

# VPN

## I- Préparatifs

### 1.La commande openvpn

On commence par l'installation des paquets comme *openvpn* et *openssl* pour le serveur et *openvpn* pour le client sur Linux.

*apt install openvpn*

*Apt instal -y openssl*

Installation de *wireshark* sur le machine serveur et client.

Placer la machine cliente dans le Vlan 60.

Commande *openvpn* essentielles à la construction du tunnel VPN et sécurisation minimale :

<i>--remote</i>	Adresse IP ou FQDN indique l'adresse de serveur VPN.
<i>--local</i>	AdresselP indique quelle IP ou interface utilisée localement.
<i>--dev</i>	Indique l'interface virtuelle.
<i>--port</i>	1194 par défaut.
<i>--proto</i>	UDP ou TCP.
<i>--verb</i>	Mode verbeux 1à5.
<i>--ifconfig</i>	Permet l'adressage virtuel du tunnel (ip remote).
<i>--genkey</i>	Génère une clef symétrique.
<i>--secret</i>	Précise le fichier contenant la clef.
<i>--push</i>	Pousse une instruction sur le client depuis le VPN.
<i>--server</i>	Réseau et masque pour distribuer les IPs virtuelles, la première est pour le serveur, désigne aussi le serveur TLS.
<i>--client</i>	Désigne un client TLS.
<i>--dh</i>	Précise les fichiers concernant la clef de Diffie Hell mans.
<i>--ca</i>	Précise les fichier concernant le certificat d'autorité
<i>--cert</i>	Précise les fichiers concernant le certificat machine.
<i>--key</i>	Précise les fichiers concernant la clef privée machine.

On créer un premier tunnel VPN avec du chiffrement symétrique, en utilisant le commande [openvpn](#) :

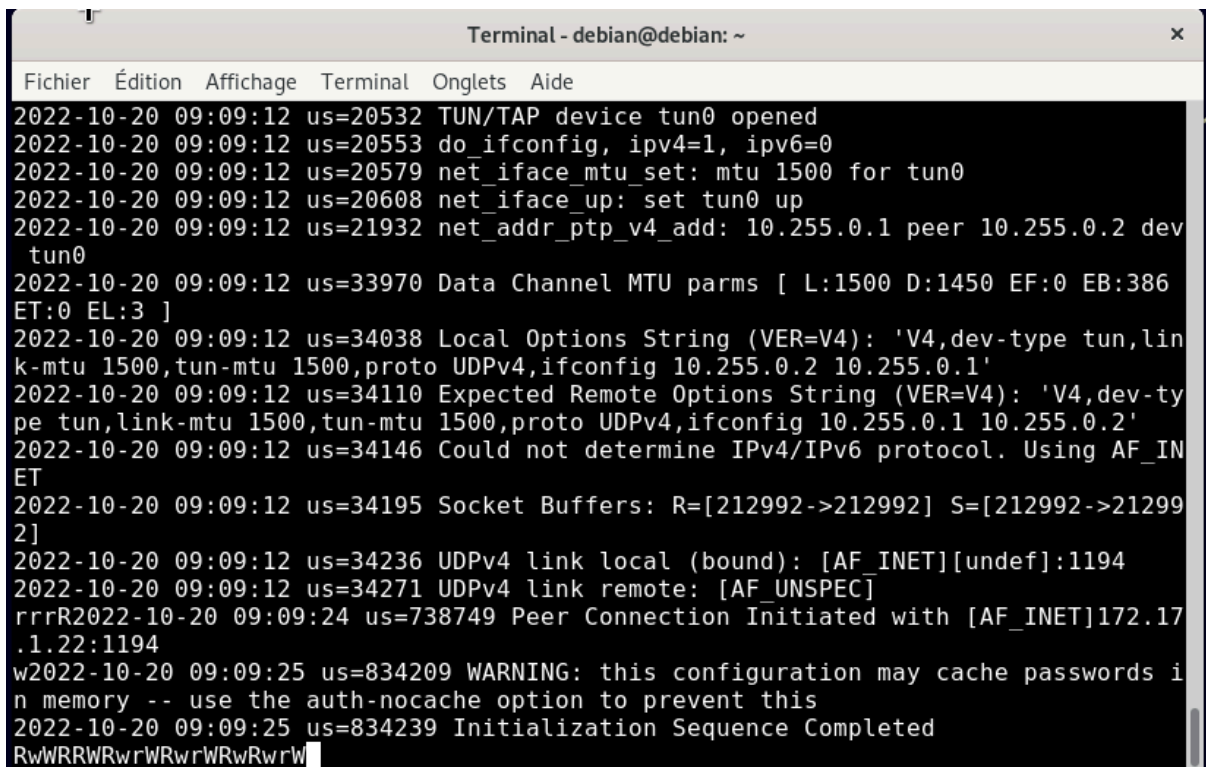
Sur le serveur :

```
Openvpn --dev tun0 --verb 5 --ifconfig 10.255.0.1 10.255.0.2
```

