TP - IPtables

`Galiegue Enzo 28/11/2022`

Iptables est une interface en ligne de commande qui permet de configurer Netfilter.

Pour installer iptables, rien de plus simple `apt install iptables`, et pour apprendre à s'en servir `man iptables`

Tables principales de Netfilter

Tables	Description
Filter	Table responsable du filtrage (bloquer ou accepter un paquet), chaque paquet passe par cette table, et traverse une seule chaînes, soit `INPUT`, `OUTPUT` ou `FORWARD`.
NAT	Table responsable de la traduction d'adresses, le premier paquet de chaques connexions passe par cette table, et détermine comment chaque paquet de la connexion vont être transformés avec `PREROUTING`, `POSTROUTING` ou `OUTPUT`.
MANGLE	Table responsable de la transformation des options des paquets, chaque paquet passe à travers cette table, comme elle est prévue pour des options avancés, elle contient toutes les chaînes nommées ci dessus.

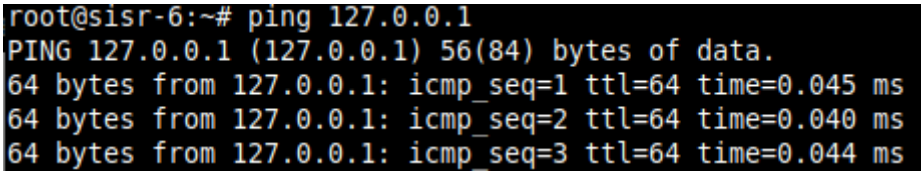
Les chaînes associées aux différents points d'entrée

Chaîne	Table	Description
PREROUTING	NAT / MANGLE	Traduit l'adresse de destination de tout les paquets entrant de cet ordinateur. Avant le routage.
POSTROUTING	NAT / MANGLE	Traduit l'adresse source de tout les paquets sortant de cet ordinateur. Après le routage.
FORWARD	FILTER / MANGLE	Tout les paquets traversant cet ordinateur.
INPUT	FILTER / MANGLE	Tout les paquets destinés à cet ordinateur.
OUTPUT	FILTER / NAT / MANGLE	Tout les paquets créés par cet ordinateur.

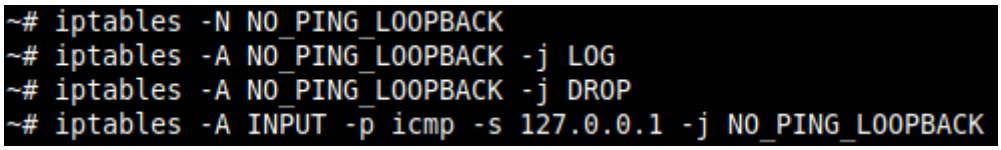
Les cibles prédéfinies les plus courantes

Cible	Description
ACCEPT	Permet d'accepter les paquets qui correspondent à une règle.
DROP	Permet de refuser les paquets qui correspondent à une règle sans avertir le demandeur que la connexion est refusée. (Conseillé)
REJECT	Permet de refuser les paquets qui correspondent à une règle en avertissant le demandeur que la connexion est refusée.
LOG	Permet de journaliser le paquet.
MASQUERADE	Permet d'acheminer le trafic sans perturber le trafic d'origine.
SNAT	Source NAT, réécrit l'adresse de source des paquets sortants à l'adresse du par-feu, permet de mapper l'adresse IP back sur l'adresse IP public, mais également d'empêcher des sources externes d'avoir une adresse directe vers les instances principales.
DNAT	Destination NAT, réécrit l'adresse de destination, qui est l'adresse du par-feu, en adresse de serveurs, iptables transfert donc le trafic entrant vers ces serveurs.
RETURN	Permet d'arrêter de parcourir la chaîne et reprendre à la règle suivante.

On peut voir que l'on peut ping l'adresse de loopback:



Les commandes à exécuter sont les suivantes:




```
iptables -N NO_PING_LOOPBACK
iptables -A NO_PING_LOOPBACK -j LOG
iptables -A NO_PING_LOOPBACK -j DROP
iptables -A INPUT -p icmp -s 127.0.0.1 -j NO_PING_LOOPBACK
```

Pour vérifier, on utilise la commande `iptables -L` pour Lister toutes les règles du pare-feu.

```
root@sisr-6:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
NO_PING_LOOPBACK  icmp --  localhost            anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain NO_PING_LOOPBACK (1 references)
target     prot opt source               destination
LOG        all  --  anywhere             anywhere             LOG level warning
DROP       all  --  anywhere             anywhere
```

