IURRETA INSTITUTUA
Goi-mailako Lanbide-Heziketako Berariazko Institutua
2005

EUSKO JAURLARITZA
GOBIERNO VASCO
HEZKUNTZA UNIBERTSITATE ETA IKERKETA SAILA
DEPARTAMENTO DE EDUCACIÓN UNIVERSIDADES E INVESTIGACIÓN

DNS

Domain Name Service

Versión: 1.0

Alfredo Barrainkua Zallo

Febrero de 2008

Índice

1. Introducción

Los usuarios de redes esperan tener acceso a una gran variedad de servicios de alto nivel. Loguearse en servidores remotos, acceder a ficheros almacenados en servidores, ver información de páginas web, bajarse ficheros, y cosas de esas. Para ello debemos indicar en qué lugar se encuentra esa información. Debemos indicar la dirección del host.

Inicialmente, se referenciaban los hosts por su dirección. Pero pronto se dieron cuanta de que esa no era una forma práctica. Los humanos somos muy malos recordando números. Recordamos más fácilmente los nombres. El siguiente paso fue mapear en una tabla las direcciones y los nombres de hosts (**/etc/hosts**). Esto permite referenciar un host tanto por su nombre como por su dirección. Aquí vemos un ejemplo de este fichero:

```
127.0.0.1         localhost.localdomain   localhost
10.22.1.12        nireserver.nire-esdkola.net   nireserver
172.24.82.14      niremakina.nire-eskola.net    www
```

Este método funciona bien, pero tiene un problema. Las tablas han de residir en cada host. Cada vez que se quiere añadir un host a la red, debemos actualizar las tablas de cada host. En redes pequeñas, esto puede no suponer un problema, pero según va creciendo la red, se hace insostenible.

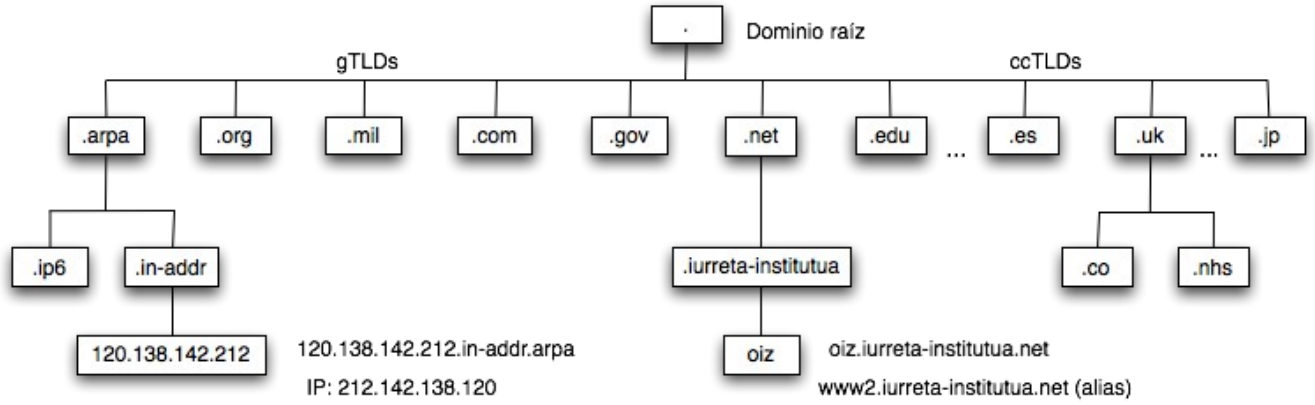
Para solucionar este problema, se diseñó una jerarquía de nombres, y se implementó en una base de datos distribuida. Por ello, para realizar el mapeo de nombres con direcciones, nos debemos de conectar primero a uno de estos servidores, y preguntarle la correlación de nombre y dirección. A esto se le llama resolución de nombres.

A este servicio se le denomina DNS (Domain Name Service). La primera implementación fue realizada en la universidad de Berkeley. Esta implementación es BIND (Berkeley Interdomain Name Daemon). Nosotros vamos a usar la versión 9.3.4 de BIND.

1.1. Jerarquía DNS

La jerarquía DNS tiene forma de árbol invertido. a jerarquía superior de la raíz (in-addr-arpa) se señaliza con un punto (.). Bajo éste, se encuentran las jerarquías **TLD** (Top Level Domain) o dominios de nivel superior. Todas las demás cuelgan de éstas. Las TLD originales o genéricas (**gTLD**) son 6, se representan con tres letras y todas ellas son de EEUU. Luego se encuentran las TLDs correspondientes a los estados o Country Code

TLDS (**ccTLD**), con las dos letras de la codificación **ISO 3166**. Últimamente se han añadido más jerarquías de varias letras, para diferenciar tipos de servicios y negocios. Además hay una jerarquía un tanto especial, pues corresponde a las direcciones inversas IP. Se utiliza para la resolución inversa. En realidad es para dar completitud semántica a la estructura. Aquí vemos un gráfico con las jerarquías.



Estructura DNS

Listado de los dominios de nivel superior (TLD).

Dominios generales	
.com	Commercial – Entidades comerciales
.edu	Education – Entidades educativas superiores
.gov	Government – Agencias gubernamentales de EEUU
.mil	Military – Organizaciones militares de EEUU
.net	Network – Organizaciones relacionadas con la infraestructura de Internet
.org	Organizations – Entidades sin ánimo de lucro / no comerciales

Dominios geográficos (ISO 3166)	
.es	España
.uk	Reino Unido
.us	EEUU
.tw	Taiwan
.de	Alemania
.jp	Japón
.mx	México

Dominios de negocios u organizaciones	
.int	Organizaciones establecidas por tratados internacionales
.biz	Negocios
.firm	Empresas o firmas
.store	Venta de bienes
.web	Sostenedores de la Web
.arts	Organizaciones culturales y de entretenimiento
.rec	Servicios de recreo
.info	Servicios de información
.name	Individuales
.eu	Unión Europea
.aero	Líneas aéreas
.coop	Organismos y asociaciones de cooperación
.museum	Museos
.pro	Profesionales (Médicos, Abogados, ...)
(.xxx)	Pornografía (No aprobado. El año pasado -2007- se decidió posponer la votación por presiones de EEUU)

1.2. Nombres

El nombre completo de un host (Full Qualified Domain Name), comienza con el nombre del host, y continuando con el subdominio y dominio, termina con el dominio de nivel superior. Todos separados por puntos. Semánticamente hablando, al final hay un punto, que corresponde a la raíz y que habría que poner. Debido a que no aporta información, y para facilitar la sintaxis, dicho punto no se pone. Muchas veces, los nombres no se ponen completos, solamente el nombre del host o el alias (CNAME). En ese caso, el sistema añade el nombre de dominio por defecto al nombre de host que hemos puesto.

1.3. Servidores raíz

Hay trece servidores raíz. Se identifican por una letra, comenzando por la "**A**" hasta la "**M**". En realidad, cada letra, no es un servidor, sino un grupo de ellos, que actúan como uno solo. Por ejemplo, el operado por el **ISC** (**Internet Software Consortium**), creadores del software ISC DHCP, y mantenedores de ISC BIND, es el "**F**". Uno de los nodos se encuentra en Madrid, y es el que habitualmente utilizamos. Podemos ver el nodo que utilizamos con la siguiente orden:

```
dig +norec @F.ROOT-SERVERS.NET HOSTNAME.BIND CHAOS TXT
```

Si en la respuesta aparece "**`mad1b.f.root-servers.org`**", lo estamos utilizando.

2. Funcionamiento del servicio DNS

Cuando ponemos un nombre, el sistema lo convierte si es preciso en un nombre FQDN. Seguidamente, y si dicho nombre no se encuentra de forma local (el fichero **/etc/hosts**), hace una petición de resolución para dicho nombre, al servidor DNS configurado como tal. Si el servidor tiene la respuesta, nos la proporcionará, indicándonos además si es autoritativa, es decir, dicho servidor tiene autoridad en dicho dominio. En caso de que no lo conozca, hay dos maneras de proceder Recursiva y no recursiva. Las solicitudes de resolución se realizan en el puerto UDP/53. La transferencia de zonas entre maestro y esclavo se realiza por el puerto TCP/53.

