Laboratorio Instalación DNS BIND CHROOT RedHat 7

Servidor 192.168.1.150

Servidor Maestro 192.168.1.150

Comprobamos que tenemos instalado el paquete **bind-chroot**

rpm -qa bind-*

En nuestro servidor, podemos encontrar los ficheros de zona de ejemplo:

 # /usr/share/doc/bind-9.9.4/sample/var/named/

Y un ejemplo del fichero de configuración de named.conf:

#/etc/named.conf

Copiamos al directorio root de nuestro servidor la carpeta ddns-192.168.1

#**cd /root/ddns-192.168.1**

[root@trasgu ddns-192.168.1]# **cp db* /var/named/chroot/var/named/**
[root@trasgu ddns-192.168.1]# **cp named.ca /var/named/chroot/var/named/**
[root@trasgu ddns-192.168.1]# **cp named.conf /var/named/chroot/etc/**

cd /var/named/chroot/var/named/

chown named.named *

[root@trasgu ~]# **tail -f /var/log/messages**

vi /etc/resolv.conf
nameserver 192.168.1.150
domain curso.esp
search curso.esp

Paramos y deshabilitamos el servicio named y arrancamos y habilitamos el servicio bind-chroot

```
# /usr/libexec/setup-named-chroot.sh /var/named/chroot on
# systemctl stop named
# systemctl disable named
# systemctl start named-chroot
# systemctl enable named-chroot
```

```
[root@trasgu ~]# netstat -tan |grep -i listen
tcp        0      0 127.0.0.1:2208          0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN
tcp        0      0 192.168.1.150:53        0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
```

```
 [root@trasgu named]# nslookup pc2
Server:         192.168.1.150
Address:        192.168.1.150#53

[root@trasgu named]# nslookup trasgu
Server:         192.168.1.150
Address:        192.168.1.150#53

Name:   trasgu.curso.esp
Address: 192.168.1.150
```

Testear la configuración de nuestro DNS:

Testear la configuración (sintaxis) del fichero de configuración named.conf

 # named-checkconf /var/named/chroot/etc/named.conf

Testear los ficheros de zona:

named-checkzone curso.esp /var/named/chroot/var/named/db.curso
zone curso.esp/IN: loaded serial 2008012003
OK

Testear al servidor maestro de DNS:

dig sercentos7.curso.esp

```
; <<>> DiG 9.9.4-RedHat-9.9.4-18.el7 <<>> sercentos7.curso.esp
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41514
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
```

```
;sercentos7.curso.esp.          IN      A

;; ANSWER SECTION:
sercentos7.curso.esp.   259200  IN      A       192.168.1.5

;; AUTHORITY SECTION:
curso.esp.              259200  IN      NS      orion.curso.esp.
curso.esp.              259200  IN      NS      sercentos7.curso.esp.

;; ADDITIONAL SECTION:
orion.curso.esp.        259200  IN      A       192.168.1.152

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: sáb ago 15 11:35:33 CEST 2015
;; MSG SIZE  rcvd: 115
```

Ficheros de configuración Maestro

named.conf

```
options {
 directory "/var/named";
 forwarders{
  80.58.0.33;

 };
 allow-transfer{
  192.168.1.150;
  192.168.1.152;
  };

};


zone "." {
 type hint;
 file "named.ca";
};
zone "curso.esp"{
 type master;
 allow-update {
 192.168.1.0/24;
};
 file "db.curso";
};
zone "dominio1.esp"{
 type master;
 allow-update {
 192.168.1.0/24;
};
 file "db.dominio1";
 };
zone "dominio2.esp"{
 type master;
 allow-update {
 192.168.1.0/24;
};
 file "db.dominio2";
 };

zone "0.168.192.IN-ADDR.ARPA"{
 type master;
 allow-update {
```

```
  192.168.1.0/24;
 };
 file "db.192.168.1";
};
```

[root@trasgu ~] **# systemctl restart named-chroot**

En todos los servidores de DNS para SElinux y firewall:

Configuración de Firewall:

#firewall-cmd --permanent --add-port=53/tcp

Reiniciamos el firewall:

#firewall-cmd –reload

Configurar SElinux:

```
#restorecon -rv /var/named/chroot/var/named
#restorecon /var/named/chroot/etc/named.conf
```

Laboratorio Servidor Maestro/Esclavo dominio curso.esp

En este laboratorio, trabajaremos con un servidor maestro (192.168.1.150 trasgu) y un servidor esclavo (192.168.1.152) orion.

En el servidor Esclavo:

[root@orion ~]# cat /etc/hosts
127.0.0.1 localhost.localdomain localhost
192.168.1.152 orion.curso.esp orion

 # cat /etc/resolv.conf
nameserver 192.168.1.150
search curso.esp
domain curso.esp

Comprobamos que podamos resolver por nuestro servidor Maestro (192.168.1.150)

[root@orion /]# nslookup orion
Server: 192.168.1.150
Address: 192.168.1.150#53

Name: orion.curso.esp
Address: 192.168.1.152

Copiamos los ficheros del laboratorio, la carpeta esclavo, en el directorio /root, de nuestro servidor esclavo.

