Configuration d'un firewall sous Linux

Présentation générale

Un pare-feu (firewall) est une passerelle que l'on place sur un réseau pour sécuriser les communications entrantes et sortantes. Il existe deux familles de pare-feu : les firewalls à filtrage de paquets et les firewalls de type proxy. Ils ne travaillent pas de la même façon et n'offrent pas les mêmes sécurités.
Selon la version du noyau Linux utilisée, plusieurs types de firewall filtrant existent :

- jusqu'à la version 2.1.102, c'est *ipfwadm* qui est implémenté,
- depuis la version 2.1.102, on utilise *ipchains*,
- à partir du noyau 2.4, *iptables/NetFilter* est implémenté en plus.

NetFilter permet d'offrir une infrastructure dédiée au filtrage/manipulation de paquets, que les utilisateurs et développeurs pourraient déployer comme un add-on construit autour du noyaux Linux. Il a été conçu pour être modulaire et extensible. *Iptables* est un module qui s'insère dans la structure de NetFilter et autorise l'utilisateur à accéder aux règles et commandes de filtrage/manipulation du noyau.
Le but de cette documentation est d'expliquer comment configurer un pare-feu avec les *iptables* sous Linux.

Démarrage

Tout d'abord *iptables* est une commande que seul **root** peut lancer. Ensuite, vérifier la version de votre noyau :

```
> uname -a
```

Si la version du noyau est antérieure à 2.4, il faudra installer *iptables* (cf. installation)
Pour voir si la commande est présente :

```
> which iptables
```

Pour voir si le module est installé :

```
> lsmod | grep iptable
```

Installation de iptables (succinct)

IpTables a besoin d'un kernel de génération 2.4 **compilé** avec des options spéciales. Ceci ne pose pas de problèmes avec les noyaux 2.4 génériques des principales distributions basées sur cette génération de kernel.

- **Le noyau**

Si vous désirez re-compiler votre kernel (`make dep; make clean; make bzImage`), il faut spécifier les options nécessaires au fonctionnement d'iptables comme CONFIG_PACKET, CONFIG_NETFILTER, CONFIG_IP_NF_CONNTRACK, CONFIG_IP_NF_FTP, CONFIG_IP_NF_IRC, CONFIG_IP_NF_IPTABLES, CONFIG_IP_NF_FILTER, CONFIG_IP_NF_NAT, CONFIG_IP_NF_MATCH_STATE, CONFIG_IP_NF_TARGET_LOG, CONFIG_IP_NF_MATCH_LIMIT, CONFIG_IP_NF_TARGET_MASQUERADE
Déplacez votre ancien répertoire de module

```
> mv /lib/module/votre_version /lib/module/votre_version.old
```

Puis compilez et installez les modules (`make modules; make modules_install`), Copiez alors le nouveau kernel

```
> cp /usr/src/linux/arch/i386/boot/bzImage /boot
```

- **Netfilter**

Récupérer sur le site officiel les sources (*http://www.samba.org/netfilter*).
Désarchiver et compiler (`make, make install`).

T. BERNARD DE DOMPSURE	Projet d'administration réseau	IUP-ASR 2004-2005

- **Chargement des modules**

Il est nécessaire de charger ces modules à chaque *boot* puis de lancer les règles de filtrage/masquage, avant de pouvoir utiliser iptables :
modprobe ip_tables
Il faut faire un script qui sera lancé à chaque démarrage (à rajouter dans `/etc/rc.d/init.d` suivant le système). Et selon les besoins, on peut éventuellement charger d'autres modules (cf. modules)

Fonctionnement d'iptables

Le noyau contient par défaut trois tables : Filter, NAT et Mangle. Une table permet de définir un comportement précis du firewall Linux.

- La table **FILTER** va contenir toutes les règles pour le filtrage des paquets.
- La table **NAT** permet d'effectuer des translations d'adresses.
- La table **MANGLE** permet de marquer des paquets entrants (PREROUTING) et générés localement (OUTPUT). Ce marquage va permettre de traiter spécifiquement les paquets marqués dans les tables de routage.

