

Passerelle VPN : OpenVPN

Debian GNU/Linux



Matthieu Vogelweith
13 août 2009

Résumé

OpenVPN [1] sous Debian Lenny [2]

...

Ce document a été rédigé en LaTeX en utilisant l'excellent Vim sous Debian GNU/Linux. Il est disponible aux formats XHTML et PDF. Les sources LaTeX sont disponibles ici : [L^AT_EX](#)

Licence

Copyright ©2009 Matthieu VOGELWEITH <matthieu@vogelweith.com>.

Vous avez le droit de copier, distribuer et/ou modifier ce document selon les termes de la GNU Free Documentation License, Version 1.3 ou ultérieure publiée par la Free Software Foundation ; avec aucune section inaltérable, aucun texte de première page de couverture, et aucun texte de dernière page de couverture. Une copie de la licence est disponible dans la page [GNU Free Documentation License](#).

Historique

- **13-08-2009** : Mise à jour pour Debian Lenny

Table des matières

Table des matières	4
1 Introduction	5
2 Passerelle OpenVPN	6
2.1 Installation des paquets	6
2.2 Configuration de base	6
2.3 Paramètres réseaux	7
2.4 Utilisateur système dédié	7
2.5 Gestion des clients	8
2.6 Gestion des journaux	8
2.7 Sécurité	9
2.8 Résumé	9
3 Client OpenVPN Linux	10
3.1 Installation des paquets	10
3.2 Configuration de base	10
3.3 Configuration des DNS	11
3.4 Démarrage du service	11
4 Client OpenVPN Windows	12
4.1 Installation	12
4.2 Configuration de base	12
4.3 Configuration des DNS	12
4.4 Démarrage du service	12
5 Client OpenVPN Mac OS X	13
5.1 Installation	13
5.2 Configuration de base	13
5.3 Configuration des DNS	14
5.4 Démarrage du service	14
6 Management du serveur	15
6.1 Activation	15
6.2 Premiers tests	15
7 Références	17

Chapitre 1

Introduction

- Simple
- Flexible
- Robuste
- Multi-plateforme

Chapitre 2

Passerelle OpenVPN

2.1 Installation des paquets

L'installation du paquet se fait, comme toujours, avec la commande suivante :

```
# aptitude install openvpn
```

Notons que le paquet installé précédemment permet aussi bien d'utiliser la machine comme passerelle (serveur) OpenVPN que comme client, voir les deux en même temps. Notons également qu'après l'installation du paquet le répertoire est totalement vide, il n'y a aucune configuration par défaut.

2.2 Configuration de base

Comme indiqué ci-dessus, aucune configuration n'est prédéfinie après l'installation du paquet. Cependant, des fichiers d'exemples très bien commentés sont disponibles dans `/usr/share/doc/openvpn/examples/`. Cette documentation propose cependant de partir d'une configuration vierge pour bien comprendre l'intérêt de chaque option.

`/etc/openvpn/server.conf` :

```
# Server configuration
dev tun
comp-lzo
persist-key
persist-tun
keepalive 10 60
server 10.10.0.0 255.255.0.0
```

L'adressage et les options de base du serveur sont définis, il faut maintenant configurer les options SSL. Dans un premier temps, comme pour tous les services qui utilisent SSL pour chiffrer les communications, il faut indiquer un certificat et sa clé privée ainsi que le certificat de l'autorité de certification qui a été utilisé pour signer le certificat du serveur. Notons que la génération de ces certificats sera détaillée par la suite dans un chapitre dédié.

Ces options SSL sont donc déclarées de la manière suivante dans `/etc/openvpn/server.conf` :

```
# SSL parameters
ca keys/ca.crt
```

```
cert keys/gateway.crt
key keys/gateway.key
dh keys/dh1024.pem
crl-verify keys/crl.pem
```

Notons que dans la configuration ci-dessus, deux paramètres indispensables ont été ajoutés :

- une clé de "Diffie Hellman" [3] qui sera utilisée pour sécuriser l'échange de clés SSL avant que la communication ne soit chiffrée. (L'option est nommée "dh").
- un fichier ou seront enregistrés les certificats clients révoqués. (L'option est nommée "crl-verify")

2.3 Paramètres réseaux

Options DHCP :

```
# VPN Gateway offer DNS parameters
push "dhcp-option DOMAIN example"
push "dhcp-option DNS 10.1.0.1"
push "dhcp-option DNS 10.1.0.2"
push "dhcp-option WINS 10.1.0.3"
push "dhcp-option NTP 10.1.0.4"
```

Routes :

```
# VPN Gateway offer this routes to clients
push "route 10.2.0.0 255.255.0.0"
```

Gestion du MTU :

```
# Fix MTU problems
mssfix 1300
```

2.4 Utilisateur système dédié

Afin d'améliorer sensiblement la sécurité du système, il est possible de faire tourner le serveur OpenVPN avec un utilisateur dédié ayant des droits restreints sur le système de fichiers de la machine. Notons que le démarrage d'OpenVPN sera toujours effectué en root mais que les droits seront modifiés immédiatement après le démarrage du service. Ce mode de fonctionnement permet notamment d'avoir les droits nécessaires pour créer les périphériques logiques (tun*) indispensables au fonctionnement d'OpenVPN.