On peut lire le contenu d'un fichier en temps réel avec la commande `tail -f /var/log/syslog`, on obtient donc ce rendu là en faisant, en parallèle un `ping 127.0.0.1`, qui lui, n'aura aucun résultat, car la requête est bloquée.

```
Dec 13 13:58:39 sisr-6 kernel: [ 5092.501739] IN=lo OUT= MAC=00:00:00:00:00:00:00:00:00:00:00:00:08:00 SRC=127.0.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=64360 DF PROTO=ICMP TYPE=8 CODE=0 ID=5 SEQ=1
Dec 13 13:58:40 sisr-6 kernel: [ 5093.508820] IN=lo OUT= MAC=00:00:00:00:00:00:00:00:00:00:00:00:08:00 SRC=127.0.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=64400 DF PROTO=ICMP TYPE=8 CODE=0 ID=5 SEQ=2
Dec 13 13:58:41 sisr-6 kernel: [ 5094.532858] IN=lo OUT= MAC=00:00:00:00:00:00:00:00:00:00:00:00:08:00 SRC=127.0.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=64650 DF PROTO=ICMP TYPE=8 CODE=0 ID=5 SEQ=3
Dec 13 13:58:42 sisr-6 kernel: [ 5095.556816] IN=lo OUT= MAC=00:00:00:00:00:00:00:00:00:00:00:00:08:00 SRC=127.0.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=64827 DF PROTO=ICMP TYPE=8 CODE=0 ID=5 SEQ=4
```

Résultat du ping:

```
root@sisr-6:~# 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 ---
7 packets transmitted, 0 received, 100% packet loss, time 6127ms
```

Le ping sur l'adresse de loopback est bien bloqué.

Pour supprimer cette règle c'est simple, avec la commande `iptables -L`, on regarde à quelle ligne se situe la règle.

```
root@sisr-6:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
NO_PING_LOOPBACK  icmp --  localhost            anywhere
```

Ici dans la chaîne `INPUT`, la règle est à la ligne 1, donc la commande pour la supprimer sera:

`iptables -D INPUT 1`

```
root@sisr-6:~# iptables -D INPUT 1
root@sisr-6:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

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

La règle est bien supprimée.

Le Serveur est en `192.168.20.110`

Le Client est en `192.168.20.95`

Les deux se peuvent se ping entre eux.

Serveur vers Client

```
root@sisr-6:~# ping 192.168.20.110
PING 192.168.20.110 (192.168.20.110) 56(84) bytes of data.
64 bytes from 192.168.20.110: icmp_seq=1 ttl=64 time=0.710 ms
64 bytes from 192.168.20.110: icmp_seq=2 ttl=64 time=0.360 ms
```

Client vers Serveur

```
user@debian:~$ ping 192.168.20.95
PING 192.168.20.95 (192.168.20.95) 56(84) bytes of data.
64 bytes from 192.168.20.95: icmp_seq=1 ttl=64 time=0.418 ms
64 bytes from 192.168.20.95: icmp_seq=2 ttl=64 time=0.398 ms
```

On exécute la commande `iptables -A INPUT -p icmp --icmp-type echo-request -j REJECT` sur le Serveur, et on essaye maintenant de ping depuis le Client.

```
user@debian:~$ ping 192.168.20.95
PING 192.168.20.95 (192.168.20.95) 56(84) bytes of data.
From 192.168.20.95 icmp_seq=1 Destination Port Unreachable
From 192.168.20.95 icmp_seq=2 Destination Port Unreachable
From 192.168.20.95 icmp_seq=3 Destination Port Unreachable
```

On a empêcher le ping du poste serveur sur le poste client.

Maintenant on va permettre au client d'accéder au serveur web uniquement en http.

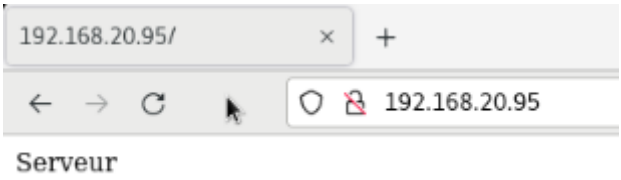
Pour se faire, on doit bloquer le port 443 qui est le port pour le https en entré et sortie.

Avec les commandes suivantes:

```
iptables -A INPUT -p tcp --destination-port 443 -j DROP
iptables -A OUTPUT -p tcp --destination-port 443 -j DROP
iptables -A INPUT -p udp --destination-port 443 -j DROP
iptables -A OUTPUT -p udp --destination-port 443 -j DROP
```