2.1. Resolución recursiva

En este modo, ante una consulta que no sabe resolver, el servidor hará una petición a otro servidor, y nos proporcionará la respuesta. Este modo se utiliza en servidores internos. No en servidores públicos. El servidor trabaja por nosotros. Además, el servidor cachea la respuesta, para poder utilizarla en caso de que se repita la consulta.

2.2. Resolución no recursiva

En este modo, el servidor nos indicará qué otro servidor puede resolvernos la consulta solicitada. Tras ello, tendremos que realizar nuevamente la consulta a este nuevo servidor. Nosotros nos encargamos de realizar el trabajo. Es el modo por defecto, y el que se utiliza en los servidores públicos de Internet. En este caso también se cachea la consulta.

3. Trabajos previos a la instalación

Antes de instalar y configurar DHCP, realizaremos una serie de configuraciones en el servidor, que si bien no son específicas de DHCP, si son imprescindibles para su correcto funcionamiento. Son detalles que hay que cuidar en un servidor, para este servicio o para cualquier otro.

3.1. Nombre del servidor

Necesitamos poder resolver el nombre de nuestro servidor independientemente de que tengamos un servidor DNS. Al inicio del sistema, puede suceder que necesitemos resolver el nombre de nuestra máquina, y no tengamos aún cargada la red o el servicio de interrogación DNS, o simplemente puede suceder que no podamos acceder al servidor DNS. Para ello, vamos a verificar que el nombre del servidor esté correctamente establecido en los ficheros **/etc/hostname** y **/etc/hosts.** He aquí el contenido del fichero **/etc/hostname**:

```
nireserver
```

El contenido del fichero **/etc/hosts**:

```
127.0.0.1       localhost
127.0.1.1       nireserver.nire-eskola.net nireserver

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

3.2. Dirección IP

La IP deberá de ser fija. Por medio de **Escritorio, Configuración del sistema, Configuración de red, IP fija**, podremos hacerlo. Otra manera es en el fichero **/etc/network/interfaces**. En este fichero aparecen todos los interfaces de red. Deberá

contener algo similar a esto:

```
auto eth0
iface eth0 inet static
      address 10.22.1.12
      netmask 255.255.255.0
      gateway 10.22.1.1
```

3.3. Servidores DNS

Debemos configurar los servidores DNS a utilizar para la resolución de nombres. Se puede realizar esta configuración a través de **Escritorio, Configuración del sistema, Configuración de red, IP fija**. Otra forma es editar directamente el fichero **/etc/resolv.conf**. Deberá contener algo similar a :

```
search euskaltel.es
nameserver 212.55.8.132
```

3.4. La hora, la hora, la hora

Debemos asegurarnos de que la hora esté correctamente configurada. Para ello, configuraremos el cliente **NTP** (**Network Time Protocol**) para que el reloj del sistema se sincronice con un servidor NTP. Lo primero es instalar los paquetes necesarios:

```
      aptitude install ntp ntpdate
```

Posteriormente, configuraremos el fichero **/etc/ntp.conf** con un servidor de tiempo. En el fichero debe aparecer una línea similar a esta:

```
server 0.pool.ntp.org
server 1.pool.ntp.org
```

Reiniciaremos el demonio.

```
      /etc/init.d/ntp restart
```

En Redes de Área Local, lo más probable es que haya un servidor NTP, que sea el que se sincroniza con un reloj que haya en Internet. En este caso, los demás equipos de la red (incluidos los equipos que den algún tipo de servicio) se sincronizarán con este servidor. Así conseguimos que sea un solo equipo el que tenga que comunicarse con un servidor de Internet, reduciendo el tráfico entre la LAN e Internet.

3.5. Deshabilitar el reinicio por teclado

La distribución Debian, por defecto tiene activado el reinicio del sistema cuando detecta la combinación de teclado **CTRL+ALT+DEL**. Debemos desactivarlo. De este modo, se deberá introducir el comando oportuno para reiniciar el sistema.

El el fichero **/etc/inittab**, debemos comentar (poner "**#**" delante) la siguiente línea, que quedará así:

```
#ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -a now
```

3.6. Acceso por la red : SSH

En un servidor es común acceder al mismo por la red. Esto nos permite acceder al mismo desde lugares remotos, o aunque estemos en la misma localización, no tener que instalar periféricos para el mismo. el acceso se puede realizar tanto en modo consola o gráfico.

Las comunicaciones conviene realizarlas cifradas. De este modo, no se transmitirán por la red nuestros datos (entre otros, contraseñas de acceso) en testo plano. Los datos se cifran con algún método criptográfico. No vamos a entrar aquí en teoría ni sistemas de certificados, ni cosas de esas. Vamos a ver simplemente un acceso de clave compartida.

El sistema de acceso que vamos a utilizar el **ssh**. Hay que mencionar que **ssh** también permite la copia de ficheros entre máquinas, de forma cifrada con **scp**. Vamos a instalarlo.

```
aptitude install ssh
```

Vamos a configurar el paquete. El fichero de configuración es **/etc/ssh/sshd_config**. Vamos a poner las siguientes líneas de esta forma:

```
Protocol 2
PermitRootLogin no
```

Si querremos securizar más el sistema, podemos evitar el acceso con password y que se usen certificados, limitarlo a unas máquinas, etc.

Para acceder al sistema como el usuario **elusuario**:

```
ssh elusuario@lamaquina.eldominio.net
```

Para copiar el fichero **mifile** de **/home/mihome** al directorio home del usuario **elusuario** en el sistema 10.22.1.59:

```
scp /home/mihome/mifile elusuario@10.22.1.59:/home/elusuario/
```

4. Instalación y configuración del servidor DNS, BIND

Vamos a instalar el paquete correspondiente al servidor BIND (Berkeley InterDomain Name Daemon). La instalación la haremos de la forma habitual.

