

Installation sécurisée d'OpenVPN

Alice de Bignicourt

CNRS/UREC

7 juin 2005

OpenVPN est un logiciel « open source » permettant de créer un VPN (Virtual Private Network ou réseau virtuel privé) basé sur SSL. Il peut être utilisé afin de relier deux, ou plus, réseaux via un tunnel chiffré à travers l'Internet. Il est facile à mettre en place, robuste et offre beaucoup de configurations possibles.

L'avantage d'OpenVPN par rapport à une solution IPsec est la simplicité de mise en œuvre.

Par ailleurs, OpenVPN n'utilise pas de protocole de communication standard. Il faut donc utiliser un client OpenVPN pour se connecter à un serveur OpenVPN.

Dans cet article il n'est pas question de refaire la documentation d'OpenVPN. Il s'adresse à un administrateur et tente :

- de lui expliquer les grands principes d'OpenVPN
- de l'aider à mettre en place une politique de sécurité
- de l'aider à configurer OpenVPN (en mode routeur) sur une machine Linux, et de lui donner un exemple d'utilisation avec des postes nomades

Sommaire

1	GENERALITES -----	2
2	OPTIONS D'INSTALLATION -----	3
2.1	CHOIX DU PROTOCOLE-----	3
2.2	CHOIX DE LA METHODE D'AUTHENTIFICATION-----	3
2.3	CHOIX DU TYPE DE TUNNEL-----	4
3	CONFIGURATION SECURISEE D'OPENVPN -----	4
3.1	SECURISATION DU SERVEUR-----	4
3.1.1	<i>Utilisation d'une clé partagée</i> -----	5
3.1.2	<i>Redéfinir la racine du serveur</i> -----	5
3.2	POLITIQUE DE FILTRAGE-----	6
3.2.1	<i>Filtrage sur le serveur OpenVPN</i> -----	7
3.2.2	<i>Filtrage sur le routeur</i> -----	7
4	EXEMPLE DE CONFIGURATION SECURISEE -----	7
4.1	VERIFICATION DES DROITS D'ACCES (--CLIENT-CONNECT <SCRIPT>)-----	8
4.2	ATTRIBUTION DES ADRESSES IP-----	9
4.2.1	<i>Le script (--client-connect <script>)</i> -----	10
4.2.2	<i>Attribuer une adresse IP fixe pour un client donné (--client-config-dir <répertoire>)</i> -----	11
4.3	CONFIGURATION DU SERVEUR-----	11
4.4	CONFIGURATION DU CLIENT-----	13
4.5	CONFIGURATION DU PARE-FEU-----	14
5	EXECUTION D'UN CLIENT OPENVPN SOUS WINDOWS SANS LES DROITS ADMINISTRATEURS. -----	14
6	TRACE DES CONNEXIONS AU NIVEAU DU SERVEUR -----	17

1 Généralités

OpenVPN est disponible sur plusieurs plateformes : Linux, Windows 2000/XP et plus récent, OpenBSD, FreeBSD, NetBSD, Mac OS X, et Solaris.

L'installation sur une machine Windows est détaillée dans le chapitre 5.

Un utilisateur sur un réseau distant veut accéder à un serveur dans la zone interne du réseau de son laboratoire. Dans cet exemple, l'architecture réseau du laboratoire est composée de deux zones :

- la « zone semi-ouverte », accessible de l'Internet
- la « zone interne » hébergeant les serveurs internes du laboratoire

Les deux premières zones sont séparées par un routeur assurant la sécurité entre elles.

Une troisième zone, nommée « zone VPN » est utilisée pour les machines distantes. Elle regroupe une plage d'adresses qui sera utilisée par le serveur OpenVPN pour celles-ci.

Dans la suite de cet article, les zones se partagent les adresses IP de la manière suivante :

- Zone semi-ouverte : 195.195.195.32/27
- Zone interne : 195.195.195.64/27
- Zone VPN : 195.195.195.224/27

Voici un schéma représentant ce que nous cherchons à mettre en place :

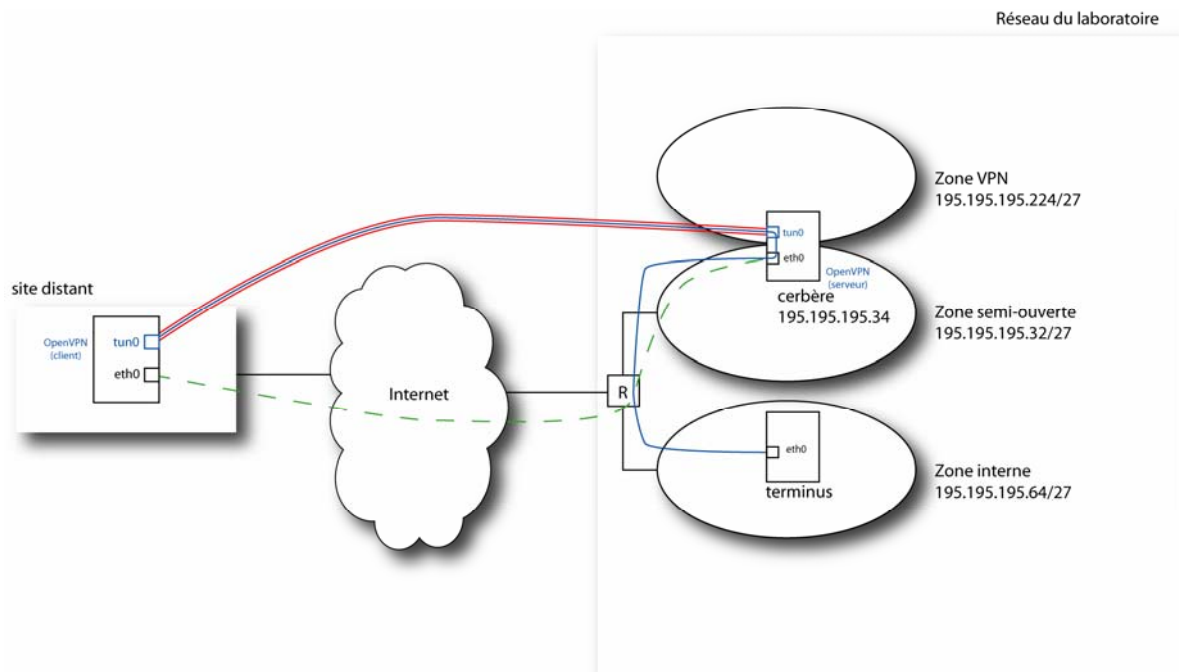


Figure 1 : Vue d'ensemble

Le processus est le suivant :

