

# Réseau Privé Virtuel et WIFI.

## Présentation :

Un Réseau privé virtuel appelé VPN est un système permettant de créer un lien direct entre des ordinateurs distants. On utilise notamment ce terme dans le travail à distance, ainsi que pour l'accès à des structures de type cloud computing. Un VPN permet d'accéder à des ordinateurs distants comme si l'on était connecté au réseau local. On peut ainsi avoir un accès au réseau interne (réseau d'entreprise, par exemple).

Un VPN dispose généralement aussi d'une passerelle permettant d'accéder à l'extérieur, ce qui permet de changer l'adresse IP source apparente de ses connexions. Cela rend plus difficile l'identification et la localisation approximative de l'ordinateur émetteur par le fournisseur de service.

## Objectif :

Installer et configurer un VPN avec le logiciel libre OpenVPN. Mise en place d'une borne WIFI avec deux SSID différents.

## Pré requis :

- Deux ordinateurs un sur Linux pour le serveur (Debian 8.2) et un sur Windows pour le client (W7).
- Avoir une connexion internet
- Avoir une IP fixe pour le serveur
- Mon serveur s'appelle openvpn et son @IP est 192.168.1.140/24. Le client est en dhcp sur le même réseau que le serveur.

## Sommaire :

- I. Installation d'OpenVPN
- II. Construction d'une PKI
- III. Configuration du serveur
- IV. Configuration Client VPN
- V. Configuration de la borne WIFI Cisco 1200 Series

## I. Installation d'OpenVPN

Avant l'installation, mettre à jour les paquets :

```
root@openvpn:~# apt-get update
```

Puis installer les paquets :

```
root@openvpn:~# apt-get install openvpn openssh-server openssl
```

OpenVPN utilise les protocoles TLS et SSL et écoute sur les ports UDP ou TCP.

## II. Construction d'une PKI

On va créer deux répertoires et copier les scripts dans ce répertoire :

```
root@openvpn:~# mkdir /etc/openvpn/easy-rsa
root@openvpn:~# cp /usr/share/easy-rsa/* /etc/openvpn/easy-rsa/
root@openvpn:~# mkdir /etc/openvpn/easy-rsa/keys
```

On va se situer dans le répertoire où il y a les scripts :

```
root@openvpn:~# cd /etc/openvpn/easy-rsa/
root@openvpn:/etc/openvpn/easy-rsa# ls
build-ca          build-key-server  list-crl          sign-req
build-dh          build-req         openssl-0.9.6.cnf vars
build-inter      build-req-pass   openssl-0.9.8.cnf whichopensslcnf
build-key        clean-all       openssl-1.0.0.cnf
build-key-pass   inherit-inter    pkitsool
build-key-pkcs12 keys             revoke-full
```

On va éditer le fichier vars et modifier ces valeurs :

```
GNU nano 2.2.6          Fichier : vars          Modifié
export CA_EXPIRE=3650

# In how many days should certificates expire?
export KEY_EXPIRE=3650

# These are the default values for fields
# which will be placed in the certificate.
# Don't leave any of these fields blank.
export KEY_COUNTRY="FR"
export KEY_PROVINCE="France"
export KEY_CITY="Caen"
export KEY_ORG="BTS SIO"
export KEY_EMAIL="pm.quantin@sts-sio-caen.info"
export KEY_OU="PPE"
```

On initialise les variables et on clean:

```
root@openvpn:/etc/openvpn/easy-rsa# source ./vars
NOTE: If you run ./clean-all, I will be doing a rm -rf on /etc/openvpn/easy-rsa/
keys
root@openvpn:/etc/openvpn/easy-rsa# ./clean-all
```

On tape cette commande pour générer deux certificats CA:

```
root@openvpn:/etc/openvpn/easy-rsa# ./build-ca
```

Ces certificats sont présents maintenant dans le dossier keys :

```
root@openvpn:/etc/openvpn/easy-rsa# ls keys/
ca.crt  ca.key  index.txt  serial
```

Maintenant, on va créer le certificat du serveur, 3 commandes à faire :

```
root@openvpn:/etc/openvpn/easy-rsa# echo 01 > keys/serial
root@openvpn:/etc/openvpn/easy-rsa# chmod -R 0700 keys/
```

On peut lancer la commande maintenant :

```
root@openvpn:/etc/openvpn/easy-rsa# ./build-key-server serveurvpn
```

Important, répondre yes (y) aux deux questions qui suivent :

```
Certificate is to be certified until May 31 07:04:54 2026 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
```

On fait le certificat du client maintenant et toujours répondre y (Yes) aux deux dernières questions :

```
root@openvpn:/etc/openvpn/easy-rsa# ./build-key client1
```

NB : sur le client même, il lui faut le ca.cert.