```
aptitude install bind9 bind9-doc gbindadmin
```

Tras la instalación, ya tenemos funcionando un servidor DNS en nuestra máquina. Más adelante lo configuraremos para prestar servicio.

De cara a la configuración es importante conocer y / o recordar algunos de los ficheros y carpetas de los equipos Linux:

Ficheros y Directorios Importantes	
/etc/	Directorio con las configuraciones del sistema.
/home/	Directorios home de los usuarios.
/var/	Directorio con los datos del sistema (hay otros).
/etc/passwd	Fichero con los usuarios locales.
/etc/shadow	Fichero de contraseñas cifradas de los usuarios locales.
/etc/group	Grupos locales y sus usuarios.
/etc/hostname	Fichero que contiene el nombre del host.
/etc/hosts	Fichero con pares de direcciones IP y nombres, para resolver nombres de hosts.
/etc/network/interfaces	Fichero con la configuración de los interfaces de red.
/etc/resolv.conf	Fichero con la configuración del resolver. Contiene el dominio por defecto y las direcciones IP de los servidores de nombres.
/etc/ntp.conf	Configuración del cliente y servidor NTP.
Ficheros y Directorios relativos a DNS	
/etc/default/bind9	Configuración del servidor DHCP en sí.
/var/lib/named/	Directorio con la configuración del servicio y las zonas.
/etc/bind/named.conf	Configuración principal del servicio DNS.

/var/cache/bind	Base de datos de las zonas del servidor DNS.
/var/log/daemon.log	Fichero por defecto donde se almacena el registro de las incidencias relativas a DNS.
/var/log/bind9-query.log	Fichero con las consultas realizadas al servidor DNS.
/etc/init.d/bind9	Script de arranque / parada / recarga del servidor.

Veamos a ver los ficheros de configuración iniciales y su contenido. Más tarde crearemos los de nuestras zonas.

Ficheros de configuración básicos	
/etc/bind/named.conf	Configuración principal. El resto de ficheros (excepto los de datos de zonas) es como si estuviesen en este. Se realiza su inclusión.
/etc/bind/named.conf.options	Las opciones a aplicar al servicio. Se incluye en el fichero principal.
/etc/bind/named.conf.local	Fichero para personalizar el servicio de la máquina. Se incluye en el fichero principal. Aquí definiremos nuestras zonas.
/etc/bind/zones.rfc1918	Definiciones de las zonas de redes privadas según RFC1918. No lo vamos a usar.
/etc/bind/db.*	Datos de las zonas que siempre han de existir. La raíz (root), la local (local), la de broadcast (255), las inversas local(127) y de broadcast (255).
/etc/bind/rndc.key	Clave compartida con el programa de gestión remota del servidor.

4.1. Configuración de BIND

Cuando estemos configurando BIND, debemos tener siempre en cuenta una cosa: **CUIDADO CON EL PUNTO FINAL DE LOS NOMBRES**. Los nombres pueden ser absolutos o relativos (por decirlo de alguna forma). Los nombres absolutos acaban en punto. Si no ponemos punto, se tomarán en relación con el nombre de la zona y, por tanto, siempre se les añadirá el nombre del dominio. Si escribimos **makina.nire-eskola.net.**, ese será nuestro ordenador con el nombre **makina**. Sin embargo, si escribimos **makina.nire-eskola.net**, el nombre del ordenador será **makina-nire-eskola.net.nire-eskola.net**. Este comportamiento se puede cambiar, pero no entraremos en esas profundidades. ¡Cuidado!

Por otra parte, la extensión DNS necesita dos archivos (de forma simple). Uno para hacer la resolución directa. Es decir, cuando pedimos la dirección IP de un nombre. El otro para la resolución inversa. Para cuando queramos saber el nombre de la máquina a partir de una dirección IP. Este segundo fichero debemos crearlo para cada subred IP.

Veamos ahora el fichero de opciones (**/etc/bind/named.conf.options**), tal y como viene con la distribución. Le vamos a quitar los comentarios para que no ocupe tanto. Tengamos en cuenta además que estamos trabajando con dos redes: 192.168.201.0/24 y 192.168.202.0/24. Otro detalle: Los comentarios en los ficheros de configuración deben de ir precedidos por "**//**" o "**#**". En los de datos de zonas deberán ir precedidos por "**;**".

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

        auth-nxdomain no;    # conform to RFC1035
        listen-on-v6 { any; };
};
```

Vemos que solo tiene tres opciones. La primera indica el directorio donde va a cachear las zonas. La segunda nos indica que no responde autoritativamente a peticiones de dominios no existentes. La tercera indica que debemos ocuparnos también de las peticiones IPv6. En nuestro caso no tenemos direcciones IPv6, por lo que no lo necesitamos.

Vamos a incluir unas opciones más, y las incluiremos comentadas, para documentarlo. Así quedará el fichero.

```
options {
      // Directorio de trabajo
      directory "/var/cache/bind";

      // fichero para escribir el registro y las estadísticas
      dump-file "/var/log/named_dump.db";
      statistics-file "/var/log/named.stats";

      // Servidores DNS a los que hacer peticiones de información de otras zonas
      forwarders { 212.55.8.132; 212.55.8.133; };

      // Primero, usar el primero
      forward first;

      // Direcciones y puertos en los que realizar la escucha
      // e peticiones de resolución y otras funciones
      // Deben ser direcciones de nuestro host
      listen-on port 53 { 127.0.0.1; 192.168.201.48; 192.168.202.48;   };

      // Aceptar peticiones de ordenadores de estas redes
      allow-query { 127.0.0.1; 192.168.201.0/24; 192.168.202.0/24;     };

      // No enviar notificaciones de actualización de zonas
      notify no;

      // No responde autoritativamente a peticiones de dominio inexistentes
      auth-nxdomain no;
```

```
      // escucha en Ipv6 (en cualquier interfaz)
      listen-on-v6 { any; };
};
```

Hay otra sección que no viene en el ejemplo original, pero que lo vamos a poner. Es la sección de registro. En ella le indicamos al servicio, dónde y cómo ha de realizar el registro. Por defecto hemos visto que realizaba los apuntes en el fichero **/var/log/daemon.log**. Con esto podemos modificarlo.

```
logging {
      // Enviar las peticiones al registro principal del sistema
      channel syslog_queries {
            syslog user;
            severity info;
      };
      category queries { syslog_queries; };

      // Enviar los errores al registro principal del sistema
      channel syslog_errors {
            syslog user;
            severity error;
      };
      category errors { syslog_errors; };

};
```

Bien. Veamos ahora las definiciones de zonas para nuestra red. Primero veremos cuales son las que siempre han de existir, y luego añadiremos las nuestras. El contenido del fichero de configuración principal (**/etc/bind/named.conf**), trae estas zonas:

```
// Servidores root. Los servidores de nombres principales de Internet
zone "." {
        type hint;
        file "/etc/bind/db.root";
};

// Zonas locales directa e inversa. Es la autoridad en esta zona (RFC1912)
zone "localhost" {
        type master;
        file "/etc/bind/db.local";
};

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

// Zona inversa que representa a la red broadcast
```

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

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

Vemos que cada zona está definida con la palabra "**zone**". Después tenemos el tipo. Pueden ser "**hint**" que es la zona raíz, en la que están los servidores raíz. Otro tipo es el "**master**". Esto indica que es un servidor principal o maestro en ese dominio / zona. El otro tipo es "**slave**". Esto indica que es un servidor secundario o esclavo. La palabra "**file**", indica en qué fichero se encuentran los datos para esa zona. Las zonas inversas se escriben al revés, siendo la raíz **in-addr.arpa** (hay que tener en cuenta que Internet nació como un proyecto de DARPA y que el sistema de direcciones estaba a sus órdenes).

4.2. Ficheros de zonas

Veamos el aspecto que presenta un fichero de datos de zona. Hay dos formas de definir algunos aspectos de una zona. Nosotros utilizaremos la que creemos que es más clara. De todos modos, en los ejemplos viene de otra forma. No importa. No tendremos que modificar estas zonas.

```
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     localhost. root.localhost. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      localhost.
@       IN      A       127.0.0.1
```

En la siguiente tabla tenemos el significado de estos campos:

$TTL	Tiempo de vida. Esto quiere decir los servidores de otros dominios pueden tener este zona en su memoria caché durante el tiempo de vida (604800 segundos)
@	Indica el nombre de nuestro dominio. También puede

	ponerse el dominio directamente (con un punto al final).
IN	Clase: IN = Internet
SOA	Registro de la clase: SOA = Start Of Authority
localhost. / @	Nombre del servidor (con punto al final). También puede ponerse “**@**”.
root.localhost.	Dirección electrónica del administrador de la zona (**root@localhost.**). Se sustituye la arroba por el punto, pues la arroba tiene un significado particular en el contexto de este fichero. Termina en punto.
Serial	Indica un número de serie para el fichero, y funciona como un control de versiones. Lo habitual es poner la fecha de modificación en formato AAAAMMDDxx, siendo xx un número creciente. Ciando se actualiza el fichero, se debe modificar este número. **MUY IMPORTANTE**!. Los esclavos lo tienen en cuanta a la hora de realizar transferencias de zona.
Refresh	Cada cuánto tiempo ha de refrescar el esclavo la información de este zona
Retry	Si el esclavo no puede refrescar la información de zona, intentarlo tras pasar este tiempo.
Expire	Después de pasar cuánto tiempo, no tienen valor para este área, los datos que están en los servidores esclavos.
Negative Cache TTL / Minimum	El tiempo que se necesita cachear una respuesta negativa.

A continuación, aparecen los servidores. En este caso, el carácter de arroba tiene el significado del nombre del dominio. También pude estar vacío. Lo veremos en nuestras zonas. La clase es Internet (**IN**). El tipo de registro es NS (Name Server). Indica que se refiere a un servidor DNS. “**localhost.**” indica la máquina donde corre el servidor. La siguiente línea indica que la máquina nuestra (**@**) de clase Internet (**IN**), tiene una dirección (**A**) 127.0.0.1.

