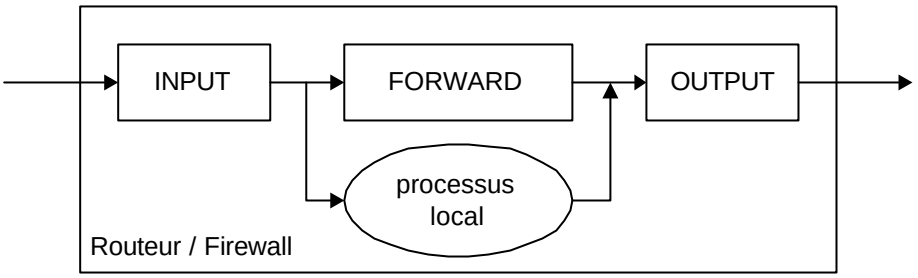
filtrage sous Linux avec IPFWADM

Un paquet entrant sur une interface du routeur/firewall peut être destiné à un processus local, ou à être émis sur une autre interface.
Un paquet sortant sur une interface du routeur/firewall peut provenir d'un processus local ou d'une autre interface.
Les paquets sont traités par les règles de filtrages Input, Forward et Output selon le schéma ci-dessous.



La commande `ipfwadm -h` affiche l'aide et les options de ipfwadm.

options de la commande ipfwadm (affiché par la commande "ipfwadm –h")		
-A	accounting	comptage
-F	forward	trafic traversant
-I	input	trafic entrant
-O	output	trafic sortant
-M	masquerading	masquage
-h	help	aide
-i	insert	insère une règle (accept, reject, deny)
-a	add	ajoute une règle
-d	delete	efface une règle
-l	list	liste les règles
-z	zero	remet à zéro le compteur de paquets
-f	flush	vide (efface) toutes les règles
-p	policy	change la règle par défaut
-c	check	teste l'acceptance du paquet
-P	protocol	protocole spécifié (tcp, udp, icmp, all)
-S	source	adr IP... port source
–D	destination	adr IP... port destination
–V		adresse de l'interface réseau
–W		nom de l'interface réseau
–b	bidirectional	correspondance bidirectionnelle
-e	extended	mode sortie étendue
-k	ack	paquet TCP avec bit ACK=1
-m		
-n		
-o		
-r		
-t		
-v	verbose	verbeux
-x		
-y	syn	paquet TCP avec bit SYN=1

Si vous utilisez le noyau 2.1.102 ou plus récent, voyez plutôt la commande IPCHAINS.
Dans les noyaux plus anciens, IP Forwarding est activé par défaut dans le noyau.
Pour cette raison, votre réseau devrait démarrer en refusant l'accès à tout et en effaçant toutes les règles ipfw en place lors du dernier fonctionnement.
Ce fragment de script devrait se trouver dans votre script de démarrage réseau (/etc/rc.d/init.d/network)

```
# setup IP packet Accounting and Forwarding
#
```

```
  #   Forwarding
  #
  # By default DENY all services
  ipfwadm -F -p deny
  # Flush all commands
  ipfwadm -F -f
  ipfwadm -I -f
  ipfwadm -O -f
```

Nous avons maintenant le firewall parfait : rien ne passe.
Créons à présent le fichier /etc/rc.d/rc.firewall. Ce script devrait autoriser le trafic email, Web et DNS.

```
#! /bin/sh
#
# rc.firewall
#
# Source function library.
. /etc/rc.d/init.d/functions
# Get config.
. /etc/sysconfig/network

# Check that networking is up.
if [ ${NETWORKING} = "no" ]
then
        exit 0
fi
case "$1" in
  start)
  echo -n "Starting Firewall Services: "
  # Allow email to got to the server
  /sbin/ipfwadm -F -a accept -b -P tcp -S 0.0.0.0/0 1024:65535 -D
192.1.2.10 25
  # Allow email connections to outside email servers
  /sbin/ipfwadm -F -a accept -b -P tcp -S 192.1.2.10 25 -D 0.0.0.0/0
1024:65535
  # Allow Web connections to your Web Server
  /sbin/ipfwadm -F -a accept -b -P tcp -S 0.0.0.0/0 1024:65535 -D
192.1.2.11 80
  # Allow Web connections to outside Web Server
  /sbin/ipfwadm -F -a accept -b -P tcp -S 192.1.2.* 80 -D 0.0.0.0/0
1024:65535
  # Allow DNS traffic
  /sbin/ipfwadm -F -a accept -b -P udp -S 0.0.0.0/0 53 -D 192.1.2.0/24
  ;;
  stop)
  echo -n "Stooping Firewall Services: "
  ipfwadm -F -p deny
  ;;
  status)
  echo -n "Now do you show firewall stats?"
  ;;
  restart|reload)
        $0 stop
        $0 start
        ;;
  *)
        echo "Usage: firewall {start|stop|status|restart|reload}"
        exit 1
esac
```

NOTE: dans cet exemple nous avons le serveur de mail (smtp) tournant sur 192.1.2.10 qui doit être capable d'envoyer et de recevoir sur le port 25, le serveur Web tournant sur 192.1.2.11.
Nous autorisons tous les postes du LAN à aller sur des serveurs Web et DNS à l'extérieur.

