

I- Etude du VPN

a. Qu'est-ce qu'un VPN ?

Un VPN (Virtual Private Network = Réseau Privé Virtuel) permet de créer une connexion sécurisée entre un ordinateur et un serveur VPN. Ce dernier servira de relai entre l'ordinateur nomade et le réseau local de l'entreprise. De plus un des gros avantages de ce système est que toutes les données entrantes et sortantes et qui passent par ce serveur sont cryptées.

b. Solutions existantes et choix d'OpenVpn :

Il existe plusieurs solutions de VPN dont les offres ont tendances à fleurir sur le marché. Il existe entre autres :

- HydeMyAss
- IPvanish
- VPNTunnel
- OpenVPN

Après quelques recherches concernant les solutions existantes, nous avons porté notre choix sur **OpenVpn** pour les raisons suivantes :

- Créé en 2002, Open est un outil open source utilisé pour construire des VPNs site à site avec le protocole SSL/TLS ou avec des clefs partagées. Son rôle est de "tunneliser", de manière sécurisée, des données sur un seul port TCP/UDP à travers un réseau non sûr comme Internet et ainsi établir des VPNs.
- La grande force d'OpenVPN est d'être extrêmement facile à installer et à configurer, ce qui est rarement le cas pour des outils utilisés pour créer des VPNs.
- OpenVPN est le premier protocole VPN conçu pour les réseaux modernes haut débit, mais il n'est pas supporté par les appareils mobiles et les tablettes. OpenVPN propose un cryptage 256 bits et est extrêmement stable et rapide sur les réseaux longue distance et à latence élevée. Il offre une plus grande sécurité que PPTP et nécessite moins d'utilisation de processeur que L2TP/IPsec. OpenVPN est le protocole recommandé pour les ordinateurs de bureau, y compris Windows, Mac OS X et Linux.
- OpenVPN est relativement facile à mettre en oeuvre ;
- C'est un logiciel libre multiplateforme ;
- Il correspond à la plupart de besoins de base ;

II- Préparation du serveur et installation d'OpenVpn :

Tout d'abord nous avons ouvert le Terminal nous sommes mis en root. Puis nous avons mis à jour des paquets de Ubuntu 12.04.

```
sudo apt-get update
```

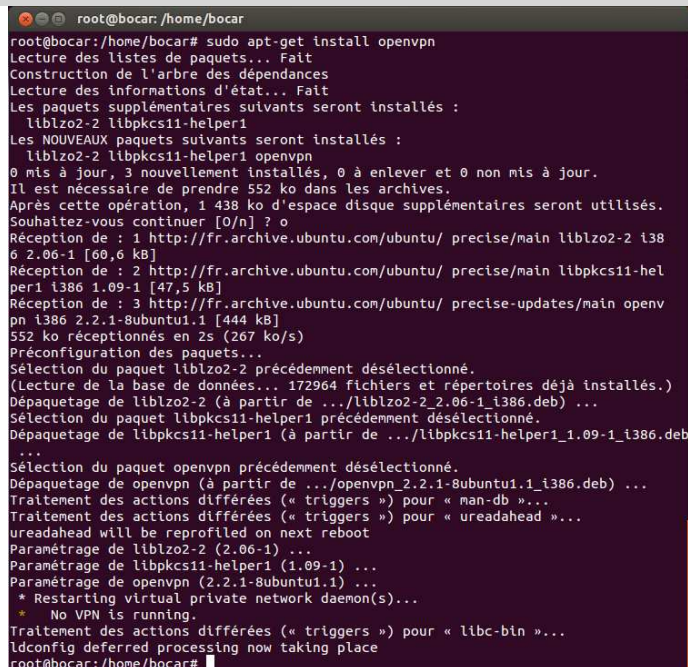
Nous avons utilisé une commande pour vérifier la présence d'un logiciel ou pas dans notre système :
openvpn et openssl (pour la sécurisation des données).

```
dpkg -l | grep nom-du-logiciel
```

OpenSSL étant souvent installé par défaut sur les machines, nous n'avons pas eu à le réinstaller après vérification par la commande ci-dessus qu'il est bien présent.