1. L'utilisateur démarre le client OpenVPN et se connecte au serveur OpenVPN sur la machine *Cerbère* (se trouvant dans la zone semi-ouverte) via Internet (traits pointillé vert).
2. A l'ouverture du tunnel, le serveur OpenVPN :
 - a. affecte des adresses IP aux extrémités du tunnel entre le serveur et la machine cliente dans la zone VPN du réseau du laboratoire.
 - b. envoie des commandes qui seront exécutées sur la machine cliente en particulier pour créer une interface réseau et modifier sa table de routage.
3. Une fois le tunnel établi, toutes les connexions entre le poste client et la zone interne du réseau du laboratoire passeront par le tunnel. Les datagrammes provenant de la machine cliente à destination du réseau du laboratoire auront une adresse source contenue dans la plage d'adresses de la « zone VPN ». La mise en place de la politique de filtrage en sera facilitée.

2 Options d'installation

2.1 *Choix du protocole*

Syntaxe : **proto tcp|udp**

OpenVPN peut utiliser le protocole UDP ou TCP. Or, pour des raisons de sécurité, le protocole UDP est de plus en plus rejeté par les parefeux en entrée des réseaux de campus ou de laboratoires, ce qui rendra l'utilisation d'OpenVPN impossible sur ces sites. Le protocole TCP est donc conseillé dans ce contexte, à partir du moment où les paquets TCP correspondants à des sessions établies sont autorisés en entrée de site.

Il est également possible de proposer les deux protocoles au niveau du serveur OpenVPN pour augmenter les possibilités de connexion aux nomades n'ayant aucun droit sur la politique de sécurité du réseau qu'ils utilisent. Pour cela, il faut lancer deux instances d'OpenVPN, donc créer deux fichiers de configuration. Chaque serveur instancié créera une interface réseau sur le serveur. Il faut donc prévoir une plage d'adresses pour chaque protocole. Les deux protocoles peuvent passer par le même numéro de port dont le numéro officiel attribué par l'IANA est 1194.

2.2 *Choix de la méthode d'authentification*

OpenVPN propose différentes méthodes d'authentification :

- Par login/mot de passe (PAM, RADIUS, ou LDAP)
- Par certificat X509

Nous ne traiterons ici que la deuxième méthode.

Le certificat pour le serveur OpenVPN doit avoir l'extension :

Netscape Cert Type: SSLServer

Le certificat client peut être un certificat personnel ou un certificat de serveur. Ce certificat et la clé privée correspondante peuvent être installés :

- Dans deux fichiers comme indiqué dans le tableau 2 (directives **cert** et **key**)
- Dans un fichier unique au format PKCS12 (directive **pkcs12**)
- Dans un magasin de certificats Windows (directive **cryptoapicert**)

La question se pose de savoir si le certificat client permet d'authentifier l'utilisateur ou le poste de travail. La réponse dépend de l'installation et du type de certificat. On authentifie l'utilisateur si le certificat est un certificat personnel conservé :

- dans un fichier PKCS12 dans un répertoire de l'utilisateur
- dans le magasin de certificats Windows de l'utilisateur
- sur un token USB ou une carte

Dans les autres cas, on authentifie le poste de travail.

2.3 Choix du type de tunnel

Syntaxe : **dev tun|tap**

Il existe deux modes pour établir des tunnels via des VPN : le mode routeur et le mode pont. Il est recommandé d'utiliser OpenVPN en mode routeur sauf dans le cas où le tunnel doit pouvoir faire transiter des paquets dont le protocole n'est pas IP.

Dans ce document, nous ne verrons que le mode routeur.

3 Configuration sécurisée d'OpenVPN

La sécurisation d'un serveur OpenVPN peut se faire sur plusieurs plans :

1. La sécurisation du serveur lui-même, l'accès à celui-ci pour éviter des connexions non autorisées.
2. La mise en place d'une politique de filtrage (aux niveaux du serveur et du routeur) pour n'autoriser les accès depuis les postes distants qu'à un ensemble de services donnés.

3.1 Sécurisation du serveur

Pour sécuriser le serveur OpenVPN, plusieurs méthodes sont disponibles par rapport à une configuration basique :

- utilisation d'une clé partagée entre le serveur et les clients.
- emprisonner le serveur OpenVPN en redéfinissant la racine du serveur avec la directive *chroot* et abaisser les privilèges en utilisant les directives *user* et *group*.

3.1.1 Utilisation d'une clé partagée

Syntaxe : **tls-auth <fichier> 0|1**

Cette méthode complète le dispositif de sécurité en utilisant une clé partagée entre le serveur et les postes clients. Elle entre en œuvre en amont de l'authentification du client. Toute demande de connexion par un client ne possédant pas la clé partagée est tout simplement ignorée. Cette option permet donc d'augmenter le niveau de sécurité, et protège :

- des attaques DoS (Déni de Service) et du « UDP port flooding »
- du scanning de ports pour déterminer ceux qui sont ouverts
- des vulnérabilités du débordement de mémoire (« buffer overflow » en anglais) dans l'implémentation de SSL/TLS
- des initialisations en SSL/TLS des machines non autorisées puisque celles-ci sont rejetées avant le processus d'établissement d'un tunnel

L'utilisation de la directive **tls-auth** demande d'avoir généré une clé. OpenVPN met à disposition une commande :

```
openvpn --genkey --secret ta.key
```

Une fois cette clé générée, elle peut être placée dans le même répertoire que celui des fichiers des certificats, et doit être distribuée à tous les clients.