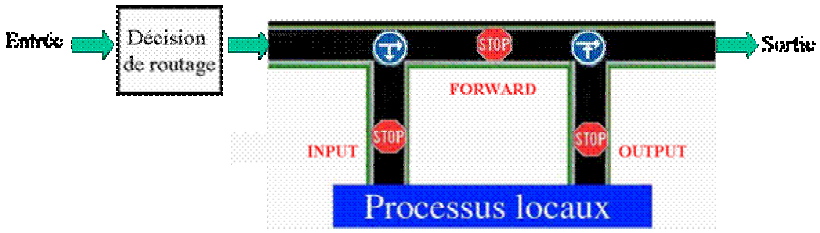
Une table est un ensemble de chaînes, elles-mêmes composées de règles.

FILTER		NAT		MANGLE
INPUT	Contrôle des paquets entrant localement sur l'hôte	**PREROUTING**	Pour faire la translation d'adresse de destination	**PREROUTING**
OUTPUT	Contrôle des paquets sortant localement sur l'hôte	**POSTROUTING**	Pour faire la translation d'adresse de la source	**OUTPUT**
FORWARD	Filtre les paquets qui passent d'une interface réseau à l'autre	**OUTPUT**	Pour modifier la destination de paquets générés localement	*Marquage des paquets entrants (PREROUTING) et générés localement (OUTPUT)*

Sur un firewall, lorsqu'un paquet arrive, la fonction de décision de routage va déterminer si le paquet est destiné à un processus local ou à un hôte sur un autre réseau. Deux cas peuvent se présenter :

- **Le paquet lui est destiné :** le paquet traverse la chaîne INPUT. S'il n'est pas rejeté, il est transmis au processus sollicité. Il sera traiter. Peu éventuellement émettre un paquet en réponse. Ce nouveau paquet traverse la chaîne OUTPUT. S'il n'est pas rejeté, il va vers la sortie.

- **Si le paquet est destiné à un hôte sur un autre réseau :** il traverse la chaîne FORWARD. S'il n'est pas rejeté, il poursuit alors sa route.



Configuration des tables

Suivant la syntaxe utilisée, la commande *iptables* va permettre de spécifier des règles de sélection des paquets IP. On va pouvoir :

- **Ajouter** des règles / chaînes.
- **Supprimer** des règles / chaînes.
- **Modifier** des règles / chaînes.
- **Afficher** les règles / chaînes

Les paquets sont récupérer suivant leurs : adresse source, adresse destination, protocole et numéro de port. Pour chaque règle de sélection, on peut soit accepter le paquet, soit l'ignorer, soit renvoyer une erreur. Pour voir les commandes les plus usuelles :

```
> man iptables
```

- **Commandes principales**

Ces options spécifient une action particulière à effectuer. Seule l'une d'elles peut être spécifiée sur la ligne de commande, sauf indication contraire.

-A	--append	Ajoute une ou plusieurs règles à la fin de la chaîne sélectionnée
-D	--delete	Permet de supprimer une chaîne
-R	--replace	Remplace une règle dans la chaîne sélectionnée
-I	--insert	Pour ajouter une chaînes
-L	--list	Pour afficher les règles de filtrages
-F	--flush	Pour supprimer toutes les règles
-N	--new	Permet de créer une nouvelle chaîne
-X	--delete-chain	Permet d'effacer une chaîne
-P	--policy	Met en place le comportement par défaut : ACCEPT, REJECT, DROP…

- **Commandes pour matcher**

-p	--protocol	Le protocole de la règle ou du paquet à vérifier : tcp, icmp, all…
-s	--source	Spécification de la source
-d	--destination	Spécifie la destination
-j	--jump	Spécifie ce qu'il faut faire si le paquet correspond à la règle
-i	--in-interface	Spécifier une interface d'entrée
-o	--out-interface	Spécifier une interface de sortie
-f	--fragment	Paquet fragmenté
-sport	--source-port	Spécifier le port source ou une plage de ports
-dport	--destination-port	Spécifier le port destination ou une plage de ports
--tcp-flags		Spécifier un flag tcp à matcher : SYN ACK FIN RST URG PSH ALL
--icmp-type		Spécifier un type de paquet icmp
--mac-source		Spécifier l'adresse MAC à matcher
--state		Etats du paquet à matcher : ESTABLISHED, NEW, INVALID,RELATED

