

Installation d'un serveur VPN STRONGSWAN et d'un client THE GREENBOW

Pré requis :

Serveur :

Une machine comportant deux cartes réseau compatibles avec Linux.
Linux Redhat 8 installé et mis à jour avec GRUB comme chargeur de démarrage.
Cette machine accède bien à Internet et route la trafic du réseau local (1).
Une adresse IP fixe sur la connexion Internet de cette passerelle.

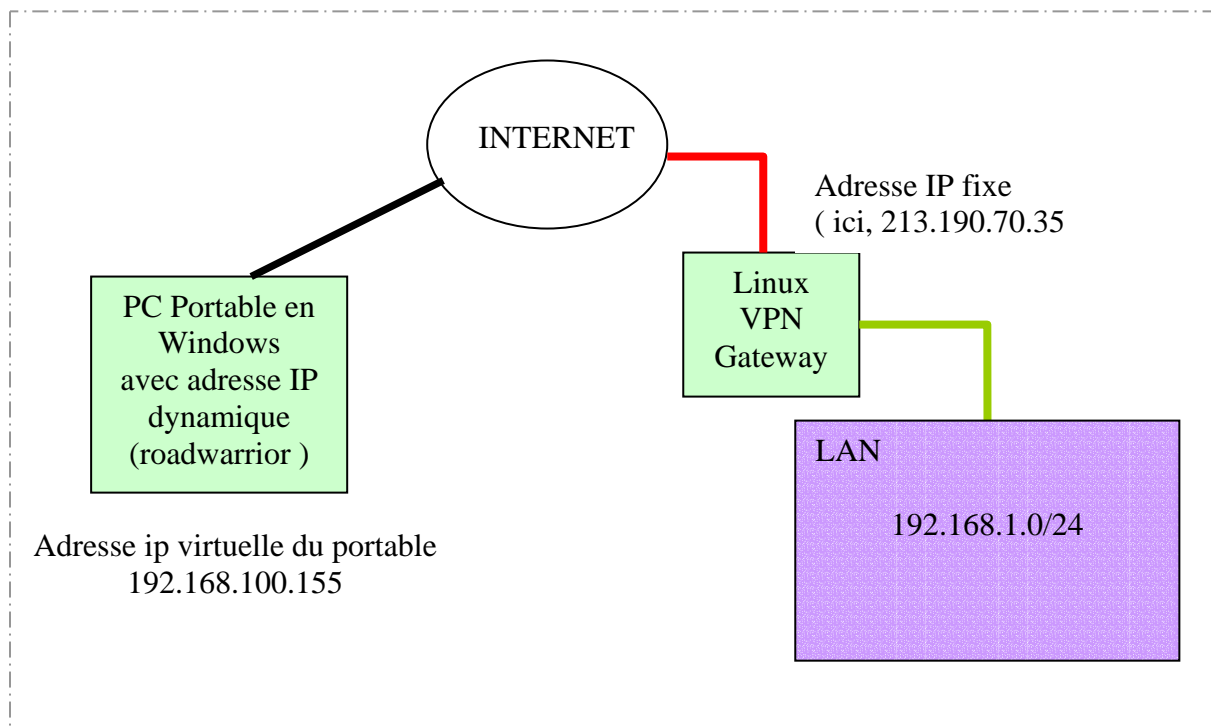
Les RPMS de STRONGSWAN téléchargés sur le site
<http://www.lamerzklan.de/~eldoc/strongswan/>

On a besoin du paquet RPM du kernel correspondant au processeur de la machine, ici, j'ai utilisé le RPM `kernel-2.4.26-4.ipsec.i386.rpm` et des commandes de STRONGSWAN `strongswan-userland-2.0.2-rh8.i386.rpm`.

Client :

Un PC en Windows.
Le client VPN THE GREENBOW (http://www.thegreenbow.fr/vpn_down.html) v2.50 et l'outil appelé « certificate » pour extraire les certificats créés sur le serveur Linux.

Schéma de l'architecture :



Installation :

Serveur :

On installe le nouveau kernel compilé avec ipsec par la commande « **rpm -ivh kernel-2.4.26-4.ipsec.i386.rpm** ».

On vérifie ensuite dans le fichier grub.conf que c'est bien ce noyau qui va être démarré par défaut, ce qui n'est pas le cas en général. Corrigez le fichier en conséquence.

