Iptables: Parfeu Linux

Noura Ouerdi

Plan

- **Définition du parfeu**
- **Iptables: Configuration des tables et chaînes du pare-feu intégré**
- **Iptables: Configurer la table NAT du pare-feu**
- **Iptables: Sauvegarder et restaurer la configuration des tables du pare-feu**

Définition du parfeu

- Un **pare-feu** (en anglais firewall) est un logiciel et/ou un matériel permettant de faire respecter la politique de sécurité du réseau, celle-ci définissant quels sont les types de communications autorisés sur ce réseau informatique.
- Il surveille et contrôle les applications et les flux de données (paquets).

Iptables: Configuration des tables et chaînes du pare-feu intégré

- Le pare-feu Iptables se compose de plusieurs tables et chaînes pour la configuration.
- Les paramètres permettent d'autoriser ou de bloquer les paquets entrant, sortant et traversant votre machine.
- Il existe 4 tables dans Iptables:
  La première est la table par défault, les nouvelles règles y sont ajoutées lorsque aucune table n'a été spécifié.
  filter => L'emplacement prévu pour ajouter des filtres sur les paquets entrant, sortant et traversant.
  nat => Permet de faire la traduction d'adresse réseau (NAT) sur différents paquets.
  mangle => Permet de modifier des paquets en changeant les champs de TOS (type de service), et d'autres …
- L'utilisation de la table filter marche de la même façon que la table par défaut.

Comprendre la table filter

- ***Filter*** est la table par défaut.
  C'est la table utilisée pour définir les règles de pare-feu de la machine. En effet, cette table ***filter*** permet de bloquer ou autoriser les paquets qui entrent et sortent.
- On y trouve les chaines:
- **INPUT** : Les paquets à destination de la machine. En général, à destination des processus locaux.
- **OUTPUT** : Les paquets sortant de la machine, en général, provenant des processus locaux.
- **FORWARD** : Les paquets qui passent à travers le serveur, par exemple, d'une interface réseau à l'autre ou à destination d'une autre machine sur le réseau. Cette chaine est utilisée lorsque la machine joue le rôle de routeur.

Comprendre la table nat

La table NAT contient les chaînes suivantes :

- **PREROUTING** : Modifier les paquets AVANT leurs routages. En clair cela, vous permet de rediriger un paquet vers une autre destination en modifiant l'IP de destination par exemple (DNAT – Destination NAT)
- **POSTROUTING** : Modifier les paquets APRES le routage. Cela permet de modifier par exemple l'IP source, SNAT (Source NAT).
- **OUTPUT :** NAT pour les paquets générés par le pare-feu.

Comprendre la table mangle

- La tables Mangle permet de modifier les paquets, elle est surtout utile pour effectuer de la QoS.
  Cette table ne doit pas être utilisé pour modifier les IPs source/destination pour effectuer du NAT.
- Les chaînes de la table Mangle. On retrouve les mêmes que précédemment, avec les mêmes conditions.
- PREROUTING
- OUTPUT
- FORWARD
- INPUT
- POSTROUTING

Iptables

- Syntaxe:

- iptables [-t table] [Action] [Options Valeur] [Cible]
  Quelque options:
  -p protocol
  -s adresse_ip: IP source
  -d adresse_ip: IP de destination
  -j Cible: règle (LOG, ACCEPT, REJECT, DENY, DROP)
  -i "Input Interface"
  -o "Output Interface"
  --dport Le numéro de port
  --line-numbers: Afficher les numéros de ligne des règles

Lister les règles actuelles (-L)

```
# 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
```

Ordre d'application des règles

- iptables traite les paquets de haut en bas, dans l'ordre exact des règles définies
- .Lorsqu'un paquet arrive sur une interface, iptables va le comparer aux règles dans chacune des chaînes pour déterminer à quelle règle il correspond en premier.
- Une fois qu'une règle correspond, elle est appliquée et le traitement des règles s'arrête là.

Fonctionnement général

Lorsqu'un paquet réseau arrive :

1. Il passe par une ou plusieurs **chaînes** en fonction de son type (entrée, sortie, transfert).
2. Il est **analysé règle par règle** dans cette chaîne.
3. La **première règle qui correspond** détermine l'action (ACCEPT, DROP, REJECT, etc.).
4. Une fois qu'une action est prise, **iptables ne lit pas les règles suivantes** pour ce paquet (sauf dans le cas de certaines chaînes personnalisées).