[root@orion esclavo]# **cd /root/esclavo**

[root@orion esclavo]# **cp named.ca /var/named/chroot/var/named/**

[root@orion esclavo]# **cp named.conf /var/named/chroot/etc/**

mkdir /var/named/chroot/var/named/slaves

chmod 770 /var/named/chroot/var/named/slaves/

 # chown named.named /var/named/chroot/var/named/slaves/

 # ls -ld /var/named/chroot/var/named/slaves/

drwxrwx--- 2 named named 6 ago 15 11:16

[root@orion esclavo]# **tail -f /var/log/messages**

Paramos y deshabilitamos el servicio named y arrancamos y habilitamos el servicio bind-chroot en nuestro servidor esclavo:

```
# /usr/libexec/setup-named-chroot.sh /var/named/chroot on
# systemctl stop named
# systemctl disable named
# systemctl start named-chroot
# systemctl enable named-chroot
```

Comprobamos que se han copiado los ficheros de zona del servidor maestro en el servidor esclavo en el directorio:

[root@orion slaves]# pwd
/var/named/chroot/var/named/slaves

[root@orion slaves]# **ls -l**
-rw-r--r-- 1 named named 427 oct 28 22:38 db.192.168.1
-rw-r--r-- 1 named named 580 oct 28 22:38 db.curso

Ficheros de configuración servidor esclavo:

named.conf

```
options {
 directory "/var/named";
 forwarders{
  8.8.8.8;
  192.168.1.1;

 };
 allow-transfer{
  192.168.1.150;
  192.168.1.3;
  };

};

zone "." {
 type hint;
 file "named.ca";
};
zone "curso.esp"{
 type slave;
 file "slaves/db.curso";
 masters { 192.168.1.150; };
};

zone "0.168.192.IN-ADDR.ARPA"{
 type slave;
 file "slaves/db.192.168.1";
 masters { 192.168.1.150; };

};
```

Comprobamos la salida del fichero log:

[root@orion ~]# **tail -f /var/log/messages**

Oct 28 22:38:25 orion named[3785]: zone 0.168.192.IN-ADDR.ARPA/IN: Transfer started.
Oct 28 22:38:25 orion named[3785]: transfer of '0.168.192.IN-ADDR.ARPA/IN' from 192.168.1.150#53: connected using 192.168.1.152#35555
Oct 28 22:38:25 orion named[3785]: zone 0.168.192.IN-ADDR.ARPA/IN: transferred serial 2008012003
Oct 28 22:38:25 orion named[3785]: transfer of '0.168.192.IN-ADDR.ARPA/IN' from 192.168.1.150#53: end of transfer
Oct 28 22:38:25 orion named[3785]: zone 0.168.192.IN-ADDR.ARPA/IN: sending notifies (serial 2008012003)
Oct 28 22:38:25 orion named[3785]: client 192.168.1.152#38611: received notify for zone '0.168.192.IN-ADDR.ARPA'
Oct 28 22:38:25 orion named[3785]: zone 0.168.192.IN-ADDR.ARPA/IN: refused notify from non-master: 192.168.1.152#38611

Configuración servidor Maestro (192.168.1.150)

named.conf

```
options {
 directory "/var/named";
 forwarders{
  80.58.0.33;
       8.8.8.8;

 };
 allow-transfer{
  192.168.1.150;
  192.168.1.152;
  };
allow-notify {192.168.1.152;}; //PERMITE NOTIFICAR LAS ZONAS EN LOS ESCLAVOS
};


zone "." {
 type hint;
 file "named.ca";
};
zone "curso.esp"{
 type master;
 allow-update {
 192.168.1.0/24;
};
 file "db.curso";
};

zone "0.168.192.IN-ADDR.ARPA"{
 type master;
 allow-update {
  192.168.1.0/24;
};
 file "db.192.168.1";

};
```

db.curso

```
$ORIGIN .
$TTL 259200     ; 3 days
curso.esp               IN SOA trasgu.curso.esp. root.curso.esp. (
                                2008012003 ; serial
                                86400      ; refresh (1 day)
                                7200       ; retry (2 hours)
                                2592000    ; expire (4 weeks 2 days)
                                172800     ; minimum (2 days)
                                )
                        NS      orion.curso.esp.
                        NS      trasgu.curso.esp.
                        MX      3 trasgu.curso.esp.
$ORIGIN curso.esp.
agendapc5               CNAME           pc5
curso                   A       192.168.1.3
fresnosa        CNAME           trasgu
localhost               A       127.0.0.1
pc2                      A       192.168.1.89
portalpc12              CNAME           pc12
portatil        A       192.168.1.2
trasgu                  A       192.168.1.150
orion                   A       192.168.1.152
webalizerpc12           CNAME           pc12
```

db.192.168.1

```
$TTL 259200     ; 3 days
0.168.192.IN-ADDR.ARPA.                 IN SOA trasgu.curso.esp. root.curso.esp. (
                                2008012003 ; serial
                                86400      ; refresh (1 day)
                                7200       ; retry (2 hours)
                                2592000    ; expire (4 weeks 2 days)
                                172800     ; minimum (2 days)
                                )
0.168.192.IN-ADDR.ARPA.         NS      orion.curso.esp.
0.168.192.IN-ADDR.ARPA.         NS      trasgu.curso.esp.
0.168.192.IN-ADDR.ARPA.         MX      3 trasgu.curso.esp.

150             IN      PTR     trasgu.curso.esp.
152             IN      PTR     orion.curso.es.
```

