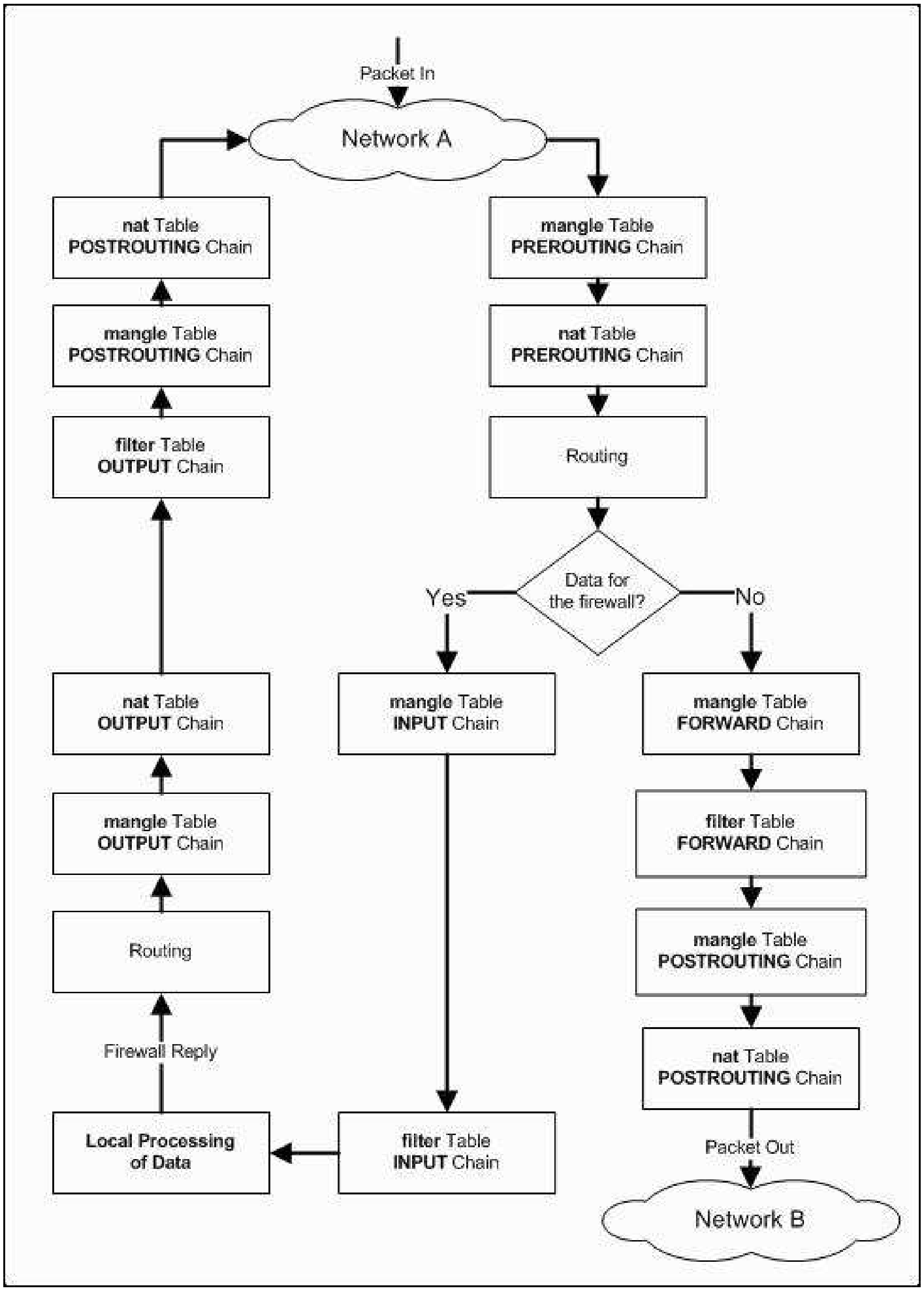
Firewall IPTABLES

Table des matières



Penser à regarder les logs :

```
# tail -f /var/log/syslog | ccze
# ccze -A </var/log/syslog | less -R
```