Ce qui n'est pas le cas d'Openvpn dont nous avons procédé à l'installation par la commande suivante :

```
sudo apt-get install openvpn
```



```
root@bocar: /home/bocar# sudo apt-get install openvpn
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  liblz2-2 libpkcs11-helper1
Les NOUVEAUX paquets suivants seront installés :
  liblz2-2 libpkcs11-helper1 openvpn
0 mis à jour, 3 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 552 ko dans les archives.
Après cette opération, 1 438 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer [O/n] ? o
Réception de : 1 http://fr.archive.ubuntu.com/ubuntu/ precise/main liblz2-2 i386 2.06-1 [60,6 kB]
Réception de : 2 http://fr.archive.ubuntu.com/ubuntu/ precise/main libpkcs11-helper1 i386 1.09-1 [47,5 kB]
Réception de : 3 http://fr.archive.ubuntu.com/ubuntu/ precise-updates/main openvpn i386 2.2.1-8ubuntu1.1 [444 kB]
552 ko réceptionnés en 2s (267 ko/s)
Préconfiguration des paquets...
Sélection du paquet liblz2-2 précédemment désélectionné.
(Lecture de la base de données... 172964 fichiers et répertoires déjà installés.)
Dépaquetage de liblz2-2 (à partir de .../liblz2-2_2.06-1_i386.deb) ...
Sélection du paquet libpkcs11-helper1 précédemment désélectionné.
Dépaquetage de libpkcs11-helper1 (à partir de .../libpkcs11-helper1_1.09-1_i386.deb) ...
Sélection du paquet openvpn précédemment désélectionné.
Dépaquetage de openvpn (à partir de .../openvpn_2.2.1-8ubuntu1.1_i386.deb) ...
Traitement des actions différées (« triggers ») pour « man-db »...
Traitement des actions différées (« triggers ») pour « ureadahead »...
ureadahead will be reprofiled on next reboot
Paramétrage de liblz2-2 (2.06-1) ...
Paramétrage de libpkcs11-helper1 (1.09-1) ...
Paramétrage de openvpn (2.2.1-8ubuntu1.1) ...
* Restarting virtual private network daemon(s)...
* No VPN is running.
Traitement des actions différées (« triggers ») pour « libc-bin »...
ldconfig deferred processing now taking place
root@bocar: /home/bocar#
```

III- Génération des certificats et clés d'authentification

OpenVPN peut fonctionner avec plusieurs types d'authentification. Nous utiliserons l'authentification par clés et certificats, plus sûre que le classique login/mot de passe.

L'installation d'OpenVPN a créé un dossier (sous Ubuntu 12.04) dans **/usr/share/doc/openvpn/examples/easy-rsa/2.0/** contenant tous les scripts permettant de générer facilement tous les certificats et clés d'authentification nécessaires au fonctionnement d'OpenVPN. Tout le processus décrit ci-après doit s'effectuer en tant que root.

Avant toute chose, nous avons créé un dossier **easy-rsa** dans le répertoire d'OpenVPN et copié les scripts originaux dedans afin de centraliser les applications et scripts :

Création du répertoire easy-rsa :

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

Copie des scripts :

```
> cp /usr/share/doc/openvpn/examples/easy-rsa/2.0/* /etc/openvpn/easy-rsa/
```

Nous avons créé ensuite un dossier **keys** devant contenir les différents certificats et clés générés :

Création du répertoire easy-rsa :

```
> mkdir /etc/openvpn/easy-rsa/keys/
```

a. Initialisation des variables de génération

À partir du dossier **/etc/openvpn/easy-rsa/**, nous avons dans un premier temps édité le fichier **vars** afin d'initialiser différentes variables servant à la génération des certificats :

Création du répertoire easy-rsa :

```
> sudoedit /etc/openvpn/easy-rsa/vars
```

Informations renseignées dans le fichier vars

```
Export KEY_CONFIG=/etc/openvpn/easy-rsa/openssl-1.0.0.cnf
```

```
Export KEY_DIR=/etc/openvpn/easy-rsa/keys/
```

```
Export KEY_COUNTRY=FR
```

```
Export KEY_PROVINCE=IDF
```

```
Export KEY_CITY=PARIS
```

```
Export KEY_ORG=vpnserv
```

```
Export KEY_EMAIL=vpnserv@integrale.lan
```

Enfin nous avons exécuté le script afin d'initialiser les variables:

Exécution du script :

```
> ./vars
```

N.B. : La commande est bien : point espace point/vars

b. Génération du certificat et de la clé d'autorité de certification

OpenVPN fonctionne sous un mode PKI (Public Key Infrastructure). Selon ce mode, le serveur et chaque client possède un certificat (appelé également clé publique) et une clé privée qui leur sont propres. Un certificat d'autorité de certification (master CA) et une clé privée sont utilisés pour signer les certificats du serveur et de chaque client. Ce master CA permet une authentification bidirectionnelle : chacun des clients et serveur authentifie donc l'autre réciproquement en vérifiant dans un premier temps que le certificat qu'ils proposent a bien été signé par le master CA.

