

FireWall

Avertissement

Ce guide est réservé aux personnes ayant un **bon niveau** dans l'administration des serveurs dédiés sous Linux. La manipulation d'un firewall peut être **dangereuse**. En effet, vous pouvez bloquer votre serveur ce qui vous forcera à le redémarrer en HARD. Si vous vous trompez dans le script final et que vous le mettez en auto démarrage, vous n'aurez plus accès à votre machine ! Faites donc **très attention** et si vous ne vous sentez pas à l'aise avec ce guide, ne paramétrez pas votre firewall !

Un firewall c'est quoi ?

C'est un programme qui va bloquer certains ports de votre machine et en ouvrir d'autre. Imaginez votre maison : vous avez une porte à l'avant et une porte à l'arrière. Celle à l'arrière ne vous sert jamais, autant la condamner. Pourquoi ? Parce que c'est un risque potentiel d'intrusion pour un voleur. Avec le firewall c'est pareil, on va fermer tous les ports dont nous n'avons pas besoin.

Quels sont les ports dont vous vous servez ?

Attention !

Avant toute chose, il faut faire très attention à ce que vous allez faire. En effet, vous risquez de vous tromper de ports et de fermer les mauvais. Imaginez si vous fermez le port SSH ! Il faudra alors soit redémarrer via telnet, soit via webmin ou alors rebooter en hard :/.

Les ports ouverts par défaut sur les serveurs OVH sont :

- 21 – ftp (le serveur FTP, à laisser selon utilisation).
- 22 – ssh (l'accès au shell crypté, à laisser !).
- 23 – telnet (l'accès au shell non crypté, à laisser en dépannage).
- 25 – smtp (le serveur de courrier sortant, à laisser dans la plupart des cas).
- 53 – dns (le serveur DNS, à laisser dans la plupart des cas).
- 80 – http (le serveur web, à laisser).
- 110 – pop3 (l'accès aux mails, à laisser dans la plupart des cas).
- 143 – imap (l'accès aux mails, à laisser si vous n'utilisez pas pop3).
- 443 – https (l'accès au web crypté, à laisser selon votre utilisation).
- 1000 – webmin (panneau de configuration du serveur, à laisser si vous vous en servez).

Ces ports sont ceux ouverts par défaut mais vous avez peut-être des logiciels lancés qui en ouvrent d'autres. À vous de savoir lesquels sont à garder ou non. Une fois votre choix effectué, passons à la mise en ?uvre.

Iptables

Iptables est un firewall très performant, installé sur tous les serveurs OVH. Le fonctionnement va être le suivant : nous allons ouvrir certains ports et fermer tout le reste. Dans cet exemple, nous allons laisser que le port 22 (SSH) et 80 (HTTP). Ce n'est qu'un exemple, c'est à vous de l'adapter par rapport à vos besoins.

Connectez-vous avec SSH en root.

La première chose à faire est de vérifier la version d'iptables :

```
$ /sbin/iptables -V
iptables v1.2.4
```

La version est ici trop ancienne. On va mettre la 1.2.9 :

```
$ cd /root
$ wget http://www.netfilter.org/files/iptables-1.2.9.tar.bz2
$ tar xvfj iptables-1.2.9.tar.bz2
$ cd iptables-1.2.9
```

```

$ make KERNEL_DIR=/usr/src/linux
$ make install KERNEL_DIR=/usr/src/linux
$ cd /sbin
$ mv iptables iptables.old
$ mv iptables-restore iptables-restore.old
$ mv iptables-save iptables-save.old
$ ln -s /usr/local/sbin/iptables iptables
$ ln -s /usr/local/sbin/iptables-restore iptables-restore
$ ln -s /usr/local/sbin/iptables-save iptables-save
$ /sbin/iptables -V
iptables v1.2.9

```

Voilà, iptables est mis à jour, on peut poursuivre.

On liste les règles existantes :

```

$ /sbin/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

```

On voit trois rubriques : Input, Forward et Output.

Nous allons nous occuper que de la rubrique Input pour le moment (pour le trafic entrant).

On autorise les ports 22 et 80 :

```

$ /sbin/iptables -A INPUT -i eth0 -p tcp --dport 22 -j ACCEPT
$ /sbin/iptables -A INPUT -i eth0 -p tcp --dport 80 -j ACCEPT

```

-A INPUT : on place notre règle en entrée du firewall.

-i eth0 : ici c'est l'interface ethernet qui nous intéresse.

-p tcp : le protocole traité est le TCP (on ne traite que celui-là pour le moment).

--dport 22 : la règle va être appliquée sur le port SSH (n° 22).

-j ACCEPT : on accepte ce trafic.