Vemos que esto es algo engorroso. No es fácil de entender de este modo, pero por el contrario, un fichero de estos no lo tenemos que modificar. Para nuestras zonas será más fácil, y además una vez creadas, solamente necesitaremos añadir hosts de vez en cuando. Algo muy fácil!.

5. Creación de zonas

Vamos a definir una red de ejemplo para crear el servidor DNS que dé servicio a la misma. se tratará de una escuela. Una escuela con dos subredes. Una de profesores y otra de alumnos. Tendrá seis servidores.

Subredes de la escuela	
Subredes	Direcciones IP
Profesores	192.168.201.0/24
Alumnos	192.168.202.0/24

Nombres de los hosts	
Máquina	Direcciones IP
nerbioi	192.168.201.1, 192.168.202.1
urumea	192.168.201.8, 192.168.202.8
ibaizabal	192.168.202.9
oka	192.168.201.10, 192.168.202.10
arga	192.168.201.11, 192.168.202.11
zadorra	192.168.201.12, 192.168.202.12

Características y servicios de la escuela		
Nombre de dominio	nire-eskola.net	
Servicio	Nombre DNS	Servidor
Servidor web de Internet	www.nire-eskola.net	212.55.8.7
Servidor web de notas y faltas	notak.nire-eskola.net	zadorra
Servidor web interno	www2.nire-eskola.net	urumea
Servidor de correo	posta.nire-eskola.net	oka
Servidor de mensajería instantánea	jabber.nire-eskola.net	ibaizabal
Cortafuegos	suhesia.nire-eskola.net	nerbioi
Diccionario Elhuyar	hiztegia.nire-eskola.net	urumea
Expedientes de alumnos	elkartuz.nire-eskola.net	ibaizabal

Características y servicios de la escuela		
Servidor para distribuir antivirus	antibirusa.nire-eskola.net	arga
Servidor FTP	ftp.nire-eskola.net	urumea

Vamos a añadir ahora la definición de las zonas propias de nuestra red. Definiremos una zona para la red, y una zona inversa para cada subred. Esto lo haremos en el fichero **/etc/bind/named.conf.local**. Añadimos esto al final.

```
// Zona directa de nuestro dominio
zone "nire-eskola.net" in {
      type master;
      file "/etc/bind/nire-eskola.net.hosts";
};

// Zona inversa de la subred 1
zone "201.168.192.in-addr.arpa" in {
      type master;
      file "/etc/bind/192.168.201.rev";
};

// Zona inversa de la subred 2
zone "202.168.192.in-addr.arpa" in {
      type master;
      file "/etc/bind/192.168.202.rev";
};
```

Vamos a crear ahora nuestros ficheros de zonas. En la definición de zonas hemos puesto que la zona directa correspondiente a nuestro dominio se encuentra en el fichero **/etc/bind/nire-eskola.net.hosts**, Está bastante claro que en este fichero se encuentran los hosts de nuestro dominio (**nire-eskola**). No? Veamos su contenido:

```
$TTL 2D
nire-eskola.net.  IN SOA           @     sare-admin.nire-eskola.net. (
                        2005061200  ; serial
                        2H          ; refresh
                        1200        ; retry
                        2W          ; expire
                        3600        ; minimum
);
                        ; Servidor de correo
            IN MX       5 posta
            IN MX       10 entrante.empresa.euskalnet.net.
                        ; Servidor DNS
            IN NS       urumea

            IN A        212.55.8.7  ; Servidor web de Internet
```

```
; Servidores de nuestro dominio
nerbioi     IN A        192.168.201.1
            IN A        192.168.202.1

urumea      IN A        192.168.201.8
            IN A        192.168.202.8

ibaizabal   IN A        192.168.202.9

oka         IN A        192.168.201.10
            IN A        192.168.202.10

arga        IN A        192.168.201.11
            IN A        192.168.202.12

zadorra     IN A        192.168.201.12
            IN A        192.168.202.12

posta       IN A        192.168.201.10

; Servidor web de Internet (en euskaltel)
www         IN A        212.55.8.7

; Aliases de los hosts
ftp         IN CNAME    urumea
www2        IN CNAME    urumea

suhesia     IN CNAME    nerbioi

jabber      IN CNAME    ibaizabal

elkartuz    IN CNAME    ibaizabal
notak       IN CNAME    zadorra
hiztegia    IN CNAME    urumea
antibirusa  IN CNAME    arga
```

Aquí aparecen algunas cosas nuevas. El tiempo de vida ($TTL) aparece con “2D” (dos días). Más fácil que en segundos. Refresh es 2H (dos horas), y expire 2W (2 semanas). Por otro lado, en lugar de referirnos a nuestro dominio (**@**), se escribe su nombre de forma explícita (**nire-eskola.net.**). Más fácil, a mi modo de entender.

En la parte siguiente tenemos nuestro(s) servidor(es) de correo. Tenemos un servidor de correo en la escuela pero, como lo tenemos dado de alta en los DNS en Euskaltel (y tenemos que hacerlo así para recoger el correo externo), Euskaltel recoge los correos que nos envían, si el servidor no está en funcionamiento. Luego, este servidor de correo envía los mensajes al nuestro, cuando se pone en marcha de nuevo (backup server). El registro para indicar los servidores de correo es **MX** (Mail Exchanger). El número que hay detrás del nombre del registro es la prioridad. Cuanto más pequeño es el número,

mayor es la prioridad. Por tanto, los servidores de correo exterior enviarán al servidor **posta.nire-eskola.net** los mensajes enviados a nuestro dominio **posta.nire-eskola.net** y, si no pueden conectarse con éste, se los darán a la máquina de Euskaltel **entrante.empresa.euskalnet.net**. El siguiente registro es de tipo **NS** (Name Server). Es decir, el servidor DNS. En nuestro caso es **urumea.** Como al final no tiene punto será **urumea.nire-eskola.net**. También tenemos un registro A que apunta nuestro servidor web en Internet. Podemos ver que en el lado izquierdo de estos registros no hay nada. Cuando no se pone nada, se coge lo anterior. En este caso, "**nire-eskola.net.**"). En el caso del servidor de correo y el de DNS, y como veremos después, en los servidores con más de una dirección, esta forma de indicación es normal. En los últimos tiempos se ha puesto de moda hacerlo también con los servidores web. esto hace, que si en un navegador ponemos solamente el dominio, directamente se referencie el servidor web.

A continuación, aparecen los servidores. Como no tienen puntos, se les añade el nombre del dominio. El tipo de registro es **A** (address). Tenemos que poner todas las direcciones IP. Si no ponemos nada a la izquierda de la clase **IN,** coge la de arriba; por tanto, no tenemos porqué repetir los nombres de las máquinas.

El siguiente grupo es el de los apodos. A la izquierda ponemos el apodo (alias), luego será **IN** (clase) y el registro **CNAME** (Canonical Name). A la derecha pondremos a qué máquina hace referencia.

Como detalle, tenemos que el servidor **ibaizabal** solamente está en la red de profesores. Esto hace que desde la red de alumnos no se pueda acceder directamente (sin pasar por un ruter) al servidor de expedientes y a la mensajería instantánea.

