TP Filtrage de paquets

Guillaume Sanchez

Politique de filtrage par défaut

1. Lancer la commande iptables -L puis observer la sortie d'écran.

```
root@U2-22:~/regles_filtrages# 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
```

2. Quelle est la politique adoptée par chacune des chaînes prédéfinies ?

Tout peux rentrer et tou peux sortir

Exploitation des règles de filtrage

1. Créer sur le poste Serveur un répertoire nommé « firewall ».

2. Copier dans ce répertoire le fichier regles-filtrage.tar.gz puis décompacter le avec la commande « tar xvzf regles-filtrage.tar.gz ». Vous devez obtenir les fichiers scripts suivants : deny-all, accept-all, block-ip et unblock-ip.

3. Rendre ces fichiers exécutables (droits d'accès 700).

4. Exécuter le script deny-all en tapant « ./deny-all ». Essayer de faire un ping localhost et 192.168.10.254. Que constatez-vous ? Décrivez les règles utilisées par ce script.

Les règles étaient déjà décommenté, mais si elle les sont, le ping fonctionne car aucune règles n'est appliqué.

5. Editer le script deny-all puis décommenter les lignes suivantes :

```
#INTERFACE_LOOPBACK=lo
#iptables -A OUTPUT -o $INTERFACE_LOOPBACK -j ACCEPT
#iptables -A INPUT -i $INTERFACE_LOOPBACK -j ACCEPT
```

6. Exécuter le script obtenu puis ressayer à nouveau les deux commandes Ping précédentes. Que constatez-vous cette fois ci ? Expliquez.

Le ping ne fonctionne plus, en effet les règles décommenté empêche les entrées et les sorties.

7. Afficher les informations concernant les paquets examinés par le pare-feu en utilisant la commande « iptables –L -v». Commenter la sortie d'écran.

```
root@U2-22:~# iptables -L -v
Chain INPUT (policy DROP 15 packets, 2404 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  lo     any     anywhere             anywhere

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

Chain OUTPUT (policy DROP 8 packets, 880 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  any    lo      anywhere             anywhere
```

8. Exécuter le script accept-all en tapant « ./accept-all » puis essayer de faire un ping localhost et 192.168.10.254. Que constatez-vous ? Décrivez les règles utilisées par ce script.

```
root@U2-22-server:~/regles_filtrages# ./accept-all
root@U2-22-server:~/regles_filtrages# iptables -L -v
Chain INPUT (policy ACCEPT 31 packets, 2152 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 16 packets, 1488 bytes)
 pkts bytes target     prot opt in     out     source               destination
root@U2-22-server:~/regles_filtrages#

root@U2-22-server:~/regles_filtrages# ping localhost
PING localhost(localhost (::1)) 56 data bytes
64 bytes from localhost (::1): icmp_seq=1 ttl=64 time=0.013 ms
64 bytes from localhost (::1): icmp_seq=2 ttl=64 time=0.024 ms
64 bytes from localhost (::1): icmp_seq=3 ttl=64 time=0.044 ms
64 bytes from localhost (::1): icmp_seq=4 ttl=64 time=0.024 ms
--- localhost ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3063ms
rtt min/avg/max/mdev = 0.013/0.026/0.044/0.011 ms
```

```
root@U2-22-server:~/regles_filtrages# ping 192.168.1.236
PING 192.168.1.236 (192.168.1.236) 56(84) bytes of data.
64 bytes from 192.168.1.236: icmp_seq=1 ttl=64 time=0.106 ms
64 bytes from 192.168.1.236: icmp_seq=2 ttl=64 time=0.051 ms
64 bytes from 192.168.1.236: icmp_seq=3 ttl=64 time=0.084 ms
64 bytes from 192.168.1.236: icmp_seq=4 ttl=64 time=0.064 ms
--- 192.168.1.236 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3066ms
rtt min/avg/max/mdev = 0.051/0.076/0.106/0.020 ms
```

il n'y a plus aucune règle de deny donc tout est de nouveau ouvert.