Sur le client :

```
Openvpn --dev tun0 --verb 5 --ifconfig 10.255.0.2 10.255.0.1 --remote 172.17.1.20
```

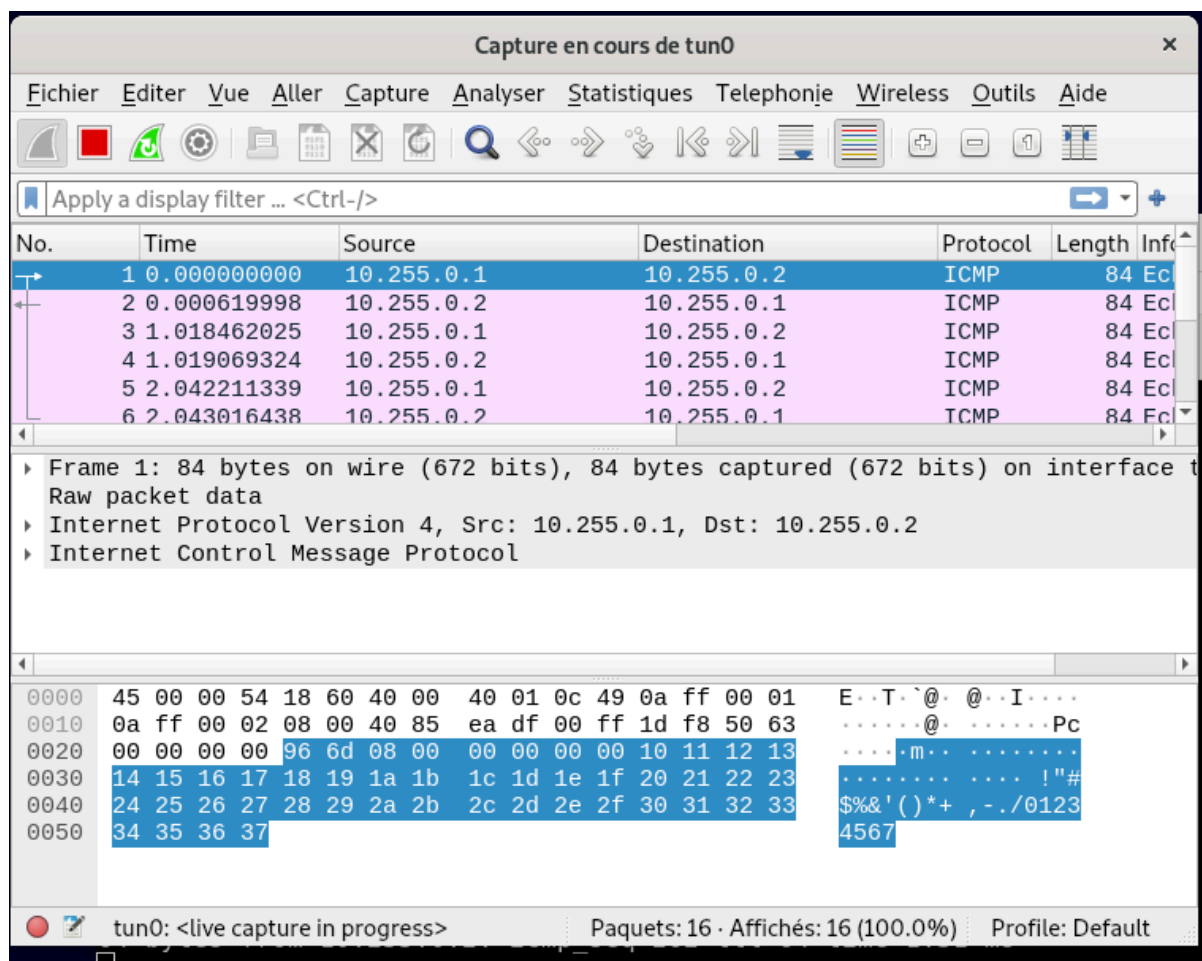
Résultat des commandes :



```
Terminal - debian@debian: ~
Fichier  Édition  Affichage  Terminal  Onglets  Aide
2022-10-20 09:09:12 us=20532 TUN/TAP device tun0 opened
2022-10-20 09:09:12 us=20553 do_ifconfig, ipv4=1, ipv6=0
2022-10-20 09:09:12 us=20579 net_iface_mtu_set: mtu 1500 for tun0
2022-10-20 09:09:12 us=20608 net_iface_up: set tun0 up
2022-10-20 09:09:12 us=21932 net_addr_ptp_v4_add: 10.255.0.1 peer 10.255.0.2 dev
tun0
2022-10-20 09:09:12 us=33970 Data Channel MTU parms [ L:1500 D:1450 EF:0 EB:386
ET:0 EL:3 ]
2022-10-20 09:09:12 us=34038 Local Options String (VER=V4): 'V4,dev-type tun,lin
k-mtu 1500,tun-mtu 1500,proto UDPv4,ifconfig 10.255.0.2 10.255.0.1'
2022-10-20 09:09:12 us=34110 Expected Remote Options String (VER=V4): 'V4,dev-ty
pe tun,link-mtu 1500,tun-mtu 1500,proto UDPv4,ifconfig 10.255.0.1 10.255.0.2'
2022-10-20 09:09:12 us=34146 Could not determine IPv4/IPv6 protocol. Using AF_IN
ET
2022-10-20 09:09:12 us=34195 Socket Buffers: R=[212992->212992] S=[212992->21299
2]
