



ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 24 – NOVIEMBRE DE 2009

“RESOLUCIÓN DE NOMBRES Y COMPARTICIÓN DE RECURSOS”

AUTORIA JOSÉ JIMÉNEZ CALERO
TEMÁTICA SISTEMAS OPERATIVOS MULTIUSUARIO. REDES
ETAPA CICLOS FORMATIVOS

Resumen

En este artículo se realiza una descripción de cómo configurar el servidor de nombres DNS lo que nos permite acceder a cualquier servidor en nuestra red mediante el empleo de un nombre en lugar de la dirección, así como realizar la compartición de archivos y otros recursos usando NFS y/o SAMBA.

Palabras clave

Servicio de nombres DNS, compartición de recursos SAMBA-NFS.

RESOLUCIÓN DE NOMBRES. INTRODUCCIÓN.

El servicio de nombres de dominio (Domain Name Service, DNS) es una base de datos distribuida que soporta delegación de autoridad para segmentos del espacio de claves. Se usa principalmente para proporcionar una correlación entre nombres de máquina y direcciones IPv4; también ofrece correlaciones entre direcciones y nombres de máquina (registros A), información sobre máquinas, soporte especial de encaminamiento para correo (registros MX), etc. Utiliza algunos tipos de registros internos para definir su propia estructura jerárquica interna, la delegación de subdominios; además, existen características para el soporte de IPv6.

Se trata de un sistema que permite traducir de nombre de dominio a direcciones IP y viceversa. Aunque Internet sólo funciona en base a direcciones IP, el DNS permite que usemos nombres de dominio que son bastante más simples de recordar. El sistema de nombres de dominios en Internet es un sistema distribuido, jerárquico, replicado y tolerante a fallos. Aunque parece muy difícil lograr todos esos objetivos, la solución no es tan compleja en realidad. El punto central se basa en un árbol que define la jerarquía entre los dominios y los subdominios. En un nombre de dominio, la jerarquía se lee de derecha a izquierda. Por ejemplo, en blabla.prueba.com, el dominio más alto es com. Para que exista una raíz del árbol, se puede ver como si existiera un punto al final del nombre: www.midominio.com., y todos los dominios están bajo esa raíz (también llamada “punto”).

Cada componente del dominio (y también la raíz) tiene un servidor primario y varios servidores secundarios.



ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 24 – NOVIEMBRE DE 2009

Todos estos servidores tienen la misma autoridad para responder por ese dominio, pero el primario es el único con derecho para hacer modificaciones en él. Por ello, el primario tiene la copia maestra y los secundarios copian la información desde él.

La raíz del sistema de dominios es servida por algunos servidores raíz principales (son aproximadamente trece en todo el mundo, de los cuales uno se encuentra en España). Todo servidor de nombres debe ser configurado con la lista de los servidores raíz principales. Estos servidores dicen qué dominios de primer nivel existen y cuales son sus servidores de nombres. Recursivamente, los servidores de esos dominios dicen qué subdominios existen y cuales son sus servidores.

DNS se transmite tanto sobre TCP como sobre UDP, en el puerto 53. Las peticiones más corrientes son sobre UDP. TCP se usa cuando el total de una consulta excede los 512 bytes, o para llevar a cabo "transferencias de zonas". Podemos destacar dos paquetes encargados de las gestiones DNS bajo Linux, 'bind' y 'djbdns'. Bind es probablemente el servidor DNS más extendido, pero a la vez el más complejo. Además, a lo largo de su historia han aparecido grandes problemas de seguridad en él, por lo que en algunas ocasiones no es aconsejable su uso. Djbdns, por el contrario, es un paquete mucho más simple que bind (posee menos funcionalidades), aunque en la mayoría de ocasiones nos resultarán suficientes. Se diseñó pensando en la sencillez y la seguridad.

