OpenVPN : Installation

Aujourd'hui, les derniers exemples de nos dirigeants nous donnent encore une très bonne raison d'utiliser OpenVPN, mais principalement il va permettre d'anonymiser les connexions d'un utilisateur (chambre d'hôtel, télé-travail, de passer un serveur proxy un peu chiant 😉, de faire une LAN party, ...), de faire son cloud privé sécurisé avec comme accès le réseau VPN, de voir un site distant, ...

Comme vous pouvez le voir dans mes exemples, il y a pleins de raison d'utiliser ce genre d'outils,

mais pour moi la vraie raison sa licence GNU GPL ce qui est à mon sens la liberté 😬

J'ai réalisé cette article avec une distribution Linux Debian 8, mais c'est faisable sur d'autres en adaptant 🤣

Prérequis :

Avant de commencer, l'installation d'OpenVPN nécessite la résolution des dépendances de celui-ci, mais pas que, nous allons aussi préparer la création de la PKI et de la compilation d'OpenVPN

```
aptitude install liblzo2-2 liblzo2-dev libssl-dev openssl libpam0g-dev dh-autoreconf git gcc g++
```

compilation

Télécharger les sources openvpn source

```
tar -xvzf openvpn-2.3.6.tar.gz
cd openvpn-2.3.6
```

La compilation n'a rien de bien compliqué

```
autoreconf -i -v -f
./configure  --enable-strict --prefix=/usr/local
make
```

autoreconf : la commande permet de mettre à jour les fichiers de configuration générés
./configure : On regroupe dans cette catégorie les macros liées à la configuration préalable et à la compilation d'une application.
Les petites options :

- **–enable-strict** : active le mode débogage “warnings”

make : Le but de l'utilitaire *make* est de déterminer automatiquement quelles sont les parties d'un gros programme qu'il faut recompiler, et d'exécuter les commandes appropriées

Une fois la compilation terminer nous allons effectuer un test et après on passe à l'installation

```
make check
make install
```

L'infrastructure à clés publiques "PKI" (Easy-rsa version 3)

Désormais, Easy-RSA n'est plus fourni en bundle avec OpenVPN, il doit être récupéré depuis un dépôt Git.

```
git clone https://github.com/OpenVPN/easy-rsa
```

Alors, libre à vous de faire cela sur la même machine, pour des raisons de facilité, j'ai volontairement fait sur la même machine et même mis à coter du fichier de configuration pour un but pratique (on est d'accord tout ce qui est pratique n'est pas forcément sécurisé ), **mais dans un environnement de production on veillera à ne pas faire cela**.

```
cd easy-rsa
cp vars.example vars
```

Éditer le fichier *vars* copié, décommenter et modifier les paramètres suivants:

```
set_var EASYRSA_REQ_COUNTRY    "US"
set_var EASYRSA_REQ_PROVINCE   "California"
set_var EASYRSA_REQ_CITY       "San Francisco"
set_var EASYRSA_REQ_ORG        "Copyleft Certificate Co"
set_var EASYRSA_REQ_EMAIL      "me@example.net"
set_var EASYRSA_REQ_OU         "My Organizational Unit"
```

Initialisez la *PKI* et construisez le *CA*.

```
cd /etc/openvpn/easy-rsa
./easyrsa init-pki
./easyrsa build-ca
```

Ensuite il y a 2 méthode :

méthode long, mais on maitrise LOL

Générer une requête de certificat et une paire de clés.

```
./easyrsa gen-req <nom-du-serveur> [nopass]
```

Utilisez l'option [nopass] pour ne pas être embêté pour une passphrase **(option recommandée pour le serveur, mais déconseillée pour les clients)**.

Faite de même sur chacun des clients :

```
./easyrsa gen-req <nom-du-client>
```

souvenez vous de la passphrase, car elle vous sera demandée à la connexion

Transférez ensuite chacun des fichiers .req (générés dans pki/reqs/) sur le serveur PKI.

```
cp pki
cp reqs/* ../
```

Sur la PKI, importez ces demandes.

```
./easyrsa import-req /chemin/vers/<nom-du-client>.req <nom-court-unique>
```

dans mon exemple :

```
./easyrsa import-req /etc/openvpn/easy-rsa/toto.req toto
```

Puis, s'il s'agit d'un .req serveur, signez la demande avec :

```
./easyrsa sign server <nom-court-unique>
```

pour un client :

```
./easyrsa sign client <nom-court-unique>
```

Méthode rapide

La commande suivante va générer et signer automatiquement les certificats clients et serveur

Création du certificat serveur :

```
./easyrsa build-server-full <nom-du-serveur> [nopass]
```

Création d'un certificat client :

```
./easyrsa build-client-full <nom-du-client> [nopass]
```

Une fois terminé, retournez dans le répertoire d'easy-rsa et générez les paramètres Diffie-Hellman :

```
./easyrsa gen-dh
```

Sur le serveur OpenVPN

les fichiers suivant vous serons nécessaire :

nom du fichier	**details**
ca.crt	certificat de l'autorité de certification
serveur.key	fichier contenant la clé RSA privé du serveur
serveur.crt	certificat x509 pour une durée de validité de 10 ans et auto-signé
dh.pem	contient les paramètres Diffie-Hellman, en gros c'est une clé

Placer ces fichiers à la racine de votre serveur OpenVPN (`/etc/openvpn`)

Voilà l'installation est finie, j'ai segmenté l'article pour que vous puissiez le digéré et me poser des questions, la configuration est très simple et surtout il existe plusieurs configurations possibles donc cela me permettra d'essayer de traités cela dans d'autres articles.

Articles en lien : OpenVPN : configuration tun avec authentification PAM
sources: OpenVPN easy-rsa 3 man OpenVPN wiki OpenVPN