

Pare-feu IPtables sous Centos

Avant de commencer !

CentOs 7

- 1) Si vous êtes sur CentOS 7 il faut désactiver et désinstaller le parefeu déjà présent.

```
[root@localhost ~]# systemctl stop firewalld
[root@localhost ~]# yum remove firewalld
Modules complémentaires chargés : fastestmirror
Résolution des dépendances
--> Lancement de la transaction de test
--> Le paquet firewalld.noarch 0:0.5.3-5.el7 sera effacé
--> Résolution des dépendances terminée

Dépendances résolues

=====
Package                Architecture      Version           Dépôt             Taille
=====
Suppression :
firewalld               noarch            0.5.3-5.el7       @anaconda         1.8 M
=====

Résumé de la transaction
=====
Supprimer 1 Paquet

Taille d'installation : 1.8 M
Est-ce correct [o/N] : o
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Suppression : firewalld-0.5.3-5.el7.noarch                1/1
  Vérification : firewalld-0.5.3-5.el7.noarch                1/1

Supprimé :
firewalld.noarch 0:0.5.3-5.el7

Terminé !
[root@localhost ~]#
```

IpTables

- 1) Vérifier le présence de Iptable (il est normalement installer sur toutes les version d'origine)

```
[root@localhost ~]# rpm -q iptables
iptables-1.4.21-28.el7.x86_64
```

- 2) Vérifier que iptables service n'est pas installé (Nous l'installerons plus tard), sinon désinstaller le.

```
[root@localhost ~]# rpm -q iptables-services
le paquet iptables-services n'est pas installé
[root@localhost ~]#
```

Apprendre les bases

Afficher l'état du pare-feu (Les règles)

- 1) Pour afficher l'état général des pare-feu on peut faire la manip suivante

```
[root@localhost ~]# 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
```

- 2) Pour choisir la table à afficher Filter, Nat, Mangle (Etat général de la table)

```
[root@localhost ~]# iptables -t filter -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
[root@localhost ~]# iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target    prot opt source                destination

Chain INPUT (policy ACCEPT)
target    prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target    prot opt source                destination

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

- 3) Pour afficher seulement certaines règles

```
[root@localhost ~]# iptables -L INPUT
Chain INPUT (policy ACCEPT)
target    prot opt source                destination
[root@localhost ~]#
```

- 4) Cette commande est la plus fréquemment utilisée. Les paramètres -vn servent à afficher plus de détail (-v) et à transformer les adresses IP en numérique (-n)

```
[root@localhost ~]# iptables -L -vn
Chain INPUT (policy ACCEPT 63 packets, 5358 bytes)
 pkts bytes target    prot opt in     out     source                   destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source                   destination

Chain OUTPUT (policy ACCEPT 62 packets, 4606 bytes)
 pkts bytes target    prot opt in     out     source                   destination
```

- 5) Exemple de commande :

```
[root@localhost ~]# iptables -t nat -L -vn --line-numbers
Chain PREROUTING (policy ACCEPT 1 packets, 576 bytes)
num  pkts bytes target    prot opt in     out     source                   destination

Chain INPUT (policy ACCEPT 1 packets, 576 bytes)
num  pkts bytes target    prot opt in     out     source                   destination

Chain OUTPUT (policy ACCEPT 45 packets, 3347 bytes)
num  pkts bytes target    prot opt in     out     source                   destination

Chain POSTROUTING (policy ACCEPT 45 packets, 3347 bytes)
num  pkts bytes target    prot opt in     out     source                   destination
[root@localhost ~]#
```

Apprendre à définir des règles

- 1) Tout d'abord, il faut interdire toutes les connexions entrantes. On le voit bien, le ping vers notre localhost ne fonctionne plus. L'option -P Permet de définir une politique de sécurité par défaut.