Rajouter dans le fichier de configuration du serveur :

```
tls-auth ta.key 0 # --- 0 pour le serveur
```

Rajouter dans le fichier de configuration des clients :

```
tls-auth ta.key 1 # --- 1 pour les clients
```

Remarque : il est vivement recommandé d'utiliser cette directive lorsque le serveur accepte des connexions provenant de la part d'adresses IP inconnues. Si à l'inverse le serveur et les clients sont connus ou ont des adresses IP fixes, la directive **remote <nom de machine | adresse IP> [port]** peut être utilisé dans les fichiers de configuration des machines aux extrémités du tunnel.

3.1.2 Redéfinir la racine du serveur

Syntaxe : **chroot <répertoire>**

Cette directive redéfinit la racine du serveur. Une fois le serveur démarré, seuls les fichiers et les répertoires situés dans le répertoire indiqué seront accessibles.

Pour l'utiliser, il faut copier la commande `sh` ainsi que tous les exécutables et toutes les bibliothèques nécessaires au serveur OpenVPN après son initialisation dans le répertoire spécifié par la directive **chroot**.

3.2 Politique de filtrage

La sécurisation des postes distants connectés via un VPN relève à peu près de la même problématique que celle concernant la connexion des portables dans le réseau du laboratoire. En effet tout se passe comme si le poste distant était connecté physiquement au réseau du laboratoire. La différence essentielle repose sur le fait que le poste distant peut être connecté à un réseau « étranger » (à la maison, dans un autre laboratoire ou dans un hôtel), donc potentiellement hostile.

Il est donc nécessaire de mettre en place une politique de filtrage qui permettra de limiter précisément les connexions possibles entre les postes clients et les machines du réseau du laboratoire.

Le filtrage peut être mis en place à 2 niveaux (cf. figure 3) :

- sur le serveur OpenVPN, par l'utilisation d'un garde-barrière (par exemple la commande *iptables*). Cela permettra :
 - de limiter les paquets en provenance du tunnel et à destination du serveur lui-même (1 sur fig. 3)
 - de limiter le routage des paquets en provenance du tunnel à destination des machines dans la même zone que le serveur OpenVPN (2 sur fig. 3)
- sur le routeur, point d'interconnexion entre les différentes zones. Cela permettra de limiter les possibilités de connexion entre les postes clients et les autres zones du réseau du laboratoire (3 sur fig. 3)

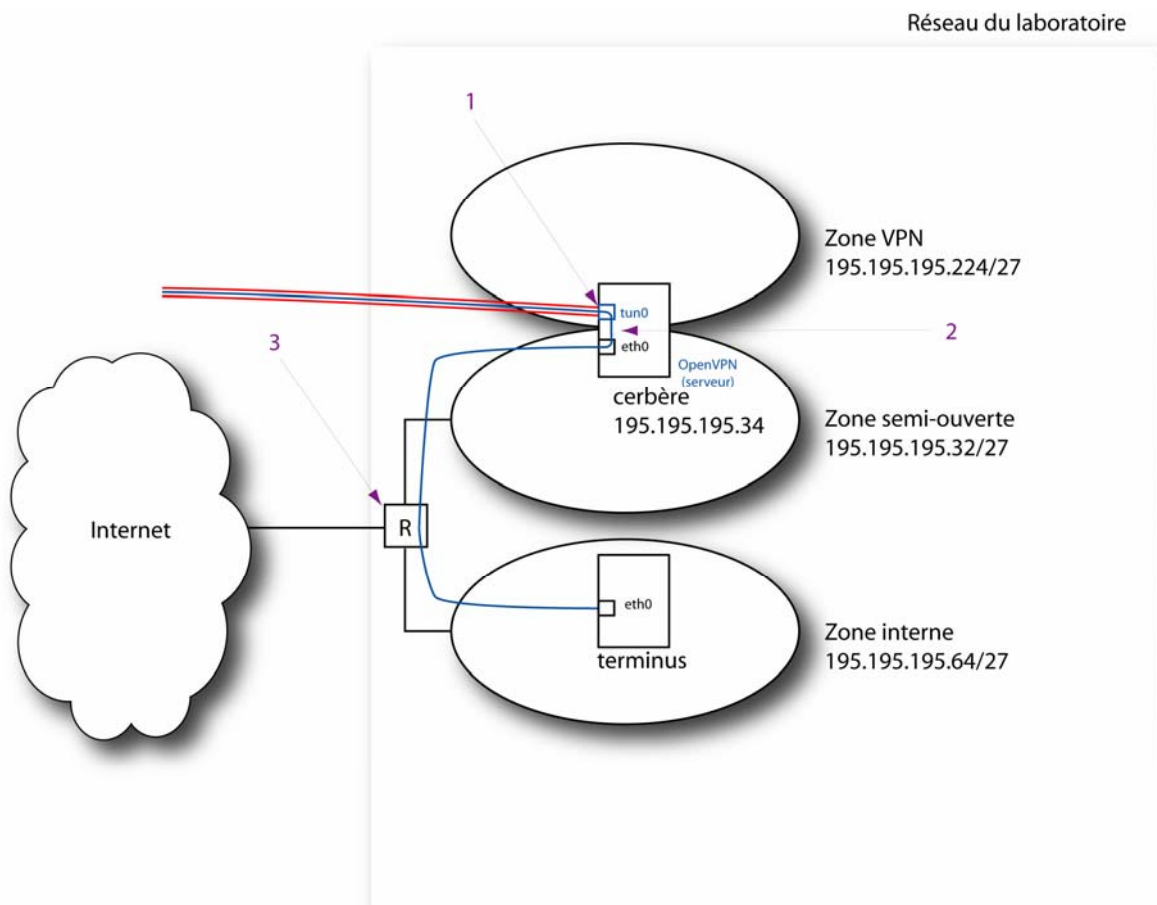


Figure 2 : Points de filtrage

La mise en place de ce filtrage permettra en particulier d'empêcher des machines éventuelles sur les réseaux des postes clients d'utiliser ce dernier comme routeur pour accéder au réseau du laboratoire.