Modifiez le paramètre « default=x » par la valeur correspondante à l'ordre de votre noyau ipsec.

Installez ensuite le paquet strongswan-userland-2.0.2-rh8.i386.rpm avec la commande « **rpm -ivh strongswan-userland-2.0.2-rh8.i386.rpm** ».

Vérifiez avec la commande « **ntsysv** » que le démon **ipsec** est coché pour démarrer au boot de la machine, puis rebootez sur votre nouveau noyau.

Client :

On lance l'exécutable d'installation « thegreenbow_vpn_client_250.exe », et on suit l'assistant.

Configuration :

Configuration du serveur STRONGSWAN :

J'ai utilisé la documentation du site de NAT CARLSON

(<http://www.natecarlson.com/linux/ipsec-x509.php>) pour configurer les certificats.

STRONGSWAN qui est un FORK de FREESWAN du même genre que OPENSWAN sur lequel est basé cette documentation.

Installation de votre Certificate Authority :

1) Ouvrez le fichier /usr/share/ssl/openssl.cnf dans votre éditeur préféré.

Ce fichier contient des valeurs par défaut pour la génération de certificats OpenSSL.

Nous allons changer les options suivantes:

'default_days ':

C'est la durée, en jours, durant laquelle vos certificats seront valides, et la valeur par défaut est de 365 jours, soit 1 an.

Je place cette valeur à '3650', ce qui donnera 10 ans de validité à notre certificats.

Puisque c'est pour l'usage interne, je suis assez confiant sur les risques de sécurité encourus d'avoir un certificat valide pendant longtemps - si vous le perdez ou autre, vous pouvez le révoquer sans problème.

La section '[req_distinguished_name] ':

Vous n'avez pas vraiment « besoin » de changer les options au-dessous du

req_distinguished_name; ce ne sont que des valeurs par défaut (telles que le lieu, le nom de compagnie, etc..) pour la génération de certificat. Je trouve plus facile de les renseigner ici que de les retaper à chaque création de certificat.

2) créez un dossier pour stocker votre CA. J'emploie généralement quelque chose comme `/var/sslca`; vous pouvez employer le nom que vous voulez. Changez les permissions du dossier en **700**, de sorte que les gens ne puissent pas accéder aux clefs privées auxquelles ils ne sont pas supposés accéder.

3) Éditez le script `/usr/share/ssl/misc/CA`, et mettez la ligne qui indique `'DAYS='days 365` à un nombre très élevé (ceci indique combien de temps le certificat de l'autorité de certificat est valide.) Soyez sûr que ce nombre est plus haut que la valeur de l'étape 1; ou bien Windows peut ne pas accepter vos certificats. Notez que si ce nombre est trop haut, il peut poser des problèmes - je le place généralement entre 15-20 ans.

4) lancez la commande `'CA -newca '`.

Renseignez les questions posées avec l'aide des exemples ci-dessous en modifiant les valeurs qui vous concernent.

L'entrée d'exemple est en rouge, et les commentaires sont en bleu.

Soyez sûr de n'employer aucun caractère non-alphanumérique, tel que des tirets, des virgules, des signes +, etc... Ces caractères peuvent rendre les choses plus difficiles.

```
nate@example:~/sslca$ /usr/share/ssl/misc/CA -newca
CA certificate filename (or enter to create)
(enter)
Making CA certificate ...
Using configuration from /usr/lib/ssl/openssl.cnf
Generating a 1024 bit RSA private key
.....
..+++
.....+++
writing new private key to './demoCA/private/./cakey.pem'
Enter PEM pass phrase: (entrez le mot de passe) C'est le mot de passe dont
vous aurez besoin pour créer tous les autres certificats.
Verifying password - Enter PEM pass phrase: (répétez le mot de passe)
-----
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) [AU]: US(enter) Écrivez votre code de pays
ici, FR pour France
State or Province Name (full name) [Some-State]: State(enter) Écrivez votre
etat/province ici, par exemple Midi-Pyrenees
Locality Name (eg, city) []:City(enter) Entrez le nom de votre ville ici
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
ExampleCo(enter) Écrivez votre nom de compagnie ici (ou laissez le blanc)
Organizational Unit Name (eg, section) []:(enter) OU, comme vous voulez. Je
le laisse habituellement vide.
Common Name (eg, YOUR name) []:CA(enter) Le nom de votre Certificate
Authority
Email Address []:ca@example.com(enter) Adresse Email
nate@example:~/sslca$
```

Créons également un fichier `crl`, dont vous aurez besoin sur votre passerelle :

```
nate@example:~/sslca$ openssl ca -gencrl -out crl.pem
```

Vous devrez mettre à jour ce fichier CRL à chaque fois que vous révoquerez un certificat.

