Fonctionnement du parefeu Netfilter

I Qu'est ce qu'un firewall ?

Un firewall ou pare-feu est un dispositif permettant de filtrer et masquer le trafic TCP/IP entre un réseau public (internet par exemple) et un réseau privé (le réseau local). Il protège le réseau contre les attaques et infiltrations venant d'éventuels pirates.

1. Fonctions de masquage et de translation d'adresses

Ces techniques sont utilisées principalement dans deux cas :

- connecter plus de stations qu'il n'y a d'adresses IP disponibles
- masquer les adresses réelles des stations derrière le routeur

a) La translation d'adresses (*NAT : Network Address Translation*)

Le routeur, depuis le réseau externe, répond à plusieurs adresses IP publiques qui sont traduites vers des adresses du réseau interne (en général des adresses privées). Cette translation peut être statique (serveurs) ou dynamique (postes de travail). Le routeur modifie *chaque paquet IP* pour remplacer l'adresse interne par l'adresse externe en sortie, et inversement en entrée.

Exemple: translation d'adresses entre un réseau de classe A privé 10.0.0.0/8 (2^{24} adresses potentielles) sur un réseau de classe C officiel 195.195.195/24

b) Le masquage d'adresses IP (*IP masquerade*) ou translation de port (PAT):

L'IP masquerade, consiste à masquer les @IP du réseau interne et à n'utiliser que l'adresse du routeur sur le réseau d'interconnexion. Cette technique est intéressante pour connecter tout un réseau local sur l'accès d'un prestataire qui ne fournit qu'une seule adresse IP (souvent dynamique). Le routeur tient, pour chaque connexion TCP/UDP initiée depuis le réseau interne, la correspondance entre les triplets (@IP interne, TCP/UDP, port utilisateur) et (@IP de sortie, TCP/UDP, port translaté). Dans chaque paquet sont modifiés par le routeur, *l'adresse IP* et le *port utilisateur de l'émetteur*.

Exemple: connexion à internet d'un réseau de classe C privé 192.168.100/24 en utilisant l'adresse d'interconnexion vers le réseau externe. Dans ce type de configuration, seul le routeur est visible de l'extérieur. C'est la solution pour partager un accès unique vers un prestataire Internet.

2. Les fonctions de filtrage :

Pour des raisons de sécurité il est souvent utile, voire indispensable d'effectuer un filtrage au niveau des paquets IP qui transitent à travers un routeur. *Exemple*: effectuer les filtrages pour autoriser en sortie tout le trafic Web et ftp et interdire tout le reste

Le filtrage s'effectue en analysant les champs (source/destinataire) des paquets qui transitent à travers le routeur. On peut ainsi filtrer sur les @IP, le protocole, les ports, etc. En général les règles de filtrage sont analysées de manière séquentielle, dès qu'une règle correspond au paquet analysé, elle est appliquée.

II Filtrage et masquage avec netfilter sous Linux

Linux peut intègrer de base (selon les options d'installation), les fonctions de routage, de filtrage et de masquage. Il s'agit du duo netfilter/iptables dans les noyaux supérieurs ou égaux à 2.4.x

Netfilter est un sous-système du noyau Linux qui permet d'inspecter les paquets IP et de les filtrer selon certaines règles. Il utilise le mécanisme de **listes d'accès** qui sont regroupées dans des "**chaînes**", elles mêmes contenues dans des **tables**. Chaque paquet IP qui arrive, traverse, sort de Linux est analysé et traité (accepté, rejeté, refusé, modifié, redirigé) en fonction des règles qui lui sont applicables. Les règles sont évaluées dans l'ordre ou elle ont été écrites. Dès qu'un règle est applicable elle est appliquée et le paquet sort de la chaîne. Netfilter permet de faire de la translation d'adresses et du masquage (translation de ports).

iptables est l'outil (la commande du sytème) qui permet d'écrire les règles.

Trois tables sont utilisées :

filter, table par défaut, qui contient les chaînes de règles de filtrage

nat, qui contient les chaînes de règles de translation d'adresse/port

mangle, qui contient les chaînes de règles de modification de paquets (que nous n'étudierons pas ici)

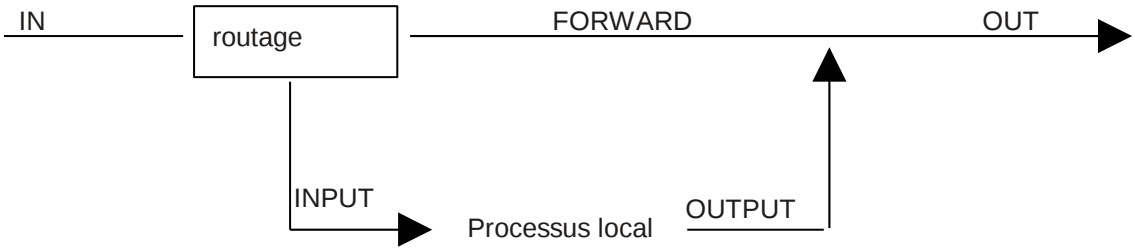
1. Le filtrage : table filter

Dans la table ***filter***, trois chaînes sont prédéfinies :

FORWARD qui contient les règles à appliquer aux paquets qui **traversent** le parefeu (paquets routés)

INPUT qui contient les règles à appliquer aux paquets entrant sur le parefeu **destinés aux processus locaux**

OUTPUT qui contient les règles à appliquer aux **paquets émis** par les processus locaux, sortant du routeur.



