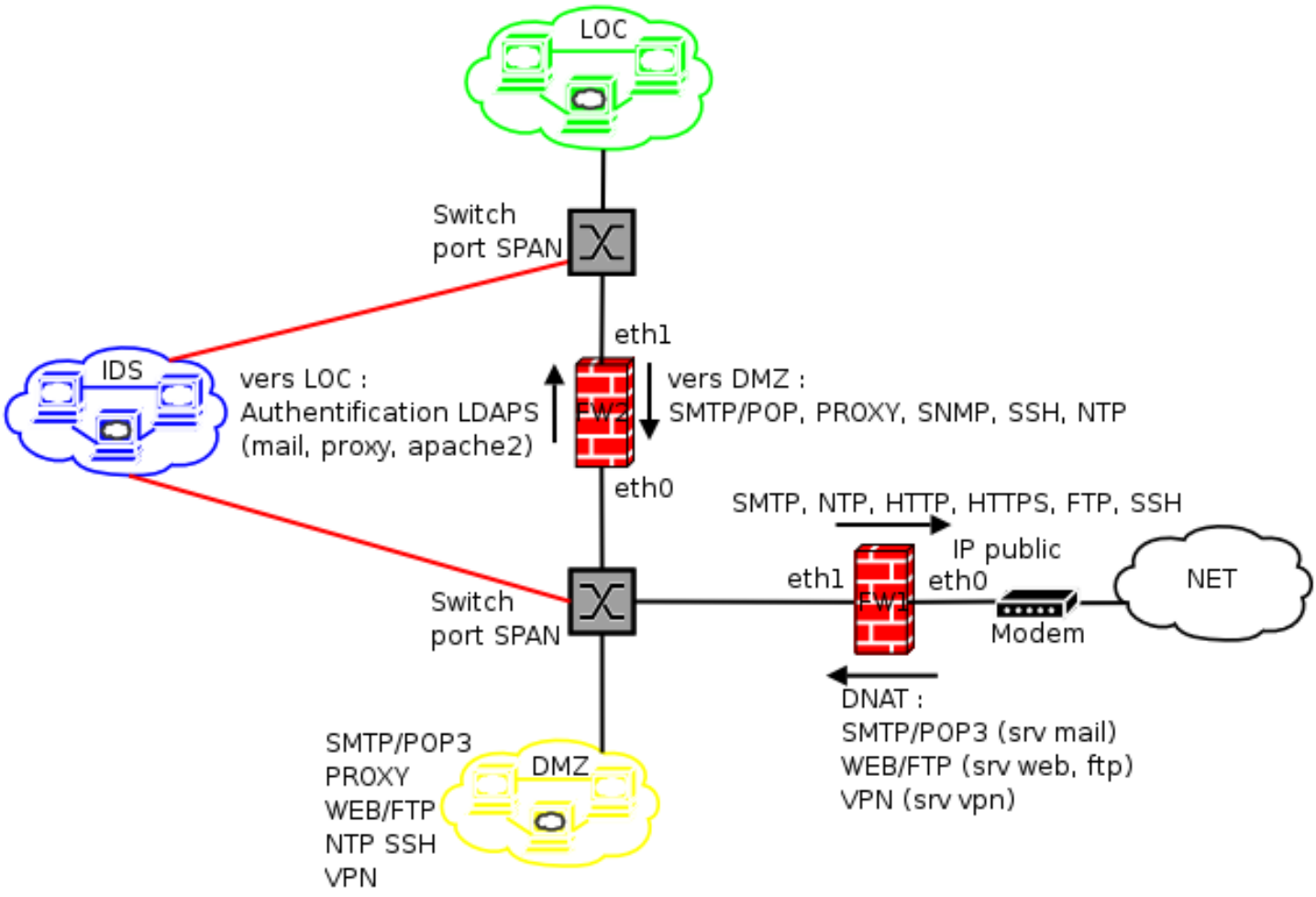
IPTables

Ce tutoriel développe la mise en place d'un Firewall avec IPTables.
La configuration proposée dans ce tutoriel est basée sur l'implantation réseau utilisée sur le Site, avec les règles suivantes :



Installation du package

Mandrake	debian
Sur les routeurs/firewall : iptables-1.2.9-5mdk	Sur les routeurs/firewall : iptables_1.2.11-10_i386.deb