2022-10-20 09:09:12 us=34236 UDPv4 link local (bound): [AF_INET][undef]:1194
2022-10-20 09:09:12 us=34271 UDPv4 link remote: [AF_UNSPEC]
rrrR2022-10-20 09:09:24 us=738749 Peer Connection Initiated with [AF_INET]172.17
.1.22:1194
w2022-10-20 09:09:25 us=834209 WARNING: this configuration may cache passwords i
n memory -- use the auth-nocache option to prevent this
2022-10-20 09:09:25 us=834239 Initialization Sequence Completed
RwWRRWRwrWRwrWRwRwrW
```

Pendant que le tunnel est en fonction on fera une série de ping vers l'interface virtuelle (10.255.0.1 ou 10.255.0.2), depuis le eth0

Résultat sur [Wireshark](#) :



Générez une clé symétrique avec la commande:

`Openvpn --genkey --secret (ex.FichierClef)`

La clé a été créée sur la machine serveur et envoyée en SSH à la machine client.

```
root@srvVPN:/home/debian# scp /home/debian/Fichierclef root@172.17.1.22:/home/debian/
root@172.17.1.22's password:
Fichierclef 100% 636 82.7KB/s 00:00
root@srvVPN:/home/debian#
```

Puis relancez le tunnel, sur le client et le serveur, en ajoutant cette clé dans les commandes, tunnel précédentes avec l'option `--secret` (nom de la clé, dans ce cas FichierClef).

