OpenVPN avec client Debian

>>> En mode routé

Description :

OpenVPN est un service VPN très sûr, opensource et simple à mettre en place. Cet article explique comment configurer un serveur OpenVPN ainsi qu'un client OpenVPN installé sur une Debian. Nous utiliserons le mode tun qui correspond au mode routé qui est plus facile à mettre en place.

OpenVPN avec client Debian

>>> En mode routé

Sommaire :

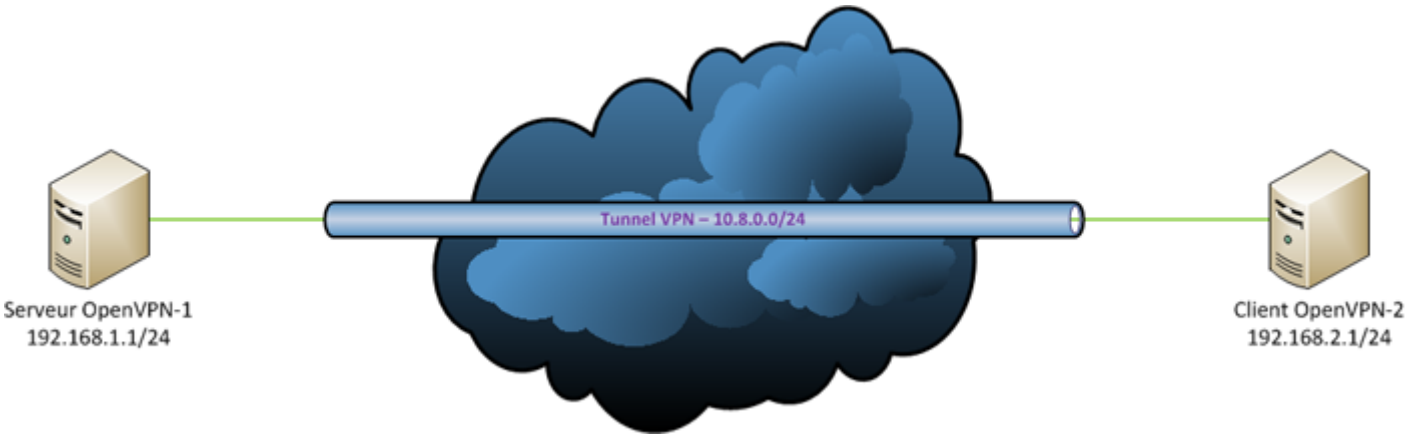
I) Explications
II) Schéma réseau
III) Installation et configuration du serveur
1) Installations
2) Création des clés et certificats
3) Création d'un utilisateur
4) Configuration du fichier server.conf
5) Activation du module tun
6) Redémarrage du service et vérifications
IV) Installation et configuration du client
1) Installations
2) Copie des clefs et certificats
3) Création d'un utilisateur
4) Configuration du fichier client.conf
5) Activation du module tun
6) Redémarrage du service et vérifications
V) Révocation d'un certificat client

I) Explications

OpenVPN permet de créer un Tunnel VPN entre un serveur et un client. Nous allons commencer par générer les clés et les certificats permettant de chiffrer les données qui vont transiter par ce tunnel. Puis nous installerons et configurerons le serveur et le client.

II) Schéma réseau

Voici un schéma de notre réseau :



III) Installation et configuration du serveur

1) Installations

L'installation est simple, taper la commande suivante :

```
aptitude install openvpn openssl liblzo2-2
```

Voilà la première étape est terminée !

2) Création des clés et certificats

Cette étape est l'une des plus complexes dans la configuration d'un serveur OpenVPN. Mais heureusement OpenVPN a tout prévu. Celui-ci possède plusieurs scripts permettant de générer plus facilement les clés et les certificats avec OpenSSL. Ces scripts sont présents dans le dossier : **/usr/share/doc/openvpn/examples/easy-rsa/2.0/**.

On commence par accéder au dossier suivant :

```
cd /usr/share/doc/openvpn/examples/easy-rsa/2.0/
```

Puis on édite le fichier **vars** :

```
vim vars
```

On se déplace à la fin du fichier et on modifie les lignes suivantes :

```
export KEY_COUNTRY=FR
export KEY_PROVINCE=France
export KEY_CITY=Cherbourg
export KEY_ORG=idum
export KEY_EMAIL=contact@idum.fr
```

Une fois le fichier modifié, la ligne suivante permet d'initialiser les variables pour les scripts :

```
bash vars
```

ou

```
source ./vars
```

Le script suivant, permet de créer ou de réinitialiser le sous-dossier **keys** :

```
bash clean-all
```

Le script suivant permet de créer dans **keys**, le certificat principal du serveur **ca.crt** et la clé correspondante **ca.key** :

```
bash build-ca
```

Voici ce qui s'affiche sur l'écran (taper sur Entrée pour sélectionner les valeurs par défaut qui sont entre crochées) :

```
Generating a 1024 bit RSA private key
.........++++++
.....................................................................++++++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [FR]:
State or Province Name (full name) [France]:
Locality Name (eg, city) [Cherbourg]:
Organization Name (eg, company) [Idum]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) [Idum CA]:
Name []:
Email Address [contact@idum.fr]:
```

Quelques champs ne sont pas renseignés par défaut, vous devrez les compléter manuellement.