[root@trasgu ~]# **tail -f /var/log/messages**

```
Oct 28 10:57:05 trasgu named[3411]: zone curso.esp/IN: loaded serial 2008012003
Oct 28 10:57:05 trasgu named[3411]: zone dominio1.esp/IN: loaded serial 2007040201
Oct 28 10:57:05 trasgu named[3411]: zone dominio2.esp/IN: loaded serial 2007040200
Oct 28 10:57:05 trasgu named[3411]: running
Oct 28 10:57:05 trasgu named[3411]: zone curso.esp/IN: sending notifies (serial 2008012003)
Oct 28 10:57:05 trasgu named[3411]: zone 0.168.192.IN-ADDR.ARPA/IN: sending notifies (serial
2008012003)
Oct 28 10:59:03 trasgu named[3411]: client 192.168.1.152#38801: transfer of 'curso.esp/IN': AXFR
started
Oct 28 10:59:03 trasgu named[3411]: client 192.168.1.152#38801: transfer of 'curso.esp/IN': AXFR
ended
Oct 28 10:59:04 trasgu named[3411]: client 192.168.1.152#35555: transfer of '0.168.192.IN-
ADDR.ARPA/IN': AXFR started
Oct 28 10:59:04 trasgu named[3411]: client 192.168.1.152#35555: transfer of '0.168.192.IN-
ADDR.ARPA/IN': AXFR ended
```

Laboratorio DNSSEC

Seguridad adicional para transferencias de zona.

Cuando se gestionan dominios a través de redes públicas, es importante considerar que si se tienen esquemas de servidores **maestros** y **esclavos**, siempre será más conveniente utilizar una **clave cifrada** en lugar de una dirección IP, debido a que esta última puede ser falsificada bajo ciertas circunstancias.

Comúnmente se definen las direcciones IP desde las cuales se permitirá transferencias de zonas, utilizando una configuración en el fichero **/var/named/chroot/etc/named.conf** como la ejemplificada a continuación, donde los servidores esclavos corresponden a los servidores con direcciones IP 192.168.1.11 y 192.168.1.12:

```
zone "mi-dominio.org" {
type master;
file "mi-dominio.org.zone";
allow-update { none; }:
allow-transfer { 192.168.1.11; 192.168.1.12; };
};
```

Lo anterior permite la transferencia de zona para los servidores con direcciones IP 192.168.1.11 y 192.168.1.12, los cuales utilizan la siguiente configuración en el fichero **/var/named/chroot/etc/named.conf**, ejemplificada a continuación, donde el servidor primario (zonas maestras) corresponde al servidor con dirección IP 192.168.1.1:

```
zone "mi-dominio.org" {
type slave;
file "mi-dominio.org.zone";
masters { 192.168.1.1; };
};
```

El inconveniente del esquema anterior es que es fácil falsificar las direcciones IP. A fin de evitar que esto ocurra, el método recomendado será utilizar una clave cifrada que será validada en lugar de la dirección IP. La llave se crea con el mandato **dnsseckeygen**, especificando un algoritmo, que puede ser **RSAMD5** o **RSA**, **DSA**, **DH** (**D**iffie **H**ellman) o **HMAC-MD5**, el tamaño de la llave en octetos (bits), el tipo de la llave, que puede ser ZONE, HOST, ENTITY o USER y el nombre específico para la clave cifrada. DSA y RSA se utilizan para **DNS Seguro** (**DNSSEC**), en tanto que **HMAC-MD5** se utiliza para **TSIG** (**T**ransfer **SIG**nature o transferencia de firma). Lo más común es utilizar **TSIG**. En el siguiente ejemplo, se generará en el directorio de trabajo actual la clave **mi-dominio.org**, utilizando **/dev/random** como fuente de datos aleatorios, un algoritmo **HMAC-MD5** tipo **HOST** de 128 octetos (bits):

Servidor Maestro:

Para el correcto funcionamiento de este laboratorio los relojes del servidor maestro y esclavo tienen que estar sincronizados, comprobaremos con el comando date en nuestros servidores hora y fecha:

Cambiar fecha y hora desde consola

#date --set "2015-06-12 11:01"

[root@trasgu /]# **cd /var/named/chroot/var/named/**

[root@trasgu /]# **dnssec-keygen -r /dev/random -a HMAC-MD5 -b 128 -n HOST curso.esp**

[root@trasgu named]# **chmod 400 Kcurso.esp.+157+17021.***

[root@trasgu named]# **chown named.named Kcurso.esp.+157+17021.***

[root@trasgu named]# **ls -l**
total 36
drwxrwx--- 2 named named 4096 ago 25 2004 data
-rw-r--r-- 1 named named 494 oct 28 10:56 db.192.168.1
-rw-r--r-- 1 named named 626 oct 28 23:41 db.curso
-rw-r--r-- 1 named named 549 oct 28 10:03 db.dominio1
-rw-r--r-- 1 named named 549 oct 28 10:03 db.dominio2
-r-------- 1 named named   53 oct 29 00:07 Kcurso.esp.+157+17021.key
-r-------- 1 named named   81 oct 29 00:07 Kcurso.esp.+157+17021.private
-rw-r--r-- 1 named named 2518 oct 28 10:04 named.ca
drwxrwx--- 2 named named 4096 jul 27 2004 slaves