Comment les règles sont priorisées :

- **Politique par défaut** : Les politiques par défaut définissent ce qui se passe pour les paquets qui ne correspondent à aucune règle dans la chaîne. Vous pouvez définir ces politiques en utilisant les commandes iptables -P.
- **Ordre des règles** : Les règles sont traitées dans l'ordre où elles sont spécifiées. La première règle qui correspond à un paquet est appliquée, et le traitement des règles s'arrête là.

Exemple

```
iptables -A INPUT -s 192.168.1.100 -j DROP
iptables -A INPUT -s 192.168.1.0/24 -j ACCEPT
```

- Dans cet exemple, si un paquet provient de l'adresse IP 192.168.1.100, il sera bloqué par la première règle et la deuxième règle ne sera pas appliquée.
- Si le paquet provient d'une autre adresse IP de la plage 192.168.1.0/24, il sera accepté grâce à la deuxième règle.

Ajouter une règle pour un port (-A ou -I)

- On va autoriser le trafic http entrant sur le port "80" de l'interface "eth0".
- Le paramètre -A (append) permet de modifier une chaîne, la règle sera créée à la fin de la chaîne.
- Le paramètre -I (insert) permet d'ajouter une règle en début de chaîne.
- -p Pour choisir le protocol, dans ce cas c'est "tcp "
- -i Pour l'interface où la règle sera appliqué
- --dport Pour le numéro de port, là c'est "80 "
- -j Pour l'autorisation

```
# iptables -A INPUT -p tcp -i eth0 --dport 80 -j ACCEPT
```

Remarque :

- En iptables**, tu dois toujours spécifier le protocole (-p) si tu veux utiliser une condition sur le port** (--dport ou --sport), sinon la commande ne sera pas acceptée (elle renverra une erreur).

- **Les ports sont des notions propres à certains protocoles**, notamment TCP et UDP.
- **Sans** préciser le protocole, **iptables ne sait pas à quoi le port fait référence.**

- On va ajouter une autre règle pour autoriser les connexions ftp entrantes.

```
# iptables -A INPUT -p tcp -i eth0 --dport 21 -j ACCEPT
```

Et aussi le SSH.

```
# iptables -A INPUT -p tcp -i eth0 --dport 22 -j ACCEPT
```

Maintenant on va afficher la liste pour voir nos 3 nouvelles règles

```
s -L -v -n
UT (policy ACCEPT 8 packets, 608 bytes)
es target     prot opt in     out     source               destination
 0 ACCEPT     tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80
 0 ACCEPT     tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0           tcp dpt:21
 0 ACCEPT     tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
```

Autoriser le trafic d'une connexion déjà établie

- On peut autoriser le trafic entrant d'une connexion déjà ouverte.

```
# iptables -A INPUT -m state --state ESTABLISHED -j ACCEPT
```

On voit cette nouvelle règle à la 4ème ligne des chaines INPUT

```
v -n
licy ACCEPT 0 packets, 0 bytes)
get     prot opt in     out     source               destination
EPT     tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80
EPT     tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0           tcp dpt:21
EPT     tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
EPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           state ESTABLISHED
```

Changer la politique (policy) d'une chaine (-P)

- Quand vous affichez vos règles, les paramètres des politiques s'affichent juste après "policy".
  Dans l'exemple ci-dessous, la chaine "INPUT" avec la politique "ACCEPT" permet d'autoriser tout le trafic entrant.

```
# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:www
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ftp
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ssh
ACCEPT     all  --  anywhere             anywhere            state ESTABLISHED
...
```

- Mais pour autoriser que le trafic entrant spécifié dans les règles INPUT, on va utiliser la commande "-P".
  Syntaxe:
  iptables -P [chaine] [politique]
  On va par exemple mettre la politique de bloquer(DROP) sur le trafic entrant(INPUT):

```
iptables -P INPUT DROP
```

On vérifie la politique:

```
# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:www
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ftp
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ssh
ACCEPT     all  --  anywhere             anywhere            state ESTABLISHED
```

L’interface Locale

- L'interface locale (aussi appelée **loopback**) joue un rôle essentiel dans le fonctionnement interne d’un système, en particulier sur les systèmes Unix/Linux. Elle est généralement représentée par :

- Nom de l’interface : **lo**
- Adresse IP : **127.0.0.1**
- Nom d’hôte associé : **localhost**

- **L'interface locale permet à une machine de communiquer avec elle-même via le réseau, sans sortir physiquement sur la carte réseau. C’est une forme de réseau interne au système.**

- **Autoriser l'interface locale**

➢ Quand vous changez de politique pensez à autoriser l'exécution de l'interface locale(loopback) pour permettre aux applications sur la machine elle-même de communiquer entre elles.

