

OpenVPN : client et serveur VPN

- Objet : Installer et configurer intégralement OpenVPN
- Niveau requis : [débutant, avisé](#)
- Commentaires : *Mise en place d'un serveur VPN et de son client.*
- Débutant, à savoir (pour le moins...) : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
- Suivi : [obsolète](#)
 - Création par  [TechDesk](#) le 30/12/2012
 - Testé par ... le ...
- Commentaires sur le forum : [ici](#)¹⁾

Un guide plus récent basé sur celui-ci est à privilégier : [OPENVPN Serveur et Client](#)

Présentation

Cette technique permet la création d'une liaison chiffrée entre votre machine et un serveur hébergé sur Internet (par exemple chez un fournisseur d'accès se trouvant en France ou à l'étranger). Tous vos accès à Internet seront alors vus à partir de l'adresse IP de ce serveur VPN et non plus par celle de votre machine.

OpenVPN n'est pas un VPN IPsec. C'est un VPN SSL se basant sur la création d'un tunnel IP (UDP ou TCP au choix) authentifié et chiffré avec la bibliothèque OpenSSL.

Quelques avantages des tunnels VPN SSL :

- Facilité pour passer les réseaux **NATés** (pas de configuration à faire)
- Logiciel clients disponibles sur **GNU/Linux, BSD, Windows et Mac OS X**

Installation Serveur

On commence par installer OpenVPN à partir des dépôts officiels :

```
apt-get update && apt-get install openvpn
```

On copie ensuite les fichiers de configurations :

```
mkdir /etc/openvpn/keys
```

```
cp -r /usr/share/easy-rsa/ /etc/openvpn
```

```
chown -R $USER /etc/openvpn/easy-rsa/
```

Configuration

À l'aide des scripts installés dans le répertoire **/etc/openvpn/easy-rsa/** nous allons configurer OpenVPN pour utiliser une authentification par clés et certificats.

On commence par éditer²⁾ le fichier **/etc/openvpn/easy-rsa/vars** :

```
export KEY_COUNTRY="FR"
export KEY_PROVINCE="FR"
export KEY_CITY="FR"
export KEY_ORG="exemple.com"
export KEY_EMAIL="exemple@exemple.com"
```

Ensuite on lance la séquence suivante qui va générer les **clés (.key)** et les **certificats (.crt)** :

```
cd /etc/openvpn/easy-rsa/
source vars
./clean-all
./build-dh
./pkitool --initca
./pkitool --server server
openvpn --genkey --secret keys/ta.key
```

On copie ensuite les clés et les certificats utiles pour le serveur dans le répertoire **/etc/openvpn/** :

```
cp keys/ca.crt keys/ta.key keys/server.crt keys/server.key keys/dh2048.pem
/etc/openvpn/
```

Puis on génère un répertoire **/etc/openvpn/jail** dans lequel le processus OpenVPN sera chrooté (afin de limiter les dégâts en cas de faille dans OpenVPN) :

```
mkdir /etc/openvpn/jail
```

Puis un autre répertoire (**/etc/openvpn/clientconf**) qui contiendra la configuration des clients :

```
mkdir /etc/openvpn/clientconf
```

Enfin on crée le fichier de configuration **/etc/openvpn/server.conf** :

[fichier /etc/openvpn/server.conf](#)

```
# Serveur TCP/443
mode server
proto tcp
port 443
dev tun

# Clés et certificats
ca ca.crt
cert server.crt
```

```
key server.key
dh dh2048.pem
tls-auth ta.key 0
cipher AES-256-CBC

# Réseau
server 10.8.0.0 255.255.255.0
push "redirect-gateway def1 bypass-dhcp"
push "dhcp-option DNS 8.8.8.8"
push "dhcp-option DNS 8.8.4.4"
keepalive 10 120

# Sécurité
user nobody
group nogroup
chroot /etc/openvpn/jail
persist-key
persist-tun
comp-lzo

# Log
verb 3
mute 20
status openvpn-status.log
log-append /var/log/openvpn.log
```

Ce fichier permet de créer un serveur VPN SSL routé basé sur le protocole TCP et utilisant le port HTTPS (443) afin de maximiser son accessibilité depuis des réseaux sécurisés par des Firewalls. Les clients obtiendront une nouvelle adresse IP dans le range 10.8.0.0/24.

