

OpenVPN

OpenVPN est un utilitaire libre qui permet de mettre en place un serveur VPN (*Virtual Private Network*) qui permet à des pairs de s'authentifier entre eux à l'aide d'une clé privée partagée à l'avance, de certificats ou de couples login/passwords. Il utilise de manière intensive la bibliothèque d'authentification OpenSSL ainsi que le protocole SSLv3/TLSv1. Disponible sous Solaris, Linux, *BSD, Mac OS X, Windows*, il offre aussi de nombreuses fonctions de sécurité et de contrôle.

1 Installation d'OpenVPN et premiers tests

OpenVPN peut être installé au moyen de yum dans la version `openvpn-2.1-0.17` avec le paquet `lzo.i386 0 :2.02-2.fc6`. Dans un premier temps, nous allons petit à petit mettre en place la configuration proposée dans la figure 1.

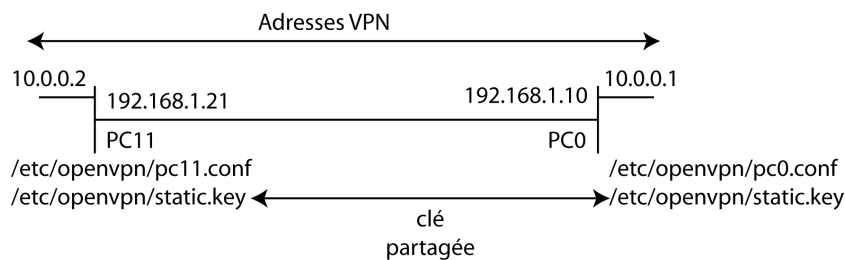


FIG. 1 – Première configuration

1.1 Premiers tests

Afin de vérifier l'installation, effectuons un test (non chiffré) où PC0 est serveur VPN et PC11 client :

```
PC0$ openvpn --dev tun0 --ifconfig 10.0.0.1 10.0.0.2
PC11$ openvpn --remote pc0.tp.sys --dev tun0 --ifconfig 10.0.0.2 10.0.0.1
```

Ces commandes démarrent le réseau virtuel (d'adresse 10.0.0.0) et d'attribuent des IP aux machines.

- (1) Notez la création d'une nouvelle interface réseau (`ifconfig`).
- (2) Vérifiez par des `ping` que le réseau fonctionne convenablement.
- (3) Installez un serveur `telnet` coté serveur pour avoir accès à un service accessible depuis le client.
- (4) Le cas échéant, vérifiez que la connexion n'est pas chiffrée.

1.2 Ajout d'une clé partagée

Il est indispensable de chiffrer les communications avec un VPN. On commence par le réaliser au moyen d'une clé partagée qu'il faut engendrer au préalable sur le serveur et distribuer au client.

- (1) engendrez une clé partagée sur le serveur `openvpn -genkey -secret static.key`, recopiez-la sur le client (de manière sécurisée).
- (2) Relancez le VPN entre les deux machines en concaténant aux commandes précédentes la directive `-secret /path/to/key` et vérifiez le bon fonctionnement du réseau.
- (3) Le cas échéant, vérifiez que la connexion est maintenant chiffrée.
- (4) Pour utiliser de la compression (et gagner ainsi de la bande passante, vous pouvez également concaténer la directive `-comp-lzo -keepalive 10 60 -float` aux commandes précédentes (et en vérifier le bon fonctionnement).

1.3 Fichiers de configuration

Pour simplifier le démarrage, il est préférable de créer un fichier de configuration. Le démarrage en devient grandement simplifié : `openvpn /path/to/pc.conf`. Fichier de configuration client :

```
dev tun
remote 192.168.39.129
ifconfig 10.0.0.2 10.0.0.1
secret /sw/etc/openvpn/static.key
comp-lzo
keepalive 10 60
float
```

- (1) Créez le fichier de configuration serveur et testez-en le bon fonctionnement.

2 Utilisation de clés SSL/TLS

La première étape pour obtenir une vraie configuration est de construire une PKI qui consiste en :

- (1) un certificat séparé (une clé publique) et une clé privée pour le serveur et pour chaque client ;
- (2) une autorité de certification qui signe les certificats précédents.

2.1 Création de l'AC et du PKI

On construit d'abord l'AC au moyen des scripts fournis dans `easy-rsa/2.0` distribué avec le paquet.