```
[root@localhost ~]# iptables -D INPUT 1
[root@localhost ~]# iptables -P INPUT DROP
[root@localhost ~]# iptables -L -vn
Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source                   destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source                   destination

Chain OUTPUT (policy ACCEPT 1 packets, 68 bytes)
 pkts bytes target    prot opt in     out     source                   destination
[root@localhost ~]# ping localhost
PING localhost (127.0.0.1) 56(84) bytes of data.
^C
--- localhost ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2004ms
```

- 2) On va donc autorisé l'envoi de paquet entrant sur le réseau local de notre machine. Le paramètre `-A` permet d'ajouter une règle, `-i` permet de sélectionner l'interface réseau et `-j` spécifie la cible de règle. Maintenant le ping vers le localhost fonctionne de nouveaux.

```
[root@localhost ~]# iptables -A INPUT -i lo -j ACCEPT
[root@localhost ~]# iptables -L -vn
Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source               destination
  0      0 ACCEPT    all  --  lo     *       0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source               destination
[root@localhost ~]# ping localhost
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.043 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.065 ms
^C
--- localhost ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1014ms
rtt min/avg/max/mdev = 0.043/0.054/0.065/0.011 ms
[root@localhost ~]# _
```

- 3) Un autre exemple (que nous supprimerons par la suite) : Acceptons le ping lancé depuis une machine extérieur en ajoutant le protocole icmp.

```
[root@localhost ~]# iptables -A INPUT -p icmp -j ACCEPT
[root@localhost ~]# iptables -L INPUT --line-number
Chain INPUT (policy DROP)
num target prot opt source destination
1 ACCEPT all -- anywhere anywhere
2 ACCEPT icmp -- anywhere anywhere
[root@localhost ~]# iptables -D INPUT 2
```

- 4) En ajoutant le protocole icmp, il y a plein de paquet qui sont autorisé, ducoup on va supprimer le protocole icmp et ajouter seulement les paquet spécifique pour le ping.

```
[root@localhost ~]# iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
[root@localhost ~]# iptables -A INPUT -p icmp --icmp-type time-exceeded -j ACCEPT
[root@localhost ~]# iptables -A INPUT -p icmp --icmp-type destination-unreachable -j ACCEPT
[root@localhost ~]# iptables -L INPUT --line-number
Chain INPUT (policy DROP)
num target prot opt source destination
1 ACCEPT all -- anywhere anywhere
2 ACCEPT icmp -- anywhere anywhere icmp echo-request
3 ACCEPT icmp -- anywhere anywhere icmp time-exceeded
4 ACCEPT icmp -- anywhere anywhere icmp destination-unreachable
[root@localhost ~]#
```

Configurer le pare-feu iptables

Commençons par le début

- 1) Il faut arrêter le pare-feu et pour cela il suffit d'accepter toutes les connexions de toutes les tables sur tous les interfaces. La table Mangle contrôle les flux réseau et peut par exemple prioriser une connexion au confort de l'utilisateur. La table Nat sert de passerelle internet pour les paquets circulant sur le réseau.

```
[root@localhost ~]# iptables -P INPUT ACCEPT
[root@localhost ~]# iptables -P OUTPUT ACCEPT
[root@localhost ~]# iptables -P FORWARD ACCEPT
[root@localhost ~]#
[root@localhost ~]# iptables -t nat -P PREROUTING ACCEPT
[root@localhost ~]# iptables -t nat -P INPUT ACCEPT
[root@localhost ~]# iptables -t nat -P OUTPUT ACCEPT
[root@localhost ~]# iptables -t nat -P POSTROUTING ACCEPT
[root@localhost ~]#
[root@localhost ~]# iptables -t mangle -P POSTROUTING ACCEPT
[root@localhost ~]# iptables -t mangle -P PREROUTING ACCEPT
[root@localhost ~]# iptables -t mangle -P INPUT ACCEPT
[root@localhost ~]# iptables -t mangle -P OUTPUT ACCEPT
[root@localhost ~]# iptables -t mangle -P FORWARD ACCEPT
[root@localhost ~]#
```

- 2) Ensuite il faut remettre à zéro tous les compteurs de paquet et d'octet dans toutes les chaînes ; le paramètre -Z (--zero) sert à cela)

