Allant être déconnectés du réseau de l'IUT, il vous faut sauvegarder ce fichier ainsi que les fichiers de scripts sur la machine passerelle.

Ce sujet de TP repose sur deux articles :

- http://olivieraj.free.fr/
- http://www.shorewall.net/

Introduction

Un firewall ou Pare-feu ou garde barrière est un dispositif informatique qui permet le passage sélectif des flux d'information entre un réseau interne et un réseau public. Il permet de protéger le réseau interne d'intrusions venant du réseau public et de limiter les accès depuis le réseau interne vers le réseau public.

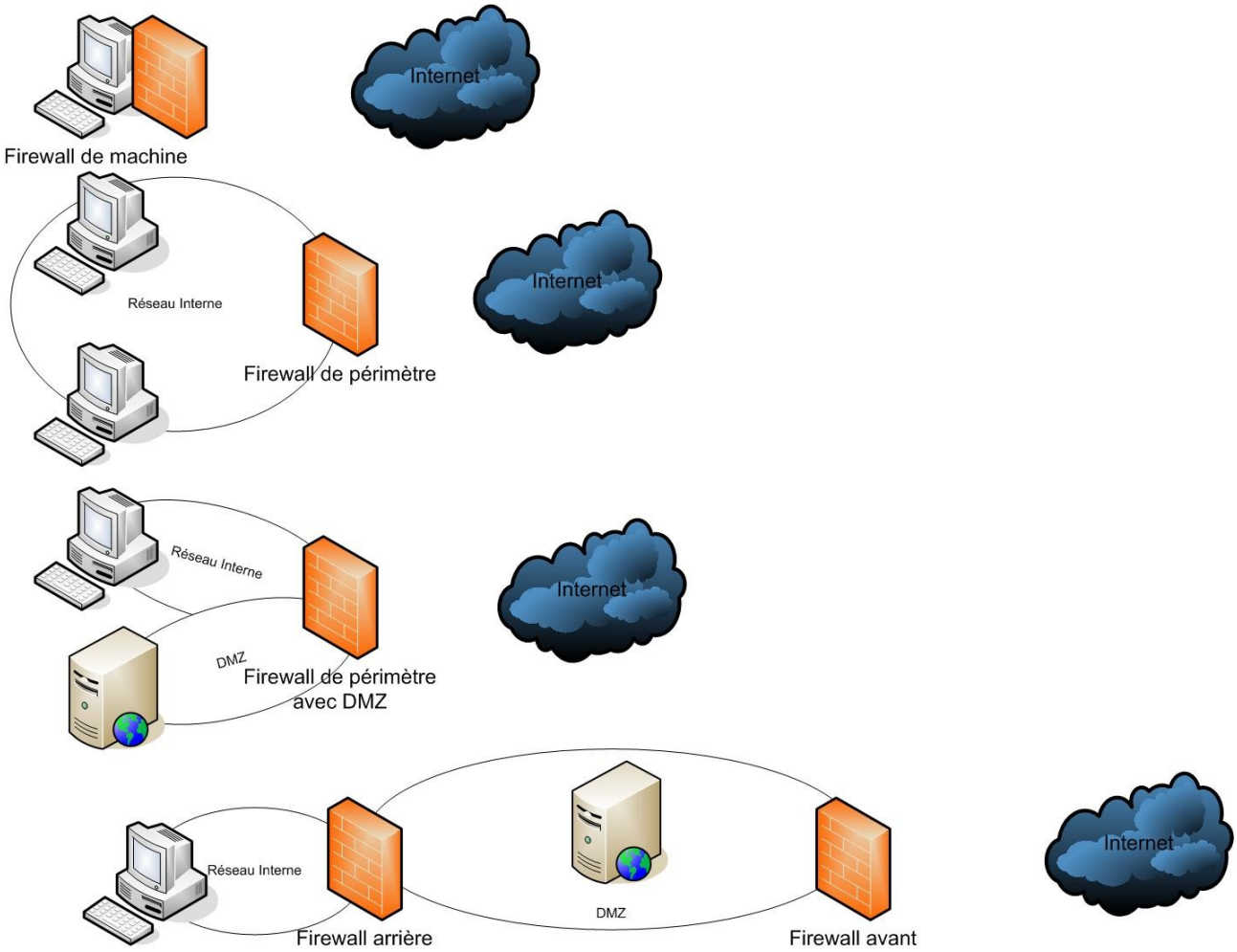
Les firewalls peuvent être logiciels ou matériels. Nous utiliserons aujourd'hui le firewall logiciel de linux : netfilter.