Enfin, on va générer les paramètres Diffie Hellman. Cela dure un assez long moment et le fichier créé est nommé dh2048.pem dans le sous répertoire keys :

```
root@openvpn:/etc/openvpn/easy-rsa# ./build-dh
```

### III. Configuration du serveur

On va créer un utilisateur spécial openvpn et son groupe sans répertoire ni shell :

```
root@openvpn:/# groupadd openvpn
root@openvpn:/# useradd -d /dev/null -g openvpn -s /bin/false openvpn
```

Puis, on va récupérer le fichier de conf du serveur :

```
root@openvpn:/# cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz /etc/openvpn/  
root@openvpn:/# gunzip /etc/openvpn/server.conf.gz
```

On l'édite et on le modifie comme suit:

```
# Any X509 key management system can be used.  
# OpenVPN can also use a PKCS #12 formatted key file  
# (see "pkcs12" directive in man page).  
ca ca.crt  
cert serveurvpn.crt  
key serveurvpn.key # This file should be kept secret  
  
# Diffie hellman parameters.  
# Generate your own with:  
# openssl dhparam -out dh1024.pem 1024  
# Substitute 2048 for 1024 if you are using  
# 2048 bit keys.  
dh dh2048.pem
```

```
# Push routes to the client to allow it  
# to reach other private subnets behind  
# the server. Remember that these  
# private subnets will also need  
# to know to route the OpenVPN client  
# address pool (10.8.0.0/255.255.255.0)  
# back to the OpenVPN server.  
push "route 192.168.1.0 255.255.255.0"
```

```
# Select a cryptographic cipher.  
# This config item must be copied to  
# the client config file as well.  
cipher BF-CBC # Blowfish (default)  
;cipher AES-128-CBC # AES  
;cipher DES-EDE3-CBC # Triple-DES
```

```
# You can uncomment this out on  
# non-Windows systems.  
user openvpn  
group openvpn
```

```
# Silence repeating messages. At most 20  
# sequential messages of the same message  
# category will be output to the log.  
mute 20
```

## IV. Configuration du client Windows.

Installer le client VPN windows via le site : [openvpn-2.0.9-gui-1.0.3-install.exe](http://openvpn-2.0.9-gui-1.0.3-install.exe)

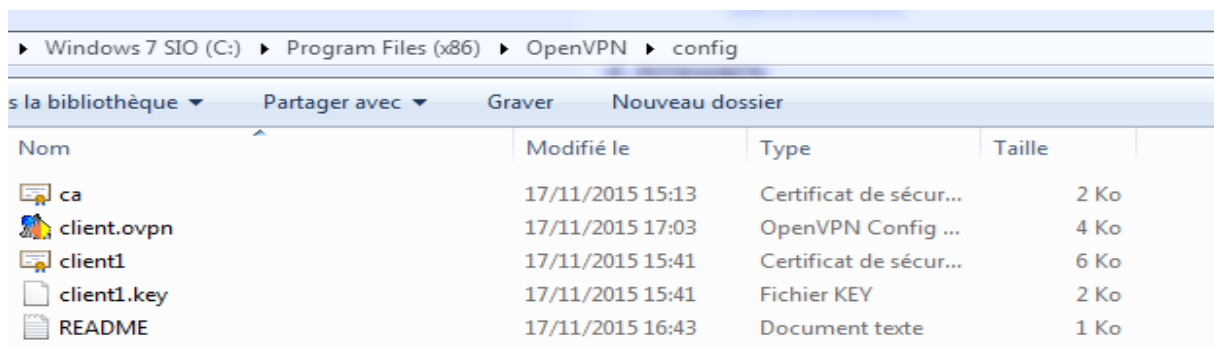
De plus, on copie le fichier de conf par défaut présent dans :

C:\Program Files\OpenVPN\Sample-config\clientopenvpn dans le sous repertoire config

Mettez dans le répertoire config le ca.crt, le client1.crt et le client1.key disponible dans le keys :

```
root@openvpn:/etc/openvpn/easy-rsa# cd keys/
root@openvpn:/etc/openvpn/easy-rsa/keys# ls
01.pem  client1.crt  index.txt          serial          serveurvpn.key
02.pem  client1.csr  index.txt.attr    serial.old
ca.crt  client1.key  index.txt.attr.old  serveurvpn.crt
ca.key  dh2048.pem  index.txt.old     serveurvpn.csr
```

