

PTI : Netfilter

Filtrage de paquets à l'aide de Netfilter (IPTABLES)

COMPETENCES ABORDEES

C21.....	Installer et configurer un microordinateur
C22.....	Installer et configurer un réseau
C23.....	Installer et configurer un dispositif de sécurité
C31.....	Assurer les fonctions de base de l'administration d'un réseau
C32.....	Assurer les fonctions de l'exploitation
C34.....	Surveiller et optimiser le trafic sur le réseau

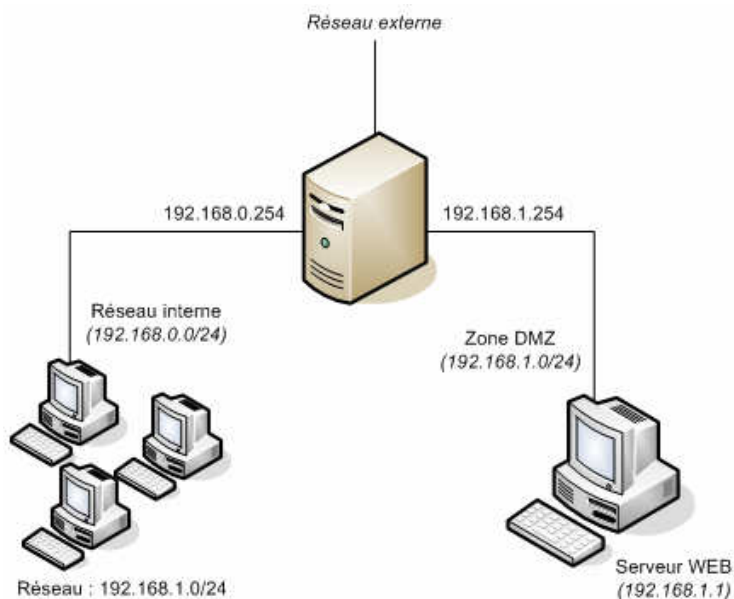
OBJECTIFS

Une société possédant un serveur Web désire le mettre aussi bien à disposition de ses employés, qu'aux intervenants extérieurs (*clients et fournisseurs de la société*). Pour cela, il a été décidé de placer le serveur Web dans une zone 'DMZ'. Ainsi, grâce à la mise en place d'un système de filtrage, le réseau interne est parfaitement isolé de l'extérieur, laissant le serveur Web toutefois accessible.

MISE EN PLACE

Etat du réseau :

- Systemes d'exploitation :
 - GNU/Linux Fedora Core 2 ;
 - GNU/Linux Slackware 10 ;
 - Microsoft Windows XP Professional.
- Caractéristiques matérielles :
 - Ethernet 10/100Mbps.
- Logiciels utilisés :
 - Apache 1.3.31 ;
 - Netfilter 1.2.11.



CONFIGURATIONS DE BASE

Configuration des interfaces du poste routeur :

- Attribution des adresses IP :

eth0 (172.16.6.50)

interface vers le réseau externe ;

eth1 (192.168.1.254)

interface vers la zone 'DMZ' ;

eth2 (192.168.0.254)

interface vers le réseau interne.

```
> ifconfig ethX XXX.XXX.XXX.XXX netmask XXX.XXX.XXX.XXX
```

- Passerelle par défaut du routeur :

```
> route add default gw 172.16.1.254
```

- Activation du routage :

Il est nécessaire d'activer le routage entre les différentes interfaces du poste routeur afin de permettre les futures communications et l'utilisation des règles de filtrage.

```
> echo 1 > /proc/sys/net/ipv4/ip_forward
```

Configuration des interfaces des autres postes :

- Attribution des adresses IP :

```
192.168.1.1      serveur Web de la zone 'DMZ' (DeMilitarized Zone) ;
192.168.0.1     poste client interne au réseau ;
172.16.6.70    poste client externe au réseau.
```

```
> ifconfig ethx xxx.xxx.xxx.xxx netmask xxx.xxx.xxx.xxx
```