Ce firewall offre pour avantage par rapport aux firewall logiciel de Windows de s'éxécuter en mode noyau à savoir qu'il ne peut être manipulé par une autre application lancée par un utilisateur non administrateur. Ce qui l'empêche, contrairement aux firewall sous windows d'être arrêté par une application malveillante.

Netfilter permet de réaliser un filtrage au niveau des adresses IP (couche réseaux) et au niveau des ports (couche transport).

La sécurisation d'une entreprise peut faire intervenir de nombreux firewalls voici quelques exemples :

- firewall de machine,
- firewall de périmètre,
- firewall de périmètre avec zone démilitarisée (DMZ : DeMilatarized Zone), une zone visible depuis le réseau externe,
- firewall avant,
- firewall arrière.



Aujourd'hui nous mettrons en oeuvre un simple firewall de périmètre sur une passerelle NAT.

Choisir les machines

Nous utiliserons trois machines :

- une sous Linux qui va servir de firewall, cette machine doit être impaire et posséder deux cartes ;
- une autre sous Linux, qui va être l'agresseur, le méchant, cette machine doit être paire et ne posséder qu'une carte ;
- enfin, une machine sous Windows qui peut-être soit paire, soit impaire.

Installation des logiciels

Avant de réaliser le câblage, sur la machine sous Linux, possédant deux carte réseaux (le firewall), vous devez installer :

- shorewall (un firewall) : *sudo apt-get install shorewall* (à taper dans une invite de commande)
- la documentation de shorewall : *sudo apt-get install shorewall-doc*

- un serveur apache : *sudo apt-get install php5*
- *Téléchargez le sujet de TP et le script.*

Sur la deuxième machine linux qui jouera le rôle de méchant, installer :

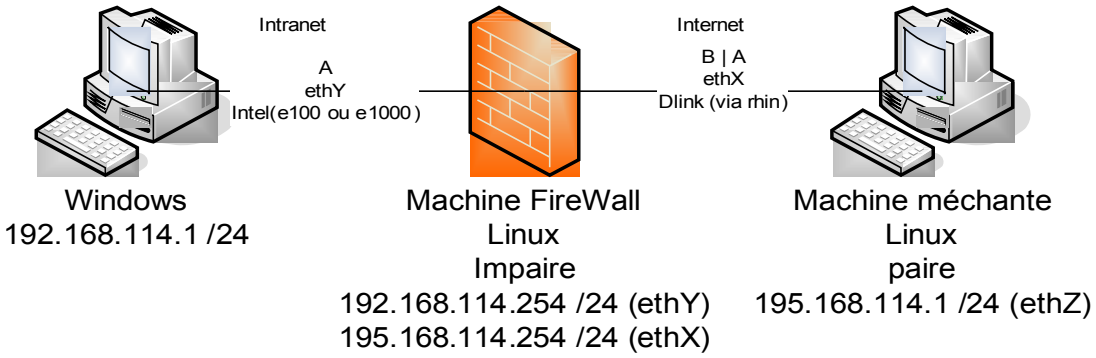
- nmap (un scanner de ports) : *sudo apt-get install nmap*
- un serveur apache : *sudo apt-get install php5*

Découverte des cartes réseaux

Pour découvrir les cartes réseaux de la passerelle, vous devez sur cette dernière :

- ouvrir un terminal
- devenir administrateur (root) : *sudo -s*
- obtenir la configuration ip : *ifconfig* (les cartes sont numérotées eth0, eth1, ...)
- connaître le drivers d'une carte : *ethtool -i ethX* (ou X est le numéro d'une carte découverte).

Plan de câblage et plan d'adressage



La machine interne 192.168.114.1 aura pour routeur par défaut 192.168.114.254.

