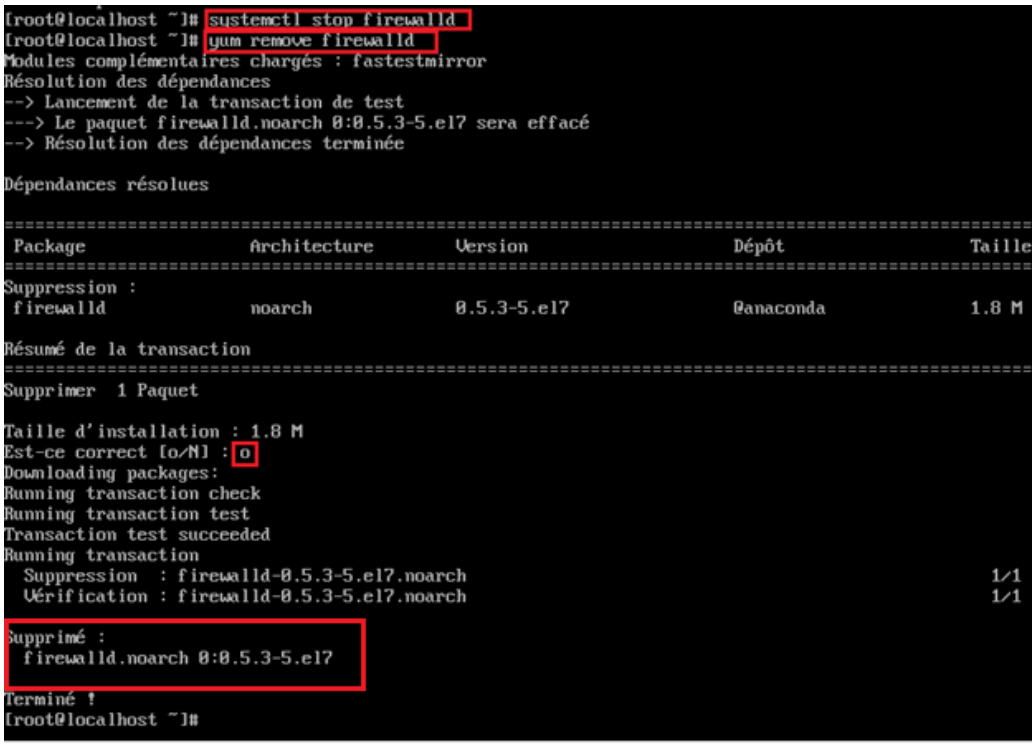
Pare-feu IPtables sous Centos

Avant de commencer !

CentOs 7

1) Si vous êtes sur CentOs 7 il faut desactiver et désinstaller le parefeu déjà présent.



IpTables

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

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'etat 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
```

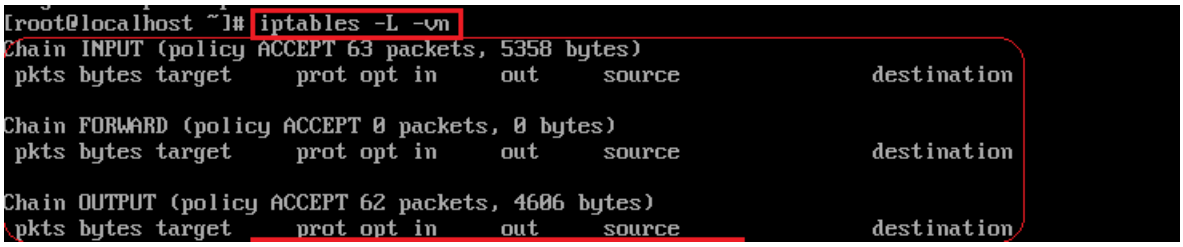
2) Pour choisir la table à afficher Filter, Nat, Mangle (Etat général de la table)

