Práctica 7 – Network Address Translation en routers GNU/Linux empleando iptables

1- Objetivos

NAT permite que una red IP parezca hacia el exterior que emplea un espacio de direcciones diferente del que en realidad usa. La utilidad más típica es hacer que una red que emplea direccionamiento privado pueda acceder a Internet convirtiendo las direcciones IP en los paquetes que envía a direcciones públicas.

En esta práctica vamos a ver conceptos básicos sobre cómo configurar NAT en lo routers GNU/Linux.

2- Introducción a iptables

Consulte la página del manual de iptables[1], en especial las secciones referentes a la tabla 'nat', que es la que vamos a utilizar. Solo el primer paquete de cada flujo llega a esta tabla, lo suficiente para crear el mapeo de traducción de direcciones.

Consulte la página del manual sobre los módulos de extensión[2], en especial las secciones referentes al target 'SNAT'.

Puede buscar tutoriales sobre iptables que le aclaren con dibujos la organización de tablas y cadenas[3].

3- Conversión estática

Creen la simple topología de la figura 1. En este primer apartado emplearemos solo un host de cada subred pero en el siguiente emplearemos más.

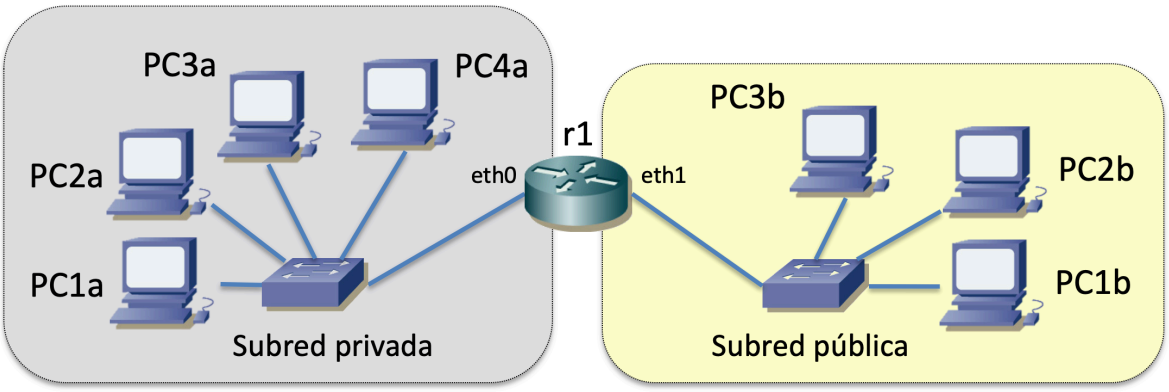

Figura 1.- Dos redes

La subred privada representará el interior de la empresa con direccionamiento privado, donde emplearemos el espacio de direcciones 192.168.1.0/24. La subred pública representará el exterior, la primera red pública a la que salga el tráfico. De momento restringiremos el universo a solo estas dos redes. Para la subred pública emplearemos el espacio de direcciones 10.0.0.0/24 (aunque oficialmente sea un bloque de

[1] `man iptables`

[2] `man iptables-extensions`

[3] https://wiki.archlinux.org/index.php/Iptables

direccionamiento privado).

- Asignen dirección a los interfaces del router y a los hosts, así como router por defecto a estos últimos. Asigne 192.168.1.1 al PC1a.

Si prueban a hacer ping entre PC1a y PC1b debería funcionar. En el caso de que realmente la red interior emplease direccionamiento privado y la red exterior fuera Internet entonces el router no debería reenviar estos paquetes. El router incluiría algún tipo de reglas de filtrado que evitasen este reenvío. En GNU/Linux es común que estos filtrados se configuren con iptables, incluyendo nuevas reglas en la tabla *filter*.