Ce qui nous donne ceci :



Windows 7 SIO (C:) > Program Files (x86) > OpenVPN > config

Nom	Modifié le	Type	Taille
ca	17/11/2015 15:13	Certificat de sécur...	2 Ko
client.ovpn	17/11/2015 17:03	OpenVPN Config ...	4 Ko
client1	17/11/2015 15:41	Certificat de sécur...	6 Ko
client1.key	17/11/2015 15:41	Fichier KEY	2 Ko
README	17/11/2015 16:43	Document texte	1 Ko

On démarre le serveur openvpn : `service openvpn start`

Sur le windows, on remarque avec un clic droit en bas à droite puis on se connecte. Sur le serveur, le tun s'est créé :

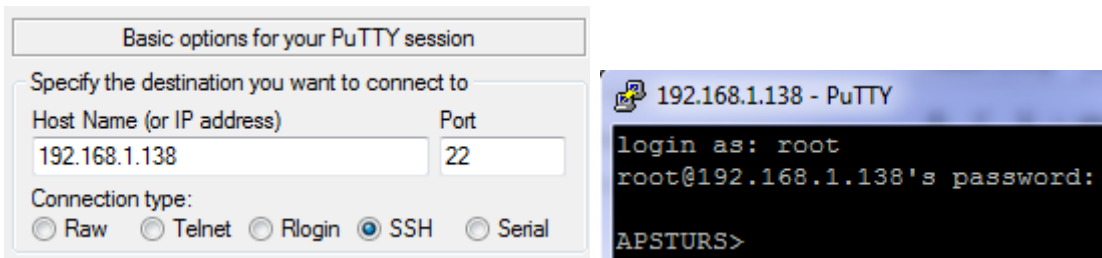
```
GNU nano 2.2.6      Fichier : /etc/openvpn/openvpn.log
Fri Nov 20 14:17:31 2015 OpenVPN 2.3.4 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO$
Fri Nov 20 14:17:31 2015 library versions: OpenSSL 1.0.1k 8 Jan 2015, LZO 2.08
Fri Nov 20 14:17:31 2015 NOTE: your local LAN uses the extremely common subnet $
Fri Nov 20 14:17:31 2015 Diffie-Hellman initialized with 2048 bit key
Fri Nov 20 14:17:31 2015 Socket Buffers: R=[212992->131072] S=[212992->131072]
Fri Nov 20 14:17:31 2015 ROUTE_GATEWAY 192.168.1.254/255.255.255.0 IFACE=eth0 H$
Fri Nov 20 14:17:31 2015 TUN/TAP device tun0 opened
Fri Nov 20 14:17:31 2015 TUN/TAP TX queue length set to 100
Fri Nov 20 14:17:31 2015 do_ifconfig, tt->ipv6=0, tt->did_ifconfig_ipv6_setup=0
Fri Nov 20 14:17:31 2015 /sbin/ip link set dev tun0 up mtu 1500
Fri Nov 20 14:17:31 2015 /sbin/ip addr add dev tun0 local 10.8.0.1 peer 10.8.0.2
Fri Nov 20 14:17:31 2015 /sbin/ip route add 10.8.0.0/24 via 10.8.0.2
```

Avec la commande ifconfig, le tunnel est bien crée :

