Filtrage de paquets

Objectif

Utiliser et maîtriser le filtrage de paquets via *Iptables*. Assurer la protection d'un système des accès non autorisés en créant des nouvelles règles de filtrage.

Description de l'outil

Le noyau Linux comporte des systèmes de filtrage IP. Plusieurs systèmes se sont succédés au fil du temps: *ipfwadm*, *ipchains* et *iptables*. Pour ce TP, nous utiliserons le dernier en date : *iptables*, appelé aussi *netfilter*.

Netfilter/Iptables est un pare-feu Linux libre qui assure la mise en place, la maintenance et l'inspection des règles de filtrage (Pour plus de détails, cliquez ici).

Remarque : Netfilter/Iptables utilise 3 tables principales : FILTER, NAT et MANGE. Dans ce TP, nous travaillerons uniquement sur la table de filtrage (FILTER).

Pour la réalisation de ce TP, vous disposer de la plateforme suivante émulée par PC (VMware).

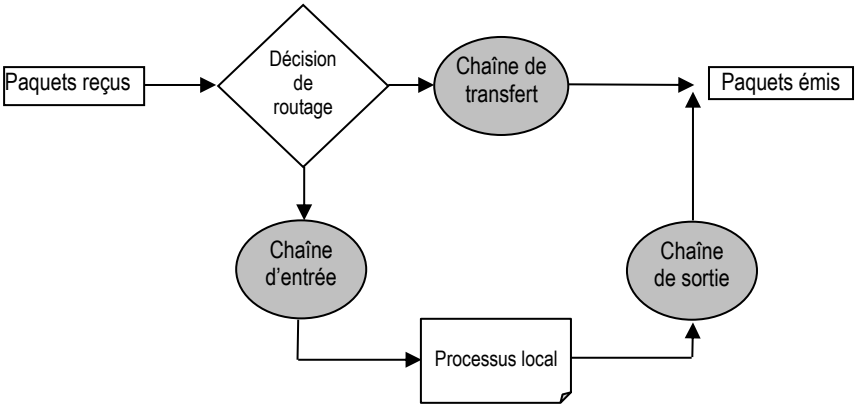
Cette plateforme est constituée d'un serveur sur lequel est installé un pare-feu et d'un poste de travail qui servira pour tester les différentes règles de filtrage implémentées dans le pare-feu.

Les chaînes et les règles de filtrage

Les règles de filtrage sont organisées en chaînes. Chaque chaîne est une succession de règles. Il y a trois chaînes prédéfinies dans la table de filtrage :

- La chaîne d'entrée (*input*) : Elle filtre les paquets IP destinés à des processus de la machine elle-même ;
- La chaîne de transfert (*forward*) : Elle filtre les paquets IP destinés à une autre machine ;
- La chaîne de sortie (*output*) : Elle filtre les paquets émis par les processus de la machine elle-même.

A chaque fois qu'un paquet arrive au pare-feu, il est orienté selon certains paramètres vers la chaîne concernée (voir schéma suivant).



Initialement, ces chaînes sont vides. Il est possible alors de définir des règles de filtrage pour chaque chaîne qui permettent de prendre une décision face à tout paquet qui la traverse. Chaque règle est un ensemble de conditions, qui lorsqu'elles sont vérifiées, vont aboutir à une décision pour le paquet traité. Les décisions principales sont:

- ACCEPT: le paquet est autorisé à traverser la chaîne
- DROP: le paquet est jeté
- REJECT: le paquet est jeté et sa source est prévenue par un paquet ICMP
- LOG : Le paquet est journalisé

Les règles sont examinées séquentiellement et la première dont les conditions correspondent au paquet traité est appliquée.

La manipulation des chaînes et des règles de filtrage s'effectue à travers la commande *iptables*. Voici les options de cette commande qui vous seront utiles lors des prochaines manipulations (Pour plus de détails, veuillez consulter le manuel d'utilisation d'iptables) :

-L : affiche la liste des règles d'une ou de toutes les chaînes existantes
-F : efface toutes les règles d'une chaîne existante
-D : efface une règle particulière d'une chaîne
-N : crée une nouvelle chaîne
-X : supprime une chaîne vide
-P : attribue la politique par défaut (ACCEPT ou DROP) à une chaîne
-A : ajoute une nouvelle règle à la fin d'une chaîne
-I : insère une nouvelle règle à une position donnée dans une chaîne

Politique de filtrage par défaut

Pour chaque chaîne, il est possible de choisir entre deux politiques de filtrage :

- *Tout ce qui n'est pas explicitement autorisé est interdit*. Les règles de filtrage de la chaîne consistent alors à laisser passer ce qui est autorisé (ACCEPT).
- *Tout ce qui n'est pas explicitement interdit est autorisé*. Les règles de filtrage de la chaîne consistent alors à bloquer ce qui est interdit (DROP ou REJECT).

1. Lancer la commande *iptables -L* puis observer la sortie d'écran.
2. Quelle est la politique adoptée par chacune des chaînes prédéfinies ?

