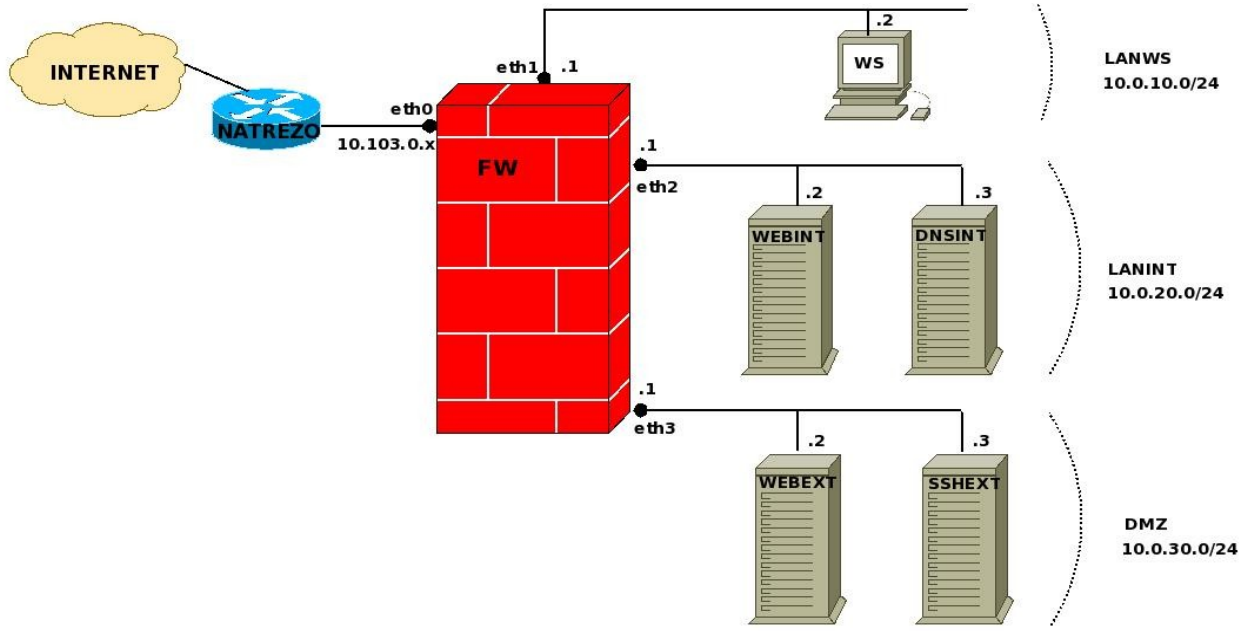
TP – FIREWALLING

Exercice 1: Firewall stateless

- Préparez la maquette suivante:



. 6 disques durs à cloner à partir de la distribution 'maison' minimale.
. 256 Mo Ram par machine suffit.

. N'oubliez pas d'activer IP_FORWARDING sur FW.

. Réseau interne 'lanws' (eth1 de FW et eth0 de WS)
. Réseau interne 'lanint' (eth2 de FW, eth0 de WEBINT et eth0 de DNSINT)
. Réseau interne 'dmz' (eth3 de FW, eth0 de WEBEXT et eth0 de SSHEXT)
. Réseau bridgé (eth0 de FW sur eth0.103)

. Configuration IP (particularités)
Passerelle sur FW → `10.103.0.1`
Passerelle sur WS → `10.0.10.1`
Passerelle sur WEBINT et DNSINT → `10.0.20.1`
Passerelle sur WEBEXT et SSHEXT → `10.0.30.1`

. Nating complet sur FW afin de pouvoir installer vos packages (il sera Enlevé par les futures règles du pare-feu).

- Sur WEBINT et WEBEXT, installez un serveur Apache écoutant sur le port 80 et présentant la page d'accueil par défaut lors d'une requête http sur ceux-ci.

```
# yum install httpd -y
# service httpd start
# chkconfig --level 3 httpd on
```

httpd.conf de WEBINT	httpd.conf de WEBEXT
...	...
ServerName 10.0.20.2	ServerName 10.0.30.2
...	...