Pour générer ce master CA et la clé correspondante, nous avons exécuté les scripts suivants à partir du dossier **/etc/openvpn/easy-rsa** :

Exécution du script clean-all :

```
> ./clean-all
```

```
> ./build-ca
```

L'exécution du script build-ca a eu pour effet la création du certificat **ca.crt** et de la clé **ca.key** dans le répertoire **/etc/openvpn/easy-rsa/keys**.

c. Génération du certificat et de la clé pour le serveur

La génération du certificat et de la clé du serveur VPN se fait simplement, par l'exécution du script build-key-server, toujours à partir du dossier **/etc/openvpn/easy-rsa** :

Génération du certificat et de la clé pour le serveur :

> ./build-key-server vpnintegrale

Différentes informations nous sont demandées dont voici un récapitulatif après avoir validé. Et nous avons répondu « y » pour « yes » aux questions

```
The Subject's Distinguished Name is as follows
countryName       :PRINTABLE:'FR'
stateOrProvinceName :PRINTABLE:'IDF'
localityName      :PRINTABLE:'Paris'
organizationName  :PRINTABLE:'integrale'
commonName        :PRINTABLE:'vpnintegrale'
name              :PRINTABLE:'admin'
emailAddress      :IA5STRING:'integrale@integrale.lan'
Certificate is to be certified until May  3 16:38:48 2023 GMT (3650 days)
Sign the certificate? [y/n]:y
```

Ce script conduit à la création des fichiers **vpnintegrale.crt** et **vpnintegrale.key** dans le dossier **/etc/openvpn/easy-rsa/keys**.

d. Génération des certificats et clés pour les clients

De la même façon, ils sont générés par l'exécution du script build-key à partir du dossier **/etc/openvpn/easy-rsa** :

Génération des certificats et clés pour le client1:

> ./build-key client1

Génération des certificats et clés pour le client2:

> ./build-key client2

```
The Subject's Distinguished Name is as follows
countryName       :PRINTABLE:'FR'
stateOrProvinceName :PRINTABLE:'IDF'
localityName      :PRINTABLE:'Paris'
organizationName  :PRINTABLE:'integrale'
commonName        :PRINTABLE:'client1'
name              :PRINTABLE:'client1'
emailAddress      :IA5STRING:'integrale@integrale.lan'
Certificate is to be certified until May  3 17:00:33 2023 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
```

N.B. : Pour le paramètre CommonName nous avons renseigné le même nom que le nom du client dans la commande de création du certificat, à savoir client1 pour client1.

Nous répétons autant de fois que possible cette manipulation en fonction du nombre de clients à paramétrer pour se connecter au serveur.

L'exécution de ce script a permis de créer les fichiers client1.crt et client1.key

e. Génération des paramètres de Diffie-Hellman

Le protocole Diffie-Hellman est un protocole de cryptographie utilisé dans les échanges de clés. Les paramètres de Diffie-Hellman sont générés par l'exécution du script `build-dh` à partir du dossier `/etc/openvpn/easy-rsa` :

Génération des paramètres de Diffie-Hellman:

```
> ./build-dh
```

Une fois le script exécuté nous avons une sortie comme sur cette image :

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

Le résultat a été la création du fichier `dh1024.pem` dans le dossier `/etc/openvpn/easy-rsa/keys`.

IV- Configuration du serveur OpenVPN et des Clients:

Après avoir fini de créer les clés et certificats d'authentification nous sommes passés à la configuration du serveur et des clients.

Afin de configurer au mieux le serveur et les clients, il a été nécessaire de déplacer les différents fichiers de configuration nécessaires dans `/etc/openvpn/`.

Des exemples de fichiers de configuration sont présents dans le dossier **`/usr/share/doc/openvpn/examples/sample-config-files/`**

Copie du fichier de **confserver.conf** dans le dossier **openvpn** :

```
>cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz /etc/openvpn/
```

Copie du fichier de **confclient.conf** dans le dossier openvpn :

```
>cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf /etc/openvpn/
```

Nous avons procédé ensuite à l'extraction du fichier `server.conf.gz`

```
>gunzip /etc/openvpn/server.conf.gz
```

Nous avons vérifié qu'ils sont bien copiés

1. Configuration Serveur

La mise en route du serveur a entraîné l'attribution automatique d'une adresse IP à l'interface tun0 du serveur. Cette adresse IP est toujours la première adresse (en .1) du réseau annoncé dans le fichier de configuration.

