**LABORATORIO No 5: Servidor DNS**

**Objetivos**

Al finalizar el taller, se espera que el estudiante esté en capacidad de:

* Instalar BIND en Linux.
* Configurar BIND para resolución de nombres de dominio.
* Iniciar el servicio DNS a través de BIND.

**Antecedentes**

Antes de iniciar el proceso de configuración, se hace necesario describir la manera como nace el servicio DNS. Para este propósito, se presenta una parte del trabajo realizado por Robles Garay dónde expone el nacimiento del servicio DNS.

La historia del sistema de nombres de dominio, DNS, se remonta a la década de los setenta, donde cada una de las computadoras conectadas a la red tenía asignada una dirección numérica (de la misma forma que nuestros teléfonos actuales tienen asignado un número telefónico). Para acceder a aquellos equipos de cómputo era necesario recordar la dirección numérica de cada uno, ya que el esquema de “nombramiento” de aquel entonces era bastante limitado.

Esta situación trajo como consecuencia la creación de un nuevo sistema de nombramiento, el cual sería conocido como Sistema de Nombres de Dominio (Domain Name System, DNS) y produciría serios problemas 15 años después.

Lo que concibió su creador, Paul Mockapetris, básicamente, fue un sistema jerárquico similar a un sistema de archivos de computadora. En esta estructura jerárquica definió niveles, de la misma forma que en un sistema de archivos hay directorios. A cada nivel de la estructura le asignó un nombre o etiqueta. El nivel cero, o raíz, no tiene nombre, el nivel 1 puede ser alguno de los que se muestran en la figura 1, .mx, .uk, .com o .net, el cual se conoce como Top Level Domain – TLD. A su vez, éstos pueden tener subclasificaciones, como en el caso de .mx que tiene debajo a .com.mx, .net.mx, .gob.mx, etc. A este nivel se le conoce como Second Level Domain – SLD.

De esta forma, los nombres de dominio se construyen por una secuencia de etiquetas separadas por un punto, empezando en el nivel más profundo hasta llegar al nivel superior. Por ejemplo, en la Figura 2 se puede apreciar que el nombre de dominio empresa.com.mx. se forma desde el último nivel llamado “empresa”, después el SLD “com” y por último el TLD “mx”.

Figura 1. Estructura jerárquica.

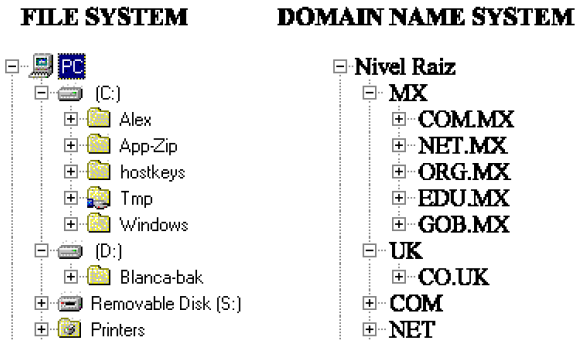
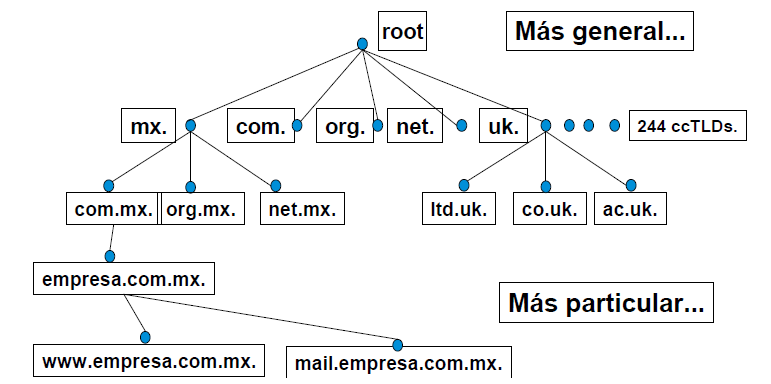


Figura 2. Estructura jerárquica del DNS.



Las etiquetas pueden tener letras, números y el guion medio “-”, pero no puede iniciar ni terminar con guión. Cada etiqueta puede llevar hasta 63 caracteres, el nombre de dominio en total puede tener hasta 255 (cualquier combinación de letras, números y guión medio). Y puede haber hasta 127 niveles (siempre y cuando no se rebase el límite de 255 caracteres).

El DNS buscaba un objetivo muy simple, desempeñar una función técnica de traducción de nombres de equipos de cómputo a su dirección numérica correspondiente, que fuera conveniente, amigable y fácil de utilizar por los usuarios de Internet, es decir proveer un esquema de interpretación entre los usuarios y las computadoras, sin que los primeros tuvieran la necesidad de recordar las direcciones numéricas de cada uno de los equipos a los que intentaban comunicarse; por ejemplo: 131.178.11.16, 200.23.1.7, en vez de www.mty.itesm.mx y www.nic.mx, respectivamente.

La realidad es que este esquema, el DNS, cumplió con su intención y más que eso, los nombres de dominio no sólo fueron una manera fácil para los usuarios de conectarse a los equipos en la red, sino que además significó una manera simple de representar ideas, productos, servicios, empresas, organismos, etc. El fenómeno del WWW le aplicó una aceleración importante al registro de dominios y pronto vendrían los problemas. A partir de este punto, se hizo imposible dar marcha atrás a un esquema de nombramiento que empezó siendo técnico y gracias a la comercialización de Internet, ese esquema acabó generando conflictos con esquemas sociales definidos anteriormente. [[[1]](#footnote-1)]

**Configuración**

Veamos como configurar BIND9 para disponer de un servidor DNS en una intranet, que resuelva dominios internos. Por ejemplo, en la intranet se utilizaran dominios que terminen en “marblestation.homeip.net” como “saturno.marblestation.homeip.net” o “luna.marblestation.homeip.net”. El servidor DNS se encargará de resolver esos dominios en sus respectivas IPs, además de resolver otros dominios de Internet como “google.com”.