- Passerelles par défaut :

```
> route add default gw 192.168.x.254
```

MISE EN PLACE DU FILTRAGE

Chargement des modules :

Pour pouvoir utiliser **netfilter**, le chargement de certains modules est nécessaire lorsque celui-ci n'a pas été compilé dans le noyau de Linux :

```
> modprobe iptables
> modprobe iptable_filter
> modprobe iptable_nat
> modprobe ipt_MASQUERADE
```

iptables	<i>module principal ;</i>
iptable_filter	<i>module de filtrage de paquets ;</i>
iptable_nat, ipt_MASQUERADE	<i>module pour la translation d'adresses ;</i>

Création d'un alias :

Afin d'obtenir un affichage plus clair et plus rapide des règles de filtrage, nous établissons un alias.

```
> alias clip="clear && iptables -v -L --line-numbers"
```

-v	<i>affichage d'informations supplémentaires ;</i>
-L	<i>listage des règles de filtrage ;</i>
--line-numbers	<i>affichage du numéro des lignes.</i>

Mise en place des règles :

- Règles par défaut :

Désormais, nous pouvons définir des règles de filtrage. Tout d'abord il est nécessaire de définir les stratégies par défaut du filtre, celles qui seront appliquées à tous les paquets pour lesquels aucune règle n'aura pu être mise en correspondance.

```
> iptables -P INPUT DROP
> iptables -P OUTPUT DROP
> iptables -P FORWARD DROP
```

- Translation d'adresse :

Ensuite, il faut activer la translation d'adresse NAT (*Network Address Translation*) pour le réseau interne et la zone 'DMZ'.

```
> iptables -A POSTROUTING -t nat -s 192.168.0.0/23 -o eth0 -j MASQUERADE
```

- Autorisation des requêtes DNS :

Pour accéder aux services DNS (*Domain Name System*) de l'extérieur, il est nécessaire d'autoriser les postes des réseaux internes à effectuer des requêtes et à recevoir les résultats de celles-ci.

```
> iptables -A FORWARD -p udp -s 212.180.1.79 --sport 53 -i eth0 -j ACCEPT
> iptables -A FORWARD -p udp -d 212.180.1.79 --dport 53 -o eth0 -j ACCEPT
> iptables -A FORWARD -p udp -s 212.180.0.137 --sport 53 -i eth0 -j ACCEPT
> iptables -A FORWARD -p udp -d 212.180.0.137 --dport 53 -o eth0 -j ACCEPT
```

- Autorisation du trafic HTTP(S) et FTP :

```
> iptables -A FORWARD -p tcp -m multiport -s 192.168.0.0/23 --dport
http,https,ftp,ftp-data -o eth0 -j ACCEPT
> iptables -A FORWARD -p tcp -m state --state RELATED,ESTABLISHED -m multiport -d
192.168.0.0/23 --sport http,https,ftp,ftp-data -i eth0 -j ACCEPT
```

Il est également nécessaire d'autoriser l'accès au serveur Web de la DMZ :

```
> iptables -A FORWARD -p tcp -d 192.168.1.1/32 --dport http -j ACCEPT
> iptables -A FORWARD -p tcp -m state --state RELATED,ESTABLISHED -s 192.168.1.1/32 -
-sport http -j ACCEPT
```

- Autorisation des réponses aux requêtes ICMP (*Internet Control Message Protocol*) :

On ne va autoriser que les paquets ICMP venant en réponse à des requêtes adressées de l'intérieur du réseau :

```
> iptables -A FORWARD -p icmp -s 192.168.0.0/23 -o eth0 -j ACCEPT
> iptables -A FORWARD -p icmp -m state --state RELATED,ESTABLISHED -d 192.168.0.0/23
-i eth0 -j ACCEPT
```

- Autorisation des connexions SSH (*Secure SHell*) sur le pare-feu :

Pour des raisons de maintenance sur le routeur, on autorise les connexions via SSH sur ce dernier :

```
> iptables -A INPUT -p tcp --dport ssh -j ACCEPT
> iptables -A OUTPUT -p tcp --sport ssh -j ACCEPT
```

- Redirection de paquets :