Configuration du Firewall principal (n°1)

Créez un fichier fw1.dmz.alex.fr.sh par exemple et rendez le exécutable.

```
#!/bin/bash
########################################
# script IPTABLES - fw1.dmz.alex.fr.sh #
########################################

# Activation de l'interface eth1 si elle ne l'était pas.
ifconfig eth1 192.168.1.10 up

# Création de différentes variables.
# Interface connectée à Internet.
IFNET="eth0"
# Votre IP fixe.
IPNET="123.123.123.1"
# Interface connectée à la DMZ.
IFDMZ="eth1"

# Variables sur eth1 (DMZ).
IPAPACHE="192.168.1.1"
IPFTP="192.168.1.1"
IPNTPDMZ="192.168.1.2"
IPPOSTFIX="192.168.1.3"
IPPOP=$IPPOSTFIX
IPSQUID="192.168.1.4"
IPSSHDMZ="192.168.1.2"
IPVPN="192.168.1.5"

# Variable pour les réponses du client FTP.
FTPCLIENTPORTS="1024:65535"
# Variables des classes privées.
LOOPBACK="127.0.0.0/8"
CLASS_A="10.0.0.0/8"
CLASS_B="172.16.0.0/12"
CLASS_C="192.168.0.0/16"
CLASS_D_MULTICAST="224.0.0.0/4"
CLASS_E_RESERVED_NET="240.0.0.0/5"
BROADCAST="192.168.1.255"

# ---------------------------------------

# Configuration au niveau du Kernel.
# On ne veut pas de spoofing.
if [ -e /proc/sys/net/ipv4/conf/all/rp_filter ]
then
  for filtre in /proc/sys/net/ipv4/conf/*/rp_filter
  do
    echo 1 > $filtre
  done
fi

# IP forwarding entre les interfaces.
echo 1 > /proc/sys/net/ipv4/ip_forward

# Refuse les ping (icmp 8)
```

```
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
# Refuse les réponses aux  broadcasts.
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
# Refuse le routage des paquets source.
echo 0 > /proc/sys/net/ipv4/conf/all/accept_source_route
# Refuse les redirections ICMP.
echo 0 > /proc/sys/net/ipv4/conf/all/accept_redirects
# Active la protection contre les erreurs ICMP.
echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
# Log les paquets spoofés, le routage des paquets source et la
redirection des paquets.
echo 1 > /proc/sys/net/ipv4/conf/all/log_martians

# On charge les modules voulus.
modprobe ipt_tcpmss
modprobe ip_conntrack_ftp
modprobe ip_nat_ftp
modprobe iptable_filter
modprobe iptable_nat

# On vide toutes les règles avant d'appliquer les nouvelles.
iptables -F
iptables -X
iptables -Z

# On réinitialise les tables.
iptables -t nat -F
iptables -t nat -X
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P POSTROUTING ACCEPT
iptables -t nat -P OUTPUT ACCEPT
iptables -t mangle -P PREROUTING ACCEPT
iptables -t mangle -P OUTPUT ACCEPT
iptables -t filter -P OUTPUT ACCEPT
iptables -t filter -P FORWARD ACCEPT
iptables -t filter -P INPUT ACCEPT

# On rajoute des chaînes de log personnalisées.
#
# On DROP les paquets fragmentés.
iptables -A INPUT -i $IFNET -f -j LOG --log-prefix '[IPTABLES
DROP_FRAGMENTS] : '
iptables -A INPUT -i $IFNET -f -j DROP

iptables -N LOG_DROP_INPUT
iptables -A LOG_DROP_INPUT -j LOG --log-prefix '[IPTABLES DROP_INPUT] : '
iptables -A LOG_DROP_INPUT -j DROP

iptables -N LOG_DROP_OUTPUT
iptables -A LOG_DROP_OUTPUT -j LOG --log-prefix '[IPTABLES DROP_OUTPUT] :
'
iptables -A LOG_DROP_OUTPUT -j DROP

iptables -N LOG_DROP_FORWARD
iptables -A LOG_DROP_FORWARD -j LOG --log-prefix '[IPTABLES DROP_FORWARD]
: '
iptables -A LOG_DROP_FORWARD -j DROP
```

```
# Politiques par défaut, on DROP tout.
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# ----------------------------------------

# On accepte tout sur la boucle locale du FW.
iptables -A OUTPUT -o lo -j ACCEPT
iptables -A INPUT -i lo -j ACCEPT

# Re-écriture des adresses source (SNAT) du réseau LOC.
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -j MASQUERADE

