Université de Bordeaux — année 2019-2020

Master CSI/Master Info. — Sécurité des réseaux

TD - OpenVPN

Le but de ce TP est de mettre en place un tunnel OpenVPN entre deux machines distantes afin de créer un VPN (réseau privé virtuel entre ces deux machines). Ce mécanisme sera mis en place entre les machines `immortal`, `nile`, et `dt` de la plate-forme décrite ci-dessous :

La topologie réseau correspondante peut être obtenue en lançant le script de démarrage `/net/stockage/aguermou/SR/TP/11/qemunet.sh` en lui fournissant la description de la topologie réseau à l'aide de l'option `-t` ainsi que l'archive contenant la configuration initialte des machines à l'aide de l'option `-a`. Ceci revient à lancer les commandes suivantes :

```
cd /net/stockage/aguermou/SR/TP/11/; ./qemunet.sh -x -t topology -a archive_tp11.tgz
```

1. Nous allons tout d'abord générer trois certificats X509 pour `dt`, `immortal` et `nile`. Attention, il est important de donner un numéro de série différent pour chacun des certificats.

   (a) Nous allons commencer par créer la clé privée de la CA :

   ```
   certtool --generate-privkey --outfile ca-key.pem
   ```

   (b) Puis nous allons générer le certificat de cette dernière (Attention le champ Common Name doit absolument être saisi).

   ```
   certtool --generate-self-signed --load-privkey ca-key.pem --outfile \
      ca-cert.pem
   ```

   (c) Ensuite, nous allons générer une clé privée pour l'une de nos machines.

   ```
   certtool --generate-privkey --outfile key.pem
   ```

   (d) Enfin, nous allons générer et signer le certificat :

   ```
   certtool --generate-certificate --load-privkey key.pem --outfile \
      cert.pem --load-ca-certificate ca-cert.pem  --load-ca-privkey \
      ca-key.pem
   ```

   Le contenu du certificat peut être consulté soit en éditant le fichier, ou en utilisant la commande :

   ```
   certtool --certificate-info --infile cert.pem
   ```

   Enfin pour générer le fichier contenant les paramètre pour le protocole de Diffie-Hellman il faut exécuter la commande suivante :

   ```
   certtool --get-dh-params > dh.pem
   ```

   Attention : Pour que l'étape suivante fonctionne correctement, vous devez lors de la génération des certificats choisir la machine qui doit jouer le rôle de serveur (son `CN` sera alors `server`) et celles qui vont jouer le rôle de clients auront un `CN` égal à `client1`, `client2`, etc.

2. Nous allons commencer par utiliser `openvpn` dans le contexte d'un tunnel IP (couche 3 de la pile TCP/IP) à l'aide d'une interface de type `tun`. Cette action se fera entre `nile` et `immortal`.

   — Sur `immortal`, le serveur, il suffit de lancer la commande suivante :

   ```
   openvpn --dev tun1 --ifconfig 10.0.1.1 10.0.1.2 --tls-server \
     --dh <fichier_dh> --ca <fichier_ca> --cert <fichier_certificat_s> \
     --key <fichier_cle_s> --reneg-sec 60 --verb 5
   ```

— Sur `nile`, le client, il faudra lancer la commande suivante :
```
openvpn --remote <@IP immortal> --dev tun1 --ifconfig 10.0.1.2 10.0.1.1
   --tls-client --ca <fichier_ca> --cert <fichier_certificat_c>
   --key <fichier_cle_c> --reneg-sec 60 --verb 5
```
Tester la configuration et analyser la traffic entre `immortal` et `nile` lorsque ces dernières communiquent entre elles.

3. Nous allons passer maintenant à la configuration d'un tunnel de niveau 2 (`tap`) entre `immortal`, `nile` et `dt`. `immortal` va jouer le rôle de serveur. Nous allons donc commencer par configurer cette dernière pour qu'elle joue ce rôle. Il faut tout d'abord créer un fichier de configuration pour OpenVPN. Le fichier doit avoir le contenu suivant :

```
port 1194
proto udp
dev tap0


script-security 3 #system

up /root/up.sh
down /root/down.sh

ca <fichier_ca_s>
cert <fichier_certificat_s>
key <fichier_cle_s>
dh <fichier_dh>

ifconfig-pool-persist ipp.txt

#verify-x509-name "CN=client"
verify-x509-name "client" name-prefix

client-to-client

# Les adresses allant de  .100 to .200 sont réservées aux clients VPN.
server-bridge 192.168.0.1 255.255.255.0 192.168.0.100 192.168.0.200

# Ajout d'une route spécifique vers le réseau 140.77.13.0/24
push "route 140.77.13.0 255.255.255.0 192.168.0.1"

keepalive 10 120
comp-lzo
persist-key
persist-tun
status openvpn-status.log
verb 3
```

Comme vous pouvez le remarquer, le fichier de configuration ci-dessus fait référence à différents fichiers contenant soit les objets cryptographiques à manipuler ou des scripts à lancer lors du démarrage ou de l'arrèt du tunnel. Le contenu du fichier `up.sh` est fourni ci-dessous :

```
#!/bin/sh
PATH=/sbin:/usr/sbin:/bin:/usr/bin
DEV=$1
```

```
brctl addbr br0
brctl addif br0 eth0

echo brctl addif br0 eth0

ifconfig eth0 0.0.0.0 promisc up
ifconfig $DEV 0.0.0.0 promisc up

brctl addif br0 $DEV
echo brctl addif br0 $DEV

ifconfig br0 192.168.0.1 netmask 255.255.255.0 broadcast 192.168.0.255
```

Le fichier `down.sh` quant à lui doit contenir :

```
#!/bin/sh
PATH=/sbin:/usr/sbin:/bin:/usr/bin

ifconfig br0 down
brctl delbr br0

ip link set eth0 promisc off
ip addr add 192.168.0.1 dev eth0
```

Une fois la configuration effectuée, il sera possible de lancer `openvpn` avec la commande suivante :

`openvpn --config <chemin_vers_fichier_conf>`

4. Concernant les machines clientes, la configuration peut se faire en copiant les bons objets cryptographiques sur chacune des machines puis dans une deuxième phase de configurer le client à l'aide du fichier de configuration suivant :

```
client
dev tap
proto udp

remote 172.16.0.2 1194
nobind
persist-key
persist-tun

ca <fichier_ca>
cert <fichier_certificat_c>
key <fichier_cle_c>

verify-x509-name "CN=server"
comp-lzo
verb 3
```

De même que pour le serveur, le lancement d'`openvpn` pour le client se fait avec la commande suivante :

`openvpn --config <chemin_vers_fichier_conf>`

Remarque : Vérifier que les adresses MAC des l'interfaces tap0 des machines sont différentes. Si ce n'est pas le cas ajouter une directive `lladdr @MAC` dans le fichier `openvpn.conf`.

5. Essaye de faire communiquer les différents clients entre eux et avec le serveur en vous mettant en écoute sur `grave` pour analyser le traffic généré.