➢ Par exemple, un serveur web peut avoir besoin de communiquer avec un serveur de base de données sur la même machine via localhost.

```
# iptables -I INPUT -i lo -j ACCEPT
```

Insérer une règle dans un certain ordre(-I)

- Pour commencer, on va lister les règles avec un numéro par ligne.

```
# iptables -L --line-numbers
Chain INPUT (policy DROP)
num  target     prot opt source               destination
1    ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:www
2    ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ftp
3    ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ssh
4    ACCEPT     all  --  anywhere             anywhere            state ESTABLISHED
```

On va insérer en "3" ème ligne de la chaine "INPUT", l'autorisation du ping entrant (requêtes ICMP).
Puis, On vérifie que la règle s'affiche en 3 ème lignes:

```
# iptables -I INPUT 3 -p icmp -j ACCEPT
```

```
# iptables -L --line-numbers
Chain INPUT (policy DROP)
num  target     prot opt source               destination
1    ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:www
2    ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ftp
3    ACCEPT     icmp --  anywhere             anywhere
4    ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ssh
5    ACCEPT     all  --  anywhere             anywhere            state ESTABLISHED
```

Supprimer une règle d'autorisation(-D)

- Pour commencer, on va lister les règles avec un numéro par ligne.

```
# iptables -L --line-numbers
Chain INPUT (policy DROP)
num  target     prot opt source               destination
1    ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:www
2    ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ftp
3    ACCEPT     icmp --  anywhere             anywhere
4    ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ssh
5    ACCEPT     all  --  anywhere             anywhere            state ESTABLISHED
```

Dans notre exemple on va supprimer les autorisations entrantes du "SSH" qui est sur la ligne "4" de la chaine "INPUT".

Syntaxe: **iptables -D [chaine] [numéro_de_ligne]**

Exemple:

```
iptables -D INPUT 4
```

Règles de la chaine FORWARD

- La chaine "FORWARD" permet d'autoriser à retransmettre (router) des paquets d'une interface à une autre.
- Il faut tout d'abord activer le forward c-à-d **Activer le routage des paquets IP, des datagrammes**
- Vérifiez d'abord l'état de l'IP Forwarding en tapant la commande suivante :

```
cat /proc/sys/net/ipv4/ip_forward
```

- Si la commande retourne la valeur 0, c'est que l'IP Forwarding n'est pas activé.

- Et si la commande retourne la valeur 1, c'est que l'IP Forwarding est activé.

Activer le routage IP et rendre le routage permanent

- Pour l'activer, il suffit de taper la commande :

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Nous voulons éditer le fichier "sysctl.conf" dans le répertoire "/etc" pour rendre cette configuration permanente:

```
nano  /etc/sysctl.conf
```

Et ajouter cette ligne si elle n'existe pas. Si la ligne existe dans le fichier, vérifier bien qu'elle n'est pas commentée (il n'y a pas de "#" en début de ligne).

```
net.ipv4.ip_forward=1
```

Pour activer les changements effectués dans Sysctl.Conf ,vous aurez besoin de la commande:

```
sysctl -p /etc/sysctl.conf
```

- Une fois le routage est activé, autoriser à retransmettre des paquets en entrée(-i) et en sortie(-o) sur l'interface "eth0"

```
iptables -A FORWARD -i eth0 -j ACCEPT
iptables -A FORWARD -o eth0 -j ACCEPT
```

Par exemple pour autoriser à retransmettre des paquets en entrée(-i) de l'interface "eth0" vers la sortie(-o) sur l'interface "eth1" et dans la deuxième ligne dans l'autre sens:

```
iptables -A FORWARD -i eth0 -o eth1 -j ACCEPT
iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT
```