Ahora veremos los ficheros de las zonas inversas. Como sólo tenemos un único dominio (y no tenemos subdominios), sólo hemos creado una zona directa. Sin embargo, se necesitan tantas zonas inversas como número de subredes. En nuestro caso, dos. He aquí el fichero de zona inversa de la red **192.168.201.0/24**. El nombre del fichero es **192.168.201.rev**.

```
$TTL 2D
201.168.192.in-addr.arpa.     IN SOA      urumea      sare-admin.nire-eskola.net. (
                        2005061200          ; serial
                        2H                  ; refresh
                        1200                ; retry
                        2W                  ; expiry
                        3600                ; minimum
);

      IN NS       urumea.nire-eskola.net.

1     IN PTR            nerbioi.nire-eskola.net.
8     IN PTR            urumea.nire-eskola.net.
10    IN PTR            oka.nire-eskola.net.
10    IN PTR            posta.nire-eskola.net.
11    IN PTR            arga.nire-eskola.net.
```

```
12    IN PTR            zadorra.nire-eskola.net.
```

Hemos comentado el registro **SOA a**l ver la zona directa. Luego tenemos el registro de nuestro servidor DNS. Finalmente, la parte del host de las direcciones IP de las máquinas, la clase IN, el registro tipo **PTR** (Pointer) y el nombre completo del host (acaba con un punto).

Ahora crearemos el fichero de la zona inversa de la red **192.168.202.0/24**. El nombre del fichero es **192.168.202.rev**.

```
$TTL 2D
202.168.192.in-addr.arpa.     IN SOA      urumea      sare-admin.nire-eskola.net. (
                        2005061200          ; serial
                        2H                  ; refresh
                        1200                ; retry
                        2W                  ; expiry
                        3600                ; minimum
);

      IN NS       urumea.nire-eskola.net.

1     IN PTR            nerbioi.nire-eskola.net.
8     IN PTR            urumea.nire-eskola.net.
9     IN PTR            ibaizabal.nire-eskola.net.
10    IN PTR            oka.nire-eskola.net.
10    IN PTR            posta.nire-eskola.net.
11    IN PTR            arga.nire-eskola.net.
12    IN PTR            zadorra.nire-eskola.net.
```

Como detalle, podemos ver que se ha puesto el servidor de correo como un nombre, y no como alias. A los servidores de correo no les gustan los alias. Es conveniente poner su nombre, pues utilizan mucho la resolución inversa para labores de protección.

6. Uso del servidor

Además de configurar hosts clientes para que hagan uso de él, podemos utilizarlo directamente para realizar búsquedas o para chequeos, o lo que queramos. En Windows disponemos de la herramienta **nslookup**. En Linux podemos utilizar tanto esta herramienta, como dig. Veámoslo.

6.1. nslookup

Tiene dos modos de funcionamiento, interactivo o no.

En el modo interactivo, nos permite hacer consultas, seleccionar el servidor, seleccionar el tipo de consulta, realizar consultas, y permanecemos en el entorno. Podemos utilizar la ayuda escribiendo **help**. Esta herramienta también se encuentra en el entorno windows. Veamos una sesión:

```
NA103-DE03:~# nslookup
> server
Default server: 10.22.1.8
Address: 10.22.1.8#53
Default server: 10.22.1.9
Address: 10.22.1.9#53
> server 192.168.201.48
Default server: 192.168.201.48
Address: 192.168.201.48#53
> server
Default server: 192.168.201.48
Address: 192.168.201.48#53
> urumea.nire-eskola.net
Server:         192.168.201.48
Address:        192.168.201.48#53

Name:   urumea.nire-eskola.net
Address: 192.168.201.8
Name:   urumea.nire-eskola.net
Address: 192.168.202.8
> ibaizabal.nire-eskola.net
Server:         192.168.201.48
Address:        192.168.201.48#53

Name:   ibaizabal.nire-eskola.net
```

```
Address: 192.168.202.9
> set type=MX
> nire-eskola.net
Server:         192.168.201.48
Address:        192.168.201.48#53

nire-eskola.net mail exchanger = 5 posta.nire-eskola.net.
nire-eskola.net mail exchanger = 10 entrante.empresa.euskalnet.net.nire-
eskola.net.
> set type=NS
> nire-eskola.net
Server:         192.168.201.48
Address:        192.168.201.48#53

nire-eskola.net nameserver = urumea.nire-eskola.net.
> set type=PTR
> 1.201.168.192.in-addr.arpa
Server:         192.168.201.48
Address:        192.168.201.48#53

1.201.168.192.in-addr.arpa      name = nerbioi.nire-eskola.net.
> exit

NA103-DE03:~#
```

En el modo no interactivo, realizamos la consulta y fin.

```
NA103-DE03:~# nslookup urumea.nire-eskola.net 192.168.201.48
Server:         192.168.201.48
Address:        192.168.201.48#53

Name:   urumea.nire-eskola.net
Address: 192.168.201.8
Name:   urumea.nire-eskola.net
Address: 192.168.202.8

NA103-DE03:~#
```

6.2. dig

Podemos utilizar dig de varias maneras. Además permite realizar consultas múltiples. Vamos a ver unos ejemplos.

6.2.1. dig @192.168.201.48 urumea.nire-eskola.net A

```
NA103-DE03:~# dig @192.168.201.48 urumea.nire-eskola.net A

; <<>> DiG 9.3.4 <<>> @192.168.201.48 urumea.nire-eskola.net A
; (1 server found)
```

```
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37728
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;urumea.nire-eskola.net.                IN      A

;; ANSWER SECTION:
urumea.nire-eskola.net. 172800  IN      A       192.168.201.8
urumea.nire-eskola.net. 172800  IN      A       192.168.202.8

;; AUTHORITY SECTION:
nire-eskola.net.        172800  IN      NS      urumea.nire-eskola.net.

;; Query time: 0 msec
;; SERVER: 192.168.201.48#53(192.168.201.48)
;; WHEN: Mon Feb 18 16:56:54 2008
;; MSG SIZE  rcvd: 86
```

6.2.2. dig @192.168.201.48 ibaizabal.nire-eskola.net A

```
NA103-DE03:~# dig @192.168.201.48 ibaizabal.nire-eskola.net A

; <<>> DiG 9.3.4 <<>> @192.168.201.48 ibaizabal.nire-eskola.net A
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7601
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; QUESTION SECTION:
;ibaizabal.nire-eskola.net.     IN      A

;; ANSWER SECTION:
ibaizabal.nire-eskola.net. 172800 IN    A       192.168.202.9

;; AUTHORITY SECTION:
nire-eskola.net.        172800  IN      NS      urumea.nire-eskola.net.

;; ADDITIONAL SECTION:
urumea.nire-eskola.net. 172800  IN      A       192.168.201.8
urumea.nire-eskola.net. 172800  IN      A       192.168.202.8

;; Query time: 0 msec
;; SERVER: 192.168.201.48#53(192.168.201.48)
;; WHEN: Mon Feb 18 16:57:09 2008
;; MSG SIZE  rcvd: 112
```

6.2.3. dig @192.168.201.48 nire-eskola.net MX

```
NA103-DE03:~# dig @192.168.201.48 nire-eskola.net MX

; <<>> DiG 9.3.4 <<>> @192.168.201.48 nire-eskola.net MX
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39239
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 1, ADDITIONAL: 3

;; QUESTION SECTION:
;nire-eskola.net.               IN      MX

;; ANSWER SECTION:
nire-eskola.net.        172800  IN      MX      10
entrante.empresa.euskalnet.net.nire-eskola.net.
nire-eskola.net.        172800  IN      MX      5 posta.nire-eskola.net.

;; AUTHORITY SECTION:
nire-eskola.net.        172800  IN      NS      urumea.nire-eskola.net.

;; ADDITIONAL SECTION:
posta.nire-eskola.net.  172800  IN      A       192.168.201.10
urumea.nire-eskola.net. 172800  IN      A       192.168.202.8
urumea.nire-eskola.net. 172800  IN      A       192.168.201.8

;; Query time: 1 msec
;; SERVER: 192.168.201.48#53(192.168.201.48)
;; WHEN: Mon Feb 18 16:57:21 2008
;; MSG SIZE  rcvd: 171
```