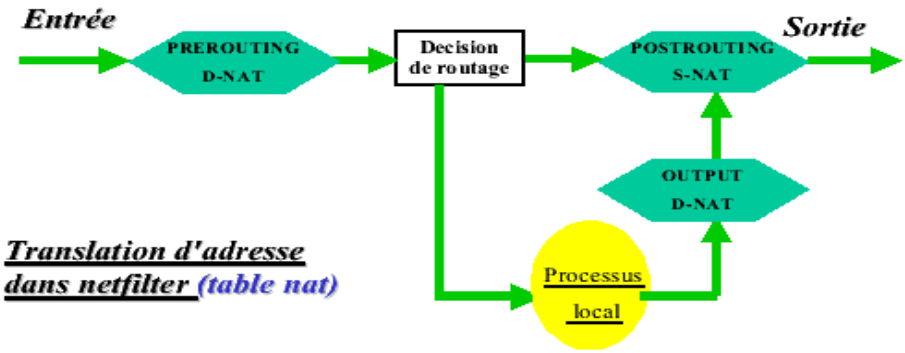
Le routeur est une « ***boîte noire*** ». Chaque paquet entrant est « ***pris en charge*** », routé par le noyau.

Les paquets qui ne font que traverser le routeur Linux, sont concernés par la chaîne ***FORWARD***. Ceux qui sont destinés aux processus internes, c'est à dire qui entrent, sont concernés pas la chaîne ***INPUT***, ceux émis par les processus internes, c'est à dire qui sortent, par la chaîne ***OUTPUT***.

L'outil iptables sous Linux permet de contrôler le trafic TCP/IP qui traverse le routeur. Il assure le filtrage des paquets TCP/IP en fonction des adresses IP sources et destination, du protocole de transport (TCP/UDP/ICMP) et du protocole applicatif (ports sources et destination). C'est un firewall « stateful » c'est-à-dire qu'il permet de filtrer les paquets en fonction de l'état de la connexion TCP associée. Par exemple, on peut décider d'autoriser l'entrée des paquets correspondant à une connexion déjà établie en refusant tous les paquets correspondant à l'établissement d'un nouvelle connexion. Pour cela, le firewall garde en mémoire l'état de chaque connexion TCP qui le traverse.

2. La translation d'adresse : table NAT

La translation d'adresse a été entièrement revue dans les noyaux 2.4. Les paquets IP sont examinés, dans la table NAT, dans les chaînes **PREROUTING**, **POSTROUTING** et **OUTPUT**.



Dans les chaînes ***PREROUTING*** et ***OUTPUT*** on ne peut que modifier l'adresse de destination du paquet (cible ***DNAT***).

Dans la chaîne ***POSTROUTING*** on ne peut que modifier l'adresse source du paquet (cible ***SNAT***). C'est cette chaîne qui servira à faire le masquage IP (translation de port)

Liens

page officielle : http://www.netfilter.org/

Léa linux :http://lea-linux.org/reseau/iptables.php3

http://www.hsc.fr/ressources/presentations/netfilter/netfilter.htm

Iptables : tutoriel (extrait simplifié)

Ajouter une nouvelle règle à une chaîne : iptables -A

Pour l'exemple, nous "droperons" tous les paquets icmp en provenance de 127.0.0.1:

```
iptables -A INPUT -s 127.0.0.1 -p icmp -j DROP
```

Supprimer les règles d'une chaîne

```
iptables -F  [nom_de_la_chaine] [-t nom_de_la table]
```

Attention! Si vous ne spécifiez aucun nom de table , seule la table filter sera vidée

suppression de toutes les règles de la table nat:

```
iptables -F -t nat
```

Traiter le paquet avec -jump, ou -j

Spécifie ce que vous désirez faire du paquet. La cible peut être une sous-chaîne que vous aurez vous-même créé ou alors une valeur spéciale. Celle-ci peut prendre la forme de **ACCEPT**, **DROP**, **.**

ACCEPT autorise le paquet à continuer son chemin, **DROP** supprime la paquet. **REJECT** refuse le paquet mais en avertissant le demandeur que sa demande de connexion lui a été refusée en lui envoyant un message ICMP port-unreachable.

