El sistema de nombres de dominio: Bind 9.2.4

Por Jaume Sabater, publicado el 27 de mayo de 2002.

Índice

Resumen.

Millones de hosts se encuentran conectados a Internet. ¿Cómo se consigue mantener la pista de todos ellos cuando pertenecen a tantos países, redes y grupos administrativos distintos? Dos piezas básicas de infraestructura mantienen todo eso junto: el sistema de nombres de dominio (DNS, del inglés **Domain Name System**), cuya función es saber quién es cada host, y el sistema de enrutado de Internet, que se encarga de conocer cómo están conectados. Este artículo hace referencia a la porción que supone el DNS en ese sistema.

Objetivos.

Los objetivos de un servidor de nombres de dominio (DNS, del inglés **Domain Name Service**) son dos:

1. Por una parte, traducir una dirección canónica en una dirección IP (del inglés, **Internet Protocol**). Por ejemplo, *linuxsilo.net* es, a fecha de creación del artículo, *66.79.182.201*.
2. Por otra parte, traducir una dirección IP en una o varias direcciones canónicas. Es lo que se conoce como traducción inversa.

Haciendo un símil, el primer punto equivaldría a buscar en una agenda el número de teléfono de una persona, dado su nombre y apellidos, mientras que el segundo sería el proceso inverso: dado un número de teléfono averiguar a qué persona corresponde.

La correlación entre una dirección IP y un nombre de dominio no tiene porqué ser única. Esto es debido a lo que se conocen como dominios virtuales. De hecho, es habitual que una dirección IP equivalga a varios nombres de dominio. Por ejemplo, la dirección IP *66.79.182.201* es equivalente a *linuxsilo.net*, *www.linuxsilo.net*, *ftp.linuxsilo.net*, *pop3.linuxsilo.net* y otros. Sin embargo, esto no significa que la misma máquina (o host) *66.79.182.201* esté ofreciendo todos esos servicios. Esto es posible gracias a lo que se conoce como enrutamiento (del inglés, **routing**) de paquetes, pero no viene al caso del artículo.

Introducción.

¿Qué es el DNS?

DNS es un sistema jerárquico con estructura de árbol. El inicio se escribe *"."* y se denomina *raíz*, al igual que en las estructuras de datos en árbol. Bajo la raíz se hallan los dominios de más alto nivel (TLD, del inglés, **Top Level Domain**), cuyos ejemplos más representativos son *ORG*, *COM*, *EDU* y *NET*, si bien hay muchos más. Del mismo modo que un árbol, tiene una raíz y ramas que de ella crecen. Si el lector está versado en ciencias de la computación, reconocerá el DNS como un árbol de búsqueda y será capaz de encontrar en él los nodos, nodos hoja y otros conceptos.

Cuando se busca una máquina, la consulta se ejecuta recursivamente en la jerarquía, empezando por la raíz. Si se desea encontrar la dirección IP de *ftp.akane.linuxsilo.net.*, el servidor de nombres (del inglés, **nameserver**) tiene que empezar a preguntar en algún sitio. Empieza mirando en su caché. Si conoce la

respuesta, pues la había buscado anteriormente y guardado en dicha caché, contestará directamente. Si no la sabe, entonces eliminará partes del nombre, empezando por la izquierda, comprobando si sabe algo de *akane.linuxsilo.net.*, luego de *linuxsilo.net.*, luego *net.* y, finalmente, de *"."*, del cual siempre se tiene información ya que se encuentra en uno de los ficheros de configuración en el disco duro. A continuación preguntará al servidor *"."* acerca de *ftp.akane.linuxsilo.net*. Dicho servidor *"."* no sabrá la contestación, pero ayudará a nuestro servidor en su búsqueda dándole una referencia de dónde seguir buscando. Estas referencias llevarán a nuestro servidor hasta el servidor de nombres que conoce la respuesta.

Así pues, empezando en *"."* encontramos los sucesivos servidores de nombres para cada nivel en el nombre de dominio por referencia. Por supuesto, nuestro servidor de nombres guardará toda la información obtenida a lo largo del proceso, a fin de no tener que preguntar de nuevo durante un buen rato.

En el árbol análogo, cada *"."* en el nombre es un salto a otra rama. Y cada parte entre los *"."* son los nombres de los nodos particulares en el árbol. Se trepa el árbol tomando el nombre que queremos (*ftp.akane.linuxsilo.net*) preguntando a la raíz (*"."*) o al servidor que sea padre desde la raíz hacia *ftp.akane.linuxsilo.net* acerca de los cuales tengamos información en la caché. Una vez se alcanzan los límites de la caché, se resuelve recursivamente preguntando a los servidores, persiguiendo las referencias (ramas) hacia el nombre.

Otro concepto del cual no se habla tanto, pero que no es menos importante, es el dominio *in-addr.arpa*, que también se encuentra anidado como los dominios "normales". *in-addr.arpa* nos permite hacernos con el nombre del host cuando tenemos su dirección. Merece la pena destacar aquí que las direcciones IP están escritas en orden inverso en el dominio *in-addr.arpa*. Si se tiene la dirección de una máquina tal como *192.168.0.1*, el servidor de nombres procederá del mismo modo que con el ejemplo *ftp.akane.linuxsilo.net*. Es decir, buscará los servidores *arpa.*, luego los servidores *in-addr.arpa.*, luego los *192.in-addr.arpa.*, luego los *168.192.in-addr.arpa.* y, por último, los servidores *0.168.192.in-addr.arpa.* En este último encontrará el registro buscado: *1.0.168.192.in-addr.arpa.*

¿Quién necesita el DNS?

El DNS define:

1. Un espacio de nombres jerárquico para los hosts y las direcciones IP.
2. Una tabla de hosts implementada como una base de datos distribuida.
3. Un traductor (del inglés, **resolver**) o librería de rutinas que permite realizar consultas a esa base de datos.
4. Enrutamiento mejorado para el correo electrónico.
5. Un mecanismo para encontrar los servicios en una red.
6. Un protocolo para intercambiar información de nombres.

Para ser auténticos ciudadanos de Internet, los sitios necesitan el DNS. Mantener un fichero local */etc/hosts* con un mapeado de todos los hosts que los usuarios puedan querer contactar no es factible.

Cada sitio mantiene una o varias piezas de la base de datos distribuida que posibilita el servicio global del sistema DNS. Su pieza de la base de datos consiste en dos o más ficheros de texto que contienen registros para cada uno de los hosts. Cada registro es una sencilla línea consistente en un nombre (normalmente el nombre de un host), un tipo de registro y diversos valores o datos.

El DNS es un sistema cliente/servidor. Los servidores (de nombres) cargan los datos de sus ficheros de DNS en memoria y los usan para responder las consultas tanto de los clientes de la red interna como de los clientes y otros servidores en la red Internet. Todos sus hosts deberían ser clientes del DNS, pero relativamente pocos necesitan ser servidores de DNS.

Si su organización es pequeña (unos pocos hosts en una única red), puede ejecutar un servidor en uno de sus equipos o pedirle a su ISP (del inglés, **Internet Services Provider**) que le proporcione ese servicio en su nombre. Un sitio de tamaño medio con diversas subredes debería tener múltiples servidores de DNS para reducir la latencia de las consultas y mejorar la productividad. Un sistema muy grande puede dividir sus dominios de DNS en subdominios y usar algunos servidores para cada subdominio.

Requerimientos y datos técnicos.

En este artículo se aprenderá a instalar y configurar BIND (del inglés, **Berkeley Internet Name Domain**) sobre un sistema Linux. Se darán por supuestos ciertos conocimientos mínimos de redes TCP/IP (del inglés, **Transmission Control Protocol / Internet Protocol**) y de administración Linux o, al menos, un conocimeinto básico del funcionamiento de un sistema de este tipo. Estos son los puntos que se tratarán y el software y hardware usado.

Software:

1. Debian GNU/Linux Sarge
2. Bind 9.2.4
3. DNS Utils
4. Bind Docs

Servicios:

1. Traducción de nombres a direcciones IP.
2. Traducción inversa (de direcciones IP a nombres).
3. Listas de control de acceso.
4. Servidores secundarios.
5. Transferencia segura de zonas entre servidores primarios y secundarios (y puertos).
6. Localización de servicios (registros SRV - RFC2052, del inglés, **Request For Comments**).
7. Respuestas parametrizadas en función del origen de la petición (vistas).
8. Uso de la herramienta *rndc*.
9. Logs a medida.

Para este artículo se usará el FQDN (del inglés, **Fully Qualified Domain Name**) *linuxsilo.net* y los servidores de nombres ns1.linuxsilo.net y ns2.linuxsilo.net. Un FQDN está formado por un host y un nombre de dominio, incluyendo el dominio de más alto nivel. Por ejemplo, *www.linuxsilo.net* es un FQDN. *www* es el host, *linuxsilo* es el dominio de segundo nivel y *net* es el dominio de más alto nivel. Un FQDN siempre empieza con el nombre del host y continua subiendo directo al dominio de más alto nivel, por lo que *ftp.akane.linuxsilo.net* es también un FQDN. *akane* no es un FQDN.

Instalación.

De este tipo de software siempre es más que recomendable tener la última versión, pues podemos hallar en ella importantes errores y fallos de seguridad corregidos, así como nuevas funcionalidades que faciliten nuestra tarea como administradores de sistemas. El proceso de instalación en una distribución Debian de Linux es tan sencillo como ejecutar (como *root*, por supuesto, del mismo modo que en el resto del artículo):

```
apt-get install bind9 bind9-doc dnsutils
```

Dependiendo de la versión de Debian, el paquete Bind9 no estará disponible (por ejemplo, en *Potato* se encuentra la versión 8), por lo que deberemos actualizar a una versión más actual (*Woody* o *Sid* en el momento de escribir este artículo) y proceder. La instalación nos deja un Bind con una configuración básica (en */etc/bind/*) y

funcionando, por lo que tan sólo deberemos configurarlo según nuestras necesidades. Empezaremos por la traducción de nombres a direcciones IP.

El paquete Debian *bind9* se instala con una configuración ya funcional para la inmensa mayoría de los servidores terminales sin que sea necesaria la acción del usuario.

El fichero de configuración *named.conf* del demonio (del inglés, **daemon**) *named* (nombre en el sistema del demonio del servidor de nombres de dominio Bind) se encuentra en */etc/bind*, de modo que todos los ficheros estáticos de configuración relacionados con Bind estén en el mismo lugar. Se recomienda encarecidamente no modificar esta configuración, más en un sistema GNU/Debian Linux. De todos modos, si es necesario hacerlo, posiblemente la mejor manera sea usando un enlace simbólico a la localización que desee usarse.

Los ficheros de datos de las zonas para los servidores raíz y las zonas de traducción de direcciones (del inglés, **forward**) y de traducción inversa (del inglés, **reverse**) para el host local (del inglés, **localhost**) se encuentran también en */etc/bind*. El directorio de trabajo (del inglés, **working directory**) de *named* es */var/cache/bind*. Por lo tanto, cualesquiera ficheros temporales generados por named, como los ficheros de la base de datos de las zonas que son secundarias para el demonio, serán escritos en el sistema de ficheros de */var*, que es a donde pertenecen. Para conseguir que esto funcione, el *named.conf* proporcionado con la instalación usa explícitamente rutas absolutas (del inglés, **fully-qualified** o **absolute pathnames**) para referenciar los ficheros en */etc/bind*.

A diferencia de anteriores paquetes Debian de Bind, los ficheros *named.conf* y todos los *db.** de la instalación se consideran ficheros de configuración. Por ello, si tan sólo se requiere una configuración "de caché" para un servidor que no ha de ser el autorizado (del inglés, **authoritative**) de ningún dominio, se puede ejecutar la configuración proporcionada tal cual. Si es necesario cambiar opciones en el *named.conf*, o incluso referentes al *init.d*, puede hacerse sin compromiso, pues las futuras actualizaciones respetarán dichos cambios, siguiendo la política de paquetes de Debian.

Si bien el lector es libre para idear la estructura que más le plazca para los servidores de los cuales necesita ser el autorizado, se sugiere que todos los ficheros *db* para las zonas de las cuales se es servidor primario (del inglés, **master**) estén en */etc/bind* (quizás incluso en una estructura de subdirectorios, dependiendo de la complejidad), y usar rutas absolutas en el fichero *named.conf*. Cualesquiera zonas de las que se es servidor secundario (del inglés, **secondary**) deberían configurarse en el *named.conf* como nombres de fichero sin ruta, de forma que los ficheros de datos terminen creándose en */var/cache/bind*. A lo largo del artículo se ilustrará este concepto para una mejor comprensión.

Traducción de nombres a direcciones IP.

El primer paso es editar el fichero */etc/bind/named.conf.options*, donde cambiaremos algunos de los valores por defecto y añadiremos todo lo necesario para que nuestro dominio sea accesible desde el exterior.

