

CHAPITRE 1

IPTABLES

Si vous ne connaissez pas encore « iptables », c'est LA commande pour gérer les connexions réseau, faire des redirections de port, blacklister une adresse IP ou tout simplement ouvrir ou fermer des ports. C'est ce que l'on appelle un pare-feu ou firewall.

Par défaut « iptables » laisse tout ouvert et ne filtre rien. Notre but c'est de fermer un peu tout ça et de garder le minimum ouvert.

Je ne rentrerai pas dans le détails de l'utilisation de « iptables » car bien trop de choses sont à voir mais nous allons l'utiliser d'une manière simple et efficace ce qui vous donnera les bases minimales pour commencer à appréhender « iptables ».

Nous allons paramétrer « iptables » de manière à l'utiliser que pour nos besoins. A l'heure actuelle, seul « apt » (la gestion des paquets Debian) et « ntpdate » (synchronisation de l'heure) font des connexions réseau. Nous allons donc tout bloquer puis ouvrir un passage à « apt » et « ntpdate ».

D'abord on supprime toutes les règles existante (des fois qu'ils y en aurait de rentrées) :

```
ALBAN@bebeserv:~$ sudo iptables -t filter -F INPUT
ALBAN@bebeserv:~$ sudo iptables -t filter -F OUTPUT
```

Maintenant on ferme les portes :

```
ALBAN@bebeserv:~$ sudo iptables -t filter -P INPUT DROP
ALBAN@bebeserv:~$ sudo iptables -t filter -P OUTPUT DROP
```

Un reflex qu'il faut avoir c'est de toujours autoriser la boucle locale (et ceci doit être toujours la première règle) :

```
ALBAN@bebeserv:~$ sudo iptables -t filter -A INPUT -i lo -j ACCEPT
ALBAN@bebeserv:~$ sudo iptables -t filter -A OUTPUT -o lo -j ACCEPT
```

Dorénavant nous sommes totalement fermé de l'extérieur. Configurons « iptables » pour laisser « apt ».

« apt » aura déjà besoin de faire des requêtes DNS (port 53 en UDP et TCP) pour résoudre les noms tel que « ftp.debian.fr » afin de pouvoir récupérer les listes et les paquets.

Ouvrons le port DNS en TCP et UDP pour l'entrée et la sortie :

```
ALBAN@bebeserv:~$ sudo iptables -t filter -A INPUT -i eth0 -d
10.0.0.2 -p tcp --dport 53 -j ACCEPT
ALBAN@bebeserv:~$ sudo iptables -t filter -A INPUT -i eth0 -d
10.0.0.2 -p udp --dport 53 -j ACCEPT
ALBAN@bebeserv:~$ sudo iptables -t filter -A OUTPUT -o eth0 -s
10.0.0.2 -p tcp --dport 53 -j ACCEPT
```

```
ALBAN@bebeserv:~$ sudo iptables -t filter -A OUTPUT -o eth0 -s
10.0.0.2 -p udp --dport 53 -j ACCEPT
```

De même ouvrons le port 80 en TCP pour que « apt » puisse télécharger les paquets :

```
ALBAN@bebeserv:~$ sudo iptables -t filter -A INPUT -i eth0 -d
10.0.0.2 -p tcp --dport 80 -j ACCEPT
ALBAN@bebeserv:~$ sudo iptables -t filter -A OUTPUT -o eth0 -s
10.0.0.2 -p tcp --dport 80 -j ACCEPT
```

Maintenant autorisons « ntpdate » qui lui utilise le port 123 en TCP et UDP :

```
ALBAN@bebeserv:~$ sudo iptables -t filter -A INPUT -i eth0 -d
10.0.0.2 -p tcp --dport 123 -j ACCEPT
ALBAN@bebeserv:~$ sudo iptables -t filter -A INPUT -i eth0 -d
10.0.0.2 -p udp --dport 123 -j ACCEPT
ALBAN@bebeserv:~$ sudo iptables -t filter -A OUTPUT -o eth0 -s
10.0.0.2 -p tcp --dport 123 -j ACCEPT
ALBAN@bebeserv:~$ sudo iptables -t filter -A OUTPUT -o eth0 -s
10.0.0.2 -p udp --dport 123 -j ACCEPT
```