# Les états de connexions :
# --state NEW
# Une nouvelle connexion est établie. (elle contient le flag SYN)
#
# --state ESTABLISHED
# La connexion analysée a déjà été établie, elle ne devrait pas contenir
de SYN ni de FIN.
#
# --state RELATED
# La connexion est en relation avec une autre connexion déjà établie.
#
# --state INVALID
# la connexion n'est pas conforme, elle contient un jeu de flags anormal.


# SYN-FLOODING PROTECTION
iptables -N syn-flood
iptables -A INPUT -i $IFNET -p tcp --syn -j syn-flood
iptables -A syn-flood -m limit --limit 1/s --limit-burst 4 -j RETURN
iptables -A syn-flood -j DROP

# Pour être sur que les nouvelles connexions TCP sont des paquets SYN.
iptables -A INPUT -i $IFNET -p tcp ! --syn -m state --state NEW -j DROP

# SPOOFING
# Refuse les paquets spoofés prétendant venir de notre IP.
iptables -A INPUT -i $IFNET -s $IPNET -j DROP
# Refuse les paquets venant des réseaux privés de classe A,B et C.
iptables -A INPUT -i $IFNET -s $CLASS_A -j DROP
iptables -A INPUT -i $IFNET -s $CLASS_B -j DROP
iptables -A INPUT -i $IFNET -s $CLASS_C -j DROP
# Refuse les paquets d'adresses multicast (classe D).
iptables -A INPUT -i $IFNET -s $CLASS_D_MULTICAST -j DROP
# Refuse les paquets d'adresses réservées de classe E.
iptables -A INPUT -i $IFNET -s $CLASS_E_RESERVED_NET -j DROP
# Refuse les paquets sur l'interface loopback.
iptables -A INPUT -i $IFNET -d $LOOPBACK -j DROP
# Refuse les paquets de broadcast.
iptables -A INPUT -i $IFDMZ -d $BROADCAST -j DROP

# On fait suivre la connexion SSH de la machine $IPSSHDMZ vers le NET.
# Les connexions SSH venant de LOC vers le Net ne sortent pas
```

```
directement,
# elles passent par une machine relais. Aucune connexions entre le FW2 et
# FW1, et inversement.
# Note : Il n'y a pas de client ou serveur SSH sur le FW principal, pour
# éviter toute prises de contrôle à distance.
iptables -A FORWARD -s $IPSSHDMZ -o $IFNET -p tcp --dport ssh -m state --
state ! INVALID -j ACCEPT
iptables -A FORWARD -d $IPSSHDMZ -i $IFNET -p tcp --sport ssh -m state --
state RELATED,ESTABLISHED -j ACCEPT

# On fait suivre les requêtes DNS venant de la DMZ vers le Net.
iptables -A FORWARD -i $IFDMZ -o $IFNET -p udp --dport domain -m state --
state ! INVALID -j ACCEPT
iptables -A FORWARD -i $IFNET -o $IFDMZ -p udp --sport domain -m state --
state RELATED,ESTABLISHED -j ACCEPT

# On fait suivre uniquement les demandes du PC Squid vers le Net en HTTP,
HTTPS et FTP.
iptables -A FORWARD -s $IPSQUID -o $IFNET -p tcp --dport http -m state --
state ! INVALID -j ACCEPT
iptables -A FORWARD -i $IFNET -d $IPSQUID -p tcp --sport http -m state --
state RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -s $IPSQUID -o $IFNET -p tcp --dport https -m state
--state ! INVALID -j ACCEPT
iptables -A FORWARD -i $IFNET -d $IPSQUID -p tcp --sport https -m state
--state RELATED,ESTABLISHED -j ACCEPT
#
iptables -A FORWARD -s $IPSQUID -o $IFNET -p tcp --dport ftp -m state --
state ! INVALID -j ACCEPT
iptables -A FORWARD -i $IFNET -d $IPSQUID -p tcp --sport ftp -m state --
state RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -s $IPSQUID -o $IFNET -p tcp --dport ftp-data -m
state --state ! INVALID -j ACCEPT
iptables -A FORWARD -i $IFNET -d $IPSQUID -p tcp --sport ftp-data -m
state --state RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -s $IPSQUID -o $IFNET -p tcp --sport $FTPCLIENTPORTS
--dport $FTPCLIENTPORTS -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i $IFNET -d $IPSQUID -p tcp --sport $FTPCLIENTPORTS
--dport $FTPCLIENTPORTS -m state --state RELATED,ESTABLISHED -j ACCEPT

# On fait suivre uniquement les demandes du PC Postfix vers le Net en
SMTP.