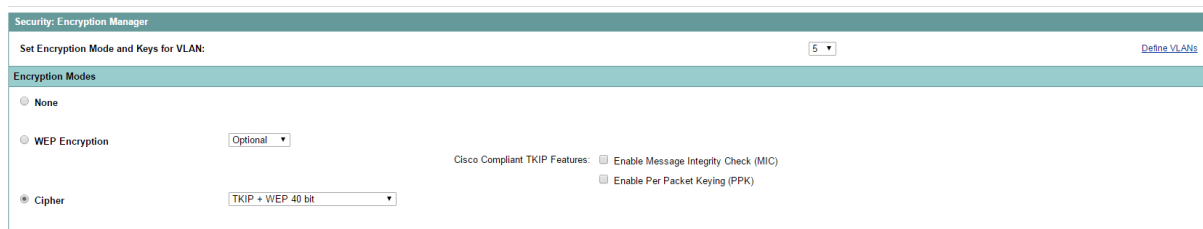
```
tun0    Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
-00
        inet adr:10.8.0.1  P-t-P:10.8.0.2  Masque:255.255.255.255
UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 lg file transmission:100
RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

### V. Configuration de la borne wifi Cisco 1200 series

Tout d’abord, il faut se connecter en console et mettre une adresse sur l’interface BV pour ainsi sur l’interface graphique. De plus, j’ai configuré SSH pour pouvoir me connecter à distance sur l’AP :



J’ai ensuite configuré les VLANs depuis l’interface graphique « Services » puis « VLAN ». Puis pour le chiffrement, je suis allé dans l’onglet Security puis « Encryption Manager » :



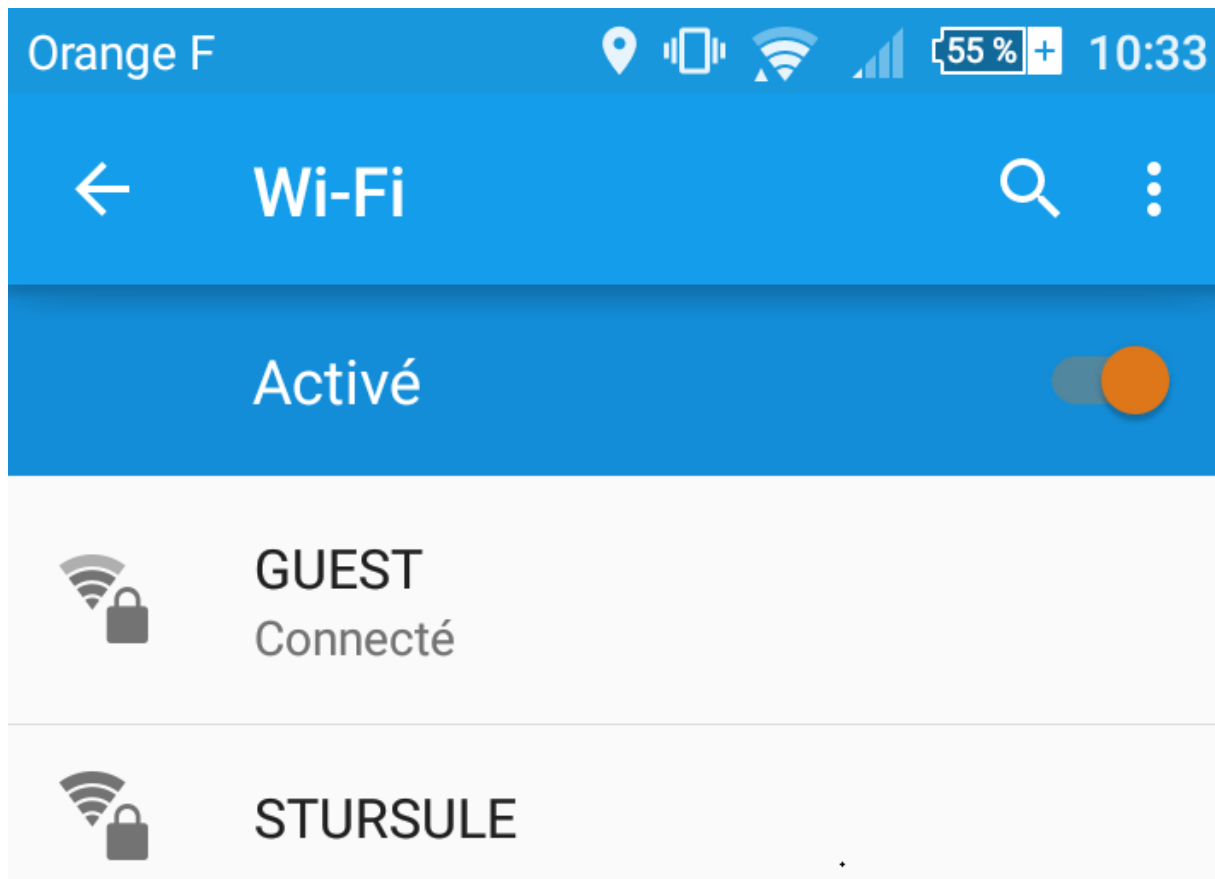
Enfin, j’ai déclaré les deux SSID depuis « SSID Manager ». On a un récapitulatif des SSID créés dans l’onglet « Express Security »

Service: Set Identifiers (SSIDs)						
SSID	VLAN	Radio	BSSID/Guest Mode	Open	Shared	Network EAP
GUEST	5	Radio0-802.11G	001b.548e.6ef0 ✓	no addition		
STURSULE	10	Radio0-802.11G	001b.548e.6ef1 ✓	no addition		

Comme dans l’onglet « Security » :

VLAN	Encryption Mode	WEP		Cipher					Key Rotation	
		MIC	PPK	TKIP	WEP40bit	WEP128bit	CKIP	CMIC		AES CCM
5	Cipher			✓	✓					
10	Cipher			✓						

Enfin, depuis un appareil mobile, on distingue bien les deux réseaux :