En este apartado trabajaremos con djbdns. En primer lugar veremos como montar un caché DNS haciendo uso de dnscache (un componente del paquete djbdns), y posteriormente configuraremos tinydns (otro componente de djbdns) para que se encargue de gestionar un hipotético dominio nuestro, por ejemplo midominio.com.

Antes de instalar el paquete djbdns, necesitaremos los paquetes daemontools (se encargan de gestionar demonios, de una forma similar a inetd) y ucspi-tcp (control de acceso tcp, trabajando de forma similar a los tcpwrappers), ambos escritos por el mismo creador de djbdns. Una vez tengamos ambos instalados, haremos lo mismo con el paquete djbdns, que contendrá todo lo necesario para poner en marcha nuestro propio servidor de nombres.

Es importante destacar que debido al tipo de licencia en que Dan J. Bernstein (creador de djbdns, entre otros) libera sus aplicaciones (no permite distribuciones binarias de su código), tendremos que descargar sus fuentes y proceder luego a su compilación. Debian nos permitirá abordar esto de una forma más sencilla.

Por ejemplo, la instalación del paquete ucspi-tcp se haría de la siguiente forma:

```
# apt-get install ucspi-tcp
```

(Instalaría el código fuente de ucspi-tcp en nuestra máquina, preparándolo para su compilación)

```
# build-ucspi-tcp
```



ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 24 – NOVIEMBRE DE 2009

(Compilaría el código que descargamos en el paso anterior, generando una vez terminado un paquete en formato deb; ahora instalaríamos el paquete como haríamos normalmente en Debian)

```
# dpkg -i ucspi-tcp-xxxx.deb
```

Como podemos observar, el procedimiento es muy sencillo. Los paquetes daemontools y djbdns se instalarían de una forma similar.

Puesta en marcha de un cache DNS con djbdns

Dnscache es el componente de djbdns encargado de la caché DNS. El uso de un caché DNS puede ahorrar mucho tráfico de red. Imaginemos por un momento que en la red de nuestra oficina hay unos cien equipos.

Un equipo de la red realiza una petición Web a Google, por lo que previamente habrá tenido que resolver su dirección (mediante una consulta DNS); seguidamente otros dos equipos de la red desean visitar la misma página Web. Es inevitable pensar que sería interesante disponer de algún mecanismo que almacenase esas consultas DNS, de forma que esa información estuviese disponible para otros usuarios, evitando así tráfico innecesario en nuestra red y una parte de nuestro preciado ancho de banda. El componente encargado de realizar eso es la caché DNS, más concretamente en nuestro caso dnscache, que forma parte de djbdns.

A continuación veremos cómo poner en marcha dnscache.

Una vez que tengamos instalado el software djbdns, nos dispondremos a configurar nuestra caché DNS. Para ello utilizaremos el comando dnscache-conf.

```
# dnscache-conf dnscache dnslog /etc/dnscache 192.168.0.1
```

donde 192.168.0.1 es la IP en la que dnscache escuchará. Por defecto, los ficheros de bitácora (logs) residen en /etc/dnscache/log/main. Si queremos que dichos registros se almacenen en /var/log/dnscache, debemos crear dicho directorio con propietario dnslog y sustituir ./main en /etc/dnscache/log/run por /var/log/dnscache.

Iniciaremos dnscache informándole a svscan acerca de él:

```
# ln -s /etc/dnscache /service
```

dnscache debería arrancar antes de 5 segundos. Si deseamos únicamente hacer uso local de dnscache ya hemos terminado, ya que por defecto dnscache no acepta consultas desde otras máquinas. Si por el contrario deseamos que toda nuestra red haga uso de él (como es nuestro caso, ya que deseamos que toda la oficina haga uso de él), deberemos detallarle qué red o ips podrán hacer uso de él. Para ello añadiremos las direcciones correspondientes a /etc/dnscache/root/ip. En nuestro caso, deseamos que toda nuestra red (192.168.0.0/24) pueda utilizarlo. El comando a ejecutar sería el siguiente.

```
# touch /etc/dnscache/root/ip/192.168.0
```



ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 24 – NOVIEMBRE DE 2009

Podemos añadir o eliminar redes sin tener que informar de ello al dnscache que está en ejecución.