En relançant Wireshark et en faisant un ping depuis l'extérieur de la trame à l'intérieure, on constate que l'intérieure n'est plus clairement lisible grâce à l'apparition du chiffrement.

**Mais seulement le tunnel est chiffré et pas les données à l'intérieur.**

## 2.Création de certificat avec openssl

Créez des répertoires de travail où seront stockées les données du tunnel VPN :

```
mkdir -p /apps/openvpn/keys/
```

```
mkdir -p /apps/openvpn/log
```

```
mkdir -p /apps/openvpn/conf
```

```
mkdir -p /apps/pki-booktic
```

Recherchez le répertoire *easy-rsa* qui est dans */usr/share/* puis copiez le contenu de ce répertoire dans le dossier */apps/pki-booktic*

La machine serveur sera l'autorité de certification.

Modification du fichier */apps/pki-booktic/easy-rsa/vars.example*

```
set_var EASYRSA_REQ_COUNTRY    "FR"
set_var EASYRSA_REQ_PROVINCE   "ILEDEFrance"
set_var EASYRSA_REQ_CITY       "PARIS"
set_var EASYRSA_REQ_ORG        "BTSSIO"
set_var EASYRSA_REQ_EMAIL      "germain@gmail.com"
set_var EASYRSA_REQ_OU         "BTSSIO2"
```

Initialisez votre infrastructure de certification dans */apps/pki-booktic*

Avec la commande. *./easyrsa init-pki*

Dans */apps/pki\_booktic/*, utilisez le script *easyrsa* pour construire le certificat de l'autorité (celui qui validera tous les autres... pas celui du serveur).

```
./easyrsa build-ca nopass
```

```

root@srvVPN:/apps/pki-booktic/easy-rsa# ./easy-rsa build-ca nopass
bash: ./easy-rsa: Aucun fichier ou dossier de ce type
root@srvVPN:/apps/pki-booktic/easy-rsa# ./easyrsa build-ca nopass
Using SSL: openssl OpenSSL 1.1.1n 15 Mar 2022
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
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.
-----
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:CAbookticgd

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/apps/pki-booktic/easy-rsa/pki/ca.crt

```

*nopass* afin de ne pas gérer de mot de passe à chaque connexion.

Création des certificats pour le serveur et le client.

*./easyrsa gen-req srvVpn nopass*

```

root@srvVPN:/apps/pki-booktic/easy-rsa# ./easyrsa gen-req srvVpn nopass
Using SSL: openssl OpenSSL 1.1.1n 15 Mar 2022
Generating a RSA private key
...+++++
.....+++++
writing new private key to '/apps/pki-booktic/easy-rsa/pki/easy-rsa-2569.wpyc4W/
tmp.6TZo5G'
-----
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.
-----
Common Name (eg: your user, host, or server name) [srvVpn]:

Keypair and certificate request completed. Your files are:
req: /apps/pki-booktic/easy-rsa/pki/reqs/srvVpn.req
key: /apps/pki-booktic/easy-rsa/pki/private/srvVpn.key

```

*./easyrsa gen-req CltVpn nopass*

```

root@srvVPN:/apps/pki-booktic/easy-rsa# ./easyrsa gen-req CltVpn nopass
Using SSL: openssl OpenSSL 1.1.1n 15 Mar 2022
Generating a RSA private key
.....+++++
.....+++++
writing new private key to '/apps/pki-booktic/easy-rsa/pki/easy-rsa-2590.SU0WZc/
tmp.ERJush'
-----
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.
-----
Common Name (eg: your user, host, or server name) [CltVpn]:

Keypair and certificate request completed. Your files are:
req: /apps/pki-booktic/easy-rsa/pki/reqs/CltVpn.req
key: /apps/pki-booktic/easy-rsa/pki/private/CltVpn.key

```

Une fois créer les certificats pour le serveur et le client, on va devoir les signer.

