OpenVPN :

Mais là solution que je vous propose est basée sur OpenVPN (http://openvpn.net/) et parce qu'on ne peut pas mieux se présenter que soit même :

Citation :

OpenVPN is a full-featured SSL VPN solution which can accomodate a wide range of configurations, including remote access, site-to-site VPNs, WiFi security, and enterprise-scale remote access solutions with load balancing, failover, and fine-grained access-controls.

OpenVPN implements OSI layer 2 or 3 secure network extension using the industry standard SSL/TLS protocol, supports flexible client authentication methods based on certificates, smart cards, and/or 2-factor authentication, and allows user or group-specific access control policies using firewall rules applied to the VPN virtual interface. OpenVPN is not a web application proxy and does not operate through a web browser.

OpenVPN est donc une solution de réseau privé virtuel qui utilise SSL (couche de cryptage). Cette solution est ouverte, multi-plateforme, libre et gratuite (ce qui ne vous empêche pas de faire une donation aux êtres humains qui travaillent sur ce projet). Elle est très configurable et parfaitement utilisable dans un environnement d'entreprise. Elle propose pour ces dernières des fonctions avancées comme la répartition de charge et la redondance.

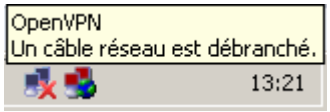
Mon but est donc d'expliquer le plus simplement possible comme mettre en place ce type réseau et accessoirement vous expliquer comment ça marche car c'est quand même intéressant. Vous vous dites que ça commence mal pour la rapidité de déploiement vu tout ce qui précède mais ne vous inquiétez pas, nous y venons%u2026

Pour rendre les choses un poil plus convivial, nous allons utiliser OpenVPN GUI (http://openvpn.se/).

Téléchargeons le package d'installation.

Pour le moment, en version stable, il s'agit de la version 1.0.3 basée sur OpenVPN 2.0.5.

Il n'y a pas de pré-requis particulier à avoir. Lancez l'installation en gardant les options par défaut. Lorsque l'on va vous demander d'installation un nouveau composant sur le système, faites « continuer ». Il s'agit d'une nouvelle interface réseau virtuelle. C'est une particularité de Windows. VMWare utilise le même mécanisme.



Une fois l'installation terminée, on ne commence pas par s'exciter sur la nouvelle icône dans la barre des tâches en pestant que rien ne marche. Il va falloir mettre les mains dans le cambouis.

Commençons par configurer un serveur OpenVPN.

Rendez-vous dans :

```
C:\Program Files\OpenVPN
/bin contient les exécutables du programme
/config contiendra la configuration, les clés et certificats
/driver contient le pilote pour l'interface réseau virtuelle
/easy-rsa est une boite à outils pour générer clés et
certificats
/log accueillera vos redoutés messages d'erreur (entre
autre)
/sample-config propose des configurations pré-remplies
```

Les présentations sont faites. La procédure que je vais vous décrire suit en grande partie le « how-to » proposé sur le site officiel.

Lancez une console (Démarrer / Exécuter / cmd.exe).
Vous commencez à paniquer ? Retournez au début et passez par la case Hamachi sans toucher aux joies du libre. Il est obligatoire d'utiliser une console pour garder les variables, ne doublez cliquez donc jamais sur les icônes des fichiers batch.

Déplacez-vous à coup(s) de commande « cd » dans C:\Program Files\OpenVPN\easy-rsa

Pour plus de clarté, les commandes à taper seront simplement précédées de « > » dans la suite de ce tutoriel.

Allons-y :
> init-config
Pour préparer le terrain et remettre les pendules à l'heure si vous vous êtes précédemment mélangé les pinceaux.

Il vous faut maintenant éditer le fichier vars.bat qui se trouve dans /easy-rsa.

Il est possible que vous ayez quelques problèmes pour modifier ces fichiers à cause des retours à ligne qui ne sont pas faits de la même façon sur Linux et Windows (tout va apparaître en ligne, les éléments étant séparés par des rectangles). Il vous faut donc un éditeur de texte un peu plus costaud que le bloc-notes. Personnellement j'utilise gVim mais c'est un peu extrémiste et vous risquez de le haïr rapidement si vous ne le connaissez pas. Je vous invite à trouver votre préférence en cherchant et testant quelques éditeurs sur Internet.

Voilà les champs qui nous intéressent :

set KEY_SIZE=1024

La documentation officielle propose aux paranoïaques d'augmenter le chiffrement de la clé à 2048. En contrepartie, les performances seront moins bonnes. J'ai laissé cette option à 1024.

