Institut National Universitaire Champollion	L2 Informatique UE Réseau 1 TP3 : Firewall

Consignes générales

- Un compte-rendu par binôme
- Justifiez vos réponses mais soyez concis

Objectifs

Le but de cette séance est de comprendre le fonctionnement d'un pare-feu..
Ce TP est à faire en binôme sur une machine Linux. Au besoin utilisez votre machine virtuelle sur le serveur utilisé en prog web ….

Introduction

Iptables est le module noyau de Linux gérant le filtrage des paquets. Couplé à l'outil NETFILTER, il constitue un bon pare-feu pour une station de travail ou un serveur.
Iptables est constitué de 5 tables de règles :
- *FILTER* : Table par défaut
- *NAT* : Table utilisée pour la mascarade lorsqu'un paquet crée une nouvelle connexion
- *MANGLE* : Table permettant de travailler sur la manipulation des paquets
- *SECURITY* : Table propre au contrôle d'accès
- *RAW* : Table contenant des exceptions aux règles de filtrage et permettant d'effectuer le suivi des connexions

Pour choisir la table à utiliser avec une commande iptables, il faut utiliser l'option -t.
Si l'option n'est pas spécifiée, on travaille par défaut sur la table *FILTER*.
- `iptables -t filter ...`

Chaque table va être composée de chaînes qui sont toutes composées d'une suite de règles. Il en existe 5 prédéfinies (*PRE-ROUTING*, *INPUT*, *FORWARD*, *OUTPUT*, *POST-ROUTING*) mais il est possible d'en créer autant que nécessaire. Ces chaînes peuvent donc être vues comme des tableaux de règles qui peuvent éventuellement être vide. En général, elles possèdent a minima une règle par défaut permettant soit d'autoriser soit d'interdire le passage de tous les paquets.
Sur un hôte terminal, la trame est reçue par la carte réseau qui en extrait le paquet IP et le fait suivre au noyau. Le paquet va ensuite traverser la chaîne *INPUT* en entrée et la chaîne *OUTPUT* en sortie. Celui-ci est ensuite transmis à la carte réseau de sortie.
Les chaînes propres au routage des paquets sont contenues dans la table *FILTER* qui permet donc de rediriger des paquets à la volée de faire du NAT etc. Les chaînes utilisées dans ce cas sont *INPUT*, *FORWARD* et *OUTPUT*. Les chaînes *PRE-ROUTING* et *POST-ROUTING* ne se retrouveront que dans la table *Mangle* et interviennent avant ou après la décision de routage.
Dans chaque chaîne le filtrage est défini sous la forme de règles, aboutissant à une action : *DROP*, *ACCEPT*, *LOG*, *REJECT*, *MASQUERADE*, *DNAT*, *SNAT*. . .
Quelques commandes générale utiles pour ce TP :
- `iptables -L` : permet d'afficher la liste des règles actives.

- iptables -X : permet de supprimer toutes les règles utilisateur
- iptables -P chaîne (DROP|ACCEPT) : permet de définir la politique par défaut à appliquer à la chaîne.
- iptables -D chaîne numéro : permet de supprimer la règle numéro de la chaîne.
- iptables -t chaîne -F : permet de supprimer toutes les règles d'une chaîne.
- iptables -A chaîne ... : permet d'ajouter une règle à la chaîne. Par défaut les règles sont ajoutées en tête de la chaîne.
- iptables -N nom : permet de créer une nouvelle chaîne.

Pour l'ajout ou la modification de nouvelles règles dans une chaîne voici quelques exemples :
- iptables -A INPUT -i eth0 -j ACCEPT : Ajoute une nouvelle règle dans la table FILTER à la chaîne INPUT qui autorise les paquets en provenance de l'interface eth0.
- iptables -A FORWARD -i eth0 -o eth0 -s 192.168.1.0/24 -d 192.168.2.1 -m state --state NEW, ESTABLISHED, RELATED -j ACCEPT : qui accepte de relayer sur l'interface eth0 tous les paquets en provenance du sous-réseau 192.168.1.0 vers la machine 192.168.2.1 toues les nouvelles connexions, les connexions déjà étalies ou les connexions liées à des connexions existantes.