iptables -A FORWARD -s $IPPOSTFIX -o $IFNET -p tcp --dport smtp -m state
--state ! INVALID -j ACCEPT
iptables -A FORWARD -i $IFNET -d $IPPOSTFIX -p tcp --sport smtp -m state
--state RELATED,ESTABLISHED -j ACCEPT

# On fait suivre uniquement les demandes NTP du serveur de temps en DMZ.
iptables -A FORWARD -s $IPNTPDMZ -o $IFNET -p udp --dport ntp -m state --
state ! INVALID -j ACCEPT
iptables -A FORWARD -i $IFNET -d $IPNTPDMZ -p udp --sport ntp -m state --
state RELATED,ESTABLISHED -j ACCEPT


# DNAT SERVEURS DEBUT ------------------------------------
#
# On souhaite que toutes les requêtes provenant d'Internet
```

```
# arrivant sur l'adresse IP publique soient redirigées vers les
# serveurs concernés :

# Redirige vers le serveur Web de la DMZ.
iptables -t nat -A PREROUTING -d $IPNET -p tcp --dport http -j DNAT --to-
destination $IPAPACHE:80
iptables -t nat -A PREROUTING -d $IPNET -p tcp --dport https -j DNAT --
to-destination $IPAPACHE:443
# On fait suivre les ports HTTP et HTTPS vers le serveur Web de la DMZ.
iptables -A FORWARD -i $IFNET -d $IPAPACHE -p tcp --dport http -m state
--state ! INVALID -j ACCEPT
iptables -A FORWARD -s $IPAPACHE -o $IFNET -p tcp --sport http -m state
--state RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i $IFNET -d $IPAPACHE -p tcp --dport https -m state
--state ! INVALID -j ACCEPT
iptables -A FORWARD -s $IPAPACHE -o $IFNET -p tcp --sport https -m state
--state RELATED,ESTABLISHED -j ACCEPT

# Redirige vers le serveur de Messagerie de la zone Local.
iptables -t nat -A PREROUTING -d $IPNET -p tcp --dport smtp -j DNAT --to-
destination $IPPOSTFIX:25
iptables -t nat -A PREROUTING -d $IPNET -p tcp --dport pop3 -j DNAT --to-
destination $IPPOSTFIX:110
# On fait suivre les demandes SMTP et POP3.
iptables -A FORWARD -i $IFNET -d $IPPOSTFIX -p tcp --dport smtp -m state
--state ! INVALID -j ACCEPT
iptables -A FORWARD -s $IPPOSTFIX -o $IFNET -p tcp --sport smtp -m state
--state RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i $IFNET -d $IPPOSTFIX -p tcp --dport pop3 -m state
--state ! INVALID -j ACCEPT
iptables -A FORWARD -s $IPPOSTFIX -o $IFNET -p tcp --sport pop3 -m state
--state RELATED,ESTABLISHED -j ACCEPT

# Redirige vers le serveur FTP de la DMZ.
iptables -t nat -A PREROUTING -d $IPNET -p tcp --dport ftp -j DNAT --to-
destination $IPFTP:21
# On fait suivre les demandes FTP.
iptables -A FORWARD -i $IFNET -d $IPFTP -p tcp --dport ftp -m state --
state ! INVALID -j ACCEPT
iptables -A FORWARD -s $IPFTP -o $IFNET -p tcp --sport ftp -m state --
state RELATED,ESTABLISHED -j ACCEPT

# Redirige les connexions OpenVPN en UDP ports 1194 et 1195 vers le
serveur OpenVPN.
iptables -t nat -A PREROUTING -d $IPNET -p udp --dport 1194 -j DNAT --to-
destination $IPVPN:1194
iptables -t nat -A PREROUTING -d $IPNET -p udp --dport 1195 -j DNAT --to-
destination $IPVPN:1195
# On fait suivre les demandes OpenVPN.
iptables -A FORWARD -i $IFNET -d $IPVPN -p udp --dport 1194 -m state --
state ! INVALID -j ACCEPT
iptables -A FORWARD -s $IPVPN -o $IFNET -p udp --sport 1194 -m state --
state RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i $IFNET -d $IPVPN -p udp --dport 1195 -m state --
state ! INVALID -j ACCEPT
iptables -A FORWARD -s $IPVPN -o $IFNET -p udp --sport 1195 -m state --
state RELATED,ESTABLISHED -j ACCEPT
```

```
#
# DNAT SERVEURS FIN ---------------------------------------


# Toutes les autres règles sont refusées et logguées.
iptables -A FORWARD -j LOG_DROP_FORWARD
iptables -A INPUT -j LOG_DROP_INPUT
iptables -A OUTPUT -j LOG_DROP_OUTPUT

echo " [config iptables FW1 terminée] "
```