→ plus d'info : voir chapitre 'Apache'

- Sur DNSINT, installez un serveur DNS de cache écoutant sur le port 53 et dont les forwarders sont les deux DNS du FAI.

```
# yum install bind bind-libs bind-utils -y
```

named.conf (reconstiué à partir de /etc/named.conf et /etc/named.rfc192.zones)

```
...
```

Attention :
- aux interfaces d'écoute
- aux forwarders
- à l'allow-query
- rappels : on a besoin que des zones ".", "localhost" et "1.0.0.127.in-addr.arpa"

resolv.conf

```
nameserver 127.0.0.1

# service named start (+ voir les logs)
# chkconfig --level 3 httpd on
# netstat -tunl
```

Vérifications :

```
# nslookup localhost
# nslookup 127.0.0.1
# nslookup www.helha.be

# rndc flush
```

→ plus d'info : voir chapitre 'DNS'

Sur toutes les autres machines :

```
# yum install bind-utils

resolv.conf
nameserver 10.0.20.3

# nslookup www.helha.be
...
Server 10.0.20.3
...
Name : www.helha.be
Address : 193.190.66.12
```

- Sur SSHEXT, installez un serveur SSH écoutant sur le port 22 et acceptant les connexions ssh par mot de passe (par défaut, il existe déjà...).

- Sur FW, installez un serveur SSH écoutant sur le port 22 et acceptant les connexions ssh par mot de passe (par défaut, il existe déjà...).

Remarque

Les commandes suivantes devraient fonctionner à partir de WS

```
# lynx 10.0.20.2
# lynx 10.0.30.2
```

- Sur FW, configurez un pare-feu répondant aux spécificités suivantes:

. Politique du "Tout est fermé" : par défaut le pare-feu bloque tout.

. Les services locaux sur celui-ci doivent pouvoir communiquer entre eux.

. Il autorise toute requête dns venant de n'importe quelle machine d'un des réseaux privés pour être résolue par le serveur de cache (DNSINT).

. L'accès sécurisé (ssh) sur FW est autorisé uniquement à partir des machines du réseau LANINT.

. Le serveur web de WEBINT ne sera accessible que par les machines du réseau LANWS.

