OpenVPN

By COCHET Anthony
www.anthony-cochet.fr

Introduction :

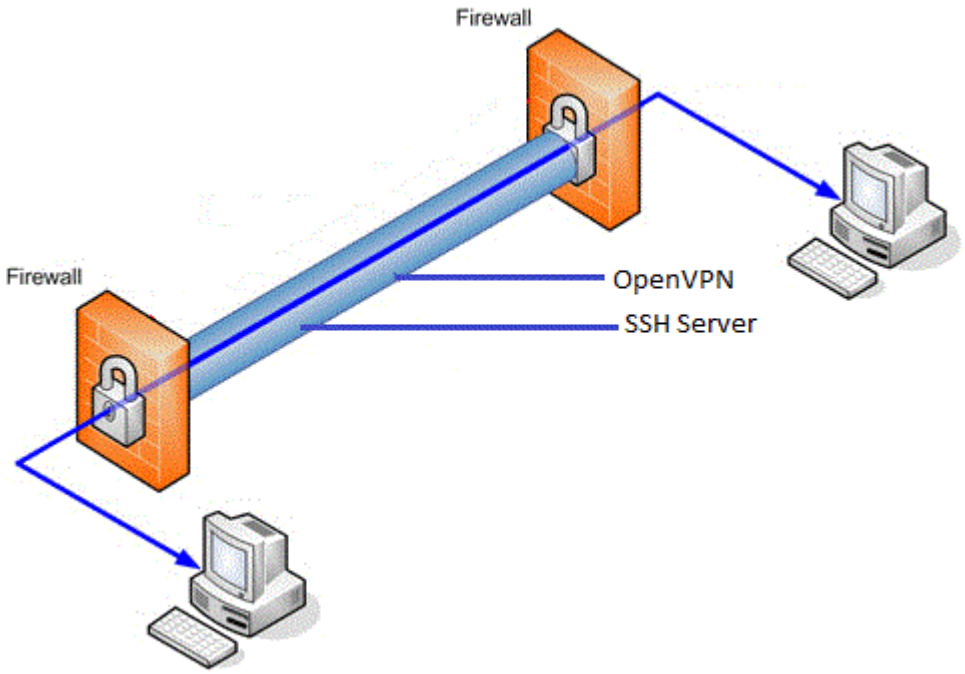
On entend de plus en plus parler des réseaux privés virtuels. Cette technique permet la création d'une liaison chiffrée entre votre machine et un serveur hébergé sur Internet (par exemple chez un fournisseur d'accès se trouvant en France ou à l'étranger). Tous vos accès à Internet seront alors vus à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

Nous allons voir comment installer et configurer son propre serveur VPN sous Debian basée sur OpenVPN, une solution libre et compatible avec des clients multi-OS (Windows/Linux).

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 NAT.
- Logiciel clients disponibles sur GNU/Linux, BSD, Windows et Mac OS X.



Installation du serveur OpenVPN :

On commence par installer OpenVPN et OpenSSL à partir des dépôts officiels:

```
sudo aptitude install openvpn openssl
```

On copie ensuite les fichiers de configurations:

```
sudo mkdir /etc/openvpn/easy-rsa/

sudo cp -r /usr/share/doc/openvpn/examples/easy-rsa/2.0/* /etc/openvpn/easy-rsa/

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

Configuration du serveur OpenVPN :

A 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 éditer le fichier **/etc/openvpn/easy-rsa/vars**:

```
export KEY_COUNTRY="FR"

export KEY_PROVINCE="60"

export KEY_CITY="CHANTILLY"

export KEY_ORG="PRODCOCHET.LAN"

export KEY_EMAIL="postmaster@prodcochet.lan"
```

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

sudo openvpn --genkey --secret keys/ta.key
```

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

```
sudo cp keys/ca.crt keys/ta.key keys/server.crt keys/server.key keys/dh1024.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) puis un autre répertoire **(/etc/openvpn/clientconf**) qui contiendra la configuration des clients:

```
sudo mkdir /etc/openvpn/jail

sudo mkdir /etc/openvpn/clientconf
```

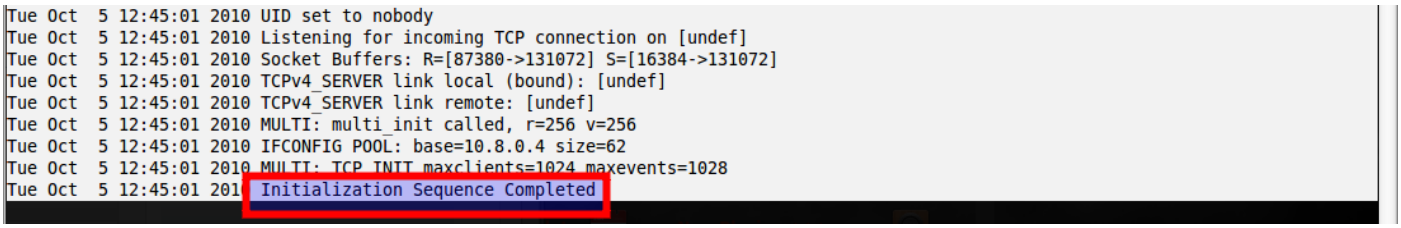
Enfin ont créé le fichier de configuration **/etc/openvpn/server.conf**:

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

# Cles et certificats
ca ca.crt
cert server.crt
key server.key
dh dh1024.pem

tls-auth ta.key 1
```

```
key-direction 0

cipher AES-256-CBC

# Reseau

server 10.8.0.0 255.255.255.0

push "redirect-gateway def1 bypass-dhcp"

push "dhcp-option DNS 208.67.222.222"

push "dhcp-option DNS 208.67.220.220"

keepalive 10 120

# Securite

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 basée sur le protocole TCP et utilisant le port HTTPS (443) enfin de maximiser son accessibilité depuis des réseaux sécurisés par des Firewalls. Les clients obtiendrons une nouvelle adresse IP dans le range 10.8.0.0/24.

On teste la configuration en saisissant la commande suivante:



Si le serveur démarre correctement, on peut terminer la configuration sur serveur OpenVPN en décommentant la dernière ligne du fichier **/etc/openvpn/server.conf** :

```
log-append /var/log/openvpn.log
```

On lance le serveur avec la commande:

```
sudo /etc/init.d/openvpn start
```

A 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 par 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 NAT entre les adresses en 10.8.0.x et son adresse IP réelle.

Configuration du routage :

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

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

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

Puis configuration la translation d'adresse (NAT):

```
sudo iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o eth0 -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):

```
sudo sh -c "iptables-save > /etc/iptables.rules"
```

Puis éditer votre fichier /etc/network/interfaces pour y ajouter la ligne suivante après la définition de votre interface réseau principale ("iface eth0 inet..." par exemple):

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

Le serveur est maintenant prêt à accueillir les clients. Nous allons donc voir dans l'activité suivante comment déclarer un client sur le serveur OpenVPN.