```
[root@localhost ~]# iptables -t mangle -Z
[root@localhost ~]# iptables -t filter -Z
[root@localhost ~]# iptables -t nat -Z
[root@localhost ~]#
```

- 3) Puis il faut supprimer toutes les règles sur toutes les tables ; Le paramètre -F (--flush) sert à supprimer toutes les chaînes et le paramètre -X (--delete-chain) sert à supprimer les règles créées par l'utilisateur. La table Filter filtre les paquets entrants, sortants et ceux qui sont relayés.

```
[root@localhost ~]# iptables -t nat -F
[root@localhost ~]# iptables -t nat -X
[root@localhost ~]# iptables -t filter -F
[root@localhost ~]# iptables -t filter -X
[root@localhost ~]# iptables -t mangle -F
[root@localhost ~]# iptables -t mangle -X
[root@localhost ~]#
```

Creation d'un script de configuration

- 1) Afin de ne pas perdre de temps, nous allons créer un script firewall.sh qui contiendra toute notre configuration. Entrez les commandes suivantes et fermé le fichier. (commande :set nu et :set nu ! pour afficher ou non les numéro de ligne dans vi)

```
#!/bin/sh
#
# firewall.sh

IPT=/usr/sbin/iptables

# Pour démarrer à zero accepter toute les connexions

$IPT -t filter -P INPUT ACCEPT
$IPT -t filter -P OUTPUT ACCEPT
$IPT -t filter -P FORWARD ACCEPT
$IPT -t nat -P PREROUTING ACCEPT
$IPT -t nat -P POSTROUTING ACCEPT
$IPT -t nat -P INPUT ACCEPT
$IPT -t nat -P OUTPUT ACCEPT
$IPT -t mangle -P PREROUTING ACCEPT
$IPT -t mangle -P POSTROUTING ACCEPT
$IPT -t mangle -P FORWARD ACCEPT
$IPT -t mangle -P INPUT ACCEPT
$IPT -t mangle -P OUTPUT ACCEPT

# Remettre les compteurs à zéro

$IPT -t filter -Z
$IPT -t nat -Z
$IPT -t mangle -Z

# Supprimer toutes les règles actives et les chaines personnalisées

$IPT -t filter -F
$IPT -t filter -X
$IPT -t nat -F
$IPT -t nat -X
$IPT -t mangle -F
$IPT -t mangle -X

[root@localhost ~]# vi /usr/local/sbin/firewall.sh_
```

```
# Remettre les compteurs à zéro

$IPT -t filter -Z
$IPT -t nat -Z
$IPT -t mangle -Z

# Supprimer toutes les règles actives et les chaînes personnalisées

$IPT -t filter -F
$IPT -t filter -X
$IPT -t nat -F
$IPT -t nat -X
$IPT -t mangle -F
$IPT -t mangle -X

# Politique par défaut

$IPT -P INPUT DROP
$IPT -P FORWARD DROP
$IPT -P OUTPUT ACCEPT

# Autoriser les connexions entrantes provenant du réseau local

$IPT -A INPUT -i lo -j ACCEPT

# Autoriser les ping provenant de l'extérieur

$IPT -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
$IPT -A INPUT -p icmp --icmp-type time-exceeded -j ACCEPT
$IPT -A INPUT -p icmp --icmp-type destination-unreachable -j ACCEPT
:wq!
```

- 2) Changer les droits de firewall.sh (700) puis exécuter le script.