```
[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
```

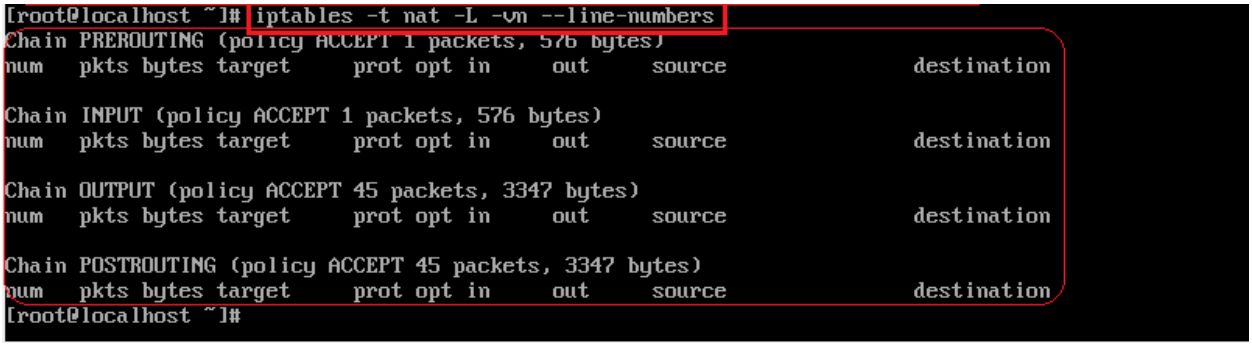
3) Pour afficher seulement certaines règles

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

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

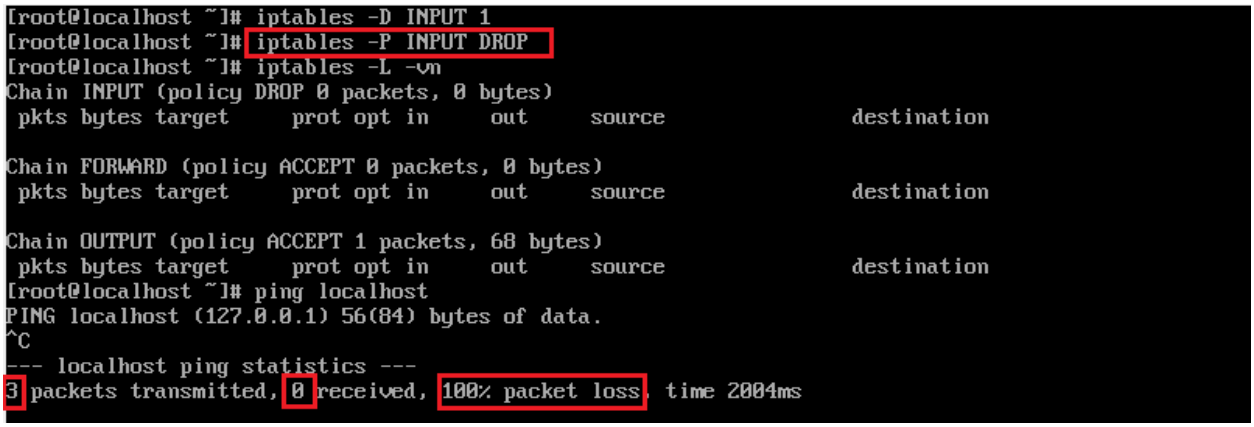


5) Exemple de commande :

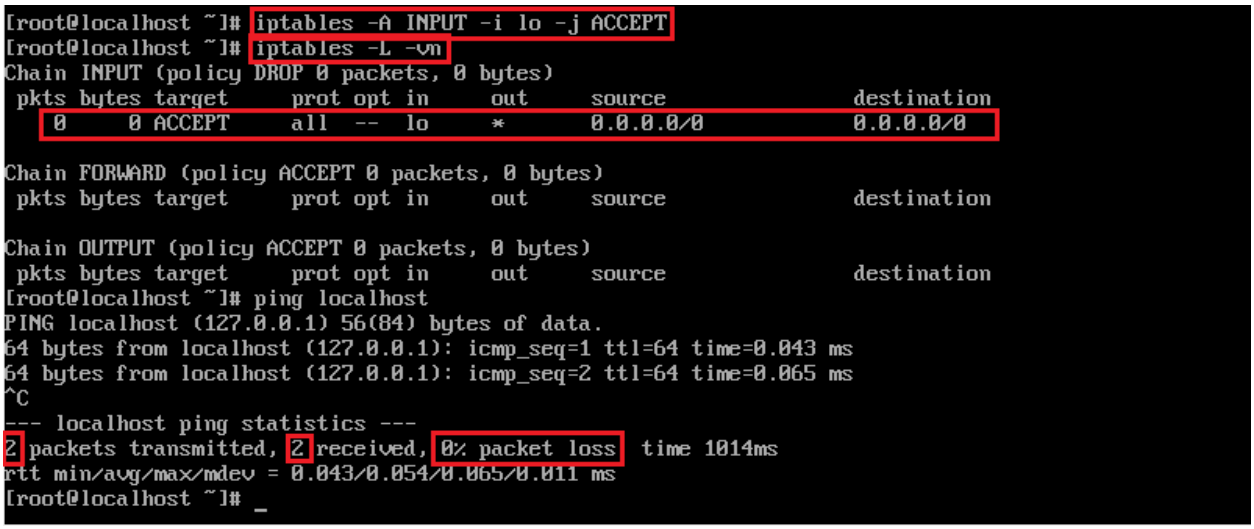


Apprendre à définir des règles

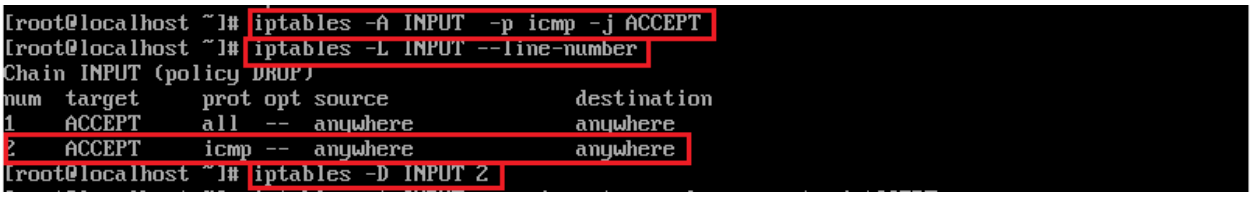