Les champs qui suivent vous permettront simplement de gagner du temps par la suite dans la création de vos certificats, ne laissez aucun champs vide. Pour les non anglophones, je traduis rapidement :

COUNTRY : sigle du pays
PROVINCE : état pour les Etats-Unis, vous pouvez mettre les deux premières lettres de votre région
CITY : la ville
ORG : Le nom de l'organisation ou entreprise

set KEY_COUNTRY=FR
set KEY_PROVINCE=LO
set KEY_CITY=Nancy
set KEY_ORG=MasterJul.net
set KEY_EMAIL=example@masterjul.net

N'oubliez pas d'enregistrer%u2026

Continuons dans les commandes :
> vars
Pour initialiser les variables que l'on vient de définir.
> clean-all
Nettoie toutes les anciennes clés générées dans /keys et prépare une nouvelle base de certificats.

Génération du certificat root et de sa clé pour le réseau :

> build-ca
Pour générer votre certificat d'autorité. Réfléchissez à un « common name » qui sera le nom de votre réseau virtuel. Cela peut être le nom de l'entreprise par exemple précédée de « OpenVPN » par exemple.

Voyons en détail ce qu'il se passe :

```
C:\Program Files\OpenVPN\easy-rsa>build-ca.bat
Loading 'screen' into random state - done
Generating a 1024 bit RSA private key
.......++++++
.....................................................++++++
writing new private key to 'keys\ca.key'
-----
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a
Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
```

```
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [FR]:
State or Province Name (full name) [LO]:
Locality Name (eg, city) [Nancy]:
Organization Name (eg, company) [MasterJul.net]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname)
[]:OpenVPN-MJ.net
Email Address [example@masterjul.net]:

C:\Program Files\OpenVPN\easy-rsa>
```

Résultat de tout cela, la création de deux importants fichier dans /easy-rsa/keys :

ca.crt : Le certificat d'authentification root. Il peut être diffusé sans sécurité particulière.

Vous devrez d'ailleurs communiquer ce certificat à vos clients.

ca.key : La clé de ce certificat, sans doute le fichier le plus important. Il permet de signer les certificats des clients. À garder dans un coffre-fort, il n'est pas nécessaire au fonctionnement du serveur. Il ne doit donc évidemment pas circuler sur une liaison non sécurisée.

Génération du certificat et de la clé pour le serveur :

C'est simple, en une commande :
> build-key-server server
N'oubliez pas de préciser l'argument sinon vous risquez d'avoir quelques surprises dans les noms de fichiers générés.

À la différence du certificat root, le « common name » doit ici être « server ».

Je vous propose à nouveau un compte-rendu complet pour éviter les erreurs :

```
C:\Program Files\OpenVPN\easy-rsa>build-key-server server
Loading 'screen' into random state - done
Generating a 1024 bit RSA private key
.............++++++
..++++++
writing new private key to 'keys\server.key'
-----
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a
Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
```

```
Country Name (2 letter code) [FR]:
State or Province Name (full name) [LO]:
Locality Name (eg, city) [Nancy]:
Organization Name (eg, company) [MasterJul.net]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname)
[]:server
Email Address [example@masterjul.net]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from openssl.cnf
Loading 'screen' into random state - done
DEBUG[load_index]: unique_subject = "yes"
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'FR'
stateOrProvinceName   :PRINTABLE:'LO'
localityName          :PRINTABLE:'Nancy'
organizationName      :PRINTABLE:'MasterJul.net'
commonName            :PRINTABLE:'server'
emailAddress          :IA5STRING:'example@masterjul.net'
Certificate is to be certified until Mar 17 17:11:19 2016
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

C:\Program Files\OpenVPN\easy-rsa>
```

Il faut donc répondre « y » aux deux questions posées pour signer le certificat et l'ajouter à la base.