3.2.1 Filtrage sur le serveur OpenVPN

La configuration du pare-feu sur le serveur OpenVPN doit :

- autoriser la connexion TCP ou UDP sur le port OpenVPN (1194, numéro de port officiel)
- autoriser les services accessibles (s'il y en a) sur le serveur OpenVPN lui-même à partir de la zone VPN
- autoriser le routage des paquets en provenance de la zone VPN à destinations des services autorisés
- interdire tout le reste

3.2.2 Filtrage sur le routeur

Il est conseillé de l'utiliser pour filtrer les paquets entre les zones. Ainsi, il sert à affiner le filtrage et/ou le routage entre des zones prédéfinies. Par rapport à l'exemple précédent, il devra :

- autoriser la connexion TCP ou UDP sur le port OpenVPN (1194, numéro de port officiel) sur le serveur OpenVPN
- Autoriser les connexions en provenance de la zone VPN et à destination de la zone interne uniquement pour les services accessibles
- Interdire tout le reste en provenance de la zone VPN

4 Exemple de configuration sécurisée

Dans cet exemple de configuration, nous souhaitons :

- faire tourner le serveur OpenVPN sous l'utilisateur 'openvpn'
directive **user** <user>
et
directive **group** <group>
- emprisonner le serveur OpenVPN
directive **chroot** <chemin>
- utiliser une clé partagée entre le serveur et les clients
- authentifier les utilisateurs par certificats et vérifier leurs droits d'accès dans un annuaire LDAP
directive **--client-connect** <script>
- attribuer une adresse IP à la machine distante en fonction du profil de l'utilisateur
directives
--ifconfig-pool <start-IP> <end-IP> [netmask],
--client-config-dir <répertoire>
et
--client-connect <script>)

Commencer par suivre les étapes suivantes :

1. Créer un utilisateur et groupe 'openvpn'.
2. Changer le propriétaire du répertoire **/etc/openvpn** de ses sous-répertoires et fichiers à l'utilisateur 'openvpn'
3. Créer le script dans le fichier spécifié par la directive **client-connect**.
4. Mettre la commande **sh**, tous les exécutables, ainsi que les bibliothèques nécessaires au script et les placer dans le répertoire spécifié par la directive **chroot**. Pour pouvoir exécuter un script en chrooté, il faut mettre la commande **sh** car OpenVPN l'appelle pour lancer le script.

Puis, nous cherchons également à différencier les droits d'accès suivant l'authentification de l'utilisateur par son certificat. Par exemple, en donnant plus de droits aux administrateurs qu'aux autres utilisateurs.

Pour cela :

1. On définit deux plages d'adresses IP :
 - a. pour les utilisateurs : 195.195.195.224/28
 - b. pour les administrateurs : 195.195.195.240/28
2. On configure au niveau du routeur et du pare-feu du serveur OpenVPN pour permettre aux plages d'adresses d'accéder aux services appropriées.
3. On va attribuer une adresse IP statique à l'administrateur authentifié par la valeur de son CN dans la plage d'adresses des administrateurs.

4.1 Vérification des droits d'accès (--client-connect <script>)

Le nom du fichier script est spécifié par la directive **client-connect**. Il est appelé lors de la mise en place du tunnel et prend en paramètre le nom d'un fichier temporaire. OpenVPN attend comme code retour un entier :

- 0 : il passe à l'étape suivante (voir le chapitre suivant).
- > 0 : il ne passe pas à l'étape suivante et refuse le client.

Par exemple, dans un premier temps, on veut vérifier les droits d'accès d'un client : voici un programme écrit en shell qui vérifie l'existence d'au moins une entrée dans un annuaire LDAP dont la valeur de l'attribut CN correspond à celle du champ CN extraite du certificat :

```

#!/bin/bash
LDAPSEARCH="/usr/bin/ldapsearch"
LDAP_OPT="-x"
LDAP_SERVER="ldap.monlabo.fr"
LDAP_PORT=389
LDAP_BASE="dc=monlabo,dc=fr"
DISPLAY_ATTRIBUTE="cn"

CN=`expr "$tls_id_0" : '.*CN=\\([^\./^,]*\\)'`

RESULT=`$LDAPSEARCH $LDAP_OPT -s sub -h $LDAP_SERVER -p $LDAP_PORT -b
"$LDAP_BASE" cn="$CN" $DISPLAY_ATTRIBUTE`

NBRESULT=`echo "$RESULT" | grep "# numEntries: " | awk '{print $3}'`

if [ -n "$RESULT" ]
then
    if [ -n "$NBRESULT" ]
    then
        if [ "$NBRESULT" -gt 0 ]
        then
            exit 0;
        fi
    fi
fi
exit 1;

```

Remarque : la variable `tls_id_0` est une variable d'environnement dont la valeur est le DN du certificat client. Pour plus d'information concernant ces variables, veuillez consulter le paragraphe « *Environmental Variables* » de la manpage d'OpenVPN.