Le script suivant permet de créer dans **keys** le certificat **OpenVPN-1-server.crt** et la clé **OpenVPN-1-server.key** pour le serveur VPN. Dans mon cas j'ai nommé le serveur **OpenVPN-1-server** :

```
bash build-key-server OpenVPN-1-server
```

Voici ce que l'on obtient à l'écran :

```
Generating a 1024 bit RSA private key
.........++++++
.........................++++++
writing new private key to 'OpenVPN-1-server.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [FR]:
State or Province Name (full name) [France]:
Locality Name (eg, city) [Cherbourg]:
Organization Name (eg, company) [Idum]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) [OpenVPN-1-server]:
Name []:
Email Address [n.salmon@idum.fr]:

Please enter the following 'extra' attributes
```

```
to be sent with your certificate request
A challenge password []:aqw
string is too short, it needs to be at least 4 bytes long
A challenge password []:azerty
An optional company name []:
Using configuration from /usr/share/doc/openvpn/examples/easy-rsa/2.0/openssl.cn
f
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'FR'
stateOrProvinceName   :PRINTABLE:'France'
localityName          :PRINTABLE:'Cherbourg'
organizationName      :PRINTABLE:'Idum'
commonName            :PRINTABLE:'OpenVPN-1-server'
emailAddress          :IA5STRING:'n.salmon@idum.fr'
Certificate is to be certified until Aug 19 21:17:45 2021 GMT (3650 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Nous allons maintenant créer la clé et le certificat pour le client OpenVPN :

```
bash build-key OpenVPN-2-client
```

On obtient ceci à l'écran :

```
Generating a 1024 bit RSA private key
.++++++
.....................++++++
writing new private key to 'OpenVPN-2-client.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [FR]:
State or Province Name (full name) [France]:
Locality Name (eg, city) [Cherbourg]:
Organization Name (eg, company) [Idum]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) [OpenVPN-2-client]:
Name []:
Email Address [n.salmon@idum.fr]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:azerty
An optional company name []:
Using configuration from /usr/share/doc/openvpn/examples/easy-rsa/2.0/openssl.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'FR'
stateOrProvinceName   :PRINTABLE:'France'
localityName          :PRINTABLE:'Cherbourg'
organizationName      :PRINTABLE:'Idum'
commonName            :PRINTABLE:'OpenVPN-2-client'
emailAddress          :IA5STRING:'n.salmon@idum.fr'
Certificate is to be certified until Aug 19 21:21:26 2021 GMT (3650 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

On génère les paramètres de Diffie Hellman :

```
bash build-dh
```

Vous devriez avoir ceci à l'écran :

```
Generating DH parameters, 1024 bit long safe prime, generator 2
This is going to take a long time
.....................................................................................+
.....................................................................+.....................
..........+.........................................................................
................+.....................................................................
+.......................+..............................+..............................
......+...................+.....................................................+..
++*++*++*
```

Pour terminer cette étape nous allons déplacer les clefs générées, taper les commandes suivantes :

```
cp ./keys/ca.crt /etc/openvpn/
cp ./keys/ca.key /etc/openvpn/
cp ./keys/OpenVPN-1-server.crt /etc/openvpn/
cp ./keys/OpenVPN-1-server.key /etc/openvpn/
cp ./keys/dh1024.pem /etc/openvpn
```

3) Création d'un utilisateur

Pour limiter les risques d'attaques sur OpenVPN, il est important que le processus d'OpenVPN fonctionne sur un utilisateur n'ayant aucun droit sur le système. Souvent, l'utilisateur **nobody** est utilisé par défaut, mais il est encore plus sécurisant de faire tourner chaque processus avec un utilisateur différent. Donc, pour le processus OpenVPN, nous allons créer l'utilisateur **openvpn** :

```
groupadd openvpn
useradd -d /dev/null -g openvpn -s /bin/false openvpn
```

4) Configuration du fichier server.conf

Encore une autre étape importante, la configuration du fichier server.conf peut être compliqués. Mais encore une fois OpenVPN a tout prévu, un exemple du fichier de configuration se trouve dans le dossier : **/usr/share/doc/openvpn/examples/sample-config-files** :

```
cd /usr/share/doc/openvpn/examples/sample-config-files
```