C'est fait, vous avez maintenant votre propre Certificate Authority que vous pouvez employer pour produire des certificats.

Maintenant, vous devrez produire d'un certificat pour chaque machine qui établira une connexion IPSec. Ceci inclut la passerelle, et chacune de vos machines clientes.

Ce paragraphe détaille comment créer un certificat, et le convertir en format nécessaire à Windows.

Générer un certificat pour la passerelle :

Encore une fois, nous emploierons le script CA.

Cette fois, au lieu de lui demander de créer un nouveau « Certificate Authority », nous lui demandons de signer un certificat:

```
nate@example:~/sslca$ /usr/share/ssl/misc/CA -newreq
Using configuration from /usr/lib/ssl/openssl.cnf
Generating a 1024 bit RSA private key
.....+++
.....+++
writing new private key to 'newreq.pem'
Enter PEM pass phrase: (entrez le mot de passe) Mot de passe pour chiffrer
la nouvelle clef privée du certificat - vous en aurez besoin!
Verifying password - Enter PEM pass phrase:(répétez le mot de passe)
-----
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) [AU]:US(enter)
State or Province Name (full name) [Some-State]:State(enter)
Locality Name (eg, city) []:City(enter)
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ExampleCo(enter)
Organizational Unit Name (eg, section) []:(enter)
Common Name (eg, YOUR name) []:host.example.com(enter) Ceci peut être un
hostname, un vrai nom, une adresse de E-mail, ou n'importe quoi d'autre.
Email Address []:user@example.com(enter) (facultatif)

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:(enter)
An optional company name []:(enter)
Request (and private key) is in newreq.pem
```

Ce que nous venons de faire est de générer une demande de certificat - c'est le même type de demande que vous enverriez à Thawte ou à Verisign pour obtenir un certificat généralement utilisé pour SSL. Pour notre usage, cependant, nous le signerons avec notre propre CA:

```
nate@example:~/sslca$ /usr/share/ssl/misc/CA -sign
Using configuration from /usr/lib/ssl/openssl.cnf
Enter PEM pass phrase: (mot de passe que vous avez entré en créant le ca)
```

```

Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName :PRINTABLE:'US'
stateOrProvinceName :PRINTABLE:'State'
localityName :PRINTABLE:'City'
organizationName :PRINTABLE:'ExampleCo'
commonName :PRINTABLE:'host.example.com'
emailAddress :IA5STRING:'user@example.com'
Certificate is to be certified until Feb 13 16:28:40 2012 GMT (3650 days)
Sign the certificate? [y/n]:y(enter)

1 out of 1 certificate requests certified, commit? [y/n]y(enter)
Write out database with 1 new entries
Data Base Updated
(certificat élagué)
Signed certificate is in newcert.pem

```

Ensuite, renommez les fichiers créés en noms qui auront plus de sens pour notre usage futur.

```

nate@example:~/sslca$ mv newcert.pem host.example.com.pem
nate@example:~/sslca$ mv newreq.pem host.example.com.key

```

C'est tout ce qui est nécessaire de faire pour les passerelles Openswan - Vous aurez besoin de ces deux fichiers ainsi que du fichier 'cacert.pem' créé dans le dossier 'demoCA', et du fichier 'crl.pem' généré plus tôt, mais je détaillerai ceci plus tard.

Création d'un certificat pour le client Greenbow Windows :

La création du certificat du client se fait exactement de la même manière que pour la passerelle.

Utilisez par contre winhost.example.com, au lieu de host.example.com dans la procédure afin de distinguer les certificats.