La table des connexions existantes est accessible sur Linux grâce à l'outil conntrack (conntrack -L pour afficher toutes les connexions gérées par netfilter). Cet outil n'est pas disponible par défaut, il faut l'installer avec notre bon vieux apt-get.
Au cas où, s'il vous faut définir un serveur dns, vous ajouterez la ligne suivante à votre fichier interfaces :
dns-nameservers ip_du_serveur_dns1 ip_du_serveur_dns$_2$

utilisez sans restriction le site : https://doc.ubuntu-fr.org/iptables
(avec les liens en fin de doc)

Activités

Exercice 1 Coupure générale !

La première étape va consister à couper toutes les communications en provenance et à destination de votre machine.
a) Commencez par vérifier que vous arrivez à surfer normalement sur Firefox
b) Affichez la liste des règles en cours d'utilisation par iptables.
c) Ajoutez une règle dans la table *FILTER* bloquant tout le trafic en *OUTPUT*.
Vérifiez que votre trafic Web n'arrive plus à sortir.
d) Ajoutez maintenant une règle supprimant tout le trafic en entrée (*INPUT*).
e) Affichez à nouveau la base de règles.

Exercice 2 Remise en marche de quelques services

Vous êtes maintenant normalement aveugle et sourd à ce qui se passe sur le réseau. Nous allons donc essayer de remettre le trafic Web en circulation.
a) Modifiez les règles précédentes pour autoriser le trafic HTTP à destination et en provenance d'un port 80. On suppose qu'aucun serveur web ne tourne sur votre machine.
b) Est-ce que la seule règle sur le port 80 est suffisante ? Pourquoi ?
c) Est il possible d'effectuer des résolutions DNS? Testez avec la commande nslookup sur www.google.fr.

d) Après avoir pris en compte le DNS, vérifiez dans Firefox que vos règles fonctionnent.
e) Vérifiez à l'aide d'un ping avec votre voisin que les autres types de trafic ne fonctionnent pas.
f) Autorisez le trafic ICMP en entrée et en sortie puis testez à nouveau le ping
g) Autorisez les communications sur le port 22 pour remettre en service votre service SSH : quel est le protocole de couche transport associé ?
h) Pour le trafic HTTP, le mieux est de restreindre le trafic en entrée aux connexions déjà établies et de n'autoriser à sortir que les connexions nouvelles ou établies : comment implémenter cette restriction ?

Exercice 3 : Mise en place de la journalisation

Dans cet exercice nous allons journaliser (*Logger*) les paquets ICMP à destination de votre machine.
Pour cela nous allons nous intéresser à l'outil *syslog* et plus particulièrement à l'option -j LOG d'iptables.
a) Commencez par créer une nouvelle chaîne que nous appellerons *PING_DROP_LOG*.
Cette chaîne va contenir deux règles : l'une qui va *logger* tous les paquets reçus grâce à la chaîne *LOG*; l'autre qui va rejeter tous les paquets reçus. Attention, l'ordre est important.
b) Ajoutez maintenant une règle à la chaîne INPUT pour permettre de rejeter tous les paquets ICMP en entrée et de les *logger* :
iptables -A INPUT -p ICMP -s 172.18.8.0/24 -j PING_DROP_LOG
c) Visualisez la table INPUT pour vérifier que vos modifications ont bien été prises en compte.
d) Testez ensuite depuis votre machine hôte que la règle fonctionne.
e) Visualisez dans un autre terminal le contenu du fichier de log : tail -f /var/log/syslog
f) Comme vous pouvez le constater, le fichier *syslog* contient énormément d'information. Pour s'y retrouver, une solution consiste à ajouter un préfixe devant chaque message à l'aide de l'option –logprefix.