9. Exécuter le script block-ip en tapant « ./block-ip 192.168.10.254 ». Essayer maintenant de faire un ping localhost et 192.168.10.254. Que constatez-vous ? Décrivez les règles utilisées par ce script.

```
root@U2-22-server:~/regles_filtrages# ./block-ip 192.168.1.236

root@U2-22-server:~/regles_filtrages# ping localhost
PING localhost(localhost (::1)) 56 data bytes
64 bytes from localhost (::1): icmp_seq=1 ttl=64 time=0.432 ms
64 bytes from localhost (::1): icmp_seq=2 ttl=64 time=0.022 ms
64 bytes from localhost (::1): icmp_seq=3 ttl=64 time=0.048 ms
64 bytes from localhost (::1): icmp_seq=4 ttl=64 time=0.051 ms

--- localhost ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3086ms
rtt min/avg/max/mdev = 0.022/0.138/0.432/0.169 ms

root@U2-22-server:~/regles_filtrages# ping 192.168.1.236
PING 192.168.1.236 (192.168.1.236) 56(84) bytes of data.

--- 192.168.1.236 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5144ms
```

Localhost n'est pas bloqué, mais la machine client ne reçoit pas le ping car il est bloqué.

```
root@U2-22-server:~/regles_filtrages# iptables -L --line-number
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    DROP       all  --  192.168.1.236        anywhere

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

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
1    DROP       all  --  anywhere             192.168.1.236
```

On voit que ces règles empêche les entrées et les sorties de 192.168.1.236 (machine client chez moi).

10. Exécuter le script unblock-ip en tapant « ./unblock-ip 192.168.10.254 » puis essayer de faire un ping localhost et 192.168.10.254. Que constatez-vous ? Décrivez les règles utilisées par ce script.

```
root@U2-22-server:~/regles_filtrages# ./unblock-ip 192.168.1.236
IP 192.168.1.236 débloquée.

root@U2-22-server:~/regles_filtrages# ping 192.168.1.236
PING 192.168.1.236 (192.168.1.236) 56(84) bytes of data.
64 bytes from 192.168.1.236: icmp_seq=1 ttl=64 time=0.028 ms
64 bytes from 192.168.1.236: icmp_seq=2 ttl=64 time=0.052 ms
64 bytes from 192.168.1.236: icmp_seq=3 ttl=64 time=0.077 ms
64 bytes from 192.168.1.236: icmp_seq=4 ttl=64 time=0.062 ms

--- 192.168.1.236 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3080ms
rtt min/avg/max/mdev = 0.028/0.054/0.077/0.017 ms
```

On remarque que l'on peut de nouveau pinguer 192.168.1.236 (machine client chez moi)

```
root@U2-22-server:~/regles_filtrages# iptables -L --line-number
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination

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

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

Les règles de blocage sur l'ip 192.168.1.236 en entrée et en sortie on disparu, il n'y a plus aucunes règles.

Création de nouvelles règles de filtrage

1. Interdire un paquet s'il ne provient pas de localhost

Autoriser paquet qui vient de localhost : `iptables -A INPUT -i lo -j ACCEPT`

Bloquer tout autre trafic entrant `iptables -A INPUT ! -i lo -j DROP`

Du coup machine serveur ping localhost :

```
root@U2-22-server:~/regles_filtrages# ping localhost
PING localhost(localhost (::1)) 56 data bytes
64 bytes from localhost (::1): icmp_seq=1 ttl=64 time=0.032 ms
64 bytes from localhost (::1): icmp_seq=2 ttl=64 time=0.023 ms
64 bytes from localhost (::1): icmp_seq=3 ttl=64 time=0.023 ms
64 bytes from localhost (::1): icmp_seq=4 ttl=64 time=0.023 ms
--- localhost ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3094ms
rtt min/avg/max/mdev = 0.023/0.025/0.032/0.004 ms
```

Machine client ne ping pas :

```
root@U2-22-client:~# ping 192.168.1.235
PING 192.168.1.235 (192.168.1.235) 56(84) bytes of data.

--- 192.168.1.235 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2075ms
```

2. Interdire le protocole ICMP à destination de localhost

```
iptables -A INPUT -p icmp -d 127.0.0.1 -j DROP

root@U2-22-server:~/regles_filtrages# iptables -A INPUT -p icmp -d 127.0.0.1 -j DROP

root@U2-22-server:~/regles_filtrages# ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3057ms
```

3. Interdire tout paquet à destination du port Telnet

```
iptables -A INPUT -p tcp --dport 23 -j DROP
```

telnet bloqué coté client :

```
root@U2-22-client:~# telnet 192.168.1.235 23
Trying 192.168.1.235...
telnet: Unable to connect to remote host: Connection refused
```

4. Interdire tout paquet sortant par eth0 dont le numéro du port destination est inférieur à 1024

```
iptables -A OUTPUT -o eth0 -p tcp --dport 0:1023 -j DROP iptables -A OUTPUT -o eth0 -p udp --dport 0:1023 -j DROP
```

Si je curl par exemple guillaume-sanchez.fr

```
root@U2-22-server:~/regles_filtrages# curl https://guillaume-sanchez.fr --interface eth0 -v
* Could not resolve host: guillaume-sanchez.fr
* Closing connection 0
curl: (6) Could not resolve host: guillaume-sanchez.fr
```

5. Interdire toute tentative d'initialisation de connexion TCP provenant de eth0

