1 Objectifs

L'objectif de ce TP est de vous initier à certaines techniques dites de *défense* afin de vous donner quelques armes face aux dangers liés aux réseaux. Pour se faire, vous allez utiliser un environnement virtuel émulant un réseau de machines sous *Debian*[1], c'est à dire la distribution que vous utilisez actuellement. L'environnement virtuel que nous allons utiliser est *NEmu*[2].

2 Un peu de vocabulaire

Il existe plusieurs mécanismes de défense contre les attaques réseaux :
— IDS (*Intrusion Detection System*)
— IPS (*Intrusion Prevention System*)
— Honey Pot (*Pot de miel*)
— Firwall (*Pare-feu*)
— etc.

Ces systèmes sont tous utilisés de manière courante dans les réseaux d'entreprises, les réseaux académiques, les réseaux militaires, etc.

2.1 IDS

Un IDS est un système permettant de détecter, et de notifier, des attaques. **arpwatch**[3] en est un example basique dans la mesure où il ne détecte que des attaques de type ARP *cahe poisoning*.

2.2 IPS

Un IPS permet, en plus de la détection, d'appliquer une réponse à une attaque et donc de fournir une défense plus ou moins automatisée. Ces systèmes opèrent de manière *intelligente* dans la mesure où ils peuvent adapter leur comportement en fonction de l'historique du trafic réseau. Ces systèmes agissent en relais invisible. En d'autres termes ce sont des machines qui ne possèdent pas d'adresse, d'un point de vue réseau, et faisant office de passerelles.

2.3 Honey Pot

Un Honey Pot est un système servant d'appât pour pirates. Le concept le plus courant est de fournir de fausses informations au pirate lorsqu'il essaye de collecter des informations ou attaquer un système. Par exemple, fournir des informations sur les services totalement abracadabrantes lorsque le pirate essaye d'en obtenir la liste par **nmap**. Coupler à un IDS, l'administrateur peut ainsi analyser le comportement et les attaques courantes et parfaire sa protection.

2.4 Firewall

Un Firewall est un système appliquant des règles de filtrage explicites et pouvant éventuellement *dumper* le trafic et faire des statistiques. Il peut également servir à faire de la translation d'adresse.

Nous allons dans ce TP voir comment configurer un pare-feu standard du noyau Linux, massivement utilisé dans le monde. Ce pare-feu se nomme **NetFilter**[4].

1. http://www.debian.org
2. http://nemu.valab.net
3. http://ee.lbl.gov
4. http://www.netfilter.org

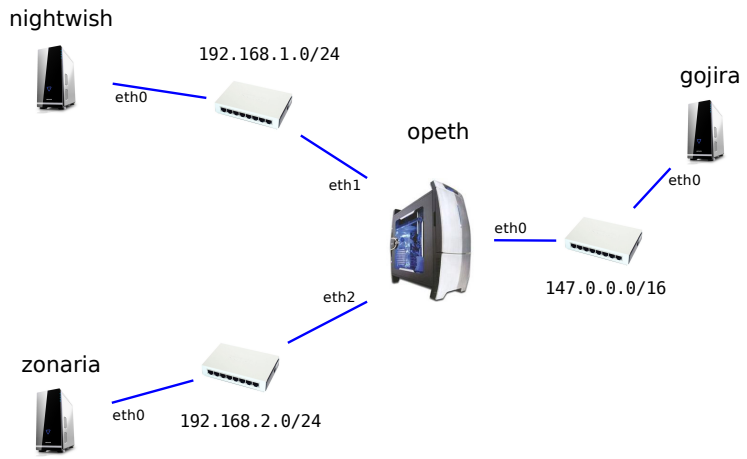
Vincent Autefage
autefage@labri.fr

3 Avant de commencer...

— Pour lancer le réseau virtuel :
```
$ source /net/ens/vince/virt/nemu-init.rc
$ nemu-kvm start
$ nemu-vnet /net/ens/vince/virt/config/netfw.py
```
— Pour quitter le réseau virtuel, tapez **quit()** dans le terminal principal.
— Pour sauvegarder le réseau virtuel, tapez **save()** et validez dans le terminal principal. Le réseau sera sauvegardé dans $\sim /netfw.tgz$.
— Pour redémarrer (violemment) le réseau virtuel, tapez **reboot** et validez dans le terminal principal.
— Pour restaurer le réseau virtuel précédemment sauvegardé :
```
$ nemu-restore ~/nefw.tgz
```
— Les éditeurs *jed*[5], *nano*[6] et *vi*[7] sont installés sur le système.

4 Le réseau

Nous allons travailler sur le réseau suivant :



Nous pouvons constater que ce réseau est composé de 4 machines inter-connectées par des switchs. La machine *opeth* est une passerelle sur laquelle nous allons configurer notre pare-feu. *nightwish* et *zonaria* font parties d'un même réseau local privé mais sont dans des sous-réseaux différents. *gojira* représente une machine externe à IP publique. Toutes les machines sont des terminaux utilisateurs standards tournant sous *Debian*. Les machines virtuelles vous sont livrées *nues*. C'est à dire qu'elles disposent uniquement des réglages élémentaires du système. Le mot de passe *root* est **plop**.

4.1 Amorçage du réseau

1) Lancez le réseau virtuel comme indiqué ci-dessus. Quatre fenêtres correspondant aux consoles de chacune des machines devraient apparaître.

4.2 Configuration générale

2) Qu'est ce qu'une IP privée ? Quelle est la différence avec une IP publique ?