On reliste tout :

```

$ /sbin/iptables -L
Chain INPUT (policy ACCEPT)
target prot opt source destination
ACCEPT tcp anywhere anywhere tcp dpt:ssh
ACCEPT tcp anywhere anywhere tcp dpt:www

```

```

Chain FORWARD (policy ACCEPT)
target prot opt source destination

```

```

Chain OUTPUT (policy ACCEPT)
target prot opt source destination

```

La rubrique Input se remplit, c'est bon signe ;).

On voit que la politique par défaut est de tout accepter => Chain INPUT (policy ACCEPT). Nous voulons bloquer tout le trafic qu'on n'aura pas au préalable autorisé. On va donc ajouter une règle qui va bloquer les autres ports. Mais il se pose alors un problème : lorsqu'une connexion va être faite depuis notre serveur vers le serveur kernel.org pour télécharger le nouveau noyau (ce n'est qu'un exemple), il va établir une connexion vers le site et va attendre sa réponse. La demande de connexion va bien partir mais comment va-t-elle revenir étant donné qu'on a tout bloqué ? Heureusement, iptables est puissant et peut aussi trier les paquets par rapport à leur état. On va donc ajouter une règle :

```
$ /sbin/iptables -A INPUT -i eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Il ne reste plus qu'à bloquer le reste (attention, c'est là où le firewall va vraiment faire effet, vérifiez donc d'avoir bien rentré vos règles sinon vous allez bloquer votre serveur !) :

```
$ /sbin/iptables -A INPUT -i eth0 -j DROP
```

On a en fait deux choix ou niveau de cette règle. Soit on droppe les paquets, c'est-à-dire que si un paquet arrive et qu'il n'est pas accepté on l'efface. Le client attendra de son côté une réponse en vain, jusqu'au timeout. La deuxième solution est de rejeter les paquets (REJECT au lieu de DROP). Si un paquet non sollicité arrive, on renvoie au client une erreur et il n'attend plus car il a une réponse négative. Rejeter les paquets est plus propre mais les jeter est plus sécurisé. En effet, imaginons quelqu'un qui vous envoie des paquets à répétition sur un mauvais port, votre serveur ne les traitera pas, alors qu'avec la règle reject, il prendra le temps de répondre.

À vous de voir ;)

Pour mettre à zéro votre firewall, tapez :

```
$ /sbin/iptables -F INPUT
```

Cette commande supprimera toutes les règles de la rubrique INPUT.

Si vous souhaitez ajouter une règle entre la première et la deuxième, tapez ceci :

```
$ /sbin/iptables -I INPUT 2 ... la suite de votre règle
```

Pour supprimer la règle n°3 tapez ceci :

```
$ /sbin/iptables -D INPUT 3
```

Pour bloquer totalement une IP :

```
$ /sbin/iptables -I INPUT 1 -s <IP> -j DROP
```

Le firewall est maintenant en action. Essayez de scanner votre serveur, vous ne verrez que les ports 22 et 80 d'ouverts. Ne vous étonnez pas si le scan est très lent, c'est à cause de la règle DROP.

IP à exclure... et à autoriser ;)

Pour un Serveur dédié :

Si vous souhaitez bloquer le protocole ICMP (les requêtes ping), vous devez laisser au moins ping.ovh.net, proxy.p19.ovh.net, proxy.rbx.ovh.net, proxy.ovh.net et proxy.rbx2.ovh.net vous pinguer. Cela permet aux

OVH

équipes d'OVH de vérifier le bon état de votre serveur.

Vous devez de plus laisser passer l'IP obtenue de la façon suivante :

L'IP de votre serveur est de la forme aaa.bbb.ccc.ddd

Vous devez laisser passer : aaa.bbb.ccc.250

Ex 213.186.57.143 doit laisser passer 213.186.57.250 pour le serveur SLA et 213.186.57.251 pour le serveur mrtg afin de pouvoir bénéficier de RTM.

Si vous êtes sur un serveur HG, laissez passer de plus l'IP aaa.bbb.ccc.249 (règle temporaire).

Si vous bloquez toutes les requêtes ping, même celles d'OVH, nous ne surveillerons plus le bon état de marche de votre serveur et s'il tombe nous n'en serons pas avertis. Pour autoriser le ping depuis nos serveurs, entrez les règles suivantes :

```
/sbin/iptables -A INPUT -i eth0 -p icmp --source proxy.ovh.net -j ACCEPT
```

```
/sbin/iptables -A INPUT -i eth0 -p icmp --source proxy.p19.ovh.net -j ACCEPT
```

```
/sbin/iptables -A INPUT -i eth0 -p icmp --source proxy.rbx.ovh.net -j ACCEPT
```