6.2.4. dig @192.168.201.48 nire-eskola.net NS

```
NA103-DE03:~# dig @192.168.201.48 nire-eskola.net NS

; <<>> DiG 9.3.4 <<>> @192.168.201.48 nire-eskola.net NS
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 966
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 2

;; QUESTION SECTION:
;nire-eskola.net.               IN      NS

;; ANSWER SECTION:
nire-eskola.net.        172800  IN      NS      urumea.nire-eskola.net.
```

```
;; ADDITIONAL SECTION:
urumea.nire-eskola.net. 172800  IN      A       192.168.201.8
urumea.nire-eskola.net. 172800  IN      A       192.168.202.8

;; Query time: 0 msec
;; SERVER: 192.168.201.48#53(192.168.201.48)
;; WHEN: Mon Feb 18 16:57:25 2008
;; MSG SIZE  rcvd: 86
```

6.2.5. dig @192.168.201.48 1.201.168.192.in-addr.arpa PTR

```
NA103-DE03:~# dig @192.168.201.48 1.201.168.192.in-addr.arpa PTR

; <<>> DiG 9.3.4 <<>> @192.168.201.48 1.201.168.192.in-addr.arpa PTR
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 58122
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

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

;; ANSWER SECTION:
1.201.168.192.in-addr.arpa. 172800 IN   PTR     nerbioi.nire-eskola.net.

;; AUTHORITY SECTION:
201.168.192.in-addr.arpa. 172800 IN     NS      urumea.nire-eskola.net.

;; ADDITIONAL SECTION:
urumea.nire-eskola.net. 172800  IN      A       192.168.202.8
urumea.nire-eskola.net. 172800  IN      A       192.168.201.8

;; Query time: 0 msec
;; SERVER: 192.168.201.48#53(192.168.201.48)
;; WHEN: Mon Feb 18 16:58:01 2008
;; MSG SIZE  rcvd: 134

NA103-DE03:~#
```

6.3. host

host es un comando para interrogar servidores DNS. Es muy simple. Veamos un ejemplo.

```
NA103-DE03:~# host urumea.nire-eskola.net 192.168.201.48
Using domain server:
Name: 192.168.201.48
Address: 192.168.201.48#53
```

```
Aliases:

urumea.nire-eskola.net has address 192.168.201.8
urumea.nire-eskola.net has address 192.168.202.8
NA103-DE03:~#
```

6.4. rndc

rndc es un comando para gestionar el servidor DNS, en línea. Se puede hacer que cargue zonas, que recargue toda la configuración, que pare, etc. Ejecutamos **rndc** sin parámetros, para que nos diga lo que puede hacer.

```
sasiroot2@zeus:~$ rndc
Usage: rndc [-c config] [-s server] [-p port]
        [-k key-file ] [-y key] [-V] command

command is one of the following:

  reload        Reload configuration file and zones.
  reload zone [class [view]]
                Reload a single zone.
  refresh zone [class [view]]
                Schedule immediate maintenance for a zone.
  retransfer zone [class [view]]
                Retransfer a single zone without checking serial number.
  freeze        Suspend updates to all dynamic zones.
  freeze zone [class [view]]
                Suspend updates to a dynamic zone.
  thaw          Enable updates to all dynamic zones and reload them.
  thaw zone [class [view]]
                Enable updates to a frozen dynamic zone and reload it.
  notify zone [class [view]]
                Resend NOTIFY messages for the zone.
  reconfig      Reload configuration file and new zones only.
  stats         Write server statistics to the statistics file.
  querylog      Toggle query logging.
  dumpdb [-all|-cache|-zones] [view ...]
                Dump cache(s) to the dump file (named_dump.db).
  stop          Save pending updates to master files and stop the server.
  stop -p       Save pending updates to master files and stop the server
                reporting process id.
  halt          Stop the server without saving pending updates.
  halt -p       Stop the server without saving pending updates reporting
                process id.
  trace         Increment debugging level by one.
  trace level   Change the debugging level.
  notrace       Set debugging level to 0.
  flush         Flushes all of the server's caches.
  flush [view]  Flushes the server's cache for a view.
```

```
  flushname name [view]
                Flush the given name from the server's cache(s)
  status        Display status of the server.
  recursing     Dump the queries that are currently recursing (named.recursing)
  validation newstate [view]
                Enable / disable DNSSEC validation.
  *restart      Restart the server.

* == not yet implemented
Version: 9.4.1-P1
sasiroot2@zeus:~$
```

7. Respuestas ordenadas y vistas

Cuando preguntamos a un servidor DNS, sobre la dirección de un host, si posee más de una dirección nos las proporciona todas. El servidor BIND, nos las proporcionará además, en un orden desconocido, y a cada pregunta le responderá con las direcciones en orden diferente. Este comportamiento es adecuado para realizar balanceos de carga entre distintos servidores. El orden se modifica de una forma circular (round robin), por defecto. Los sistemas Windows, ordenan esas respuestas, por lo que no tienen capacidad de participar en el balanceo de carga. Los sistemas Unix / Linux en cambio, utilizan la primera dirección proporcionada, y si no obtienen respuesta, la siguiente, y así sucesivamente. En nuestro caso, nos conviene que los host tengan distintas direcciones para que los datagramas no pasen por el cortafuegos, pero deseamos que utilicen la dirección de su segmento de red. Cómo solucionarlo? De dos maneras: Ordenando las respuestas, y proporcionando diferentes vistas.

7.1. Respuestas ordenadas

Le podemos indicar al servidor BIND que nos ordene las respuestas, en función de la dirección IP del solicitante. A eso se le llama ordenamiento de direcciones. Para ello, en el fichero de configuración, y en la sección de opciones, le indicamos las listas de ordenamiento.

```
sortlist {
      { 192.168.201/24; { 192.168.201/24; 192.168.202/24; }; };
      { 192.168.202/24; { 192.168.202/24; 192.168.201/24; }; };
};
```

Como se puede ver en el ejemplo, las solicitudes realizadas desde la red 192.168.201.0, serán respondidas con direcciones de las redes 192.168.201.0 y 192.168.202.0, y en ese orden. Con las otras redes, se hace lo mismo.

7.2. Vistas DNS (views)

Con las vistas. el servidor DNS, nos proporcionará respuestas diferentes, dependiendo la red desde la cual se han realizado las peticiones. De esta manera, las máquinas Linux se conectarán más rápido y a los sistemas Windows les es indiferente. El problema más importante, es que debemos crear ficheros de todas las zonas por cada vista que deseamos tener. Además. el fichero de configuración habrá de tener distintas definiciones de zonas para cada vista. Esto hace que sea un poco engorroso. En este curso no las vamos a implementar, pues además del engorro de su creación tienen otro problema. Las vistas

también influyen en la actualización dinámica de las zonas por parte de un servidor DHCP. Al mostrar cada vista en función de la dirección IP del solicitante, cuando se actualizan las zonas, solamente se pueden actualizar de la manera en que por la dirección IP utilizada, permita la vista. Esto hace que el servidor DHCP pueda actualizar unos u otras en función de la dirección IP empleada para ello. El resultado es una tremenda complejidad. En este caso, no se recomienda su uso. Nosotros no lo vamos a implementar, pues queremos realizar las actualizaciones dinámicas, y llevaría el curso, más allá de lo pretendido.

8. Servidor DNS Esclavo (slave)

Tenemos un servidor DNS. Dos redes internas, una serie de servidores y un montón de clientes. Todo esto repartido en varios edificios. Qué ocurre si en uno de estos edificios hay un corte de electricidad o si el servidor DNS se estropea? Todos quietos? (menos el administrador de la red, claro). Necesitaremos por lo tanto uno o dos servidores DNS más. En este capitulo veremos cómo poner en marcha un servidor esclavo (slave). En nuestra escuela, el servidor esclavo será **oka**.

Se le llama servidor esclavo, pero eso no quiere decir que tenga menos autoridad que el maestro (master) en el dominio (o en las áreas que le corresponden). Quiere decir simplemente, que los cambios se hacen en el maestro, y luego se propagan al esclavo. Además, un servidor puede ser maestro de unas zonas, y esclavo de otras.

La transferencia de zonas entre servidores maestro y esclavo se realiza cuando un maestro envía una notificación de cambio de los datos de la zona a un esclavo, o cuando ha expirado el tiempo de validez de los datos de esa zona. Se realiza por el puerto TCP/53.