La machine externe 195.168.114.1 n'aura pas de routeur par défaut.

Mettre en œuvre le plan de câblage et respectant le schéma précédent.

Affecter les adresse IP : System -> Administration -> Réseaux -> choisir une connexion -> cliquer sur propriétés -> mettre une adresse IP statique.

Remarque : pour forcer la mise à jour de la configuration réseaux : *sudo /etc/init.d/networking restart*

Une fois le plan de câblage mis en œuvre et les adresses IP positionnées tester la connectivité entre les machines de l'Intranet et les machines représentant Internet avec des *ping*.

La connectivité acquise activer le routage sur la machine passerelle avec la commande suivante :

echo 1 > /proc/sys/net/ipv4/ip_forward

Découverte d'une machine

Le ping en braodcast étant verrouillé sur les Ubuntu utiliser : *nmap -sP 195.168.114.1-254*

La machine découverte, nous pouvons nous intéresser à ses services avec un scan de port. Depuis la machine 195.168.114.1 lancer un scan sur la machine 195.168.114.254 avec la commande nmap.

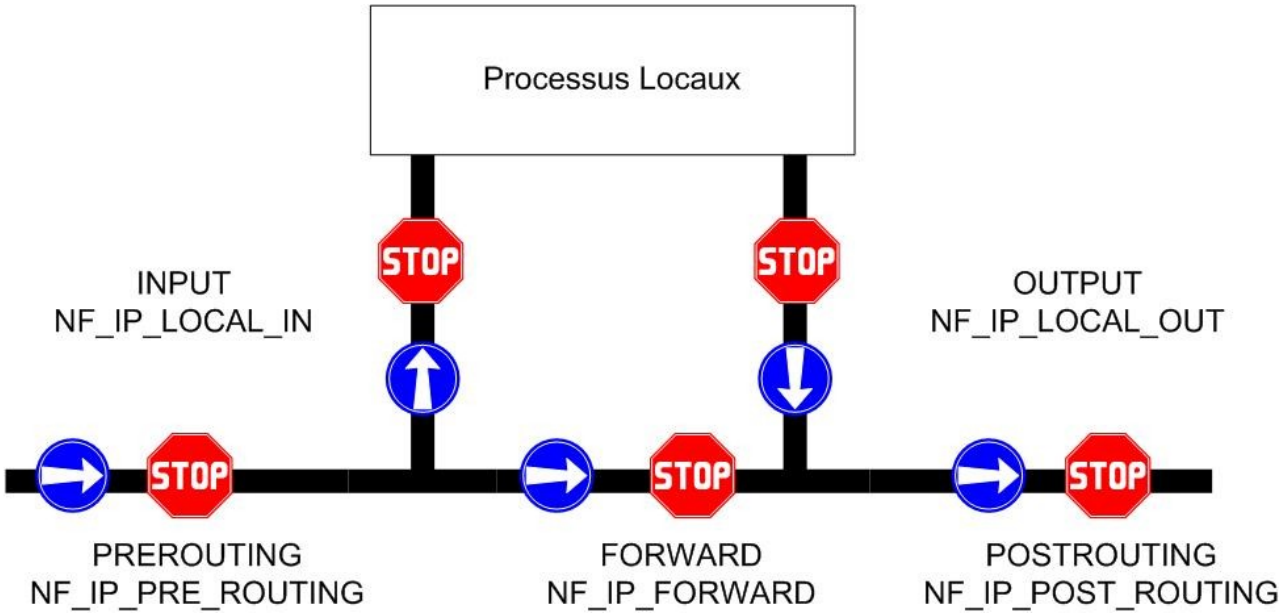
Normalement vous devriez trouver un serveur Web présent, celui que vous avez installé, mais qu'elle est sa version ? La commande *wget -d –no-proxy 195.168.114.254* peut vous aider.

Sachant que tout serveur possède des failles de sécurité et que les serveurs NFS (Network File System : Service sur un réseau ouvrant à un ordinateur l'accès à des fichiers sur une autre machine) ou SMB (Server Message Block : Protocole utilisé pour partager des ressources entre des réseaux Linux et Windows) sont particulièrement sensibles. Êtes vous satisfait de votre configuration ?

