Iptables y el servicio de Traduccion de Direcciones de Redes (NAT)

Por Haller Javier Bracho Hernandez
hbracho@linux.org.ve

Objetivo: Implementar un router/firewall con NAT

Que es IPTABLES ?

Iptables, se usa para configurar, mantener e inspeccionar las reglas de cortafuegos IP del nucleo Linux. Es un descendiente directo de ipchains (que vino de ipfwadm, que vino del ipfw IIRC de BSD), con extensibilidad. Los módulos del kernel pueden registrar una tabla nueva, e indicarle a un paquete que atraviese una tabla dada. Este método de selección de paquetes se utiliza para el filtrado de paquetes, para la **Traducción de Direcciones de Red (NAT)** y para la manipulación general de paquetes antes del enrutamiento. Una de las ventajas de iptables sobre ipchains es que es pequeño y rápido.

Razones para implementar NAT

El servicio de Traducción de Direcciones de Redes IP es algo así como el hermano mayor estandarizado del servicio de Enmascarado IP de Linux. NAT proporciona características que no posee el Enmascarado IP que lo hacen eminentemente más apropiado para su uso en los diseños de encaminamiento de cortafuegos corporativos y en instalaciones a mayor escala.

Las razones principales para implementar NAT son:

1.- Conexiones con módem a Internet o por banda ancha

La mayoría de los ISP (Proveedor de Servicios de Internet) le dan una sola dirección IP cuando se conecta con ellos. Puede enviar paquetes con cualquier dirección que le plazca, pero sólo obtendrá respuestas a los paquetes con esa IP de origen. Si desea utilizar varias máquinas diferentes (como una red casera) para conectar a Internet a través de un enlace, necesita NAT.

Este es, de lejos, el uso más común de NAT hoy en día, conocido normalmente como «enmascaramiendo» (masquerading) en el mundo de Linux. Algunos prefieren llamarle SNAT, porque se cambia la dirección de origen (source) del primer paquete.

2.- Varios servidores

Puede que quiera cambiar el destino de los paquetes que entran en su red. Con frecuencia esto se debe (como antes), a que sólo tiene una dirección IP, pero desea que la gente sea capaz de llegar a las máquinas detrás de la que tiene la IP «real». Si reescribe el destino de los paquetes entrantes, podrá conseguirlo.

Una variante común de esto es el balanceo de carga, en la cual se toma un cierto número de máquinas, repartiendo los paquetes entre ellas. Este tipo de NAT se llamó reenvío de puerto (port-forwarding) en anteriores versiones de Linux.

3.- Proxy transparente

Hay veces que deseará simular que cada paquete que pase por su máquina Linux esté destinado a un programa en la propia máquina. Esto se utiliza para hacer proxies transparentes: un proxy es un programa que se pone entre su red y el mundo real, filtrando las comunicaciones entre ambos. La parte transparente se debe a que su red nunca tendrá por qué enterarse de que está comunicándose con un proxy, a menos, claro, que el proxy no funciones.

Se puede configurar Squid para que trabaje de esta manera, y a esto se le llamó redirección o proxy transparente en anteriores versiones de Linux.

Como usar iptables

Necesita crear reglas NAT que le digan al núcleo qué conexiones cambiar, y cómo hacerlo. Para ello, usaremos la muy versátil herramienta iptables, y le diremos que altere la tabla de NAT usando la opción «-t nat».

La tabla de reglas NAT contiene tres listas llamadas «cadenas»: cada regla se examina por orden hasta que una coincide. Las tres cadenas se llaman PREROUTING (para Destination NAT, según los paquetes entran), POSTROUTING (para SOURCE NAT, según los paquetes salen), y OUTPUT (para Destination NAT con los paquetes generados en la propia máquina).

Iptables toma cierto número de decisiones estándar que se listarán ahora. Todas las opciones con doble guión pueden ser abreviadas, siempre que iptables pueda distinguirlas de otras opciones posibles. Si el núcleo tiene la implementación de iptables como módulo, necesitará cargar el módulo ip_tables.o antes: «modprobe ip_tables».

