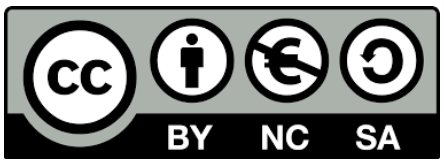
DNS

Antonio Boronat Pérez

SMX2 - Servicios en Red



Índice

Introducción

El servicio de nombres de dominio DNS (Domain Name System) se encarga, básicamente, de la traducción de nombres canónicos de sistema a sus direcciones IP y también de lo que se conoce como resolución inversa, que consiste en partiendo de la IP obtener los nombres que tiene asociados ya que un sistema puede tener más de un nombre.

Repasemos brevemente la historia de este servicio básico para el funcionamiento de las redes actuales e Internet.
Inicialmente, en la red precursora de Internet, ARPAnet, la traducción entre nombres de sistemas y sus IP se realizaba mediante un fichero de texto denominado HOSTS.TXT que era mantenido de forma centralizada y después los sistemas de la red obtenían una copia sobre la que realizar sus consultas. Este mecanismo funcionó bien mientras el número de sistemas conectados a la red se mantenía en decenas de sistemas, pero a medida que aumentaba el número de nuevos sistemas, mantener el fichero HOSTS.TXT actualizado se hacía más difícil, hasta el punto de no ser operativo. Así en 1984 se publicó la primera versión de DNS, que ha sido adoptado como el servicio para la resolución de nombres en redes basadas en TCP/IP. De todas formas, en las implementaciones de sistemas tipo *NIX todavía disponemos de un fichero similar a HOSTS.TXT, el */etc/hosts* donde podemos establecer las correspondencias entre nombres e IP's. Copiando dicho fichero en los diferentes sistemas de nuestra red funcionaría, la pega es que cada vez que debamos realizar algún cambio nombre-IP deberá realizarse en todos los ficheros *hosts* de todos los sistemas de la red.
El sistema DNS tiene una estructura jerárquica en forma de árbol, similar a la estructura de ficheros de los sistemas tipo UNIX, de forma que los servidores DNS conocen la información de zonas de ese árbol, cuando se les consulta, si tienen la respuesta contestan al cliente y si no entonces reenvían la consulta a otro servidor DNS de rango superior en la jerarquía, cuando se llega a quien conoce la respuesta esta recorre el camino inverso hasta el cliente.
https://es.wikipedia.org/wiki/Sistema_de_nombres_de_dominio

Descripción y funcionamiento de DNS

La configuración y funcionamiento del servicio DNS la tomaremos del documento DNS_linuxsilo-net.pdf la descripción realizada en este artículo queda bastante clara, aunque en alguno de los ejemplos la sintaxis puede variar respecto a la versión actual de DNS/BIND.

Para las configuraciones de prueba se indicarán las IP a utilizar. Y también cuando sea necesario se proporcionarán ejemplos de ficheros de configuraciones probadas en la versión utilizada del servicio.

Instalación del servicio:
 # apt-get update
 # apt-get install bind9 bind9-doc dnsutils

Para empezar seguiremos el análisis y si es el caso modificación de los ficheros de configuración que encontramos en el directorio */etc/bind*:

`named.conf.options` el valor de la IP `forwarders` a `194.179.1.100` que corresponde a un servidor DNS de Movistar Telefónica, también comentaremos las referencias a IPv6 *listen-on-v6 { any };* y el uso de *dnssec-validation auto;* .
El fichero `named.conf` no lo modificaremos, incorpora las configuraciones separadas de diferentes zonas.
El fichero `named.conf.local` contendrá la configuración de las zonas que vamos a crear en nuestra red y que vamos a administrar:

Esta primera sentencia limita a la máquina local el uso de la herramienta *rndc* que puede usarse para gestionar el servicio de forma remota.

```
controls {
inet 127.0.0.1 port 953 allow { any; } keys { "rndc-key"; };
};
```

A continuación, crea una lista de control de acceso (ACL) donde especificar sistemas que actúen como servidores DNS secundarios, el uso de estas listas sería más interesante si en lugar de un solo sistema dispusiéramos de varios.

```
acl slaves {
192.168.1.113;
};
```

Ahora definimos una zona DNS de nuestra red, en la que somos administradores (*master*), indicamos el fichero que guardará la base de datos que relaciona nombres - IP, además con *allow-query* especificamos quién puede consultar esta zona DNS, con *any* establecemos que cualquiera. la sentencia *allow-transfer* indica que sólo a los sistemas especificados se les permite obtener una copia de la base de datos de esta zona para que actúen como servidores DNS secundarios.