Ceci n'est pas une sécurité parfaite. Parce que le port 80 n'est pas réservé au Web, un petit hacker peut utiliser ce port pour créer un VPN à travers le firewall. La parade consiste à monter un proxy web et à ne permettre qu'au proxy de traverser le firewall pour aller vers un serveur Web extérieur.

Vous pouvez souhaiter mesurer le trafic traversant votre firewall. Ce script va compter chaque paquet. Vous pouvez ajouter une ligne ou deux pour compter les paquets allant vers un système donné.

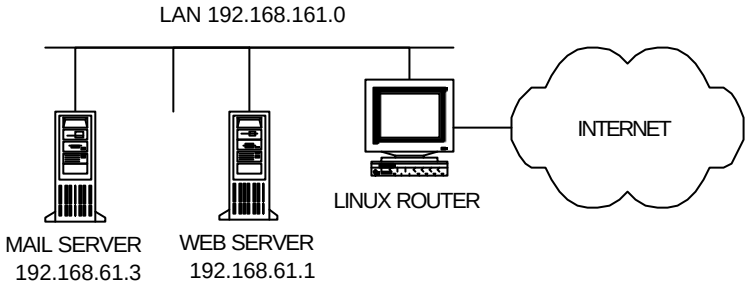
```
# Flush the current accounting rules
ipfwadm -A -f
# Accounting
/sbin/ipfwadm -A -f
/sbin/ipfwadm -A out -i -S 192.1.2.0/24 -D 0.0.0.0/0
/sbin/ipfwadm -A out -i -S 0.0.0.0/0 -D 192.1.2.0/24
/sbin/ipfwadm -A in -i -S 192.1.2.0/24 -D 0.0.0.0/0
/sbin/ipfwadm -A in -i -S 0.0.0.0/0 -D 192.1.2.0/24
```

Si vous ne souhaitez qu'un simple filtrage, vous pouvez stopper ici.
AUTRE EXEMPLE :

Par défaut, interdire tout passage, comme dans toute configuration de firewall :
`ipfwadm -F -p deny`

Vider toutes les règles de filtrage, maintenant nous avons un firewall absolu, rien ne peut passer au travers :

```
ipfwadm -F -f
ipfwadm -I -f
ipfwadm -O -f
```



En fonction des applications, nous allons maintenant autoriser quelques services à passer à travers le firewall. L'entreprise prise ici comme exemple possède une adresse privée de classe C : 192.168.61.0 à 192.168.61.255. Parmi ces adresses, quelques-unes sont remarquables :

- serveur web : 192.168.61.1
- serveur de messagerie : 192.168.61.3

Autoriser l'accès de l'e-mail (SMTP:25) au serveur :
`ipfwadm -F -a accept -b -P tcp -S 0.0.0.0/0 1024:65535 -D 192.168.61.3 25`

Autoriser les connexions e-mail (SMTP:25) vers les serveurs extérieurs :
`ipfwadm -F -a accept -b -P tcp -S 192.168.61.3 25 -D 0.0.0.0/0 1024:65535`

Autoriser les connexions web (HTTP:80) vers le serveur web de l'entreprise (trafic HTTP entrant) :
`ipfwadm -F -a accept -b -P tcp -S 0.0.0.0/0 1024:65535 -D 192.168.61.1 80`

Autoriser les connexions web de toutes les machines du réseau local vers les serveurs web extérieurs
`ipfwadm -F -a accept -b -P tcp -S 196.168.61.* 80 -D 0.0.0.0/0 1024:65535`

Autoriser le trafic DNS :
`ipfwadm -F -a accept -b -P udp -S 0.0.0.0/0 53 -D 196.168.61.1/24`

Masquer un réseau entier par le firewall :

```
ipfwadm -F -p deny
ipfwadm -F -a m -S 192.168.61.0/24 -D 0.0.0.0/0
```

Ou bien masquer certaines adresses seulement :

```
ipfwadm -F -p deny
ipfwadm -F -a m -S 192.168.61.4/32 -D 0.0.0.0/0
ipfwadm -F -a m -S 192.168.61.10/32 -D 0.0.0.0/0
```

Selon les besoins, ne pas oublier d'insérer les modules nécessaires au masquage de certains protocoles :

```
modprobe ip_masq_ftp.o
modprobe ip_masq_irc.o
modprobe ip_masq_cuseeme.o
```

Tous ces paramètres ne sont pas évidents à remplir quand on veut obtenir une solution qui corresponde aux besoins de l'entreprise : autoriser tel ou tel trafic, en interdire un autre… De plus, comme habituellement dans le monde Unix, les paramètres à entrer derrière les commandes sont difficiles à retenir.

Pour plus de convivialité il existe une interface graphique qui reprend les fonctionnalités de ipfwadm : Easyfw.