Mise en place du pare-feu

Lors de la mise en place d'un pare-feu, on doit :

- effacer toute les règles existantes et s'assurer qu'aucune règle n'est appliquée

```
> iptables -F
> iptables –L
```

- appliquer une politique par défaut qui refuse tous les paquets

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


- Pour logguer tout ce qu'on jette :

```
> iptables -N LOG_DROP
> iptables -A LOG_DROP -j LOG --log-prefix '[IPTABLES DROP] : '
> iptables -A LOG_DROP -j DROP
```

Et ensuite

```
> iptables -A FORWARD -j LOG_DROP
> iptables -A INPUT -j LOG_DROP
> iptables -A OUTPUT -j LOG_DROP
```

- Pour accepter tout ce qui se passe sur l'interface `lo` par exemple

```
> iptables -A INPUT -i lo -j ACCEPT
> iptables -A OUTPUT -o lo -j ACCEPT
```

- Pour refuser les connexion sortantes vers les services non sécurisés telnet, FTP, et rsh

```
> iptables -A INPUT -t DENY -p tcp --dport telnet,ftp,shell
```

- Pour accepter le trafic FTP

```
> iptables -A INPUT -i eth0 -p tcp -sport 21 -m state --state ESTABLISHED -j ACCEPT
```

(pour accepter les trames FTP qui rentrent sur l'interface eth0 seulement si c'est une connexion déjà établis)

```
> iptables -A OUTPUT -o eth0 -p tcp -dport 21 -m state --state NEW,ESTABLISHED -j ACCEPT
```

(pour accepter les trames FTP qui sortent si c'est une nouvelle connexion ou une connexion déjà établis)

- Pour empêcher un ping sur une machine du réseau 192.168.2.0

```
> iptables –A FORWARD –i eth0 –d 192.168.2.0 –p icmp –j DROP
```

- Pour supprime toutes les trames rentrantes dans l'espace utilisateur, sauf HTTP (avec '!')

```
> iptables -I INPUT -p tcp -sport ! 80 -j DROP
```

Mise en place de modules

Les modules permettent de rajouter des fonctionnalités à NetFilter. Prenons l'exemple pour FTP.
Charger le module :

```
> modprobe ip_conntrack_ftp
```

Ensuite, il faut vérifier que le trafic FTP est autorisé (cf. Pour accepter le trafic FTP). C'est indispensable pour que la connexion puisse s'établir.
Ensuite c'est ici que ip_conntrack_ftp va servir :

```
> iptables -A INPUT -i eth0 -p tcp -sport 20 -m state --state ESTABLISHED,RELATED
-j ACCEPT
```

(RELATED signifiant que le paquet initie une nouvelle connexion)

```
> iptables -A OUTPUT -o eth0 -p tcp -dport 20 -m state --state ESTABLISHED -j
ACCEPT
```

Enfin pour que le serveur puisse établir la connexion pour les données (en mode actif) :

```
> iptables -A INPUT -i eth0 -p tcp --sport 1024:65535 --dport 1024:65535 -m state
--state ESTABLISHED -j ACCEPT
> iptables -A OUTPUT -o eth0 -p tcp --sport 1024:65535 --dport 1024:65535 -m state
--state ESTABLISHED,RELATED -j ACCEPT
```

(`sport` et `deport` signifie l'intervalle des ports acceptés. En effet, la connexion se fait sur le port 20/21 mais les transferts se font sur d'autres ports

De nombreuses autres règles de filtrages relativement fines peuvent être mise en place avec `iptables`. Nous avons vu seulement la table FILTER. Les autres tables fonctionnent de la même manière. Il y a certaines limitations de NetFilter : par exemple il n'intègre pas des fonctionnalités de limitation de bande passante ou il n'intègre pas des fonctionnalités d'introduction d'un pourcentage d'erreur dans les communications.

Bibliographie

http://www.samba.org/netfilter
http://doc-libre.equinux.ca
http://www.ze-linux.org