Vamos a hacer que el router cambie la direccción del PC1a en los paquetes que reenvíe. Para ello:

- Establezcan una regla de conversión estática con el comando:

```
# iptables --table nat --append POSTROUTING --source 192.168.1.1 ––out-
interface eth1 --jump SNAT --to-source 10.0.1.1
```

Consulte las secciones del manual que se han comentado en el comienzo de la práctica. En este comando estamos añadiendo (--append) una nueva regla en la cadena POSTROUTING de la tabla (--table) 'nat'. Esta tabla se emplea para crear los mapeos de la traducción de direcciones. La cadena POSTROUTING se consulta cuando el paquete ya ha pasado por la tabla de rutas, con lo que se sabe por qué interfaz va a salir. Con la opción --out-interface estamos indicando que nos interesan los paquetes que van a salir por el interfaz eth1 (que es el conectado a la subred pública) y con --source los que vienen de la dirección IP 192.168.1.1. Lo que queremos hacer con los paquetes que cumplen la regla anterior es pasarlos al target llamado SNAT. La documentación sobre este target está en la página de manual sobre extensiones y ahí vemos que la opción --to-source nos permite indicar a qué dirección queremos hacer el cambio de la dirección origen del paquete.

En este caso estamos pidiendo que se cambie la dirección IP origen de los paquetes que vengan de 192.168.1.1 y vayan a salir por eth1, a la dirección 10.0.1.1. Esta regla hace que una vez salga un paquete que la cumpla se cree un mapeo que pueden emplear también los paquetes entrantes. Como PC1b tiene como siguiente salto en la ruta por defecto la dirección IP de eth1 del router los paquetes de respuesta llegarán correctamente a él.

Puede ver las reglas añadidas con:

```
# iptables --table nat --numeric --list
```

Hagan de nuevo el ping desde PC1a a PC1b para ver cómo ahora los paquetes que llegan a PC1b vienen de la dirección IP a la que se ha hecho la conversión.

Emplee ahora el comando nc para crear una conexión TCP de PC1a a PC1b. Puede ver el mapeo concreto que se está haciendo con el siguiente comando:

```
# cat /proc/net/ip_conntrack
```

Puede usar nc también para enviar datagramas UDP.

Puede borrar las entradas en la tabla nat con el comando:

```
# iptables --flush --table nat
```

4- Conversión dinámica

Ahora vamos a crear un conjunto (*pool*) de direcciones globales que el router empleará para convertir las direcciones internas a ellas. Esto se puede hacer con opciones muy similares al apartado anterior. Consulte las secciones del manual sobre estas opciones.

- Configure que todas las direcciones de la subred 192.168.1.0/24 se mapeen a las

direcciones entre 10.0.1.1 y 10.0.1.3. Eso son 3 direcciones disponibles.

- Emplee nc para enviar datagramas UDP desde PCs de la subred privada a máquinas de la subred pública. Usando tcpdump compruebe a qué dirección y puerto públicos se hacen los mapeos.
- Compruebe cómo es el mapeo si dos hosts de la subred privada envían datagramas UDP al mismo host de la subred pública empleando el mismo puerto cliente

Las entradas caducan tras cierto periodo de inactividad. Puede ver el estado de ese contador de tiempo en /proc/net/ip_conntrack. También puede modificar su valor máximo empleando sysctl.

- Mande un datagrama UDP empleando nc de un host interno a uno externo y compruebe cómo se ha creado el mapeo en el NAT
- Compruebe cómo va agotando su tiempo de vida si no envía más datagramas, y cómo desaparece por inactividad
- Pruebe a cambiar el valor máximo de dicho timer empleando sysctl[4]

Punto de control: Muestre al profesor de prácticas este último escenario y que ha entendido el funcionamiento básico de un NAT.

5- Sobrecarga de la dirección exterior del router

En este apartado vamos a configurar que el router convierta todas las direcciones internas en *su* dirección externa. Emplee para ello la misma topología de la figura 1.

- Configure una regla que cree mapeos SNAT para todas las direcciones de la subred privada donde la dirección origen se cambie por la del interfaz eth1 del router/NAT.
- Empleando varios hosts internos que se comunican con uno o varios hosts externos vean los mapeos que se crean, así como el tráfico que llega a esos hosts

6- Interconexión de redes públicas y privadas

En este parte recrearemos un escenario donde una empresa tiene redes con direccionamiento privado y direccionamiento público (figura 2).