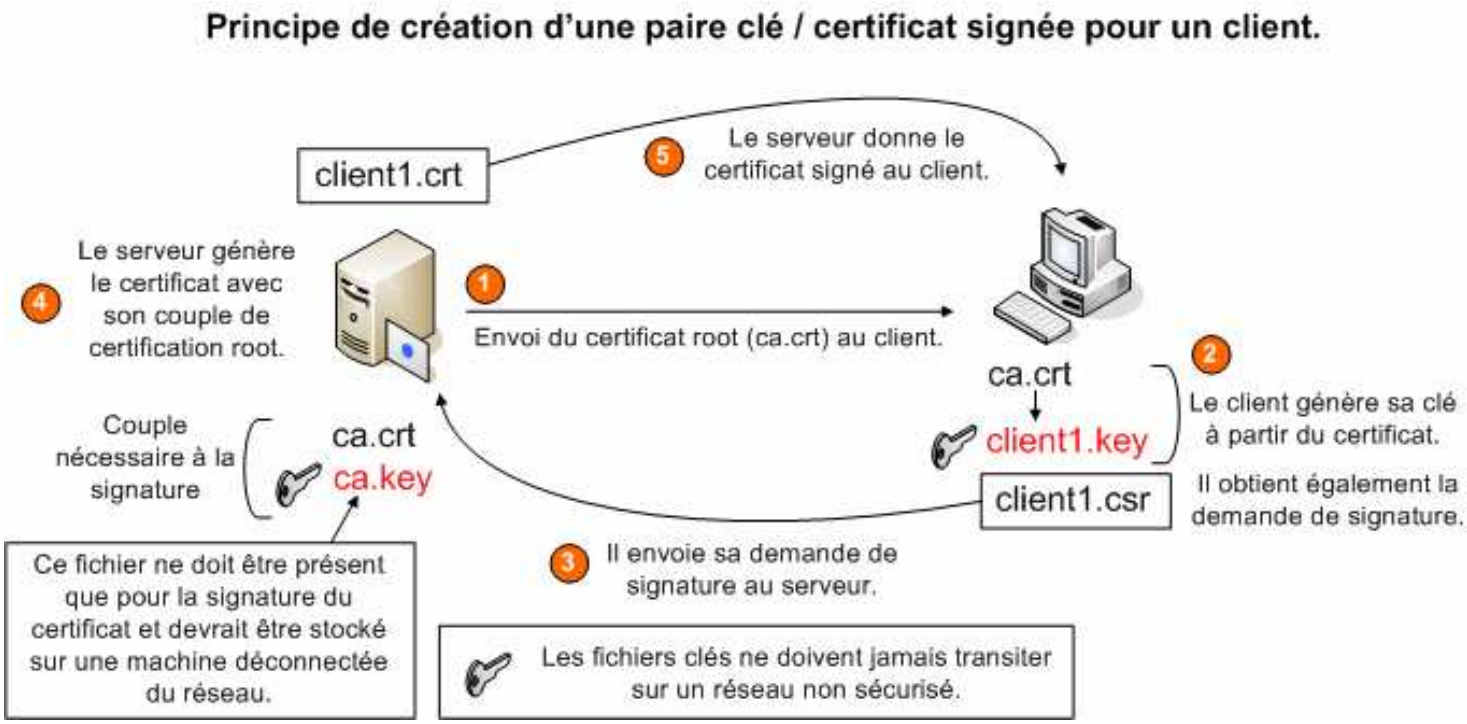
Nous avons trois nouveaux fichiers :
server.crt : Le certificat du serveur.
server.key : Sa clé, confidentielle.
server.csr : Une demande de signature de certificat, inutile puis que l'on vient de le signer.

Génération du certificat et de la clé pour les clients :

Il y a deux méthodes :

Génération locale des clés : on génère les clés sur le serveur et on lui envoie par un moyen sécurisé. Et j'insiste bien sur ce point. On ne doit jamais faire transiter les clés sur un réseau non sécurisé

Génération de la clé et d'une demande de certificat signé :



C'est le client qui génère lui-même sa clé avec le certificat root (« ca.crt » qu'on lui aura envoyé au préalable sans sécurité particulière). Il a également un fichier « .csr » qu'il peut envoyer sans crainte au serveur. Ce dernier va pouvoir générer un certificat signé en utilisant son certificat et sa clé root. Dès lors, le serveur donne au client son certificat « client1.crt ». C'est la méthodologie logique de ce type d'authentification qui apporte la sécurité nécessaire.

Je développe chaque méthode. Vous comprendrez mieux pourquoi certaines personnes se contentent de la première au dépend d'une faille de sécurité dans la procédure (envoi de la clé par un simple e-mail par exemple).

Préambule :
Notez bien que le « common name » doit être unique pour chaque client. Vous entrerez donc pour chaque couple clé / certificat généré ici successivement « client1 », « client2 », « client3 ».
Il existe une directive permettant à plusieurs clients de partager un même certificat mais c'est une pratique vivement déconseillée. Actuellement si un client tente de se connecter avec un certificat déjà en cours d'utilisation, la première connexion est perdue.

Génération locale des clés :

Nous sommes toujours dans C:\Program Files\OpenVPN\easy-rsa

> build-key client1
> build-key client2
> build-key client3

C'est tout ce qu'il y a faire. Je reviendrai après la seconde méthode sur ce que l'on doit faire des fichiers générés.

Génération de la clé et d'une demande de certificat signé :

Pré-requis :
Vous avez installé un OpenVPN GUI sur votre machine client. Pour le moment rien ne distingue encore vraiment le client du serveur.

Ouvrez une console et tapez :
> init-config

Editez le fichier vars.bat comme pour le serveur, puis continuez :

> vars
> clean-all

Ok, on passe aux choses sérieuses, copiez le certificat root généré par le serveur « ca.crt » (et surtout pas le confidentiel « ca.key ») dans /easy-rsa/keys. Dans un souci de paranoïa élevé et pour éviter une attaque de type « homme dans le milieu », vous devriez vérifier que le client a bien reçu (si vous l'avez envoyé par une connexion non sécurisée) le bon certificat root en le contactant par téléphone et en vérifiant ensemble un md5sum du fichier.

