

Cómo instalar un sencillo servidor DNS

(c) Guillermo Ballester Valor
gbv@oxixares.com
Ogijares, Granada, España

Version 0.1.1 (01-Sep-2003)
Este documento tiene [licencia GPL](#).

Sumario:

- [1.- Introducción](#)
 - [Servidor cache DNS.](#)
 - [Servidor maestro de un dominio.](#)
 - [2.- Instalación del software](#)
 - [2.1 Desde RPM](#)
 - [2.2 Desde código fuente.](#)
 - [3.- Configuración](#)
 - [3.1 Ficheros básicos de configuración.](#)
 - [4.- Configuración final](#)
 - [5.- Bibliografía](#)
-

1.- Introducción

Cuando indicamos a nuestro navegador Web que, por ejemplo, vaya a una página en **www.gnu.org**, éste tiene que realizar algunas tareas antes de saber dónde buscar.

Como es bien sabido, los ordenadores conectados a Internet se identifican por un número, la dirección **IP**. En este aspecto funciona igual que los teléfonos. Si queremos llamar a alguien cuyo nombre y apellidos conocemos, o bien sabemos de antemano el número, o se tiene que consultar en una guía o listín telefónico. En internet es lo mismo, de alguna forma el navegador tiene que averiguar qué número tiene el servidor de cada una de las páginas o documentos que se consultan. ¿Cómo se hace?

Sin entrar en detalles, básicamente diremos que nuestro ordenador realiza los siguientes pasos

1. Se consultan los ficheros **/etc/nsswitch.conf** y **/etc/host.conf**. Si la librería del sistema utilizada es muy antigua (libc4 o anterior) entonces solamente se consulta el segundo fichero. Para libc5 y librería GNU C 2.x se lee el primero. Si existen ambos ficheros entonces deben de ser coherentes. En esencia, indican

el orden que hay que seguir en la búsqueda. En el fichero **/etc/nsswitch.conf** debemos tener una línea

```
hosts: files dns
```

De existir **/etc/host.conf** deberíamos tener, por ejemplo, la línea

```
order hosts, bind
```

Ambos ficheros indican que se busque primero en los ficheros locales, normalmente en el fichero **/etc/hosts** y después se realice una **consulta DNS**, los términos *dns* o *bind* se refieren a éste último tipo de consulta.

2. Suponiendo el orden descrito en el punto anterior, mira si en el fichero **/etc/hosts** existe una línea con la forma

```
dirección_ip servidor_buscado
```

Si esto es así la búsqueda ha terminado, pero no es lo habitual. De esta forma se recurre a la consulta DNS

3. La consulta DNS comienza con la inspección del fichero **/etc/resolv.conf**. En este fichero tenemos una serie de líneas con el formato

```
nameserver dirección_ip
```

Es la lista de servidores DNS que tuvimos que implementar en su día para poder acceder a la red, y que nos facilitó nuestro Proveedor de Servicios de Internet. Primero envía una consulta al primero de ellos (DNS primario), en caso de fallo al secundario y así normalmente tenemos hasta 3 servidores.

4. Cada uno de los servidores DNS consultados, busca si en su memoria caché por casualidad existe esa información y todavía es válida, en cuyo caso la sirve rápidamente. En caso negativo consulta a otros servidores de jerarquía superior, los cuales a su vez extienden la búsqueda a otros superiores en caso de fallo. Así hasta llegar a alguno de los servidores raíces en Internet. Si los servidores raíz no conocen el nombre, entonces se devuelve un fallo de búsqueda siguiendo el camino inverso. Si los datos del dominio buscado se hallan en el cache de alguno de los servidores, entonces se responde con la información ahí almacenada. En caso negativo, si el dominio es válido, un servidor raíz dirigirá la solicitud al **servidor maestro** del dominio. Estos datos, al ser devueltos al solicitante, irán actualizando los cachés de los servidores por los que pasa la información.