Configuration du Firewall secondaire (n°2)

Créez un fichier fw2.alex.fr.sh par exemple et rendez le exécutable.

```
#!/bin/bash
####################################
# script IPTABLES - fw2.alex.fr.sh #
####################################

# Création de différentes variables.
# Interface connectée à la DMZ.
IFDMZ="eth0"
# Interface connectée au réseau LOC.
IFLOC="eth1"

# Variables sur LOC.
IPADMIN="192.168.0.11"
IPNTPLOC="192.168.0.7"
IPSMNP="192.168.0.11"

# Variables sur DMZ.
IPNTPDMZ="192.168.1.2"
IPPOSTFIX="192.168.1.3"
IPPOP=$IPPOSTFIX
IPPROXY="192.168.1.4"
# Si vous utilisez Squid (3128) ou DansGuardian (8080)
PORTPROXY="3128"
IPSSHDMZ="192.168.1.2"

# ----------------------------------------

# Configuration au niveau du Kernel.
# On ne veut pas de spoofing.
if [ -e /proc/sys/net/ipv4/conf/all/rp_filter ]
then
  for filtre in /proc/sys/net/ipv4/conf/*/rp_filter
  do
    echo 1 > $filtre
  done
fi

# IP forwarding entre les interfaces.
echo 1 > /proc/sys/net/ipv4/ip_forward

# Refuse les ping (icmp 8)
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
# Refuse les réponses aux  broadcasts.
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
# Refuse le routage des paquets source.
echo 0 > /proc/sys/net/ipv4/conf/all/accept_source_route
# Refuse les redirections ICMP.
echo 0 > /proc/sys/net/ipv4/conf/all/accept_redirects
# Active la protection contre les erreurs ICMP.
echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
# Log les paquets spoofés, le routage des paquets source et la
redirection des paquets.
echo 1 > /proc/sys/net/ipv4/conf/all/log_martians
```

```
# On charge les modules voulus.
modprobe ipt_tcpmss
modprobe iptable_filter
modprobe iptable_nat

# On vide toutes les règles avant d'appliquer les nouvelles.
iptables -F
iptables -X
iptables -Z

# On réinitialise les tables.
iptables -t nat -F
iptables -t nat -X
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P POSTROUTING ACCEPT
iptables -t nat -P OUTPUT ACCEPT
iptables -t mangle -P PREROUTING ACCEPT
iptables -t mangle -P OUTPUT ACCEPT
iptables -t filter -P OUTPUT ACCEPT
iptables -t filter -P FORWARD ACCEPT
iptables -t filter -P INPUT ACCEPT

# On rajoute des chaînes de log personnalisées.
iptables -N LOG_DROP_INPUT
iptables -A LOG_DROP_INPUT -j LOG --log-prefix '[IPTABLES DROP_INPUT] : '
iptables -A LOG_DROP_INPUT -j DROP

iptables -N LOG_DROP_OUTPUT
iptables -A LOG_DROP_OUTPUT -j LOG --log-prefix '[IPTABLES DROP_OUTPUT] :
'
iptables -A LOG_DROP_OUTPUT -j DROP

iptables -N LOG_DROP_FORWARD
iptables -A LOG_DROP_FORWARD -j LOG --log-prefix '[IPTABLES DROP_FORWARD]
: '
iptables -A LOG_DROP_FORWARD -j DROP

# Politiques par défaut, on DROP tout.
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# ----------------------------------------

