.loracep.org
```

[1] En inglés, *Top Level Domain*

[2] *Domain Information Groper*. Esta herramienta sustituye a otra anterior que se llama `nslookup`.

```
; <<>> DiG 9.5.1-P3 <<>> www.loracep.org
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18420
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.loracep.org.               IN      A

;; ANSWER SECTION:
www.loracep.org.        47      IN      A       212.36.75.197

;; Query time: 19 msec
;; SERVER: 62.42.230.24#53(62.42.230.24)
;; WHEN: Sun Mar 14 23:26:54 2010
;; MSG SIZE  rcvd: 49
```

La instrucción `dig` nos será de gran ayuda para consultar a los servidores de nombres. Una llamada típica a `dig` es de la forma:

```
dig @servidor_de_nombres recurso tipo_de_recurso
```

servidor_de_nombres es el servidor de nombres al que vamos a preguntar. En caso de que no lo especifiquemos, preguntará a los servidores de nombres que estén en el fichero `/etc/resolv.conf`

recurso es el nombre o dirección del que queremos consultar información.

tipo_de_recurso es el tipo del recurso que buscamos. Si no especificamos ninguno, buscará el tipo A por defecto.

Un servidor de nombres nos ofrece varios tipos de recursos, los más importantes son:

A (*Address*) Nos da la correspondencia de dirección simbólica a dirección IP

CNAME (*canonical name*) Nos especifica un alias o apodo para una dirección simbólica

MX (*mail exchanger*) Indica la máquina o las máquinas que recibirán el correo

NS (*name server*) Indica los servidores de nombres oficiales para el dominio

PTR (*pointer*) Nos da la resolución inversa de una dirección IP a una dirección simbólica

SOA (*start of authority*) Autoridad sobre el Dominio de nombres.

Exprimamos un poco más el comando `dig`. Vamos a realizar una consulta amplia sobre el dominio *loracep.org*:

```
avatar:~$ dig -t ANY loracep.org
```

```
; <<>> DiG 9.5.1-P3 <<>> -t ANY loracep.org
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 34900
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 3

;; QUESTION SECTION:
```

```
;loracep.org.                   IN      ANY

;; ANSWER SECTION:
loracep.org.            2560    IN      SOA     ns1.cdmon.net.\
 hostmaster.loracep.org. 1204055656 10000 3600 604800 21600
loracep.org.            21600   IN      NS      ns1.cdmon.net.
loracep.org.            21600   IN      NS      ns2.cdmon.net.
loracep.org.            21600   IN      NS      ns3.cdmon.net.
loracep.org.            900     IN      MX      10 mail.loracep.org.
loracep.org.            900     IN      A       212.36.75.197

;; ADDITIONAL SECTION:
ns2.cdmon.net.          82984   IN      A       212.36.75.129
ns3.cdmon.net.          82986   IN      A       95.211.8.207
ns1.cdmon.net.          82986   IN      A       212.36.74.129

;; Query time: 77 msec
;; SERVER: 62.42.230.24#53(62.42.230.24)
;; WHEN: Sat Mar 20 12:47:43 2010
;; MSG SIZE  rcvd: 224
```

Que nos devuelve el registro SOA, `hostmaster@loracep.org`[3] que es la dirección de correo del responsable de la zona, los nombres y direcciones IP de los servidores DNS, el servidor de correo y la dirección genérica del dominio en caso de que no se especifique un FQDN en la consulta.

El registro indica que para todas las direcciones de correo electrónico del dominio loracep.org[4], como por ejemplo *yomismo@loracep.org*, deben dirigirse a los *intercambiadores de correo*[5].

Preguntemos por un registro CNAME. El registro CNAME se suele utilizar como un alias o pseudónimo de otra u otras máquinas. ¿Qué utilidad puede tener ésto? Por ejemplo, los servicios de Internet suelen prestarse en direcciones estandarizadas. Si queremos ver el Boletín Oficial del Estado y no sabemos con certeza la dirección, una de las primeras que probaremos si tenemos cierta experiencia con internet será `www.boe.es`. Nuestra máquina con el servidor web, no tiene porqué llamarse www[6] y además nos permite cambiar rápidamente a otra máquina sin demasiados problemas en nuestra red. Veamos un ejemplo de registro CNAME:

```
avatar:~$ dig -t CNAME informatica.gonzalonazareno.org
```

```
; <<>> DiG 9.5.1-P3 <<>> -t CNAME informatica.gonzalonazareno.org
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43593
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;informatica.gonzalonazareno.org. IN    CNAME

;; ANSWER SECTION:
informatica.gonzalonazareno.org. 159 IN CNAME    lavadora.gonzalonaza\
 reno.org.