Version maison :

```
# tail -f /var/log/syslog | ./sed.sh
# cat /var/log/syslog | ./sed.sh | less -R
```

Fichier *sed.sh* :

```
sed "\
s/\(bad-in\)/`echo -e "\033[1;91m"`\1`echo -e "\033[0;39m"`/;\
s/\(SRC=\)/`echo -e "\033[0;91m"`\1`echo -e "\033[0;39m"`/;\
s/\(DPT=\)/`echo -e "\033[0;91m"`\1`echo -e "\033[0;39m"`/;\
"
```

1 Les tables

Elles correspondent à des classes de traitements que l'on peut effectuer sur les paquets IP. Lorsqu'on parametre netfilter, le choix de la table permet de déterminer le type d'action qu'on désire effectuer. Et c'est le choix de la chaine qui va déterminer le moment ou cela doit se passer. Il existe trois tables.

- **filter** : Accepter ou rejeter un paquet.
- **nat** : Modifier l'adresse/port source/destination d'un paquet.

2 Les chaînes

Lorsqu'un paquet IP est pris en charge par le noyau, il effectue un voyage dans les pilotes de cartes réseaux, dans la pile IP, voir, dans les applications qui le traitent. netfilter définit pour chaque table un certain nombre de points de passage pertinents où il est possible d'examiner un paquet, afin de le filtrer/modifier.

Ces points de passage, ce sont les chaines. Pour chaque table, il existe un certain nombre de chaines de base qui representent autant de points de passage :

- **PREROUTING** : Le paquet est pris en charge par l'interface réseau. Il s'apprete à être routé.
- **INPUT** : Le paquet est destiné à l'hôte sur lequel nous définissons les règles. Il nous est destiné.
- **FORWARD** : Le paquet ne nous est pas destiné, et nous sommes une passerelle.
- **OUTPUT** : Le paquet est émis par nous.
- **POSTROUTING** : Le paquet s'apprete à sortir par l'interface réseau.

Remarque : j'ai l'impression que lors du port forwarding le paquet passe dans la chaîne PREROUTING puis FORWARD.

3 Masquerade

Pour héberger un réseau privé *(192.168.2.0/24)* et lui ouvrir la passerelle *(eth1)* :

```
(192.168.2.0/24) ---eth2- [Routeur] -eth1--- (exterieur)

# iptables -t nat -A POSTROUTING -s 192.168.2.0/24 -o eth1 -j MASQUERADE
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

4 Routage

```
# iptables -Z -Lv | less
```

Mes règles : */etc/network/if-pre-up.d/firewall*

4.1 Règles communes

4.1.1 Variables

```
#!/bin/sh
test "$IFACE" = "eth2" || exit 0
echo "Lancement du firewall sur $IFACE"

# Mes variables
BAD_IFACE=eth2
DMZ_IFACE=eth1
DMZ_IFACE2=eth3
WIFI_IFACE=ath0
DMZ_ADDR=192.168.2.0/255.255.255.0
DMZ_ADDR2=192.168.3.0/255.255.255.0
WIFI_ADDR=192.168.1.0/255.255.255.0
ORDI_NARVAL=192.168.2.2
```

4.1.2 Purge

```
# Effacement de toutes les regles
iptables -F
iptables -t nat -F
iptables --delete-chain
iptables --table nat --delete-chain

iptables -N debug-and-drop
iptables -A debug-and-drop -j LOG --log-prefix "debug : "
iptables -A debug-and-drop -j DROP
```

4.1.3 icmp

```
iptables -N icmp-acc

# on accepte certains type de ping
iptables -A icmp-acc -p icmp --icmp-type destination-unreachable -j ACCEPT
iptables -A icmp-acc -p icmp --icmp-type source-quench -j ACCEPT
iptables -A icmp-acc -p icmp --icmp-type time-exceeded -j ACCEPT
iptables -A icmp-acc -p icmp --icmp-type echo-request -j ACCEPT
iptables -A icmp-acc -p icmp --icmp-type echo-reply -j ACCEPT

