1 Objectifs

L'objectif de ce TP est de vous initier à certaines techniques dites de *proxy* afin de sécuriser une connexion réseau. Pour ce 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 Principe

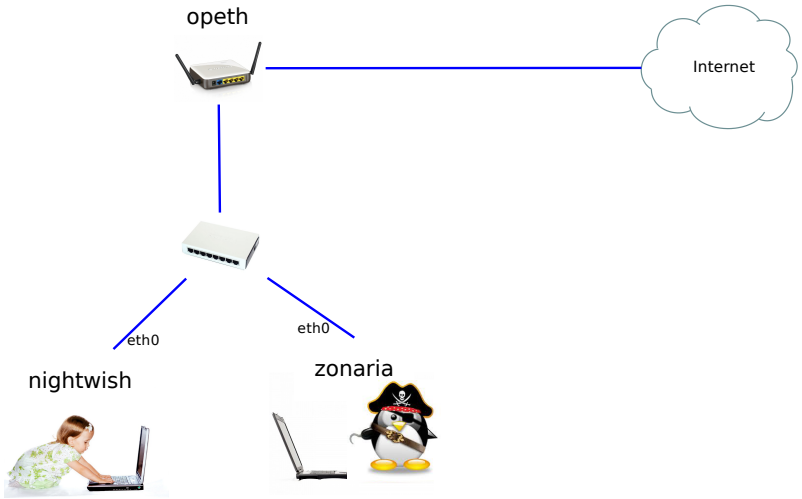
2.1 SSH

SSH[3] est un protocole de communication sécurisée pouvant autant fonctionner de manière symétrique qu'asymétrique. Originellement, le protocole SSH permettait uniquement d'accéder au *shell* d'une machine distante de manière cryptée. On peut maintenant utiliser SSH pour faire ce que l'on appel du *tunneling*, c'est à dire de l'encapsulation de paquets de protocole *a* dans d'autres paquets de protocole *b*. Avec SSH, les *super-paquets* sont cryptés. Pour réaliser cet exploit, le principe est de créer une *socket* SSH entre une machine et une passerelle, et d'envoyer les paquets devant être sécurisés (par exemple HTTP) au travers de cette *socket*. On appelle ce mécanisme *SOCKS*.

Le but de ce TP est de mettre en place un *proxy SSH* ayant pour but de sécuriser une connexion à l'aide d'un modèle de cryptage symétrique. Nous nous intéresserons à la configuration asymétrique dans un second temps.

3 Le réseau

Nous allons travailler sur le réseau suivant :



Nous pouvons constater que ce réseau est composé de 3 machines inter-connectées à un switch. La machine *opeth* est un routeur/passerelle de type *box* (comme la *freebox*, *neufbox*, etc.). *opeth* est reliée à internet sur une des ses interfaces. Les 2 autres 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**.

1. http://www.debian.org
2. https://gitlab.com/v-a/nemu
3. http://openssh.org

Vincent Autefage
vincent.autefage@u-bordeaux.fr

4 Avant de commencer...

- Dans un terminal régulier :
  - Pour lancer le réseau virtuel :
    `/net/ens/nemu/nemu-vnet netproxy`
  - Pour restaurer le réseau virtuel précédemment sauvegardé :
    `/net/ens/nemu/nemu-restore ~/vnet/netproxy.tgz`
- Dans le terminal de *NEmu* :
  - 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 `~/vnet/netproxy.tgz`
  - Pour redémarrer (violemment) l'ensemble du réseau virtuel, tapez **reboot()** et validez dans le terminal principal
  - Pour redémarrer une seule machine virtuelle : **RebootVNode('<nom de la VM>')**
- Dans le terminal de chaque machine virtuelle :
  - Vous êtes administrateur de la machine virtuelle : compte **root** et mot de passe **plop**
  - Vous disposez d'éditeurs de texte simples dans les machines virtuelles : *micro*, *nano* et *vim*
  - Démarrez l'interface graphique par **startx**
  - Redémarrez une machine virtuelle par **reboot**
  - Arrêtez proprement une machine virtuelle par **poweroff** ou **halt**
  - Souris piégée dans une machine virtuelle : **CTRL+ALT+G**
  - Le pavé numérique n'est pas directement disponible : taper 2 fois sur la touche **Verr Num** pour l'activer
  - L'affichage dans une fenêtre est trop long : **Shift+PageUp** ou **Shift+PageDown** pour naviguer

4.1 Amorçage du réseau

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

4.2 Configuration générale

2) Configurez les interfaces réseaux de *nightwish* et *zonaria* à l'aide de la commande **dhclient**.