Présentation de netFilter

Un firewall étudie les paquets IP et choisit de les modifier, de les détruire ou de les laisser passer en fonction de leurs adresses IP et éventuellement des ports pour ceux qui transportent une entête de segment. Rappelez vous, suite à la fragmentation IP, un segment peut ne pas tenir en entier dans un paquet.

NetFilter fait parti du *Kernel* Linux sa tâche est de faire du filtrage de paquets réseaux. Appartenant au noyau, il n'est pas configurable avec un fichier de configuration, ni fourni avec une interface graphique, le seul moyen d'accès est la commande "*iptable*".



Les paquets peuvent suivre plusieurs chemins, soient-ils traversent la machine, soient ils vont ou partent des applications locales. Sur leur chemin vont se trouver des points de contrôle: des hooks (NF_IP_PRE_ROUTING, ...). A chaque point de contrôle le véhicule (le paquet IP) est inspecté puis soit modifié, soit détruit, soit laissé tel quel. Le choix est fait en fonction d'un ensemble de règles, les chaînes (PREROUTING, ...), qui exprime le comportement en fonction de l'origine, de la destination du véhicule, de sa taille, ...

Le hook :

- NF_IP_PRE_ROUTING reçoit le paquet brut de l'interface réseau,
- NF_IP_LOCAL_IN est le point de contrôle avant la remise aux couches applicatives,
- NF_IP_LOCAL_OUT est le point de contrôle qui voit passer les paquets générés par les applications,
- NF_IP_FORWARD est le point de contrôle pour les paquets qui sont simplement routés,
- NF_IP_POST_ROUTING est le point de contrôle qui précède la remise à l'interface de sortie.

Les chaines qui contiennent les règles sont manipulées au travers de tables :

- Filter (INPUT, OUTPUT, FORWARD) responsable du filtrage des paquets
- NAT (PREROUTING, OUTPUT, POSTROUTING) responsable de la translation d'adresse
- Mangle (Toutes les chaînes) qui va marquer les paquets pour assurer une Qualité de Service (QoS) en vous permettant par exemple de surfer pendant un gros téléchargement.

A ces chaînes (ensemble de règles) peuvent s'ajouter des chaînes utilisateur, de nouvelles check list pour le point de contrôle.

Les règles correspondent à un ensemble de critères, à ces régles sont associées des cibles, des actions a effectuer si la règle s'applique :

- DROP, le paquet est détruit
- ACCEPT, le paquet est accepté
- LOG/ULOG, le paquet est accepté mais est louche aussi se souvient ton de lui
- MASQUERADE, le paquet va être modifié
- MARK, le paquet est marqué
- Chaîne utilisateur, le paquet va suivre une nouvelle check liste définie par l'utilisateur.

L'outil IP table permet dont de rajouter, supprimer, modifier et afficher des règles et des chaînes.

La commande *iptables -I INPUT -p tcp -sport 80 -j DROP* permet de supprimer tous les paquets contenant un segment de port source 80 (http) à destination des applications.

Nous allons tester les commandes suivantes :

- suppression des chaînes existantes :
  - *iptables -t filter -F* //suppression des tables prédéfinies
  - *iptables -t filter -X* //suppression des tables utilisateur
- Définition de la politique (ensemble de règles) par défaut
  - *iptables -t filter -P INPUT DROP* //On interdit tout ce qui remonte
  - *iptables -t filter -P OUTPUT DROP* //On interdit tout ce qui descend
  - *iptables -t filter -P FORWARD DROP* //On interdit tout ce qui traverse

Essayer de pinguer depuis la passerelle une autre machine. Conclusion ?

Nous allons maintenant amoindrir notre belle configuration parfaite en autorisant certains trafics :

- Nous allons autoriser tous les trafics depuis notre machine vers elle même :
  - *iptables -t filter -A OUTPUT -o lo -s 127.0.0.0/8 -d 127.0.0.0/8 -j ACCEPT* //lo est l'interface de boucle locale et OUTPUT les paquets sortants. Pinguer 127.0.0.1 quel est le changement et pourquoi ?
  - *iptables -t filter -A INPUT -i lo -s 127.0.0.0/8 -d 127.0.0.0/8 -j ACCEPT* //INPUT les paquets entrants

Essayer un ping localhost, il doit marcher. Mais vous ne pouvez toujours pas pinguer les machines de votre réseau interne (192.168.114.1).