Pour une raison que j'ignore, il n'y pas autant de scripts batch pour Windows que de scripts Unix et il nous manque donc deux scripts que je vous propose de « refaire » en batch.

/build-req.bat : Il permet de générer le fichier « .csr » nécessaire à la demande du certificat.

```
@echo off
cd %HOME%
rem build a request for a cert that will be valid for ten
years
openssl req -days 3650 -nodes -new -keyout %KEY_DIR%\%1.key
-out %KEY_DIR%\%1.csr -config %KEY_CONFIG%
rem delete any .old files created in this process, to avoid
future file creation errors
del /q %KEY_DIR%\*.old
```

Vous copiez tout cela dans votre éditeur préféré (évitez Word tout de même, ça marchera moins bien%u2026) et vous enregistrez le tout sous build-req.bat

C'est en fait une version coupée de « build-key ». Vous pouvez éventuellement utiliser « build-key client1 » pour générer la demande, elle sera créée correctement mais vous aurez une erreur car le client ne pourra pas signer directement son certificat (ce que tente de faire « build-key »).

N'oubliez pas de donner un argument au programme et de spécifier son « common name » unique. En pratique, cela peut être le nom de l'ordinateur.

Vous voilà donc muni d'un nouveau fichier, « client1.csr ». Transmettez-le de la façon que vous souhaitez au serveur afin qu'il puisse générer votre certificat.

Sur le serveur, il nous manque le script pour générer le certificat signé, « sign-req.bat ». C'est en fait la seconde partie du script « build-key.bat » :

```
@echo off
cd %HOME%
rem sign the cert request with our ca, creating a cert/key
pair
openssl ca -days 3650 -out %KEY_DIR%\%1.crt -in
%KEY_DIR%\%1.csr -config %KEY_CONFIG%
rem delete any .old files created in this process, to avoid
future file creation errors
del /q %KEY_DIR%\*.old
```

Vous l'enregistrez sous « sign-req.bat » dans /easy-rsa. Le fichier « client1.csr » doit être copié dans /easy-rsa/keys.

Vous devez, pour signer un certificat, avoir le certificat root (« ca.crt ») et sa clé (« ca.key ») dans le même répertoire que la demande de signature.

Appelez depuis une console avec les variables initialisées correctement (« vars.bat ») :
> sign-req client1

Vous retrouvez un fichier « client1.crt » que vous pouvez communiquer sans crainte au client.

C'est fini !

Note concernant la protection des certificats des clients :
Il est possible d'assigner un mot de passe aux certificats des clients. Il suffit pour cela d'enlever le paramètres « -nodes » dans le fichier « build-key.bat » et le mot de passe vous sera demandé à la création.
Même si cela renforce toujours plus la sécurité, cette mesure risque de devenir rapidement contraignante vous obligeant à rentrer le mot de passe à chaque utilisation du certificat.

Il serait bon de faire un petit résumé avec tous ces fichiers%u2026

```
Nom          Description                          Confidentiel
Emplacement
ca.crt          Certificat root               Non
/config de toutes les machines
ca.key Clé du certificat root       Oui            Coffre-
fort
server.crt Certificat du serveur       Non
 /config du server
server.key Clé du certificat du serveur   Oui
 /config du server
client1.csr Demande de signature       Non             Peut
être détruit
```

```
client1.crt Certificat du client                Non
  /config du client
client1.key Clé du certificat du client     Oui
/config du client
```

Vérifiez donc que tout est déjà au bon endroit.

On continue et termine de s'amuser avec les scripts pour le serveur.

Nous devons maintenant générer les paramètres Diffie Hellman (http://www.rsasecurity.com/rsalabs/node.asp?id=2248) pour le serveur :
> build-dh

Cette opération consiste dans la génération dans grand nombre premier. Pour ajouter de l'entropie au processus vous pouvez agiter le pointeur de la souris. Cette opération dure plusieurs dizaines de secondes à plusieurs minutes selon votre configuration.

Le fichier généré s'appelle « dh1024.pem » et doit uniquement être copié dans le /config du serveur. Ce fichier n'est pas confidentiel.

On pourrait s'en arrêter là, mais on va ajouter une dernière clé :
> openvpn --genkey --secret ta.key

Cette clé va permettre d'ajouter une couche de sécurité supplémentaire avec une authentification TLS. Cela va créer un pare-feu HMAC et aidera à contrer les attaques de type DoS et le flood de port UDP.