Rappel :

```
# dhclient <iface>
```

3) Vérifiez que la configuration est effective à l'aide des commandes **ifconfig**, **route** et **ping**.

Rappels :

```
# ifconfig <iface>
# route -n
# ping <@IP>
```

4) Enregistrer la configuration de manière perenne dans `/etc/network/interfaces` :

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp
```

5) Tentez d'effectuer la commande suivante (sur *nightwish* et *zonaria*) pour vérifier que l'accès à internet fonctionne.

```
# wget www.labri.fr
```

Vincent Autefage
vincent.autefage@u-bordeaux.fr

5 Proxy sécurisé

5.1 Configuration symétrique du proxy

L'intérêt de ce système est qu'il nécessite seulement un serveur SSH ainsi qu'un compte utilisateur standard sur la passerelle. Nous allons ici réaliser ce tunnel sécurisé de *nightwish* vers *opeth*.

6) Passez en mode graphique sur *nightwish*.

7) Un compte utilisateur nommé *toto* existe déjà sur *opeth*. Son mot de passe est également **toto**. Essayez de vous connectez sur ce compte depuis *nightwish* à l'aide de la commande suivante.

```
# ssh -D 127.0.0.1:8080 -N <username>@<@IP opeth>
```

8) Configurez le proxy du navigateur de *nightwish* pour qu'il utilise ce tunnel (*google* est encore votre ami).

9) Essayez d'accéder à site web depuis *nightwish*.

5.2 Tentative d'espionnage

5.2.1 Rappels sur ARP

Lorsque une machine fait une requête à une autre machine sur un même réseau local, la machine appelante effectue une requête ARP de manière à acquérir l'adresse physique (appelée aussi MAC) de la machine qu'elle cherche à joindre.

10) La table de correspondance entre adresse IP et MAC peut être consultée sur une machine grâce à la commande suivante.

```
# arp -n
```

5.2.2 Rappels sur l'attaque *Man In The Middle*

Les requêtes (*broadcast*) et réponses (*unicast*) ARP permettent de faire le lien ; le principe est donc de *flooder*, c'est à dire inonder la victime (ici *nightwish*), de réponses ARP de manière à lui faire croire que l'adresse IP de la passerelle (ici *opeth*) qu'il souhaite contacter correspond à notre machine pirate (ici *zonaria*). Il faut ensuite transmettre les messages IP de la victime à la véritable passerelle. De cette manière, notre machine pirate jouera le rôle de relais entre la victime et l'extérieur. Nous pourrons ainsi espionner toutes ses communications. Ce type d'attaque par vole d'identité s'appelle du *spoofing*. La technique qui consiste à s'insérer dans une communication en tant que relais intermédiaire illicite s'appelle l'attaque de *l'homme du milieu* ou *Man In The Midlle* (MITM).

```
# arpspoof -t <@IP victime> <@IP vraie passerelle>
```

5.2.3 À l'abordage !

11) Mettre en place une procédure de *man in the middle* de *zonaria* sur *nightwish* avec **arpspoof** et **wireshark**.

12) Essayez d'accéder à un site web depuis *nightwish*.

13) Essayez d'espionner le contenu de la communication entre *nighwtsih* et *opeth* depuis *zonaria* ?

14) Cela fonctionne-t-il ?

15) Stoppez la procédure d'espionnage sur *zonaria*.

Vincent Autefage
vincent.autefage@u-bordeaux.fr

5.3 Configuration asymétrique du proxy

Le principe est d'utiliser une paire de clés à la place d'un mot de passe. Cela permet entre autre d'établir des communications de manière totalement automatisée sans intervention de l'utilisateur. Nous allons ici réaliser ce tunnel asymétrique sécurisé de *nightwish* vers *opeth*.

16) Rendez vous dans le répertoire *.ssh* du *home* de votre utilisateur sur *nightwish*.

17) Créez votre couple de clé SSH à l'aide de la commande suivante.

```
$ ssh-keygen -b 2048 -t rsa
```

18) Publiez la **clé publique** sur *opeth*.

Rappel :

```
# ssh-copy-id -i <clé publique> <login>@<server>
```

19) Essayez de vous connecter sur ce nouveau compte depuis *nightwish* à l'aide de la commande suivante.

```
# ssh -D 127.0.0.1:8080 -N <username>@<opeth>
```

Le mot de passe n'est maintenant plus demandé.

20) Configurez le proxy du navigateur de *nightwish* pour qu'il utilise ce tunnel.

21) Essayez d'accéder à un site web depuis *nightwish*.

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

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

Vincent Autefage
vincent.autefage@u-bordeaux.fr