```
root@sisr-6:~# iptables -A INPUT -p tcp --destination-port 443 -j DROP
root@sisr-6:~# iptables -A OUTPUT -p tcp --destination-port 443 -j DROP
root@sisr-6:~# iptables -A INPUT -p udp --destination-port 443 -j DROP
root@sisr-6:~# iptables -A OUTPUT -p udp --destination-port 443 -j DROP
```

En HTTP, la requête est passée.



En HTTPs, la requête ne passe pas.

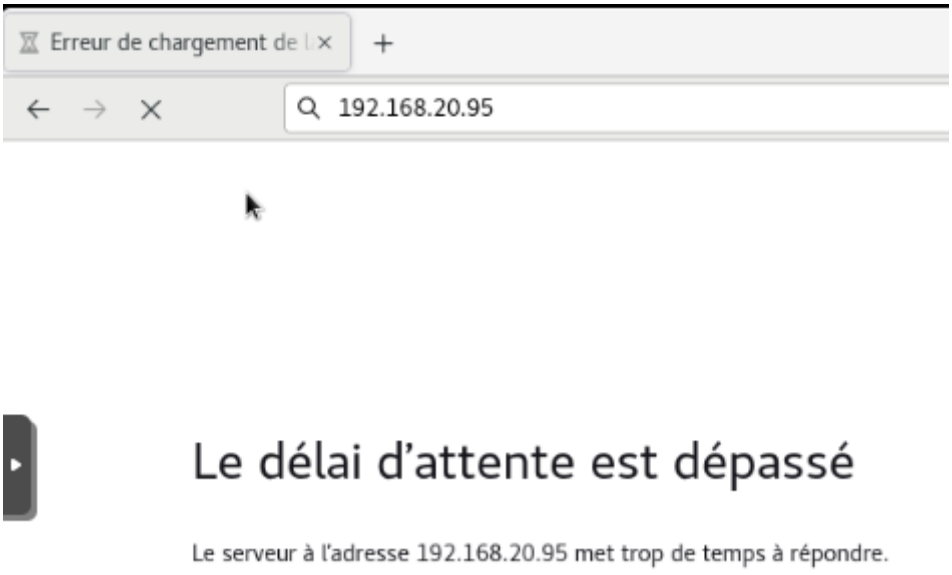
Au lieu de refaire une adresse IP sur la carte réseau, on va temporairement bloquer l'adresse IP déjà présente, et supprimer la règle une fois les tests faits.

On fait simplement la commande:

`iptables -A INPUT -s 192.168.20.110 -j DROP`

```
root@sisr-6:~# iptables -A INPUT -s 192.168.20.110 -j DROP
```

Voici le résultat du blocage:



On peut supprimer la règle avec `iptables -D INPUT 3`

```
root@sisr-6:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  anywhere             anywhere             tcp dpt:https
DROP       udp  --  anywhere             anywhere             udp dpt:443
DROP       all  --  192.168.20.110       anywhere
```

```
root@sisr-6:~# iptables -D INPUT 3
root@sisr-6:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  anywhere             anywhere             tcp dpt:https
DROP       udp  --  anywhere             anywhere             udp dpt:443
```

Pour refuser toute connexion telnet sur le port 23, il suffit de faire la commande:

`iptables -A INPUT -p tcp --destination-port 23 -j DROP`

Ce qui fait que lorsqu'on se connecte depuis le client en telnet, on obtient:

```
user@debian:~$ telnet 192.168.20.95
Trying 192.168.20.95...
telnet: Unable to connect to remote host: Connection refused
```

Maintenant on construit les règles qui répondent aux demandes suivantes:

- Votre poste client ne peut que consulter votre serveur web

```
iptables -A INPUT -s 192.168.20.110 -p tcp --dport 80 -j ACCEPT
iptables -A OUTPUT -s 192.168.20.110 -p tcp --dport 80 -j ACCEPT
```

- Le poste client ne peut pas pinguer votre serveur

```
iptables -A INPUT -s 192.168.20.110 -p icmp --icmp-type echo-request -j REJECT
```

- Le poste client ne peut pas être pingué

```
iptables -A OUTPUT -p icmp -d 192.168.20.110 -j DROP
```

- Votre serveur web est uniquement serveur web

```
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 80 -j ACCEPT
```

- Seules les connexions établies sont acceptées

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

Pour vérifier toutes ces commandes, on liste avec `iptables -L`

```
root@sisr-6:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  anywhere             anywhere             tcp dpt:https
DROP       udp  --  anywhere             anywhere             udp dpt:443
ACCEPT     tcp  --  192.168.20.110       anywhere             tcp dpt:http
REJECT     icmp --  192.168.20.110       anywhere             icmp echo-request reject-with icmp-port-unreachable
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTABLISHED

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  anywhere             anywhere             tcp dpt:https
DROP       udp  --  anywhere             anywhere             udp dpt:443
ACCEPT     tcp  --  192.168.20.110       anywhere             tcp dpt:http
DROP       icmp --  anywhere             192.168.20.110
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTABLISHED

Chain NO_PING_LOOPBACK (0 references)
target     prot opt source               destination
LOG        all  --  anywhere             anywhere             LOG level warning
DROP       all  --  anywhere             anywhere
```