Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

Haciendo funcionar el Policy Routing junto a DNAT (22153 lectures)

Per **Eduard Llull**, ***Daneel*** ()

Creado el 30/01/2005 21:21 modificado el 30/01/2005 21:21

Los que hayais intentado dar servicio por dos líneas en un linux que también redirige algunos puertos (DNAT) a máquinas internas seguro que os habreis encontrado con el problema de que las conexiones del exterior hacia una de las IPs públicas funciona, pero las de la otra IP pública, no. ¡Y eso que teneis el policy routing y las reglas de IPTables bien configuradas!

En este artículo os muestro la solución que yo he encontrado.

Los que os esteis encontrando este problema podeis saltaros la introducción e ir directamente a la sección Solución 2.

¿Qué es policy routing?

El enrutado de los paquetes en redes IP se decide por la dirección destino de dicho paquete. Pero en ocasiones nos podemos encontrar con que tenemos dos líneas conectadas al mismo firewall, y nos puede interesar enrutar el trafico salliente del firewall hacia una línea u otra según la direccion origen, o, si tenemos una línea de mayor ancho de banda pero también con mayor latecia, nos puede interesar enrutar trafico hacia una línea o la otra segun el ToS. En resumidas cuentas, el policy routing es el proceso de enrutar los paquetes por cualquier otro dato que no sea la dirección destino del paquete.

¿Y DNAT?

DNAT son las siglas de Destination Network Address Translation. Muy por encima, y en el caso que nos ocupa en este artículo, el DNAT nos permite redirigir puertos hacia máquinas internas. Para más información podeis visitar esta introducción al NAT[1].

Esquema del montaje

```
   /----\      ISP1    +----------+
 /-       --\    +----+          |
/ Internet |----|    | Firewall |----- Red Interna
\          -/    +----+          |
 \-------/     ISP2    +----------+
```

En el esquema tenemos una red, llamemosle interna, conectada a Internet por dos líneas/ISPs. El firewall es un GNU/linux con IPTables.

Lo que queremos conseguir es que se puedan usar las dos líneas para que usuarios se puedan conectar a nuestros servidores (ya sean web, de correo, etc.)

Entrando un poco más en detalle, alrededor del firewall tenemos dos LANs, la pública y la privada. Supondremos las siguientes direcciones:

- Direcciones red pública:
  - Router ISP1: 10.0.0.1/24
  - Router ISP2: 10.0.1.1/24

- Firewall eth0 (ISP1): 10.0.0.2/24
- Firewall eth0:0 (ISP2): 10.0.1.2/24

- Direcciones red privada:
  - Firewall eth1: 192.168.0.1/24
  - Servidor: 192.168.0.10/24

***Nota**: Yo siempre que puedo utilizo la eth0 como la red externa y la eth1 como la interna. La regla pnemotécnica es: de eth0, el 0 se parece a O de Outside, de eth1, el 1 se parece a la I de Inside :-)*

Preliminares

Para poder utilizar el policy routing es necesario tener las correspondientes opciones compiladas en el kernel, ya sea dentro del mismo o como módulos. Para el NAT, hay que compilar tamben el soporte de netfiler. Importante, porque lo necesitaremos más adelante, compilar el soporte para "Connection tracking march support".

Opciones imprescindibles:

```
Device Drivers  --->
  Networking support  --->
    [*] Networking support
        Networking options  --->
          [*] TCP/IP networking
          [*]   IP: advanced router
          [*]     IP: policy routing
          [*]       IP: use netfilter MARK value as routing key
          [*] Network packet filtering (replaces ipchains)  --->
                IP: Netfilter Configuration  --->
                   Connection tracking (required for masq/NAT)
                   IP tables support (required for filtering/masq/NAT)
                           netfilter MARK match support
                           Connection tracking match support
                           Full NAT
                         Packet mangling
                           MARK target support
```

Redirigiendo el puerto 80/tcp

En este artículo no me meteré en cuestiones de filtrado. Supondremos que no filtramos ningun puerto. Si se quiere hacer filtrado de paquetes, hay otros artículos en Bulma que hablan sobre esto (como este[2], por ejemplo). Redirigimos el puerto 80:

```
# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT
--to=192.168.0.10
```

No hemos indicado la IP destino porque queremos redirigir el puerto tanto si se conectan a la 10.0.0.2 como a la 10.0.1.2.

Configurando el policy routing

Como base, esta muy bien el apartado 4.2 del Linux Advanced Routing & Traffic Control HOWTO[3]. Siguendo dicho documento configuramos lo siguiente:

Añadimos los nombres de las tablas para poder trabajar con nombres descriptivos en lugar de con números

```
 # echo 101 isp1 >> /etc/iproute2/rt_tables
# echo 102 isp2 >> /etc/iproute2/rt_tables
```

Rutas para la línea del ISP1

```
 # ip route add 10.0.0.0/24 dev eth0 src 10.0.0.2 table isp1
# ip route add default via 10.0.0.1 table isp1
```

Rutas para la línea del ISP2

```
 # ip route add 10.0.1.0/24 dev eth0 src 10.0.1.2 table isp2
# ip route add default via 10.0.1.1 table isp2
```

Rutas principales

```
 # ip route add 10.0.0.0/24 dev eth0 src 10.0.0.2
# ip route add 10.0.1.0/24 dev eth0 src 10.0.1.2
# ip route add default via 10.0.0.1 (o 10.0.1.1 si queremos que por defecto se
envie el trafico por la línea del ISP2)
```

Finalmente añadimos las reglas de policy routing, lo que salga del router con origen 10.0.0.2 lo enviamos hacia el router del ISP1 y lo que salga con origen 10.0.1.2, lo enviamos al router del ISP2.

```
 # ip rule add from 10.0.0.2 table isp1
# ip rule add from 10.0.1.2 table isp2
```

No hemos hecho nada más que seguir el ejemplo del HOWTO.