. Les services spécifiques de la DMZ ne seront accessibles que de l'extérieur.

```
#!/bin/bash
#Demarrage automatique du firewall
# chkconfig: 3 98 99
# → a condition d'avoir ajouter le lien du demon dans le runlevel de demarrage
#                       → chkconfig --add fwless
#
#                            OU
#
# Ajoutez "/etc/init.d/fwless start" dans /etc/rc.local
# pour demarrer le firewall automatiquement au lancement de la machine
#
# DEBUG: iptables -L
#     iptables -L -v
#         iptables -S
#
#
################################################################################
#                          VARIABLES                                           #
################################################################################
lanws=10.0.10.0/24
lanint=10.0.20.0/24
dmz=10.0.30.0/24
ipext=10.103.0.x
dnsfai1=109.88.203.3
dnsfai2=62.197.111.140
webext=10.0.30.2
sshext=10.0.30.3
webint=10.0.20.2
dnsint=10.0.20.3

#########################################################################
#            ARRET (TOUT OUVERT) MEME L'ACCES AUX SERVICES INTERNES        #
#########################################################################
stop()
{
# On nettoye toutes les regles des tables filter et nat
iptables -F
iptables -X # Pour les regles utilisateurs
iptables -t nat -F
iptables -t nat -X # Pour les regles utilisateurs

# On ouvre tout
for chaine in INPUT OUTPUT FORWARD
do
  iptables -P $chaine ACCEPT
done

for chaine in PREROUTING POSTROUTING OUTPUT
do
  iptables -t nat -P $chaine ACCEPT
done

# Et meme les cibles -j MASQUERADE et -j DNAT
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT
                                              --to-destination $webext:80
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 22 -j DNAT
                                                    --to-destination $sshext:22

echo "[Done.]"
}
```

```
#############################################################################
#     DEMARRAGE (TOUT FERME) SAUF LES ACCES AUTORISES POUR L'EXERCICE     #
#############################################################################

start()
{
# On nettoye toutes les regles des tables filter et nat
#########################################################
iptables -F
iptables -X # Pour les regles utilisateurs
iptables -t nat -F
iptables -t nat -X # Pour les regles utilisateurs

# On ferme tout (pas de DROP sur nat)
#####################################
for chaine in INPUT OUTPUT FORWARD
do
  iptables -P $chaine DROP
done

# Les services locaux doivent pouvoir communiquer entre eux
############################################################
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

# On ouvre la translation d'adresse (on aurait pu travailler avec MASQUERADE)
# (Sinon il n'y aurait pas de nating et l'accès à l'extérieur de l'intérieur
#  serait tout simplement impossible)
#############################################################################
###iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
###  ou
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to $ipext

# Ouvre la resolution dns vers dnsint pour tous les hosts des lans
##################################################################
for lan in $lanws $lanint $dmz
do
  # Lan vers dnsint
  iptables -A FORWARD -p udp --dport 53 -s $lan -d $dnsint -j ACCEPT
  iptables -A FORWARD -p udp --sport 53 -s $dnsint -d $lan -j ACCEPT
done

for dnsfai in $dnsfai1 $dnsfai2
do
  # dnsint vers dnsfai
  iptables -A FORWARD -i eth2 -o eth0 -p udp --dport 53 -s $dnsint -d $dnsfai -j ACCEPT
  iptables -A FORWARD -i eth0 -o eth2 -p udp --sport 53 -s $dnsfai -d $dnsint -j ACCEPT
done

# Autoriser l'acces ssh sur fw des hosts de lanint
##################################################
iptables -A INPUT -i eth2 -s $lanint -p tcp --dport 22 -j ACCEPT
iptables -A OUTPUT -o eth2 -d $lanint -p tcp --sport 22 -j ACCEPT

# Autoriser l'acces http a webint uniquement a partir des hosts de lanws
########################################################################
iptables -A FORWARD -i eth1 -o eth2 -p tcp --dport 80 --sport 1024: -s $lanws
                                                                    -d $webint -j ACCEPT
iptables -A FORWARD -i eth2 -o eth1 -p tcp --dport 1024: --sport 80 -s $webint
                                                                     -d $lanws -j ACCEPT
```

```
# Autoriser l'acces aux services web et ssh sur les hosts de la dmz a partir du web
#################################################################################
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT
                                                        --to-destination $webext:80
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 22 -j DNAT
                                                        --to-destination $sshext:22
# Ne pas oublier le forward car il est DROP.
# En effet, il faut autoriser celui-ci apres un PREROUTING
iptables -A FORWARD -i eth0 -o eth3 -p tcp --dport 80 -j ACCEPT
iptables -A FORWARD -i eth3 -o eth0 -p tcp --sport 80 -j ACCEPT
iptables -A FORWARD -i eth0 -o eth3 -p tcp --dport 22 -j ACCEPT
iptables -A FORWARD -i eth3 -o eth0 -p tcp --sport 22 -j ACCEPT

echo "[Done.]"
}

########################################################
#       GESTION DU PASSAGE DE PARAMETRE AU SCRIPT      #
########################################################
case "$1" in
     start)             echo "Firewall is starting ..."
                        start;;
     stop)              echo "Firewall is stopping ... "
                        stop;;
     restart)           echo "Firewall is stopping ... "
                        stop
                        echo "Firewall is starting ..."
                        start;;
     status)            iptables -L
                        iptables -t nat -L;;
     *)                 echo "Usage: $0 {start|stop|restart|status}";;
esac
```