On laisse passer tous les paquets sortants:

```
iptables -A OUTPUT -j ACCEPT
```

Spécifier une politique par défaut pour les chaines de la table filter

C'est l'action qui sera appliquée si aucune règle ne s'applique au paquet.

Par défaut un paquet entrant est refusé:

```
iptables -P INPUT DROP
```

Spécifier une adresse source ou destination

Refuser tout ce qui provient de l'hôte 195.52.4.1:

```
iptables -A INPUT -s  195.52.4.1 -j DROP
```

Laisser passer tout ce qui est destiné au réseau 192.168.0.0:

```
iptables -A OUTPUT -d  192.168.0.0/16 -j ACCEPT
```

Mentionner un protocole avec -p

Refuser les entrées des paquets du protocole ICMP (ping entre autres):

```
iptables -A INPUT  -p icmp -j DROP
```

Mentionner les ports avec --sport, --source-port, --dport, --destination-port

Laisser sortir les requêtes Web :

```
iptables -A OUTPUT  -j ACCEPT -p tcp --dport 80
```

Accepter les réponses web:

```
iptables -A INPUT -j ACCEPT -p tcp --sport 80
```

Spécifier une interface avec -i, --in-interface ou -o, --out-interface

```
iptables -A INPUT -i eth0 -o ppp0
iptables -A INPUT -i lo
```

Spécifiez l'état de la connexion à laquelle appartient le paquet.

Netfilter est un firewall « stateful » ce qui veut dire qu'il peut accepter ou refuser les paquet en fonction de 'état en cours des connexions. Netfilter garde en dans une table d'états, les connexions en cours. Cela permet d'associer que tel client (adresse IP cliente) vers tel serveur (adresse IP serveur) est en train de faire telle chose (connexion du port source x vers le port destination y). On pourra donc décider de spécifier les paquets correspondant à un connexion établie (ESTABLISHED), un nouvelle connexion (NEW) ou une nouvelle connexion associée à un connexion déjà établie (RELATED). C'est le cas par exemple des connexion de données FTP qui sont associée à une connexion de type commande pré-établie, ou des messages d'erreur icmp correspondant à un requête émise par le routeur.

Accepter les paquets correspondant à des connexion établies ou associées:

```
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

Exemple d'un script de configuration de Netfilter :

Cas d'une connexion internet partagée par un réseau local. Le routeur filtrant possède une interface externe (eth1) et une interface interne (eth0) :

```
#activation du routage
echo 1 > /proc/sys/net/ipv4/ip_forward

#chargement des modules nécessaires
/sbin/modprobe iptable_nat
/sbin/modprobe ip_tables
/sbin/modprobe ip_conntrack
/sbin/modprobe iptable_filter
/sbin/modprobe iptable_nat
/sbin/modprobe ipt_state
/sbin/modprobe ipt_MASQUERADE

#effacement des règles des tables
iptables -t nat -F
iptables -F

#potitiques par défaut
iptables -P FORWARD DROP
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT

#on accepte l'entrée de paquets sur l'interface localhost (indispensable sinon ca bloque)
iptables -A INPUT  -i lo -j ACCEPT

#on accepte toute entrée sur eth0
iptables -A INPUT -i eth0 -j ACCEPT

#on n'accepte que l'entrée de paquet correspondant à des connexions établies ou des #connexions relatives à des connexions établies (connexion de données ftp par exemple ou messages d'erreur icmp)
iptables -A INPUT -i eth1 -m state --state RELATED,ESTABLISHED  -j ACCEPT

#on route tous les paquets venant de eth0 (interface interne)
iptables -A FORWARD -i eth0 -j ACCEPT

#pour l' autre interface (externe) on ne route que les paquets correspondant à des #connexions établies ou des #connexions relatives à des connexions établies
iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT

#on active l'ip masquerade pour les adresses du réseau 10.0.0.0 sortant sur l'interface #externe
iptables -t nat -A POSTROUTING -s 10.0.0.0/8 -o eth1 -j MASQUERADE
```

Création d'un fichier de script.

Ce fichier sera créé par vi et appelé /etc/firewall.sh. Il contiendra l'ensemble des règles de notre firewall. Pour l'exécuter il faudra d'abord le rendre exécutable par la commande « chmod 700 /etc/firewall.sh » puis de taper /etc/firewall.sh, ce qui lancera l'interprétation des commandes du script.

Pour visualiser les règles créées par l'exécution du script:

```
iptables -L (affiche les règles de la table filter)
iptables -F -t nat (affiche les règles de la table nat)
```

Pour automatiser le filtrage lors du démarrage du routeur, il faut lancer le script de création des règles /etc/firewall.sh au démarrage du système. Pour cela, éditez le fichier /etc/rc.local et ajouter la ligne suivante à la fin :

```
/etc/firewall.sh
```