Cette clé est à copier dans le répertoire /config de chaque machine et sa prise en compte nécessitera une modification dans le fichier de configuration. Ce fichier est partagé entre les machines et secret. Vous devriez donc utiliser une connexion sécurisée préexistante pour le transmettre aux clients.

On résume à nouveau :

Dans le répertoire /config du serveur, vous avez copié : « ca.crt », « server.crt », « server.key », « dh1024.pem » et « ta.key ».

Dans le répertoire /config d'un client, on trouve : « ca.crt », « client1.crt » , « client1.key », « ta.key » .

Fichiers de configuration

Le serveur :

Copiez le fichier « server.ovpn » qui se trouve dans /sample-config dans /config.

Editez-le, voici quelques paramètres essentiels :

port 1194
Le port utilisé par OpenVPN. Vous devez rediriger le port si vous êtes derrière un routeur. Vous pouvez créer plusieurs VPN en utilisant des ports différents.

client-to-client
Il faut « dé-commenter » ce paramètre (enlever le point-virgule) afin que les clients puissent communiquer entre eux, sinon ils ne pourront communiquer qu'avec le serveur.

J'ai vu un tutoriel proposant de faire un VPN par client avec un port différent et de créer des ponts réseaux entre chaque connexion. C'est assez saugrenu à moins d'avoir un intérêt vraiment très spécial à le faire, je n'ai pas encore trouvé.

tls-auth ta.key 0 # This file is secret
Dé-commentez cette ligne si vous utilisez un fichier d'authentification TLS « ta.key » sur les machines.

Le client :

Copiez le fichier « client.ovpn » qui se trouve dans /sample-config dans /config.

Editez-le, voici quelques paramètres essentiels :

remote xx.xx.xx.xx 1194
Spécifiez ici l'adresse (ou le nom NetBIOS) et le port sur lequel joindre le serveur OpenVPN. Vous pouvez spécifier plusieurs adresses de serveur pour répartir la charge automatiquement et proposer une redondance.

cert client1.crt
key client1.key
Pensez à changer ces paramètres en fonctions des fichiers qui sont présents dans le répertoire /config de la machine.

ns-cert-type server
Dé-commentez cette ligne pour ajouter une vérification du certificat du serveur. Le serveur est préconfiguré correctement avec les fichiers par défaut proposés dans /easy-rsa. Cette protection permet d'éviter une attaque du type « homme dans le milieu ».

tls-auth ta.key 1
Dé-commentez cette ligne si vous utilisez un fichier d'authentification TLS « ta.key » sur les machines.

Félicitations ! C'est fini ! 

Double-cliquez sur l'icône avec les deux ordinateurs et le globe sur le serveur, puis les clients.

On croise les doigts...

```
Mon Mar 20 20:49:53 2006 OpenVPN 2.0.5 Win32-MinGW [SSL]
[LZO] built on Nov  2 2005
Mon Mar 20 20:49:53 2006 Diffie-Hellman initialized with
1024 bit key
Mon Mar 20 20:49:53 2006 Control Channel Authentication:
using 'ta.key' as a OpenVPN static key file
Mon Mar 20 20:49:53 2006 Outgoing Control Channel
Authentication: Using 160 bit message hash 'SHA1' for HMAC
authentication
Mon Mar 20 20:49:53 2006 Incoming Control Channel
Authentication: Using 160 bit message hash 'SHA1' for HMAC
authentication
Mon Mar 20 20:49:53 2006 TLS-Auth MTU parms [ L:1542 D:166
EF:66 EB:0 ET:0 EL:0 ]
Mon Mar 20 20:49:54 2006 TAP-WIN32 device [OpenVPN] opened:
\\.\Global\{5ECDB806-2DF8-4518-9340-C4AA8B4323B3}.tap
Mon Mar 20 20:49:54 2006 TAP-Win32 Driver Version 8.1
Mon Mar 20 20:49:54 2006 TAP-Win32 MTU=1500
Mon Mar 20 20:49:54 2006 Notified TAP-Win32 driver to set a
DHCP IP/netmask of 10.8.0.1/255.255.255.252 on interface
{5ECDB806-2DF8-4518-9340-C4AA8B4323B3} [DHCP-serv: 10.8.0.2,
lease-time: 31536000]
Mon Mar 20 20:49:54 2006 Sleeping for 10 seconds...
Mon Mar 20 20:50:04 2006 Successful ARP Flush on interface
[3] {5ECDB806-2DF8-4518-9340-C4AA8B4323B3}
Mon Mar 20 20:50:04 2006 route ADD 10.8.0.0 MASK
255.255.255.0 10.8.0.2
Mon Mar 20 20:50:04 2006 Route addition via IPAPI succeeded
Mon Mar 20 20:50:04 2006 Data Channel MTU parms [ L:1542
D:1450 EF:42 EB:135 ET:0 EL:0 AF:3/1 ]
Mon Mar 20 20:50:04 2006 UDPv4 link local (bound):
[undef]:1194
Mon Mar 20 20:50:04 2006 UDPv4 link remote: [undef]
Mon Mar 20 20:50:04 2006 MULTI: multi_init called, r=256
v=256
Mon Mar 20 20:50:04 2006 IFCONFIG POOL: base=10.8.0.4
size=62
Mon Mar 20 20:50:04 2006 IFCONFIG POOL LIST
Mon Mar 20 20:50:04 2006 client1,10.8.0.4
Mon Mar 20 20:50:04 2006 client2,10.8.0.8
Mon Mar 20 20:50:04 2006 client5,10.8.0.12
Mon Mar 20 20:50:04 2006 client4,10.8.0.16
Mon Mar 20 20:50:04 2006 Initialization Sequence Completed
```