La opción más importante aquí es la opción de selección de tabla, «-t». Para todas las operaciones de NAT, querrá usar «-t nat» para la tabla NAT. La segunda más importante es «-A» para añadir una nueva regla al final de una cadena («-A POSTROUTING»), o «-I» para insertarla al principio («-I PREROUTING»).

Puede especificar el origen («-s» o «--source») y el destino («-d» o «--destination») de los paquetes sobre los que quiere hacer NAT. Estas opciones pueden ir seguidas por una IP sencilla (192.168.1.1), un nombre (www.linux.org), o una dirección de red (192.168.1.0/24 o 192.168.1.0/255.255.255.0).

Puede especificar qué interfaz de entrada («-i» o «--in-interface») o de salida («-o» o «--out-interface») mirar, pero lo que puede especificar depende de en qué cadena esté poniendo la regla: en PREROUTING sólo puede elegir la interfaz de entrada, y en POSTROUTING (y OUTPUT) sólo la de salida. Si usa la equivocada, iptables le avisará con un mensaje de error.

Dije antes que se puede especificar una dirección de origen y destino. Si omite la opción de origen, entonces será cualquier dirección de origen. Si omite la de destino, será cualquier dirección de destino.

También puede indicar un protocolo específico («-p» o «--protocol»), como TCP o UDP; sólo los paquetes de este protocolo coincidirán con la regla. La razón principal para hacer esto es que especificar uno de los protocolos tcp o udp permite más opciones: específicamente las opciones «--source-port» y «--destination-port» (abreviadas «--sport» y «--dport»).

Estas opciones le permiten especificar que sólo los paquetes con un determinado origen y destino coincidirán con la regla. Esto es útil para redireccionar peticiones web (puertos TCP 80 u 8080) y dejar los demás paquetes tranquilos.

Estas opciones deben seguir a la «-p» (que tiene el efecto secundario de cargar la biblioteca compartida de extensión para ese protocolo). Puede usar números de puerto, o un nombre de fichero /etc/services.

Todos los diferentes parámetros por los que se puede seleccionar un paquete vienen enumerados con toda clase de dolorosos detalles en la página de manual (man iptables).

Enmascaramiento

Esto es lo que la mayoría de la gente quiere. Si tengo una conexión PPP con IP dinámica simplemente querré decirle a mi máquina que todos los paquetes que salgan de la red interna deberían aparentar salir de la máquina que tiene el enlace PPP.

```
# Agrega (-A) una regla a la tabla NAT (-t nat), después del encaminamiento
(POSTROUTING) para todos los paquetes que salgan por ppp0 (-o ppp0) enmascarando la
conexión (-j MASQUERADE).
iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE

# Poner en marcha el reenvío de IP (IP forwarding)
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Si la salida a internet la tengo por la interfaz de red eth1, entonces la instruccion quedará asi:

```
# Enmascarar todo lo que salga por eth1
iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

Reenvío de puertos

Puede usar «iptables -t nat» para hacer reenvío de puertos en caso de querer colocar, por ejemplo, un servidor web en la red interna (con un ip no certificado) pero que se vea desde internet.

```
# Agrega una regla previa al encaminamiento (-A PREROUTING) a la tabla NAT (-t nat) de
manera que los paquetes TCP (-p tcp) que vayan a 1.2.3.4 (-d 1.2.3.4), puerto 80 (--dport 80)
tengan una correspondencia de destino (-j DNAT) con 192.168.1.1, puerto 80 (--to
192.168.1.1:80).
iptables -A PREROUTING -t nat -p tcp -d 1.2.3.4 --dport 80 -j DNAT --to 192.168.1.1:80
```

Si desea que esta regla altere también las conexiones locales (aquellas que se originen en la propia máquina que hace NAT), puede insertar la misma regla en la cadena OUTPUT (que es para los paquetes locales de salida):

```
# Linux 2.4
iptables -A OUTPUT -t nat -p tcp --dport 80 -j DNAT --to 192.168.1.1:80
```