1) Tout d'abord, il faut interdire toutes les connexions entrantes. On le voit bien, le ping vers notre localhost ne fonctionne plus. L'option -P Permet de définir une politique de sécurité par défaut.



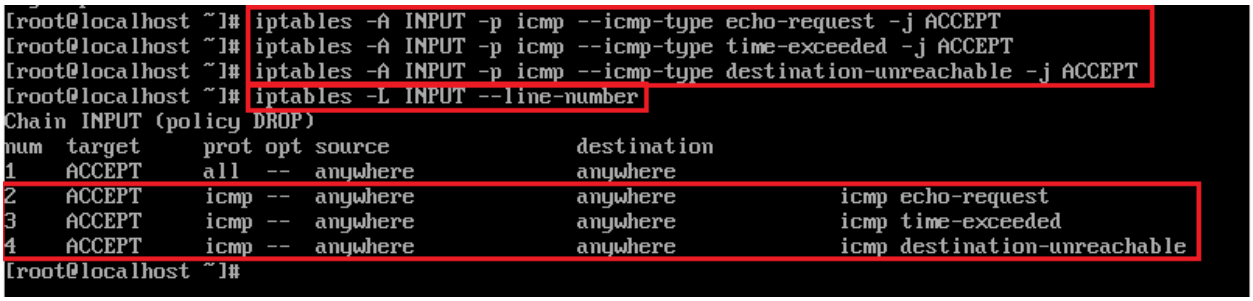
2) On va donc autorisé l'envoie de paquet entrant sur le réseau local de notre machine. Le paramètre –A permet d'ajouter une règle, -i permet de sélectionner l'interface réseau et -j spécifie la cible de règle. Maintenant le ping vers le localhost fonctionne de nouveaux.



