Responsable du cours (ST2EH) : Yousfi Souheib

TP Firewall iptables

Objectif du TP :

Ce TP a pour objectif de vous familiariser avec l'outil 'iptables' pour la gestion des règles de filtrage de paquets sur un système Linux. Vous apprendrez à configurer et à gérer les règles de pare-feu pour contrôler le trafic réseau.

1 Partie 1 :

'iptables' est un outil en ligne de commande permettant de configurer le pare-feu Netfilter dans le noyau Linux. Il est utilisé pour gérer les tables de règles qui filtrent et redirigent le trafic réseau.

1. Pour lister toutes les règles de iptables :

```
┌──(kali㉿kali)-[~/Downloads]
└─$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

   — Pour la chaîne INPUT c'est les adresses du réseau local et de la NAT (si elle existe).
   — Pour la chaîne OUTPUT c'est les paquets envoyés à partir de la machine elle-même se rendront à cette chaîne, comme par exemple le ping sur le DNS public de google 8.8.8.8. Le Forward, généralement en cas de passerelle, objectif de la deuxième partie.

2. Afficher les noms et les adresses des interfaces réseaux :

```
┌──(kali㉿kali)-[~/Downloads]
└─$ ip route show table local
local 127.0.0.0/8 dev lo proto kernel scope host src 127.0.0.1
local 127.0.0.1 dev lo proto kernel scope host src 127.0.0.1
broadcast 127.255.255.255 dev lo proto kernel scope link src 127.0.0.1
local 192.168.64.3 dev eth0 proto kernel scope host src 192.168.64.3
broadcast 192.168.64.255 dev eth0 proto kernel scope link src 192.168.64.3
```

3. Tester le ping sur l'adresse locale avant d'initier une règle.

```
┌──(kali㉿kali)-[~/Downloads]
└─$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.208 ms
```

4. Ajouter, à l'aide de l'option append, une règle pour bloquer le protocole ICMP en utilisant DROP, spécifiquement pour une destination locale.

```
┌──(kali㉿kali)-[~/Downloads]
└─$ sudo iptables -A INPUT -d 127.0.0.1 -p icmp -j DROP

┌──(kali㉿kali)-[~/Downloads]
└─$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
^C
── 127.0.0.1 ping statistics ──
3 packets transmitted, 0 received, 100% packet loss, time 2037ms
```

5. Vérifier le ping sur l'adresse locale ainsi que la table des règles.

```
┌──(kali㉿kali)-[~/Downloads]
└─$ sudo iptables -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    DROP       icmp --  anywhere             localhost

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
```

6. Bloquer le ping sur l'adresse 8.8.8.8, puis tester le.

```
┌──(kali㉿kali)-[~/Downloads]
└─$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=114 time=7.51 ms
^C
── 8.8.8.8 ping statistics ──
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 7.508/7.508/7.508/0.000 ms

┌──(kali㉿kali)-[~/Downloads]
└─$ sudo iptables -A OUTPUT -d 8.8.8.8 -p icmp -j DROP

┌──(kali㉿kali)-[~/Downloads]
└─$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
^C
── 8.8.8.8 ping statistics ──
3 packets transmitted, 0 received, 100% packet loss, time 2035ms
```

7. Si vous voulez bloquer un site donné, prenez comme exemple facebook.

```
┌──(kali㉿kali)-[~/Downloads]
└─$ ping facebook.com
PING facebook.com (185.60.219.35) 56(84) bytes of dat
a.
64 bytes from edge-star-mini-shv-01-cdg4.facebook.com
 (185.60.219.35): icmp_seq=1 ttl=57 time=6.71 ms
```

```
┌──(kali㉿kali)-[~/Downloads]
└─$ sudo iptables -A INPUT -s facebook.com -j DROP
```

```
┌──(kali㉿kali)-[~/Downloads]
└─$ sudo iptables -L
[sudo] password for kali:
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

DROP       icmp --  anywhere             localhost

DROP       all  --  edge-star-mini-shv-01-cdg4.facebo
ok.com  anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination


Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

DROP       icmp --  anywhere             dns.google


┌──(kali㉿kali)-[~/Downloads]
└─$ ping facebook.com
PING facebook.com (185.60.219.35) 56(84) bytes of dat
a.
^C
── facebook.com ping statistics ──
32 packets transmitted, 0 received, 100% packet loss, time 31
728ms
```

8. Affichant les règles numérotées de notre chaîne

```
┌──(kali㉿kali)-[~/Downloads]
└─$ sudo iptables -L  -v --line-numbers
Chain INPUT (policy ACCEPT 314 packets, 59576 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1        3   252 DROP       icmp --  any    any     anywhere             localhost
2       32  2688 DROP       all  --  any    any     edge-star-mini-shv-01-cdg4.facebook.com  anywhere

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

Chain OUTPUT (policy ACCEPT 239 packets, 29036 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1        3   252 DROP       icmp --  any    any     anywhere             dns.google
```

9. Supprimant une règle de notre chaîne.

```
┌──(kali㉿kali)-[~/Downloads]
└─$ sudo iptables -D INPUT 2

┌──(kali㉿kali)-[~/Downloads]
└─$ sudo iptables -L  -v --line-numbers
Chain INPUT (policy ACCEPT 480 packets, 83035 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1        3   252 DROP       icmp --  any    any     anywhere             localhost

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

Chain OUTPUT (policy ACCEPT 402 packets, 51483 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1        3   252 DROP       icmp --  any    any     anywhere             dns.google

┌──(kali㉿kali)-[~/Downloads]
└─$ ping facebook.com
PING facebook.com (157.240.202.35) 56(84) bytes of data.
64 bytes from edge-star-mini-shv-02-cdg4.facebook.com (157.240.202.35): icmp_seq=1 ttl=57 time=6.72 ms
```

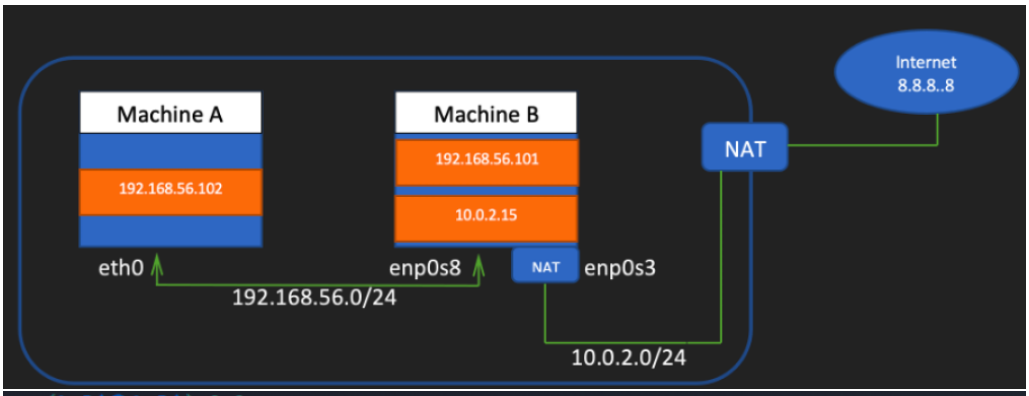
10. Si on veut ajouter une -policy de blocage de tous les sites

```
┌──(kali㉿kali)-[~/Downloads]
└─$ sudo iptables -P INPUT DROP

┌──(kali㉿kali)-[~/Downloads]
└─$ sudo iptables -L  -v --line-numbers
Chain INPUT (policy DROP 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1        3   252 DROP       icmp --  any    any     anywhere             localhost

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

Chain OUTPUT (policy ACCEPT 542 packets, 69560 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1        3   252 DROP       icmp --  any    any     anywhere             8.8.8.8

┌──(kali㉿kali)-[~/Downloads]
└─$ ping facebook.com
^C

┌──(kali㉿kali)-[~/Downloads]
└─$ ping youtube.com
^C
```

La politique par défaut définie avec -P ne s'affiche pas comme une règle individuelle lorsque vous utilisez la commande iptables -L. Elle est visible en tant que paramètre global en haut de la chaîne correspondante.

2 Partie 2 :

Dans cette section, nous nous concentrons sur la chaîne FORWARD. Sur une machine Kali, nous désactivons la NAT et activons le réseau local. Ensuite, sur une machine Ubuntu, nous activons à la fois le réseau local et la NAT. L'objectif est que Kali accède à Internet via l'interface NAT d'Ubuntu, qui jouera le rôle de passerelle pour Kali vers la NAT de la machine physique.




