

# $\begin{array}{c} \textbf{COMPTE RENDU} \\ \textbf{TP } \, \textbf{N}^{\circ} \textbf{4} - \textbf{FIREWALL NETFILTER} \end{array}$

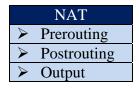
Master 1 IRS P15 Année 2012-2013

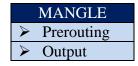
## 1) Prise en main:

Tout au long de ce TP, nous allons nous familiariser avec l'outil IPTABLE qui permet de gérer le pare-feu sous linux. L'outil recense trois tables (FILTER, NAT et MANGLE) qui contiennent par défaut des chaines telles qu'Input, Ouput, Prerouting... Dans la suite du TP, nous verrons que l'utilisateur pourra ajouter des chaines personnalisées dites "chaines utilisateurs" qui seront rattachées à l'une des chaines existantes. Cela permet de simplifier la gestion des tables lorsqu'il a beaucoup de règles mises en place.

Voici un schéma résumé des différentes tables :

	FILTER
>	Input
>	Output
>	Forward





Avant toute chose, il faut s'assurer qu'IPTABLE est présent sur le système (yum list installed iptables ou par dpkg --list "iptables" suivant la distribution choisie).

Si tel n'est pas le cas, il faut l'installer via la commande : yum install iptables

iptables -L permet de lister les règles qui sont actives.

iptables -F permet de supprimer l'ensemble des règles de toutes les chaines

iptables -D numero\_de\_la\_règle permet d'effacer une règle d'une chaine (iptables -L -- line-numbers pour afficher les règles avec les numéros de ligne)

iptables -F nom de la chaine permet d'effacer toutes les règles d'une chaine

iptables -X nom\_de\_la\_chaine permet de supprimer une chaine utilisateur

iptables --delete-chain permet de supprimer toutes les chaines utilisateur

iptables -I nom\_de\_la\_chaine numero\_de\_la\_chaine ma\_règle permet d'insérer une règles à une position désirée parmi celles déjà présentes.

Le fichier /etc/sysconfig/iptables contient l'ensemble des règles actives (c'est ce fichier que la commande *iptables -L* affiche.

La commande : service iptables save permet de sauvegarder la configuration courante dans le fichier /etc/sysconfig/iptables. Lors du redémarrage du service iptables via la commande service iptables restart, les règles contenues dans ce fichier sont lues et chargées dans iptables.

<u>NB</u>: Pour connaitre l'ensemble des commandes se référents à l'outil iptables, il suffit de consulter le manuel (man iptables).

# 2) Filtrage de ports :

## a. Exercice 1:

Dans cet exercice, nous allons interdire les connexions entrantes sur le port 22 du service SSH, le port 25 du service SMTP et n'autoriser qu'une seule machine à s'y connecter. Il faut bien garder à l'esprit que <u>les règles sont exécutées dans l'ordre</u> et il faut donc être extrêmement vigilent sur ce point.

```
-A INPUT : ajoute en bas de la liste les règles de la chaine INPUT (A=append)
-I INPUT 2 : ajoute la règles en position 2 (I=insert)
```

-D INPUT 5 : supprime la règles en position 5 (D=delete)

-p tcp: pour les paquets TCP

--dport 22 : pour les paquets à destination du port 22 (dport=port destination et sport=port source)

-m multiport --dports 11,22,33 : pour les paquets à destination des ports 11, 22 et 33. Il fait séparer les différents ports par une virgule.

-s 10.2.3.4 : spécifie une adresse IP source. Ici, la règle s'applique uniquement pour une connexion depuis l'adresse IP 10.2.3.4 (option -d pour une IP de destination) -j DROP : pour supprimer les paquets (-j = --jump spécifie l'action à réaliser)

## • Règle pour le service SSH:

```
//on crée une chaine utilisateur nommée "ssh"
iptables -N ssh
```

//on place la chaine dans la table FILTER (input)

```
iptables -A INPUT -j ssh
```

//on autorise uniquement la machine 192.168.11.2 à se connecter sur notre machine via le port 22

```
iptables -A ssh -p tcp --dport 22 -s 192.168.11.2 -j ACCEPT
```

//on log l'accès refusé puis on drop, mais pas l'inverse !! Les logs sont stockés dans le fichier "/var/log/message" iptables -A ssh -j LOG --log-prefix 'acces SSH bloque'

//on interdit tout le reste

iptables -A ssh -j DROP

#### • Règle pour le service SMTP :

```
//on crée une chaine utilisateur nommée "mail"
iptables -N mail

//on place la chaine dans la table FILTER (input)
iptables -A INPUT -j mail

//on autorise uniquement la machine 192.168.11.2 à se connecter sur notre machine via le port 25
iptables -A mail -p tcp --dport 25 -s 192.168.11.2 -j ACCEPT

//on log l'accès refusé
iptables -A mail -j LOG --log-prefix 'acces mail bloque'
```

//on interdit tout le reste

iptables -A mail -j DROP

#### b. Exercice 2:

Nous allons interdire l'accès http et https de tous les utilisateurs, excepté pour root.

