

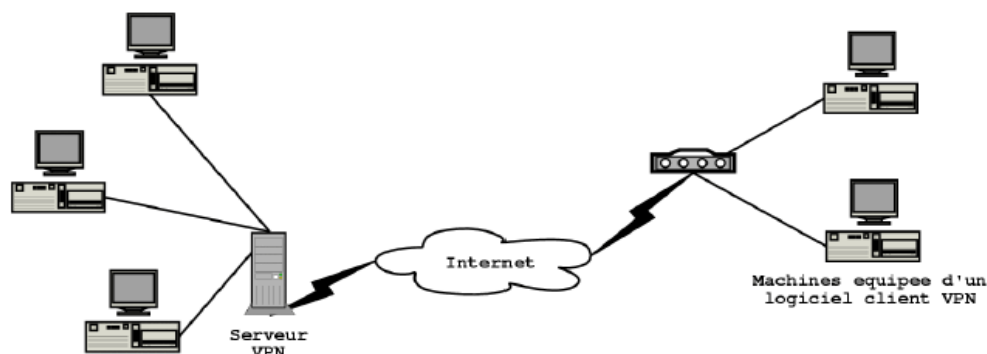
 Annexe - VPN avec OpenVPN (Virtual Private Network)	
Sommaire :	
I - Introduction.....	1
I.1. Présentation.....	1
I.2. OpenVPN.....	2
I.3. IPSec.....	2
I.4. Objectif.....	2
II - Serveur OpenVPN.....	2
II.1. Installation d'OpenVPN sur le serveur.....	2
II.2. Configuration d'OpenVPN sur le serveur.....	2
II.3. Gestion des certificats SSL.....	4
II.4. Démarrage du serveur OpenVPN.....	4
II.5. Pour aller plus loin.....	5
III - Client OpenVPN.....	5
III.1. Installation d'OpenVPN sur le client.....	5
III.2. Gestion des certificats coté client.....	5
III.3. Test du client OpenVPN.....	6
IV - Clients OpenVPN graphiques.....	7
IV.1. Client OpenVPN pour Windows.....	7
IV.2. Client OpenVPN pour Ubuntu avec Gnome.....	7

I - Introduction

I.1. Présentation

L'intérêt d'un réseau privé virtuel **VPN** (Virtual Private Network), c'est d'offrir une **communication sécurisée** entre **deux sites distants** au **travers d'Internet**.

Au moyen d'un **VPN**, un **télétravailleur** peut accéder à distance au réseau du quartier général de sa société. Via Internet, il est possible de construire un **tunnel VPN sécurisé** entre le **PC du télétravailleur** et un serveur **VPN** installé au quartier général de la société.



Les **VPN** sont donc élaborés à partir de l'infrastructure des réseaux publics existants mais sont protégés par des **techniques d'encryptage des données** afin de garantir la **confidentialité**. En effet, cette solution est moins coûteuse que la mise en place de liaisons spécialisées mais beaucoup plus critique pour l'information.

Si l'on passe en revue les protocoles disponibles on en trouve essentiellement deux : **OpenVPN** et **IPSec**.

1.2. OpenVPN

OpenVPN est l'un des **serveurs de VPN libres les plus populaires** en raison de sa simplicité de configuration et d'utilisation, mais par contre il n'est pas standardisé.

Lorsqu'on se connecte à **OpenVPN**, le système crée une nouvelle **interface réseau virtuelle**, qui fonctionne comme n'importe quelle autre interface et qui a une **adresse IP privée**, connectée à un **réseau virtuel** géré par le serveur **OpenVPN**. Les communications sur ce **réseau virtuel** sont chiffrées par **SSL/TLS**.

Les communications d'**OpenVPN** se font au travers d'un **tunnel** qui utilise **UDP** (port **1194**) ou **TCP** (port **443**). C'est par dessus cette communication classique qu'un **réseau virtuel** est créé.