Llegados a este punto, conviene aclarar que el término **servidor DNS** suele utilizarse para dos tipos de servicios totalmente distintos:

- **Servidor caché DNS.** Traductor de URL a direcciones IP y viceversa. Es el servicio habitual, todos los Proveedores de Servicios de Internet deben suministrar varios a sus clientes. De esta forma, cuando un ordenador necesita traducir nombres a IP (o viceversa) se recurre a estos servidores. Si logramos implementar con éxito el servidor siguiendo las instrucciones de este documento, habremos instalado uno de esos servidores (pero normalmente para uso exclusivo de nuestra red local).
- **Servidor maestro de un Dominio.** Supongamos que hemos registrado el dominio *mi_dominio.com*. Uno de los requisitos para registrarlo ha sido suministrar la dirección IP del servidor DNS que contiene los datos del dominio. Este servidor, en la IP facilitada al registrar el dominio, actúa como maestro. Si modificamos en él un dato, se propagará esta modificación por la Internet. De forma muy básica, cuando alguien en la red busca *mi_dominio.com* por primera vez, ningún servidor cache tendrá conocimiento de nuestro nuevo dominio, salvo los servidores raíces (en este caso los que controlan **.com**) que saben cuál es el servidor maestro del dominio. De esta forma, el servidor DNS del visitante tiene que llegar a nuestro servidor maestro, recoger la información del dominio requerida y pasarla al solicitante. Al mismo tiempo, este servidor guardará la información en el caché, de forma que si de nuevo se produce otra solicitud de información de *mi_dominio.com*, ya la tiene en caché, y puede servirla sin consultarla al servidor maestro del dominio. Es interesante notar que, dentro de la información que se recoge de un dominio, está el tiempo de expiración tras el cuál la información del cache ya no es válida.

El objetivo de este documento es, fundamentalmente, instalar nuestro servidor local como servidor DNS primario funcionando como servidor caché, de forma que las consultas a lugares habituales se resuelven en pocos milisegundos en vez de superar, en algunos casos, una espera superior al segundo. La configuración como servidor maestro de algún dominio es algo más complicada, pero se darán indicaciones de cómo hacerlo.

2.- Instalación del software.

Como siempre, resulta mucho más sencillo utilizar los paquetes que ya vienen preparados en las distribuciones, aunque puede configurarse y compilarse desde el código fuente. El nombre usual del paquete en las distribuciones es **bind9**. Es un código desarrollado por la [Internet Software Consortium](#). La última versión puede conseguirse [aquí](#)

A continuación daremos dos breves descripciones de cómo

puede instalarse el software, la primera a través de RPM, utilizando como ejemplo la distribución SuSE 8.x. La segunda, compilando e instalando directamente desde el código fuente.

2.1 - Desde RPM.

En la distribución SuSE 8.0 y 8.1 los paquetes a instalar son **bind9** y **bind9-utils**. Si ya tiene instalados la versión bind8, entonces es conveniente desinstalar previamente el paquete

```
#rpm -e bind8
```

Si se utiliza Yast2 para la instalación de paquetes, los pasos para instalar serían

Lanzar yast2 como root:

```
#yast2
```

Elegir los paquetes a instalar,

```
software-> Instalar/desinstalar software  
->Red/servidor
```

Se seleccionan **bind9** y **bind9-utils**, se pulsa el botón aceptar y ya se tiene el software instalado. Pero todavía hace falta configurarlo para que funcione. Pasar al apartado 3.

En el caso de utilizar **apt-get** para actualizar paquetes, como root

```
#apt-get install bind9 bind9-utils
```

Una vez instalado el software, hay que configurarlo, lo cual se describe en el apartado 3.

2.2 - Desde código fuente.

Si se utiliza el código fuente, el método de instalación es el habitual. Desde [aquí](#) podemos conseguir la última versión (suponemos en los ejemplos la 9.x.x). Desde un directorio descomprimos el fichero

```
> tar -xvzf bind-9.x.x.tar.gz
```

si está comprimido con gzip. En caso de que la compresión sea con bzip2

```
> tar -xvjf binf-9.x.x.tar.bz2
```

Se cambia al directorio creado,

```
> cd bind-9.x.x
```

Deberíamos leer el fichero README en ese directorio. El paso siguiente es configurar, podemos ver las opciones de configuración con

```
> ./configure --help
```

Una vez elegidas las opciones, se configura

```
> ./configure
```

El siguiente paso es la compilación,

```
> make
```

y por ultimo, como root, se procede a la instalación

```
# make install
```

con ello, el software quedará instalado. Ahora queda configurarlo.

3.- Configuración.

Una vez instalado el software, hay que configurarlo. En este punto, todo depende de qué configuración de red tenemos y qué tipo de servidor queremos configurar. Lo mas común es hacerlo como servidor caché DNS y quizás como maestro de algún dominio.