[root@trasgu named]# **cat Kcurso.esp.+157+17021.key**
curso.esp. IN KEY 512 3 157 **KuEeKa7Mxj9UUJtgoc7tSQ==**

[root@trasgu named]# cat Kcurso.esp.+157+17021.private
Private-key-format: v1.2
Algorithm: 157 (HMAC_MD5)
Key: **KuEeKa7Mxj9UUJtgoc7tSQ==**

named.conf

//Añadimos lo que tenemos en negrita

```
key curso.esp {
   algorithm HMAC-MD5;
   secret "KuEeKa7Mxj9UUJtgoc7tSQ==";
};

options {
 directory "/var/named";
 forwarders{
  80.58.0.33;

 };
allow-transfer { key curso.esp; };
```

[root@trasgu named]# **systemctl restart named-chroot**

tail -f /var/log/messages

```
Oct 29 00:37:31 trasgu named[4511]: shutting down: flushing changes
Oct 29 00:37:31 trasgu named[4511]: stopping command channel on 127.0.0.1#953
Oct 29 00:37:31 trasgu named[4511]: stopping command channel on ::1#953
Oct 29 00:37:31 trasgu named[4511]: no longer listening on 127.0.0.1#53
Oct 29 00:37:31 trasgu named[4511]: no longer listening on 192.168.1.150#53
Oct 29 00:37:31 trasgu named[4511]: exiting
Oct 29 00:37:33 trasgu named[4688]: starting BIND 9.3.6-P1-RedHat-9.3.6-
20.P1.el5_8.6 -u named -t /var/named/chroot
Oct 29 00:37:33 trasgu named[4688]: adjusted limit on open files from 1024 to 1048576
Oct 29 00:37:33 trasgu named[4688]: found 1 CPU, using 1 worker thread
Oct 29 00:37:33 trasgu named[4688]: using up to 4096 sockets
Oct 29 00:37:33 trasgu named[4688]: loading configuration from '/etc/named.conf'
Oct 29 00:37:33 trasgu named[4688]: using default UDP/IPv4 port range: [1024,
65535]
Oct 29 00:37:33 trasgu named[4688]: using default UDP/IPv6 port range: [1024,
65535]
Oct 29 00:37:33 trasgu named[4688]: listening on IPv4 interface lo, 127.0.0.1#53
Oct 29 00:37:33 trasgu named[4688]: listening on IPv4 interface eth1,
192.168.1.150#53
Oct 29 00:37:33 trasgu named[4688]: zone 'curso.esp' allows updates by IP address,
which is insecure
Oct 29 00:37:33 trasgu named[4688]: zone 'dominio1.esp' allows updates by IP address,
which is insecure
Oct 29 00:37:33 trasgu named[4688]: zone 'dominio2.esp' allows updates by IP address,
which is insecure
Oct 29 00:37:33 trasgu named[4688]: zone '0.168.192.IN-ADDR.ARPA' allows updates
by IP address, which is insecure
Oct 29 00:37:33 trasgu named[4688]: command channel listening on 127.0.0.1#953
Oct 29 00:37:33 trasgu named[4688]: command channel listening on ::1#953
Oct 29 00:37:33 trasgu named[4688]: zone 0.168.192.IN-ADDR.ARPA/IN: loaded
serial 2008012003
Oct 29 00:37:33 trasgu named[4688]: zone curso.esp/IN: loaded serial 2013102804
```

Oct 29 00:37:33 trasgu named[4688]: zone dominio1.esp/IN: loaded serial 2007040201
Oct 29 00:37:33 trasgu named[4688]: zone dominio2.esp/IN: loaded serial 2007040200
Oct 29 00:37:33 trasgu named[4688]: running
Oct 29 00:37:33 trasgu named[4688]: zone curso.esp/IN: sending notifies (serial 2013102804)
Oct 29 00:37:33 trasgu named[4688]: zone 0.168.192.IN-ADDR.ARPA/IN: sending notifies (serial 2008012003)

Servidor esclavo

named.conf

//Añadimos lo que tenemos en negrita

```
key curso.esp {
   algorithm HMAC-MD5;
   secret "KuEeKa7Mxj9UUJtgoc7tSQ==";
};

server 192.168.1.150 {
      keys { curso.esp; };
};

options {
 directory "/var/named";
 forwarders{
  8.8.8.8;
  192.168.1.1;
```

systemctl restart named-chroot

[root@orion ~]# **tail -f /var/log/messages**

Oct 29 00:46:10 orion named[4552]: loading configuration from '/etc/named.conf'
Oct 29 00:46:10 orion named[4552]: using default UDP/IPv4 port range: [1024, 65535]
Oct 29 00:46:10 orion named[4552]: using default UDP/IPv6 port range: [1024, 65535]
Oct 29 00:46:10 orion named[4552]: listening on IPv4 interface lo, 127.0.0.1#53
Oct 29 00:46:10 orion named[4552]: listening on IPv4 interface eth0, 192.168.1.152#53
Oct 29 00:46:10 orion named[4552]: command channel listening on 127.0.0.1#953
Oct 29 00:46:10 orion named[4552]: command channel listening on ::1#953
Oct 29 00:46:10 orion named[4552]: zone 0.168.192.IN-ADDR.ARPA/IN: loaded serial 2008012003
Oct 29 00:46:10 orion named[4552]: zone curso.esp/IN: loaded serial 2013102804
Oct 29 00:46:10 orion named[4552]: running
Oct 29 00:46:10 orion named[4552]: zone 0.168.192.IN-ADDR.ARPA/IN: sending notifies (serial 2008012003)
Oct 29 00:46:10 orion named[4552]: zone curso.esp/IN: sending notifies (serial 2013102804)
Oct 29 00:46:10 orion named[4552]: client 192.168.1.152#11029: received notify for zone '0.168.192.IN-ADDR.ARPA'
Oct 29 00:46:10 orion named[4552]: zone 0.168.192.IN-ADDR.ARPA/IN: refused notify from non-master: 192.168.1.152#11029