;; Query time: 14 msec
;; SERVER: 62.42.230.24#53(62.42.230.24)
```

[3] En la respuesta DNS de sustituye la @ por un punto

[4] Y de sus subdominios en caso de que no tengan especificados los suyos propios.

[5] *Mail eXchanger*, de donde viene MX.

[6] Sería un nombre bastante feo

```
;; WHEN: Sat Mar 20 19:36:18 2010
;; MSG SIZE  rcvd: 72
```

La línea importante en esta consulta es la que nos dice que `informatica.gonzalonazareno.org` es un apodo (CNAME) de la máquina `lavadora.gonzalonazareno.org`. Si esa máquina se cae, una posible solución es cambiar el registro CNAME a otra máquina. El resto de usuarios (de todo el mundo) seguirán apuntando sus navegadores a la misma dirección sin enterarse del problema.

El recurso PTR es un poco más complicado. Veamos. Para que el mismo sistema funcione tanto para pedir conversiones de direcciones simbólicas a direcciones IP, como al revés, de direcciones IP a direcciones simbólicas se crea el recurso PTR y un dominio especial de nombre `in-addr.arpa`.

Para realizar una consulta tipo PTR con dig utilizamos la siguiente sintaxis:

```
avatar:~$ dig -x 194.224.52.36
```

```
; <<>> DiG 9.5.1-P3 <<>> -x 194.224.52.36
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4959
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

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

;; ANSWER SECTION:
36.52.224.194.in-addr.arpa. 172800 IN   PTR     ns1.telefonica-data.com.

;; Query time: 36 msec
;; SERVER: 62.42.230.24#53(62.42.230.24)
;; WHEN: Sat Mar 20 19:41:58 2010
;; MSG SIZE  rcvd: 81
```

Vemos que nos devuelve que la dirección IP `194.224.52.36` se corresponde con la dirección simbólica `ns1.telefonica-data.com`, pero antes da una información un poco rara. Como en las direcciones simbólicas la jerarquía va de derecha a izquierda y en las direcciones IP de izquierda a derecha, se emplea un truco. Todas las direcciones IP se colocan bajo el dominio `in-addr.arpa` y se va poniendo cada uno de los bytes de la dirección IP de derecha a izquierda. Así `194.224.52.36` queda como `36.52.224.194.in-addr.arpa`. Veamos la misma consulta de otra manera:

```
avatar:~$ dig -t PTR 36.52.224.194.in-addr.arpa
```

```
; <<>> DiG 9.5.1-P3 <<>> -t PTR 36.52.224.194.in-addr.arpa
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60496
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

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

;; ANSWER SECTION:
36.52.224.194.in-addr.arpa. 172800 IN   PTR     ns1.telefonica-data.com.

;; Query time: 42 msec
;; SERVER: 62.42.230.24#53(62.42.230.24)
```

```
;; WHEN: Sat Mar 20 19:43:52 2010
;; MSG SIZE  rcvd: 81
```

Correcto. Nos dice que estamos hablando de `ns1.telefonica-data.com` y es un registro de tipo PTR (*PoinTeR*).

3. Servidores de Nombres

Seguro que el DNS os ha deparado muchas sorpresas. Pues aún hay más. El hecho de configurar un Servidor de Nombres es muy divertido, aunque los ficheros de configuración del servidor de nombres que vamos a utilizar (bind) son bastante delicados.

El servidor de nombres por excelencia es el demonio `named`, que es parte del paquete BIND, preparado y coordinado por el *Internet Software Consortium*.

Un servidor de nombres puede estar configurado de alguna de estas formas:

master Es el *dueño* del dominio[7], en el que se hacen las modificaciones para ese dominio, responde las consultas que se le hagan y se encarga de propagarlo al resto.

slave Son servidores de nombres del dominio y así se encargan de resolver las preguntas que se les hagan. Pero cada cierto tiempo le preguntan a su servidor *master* del que dependen para actualizar su información.

caching-only Solamente constituyen un caché de datos para optimizar las respuestas. Por ejemplo, podemos montar uno de este tipo en nuetro equipo u organización para que todos los puestos clientes le pregunten a él. Sirve para optimizar las respuestas y el uso de la línea de comunicaciones, pero además simplifica la política de seguridad. Para las peticiones de resolución DNS, los clientes no pueden atravesar el cortafuegos y sí esta única máquina.

forwarding Redirige las peticiones a otros servidores de nombres. Normalmente los servidores DNS que hacen forwarding también cachean y se utilizan cuando se realizan las peticiones a otro servidor DNS intermedio (el del Proveedor de Servicios de Internet por ejemplo) en lugar de a los servidores raiz.

4. Instalación y configuración del servidor bind9

Para realizar la instalación del servidor DNS bind9 tenemos que ejecutar la siguiente instrucción:

```
avatar:~# aptitude install bind9
```

Simplemente con esto tendremos un servidor DNS caché que realiza consultas directamente a los servidores DNS raiz y puede servir para agilizar las consultas DNS en nuestra organización. Si además queremos que actúe como servidor maestro de zonas locales (en nuestro caso serán *example.com* y *2.168.192.in-addr.arpa*) son necesarios los pasos siguientes.

Los ficheros de configuración del servidor se encuentran en el directorio */etc/bind*. Para crear las las zonas locales de resolución de nombres tenemos que modificar el fichero named.conf.local (para ver el ejemplo vamos a suponer que el dominio es *example.com*, la red es la `192.168.2.0`. Incluimos las siguientes líneas en el fichero:

[7] Zona es el término empleado

/etc/bind/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 "example.com" {
type master;
file "db.example.com";
};

zone "2.168.192.in-addr.arpa" {
type master;
file "db.192.168.2";
};
```