Exploitation des règles de filtrage

1. Créer sur le poste Serveur un répertoire nommé « *firewall* ».
2. Copier dans ce répertoire le fichier *regles-filtrage.tar.gz* puis décompacter le avec la commande « *tar xvzf regles-filtrage.tar.gz* ». Vous devez obtenir les fichiers scripts suivants : *deny-all*, *accept-all*, *block-ip* et *unblock-ip*.
3. Rendre ces fichiers exécutables (droits d'accès 700).
4. Exécuter le script *deny-all* en tapant « *./deny-all* ». Essayer de faire un ping localhost et 192.168.10.254. Que constatez-vous ? Décrivez les règles utilisées par ce script.
5. Editer le script *deny-all* puis décommenter les lignes suivantes :

```
#INTERFACE_LOOPBACK=lo

#iptables -A OUTPUT –o $INTERFACE_LOOPBACK –j ACCEPT
#iptables -A INPUT –i $INTERFACE_LOOPBACK –j ACCEPT
```

6. Exécuter le script obtenu puis ressayer à nouveau les deux commandes Ping précédentes. Que constatez-vous cette fois ci ? Expliquez.
7. Afficher les informations concernant les paquets examinés par le pare-feu en utilisant la commande « *iptables –L -v*». Commenter la sortie d'écran.
8. Exécuter le script *accept-all* en tapant « *./accept-all* » puis essayer de faire un ping localhost et 192.168.10.254. Que constatez-vous ? Décrivez les règles utilisées par ce script.
9. Exécuter le script *block-ip* en tapant « *./block-ip 192.168.10.254* ». Essayer maintenant de faire un ping localhost et 192.168.10.254. Que constatez-vous ? Décrivez les règles utilisées par ce script.
10. Exécuter le script *unblock-ip* en tapant « *./unblock-ip 192.168.10.254* » puis essayer de faire un ping localhost et 192.168.10.254. Que constatez-vous ? Décrivez les règles utilisées par ce script.

Création de nouvelles règles de filtrage

Dans cette partie, il vous est demandé de créer des nouvelles règles de filtrage permettant d'assurer des exigences de sécurité particulières. Pour chaque règle, vous devez :

- commencer par effacer toutes les règles existantes dans les chaînes prédéfinies puis s'assurer qu'aucune règle n'est appliquée. Ensuite, vous devez choisir puis appliquer une politique de filtrage par défaut.
- inclure dans le rapport, une description détaillée de toutes les règles de filtrage demandées, ainsi que la description du test réalisé sur chacune d'elles, à savoir la sortie d'écran des commandes utilisées pour le test et/ou la capture d'écran d'un sniffer (Ethereal par exemple)

Voici les exigences de sécurité que doit assurer le pare-feu :

1. Interdire un paquet s'il ne provient pas de localhost
2. Interdire le protocole ICMP à destination de localhost
3. Interdire tout paquet à destination du port Telnet
4. Interdire tout paquet sortant par eth0 dont le numéro du port destination est inférieur à 1024
5. Interdire toute tentative d'initialisation de connexion TCP provenant de eth0
6. Interdire toute réponse à un Ping
7. Interdire tout paquet entrant par eth0 dont l'adresse mac n'est pas celle du poste de travail (voir le schéma à la page 1)
   Remarque : Attention, vous ne pouvez utiliser le filtrage par adresse mac que sur la table INPUT
8. Interdire les paquets provenant du sous-réseau local 192.168.10.0/24 sauf ceux en provenance du poste de travail.
   Remarque : Vous devez utiliser, dans ce cas, deux règles de filtrage.
   Appliquer ces règles puis essayer d'établir depuis le poste de travail, une connexion FTP vers le pare-feu. Inversez l'ordre de ces deux règles puis réessayez l'opération. Que remarquez-vous ? Quelle conclusion pouvez-vous en tirer ?
9. Écrire une règle qui laisse passer 5 tentatives de connexion TCP avec une fréquence de 2 tentatives par minute.
10. Créer une nouvelle chaîne qui journalise puis rejette tout paquet qui la traverse. Les paquets journalisés doivent être précédés par le préfixe [FIREWALL DROP]. Renvoyer ensuite sur cette nouvelle chaîne tout paquet entrant qui demande l'établissement d'une nouvelle connexion.

Vous avez certainement remarqué que pour toutes les règles de filtrage créer précédemment, la décision de laisser passer ou de bloquer un paquet est prise uniquement sur la base des informations contenues dans le paquet lui-même. On parle dans ce cas de filtrage sans état (*stateless*). Netfilter/Iptables met en oeuvre un mécanisme de suivi de connexions (option *--state*) lui permettant d'assurer également un filtrage avec état (statefull). Dans ce cas, la décision de laisser passer ou de bloquer un paquet prend également en compte les paquets déjà parvenus au pare-feu.