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

La zona siguiente expresa la zona para las resolución inversa, las especificaciones aplicables son las mismas.

```
zone "1.168.192.in-addr.arpa" {
type master;
file "/etc/bind/db.192.168.1";
allow-query { any; };
allow-transfer { slaves; };
};
```

A continuación, quedará por dar los valores a los ficheros de las bases de datos de cada zona, tomar como ejemplo: db.local, db.depinfo.iesjc y db.192.168.1 (bind.zip). Si nuestra rede tiene una máscara, por ejemplo de 16 bits, en la resolución inversa puedes tomar como ejemplo el contenido de bind_res_inversa_mascara_16bits.zip .

En cada sistema que deba tomar como servidor de DNS nuestro sistema debemos realizar su configuración, la podemos realizar de tres formas:

- Si el sistema tiene una IP estática, debemos poner los valores en el fichero con formato yaml usado por *netplan*. Por ejemplo:

01-network-manager-all.yaml:

```
network:
version: 2
renderer: NetworkManager
ethernets:
    enp0XX:
        dhcp4: no (o se puede eliminar la línea)
        addresses: (hay dos sintaxis, supongo equivalentes)
            - 192.168.xxx.xxx/24 ó [192.168.xxx.xxx/24]
        gateway4: 192.168.xxx.yyy
        nameservers:
         search: [midominio.org]
         addresses: [xxx.xxx.xxx.xxx, yyy.yyy.yyy.yyy]
```

- Si es cliente de DHCP debería configurarse mediante los valores que recibe de este servicio.
- También podemos establecer los valores usando las órdenes:

```
#systemd-resolve --set-dns=xxx.xxx.xxx.xxx -set-domain=midominio.org --
inferface=enp0sX
```

Para comprobar la configuración de DNS usa la orden: `$ resolvectl status`

(Para que este servicio de DNS realice consultas a servidores externos, las IP de las máquinas virtuales deben estar en el mismo segmento de red que el router o del equipo de salida a Internet).

Para realizar consultas a DNS podemos usar la orden host, por ejemplo:

```
$ host debian
$ host 192.168.1.112
```

Descripción Bases de Datos de Zona

SOA » (start of Authority) indica que es una zona de autoridad y los registros que van a continuación tienen información de autoridad. Normalmente, en la parte de *dominio* está la @ indicando el nombre del dominio. Estructura:

@ IN SOA <primary-name-server> <hostmaster-email> (

<serial-number>

<time-to-refresh>

<time-to-retry>
<time-to-expire>
<minimum-TTL>)

TTL *»* Time to Live, Tiempo de validez de la información del registro DNS expresado en segundo. Pasado este tiempo el cliente debe volver a consultar esta información.
A » (Dirección IPv4) asocia un nombre con una dirección IP.
AAAA *»* (Dirección IPv6)
CNAME » (Canonical Name) asocia un alias al nombre oficial de un ordenador, al cual, previamente ha sido declarado con un registre de tipo A. No se recomienda su uso, ya que puede dar problemas.
nombre_alias IN CNAME nombre_definido.dominios.
*NS » (Name Server) a*punta a un servidor dns, normalmente de nivel superior si estemos en una zona subordinada.
PTR *» (puntero)* traducción de dirección a nombre, resolución inversa.
MX » (Mail eXchanger) controla el encaminado del correo.
LOC » (Localització) localización geográfica.
TXT » (Text) comentarios para presentar información.
RP » (Persona Responsable) Especifica la persona de contacte de cada ordenador.