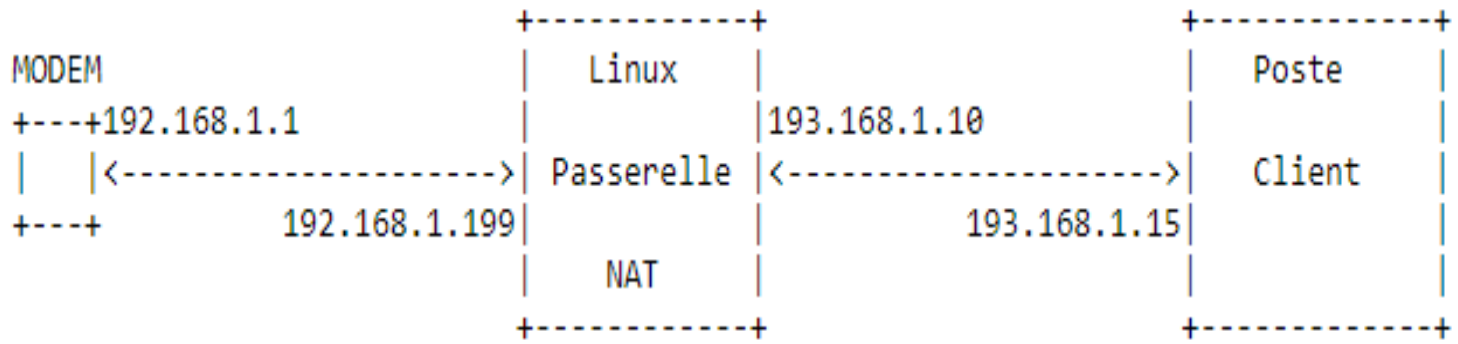
Réinitialiser la configuration -F

- Pour nettoyer toutes les règles de la configuration de votre pare-feu:

```
iptables -F
iptables -X
```

Iptables: Configurer la table NAT du pare-feu

- La table nat permet de faire la translation d'adresse réseau (NAT) sur différents paquets.
- La traduction d'adresse est gérée avec les chaînes PREROUTING, POSTROUTING et OUTPUT.
  Voici le structure de test utilisée pour les exemples qui suivent:



Le DNAT ou NAT Destination

- La chaîne PREROUTING (avant routage) permet de modifier que l'adresse de destination mais conserve l'adresse source.
  C'est ce qu'on appelle faire du DNAT (NAT destination).

- Dans cet exemple, on va rediriger le trafic en destination du réseau "195.111.222.0" vers le réseau "193.168.1.0".

- **iptables -t nat -A PREROUTING -d 195.111.222.0/24 -j DNAT --to-destination 193.168.1.0/24**

Le SNAT ou NAT Source

- La chaîne POSTROUTING (après routage) permet de modifier que l'adresse source mais conserve l'adresse de destination.

- C'est ce qu'on appelle faire du SNAT (NAT source).

- Dans cet exemple, on va substituer l'adresse source du trafic privé sortant vers l'extérieur par une des adresses publiques spécifiées.

- **iptables -t nat -A POSTROUTING -s 193.168.1.0/24 -j SNAT --to-source 192.168.1.5-192.168.1.8**

L'IP Masquerade

- Le principe est d'utiliser une adresse IP publique source pour cacher derrière elle toutes les adresses IP du réseau privé.
- Dans cet exemple avec du SNAT, tous les paquets provenant du réseau privé 193.168.1.0 seront perçus comme provenant de l'IP public 192.168.1.199.
  iptables -t nat -A POSTROUTING -s 193.168.1.0/24 -j SNAT --to-source 192.168.1.199

  Il y a aussi l'option "MASQUERADE" qui a le même but.
  iptables -t nat -A POSTROUTING -s 193.168.1.0/24 -j MASQUERADE

  Vous pouvez utiliser d'autres options, comme choisir par rapport à votre interface réseau , ici "eth0" et au numéro de port, dans ce cas "80".
  iptables -t nat -A POSTROUTING -o eth0 --dport 80 -j MASQUERADE

Afficher la table NAT du parfeu

```
# iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination

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

Supprimer une règle de la table NAT(-D)

- Pour commencer, on va lister les règles avec un numéro par ligne.

```
# iptables -t nat -L --line-numbers
Chain PREROUTING (policy ACCEPT)
num  target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
num  target     prot opt source               destination
1    MASQUERADE  all  --  193.168.1.0/24       anywhere

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

Dans notre exemple on va supprimer la règle "1" de la chaine "POSTROUTING".

Syntaxe:

iptables -t nat -D [chaine] [numéro_de_ligne]

Exemple:

```
# iptables -t nat -D POSTROUTING 1
```

Iptables: Sauvegarder et restaurer la configuration des tables du pare-feu

- Pour sauvegarder les règles des tables du pare-feu "Iptables" dans un fichier texte "iptables-regles.txt":

```
# iptables-save > iptables-regles.txt
```

Pour restaurer la configuration du firewall à partir du fichier texte "iptables-regles.txt":

```
# iptables-restore < iptables-regles.txt
```