*[./easyrsa sign-req server srvVpn](#)*

```

root@srvVPN:/apps/pki-booktic/easy-rsa# ./easyrsa sign-req server srvVpn
Using SSL: openssl OpenSSL 1.1.1n 15 Mar 2022

You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a server certificate for 825 days:

subject=
  commonName              = srvVpn

Type the word 'yes' to continue, or any other input to abort.
  Confirm request details: yes
Using configuration from /apps/pki-booktic/easy-rsa/pki/easy-rsa-2763.bZrkuy/tmp.ynCJc6
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName      :ASN.1 12:'srvVpn'
Certificate is to be certified until Jan 22 09:43:02 2025 GMT (825 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /apps/pki-booktic/easy-rsa/pki/issued/srvVpn.crt

```

*./easyrsa sign-req client CltVpn*

```
root@srvVPN:/apps/pki-booktic/easy-rsa# ./easyrsa sign-req client CltVpn
Using SSL: openssl OpenSSL 1.1.1n 15 Mar 2022

You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a client certificate for 825 days:

subject=
  commonName          = CltVpn

Type the word 'yes' to continue, or any other input to abort.
Confirm request details: yes
Using configuration from /apps/pki-booktic/easy-rsa/pki/easy-rsa-2847.YMXRUL/tmp.nlkyl3
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName      :ASN.1 12:'CltVpn'
Certificate is to be certified until Jan 22 09:44:13 2025 GMT (825 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /apps/pki-booktic/easy-rsa/pki/issued/CltVpn.crt
```

Création de la clé de session (Diffie-Hellman).

Commande *./easyrsa gen-dh*

Création d'une signature électronique pour authentifier la connexion client-serveur, on la placera dans le fichier */apps/openvpn/keys* avec la commande :

*Openvpn --genkey --secert /apps/openvpn/keys/bookticsign.key*

Suite à ça on va déplacer les clés créer pour le client dans la VM Client, en utilisant la commande *scp*, et les clés seront mis dans le répertoire */apps/openvpn/keys*.

```
root@srvVPN:/home/debian# scp -r /apps/openvpn/keys root@172.17.1.22:/apps/openvpn/ █
```

```
root@ClientVPN:/home/debian# cd /apps/openvpn/keys
root@ClientVPN:/apps/openvpn/keys# ls
bookticsign.key  ca.crt  ca.key  CltVpn.crt  CltVpn.key  dh.pem
root@ClientVPN:/apps/openvpn/keys# █
```

## II- Mise en place du tunnel openvpn

### 1. Configuration serveur

Afin de démarrer une instance de serveur VPN, *openvpn* vient chercher dans */etc/openvpn*

Un fichier (.conf), qui lui permettra de savoir avec quelles options lancer le tunnel. Nous allons créer un lien pour le fichier .conf afin qu'il aille le chercher dans le répertoire.

On va créer un fichier conf dans */apps/openvpn/conffiles/* appelé *bookticVPN.conf*.

Puis on crée un lien symbolique (comme un raccourci) depuis */etc/openvpn* vers le nouveau fichier *bookticVPN*. Le service *openvpn* cherche les fichiers conf dans */etc/openvpn/* donc notre lien permettra d'aller chercher le nôtre à son emplacement réel :