Es decir, crearemos dos ficheros que incluirán respectivamente las entradas para la zona de resolución directa (*db.example.com*) e inversa (*db.192.168.2*). Estos ficheros se deben crear en el directorio de trabajo, que en este caso es */var/cache/bind*, y sus permisos y propietarios deben ser los siguientes:

```
-rw-rw---- 1 bind bind 313 mar 14 16:25 db.192.168.2
-rw-rw---- 1 bind bind 440 mar 14 16:25 db.example.com
```

Además de definir las zonas de resolución directa e inversa se ha añadido el fichero *zones.rfc1918* para que las consultas DNS a direcciones IP privadas (RFC 1918) se responda con localhost, salvo para el segmento `192.168.0.0/16`, por lo que el fichero zones.rfc1918 debe tener el siguiente contenido:

/etc/bind/zones.rfc1918

```
zone "10.in-addr.arpa" { type master; file "/etc/bind/db.empty"; };
zone "16.172.in-addr.arpa" { type master; file "/etc/bind/db.empty"; };
zone "17.172.in-addr.arpa" { type master; file "/etc/bind/db.empty"; };
zone "18.172.in-addr.arpa" { type master; file "/etc/bind/db.empty"; };
zone "19.172.in-addr.arpa" { type master; file "/etc/bind/db.empty"; };
zone "20.172.in-addr.arpa" { type master; file "/etc/bind/db.empty"; };
zone "21.172.in-addr.arpa" { type master; file "/etc/bind/db.empty"; };
zone "22.172.in-addr.arpa" { type master; file "/etc/bind/db.empty"; };
zone "23.172.in-addr.arpa" { type master; file "/etc/bind/db.empty"; };
zone "24.172.in-addr.arpa" { type master; file "/etc/bind/db.empty"; };
zone "25.172.in-addr.arpa" { type master; file "/etc/bind/db.empty"; };
zone "26.172.in-addr.arpa" { type master; file "/etc/bind/db.empty"; };
zone "27.172.in-addr.arpa" { type master; file "/etc/bind/db.empty"; };
zone "28.172.in-addr.arpa" { type master; file "/etc/bind/db.empty"; };
zone "29.172.in-addr.arpa" { type master; file "/etc/bind/db.empty"; };
zone "30.172.in-addr.arpa" { type master; file "/etc/bind/db.empty"; };
zone "31.172.in-addr.arpa" { type master; file "/etc/bind/db.empty"; };
//zone "168.192.in-addr.arpa" { type master; file "/etc/bind/db.empty"; };
```

Y su contenido podría ser (incluyendo sólo el propio servidor DNS de forma estática):

/var/cache/bind/db.example.com

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

```
        21600 ; minimum (6 hours)
)
        NS      avatar
avatar  A       192.168.2.1
```

/var/cache/bind/db.192.168.2

```
$ORIGIN 2.168.192.in-addr.arpa.
$TTL 86400 ; 1 day
@       IN      SOA     avatar   hostmaster (
        1 ; serial
        21600 ; refresh (6 hours)
        3600 ; retry (1 hour)
        604800 ; expire (1 week)
        21600 ; minimum (6 hours)
)
        NS      avatar.example.com.
1       PTR     avatar.example.com.
```

Para reiniciar el servidor DNS tenemos que ejecutar

```
avatar:~# /etc/init.d/bind9 restart
```

Tras reiniciar el servicio bind es conveniente mirar el fichero */var/log/syslog* para comprobar si se ha producido algún fallo. El reinicio correcto del servidor bind9 produce la siguiente salida:

```
named[2176]: starting BIND 9.5.1-P3 -u bind
...
named[2176]: zone 2.168.192.in-addr.arpa/IN: loaded serial 1
...
named[2176]: zone example.com/IN: loaded serial 1
named[2176]: zone localhost/IN: loaded serial 2
named[2176]: running
named[2176]: zone example.com/IN: sending notifies (serial 1)
```

4.1. Prueba de funcionamiento del servidor DNS

Utilizando algún cliente DNS (preferentemente dig), haremos consultas al servidor DNS local y comprobaremos si responde correctamente, por ejemplo:

```
avatar:~$ dig @127.0.0.1 avatar.example.com
```

```
; <> DiG 9.5.0-P2 <> @127.0.0.1 avatar.example.com
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29030
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;avatar.example.com.          IN        A

;; ANSWER SECTION:
avatar.example.com. 86400 IN    A       192.168.2.1

;; AUTHORITY SECTION:
example.com. 86400      IN      NS      avatar.example.com.
```

```
;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Tue Feb 23 17:27:49 2010
;; MSG SIZE  rcvd: 77
```