IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

TP - Réseaux privés virtuels

FIP IR312

Automne 2018

1 Le réseau émulé

1.1 Netkit

Netkit permet de simuler un réseau sur un PC. Il repose sur «User Mode Linux» qui permet de lancer un noyau Linux comme une application utilisateur et de créer ainsi une machine virtuelle. Les machines virtuelles sont reliées entre elle par des zones de collision se comportant comme un hub Ethernet. Chaque machine a son propre système de fichier et peut utiliser son propre noyau. Il s'agit donc d'un réseau de machines hétérogènes. Chaque machine est contrôlée par l'utilisateur (`root`) via une console de contrôle (ici un terminal `xterm`).

On peut simuler des réseaux assez importants avec netkit car la mémoire est très efficacement utilisée. Dans le cas qui nous intéresse, nous nous contenterons de 5 machines. La description du réseau constitue un laboratoire. Elle est entièrement contenue dans un répertoire (ici `~/lab-vpn`). Les principales commandes de contrôle doivent être exécutées dans le répertoire de base du laboratoire.

— `lstart` permet de lancer les différentes machines ;
— `lhalt` permet de les arrêter ;
— `lclean` permet de nettoyer tous les fichiers temporaires en cas de problème.

Chaque machine donne accès aux répertoires `/hostlab` et `/hosthome` correspondant respectivement au répertoire de description du labo et au répertoire `${HOME}` sur la machine hôte.

1.2 Topologie du réseau de test

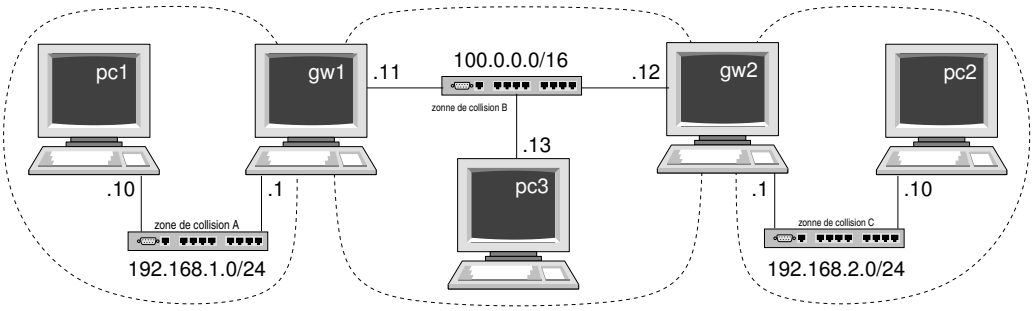

FIGURE 1 – Le réseau utilisé

La topologie du réseau est présentée sur la figure 1. Le réseau est constitué de trois sous réseaux :

— un réseau fédérateur `100.0.0.0/16` ;
— deux sous réseau privés (`192.168.1.0` et `192.168.2.0`) abritant chacun une machine (resp. `pc1` et `pc2`) et connectés au réseau fédérateur par une passerelle (resp. `gw1` et `gw2`) ;

— un PC isolé sur le réseau fédérateur (`pc3`).

Pour simplifier les choses, les adresses sont fixes et on n'a pas activé la traduction d'adresse (NAT) sur les passerelles. On souhaite relier les deux réseaux privés entre eux par un tunnel (configuration réseau à réseau).

Lancez le réseau avec `lstart`. Prenez une minute pour vous familiariser avec le réseau. Identifiez les adresses et les interfaces utilisées par toutes les machines (`ifconfig`). Vérifiez le contenu des routes (`route`). Les tables de routage sont probablement incomplètes. Complétez-les de manières à ce que `gw1` et `gw2` puissent se joindre (`ping`), ainsi que `pc1` et `pc2`, et finalement que chacun puisse joindre tous les autres.

2 Un VPN SSL : OpenVPN

2.1 Création d'un VPN réseau à réseau

Il s'agit de créer un réseau entre les machines passerelle `gw1` et `gw2`. Ce réseau est un tunnel sur lequel la machine 1 aura par exemple l'adresse `192.168.0.1` et la machine 2 l'adresse `192.168.0.2`.