4.2 Attribution des adresses IP

La méthode d'affectation des adresses IP par OpenVPN est basée sur trois directives :

- **--ifconfig-pool <start-IP> <end-IP> [netmask]**
 Cette directive permet de définir le pool d'adresses IP qui seront affectées **dynamiquement** aux clients OpenVPN
- **--client-config-dir <répertoire>**
 Cette directive optionnelle permet de définir un répertoire qui contiendra des fichiers pour attribuer une **adresse IP statique** (correspondant à une valeur du champ CN dans le certificat client).
- **--client-connect <script>**
 cette directive optionnelle permet de définir un script qui sera exécuté par le serveur pour vérifier les droits d'accès et éventuellement générer un fichier pour attribuer une **adresse IP statique**

L'algorithme utilisé par OpenVPN est le suivant :

1. si la directive **client-connect** est utilisée ET si le script crée un fichier temporaire contenant la commande **ifconfig-push**, les adresses IP indiquées seront affectées aux extrémités du tunnel
2. sinon, si la directive **client-config-dir** est utilisée ET qu'un fichier ayant comme nom la valeur du CN du certificat du client existe dans le répertoire spécifié par cette directive, la commande **ifconfig-push** contenue dans ce fichier attribuera les adresses IP.
3. sinon, le serveur OpenVPN attribuera les adresses parmi celles libres dans le pool d'adresses défini par la directive **ifconfig-pool**

Remarques :

- La directive **server** spécifie plusieurs autres directives. Ainsi, spécifier :

```
server 195.195.195.224 255.255.255.240
```

revient à spécifier en mode routeur :

```
mode server
tls-server

ifconfig 195.195.195.225 195.195.195.226
ifconfig-pool 195.195.195.228 195.195.195.254
route 195.195.195.224 255.255.255.240

push "route 195.195.195.224"
```

- Attention, pour des problèmes de compatibilité avec le client Windows et le driver TAP-Win32, la directive **ifconfig-pool** peut uniquement utiliser les couples d'adresses dont le dernier octet correspond aux couples suivants :

[1, 2]	[5, 6]	[9, 10]	[13, 14]	[17, 18]
[21, 22]	[25, 26]	[29, 30]	[33, 34]	[37, 38]
[41, 42]	[45, 46]	[49, 50]	[53, 54]	[57, 58]
[61, 62]	[65, 66]	[69, 70]	[73, 74]	[77, 78]
[81, 82]	[85, 86]	[89, 90]	[93, 94]	[97, 98]
[101, 102]	[105, 106]	[109, 110]	[113, 114]	[117, 118]
[121, 122]	[125, 126]	[129, 130]	[133, 134]	[137, 138]
[141, 142]	[145, 146]	[149, 150]	[153, 154]	[157, 158]
[161, 162]	[165, 166]	[169, 170]	[173, 174]	[177, 178]
[181, 182]	[185, 186]	[189, 190]	[193, 194]	[197, 198]
[201, 202]	[205, 206]	[209, 210]	[213, 214]	[217, 218]
[221, 222]	[225, 226]	[229, 230]	[233, 234]	[237, 238]
[241, 242]	[245, 246]	[249, 250]	[253, 254]	

4.2.1 Le script (--client-connect <script>)

Cette directive peut également être utilisé pour spécifier une commande à exécuter par OpenVPN. On peut donc l'utiliser pour attribuer une adresse IP fixe. Le principe est sensiblement le même que vu précédemment :

- 0 : le serveur regarde si une adresse IP a été attribuée. Si oui, il autorise le client à établir un tunnel. Sinon, il passe à l'étape suivante.
- < 0 : il ne passe pas à l'étape suivante et refuse le client.

Pour attribuer une adresse IP fixe pour une personne cherchant à se connecter, il suffit de rajouter dans le script (chap. 5.1 Vérification des droits d'accès) les lignes :

```
if [ $tls_id_0 = "la valeur du DN" ]
then
echo 'ifconfig-push 195.195.195.229 195.195.195.230' > $1
fi
```

Dans ce cas, si l'exécution du script renvoie la valeur 0, OpenVPN exécute le fichier temporaire (\$1) qui contient la commande

```
'ifconfig-push 195.195.195.229 195.195.195.230'
```

et donc attribue l'adresse 195.195.195.129 du côté serveur du tunnel, et l'adresse 195.195.195.130 du côté client du tunnel.

4.2.2 Attribuer une adresse IP fixe pour un client donné (--client-config-dir <répertoire>)

La directive **client-config-dir** permet de spécifier un répertoire qui contient des fichiers dont le nom est la valeur du CN du sujet du certificat présenté auquel on veut attribuer une adresse fixe. OpenVPN scanne ce répertoire cherchant une configuration spécifique à ce client. Attention : tout caractère non accepté dans le CN est remplacé par un caractère « souligné » ('_'). Pour le CN, les caractères acceptés sont : alphanumérique, souligné ('_'), tiret ('-'), alpha ('@'). Pour plus d'information, consultez le paragraphe « String Types and Remapping » de la manpage d'OpenVPN.

Par exemple, l'administrateur veut attribuer une adresse IP fixe pour Jean Dupond. La valeur de son CN est 'Jean Dupond' (avec un espace entre Jean et Dupond).

1. Il spécifie, dans le fichier de configuration du serveur, le répertoire **ccd** par la directive **client-config-dir <racine openvpn>/ccd**.
2. Il crée alors le fichier **<racine openvpn>/ccd/Jean_Dupond** dans lequel il met la ligne suivante :

```
ifconfig-push 195.195.195.241 195.195.195.242
```

Lorsque le tunnel sera établi, le poste client aura pour adresse 195.195.195.242 et l'autre extrémité du tunnel aura pour adresse 195.195.195.241.

4.3 Configuration du serveur

Copier et modifier le fichier d'exemple fourni dans les sources (**<racine openvpn>/openvpn-2.0_rc19/server.conf**) et le mettre par exemple dans le répertoire **/etc/openvpn** (notre répertoire de fichiers relatifs à OpenVPN).