3.1 - Ficheros básicos de configuración.

El binario que realiza el servicio DNS se llama **named**. El fichero de configuración en muchas distribuciones es **/etc/named.conf**. El utilizado por SuSE puede servir como base. Si se utilizan los rpms de SuSE para instalar, todos estos ficheros ya estan convenientemente instalados en el directorio **/var/named**.

```
# Copyright (c) 2001 SuSE GmbH Nuernberg, Germany
#
# Author: Frank Bodammer <feedback@suse.de>
#
# Traducción y comentarios:
#      Guillermo Ballester Valor
gbv@oxixares.com
#
# /etc/named.conf
#
# Este es un ejemplo de fichero de configuración
de name para BIND9
# Funciona como servidor caché sin modificación.
#
# Un ejemplo para hacer de servidor maestro de tu
propio dominio puedes
# verlo en
```

```

/usr/share/doc/packages/bind9/sample-config.
#
# La descripción de todas las opciones posibles
puede verse en
# /usr/share/doc/packages/bind9/misc/options.

options {

    # Aquí se define el nombre del directorio
de trabajo

    directory "/var/named";

    # La Línea forwarders se puede
descomentar, cambiando los
    # IP por los que facilita el Proveedor de
Servicios de Internet
    # Si se habilita, el servidor puede pasar
la petición a los
    # servidores indicados. En general, no es
necesario habilitar
    # esta opción si en el fichero
'resolv.conf' ya están esos
    # servidores que suministra el ISP.

    #forwarders { 10.11.12.13; 10.11.12.14;
};

    # Si se habilita la siguiente línea,
antes que intentar buscar la información
    # por sí mismo, el servidor la solicita
la información a los 'forwarders'
    # definidos anteriormente

    #forward first;

    # Con la siguiente línea, se define la
interface y el puerto en las
    # que named escucha las peticiones DNS.
Por defecto escucha en todas
    # las interfaces definidas en el sistema.
El puerto definido por defecto
    # es el 53, y puede cambiarse en esta
línea.

    #listen-on port 53 { 127.0.0.1; };

    # Esta línea activa la escucha de
interfaces IPV6
    # Los valores permitidos son 'any' y
'none'
    # o una lista de direcciones IPV6 se
puede utilizar solamente
    # con el kernel 2.4 en esta versión.

    listen-on-v6 { any; };

    # Las siguientes tres líneas se pueden
necesitar si hay un firewall
    # entre el servidor e internet.

    #query-source address * port 53;
    #transfer-source * port 53;
    #notify-source * port 53;

```

```
# La línea allow-query contiene una lista
de redes o direcciones IP
# para aceptar o rechazar peticiones. La
opción por defecto es aceptar
# todas las peticiones de todas las
direcciones.
```

```
#allow-query { 127.0.0.1; };
```

```
# Si notify se pone a yes (por defecto),
se envían mensajes 'notify' a otros
# servidores cuando cambian los datos de
alguna zona. En vez de utilizar esta
# opción de forma global, es mejor
hacerlo individualmente en la definición
# de cada zona
```

```
notify no;
```

```
};
```

```
# Las siguientes tres definiciones no necesitan
ninguna modificación.
# La primera define 'localhost', la segunda la
inversa para 'localhost'
# y la tercera es la definición de los servidores
raíz "."
```

```
zone "localhost" in {
    type master;
    file "localhost.zone";
};
```

```
zone "0.0.127.in-addr.arpa" in {
    type master;
    file "127.0.0.zone";
};
```

```
zone "." in {
    type hint;
    file "root.hint";
};
```

```
# Aquí puedes introducir las definiciones de zona
para tus propios
# dominios.
```

En las definiciones de zona que hay al final del fichero anterior, se hace referencia a tres ficheros de configuración de zona. EL fichero **127.0.0.zone**

```
$TTL 1W
@           IN SOA      localhost.
root.localhost. (
                        20030325      ;
Numero de serie
                        2D             ;
refresh
                        4H             ;
retry
                        6W             ;
expiry
                        1W )          ;
minimum
```

```
1          IN NS      localhost.
          IN PTR     localhost.
```

Para el fichero **localhost.zone**

```
$TTL 1W
@          IN SOA    @      root (
                                20030325      ;
Numero de serie
                                2D           ;
refresh
                                4H           ;
retry
                                6W           ;
expiry
                                1W )        ;
minimum

          IN NS      @
          IN A       127.0.0.1
```

El fichero **root.hint** contiene la información de los servidores raíz. Puede obtenerse la última versión mediante ftp anónimo desde **Internic** [aquí](#).