Ca s'allume en vert, ça marche !

Repérez les lignes « Learn : » qui indiquent les adresses IP que l'on vient d'attribuer à un client et tentez de pinger les machines entre elles.

```
Mon Mar 20 20:51:15 2006 MULTI: multi_create_instance called
Mon Mar 20 20:51:15 2006 192.168.0.254:3972 Re-using SSL/TLS
context
```

```
Mon Mar 20 20:51:15 2006 192.168.0.254:3972 LZO compression
initialized
Mon Mar 20 20:51:15 2006 192.168.0.254:3972 Control Channel
MTU parms [ L:1542 D:166 EF:66 EB:0 ET:0 EL:0 ]
Mon Mar 20 20:51:15 2006 192.168.0.254:3972 Data Channel MTU
parms [ L:1542 D:1450 EF:42 EB:135 ET:0 EL:0 AF:3/1 ]
Mon Mar 20 20:51:15 2006 192.168.0.254:3972 Local Options
hash (VER=V4): '14168603'
Mon Mar 20 20:51:15 2006 192.168.0.254:3972 Expected Remote
Options hash (VER=V4): '504e774e'
Mon Mar 20 20:51:15 2006 192.168.0.254:3972 TLS: Initial
packet from 192.168.0.254:3972, sid=1c6751ff 9e693f3c
Mon Mar 20 20:51:17 2006 192.168.0.254:3972 VERIFY OK:
depth=1, /C=FR/ST=LO/L=Nancy/O=MasterJul.net/CN=OpenVPN-
MJ.net/emailAddress=example@masterjul.net
Mon Mar 20 20:51:17 2006 192.168.0.254:3972 VERIFY OK:
depth=0,
/C=FR/ST=LO/O=MasterJul.net/CN=client4/emailAddress=
example-client@masterjul.net
Mon Mar 20 20:51:17 2006 192.168.0.254:3972 Data Channel
Encrypt: Cipher 'BF-CBC' initialized with 128 bit key
Mon Mar 20 20:51:17 2006 192.168.0.254:3972 Data Channel
Encrypt: Using 160 bit message hash 'SHA1' for HMAC
authentication
Mon Mar 20 20:51:17 2006 192.168.0.254:3972 Data Channel
Decrypt: Cipher 'BF-CBC' initialized with 128 bit key
Mon Mar 20 20:51:17 2006 192.168.0.254:3972 Data Channel
Decrypt: Using 160 bit message hash 'SHA1' for HMAC
authentication
Mon Mar 20 20:51:17 2006 192.168.0.254:3972 Control Channel:
TLSv1, cipher TLSv1/SSLv3 DHE-RSA-AES256-SHA, 1024 bit RSA
Mon Mar 20 20:51:17 2006 192.168.0.254:3972 [client4] Peer
Connection Initiated with 192.168.0.254:3972
Mon Mar 20 20:51:17 2006 client4/192.168.0.254:3972 MULTI:
Learn: 10.8.0.18 -> client4/192.168.0.254:3972
Mon Mar 20 20:51:17 2006 client4/192.168.0.254:3972 MULTI:
primary virtual IP for client4/192.168.0.254:3972: 10.8.0.18
Mon Mar 20 20:51:18 2006 client4/192.168.0.254:3972 PUSH:
Received control message: 'PUSH_REQUEST'
Mon Mar 20 20:51:18 2006 client4/192.168.0.254:3972 SENT
CONTROL [client4]: 'PUSH_REPLY,route 10.8.0.0
255.255.255.0,ping 10,ping-restart 120,ifconfig 10.8.0.18
10.8.0.17
```

Tentative de ping du serveur vers un client :

```
C:\Program Files\OpenVPN\easy-rsa>ping 10.8.0.18

Envoi d'une requête 'ping' sur 10.8.0.18 avec 32 octets de
données :

Réponse de 10.8.0.18 : octets=32 temps=1 ms TTL=128
Réponse de 10.8.0.18 : octets=32 temps=1 ms TTL=128
Réponse de 10.8.0.18 : octets=32 temps=1 ms TTL=128
Réponse de 10.8.0.18 : octets=32 temps=1 ms TTL=128

Statistiques Ping pour 10.8.0.18:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
```

```
Durée approximative des boucles en millisecondes :
    Minimum = 1ms, Maximum = 1ms, Moyenne = 1ms

C:\Program Files\OpenVPN\easy-rsa>
```

Explication de l'utilisation du mode bridge

I - Quel est le problème ?