Haremos la instalación del software igual que al principio. Los script de inicio serán los mismos. Los archivos **/etc/bind/named.conf.options** y **/etc/bind/named.conf.local** deberán ser modificados un poco. Pondremos las áreas DNS igual pero donde pone **master** pondremos **slave**. Pondremos los archivos de zona en el directorio **/etc/bind/esclavo/**. Debemos de tener en cuenta que el proceso **bind** debe escribir en este directorio**.** Por lo tanto, el dueño del directorio será **bind**. También deberemos decirle desde dónde deberá actualizarse, y si el master puede actualizarlo o no. Por ultimo, autentificaremos al esclavo por medio de claves **TSIG**.

8.1. Configuración del esclavo

En primer lugar comprobaremos la configuración del esclavo. Copiaremos del maestro los archivos **/etc/bind/named.conf.options** y **/etc/bind/named.conf.local**. Como nuestro esclavo es **oka** y como el servidor oka tiene dos direcciones IP, deberemos colocar esas direcciones en el archivo **/etc/bind/named.conf.options**.

```
# Puertos en los que esta a la espera de conexiones
listen-on port 53 { 127.0.0.1; 192.168.201.10; 192.168.202.10; };
```

Dejaremos el **Allow query** como estaba ya que deberá responder a las preguntas de

toda nuestra red. Ahora, deberemos indicar en cada zona, que es un servidor esclavo. Esto no es necesario en las zonas **localhost.zone** y **0.0.127.in-addr.arpa** , ya que no se modifican y por lo tanto no se actualizan. Dejaremos esas áreas y el área (**root.hint**) como maestro. Por otro lado deberemos indicarle al esclavo desde qué dirección IP debe actualizar las zonas. A todos les pondremos la misma. La dirección IP(**192.168.202.8**) de la subred de profesores de nuestro maestro (**urumea**). Veamos como quedaría esto.

```
zone "nire-eskola.net" {
      type slave;
      masters {192.168.202.8; };
      transfer-source 192.168.202.10;
      file "esclavo/nire-eskola.net.hosts";
};

# Zona inversa de la subred 1
zone "201.168.192.in-addr.arpa" in {
      type slave;
      masters {192.168.202.8; };
      transfer-source 192.168.202.10;
      file "esclavo/192.168.201.rev";
};

# Zona inversa de la subred 2
zone "202.168.192.in-addr.arpa" in {
      type slave;
      masters {192.168.202.8; };
      transfer-source 192.168.202.10;
      file "esclavo/192.168.202.rev";
};
```

Hemos puesto que el esclavo almacena las zonas de las cuales es clavo, dentro del directorio **/etc/bind/esclavo/**. Debemos de crearlo y darle la propiedad a BIND.

```
mkdir /etc/bind/esclavo
chown bind:bind /etc/bind/esclavo
```

8.2. Configuración del Maestro

Ahora habilitaremos en el maestro, la transferencia de zonas, para que el esclavo las pueda leer. Para hacer esto, en el archivo maestro **/etc/bind/named.conf.options**, debemos cambiar la sección options. En esta sección, al final, pondremos:

```
allow-transfer {192.168.200.10; 192.168.202.10; };
```

De esta manera podremos actualizar las áreas desde cualquier dirección IP del servidor esclavo DNS. Si queremos transferir unas zonas si y otras no, deberemos poner en cada zona que queramos transferir, la opción “**allow-transfer**” y quitarla de las opciones generales.

Si actualizamos el maestro, cuando se actualizara el esclavo? Después de pasar el tiempo **Refresh**. Si queremos actualizarlo seguido, debemos decírselo al maestro para que avise al esclavo. Para eso debemos hacer un pequeño cambio. En el archivo maestro **/etc/bind/named.conf.options**, en la sección options, donde teníamos puesto "**notify no**", debemos de poner:

```
notify yes;
```

A quien se notifica? Se les notifica a los servidores DNS que están en las zonas modificadas. Por lo tanto, debemos poner también el servidor esclavo en los archivos de las zonas. Debajo del registro actual NS pondremos el esclavo:

```
IN NS oka.nire-eskola.net.
```

Con todo esto le hemos dicho que la maquina **oka** tiene permiso para transferir las zonas. Recarguemos pues las zonas en los dos servidores.

```
rndc reload
```

Listo, en marcha!

Probémoslo. Cambiar el archivo de resolución directa de la zona y poner al final:

```
_jabber._tcp.nire-eskola.net. 86400 IN SRV 5 0 5269 jabber.nire-eskola.net.
_xmpp-server._tcp.nire-eskola.net. 86400 IN SRV 5 0 5269 jabber.nire-eskola.net.
_xmpp-client._tcp.nire-eskola.net. 86400 IN SRV 5 0 5222 jabber.nire-eskola.net.
```

Hemos utilizado los registros **SRV** (Service). Ahora cambiaremos el numero serial (ampliar). Para cargar el archivo de área poner la orden siguiente:

```
rndc reload
```

Y ahora que? Hacer preguntas al servidor. Primero al maestro.

```
dig @urumea.nire-eskola.net _jabber._tcp.nire-eskola.net any +short
```

La respuesta deberá ser.

```
10 0 5269 jabber.nire-eskola.net.
```

Haremos lo mismo preguntándole al esclavo. Responde? Estupendo. Si no responde entonces tenemos un problema.

AVISO: la tarea de avisar (notify) no la hace solamente el maestro. Los esclavos también deben hacerlo, y si existen dos esclavos, estos se avisan mutuamente. En el caso de que los dos se actualicen del mismo maestro, esta forma de actuar no es aconsejable. Por lo tanto, en los esclavos, en la sección de opciones pondremos, `notify no;`. Por otro lado, en el maestro también, en **localhost** y en las zonas inversas **0.0.127.in-addr.arpa**, como no hay actualizaciones pondremos la opción "`notify no;`", pero solamente

DENTRO DE ESTAS ZONAS!

8.3. Seguridad

Hasta ahora hemos hecho las transferencias de zona sin autentificar el maestro y el esclavo. Esto es, nos fiamos de que una maquina con otro IP es nuestro servidor DNS. Pero si alguien nos roba la IP? A esto se le llama **IP spoofing**. Por medio de esta técnica puede rellenar nuestros registros con datos falsos y enviar a nuestros clientes a lugares falsos (DNS poisoning). Para que esto no ocurra, y para autentificar los servidores, utilizaremos claves **TSIG**.

Al archivo que contiene la clave le llamaremos **dns.nire-eskola.net**. Recordar que si se instala de nuevo el sistema, o el servidor DNS, o si este ultimo es cambiado de maquina… De nuevo deberemos crear una clave (y pasarla al otro servidor), o copiar la misma.

```
dnssec-keygen -a HMAC-MD5 -b 128 -n HOST dns.nire-eskola.net
```

El resultado será algo así:

```
Kdns.nire-eskola.net+157+60939
```

En el directorio de trabajo se ha creado un archivo de texto. El nombre del archivo será el que nos ha aparecido en la pantalla, más "`.key`". En nuestro caso, `Kdns.nire-eskola.net.+157+60939.key`. Dentro de este archivo nos aparece algo parecido a esto (sólo como ejemplo)

```
dns.nire-eskola.net IN KEY 512 3 157 0mYzjU4Fnp7vbwldkJJsdw==
```

El ultimo campo es una clave de 128 bits codificada en base64. Cada vez que se crea una clave, esta sección será distinta. Debemos poner esta clave en los dos servidores DNS.

En el maestro, en el archivo **/etc/bind/named.conf.options**, definiremos la clave y para las transferencias de zona sólo aceptaremos las peticiones que vengan con esta clave. Encima de la sección de opciones pondremos lo siguiente:

```
key dns.nire-eskola.net {
      algorithm hmac-md5;
      secret "0mYzjU4Fnp7vbwldkJJsdw==";
      };
```

En la sección de opciones (**options**), por ultimo, quitaremos los **allow-transfer** y los IP de los esclavos que teníamos; en su lugar pondremos:

```
allow-transfer { key dns.nire-eskola.net; };
```

De esta forma, solo aceptará las peticiones de transferencia de zona, firmadas con esa

clave.