La última versión es:

```
;      This file holds the information on root
name servers needed to
;      initialize cache of Internet domain name
servers
;      (e.g. reference this file in the "cache
. <file>"
;      configuration file of BIND domain name
servers).
;
;      This file is made available by InterNIC
;      under anonymous FTP as
;      file
/domain/named.root
;      on server          FTP.INTERNIC.NET
;
;      last update:      Nov 5, 2002
;      related version of root zone:
2002110501
;
;
; formerly NS.INTERNIC.NET
;
.          3600000      IN      NS
A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.      3600000      A
198.41.0.4
;
; formerly NS1.ISI.EDU
;
.          3600000      NS
B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.      3600000      A
128.9.0.107
;
; formerly C.PSI.NET
;
.          3600000      NS
C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.      3600000      A
192.33.4.12
```

```

;
; formerly TERP.UMD.EDU
;
.           3600000      NS
D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET.   3600000      A
128.8.10.90
;
; formerly NS.NASA.GOV
;
.           3600000      NS
E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET.   3600000      A
192.203.230.10
;
; formerly NS.ISC.ORG
;
.           3600000      NS
F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET.   3600000      A
192.5.5.241
;
; formerly NS.NIC.DDN.MIL
;
.           3600000      NS
G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET.   3600000      A
192.112.36.4
;
; formerly AOS.ARL.ARMY.MIL
;
.           3600000      NS
H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET.   3600000      A
128.63.2.53
;
; formerly NIC.NORDU.NET
;
.           3600000      NS
I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET.   3600000      A
192.36.148.17
;
; operated by VeriSign, Inc.
;
.           3600000      NS
J.ROOT-SERVERS.NET.
J.ROOT-SERVERS.NET.   3600000      A
192.58.128.30
;
; housed in LINX, operated by RIPE NCC
;
.           3600000      NS
K.ROOT-SERVERS.NET.
K.ROOT-SERVERS.NET.   3600000      A
193.0.14.129
;
; operated by IANA
;
.           3600000      NS
L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET.   3600000      A
198.32.64.12
;
; housed in Japan, operated by WIDE
;
.           3600000      NS
M.ROOT-SERVERS.NET.

```

```
M.ROOT-SERVERS.NET.      3600000      A
202.12.27.33
; End of File
```

Con los ficheros descritos hasta ahora, tendríamos configurado un simple servidor **cache DNS**. Si el lector no pretende crear un servidor maestro para ningún dominio, puede seguir en el [apartado 4](#).

3.2 Configuración de servidor maestro de un dominio

Normalmente, al registrar un dominio en Internet, la empresa o Agente registrador además de realizar el registro nos proporciona el servicio de **servidor maestro**. Esto es, cualquier dato referente a nuestro dominio tendrá que ser buscado en el servidor que se nos facilita. Normalmente, las empresas que nos registran el dominio nos facilitan el acceso remoto al servidor maestro para poder introducir la información deseada.

Si tengo una línea ADSL con una IP fija XXX.XXX.XXX.XXX en la que un servidor Web quiero que preste los servicios al dominio **mi_dominio.com**, tengo que incluir esa información en el servidor maestro. También puedo decirle qué máquina sirve como agente de correo para ese dominio, qué otra como servidor ftp ...

Realmente, la forma en que se incluye toda esa información es la misma para todos los servidores DNS de la Internet, lo único que varía en cada caso es la complejidad. Se puede gestionar el servidor maestro de un simple dominio compuesto por una sola máquina o la de un dominio que representa una gran red extendida por todo el mundo. Escribir en profundidad cómo se configura un DNS en todos los casos es una tarea que desborda este pequeño documento. No obstante, daremos alguna indicación de cómo configurar en casos sencillos.

Como ya se ha dicho, la configuración de **BIND9** se introduce en el fichero **/etc/named.conf**. De la lectura del [ejemplo](#) en este documento puede inferirse su funcionamiento básico. Este puede ser muy complejo, si se quiere profundizar puede consultar [este documento pdf](#) en inglés. En muchos casos, puede ser adecuada la configuración básica del ejemplo, añadiendo al final del fichero la definición de la zona o zonas para nuestro dominio.