```
/sbin/iptables -A INPUT -i eth0 -p icmp --source proxy.rbx2.ovh.net -j ACCEPT
```

```
/sbin/iptables -A INPUT -i eth0 -p icmp --source ping.ovh.net -j ACCEPT
```

```
/sbin/iptables -A INPUT -i eth0 -p icmp --source IP.250 -j ACCEPT # IP = aaa.bbb.ccc obtenue selon la règle précédente
```

```
/sbin/iptables -A INPUT -i eth0 -p icmp --source IP.249 -j ACCEPT # temporaire, seulement pour serveurs HG
```

```
/sbin/iptables -A INPUT -i eth0 -p icmp --source IP.251 -j ACCEPT # IP pour system de monitoring
```

Au niveau de SSH, si vous souhaitez restreindre l'accès seulement depuis votre IP, il vous est conseillé de laisser aussi cache.ovh.net. En effet, en cas de problème sur votre machine, nous pourrions intervenir dessus et vous dépanner. Si vous fermez le port 22 pour les techniciens d'OVH, nous ne pourrions vous aider si votre machine est bloquée.

Pour autoriser le SSH depuis nos serveurs, entrez la règle suivante :

```
/sbin/iptables -A INPUT -i eth0 -p tcp --dport 22 --source cache.ovh.net -j ACCEPT
```

Si vous avez un filer RAID, ne pas oublier d'autoriser les connexions NFS. Pour cela, nous pouvons être large en autorisant tout ce qui vient du réseau interne 192.168.0.0/16 :

```
/sbin/iptables -A INPUT -i eth0 -p tcp --source 192.168.0.0/16 -j ACCEPT
```

```
/sbin/iptables -A INPUT -i eth0 -p udp --source 192.168.0.0/16 -j ACCEPT
```

Si vous avez une configuration en Cluster, il faut également autoriser le port 79 pour que OCO puisse communiquer avec le répartiteur de charge.

```
/sbin/iptables -A INPUT -i eth0 -p tcp --dport 79 -j ACCEPT
```

Si votre serveur se trouve à Roubaix 3, il est nécessaire que vous récupériez votre serveur de monitoring comme pour un RPS :

Vous trouverez alors un serveur ayant un nom du type sla-rbx3-2.ovh.net que vous devez ajouter aux exceptions de votre firewall. Le tcpdump doit être fait sur l'ip fixe :

```
tcpdump host ip.fixe.du.serveur | grep ICMP
```

Pour un Serveur RPS :

L'interface monitoré par nos services est la eth0 (ou dummy0) les règles de firewall s'appliqueront sur celle-ci.

Si vous bloquez toutes les requêtes ping, même celles d'OVH, nous ne surveillerons plus le bon état de marche de votre serveur et s'il tombe nous n'en serons pas avertis. Pour autoriser le ping depuis nos serveurs, entrez les règles suivantes :

```
/sbin/iptables -A INPUT -i eth0 -p icmp --source proxy.ovh.net -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p icmp --source proxy.p19.ovh.net -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p icmp --source proxy.rbx.ovh.net -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p icmp --source proxy.rbx2.ovh.net -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p icmp --source ping.ovh.net -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p tcp --dport 22 --source cache.ovh.net -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p icmp --source IP.251 -j ACCEPT # IP pour system de monitoring RTM
/sbin/iptables -A INPUT -i eth0 -p icmp --source IP.250 -j ACCEPT # IP pour system de monitoring SLA
```

Pour récupérer la liste complète des serveurs de monitoring de votre rps, utilisez cette commande :

```
tcpdump host votre.ip.fail.over | grep ICMP
```

Ajoutez ensuite les ips ou les noms trouvés comme ci-dessus :

```
/sbin/iptables -A INPUT -i eth0 -p icmp --source ip.trouvée.avec.tcpdump -j ACCEPT
```

Il faudra également autoriser votre filer, pour le retrouver utiliser la commande :

```
r12xxx ~ # netstat -tanpu | grep iscsi
tcp 0 0 91.121.xx.xx:38632 91.121.191.16:3260 ESTABLISHED 3097/iscsid
```

L'IP de votre filer est donc : 91.121.191.16

La regle a ajouter sera :

```
/sbin/iptables -A INPUT -i eth0 -p tcp --source 91.121.191.16 -j ACCEPT
```

Exemple de configuration complète

Voici un exemple de script complet pour protéger votre serveur via iptables. Il est assez permissif dans le sens où la plupart des services présents sur votre machine sont accessibles, mais il peut servir de base pour votre propre configuration :