- Nous allons autoriser tous les trafics entre et notre passerelle et le réseau interne.
  - *iptables -t filter -A INPUT -i ethy -s 192.168.114.0/24 -d 192.168.114.0/24 -j ACCEPT*

- *iptables -t filter -A OUTPUT -o ethy -s 192.168.114.0/24 -d 192.168.114.0/24 -j ACCEPT*

Oh joie, oh merveille vous devez pouvoir pinguer 192.168.114.1. Mais pouvons nous accéder à Internet, toujours pas. Là c'est plus compliqué nous ne pouvons pas faire une confiance absolue au réseau Internet. Il faut être plus fin.

- Nous allons autoriser le trafic http
  - *iptables -t filter -A OUTPUT -o ethx -s 195.168.114.0/24 -d 0.0.0.0/0 -p tcp --dport 80 -j ACCEPT* //Pour sortir
  - *iptables -t filter -A INPUT -i ethx -s 0.0.0.0/0 -d 195.168.114.0/24 -p tcp --sport 80 -j ACCEPT* //Pour recevoir la réponse

Bon normalement cela devrait fonctionner. Tester avec l'url http://195.168.114.1. Et oui, ça ne marche pas nous avons oublié le parcours de nos paquets, ils partent d'un processus local (le client web) et sont donc soumis aux règles de l'interface locale qui bloque tout trafics en dehors de ceux sortant et venant de cette interface.

Le fichier *iptables-basic-2.sh* vous permet de corriger ce problème.

Somme nous content ? Depuis la machine internet un classique nmap 195.168.114.254 ne donne rien, mais il est possible d'utiliser d'autres options comme par exemple se faire passer pour un serveur Web (port 80). Exécuter la commande *nmap 195.168.114.254 -g 80* et constatez.

Pourquoi cela ne marche-t-il pas ?

Nous acceptons les paquets transportant un segment dont le port source et 80 mais nous n'avons pas demandé d'ouverture de connexion. Une solution existe, le module conntrak (suivie de connexion) permet de se souvenir des connexions initiées par notre volonté en imposant trois état :

- NEW, une demande d'ouverture de connexion
- ESTABLISHED : une connexion établie à notre demande
- RELATED : une connexion en liaison avec une connexion établie à notre demande (FTP, IRC et autre protocoles qui utilisent plusieurs ports dont l'un n'est pas connu à l'avance).
- INVALID : un paquet contenant un segment qui n'est pas associé à une connexion existante.

Avec ce module nous pouvons exprimer de nouvelle règles comme :

- iptables -t filter -A OUTPUT -o eth0 -s 195.168.114.254 -d 0.0.0.0 -p all -m state –state ! INVALID -j ACCEPT //accepter toutes les segments sortants valides
- iptables -t filter -A INPUT -i eth1 -s 0.0.0.0 -d 195.168.114.254 -p all -m state –state RELATED, ESTABLISHED -j ACCEPT //accepter toutes les segments entrants liés aux connexions initiées.

Nous pouvons continuer ainsi pour par exemple rendre notre firewall applicatif comme ZoneAlarm, NortonInternetSecurity et bien d'autres en activant le module ip_queue. Ainsi notre firewall identifiera les applications et non plus des ports et des adresses IP.

Tous cela devient vous l'avez compris bien compliqué. Dans la plus part des cas les fonctionnalités et les filtres attendus d'un firewall sont les mêmes aussi pourquoi ne pas utiliser un utilitaire de configuration comme shorewall ou une distribution Linux dédiée comme IPCop.

Mise en œuvre de shorewall

On abandonne l'utilisation directe des iptables et on utilise un utilitaires de configuration pour les piloter.

Sur la machine passerelle "shorewall" un service de configuration des iptables est installé.

Comme tout service Linux, ses fichiers de configuration se trouvent dans /etc.