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



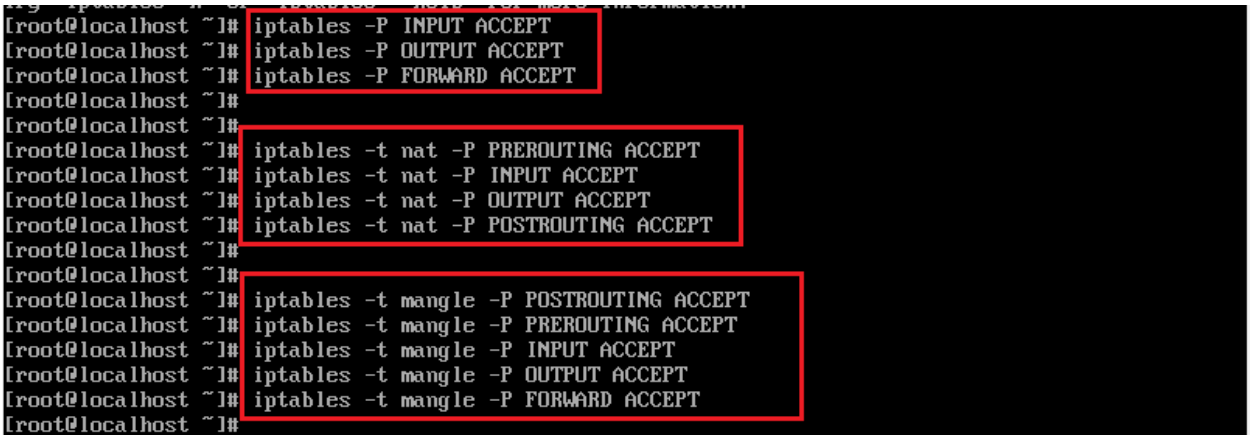
4) En ajoutant le protocol icmp, il y a plein de paquet qui sont autorisé, ducoup on va supprimer le protocol icmp et ajouter seulement les paquet spécifique pour le ping.



Configurer le pare-feu iptables

Commençons par le début