A menos que seamos un proveedor de servicios de internet, se nos habrán proporcionado una o más direcciones IP de servidores de nombres estables, que seguramente querremos usar como redireccionadores (del inglés, **forwarders**), si bien no es imprescindible para conseguir los objetivos básicos de este artículo. Para ello deberemos descomentar el bloque casi al principio del fichero:

```
// forwarders {
//   0.0.0.0;
// };
```

Y dejarlo en algo como esto:

```
forwarders {
  66.79.160.3;
};
```

Donde las IPs son las correspondientes a nuestro ISP. Esta directiva le indica a nuestro servidor que pase a otro servidor de nombres todas las peticiones para las cuales no es el autorizado o no tiene la respuesta en caché. En el caso de no especificarlos, se usarán los servidores raíz de DNS. Otras opciones interesantes de Bind (dentro de la directiva *options* y finalizadas en punto y coma) son:

1. `pid-file "/var/run/named.pid";`, que definiría la localización del fichero que contiene el PID (del inglés, **Process IDentificator**) del demonio *named*,
2. `stacksize 30M;`, que determinaría un tamaño de pila de treinta megabytes,
3. `datasize 20M;`, que especificaría un tamaño máximo de memoria dedicado a almacenar datos de veinte megabytes,
4. `transfer-format many-servers;`, que provocaría la transferencia en paralelo de varias zonas a los servidores secundarios, acelerando el proceso,
5. `allow-transfer { slaves; };`, que acotaría globalmente las transferencias de zonas a los servidores secundarios en la lista *slaves* (ver más abajo el uso de listas de control de acceso),
6. y `version "DNS server";`, que ocultaría la versión de Bind que se está ejecutando, en aras a una mayor seguridad del sistema.

Acto seguido se procederá a dar de alta las zonas para nuestros dominios. Si abrimos con un editor de textos el */etc/bind/named.conf* que viene por defecto con la instalación encontramos cinco zonas:

- La raíz (el punto)
- localhost
- 127.in-addr.arpa
- 0.in-addr.arpa
- 255.in-addr.arpa

Mediante la primera damos a conocer los servidores raíz a nuestro servidor de DNS, mientras que con las otras cuatro nos hacemos cargo de la traducción normal e inversa del localhost. A partir de aquí, abrimos el fichero */etc/bind/named.conf.local* y en él creamos la zona de nuestro dominio:

```
zone "linuxsilo.net" {
  type master;
  file "/etc/bind/db.linuxsilo.net";
  allow-query { any; };
  allow-transfer { slaves; };
};
```

El orden de las zonas es completamente irrelevante, pero se recomienda dejarlas en orden alfabético para una más fácil localización en el futuro. Nótese que el nombre de la zona no termina en *"."* (punto). Este es el cometido de los parámetros de cada zona:

1. `type master;` significa que el servidor de dominios es primario o maestro de la zona. Más adelante, al configurar servidores secundarios, se usará *type slave;*.

2. `file "/etc/bind/db.linuxsilo.net";` es el fichero donde especificaremos la configuración de esa zona. Nótese que se usa una ruta absoluta, siguiendo la política de directorios de Debian. El contenido de este fichero se especificará en breve.

3. `allow-query { any; };` significa que se permiten consultas (del inglés, **queries**) externas a la zona. Esto es algo útil y necesario, a menos que se quiera ser muy paranoico con la seguridad. Simplemente se ofrece de forma técnicamente ordenada la información que es públicamente accesible.

4. `allow-transfer { slaves; };` posibilita la transferencia automática de esta configuración a los servidores secundarios de las zonas bajo nuestro control que se especifiquen en la lista *slaves*. Se profundizará más en el punto de transferencia de zonas.

Seguramente, el lector se habrá percatado ya de que se han usado dos palabras especiales, `any` y `slaves`, que requieren una mención especial. Efectivamente, además de hacer notar la sintaxis similar a la del lenguaje de programación C, con la que se debe ser extremamente cuidadoso, hay dos comentarios extras que hacer:

1. `any` es una palabra reservada de la sintaxis de bind que significa "cualquier dirección IP", como era lógico. Su uso es muy común y necesario. Otras palabras reservadas importantes son `none`, que significa "ningún host", `localhost`, que significa el host local desde cualquiera de las interfaces del sistema, y `localnets`, que representa a todos los hosts de las redes para las cuales el sistema tiene una interfaz.

2. `slaves`, en cambio, no es ninguna palabra reservada de bind, sino que corresponde al concepto de lista de control de acceso (ACL, del inglés, **Access Control List**). Estas listas de direcciones IP nos ahorran trabajo pues, de este modo, tan sólo tenemos que especificarlas una vez y, dado que les asginamos un identificador de grupo, podemos referenciarlas de forma más simple y rápida. Este es el código de la ACL usada en el ejemplo que, por supuesto, debe especificarse en algún lugar del documento antes de ser usada:

```
acl "slaves" {
  213.96.79.79;
};
```

El lector se habrá dado cuenta en seguida de las grandes ventajas de usar estas listas, bien sea porque la lista se use en varias zonas, bien porque tengamos más de un servidor esclavo. Nótese que en los identificadores de las ACL se diferencian mayúsculas y minúsculas (en inglés, **case sensitive**).

A continuación se detalla el contenido del fichero de datos de la zona *linuxsilo.net*:

```
;
; BIND data file for zone linuxsilo.net
;
$TTL 604800
@ IN SOA linuxsilo.net. hostmaster.linuxsilo.net. (
  2005052401   ; Serial yyyy/mm/dd/id
       10800   ; Refresh (3 hours)
        7200   ; Retry (2 hours)
     1296000   ; Expire (15 days)
      172800 ) ; Negative Cache TTL (2 days)

@ IN NS    ns1.linuxsilo.net.
@ IN NS    ns2.linuxsilo.net.
@ IN MX    20 mx1.linuxsilo.net.
@ IN MX    30 mx2.linuxsilo.net.
@ IN TXT   "Linux Silo Dot Net"
@ IN HINFO "Intel Pentium IV" "Debian Linux"
@ IN LOC   39 34 58 N 2 38 2 E 100m 10000m 20m 100m

@       IN A 66.79.182.201
ns1     IN A 66.79.182.201
ns2     IN A 213.96.79.79
mx1     IN A 66.79.182.201
mx2     IN A 213.96.79.79
www     IN A 66.79.182.201
www2    IN A 66.79.182.201
webmail IN A 66.79.182.201

ssh.tcp SRV 0 0 22 linuxsilo.net.
smtp.tcp SRV 0 0 25 mx1.linuxsilo.net.
http.tcp SRV 0 3 80 linuxsilo.net.
http.tcp SRV 0 1 80 www2.linuxsilo.net.
https.tcp SRV 1 0 443 linuxsilo.net.
pop3s.tcp SRV 0 0 995 mx1.linuxsilo.net.

*.tcp SRV 0 0 0 .
*.udp SRV 0 0 0 .
```

Se comentan acto seguido todas y cada una de las directivas y opciones de estos ficheros de configuración (un punto y coma, *";"*, indica que todo lo que hay a su derecha es un comentario):

1. `$TTL 604800`: directiva obligatoria a partir de la versión 9 de Bind (RFC1035 y RFC2308), indica el tiempo de vida (TTL, del inglés, **Time To Live**) de la información contenida en el fichero. Es decir, el tiempo máximo de validez, tras el cual deberá refrescarse o actualizarse (para comprobar que no haya cambiado). Es lo que se conoce como caché positiva/negativa (del inglés, **positive/negative caching**), como se especifica en el RFC2308. Por defecto se usan segundos (604800 segundos equivale a siete días exactos), pero pueden usarse también semanas (`$TTL 1w`), días (`$TTL 7d`), horas (`$TTL 168h`) y minutos (`$TTL 10080m`). Estas abreviaturas se usan asimismo en el registro SOA, que se explica a continuación.

   Otra directiva interesante, aunque no se use en los ejemplos, es `$INCLUDE <zone-file>`, que hace que *named* incluya otro fichero de zona en el lugar donde la directiva se usa. Esto permite almacenar parámetros de configuración comunes a varias subzonas en un lugar separado del fichero de la zona principal.

2. `@ IN SOA linuxsilo.net. hostmaster.linuxsilo.net.`: el registro SOA (del inglés, **Start Of Authority**) se encuentra siempre tras las directivas y proclama información relevante sobre la autoridad de un dominio al servidor de nombres. Es siempre el primer recurso en un fichero de zona. El símbolo *"@"*

(arroba) equivale a la directiva `$ORIGIN` (o el nombre de la zona si dicha directiva no se ha usado - caso más frecuente) como espacio de nombres de dominio definido por este registro. Este sería el esqueleto de este registro:

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

El servidor de nombres primario que es el autorizado de este dominio se usa en `<primary-name-server>` y el correo electrónico de la persona a contactar acerca de este espacio de nombres (del inglés, **namespace**) se sustituye en `<hostmaster-email>` (nótese que no tiene porqué corresponder con una dirección del propio dominio).

El campo `<serial-number>` es un número que se incrementa cada vez que se modifica un fichero de una zona, de forma que Bind se dé cuenta de que tiene que recargar esta zona. Se recomienda usar la fecha de modificación en formato *AAAAMMDD*, donde *AAAA* es el año en formato de cuatro cifras, *MM* es el mes en dos cifras, y *DD* es el día de mes en dos cifras, seguido de un número de dos cifras, empezando por el 01. De este modo se podrán realizar hasta cien cambios por día. El campo `<time-to-refresh>` le dice a los servidores secundarios (esclavos) cuánto tiempo deben esperar antes de preguntar a su servidor principal (maestro) si se ha hecho algún cambio en la zona. El valor del campo `<serial-number>` es usado por los esclavos para determinar si se está usando información anticuada que deba actualizarse.

El campo `<time-to-retry>` especifica a los servidores esclavos el intervalo de tiempo a esperar antes de solicitar una actualización en el caso de que el servidor de nombres principal no esté respondiendo. Si el servidor maestro no ha respondido a la petición de actualización antes de que expire el tiempo del campo `<time-to-expire>`, el esclavo dejará de actuar como servidor el autorizado de ese espacio de nombres (zona). El campo `<minimum-TTL>` solicita a otros servidores de dominio que almacenen en su caché la información de esta zona durante al menos la cantidad de tiempo en él especificada.

Nótese que el campo `<primary-name-server>` termina en un punto, que es obligatorio poner, y que representa, según lo explicado en el apartado introductorio del artículo, el servidor de nombres *raíz*. Asimismo, este punto aparecerá en todas las referencias explícitas al dominio a lo largo del fichero. Cuando se configura un host o subdominio, por ejemplo *ftp*, se hace una referencia implícita y Bind añade automáticamente el dominio, que saca de la *"@"* del registro SOA. En cualquier caso, es posible usar referencias implícitas o explícitas indistintamente.

3. `NS ns1.linuxsilo.net.` y `NS ns2.linuxsilo.net.`: indican los servidores de nombre que tienen autoridad sobre el dominio. Nótese que la arroba nos ahorra tener que escribir el nombre del dominio completo. De hecho, el prefijo, *IN* también es prescindible. Esta omisión es posible gracias a que Bind toma las características omitidas del registro SOA anterior, es decir, *@ IN*. Desde luego, ambas formas son correctas.

4. `MX 20 ns1.linuxsilo.net.`: se trata de un registro MX (del inglés, **Mail eXchanger**) e indica dónde mandar el correo destinado a un espacio de nombres controlado por esta zona. El dígito que sigue a la palabra *MX* representa la prioridad respecto a otros registros MX para la zona, que se especificarían en posteriores líneas (`MX 30 ns2.linuxsilo.net.`), siguiendo el mismo formato pero variando dicho dígito (incrementándolo a medida que pierdan prioridad frente a anteriores registros). Es decir, cuanto más bajo es el valor de preferencia, mayor prioridad adquiere.

5. `TXT "LinuxSilo.net DNS server"`: este es un registro a descriptivo, en texto plano (del inglés, **plain text**), del servidor. Puede usarse libre y arbitrariamente para propósitos diversos. Aparecerá como resultado de una consulta sobre este tipo de registro hecha al servidor de nombres sobre esta zona.

6. `HINFO "Intel Pentium IV" "Debian Linux"`: otro registro, también a título informativo y totalmente opcional (del inglés, **Host INFOrmation**), cuyo propósito es informar sobre el hardware y el sistema operativo, en este orden, delimitados por dobles comillas y separados por un espacio o tabulador, de la máquina sobre la cual el servidor de nombres se ejecuta. Tanto este tipo de registro (HINFO) como el anterior (TXT) pueden usarse en cada uno de los subdominios (no únicamente en el dominio principal de la zona), como se verá más abajo.

7. `LOC 39 34 58 N 2 38 2 E 100m 10000m 20m 100m`: registro de localización geográfica del servidor, de nuevo opcional, que es usado por las herramientas de representación gráfica de localizaciones de servidores, por ejemplo las de la asociación CAIDA (del inglés, **Cooperative Association for Internet Data Analysis**) y otras. Puede encontrarse información sobre este tipo de registro en el RFC1876. Las coordenadas (latitud, longitud y diámetro del objeto) se encuentran en formato WGS-84 (del inglés, **World Geodetic System**, del año 1984). La localización usada en el artículo corresponde a Palma, Mallorca, Islas Baleares, España.