# On accepte tout sur la boucle locale du FW.
iptables -A OUTPUT -o lo -j ACCEPT
iptables -A INPUT -i lo -j ACCEPT

# Re-écriture des adresses source (SNAT) du réseau LOC.
iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -j MASQUERADE

# On fait suivre la connexion SSH de la machine $IPADMIN vers la machine
$IPSSHDMZ.
iptables -A FORWARD -s $IPADMIN -d $IPSSHDMZ -p tcp --dport ssh -m state
--state ! INVALID -j ACCEPT
iptables -A FORWARD -s $IPSSHDMZ -d $IPADMIN -p tcp --sport ssh -m state
```

```
--state RELATED,ESTABLISHED -j ACCEPT

# On accepte la connexion SSH de la machine $IPADMIN vers le FW2.
iptables -A INPUT -s $IPADMIN -p tcp --dport ssh -j ACCEPT
iptables -A OUTPUT -d $IPADMIN -p tcp --sport ssh -j ACCEPT

# On fait suivre les demandes au Proxy venant de LOC vers le PC Squid en
DMZ.
# Note : Nous n'avons pas besoin relayer les requêtes DNS car tout passe
par le proxy.
iptables -A FORWARD -i $IFLOC -d $IPPROXY -p tcp --dport $PORTPROXY -m
state --state ! INVALID -j ACCEPT
iptables -A FORWARD -s $IPPROXY -o $IFLOC -p tcp --sport $PORTPROXY -m
state --state RELATED,ESTABLISHED -j ACCEPT

# On fait suivre les demandes SMTP et POP3 venant de LOC vers le PC
Postfix en DMZ.
iptables -A FORWARD -i $IFLOC -d $IPPOSTFIX -p tcp --dport smtp -m state
--state ! INVALID -j ACCEPT
iptables -A FORWARD -s $IPPOSTFIX -o $IFLOC -p tcp --sport smtp -m state
--state RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i $IFLOC -d $IPPOP -p tcp --dport pop3 -m state --
state ! INVALID -j ACCEPT
iptables -A FORWARD -s $IPPOP -o $IFLOC -p tcp --sport pop3 -m state --
state RELATED,ESTABLISHED -j ACCEPT

# On fait suivre les requêtes NTP du serveur de temps de LOC vers le PC
NTP en DMZ.
iptables -A FORWARD -s $IPNTPLOC -d $IPNTPDMZ -p udp --dport ntp -m state
--state ! INVALID -j ACCEPT
iptables -A FORWARD -s $IPNTPDMZ -d $IPNTPLOC -p udp --sport ntp -m state
--state RELATED,ESTABLISHED -j ACCEPT

# On fait suivre les demandes LDAPS venant de la DMZ vers LOC.
iptables -A FORWARD -i $IFDMZ -o $IFLOC -p tcp --dport ldaps -m state --
state ! INVALID -j ACCEPT
iptables -A FORWARD -i $IFLOC -o $IFDMZ -p tcp --sport ldaps -m state --
state RELATED,ESTABLISHED -j ACCEPT

# On fait suivre les requêtes SNMP de supervision venant du PC $IPSMNP
vers le serveur NTP de la DMZ.
iptables -A FORWARD -s $IPSMNP -d $IPNTPDMZ -p udp --dport snmp -m state
--state ! INVALID -j ACCEPT
iptables -A FORWARD -s $IPNTPDMZ -d $IPSMNP -p udp --sport snmp -m state
--state RELATED,ESTABLISHED -j ACCEPT


# Toutes les autres règles sont refusées et logguées.
iptables -A FORWARD -j LOG_DROP_FORWARD
iptables -A INPUT -j LOG_DROP_INPUT
iptables -A OUTPUT -j LOG_DROP_OUTPUT

echo " [config iptables FW2 terminée] "
```

