2ème année 2014-2015

Le filtrage de niveau IP

Janvier 2015

Objectifs

Filtrage : Le filtrage permet de choisir un comportement à adopter vis à vis des différents paquets émis ou reçus par une station. Il permet donc de mettre en place des fonctionnalité de type *firewall*.

Nous observerons ensuite comment il peut être couplé avec des mécanismes équivalents au niveau applicatif.

1 Filtrage de type *firewall*

L'utilisation d'une connexion directe présente l'avantage de faciliter l'accès à l'information et la publication de l'information. Elle présente cependant également un gros inconvénient, celui de la sécurité.

En effet, s'il est possible d'accéder à l'information et plus généralement aux machines de façon "amicale", il est probablement tout aussi simple d'y accéder à des fins moins avouables !

Une connexion directe à Internet implique donc traditionnellement la mise en place de systèmes de sécurité et de veille. La plupart de ces systèmes est en général rassemblée dans un "firewall" (littéralement "mur pare-feu").

De par sa fonction de protection du réseau, un firewall doit avoir une position particulière dans ce réseau. Il se trouve en général sur le routeur d'entrée dans le réseau, comme dans la figure 1.

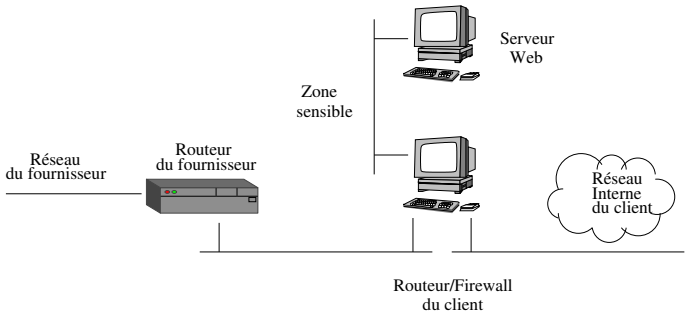

FIGURE 1 – Positionnement du firewall.

Le firewall se place donc entre Internet et le réseau interne à l'organisation. Cependant, les machines de ce réseau interne n'ayant pas toutes les mêmes contraintes de sécurité, elles sont généralement divisées en deux zones :

— le réseau interne, qui contient la plupart des stations de travail et qui doit donc être aussi isolée que possible d'Internet ;

— la zone démilitarisée (ou DMZ) qui contient les serveurs d'informations susceptibles d'être accédés depuis des stations situées ailleurs sur Internet.

Le système Linux intègre des fonctionnalités de firewall, certaines d'entre elles (celles relevant du filtrage IP) pouvant être mises en place grâce à la commande `iptables`.

1.1 Fonctionnement du firewall sous Linux

Le fonctionnement est basé sur des chaînes ; chaque chaîne étant une suite ordonnée de règles. Une règle définit une condition à évaluer sur chaque paquet (par exemple d'où il provient) et un comportement à adopter pour les paquets qui vérifient cette condition (les refuser par exemple).

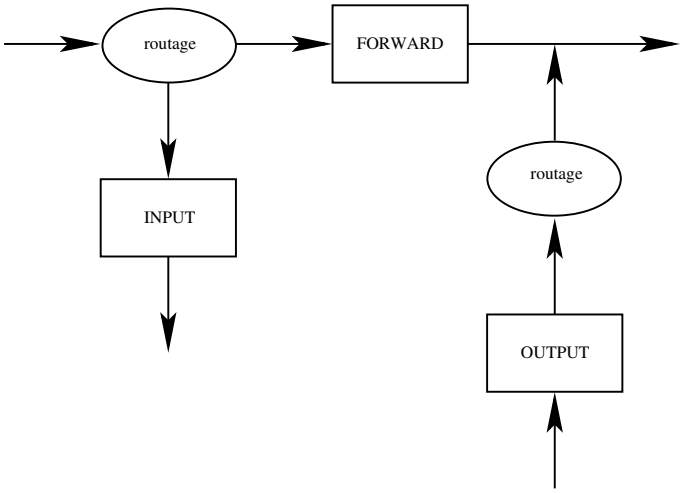

FIGURE 2 – Les chaînes du filtrage.

Il existe au minimum trois chaînes liées au filtrage, illustrées par la figure 2 :

— la chaîne d'entrée (`INPUT`), qui concerne tous les paquets entrants dont le firewall est le destinataire final ;
— la chaîne de sortie (`OUTPUT`), pour chaque paquet généré par le firewall ;
— la chaîne de réexpédition (`FORWARD`), pour les paquets que le firewall doit faire suivre à une autre station.

L'administrateur peut manipuler les chaînes par la commande `iptables`. Cette commande admet de nombreuses options que nous ne décrirons pas ici (voir son manuel en ligne). Voyons cependant quelques exemples.

Pour ajouter une nouvelle règle à la fin d'une chaîne, on utilise par exemple :

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

Ici on refuse (`DROP`) tous les paquets entrants (`INPUT`) en provenance (`-s`) de l'adresse 127.0.0.1 pour le protocole (`-p`) ICMP et à destination de la machine locale.

On peut alors observer l'état des chaînes :

```
# iptables -L
Chain INPUT (policy ACCEPT):
target     prot   opt      source     destination ports
DROP       icmp   ------   localhost  anywhere    any ->   any
```

```
Chain FORWARD (policy ACCEPT):
Chain OUTPUT (policy ACCEPT):
```

puis on peut détruire la première règle de la chaîne `input` de la façon suivante :

```
# iptables -D INPUT 1
```

1.2 Définition de règles

La mise en place de mécanismes de sécurité avec un firewall Linux consiste donc en l'écriture de règles rassemblées dans différentes chaînes.