   El formato a seguir es el siguiente: `<owner><TTL><class> LOC ( d1 [m1 [s1]] {"N"|"S"} d2 [m2 [s2]] {"E"|"W"} alt["m"] [siz["m"] [hp["m"] [vp["m"]]]] )`. Donde:

Parámetro	Significado	Unidad	Valores	Comentario
d1	Latitud (grados)	º	0..90	Porción en grados de la latitud
m1	Latitud (minutos)	'	0..59	Porción en minutos de la latitud. Si se omite se toma por defecto 0'
s1	Latitud (segundos)	"	0..59,999	Porción en segundos de la latitud. Si se omite se toma por defecto 0"
N/S	Latitud (hemisferio)		N/S	Hemisferio terrestre norte/sur
d2	Longitud (grados)	º	0..180	Porción en grados de la longitud
m2	Longitud (minutos)	'	0..59	Porción en minutos de la longitud. Si se omite se toma por defecto 0'
s2	Longitud (segundos)	"	0..59,999	Porción en segundos de la longitud. Si se omite se toma por defecto 0"
E/W	Longitud		E/W	Longitud E=este/W=oeste
alt	Altitud	m	-100000.00 .. 42849672,95	Altitud con precisión de 0.01 m.
siz	Tamaño	m	0..90000000,00	Diámetro de la esfera que contiene el punto indicado. Si se omite se toma por defecto 1 m.
hp	Precisión horizontal	m	0..90000000,00	Precisión horizontal en metros. Si se omite se toma por defecto 10.000 m.

vp	recisión vertical	m	0..90000000,00	Precisión vertical en metros. Si se omite se toma por defecto 10 m.

8. `localhost A 127.0.0.1`: registro que relaciona el host local con su IP de loopback.

9. `linuxsilo.net. A 66.79.182.201`: registro que relaciona el nombre de dominio de segundo nivel (el "principal" de la zona) con la IP donde está hospedado. Este es el registro más usado, pues cualquier petición a *linuxsilo.net* será resuelta mediante este registro, se use el protocolo de comunicaciones que se use (por ejemplo, *http://linuxsilo.net*).

10. `ns1 A 66.79.182.201`: a partir de aquí empieza la traducción de subdominios del dominio para el cual somos el autorizado: los dominios de tercer nivel y sucesivos. Fíjese el lector en que debe crearse un registro para cada uno, sin posibilidad de "agrupar" de ningún modo. Asimismo, nótese que, al ser subdominios de la zona, se ha omitido el sufijo *linuxsilo.net.*, que se encuentra implícito debido a que no termina en *"."* (punto). Es simplemente una cuestión de claridad y ahorro de espacio, pues las representaciones en ambas zonas son - repetimos de nuevo - igualmente correctas. Otros registros similares se citan, agrupados, a continuación:

```
ns2  A     213.96.79.79
     TXT   "LinuxSilo.net secondary nameserver"
     HINFO "Intel Pentium MMX" "Debian Linux"
www  A     66.79.182.201
pop3 A     66.79.182.201
smtp A     66.79.182.201
ftp  A     66.79.182.201
ts   A     213.96.79.79
     TXT   "LinuxSilo.net Team Speak server"
     HINFO "Intel Pentium MMX" "Debian Linux"
```

Dese cuenta el lector de que se han usado dos direcciones IP distintas, lo que indicaría a priori que, en realidad, todos estos hosts (dominios de tercer nivel) se encuentran tan sólo en dos máquinas distintas. Pero esto no tiene porqué ser cierto, pues podría tenerse una misma IP pública pero varias máquinas sirviendo los distintos puertos usados en estos servicios, gracias a la acción de un router.

A propósito del concepto de *alias* (`www`, `pop3`, `smtp` y `ftp` son de hecho el mismo host) existe una controvertida discusión sobre si es mejor usar el tipo de registro *CNAME* (del inglés, **Canonical NAME**) o *IN A*. Muchos gurús de Bind recomiendan no usar registros *CNAME* en absoluto, si bien esa discusión se escapa de los objetivos de este artículo. En cualquier caso, es muy recomendable seguir la regla de que los registros *MX*, *CNAME* y *SOA* nunca deben referenciar un registro *CNAME*, sino exclusivamente algo con un registro tipo *"A"*. Por lo tanto, no es aconsejable usar:

```
web CNAME www
```

Pero sí sería correcto:

```
web CNAME ns
```

También es seguro asumir que un *CNAME* no es un host adecuado para una dirección de correo electrónico: *webmaster@www.linuxsilo.net*, sería incorrecta dada la configuración de arriba. La manera de evitar esto es usar registros *"A"* (y quizás algunos otros también, como el registro *MX*) en su lugar. El autor de este artículo se decanta por el uso de *IN A* y recomienda dicha práctica.

Traducción inversa.

En estos momentos, los programas son ya capaces de convertir los nombres en *linuxsilo.net* y *balearikus-party.org* a direcciones a las cuales pueden conectarse. Pero también se requiere una zona inversa, capaz de permitir al DNS convertir una dirección en un nombre. Este nombre es usado por muchos servidores de diferentes clases (FTP, IRC, WWW y otros) para decidir si quieren "hablar" con el cliente o no y, si es el caso, quizás incluso cuánta prioridad se le debe asignar. Para poder tener acceso completo a todos estos servicios en Internet es necesario una zona inversa.

En el fichero */etc/bind/named.conf* hallamos varias zonas inversas que vienen por defecto con la instalación, justo debajo de dos líneas de comentario como estas:

```
// be authoritative for the localhost forward and reverse zones, and for
// broadcast zones as per RFC 1912
```

Ahí podemos encontrar la traducción de zonas inversas para `localhost`, `127.in-addr.arpa`, `0.in-addr.arpa` y `255.in-addr.arpa`, que no es necesario modificar para nada excepto en el primer caso. Tras ellas deberemos añadir nuestra zona: `38.127.217.in-addr.arpa` (recuérdese que se escriben en orden inverso, como se explica en el apartado introductorio de este artículo):

```
zone "38.127.217.in-addr.arpa" {
  type master;
  file "/etc/bind/db.217.127.38";
};
```

La sintaxis es idéntica a la utilizada en las zonas de traducción de nombres explicadas en el punto anterior, y los comentarios anteriores mantienen su validez aquí. Pasemos a ver el contenido del fichero */etc/bind/db.217.127.38*:

```
;
; BIND reverse data file for zone 217.127.38
;
$TTL 604800
@ IN SOA linuxsilo.net. hostmaster.linuxsilo.net. (
  2001081501   ; Serial
       10800   ; Refresh (3 hours)
        7200   ; Retry (2 hours)
     1296000   ; Expire (15 days)
      172800 ) ; Negative Cache TTL (2 days)

@   IN NS  ns1.linuxsilo.net.
       NS  ns2.linuxsilo.net.
156 IN PTR ns1.linuxsilo.net.
```

Este sería el aspecto de la zona inversa `localhost`, que deberemos modificar ligeramente a partir del original:

```
;
; BIND data file for local loopback interface
;
$TTL 604800
@ IN SOA localhost. hostmaster.linuxsilo.net. (
  2001061501   ; Serial
      604800   ; Refresh
       86400   ; Retry
     2419200   ; Expire
      604800 ) ; Negative Cache TTL

  IN NS  ns1.linuxsilo.net.
1 IN PTR localhost.ns1.linuxsilo.net.
```

El mapeado inverso de la dirección del host local (127.0.0.1) nunca cambia, por lo que los tiempos entre cambios son largos. Nótese el número de serie, que codifica la fecha: el fichero fue cambiado por última vez durante el verano del 2001, fecha en que el servidor fue creado. Nótese asimismo que sólo el servidor maestro se lista en el dominio *localhost*. El valor de *"@"* aquí es *0.0.127.in-addr.arpa.*

De nuevo, los conceptos son los mismos (la *"@"* - arroba - indica el dominio de la zona *linuxsilo.net.*, el *"."* - punto - del final hace referencia al servidor de nombres raíz y el registro *"SOA"* tiene exactamente la misma estructura y funcionalidad), excepto las dos últimas líneas:

1. `@ IN NS ns1.linuxsilo.net.` y `NS ns2.linuxsilo.net.`: indican a qué servidores de nombres debe preguntarse por la traducción inversa de una dirección IP de esta zona.

2. `156 IN PTR ns1.linuxsilo.net.`: este es el registro que se usará para devolver el nombre que queremos que corresponda con la dirección IP que nos pertenece (cuidado al crear estos registros, pues debe hacerse referencia exclusivamente a direcciones IP que sean de nuestra propiedad o provocaríamos un conflicto). En este caso se indica que la dirección 156 (implícitamente se le añade el sufijo *.38.127.217.in-addr.arpa*, lo que indica que se trata de "nuestra" dirección IP *66.79.182.201*) equivale al host *ns1.linuxsilo.net*.

   Es obvio que aquí "falta información", pues la dirección IP *66.79.182.201* equivale, en realidad, a más hosts, tal y como hemos especificado en el fichero */etc/bind/db.linuxsilo.net*. Esto es cierto, pero el autor es de la opinión de que es redundante añadir líneas del estilo:

   ```
   156 IN PTR ftp.linuxsilo.net.
   156 IN PTR pop3.linuxsilo.net.
   156 IN PTR smtp.linuxsilo.net.
   156 IN PTR www.linuxsilo.net.
   [..]
   ```

   Es decir, se estima más adecuado especificar un único FQDN por IP. Por supuesto, si se poseyera un rango de direcciones IP, por ejemplo de *66.79.182.201* a *217.127.38.160*, ambos inclusive, aparecerían registros similares a los siguientes (variarían en función del caso concreto):

   ```
   156 IN PTR ns1.linuxsilo.net.
   157 IN PTR ftp.linuxsilo.net.
   158 IN PTR smtp.linuxsilo.net.
   159 IN PTR ssh.linuxsilo.net.
   160 IN PTR www.linuxsilo.net.
   ```

   Para este ejemplo, se deduce que la zona *linuxsilo.net.* está dividida en cinco máquinas distintas, una para cada uno de los servicios mencionados (NS, FTP, SMTP, SSH y WWW, respectivamente).

¿Por qué la traducción inversa no funciona? Hay una serie de "atenciones especiales" que prestar en este punto que a menudo se pasan por alto al configurar un servidor de nombres de este tipo. Se discuten a continuación dos errores comunes en las traducciones inversas:

1. **La zona inversa no ha sido delegada**. Cuando se solicita un rango de direcciones IP y un nombre de dominio a un proveedor de servicios, el nombre de dominio es generalmente delegado por norma. Una delegación es el servidor de nombres específico que permite ir saltando de un servidor de nombres a otro tal y como se explica en la sección introductoria de este artículo.

   La zona inversa también debe ser delegada. Si se obtiene la red *217.127.38* con el dominio *linuxsilo.net* a través de un proveedor, es preciso que dicho proveedor añada un registro *NS* para nuestra zona inversa así como para nuestra zona directa. Si se sigue la cadena desde *in-addr.arpa* hacia arriba hasta llegar a nuestra red, probablemente se encontrará una fractura en la cadena, muy probablemente a la altura de nuestro proveedor de servicios. Habiendo encontrado el eslabón roto, contacte con su proveedor de servicios y pídales que corrijan el error.

2. **Su subred no pertenece a una clase definida**. Este es un concepto más avanzado, pero las subredes sin clase (del inglés, **classless subnet**) son muy comunes en la actualidad y probablemente tenga una si la suya es una empresa pequeña.

   Una subred sin clase es lo que consigue que Internet siga funcionando hoy en día. Hace algunos años se discutía mucho sobre la falta de direcciones IP. Los cerebros del IETF (del inglés, **Internet Engineering Task Force**), que mantienen Internet funcionando, se exprimieron la cabeza y hallaron la solución al problema, aunque a cierto coste. El precio es que se obtiene menos que una subred de tipo "C" y algunas cosas pueden dejar de funcionar.

   La primera parte del problema es que su ISP debe entender la técnica utilizada. No todos los pequeños proveedores de servicios tienen un conocimiento práctico de su funcionamiento, por lo que quizás deba usted explicárselo y ser algo insistente (aunque asegúrese primero que lo entiende). Entonces, ellos deberán preparar una zona inversa en su servidor cuya correctitud puede ser examinada mediante la utilidad *dig* del paquete *dnsutils*.

   La segunda y última parte del problema es que usted debe entender la problemática y su solución. Si no está seguro, haga aquí una pausa y busque más información sobre ello. Sólo entonces, debería usted configurar su zona inversa para su red sin clase.