C'est un problème qui revient souvent sur le topic, voilà une explication de la situation.

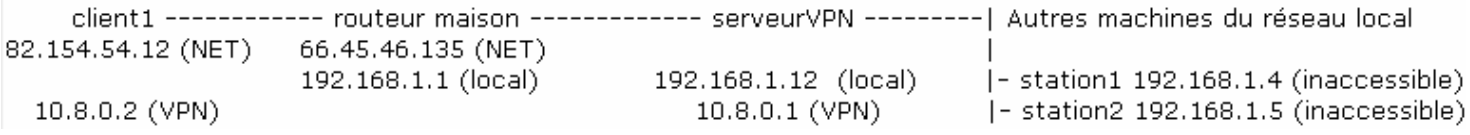
Vous avez chez vous un modem-routeur qui cache un réseau local de plage d'adresse 192.168.1.x.
Vous avez mis en place un serveur VPN, c'est une de vos machines de votre réseau, son adresse IP dans votre réseau local est 192.168.1.12.
Ce serveur possède sa propre plage d'adresse pour le VPN, qui est par défaut de la forme 10.8.0.x.

Lorsque vous souhaitez vous connecter sur votre VPN depuis l'extérieur, vous entrez l'adresse IP publique sur vous avez sur Internet, la redirection de port sur votre routeur vous amène sur la machine dédiée à OpenVPN.

Celui-ci va vous donner une adresse disponible dans la plage 10.8.0.x., mettons 10.8.0.2 (10.8.0.1 est l'adresse du serveur).

Vous arrivez à pinger 10.8.0.1, le serveur VPN. Vous arrivez également à pinger 192.168.1.12 c'est l'interface est sur la même machine.

Mais, il est impossible d'atteindre les autres machines de votre réseau local, derrière le serveur VPN.



Il y a deux façons d'obtenir ce que vous souhaitez faire, **le mode bridge (pont)** ou **le routage**.

II - La connexion de pont (ou bridge)

1 - Comment cela se passe-t-il en théorie ?

Avec le mode bridge, votre réseau VPN va utiliser la même plage d'adresses que votre réseau local. Autrement dit, votre machine sur laquelle est installé OpenVPN n'aura plus qu'une adresse IP, de la forme 192.168.1.x, nous choisirons 192.168.1.120 pour l'exemple. C'est une adresse IP fixe.
Dans le fichier de configuration, nous préciserons alors la plage que votre serveur

VPN attribuera aux clients. Mettons de 192.168.1.128 à 192.168.1.254. Du coup, quand votre client se connectera au VPN, celui-ci va lui donner la première adresse disponible, 192.168.1.128 et ce client fera littéralement "parti" de votre réseau local.

2 - Passons aux choses pratiques :

2.1 - Dans les fichiers de configuration :

- Côté serveur comme client :
Remplacer "dev tun" par "dev tap"

```
;dev tun
dev tap
```

- Sur le serveur, commentez la ligne "server" de cette façon :

```
;server 10.8.0.0 255.255.255.0
```

- Sur le serveur, vous allez donner l'adresse IP fixe du VPN sur le réseau local et la plage mise à disposition :
server-bridge adresse-serveur masque debut-plage fin-plage

```
server-bridge 192.168.1.120 255.255.255.0 192.168.1.128
192.168.1.254
```

2.2 - Dans les connexions réseau de Windows (Panneau de configuration) :

