1.3 IPTABLES

→ Hecho por: Izan Navarro Lujan Practica 1.3

INDICE

EJERCICIO 1: Comprueba si iptables está instalado en tu sistema operativo.

Adjunta capturas de pantalla y justifica la respuesta. De no ser así instálalo y comprueba que el servicio está activo.

El comando utilizado para mostrar la versión del iptables en la VM:

```
vboxuser@izan:~$ sudo iptables -V
[sudo] contraseña para vboxuser:
iptables v1.8.7 (nf_tables)
vboxuser@izan:~$
```

EJERCICIO 2: Estudia mediante el comando man, el manual de iptables. Comenta, los argumentos más interesantes que consideres de esta herramienta y por qué. Como venimos haciendo y va a ser la dinámica de las prácticas, se va a estudiar y analizar el manual de las herramientas, con el fin de conocer todas las funciones que ofrece, esto va a permitir al alumno, desarrollar la habilidad de saber cómo y dónde buscar, en el caso de posible duda.

Ejecuto el comando “man iptables” el cual me mostrará la librería de opciones que puedo hacer con iptables y que función hace cada letra.

```
vboxuser@izan:~$ man iptables
```

```
-A, --append chain rule-specification
       Append one or more rules to the end of the selected chain.  When
       the source and/or destination names resolve to more than one ad-
       dress, a rule will be added for each possible  address  combina-
       tion.

-C, --check chain rule-specification
       Check  whether  a  rule matching the specification does exist in
       the selected chain. This command uses the same logic  as  -D  to
       find  a matching entry, but does not alter the existing iptables
       configuration and uses its exit  code  to  indicate  success  or
       failure.

-D, --delete chain rule-specification
-D, --delete chain rulenum
       Delete one or more rules from the selected chain.  There are two
       versions of this command: the rule can be specified as a  number
       in  the  chain  (starting  at 1 for the first rule) or a rule to
       match.

-I, --insert chain [rulenum] rule-specification
       Insert one or more rules in the selected chain as the given rule
       number.   So, if the rule number is 1, the rule or rules are in-
       serted at the head of the chain.  This is also the default if no
       rule number is specified.

-R, --replace chain rulenum rule-specification
       Replace a rule in the selected chain.  If the source and/or des-
       tination names resolve to multiple addresses, the  command  will
       fail.  Rules are numbered starting at 1.
```

- `-A` → Añadir una regla al final de una cadena.
- `-I` → Insertar una regla en una posición específica (por ejemplo al principio).
- `-D` → Eliminar una regla.
- `-L` → Listar reglas activas.
- `-F` → Limpiar (flush) todas las reglas.
- `-P` → Cambiar la política por defecto de una cadena.
- `-s` → Especificar dirección origen.
- `-d` → Especificar dirección destino.
- `-p` → Protocolo (tcp, udp, icmp).
- `--dport` → Puerto destino.
- `-j` → Acción (ACCEPT, DROP, REJECT, MASQUERADE…).

EJERCICIO 3: Comprueba el estado de las tablas filter y NAT, para comprobar si hay alguna configuración previa. Si hubiera alguna configuración, ¿Cómo borrarías y guardarías la configuración de iptables?

Para mostrar filter y NAT uso –t filter/nat -L –n -v:

```
vboxuser@izan:~$ sudo iptables -t filter -L -n -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
vboxuser@izan:~$ sudo iptables -t nat -L -n -v
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
vboxuser@izan:~$
```

Para guardar y borrar la configuración de Iptables (-F/-netfilter-persisntent save):

```
vboxuser@izan:~$ sudo iptables -F
vboxuser@izan:~$ sudo iptables -t nat -F
```

```
vboxuser@izan:~$ sudo netfilter-persistent save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables save
```

EJERCICIO 4: Establece la política por defecto (INPUT, OUTPUT, FORWARD), en la tabla filter. ¿En qué consiste la política por defecto? Como buena práctica en ciberseguridad, primero se bloquea todo el tráfico y luego se va permitiendo lo necesario. Con esto solo tenemos permitido el tráfico estrictamente necesario, reduciendo los vectores de entradas de amenazas.

Con estos comandos bloqueo todo el tráfico de red reduciendo la entrada de amenazas.

```
vboxuser@izan:~$ sudo iptables -P INPUT DROP
vboxuser@izan:~$ sudo iptables -P OUTPUT DROP
vboxuser@izan:~$ sudo iptables -P FORWARD DROP
```

EJERCICIO 5: Una vez realizado el punto 4, comprueba que no puedes acceder, mediante SSH a la máquina virtual.