Démarrage du serveur

On lance le serveur avec la commande :

```
service openvpn start
```

À ce stade les machines clientes vont pouvoir se connecter au serveur VPN. Par contre impossible d'aller plus loin que ce dernier car l'adresse 10.8.0.x ne sera pas routée en dehors de votre serveur. Il faut donc configurer le serveur pour qu'il joue le rôle de routeur entre l'interface VPN (tun0) et l'interface physique (eth0) et de NATeur entre les adresses en 10.8.0.x et son adresse IP réelle.

Configuration du routage:

```
sh -c 'echo 1 > /proc/sys/net/ipv4/ip_forward'
```

Pour rendre ce paramétrage de routage permanent (même après un reboot), il faut ajouter la ligne suivante au fichier **/etc/sysctl.conf** :

```
net.ipv4.ip_forward = 1
```

Puis configurer la translation d'adresse (NAT).

Dans la commande qui suit, remplacer `voire_interface` par l'interface réelle de votre carte réseau.

Avant stretch c'était genre `eth0`... :

```
iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o voire_interface -j MASQUERADE
```

Pour rendre cette règle de NAT persistante après un reboot de votre serveur, il faut commencer par créer un script de chargement de règles de Firewall (ou utiliser un script existant) :

```
iptables-save > /etc/iptables.rules
```

On peut aussi simplement ajouter la ligne suivante dans la configuration de son interface réseau (généralement la section `eth0` dans **`/etc/network/interface`**)

```
post-up iptables-restore < /etc/iptables.rules
```

Le serveur est maintenant prêt à accueillir les clients.

Installation Client

```
apt-get install openvpn
```

Configuration

Nous allons créer les clés pour le client "**client1**", pour cela il faut saisir **sur le serveur** les commandes suivantes :

```
cd /etc/openvpn/easy-rsa
source vars
./build-key client1
```

Si vous souhaitez protéger l'accès à vos clés par un mot de passe (c'est à dire qu'un mot de passe sera demandé à la montée du tunnel VPN), il faut utiliser la commande **`./build-key-pass`** en lieu et place de **`./buil-key`**.

Le script **`./build-key`** va générer 3 fichiers dans le répertoire **`/etc/openvpn/easy-rsa/keys`** :

- `client1.crt`: Certificat pour le client
- `client1.csr`: Certificat à garder sur le serveur
- `client1.key`: Clés pour le client

On copie les fichiers nécessaires dans un sous répertoire du répertoire **`/etc/openvpn/clientconf/`** préalablement créé :

```
mkdir /etc/openvpn/clientconf/client1/
```

```
cp /etc/openvpn/ca.crt /etc/openvpn/ta.key keys/client1.crt keys/client1.key  
/etc/openvpn/clientconf/client1/
```

On va ensuite dans le répertoire **/etc/openvpn/clientconf/client1/** :

```
cd /etc/openvpn/clientconf/client1/
```

Puis on crée le fichier `client.conf` (il faut remplacer **A.B.C.D** par l'adresse publique de votre serveur VPN que vous pouvez obtenir avec la commande "**wget -qO- whatismyip.org**") :

[client.conf](#)

```
# Client  
client  
dev tun  
proto tcp-client  
remote A.B.C.D 443  
resolv-retry infinite  
cipher AES-256-CBC  
  
# Clés  
ca ca.crt  
cert client1.crt  
key client1.key  
tls-auth ta.key 1  
  
# Sécurité  
nobind  
persist-key  
persist-tun  
comp-lzo  
verb 3
```

Pour assurer la compatibilité avec le **client Windows OpenVPN**, on fait une copie du fichier **client.conf** vers **client.ovpn** :

```
cp client.conf client.ovpn
```

On devrait ainsi avoir les fichiers suivants dans le répertoire **/etc/openvpn/clientconf/client1/**:

- `ca.crt`: Certificat du serveur
- `client.conf`: Fichier de configuration du client OpenVPN (Linux, BSD, MacOS X)
- `client.ovpn`: Fichier de configuration du client OpenVPN (Windows)
- `client1.crt`: Certificat du client
- `client1.key`: Clés du client
- `ta.key`: Clés pour l'authentification

Il ne reste plus qu'à mettre ces fichiers dans une archive **zip**³⁾ et de la transmettre sur le **PC client**:

```
zip client1.zip *.*
```

Merci à **TechDesk** l'auteur avisé et attentif de l'ensemble de ce tuto ! 😊

Lien

- [OpenVPN Client-gui](#) Cryptage entre PC Client et Serveur en graphique.

1)

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

2)

[nano](#)

3)

ou [tar](#)

From:
<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:
<http://debian-facile.org/doc:reseau:openvpn>

Last update: **13/03/2021 05:40**