Resolución de nombres con djbdns

Tinydns es el componente encargado de realizar esta tarea. En este apartado vamos a montar un servidor de nombres de dominio que gestionará nuestro recién adquirido dominio, 'midominio.com'.

Deseamos que nuestro servidor de nombres pueda resolver www.midominio.com, para nuestro portal Web, ftp.midominio.com, para subir ciertos archivos, y mail.midominio.com, que será nuestro 'mail exchanger' (nuestro registro MX).

Para configurar tinydns haremos uso del comando tinydns-conf. Ejecutaremos la siguiente orden:

```
# tinydns-conf tinydns dnslog /etc/tinydns 80.150.10.2
```

Pues bien, 192.168.0.2 es la IP en la que tinydns debe escuchar. Es importante destacar que no podemos utilizar la misma dirección que usamos cuando montamos nuestra caché DNS. Por defecto, los ficheros de bitácora habitan en /etc/tinydns/log/main. No deberíamos tener nada distinto a ficheros de configuración en /etc; al igual que en el caso anterior, podemos modificar donde se guardarán los registros. Para almacenarlos en /var/log/tinydns, crearemos este directorio con propietario dnslog y sustituiremos ./main en /etc/tinydns/log/run por /var/log/tinydns.

Arrancaremos tinydns informándole a svscan de su existencia:

```
# ln -s /etc/tinydns /service
```

tinydns ha de arrancar antes de 5 segundos.

Ahora configuraremos los nombres que tinydns resolverá. Dicha configuración se almacena en /etc/tinydns/root/data. Escribiremos este fichero para que se encargue de la gestión de nuestro dominio.

El resultado sería el siguiente:

```
# definimos el servidor de nombres autorizado
.midominio.com::ns1.midominio.com
# y el intercambiador de correo
@midominio.com::mail.midominio.com
# las direcciones para las máquinas www, ftp
=www.midominio.com:80.150.10.1
=ftp.midominio.com:80.150.10.2
# la máquina ftp también es ns1, servidor de nombres
+ns1.midominio.com:80.150.10.2
# y también nuestro servidor de correo
```



ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 24 – NOVIEMBRE DE 2009
+mail.midominio.com:80.150.10.2

Tras editarlo, nos situaremos en el directorio /etc/tinydns/root y ejecutaremos "make". De esta forma compilaremos el fichero data.cdb que lee tinydns. A partir de este momento, nuestro servidor de nombres, que escucha en la dirección 80.150.10.2, resolverá las direcciones de nuestro dominio.

Como hemos visto, djbdns es un paquete sencillo pero completo, que nos permitirá crear nuestro propio servidor de nombres de una forma rápida, cómoda y segura.

COMPARTICIÓN DE RECURSOS. INTRODUCCIÓN.

El sistema de ficheros de red (NFS - Network File System) nos permite hacer uso de unidades compartidas por sistemas remotos.

Antes de configurar NFS en nuestras máquinas, pondremos a punto los requisitos previos. En primer lugar, tendremos que decidir si nuestra máquina actuará como cliente NFS, servidor, o ambos. En caso de que necesitemos ejercer de clientes, tendremos que habilitar el soporte NFS en el núcleo de nuestro sistema. Por el contrario, si sólo queremos exportar algún directorio, no lo necesitaremos. Aunque como hemos mencionado no necesitamos compilar este soporte para actuar de servidores (ya que el paquete nfs-userserver gestiona el servicio correctamente), es recomendable hacerlo, ya que la solución basada en el kernel es más rápida.

La primera opción de configuración que aparece sería necesaria para conectarnos a un servidor NFS, mientras que la segunda es recomendable para ofrecer el servicio. Podremos compilarlo tanto de forma estática como módulo.