//on crée une chaine utilisateur nommée "web-sortant"

iptables -N web-sortant

//on place la chaine dans la table FILTER (output)

iptables -A OUTPUT -j web-sortant

//on autorise l'accès internet (http et https) uniquement à l'utilisateur "root"

iptables -A web-sortant -p tcp -m multiport --dports 80,443 -m owner --uid-owner root -j ACCEPT

//si on est pas root, on supprime tous les paquets sortants vers http et https

iptables -A web-sortant -p tcp -m multiport --dports 80,443 -j DROP

### c. Exercice 3:

Nous allons créer une règle pour nous prémunir des attaques de type "ping flood" en limitant le nombre de paquets ICMP que notre machine va accepter par seconde. Dans cet exemple nous avons fixé la valeur à 1 paquet maximum par seconde.

iptables -A INPUT -p icmp -m limit --limit 1/second

## 3) Suivi de connexion (conntrack):

Conntrack permet de réagir intelligemment sur une connexion donnée, suivant son état (NEW, ETABLISHED, RELATED ou INVALID).

Le module "state" va permettre de détecter un flux et d'ouvrir dynamiquement des ports du pare feu vers le client. Prenons l'exemple du protocole FTP en mode passif : Si nous créons une règle utilisant "state", cela va permettre d'écouter sur le port 20 et 21, puis dès qu'une connexion est établie entre le serveur et le client d'ouvrir le port choisi par le serveur pour le transfert des données.

//commençons par charger le module dans le noyau permettant de gérer le suivi de connexion ftp. Attention ce module ne gère que les connexions ftp passives. Sinon, il faut utiliser le module "nf\_conntrack\_ftp"

modprobe ip\_conntrack\_ftp

//on efface toutes les règles existantes

iptables -F

//on interdit tous les paquets

iptables -A INPUT -j DROP

On constate que la connexion au serveur FTP en mode passif est impossible puisque les paquets sont supprimés (dropés). Pour autoriser la connexion, il faut ajouter la règle suivante en première position :

iptables -I INPUT 1 -p tcp -m multiport --sports 20,21 -m state --state ESTABLISHED -j ACCEPT

Mémo connexion ftp mode passif

[root@toto:/]#ftp ftp> passive ftp>open 192.168.1.1

[root@toto:/]#ftp

4/6

## 4) Exercice de réflexion ;

A) Il existe deux types de politique par défaut pouvant régir un filtre : restrictive et permissive.

B)

1/ Interdire tout service quel qu'il soit entrant ou sortant (règles à ajouter en dernier):

```
iptables -A INPUT -j DROP
iptables -A OUTPUT -j DROP
```

2/ Une connexion TCP établie doit être définitivement acceptée :

```
iptables -A INPUT -p tcp -m state --state ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp -m state --state ESTABLISHED -j ACCEPT
```

3/ Autoriser les connexions sortantes vers le web (80, http) quel que soit le serveur visité :

```
iptables -A OUTPUT -p tcp --sport 80 -m state --state NEW -j ACCEPT
```

4/ Autoriser les connexions mail (25, smtp) vers notre machine.

```
iptables -A INPUT -p tcp --dport 25 -m state --state NEW -j ACCEPT
```

5/ Autoriser les connexions ftp vers l'extérieur ainsi que les connexions de données associées sur le port 20 (ftp-data) telles que les commandes get et put

```
iptables -A OUTPUT -p tcp -m multiport --sports 20,21 -m state --state NEW
-j ACCEPT
```

6/ Autoriser toutes les connexions vers l'extérieur mais pas celles entrantes :

```
iptables -I INPUT 2 -p tcp -m state --state NEW -j REJECT
iptables -I OUTPUT 2 -p tcp -m state --state NEW -j ACCEPT
```

7/ Autoriser les ping et traceroute vers notre machine :

```
iptables -A INPUT -p icmp -m state --state NEW -j REJECT
iptables -A INPUT -p udp --dport 33434:33523 -m state --state NEW -j ACCEPT
```

**C**)

On souhaite protéger tout un réseau local placé derrière notre machine. Notre machine aura donc un rôle de "routeur pare-feu". Pour notre exemple eth0 sera l'interface reliée au réseau local et eth1 l'interface réseau relié vers l'extérieur.

```
-i: carte réseau d'entrée (in interface).-o: carte réseau de sortie (out interface)
```

```
//on accepte les connexions établies (et celles qui y sont liées)
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

//on accepte uniquement les nouvelles connexions depuis le réseau local vers l'extérieur
iptables -A FORWARD -i eth0 -o eth1 -p tcp -m state --state NEW -j ACCEPT

//on accepte les connexions ftp, ssh, http et https
iptables -A FORWARD i eth0 -o eth1 -p tcp -multiport --sport 21,22,80,443 -
j ACCEPT

//on bloque tout le reste
iptables -A FORWARD -j DROP
```

#### D)

• Nous allons bloquer l'accès aux paquets arrivants sur l'interface externe avec une adresse source correspondant à une adresse du réseau interne :

```
iptables -A INPUT -m iprange --src-range 192.168.10.0-10.255.255.255 -j
DROP
```

 Nous allons bloquer l'accès aux paquets de l'Internet qui ont pour adresse IP une adresse privée (10.0.0.0 à 10.255.255.255, 172.16.0.0 à 172.31.255.255 et 192.168.0.0 à 192.168.255.255)

```
iptables -A INPUT -m iprange --src-range 10.0.0.0-10.255.255.255 -j DROP iptables -A INPUT -m iprange --src-range 172.16.0.0-172.31.255.255 -j DROP iptables -A INPUT -m iprange --src-range 192.168.0.0-192.168.255.255 -j DROP
```