# On ignore tout le reste
iptables -A icmp-acc -j LOG --log-prefix "Icmp: "
iptables -A icmp-acc -j DROP
```

4.1.4 tcp

```
iptables -N tcp-acc

# Chaine pour logguer les paquets avant de les bloquer
iptables -N log-and-drop-tcp
iptables -A log-and-drop-tcp -j LOG --log-prefix "Tcp: "
iptables -A log-and-drop-tcp -j DROP

# On elimine les paquets ayant tous les flags TCP actives ansi que ceux
```

```
# avec aucun flag active (souvent utilise par les scans de Nmap)
iptables -A tcp-acc -p tcp --tcp-flags ALL ALL  -j log-and-drop-tcp
iptables -A tcp-acc -p tcp --tcp-flags ALL NONE -j log-and-drop-tcp

# On accepte les paquets appartenants a une connexion deja etablie
iptables -A tcp-acc -m state --state INVALID -j log-and-drop-tcp
iptables -A tcp-acc -m state --state RELATED,ESTABLISHED -j ACCEPT

# Rq: pour acceder au sites ftp, il faut ajouter les 2 modules suivants.
# Ensuite, les connexions distantes ftp sont geres (RELATED ci dessus)
# - ip_nat_ftp              2752  0
# - ip_conntrack_ftp       71568  1 ip_nat_ftp

# Rq: retour a l'appelant
```

4.2 Pre-routing

Ici le port forwarding va dérouter les paquets destinés à la chaîne INPUT et les envoyer sur la chaîne FORWARD.

```
# emule pour avoir un hight id
#iptables -t nat -A PREROUTING -p tcp --dport 4662 -j DNAT --to $ORDI_NARVAL
#iptables -t nat -A PREROUTING -p udp --dport 4672 -j DNAT --to $ORDI_NARVAL

# bittorent
iptables -t nat -A PREROUTING -p udp --dport 30163 -j DNAT --to $ORDI_NARVAL
iptables -t nat -A PREROUTING -p tcp --dport 6881:6889 -j DNAT --to $ORDI_NARVAL

# tunnel ssh
iptables -t nat -A PREROUTING -p tcp --dport 222 -j DNAT --to $ORDI_NARVAL:22

# freebox tv  ($ vlc http://mafreebox.freebox.fr/freeboxtv/playlist.m3u)
iptables -t nat -A PREROUTING -s 212.27.38.253 -j DNAT --to $ORDI_NARVAL
```

4.3 Input

```
iptables -N bad-in
iptables -N dmz-in

# On transmet les pings
iptables -A INPUT -p icmp -j icmp-acc

# On teste les flags tcp
iptables -A INPUT -p tcp -j tcp-acc

# Selon le sens d'arrivee des paquets on transmet a la chaine correspondante
iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT -i $BAD_IFACE -j  bad-in
iptables -A INPUT -i $DMZ_IFACE -j  dmz-in
iptables -A INPUT -i $DMZ_IFACE2 -j dmz-in
iptables -A INPUT -i $WIFI_IFACE -j dmz-in

# On ignore tout le reste
iptables -A INPUT -j LOG --log-prefix "in: "
iptables -A INPUT -j DROP
```

4.3.1 Exterieur vers serveur

C'est là que l'on voit beaucoup de scans.