Voici le fichier de configuration correspondant à cet article du côté serveur :

```
#Fichier de configuration de OpenVPN partie SERVEUR

#-----Securite-----
# --- les serveur tourne sous l'utilisateur et sous le groupe openvpn
user openvpn
group openvpn

chroot /etc/openvpn # --- modifie la racine pour le serveur OpenVPN

# ----- Vérification sur les clients -----
# --- Script appelé lors de l'établissement de la connexion
client-connect "/script/connect"
client-config-dir /ccd

#----- Réseau -----
dev tun # --- Mode routeur

# --- OpenVPN va assigner la 1ere adresse (.1) a l'extrémité locale
# --- du tunnel (serveur) et
# --- va fournir aux clients une adresse dans ce réseau en incrémentant
server 195.195.195.224 255.255.255.240

# --- Ajoute la route suivante (zone VPN des administrateurs) à la table
# --- de routage locale une fois le tunnel établi
# --- route reseau masque [passerelle] [metric]
route 195.195.195.240 255.255.255.240

# --- Ce tunnel utilisera tcp
proto tcp
port 1194

# --- Options de persistance pour permettre l'accès a certaines ressources
# --- lors d'un re-démarrage
persist-key
persist-tun
keepalive 10 60

# --- Commande envoyé au client pour modifier sa table de routage
# --- Dans ce cas, on redirige tous les paquets à destination de
# --- 195.195.195.64/27 vers le tunnel.
push "route 195.195.195.64 255.255.255.224"

# ----- Certificats -----
# --- clé partagée utilisée lors de l'initialisation du tunnel
tls-auth ssl-tls/ta.key 0
# --- Paramètres Diffie-Hellman
dh ssl-tls/dh1024.pem
ca ssl-tls/ca.crt
cert ssl-tls/serveur.pem
key ssl-tls/serveur.key # Fichier à protéger

# ----- Logs -----
verb 5 # --- niveau de log
log-append log/openvpn.log # --- Fichier de log

# --- Fichier de status des connexions actuelles du serveur
# --- (mis à jour toutes les minutes)
status log/openvpn-status.log
comp-lzo # --- Utiliser la librairie LZ0
```

4.4 Configuration du client

Voici l'exemple de fichier de configuration d'un poste client OpenVPN correspondant à cet article :

```
# Fichier de configuration de OpenVPN partie CLIENT
# -- Spécifie la partie client
client

# --- Utiliser les mêmes options que sur la partie serveur
dev tun
proto tcp

# --- Forcer la récupération de
# --- la configuration donnée par le serveur
pull

# --- le hostname/IP et port du serveur.
# --- il est possible d'avoir plusieurs entrées
# --- remote cerbere 1194
remote cerbere.monlabo.fr 1194

# --- Essaie de résoudre le nom
resolv-retry infinite

# --- La plupart des clients n'ont pas besoin de faire un bind
nobind

# --- Préserve certains états entre deux exécutions
persist-key
persist-tun

# --- paramètres SSL/TLS
pkcs12 "C:\\Program Files\\OpenVPN\\config\\certificat.p12"
# --- Ou alors, remplacer le fichier p12 par les trois fichiers
#ca ca.crt
#cert client.crt
#key client.key

# --- Vérifie que le serveur possède un certificat
# --- dont l'attribut nsCertType a pour valeur « server »
ns-cert-type server

# --- active la compression LZ0 pour optimiser les échanges.
comp-lzo

# --- Niveau de log
verb 5

# Clé partagée
tls-auth "C:\\Program Files\\OpenVPN\\config\\ta.key" 1
tls-client
```

Remarque : pour des problèmes de modification de table de routage, un client Windows doit être exécuté avec les droits administrateurs. Cependant, un utilisateur n'ayant pas les droits administrateurs peut utiliser le client d'OpenVPN (cf. chapitre 6 Exécution d'OpenVPN sous Windows sans les droits administrateurs).

4.5 Configuration du pare-feu

Voici un exemple de fichier de configuration pour iptables (`/etc/sysconfig/iptables`) :

```
*filter
: INPUT DROP [0:0]
: FORWARD DROP [0:0]
: OUTPUT DROP [0:0]
: RULE_1 - [0:0]
: RULE_2 - [0:0]
: LOGANDDROP - [0:0]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i tun0 -s 195.195.195.240/28 -d 195.195.195.225 -p tcp -m tcp
--dport 22 -j ACCEPT
-A INPUT -i tun0 -s 195.195.195.240/28 -d 195.195.195.225 -p icmp -m icmp -
-icmp-type 8/0 -j ACCEPT
-A INPUT -j LOGANDDROP
-A FORWARD -i tun0 -s 195.195.195.128/24 -d 195.195.195.64/27 -j RULE_2
-A FORWARD -o tun0 -j ACCEPT
-A FORWARD -j LOGANDDROP
-A OUTPUT -o tun0 -j ACCEPT
-A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A OUTPUT -j LOGANDDROP
-A RULE_1 -p tcp -m tcp --dport 1194 -j ACCEPT
-A RULE_2 -p icmp -m icmp --icmp-type 8/0 -j ACCEPT
-A RULE_2 -p tcp -m tcp -m multiport --dports
22, 25, 80, 143, 443, 389, 137, 138, 139, 993, 445 -j ACCEPT
-A RULE_2 -p udp -m udp -m multiport --dports 137, 138 -j ACCEPT
-A LOGANDDROP -j LOG --log-prefix "RULE LOGANDDROP -- DENY " --log-level 6
-A LOGANDDROP -j DROP
```

Dans cet exemple, nous autorisons :

- l'accès au serveur OpenVPN en ssh (port 22) seulement pour les machines dont l'adresse IP est dans le sous-réseau 195.195.195.240/28
- le routage des paquets en provenance de la zone VPN vers la zone interne 195.195.195.64/27 seulement pour les protocoles autorisés : SSH, SMTP, HTTP, IMAP, LDAP et SAMBA.

5 Exécution d'un client OpenVPN sous Windows sans les droits administrateurs.

Si l'utilisateur Windows ne veut/peut pas utiliser un compte ayant les droits administrateurs, il est quand même envisageable d'utiliser le client OpenVPN.