   Pero hay aún otra trampa oculta en este concepto. Los servidores de nombres de dominio antiguos no serán capaces de seguir el registro *CNAME* en la cadena de traducciones y errarán en la traducción inversa de su máquina. Esto puede terminar en la asignación de una clase de acceso incorrecta por parte de un servicio, una denegación de servicio o algo a medio camino entre ambos. Si se encuentra en este caso, la única solución es que su ISP inserte directamente su registro *PTR* directamente en su zona de red sin clase en lugar de usar registros *CNAME*.

   Algunos ISP ofrecen diversas alternativas para tratar este problema, como formularios web que le permitirán introducir su mapa de registros de traducción inversa, etc.

Servidores secundarios

Una vez se han configurado correctamente las zonas en el servidor principal (maestro), es necesario preparar al menos un servidor secundario (esclavo), que proporcionará robustez y fiabilidad. Si el servidor maestro cae los usuarios aún serán capaces de obtener información del esclavo acerca de las zonas que se representan. El servidor esclavo debería estar lo más lejos posible del maestro, debiendo ambos compartir la menor cantidad posible de las siguientes características: suministro eléctrico, red de área local (LAN, del inglés, **Local Area Network**), ISP, ciudad y país. Si todas ellas son distintas entre el maestro y el esclavo, entonces se tiene un servidor secundario realmente bueno.

Un servidor esclavo es simplemente un servidor de nombres que replica los ficheros de las zonas de un maestro. Se configuran tal que así:

```
zone "balearikus-party.org" {
  type slave;
  file "sec.balearikus-party.org";
  allow-query { any; };
  masters { 66.79.182.201; };
};

zone "linuxsilo.net" {
  type slave;
  file "sec.linuxsilo.net";
  allow-query { any; };
  masters { 66.79.182.201; };
};
```

Nótese que la estructura es la misma que para el servidor primario, cambiando únicamente algunos parámetros:

1. `type slave;`: indica que el servidor es esclavo para esta zona.

2. `file "sec.balearikus-party.org";` y `file "sec.linuxsilo.net";`: como se ha explicado en la introducción del artículo, para seguir la política de directorios de Debian, los archivos temporales de las zonas generados automáticamente por el servidor secundario deben guardarse en el directorio por defecto */var/cache/bind*, por lo que tan sólo se especifican ficheros (sin ruta, o con ruta relativa implícita, que es lo mismo). Véase el punto siguiente para más información sobre el contenido de estos ficheros.

3. `allow-query { any; };`: mismo concepto que en el servidor primario.

4. `masters { 66.79.182.201; };`: define qué servidor es maestro para esta zona (de la cual, recordemos, se es esclavo). Podría haberse usado una ACL aquí, de la misma manera que se hace en el */etc/bind/named.conf* del maestro, pero no se ha estimado oportuno pues existe un único maestro para ambas zonas. De todos modos, si el lector debe administrar una red de servidores de nombres, donde el papel de maestro y esclavo es desempeñado a la vez por el mismo host en función de la zona, sería entonces muy conveniente crear varias ACL, de forma que se facilite el mantenimiento y el control en la asignación de maestro y esclavos para cada zona.

Las demás opciones de configuración se usarían de modo idéntico al del servidor maestro, siempre y cuando las condiciones sean las mismas. Es decir, se aplican las mismas directivas (por ejemplo, *options*, en la cual incluiríamos la opción *forwarders*) y posibilidades.

Por último, se desea destacar este apartado un aspecto que no debe pasarse por alto: las zonas inversas, aunque especiales, también son zonas y deben transferirse del servidor primario a los secundarios. En este momento, el que hasta ahora era servidor primario pasa a ser además servidor secundario, pues *ns2.linuxsilo.net* es maestro de su zona inversa (*79.96.213.in-addr.arpa*), que transferirá a *ns1.linuxsilo.net*, convirtiéndolo en esclavo únicamente para esa zona. Del mismo modo, *ns1.linuxsilo.net* actuará como maestro de su zona inversa (*38.127.217.in-addr.arpa*), que transferirá a *ns2.linuxsilo.net* al igual que venía ocurriendo con las zonas *balearikus-party.org* y *linuxsilo.net*. A continuación se presentan los cambios en los ficheros de configuración. En el *named.conf* de *ns1.linuxsilo.net*:

```
zone "38.127.217.in-addr.arpa" {
  type master;
  file "/etc/bind/db.217.127.38";
  allow-transfer { slaves; };
};

zone "79.96.213.in-addr.arpa" {
  type slave;
  file "sec.db.213.96.79";
  masters { 213.96.79.79; };
};
```

Y en el *named.conf* de *ns2.linuxsilo.net*:

```
zone "79.96.213.in-addr.arpa" {
  type master;
  file "/etc/bind/db.213.96.79";
  allow-transfer { 66.79.182.201; };
};

zone "38.127.217.in-addr.arpa" {
  type slave;
  file "sec.db.217.127.38";
  masters { 66.79.182.201; };
};
```

El contenido de las zonas se mantendría exactamente igual. Tras estos cambios, en el directorio */var/cache/bind* de *ns1.linuxsilo.net* aparecería el fichero *sec.db.213.96.79* y en el mismo directorio de *ns2.linuxsilo.net* aparecería el fichero *sec.db.217.127.38*, todo gracias a la transferencia automática de zonas que pasa a verse a continuación.

Transferencia segura de zonas.

El lector se habrá dado cuenta de que no se ha comentado nada de los ficheros *sec.balearikus-party.org* y *sec.linuxsilo.net* especificados en las directivas *zone* del servidor secundario. Esto es debido a que usaremos un procedimiento que permitirá que esos ficheros se creen de forma automatizada a partir de los que creemos en el servidor primario, de forma que las tareas de mantenimiento se facilitan enormemente.

Para ello, se deberán haber utilizado, como se ha hecho en el ejemplo, las opciones `allow-transfer { slaves; };` y `masters { 66.79.182.201; };` en las zonas definidas en los ficheros */etc/bind/named.conf* de los servidores primario y secundario, respectivamente. Esto permitirá que, realizados los cambios deseados en el fichero */etc/bind/db.balearikus-party.org* o */etc/bind/db.linuxsilo.net*, incluyendo el incremento del número de serie identificativo del registro SOA, y habiéndole ordenado al servidor de nombres que recargue una, varias o todas las zonas, estos cambios se reflejen en el secundario de forma que se generen los correspondientes ficheros */var/cache/bind/sec.balearikus-party.org* y */var/cache/bind/sec.linuxsilo.net*.

Para que esta transferencia de zonas se haga de forma segura y controlada, impondremos ciertas restricciones en el */etc/bind/named.conf* y generaremos claves que nos asegurarán la privacidad en la comunicación. Estas son las líneas que añadiremos en el */etc/bind/named.conf* del servidor primario (*66.79.182.201* en el ejemplo):

```
controls {
  inet 127.0.0.1 allow {
    127.0.0.1;
  }
  keys {
    "2002052101.linuxsilo.net.tsigkey.";
  };
};

server 213.96.79.79 {
  keys {
    "2002052101.linuxsilo.net.tsigkey.";
  };
};
```

Y estas las que añadiremos en el */etc/bind/named.conf* del secundario:

```
controls {
  inet 127.0.0.1 allow {
    127.0.0.1;
  }
  keys {
    "2002052101.linuxsilo.net.tsigkey.";
  };
};

server 66.79.182.201 {
  keys {
    "2002052101.linuxsilo.net.tsigkey.";
  };
};
```

Acto seguido se explica el significado de ambas directivas:

1. `controls { inet 127.0.0.1 allow { 127.0.0.1; } keys { "2002052101.linuxsilo.net.tsigkey."; } };` es la directiva que ciñe el control sobre el servidor a través de la clave `2002052101.linuxsilo.net.tsigkey.` únicamente al host local. Es decir, deberemos habernos conectado (habitualmente de forma remota mediante SSH) al servidor y, desde allí, ejecutar los comandos que controlan las acciones de Bind (normalmente mediante la utilidad *rndc*, que se explicará más adelante). De aquí se deduce que, tanto la transferencia remota de zonas como el control sobre el servidor (recarga de zonas, parada, arranque, etc.) se realiza a través de esta clave cifrada. Además, se deduce también que mediante esta restricción no será posible controlar Bind remotamente, como ya se ha dicho. Esta es la opción por defecto y la que el autor de este artículo recomienda.

2. `server 213.96.79.79 { keys { "2002052101.linuxsilo.net.tsigkey."; }; };` es una directiva que indica al servidor cuándo debe usarse la clave. Al usar esta cláusula se obliga al servidor a usar cierta clave cuando se comunique con una determinada dirección IP. Para cada servidor es conveniente especificar una directiva *server*, especificando la dirección IP de la otra máquina y el nombre de la clave a utilizar. En el ejemplo se usa la misma clave para la comunicación entre servidores y para el control del servidor desde el host local - habiendo accedido por SSH - mediante la utilidad *rndc*

Nótese que, si se cambia la clave, la herramienta *rndc* puede dejar de funcionar correctamente. Al realizar este proceso se recomienda para el servidor (*rndc stop* o */etc/init.d/bind9 stop*), sustituir las claves y arrancarlo de nuevo (*/etc/init.d/bind9 start*). Antes de explicar cómo crear una clave de este tipo, veamos el verdadero porqué de su necesidad y los problemas que un fallo de seguridad podría causar.

Acerca de los puertos, Bind usa el 53 TCP para las transferencias y el 53 UDP para las consultas.

Qué es una TSIG y para qué se necesita

El DNS trabaja sobre un modelo pregunta-respuesta. Si un cliente necesita información del DNS, manda una petición al servidor de DNS y éste le devuelve una respuesta. Hasta hace poco sólo era posible basarse en la dirección IP de origen para discernir si debía o no contestarse una consulta. Pero esto no es precisamente "ideal". La autenticación basada únicamente en la dirección IP de origen se considera insegura. Las transacciones firmadas (TSIG, del inglés, **Transaction SIGnatures**) añaden las firmas criptográficas como método de autenticación en una conversación del DNS. Se usa una clave secreta compartida para establecer la confianza entre las partes involucradas.

Las TSIG se usan para asegurar que la información del DNS que pretende provenir de cierto servidor es realmente de ese servidor. Se usan principalmente para la autenticación en la transferencia de zonas entre el servidor de nombres primario y los secundarios. Se quiere asegurar que los servidores secundarios no serán nunca engañados para que acepten una copia de una zona para la cual es el autorizado de un impostor que escucha en la dirección IP del servidor primario.

Las transacciones firmadas se definen en el RFC2845.

En el ejemplo anterior se ha usado la clave *tsigkey.linuxsilo.net.20010922* para autenticar el tráfico del DNS entre los dos servidores, el primario (*66.79.182.201*) y el secundario (*213.96.79.79*).

Creación de una clave TSIG

En la instalación por defecto del paquete Debian se facilita una clave TSIG previamente generada y totalmente funcional. Pero es, desde luego, la misma para todo aquel que se instala ese paquete. Por lo tanto, es más que recomendable cambiarla. A continuación se muestra cómo generar una clave particular y cómo usarla para que la transferencia de zonas se haga de forma segura.

TSIG usa una clave secreta compartida que es incorporada a una dispersiíon (del inglés, **hash**) MD5 de la información a ser firmada. Bind viene con una herramienta para crear este tipo de claves, llamada *dnssec-keygen*, cuyos parámetros son numerosos (ejecute *dnssec-keygen –help* para ver la lista completa y *man dnssec-keygen* para la página del *man* a propósito de esta utilidad). Estos son los pasos a seguir para crear rápidamente una clave:

1. Mediante la ejecución del comando `dnssec-keygen -a HMAC-MD5 -b 512 -n HOST 2002052101.linuxsilo.net.tsigkey.` se crea una clave llamada *2002052101.linuxsilo.net.tsigkey.* usando el algoritmo *HMAC-MD5*, de 512 bits (del inglés, **BInary digiT**) y tipo *HOST* (que es precisamente el uso para el cual va destinada).

2. De los dos ficheros generados, `K2002052101.linuxsilo.net.tsigkey.+157+30191.key` y `K2002052101.linuxsilo.net.tsigkey.+157+30191.private`, se usará sólo el segundo. Se aprovecha para mencionar que el formato de salida de los nombres de los ficheros generados es *Knnnn.+aaa+iiiii*, donde *nnnn* es el nombre de la clave, *aaa* es la representación numérica del algoritmo e *iiiii* es la marca/huella del identificador de la clave (del inglés, **footprint**). Por supuesto, los nombres de cada clave generada deben ser únicos, es decir, dos claves no deberían compartir jamás el mismo nombre, de aquí el nombre tan inusual que se le ha dado. La clave, propiamente dicha, es el conjunto de caracteres que se encuentra tras la palabra *Key:* en la última línea del fichero con sufijo *.private*.