```
┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group def
ault qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
 group default qlen 1000
    link/ether 08:00:27:1b:bf:4d brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.9/24 brd 192.168.56.255 scope global dynamic noprefixrout
e eth0
       valid_lft 508sec preferred_lft 508sec
    inet6 fe80::a00:27ff:fe1b:bf4d/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

```
souhelb@souhelb-VirtualBox:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:d5:6e:af brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 78937sec preferred_lft 78937sec
    inet6 fe80::6da7:785e:df25:231b/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:54:98:0c brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.5/24 brd 192.168.56.255 scope global dynamic noprefixroute enp0s8
       valid_lft 338sec preferred_lft 338sec
    inet6 fe80::a13b:d59:fe76:3c6f/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

— Tester sur kali, que vous ne pouvez pas accéder à internet.

— Sur Kali, ajouter la machine Ubuntu comme gateway.

```
┌──(kali㉿kali)-[~]
└─$ sudo route add default gw 192.168.56.5
[sudo] password for kali:

┌──(kali㉿kali)-[~]
└─$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.168.56.5    0.0.0.0         UG    0      0        0 eth0
192.168.56.0    0.0.0.0         255.255.255.0   U     100    0        0 eth0
```

— Même si le message change du ping 8.8.8.8, mais on n'a toujours pas accès à internet..

— Sur la machine Ubuntu, activer le forward.

```
souheib@souheib-VirtualBox:~$ cat /etc/sysctl.conf | grep net.ipv4
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1
#net.ipv4.tcp_syncookies=1
net.ipv4.ip_forward=1
#net.ipv4.conf.all.accept_redirects = 0
# net.ipv4.conf.all.secure_redirects = 1
#net.ipv4.conf.all.send_redirects = 0
#net.ipv4.conf.all.accept_source_route = 0
#net.ipv4.conf.all.log_martians = 1
souheib@souheib-VirtualBox:~$ sudo sysctl -p
net.ipv4.ip_forward = 1
```

— Ajoutez ensuite cette règle qui permet aux machines du sous-réseau local 192.168.56.0/24 d'accéder à Internet via l'interface ens0s3. Le masquage substitue l'adresse IP source des paquets par l'IP publique de l'interface, permettant ainsi à toutes les machines du réseau local de partager cette adresse pour leurs communications externes.

```
souheib@souheib-VirtualBox:~$ sudo iptables -t nat -A POSTROUTING -s 192.168.56.0/24 -o enp0s3 -j MASQUERADE
```

— Tester sur Kali, l'accés à internet, puis ajouter une règle sur la machine Ubuntu pour qu'elle accepte le traffic si la source est la machine de Kali.

```
souheib@souheib-VirtualBox:~$ sudo iptables -A FORWARD -s 192.168.56.9 -o enp0s3 -j ACCEPT
souheib@souheib-VirtualBox:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  192.168.56.9         anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

3 Annexe : Options Principales d'iptables

— `-A`, `-append` : Ajouter une règle à la fin d'une chaîne.

— `-I`, `-insert` : Insérer une règle à une position spécifique.

— `-D`, `-delete` : Supprimer une règle d'une chaîne.

— `-P`, `-policy` : Définir la politique par défaut d'une chaîne.

— `-L`, `-list` : Lister toutes les règles actuelles.

— `-F`, `-flush` : Supprimer toutes les règles d'une chaîne.

— `-t`, `-table` : Spécifier la table à utiliser (`filter`, `nat`, `mangle`, `raw`).

— `-p`, `-protocol` : Spécifier le protocole (`tcp`, `udp`, etc.).

— `-dport`, `-sport` : Spécifier les ports source ou destination.

— `-j`, `-jump` : Spécifier l'action à appliquer (`ACCEPT`, `DROP`, `REJECT`, etc.).

♣ S.Y. ♣
Bon travail