Pour terminer, il est nécessaire de rediriger les requêtes effectuées sur le port 80 (HTTP) du routeur vers celui du serveur Web de la zone 'DMZ' :

```
> iptables -t nat -A PREROUTING -p tcp -i eth0 -d 172.16.6.50 --dport 80 -j DNAT --
to-destination 192.168.1.1:80
```

SCRIPT

Afin de faciliter la mise en place et l'arrêt des règles de filtrage, un script a été mis en place :

```
#!/bin/sh
export IPT="/usr/sbin/iptables"
export INT_EXT="eth0"
export INT_DMZ="eth1"
export INT_INT="eth2"

fw_start() {
  #Preparer le systeme a router et filtrer
  echo 1 > /proc/sys/net/ipv4/ip_forward
  modprobe ip_tables
  modprobe iptable_filter
  modprobe iptable_nat

  #Purger le filtre par precaution
  $IPT -t filter -F
  $IPT -t nat -F
```

```

#Appliquer les strategies par default : tout refuser
$IPT -P INPUT DROP
$IPT -P OUTPUT DROP
$IPT -P FORWARD DROP

#Activer la translation d'adresses (NAT) pour l'exterieur
$IPT -A POSTROUTING -t nat -s 192.168.0.0/23 -o $INT_EXT -j MASQUERADE

##### Creation des regles de filtrage (par ordre de priorite) #####

#1) Activer le FORWARDING pour la resolution de nom DNS
$IPT -A FORWARD -p udp -s 212.180.1.79 --sport 53 -i $INT_EXT -j ACCEPT
$IPT -A FORWARD -p udp -d 212.180.1.79 --dport 53 -o $INT_EXT -j ACCEPT
$IPT -A FORWARD -p udp -s 212.180.0.137 --sport 53 -i $INT_EXT -j ACCEPT
$IPT -A FORWARD -p udp -d 212.180.0.137 --dport 53 -o $INT_EXT -j ACCEPT

#2) Autoriser les echanges HTTP(S), FTP
$IPT -A FORWARD -p tcp -m state --state RELATED,ESTABLISHED -m multiport -d
192.168.0.0/23 --sport http,https,ftp,ftp-data -i $INT_EXT -j ACCEPT
$IPT -A FORWARD -p tcp -m multiport -s 192.168.0.0/23 --dport http,https,ftp,ftp-
data -o $INT_EXT -j ACCEPT
#Reponses HTTP venant du serveur WEB (DMZ)
$IPT -A FORWARD -p tcp -d 192.168.1.1/32 --dport http -j ACCEPT
$IPT -A FORWARD -p tcp -m state --state RELATED,ESTABLISHED -s 192.168.1.1/32 --
sport http -j ACCEPT

#3) Autoriser certains echanges ICMP (reponses < exterieur)
$IPT -A FORWARD -p icmp -m state --state RELATED,ESTABLISHED -d 192.168.0.0/23 -i
$INT_EXT -j ACCEPT
$IPT -A FORWARD -p icmp -s 192.168.0.0/23 -o $INT_EXT -j ACCEPT

#4) Autoriser les connexions SSH sur le serveur (routeur)
$IPT -A INPUT -p tcp --dport ssh -j ACCEPT
$IPT -A OUTPUT -p tcp --sport ssh -j ACCEPT

#5) Rediriger les entrees sur port 80 sur le serveur WEB (DMZ)
$IPT -t nat -A PREROUTING -p tcp -i $INT_EXT -d 172.16.6.50 --dport 80 -j DNAT --
to-destination 192.168.1.1:80
}

fw_stop() {
    $IPT -t filter -F
    $IPT -t nat -F

    modprobe -r ipt_state
    modprobe -r ipt_MASQUERADE
    modprobe -r iptable_filter
    modprobe -r iptable_nat
    modprobe -r ip_conntrack
    modprobe -r ipt_multiport
    modprobe -r ip_tables
}

case "$1" in
'start')
    fw_start
    ;;
'stop')
    fw_stop
    ;;
*)
    echo "usage $0 start|stop"
esac

```