3. El siguiente paso es editar el fichero */etc/bind/rndc.key* y sustituir la clave que viene por defecto por la que acaba de ser generada. Para ello, es suficiente con cambiar el nombre de la clave por defecto de *"rndc.key"* al que le hemos dado al crearla, en este caso *2002052101.linuxsilo.net.tsigkey.*. Por último, hay que cambiar el valor del campo *secret* por el valor de la clave generada que, como se acaba de decir más arriba, se encuentra tras la palabra *Key:* en la última línea del fichero terminado en *.private*. En el caso de ejemplo usado para el artículo, se ha generado la clave *wlnQbRQM/76rol0xGkEdm [..] MMlUFR7HpenQ==*, pero el lector no debe tomar ésta como una referencia, pues variará en cada nueva generación. Este es el contenido del fichero *K2002052101.linuxsilo.net.tsigkey.+157+30191.private*:

```
Private-key-format: v1.2
Algorithm: 157 (HMAC_MD5)
Key: wlnQbRQM/76rol0xGkEdm [..] MMlUFR7HpenQ==
```

   Este es el contenido del fichero */etc/bind/rndc.key* que viene por defecto con la instalación del paquete Debian:

```
key "rndc-key" {
  algorithm hmac-md5;
  secret "bsty5LYDsO8infm+n2JNsw==";
};
```

   Y este es, finalmente, el fichero */etc/bind/rndc.key* resultante del uso de la nueva clave:

```
key "2002052101.linuxsilo.net.tsigkey." {
  algorithm hmac-md5;
  secret "wlnQbRQM/76rol0xGkEdm [..] MMlUFR7HpenQ==";
};
```

Los dos ficheros creados al generar la clave, terminados en *.key* y *.private*, pueden ser eliminados sin problemas. El lector, durante sus pruebas, se dará cuenta de que, si se omite el *"."* al final del nombre de la clave, *dnssec-keygen* lo añadirá automáticamente, pudiéndose crear confusión acerca de si el nombre que corresponde a la clave generada es con punto o sin punto al final. Llegados a este punto, el autor recomienda usar el nombre tal y como se generó, aunque con una simple prueba de ensayo-error (comprobar que se transfieren las zonas) se puede llegar fácilmente a la solución correcta.

Por último, resaltar que una clave secreta es eso: secreta. Por lo tanto, es preciso que sea copiada e instalada en ambos servidores de modo seguro. Además, se recomienda encarecidamente cambiar los permisos de los ficheros */etc/bind/named.conf* y */etc/bind/rndc.key*, tanto del servidor primario como del secundario, a 600 (*chmod 600 /etc/bind/named.conf* y *chmod 600 /etc/bind/rndc.key*), de manera que sean únicamente accesibles por el usuario *root*.

La verificación de una TSIG requiere la posibilidad de escribir un fichero temporalmente. Asegúrese de que *named* tiene permisos de escritura en su directorio por defecto (cláusula *directory* de la directiva *options*, que en Debian es, por defecto, */var/cache/bind/*). La implementación de Microsoft de las TSIG no usa el algoritmo del RFC2845 (HMAC-MD5). El GSS-TSIG de Microsoft no cumple el estándar y, consecuentemente, no interoperará adecuadamente con Bind.

Más información en los documentos RFC2535, RFC2845 y RFC2539

Comentarios sobre la actualización dinámica, la seguridad de las TSIG y las ACL

1. Es crítico que la clave sea guardada en secreto, lo que significa, por ejemplo, que:
   a. *named.conf* y *rndc.key* no deben tener permisos de lectura para nadie que no sea *named* o el usuario que ejecute *rndc* o *nsupdate*.
   b. la clave no debe ser transmitida por emails, a menos que estén cifrados.
   c. cualquiera a quien le dé esta clave es de confianza: por ello, désela exclusivamente a quienes la necesiten, y nunca a personas de las que desconfíe.
   d. debe considerar cambiar de clave cada cierto tiempo, después de cambios en el personal, o si se tienen sospechas de que se pueda haber comprometido el secreto.

2. Si ambos hosts están en la misma subred, es más difícil monitorizar (del inglés, **spoofing**) la dirección IP que hacerse con una copia de la clave (por ejemplo si los routers en contacto con el exterior filtran IPs monitorizadas), de modo que una ACL de direcciones IP sería más efectiva.

3. Es igualmente válido especificar una ACL o una clave TSIG. Por ejemplo `allow-update {key updater; updaters; };`, que significaría que tanto la TSIG como la ACL de direcciones IP son válidas para las actualizaciones.

4. No es posible requerir a la vez una TSIG y control de acceso por IP. Los desarrolladores de Bind no creen que esto sea útil, pues ellos se concentran en el control a nivel de usuario más que a nivel de host de cara a las actualizaciones dinámicas (de aquí el énfasis puesto en la nueva política de actualizaciones que permite a los usuarios con direcciones IP dinámicas actualizar sus registros en el DNS).

5. Las actualizaciones dinámicas no pueden añadir o eliminar dominios, tan sólo registros de esos dominios.

6. Un host cliente que quiera actualizar un servidor Bind únicamente necesita el binario *nsupdate* y la clave apropiada. No se requieren otros binarios o librerías (del inglés, **libraries**) adicionales.

7. *nsupdate* soporta el parámetro *"-d"* para tareas de depuración (del inglés, **debugging**).

8. *nsupdate* también puede usar *TCP* (del inglés, **Transmission Control Protocol**) en lugar de *UDP* (del inglés, **User Datagram Protocol**) para las actualizaciones (parámetro *"-v"*), lo que proporciona un mejor rendimiento si son muchas las actualizaciones a realizar y mayor seguridad ya que *TCP* es un protocolo orientado a conexión. Además, una conexión *TCP* tiene la posibilidad de ser dirigida (del inglés, **piped**) a través de un canal (del inglés, **tunnel**) *SSH* (del inglés, **Secure SHell**)para más seguridad (encriptación y control de acceso).

9. La política de actualizaciones es una nueva característica de Bind 9 que permite que las actualizaciones se restrinjan a ciertos nombres específicos. Por

ejemplo, para permitir que un usuario de *ADSL* (del inglés, **Asymmetric Digital Subscriber Line**) o *DHCP* (del inglés, **Dynamic Host Configuration Protocol**) pueda actualizar el nombre de su propio host (es decir, aquellos a que cambian de dirección IP). Con esta política de actualizaciones, puede configurarse una lista de claves por host y permitir a cada clave que actualice únicamente el host o zona asociada.

Riesgos a los que expone un Bind inseguro

¿Es realmente necesario preocuparse también por el DNS? Bien, un DNS comprometido puede exponerse a algunos riesgos interesantes:

1. Un atacante puede obtener información muy interesante si se permiten transferencias de zonas: la lista completa de hosts y encaminadores (del inglés, **routers**) con sus direcciones IP, nombres y, posiblemente, comentarios indicando su situación, etc.

2. Denegación de servicio (del inglés, **Denial of service**): si todos sus servidores de DNS caen,

    - Su sitio web (del inglés, **website**) ya no es visible (los otros websites no pueden traducir su dirección IP).

    - Los correos electrónicos ya no pueden ser enviados (algunos sitios en Internet con los cuales se intercambia información a menudo habrán guardado en su caché los registros de DNS, pero eso no durará más que unos pocos días).

    - Un atacante podría inciar un falso servidor de DNS que finge ser el suyo y envía información de DNS falsa a Internet acerca de su dominio. Es decir, pérdida de integridad - veáse la siguiente sección.

3. Pérdida de integridad: si un atacante puede cambiar los datos del DNS o facilitar (mediante spoofing) a otros sitios falsa información (esto se conoce como envenenamiento de DNS (del inglés, **DNS poisoning**), la situación se vuelve muy peliaguda:

    - Falsificar (del inglés, **fake**) su website, de manera que parezca el suyo, y capturar las entradas de los usuarios que iban destinadas a su sitio, por lo que se estaría hablando de robar cualquier cosa, desde nombres de usuario (del inglés, **logins**) y contraseñas (en inglés, **passwords**) hasta números de tarjetas de crédito.

    - Todo el correo podría ser redirigido a un servidor repetidor (del inglés, **relay**) que podría copiar, cambiar o borrar correo antes de pasarlo a su sitio.

    - Si su cortafuegos (del inglés, **firewall**) o cualquier host accesible desde Internet usa nombres de host de DNS (del inglés, **DNS hostnames**) para autenticarse o para relaciones de confianza, éstas pueden ser completamente comprometidas, especialmente si un débil filtro de paquetes es quien protege los servidores de Internet y la Intranet. Imagine un proxy web configurado para permitir peticiones proxy sólo desde **.midominio.com*. El atacante añade su host al dominio, por lo que el proxy web pasa a permitir peticiones que provengan de él, permitiendo al atacante acceso por HTTP a la Intranet. Imagine un administrador de sistemas que usa SSH (gran invento criptográfico), pero los hosts cortafuegos tienen un *.shosts* confiando en *admin.midominio.com*, donde *admin* es la estación de trabajo del administrador. Si el atacante puede sustituir la entrada para *admin.midominio.com* en el DNS, pasa a tener un acceso libre y sin necesidad de contraseña a los hosts del cortafuegos.

El DNS se ha convertido en el objetivo favorito de los hackers, como prueban las herramientas para realizar ataques automáticos y los gusanos que usan los fallos del DNS que aparecieron durante el invierno de 2001.

Entonces, ¿qué medidas es necesario tomar?

Los riesgos de Bind pueden ser reducidos considerablemente con algunas medidas de prevención:

1. Aislamiento de los recursos: use un servidor dedicado y asegurado para el DNS de Internet, no lo comparta con otros servicios y, especialmente, no permita el acceso remoto de usuario. Minimizar los servicios y usuarios significa reducir la cantidad de software ejecutándose y, por lo tanto, la probabilidad de exponerse a ataques de red. La separación previene contra la posibilidad de que otros servicios o usuarios localicen debilidades en el sistema y las usen para atacar a Bind.

2. Redundancia: instale un secundario en una conexión a Internet diferente (rama alejada de su empresa, otro ISP, etc.). Si su sitio cae, al menos el resto de sitios no pensarán que usted ha "dejado de existir", sino que tan sólo creerán que "no está disponible", por lo que, por ejemplo, sus emails no se perderán sino que entrarán en una cola de espera (típicamente de hasta cuatro días).

3. Use la última versión.

4. Control del acceso: restrinja la transferencia de zonas para minimizar la cantidad de información que esté disponible en su red para los atacantes. Considere el uso de transacciones firmadas. Considere restringir o no permitir las consultas recursivas.

5. Ejecute Bind con los mínimos privilegios: como usuario no *root*, con una *umask* muy restrictiva (por ejemplo, *177*).

6. Mayor aislación de recursos: ejecute Bind en un entorno (del inglés, **jail**) *chroot*, de modo que sea mucho más difícil que un demonio Bind comprometido dañe el sistema operativo o comprometa otros servicios.

7. Configure Bind para que no informe de su versión. Algunas personas no creen en esta medida, pues es "seguridad por ocultación", pero entienda que, al menos, ayudará contra jovencillos con scripts que rastrean la red buscando objetivos obvios. Defenderse de los profesionales es otro asunto.

8. Detección: monitorice los logs buscando actividad inusual y cambios no autorizados en el sistema mediante un analizador de integridad.

9. Manténgase continuamente al día de las novedades y asegúrese que se le notifica la salida de nuevos problemas de Bind en un tiempo razonable.

Servidores recursivos y no recursivos.

Los servidores de nombres pueden actuar recursivamente o no permitirla. Si un servidor no recursivo tiene la respuesta a una petición cacheada de una transacción previa o es el autorizado del dominio al cual la consulta pertenece, entonces proporciona la respuesta apropiada. De otro modo, en lugar de devolver una contestación real, devuelve una referencia al servidor autorizado de otro dominio que sea más capaz de saber la respuesta. Un cliente de un servidor no recursivo debe estar preparado para aceptar referencias y actuar en consecuencia.

Aunque los servidores no recursivos puedan parecer perezosos, tienen habitualmente un buen motivo para deshacerse del trabajo extra. Los servidores raíz y los servidores de más alto nivel son todos no recursivos, pero es que 10.000 consultas por segundo bien son una excusa para serlo.

Un servidor recursivo devuelve únicamente respuestas reales o mensajes de error. Se encarga de seguir las referencias por si mismo, descargando al cliente de esa tarea. El procedimiento básico para traducir una consulta es, esencialmente, el mismo; la única diferencia es que el servidor de nombres se preocupa de de hacerse cargo de las referencias en lugar de devolverlas al cliente.

Localización de servicios

Un registro SRV especifica la localización de los servicios ofrecidos por un dominio. Por ejemplo, el registro SRV permite consultar un dominio remoto directamente y preguntarle por el nombre de su servidor FTP. Hasta ahora, en la mayoría de ocasiones, había que probar suerte. Para contactar el servidor FTP de un dominio remoto, uno esperaba que el administrador de sistemas de ese dominio hubiese seguido el estándar (el gusto mejor dicho) actual y tuviese un *CNAME* para *ftp* en su servidor de DNS.

Los registros SRV adquieren mucha importancia en este tipo de consultas y son realmente una mejor manera para los administradores de sistemas de trasladar servicios y controlar su uso. Sin embargo, deben ser solicitados y analizados explícitamente por los clientes, por lo que sus efectos se irán viendo gradualmente a medida que pase el tiempo.