Exercice 2: Firewall statefull

Idem 'Exercice 1' mais en rendant le pare-feu le plus 'statefull' possible.

```
#!/bin/bash
#Demarrage automatique du firewall
# chkconfig: 3 98 99
# --> a condition d'avoir ajouter le lien du demon dans le runlevel de demarrage
#                        --> chkconfig --add fwfull
#
#                              OU
#
# Ajoutez "/etc/init.d/fwfull start" dans /etc/rc.local
# pour demarrer le firewall automatiquement au lancement de la machine
#
# DEBUG: iptables -L
#        iptables -L -v
#        iptables -S
#
################################################################################
#                           VARIABLES                                          #
################################################################################
lanws=10.0.10.0/24
lanint=10.0.20.0/24
dmz=10.0.30.0/24
ipext=10.103.0.x
dnsfai1=109.88.203.3
dnsfai2=62.197.111.140
webext=10.0.30.2
sshext=10.0.30.3
webint=10.0.20.2
dnsint=10.0.20.3

################################################################################
#             ARRET (TOUT OUVERT) MEME L'ACCES AUX SERVICES INTERNES         #
################################################################################

stop()
{
# On nettoye toutes les regles des tables filter et nat
iptables -F
iptables -X # Pour les regles utilisateurs
iptables -t nat -F
iptables -t nat -X # Pour les regles utilisateurs

# On ouvre tout
for chaine in INPUT OUTPUT FORWARD
do
  iptables -P $chaine ACCEPT
done

for chaine in PREROUTING POSTROUTING OUTPUT
do
  iptables -t nat -P $chaine ACCEPT
done

# Et meme les cibles -j MASQUERADE et -j DNAT
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT
                                                --to-destination $webext:80
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 22 -j DNAT
                                                --to-destination $sshext:22

echo "[Done.]"

}
```

```
#############################################################################
#     DEMARRAGE (TOUT FERME) SAUF LES ACCES AUTORISES POUR L'EXERCICE     #
#############################################################################

start()
{
# On nettoye toutes les regles des tables filter et nat
############################################################
iptables -F
iptables -X # Pour les regles utilisateurs
iptables -t nat -F
iptables -t nat -X # Pour les regles utilisateurs

# On ferme tout (pas de DROP sur nat)
#####################################
for chaine in INPUT OUTPUT FORWARD
do
  iptables -P $chaine DROP
done

# Les services locaux doivent pouvoir communiquer entre eux
############################################################
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

# On ouvre la translation d'adresse (on aurait pu travailler avec MASQUERADE)
#############################################################################
#iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to $ipext

# Ouvre la resolution dns vers dnsint pour tous les hosts des lans
##################################################################
for lan in $lanws $lanint $dmz
do
  # Lan vers dnsint
  iptables -A FORWARD -p udp --sport 1024:65535 --dport 53 -s $lan -d $dnsint
                                                  -m state --state NEW,ESTABLISHED -j ACCEPT
  iptables -A FORWARD -p udp --sport 53 --dport 1024:65535 -s $dnsint -d $lan
                                                  -m state --state ESTABLISHED -j ACCEPT
done

for dnsfai in $dnsfai1 $dnsfai2
do
  iptables -A FORWARD -i eth2 -o eth0 -p udp --sport 1024:65535 --dport 53
                         -s $dnsint -d $dnsfai -m state --state NEW,ESTABLISHED   -j ACCEPT
  iptables -A FORWARD -i eth0 -o eth2 -p udp --sport 53 --dport 1024:65535
                         -s $dnsfai -d $dnsint -m state --state ESTABLISHED -j ACCEPT
done

# Autoriser l'acces ssh sur fw des hosts de lanint
##################################################
iptables -A INPUT -i eth2 -s $lanint -p tcp --dport 22
                                                  -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth2 -d $lanint -p tcp --sport 22
                                                  -m state --state ESTABLISHED -j ACCEPT

# Autoriser l'acces http a webint uniquement a partir des hosts de lanws
########################################################################
iptables -A FORWARD -i eth1 -o eth2 -p tcp --dport 80 --sport 1024: -s $lanws -d $webint
                                                  -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i eth2 -o eth1 -p tcp --dport 1024: --sport 80 -s $webint -d $lanws
                                                  -m state --state ESTABLISHED -j ACCEPT

# Autoriser l'acces aux services web et ssh sur les hosts de lanint a partir de l'inter-
net
```

```
#############################################################################################
#
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT
                                                    --to-destination $webext:80
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 22 -j DNAT
                                                    --to-destination $sshext:22
# Ne pas oublier le forward car il est DROP.
# En effet,il faut autoriser celui-ci apres un PREROUTING
iptables -A FORWARD -i eth0 -o eth3 -p tcp --dport 80
                                                    -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i eth3 -o eth0 -p tcp --sport 80
                                                    -m state --state ESTABLISHED -j ACCEPT

iptables -A FORWARD -i eth0 -o eth3 -p tcp --dport 22
                                                    -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i eth3 -o eth0 -p tcp --sport 22
                                                    -m state --state ESTABLISHED -j ACCEPT

echo "[Done.]"
}

#########################################################
#       GESTION DU PASSAGE DE PARAMETRE AU SCRIPT     #
#########################################################

case "$1" in
start) echo "Firewall is starting ..."
       start;;

stop) echo "Firewall is stopping ... "
      stop;;

restart) echo "Firewall is stopping ... "
       stop
         echo "Firewall is starting ..."
         start;;

status) iptables -L
        iptables -t nat -L;;

*) echo "Usage: $0 {start|stop|restart|status}";;
esac
```