Proxy Transparente

Como dijimos anteriormente, un proxy es un programa que se pone entre su red y el mundo real, filtrando las comunicaciones entre ambos. No vamos a profundizar en éste tema. En pocas palabras:

```
# Envía el tráfico que entra dirigido al puerto 80 (web) a nuestro proxy squid (transparente)
iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 -j REDIRECT --to-port 3128
```

Firewall

Un cortafuegos en el mundo de las redes de computacionales es un dispositivo lógico que protege una red privada del resto de la red (pública). Una manera sencilla y muy básica de implementar un cortafuegos sería de la siguiente manera:

```
iptables -A INPUT -j DROP
iptables -A FORWARD -j DROP
```

Estas dos instrucciones impiden el acceso a la máquina y cualquier acceso que se haga a traves de ella. Estas instrucciones sellan "hermeticamente" la maquina. Pero hay que tener cuidado pues si éste es la única regla, entonces ni siquiera se podra hacer login ! Para evitar este problema le indicaremos al firewall que todas las conexiones que se hagan localmente (interfaz loopback) las acepte. Esto lo haremos asi:

```
iptables -I INPUT -d 127.0.0.1 -i lo -j ACCEPT
```

Estas tres reglas implementan un firewall muy simple y básico. Desde luego, con estas reglas, la red interna que estamos protegiendo no tendrá salida a internet por lo que debemos hablilitar otras cosas, ademas de que la administracion remota del firewall será imposible. Tambien hay que tomar en cuenta la protección contra ciertos ataques (syn flooding, DoS, etc).

Veamos algunas ejemplos:

```
iptables -A INPUT -s 192.168.0.0/24 -i eth0 -j ACCEPT
```

Esta instrución significa que acepte cualquier paquete que venga de la red 192.168.0.0/24 por la interfaz de red eth0.

```
iptables -A INPUT -s 172.16.0.0/16 -p icmp -j DROP
```

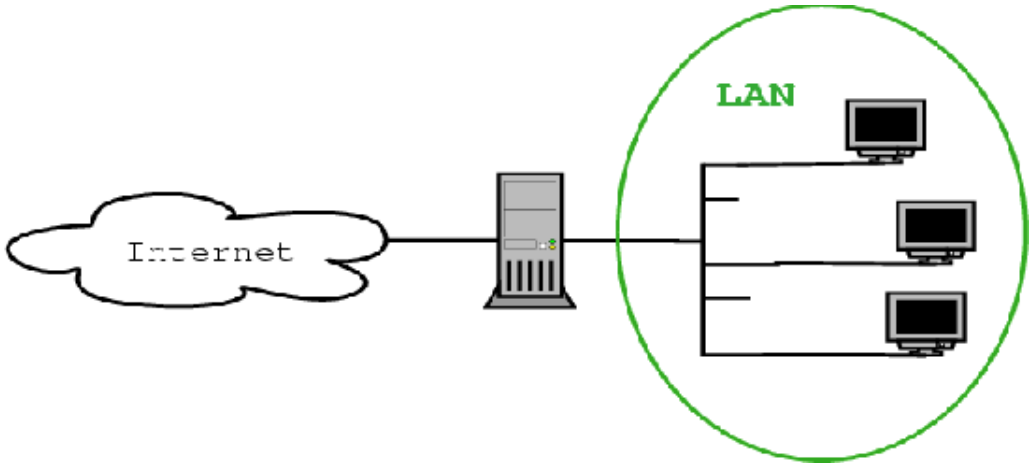
Niega cualquier paquete del protocolo icmp (ping) que venga de la red 172.16.0.0/16.

```
iptables -A FORWARD -s 192.168.1.0/24 -d 192.168.1.0/24 -i eth1 -j ACCEPT
```

Acepta cualquier paquete que venga y vaya a la misma red (192.168.1.0/24) por la interfaz eth1.