3) Configurez toutes les interfaces réseaux à l'aide des commandes **ifconfig** et **route**.

5. http://www.jedsoft.org/jed
6. http://www.nano-editor.org
7. http://vim.sourceforge.net

Vincent Autefage
autefage@labri.fr

Rappels :

```
# ifconfig <iface> <@IP> [netmask <@NETMASK>]
# route -n
# route add default gw <@IP passerelle>
# route del default gw <@IP passerelle>
Exemple :
ifconfig eth0 192.168.0.1 netmask 255.255.255.0
route add default gw 192.168.0.1
```

4) Activez l'*IP forwarding* sur *opeth*.

Rappel :

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

5) Vérifiez que la configuration est effective à l'aide de la commande **ping**.

5 Principe et fonctionnement

NetFilter est un module noyau qui se manipule à l'aide de la commande **iptables**. Il dispose de plusieurs tables (modes) :

- *filter* : filtrage de paquets.
- *nat* : translation d'adresses.
- *mangle* : altération de l'entête des paquets.
- *raw* : marquage de paquets.
- *security* : contrôle d'accès mandataire.

Nous allons ici nous concentrer sur *nat* et *filter*.

5.1 DMZ

Une DMZ (*demilitarized zone*) est une zone particulière d'un réseau privé qui héberge des services publics (serveur HTTP, FTP, DNS, etc.). Une DMZ répond à un règlement de connexions strict :

1. Le trafic *externe* $\leftrightarrows$ DMZ est **autorisé**.
2. Le trafic *interne* $\rightarrow$ DMZ est **autorisé**.
3. Le trafic *interne* $\leftarrow$ DMZ est **interdit**.

5.2 Usages

```
Usage général :
      iptables {--table|-t} <table> [<action> <zone>] [ordres...]
Usages courants :
      . Voir table : iptables -t <table> {-L|--list} [-n|--line-numbers]
      . Vider table : iptables -t <table> {-F|--flush}
      . Règle défaut : iptables -t <table> {-P|--policy} <zone> [ordres...]
      . Ajouter règle : iptables -t <table> {-A|--append} <zone> <tests> {-j|--jump} <action>
Tests :
      . {-d|--destination} {<@IP>|<@IP réseau>}
      . {-s|--source} {<@IP>|<@IP réseau>}
      . {-i|--in-interface} <iface>
      . {-o|--out-interface} <iface>
      . {-p|--protocol} <protocole>
      . {--dport|--destination-port} <port>
      . {--sport|--source-port} <port>
```

Vincent Autefage
autefage@labri.fr

```
      . -m state --state <état>
États :
      . NEW
      . ESTABLISHED
      . INVALID
```

5.2.1 nat

```
Zones :
      . PREROUTING
      . POSTROUTING
Actions :
      . DNAT --to-destination <@IP>[:<port>]
      . SNAT --to-source <@IP>[:<port>]
      . MASQUERADE
```

5.2.2 filter

```
Zones :
      . INPUT
      . FORWARD
      . OUTPUT
Actions :
      . ACCEPT
      . DROP
      . REJECT
```

6 Protect it !

L'ensemble des règles seront effectuées sur la passerelle *opeth*.

6) Créez le fichier **/etc/init.d/fw** et donnez lui les droits d'exécutions à l'aide des commandes **touch** et **chmod**. Ce script devra être exécuté à chaque mortification de son contenu.

Rappels :

```
# touch <fichier>
# chmod <droits> <fichier>
Un script-shell commence par la ligne suivante :
#!/bin/bash
```

7) Mettre en place une politique d'effacement (*flush*) sur l'ensemble des tables de *nat* et *filter* en début de fichier.

8) Mettre en place des politiques *filter* par défaut qui *drop* l'ensemble du trafic.

9) Autorisez l'ensemble du trafic de *nightwish* vers *zonaria*.

10) Essayez d'effectuer un **ping** de *nightwish* vers *zonaria*. Cela marche-t-il ? Pourquoi ?

11) N'autorisez que le trafic déjà établi de *zonaria* vers *nightwish*.

12) Essayez d'effectuer un **ping** de *nightwish* vers *zonaria*. Cela marche-t-il mieux ?

13) Mettre en place un *nat* dynamique depuis le sous-réseau de *nightwish* vers l'extérieur.

14) Autorisez le trafic du sous-réseau de *nightwish* vers l'extérieur.

15) Mettre en place un *NAT statique* depuis l'extérieur vers *zonaria*.

16) Essayez d'effectuer un **ping** de *gojira* vers *zonaria*. Cela marche-t-il ? Pourquoi ?

17) Autorisez le trafic ICMP entre *zonaria* et l'extérieur.

18) Essayez d'effectuer un **ping** de *gojira* vers *zonaria*. Cela marche-t-il mieux ?

19) Autorisez le trafic HTTP entre *zonaria* et *gojira*.

20) Écrivez une page simple dans le répertoire */var/www* de *zonaria*.

21) Lancez le serveur web à l'aide du script suivant :

```
# /etc/init.d/lighttpd start
```

22) Essayez d'accéder au site web de *zonaria* depuis *gojira*.

23) Peut-on dire que *zonaria* répond aux critères d'une DMZ ?

24) Éteignez chaque machine correctement à l'aide de la commande **halt**.

25) Clôturez l'environnement virtuel à l'aide de la commande **quit()**.

Vincent Autefage
autefage@labri.fr