1) Il faut arreter 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 priorisé une connexion au confort de l'utilisateur. La table Nat sert de passerelle internet pour les paquets circulant sur le réseau.

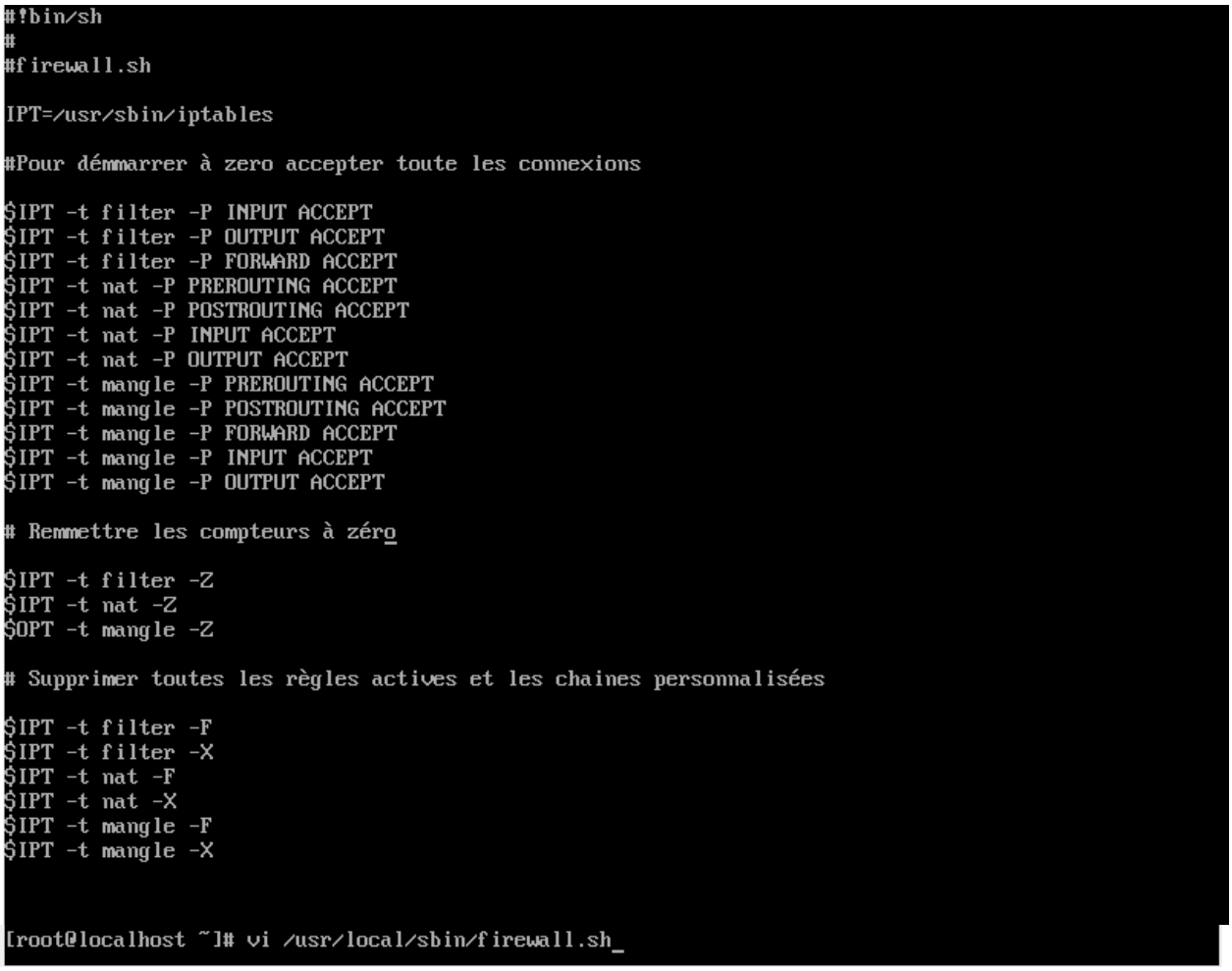


2) Ensuite il faut remettre à zéro tous les compteurs de paquet et d'octet dans toutes les chaines ; le paramètre -Z (--zero sert a cela)

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

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)