Ejercicio práctico

Vamos a crear un archivo llamado rc.firewall que solucione la siguiente situación:



Se dispone de 32 estaciones de trabajo y tres servidores. Se toma la red 192.168.0.0/24 para identificar a todas las maquinas y se distribuyen de la siguiente manera:

Desde 192.168.0.11 - 192.168.0.42 para las Estaciones.
192.168.0.5 para el Servidor web.
192.168.0.4 para el Servidor de correo.
192.168.0.3 para elServidor ftp.
192.168.0.2 queda apartado para una futura impresora en red.
192.168.0.1 para la interfaz interna del Firewall (eth0).
200.200.200.200 es el único IP certificado que nos asigno nuestro ISP y que se asignara a la interfaz externa del firewall (eth1).

Las reglas a implantar son:

1. Los tres servidores deben verse desde internet
2. Se debe hablilitar el acceso a internet a todas las maquinas de la LAN
3. Cualquier otro servicio debe ser denegado por el firewall
4. La administracion remota del firewall solo la puede realizar la maquina con el IP 150.150.150.150

Nuestro archivo rc.firewall será parecido a lo siguiente:

```
# Primero pongamos a funcionar nuestro firewall:

iptables -A INPUT -j DROP
iptables -A FORWARD -j DROP
iptables -I INPUT -d 127.0.0.1 -i lo -j ACCEPT
iptables -I INPUT -s 150.150.150.150 -j ACCEPT

# Ahora enmascaremos las maquinas y démosle salida a internet:

iptables -I FORWARD -s 192.168.0.0/24 -i eth0 -j ACCEPT
iptables -t nat  -A POSTROUTING -o eth1 -j  MASQUERADE
```

Luego filtremos (reenviemos los puertos a los servidores respectivos; puerto 80 a 192.168.0.5, 25 y 110 para 192.168.0.4 y 21 para 192.168.0.3)

```
iptables -A PREROUTING -t nat -p tcp -d 200.200.200.200 --dport 80 -j DNAT --to 192.168.0.5:80
iptables -A PREROUTING -t nat -p tcp -d 200.200.200.200 --dport 25 -j DNAT --to 192.168.0.4:25
iptables -A PREROUTING -t nat -p tcp -d 200.200.200.200 --dport 110 -j DNAT --to 192.168.0.4:110
iptables -A PREROUTING -t nat -p tcp -d 200.200.200.200 --dport 21 -j DNAT --to 192.168.0.3:21

iptables -A OUTPUT -t nat -p tcp  --dport 80 -j DNAT --to 192.168.0.5:80
iptables -A OUTPUT -t nat -p tcp  --dport 25 -j DNAT --to 192.168.0.4:25
iptables -A OUTPUT -t nat -p tcp  --dport 110 -j DNAT --to 192.168.0.4:110
iptables -A OUTPUT -t nat -p tcp  --dport 21 -j DNAT --to 192.168.0.3:21
```

Para que el ftp funciones correctamente debemos activar ciertos modulos:

```
modprobe  ip_conntrack_ftp
modprobe  ip_nat_ftp
```

Por último ponemos en marcha el reenvío de IP (IP forwarding)

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

#La usurpacion de la direccion IP (Spoofing en ingles) consiste en hacerle creer a uno que un paquete que viene del mundo externo viene de la interface por el cual llega. Esta técnica es muy usada por los crackers, pero usted puede hacer que el kernel prevenga este tipo de intrusion.
#Solo con ingresar la siguiente línea se vuelve imposible este tipo de ataque.

```
echo 1 > /proc/sys/net/ipv4/conf/all/rp_filter
```

Este es un ejemplo sencillo de como resolver la situacion antes expuesta. Para una mayor amplitud sobre los temas de cortafuegos, NAT, proxies, etc. pueden consultar los diferentes HOWTO de Linux:

- Firewall HOWTO
- NAT HOWTO
- IPTABLES HOWTO
- Transparent Proxy HOWTO
- etc (man iptables)

NOTA: Este documento es una recopilación de los distintos HOWTO que existen sobre el tema.