```
iptables -A INPUT -i eth0 -p tcp --syn -j DROP
```

Après application de la règle, impossible de se connexter en SSH par exemple depuis la machine client :

```
root@U2-22-client:~# ssh root@192.168.1.235
ssh: connect to host 192.168.1.235 port 22: Connection timed out
```

6. Interdire toutes réponses à un Ping

```
iptables -A OUTPUT -p icmp --icmp-type echo-reply -j DROP
```

Après application de la règle, la machine client ne ressois pas de réponse au ping :

```
root@U2-22-client:~# ping 192.168.1.235
PING 192.168.1.235 (192.168.1.235) 56(84) bytes of data.
--- 192.168.1.235 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5149ms
```

7. Interdire tout paquet entrant par eth0 dont l'adresse mac n'est pas celle du poste de travail (voir le schéma à la page 1) Remarque : Attention, vous ne pouvez utiliser le filtrage par adresse mac que sur la table INPUT

```
iptables -A INPUT -i eth0 -m mac --mac-source bc:24:11:b7:f8:21 -j ACCEPT iptables -A INPUT -i eth0 -j DROP
```

Après application des règles, la machine client qui à pour mac bc:24:11:b7:f8:21 peut faire un ping :

```
root@U2-22-client:~# ping 192.168.1.235
PING 192.168.1.235 (192.168.1.235) 56(84) bytes of data.
64 bytes from 192.168.1.235: icmp_seq=1 ttl=64 time=0.036 ms
64 bytes from 192.168.1.235: icmp_seq=2 ttl=64 time=0.041 ms
64 bytes from 192.168.1.235: icmp_seq=3 ttl=64 time=0.039 ms
--- 192.168.1.235 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2069ms
rtt min/avg/max/mdev = 0.036/0.038/0.041/0.002 ms
```

Mais impossible de faire un ping depuis ma machine hôte alors qu'elles sont toutes sur le même réseau :

```
nk@latitude-3590:~$ ping 192.168.1.235
PING 192.168.1.235 (192.168.1.235) 56(84) bytes of data.
--- 192.168.1.235 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5117ms
```

8. Interdire les paquets provenant du sous-réseau local 192.168.10.0/24 sauf ceux en provenance du poste de travail. Remarque : Vous devez utiliser, dans ce cas, deux règles de filtrage. Appliquer ces règles puis essayer d'établir depuis le poste de travail, une connexion FTP vers le pare-feu. Inversez l'ordre de ces deux règles puis réessayez l'opération. Que remarquez-vous ? Quelle conclusion pouvez-vous en tirer ?

Cas 1 : Ordre correct (autorisation avant interdiction)

Autoriser le poste de travail

```
iptables -A INPUT -s 192.168.1.236 -j ACCEPT
```

Interdire le reste du sous-réseau

```
iptables -A INPUT -s 192.168.1.0/24 -j DROP
```

Le poste 192.168.1.236 peut accéder au pare-feu. Tous les autres hôtes du réseau 192.168.10.0/24 sont bloqués.

Cas 2 : Ordre inversé (interdiction avant autorisation)

Interdire le sous-réseau

```
iptables -A INPUT -s 192.168.1.0/24 -j DROP
```

Autoriser le poste de travail

```
iptables -A INPUT -s 192.168.1.236 -j ACCEPT
```

Le poste 192.168.1.236 est aussi bloqué. Iptables lit les règles dans l'ordre, et la première règle correspondante est appliquée immédiatement. Donc le paquet venant de 192.168.1.236 correspond déjà à la règle de DROP du réseau entier, et ne va jamais jusqu'à la règle ACCEPT.

9. Écrire une règle qui laisse passer 5 tentatives de connexion TCP avec une fréquence de 2 tentatives par minute.

```
iptables -A INPUT -p tcp --syn -m limit --limit 2/minute --limit-burst 5 -j ACCEPT

iptables -A INPUT -p tcp --syn -j DROP
```

Pour réaliser un test de la règle de 5 connexion, j'ai fais un boucle depuis la machine client :