```
# Remmettre les compteurs à zéro

$IPT -t filter -Z
$IPT -t nat -Z
$OPT -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

# Politique par défaut

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

# Autorisé 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 ~]#
```

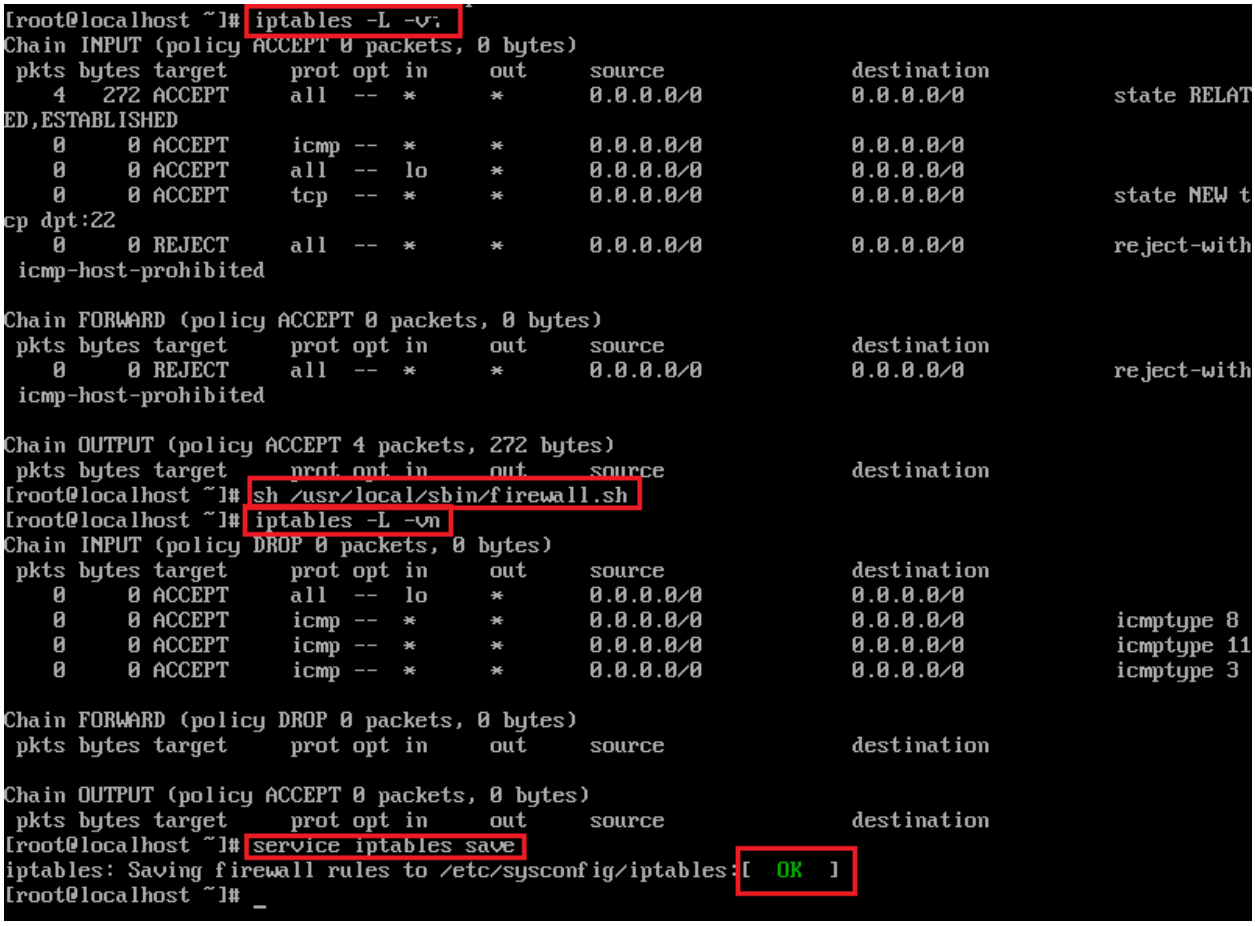
```
[root@localhost ~]# iptables -L -vn
Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0            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 paquets iptables-services puis activer et démarrer le services. Si vous n'arriver 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/sys
tem/iptables.service.
[root@localhost ~]# systemctl start iptables
[root@localhost ~]# _
```

2) Le paquet installé précedement ajoute ces propres règle en plus des notre. Relancez le script pour effacer ces règles puis savegardé les règles.



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'etre submergé par les logs, ont les limites a 2/min(2message similaire).

```
# Enregistrer les paquets refusés (Cette règle doit ce 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   # Apres l'enregistrement toutes les autres connexions entrantes sont définit
#ivement bloquées

# Sauvegarder les règles
:wq!_
```

Limiter l'accès à SSH

1) Limité les accès au connexion via SSH

Tester le pare feu

1) Installer nmap

```
 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 raport (vu de l'extérieur sur les ports ouverts). Etant donnée que je n'ai aucun service qui tourne je n'ai aucun port en ecoute depuis l'extérieur