Sur les routeurs/firewall clients OpenVPN

```
#!/bin/bash
########################################
# script IPTABLES - firewall clients VPN#
########################################

# Variables contenant le nom de vos interfaces.
IFNET="eth0"
IFLOC="eth1"

# Variables des classes privées.
LOOPBACK="127.0.0.0/8"
CLASS_A="10.0.0.0/8"
CLASS_B="172.16.0.0/12"
CLASS_C="192.168.0.0/16"
CLASS_D_MULTICAST="224.0.0.0/4"
CLASS_E_RESERVED_NET="240.0.0.0/5"
BROADCAST="192.168.5.255"

# ---------------------------------------

# Configuration au niveau du Kernel.
# On ne veut pas de spoofing.
if [ -e /proc/sys/net/ipv4/conf/all/rp_filter ]
then
  for filtre in /proc/sys/net/ipv4/conf/*/rp_filter
  do
    echo 1 > $filtre
  done
fi

# IP forwarding entre les interfaces.
echo 1 > /proc/sys/net/ipv4/ip_forward

# Refuse les ping (icmp 8)
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
# Refuse les réponses aux  broadcasts.
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
# Refuse le routage des paquets source.
echo 0 > /proc/sys/net/ipv4/conf/all/accept_source_route
# Refuse les redirections ICMP.
echo 0 > /proc/sys/net/ipv4/conf/all/accept_redirects
# Active la protection contre les erreurs ICMP.
echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
# Log les paquets spoofés, le routage des paquets source et la
redirection des paquets.
echo 1 > /proc/sys/net/ipv4/conf/all/log_martians

# On charge les modules voulus.
modprobe ipt_tcpmss
modprobe ip_conntrack_ftp
modprobe ip_nat_ftp
modprobe iptable_filter
modprobe iptable_nat

# On vide toutes les règles avant d'appliquer les nouvelles.
```

```
iptables -F
iptables -X
iptables -Z

# On réinitialise les tables.
iptables -t nat -F
iptables -t nat -X
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P POSTROUTING ACCEPT
iptables -t nat -P OUTPUT ACCEPT
iptables -t mangle -P PREROUTING ACCEPT
iptables -t mangle -P OUTPUT ACCEPT
iptables -t filter -P OUTPUT ACCEPT
iptables -t filter -P FORWARD ACCEPT
iptables -t filter -P INPUT ACCEPT

# On rajoute des chaines de log personnalisées.
iptables -N LOG_DROP_INPUT
iptables -A LOG_DROP_INPUT -j LOG --log-prefix '[IPTABLES DROP_INPUT] : '
iptables -A LOG_DROP_INPUT -j DROP

iptables -N LOG_DROP_OUTPUT
iptables -A LOG_DROP_OUTPUT -j LOG --log-prefix '[IPTABLES DROP_OUTPUT] :
'
iptables -A LOG_DROP_OUTPUT -j DROP

iptables -N LOG_DROP_FORWARD
iptables -A LOG_DROP_FORWARD -j LOG --log-prefix '[IPTABLES DROP_FORWARD]
: '
iptables -A LOG_DROP_FORWARD -j DROP

# Politiques par défaut, on DROP tout.
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# ----------------------------------------

# On accepte tout sur la boucle locale du FW.
iptables -A OUTPUT -o lo -j ACCEPT
iptables -A INPUT -i lo -j ACCEPT

# On accepte tout sur l'interface réseau LOC.
iptables -A OUTPUT -o $IFLOC -j ACCEPT
iptables -A INPUT -i $IFLOC -j ACCEPT

# Re-écriture des adresses source (SNAT) du réseau LOC.
iptables -t nat -A POSTROUTING -s 192.168.5.0/24 -j MASQUERADE


# ---- OpenVPN ----
# On accepte tout les paquets pour les interfaces TUN0 ou TAP0.
iptables -A INPUT -i tun0 -j ACCEPT
iptables -A OUTPUT -o tun0 -j ACCEPT
iptables -A FORWARD -i tun0 -j ACCEPT
#iptables -A INPUT -i tap0 -j ACCEPT
#iptables -A OUTPUT -o tap0 -j ACCEPT
```

```
#iptables -A FORWARD -i tap0 -j ACCEPT
#
# On accepte tout les paquets OpenVPN en UDP sur les ports 1194 ou 1195.
iptables -A INPUT -p udp --dport 1194 -j ACCEPT
iptables -A OUTPUT -p udp --dport 1194 -j ACCEPT
#iptables -A INPUT -p udp --dport 1195 -j ACCEPT
#iptables -A OUTPUT -p udp --dport 1195 -j ACCEPT
#
# On fait suivre toutes les requêtes de Local vers les tunnels TUN0 ou
TAP0.
iptables -A FORWARD -i $IFLOC -o tun0 -j ACCEPT
iptables -A FORWARD -o $IFLOC -i tun0 -j ACCEPT
#iptables -A FORWARD -i $IFLOC -o tap0 -j ACCEPT
#iptables -A FORWARD -o $IFLOC -i tap0 -j ACCEPT
# -----------------

#
# Ajoutez vos propres règles de connexions ...
#
# On fait suivre les requêtes DNS de Local vers le Net.
iptables -A FORWARD -i $IFLOC -o $IFNET -p udp --dport domain -m state --
state ! INVALID -j ACCEPT
iptables -A FORWARD -o $IFLOC -i $IFNET -p udp --sport domain -m state --
state RELATED,ESTABLISHED -j ACCEPT
#
# Ajoutez vos propres règles sur les requêtes (Web, FTP, ...)
#

# Toutes les autres règles sont refusées et logguées.
iptables -A FORWARD -j LOG_DROP_FORWARD
iptables -A INPUT -j LOG_DROP_INPUT
iptables -A OUTPUT -j LOG_DROP_OUTPUT

echo " [config iptables FW terminée] "
```