*ln -s /apps/openvpn/conffiles/bookticVPN.conf /etc/openvpn/bookticVPN.conf*

# pour voir le résultat : *ln -l /etc/openvpn*

Fichier *bookticVPN.conf*.

```
GNU nano 5.4 /etc/openvpn/bookticVPN.conf
proto udp
dev tun
ca /apps/openvpn/keys/ca.crt
cert /apps/openvpn/keys/srvVpn.crt
key /apps/openvpn/keys/srvVpn.key
dh /apps/openvpn/keys/dh.pem
tls-auth /apps/openvpn/keys/bookticsign.key 0

server 10.255.0.0 255.255.255.0
client-to-client
explicit-exit-notify 1
keepalive 10 120
persist-key
persist-tun
cipher AES-256-CBC
#compress lz4-v2
status openvpn-status.log
log /apps/openvpn/log/openvpn.log
log-append /apps/openvpn/log/openvpn.log
verb 5
```



## 2. Configuration client

Création du fichier de configuration VPN du client, on a placé le fichier dans le répertoire :

[/apps/openvpn/conffiles/bookticVPN.conf](#)

```
client
dev tun
proto udp
remote 172.16.19.65 1196

resolv-retry infinite
nobind

persist-key
persist-tun
mute-replay-warnings

ca /apps/openvpn/keys/ca.crt
cert /apps/openvpn/keys/CltVpn.crt
key /apps/openvpn/keys/CltVpn.key
tls-auth /apps/openvpn/keys/bookticsign.key 1

log /var/log/openvpn/openvpn.log
cipher AES-256-CBC
verb 5
```

Maintenant il faut redémarrer le service avec la commande: [systemctl restart openvpn](#)

De chaque côté vous devez vous obtenir (Initialization Sequence Complete) en faisant un ping L'adresse virtuelle VPN du serveur

## III Mise en route du routage sur le serveur et redirection sur le pare-feu

### 1.Préparation

Sur la machine client se mettre en DHCP sur le VLAN WAN, testez la connexion.

## 2.Routage du serveur VPN

Nous allons faire en sorte que le serveur VPN prenne en charge le routage et masque les différentes IP venant du Tunnel.

Dans le fichier `/etc/sysctl.conf` modifier cette ligne:

```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

Qui permettra au serveur VPN de faire le pont entre l'interface virtuelle et réel.

Redémarrez le service : `systemctl restart procps`

NAT dynamique :

Masquez toutes les IPs sortant du tunnel vers le réseau

```
root@srvVPN:/home/debian# sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
root@srvVPN:/home/debian#
```

Dans le fichier de configuration du serveur on ajoute la ligne suivante :

```
push "route 172.17.0.0 255.255.0.0"
```

On peut ajouter autant de push route que de vlan.

Modifié l'adresse du remote dans le fichier .conf du client, avec l'IP WAN du Pfsense  
172.16.19.65 .

Maintenant on doit redémarrer le service `openvpn` avec la commande `openvpn@bookticVPN`  
puis on vérifie les logs du client avec la commande `tail -f /var/log/openvpn/openvpn.log` et  
normalement il doit y avoir marqué (initialization Sequence Completed).

```
2023-03-23 12:01:06 us=1209 OPTIONS IMPORT: timers and/or timeouts modified
2023-03-23 12:01:06 us=1217 OPTIONS IMPORT: --ifconfig/up options modified
2023-03-23 12:01:06 us=1222 OPTIONS IMPORT: route options modified
2023-03-23 12:01:06 us=1227 OPTIONS IMPORT: peer-id set
2023-03-23 12:01:06 us=1232 OPTIONS IMPORT: adjusting link_mtu to 1624
2023-03-23 12:01:06 us=1237 OPTIONS IMPORT: data channel crypto options modified
2023-03-23 12:01:06 us=1243 Data Channel: using negotiated cipher 'AES-256-GCM'
2023-03-23 12:01:06 us=1254 Data Channel MTU parms [ L:1552 D:1450 EF:52 EB:406 ET:0 EL:3 ]
2023-03-23 12:01:06 us=1300 Outgoing Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
2023-03-23 12:01:06 us=1307 Incoming Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
2023-03-23 12:01:06 us=1314 Preserving previous TUN/TAP instance: tun0
2023-03-23 12:01:06 us=1367 Initialization Sequence Completed
WRwRwWRwRw
```