A ce stade nous pourrions nous arrêter là tout es fonctionnel. Mais allons plus loin.
Maintenant ajoutons une règle qui dira que toute connexion déjà établie avec le serveur donc déjà autoriser par « iptables » sera elle aussi autorisée (même sur un port fermé par exemple).

```
ALBAN@bebeserv:~$ sudo iptables -t filter -A INPUT -i eth0 -d
10.0.0.2 -m state --state RELATED,ESTABLISHED -j ACCEPT
ALBAN@bebeserv:~$ sudo iptables -t filter -A OUTPUT -o eth0 -s
10.0.0.2 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

Dernière ces règles nous allons loguer dans le fichier « /var/log/debug » tout les paquets que nous allons bloquer avant de les bloquer :

```
ALBAN@bebeserv:~$ sudo iptables -t filter -A INPUT -i eth0 -d
10.0.0.2 -j LOG --log-prefix "Iptables INPUT dropped : " --log-level
debug
ALBAN@bebeserv:~$ sudo iptables -t filter -A OUTPUT -o eth0 -s
10.0.0.2 -j LOG --log-prefix "Iptables OUPUT dropped : " --log-level
debug
```

Il est est terminer des règles.
Maintenant voyons quelques commandes utiles.
Sauver votre configuration :

```
ALBAN@bebeserv:~$ sudo iptables-save > ~/iptables-sauvegarde
```

Restaurer votre configuration :

```
ALBAN@bebeserv:~$ sudo iptables-restore ~/iptables-sauvergade
```

Voir votre configuration complète :

```
ALBAN@bebeserv:~$ sudo iptables -L -v --line-numbers
```

Vérifier le log en temps réel (paquets rejetés) : (tapez « [CTRL + C] » pour sortir)

```
ALBAN@bebeserv:~$ sudo tail -f /var/log/debug
```

Quand le serveur redémarre vos règles « iptables » seront perdues, il faut donc automatiser la mise en place de ces règles.

Allez dans « /etc/network » :

```
ALBAN@bebeserv:~$ cd /etc/network
ALBAN@bebeserv:/etc/network$
```

Créez le fichier « iptables » à côté du fichier « interface » :

```
ALBAN@bebeserv:/etc/network$ sudo nano iptables
```

Entrez les mêmes lignes que nous avons du taper pour configurer « iptables » (attention aux fautes de frappes !) :

```
iptables -t filter -F INPUT
iptables -t filter -F OUTPUT

iptables -t filter -Z INPUT
iptables -t filter -Z OUTPUT

iptables -t filter -P INPUT DROP
iptables -t filter -P OUTPUT DROP

iptables -t filter -A INPUT -i lo -j ACCEPT
iptables -t filter -A INPUT -i eth0 -d 10.0.0.2 -p tcp --dport 53 -j
ACCEPT
iptables -t filter -A INPUT -i eth0 -d 10.0.0.2 -p udp --dport 53 -j
ACCEPT
iptables -t filter -A INPUT -i eth0 -d 10.0.0.2 -p tcp --dport 80 -j
ACCEPT
iptables -t filter -A INPUT -i eth0 -d 10.0.0.2 -p tcp --dport 123
-j ACCEPT
iptables -t filter -A INPUT -i eth0 -d 10.0.0.2 -p udp --dport 123
-j ACCEPT
iptables -t filter -A INPUT -i eth0 -d 10.0.0.2 -m state --state
RELATED,ESTABLISHED -j ACCEPT
iptables -t filter -A INPUT -i eth0 -d 10.0.0.2 -j LOG --log-prefix
"Iptables INPUT dropped : " --log-level debug

iptables -t filter -A OUTPUT -o lo -j ACCEPT
iptables -t filter -A OUTPUT -o eth0 -s 10.0.0.2 -p tcp --dport 53
-j ACCEPT
iptables -t filter -A OUTPUT -o eth0 -s 10.0.0.2 -p udp --dport 53
```