Autres commandes IPTables

Affiche vos chaînes vides :
```
[root@fw user]# iptables -L -n
```

Affiche les régles de NAT :
```
[root@fw user]# iptables -t nat -n -L
```

Affiche toutes les chaînes :
```
[root@fw user]# iptables -L -n -v
```

La commande iptables-save envoie sur l'écran le contenu des chaînes de toutes les tables dans un format relativement lisible :
```
[root@fw user]# iptables-save
```

Cette commande a un autre avantage.
En dirigeant sa sortie vers un fichier, vous obtenez un fichier de configuration qui sera exploitable par un autre script : `iptables-restore`
```
[root@fw root]# iptables-save > maconfig.iptables
```

Vous pourrez restaurer intégralement votre configuration depuis ce fichier :
```
[root@fw root]# iptable-restore < maconfig.iptables
```

Quand vous avez déterminé que votre configuration est bonne :
```
[root@fw user]# /etc/init.d/iptables save
Sauvegarde des règles courantes dans /etc/sysconfig/iptables : [ OK ]
```

Au prochain démarrage du PC vos règles IPTables seront automatiquement prises en compte.

Pour restaurer l'état des tables au moyen du fichier /etc/sysconfig/iptables :
```
[root@fw user]# /etc/init.d/iptables start (ou restart)
Application des règles "iptables" du pare-feu : [ OK ]
```

Pour vérouiller votre machine (plus rien ne passe) :
```
[root@fw user]# /etc/init.d/iptables panic
Passage de la politique des cibles à DROP : [ OK ]
```

Pour vider les règles IPTables

Pour effacer toutes vos règles active (flush IPTables) :

```
[root@fw user]# /etc/init.d/iptables stop
Retour à la police par défaut 'ACCEPT' pour les règles incluses : [OK]
```

Ou créez un fichier flushIPtables.sh par exemple

```
#!/bin/bash
##################
# flush IPTABLES #
##################

iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT
iptables -t nat -P OUTPUT ACCEPT
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P POSTROUTING ACCEPT
iptables -t mangle -P PREROUTING ACCEPT
iptables -t mangle -P OUTPUT ACCEPT
iptables -t filter -P OUTPUT ACCEPT
iptables -t filter -P FORWARD ACCEPT
iptables -t filter -P INPUT ACCEPT
iptables -F
iptables -t nat -F
iptables -X
iptables -t nat -X

echo 0 > /proc/sys/net/ipv4/ip_forward

echo " [flush iptables terminé]"
```

Sources :

http://lea-linux.org/reseau/

http://christian.caleca.free.fr/netfilter/iptables.htm

Document mis à jour : 11/12/05