Parmi les fichiers qui vous intéressent pour la suite vous trouverez :

- /etc/shorewall/shorewall.conf le fichier principal de configuration que nous n'utiliserons pas aujourd'hui. Si vous avez une liaison ADSL ou si vous êtes déjà caché derrière du NAT vous pourrez y jeter un coup d'œil .
- /etc/shorewall/zones les différentes zones de votre système : net (Internet), loc (l'Intranet). Une troisième zone existe mais n'est pas listée ici : fw (la machine firewall).
- /etc/shorewall/rules qui permet de définir des exceptions à une politique. Une règle est une action à exécuter lorsque des critères sont remplis. Par exemple :

#ACTION	SOURCE	DEST	PROTO	DEST PORT(S)
ACCEPT	net	fw	tcp	80

Ouvre le port 80 de votre firewall à la zone net (Internet). Ce fichier permet aussi de configurer le miroir et la translation de port (Destination NAT). Par exemple :

#ACTION	SOURCE	DEST	PROTO	DEST PORT(S)
DNAT	net	loc:192.168.114.1	tcp	80

Permet de cacher sur la machine 192.168.114.1 un serveur web visible sur l'adresse externe de la passerelle.

- /etc/shorewall/policy contient des politiques par défaut, si une exception n'est pas applicable alors la politique par défaut est appliquée. Une politique peut être par exemple :

#SOURCE	DEST	POLICY	LOG LEVEL	LIMIT:BURST
net	fw	DROP		

#SOURCE	DEST	POLICY	LOG LEVEL	LIMIT:BURST
Loc	net	ACCEPT		

Permet d'indiquer que l'on souhaite autoriser tout trafic depuis la zone loc (l'Intranet) vers toute destination et que l'on refuse tout trafic venant de la zone net (d'Internet) à destination du firewall. L'algorithme d'utilisation des règles et politiques est le suivant : *Pour chaque connexion demandant à entrer dans le firewall, la requête est en premier lieu comparée par rapport au fichier /etc/shorewall/rules. Si aucune règle dans ce fichier ne correspond à la demande de connexion alors la première politique dans le fichier /etc/shorewall/policy qui y correspond sera appliquée. Si cette politique est REJECT ou DROP la requête est dans un premier temps comparée aux règles contenues dans le fichier /etc/shorewall/common, si ce fichier existe; sinon les règles dans le fichier /etc/shorewall/common.def sont vérifiées.* Bref ici, du monde extérieur nous n'autorisons que le trafic Web.

- /etc/shorewall/interfaces
  Permet de spécifier d'associer des interfaces à des zones. Si vous avez respecté le plan de câblage, il n'y a rien à changer.
- /etc/shorewall/masq
  Permet de configurer l'IP masquerading et le SNAT. Vous allez devoir le bricoler.

La commande « /etc/init.d/shorewall restart » vous permet de relancer votre firewall après modifications.

Mettre en marche le firewall

Avant de vous lancer notre firewall, il nous faut placer et modifier quelques fichiers :

1. Recopier une configuration par défaut :
   cp /usr/share/doc/shorewall/examples/two-interfaces/ /etc/shorewall*
2. Dans */etc/default/shorewall* mettre *startup=1*
3. Dans */etc/shorewall.conf* mettre *SARTUP=Yes*
4. Tester si le firewall démarre : */etc/init.d/shorewall restart*

Il vous faudra après chaque modification relancer le firewall.

A faire :

1. Configurer les interfaces (*/etc/shorewall/interfaces*)
2. Mettre en œuvre le NAT (*/etc/shorewall/masq*) en rajoutant *ethx ethy* à l'endroit indiqué, puis tester votre configuration avec des pings et des consultations de page web.
3. autoriser tous les trafics depuis l'intranet vers internet et depuis le firewall vers les autres zones (*/etc/shorewall/policy*)
4. Mettre en œuvre le miroir de port pour cacher un serveur web sur la machine de l'Intranet (192.168.114.1) visible depuis Internet. Le fichier a modifier est rules. Je vous conseille d'utiliser le man de /etc/shorewall/rules et de regarder DNAT dans les exemples.

5. Restreindre l'accès à Internet aux seuls serveurs Web depuis l'Intranet. A vous de trouver la solution.