Les connexions suivies par le noyau Linux peuvent être dans l'un des états suivants:

- NEW: le paquet est le premier d'une connexion (ex: paquet TCP SYN)
- ESTABLISHED: le paquet fait partie d'une connexion déjà établie
- RELATED: le paquet est en rapport avec une autre connexion déjà établie
- INVALID: paquet non identifié

11. Positionnez la politique de filtrage par défaut à DROP pour les trois chaînes prédéfinies
12. Autoriser tout paquet sortant relatif à une connexion déjà établie ou en rapport avec une connexion déjà établie
13. Interdire tout paquet sortant relatif à une connexion de type INVALID
14. Autoriser tout paquet créant une nouvelle connexion en entrée à destination du port 80
15. Utiliser le navigateur web du poste de travail pour accéder à l'adresse URL « *http://www.network.net/site* ». Que constatez-vous ?
16. A présent, essayer l'adresse URL suivante « *http://192.168.10.2/site* ». Que constatez-vous cette fois ci ? Un problème qui se pose. Lequel ?
17. Que doit-on faire pour le résoudre ?

Analyse d'un script de filtrage

Soit les règles de filtrage suivantes. Elles forment ensemble un script permettant de mettre en place une protection minimaliste d'un pare-feu.

```
iptables -F INPUT
iptables -F OUTPUT
iptables -F FORWARD
```

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

```
iptables -N log-and-drop
iptables -A log-and-drop -j LOG --log-prefix "drop "
iptables -A log-and-drop -j DROP
```

```
iptables -A FORWARD -p tcp --tcp-flags ALL ALL -j log-and-drop
iptables -A FORWARD -p tcp --tcp-flags ALL NONE -j log-and-drop
```

```
iptables -A FORWARD -i eth+ -s 224.0.0.0/4 -j log-and-drop
iptables -A FORWARD -i eth+ -s 192.168.0.0/16 -j log-and-drop
iptables -A FORWARD -i eth+ -s 172.16.0.0/12 -j log-and-drop
iptables -A FORWARD -i eth+ -s 10.0.0.0/8 -j log-and-drop
```

```
iptables -A FORWARD -m state --state INVALID -j log-and-drop
iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
```

```
EXT_IFACE="eth0"
DMZ_IFACE="eth1"
iptables -N ext-dmz
iptables -N dmz-ext
DMZ_ADDR=192.168.10.0/24
```

```
      iptables -A FORWARD -s $DMZ_ADDR -i $DMZ_IFACE -o $EXT_IFACE -j dmz-ext
      iptables -A FORWARD -o $DMZ_IFACE -j ext-dmz
      iptables -A FORWARD -j log-and-drop
```

```
      iptables -N icmp-accept
      iptables -A icmp-accept -p icmp --icmp-type destination-unreachable -j
ACCEPT
      iptables -A icmp-accept -p icmp --icmp-type source-quench -j ACCEPT
      iptables -A icmp-accept -p icmp --icmp-type time-exceeded -j ACCEPT
      iptables -A icmp-accept -p icmp --icmp-type echo-request -j ACCEPT
      iptables -A icmp-accept -p icmp --icmp-type echo-reply -j ACCEPT
      iptables -A icmp-accept -j log-and-drop
```

```
      iptables -A ext-dmz -p tcp --dport smtp -j ACCEPT
      iptables -A ext-dmz -p udp --dport domain -j ACCEPT
      iptables -A ext-dmz -p tcp --dport domain -j ACCEPT
      iptables -A ext-dmz -p tcp --dport www -j ACCEPT
      iptables -A ext-dmz -p tcp --dport https -j ACCEPT
      iptables -A ext-dmz -p tcp --dport ssh -j ACCEPT
      iptables -A ext-dmz -p icmp -j icmp-accept
      iptables -A ext-dmz -j log-and-drop
```

```
      iptables -A dmz-ext -p tcp --dport smtp -j ACCEPT
      iptables -A dmz-ext -p tcp --sport smtp -j ACCEPT
      iptables -A dmz-ext -p udp --dport domain -j ACCEPT
      iptables -A dmz-ext -p tcp --dport domain -j ACCEPT
      iptables -A dmz-ext -p tcp --dport www -j ACCEPT
      iptables -A dmz-ext -p tcp --dport https -j ACCEPT
      iptables -A dmz-ext -p tcp --dport telnet -j ACCEPT
      iptables -A dmz-ext -p icmp -j icmp-accept
      iptables -A dmz-ext -j log-and-drop
```

```
      iptables -N ext-frw
      iptables -N dmz-frw

      iptables -A INPUT -i $EXT_IFACE -j ext-frw
      iptables -A INPUT -i $DMZ_IFACE -j dmz-frw
```

```
      iptables -A ext-frw -p icmp -j icmp-accept
      iptables -A ext-frw -p tcp --dport ssh -j ACCEPT
      iptables -A ext-frw -p tcp --sport ssh -j ACCEPT
      iptables -A ext-frw -j log-and-drop
```

```
      iptables -A dmz-frw -p icmp -j icmp-accept
      iptables -A dmz-frw -j ACCEPT
```

1. Expliquer pour chaque partie, les règles utilisées et leur rôle.
2. Réaliser pour chaque partie les tests qui permettent d'en vérifier le bon fonctionnement. Inclure dans le rapport la sortie et/ou les captures d'écran des différentes commandes utilisées lors des tests.

3. Selon le script précédent, dire où se situe exactement le pare-feu dans l'architecture réseau qu'il protège puis donner une description générale de la protection qu'il permet d'assurer.