En el esclavo, en el archivo **/etc/bind/named.conf.options**, al principio, pondremos la definición de la clave:

```
key dns.nire-eskola.net {
      algorithm hmac-md5;
      secret “0mYzjU4Fnp7vbwldkJJsdw==”;
      };
```

Luego, después de la sección de opciones, pondremos lo siguiente:

```
server 172.16.2.8 {
      keys dns.nire-eskola.net;
};
```

Listo, reiniciamos los dos servidores y Buena Suerte!

AVISO: Cuidado con la hora de los servidores. Con una diferencia de 8 minutos no funciona y además el aviso no es muy claro. El BIND 9.2.2 da el siguiente aviso “`request has invalid signature: tsig verify failure`”. El BIND 9.3.1sin embargo da el siguiente aviso “`failure trying master 172.16.2.8#53 (source 0.0.0.0.#0): clocks are unsynchronized`” . Este ultimo es mucho más claro. BIND 9.3.4 vuelve a mostrar el mismo mensaje que BIND 9.2.2, pero añade al final la coletilla “**(BADTIME)**”, lo cual deja bastante claro el origen del problema.

NOTA: Si tenemos diferentes esclavos para diferentes zonas, conviene utilizar diferentes claves.

8. Actualizaciones dinámicas desde DHCP

Si tenemos el servidor DHCP de ISC entonces podremos actualizar automáticamente el servidor DNS. Las IP de los servidores serán fijas y nosotros mismos las introduciremos en el servidor DNS, pero dar de alta las IPs de los clientes a mano es muy aburrido. Además, normalmente no necesitamos conectarnos a un cliente. Por eso es mejor hacerlo automáticamente.

Para ello debemos decirle al servidor DHCP, cual servidor DNS hay que actualizar y cómo. Deberemos decirle al DNS, quien tiene permiso para cambiar su configuración. No daremos este permiso a cualquiera. En este caso, cualquiera podría tener capacidad de contaminar nuestros registros DNS. Para autentificar el servidor DHCP utilizaremos claves. Empecemos a trabajar!

AVISO: En las versiones mas viejas que el BIND 9.3.0 no se podrá hacer una actualización dinámica en el caso de que en ese área existan maquinas actualizadas a mano. Para hacer estas pruebas se ha utilizado BIND 9.3.4.

En primer lugar crearemos la clave TSIG para firmar las actualizaciones. Al archivo donde esta la clave le llamaremos **ddns.nire-eskola.net.** Recordad que si se instalan de nuevo el sistema o el servidor DHCP, o si este ultimo se cambia de maquina..., hay que crear de nuevo la clave (y pasárselo al servidor DNS), o copiar la misma.

Recordad: Si la clave la hemos creado anteriormente, no necesitamos realizar este paso.

```
dnssec-keygen -a HMAC-MD5 -b 128 -n HOST ddns.nire-eskola.net
```

El resultado será algo parecido a esto:

```
Kddns.nire-eskola.net+157+60939
```

En el directorio de trabajo se ha creado un archivo de texto. El nombre de este archivo será el que nos ha aparecido en pantalla más "`.key`". En nuestro caso, `Kddns.nire-eskola.net.+157+60939.key`. Dentro de este archivo nos aparecerá lo siguiente (recordad que son sólo ejemplos):

```
ddns.nire-eskola.net IN KEY 512 3 157 0mYzjU4Fnp7vbwldkJJsdw==
```

Ahora configuraremos el servidor DHCP. En la sección general del archivo **/etc/dhcp3/dhcpd.conf**, añadiremos las siguientes líneas. En la primera parte pondremos el comportamiento. Luego la clave para autorizar las actualizaciones en el servidor DNS. Por ultimo la lista de zonas por actualizar y sus características. En concreto las zonas de búsqueda directa e inversa.

```
ddns-uptades on;
ddns-update-style interim;
ddns-domainname “nire-eskola.net”;
ddms-rev-domainname “in-addr.arpa”;
ignore client-updates;

key ddns.nire-eskola.net {
      algorithm hmac-md5;
      secret “0mYzjU4Fnp7vbwldkJJsdw==”;
}

zone nire-eskola.net { primary 127.0.0.1; key ddns.nire-eskola.net; }
```

Luego, dentro de la sección de cada subred, ponemos la zona a actualizar y la clave a utilizar. En la subred 1:

```
zone 201.168.192.in-addr.arpa {
      primary 127.0.0.1;
      key ddns.nire-eskola.net;
}
```

En la subred 2:

```
zone 202.168.192.in-addr.arpa {
      primary 127.0.0.1;
      key ddns.nire-eskola.net;
}
```

Por último configuraremos el servidor DNS para que acepte las actualizaciones dinámicas enviadas desde el servidor DHCP. Primero definiremos la clave. La pondremos en el fichero **/etc/bind/named.conf.options**, debajo de la otra clave que usamos para autentificar al esclavo:

```
key ddns.nire-eskola.net {
      algorithm hmac-md5;
      secret “0mYzjU4Fnp7vbwldkJJsdw==”;
};
```

Luego en cada zona, en el fichero **/etc/bind/named.conf.local**, indicaremos qué política utilizaremos para hacer las actualizaciones y qué registro actualizar. Veámoslo. En las áreas directas, tendrá permiso para cambiar los registros A y TXT. Los registros TXT son necesarios para saber luego, al borrar, quién los creó.

```
	zone "nire-eskola.net" {
		type master;
		file "nire-eskola.net.hosts";
		update-policy {
			grant ddns.nire-eskola.net wildcard *.nire-eskola.net. A TXT; };
	};
```

En las áreas inversas solo se pueden cambiar los registros PTR. Finalizado. Para probarlo, enciende y apaga máquinas que obtienen la dirección IP por medio del servicio DHCP.

```
	zone "201.168.192.in-addr.arpa" in {
		type master;
		file "192.168.201.rev";
		update-policy {
			grant ddns.nire-eskola.net wildcard *.201.169.192.in-addr.arpa.
PTR; };
	};

	zone "202.168.192.in-addr.arpa" in {
		type master;
		file "192.168.202.rev";
		update-policy {
			grant ddns.nire-eskola.net wildcard *.202.169.192.in-addr.arpa.
PTR; };
	};
```

IMPORTANTE: BIND tiene que crear ficheros “journal” con las actualizaciones de zonas. Para ello, necesita escribir en el directorio **/etc/bind/**. Lo permitiremos con el siguiente comando:

```
	chmod g+w /etc/bind
```

A disfrutar!

9. Otras herramientas

9.1. gbindadmin

gbindadmin es un interfase gráfico para gestionar servidores DNS. Permite la creación de zonas, la generación de claves, el arranque y parada del servicio, y la carga de zonas. Además, permite definir servidores de correo, y de DNS. Es un poco limitado, pero puede ser de ayuda al principiante.

10. Referencias

Para más información, en DEBIAN, mirar en **/usr/share/doc/bind9** y **/usr/share/doc/bind9-doc**.

Lo que nunca hay que olvidar:

```
man named.conf
man gbindadmin
man nslookup
man dig
man host
```

Libros:

Essential System Administration, AEleen Frisch

TCP/IP Network Administration, Craig Hunt

DNS & BIND, Paul Albitz & Cricket Liu

DNS & BIND Cookbook, Cricket Liu

En la web:

Los RFC1034 y RFC1035 se encuentran en:

ftp://ftp.rfc-editor.org/in-notes/rfc1034.txt

ftp://ftp.rfc-editor.org/in-notes/rfc1035.txt

11. Autor

Alfredo Barrainkua Zallo

Iurreta Institutuko Sare Administraria

alfredobz@iurreta-institutua.net