Firewall

Les chaînes

Les chaînes se positionnent à 5 endroits différents :

- PREROUTING : situé entre les cartes réseaux et le niveau chargé de déterminer si le trafic est local ou juste sensé traverser la machine.
- INPUT : situé entre le choix trafic local ou pas et l'accès aux programmes locaux (serveurs)
- FORWARD : utilisée par les paquets qui ne font que traverser la machine, après les avoir séparés du trafic local et avant de déterminer la carte réseau de sortie
- OUTPUT : situé entre les programmes locaux et la détermination de la carte réseau de sortie
- POSTROUTING : situé après avoir déterminé l'interface de sortie, juste avant la dite interface.

A ces 5 chaînes standard on peut ajouter des chaînes créées par l'utilisateur. Pour des raisons de lisibilité, on utilisera de préférence des noms en majuscules pour ces dernières

Les tables

Le firewall est composé de différentes tables utilisées pour des tâches différentes

- filter : table par défaut (pas nécessaire de la préciser), sert à déterminer quels paquets peuvent passer ou doivent être bloqués. Contient les chaînes INPUT, OUTPUT et FORWARD
- nat : utilisée pour le NAT (Network Address Tranlation). Contient les chaînes PREROUTING et POSTROUTING
- mangle : permet de modifier les paquets, par exemple pour les "tagger"[1], changer les champs QoS, …
- raw : utilisé pour des traitements de bas niveau, par exemple éviter le suivi par le connexion tracking

En général, on ne se sert que des tables filter et nat, la mise en place de QoS étant assez complexe.

Configuration d'un Firewall

Pour configurer un firewall, on utilisera un script qui effectuera les opérations suivantes :

- Charger les modules de "connexion tracking" nécessaires (le conntracking si on désire utiliser l'état des connexions, les modules spécifiques à des protocoles si ils sont nécessaires pour le NAT)

1 Ce tagging sera par exemple utilisé pour configurer des règles de QoS au niveau de l'interface de sortie plus facilement.

- Mettre des policy par défaut de DROP sur les files filter standard et les vider, vider les files nat.
- Effacer si nécessaire puis recréer les files "utilisateur"
- créer les règles du firewall, d'abord les règles de nat puis celles de filtrage

Pour les règles de filtrage, on commencera généralement par tester quelques situations génériques et aiguiller vers des files utilisateur spécifique. Un paquet ne devrait traverser qu'une seule de ces files utilisateurs.

La commande iptables

Cette commande aura généralement la forme

```
iptables -t table -commande file paramètres
```

si la table est filter, on omettra l'option -t. Les principales commandes sont

-P	Action par défaut (généralement DROP)
-X	Efface une file utilisateurs
-N	Crée une file utilisateur
-L	Liste les règles de la file transmise. On peut utiliser -n pour garder les valeurs numériques
-A	Ajoute une règle à la fin de la liste de règles
-F	Vide toutes les règles de la file

Les règles utiliseront généralement une cible précisée par l'option -j. Les cibles suivantes sont couramment utilisées

DROP	Le paquet est bloqué
ACCEPT	Le paquet est autorisé
REJECT	Le paquet est refusé avec un message d'erreur (ICMP) retourné à l'expéditeur
LOG	Le paquet est loggé dans les logs système. Le traitement continue dans les règles qui suivent
File utilisateur	Le traitement continue dans une file utilisateur. Si aucune règle ne le valide ou le refuse, le traitement reviendra où il en était

Connexion Tracking

Le module conntrack doit être chargé par la commande modprobe. L'ancien nom était ip_conntrack, le nouveau nf_conntrack. Les deux noms sont normalement acceptés.

On pourra alors utiliser -m state –state XXX pour tester l'état d'un paquet où XXX est une série de valeurs séparées par des , dans la liste ci-dessous :

NEW	Nouvelle connexion
ESTABLISHED	Connexion déjà existant
RELATED	Connexion liée à une connexion existante

INVALID	Etat incorrect

On testera souvent le type NEW avec les flags TCP pour s'assurer qu'il s'agit bien d'un paquet SYN. Autoriser les ESTABLISHED permet de gagner pas mal de temps de traitement (pas besoin de refaire les tests, ils ont déjà été fait)

On acceptera généralement ESTABLISHED et RELATED en début de script et on testera NEW et le début de connexion pour accepter/refuser une connexion.

Pour activer un helper, dans la table raw, utiliser -j CT --helper protohelper (protohelper = ftp, irc, ...)

TCP/IP

Au niveau IP, on pourra tester l'adresse source à l'aide de -s x.x.x.x/n où n est le masque (en bits à 1, souvent 24 pour 255.255.255.0) ou tester la destination à l'aide de -d x.x.x.x/n.

On pourra également tester le protocole avec -p (par exemple, -p tcp ou -p udp).

En TCP et en UDP, il sera également possible de tester le port à l'aide de --sport ou --dport suivi du numéro de port.

En, TCP, on pourra tester les drapeaux pour la combinaison indiquant une nouvelle connexion à l'aide de --syn.

On utilisera l'inverse d'un test en le précédent de !

Limitation de débit et black/white listing

L'extension "recent" (activée par -m recent) permet de limiter le nombre de connexions par unité de temps.

Par exemple :

```
iptables -A INPUT -p tcp --dport 22 -m state --state NEW -m recent --set --name
SSH
iptables -A INPUT -p tcp --dport 22 -m state --state NEW -m recent --update --
seconds 600 --hitcount 3 --rttl --name SSH -j DROP
iptables -A  INPUT -p tcp --dport 22 -j ACCEPT
```

La première ligne s'assure que l'entrée existe dans la liste SSH, la seconde permet de limiter à 3 connexions dans une tranche de 10 (600 secondes) minutes.0

On peut aussi tester une liste pour bloquer une série d'adresses. On utilisera

```
iptables -A INPUT -m recent --rcheck --name BANNED -j DROP
```

Er on peuplera la liste en écrivant "/" dans /proc/net/xt_recent/NOMLISTE pour la vider suivi de "+ip numérique" pour ajouter des entrées.

NAT

Pour effectuer du NAT, on se placera en POSTROUTING pour du Source NAT et en PREROUTING pour du destination NAT.

On utilisera les cibles

SNAT	Change l'adresse source en celle précisée par --to-source *adresse IP*.
DNAT	Change l'adresse destination en celle précisée par --to-destination . On peut préciser un port en le séparant par un :
MASQUERADE	Effectue du SNAT en utilisant l'adresse de la carte de sortie.
REDIRECT	Redirige vers la machine avec le port précisé dans --to-port.

Pour que le NAT fonctionne, ne pas oublier

- conntracking
- ouvrir le chemin au niveau des règles FORWARD
- dans la table nat

Helpers

Pour les protocoles qui utilisent des trafics "RELATED", un helper doit être actif pour marquer les connexions en question. Cela se fait avec la cible CT en PREROUTING dans la table raw (afin d'être pris en compte par les commandes suivantes)

```
iptables -A PREROUTING -t raw -p tcp --dport 21 -d 1.2.3.4 -j CT --helper ftp
```