Oct 29 00:46:10 orion named[4552]: client 192.168.1.152#1507: received notify for zone 'curso.esp'

Laboratorio DDNS

En este laboratorio, tendremos que montar un servidor DHCP, y verificar como actualiza las zonas de DNS dinámicamente, tanto la zona directa como la zona inversa del dominio curso.esp, en el servidor DNS que tiene definida la zona como maestro (192.168.1.150).

Configuración DHCP:

dhcpd.conf

```
authoritative;
default-lease-time 86400;
max-lease-time 86400;
ddns-updates on;
ddns-update-style interim;
shared-network miredlocal {
 subnet 192.168.1.0 netmask 255.255.255.0 {
 option routers 192.168.1.1;
 option subnet-mask 255.255.255.0;
 option broadcast-address 192.168.1.255;
 option domain-name "curso.esp";
 option domain-name-servers 192.168.1.150, 80.58.0.33;
 option netbios-name-servers 192.168.1.1;
 range 192.168.1.50 192.168.1.90;
 }
 host m253 {
  option host-name "m253.mi-red-local.com";
  hardware ethernet 00:50:BF:27:1C:1C;
  fixed-address 192.168.1.253;
  }
 host m254 {
  option host-name "m254.mi-red-local.com";
  hardware ethernet 00:01:03:DC:67:23;
  fixed-address 192.168.1.254;
  }
}
```

Descripción de las Opciones del Archivo

authoritative: Cuando hay dos servidores dhcp en la red el que tenga este parámetro es el que va a servir a la red y cuando este servidor este caído servirá a la red el otro servidor.

ddns-updates: Activa la actualización DNS mediante los valores asignados por DHCP.

ddns-update-style: Define el método de actualización automática de las DNS. Los valores pueden ser ad-hoc, interim y none.

default-lease-time: Especifica la cantidad de tiempo, en segundos, que será mantenida una asignación de direcciones, siempre y cuando el cliente no haya especificado algo concreto.

ignore allow / client-updates: Permite la actualización de las asignaciones (allow) de un cliente a requerimiento de este, o bien las asignaciones se actualizan cuando el servidor así lo requiera (ignore).

max-lease-time: Especifica la cantidad máxima de tiempo, en segundos, que será mantenida una asignación de direcciones. No está sujeta a esta especificación la asignación dinámica BOOTP.

netmask: Define la mascara de red de la Subred

not authoritative: La función de este parámetro es justo la contraria del anterior. Es decir: la configuración del servidor de DHCP no es concluyente y los clientes mal configurados que sean detectados por el servidor, seguirán con su configuración intacta.

one-lease-per-client: Cuando la opción se iguala a on y un cliente solicita una asignación de dirección (DHCPREQUEST), el servidor libera de forma automática cualquier otra asignación asociada a dicho cliente. Con esto se supone que si el cliente solicita una nueva asignación es porque ha olvidado que tuviera alguna, luego tiene un sólo interfaz de red. No dándose esta situación entre los clientes no es muy aconsejable el uso de esta opción.

option broadcast-address: Define la dirección de broadcast de la Red.

option domain-name-servers: Define el nombre de los servidores DNS.

option nis-servers: Define la lista de servidores NIS (Sun Network Information Server) disponibles. Los servidores se listan en orden de preferencia. Para establecer el nombre del dominio NIS, se usará option nis-domain <nombre>.

option routers: Define el router , gateway o pasarela de enlace listadas en orden de preferencia.

option subnet-mask: Definición de la máscara de subred general.

server-identifier: Identifica la máquina donde se aloja el servidor de DHCP. Su uso se aplica cuando la máquina en cuestión tiene varias direcciones asignadas en un mismo interfaz de red.

shared-network: Declaración de Subred compartida.

subnet: Declaración de Subred.

[root@trasgu ~]# **service dhcpd start**

Para que funcione las actualizaciones dinámicas de las zonas de DNS, de nuestro servidor DNS, en el fichero principal de configuración del DNS tiene que tener las siguientes entradas:

named.conf

```
zone "curso.esp"{
type master;
 allow-update {
 192.168.1.0/24;
};

zone "0.168.192.IN-ADDR.ARPA"{
 type master;
 allow-update {
  192.168.1.0/24;
};
 file "db.192.168.1";
};
```

Si estamos probando la configuración con un **cliente linux**, esta es la configuración correcta del fichero de red, para que pueda actualizar dinámicamente sin ningún problema:

vi /etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE=eth0
ONBOOT=yes BOOTPROTO=dhcp
DHCP_HOSTNAME=servidor1
HWADDR=00:0c:29:18:e4:a0
TYPE=Ethernet
```

Podemos probar la configuración DHCP, solicitándola desde el cliente:

```
# dhcpclient -r eth0
# service network restart
```

Importante:

Para que los ficheros de zona se puedan actualizar dinámicamente, desde nuestro servidor DCHP, en Centos, tendremos que darle permiso al siguiente directorio:

[root@trasgu var]# **cd /var/named/chroot/var**

[root@trasgu var]# **chmod 770 named**

[root@trasgu var]# **ls -ld named**/
drwxrwx--- 4 root named 4096 oct 29 13:02 named/

Esto permite que se creé en el directorio donde se almacenan las zonas **/var/named/chroot/var, los siguientes archivos:**

-rw-r--r-- 1 named named 791 oct 29 12:47 **db.curso.jnl**
-rw-r--r-- 1 named named 770 oct 29 12:47 **db.192.168.1.jnl**

Estos archivos nunca se tienen que manipular manualmente, es el servidor DHCP, el que se encarga de manipularlos.

[root@trasgu ~]# **tail -f /var/log/messages**
Oct 29 12:47:49 trasgu named[5164]: client 192.168.1.150#32773: updating zone 'curso.esp/IN': adding an RR at 'servidor1.curso.esp' A
Oct 29 12:47:49 trasgu named[5164]: client 192.168.1.150#32773: updating zone 'curso.esp/IN': adding an RR at 'servidor1.curso.esp' TXT
Oct 29 12:47:49 trasgu named[5164]: journal file db.curso.jnl does not exist, creating it
Oct 29 12:47:49 trasgu dhcpd: Added new forward map from servidor1.curso.esp to 192.168.1.90
Oct 29 12:47:49 trasgu named[5164]: client 192.168.1.150#32773: updating zone '0.168.192.IN-**ADDR.ARPA/IN': deleting rrset at '90.0.168.192.in-addr.arpa' PTR**
Oct 29 12:47:49 trasgu named[5164]: client 192.168.1.150#32773: updating zone '0.168.192.IN-ADDR.ARPA/IN': adding an RR at '90.0.168.192.in-addr.arpa' PTR
Oct 29 12:47:49 trasgu named[5164]: journal file db.192.168.1.jnl does not exist, creating it
Oct 29 12:47:49 trasgu named[5164]: zone curso.esp/IN: sending notifies (serial 2013102806)
Oct 29 12:47:49 trasgu dhcpd: added reverse map from 90.0.168.192.in-addr.arpa. to servidor1.curso.esp
Oct 29 12:47:49 trasgu named[5164]: zone 0.168.192.IN-ADDR.ARPA/IN: sending notifies (serial 2008012004)
Oct 29 12:47:49 trasgu dhcpd: DHCPREQUEST for 192.168.1.90 from 00:0c:29:d0:81:50 (servidor1) via eth1
Oct 29 12:47:49 trasgu dhcpd: DHCPACK on 192.168.1.90 to 00:0c:29:d0:81:50 (servidor1) via eth1

Seguridad añadida a nuestros servidores DNS BIND

La sentencia **acl** (o sentencia de control de acceso) define grupos de hosts a los que se les puede permitir o negar el acceso al servidor de nombres.

Una declaración acl tiene la siguiente forma:

acl miacl {

192.168.1.0/24;

127.0.0.1;

};

allow-transfer

{

miacl;

};

allow-query

{

miacl;

};

listen-on

{

miacl;

};

En esta declaración, sustituya <*acl-name*> con el nombre de la lista de control de acceso y reemplace <*match-element*> con una lista de direcciones IP separada por puntos y comas. La mayoría de las veces, una dirección IP individual o notación de red IP (tal como 10.0.1.0/24) es usada para identificar las direcciones IP dentro de la declaración acl.

La siguiente lista de control de acceso ya están definidas como palabras claves para simplificar la configuración:

- any — Hace coincidir todas las direcciones IP.
- localhost — Hace coincidir cualquier dirección IP que se use el sistema local.

- localnets — Hace coincidir cualquier dirección IP en cualquier red en la que el sistema local está conectado.

- none — No concuerda ninguna dirección IP.

Cuando lo utilice con otras pautas (tales como declaraciones options), las declaraciones acl pueden ser muy útiles al asegurar el uso correcto de su servidor de nombres BIND.

El ejemplo siguiente define dos listas de control de acceso y utiliza una declaración options para definir cómo son tratadas en el servidor de nombres:

```
acl denegados {
   10.0.2.0/24;
   192.168.0.0/24;
 };

acl red-hats {
   10.0.1.0/24;
 };

options {
   blackhole { dnegados; };
   allow-query { red-hats; };
   allow-recursion { red-hats; };
 }
```

Este ejemplo contiene dos listas de control de acceso, denegados y red-hats.

Los hosts en la lista denegados se les niega el acceso al servidor de nombres, mientras que a los hosts en la lista red-hats se les dará acceso normal.

En este otro ejemplo, solo se permite realizar consultas a los equipos declarados en la acl redlocal.