```
APSTURS#sh run
Building configuration...
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
hostname APSTURS
enable secret 5 $1$dKpB$D2QORDI0UIHbGqFsRCSuq0
ip subnet-zero
ip domain name fwl.com
ip name-server 10.103.0.5
ip ssh version 2
no aaa new-model
dot11 vlan-name guest vlan 10
```

```
dot11 vlan-name stursule vlan 20
dot11 ssid GUEST
    vlan 5
    authentication open
    guest-mode
    mbssid guest-mode
dot11 ssid STURSULE
    vlan 10
    authentication open
    authentication key-management wpa
    mbssid guest-mode
    wpa-psk ascii 7 02160D5E19140A2C4D5C001C
username Cisco password 7 112A1016141D
username root password 7 081343411D485744
bridge irb
interface Dot11Radio0
    ip address 192.168.1.137 255.255.255.0
    no ip route-cache
encryption vlan 10 mode ciphers tkip
encryption vlan 5 key 1 size 40bit 7 DD5A59824B7D transmit-key
    encryption vlan 5 mode ciphers tkip wep40
ssid GUEST
ssid STURSULE
    mbssid

speed basic-1.0 basic-2.0 basic-5.5 6.0 9.0 basic-11.0 12.0 18.0 24.0
36.0 48.0 54.0

station-role root
interface Dot11Radio0.5
    encapsulation dot1Q 5 native
    no ip route-cache
```



```
bridge-group 1
bridge-group 1 subscriber-loop-control
bridge-group 1 block-unknown-source
no bridge-group 1 source-learning
no bridge-group 1 unicast-flooding
bridge-group 1 spanning-disabled
interface Dot11Radio0.10
encapsulation dot1Q 10
no ip route-cache
bridge-group 10
bridge-group 10 block-unknown-source
no bridge-group 10 source-learning
no bridge-group 10 unicast-flooding
bridge-group 10 spanning-disabled
interface Dot11Radio0.15
encapsulation dot1Q 15
no ip route-cache
bridge-group 15
bridge-group 15 subscriber-loop-control
bridge-group 15 block-unknown-source
no bridge-group 15 source-learning
no bridge-group 15 unicast-flooding
bridge-group 15 spanning-disabled
interface Dot11Radio0.20
encapsulation dot1Q 20
no ip route-cache
bridge-group 20
bridge-group 20 subscriber-loop-control
bridge-group 20 block-unknown-source
```

```
no bridge-group 20 source-learning
no bridge-group 20 unicast-flooding
bridge-group 20 spanning-disabled
interface Dot11Radio0.25
encapsulation dot1Q 25
no ip route-cache
bridge-group 25
bridge-group 25 subscriber-loop-control
bridge-group 25 block-unknown-source
no bridge-group 25 source-learning
no bridge-group 25 unicast-flooding
bridge-group 25 spanning-disabled
interface FastEthernet0
ip address 192.168.1.139 255.255.255.0
no ip route-cache
duplex auto
speed auto
hold-queue 160 in
interface FastEthernet0.5
encapsulation dot1Q 5 native
no ip route-cache
bridge-group 1
no bridge-group 1 source-learning
bridge-group 1 spanning-disabled
interface FastEthernet0.10
encapsulation dot1Q 10
no ip route-cache
bridge-group 10
no bridge-group 10 source-learning
```

```
bridge-group 10 spanning-disabled
interface FastEthernet0.15
encapsulation dot1Q 15
no ip route-cache
bridge-group 15
no bridge-group 15 source-learning
bridge-group 15 spanning-disabled
interface FastEthernet0.20
encapsulation dot1Q 20
bridge-group 20
no bridge-group 20 source-learning
bridge-group 20 spanning-disabled
interface FastEthernet0.25
encapsulation dot1Q 25
no ip route-cache
bridge-group 25
no bridge-group 25 source-learning
bridge-group 25 spanning-disabled
interface BVI1
ip address 192.168.1.138 255.255.255.0
no ip route-cache
ip default-gateway 192.168.1.254
ip http server
control-plane
line con 0
line vty 0 4
login local
transport input ssh
end
```