Il faut ensuite convertir ce certificat au format p12 reconnu par Windows.

```

$ openssl pkcs12 -export -in winhost.example.com.pem -inkey
winhost.example.com.key -certfile demoCA/cacert.pem -out
winhost.example.com.p12

```

Installation des certificats :

1) Installez les fichiers à leur endroit respectifs (créez les dossiers de destination si le RPM ne l'a pas fait pour vous).

```

$ cp /var/sslca/host.example.com.key /etc/ipsec.d/private
$ cp /var/sslca/host.example.com.pem /etc/ipsec.d/certs
$ cp /var/sslca/winhost.example.com.key /etc/ipsec.d/private
$ cp /var/sslca/winhost.example.com.pem /etc/ipsec.d/certs
$ cp /var/sslca/demoCA/cacert.pem /etc/ipsec.d/cacerts
$ cp /var/sslca/crl.pem /etc/ipsec.d/crls/crl.pem

```

Une fois que vous avez créé le certificat p12 du client Windows, copiez le sur une disquette (2)

Configurer Openswan sur la passerelle :

Pour info, ma machine de test s'appelle fw2.caplaser.net et j'ai créé les certificats avec ce nom. Adaptez les lignes suivantes à votre cas.

Configuration du fichier ipsec.secrets :

Ajoutez la ligne suivante au fichier **/etc/ipsec.secrets** :

```
: RSA host.example.com.key "password"
```

Le mot de passe indiqué ici est le mot de passe utilisé lors de la création du certificat SSL.

Voici le fichier **ipsec.secrets** résultant de mes tests où j'ai supprimé le contenu du certificat de base qui prend pas mal de place☺.

```
: RSA {
  # RSA 2192 bits fw2.caplaser.net Fri Sep 24 13 :26 :26 2004
  # for signatures only, UNSAFE FOR ENCRYPTION
  #tout un tas de trucs déjà en place et à ne pas supprimer car la commnde
  "ipsec verify" n'aime pas du tout ça...
}
# do not change the indenting of that « } »
# La ligne que j'ai ajouté
: RSA fw2.caplaser.net.key "password"
```

Configuration du fichier ipsec.conf :

Voici mon fichier **/etc/ipsec.conf**

```
# /etc/ipsec.conf - strongSwan Isec configuration file
# RCSID $Id : ipsec.conf.in,v 1.2 2004/03/15 21 :03 :06 as Exp $

# This file : /usr/share/doc/freeswan/ipsec.conf-sample
#
# Manual : ipsec.conf.5
#
# Help :
# http://www.strongsec.com/freeswan/install.htm

version 2.0 # conforms to second version of ipsec.conf specification

# basic configuration
config setup
    interfaces=%defaultroute
    nat_traversal=yes
    virtual_private=%v4:10.0.0.0/8,%v4:172.16.0.0/12,%v4:192.168.0.0/16
    # Debug-logging controls :«none» for (almost) none,«all» for lots.
    Klipsdebug=none
    plutodebug=none
    # crlcheckinterval=600
    # strictcrlpolicy=yes

conn %default
    keyingtries=1
    compress=yes
```

```

        disablearrivalcheck=no
        authby=rsasig
        lefttrsasigkey=%cert
        righttrsasigkey=%cert

# OE policy groups are disabled by default
conn block
    auto=ignore

conn clear
    auto=ignore

conn private
    auto=ignore

conn private-or-clear
    auto=ignore

conn clear-or-private
    auto=ignore

conn packetdefault
    auto=ignore

# Add connections here.

Conn roadwarrior-net
    leftsubnet=192.168.1.0/24
    also=roadwarrior

conn roadwarrior
    left=%defaultroute
    leftcert=fw2.caplaser.net.pem
    right=%any
    rightsubnet=vhost:%no,%priv
    auto=add
    pfs=yes

```

Démarrer le service ipsec :

Une fois configuré, on peut démarrer le service ipsec, avec la commande :
/etc/init.d/ipsec restart

J'utilise restart au lieu de start, car le RPM active le service ipsec et on a déjà rebooté la machine. Le service ipsec est donc normalement déjà en fonctionnement avec la conf par défaut.

Les Logs de Strongswan se trouvent dans **/var/log/secure**.

Ces derniers vous indiqueront si vous avez des problèmes de configuration.

(1) Vérifiez que la valeur `net.ipv4.ip_forward` dans le fichier **/etc/sysctl.conf** soit bien à 1.