- El tráfico de (desde o hacia) los hosts con direccionamiento público se debe enrutar con normalidad, tanto con las redes públicas (la Internet) como con las redes privadas de la empresa
- El tráfico de la red privada de la empresa hacia el exterior de la misma debe sufrir un proceso de NAT, saliendo todo el tráfico con la dirección IP del interfaz externo del NAT
- Emplee 192.168.1.0/24 para la subred privada y partes de 10.0.0.0/8 para las subredes públicas.
- (Opcional) Se debe descartar cualquier tráfico que llegue del exterior hacia el NAT y que vaya hacia la subred privada (no se debe enrutar, solo debe pasar por el procedimiento de NAT si ha habido antes el establecimiento de un mapeo hacia el exterior).

[4] `sysctl net.netfilter.nf_conntrack_udp_timeout=VALOR`

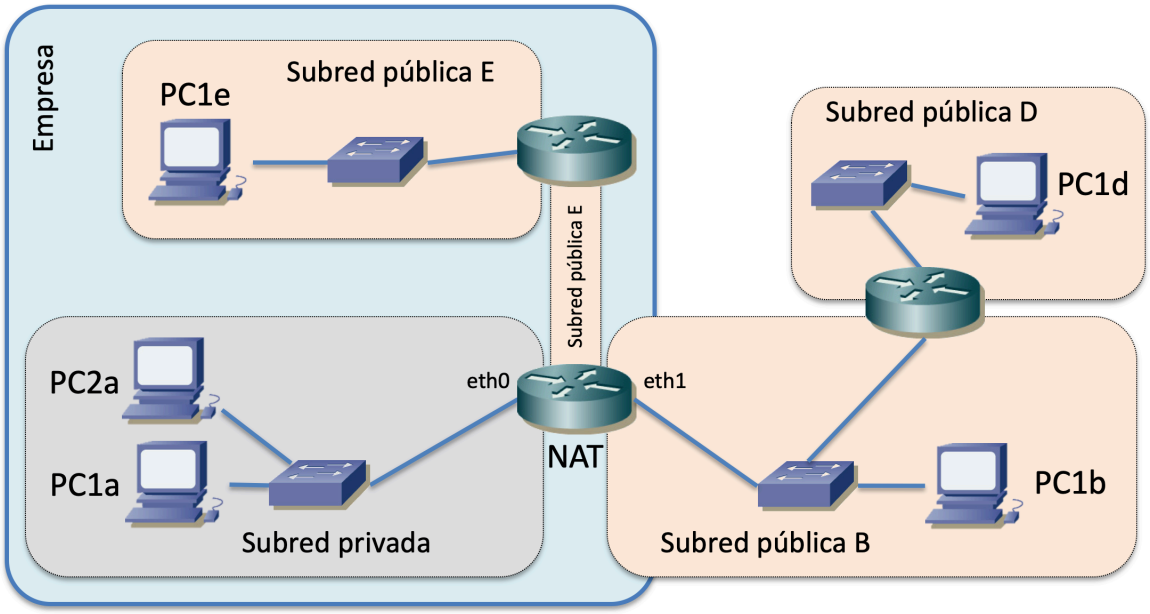

Figura 2.- Escenario combinando direccionamiento público y privado

Punto de control: Muestre al profesor de prácticas los experimentos que ha llevado a cabo y las conclusiones que se extraen de ellos respecto al comportamiento de este NAT.

7- Port forwarding

En el escenario del apartado anterior añada una configuración de *port forwarding*. Supondremos que PC1a tiene corriendo un servidor TCP en el puerto 8080 (puede emplear nc para ello). Configure que el tráfico que venga del exterior de la empresa dirigido al puerto TCP 80 de la dirección IP pública del router/NAT se redirija a ese servidor interno.