On extrait l'archive :

```
gunzip server.conf.gz
```

On copie ensuite le fichier server.conf dans le dossier d'OpenVPN, et on ouvre ce dossier :

```
cp server.conf /etc/openvpn/

cd /etc/openvpn/
```

Il suffit d'adapter ce fichier en fonction des besoins. Voici par exemple le fichier de configuration que j'utilise :

```
vim /etc/openvpn/server.conf
```

Ce qui nous donne :

```
;Adresse IP de l'interface local
local 192.168.1.1

;Port en écoute utilisé pour la connexion VPN
port 1194

;Protocole utilisé
proto tcp

;Type d'interface réseau virtuelle créée
dev tun

;Nom des fichiers servant à l'authentification des clients via OpenSSL
ca ca.crt
cert OpenVPN-1-server.crt
key OpenVPN-1-server.key
dh dh1024.pem

;Adresse du réseau virtuel (Le serveur aura l'adresse 10.8.0.1) adresse dans le VPN
server 10.8.0.0 255.255.255.0

;Cette ligne ajoute sur le client la route du réseau du serveur
;push "route 192.168.0.0 255.255.255.0"

;Ces lignes indiquent aux clients l'adresse des serveur DNS et WINS
;push "dhcp-option DNS 192.168.0.2"
;push "dhcp-option DOMAIN MonDomaine.com"
;push "dhcp-option WINS 192.168.0.3"

# Cette ligne permet aux clients de voire les autres clients
;client-to-client

keepalive 10 120

;Cette ligne active la compression
comp-lzo

;Ces lignes indiquent un user et un group particulier pour le processus
user openvpn
group openvpn

;Ces lignes permettent de rendre persistante la connexion
persist-key
persist-tun

status openvpn-status.log

;Cette ligne permet d'indiquer le niveau de log souhaité (de 1 à 9)
verb 3
;mute5
```

---

5) Activation du module tun

L'activation du module Tun est importante pour le fonctionnement du tunnel et de ses interfaces virtuels.

Pour activer le module taper :

```
modprobe tun
```

Pour que le module se charge au démarrage éditer le fichier **/etc/modules** :

```
vim /etc/modules
```

Et ajouter :

```
tun
```

6) Redémarrage du service et vérifications

On redémarre le service **openvpn** pour que les paramètres soient pris en compte :

```
/etc/init.d/openvpn restart
```

Au moindre problème on regarde le fichier de log :

```
tail -25 /var/log/syslog
```

Et on vérifie la configuration des interfaces réseaux :

```
ifconfig
```

Vous devriez avoir cette interface en plus :

```
tap0      Link encap:Ethernet  HWaddr 86:c7:dd:3f:ed:9a
          inet adr:10.8.0.1  Bcast:10.8.0.255  Masque:255.255.255.0
          adr inet6: fe80::84c7:ddff:fe3f:ed9a/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:100
          RX bytes:0 (0.0 B)  TX bytes:468 (468.0 B)
```

Le serveur OpenVPN ayant toujours l'adresse **10.8.0.1**

IV) Installation et configuration du client

1) Installations

L'installation du client est identique à celle du serveur, car c'est le même logiciel qui fait office de serveur ou de client en fonction de sa configuration :

```
aptitude install openvpn liblzo2-2
```

2) Copie des clefs et certificats

On copie les clés créées précédemment dans le dossier **/etc/openvpn** :

- ca.crt
- OpenVPN-2-client.crt
- OpenVPN-2-client.key

Le plus simple pour copier les clés d'un serveur à un autre est d'utiliser un FTP.

3) Création d'un utilisateur

Pour limiter les risques d'attaques sur OpenVPN, il est important que le processus d'OpenVPN fonctionne sur un utilisateur n'ayant aucun droit sur le système. Souvent, l'utilisateur **nobody** est utilisé par défaut, mais il est encore plus sécurisant de faire tourner chaque processus avec un utilisateur différent. Donc pour le processus OpenVPN, nous allons créer l'utilisateur **openvpn** :

```
groupadd openvpn
useradd -d /dev/null -g openvpn -s /bin/false openvpn
```

4) Configuration du fichier client.conf

Il suffit d'adapter ce fichier en fonction des besoins. Voici par exemple le fichier de configuration que j'utilise :

```
vim /etc/openvpn/client.conf
```