SOA :

origin » nombre absoluto de un servidor principal de este dominio. Los nombre absolutos acaban con un punto ".", p.e. debian.depinfo.iesjc.
contact » e-mail del responsable del servicio DNS. Se pone el usuario e-mail.subdominio. ... .dominio. (también acaba en punto), p.e. root.depinfo.iesjc. .
*serial » e*s un número que incrementa el administrador cada vez que modifica el fichero. Y tiene como finalidad indicar a los servidores secundarios que la información se ha modificado, así, cada cierto tiempo piden al servidor primario los registros SOA y miran si el número de serie ha cambiado, en caso afirmativo, toman el fichero completo para tener la información actualizada.
refresh » indica cada cuanto tiempo los servidores secundarios han de solicitar los registres SOA al principal.

retry » indica el tiempo que ha de esperar antes de volver a acceder al servidor principal si ha fallado la conexión anterior.
expire » si no se ha podido conectar con el servidor primario, pasado el tiempo aquí indicado se descartará toda la información de esta zona. Se recomienda poner 42 días (en segundos).
minim » es el límite temporal si no se ha definido en los registres con información de zona.

Órdenes comprobar el funcionamiento de DNS

Para validar los ficheros que contienen las configuraciones de las zonas usaremos:

```
named-checkconf -p fichero_con_zonas →
 #named-checkconf   -p   named.conf.local
```

si no hay errores mostrará las zonas definidas y si encuentra errores nos indicará cual es y en qué línea.
Para validar los ficheros con las bases de datos de zona usaremos:

```
named-checkzone nom_zona fich_bd_esa_zona -->
# named-checkzone  depinfo.iesjc  db.depinfo.iesjc
```

si no hay errores indicala zona a cargar y OK, pero si hay errores nos los indica.
Para consultar al servicio disponemos de varias órdenes:

```
host nombre o IP →
 $ host ord1.depinfo.iesjc
$ host 192.168.0.3
```

nos responderá, si ponemos el nombre del sistema con su IP y si ponemos una IP nos devuelve el nombre del sistema.

```
nslookup [nombre]  -->
$ nslookup
$ nslookup     www.google.es
```

permite realizar consultas a diferentes servidores DNS tanto de forma interactiva como una sola consulta. Si no indicamos el nombre de un sistema entramos en el modo interactivo de forma que en el indicador podemos realizar consultas sucesivas. Y si a la orden la acompañamos del nombre de un sistema a consultar nos responderá sólo a este.
Documentación sobre nslookup

dig es una orden de consulta de DNS que ofrece como respuesta una información más completa que las anteriores. La sintaxis básica de *dig*:

`dig` --> Realiza una consulta a la raíz del servicio de DNS y muestra la información correspondiente al servidor configurado en /etc/resolv.conf

`dig nombre_host` --> Devuelve toda la información asociada a este host disponible en el servicio DNS consutado.

`dig -x IP` --> Como el anterior, pero para la resolución inversa.

Documentación sobre dig

Servidor DNS secundario y Transferencia de Zonas

En los servicios DNS, los servidores secundarios o esclavos guardan una copia de la base de datos o una parte de ella, del servicio DNS de forma que si se produce un fallo en el servidor principal las consultas podrán ser satisfechas por los servidores secundarios. Dada su misión como alternativa ante fallos del servidor principal es obvio que los servidores secundarios deberían compartir las mínimas infraestructuras con el servidor principal, como por ejemplo alimentación eléctrica, si falla en los dos sistemas perdemos el servicio, o la conexión a un switch si los dos están conectados al mismo componente de red, si falla este perderemos la conectividad con el servicio.

El paso de información entre el servidor primario y los secundarios se realiza de forma automática, sólo se requiere configurar los servicios de forma que se complemente sus parámetros de IP, zonas a transferir y los permisos o claves correspondientes para que las transferencias de zonas se realice forma segura.

Para la configuraciones de ejemplo seguiremos los ficheros contenidos en bind_secundario.zip.

Para el secundario los ficheros de configuración a modificar en */etc/bind* son:

`named.conf.options` modificamos `forwarders` como en el principal.