Deux méthodes sont possibles :

- L'utilisateur connaît le mot de passe d'un compte administrateur : il peut faire « exécuter sous » (ou « run as ») et donner le mot de passe au moment de lancer le programme. Ou encore, il peut créer un raccourci et le configurer pour être exécuté sous un autre utilisateur. Sous Windows XP :
 1. Créer un raccourci qui pointe sur le programme `openvpn-gui` (`c:\program files\openvpn\bin\openvpn-gui.exe`)
 2. Afficher les propriétés en cliquant droit dessus.

3. Cliquer sur '**Avancé...**' et sélectionner '**Exécuter en utilisant d'autres informations d'identification**'.

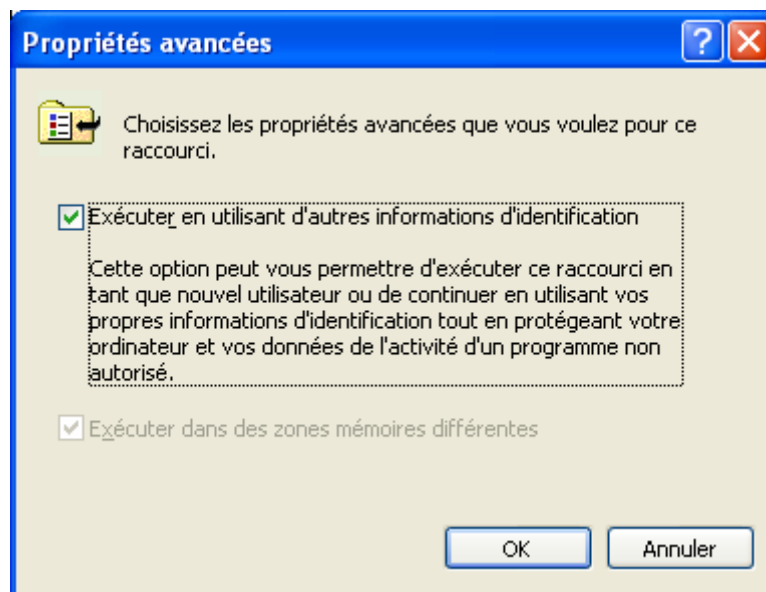


Figure 3 : Changer d'utilisateur pour l'exécution

4. Par la suite, à chaque lancement d'exécution du programme OpenVPN en double-cliquant sur ce raccourci, un identifiant et mot de passe seront demandés.
- L'utilisateur ne connaît pas de mot de passe d'un compte administrateur. L'administrateur de la machine doit spécifier au moment de l'installation que le client OpenVPN est un service windows. Ce service ne doit pas être démarré automatiquement, mais de manière explicite par l'utilisateur final. Il doit aussi stocker le certificat dans le magasin de certificats de Windows pour pouvoir signer la clé privée. Dans ce cas, il est impossible d'utiliser le certificat au format p12.

- Vérifier que le programme est installé comme service Windows en allant dans Démarrer → Panneau de configuration → Outils d'administration → Services.

 OpenVPN Service
 Manuel
Système local

- Télécharger et installer le programme SubInACL.exe (lien : <http://www.microsoft.com/downloads/details.aspx?FamilyID=e8ba3e56-d8fe-4a91-93cf-ed6985e3927b&displaylang=en>)
- Donner les privilèges (à 'user') pour démarrer et arrêter le service OpenVPN. Pour cela, se loguer en tant qu'administrateur et exécuter la commande dans le répertoire où se trouve l'exécutable subinacl.exe

```
subi nacl /SERVI CE "OpenVPNServi ce" /GRANT=user=TO
```

- Modifier dans la table des registres (avec la commande '**regedit**') la valeur du paramètre HKEY_LOCAL_MACHINE\SOFTWARE\OpenVPN-GUI \servi ce_onl y à 1 et le paramètre HKEY_LOCAL_MACHINE\SOFTWARE\OpenVPN-GUI \al l ow_servi ce à 1.

- Ajouter le certificat dans le magasin des certificats de Windows en utilisant la commande en ligne 'mmc' :
 - En ligne de commande, lancer mmc
 - **Fichier → Ajouter/supprimer un composant logiciel enfichable...**
 - **Ajouter...**
 - Sélectionner les certificats et cliquer sur le bouton '**Ajouter**'
 - Choisir '**Le compte de l'ordinateur**' et cliquer sur 'Suivant'
 - Choisir '**L'ordinateur local**' et cliquer sur '**Terminer**'
 - Dans la fenêtre 'Console1', déployer l'arborescence des certificats.
 - Cliquer droit sur '**Autorités de certification racine de confiance**', '**Toutes les tâches...**' → '**Importer**' et suivre les étapes pour importer le certificat de l'AC racine
 - Puis importer le certificat dans 'Personnel'

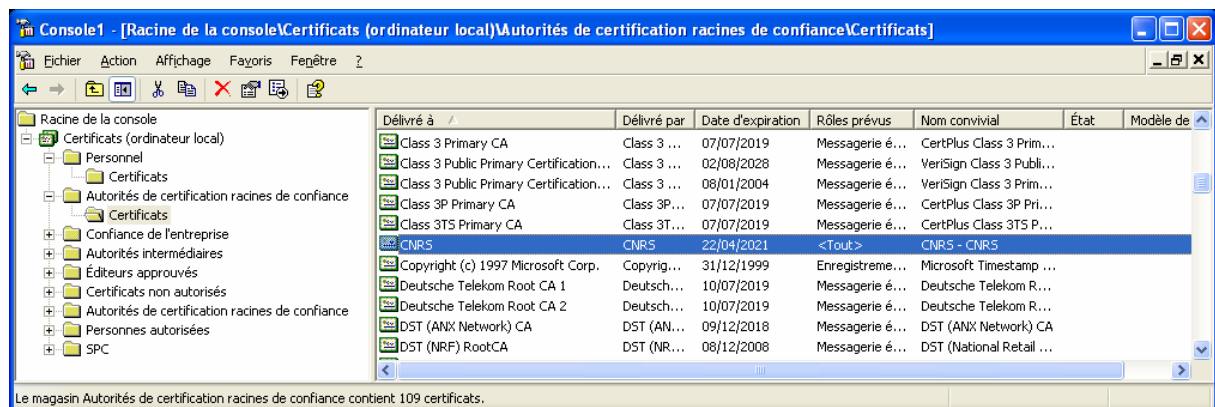


Figure 4 : Fenêtre « Console1 »