```
[root@localhost ~]# chmod 700 /usr/local/sbin/firewall.sh
[root@localhost ~]# sh /usr/local/sbin/firewall.sh
[root@localhost ~]#
```

```
[root@localhost ~]# iptables -L -vn
Chain INPUT (policy DROP 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source               destination
    0     0 ACCEPT    all  --  lo     *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT    icmp  --  *      *       0.0.0.0/0            0.0.0.0/0            icmptype 8
    0     0 ACCEPT    icmp  --  *      *       0.0.0.0/0            0.0.0.0/0            icmptype 11
    0     0 ACCEPT    icmp  --  *      *       0.0.0.0/0            0.0.0.0/0            icmptype 3

Chain FORWARD (policy DROP 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 1 packets, 76 bytes)
  pkts bytes target    prot opt in     out     source               destination
[root@localhost ~]# _
```

Rendre le pare-feu persistant

- 1) Installer le paquet iptables-services puis activer et démarrer le service. Si vous n'arrivez pas à télécharger le paquet changer la politique des entrées par défaut.

```

--> Lancement de la transaction de test
---> Le paquet iptables-services.x86_64 0:1.4.21-28.el7 sera installé
--> Résolution des dépendances terminée

Dépendances résolues

=====
Package                        Architecture      Version           Dépôt             Taille
=====
Installation :
iptables-services             x86_64            1.4.21-28.el7     base              52 k
=====

Résumé de la transaction
=====
Installation    1 Paquet

Taille totale des téléchargements : 52 k
Taille d'installation : 26 k
Is this ok [y/d/N]: y
Downloading packages:
iptables-services-1.4.21-28.el7.x86_64.rpm                | 52 kB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installation : iptables-services-1.4.21-28.el7.x86_64          1/1
  Vérification : iptables-services-1.4.21-28.el7.x86_64          1/1

Installé :
iptables-services.x86_64 0:1.4.21-28.el7

Terminé !
[root@localhost ~]# systemctl enable iptables
Created symlink from /etc/systemd/system/basic.target.wants/iptables.service to /usr/lib/systemd/system/iptables.service.
[root@localhost ~]# systemctl start iptables
[root@localhost ~]# _

```

- 2) Le paquet installé précédemment ajoute ces propres règles en plus des nôtres. Relancez le script pour effacer ces règles puis sauvegardez les règles.


```

[root@localhost ~]# iptables -L -v;
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target     prot opt in     out     source            destination
  4    272 ACCEPT     all  --  *      *       0.0.0.0/0         0.0.0.0/0         state RELATED,ESTABLISHED
  0      0 ACCEPT     icmp --  *      *       0.0.0.0/0         0.0.0.0/0
  0      0 ACCEPT     all  --  lo     *       0.0.0.0/0         0.0.0.0/0
  0      0 ACCEPT     tcp  --  *      *       0.0.0.0/0         0.0.0.0/0         state NEW tcp dpt:22
  0      0 REJECT     all  --  *      *       0.0.0.0/0         0.0.0.0/0         reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target     prot opt in     out     source            destination
  0      0 REJECT     all  --  *      *       0.0.0.0/0         0.0.0.0/0         reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT 4 packets, 272 bytes)
  pkts bytes target     prot opt in     out     source            destination
[root@localhost ~]# sh /usr/local/sbin/firewall.sh
[root@localhost ~]# iptables -L -vn
Chain INPUT (policy DROP 0 packets, 0 bytes)
  pkts bytes target     prot opt in     out     source            destination
  0      0 ACCEPT     all  --  lo     *       0.0.0.0/0         0.0.0.0/0
  0      0 ACCEPT     icmp --  *      *       0.0.0.0/0         0.0.0.0/0         icmptype 8
  0      0 ACCEPT     icmp --  *      *       0.0.0.0/0         0.0.0.0/0         icmptype 11
  0      0 ACCEPT     icmp --  *      *       0.0.0.0/0         0.0.0.0/0         icmptype 3

Chain FORWARD (policy DROP 0 packets, 0 bytes)
  pkts bytes target     prot opt in     out     source            destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target     prot opt in     out     source            destination
[root@localhost ~]# service iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables: [ OK ]
[root@localhost ~]# _

```

Quelques règles et services de bases

1) Ajoutez les règles suivantes