Para comprobar el policy routing, usaremos el traceroute indicandole con que IP origen queremos que haga las trazas:
traceroute -n -s 10.0.0.2 www.cisco.com (debería salir por la linea del ISP1)
traceroute -n -s 10.0.1.2 www.cisco.com (debería salir por la linea del ISP2)

Primer intento: no funciona

Ahora, si desde una máquina externa nos conectamos al puerto 80 de 10.0.0.2 con un navegador, debería salir la web que tenemos en el servidor de la red interna. Pero si intentamos hacer lo mismo con la IP 10.0.1.2, no nos funcionará

Con un 'tcpdump -e -i eth0', si nos fijamos en las MACs, veremos que el firewall está enviando los paquetes con dirección origen 10.0.1.2 hacia el router que no toca, el 10.0.0.1. ¿Y como es eso? ¿A caso no hemos comprobado anteriormente el policy routing desde el firewall y había funcionado? La pequeña gran diferencia es que en este caso tenemos un DNAT.

Explicación

Para entender el motivo por el que no funciona hay que tener claro el camino que sigue el paquete dentro del kernel del firewall. El siguiente esquema está sacado del IPTables tutorial[4]:

Cuando un paquete va del cliente al servidor, lo primero que se encuentra (que sea relevante para esta explicación) es la lista PREROUTING de la tabla nat, donde se hace el DNAT. En el DNAT se cambia la dirección destino del paquete, y al enrutarse posteriormente, como el kernel ve que va dirigido al servidor, mete el paquete hacia la red interna.

El problema está en que cuando las respuestas del servidor hacia el cliente pasan por el firewall, no se deshace el DNAT antes del enrutado. El kernel primero enrutará el paquete y luego le cambiará la IP origen, deshaciendo el DNAT. ¡Pero nuestro policy routing estaba enrutando según la dirección origen! Ahí esta el problema, no podemos enrutar por dirección origen los paquetes que son respuesta de otro que sufrió un proceso de DNAT.

Solución 1: añadir IPs en los servidores

Una solución que he visto en algunos foros pasa por añadir una segunda IP al servidor. Entonces deberíamos cambiar la regla de DNAT para que el firewall enviara los paquetes que hubiera recibido por el ISP1 a una de las IPs del servidor, y los que hubiera recibido por el ISP2, a la segunda IP. Luego podríamos enrutar según la IP del servidor que respondiera. Pero esta solución no me gusta porque implica complicar la configuración de los servidores, y también porque me gusta mantener la arquitectura de la red lo más independiente de los servidore que pueda, es decir, que sea transparente para los servidores. Como creo que hay otra solución mejor no me voy a extender más sobre esta.

Solución 2: enrutar por marcas gracias al modulo conntrack de IPTables

En "Enrutado en base a marcas de paquetes. Iproute + Iptables."[5] , se explica como enrutar paquetes mediante su marcado. Ahora sólo nos hace falta saber como podemos marcar esos paquetes para que el policy routing funcione como queremos. Por suerte, alguien creo el modulo 'connection track match' para IPTables (no confundir con el modulo conntrack que se encarga de mantener unas tablas en el kernel para poder deshacer el DNAT). Yo lo he probado con un kernel 2.4.20 en una debian woody, pero me he tenido que instalar las IPTables de backports por que el paquete oficial no soportaba las reglas de conntrack.

De las características del modulo conntrack, a nosotros nos interesa la capacidad de poder marcar los paquetes según la dirección destino que tenían los paquetes que iniciaron la conexión cuando estos llegaron al firewall provenientes del cliente, antes de aplicarle el DNAT. Es decir, según la dirección a la que el cliente se estaba intentando conectar. Las reglas de IPTables que nos solucionaran el entuerto son:

```
 # iptables -t mangle -A PREROUTING -m conntrack --ctorigdst 10.0.0.2 -j MARK
--set-mark=1
# iptables -t mangle -A PREROUTING -m conntrack --ctorigdst 10.0.1.2 -j MARK
--set-mark=2
```

Ahora solo hace falta que enrutemos según la marca del paquete:

```
 # ip rule add fwmark 1 table isp1
# ip rule add fwmark 2 table isp2
```

Comprobación 2: ¡ahora sí!

Ahora podemos volver a intentar conectarnos al puerto 80, tanto por una IP como por la otra que ya funcionará.

Lista de enlaces de este artículo:

1. http://linux-ip.net/html/nat-overview.html
2. http://bulma.net/body.phtml?nIdNoticia=1522
3. http://lartc.org/howto/lartc.rpdb.multiple-links.html
4. http://iptables-tutorial.frozentux.net/iptables-tutorial.html
5. http://bulma.net/body.phtml?nIdNoticia=1615

E-mail del autor: daneel _ARROBA_ bulma.net
Podrás encontrar este artículo e información adicional en: http://bulma.net/body.phtml?nIdNoticia=2145