```
root@U2-22-client:~# for i in {1..10}; do ssh -o ConnectTimeout=1 root@192.168.1.235; done
root@192.168.1.235's password:
Linux U2-22-server 6.8.12-8-pve #1 SMP PREEMPT_DYNAMIC PMX 6.8.12-8 (2025-01-24T12:32Z) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Nov 18 10:20:11 2025 from 192.168.1.236
root@U2-22-server:~# exit
logout
Connection to 192.168.1.235 closed.
root@192.168.1.235's password:
Linux U2-22-server 6.8.12-8-pve #1 SMP PREEMPT_DYNAMIC PMX 6.8.12-8 (2025-01-24T12:32Z) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Nov 18 10:22:23 2025 from 192.168.1.236
root@U2-22-server:~# exit
logout
Connection to 192.168.1.235 closed.
root@192.168.1.235's password:
Linux U2-22-server 6.8.12-8-pve #1 SMP PREEMPT_DYNAMIC PMX 6.8.12-8 (2025-01-24T12:32Z) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Nov 18 10:22:26 2025 from 192.168.1.236
root@U2-22-server:~# exit
logout
Connection to 192.168.1.235 closed.
root@192.168.1.235's password:
Linux U2-22-server 6.8.12-8-pve #1 SMP PREEMPT_DYNAMIC PMX 6.8.12-8 (2025-01-24T12:32Z) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
```

```
Last login: Tue Nov 18 10:22:30 2025 from 192.168.1.236
root@U2-22-server:~# exit
logout
Connection to 192.168.1.235 closed.
root@192.168.1.235's password:
Linux U2-22-server 6.8.12-8-pve #1 SMP PREEMPT_DYNAMIC PMX 6.8.12-8 (2025-01-24T12:32Z) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Nov 18 10:22:34 2025 from 192.168.1.236
root@U2-22-server:~# exit
logout
Connection to 192.168.1.235 closed.
ssh: connect to host 192.168.1.235 port 22: Connection timed out
ssh: connect to host 192.168.1.235 port 22: Connection timed out
ssh: connect to host 192.168.1.235 port 22: Connection timed out
ssh: connect to host 192.168.1.235 port 22: Connection timed out
ssh: connect to host 192.168.1.235 port 22: Connection timed out
```

Les 5 premières connexion SSH fonctionne, mais pas les 5 dernières. Si je refais la mainipe 2 minutes plus tard, je peux de nouveau me connecter en ssh.

10. Créer une nouvelle chaîne qui journalise puis rejette tout paquet qui la traverse. Les paquets journalisés doivent être précédés par le préfixe [FIREWALL DROP]. Renvoyer ensuite sur cette nouvelle chaîne tout paquet entrant qui demande l'établissement d'une nouvelle connexion.

Pas compris

11. Positionnez la politique de filtrage par défaut à DROP pour les trois chaînes prédéfinies

Simplement c'est 3 commandes la :

```
iptables -P INPUT DROP

iptables -P OUTPUT DROP

iptables -P FORWARD DROP
```

12. Autoriser tout paquet sortant relatif à une connexion déjà établie ou en rapport avec une connexion déjà établie

Pour autoriser tout paquet sortant qui appartient à une connexion déjà établie ou en rapport avec une connexion existante, on utilise le module state ou conntrack d'iptables.

```
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

13. Interdire tout paquet sortant relatif à une connexion de type INVALID

Pour bloquer tout paquet sortant qui est de type INVALID, on utilise le module state (ou conntrack) :

```
iptables -A OUTPUT -m state --state INVALID -j DROP
```

14. Autoriser tout paquet créant une nouvelle connexion en entrée à destination du port 80

Pour autoriser tout paquet entrant qui initie une nouvelle connexion TCP vers le port 80, on utilise le module state ou conntrack pour détecter les paquets NEW :

```
iptables -A INPUT -p tcp --dport 80 -m state --state NEW -j ACCEPT
```

Un test simple avec un curl pour bien montrer que ça marche :

```
root@U2-22-client:~# curl http://192.168.1.235

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Debian Default Page: It works</title>
  </head>
  <body>
        <p>Vous êtes sur la machine 192.168.1.235 de Guillaume Sanchez
  </body>
</html>
```

15. Utiliser le navigateur web du poste de travail pour accéder à l'adresse URL « http://www.network.net/site ». Que constatez-vous ?

16. A présent, essayer l'adresse URL suivante « http://192.168.10.2/site ». Que constatez-vous cette fois ci ? Un problème qui se pose. Lequel ?

17. Que doit-on faire pour le résoudre ?

Analyse d'un script de filtrage

1. Expliquer pour chaque partie, les règles utilisées et leur rôle.

2. Réaliser pour chaque partie les tests qui permettent d'en vérifier le bon fonctionnement. Inclure dans le rapport la sortie et/ou les captures d'écran des différentes commandes utilisées lors des tests.

3. Selon le script précédent, dire où se situe exactement le pare-feu dans l'architecture réseau qu'il protège puis donner une description générale de la protection qu'il permet d'assurer.