— Sur chacune des machines gateway (identifiée par X l'autre étant la machine Y), lancer :
`openvpn --remote ipY --port 8000 --dev tun1 \`
`--ifconfig 192.168.0.X 192.168.0.Y --verb 5`
Réflechissez bien à la valeur d'*ipY* et n'oubliez pas de substituer X et Y. Note : le caractère *backslash* (\) ne fait pas partit de la commande, c'est juste pour protéger le retour à la ligne...
— Quel est le rôle de `tun1` ?
— Il faut maintenant redéfinir les routes. Passez vos processus `openvpn` en tâche de fond (sans les arrêter). [1] Puis à l'aide de la commande `route` arrêter et remplacer la route vers le réseau contrôlé par Y. Expliquez le sens de l'opération (à la fois interface et adresses). Faites `man route` si vous ignorez la syntaxe.
— Échangez du trafic entre les deux machines : vous pouvez commencer par `ping`
— Capturez le trafic avec `tcpdump` (par exemple sur `pc1` et `pc3`). Utilisez par exemple :
`tcpdump -U -w /hostlab/capture[13].pcap -i eth0`
(c.à.d. la capture sur le `pc1` s'appelle `capture1.pcap`, et `capture3.pcap` sur le `pc3`.)
— Le fichier est écrit dans le répertoire de base du TP. Vous pouvez le regarder avec Wireshark :
  — Quel type de trafic est échangé ?
  — Il faut décoder la payload en choisissant le bon filtre.
— Pouvez vous en tirer des conclusions ? Regardez les adresses et expliquez l'encapsulation.

L'option compression
Vous pouvez rajouter de la compression avec l'option `--comp-lzo`. Recommencez l'expérience (utilisez `killall` pour arrêter OpenVPN et n'oubliez pas de remettre les routes en place). Refaîtes une capture sur `pc3`. Qu'avez vous perdu ?

Utilisez un fichier de configuration
À partir de cette étape il est conseillé d'utiliser des fichiers de configuration. Créez deux fichiers `vpn-gw1.conf` et `vpn-gw2.conf` en reprenant chacune des options passées à OpenVPN à raison d'une option par ligne et en supprimant les deux tirets (`--`) en tête d'option. Vérifiez que

1. Faites <Ctrl-Z>, puis `bg` ; ou relancez `openvpn` avec le caractère `&` en fin de ligne de commande.

votre fichier est correct en relançant OpenVPN cette fois ci avec pour seule option `--config <fichier>`.

2.2 Sécurisation du VPN

2.2.1 Chiffrement

Vous allez créer une clé pour le chiffrement des données que vous mettrez dans `/hostlab/key`. Elle sera ainsi accessible aux deux machines passerelles. Dans la réalité, il faudrait probablement un coursier pour les échanger.

La génération est très simple et se fait directement avec OpenVPN :
`openvpn --genkey --secret /hostlab/key`

Ajoutez l'option `--secret /hostlab/key` à la commande de lancement d'OpenVPN et relancez le RPV. Regardez de nouveau le trafic avec *tcpdump/wireshark*. Que pouvez vous en conclure ?

2.2.2 Génération de certificats

Nous allons créer des clés RSA et les faire signer par une autorité de certification (CA) qui sera installée sur `pc1` (la seule machine pour laquelle le fichier de configuration ssl a été correctement défini [2]).

Rappel : l'autorité de certification signe les certificats qui contiennent l'identité et la clé publique d'une machine. Ce faisant, il garantit que le possesseur de la clé privé est bien celui qui est décrit. L'autorité de certification émet un certificat signé par elle même. C'est par lui que nous allons commencer :

1. Génération de la clé privé et du certificat (sur `pc1`) :
   ```
   openssl req -subj '/C=fr/O=maboite/CN=ca.maboite.com/' -new -x509 \
           -keyout priv-ca.pem -out crt-ca.pem
   ```
2. Génération de la clé privé et de la demande de certificat pour `gw1` (sur `gw1`) :
   ```
   openssl req -subj '/C=fr/O=maboite/CN=gw1.maboite.com/' -new -nodes \
           -keyout priv-gw1.pem -out csr-gw1.pem
   ```
3. Génération du certificat (sur `pc1` après transfert de la demande) :
   ```
   openssl ca -cert crt-ca.pem -keyfile priv-ca.pem \
           -out crt-gw1.pem -in csr-gw1.pem
   ```
4. Rapatriez sur `gw1` : le certificat de `gw1` généré sur `pc1`, et une copie du certificat de `CA`.

Expliquez en regardant le manuel chaque option, en particulier :
— Que génère-t-on à chaque étape ?
— Quel est le rôle de `-x509` dans la première opération ?
— Quel est le rôle de `-nodes` (à lire *"no DES"*). Pourquoi ne l'utilise-t-on pas avec la clé du CA ? Pourquoi est-elle utile pour les clés des machines ? (`man req`)
— Que décrit l'argument de `-subj`. Vous pouvez aussi entrer interactivement le contenu en supprimant cette option. Que signifient les différents champs déclarés ?

Répétez l'opération pour `gw2`, normalement vous avez maintenant 3 fichiers sur chaque passerelle correspondant à une clé privée et deux certificats.

2. Si les commandes échouent en réclamant des fichiers dans `demoCA`, la cause probable est que vous êtes sur la mauvaise machine.

2.2.3 Génération du fichier de paramètres pour l'échange de Diffie-Hellman

Ce fichier n'est calculé que d'un coté. Dans notre exemple sur `gw1` :

```
openssl dhparam -out dh1024.pem 1024
```

Que venez vous de générer ?