Dans un premier temps il est donc nécessaire de créer un utilisateur système **openvpn** et de donner les droits à cet utilisateur sur le répertoire `/etc/openvpn` :

```
# adduser --system --group --home /etc/openvpn openvpn
# chown -R openvpn:openvpn /etc/openvpn/
```

Maintenant que l'utilisateur est créé, il reste à modifier la configuration d'OpenVPN dans `/etc/openvpn/server.conf` pour qu'il utilise cet utilisateur et que le service soit cantonné au répertoire `/etc/openvpn` :

```
# Drop root privileges
chroot /etc/openvpn
user openvpn
group openvpn
```

Après un redémarrage du service, il est possible de vérifier dans les journaux que ces commandes ont bien été appliquées et que le service tourne bien avec l'utilisateur **openvpn** :

```
# /etc/init.d/openvpn restart
# tail -n 100 /var/log/openvpn.log | grep openvpn
xxx xxx x xx:xx:xx xxxx us=680077 chroot to '/etc/openvpn' and cd to '/' succeeded
xxx xxx x xx:xx:xx xxxx us=680123 GID set to openvpn
xxx xxx x xx:xx:xx xxxx us=680189 UID set to openvpn
# ps -ef | grep openvpn
openvpn 4665 1 0 Jan17 ? 00:03:19 /usr/sbin/openvpn --writepid /var/run/
openvpn.server.pid --daemon ovpn-server --status /var/run/openvpn.server.status 10
--cd /etc/openvpn --config /etc/openvpn/server.conf
```

2.5 Gestion des clients

Par défaut, les clients connectés au VPN peuvent accéder à la passerelle elle-même et aux réseaux qui ont été "offert" par la passerelle. Si les clients doivent pouvoir se joindre entre eux, il faut ajouter l'option suivante dans le fichier de configuration de la passerelle :

```
client-to-client
```

- Utilisation des CCD.

```
client-config-dir ccd
```

- Ajout des routes dispo derrière les clients via `iroute`.

2.6 Gestion des journaux

Il est important d'enregistrer les journaux du serveur VPN pour déceler d'éventuels problèmes ou avoir une bonne vision de l'état du système. Afin de ne pas rendre le syslog illisible, il peut-être intéressant de stocker les journaux dans un fichier dédié à cet effet. Pour cela, ajouter les options suivantes dans le fichier de configuration du serveur, soit `/etc/openvpn/server.conf` :

```
# Logging configuration
status openvpn-status.log
log-append /var/log/openvpn.log
verb 5
```

Notons que l'option "status" permet d'enregistrer l'état courant du serveur dans un fichier. Il sera alors possible de voir à chaque instant quels sont les clients connectés et leur adresse dans le VPN.

ATTENTION, si les journaux sont écrits dans un fichier dédié comme c'est le cas dans la configuration ci-dessus, il faut absolument mettre en place une rotation des ces journaux. Dans le cas contraire, OpenVPN tombe lorsque le fichier atteint 2Go.

Pour cela, il suffit de créer le fichier `/etc/logrotate.d/openvpn` et d'y ajouter les directives suivantes :

```
/var/log/openvpn.log {
    rotate 4
    daily
    copytruncate
    compress
    missingok
    notifempty
}
```

Ainsi, une rotation des journaux d'OpenVPN sera effectuée chaque jour et les 4 derniers fichiers seront conservés. Bien sur ces valeurs doivent être adaptées en fonction de l'utilisation du serveur : si les journaux sont trop importants il est possible de diminuer le niveau de verbosité d'OpenVPN ou de modifier les paramètres de la rotation.

ATTENTION : L'option "copytruncate" est très importante. Lorsque cette option est activée, logrotate ne va pas déplacer le fichier de log mais va simplement le "tronquer" après l'avoir copié. De cette façon, le démon OpenVPN n'est pas obligé de fermer ce fichier pour pouvoir écrire dans le nouveau.

2.7 Sécurité

- tls-auth

2.8 Résumé

- Fonctions de la passerelle VPN mise en place.