```

-j ACCEPT
iptables -t filter -A OUTPUT -o eth0 -s 10.0.0.2 -p tcp --dport 80
-j ACCEPT
iptables -t filter -A OUTPUT -o eth0 -s 10.0.0.2 -p tcp --dport 123
-j ACCEPT
iptables -t filter -A OUTPUT -o eth0 -s 10.0.0.2 -p udp --dport 123
-j ACCEPT
iptables -t filter -A OUTPUT -o eth0 -s 10.0.0.2 -m state --state
RELATED,ESTABLISHED -j ACCEPT
iptables -t filter -A OUTPUT -o eth0 -s 10.0.0.2 -j LOG --log-prefix
"Iptables OUPUT dropped : " --log-level debug

```

Enregistrez le fichier « [CTRL + O] » et quittez « [CTRL + X] » :

Créez le fichier de démarrage,

```

ALBAN@bebeserv:/etc/network$ cd /etc/init.d/
ALBAN@bebeserv:/etc/init.d$ sudo nano iptables-conf

```

Entrez ceci :

```

#!/bin/sh
set -e

iptables_start() {
    if [ -f /etc/network/iptables ]; then
        . /etc/network/iptables
    fi
}

iptables_stop() {
    iptables -t filter -F INPUT
    iptables -t filter -F OUTPUT
    iptables -t filter -P INPUT ACCEPT
    iptables -t filter -P OUTPUT ACCEPT
}

case "$1" in
    start)
        echo -n "Apply Iptables configuration"
        iptables_start
        echo "."
        ;;
    stop)
        echo -n "Clear Iptables configuration"
        iptables_stop
        echo "."
        ;;
    restart)
        echo -n "Reloading Iptables configuration"
        iptables_stop
        iptables_start
        echo "."
        ;;
)

```

```
*)
    echo "Usage: /etc/init.d/iptables-conf {start|stop|
restart}"
    exit 1
esac

exit 0
```

Donnons les bons droits à notre fichier :

```
ALBAN@bebeserv:/etc/init.d$ sudo chmod +x iptables-conf
```

Enregistrez et quittez.

Rajoutez le script de démarrage dans le mode voulu avec la priorité voulue :

```
ALBAN@bebeserv:/etc/init.d$ sudo update-rc.d iptables-conf start 99
2 3 4 5 . stop 20 0 1 6 .
[...]
```

Testons le tout, affichons la configuration actuelle :

```
ALBAN@bebeserv:/etc/init.d$ sudo iptables -L -v --line-numbers
[...]
```

Stoppez le firewall :

```
ALBAN@bebeserv:/etc/init.d$ sudo /etc/init.d/iptables-conf stop
[...]
```

Vérifiez que tout est vide et autorisé :

```
ALBAN@bebeserv:/etc/init.d$ sudo iptables -L -v --line-numbers
[...]
```

Démarrons le firewall :

```
ALBAN@bebeserv:/etc/init.d$ sudo /etc/init.d/iptables-conf start
[...]
```

Vérifions que toutes les règles sont bien présentes :

```
ALBAN@bebeserv:/etc/init.d$ sudo iptables -L -v --line-numbers
[...]
```

Modifions nous même la configuration actuelle :

```
ALBAN@bebeserv:/etc/init.d$ sudo iptables -F INPUT
[...]
ALBAN@bebeserv:/etc/init.d$ sudo iptables -L -v --line-numbers
[...]
```

Redémarrons le firewall pour vérifier que tout revient comme nous l'avions configuré :

```
ALBAN@bebeserv:/etc/init.d$ sudo /etc/init.d/iptables-conf restart
[...]
ALBAN@bebeserv:/etc/init.d$ sudo iptables -L -v --line-numbers
[...]
```

Voilà c'est terminé pour ce chapitre !

```
ALBAN@bebeserv:/etc/init.d$ cd ~
ALBAN@bebeserv:~$
```