Dans notre cas, ayant défini notre réseau Vpn en 10.8.0.0. L'adresse IP du serveur Vpnest donc 10.8.0.1.

Nous avons renseigné les bons paramètres au fichier **/etc/openvpn/server.conf** :

Explication du contenu du fichier de conf server

```
#numéro du port utilisé
port1194
#protocole de communication
protoudp
#type d'interface
devtun
#emplacement du master CA
ca /etc/openvpn/easy-rsa/keys/ca.crt
#emplacement du certificat du serveur
cert /etc/openvpn/easy-rsa/keys/vpnintegrale.crt
#emplacement de la clé du serveur
key /etc/openvpn/easy-rsa/keys/vpnintegrale.key
#emplacement du fichier Diffie-Hellman
dh /etc/openvpn/easy-rsa/keys/dh1024.pem
#quelle sera l'adresse du réseau virtuel créé par le VPN
#l'adresse du serveur VPN sera ici 10.8.0.1
server10.8.0.0255.255.255.0
keepalive10120
#type d'encryptage des données
cipher AES-128-CBC
#activation de la compression
comp-lzo
#nombre maximum de clients autorisés
max-clients 10
#pas d'utilisateur et groupe particuliers pour l'utilisation du VPN
user nobody
group nobody
#pour rendre la connexion persistante
persist-key
persist-tun
#fichier de log
status openvpn-status.log
log openvpn.log
#niveau de verbosité
verb5
```

La configuration est terminée. Nous pouvons démarrer le serveur par la commande :

Démarrage du serveur:

```
> / etc/init.d/openvpnstart
```


Nous avons vérifié que tout s'est bien passé jusqu'à présent en vérifiant la création et la bonne configuration de l'interface tun0 :

```
bocar@bocar:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr d4:3d:7e:28:7f:de
          inet addr:192.168.1.53  Bcast:192.168.1.255  Masque:255.255.255.0
          adr inet6: fe80::d63d:7eff:fe20:7fde/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Packets reçus:226 erreurs:0 :0 overruns:0 frame:0
          TX packets:77 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:18509 (18.5 KB) Octets transmis:10561 (10.5 KB)
          Interruption:40 Adresse de base:0x4000

lo        Link encap:Boucle locale
          inet addr:127.0.0.1  Masque:255.0.0.0
          adr inet6: ::1/128 Scope:Hôte
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          Packets reçus:151 erreurs:0 :0 overruns:0 frame:0
          TX packets:151 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          Octets reçus:12744 (12.7 KB) Octets transmis:12744 (12.7 KB)

tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          -00
          inet addr:10.8.0.1  P-t-P:10.8.0.2  Masque:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          Packets reçus:0 erreurs:0 :0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:100
          Octets reçus:0 (0.0 B) Octets transmis:220 (220.0 B)

bocar@bocar:~$
```

2. Configuration des clients

La connexion au serveur VPN est possible via des clients fonctionnant aussi bien sous Linux que sous Windows. Dans notre cas nos clients sont sous Windows.

Tout d'abord, pour fonctionner, les clients ont besoin de 4 fichiers provenant du serveur :

- 1. ca.crt
- 2. client1.crt
- 3. client1.key
- 4. le fichier de configuration client.conf

Après avoir récupéré ces fichiers sur le poste client nous effectuons les paramétrages nécessaires principalement le fichier client.conf permettant de se connecter au serveur défini précédemment :

Explication du contenu du fichier de conf client (Exemple du client1)

```
#pour signaler que c'est un client !
client
#type d'interface
devtun
#protocole de communication
protoudp
#adresse ip publique du réseau dans lequel le serveur est installé + port identique au serveur
remoteA.B.C.D 1194
#tentative de connexion infinie
resolv-retryinfinite
nobind
#pour rendre la connexion persistante
persist-key
```

```
persist-tun
#pour cacher les avertissements
mute-replay-warnings
#emplacement du master CA
ca ca.crt
#emplacement du certificat client
cert client1.crt
#emplacement de la clé privée du client
key client1.key
#type d'encryptage des données
cipher AES-128-CBC
#activation de la compression
comp-lzo
#niveau de verbosité
verb5
```

Il est important de noter pour le client windows qu'après avoir édité le fichier client.conf , nous avons modifié son extension en .ovpn, il devient alors client.ovpn

Installation Openvpn Client :

Sur chaque poste client Windows alors nous avons installé une application spécifique, OpenVpn GUI téléchargée ici : http://openvpn.net/?option=com_content&id=357

Elle se présente ainsi au premier démarrage :