Chapitre 3

Client OpenVPN Linux

3.1 Installation des paquets

Comme indiqué dans le chapitre précédent, le paquet à installer sous Linux est le même coté client et coté serveur :

```
# aptitude install openvpn
```

Comme sur le serveur la configuration par défaut est vide. La mise en place d'une configuration de type "client" est l'objet du paragraphe suivant.

3.2 Configuration de base

La configuration des clients est beaucoup plus simple que celle du serveur : il suffit de définir les éléments de base et les certificats SSL à utiliser pour établir la connexion. Une configuration type permettant d'établir une connexion sur le serveur présenté dans le chapitre précédent peut être obtenue en enregistrant les informations ci-dessous dans le fichier **/etc/openvpn/vpn-example.conf** sur le client :

```
# Client configuration
client
remote vpn.example.org
nobind
dev tun
comp-lzo

# SSL Configuration
ca ca.crt
cert client.crt
key client.key

# Logging configuration
log /var/log/openvpn.log
verb 3
```

Remarquons simplement l'option **nobind** qui permet à OpenVPN d'écouter sur un port aléatoire plutôt que d'utiliser le port 1194. Avec cette option il est donc impossible de connaître à l'avance le port d'écoute d'OpenVPN.

Bien entendu, il faut copier les certificats à l'endroit indiqué dans le fichier de configuration ci-dessus :

```
# cp ca.crt client.crt client.key /etc/openvpn/  
# chmod 600 /etc/openvpn/client.key
```

3.3 Configuration des DNS

- mise à jour de resolvconf.

3.4 Démarrage du service

Si plusieurs connexions VPN sont configurées sur le même client, il peut être utile de ne pas toutes les démarrer en même temps lors du lancement d'OpenVPN. Pour cela, il suffit par exemple de désactiver toutes les connexions au lancement du démon puis de démarrer chaque tunnel à la demande. Pour désactiver l'établissement automatique des tunnels, il suffit de modifier l'option suivante dans /etc/default/openvpn :

```
AUTOSTART="none "
```

Ensuite, pour démarrer le VPN "vpn-example", il suffit d'exécuter la commande suivante :

```
# /etc/init.d/openvpn start vpn-example
```

Notons que c'est exactement le même principe pour l'arrêt des VPNs : il suffit de passer le nom du VPN en argument au script d'init.

Chapitre 4

Client OpenVPN Windows

Le client Officiel OpenVPN est disponible sous Windows, accompagné d'un interface graphique permettant de lancer ou couper le VPN à partir de la barre des tâches.

4.1 Installation

L'installation du client est relativement simple, il faut cependant utiliser l'installateur en version 2.1. Cette version est encore étiquetée en "bêta" mais reste tout a fait utilisable en production tout en apportant de nombreuses améliorations comme :

- Elle fonctionne sous Vista 32 et 64 bits
- L'interface graphique fait maintenant partie de l'installateur officiel.
- Les privilèges administrateurs ne sont plus requis pour faire fonctionner le VPN.

L'installateur est disponible dans la page téléchargement du site officiel ou directement à l'adresse suivante :

```
http://openvpn.net/release/openvpn-2.1\_rc19-install.exe
```

4.2 Configuration de base

4.3 Configuration des DNS

4.4 Démarrage du service

Chapitre 5

Client OpenVPN Mac OS X

Il existe plusieurs interfaces OpenVPN sous Mac OS X, notamment Tunnelblick [4] et Viscosity [5]. Viscosity est un produit assez jeune mais l'interface est très soignée et il fonctionne très bien. Malheureusement il ne fonctionne que sous MacOS 10.5 et ne sera plus gratuit à partir du 31/10/2008. Ce paragraphe détaillera donc la mise en place de TunnelBlick qui est disponible sous licence GPLv2.

5.1 Installation

L'installation du client Mac OS X est très simple, il suffit de :

- Télécharger le fichier .dmg sur le site de Tunnelblick [4] ;
- Monter l'image .dmg et déplacer le répertoire Tunnelblick dans le dossier des applications.

5.2 Configuration de base

Avant de lancer Tunnelblick pour la première fois, on peut déjà créer la configuration du client. Tunnelblick utilisant un OpenVPN standard, la configuration est exactement la même que pour le client Linux :

```
# Client configuration
client
remote vpn.example.org
nobind
dev tun
comp-lzo

# SSL Configuration
ca ca.crt
cert client.crt
key client.key

# Logging configuration
log /var/log/openvpn.log
verb 3
```

Enfin, créer simplement un répertoire OpenVPN dans le dossier "Bibliothèque" de l'utilisateur et ajouter les fichiers suivants dans ce répertoire :

- Le fichier de configuration OpenVPN client.conf
- Les 3 certificats SSL.