Los registros SRV se parecen a registros MX generalizados con campos que permiten al administrador local guiar y balancear la carga de las conexiones provenientes del mundo exterior. El formato es

```
servicio.proto.nombre [ttl] IN SRV pri wt puerto destino
```

donde *servicio* es uno de los servicios definidos en la base de datos de números asignada por la IANA, *proto* puede ser *tcp* o *udp*, *nombre* es el dominio al cual el servicio hace referencia, *pri* es una prioridad al estilo de los registros MX, *wt* es el peso usado para balancear la carga entre diferentes servidores, *puerto* es el puerto en el cual el servicio escucha, y *destino* es el nombre de host del servidor en el cual se provee ese servicio. El registro *A* del destino habitualmente es devuelto de forma automática junto a la respuesta enviada a una consulta SRV. Un valor *"0"* para el parámetro *wt* significa que no se realiza ningún tipo especial de balanceo de carga. Un valor de *"."* para el destino significa que el servicio no se ejecuta en ese sitio.

En la zona *linuxsilo.net* del ejemplo, adaptado del RFC2052 (donde se define SRV), se tiene lo siguiente:

```
ftp.tcp         SRV 0 0 21   ftp.linuxsilo.net.
ssh.tcp         SRV 0 0 22   linuxsilo.net.
telnet.tcp      SRV 0 0 23   linuxsilo.net.
smtp.tcp        SRV 0 0 25   smtp.linuxsilo.net.

; 3/4 de las conexiones al principal, 1/4 al secundario
http.tcp        SRV 0 3 80   linuxsilo.net.
http.tcp        SRV 0 1 80   ns2.linuxsilo.net.

; para que funcionen tanto http://www.linuxsilo.net como http://linuxsilo.net
http.tcp.www    SRV 0 3 80   linuxsilo.net.
http.tcp.www    SRV 0 1 80   ns2.linuxsilo.net.

; servidor principal en el puerto 443, secundario - en caso de fallo - en otra máquina y otro puerto
https.tcp       SRV 1 0 443  linuxsilo.net.
https.tcp       SRV 2 0 4443 ns2.linuxsilo.net.
https.tcp.www   SRV 1 0 443  linuxsilo.net.
https.tcp.www   SRV 2 0 443  ns2.linuxsilo.net.
pop3s.tcp       SRV 0 0 995  pop3.linuxsilo.net.

*.tcp           SRV 0 0 0    .
*.udp           SRV 0 0 0    .
```

Este ejemplo ilustra el uso tanto el parámetro *wt* (del inglés, **weigth**) para HTTP como el parámetro de prioridad para HTTPS. Ambos servidores HTTP serán usados, dividiéndose el trabajo entre ellos. El servidor secundario *ns2.linuxsilo.net* sólo será usado para HTTPS cuando el principal no esté disponible. Todos los servicios no especificados están excluidos. El hecho de que el demonio de, por ejemplo, *finger* no aparezca en el DNS no signifia que no se esté ejecutando, sino tan sólo que no se podrá localizar ese servicio a través de DNS.

Microsoft usa los registros SRV estándar en Windows 2000, pero los inserta en el sistema de DNS de una manera incompatible e indocumentada.

Tipos de registros del DNS

	Tipo	Nombre	Función
Zona	SOA	Start Of Authority	Define una zona representativa del DNS
	NS	Name Server	Identifica los servidores de zona, delega subdominios
Básicos	A	Dirección IPv4	Traducción de nombre a dirección
	AAAA	Dirección IPv6 original	Actualmente obsoleto
	A6	Dirección IPv6	Traducción de nombre a dirección IPv6
	PTR	Puntero	Traducción de dirección a nombre
	DNAME	Redirección	Redirección para las traducciones inversas IPv6
	MX	Mail eXchanger	Controla el enrutado del correo
Seguridad	KEY	Clave pública	Clave pública para un nombre de DNS
	NXT	Next	Se usa junto a DNSSEC para las respuestas negativas
	SIG	Signature	Zona autenticada/firmada
Opcionales	CNAME	Canonical Name	Nicks o alias para un dominio
	LOC	Localización	Localización geográfica y extensión
	RP	Persona responsable	Especifica la persona de contacto de cada host
	SRV	Servicios	Proporciona la localización de servicios conocidos
	TXT	Texto	Comentarios o información sin cifrar

Vistas

Las vistas (del inglés, **views**) son una nueva característica de Bind 9 que permite mostrar a las máquinas internas una visión distinta de la jerarquía de nombres de DNS de la que se ve desde el exterior (se entiende "interior" y "exterior" respecto del router que da salida a la empresa a Internet). Por ejemplo, le permite revelar todos los hosts a los usuarios internos pero restringir la vista externa a unos pocos servidores de confianza. O podría ofrecer los mismos hosts en ambas vistas pero proporcionar registros adicionales (o diferentes) a los usuarios internos.

Este tipo de configuración (llamada en ocasiones "DNS partido", del inglés **"split DNS"**) se está haciendo muy popular. En el pasado, se implementaba configurando servidores separados para las versiones interna y externa de la realidad. Los clientes locales apuntaban a los servidores de distribución que contenían la versión interna de la zona, mientras que los registros NS de la zona padre apuntaban a servidores que corrían la versión externa. La sentencia `view` de Bind 9 simplifica la configuración permitiendo tener juntos ambos conjuntos de datos en la misma copia de *named*. *named* busca correspondencias en listas de direcciones para adivinar qué clientes deben recibir qué datos.

La sentencia `view` empaqueta un lista de acceso que controla quién ve la vista, algunas opciones que se aplican a todas las zonas en la vista y, finalmente, las propias zonas. La sintaxis es:

```
view "nombre-de-la-vista" {
  match-clients { address_match_list; };
  opcion-de-vista; ...
  sentencia-de-zona; ...
};
```

La cláusula `match-clients` controla quién puede ver la vista. Las vistas son procesadas en orden secuencial, por lo que las más restrictivas deben ir primero. Las zonas en distintas vistas pueden tener el mismo nombre. Las vistas son una proposición de todo o nada; si las usa, todas las sentencias `zone` en su fichero *named.conf* deben aparecer dentro del contexto de una vista.

Este es el ejemplo para los dominios *linuxsilo.net* y *balearikus-party.org*, creado a partir de la documentación de Bind 9 que sigue el esquema de DNS partido descrito más arriba. Las dos vistas definen ambas zonas, pero con diferentes registros.

```
acl "lan" {
  192.168.0.0/24;
};

// View for all computers on local area network

view "internal" {
  match-clients { lan; };
  recursion yes;

  // be authoritative for the localhost forward and reverse zones, and for
  // broadcast zones as per RFC 1912

  // prime the server with knowledge of the root servers

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

  // Resto de zonas inversas por defecto omitidas para abreviar

  zone "38.127.217.in-addr.arpa" {
    type master;
    file "/etc/bind/db.217.127.38";
    allow-transfer { slaves; };
  };

  zone "79.96.213.in-addr.arpa" {
    type slave;
    file "sec.db.213.96.79";
    masters { 213.96.79.79; };
  };

  zone "0.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.192.168.0";
  };

  // add entries for other zones below here

  zone "balearikus-party.org" {
    type master;
    file "/etc/bind/db.balearikus-party.org.internal";
  };

  zone "linuxsilo.net" {
    type master;
    file "/etc/bind/db.linuxsilo.net.internal";
  };

};

// View for all computers outside the local area network

view "external" {
```

```
  match-clients { any; };
  recursion no;

  // be authoritative for the localhost forward and reverse zones, and for
  // broadcast zones as per RFC 1912

  // prime the server with knowledge of the root servers

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

  // Resto de zonas inversas por defecto omitidas para abreviar

  zone "38.127.217.in-addr.arpa" {
    type master;
    file "/etc/bind/db.217.127.38";
    allow-transfer { slaves; };
  };

  zone "79.96.213.in-addr.arpa" {
    type slave;
    file "sec.db.213.96.79";
    masters { 213.96.79.79; };
  };

  // add entries for other zones below here

  zone "balearikus-party.org" {
    type master;
    file "/etc/bind/db.balearikus-party.org";
    allow-query { any; };
    allow-transfer { slaves; };
  };

  zone "clan-bin.org" {
    type master;
    file "/etc/bind/db.clan-bin.org";
    allow-query { any; };
    allow-transfer { slaves; };
  };
};
```

La red local interna es *192.168.0.0*, de aquí que se use una lista de acceso que engloba a cualquier host que sea de esa red (red *192.168.0.0*, máscara *255.255.255.0*, especificada como suma de unos binarios, es decir, *24*). Esta nueva situación nos lleva a precisar una nueva definición de zona inversa, la correspondiente a la red local *0.168.192.in-addr.arpa*, que se muestra a continuación:

```
;
; BIND reverse data file for zone 192.168.0
;
$TTL 604800
@ IN SOA linuxsilo.net. hostmaster.linuxsilo.net. (
  2001081501   ; Serial
       10800   ; Refresh (3 hours)
        7200   ; Retry (2 hours)
     1296000   ; Expire (15 days)
      172800 ) ; Negative Cache TTL (2 days)

@ IN NS  ns1.linuxsilo.net.
1 IN PTR ns1.linuxsilo.net.
```

Nótese que, ahora, las zonas inversas, tanto las que se proporcionan con la instalación por defecto para el funcionamiento básico como las definidas por el administrador, se reparten adecuadamente entre ambas vistas. En cambio, las zonas directas son duplicadas, una ocurrencia para cada vista. Por supuesto, los ficheros de zona apuntados contienen registros distintos, en consonancia con la vista. Acto seguido se facilitan los registros de los ficheros de zonas directas internas (las externas se mantienen igual, por lo que son válidas las expuestas anteriormente en este artículo).

```
;
; BIND data file for zone balearikus-party.org, internal view
;
$TTL 604800
@ IN SOA balearikus-party.org. hostmaster.linuxsilo.net. (
  2002051001   ; Serial yyyy/mm/dd/id
       10800   ; Refresh (3 hours)
        7200   ; Retry (2 hours)
     1296000   ; Expire (15 days)
      172800 ) ; Negative Cache TTL(2 days)

balearikus-party.org. IN NS ns1.linuxsilo.net.
balearikus-party.org. IN MX 5 ns1.linuxsilo.net.

localhost             IN A 127.0.0.1
balearikus-party.org. IN A 192.168.0.1

www  IN A 192.168.0.1
pop3 IN A 192.168.0.1
```

```
- -
smtp IN A 192.168.0.1
ftp  IN A 192.168.0.1

;
; BIND data file for zone linuxsilo.net, internal view
;
$TTL 604800
@ IN SOA linuxsilo.net. hostmaster.linuxsilo.net. (
  2002051001   ; Serial yyyy/mm/dd/id
       10800   ; Refresh (3 hours)
        7200   ; Retry (2 hours)
     1296000   ; Expire (15 days)
      172800 ) ; Negative Cache TTL(2 days)

NS ns1.linuxsilo.net.
MX 5 ns1.linuxsilo.net.

localhost      A 127.0.0.1
linuxsilo.net. A 192.168.0.1

ns1       A 192.168.0.1
ns2       A 213.96.79.79
www       A 192.168.0.1
pop3      A 192.168.0.1
smtp      A 192.168.0.1
ftp       A 192.168.0.1
ts        A 213.96.79.79
akane     A 192.168.0.1
ranma     A 192.168.0.6
genma     A 192.168.0.5
kasumi    A 192.168.0.4
nabiki    A 213.96.79.79
primetime A 192.168.0.3
```

La herramienta RNDC

El comando *rndc* es una útil herramienta para manipular *named*. La siguiente tabla muestra algunas de las opciones que acepta. Los parámetros que provocan la creación de ficheros lo harán en el directorio especificado como *home* de *named* en el */etc/bind/named.conf* (cláusula *directory*, cuyo valor por defecto es */var/cache/bind* en Debian).

Comando	Función
help	Lista los opciones de rndc disponibles
status	Muestra el estado actual del *named* en ejecución
trace	Incrementa el nivel de depuración en 1
notrace	Desactiva la depuración
dumpdb	Vuelca la base de datos de DNS a *named_dump.db*
stats	Vuelca estadísticas a *named.stats*
reload	Recarga *named.conf* y los ficheros de zonas
reload *zona*	Recarga sólo la zona especificada
restart	Reinicia *named*, vaciando la caché
querylog	Activa el seguimiento de las consultas entrantes

Rndc usa el puerto 953 UDP para el control remoto. Si se siguen las pautas mostradas en este artículo, no es necesario que ese puerto sea accesible desde el exterior - configurarlo en el router - pues el control se hará siempre desde el host local y las transferencias de zonas se realizan por el puerto 53 TCP

Personalización de los logs

Los logs en Bind se configuran con la sentencia `logging` en el *named.conf*. Primero se definen canales, que son los posibles destinos de los mensajes. Luego se les dice a varias categorís de mensajes que vayan a un canal particular.