- (1) Editez et mettez à jour le fichier `vars` qui va contenir les informations X509 de certification.
- (2) Faites l'initialisation des PKI par

```
. vars
./clean-all
./build-ca
```

- (3) Construisez les paramètres de l'échange de clés par Diffie-Hellman par `./build-dh` ;
- (4) Construisez le certificat et la clé privée du serveur par `./build-key-server pc0` en répondant affirmativement aux deux questions posées.
- (5) Construisez enfin la clé du client (ou autant de clés que de clients) par `./build-key pc11`.
- (6) Copiez les clés dans les répertoires concernés pour le serveur et les clients.
- (7) Copiez depuis le répertoire `sample-config-files` les fichiers `server.conf` et `client.conf` et adaptez-les à votre configuration.
- (8) Adaptez et utilisez le script de démarrage `/etc/init.d/openvpn` pour démarrer automatiquement le serveur.
- (9) Testez votre configuration.

A la fin de l'étape (5), le répertoire `/easy-rsa/2.0/keys/` doit contenir :

fichier	pour	rôle	secret ?
<code>ca.crt</code>	serveur+clients	cert. racine AC	non
<code>ca.key</code>	AC=serveur	clé racine AC	oui
<code>dh1024.pem</code>	serveur	paramètres DH	non
<code>pc0.crt</code>	serveur	cert serveur	non
<code>pc0.key</code>	serveur	clé serveur	oui
<code>pc11.crt</code>	client	cert client	non
<code>pc11.key</code>	client	clé client	oui

3 Pontage

L'intérêt d'un VPN est surtout de permettre d'accéder à un réseau privé physique par l'intermédiaire du VPN hébergé sur le serveur VPN. On se place dans la configuration réseau du TP1, i.e. avec une passerelle avec deux interfaces réseaux en faisant de la traduction d'adresses réseaux. Le nouveau schéma du réseau devient celui de la figure 2.

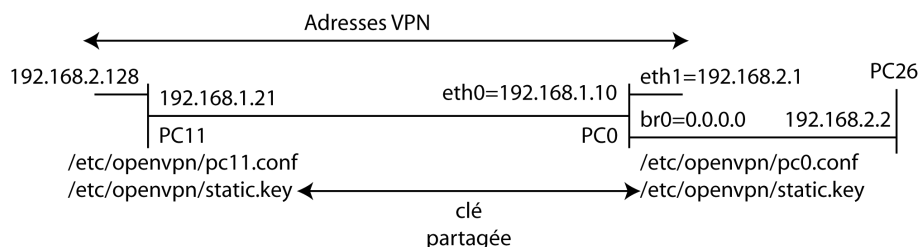


FIG. 2 – Seconde configuration.

Pour réaliser le pontage (*bridge*), il faut préalablement installer `bridge-utils.i386` par yum et recopier les scripts `bridge-start` et `bridge-stop` dans `/etc/openvpn`. Pour le TP, nous utilisons la configuration suivante :

config.	paramètre <code>bridge-start</code>	valeur
interface eth.	<code>eth</code>	<code>eth1</code>
IP locale	<code>ip</code>	<code>192.168.2.1</code>
masque local	<code>eth_netmask</code>	<code>255.255.255.0</code>
broadcast	<code>eth_broadcast</code>	<code>192.168.2.255</code>
IP clients VPN		<code>192.168.2.128 à 192.168.2.254</code>
iface virtuelle bridge	<code>br</code>	<code>br0</code>
iface virtuelle TAP	<code>tap</code>	<code>tap0</code>

- Définissez le réseau comme décrit dans la figure 2 sur `eth1`.
- Modifiez le script `bridge-start` en lui donnant les valeurs convenables pour `br`, `eth`, `eth_ip`, `eth_netmask` et `eth_broadcast`.
- Vérifiez la création des deux interfaces réseau `br0` et `tap0` par `ifconfig`.

Il faut ensuite modifier le fichier de configuration d'`openvpn` en retirant `dev tun` et en la remplaçant par `dev tap0`. Il faut également commenter la ligne `server` et la remplacer par `server-bridge 192.168.2.1 255.255.255.0 192.168.2.128 192.168.2.254`. Notons que les adresses seront attribuée dynamiquement, comme s'il y avait un serveur `dhcp`.

Il faut ensuite activer l'`ip_forwarding` et mettre à jour les règles de firewall de linux :

```
iptables -A INPUT -i tap0 -j ACCEPT
iptables -A INPUT -i br0 -j ACCEPT
iptables -A FORWARD -i br0 -j ACCEPT
```

On peut alors initialiser le VPN en démarrant successivement : `bridge-start` puis `openvpn` et le stopper en arrêtant successivement `openvpn` puis `bridge-stop`. Coté client, il faut commenter la ligne qui débute par `dev tun` et la remplacer par `dev tap`.

4 Firewalling

Avec un VPN, il faut également configurer le firewall pour permettre le bon fonctionnement du VPN.