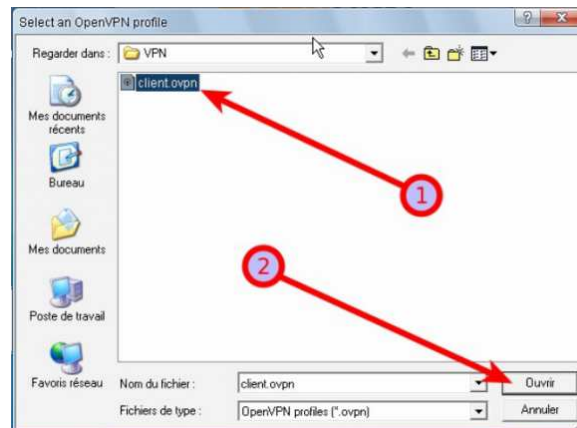
Nous avons déclaré le Vpn sur Windows en cliquant sur le bouton + pour ajouter une nouvelle connexion VPN.



Ensuite nous avons sélectionné l'option d'importation locale (1) et on cliqué sur Import (2):



Nous avons sélectionné ensuite le fichier **client.ovpn** qui se trouve dans le dossier des fichiers importés du serveur



Enfin nous avons sauvegardé la configuration :



Et la nouvelle connexion Vpn apparait dans la fenêtre :



V- Tests de connexion :

a. En local

Nous avons d'abord fait un test de connexion en local en renseignant dans le fichier client.ovpn (ouvert dans un éditeur texte) l'adresse du serveur sur le réseau local.



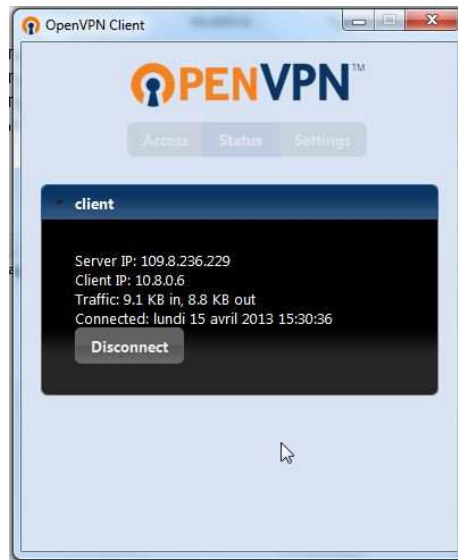
b. Test de connexion à l'extérieur du réseau

Pour procéder à une connexion de l'extérieur nous avons renseigné l'adresse IP publique du routeur d'Intégrales et le port 1194 dans le fichier client.ovpn.

Ensuite nous avons effectué une redirection de port sur le routeur.



Et après avoir lancé la connexion, nous remarquons qu'elle a réussi :



VI- Mise en production et Administration :

Serveur installé, attribution d'une adresse IP dans le plan d'adressage du réseau, ajouté au réseau, exclusion d'adresse IP sur le serveur DHCP pour pas de conflits.

Tests d'intégration :

Application accessible sur le réseau local : <http://10.8.0.3/Medianet>

Administration graphique

VII- Recherche de formations sur internet pour la veille technologique :

- Page documentation sur Openvpn sur le site de la communauté francophone d'utilisateurs d'Ubuntu
<http://doc.ubuntu-fr.org/openvpn>
- Un cours en ligne sur l'implémentation d'Openvpn sous un Linux Debian
<http://www.authsecu.com/cours-formation-elearning/open-vpn-debian.php>
- Le site officiel des développeurs d'**OpenVPN Technologies, Inc.** avec liste de diffusion ou de discussion, support et forums
- <http://openvpn.net>

⋮
<http://openvpn.net/index.php/open-source/faq.html>

<http://openvpn.net/index.php/open-source/documentation.html>

Nous sommes abonnés aux listes de diffusions listes de diffusion sur le site d'Openvpn pour être tenu au courant des informations récentes concernant le logiciel.

VIII- Conclusion

Ce travail a été pour nous très enrichissant. Face à la nombreuse documentation existante, notre principal problème a été d'abord de faire un tri par rapport aux informations dont nous avons vraiment besoin, compte tenu de la configuration du contexte qui nous a été proposé.

Au-delà du service des commerciaux, la mise en place du tunnel Vpn présente un intérêt certain pour l'ensemble de l'infrastructure d'Intégrales Technologies. En effet, il peut permettre aux administrateurs du réseau d'accéder aux autres serveurs comme l'Active Directory et permettre ainsi une gestion à distance. Il suffira juste de créer et configurer de nouveaux clients.