Vous devez obtenir ceci :

```
# Specify that we are a client and that we
# will be pulling certain config file directives
# from the server.
client


# Use the same setting as you are using on
# the server.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
;dev tap
dev tun


# Are we connecting to a TCP or
# UDP server?  Use the same setting as
```

```
# on the server.
proto tcp
;proto udp


# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
remote 192.168.1.1 1194


# Keep trying indefinitely to resolve the
# host name of the OpenVPN server.  Very useful
# on machines which are not permanently connected
# to the internet such as laptops.
resolv-retry infinite


# Most clients don't need to bind to
# a specific local port number.
nobind

# Downgrade privileges after initialization (non-Windows only)
user openvpn
group openvpn

# Try to preserve some state across restarts.
persist-key
persist-tun

# SSL/TLS parms.
# See the server config file for more
# description.  It's best to use
# a separate .crt/.key file pair
# for each client.  A single ca
# file can be used for all clients.
ca ca.crt
cert OpenVPN-2-client.crt
key OpenVPN-2-client.key


# Verify server certificate by checking
# that the certicate has the nsCertType
# field set to "server".  This is an
# important precaution to protect against
# a potential attack discussed here:
#  http://openvpn.net/howto.html#mitm
#
# To use this feature, you will need to generate
# your server certificates with the nsCertType
# field set to "server".  The build-key-server
# script in the easy-rsa folder will do this.
ns-cert-type server


# Enable compression on the VPN link.
# Don't enable this unless it is also
# enabled in the server config file.
comp-lzo


# Set log file verbosity.
verb 3
```

---

5) Activation du module tun

L'activation du module Tun est importante pour le fonctionnement du tunnel et de ses interfaces virtuels. Pour activer le module taper :

```
modprobe tun
```

Pour que le module se charge au démarrage éditer le fichier **/etc/modules** :

```
vim /etc/modules
```

Et ajouter :

```
tun
```

6) Redémarrage du service et vérifications

On redémarre le service **openvpn** pour que les paramètres soient pris en compte :

```
/etc/init.d/openvpn restart
```

Au moindre problème on regarde le fichier de log :

```
tail -25 /var/log/syslog
```

Et on vérifie la configuration des interfaces réseaux :

```
ifconfig
```

Vous devriez avoir cette interface en plus :

```
tap0      Link encap:Ethernet  HWaddr 86:c7:dd:3f:ed:9a
          inet adr:10.8.0.6  Bcast:10.8.0.255  Masque:255.255.255.0
          adr inet6: fe80::84c7:ddff:fe3f:ed9a/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:100
          RX bytes:0 (0.0 B)  TX bytes:468 (468.0 B)
```

V) Révocation d’un certificat client

Imaginer qu’un client se fasse voler son certificat ou que le client ne doit plus avoir accès au VPN, il est nécessaire de pouvoir révoquer son certificat pour qu’il ne puisse plus être utilisé.

Commencer par vous assurer que le certificat du client en question est bien présent dans le dossier **/usr/share/doc/openvpn/examples/easy-rsa/2.0/keys/**

Dans mon cas le certificat client se nomme : OpenVPN-2-client

On commence par se rendre dans le dossier **/usr/share/doc/openvpn/examples/easy-rsa/2.0/** :

```
cd /usr/share/doc/openvpn/examples/easy-rsa/2.0
```

On réinitialise les variables :

```
source ./vars
</vars>

Et on tape la commande suivante :

<code>
./revoke-full OpenVPN-2-client
```

Vous devriez voir apparaitre ceci à l'écran :

```
 ./revoke-full OpenVPN-2-client
Using configuration from /usr/share/doc/openvpn/examples/easy-rsa/2.0/openssl.cnf
Adding Entry with serial number 02 to DB for /C=FR/ST=France/L=Cherbourg/O=Idum/CN=OpenVPN-2-
client/emailAddress=contact@idum.fr
Revoking Certificate 02.
Data Base Updated
Using configuration from /usr/share/doc/openvpn/examples/easy-rsa/2.0/openssl.cnf
OpenVPN-2-client.crt: /C=FR/ST=France/L=Cherbourg/O=Idum/CN=OpenVPN-2-client/emailAddress=contact@idum.fr
error 23 at 0 depth lookup:certificate revoked
```

On ajoute la ligne ci-dessous dans le fichier **server.conf**, qui obligera le serveur à vérifier le fichier crl.pem pour connaitre les certificats révoké avant d'accepter la communication.

```
crl-verify crl.pem
```

On redémarre le service **openvpn** sur le serveur et sur le client :

```
/etc/init.d/openvpn restart
```

Pour finir on vérifie que nous avons toujours les interfaces tun0 :

```
ifconfig
```

et que la communication fonctionne :

```
ping 10.8.0.6

PING 10.8.0.6 (10.8.0.6) 56(84) bytes of data.
From 192.168.2.254 icmp_seq=1 Destination Host Unreachable
From 192.168.2.254 icmp_seq=2 Destination Host Unreachable
From 192.168.2.254 icmp_seq=3 Destination Host Unreachable
From 192.168.2.254 icmp_seq=4 Destination Host Unreachable
--- 10.8.0.6 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3003ms
```

Idum