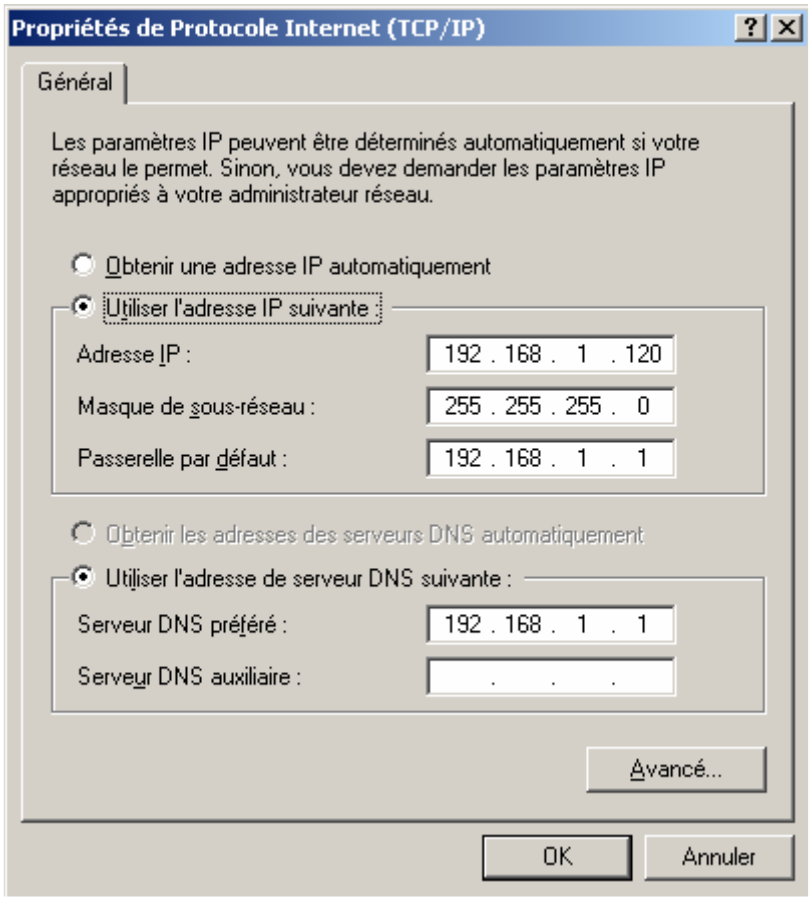
1 - Assurez-vous de ne pas avoir de connexion partageant un accès à Internet (cela empêche le bridging).
2 - Sélectionnez votre adaptateur réseau utilisé sur votre LAN **et** le virtuel de OpenVPN (TAP-Win32 Adapter). Pour cela, vous pouvez utiliser la touche CTRL + le clic de souris sur chacun des adaptateurs.
3 - Faites un clic droit sur l'une des connexions puis "Créer un pont".

A partir de ce moment, les deux connexions vont être liées et elles auront une interface commune, configurable sur le pont.

Sur cette image, la connexion TAP est "connectée" car le VPN est lancé.

4 - Faites un clic droit sur le pont, "Propriétés".
5 - Sélectionnez dans le second cadre "Protocole Internet (TCP/IP)" et faites "Propriétés".
6 - Choisissez "utilisez l'adresse IP suivante" et rentrez les informations comme

indiqué (pour rester en phase avec les paramètres cités précédemment). Dans cette configuration, vous avez un modem-routeur sur l'adresse 192.168.1.1.



7 - Il ne reste plus qu'à lancer le VPN (double clic sur l'icône prêt de l'horloge).
8 - Le client initie la connexion de son côté, ça marche ? Merveilleux, bienvenue chez vous. 😉

Je@nb a écrit :

Pour le routage tu as 2 possibilités :

- mettre sur chaque pc du lan une route vers ton vpn avec comme nexthop le serveur vpn
- configurer une route statique sur le routeur de la maison qui dit de router le réseau du vpn vers le serveur du vpn.

Ainsi dans le premier cas le routage est direct (chaque client sait où aller)
Dans le 2ème ils continuent à utiliser leur default gateway mais celle ci décide soit d'envoyer sur le serveur vpn les paquets si ils sont à destination du réseau du vpn soit d'envoyer vers le net les paquets à destinations des autres réseaux.
Après une solution mais qui n'est peut être bien utile est de mettre en place un mécanisme de routage dynamique comme RIP ou OSPF mais pour 2 machines ça ne vaut pas la peine

III - Questions et remarques

Pourquoi ne pas utiliser le DHCP du routeur ?

Parce que votre DHCP ne saurait pas différencier un client VPN d'un client de votre réseau local et lui donnerait des informations erronnées (mauvaise passerelle...). Il existe une façon de le faire pour Linux mais pas sous Windows à ma connaissance. Il faut voir du côté de votre routeur.

Toutefois, je ne pense pas que cette solution soit judicieuse car vous risquez de vous mélanger les pinceaux. Avec un serveur en IP fixe et une plage d'adresse dédiée, vous savez où vous allez.

Ca ne m'arrange pas cette histoire de pont. / J'aimerais bien garder deux plages distinctes.

Dans ce cas, il faut passer par le routage. En gros, il faut "pousser" des routes sur les clients pour qu'ils sachent quoi faire des requêtes allant vers votre réseau local. En contrepartie, il faut que les postes derrière votre routeur VPN connaissent la route de retour vers le serveur VPN. J'écrirai une explication plus détaillée sur cette façon de procéder.

Remerciements aux différents tutoriels et documentations trouvés sur Internet et en particulier : http://www.sbeattyconsulting.com/blog/index.php?p=3 et le « how-to » de OpenVPN.

MasterJul