```
# Autoriser les paquets entrants si la connexion est déjà établie
```

```
$IPT -A INPUT -m state --state ESTABLISHED -j ACCEPT
```

```
# Ouvrir le port 22 pour se connecter en SSH
```

```
$IPT -A INPUT -p tcp -i $IFACE --dport 22 -j ACCEPT
```

```
# Autoriser un serveur web
```

```
$IPT -A INPUT -p tcp -i $IFACE --dport 80 -j ACCEPT
```

```
# Autoriser le port 123 pour le serveur NTP
```

```
$IPT -A INPUT -p udp -i $IFACE --dport 123 -j ACCEPT
```

```
#Autoriser le serveur dnsmasq
```

```
$IPT -A INPUT -p tcp -i $IFACE --dport 53 -j ACCEPT
```

```
$IPT -A INPUT -p udp -i $IFACE --dport 53 -j ACCEPT
```

```
$IPT -A INPUT -p udp -i $IFACE --dport 67:68 -j ACCEPT
```

```
# Sauvegarder les règles
```

```
$SERVICE iptables save_
```

```
-- INSERT --
```

```
# Autoriser Serveur Mariadb
```

```
$IPT -A INPUT -p tcp -i $IFACE --dport 3306 -j ACCEPT
```

```
$IPT -A INPUT -p udp -i $IFACE --dport 3306 -j ACCEPT
```

2) Relancez le script.

```
$SERVICE iptables save
```

```
"/usr/local/sbin/firewall.sh" 83L, 1919C written
```

```
[root@localhost ~]# sh /usr/local/sbin/firewall.sh
```

```
iptables: Saving firewall rules to /etc/sysconfig/iptables:[ OK ]
```

```
[root@localhost ~]#
```

Enregistrer les paquets rejetés

1) Le chemin du fichier log est le suivant /var/log/messages. Afin d'éviter d'être submergé par les logs, on les limite à 2/min(2message similaire).

```
# Enregistrer les paquets refusés (Cette règle doit se placer après la dernière règle de filtrage)
```

```
$IPT -A INPUT -j LOG --log-prefix "++ IPv4 packet rejected ++"
```

```
$IPT -A INPUT -m limit --limit 2/min -j LOG \ --log-prefix "++ IPv4 packet rejected ++"
```

```
$IPT -A INPUT -j DROP # Après l'enregistrement toutes les autres connexions entrantes sont définitivement bloquées
```

```
# Sauvegarder les règles
```

```
:wq!_
```

Limiter l'accès à SSH

1) Limité les accès au connexion via SSH

```
# Ouvrir le port 22 pour se connecter en SSH
$IPT -A INPUT -p tcp -i $IFACE --dport 22 -j ACCEPT

# SSH limité en provenance de l'exterieur. Au bout de 3 echec il faudra attendre 60s avant de pouvoir
essayer de se reconnecter._

$IPT -A INPUT -p tcp -i $IFACE --dport 22 -m state \ --state NEW -m recent --set --name SSH
$IPT -A INPUT -p tcp -i $IFACE --dport 22 -m state \ --state NEW -m recent --update --seconds 60 --h
itcount 2 \ --rttl --name SSH -j DROP
$IPT -A INPUT -p tcp -i $IFACE --dport 22 -j ACCEPT

# Autoriser un serveur web
```

Tester le pare feu

1) Installer nmap

```
[root@localhost ~]# yum install nmap_
```

```

libpcap          x86_64          11:1.5.3-11.el7          base          138 k
nmap-ncat        x86_64          2:6.40-16.el7           base          206 k

Résumé de la transaction
=====
Installation      1 Paquet (+2 Paquets en dépendance)