```
/sbin/iptables -A INPUT -i eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p tcp --dport 25 -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p tcp --dport 53 -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p udp --dport 53 -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p tcp --dport 80 -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p tcp --dport 110 -j ACCEPT
```

OVH

```
/sbin/iptables -A INPUT -i eth0 -p tcp --dport 443 -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p tcp --dport 10000 -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p tcp --dport 21 --source xx.xx.xx.xx -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p tcp --dport 22 --source cache.ovh.net -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p tcp --dport 22 --source xx.xx.xx.xx -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p icmp --source proxy.ovh.net -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p icmp --source proxy.p19.ovh.net -j ACCEPT

/sbin/iptables -A INPUT -i eth0 -p icmp --source proxy.rbx.ovh.net -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p icmp --source proxy.rbx2.ovh.net -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p icmp --source ping.ovh.net -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p icmp --source xxx.xxx.xxx.250 -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p icmp --source xxx.xxx.xxx.251 -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p tcp --source 192.168.0.0/16 -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p udp --source 192.168.0.0/16 -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -p tcp --dport 79 -j ACCEPT
/sbin/iptables -A INPUT -i eth0 -j REJECT
```

Dans ces règles, il faut bien sûr remplacer xx.xx.xx.xx par l'adresse IP de la machine dont vous vous servez pour accéder à votre serveur en FTP et SSH.

Automatiser le firewall

Une fois votre firewall parfaitement configuré, il ne vous reste plus qu'à créer un script qui se lancera à chaque démarrage de votre serveur. Voici un exemple à placer dans un fichier nommé par exemple "firewall" dans le répertoire /etc/init.d/ :

```
#!/bin/sh
# chkconfig: 3 21 91
# description: Firewall

IPT=/sbin/iptables

case "$1" in
start)
$IPT -F INPUT
$IPT -A INPUT -i eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPT -A INPUT -i eth0 -p tcp --dport 25 -j ACCEPT
$IPT -A INPUT -i eth0 -p tcp --dport 53 -j ACCEPT
$IPT -A INPUT -i eth0 -p udp --dport 53 -j ACCEPT
$IPT -A INPUT -i eth0 -p tcp --dport 80 -j ACCEPT
$IPT -A INPUT -i eth0 -p tcp --dport 110 -j ACCEPT
$IPT -A INPUT -i eth0 -p tcp --dport 443 -j ACCEPT
$IPT -A INPUT -i eth0 -p tcp --dport 10000 -j ACCEPT
$IPT -A INPUT -i eth0 -p tcp --dport 21 --source xx.xx.xx.xx -j ACCEPT
$IPT -A INPUT -i eth0 -p tcp --dport 22 --source cache.ovh.net -j ACCEPT
$IPT -A INPUT -i eth0 -p tcp --dport 22 --source xx.xx.xx.xx -j ACCEPT
$IPT -A INPUT -i eth0 -p icmp --source proxy.ovh.net -j ACCEPT
$IPT -A INPUT -i eth0 -p icmp --source proxy.p19.ovh.net -j ACCEPT
$IPT -A INPUT -i eth0 -p icmp --source proxy.rbx.ovh.net -j ACCEPT
$IPT -A INPUT -i eth0 -p icmp --source proxy.rbx2.ovh.net -j ACCEPT
$IPT -A INPUT -i eth0 -p icmp --source xxx.xxx.xxx.251 -j ACCEPT
$IPT -A INPUT -i eth0 -p icmp --source xxx.xxx.xxx.250 -j ACCEPT
```

```
$IPT -A INPUT -i eth0 -p icmp --source ping.ovh.net -j ACCEPT
$IPT -A INPUT -i eth0 -p tcp --source 192.168.0.0/16 -j ACCEPT
$IPT -A INPUT -i eth0 -p udp --source 192.168.0.0/16 -j ACCEPT
$IPT -A INPUT -i eth0 -p tcp --dport 79 -j ACCEPT
$IPT -A INPUT -i eth0 -j REJECT
exit 0
;;

stop)
$IPT -F INPUT
exit 0
;;
*)
echo "Usage: /etc/init.d/firewall {start|stop}"
exit 1
;;
esac
```

Donnez-lui les droits 700 et tapez `"/etc/init.d/firewall start"` pour le démarrer et `"/etc/init.d/firewall stop"` pour l'arrêter. Pour le lancer automatiquement au démarrage :

```
$ /sbin/chkconfig --level 3 firewall on
```

```
$ /sbin/chkconfig --level 06 firewall off
```

Vérifiez avant de mettre le script à chaque démarrage du serveur qu'il soit bon sinon votre serveur sera définitivement bloqué ! La communication entre le service RTM et votre serveur nécessite également que vous autorisez les connexions entrant sortant sur les ports UDP 6100 a 6200.