Término	Significado
canal	Un lugar a donde los mensajes pueden ir: syslog, un fichero o */dev/null*
categoría	Una clase de mensajes que Bind puede generar; por ejemplo, mensajes sobre actualizaciones dinámicas o mensajes acerca de respuestas a consultas
módulo	El nombre del módulo de origen que genera un mensaje
lugar	El nombre de un lugar syslog. DNS no tiene su propio destino, por lo que tendrán que escogerse los estándar.
importancia	Lo "malo" que es un mensaje de error; a lo que syslog se refiere como prioridad

Cuando se genera un mensaje, se le asigna una categoría, un módulo y una importancia en su punto de origen. Después es distribuido a todos los canales asociados con esa categoría y módulo. Cada canal tiene un filtro de importancia que define qué nivel de importancia debe tener un mensaje para pasar. Los canales que llevan al syslog también son filtrados de acuerdo a las reglas del */etc/syslog.conf*.

Este es el esqueleto de una sentencia `logging`:

```
logging {
  definición_de_canal;
  definición_de_canal;
  ...
  category nombre_categoría {
    nombre_canal;
    nombre_canal;
    ...
  };
};
```

Una *definición_de_canal* es ligeramente diferente dependiendo de si el canal es un fichero o un canal syslog. Se debe elegir `file` o `syslog` para cada canal; un canal no puede ser ambas cosas a la vez.

```
channel "nombre_del_canal" {
  file ruta [versions númvers | unlimited] [size sizespec];
  syslog facility;
  severity importancia;
  print-category yes | no;
  print-severity yes | no;
  print-time yes | no;
};
```

En un fichero, *númvers* especifica cuántas versiones de copia de un fichero guardar, y `sizespec` dice lo grandes que pueden llegar a ser esos ficheros (por ejemplo, `2048`, `100k`, `20m`, `15g`, `unlimited`, `default`).

En el caso de syslog, *facility* especifica que nombre de lugar usar al guardar el mensaje. Puede ser cualquiera de los estándar. En la práctica, sólo *daemon* y de *local0* a *local7* son elecciones razonables.

El resto de sentencias en una definición de canal son opcionales. *importancia* puede tomar los valores (en orden descendente) `critical`, `error`, `warning`, `notice`, `info` o `debug` (con un nivel numérico opcional, por ejemplo `severity debug 3`). El valor `dynamic` también es válido y representa el nivel de depuración actual del servidor.

Las diversas opciones `print` añaden o suprimen prefijos del mensaje. El syslog incluye la fecha y hora y el host de origen en cada mensaje guardado, pero no la importancia o la categoría. En Bind 9, el fichero de origen (módulo) que generó el mensaje también está disponible como opción `print`. Adquiere sentido entonces activar `print-time` sólo para canales fichero, pues los registros del syslog ponen la fecha y hora ellos solos.

A continuación se listan los cuatro canales predefinidos por defecto, que deberín ser suficiente para la mayoría de casos:

Nombre del canal	Lo que hace
default_syslog	Manda importancia `info` al syslog con el destino daemon
default_debug	Guarda en el fichero *named.run*, importancia puesta a `dynamic`
default_stderr	Manda mensajes a la salida de error estándar de *named*, importancia `info`
null	Se descartan todos los mensajes

La configuración de `logging` por defecto de Bind 9 es:

```
logging {
  category default {
    default_syslog;
    default_debug;
  };
};
```

Debería echar un vistazo a los ficheros log cuando haga grandes cambios en Bind, y quizás incrementar el nivel de depuración. Entonces, reconfigúrelo para preservar únicamente mensajes importantes una vez *named* esté estable. Algunos de los mensajes de log más comunes se listan a continuación:

- *Lame server*. Si recibe este mensaje acerca de una de sus zonas es que ha configurado mal alguna cosa. El mensaje es relativamente poco importante si es sobre alguna zona en Internet, pues significa que es problema de algún otro.
- *Bad referral*. Este mensaje indica una descoordinacín en la comunicación entre los servidores de nombres de una zona.
- *Not authoritative for*. Un servidor esclavo no es capaz de obtener información representativa de una zona. Quizás está apuntando al maestro equivocado o quizás el maestro ha tenido algún problema cargando esa zona.
- *Rejected zone*. *named* rechazó esa zona porque contenía errores.
- *No NS RRs found*. El fichero de una zona no tiene registros NS tras el registro SOA. Podría ser que no están o que no empiezan con un tabulador o un espacio en blanco. En este último caso, los registros no se interpretan correctamente.
- *No default TTL set.*. La mejor manera de establecer el TTL por defecto es con una cláusula `$TTL` al principio del fichero de la zona. Este mensaje de error indica que el `$TTL` no está presente.
- *No root name server for class*. Su servidor está teniendo problemas para encontrar los servidores raíz. Compruebe su fichero */etc/bind/db.root* y la conexión a Internet de su servidor.
- *Address already in use*. El puerto en el que *named* quiere ejecutarse ya está siendo usado por otro proceso, probablemente otra copia de *named*. Si no ve otra copia de *named* en memoria, podría haberse colgado, dejando el socket de control de *rndc* abierto.

Taxonomía de un servidor de nombres

Tipo de servidor		Descripción
Inglés	**Español**	
authoritative	autorizado	Un representante oficial de una zona.
master	maestro	El repositorio principal de los datos de una zona; lee los datos de ficheros del disco.
slave	esclavo	Obtiene los datos del maestro.
stub	N/A	Parecido a un esclavo, pero sólo copia los datos del servidor de nombres (no los datos del equipo).
distribution	distribución	Un servidor que sólo es visible[a] desde dentro de un dominio (un "servidor oculto").
nonauthoritative[b]	no autorizado	Responde una consulta a partir de su caché; desconoce si los datos son aún válidos.
caching	reserva	Guarda los datos de consultas previas; habitualmente no tiene zonas locales.
forwarder	redireccionador	Realiza consultas en nombre de muchos clientes; mantiene una caché grande.
recursive	recursivo	Consulta en su nombre hasta que devuelve una respuesta o un error.
nonrecursive	no recursivo	Le pasa a otro servidor si no es capaz de responder a la consulta.

a. Un servidor de distribución puede ser visible por cualquiera que conozca su dirección IP.
b. Hablando estrictamente, "no autorizado" es un atributo de la respuesta a una consulta DNS, no de un servidor.

Tipos de sentencias usadas en el *named.conf*

Sentencia	Descripción
include	Interpola un fichero (p.e., claves de confianza accesibles sólo por **named**).
options	Establece opciones globales de configuración del servidor de nombres y valores por defecto.
server	Especifica opciones preservidor.
key	Define información de autenticación.
acl	Define listas de control de acceso.
zone	Define una zona de registro de recursos.
trusted-keys	Usa claves previamente configuradas.
controls	Define canales utilizados para controlar el servidor de nombres con **rndc**.
logging	Especifica categorías de logs y sus destinos.
view	Define una vista de un espacio de nombres.

Ejemplo de personalización de logs

```
// Definimos tres canales de logs (mensajes importantes del
// syslog, depuración media y mensajes de carga de zonas)
// y luego les asignamos categorías a cada uno.
logging {
  channel syslog_errors {
    syslog local1;
    severity error;
  };
  channel moderate_debug {
    severity debug 3;    // nivel 3 de depuración
    file "debug.log";    // al fichero debug.log
    print-time yes;      // fecha actual a las entradas del log
    print-category yes;  // imprimir el nombre de la categoría
    print-severity yes;  // imprimir el nivel de gravedad
  };
  channel no_info_messages {
    syslog local2;
    severity notice;
  };
  category parser {
    syslog_errors;
    default_syslog;
  };
  category lame-servers { null; };  // No guardar este tipo en los logs
  category load { no_info_messages; };
  category default {
    default_syslog;
    moderate_debug;
  };
};
```

Tabla de caracteres especiales utilizados en los registros de recursos

Caracter	Significado
;	Introduce un comentario
@	El nombre de dominio actual
()	Permite partir una sentencia en más de una línea
*	Comodín (sólo en el nombre del campo).

Tabla de mecanismos de seguridad en el *named.conf*

Característica	Sentencias	Qué especifica
allow-query	options, zone	Quién puede consultar la zona o servidor.
allow-transfer	options, zone	Quién puede solicitar transferencias de zonas.
allow-update	zone	Quién puede hacer actualizaciones dinámicas.
blackhole	options	Qué servidores deben ignorarse completamente.
bogus	server	Qué servidores no deben ser jamás consultados.
acl	varios	Listas de control de acceso.

Tabla de categorías de logging en Bind 9

Categoría	Qué incluye
default	Categorís sin una asignación explícita de canal.
general	Mensajes sin clasificar.
config	Análisis y procesado de ficheros de configuración.
queries/client	Un mensaje corto de log por cada consulta que el servidor recibe.
dnssec	Mensajes de DNSSEC.
lame-servers	Servidores que se supone que sirven una zona, pero no lo están[a].
statistics	Estadísticas agrupadas del servidor de nombres.
panic	Errores fatales (duplicados en esta categoría).
update	Mensajes sobre actualizaciones dinámicas.
ncache	Mensajes sobre caché negativa.
xfer-in	Transferencias de zonas que el servidor está recibiendo.
xfer-out	Transferencias de zonas que el servidor está enviando.
db/database	Mensajes sobre operaciones con bases de datos.
packet	Volcados de paquetes recibidos y enviados[b].
notify	Mensajes acerca del protocolo de notificaciones "zona modificada".
cname	Mensajes del tipo "...points to a CNAME".
security	Peticiones aprobadas/denegadas.
os	Problemas del sistema operativo.
insist	Comprobaciones de fallos de consistencia interna.
maintenance	Sucesos periódicos de mantenimiento.
load	Mensajes de carga de zonas.
response-checks	Comentarios sobre paquetes de respuesta malformados o inválidos.
resolver	Traducción de DNS, p.e., búsquedas recursivas para clientes.
network	Operaciones de red.

a. Bien la zona padre o bien la zona hija podrían ser la culpable; es imposible determinarlo sin investigarlo.
b. Obligatoriamente debe ser un canal simple.

Chroot

Este apartado describe algunas precauciones extras relacionadas con la seguridad que puede usted tomar al instalar BIND. Se explica cómo configurar BIND de manera que resida en una *jaula chroot*, lo que significa que no pueda ver o acceder a ficheros fuera de su propio reducido árbol de directorios. También se explica cómo configurarlo para que se ejecute como un usuario diferente a *root*.

La idea que hay detrás de un chroot es bastante sencilla: acotar el acceso que un individuo malicioso pueda obtener explotando vulnerabilidades de BIND. Por esa misma razón es bueno ejecutarlo como un usuario no root (en GNU/Debian Linux a partir de la versión 9.2.4-5). Cuando se ejecuta BIND (o cualquier otro proceso) en una jaula chroot, el proceso simplemente es incapaz de ver cualquier otra parte del sistema de ficheros que se encuentre fuera de la jaula. Por ejemplo, en este

apartado configuraremos BIND en modo chroot en el directorio */var/lib/named*. Entonces, para BIND, el contenido de este directorio será la raíz */*. Nada fuera de este directorio le será accesible. Muy probablemente ya se ha encontrado usted ante una jaula chroot con anterioridad, por ejemplo al acceder mediante FTP a un sistema público.

Este proceso debería considerarse un complemento de las precauciones de seguridad habituales (ejecutar la última versión, usar listas de control de acceso, etc.), y nunca como una manera de reemplazarlas.

Crear el usuario

Tal y como se menciona en la introducción, no es una buena idea ejecutar BIND como root. Por ello, antes de empezar, se creará un usuario separado para BIND. Nótese que nunca debería usarse un usuario genérico existente del tipo *nobody* para este propósito. Este proceso es realizado automáticamente por el script de instalación del paquete Debian, pero a continuación se resume el procedimiento manual para conseguirlo.

Es necesario añadir una línea parecida a esta en el fichero */etc/passwd*:

```
bind:x:103:103::/var/cache/bind:/bin/false
```

Y una similar a la siguiente en el fichero */etc/group*:

```
bind:x:103:
```

En este ejemplo no sólo vamos a ejecutar BIND como un usuario no root, sino que también lo haremos en un entorno chroot (la instalación por defecto en Debian únicamente cubre el primer aspecto en la actualidad).

Estas líneas crean un usuario y un grupo llamados *bind* para BIND. El lector debe asegurarse de que tanto el UID (del inglés, **User Identifier**) como el GID (del inglés, **Group Identifier**) son únicos en su sistema (ambos 103 en este ejemplo). La consola se ha dejado en */bin/false* porque este usuario jamás tendrá necesidad de hacer un login.