2.2.4 Démarrer openvpn

Vous devez écrire les deux fichiers de configuration dans lesquels vous préciserez :

- l'adresse de l'autre extrémité, le port, l'interface réseau virtuelle utilisée, la configuration des interfaces du tunnel (comme pour la version précédente)
- le fait que l'un est serveur et l'autre client pour TLS (`tls-server` et `tls-client`)
- du bon côté l'emplacement du fichier de paramètre Diffie-Helmann (`dh`)
- l'emplacement du certificat de l'autorité de certification (`ca`)
- l'emplacement du certificat de la passerelle (`cert`)
- l'emplacement de la clé privé de la passerelle (`key`)

Lancez OpenVPN et remettez les routes en place.

3 VPN IPSEC - Openswan

3.1 Création de la configuration

Le fichier de configuration doit être créé dans `/etc/ipsec.conf`.

```
version 2

config setup
  interfaces="ipsec0=eth1"
  uniqueids=yes

conn netkit
  keyingtries=0
  authby=secret
  auto=start
  # Definition of left side
  # Definition of right side
```

Regardez dans le manuel (`ipsec.conf`) le sens de `authby`.

Pour chaque coté que nous appellerons dorénavant *`dir`* (i.e. `left` ou `right`), vous devez préciser :

- *`dir`*`=<ip>` l'adresse externe de la passerelle ;
- *`dir`*`subnet=<ip>/<size>` la spécification du sous réseau ;
- *`dir`*`nexthop=<ip>` vers quel noeud externe sont relayés les messages de la passerelle pour atteindre l'autre coté.

Enfin vous devez préciser la clé utilisée dans `/etc/ipsec.secrets` :

```
ipleft ipright: PSK "chaine de caractères partégée"
```

Notez qu'en pratique vous pouvez omettre les adresses IP, tant que vous n'avez qu'une connexion.

Vous pouvez utiliser le même fichier de configuration sur les deux machines, pourquoi ?

Vous devez copier le fichier décrivant la clé sur les deux machines.

Vous pouvez maintenant lancer le VPN en tapant :
`ipsec setup start`
(Syntaxe générale : `ipsec setup {start|stop|restart|reload|version}`)

3.2 Observation du résultat

Vérifiez que vous pouvez bien échanger des données entre `pc1` et `pc2` et entre `pc1` et `pc3`.

Faîtes une capture et regardez la avec Wireshark. Comment est encapsulé le trafic ? Comparez avec la solution OpenVPN. Vous pouvez regarder les négociations entre passerelles dans `/var/log/auth.log`. C'est un bon endroit pour comprendre ce qui se passe si votre configuration ne marche pas. Faites également `ipsec auto --status`

3.3 Utilisation des clés RSA et certificats X509

Dans `/etc/ipsec.d` vous trouverez un certain nombre de répertoires. En particulier :
— `private` pour les clés privées de la passerelle ;
— `cacerts` pour les certificats de l'autorité de confiance qui garantit l'autre extrémité ;
— `certs` pour les certificats assurant l'identité de la passerelle.

Copiez sur chacune des passerelles, les bons fichiers (déjà générés pour OpenVPN) dans ces trois répertoires.

Modifiez `ipsec.secrets` pour utiliser la clé privé RSA (regardez le manuel pour trouver quelle option choisir et comment spécifier le fichier de clé privé.[3] Attention, la clé n'est plus symétrique !

Modifiez `ipsec.conf` :
— Modifiez la méthode d'authentification en `rsasig`.
— Ajoutez pour chaque coté :
  — l'identification du certificat avec *dir*`cert=fichier.pem`
  — l'identifiant attendu avec *dir*`id="identifiant"`
    Par exemple pour `gw1`, c'est `/C=fr/O=maboite/CN=gw1.maboite`
    D'où vient cet identifiant ?

Partagez votre fichier de configuration.
Vous pouvez maintenant arrêter OpenSwan (`ipsec setup stop`) et le redémarrer.

4 Configuration Road warrior

Si vous êtes arrivé ici avant la fin du TP, félicitations !

Pour terminer, voici un sujet plus ouvert. Commencez par arrêter les VPN et installer une route statique entre `gw1` et `gw2` comme en tout début de TP. Désormais il s'agit d'installer un VPN à la demande de `pc2` ou `pc3` leur permettant d'accéder à `pc1` à travers la passerelle `gw1`. C'est une configuration *«road-warrior»*.

Commencez par OpenVPN. Vous aurez besoin de nouveaux certificats, mais l'infrastructure de base est en place. Le problème principal est l'allocation des adresses pour `pc2` et `pc3`.

3. Il faut donner le nom complet du fichier mais pas son répertoire. OpenSwan (en fait le composant *pluto* assurant l'échange de clé) la trouvera si elle est à sa place.