- S'assurer que l'utilisateur a les droits d'écritures sur le fichier de log. (<sources openvpn>/log/client.log)
- Modifier le fichier de configuration du client en supprimant la directive **pkcs12**, et en ajoutant les options **ca** et **cryptoapi** :

```
#pkcs12 "C:\Program Files\OpenVPN\config\certificat.p12"
# --- copie de l'empreinte numérique
cryptoapi cert "THUMB:a9 54 e9 35 e2 53 33 c9 d0 54 4c f8 33 34 57 8c a8 a5
14 b9"
# --- AC racine (la/les AC intermédiaires ne sont pas demandées)
ca "C:\Program Files\OpenVPN\config\CNRS.crt"
```

Remarque: l'empreinte numérique à copier est la propriété nommée '*Empreinte numérique*' et non '*Identificateur de la clé du sujet*'.

6 Trace des connexions au niveau du serveur

Ainsi installé, OpenVPN peut tracer toute connexion établie avec le serveur OpenVPN visible dans le fichier de log du serveur. Il est donc possible de retrouver quelle connexion a démarré à quel date et le CN du certificat correspondant :

```

...
Wed May 11 10:10:17 2005 Initialization Sequence Completed
...
Wed May 11 10:12:20 2005 200.200.200.200:1290 VERIFY OK: depth=1,
 /C=FR/O=ORG/CN=Autorité-de-Certification
Wed May 11 10:12:20 2005 200.200.200.200:1290 VERIFY OK: depth=0,
 /C=FR/O=ORG/OU=monlabo/CN=Jean_Dupond/emailAddress=jean.dupond@monlabo.fr
Wed May 11 10:12:20 2005 200.200.200.200:1290 Data Channel Encrypt: Cipher
 'BF-CBC' initialized with 128 bit key
Wed May 11 10:12:20 2005 200.200.200.200:1290 Data Channel Encrypt: Using
 160 bit message hash 'SHA1' for HMAC authentication
Wed May 11 10:12:20 2005 200.200.200.200:1290 Data Channel Decrypt: Cipher
 'BF-CBC' initialized with 128 bit key
Wed May 11 10:12:20 2005 200.200.200.200:1290 Data Channel Decrypt: Using
 160 bit message hash 'SHA1' for HMAC authentication
Wed May 11 10:12:20 2005 200.200.200.200:1290 Control Channel: TLSv1,
 cipher TLSv1/SSLv3 DHE-RSA-AES256-SHA, 2048 bit RSA
Wed May 11 10:12:20 2005 200.200.200.200:1290 [Jean_Dupond] Peer Connection
 Initiated with 200.200.200.200:1290
Wed May 11 10:12:20 2005 Jean_Dupond/200.200.200.200:1290 OPTIONS IMPORT:
 reading client specific options from: /ccd/Jean_Dupond
Wed May 11 10:12:20 2005 Jean_Dupond/200.200.200.200:1290 MULTI: Learn:
 195.195.195.241 -> Jean_Dupond /200.200.200.200:1290
Wed May 11 10:12:20 2005 Jean_Dupond/200.200.200.200:1290 MULTI: primary
 virtual IP for Jean_Dupond /200.200.200.200:1290: 195.195.195.241
Wed May 11 10:12:21 2005 Jean_Dupond/200.200.200.200:1290 PUSH: Received
 control message: 'PUSH_REQUEST'
Wed May 11 10:12:21 2005 Jean_Dupond/200.200.200.200:1290 SENT CONTROL
 [Jean_Dupond]: 'PUSH_REPLY,route 195.195.195.64 255.255.255.224,route
 195.195.195.225,ping 10,ping-restart 60,ifconfig 195.195.195.241
 195.195.195.242' (status=1)

```

Dans cet exemple, nous pouvons voir qu'un client OpenVPN, dont l'adresse IP est 200.200.200.200 a demandé l'établissement d'un tunnel, en présentant :

- un certificat dont le DN est
/C=FR/O=ORG/OU=monlabo/CN=Jean_Dupond/emailAddress=jean.dupond@monlabo.fr .

Nous pouvons suivre toutes les étapes de vérification des droits d'accès au serveur OpenVPN :

- Vérification du certificat
- Echanges de données encryptée et décryptée avec la clé partagée
- Lecture d'une configuration spécifique au CN dont la valeur est **Jean_Dupond**
- Attribution de l'adresse IP **195.195.195.241** dans la zone VPN
- Envoi des commandes de modification de la table de routage

Suivant le même principe, on peut retrouver à quel instant le serveur OpenVPN a détecté une inactivité du client :

```
Wed May 11 10:17:13 2005 Jean_Dupond/200.200.200.200:1290 [Jean_Dupond]
Inactivity timeout (--ping-restart), restarting
Wed May 11 10:17:13 2005 Jean_Dupond/200.200.200.200:1290
SIGUSR1[soft,ping-restart] received, client-instance restarting
```

Remarque : pour des questions de confidentialité sur les données conservées dans ce fichier, il est vivement recommandé de le rendre accessible en lecture/écriture qu'à l'administrateur du serveur.

Références

- OpenVPN - An Open Source SSL VPN Solution : <http://openvpn.net/>
- OpenVPN GUI for Windows : <http://openvpn.se/>
- OpenVPN-GUI for Mac OS X : <http://rechenknecht.net/OpenVPN-GUI/>
- VPN TLS avec OpenVPN de Matthieu Herrb : <ftp://ftp.laas.fr/pub/ii/matthieu/openvpn.pdf>
- [Project Info - OpenVPN](http://sourceforge.net/projects/openvpn/) : <http://sourceforge.net/projects/openvpn/>