Taille totale des téléchargements : 4.3 M
Taille d'installation : 17 M
Is this ok [y/d/N]: y
Downloading packages:
(1/3): libpcap-1.5.3-11.el7.x86_64.rpm          | 138 kB  00:00:00
(2/3): nmap-ncat-6.40-16.el7.x86_64.rpm        | 206 kB  00:00:00
(3/3): nmap-6.40-16.el7.x86_64.rpm            | 3.9 MB  00:00:01
-----
Total                                              3.4 MB/s | 4.3 MB  00:00:01
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installation : 14:libpcap-1.5.3-11.el7.x86_64          1/3
  Installation : 2:nmap-ncat-6.40-16.el7.x86_64        2/3
  Installation : 2:nmap-6.40-16.el7.x86_64             3/3
  Vérification : 14:libpcap-1.5.3-11.el7.x86_64        1/3
  Vérification : 2:nmap-ncat-6.40-16.el7.x86_64        2/3
  Vérification : 2:nmap-6.40-16.el7.x86_64             3/3

Installé :
  nmap.x86_64 2:6.40-16.el7

Dépendances installées :
  libpcap.x86_64 14:1.5.3-11.el7          nmap-ncat.x86_64 2:6.40-16.el7

Terminé !

```

- 2) Utiliser la commande suivante pour avoir un rapport (vu de l'extérieur sur les ports ouverts).
Etant donnée que je n'ai aucun service qui tourne je n'ai aucun port en écoute depuis l'extérieur

```

[root@Serveur_gcm ~]# nmap Serveur-gcm

Starting Nmap 6.40 ( http://nmap.org ) at 2019-08-01 20:37 CEST
Failed to resolve "Serveur-gcm".
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.08 seconds

```

- 3) La commande netstat est aussi utile pour traquer les ports en écoute non utilisé.

```

[root@Serveur_gcm ~]# netstat -untap
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale      Adresse distante     Etat      PID/Program name
tcp        0      0 0.0.0.0:22          0.0.0.0:*             LISTEN    3167/sshd
tcp        0      0 0.0.0.0:22          0.0.0.0:*             LISTEN    3167/sshd
tcp6       0      0 :::22              :::*                  LISTEN    3167/sshd
tcp6       0      0 :::1:25            :::*                  LISTEN    3547/master
udp        0      0 0.0.0.0:68         0.0.0.0:*             2913/dhclient
udp        0      0 0.0.0.0:68         0.0.0.0:*             2814/chronyd
udp6       0      0 :::1:323           :::*                  2814/chronyd

```

Sauvegarder et Restaurer les règles

- 1) Sauvegarder les règles dans un fichier

```
Using username "root".  
Last login: Wed Jan  8 20:35:33 2020  
[root@pc-171 ~]# iptables-save > /etc/firewall configuration
```

- 2) Appliquer les règles à chaque redémarrage de la machine avant de lancer les interfaces réseaux. Aller dans le fichier de lancement des interfaces et ajouter les 2 lignes ci-dessous.

```

[root@pc-171 ~]# vi /etc/init.d/network
#!/bin/bash
#
# network          Bring up/down networking
#
# chkconfig: 2345 10 90
# description: Activates/Deactivates all network interfaces configured to \
#              start at boot time.
#
### BEGIN INIT INFO
# Provides: $network
# Should-Start: iptables ip6tables NetworkManager-wait-online NetworkManager $network-pre
# Short-Description: Bring up/down networking
# Description: Bring up/down networking
### END INIT INFO

# Restaurer les règles du parefeu
/usr/sbin/iptables-restore < /etc/firewall_configuration

# Source function library.
. /etc/init.d/functions

if [ ! -f /etc/sysconfig/network ]; then
    exit 6
fi

. /etc/sysconfig/network

if [ -f /etc/sysconfig/pcmcia ]; then
    . /etc/sysconfig/pcmcia
fi

# Check that networking is up.
[ "${NETWORKING}" = "no" ] && exit 6

# if the ip configuration utility isn't around we can't function.
[ -x /sbin/ip ] || exit 1

```

Sources :

- <https://www.microlinux.fr/iptables/>
- <https://debian-facile.org/doc:reseau:iptables-pare-feu-pour-un-client>

Secci Mathieu