Intento hacer ssh a mi VM (iniciándola previamente) y de primera mano me da rechazado.

```
vboxuser@izan:~$ sudo systemctl start ssh
```

```
PS C:\Users\ESP> ssh xboxuser@192.168.56.101
ssh: connect to host 192.168.56.101 port 22: Connection timed out
```

EJERCICIO 6: Permite la conexión SSH desde tu máquina anfitrión hacia tu máquina virtual, donde tienes configurado el iptables, y que regla emplearías para ello.

Agrego el comando que acepta los INPUT por el puerto 22 y al iniciar el ssh en mi VM, cuando lo llamo desde mi máquina anfitriona, me conecta sin problemas. :)

```
vboxuser@izan:~$ sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

```
vboxuser@izan:~/Escritorio$ sudo iptables -A OUTPUT -m state --state ESTABLISHED
,RELATED -j ACCEPT
vboxuser@izan:~/Escritorio$ sudo systemctl start ssh
vboxuser@izan:~/Escritorio$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: e>
     Active: active (running) since Tue 2025-10-21 18:22:38 CEST; 2min 37s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 759 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 776 (sshd)
      Tasks: 1 (limit: 4606)
     Memory: 3.7M
        CPU: 58ms
     CGroup: /system.slice/ssh.service
             └─776 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

```
C:\Users\ESP>ssh vboxuser@192.168.56.101
The authenticity of host '192.168.56.101 (192.168.56.101)' can't be established.
ED25519 key fingerprint is SHA256:dFizHi5phYSQaDOo9OWFNgC4PjpC8vo8tkuE1t9bpC4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.101' (ED25519) to the list of known hosts.
vboxuser@192.168.56.101's password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-85-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

El mantenimiento de seguridad expandido para Applications está desactivado

Se pueden aplicar 292 actualizaciones de forma inmediata.
216 de estas son actualizaciones de seguridad estándares.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

vboxuser@izan:~$
```

EJERCICIO 7: Lanza un ping, desde tu máquina anfitrión a la máquina virtual, donde tienes configurado el iptables. ¿Funciona? Permite la conexión ICMP desde tu máquina anfitrión hacia tu máquina virtual y que regla emplearías para ello.

Permitimos la conexión ICMP desde mi maquina anfitriona hacia mi VM, poniendo el comando de abajo dentro de la VM y posteriormente, haciendo ping a la ip asociada a mi máquina.

```
sudo iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
```

```
C:\Users\ESP>ping 192.168.56.101

Haciendo ping a 192.168.56.101 con 32 bytes de datos:
Respuesta desde 192.168.56.101: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.56.101: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.56.101: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.56.101: bytes=32 tiempo<1m TTL=64

Estadísticas de ping para 192.168.56.101:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 1ms, Media = 0ms

C:\Users\ESP>
```

EJERCICIO 8: Que reglas emplearías, para permitir el tráfico FTP, HTTP, HTTPS, DNS, NTP desde toda la LAN 192.168.0.0/24. Introdúcelas en iptables y muestra una captura de la tabla filter.

Uso el comando que tenemos abajo para poder habilitar por tcp los puertos asignados a los diferentes tipos de tráfico que quiero permitir (FTP, HTTP, HTTPS, DNS, NTP = por orden en las capturas de pantalla).

```
vboxuser@izan:~/Escritorio$ sudo iptables -A INPUT -p tcp --dport 21 -s 192.168.
0.0/24 -j ACCEPT
vboxuser@izan:~/Escritorio$ sudo iptables -A INPUT -p tcp --dport 80 -s 192.168.
0.0/24 -j ACCEPT
```

```
vboxuser@izan:~/Escritorio$ sudo iptables -A INPUT -p tcp --dport 443 -s 192.168
.0.0/24 -j ACCEPT
vboxuser@izan:~/Escritorio$ sudo iptables -A INPUT -p tcp --dport 53 -s 192.168.
0.0/24 -j ACCEPT
vboxuser@izan:~/Escritorio$ sudo iptables -A INPUT -p udp --dport 53 -s 192.168.
0.0/24 -j ACCEPT
```

```
vboxuser@izan:~/Escritorio$ sudo iptables -L -n -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

   96  7569 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
        tcp dpt:22
    0     0 ACCEPT     icmp --  *      *       192.168.0.0          0.0.0.0/0

    0     0 ACCEPT     tcp  --  *      *       192.168.0.0/24       0.0.0.0/0
        tcp dpt:21
    0     0 ACCEPT     tcp  --  *      *       192.168.0.0/24       0.0.0.0/0
        tcp dpt:80
    0     0 ACCEPT     tcp  --  *      *       192.168.0.0/24       0.0.0.0/0
        tcp dpt:443
    0     0 ACCEPT     tcp  --  *      *       192.168.0.0/24       0.0.0.0/0
        tcp dpt:53
    0     0 ACCEPT     udp  --  *      *       192.168.0.0/24       0.0.0.0/0
        udp dpt:53
```

EJERCICIO 9: Ya que estamos usando como cortafuegos la máquina virtual, vamos a configurarla como router, para que todas las máquinas de la LAN salgan a internet a través de esta y solo mediante las reglas, establecidas.