```
# On accepte les services suivants :
iptables -A bad-in -p tcp --dport ssh -j ACCEPT
iptables -A bad-in -p tcp --dport www -j ACCEPT
iptables -A bad-in -p tcp --dport https -j ACCEPT
iptables -A bad-in -p udp --dport domain -j ACCEPT

iptables -A bad-in -p udp --sport domain -j ACCEPT
iptables -A bad-in -p udp --sport ntp -j ACCEPT

# On ignore tout le reste (sans le logguer car il y en a trop)
#iptables -A bad-in -j LOG --log-prefix "bad-in: "
iptables -A bad-in -j DROP
```

4.3.2 Intérieur vers serveur

```
# on accepte tout
iptables -A dmz-in -j ACCEPT
```

4.4 Forward

```
iptables -N bad-dmz
iptables -N dmz-bad
iptables -N dmz-dmz

# On transmet les pings
iptables -A FORWARD -p icmp -j icmp-acc

# On teste les flags tcp
iptables -A FORWARD -p tcp -j tcp-acc

# Selon le sens de transit des paquets on transmet a la chaine correspondante
iptables -A FORWARD -i $DMZ_IFACE  -o $BAD_IFACE -j dmz-bad
iptables -A FORWARD -i $DMZ_IFACE2 -o $BAD_IFACE -j dmz-bad
iptables -A FORWARD -i $WIFI_IFACE -o $BAD_IFACE -j dmz-bad
iptables -A FORWARD -i $DMZ_IFACE  -j dmz-dmz
iptables -A FORWARD -i $DMZ_IFACE2 -j dmz-dmz
iptables -A FORWARD -i $WIFI_IFACE -j dmz-dmz
iptables -A FORWARD -o $DMZ_IFACE  -j bad-dmz
iptables -A FORWARD -o $DMZ_IFACE2 -j bad-dmz
iptables -A FORWARD -o $WIFI_IFACE -j bad-dmz

# On ignore tout le reste
iptables -A FORWARD -j LOG --log-prefix "fwd: "
iptables -A FORWARD -j DROP
```

4.4.1 Extérieur vers intérieur

```
# Chaine pour logguer les paquets avant de les bloquer
iptables -N log-and-drop-bad-dmz
iptables -A log-and-drop-bad-dmz -j LOG --log-prefix "bad-dmz: "
iptables -A log-and-drop-bad-dmz -j DROP
```

```
# On bloque les paquets provenant des classes d'adresses reservees
# ainsi que le multicast
iptables -A bad-dmz -s 224.0.0.0/4    -j log-and-drop-bad-dmz
iptables -A bad-dmz -s 192.168.0.0/16 -j log-and-drop-bad-dmz
iptables -A bad-dmz -s 10.0.0.0/8     -j log-and-drop-bad-dmz

# ping google.fr
iptables -A bad-dmz -p udp --sport domain -j ACCEPT
iptables -A bad-dmz -p tcp --sport domain -j ACCEPT

# cf PREROUTING !!
iptables -A bad-dmz -p tcp --dport ssh -j ACCEPT
#iptables -A bad-dmz -p tcp --dport 4662 -j ACCEPT
#iptables -A bad-dmz -p udp --dport 4672 -j ACCEPT
iptables -A bad-dmz -p udp --dport 30163 -j ACCEPT
iptables -A bad-dmz -p tcp --dport 6881:6889 -j ACCEPT
iptables -A bad-dmz -s 212.27.38.253 -j ACCEPT

# On ignore tout le reste
iptables -A bad-dmz -j log-and-drop-bad-dmz
```

4.4.2 Intérieur vers extérieur

```
# on accepte tout
iptables -A dmz-bad -j ACCEPT
```

4.4.3 Intérieur vers Intérieur

```
# on accepte tout
iptables -A dmz-dmz -j ACCEPT
```

4.5 Output

```
# On laisse tout sortir du serveur
iptables -A OUTPUT -o lo -j ACCEPT
iptables -A OUTPUT -j ACCEPT
```

4.6 Post-routing

```
# Partage de la connection
iptables -t nat -A POSTROUTING -o $BAD_IFACE -j MASQUERADE
```

4.7 Kernel

```
# Attention, on active la passerelle qu'une fois toutes les regles
# de filtrage sont operationelles.
echo 1 > /proc/sys/net/ipv4/ip_forward
```