`named.conf.local` modificamos según las zonas que deben transferirse a nuestro servidor secundario, básicamente cambia type que ahora vale `slave,` se indica el nombre de los ficheros sobre los que se realiza la copia de la base de datos, pero sin especificar la ruta que por defecto es */var/cache/bind* y con la directiva *master* indicamos la IP del servidor principal del cual esperamos la copia de esa zona, según nuestro ejemplo:

```
controls {
inet 127.0.0.1 allow { 127.0.0.1; } keys { "rndc-key"; };
};

zone "depinfo.iesjc" {
type slave;
file "sec.db.depinfo.iesjc";
allow-query { any; };
masters { 192.168.1.111; };
};

zone "1.168.192.in-addr.arpa" {
type slave;
file "sec.db.192.168.1";
masters { 192.168.1.111; };
};
```

Recordemos que para que nuestros sistemas conozcan este servidor secundario deberá configurarse como se ha visto anteriormente.

Una vez configurado el servidor secundario reiniciamos el servicio y comprobamos que en */var/cache/bind* se han creado los ficheros con las copias de las bases de datos. En caso afirmativo, para probar el funcionamiento del servidor secundario tan solo debemos parar el servicio del sistema principal y comprobamos que las consultas siguen funcionando, si paramos también el secundario comprobaremos que el DNS ya no funciona.

DNS sobre diferentes segmentos de red

Ahora nos plateamos un servicio DNS que trabaja sobre dos segmentos de red, en realidad, sólo es un ejemplo que presenta la problemática cuando disponemos de más de un segmento.

Partimos de un dominio (depinfo.iesjc) que está implantado sobre dos segmentos de red con direcciones 192.168.**1**.XXX y 192.168.**2**.XXX, la base de datos para la resolución normal de nombre a IP la podemos dejar con una sola zona tal y como está definida en los ejemplos visto hasta

ahora y en el fichero de base de datos (db.depinfo.iesjc) se irán añadiendo los registros nombre-IP de los dos segmentos. En cambio, para la resolución inversa usaremos zonas diferentes para cada segmento, así que definiremos dos zonas: *1.168.192.in-addr.arpa* para los ordenadores que están en el primer segmento y *2.168.192.in-addr.arpa* para los que pertenecen al segundo segmento. Cada zona tendrá su fichero de base de datos donde guardará los registros IP-nombre necesarios en la resolución inversa de esa zona. Los ficheros de configuración los encontraréis en bind_2_segmentos_red.zip.

Actualizar DNS mediante DHCP

Los pasos a seguir los encontramos en la página web:

http://lani78.wordpress.com/2008/08/12/dhcp-server-update-dns-records/ versión en PDF del artículo.

Una configuración de ejemplo sencilla para bind9 y dhcp3 en dhcp_update_dns.zip

Directivas que permiten este proceso en la configuración del servicio DHCP en /etc/dhcp/dhcpd.conf añadimos:

```
ddns-update-style interim; # Activa la actualización de DNS
ignore client-updates; # Impide que el cliente ponga su FQDN
ddns-domainname "depinfo.iesjc."; # Indica el nombre del dominio
ddns-rev-domainname "in-addr.arpa."; # Nombre del dominio inverso
```

Añadimos las zonas que van a ser actualizadas en DNS:

```
zone depinfo.iescj. { primary 127.0.0.1; }

zone 150.168.192.in-addr.arpa. { primary 127.0.0.1; }
```

También cabe indicar, que los clientes de DHCP deben enviar su nombre al servidor para que este pueda añadirlo al registro de alquiler y pasarlo a DNS. Este nombre se debe indicar en el fichero de configuración del cliente DHCP, por ejemplo en **/etc/dhcp/dhclient.conf** en la opción **send host-name "nombre"; Nota**: Es posible que la aplicación de seguridad AppArmor limite el acceso a los ficheros de /etc/bind. Si es el caso, deben crearse los ficheros de las bases de datos en /var/lib/bind. O bien, se puede

deshabilitar el servicio AppArmor para facilitar las pruebas de DHCP + DNS con las órdenes: # service apparmor stop

#systemctl disable apparmor