Una vez tengamos el soporte correspondiente añadido al núcleo, instalaremos los paquetes necesarios. En Debian, los paquetes encargados del servicio nfs son:

- nfs-common

Soporte necesario tanto para el cliente como para el servidor.

- nfs-user-server

Nos aporta lo necesario para montar un servidor NFS sin necesidad de soporte en el núcleo.

- nfs-kernel-server

Nos aporta lo necesario para montar un servidor NFS basado en el núcleo (requiere añadir soporte). En nuestro caso haremos uso de nfs-common y nfs-user-server. Instalaremos dichos paquetes haciendo uso de la potente herramienta apt-get.



ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 24 – NOVIEMBRE DE 2009

Como podemos observar en la figura, el paquete portmap se encuentra entre las dependencias de nfs-userserver. Portmap es necesario para realizar RPCs (Remote Procedure Calls) al sistema, por lo que será necesario que esté instalado y ejecutándose antes de lanzar el servidor NFS.

Ahora que tenemos todo lo necesario, veamos las configuraciones del cliente y el servidor.

Configuración del servidor

La configuración de nuestro servidor se realiza en el fichero `/etc/exports`. En él decidiremos qué deseamos compartir y a quien.

Con esto estamos indicando que vamos a exportar el directorio `/home/mp3`, permitiendo acceso a los equipos pertenecientes a la red `192.168.0.0/24`, en modo de lectura y escritura.

Nuestros clientes ya pueden hacer uso de nuestro recurso compartido. Esta configuración que hemos empleado es un caso extremadamente sencillo. Puede encontrar mucha más información sobre el archivo `/etc/exports` en la página del manual `'man exports'`.

Configuración del cliente

Hacer uso de un recurso compartido mediante NFS es sencillo. Bastará con que montemos dicho espacio como si se tratase de otro dispositivo, como un cdrom o un disquete.

Una vez montado, trabajaremos con esa ruta como si se tratase de una ruta local, y NFS se encargará de trabajar bajo los telones, resultando un proceso transparente para al usuario.

Si lo deseamos, podemos añadir el recurso compartido al fichero `fstab` para que se monte de forma automática al iniciar la máquina.

Para ver algunas opciones que podemos usar durante el montaje, podemos consultar la página del manual del comando `mount`, concretamente en el apartado que hace referencia a `nfs`.

Sería muy interesante restringir el uso de estos servicios únicamente a las máquinas que deseemos. Para ello podríamos hacer uso de los `tcp wrappers`, indicando en los archivos `hosts.allow` y `hosts.deny` las máquinas que podrán establecer conexiones con nuestro servidor.

Instalando y configurando samba

En este apartado veremos la manera de usar el paquete Samba, que dota a Linux de soporte para el protocolo Session Message Block (SMB), también llamado NetBIOS o LanManager. El protocolo SMB es usado por los sistemas operativos de Microsoft para compartir discos e impresoras. Usando el paquete de herramientas Samba, creado por Andrew Tridgell, las máquinas UNIX (incluyendo Linux) pueden compartir discos e impresoras con servidores Windows.



ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 24 – NOVIEMBRE DE 2009

Entre las cosas que podemos realizar con samba, destacaremos las siguientes:

- Compartir una unidad de Linux con máquinas Windows.
- Compartir una unidad de Windows con máquinas Linux.
- Compartir una impresora de Linux con máquinas Windows.
- Compartir una impresora de Windows con máquinas Linux.

En este punto veremos detallado alguno de estos casos, para comprender mejor el funcionamiento de samba. Antes de comenzar instalaremos el software necesario. En Debian, bastará con ejecutar como root el siguiente comando:

```
# apt-get install samba smbclient smbfs
```

Esto nos instalará todo lo necesario para compartir recursos y acceder a otros recursos compartidos por máquinas Windows.

La instalación de samba nos preguntará en primer lugar el grupo de trabajo al que pertenecerá nuestra máquina.