• Edita el fichero /etc/sysctl.conf, para habilitar el enrutamiento en la máquina.

Entramos dentro del archivo con “nano” y descomentamos la línea que hay puesta abajo.

```
vboxuser@izan:~/Escritorio$ sudo nano /etc/sysctl.conf
```

```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

• Edita el fichero /proc/sys/net/ipv4/icmp_echo_ignore_all, para habilitar la respuesta del protocolo ICMP

```
vboxuser@izan:~/Escritorio$ echo 0 | sudo tee /proc/sys/net/ipv4/icmp_echo_ignore_all
0
```

--> 0 significa que responderá a los pings.

• Crea una regla POSTROUTING en la tabla nat, utilizando el parámetro -j MASQUERADE y explica cuál es la función de este último parámetro.

Aquí el comando que creará una regla POSTROUTING:

```
vboxuser@izan:~/Escritorio$ sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

-j MASQUERADE?

Permite que todas las máquinas de la LAN salgan a Internet usando la IP pública del firewall, reescribiendo la IP de origen de los paquetes. Es una forma dinámica de NAT.

• Crea una regla FORWARD para aceptar que el tráfico pueda pasar a través de nuestra máquina iptables.

```
vboxuser@izan:~/Escritorio$ sudo iptables -A FORWARD -s 192.168.0.0/24 -o eth0 -j ACCEPT
vboxuser@izan:~/Escritorio$ sudo iptables -A FORWARD -d 192.168.0.0/24 -m state --state ESTABLISHED,RELATED
-j ACCEPT
```

--> Tráfico **de la LAN a Internet**.

--> Tráfico de **respuesta desde Internet hacia LAN** si es parte de una conexión establecida.

EJERCICIO 10: Crea una regla para aislar tu equipo anfitrión y que no pueda acceder a internet, para ello crea una regla en la tabla FORWARD.

• Utiliza iptables -A Forward y adjunta una captura de la tabla FILTER ¿Tenemos conectividad? ¿Por qué no ha funcionado?

```
vboxuser@izan:~/Escritorio$ sudo iptables -A FORWARD -s 10.250.215.72
```

```
C:\Users\ESP>ping 8.8.8.8

Haciendo ping a 8.8.8.8 con 32 bytes de datos:
Tiempo de espera agotado para esta solicitud.

Estadísticas de ping para 8.8.8.8:
    Paquetes: enviados = 1, recibidos = 0, perdidos = 1
    (100% perdidos),
Control-C
^C
C:\Users\ESP>
```

No, no tenemos conectividad ya que la VM Ubuntu hace de router/cortafuegos, y todas las demás máquinas (incluido el host) pasan por ella para salir a Internet.

EJERCICIO 11: Averigua el numero de la regla que has insertado en el Ejercicio 10 y elimínala. A continuación, en lugar de añadir inserta en el número 1 la regla anterior con el parámetro -I

```
vboxuser@izan:~/Escritorio$ sudo iptables -L FORWARD --line-numbers
Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  192.168.0.0/24       anywhere
2    ACCEPT     all  --  anywhere             192.168.0.0/24       state RELATED,ESTABLISHED
3    DROP       all  --  izan                 anywhere
4                all  --  10.250.215.72        anywhere
vboxuser@izan:~/Escritorio$ sudo iptables -D FORWARD 3
vboxuser@izan:~/Escritorio$ sudo iptables -I FORWARD 1 -s 192.168.56.101 -j DROP
vboxuser@izan:~/Escritorio$ sudo iptables -L FORWARD --line-numbers
Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
1    DROP       all  --  izan                 anywhere
2    ACCEPT     all  --  192.168.0.0/24       anywhere
3    ACCEPT     all  --  anywhere             192.168.0.0/24       state RELATED,ESTABLISHED
4                all  --  10.250.215.72        anywhere
vboxuser@izan:~/Escritorio$
```

El comando “--line-numbers" muestra las diferentes chain FORWARD que se han creado y con “-I” modifico la “source” elegida y le cambio su posición.