OpenVPN

OpenVPN est une solution logicielle open source de création de tunnels sécurisés (VPN). Contrairement aux VPN usuels, elle ne s'appuie pas sur IPSEC mais sur SSL. Elle assure des services d'authentification, de confidentialité et de contrôle d'intégrité.

1. Les modes de fonctionnement OpenVPN

OpenVPN supporte deux types principaux de fonctionnement réseau : le mode point-à-point et le mode site-à-site. Dans tous les cas, des services d'authentification et de confidentialité sont assurés en standard.

a. Authentification

Les extrémités de tunnel, c'est-à-dire les deux machines assurant le cryptage des flux sortants et le décryptage des flux entrants, doivent être mutuellement authentifiées. Il ne faut pas qu'il y ait de doute sur l'authenticité du correspondant. OpenVPN supporte plusieurs modes d'authentification, mais les deux plus courants sont l'authentification par clé partagée, et l'authentification par certificats numérique X509. La première solution est infiniment plus simple à mettre en œuvre mais passe pour être moins sécurisée. La seconde, si elle est recommandée, est toutefois beaucoup plus difficile à déployer si on n'a pas une connaissance intime des infrastructures à clés publiques qui permettent de générer les certificats. Il est souvent préférable d'avoir une solution à clé partagée qui fonctionne correctement plutôt qu'une infrastructure à clé publique bancale mal maitrisée et donc difficile à maintenir.

b. Confidentialité

La confidentialité des communications est assurée par la bibliothèque OpenSSL. Le cryptage des échanges est assuré par l'algorithme Blowfish par défaut, mais les algorithmes symétriques courants sont utilisables (AES notamment).

c. Fonctionnement réseau

Le mode de fonctionnement le plus simple et le plus facile à appréhender est le mode point-à-point dans lequel les deux protagonistes du vpn sont ceux qui doivent communiquer ensemble de façon sécurisée : ils sont à les fois les extrémités de tunnel et les extrémités de trafic. Il est aussi possible de relier deux réseaux entre eux en mode site-à-site. Deux serveurs OpenVPN assurent alors la mise en place du tunnel, mais les extrémités de trafic sont les deux réseaux reliés. Les serveurs OpenVPN assurent alors un rôle de routage entre les réseaux. Enfin, il est possible de faire du VPN d'accès distant dans lequel une machine est reliée à un réseau.

OpenVPN peut fonctionner en mode bridgé, dans ce cas il mettra en connexion deux réseaux distants, un peu comme si on avait ajouté un câble entre les switches des deux réseaux à relier, fût-il un câble de 200 km. Ce mode de fonctionnement peut être considéré comme anecdotique, et le mode routé est de loin le plus utilisé.

Les paquets cryptés sont transportés par UDP par défaut mais l'utilisation de TCP est possible.

2. Création d'un tunnel point-à-point

a. Gestion de l'authentification

La méthode d'authentification par clé partagée suppose la présence d'un fichier de clé au format reconnu par OpenVPN. Ce fichier doit être présent sur le serveur et le client, et donc copié par un moyen sécurisé. (clé usb, scp) Le fichier peut être généré directement par la commande **openvpn**.

Génération du fichier de clé secrète

```
openvpn --genkey --secret fichier_cle
```

Où *fichier_cle* représente le fichier contenant la clé secrète.

Exemple de génération de clé

On génère ici un fichier de clé secrète qui permettra l'authentification entre les machines aux extrémités du tunnel.

```
alpha:/etc/openvpn# openvpn --genkey --secret secret.key
alpha:/etc/openvpn# cat secret.key
#
# 2048 bit OpenVPN static key
#
-----BEGIN OpenVPN Static key V1-----
ae11344ce37de44dcce059ecf9fa573f
a2694d5531bc7ed144a12a099c4ef8ce
(... )
1d37552cd4f29ff6b719588056a60777
579cc2aff71bf339f5293bf08f2ce4df
-----END OpenVPN Static key V1-----
alpha:/etc/openvpn#
```

b. Fichiers de configuration

Les fichiers de configuration se trouvent par défaut dans un répertoire **/etc/openvpn**. Si l'usage veut que les fichiers portent les noms client.conf et serveur.conf, n'importe quel fichier avec l'extension .conf fera l'affaire.

Format du fichier de configuration OpenVPN

```
remote serveur
dev tun
ifconfig IP_locale IP_distante
secret fichier_cle
route réseau_distant masque
```

Fichier de configuration OpenVPN : directives courantes	
remote *serveur*	Sur le client uniquement. *serveur* indique le nom ou l'adresse ip du serveur auquel connecter le VPN.
dev tun	Crée une d'encapsulation de type tunnel (par opposition à l'encapsulation ethernet bridgée).
ifconfig *IP_locale IP_distante*	Établit les adresses locales et distantes des extrémités de trafic. Ces adresses seront visibles sous forme d'interface virtuelle dans la configuration réseau de l'hôte.
secret *fichier_cle*	Indique le fichier contenant la clé partagée, identique sur les deux machines.
route *réseau_distant masque*	Paramètre client : indique l'adresse du réseau privé derrière le serveur pour que le trafic à destination de ce réseau soit correctement routé par le VPN.

Exemple de fichiers de configuration OpenVPN

Fichier de configuration côté serveur.

```
alpha:/etc/openvpn# cat server.conf
dev tun
ifconfig 10.8.0.1 10.8.0.2
secret secret.key
```

Fichier de configuration côté client.

```
beta:/etc/openvpn# cat client.conf
remote alpha
```

```
dev tun
ifconfig 10.8.0.2 10.8.0.1
secret secret.key
route 192.168.1.0 255.255.255.0
```

c. Mise en œuvre du tunnel vpn

Une fois les fichiers créés sur le serveur et le client, il suffit de démarrer de part et d'autre le service par son script de démarrage.

La validation de fonctionnement peut se faire par un ping entre les deux adresses de tunnel. Une capture de trames permettra aussi d'observer un trafic entre les deux machines sur le port UDP/1194 par défaut.

Exemple de test d'un tunnel point-à-point

On lance le service par son script normalisé, on vérifie la présence d'une interface virtuelle, et on contrôle le fonctionnement du tunnel par un trafic quelconque.

```
beta:~# ifconfig tun0
tun0: erreur lors de la recherche d'infos sur l'interface: Périphérique non trouvé
 beta:~# /etc/init.d/openvpn start
Starting virtual private network daemon: client.
beta:~# ifconfig tun0
tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet adr:10.8.0.2  P-t-P:10.8.0.1  Masque:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:100
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

beta:~# ping 10.8.0.1
PING 10.8.0.1 (10.8.0.1) 56(84) bytes of data.
64 bytes from 10.8.0.1: icmp_seq=1 ttl=64 time=0.864 ms
```