5.3 Configuration des DNS

Comme sous les autres plates-formes, la configuration des DNS fournie par le serveur OpenVPN demande un peu plus de travail ...

5.4 Démarrage du service

Lors du lancement de Tunnelblick, une icône de "tunnel" apparaît prêt de l'horloge en haut à droite. Pour établir le tunnel OpenVPN, il suffit maintenant de cliquer sur cette icône pour sur **connect vpn-example**. Pour vérifier que tout c'est déroulé correctement, il est également possible de visualiser les logs à l'aide du menu **details**.

Chapitre 6

Management du serveur

OpenVPN dispose d'un module d'administration accessible via une socket TCP. Ce module permet d'administrer le serveur OpenVPN à distance via un programme externe.

6.1 Activation

Pour activer le module de management, il suffit de définir l'adresse et le port d'écoute dans le fichier de configuration du serveur. Par exemple, pour que le serveur écoute sur toutes les interfaces sur le port 11940, ajouter la ligne suivante :

```
management 127.0.0.1 11940 management.secret
```

Notons que l'option ci-dessus utilise également un fichier **management.secret** qui sera utilisé pour authentifier l'utilisateur lors de la connexion à l'interface de management. Il faut donc enregistrer un mot de passe dans ce fichier puis redémarrer le service pour que les modifications soient prises en compte :

```
# echo "mypassword" > /etc/openvpn/management.secret
# chown openvpn:openvpn /etc/openvpn/management.secret
# chmod 600 /etc/openvpn/management.secret
# /etc/init.d/openvpn restart
```

Il est possible de vérifier que le service écoute bien sur le réseau avec la commande suivante :

```
# netstat -naptu | grep 11940
```

6.2 Premiers tests

Dans un premier temps, il est possible de tester le service de management avec un simple telnet :

```
# telnet 127.0.0.1 11940
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^['.
```

```

ENTER PASSWORD:mypassword
SUCCESS: password is correct
>INFO:OpenVPN Management Interface Version 1 -- type 'help' for more info
help
Management Interface for OpenVPN 2.1_rc11 x86_64-pc-linux-gnu [SSL] [LZO2] [EPOLL] [
  PKCS11] built on Sep 18 2008
Commands:
auth-retry t          : Auth failure retry mode (none,interact,nointeract).
bytecount n           : Show bytes in/out, update every n secs (0=off).
echo [on|off] [N|all] : Like log, but only show messages in echo buffer.
exit|quit             : Close management session.
forget-passwords      : Forget passwords entered so far.
help                  : Print this message.
hold [on|off|release] : Set/show hold flag to on/off state, or
                       release current hold and start tunnel.
kill cn                : Kill the client instance(s) having common name cn.
kill IP:port           : Kill the client instance connecting from IP:port.
log [on|off] [N|all]  : Turn on/off realtime log display
                       + show last N lines or 'all' for entire history.
mute [n]               : Set log mute level to n, or show level if n is absent.
needok type action     : Enter confirmation for NEED-OK request of 'type',
                       where action = 'ok' or 'cancel'.
needstr type action    : Enter confirmation for NEED-STR request of 'type',
                       where action is reply string.
net                    : (Windows only) Show network info and routing table.
password type p        : Enter password p for a queried OpenVPN password.
pkcs11-id-count        : Get number of available PKCS#11 identities.
pkcs11-id-get index    : Get PKCS#11 identity at index.
client-auth CID KID    : Authenticate client-id/key-id CID/KID (MULTILINE)
client-auth-nt CID KID : Authenticate client-id/key-id CID/KID
client-deny CID KID R  : Deny auth client-id/key-id CID/KID with reason text R
client-kill CID        : Kill client instance CID
client-pf CID          : Define packet filter for client CID (MULTILINE)
signal s               : Send signal s to daemon,
                       s = SIGHUP|SIGTERM|SIGUSR1|SIGUSR2.
state [on|off] [N|all] : Like log, but show state history.
status [n]             : Show current daemon status info using format #n.
test n                 : Produce n lines of output for testing/debugging.
username type u        : Enter username u for a queried OpenVPN username.
verb [n]               : Set log verbosity level to n, or show if n is absent.
version                : Show current version number.
END
quit
Connection closed by foreign host.

```

Chapitre 7

Références

- [1] Site officiel de openvpn. www.openvpn.net.
- [2] Debian lenny. www.debian.org/releases/lenny/.
- [3] Echange de clés diffie-hellman. fr.wikipedia.org/wiki/Echange_de_cles_Diffie-Hellman.
- [4] Site officiel de tunnelblick. code.google.com/p/tunnelblick/.
- [5] Site officiel de viscosity. viscosityvpn.com.
- [6] Site officiel du projet debian. www.debian.org.