```
acl redlocal {

  //  10.0.2.0/24;

    192.168.0.150;

    127.0.0.1;

 };

options {

 directory "/var/named";

 forwarders{

  80.58.0.33;

 };


allow-query { redlocal; };

};
```

Seguridad adicional en DNS para uso público.

Un **DDoS** (**D**istributed **D**enial **o**f **S**ervice) es una ampliación del ataque **DoS**, se efectúa con la instalación de varios agentes remotos en muchas computadoras que pueden estar localizadas en diferentes puntos del mundo. El atacante consigue coordinar esos agentes para así, de forma masiva, amplificar el volumen del saturación de información (flood), pudiendo darse casos de un ataque de cientos o millares de computadoras dirigido a una máquina o red objetivo. Esta técnica se ha revelado como una de las más eficaces y sencillas a la hora de colapsar servidores, la tecnología distribuida ha ido haciendo más sofisticada hasta el punto de otorgar poder de causar daños serios a personas con escaso conocimiento técnico.

Un DNS configurado para permitir consultas recursivas indiscriminadamente puede permitir al servidor sufrir o bien participar de un **DDoS**. Solución al problema consiste añadir en el fichero **/etc/named.conf**, en la sección de opciones (options), el parámetro **allow-recursion** definiendo la red, las redes o bien los ACL que tendrán permitido realizar todo tipo de consultas en el DNS, sean locales o de otros dominios.

Fichero /etc/named.conf

```
options {
        directory "/var/named/";
        dump-file "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        allow-recursion {
                127.0.0.1;
                192.168.1.0/24;
        };
        forwarders {
                200.33.146.209;
                200.33.146.217;
        };
        forward first;
};
zone "." {
        type hint;
        file "named.ca";
};
zone "dominio.com" {
        type master;
        file "dominio.com.zone";
        allow-update { none; };
};
zone "1.243.148.in-addr.arpa" {
        type master;
        file "1.243.148.in-addr.arpa.zone";
        allow-update { none; };
};
```

Lo anterior hace que solo 192.168.1.0/24 pueda realizar todo tipo de consultas en el DNS, ya sea para un nombre de dominio hospedado de forma local y otros dominios resueltos en otros servidores, solo podrá realizar consultas sobre las zonas **dominio.com** y **1.243.148.in-addr.arpa**, que están hospedados de forma local.

Significa que el servidor DNS puede permitir a cualquiera realizar consultas recursivas. Si se trata de un DNS que se desea pueda ser consultado por cualquiera, como puede ser el caso del DNS de un ISP, esto es normal y esperado. Si se trata de un servidor que sólo debe consultar la red local o bien que se utiliza para propagar dominios alojados de manera local, si es conveniente tomar medidas al respecto.

Otra solución al problema si estamos utilizando view en BIND

Solución al problema es modificar el archivo **named.conf**, donde se añade en la sección de vista local (view "local") la opción **recursion yes;** y una o más líneas que definan la red o las redes que tendrán permitido realizar todo tipo de consultas.

```
options {
        directory "/var/named";
                dump-file "/var/named/data/cache_dump.db";
                statistics-file "/var/named/data/named_stats.txt";
                memstatistics-file "/var/named/data/named_mem_stats.txt";
        forwarders { 192.168.70.1; };
        forward first;
                dnssec-enable yes;
                dnssec-validation yes;
                dnssec-lookaside auto;
                bindkeys-file "/etc/named.iscdlv.key";
};

logging {
                channel default_debug {
                                file "data/named.run";
                                severity dynamic;
                };
};

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

include "/etc/rndc.key";

view "local" {
                match-clients {
                                127.0.0.0/8;
                                10.0.0.0/8;
                                172.16.0.0/12;
                                192.168.0.0/16;
                };
                recursion yes;
                include "/etc/named.rfc1912.zones";
                zone "." IN {
                                type hint;
                                file "named.ca";
                };
};
```

Lo anterior hace que sólo se puedan realizar todo tipo de consultas en el DNS desde 127.0.0.0/8, 10.0.0.0/8, 172.16.0.0/12 y 192.168.0.0/16, ya sea para un nombre de dominio alojado de manera local y otros dominios resueltos en otros servidores (ejemplo: *www.yahoo.com*, *www.google.com*, etc). El resto del mundo sólo podrá realizar consultas sobre los dominios alojados de manera local y que estén configurado para permitirlo.

En la **siguiente** configuración de **ejemplo**, se pretende lograr lo siguiente:

- Red Local: cualquier tipo de consulta hacia dominios externos y locales (es decir, www.yahoo.com, www.google.com, curso.esp, además de ***midominio.com***).

- Resto del mundo: sólo puede hacer consultas para la zona de ***midominio.com***

De este modo se impide que haya consultas recursivas y con esto impedir la posibilidad de sufrir/participar de un ataque DDoS.