1.3. IPSec

IPSec est standardisé, il a l'avantage d'être supporté par toutes les plateformes grand public mais aussi professionnelles comme **Cisco**. Il utilise obligatoirement les ports **UDP 500** et **1701** et parfois **4500** pour passer le **NAT**. Par contre sa configuration est assez complexe.

1.4. Objectif

Notre objectif est de connecter un client nomade à un réseau d'entreprise de la manière la plus simple possible tout en garantissant un niveau de sécurité satisfaisant.

Notre **serveur VPN** aura pour adresse **10.8.0.1 / 24** et le réseau interne **192.168.40.0 / 24** sera joignable au travers de cette connexion VPN. Les **clients VPN** auront une adresse dans le réseau **10.8.0.0 / 24**.

II - Serveur OpenVPN

II.1. Installation d'OpenVPN sur le serveur

Avec **OpenVPN**, le même exécutable est utilisé pour le **serveur** et les **clients**. Sur le **serveur**, il faut installer le paquetage **openvpn** :

```
#apt-get install openvpn
```

Remarque : Des exemples de fichiers de configuration sont disponibles dans le dossier **/usr/share/doc/openvpn/examples/sample-config-files**. Pour décompresser les fichiers portant l'extension **.gz**, il suffit de taper la commande **gzip -d fichier.gz**.

II.2. Configuration d'OpenVPN sur le serveur

Les réseaux virtuels d'**OpenVPN** sont paramétrés au travers de fichiers de configuration (**.conf**) contenus dans **/etc/openvpn**. Généralement la configuration d'**OpenVPN** doit contenir les lignes suivantes (**/etc/openvpn/serveur.conf**) :

```
#serveur UDP/1194
dev tun
```

```
mode server
proto udp
port 1194
#tls-server
#serveur TCP/443
#mode server
#dev tun
#proto tcp
#port 443
# clés et certificats
ca /etc/openvpn/server/bts-informatique-cacert.pem
cert /etc/openvpn/server/serveur-linux-cert.pem
key /etc/openvpn/server/serveur-linux-key.pem
dh /etc/openvpn/server/dh2048.pem
#tls-auth ta.key 0
tls-version-min 1.2
cipher AES-256-CBC
# network
server 10.8.0.0 255.255.255.0
max-clients 100
keepalive 10 120
push "redirect-gateway def1 bypass-dhcp"
push "route 192.168.40.0 255.255.255.0"
#Security
persist-tun
persist-key
#compress lz4
comp-lzo
user nobody
group nogroup
# Log
verb 3
mute 20
# debug
status openvpn-status.log
log-append /var/log/openvpn.log
```

Rôle des différents paramètres :

- Le **port** peut être modifié, **1194** pour **UDP** et **443** pour **TCP** sont les ports officiellement attribués à **OpenVPN** ;
- Le **protocole** peut être **TCP** ou **UDP** ;
- **dev** peut prendre 2 valeurs : **tun** qui permet de créer un tunnel IP (liaison point à point de niveau 3 OSI) ou **tap** qui permet de créer un tunnel Ethernet (une interface réseau de niveau 2 OSI qu'il faudra configurer) ;
- Les paramètres **ca**, **cert**, **key** et **dh** sont liés au certificat **SSL** à utiliser ;
- Avec la directive **comp-lzo** , **OpenVPN** compresse les paquets traversant le lien VPN, améliorant ainsi le débit ;
- **persist-key** et **persist-tun**, permettent de conserver certains éléments lors d'un redémarrage d'**OpenVPN** ;
- **verb** définit le niveau de verbosité de **0** à **9** (bavardage) d'**OpenVPN** dans les **logs** ;
- La directive **mute**, permet de limiter les répétitions à inscrire dans les logs d'un message qui se répète ;
- La directive **server**, permet de fixer la configuration réseau. Ici le serveur aura l'adresse **10.8.0.1 / 24** et les clients auront une adresse dans le réseau **10.8.0.0 / 24** ;
- La directive **push**, permet de router les réseaux internes. **push "route 192.168.40.0 255.255.255.0 "** indique que le réseau interne **192.168.40.0 / 24** sera joignable au

travers de cette connexion VPN. Si tous les flux doivent passer par le VPN, alors il faut indiquer aux clients que le serveur VPN doit être configuré comme passerelle par défaut avec la directive : **push "redirect-gateway def1 bypass-dhcp"** ;