▷ **Exercice 1 : Test des règles**

Expérimentez la commande `iptables` *en créant des règles et en vérifiant leur effet. Évitez d'utiliser trop de règles simultanément afin de bien maîtriser ce que vous faîtes.* ■

1.3 Choix d'une politique de sécurité

La définition puis la mise en place d'une véritable politique de sécurité dépassent la responsabilité de l'administrateur réseau. La politique doit en effet être définie (ou au moins validée) par les instances dirigeantes car elle doit correspondre aux besoins, aux risques et aux moyens de l'organisation.

La mise en place d'un ensemble de chaînes et de règles ne se fait donc pas "au jour le jour" et doit être planifiée dans sa globalité. L'efficacité de chaque règle peut en effet être remise en cause par une autre règle.

▷ **Exercice 2 : Développement d'une politique**

Définissez une politique cohérente (mais simple). Par exemple on interdira tous les accès sortants et entrants sur le réseau interne, mais on autorisera tous les accès Web qui en sortent. On autorisera en revanche tous les accès vers et depuis la DMZ *sauf les accès telnet (sauf venant du réseau interne).*

Mettez alors en place les règles nécessaires et vérifiez le bon fonctionnement. ■

2 Filtrage *via* une passerelle applicative

Tout comme pour une liaison par réseau commuté, il peut être intéressant de mettre en œuvre des mécanismes permettant d'optimiser l'utilisation de la bande passante de la connexion à Internet.

De tels mécanismes nécessitent généralement une souplesse ne pouvant être fournie que par une passerelle applicative.

2.1 Proxy cache

La méthode traditionnelle est d'utiliser un proxy cache, c'est à dire un proxy (tel que vu dans le TP n° 2) qui conserve les fichiers dans un cache disque local de façon à pouvoir les fournir plus rapidement (et sans utiliser la connexion à Internet) lors d'une prochaine requête.

Le serveur Web Apache intègre ces fonctions et permet donc une gestion centralisée des services Web, proxy et cache. Il est cependant souvent préférable, en particulier à des fins de souplesse d'administration et d'efficacité, de préférer utiliser un outil dédié tel que *Squid*. Il s'agit d'un proxy cache applicatif intégrant les protocoles HTTP, HTTPS et FTP.

▷ **Exercice 3 : Mise en place du cache**

Lancez l'utilitaire Squid (via la commande `/etc/init.d/squid start`*) et assurez-vous de son bon fonctionnement.*

Sa configuration est faite par le fichier `/etc/quid/squid.conf`*. Notez qu'il est configuré pour écouter sur le port 3128.* ■

L'efficacité d'un cache repose sur son utilisation (!) ; il est donc important dans une organisation utilisant un cache de s'assurer que tous les postes de travail passent par le proxy cache. Dans une grosse organisation, ceci n'est pas possible ; l'administrateur réseau doit donc procéder autrement.

Une solution est fournie par le firewall qui peut interdire les requêtes Web sortantes. Une station que l'on aurait alors oublié de configurer pour utiliser le proxy ne pourrait pas accéder à Internet.

▷ **Exercice 4 : Forcer l'utilisation du cache**

Changer votre politique de sécurité pour y intégrer une contrainte empéchant les utilisateurs du réseau interne d'accéder directement aux serveur Web d'Internet. ■

Ajoutons que l'utilisation conjointe d'une telle pratique et des possibilités de filtrage offertes par certains proxy permet également de s'assurer que les pages Web consultées par les personnes de l'organisation sont "politiquement correctes".

2.2 Proxy transparent

L'utilisation obligatoire d'un proxy nécessite tout de même la configuration (et la maintenance en cas de changement) des stations vis-à-vis de ce proxy.

La notion de proxy transparent permet de supprimer toute cette configuration et donc de rendre parfaitement transparent à l'utilisateur final l'existence même du proxy.

Linux intègre une partie de cette notion dans son "module" de gestion du firewall. En effet le comportement `REDIRECT` (qui peut être associé à n'importe quelle règle, comme tout autre comportement, mais uniquement sur les chaînes `PREROUTING` ou `OUTPUT` dans la table `nat`) signifie que tout paquet validé par la condition liée à la règle sera redirigé vers un port de la machine locale.

Par exemple

 # iptables -t nat -A PREROUTING -s 147.127.0.0/16 -p tcp --destination-port www -j REDIRECT --to-ports 8000

Cette commande ajoute une règle à la fin de la chaîne d'entrée spécifiant que tout paquet provenant du réseau de classe B 147.127 à destination d'un serveur Web doit être redirigé vers le port 8000 du firewall.

Si la machine hébergeant le firewall offre un proxy qui fonctionne sur le port 8000, alors toutes les requêtes seront traitées par ce proxy.

▷ **Exercice 5 : Automatiser l'utilisation du proxy**

Réalisez la configuration nécessaire pour forcer et automatiser l'utilisation le votre proxy squid

par l'ensemble des clients du réseau. ■

Attention cependant au fait que le proxy en question doit supporter le proxy transparent. Ce n'est pas le cas, à l'heure actuelle, du serveur Apache, mais un petit utilitaire, nommé `tproxy` permet de contourner ce problème :

```
# tproxy -s tproxy-port proxy-host proxy-port
```