```
options {
        directory "/var/named";
                dump-file "/var/named/data/cache_dump.db";
                statistics-file "/var/named/data/named_stats.txt";
                memstatistics-file "/var/named/data/named_mem_stats.txt";
        forwarders {192.168.0.1; };
        forward first;
                dnssec-enable yes;
                dnssec-validation yes;
                dnssec-lookaside auto;
                bindkeys-file "/etc/named.iscdlv.key";

};

logging {
                channel default_debug {
                                file "data/named.run";
                                severity dynamic;
                };
};

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

include "/etc/rndc.key";

view "publico" {
                match-clients { any; };
                recursion no;
                zone "." IN {
                                type hint;
                                file "named.ca";
                };
                zone "midominio.com" {
                     type master;
                     file "data/midominio.com.zone";
                     allow-update { none; };
                    allow-transfer { 204.13.249.75; 208.78.69.75; 91.198.22.75; };
                };
};

view "local" {
                match-clients {
                                127.0.0.0/8;
                                10.0.0.0/8;
                                172.16.0.0/12;
                                192.168.0.0/16;
                };
                recursion yes;
                include "/etc/named.rfc1912.zones";
                zone "." IN {
                                type hint;
                                file "named.ca";
                };
```

```
        zone "curso.esp" {
            type master;
            file "db.curso";
            allow-update { none; };
            allow-transfer { 192.168.0.2; };
        };
};
```

Seguridad adicional en DNS para uso exclusivo en red local.

Si se va a tratar de un servidor de nombres de dominio para uso exclusivo en red local, y se quieren evitar problemas de seguridad de diferente índole, puede utilizarse el parámetro **allow-query**, el cual servirá para especificar que solo ciertas direcciones podrán realizar consultas al servidor de nombres de dominio. Se pueden especificar directamente direcciones IP, redes completas o listas de control de acceso que deberán definirse antes de cualquier otra cosa en el fichero **/etc/named.conf**.

Fichero /etc/named.conf

```
acl "redlocal" {
                    127.0.0.1;
                    192.168.1.0/24;
                    192.168.2.0/24;
                    192.168.3.0/24;
};

options {
        directory "/var/named/";
        dump-file "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        allow-recursion { redlocal; };
        forwarders {
                    200.33.146.209;
                    200.33.146.217;
        };
        forward first;
        allow-query {
                    redlocal;
```

```
                        192.168.1.15;
                        192.168.1.16;
                };
};
zone "red-local" {
                type master;
                file "red-local.zone";
                allow-update { none; };
};
zone "1.168.192.in-addr.arpa" {
                type master;
                file "1.168.192.in-addr.arpa.zone";
                allow-update { none; };
};
```

Laboratorio Configurar bind logging con bind-chroot

En este laboratorio, configuraremos el loggind, del servicio de bind, para que nos recoja en un log, las consultas, y el resto de trazas en otro fichero log.

#ln -sf /var/named/chroot/var/log/dns.log /var/log/dns.log

#ln -sf /var/named/chroot/var/log/dns_queries.log /var/log/dns_queries.log

/etc/named.conf:

//Añadimos lo que tenemos en negrita

```
options {
 directory "/var/named";
 forwarders{
  80.58.0.33;

 };

allow-query { redlocal; };
 allow-transfer { key curso.esp; };

};
```

```
logging {
        channel log_dns {
                file "/var/log/dns.log" versions 3 size 10m;
                print-category yes;
                print-severity yes;
                print-time yes;
        };
        channel log_queries {
                file "/var/log/dns_queries.log" versions 3 size 20m;
                print-category yes;
                print-severity yes;
                print-time yes;
        };
        category default {log_dns;};
        category queries {log_queries;};
        category lame-servers { null;};
        category edns-disabled { null; };
};
```

 # systemctl restart named-chroot

A continuación, tendremos que verificar que los archivos de log, estan recibiendo la información correctamente.

Category	Description
client	Processing of client requests.
config	Configuration file parsing and processing.
database	Messages relating to the databases used internally by the name server to store zone and cache data.
default	Logs all values which are not explicitly defined in category statements i.e. if this is the only category defined it will log all categories listed in this table with the exception of queries which are not turned on by default.
delegation-only	Logs queries that have returned NXDOMAIN as the result of a delegation-only zone or a delegation-only statement in a hint or stub zone declaration.
dispatch	Dispatching of incoming packets to the server modules where they are to be processed.
dnssec	DNSSEC and TSIG protocol processing.
general	Anything that is not classified as any other item in this list defaults to this category..
lame-servers	Lame servers. Mis-configuration in the delegation of domains discovered by BIND 9 when trying to authoritative answers. If the volume of these messages is high many users elect to send them to the null channel e.g. category lame-servers {null;}; statement.
network	Logs all network operations.
notify	Logs all NOTIFY operations.
queries	Logs all query transactions. The querylog statement may be used to override this category statement. This entry can generate a substantial volume of data very quickly. This category is not turned on by default and hence the default type above will not log this information.
resolver	Name resolution including recursive lookups performed on behalf of clients by a caching name server.
rpz	All operations related to Response Policy Zone (RPZ) processing. Even when RPZ zones are disabled (using **policy disabled** parameter in the response-policy statement) the operation is completed, logged then discarded (the real response is returned to the user).
rate-limit	All operations related to one or more rate-limit statements in the options or view clauses.
security	Approval and denial of requests.
unmatched	No matching view clause or unrecognized class value. A one line summary is also logged to the client category. By default this category is sent to the null channel.

update	Logging of all dynamic update (DDNS) transactions.
update-security	Approval and denial of update requests used with DDNS.
xfer-in	Details of zone transfers the server is receiving.
xfer-out	Details of zone transfers the server is sending.

Severity	Description
critical	only critical errors.
error	error and above.
warning	warning and above.
notice	notice and above.
info	info and above - log starting to get chatty.
debug	debug and above. Various debug levels can be defined with 'debug 0' meaning no debugging.
dynamic	debug and above. Means assume the global debug level defined by either the command line parameter **-d** or by running **rndc trace**