- Pour des raisons de sécurité, les clients ne peuvent pas se voir entre eux. Si on désire qu'ils puissent se voir on rajoute la directive **client-to-client** ;
- La directive **max-clients**, permet de limiter le nombre de clients autorisés à se connecter au serveur **OpenVPN** ;
- La directive **log-append** permet de spécifier le fichier de **logs**.

Ce fichier de configuration permet donc de créer un **serveur VPN** routé basée sur le protocole **UDP** et utilisant le port **1194**. Les clients obtiendront une nouvelle adresse IP dans la plage **10.8.0.0/24**.

II.3. Gestion des certificats SSL

OpenVPN est livré avec un ensemble de scripts permettant de gérer ces certificats, nommé **Easy-RSA** dans sa **version 2**. L'ensemble des scripts d'**Easy-RSA** est disponible dans **/usr/share/easy-rsa/**.

Dans notre cas, nous allons générer les certificats avec l'outil **TinyCA** utilisé dans un précédent TP. Grâce à cet outil, nous disposons des fichiers suivants :

- **bts-informatique-cacert.pem**, le certificat de notre **CA** ;
- **serveur-linux-cert.pem**, le certificat de **srv-linux** ;
- **serveur-linux-key.pem**, la clé privée de **srv-linux** ;
- **client-linux-cert.pem**, le certificat de **client-linux** ;
- **client-linux-key.pem**, la clé privée de **client-linux** ;
- **client-w7-cert.pem**, le certificat de **client-w7** ;
- **client-w7-key.pem**, la clé privée de **client-w7**.

Il faudra par un moyen sécurisé, copier ces différents fichiers dans le répertoire **/etc/openvpn/server** de **serveur-linux**.

D'autre part, il faut créer sur le serveur, un paramètre « **Diffie Hellman** » au moyen d'**OpenSSL**, qui servira à générer les clés de session. En effet, la cryptographie « asymétrique », très gourmande en ressources, n'est utilisée que dans la phase d'authentification et d'établissement du tunnel. Par la suite, les données sont chiffrées de manière symétrique avec une clé partagée, communiquée par le serveur au client (à travers un chiffrement asymétrique, bien entendu). Pour cela on exécute la commande suivante (qui dure longtemps) :

```
openssl dhparam -out /etc/openvpn/server/dh2048.pem 2048
```

On peut maintenant tester la configuration en saisissant la commande suivante :

```
cd /etc/openvpn/server  
openvpn --config server.conf &
```

II.4. Démarrage du serveur OpenVPN

Si tout va bien, on peut lancer le serveur **VPN** en saisissant l'une des commandes suivantes :

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

Une fois le serveur **OpenVPN** en fonction, on peut constater à l'aide de la commande **ip**

route, que la route pour le réseau virtuel est en place, il utilise l'interface virtuelle **tun0** :

```
#ip route
default via 192.168.40.254 dev eth0
10.8.0.0/24 via 10.8.0.2 dev tun0
10.8.0.2 dev tun0 proto kernel scope link src 10.8.0.1
192.168.40.0/24 dev eth0 proto kernel scope link src 192.168.40.1
```

II.5. Pour aller plus loin

Si le **serveur VPN** joue le rôle de **passerelle**, il faudra activer la transmission de paquets en modifiant le fichier **/etc/sysctl.conf** et en décommentant la ligne :

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

On active la modification en exécutant la commande **sysctl -p**.