Si ce n'est pas le cas, modifiez la et tapez la commande :

```
/etc/init.d/network restart
```

Vérifiez également que le MASQUERADE est bien actif pour le trafic de votre réseau local sortant vers internet, sinon, tapez les commandes :

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE  
iptables-save > /etc/sysconfig/iptables
```

Attention, remplacer eth0 par l'interface qui est sur Internet pour votre machine, par exemple ppp0 pour de l'adsl .

Reportez vous à une documentation sur le routage et le firewall si vous avez besoin de plus d'informations à ce sujet.

(2) commandes utiles :

```
mount /mnt/floppy
```


```
copy mon_certificat.p12 /mnt/floppy
```

```
umount /mnt/floppy
```

Configuration du client VPN THE GREENBOW :

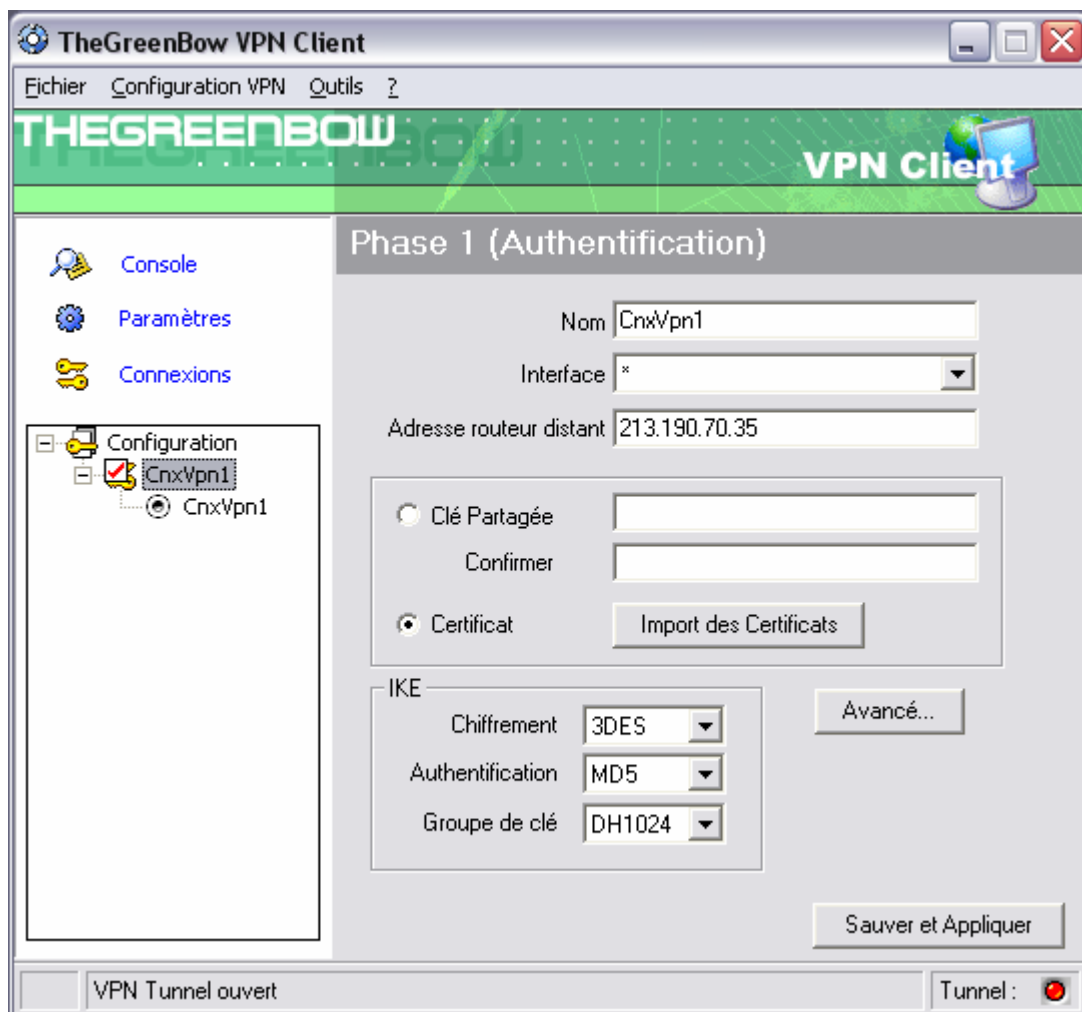
Créez les certificats nécessaires à partir du fichier p12 copié précédemment à partir du serveur et copiez les dans le dossier de votre choix (cf doc <http://www.thegreenbow.fr/doc/greenbow-x509.pdf>).



Lancer l'icône présente sur votre bureau Windows et vous verrez apparaître cette icône bleue en bas de votre écran .

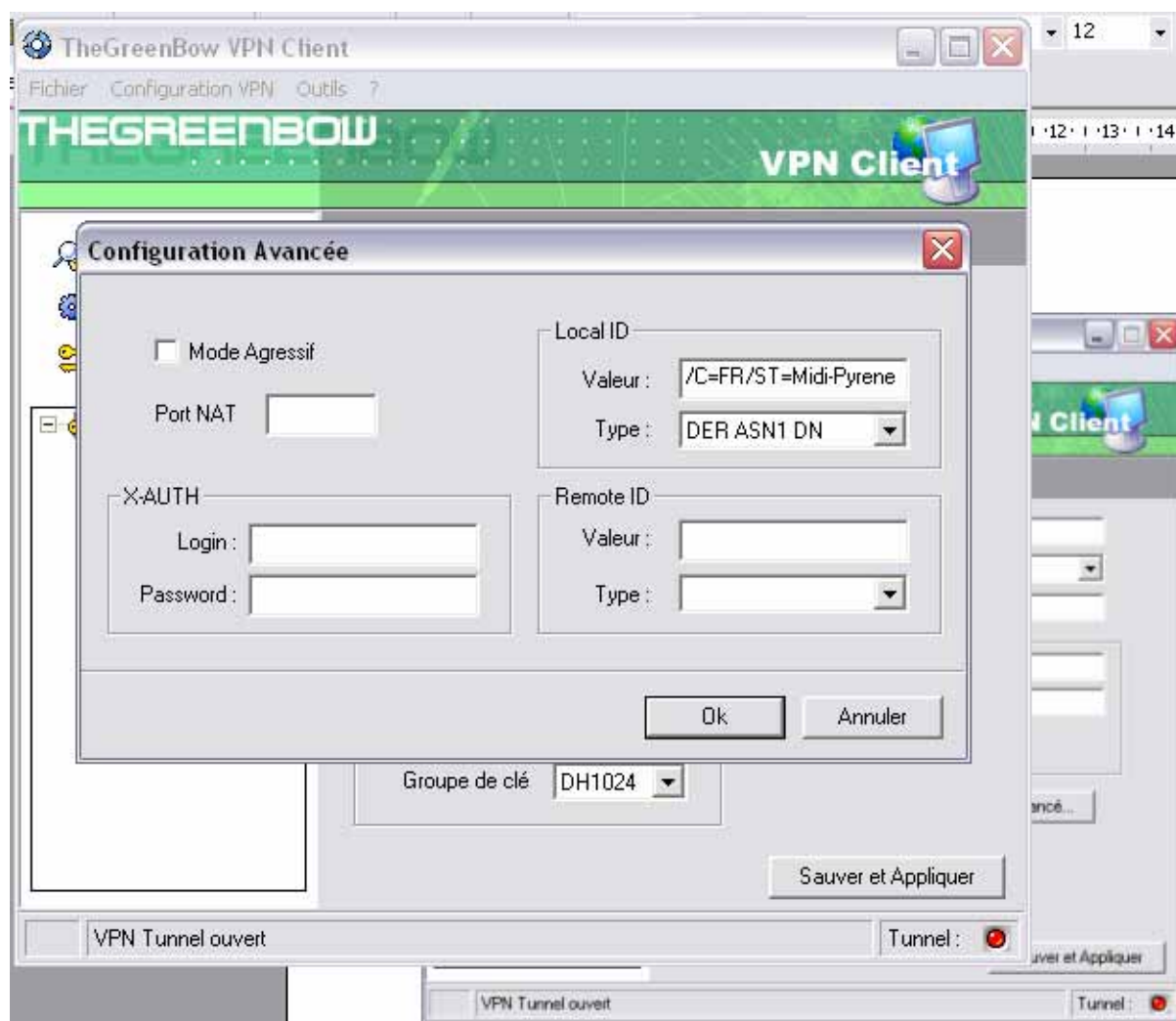
Faites un clic droit sur l'icône et sélectionnez « Connexions ».

Créez une nouvelle phase 1 en modifiant l'adresse 213.190.70.35 par l'adresse ip fixe Internet de votre passerelle Linux :