Estructura de directorios

Acto seguido es necesario crear una estructura de directorios para la jaula chroot en la cual se ejecutará BIND. Puede hacerse en cualquier lugar del sistema de ficheros; aquellos más paranoicos incluso querrán ponerlo en un volumen separado. Se usará */var/lib/named*. Empiece por crear la siguiente estructura de directorios:

```
/var/lib/
  +--- named
        +--- dev
        +--- etc
        +--- var
              +--- run
              +--- cache
                    +--- bind
        +--- usr
              +--- sbin
```

Mediante el comando *mkdir* podría crearse la mencionada estructura de directorios. Estos son los comandos a ejecutar y sus parámetros:

```
# mkdir -p /var/lib/named
# cd /var/lib/named
# mkdir -p dev etc/bind var/run var/cache/bind usr/sbin
```

Copiar los ficheros necesarios

Asumiendo que usted ya ha realizado una instalación estándar de BIND 9 y que lo está usando, tendrá, por lo tanto, un fichero *named.conf* y diversos ficheros de zonas. Esos ficheros deben ser movidos (o copiados, para mayor seguridad) dentro de la jaula chroot, de modo que BIND sea capaz de encontrarlos. *named.conf* y los ficheros de zonas van dentro de */var/lib/named/etc/bind*. Por ejemplo:

```
# cp -p /etc/bind/* /var/lib/named/etc/bind/
# cp -a /var/cache/bind/* /var/lib/named/var/cache/bind/
# cp /usr/sbin/named /var/lib/named/usr/sbin/
```

Además, será preciso copiar también las librerís que el ejecutable */usr/sbin/named* necesita, que pueden obtenerse ejecutando el comando *ldd /usr/sbin/named*. Con los siguientes comandos crearemos los subdirectorios necesarios y copiaremos las librerís:

```
# cd /var/lib/named
# mkdir -p usr/lib lib
# cp /usr/lib/liblwres.so.1 usr/lib/
# cp /usr/lib/libdns.so.5 usr/lib/
# cp /usr/lib/libcrypto.so.0.9.6 usr/lib/
# cp /usr/lib/libisccfg.so.0 usr/lib/
# cp /usr/lib/libisccc.so.0 usr/lib/
# cp /usr/lib/libisc.so.4 usr/lib/
# cp /lib/libnsl.so.1 lib/
# cp /lib/libpthread.so.0 lib/
# cp /lib/libc.so.6 lib/
# cp /lib/libdl.so.2 lib/
# cp /lib/ld-linux.so.2 lib/
```

BIND va a necesitar escribir dentro de los subdirectorios */var/lib/named/var/cache/bind* y */var/lib/named/var/run*, en el primero para guardar las zonas de las cuales esté actuando como servidor esclavo y en el segundo para guardar información estadística de su ejecución. Por lo tanto, es pertinente ejecutar las dos instrucciones siguientes:

```
# chown -R bind:bind /var/lib/named/var/cache/bind
# chown -R bind:bind /var/lib/named/var/run
```

Ficheros de sistema

Una vez que BIND esté ejecutándose en la jaula chroot, no será capaz de acceder a ficheros que se encuentren fuera de la jaula de ningún modo. Sin embargo,

necesita acceder a algunos ficheros esenciales. Uno de los ficheros que necesita dentro de su jaula es */dev/null*. Otros son */dev/zero* y */dev/random*. Pueden usarse los siguientes comandos:

```
# mknod /var/lib/named/dev/null c 1 3
# mknod /var/lib/named/dev/random c 1 8
# mknod /var/lib/named/dev/zero c 1 5
# chmod 666 /var/lib/named/dev/{null,random,zero}
```

También será necesario otro fichero en el directorio */etc* dentro de la jaula. Es preciso copiar */etc/localtime* ahí dentro de modo que BIND guarde los logs con fechas correctas. El siguiente comando se resolvería el problema:

```
# cp /etc/localtime /var/lib/named/etc/
```

Logging

Normalmente, BIND guarda los logs a través de *syslogd*, el demonio del sistema encargado de guardar los logs. Sin embargo, este tipo de logging se lleva a cabo enviando las entradas del log a un socket especial, */dev/log*. Debido a que se encuentra fuera de la jaula, BIND no va a poder usarlo. Afortunadamente, hay una solución para este problema.

Todo lo que hay que hacer es añadir el parámetro *-a /var/lib/named/dev/log* a la línea de comandos que lanza el syslogd. En Debian, este script se encuentra en */etc/init.d/sysklogd*. Debe buscar las líneas siguientes:

```
# Options for start/restart the daemons
# For remote UDP logging use SYSLOGD="-r"
#
SYSLOGD=""
```

Y cambiar la última de las líneas por esta:

```
SYSLOGD="-a /var/lib/named/dev/log"
```

Una vez reiniciado el demonio, debería ver un fichero en */var/lib/named/dev* llamado *log*, tal que así:

```
srw-rw-rw- 1 root root 0 Nov 28 14:22 log
```

Finalmente, remarcar que, en el caso de una actualización del paquete *sysklogd*, podrían perderse los cambios realizados en dicho script por una potencial sobreescritura del fichero existente. Por lo tanto, se recomienda tener cuidado al actualizar.

Endureciendo los permisos

Antes de nada, siéntase libre de restringir acceso en todo el directorio */var/lib/named* únicamente al usuario bind.

```
# chown bind:bind /var/lib/named
# chmod 700 /var/lib/named
```

Si desea aumentar aún más las restricciones, en los sistemas Linux puede conseguirse la inmutabilidad de algunos de los ficheros usando la herramienta *chattr* en los sistemas de ficheros *ext2* y *ext3*.

```
# cd /var/lib/named
# chattr +i etc etc/localtime var
```

Consulte la página del *man* para más información: *man chattr*. Sería interesante poder hacer esto mismo sobre el directorio *dev* pero, desgraciadamente, eso imposibilitaría que el *syslogd* crease el socket *dev/log*. También puede elegir activar el bit de inmutabilidad en otros ficheros dentro de la jaula, como los de las zonas primarias, en el caso de que no vayan a cambiar.

Instalación

Si se desea realizar una instalación a partir de los fuentes, se recomienda usar el paquete Debian de fuentes *bind9*, disponible mediante el comando *apt-get source bind9*. Luego tan sólo sería necesario ejecutar un *./configure* y un *make* o, más fácil aún, *dpkg-buildpackage*, que nos hará los dos pasos anteriores y nos dejará listos una serie de paquetes Debian que nos evitarán tener que hacer un *make install* a mano.

Suponiendo que usted ya dispone de una instalación de BIND 9 en su sistema, tan sólo deberá modificar ligeramente el script de arranque del demonio para que use estos parámetros:

`-u bind`, que le indica a BIND el usuario con el que debe ejecutarse.
`-t /var/lib/named`, que le dice a BIND que haga un chroot sobre sí mismo dentro de la jaula que se le ha preparado.
`-c /etc/named.conf`, que le dice a BIND dónde encontrar su fichero de configuración dentro de la jaula.

En Debian, es muy fácil cambiar el script de inicio de BIND, que encontrará en */etc/init.d/bind9*, para que acepte estas nuevas opciones. Tan sólo debe buscar estas líneas:

```
# for a chrooted server: "-u nobody -t /var/lib/named"
OPTS=""
```

Y cambiar la última línea por la siguiente:

```
OPTS="-u bind -t /var/lib/named -c etc/named.conf"
```

Cambie, por último, el ejecutable que se llama desde ese script de */usr/sbin/named* a */var/lib/named/usr/sbin/named*, de manera que sea el ejecutable de dentro de la jaula el que el script llame y no el original.

Si su versión del paquete Debian de BIND9 es la 9.2.4-5 o superior, BIND se estará ejecutando con el usuario bind y el UID/GID 103, por lo que puede saltarse los pasos referidos a la creación del usuario y el grupo y pasar directamente al enjaulamiento. Si la versión de su paquete es inferior (en GNU/Debian Woody es la 9.2.4-4), entonces deberá seguir todos los pasos.

Cambios en la configuración

Deberá usted también cambiar o añadir unas pocas opciones a su *named.conf* a fin de mantener ciertos directorios en orden. En particular, debería añadir (o cambiar,

si ya las tiene) las siguientes directivas en la sección de opciones:

```
directory "/etc/bind";
pid-file "/var/run/named.pid";
statistics-file "/var/run/named.stats";
```

Ya que este fichero va a ser leído por el demonio *named*, todas las rutas van a ser, evidentemente, relativas a la jaula chroot. En el momento de escribir esta parte del documento, BIND 9 no soporta muchas de las estadísticas y ficheros de volcado que las versiones previas soportaban. Presumiblemente, versiones posteriores sí lo harán; si está ejecutando una de esas versiones, quizás deba añadir algunas entradas adicionales para que BIND las escriba en el directorio */var/run* también.

Arrancando BIND

Ahora ya tan sólo queda por hacer el paso más elemental: arrancar de nuevo el demonio BIND. Para ello, es preciso ejecutar el siguiente comando en una distribución GNU/Debian Linux:

```
# /etc/init.d/bind start
```

Recursos en línea

- The DNS Resources Directory
- DNS HowTo
- Securing DNS with Transaction Signatures
- All About DNS
- DNS Cómo

Los RFC

Los RFC que definen el sistema de DNS están disponibles en www.rfc-editor.org. Las ideas iniciales y en desarrollo aparecen primero como borradores y son más tarde formalizadas como RFC. A continuación se listan un conjunto relacionado con Bind, incluidos los que han supuesto que Bind 9 se haya reescrito desde cero (estos documentos están todos en inglés):

Los originales y definitivos estándares:

- 1034 - Domain Names: Concepts and Facilities.
- 1035 - Domain Names: Implementation and Specification.

Estándares propuestos:

- 1995 - Incremental Zone Transfers in DNS.
- 1996 - A Mechanism for Prompt Notification of Zone Changes.
- 2136 - Dynamic Updates in the Domain Name System.
- 2181 - Clarifications to the DNS Specification.

RFC de seguimiento de nuevos estándares:

- 2535 - Domain Name System Security Extensions.
- 2671 - Extension Mechanisms for DNS (EDNS0).
- 2672 - Non-Terminal DNS Name Redirection (DNAME).
- 2673 - Binary Labels in the Domain Name System.

RFC diversos:

- 1535 - A Security Problem... with Widely Deployed DNS Software.
- 1536 - Common DNS Implementation Errors and Suggested Fixes.
- 1982 - Serial Number Arithmetic.
- 2536-2541 - Varios RFC sobre DNSSEC.

Tipos de registros de recursos:

- 1183 - New DNS RR Definitions: AFSDB, RP, X25, ISDN, RT.
- 1706 - DNS NSAP Resource Records.
- 1876 - A Means for Expressing Location Information in DNS.
- 2052 - A DNS RR for Specifying the Location of Services (SRV).
- 2168 - Resolution of Uniform Resource Identifiers using DNS.
- 2230 - Key Exchange Delegation Record for the DNS.

DNS e Internet:

- 1101 - DNS Encoding of Network Names and Other Types.
- 1123 - Requirements for Internet Hosts: Application and Support.
- 1591 - Domain Name System Structure and Delegation.
- 2317 - Classless in-addr.arpa Delegation.

Operaciones de DNS:

- 1537 - Common DNS Data File Configuration Errors.
- 1912 - Common DNS Operational and Configuration Errors.
- 2182 - Selection and Operation of Secondary DNS Servers.
- 2219 - Use of DNS Aliases for Network Services.

Otros RFC relacionados con el DNS:

- 1464 - Using DNS to Store Arbitrary String Attributes.
- 1713 - Tools for DNS debugging.
- 1794 - DNS Support for Load Balancing.
- 2240 - A Legal Basis for Domain Name Retrieval.
- 2345 - Domain Names and Company Name Retrieval.
- 2352 - A Convention for Using Legal Names as Domain Names.

Historial de revisiones

Fecha	Versión	Cambios
2002-05-27	1.0	Documento inicial
2002-05-28	1.0.1	Añadido nuevo enlace a recurso en línea
2002-05-29	1.0.2	Correción del protocolo usado para las transferencias de zonas.
2002-05-30	1.0.3	Diversos cambios en las zonas inversas, tanto en los ejemplos como en la explicación de su funcionamiento.
2002-06-01	1.0.4	Bind 9.2.4 ha sido añadido a Debian Sid. Actualización del artículo tras comprobar el *changelog*.
2002-06-03	1.0.5	Corregidos algunos errores gramaticales y añadidos varios enlaces a referencias usadas en el artículo.
2003-03-03	1.1	Añadidas varias secciones nuevas al artículo, principalmente tablas de resumen y el chroot, y un índice de contenidos.
2003-03-05	1.1.1	Corregidos algunos errores tipográficos en la sección de chroot.
2003-03-06	1.1.2	Corregida aún más la sección de chroot, añadiendo las librerías a la jaula y cambiando la ruta del ejecutable a llamar.
2005-02-22	2.0	Reescrito el artículo. Ahora se adecúa a la nueva disposición de los ficheros de configuración del paquete Debian de Bind 9.2.4. Se han simplificado los ejemplos para hacerlos m´s claros e ilustrativos, y se incluye un único dominio de ejemplo, pero con un servidor maestro y otro esclavo igualmente. Añadida más información en varios aspectos, como el de los registros DNS-LOC.