Si on veut rendre accessible un réseau situé derrière notre passerelle VPN (sur le réseau **192.168.10.0/24** par exemple), il faudra rajouter la directive suivante dans le fichier de configuration du serveur :

```
push "route 192.168.10.0 255.255.255.0"
```

III - Client OpenVPN

III.1. Installation d'OpenVPN sur le client

Sur le **client**, il faut installer les paquets **openvpn** et **openssl** :

```
#apt-get install openvpn openssl
```

III.2. Gestion des certificats coté client

Une fois le certificat de l'autorité de certification et le certificat serveur créés, il faut créer un **certificat** pour **chaque client**. Cela a été réalisé auparavant avec **TinyCA**.

On copie les fichiers nécessaires (le certificat de la CA, le certificat et la clé du client) dans le dossier **/etc/openvpn/client** du **client**.

On va ensuite dans le répertoire **/etc/openvpn/client/** et on crée par exemple le fichier **client-linux.conf** contenant les lignes suivantes :

```
#client UDP/1194
client
dev tun
proto udp
port 1194
#client TCP/443
#client
#dev tun
#proto tcp-client
resolv-retry infinite
cipher AES-256-CBC
auth SHA512
auth-nocache
#tls-version-min 1.2
```

```
# serveur VPN
remote 192.168.40.1 1194
#remote 192.168.40.1 443
# clés et certificats
ca bts-informatique-cacert.pem
cert client-linux-cert.pem
key client-linux-key.pem
#Security
nobind
persist-tun
persist-key
#compress lz4
comp-lzo
# Log
verb 3
# debug
status openvpn-status.log
log-append /var/log/openvpn.log
```

Puis on valide cette configuration en saisissant la commande suivante :

```
#cd /etc/openvpn/client
#openvpn --config client-linux.conf &
```

Puis on peut visualiser les logs dans le fichier **/var/log/openvpn.log** en tapant la commande **tail -f /var/log/openvpn.log**.

III.3. Test du client OpenVPN

Une fois le client **OpenVPN** en fonction, on peut constater à l'aide de la commande **ip route**, que la route pour le réseau virtuel est en place, il utilise l'interface virtuelle **tun0** :

```
#ip route
default via 192.168.40.254 dev eth0
10.8.0.5 dev tun0 proto kernel scope link src 10.8.0.6
192.168.40.1 via 192.168.40.254 dev eth0
10.8.0.1 via 10.8.0.5 dev tun0
192.168.40.0/24 via 10.8.0.5 dev tun0
192.168.40.0/24 dev eth0 proto kernel scope link src 192.168.40.11
0.0.0.0/1 via 10.8.0.5 dev tun0
128.0.0.0/1 via 10.8.0.5 dev tun0
```

Il suffit d'exécuter une commande ping vers le serveur :

```
#ping -c 1 10.8.0.1
```

De même depuis le serveur, une requête ping fonctionne vers le client :

```
#ping -c 1 10.8.0.6
```

IV - Clients OpenVPN graphiques

IV.1. Client OpenVPN pour Windows

Il faut télécharger le logiciel depuis le site d'**OpenVPN** puis l'installer.

Il suffit ensuite de placer les fichiers de configuration ainsi que les **clés** et les **certificats** dans le dossier **C:\Programmes\OpenVPN\config**. Il faut que l'extension du fichier de configuration du client soit **ovpn**.

On lance **OpenVPN**, puis à l'aide d'un **clic droit**, on peut se connecter au **serveur VPN**.

IV.2. Client OpenVPN pour Ubuntu avec Gnome

Il faut installer les packages **network-manager-openvpn** et **network-manager-openvpn-gnome** :

#apt install network-manager-openvpn network-manager-openvpn-gnome

On commence par **copier** tous les fichiers sur le client dans un dossier puis depuis le menu **Paramètres Système / Réseau**, Il faut **ajouter un VPN** puis **importer depuis un fichier** :



On sélectionne le fichier de configuration du client :



On saisie la clé privée utilisée lors de la génération des clés du client. Il ne reste plus qu'à se connecter au serveur VPN à l'aide du menu de **network-manager**.