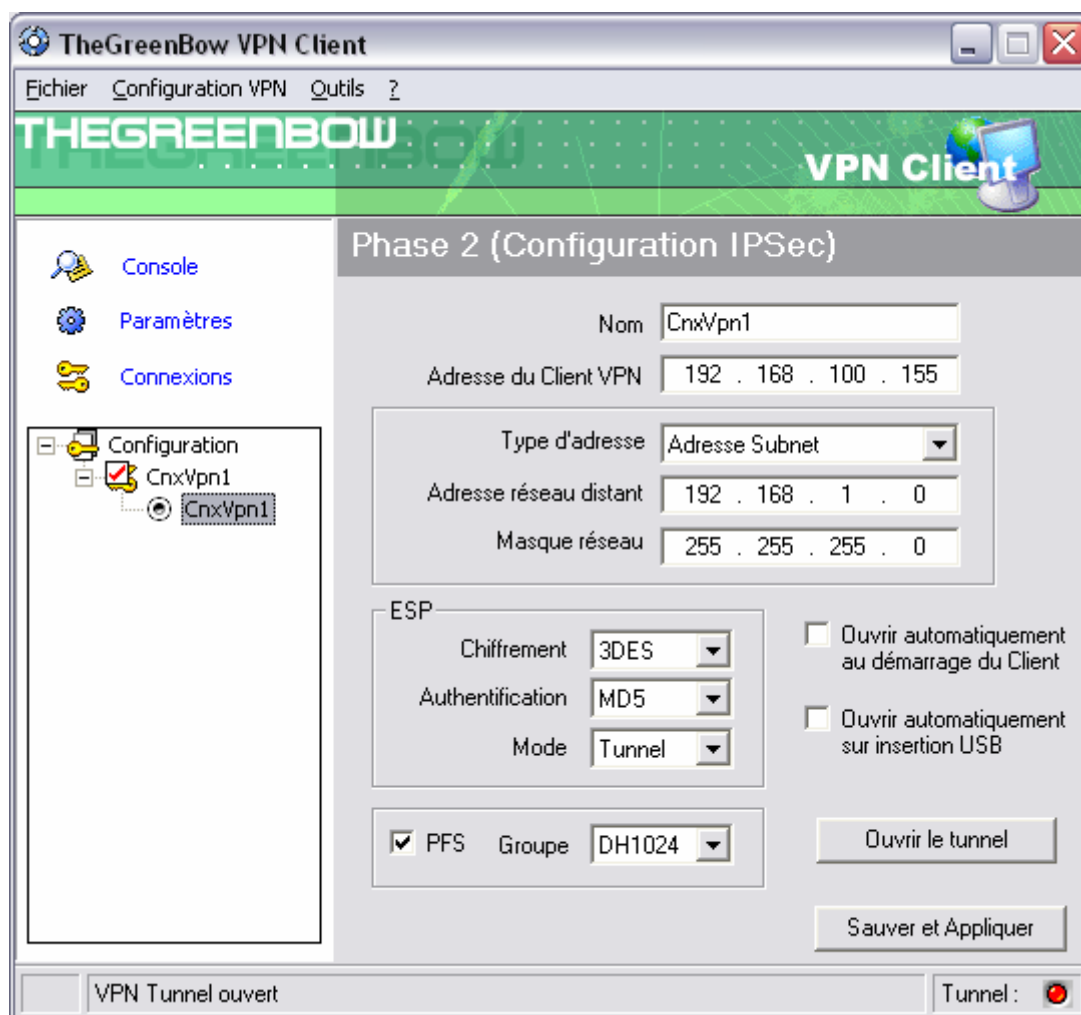


Sélectionnez « Certificat » et importez les certificats que vous venez de créer avec l'utilitaire « certificate » comme indiqué dans la documentation de ce dernier.

Cliquez sur le bouton « Avancé... » et faites un copier coller de la valeur « DER ASN1 DN » créée par « certificate ».



Créez ensuite une phase 2 comme ceci :



Une fois les paramètres sauvés, vous pouvez cliquer sur « Ouvrir le tunnel ». Ce dernier devrait monter correctement.

En cas de messages d'erreur, jetez un œil sur la doc suivante : http://www.thegreenbow.fr/doc/vpn_troubleshooting_en.pdf

Une fois connecté, vous pouvez pinguer les machines de votre réseau local distant.