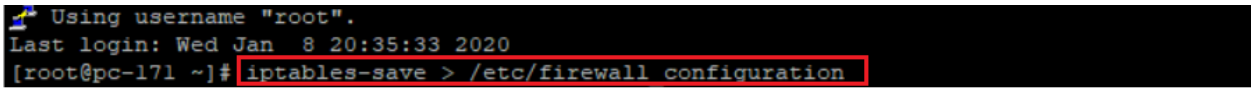
```
[root@Serveur_qcm ~]# nmap Serveur-qcm

Starting Nmap 6.40 ( http://nmap.org ) at 2019-08-01 20:37 CEST
Failed to resolve "Serveur-qcm".
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 ecoute non utilisé.

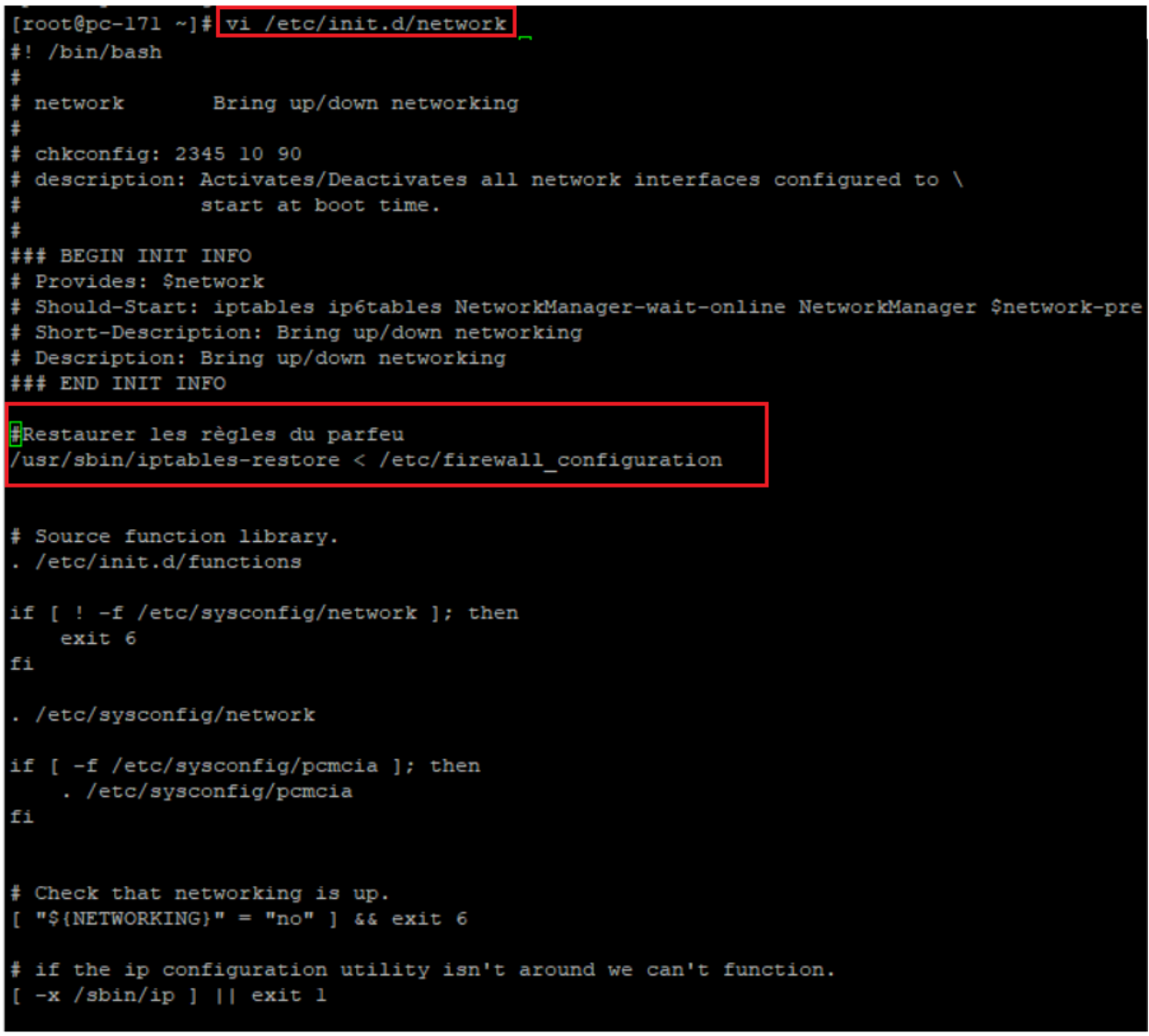
```
[root@Serveur_qcm ~]# netstat -untap
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale          Adresse distante        Etat       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      3167/sshd
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      3547/master
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      0 127.0.0.1:323           0.0.0.0:*                           2814/chronyd
udp6       0      0 ::1:323                 :::*                                2814/chronyd
[root@Serveur_qcm ~]# _
```

Sauvegarder et Restaurer les règles

1) Sauvegarder les règles dans un fichier



2) Appliquer les regles à chaque redemarrage de la machine avant de lancer les interfaces réseaux. Aller dans le fichier de lancement des interface 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 parfeu
/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