En lo que sigue supondremos que el dominio es **mi_dominio.com** y que la dirección IP asignada es **201.202.203.204**. Ello implica que al registrar el dominio **mi_dominio.com** ha habido que suministrar esa dirección IP como dirección IP del servidor maestro. Si nuestro registrador de dominio ha suministrado otro servidor DNS entonces **NO** podemos configurar en nuestro ordenador un servidor maestro para ese dominio, pero lo que sigue puede

ayudarnos a configurar de forma remota el servidor DNS que el registrador pone a nuestra disposición.

Los datos de cada dominio suelen escribirse en ficheros separados, en el caso del ejemplo que sigue en dos ficheros "**mi_dominio.com.zone**" y "**203.202.201.zone**" ambos en el directorio `/var/named` . El primero de ellos servirá, en esencia, para que cualquiera en la internet que solicite información de **xxx.mi_dominio.com** obtenga como respuesta una dirección IP a la que dirigirse. El segundo se refiere a una búsqueda inversa: Cualquiera que solicite información acerca de qué host es el que tiene como dirección IP **201.202.203.XXX** obtendrá como respuesta un nombre de host.

```
# Aquí puedes introducir las definiciones de zona
para tus propios
# dominios.
zone "mi_dominio.com" {
    type master;
    file "zone/land-5.com.zone";
};

zone "203.202.201.in-addr.arpa" {
    type master;
    file "203.202.201.zone";
};
```

NOTA IMPORTANTE: Este segundo tipo de zona presenta frecuentemente muchos problemas. Así como generalmente disponemos de la libertad de hacer lo que queramos con el nombre de dominio que hemos registrado (colocarlo en el Host que deseemos con la IP que dispongamos), **NO** ocurre lo mismo con la resolución inversa de la IP. La diferencia es que nosotros podemos utilizar el servidor maestro de nuestro dominio y allí indicar cómo se traducen de nombres a direcciones IPs: en el dueño del dominio se da **delegado** la posibilidad de especificar registros tipo nombres->IP. En cambio, la posibilidad inversa **no** suele ser delegada. Por ejemplo, aunque dispongamos de una conexión ADSL con IP fijo, es habitual que nuestro Proveedor de Servicios tenga ya fijada una zona de resolución inversa para nuestra IP, de forma que nosotros **NO** podremos escribir una.

De todas formas, siempre puede solicitarse al Proveedor de Servicios la delegación de la resolución inversa , aunque me temo que o bien la respuesta será negativa o ello implicará mayores costes :(.

4.- Configuración final.


```
gnu.org.                86400   IN      NS      ns4.gnu.org.

;; ADDITIONAL SECTION:
ns4.gnu.org.           86400   IN      A       193.201.200.170

;; Query time: 122 msec
;; SERVER: 192.168.0.4#53(192.168.0.4)
;; WHEN: Mon Sep 1 18:41:11 2003
;; MSG SIZE rcvd: 133
```

Entonces nuestro servidor funciona. Fíjese en la última parte de la salida, se supone que la IP de nuestro servidor es 192.168.0.4. Ha tardado 122 milisegundos en obtener una respuesta. Si repetimos de nuevo la misma consulta, la última parte será parecida a

```
;; Query time: 2 msec
;; SERVER: 192.168.0.4#53(192.168.0.4)
;; WHEN: Mon Sep 1 18:55:07 2003
;; MSG SIZE rcvd: 133
```

¡Ahora solo ha tardado 2 milisegundos!. Ya tiene esa información en el caché (que expira 86400 segundos después de la primera consulta). Nuestro servidor está listo para trabajar :-). Es importante notar que **named** no genera ningún fichero, el caché está en memoria. De tal forma cuando se interrumpe el demonio o se inicia el sistema el caché parte de cero.

Queda por comentar cómo se hace para que al arrancar el sistema se active **named**. Esto dependerá en cada caso de la distribución. En el caso de SuSE 8.2 esto se realiza fácilmente con **yast2**

```
yast2 -> Runlevel editor -> RunLevel properties
-> Activar named en niveles 3 y 5
```

5.- Bibliografía.

El documento básico es la traducción al español de [Cómo ser un administrador DNS en poco tiempo.](#)