Instalamos BIND9 y nos desplazamos a su directorio de configuración:

$sudo apt-get install bind9 dnsutils

$cd /etc/bind/

Editamos named.conf.local y añadimos la zona “myenterprise.homeip.net”, haciendo referencia a su fichero de configuración:

zone "myenterprise.homeip.net" {

type master;

file "/etc/bind/db.myenterprise";

};

Creamos el fichero de configuración “db.myenterprise” a partir de “db.local”:

$sudo cp db.local db.myenterprise

Editamos “db. myenterprise”, reemplazamos la palabra “localhost” por “myenterprise.homeip.net”, cambiamos la IP “127.0.0.1″ por la que queramos asignar al dominio y añadimos al final del fichero todos los A, MX y CNAME que queramos, quedando:

;

; BIND data file for local loopback interface

;

$TTL 604800

@ IN SOA myenterprise.homeip.net. root.myenterprise.homeip.net. (

1 ; Serial

604800 ; Refresh

86400 ; Retry

2419200 ; Expire

604800 ) ; Negative Cache TTL

;

@ IN NS myenterprise.homeip.net.

@ IN A 192.168.10.1

@ IN MX 0 myenterprise.homeip.net.

www IN A 192.168.10.1

saturno IN CNAME myenterprise.homeip.net.

En este ejemplo, se muestra el dominio a resolver, ‘myenterprise.homeip.net.’ y el segundo es la cuenta de correo del administrador, ‘root.myenterprise.homeip.net.’ (Remplazando el primer punto por arroba, lo que dejaría ‘root@myenterprise.homeip.net’). Debemos notar que al final de cada dominio viene un punto, que identifica la raíz de este. El resto de los parámetros son:

Serial: es un identificador del archivo, puede tener un valor arbitrario pero se recomienda que tenga la fecha con una estructura AAAA-MM-DD y un consecutivo.

Refresco: número de segundos que un servidor de nombres secundario debe esperar para comprobar de nuevo los valores de un registro.

Reintentos: número de segundos que un servidor de nombres secundario debe esperar después de un intento fallido de recuperación de datos del servidor primario.

Expiración: número de segundos máximo que los servidores de nombre secundarios retendrán los valores antes de expirarlos.

TTL mínimo: Significa Time To Live y es el número de segundos que los registros se mantienen activos en los servidores NS caché antes de volver a preguntar su valor real.

A continuación se definen los registros necesarios, cuyos tipos ya han sido explicados anteriormente en este documento.

Cada vez que se cambia la configuración de BIND9, debemos reiniciar el demonio:

$ sudo service bind9 restart

Para que nuestra máquina utilice el servidor de DNS que hemos configurado, debemos editar “/etc/resolv.conf” y dejamos únicamente la línea:

nameserver 127.0.0.1

Se debería hacer lo mismo con el resto de máquinas de la intranet que vayan a utilizar el servidor, con la única diferencia que habrá que substituir la IP 127.0.0.1 por la IP que tenga el servidor en la red. Para comprobar el correcto funcionamiento, utilizamos el comando “host” el cual sirve para resolver dominios:

$ host myenterprise.homeip.net

myenterprise.homeip.net has address 192.168.48.32

myenterprise.homeip.net mail is handled by 0 myenterprise.homeip.net.

$ host saturno.myenterprise.homeip.net

saturno. myenterprise.homeip.net is an alias for myenterprise.homeip.net.

myenterprise.homeip.net has address 192.168.48.32

saturno.myenterprise.homeip.net is an alias for myenterprise.homeip.net.

saturno.myenterprise.homeip.net is an alias for myenterprise.homeip.net.

myenterprise.homeip.net mail is handled by 0 myenterprise.homeip.net.

Si deseamos también disponer de resolución de dominios inversa, es decir, que podamos preguntar por la IP “192.168.10.1″ y el servidor DNS nos diga que pertenece a myenterprise.homeip.net, debemos añadir a “/etc/bind/named.conf.local”:

zone "10.168.192.in-addr.arpa" {

type master;

file "/etc/bind/db.192";

};

Creamos el archivo de configuración “/etc/bind/db.192″ a partir del “/etc/bind/db.127″:

$cd /etc/bind/

$sudo cp db.127 db.192

Editamos “/etc/bind/db.192″, sustituimos “localhost” por “myenterprise.homeip.net” y cambiamos la última línea:

;

; BIND reverse data file for local loopback interface

;

$TTL 604800

@ IN SOA myenterprise.homeip.net. root.myenterprise.homeip.net. (

1 ; Serial

604800 ; Refresh

86400 ; Retry

2419200 ; Expire

604800 ) ; Negative Cache TTL

;

@ IN NS myenterprise.homeip.net.

1 IN PTR myenterprise.homeip.net.

La última línea indica que la IP [192.168.10].1 (escrita a la inversa y omitiendo el 192 que ya se especificó en “named.conf.local”) corresponde al dominio myenterprise.homeip.net.

Podemos comprobar su funcionamiento reiniciando el servicio BIND9 y realizando una consulta:

$ sudo service bind9 restart

$ host 192.168.10.1

1.10.168.192.in-addr.arpa domain name pointer myenterprise.homeip.net.

1. []**Robles Garay, Oscar.** *¿Qué es el DNS?***.** s.l. : Boletín de política informática, 2003, Vol. 1. pp. 54-56. [↑](#footnote-ref-1)