Posteriormente, nos preguntará si deseamos utilizar contraseñas cifradas, a lo que responderemos afirmativamente.

En tercer lugar, tendremos que elegir el modo de ejecución de samba: bien en forma de demonios independientes o a través de inetd. A través de inetd podemos establecer algunas restricciones sobre el uso del servicio, pero el uso de demonios independientes es más eficiente en cuanto a rendimiento se refiere. En nuestro caso elegiremos demonios independientes.

En el siguiente paso se nos ofrecerá la posibilidad de crear la base de datos de las contraseñas encriptadas. Si seleccionamos hacer uso de la encriptación deberemos crearla. Si optamos por no hacerlo, hemos de tener en cuenta que habrá que configurar las máquinas Windows para que hagan uso de claves en texto plano.

De esta forma habrá concluido la instalación de samba. Ahora sólo nos queda ajustar el archivo de configuración a nuestras necesidades. En este tema trataremos algunos casos de ejemplo con samba, pero nos resultaría imposible describir todas sus funcionalidades. Sería muy recomendable, por tanto, que se leyeran las páginas del manual para obtener una información mucho más completa

Para iniciar samba escribiremos:

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

Compartiendo un directorio a una red Windows



ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 24 – NOVIEMBRE DE 2009

El fichero de configuración de samba es smb.conf, que se encuentra bajo el directorio /etc/samba (en Debian). Todo lo referente a la configuración de samba se gestiona en este fichero, así como todos los recursos que deseemos compartir.

Entre las opciones generales más importantes que podemos encontrarnos en el apartado global del fichero de configuración, cabe destacar:

NETBIOS NAME. Nombre netbios que le asignaremos a nuestra máquina.

SERVER STRING. Descripción de nuestro servidor.

WORKGROUP. Grupo de trabajo al que perteneceremos.

Estas son sólo las opciones más básicas. Samba dispone de más de 200 (suficientes para escribir un manual completo sobre él), por lo que se recomienda la lectura de un documento más especializado si deseamos profundizar en la materia.

Para compartir un directorio, por ejemplo /home/ftp, añadiremos lo siguiente a nuestro fichero de configuración:

```
[ftp]
comment = Archivos públicos de nuestro ftp
path = /home/ftp
read only = no
writable = yes
public = yes
```

Una vez que hayamos añadido lo siguiente, deberemos forzar a samba a que relea su archivo de configuración:

```
# /etc/init.d/samba reload
```

Ahora las máquinas de nuestra red podrán acceder al recurso compartido por nosotros (ftp). Según hemos establecido con las opciones propuestas, todos los usuarios podrán acceder al recurso compartido, así como modificar su contenido.

Existen algunas interfaces gráficas, como swat, que pueden facilitarnos la labor de compartir nuevos recursos, así como de gestionar las opciones generales de configuración.



ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 24 – NOVIEMBRE DE 2009
APLICACIÓN DIDÁCTICA.

Teniendo en cuenta el currículo de Educación Secundaria los contenidos de este artículo pueden usarse en el 4º curso de la ESO en la asignatura Informática Aplicada, asimismo también son aplicables para los módulos “Sistemas Informáticos Monousuario y Multiusuario” y “Sistemas Multiusuario y en Red” de los Ciclos Formativos de Grado Superior “Administración de Sistemas Informáticos” y “Desarrollo de Aplicaciones Informáticas” respectivamente.

BIBLIOGRAFÍA.

Pérez Agudín, J. (2009). *Debian GNU/LINUX*. Madrid: Anaya Multimedia.

Adelstein, T. y Lubanovic, B. (2007). *Administración de Sistemas Linux*. Anaya Multimedia.

Schroder, C. (2008). *Redes en Linux: Guía de Referencia*. Madrid: Anaya Multimedia.

Autoría

- Nombre y Apellidos: JOSÉ JIMÉNEZ CALERO
- Centro, localidad, provincia: CÓRDOBA.
- E-MAIL: jose.jimenezcalero@gmail.com