Table des matières

1) Prépapration

1) Configuration de la machine Open VPN :

- IP = 10.214.3.2

- Masque = 255.255.0.0

- Route = 10.214.255.254

Configuration du client interne :

- IP = 192.168.1.1

- Masque = 255.255.255.0

- Route = 192.168.1.254

Configuration du Firewall :

- IP = 10.214.4.2

- Masque = 255.255.0.0

- Route = 10.214.255.254

- IP = 192.168.1.254

- Masque = 255.255.255.0

-Iptables = iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -j SNAT --to-source 10.214.4.2

2) Pour installer le paquet openvpn j'utilise la commande suivante :

```
root@debian:~/lzo-1.08# apt-get install openvpn
```

Pour installer la librairie lzo j'utilise les commandes suivantes :

```
root@debian:~/lzo-1.08# wget
http://www.oberhumer.com/opensource/lzo/download/lzo-1.08.tar.gz
```

```
root@debian:~/lzo-1.08# tar -zxvf lzo-1.08.tar.gz
```

```
root@debian:~/lzo-1.08# cd lzo-1.08
```

```
root@debian:~/lzo-1.08# apt install make
```

```
root@debian:~/lzo-1.08# ./configure && make && make check && make test
```

2. Utilisation de SSL/TLS

2.1 Qu'est-ce que SSL/TLS

TLS est le successeur de SSL. Ce sont des protocoles de sécurisation des échanges sur internet.

Ils fonctionnent avec un système de client-serveur. Il permet d'avoir :

- Un système d'authentification du serveur

- Session chiffrée

- Un système d'authentification du client

2.2 X509

X509 est une norme gérant les certificats à clé publique

Il crée un format de certificats électroniques

2.3 On construit d'abord l'autorité de certification (CA)

Je lance la commande make-cadir

Il faut que le dossier qu'on appelle CA (dans l'exemple) n'existe pas

```
root@debian:~# make-cadir /root/travail/CA
```

Le dossier de travail est bien créé

Je cat le fichier vars

Je remarque que ce fichier sert à la configuration de easy-rsa 3

```
root@debian:~/travail/CA# cat vars
# Easy-RSA 3 parameter settings
```

On y voit beaucoup de paramètres

La commande ./easyrsa init-pki sert à lancer pki

La commande .easyrsa build-ca sert à créer le CA. On y rentre plusieurs paramètres comme la passphrase ou le nom du CA

Le dossier pki contient tous les fichiers de configurations de easyrsa pour le CA

2.4 Qu'est-ce qu'un échange de Diffie-Hellman

Avec la commande ./easyrsa gen-dh

On génère les paramètres de Diffie-Hellman, avec une clé chiffrée de 2048 bits pour le SSL

On voit dans le dossier pki qu'un fichier dh.pem a été créé

```
root@debian:~/travail/CA# ls pki
ca.crt         index.txt        private  revoked
certs_by_serial  issued             renewed  safessl-easyrsa.cnf
dh.pem           openssl-easyrsa.cnf  reqs     serial
```

2.5 Construisez le certificat et la clé privée du serveur

En faisant la commande ./easyrsa build-server-full firewall

On voit que certains fichier on été remplacés (en rouge) et d'autres créés (en bleu)

```
ca.crt         index.txt    openssl-easyrsa.cnf  revoked
certs_by_serial  index.txt.attr  private          safessl-easy
dh.pem           index.txt.old       renewed          serial
extensions.temp  issued              reqs             serial.old
```

On a aussi créé une autre clé pour le serveur, qui se trouve dans pki/private

```
root@debian:~/travail/CA# ls pki/private/
```

```
ca.key        firewall.key
```

On a aussi un certificat dans le dossier issued

On a aussi un fichier .req crée dans le dossier reqs

2.6 Construisez la clé du client OpenVPN

On fait : ./easyrsa build-client-full client

On nouveau fichier index.txt.attr est créé (mais il contient les mêmes valeurs)

On a une clé et un certificat qui sont crées, dans issued et private

```
root@debian:~/travail/CA/pki# ls issued/
client.crt  firewall.crt
```

On a aussi un fichier .req crée dans le dossier reqs, soit les mêmes choses que pour le serveur

Je scp la clé et le certificat

2.7 Configurez OpenVPN pour utiliser des certificats SSL

Je copie le fichier server.conf.gz dans le dossier openvpn

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

et de même pour le client

```
root@debian:/etc/openvpn# cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf /etc/openvpn/
```

Je cat le fichier. Je vosi qu'il y a 2 façons de faire du VPN. Je peux le router ou faire un tunnel ethernet. Je vois aussi l'option pour les fichiers de certificats et la clé

```
# "dev tun" will create a routed IP tunnel,
# "dev tap" will create an ethernet tunnel.
```

```
# 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 server.crt
```

```
key server.key  # This file should be kept secret
```

Je reste en tunnel routé

Je met la location des fichiers de certifications

```
# Any X509 key management system can be used.

ca /root/travail/CA/pki/ca.crt
cert /root/travail/CA/pki/issued/server.crt
key /root/travail/CA/pki/private/server.key  # This file should be kept secret
```

Je met la range IP que le serveur VPN va attribuer au clients

```
server 192.168.0.0 255.255.255.0
```

Je met une route pour accéder au réseau interne

```
push "route 192.168.1.0 255.255.255.0"
```

Je décommente 2 lignes pour que tout le monde puisse se connecter

```
# non-Windows systems.
user nobody
group nogroup
```

Je lance le serveur avec la commande :

```
root@debian:/etc/openvpn# openvpn --config /etc/openvpn/server.conf –
askpass
```

Il me demande le pass de clé clé, je le rentre et openvpn se lance

Je regarde dans le fichier log pour confirmer grâce à l'option rentrée dans le fichier de configuration :

log openvpn.log

j'ouvre ce fichier :

```
Sun Oct 20 13:50:53 2019 us=92793 Initialization Sequence Completed
```

Je peux voir aussi en faisant ip a s que la carte réseau tun0 est bien crée

```
root@debian:/etc/openvpn# ip a s
```

```
7: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast state UNKNOWN group default qlen 100
    link/none
    inet 192.168.10.1 peer 192.168.10.2/32 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::54b3:7191:5c38:8aec/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
```

3. Pontage

3.1 Serveur Configuration

Je crée alors le fichier de conf pour le client :

```
client
dev tun
proto udp
remote 10.1.1.254 1194
#Tentative de connexion infinie
resolv-retry infinite
nobind
#Pour rendre la connexion persistante
persist-key
persist-tun
ca /root/clé/ca.crt
cert /root/clé/client.crt
key /root/clé/client.key
cipher AES-128-CBC
```

Je lance openvpn avec la commande :

```
openvpn client.conf
```

Il me demande ma passphrase

La connexion avec le serveur se fait et fonctionne

```
Sun Oct 20 13:58:51 2019 us=594331 Control Channel: TLSv1.3, cipher
TLSv1.3 TLS_AES_256_GCM_SHA384, 2048 bit RSA
....
Sun Oct 20 13:51:05 2019 us=840465 Initialization Sequence Completed
```

On voit que c’est encrypté en TLS

Je fais un ip a s pour voir si ma carte réseau s’est bien montée

```
root@debian:/etc/openvpn# ip a s

5: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast state UNKNOWN group default qlen 100
    link/none
    inet 192.168.10.6 peer 192.168.10.5/32 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::fa43:c7b3:4f3b:1b95/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
```

Le serveur se trouve en 192.168.10.1.

Je tente de le ping

Le ping fonctionne :

```
root@debian:/etc/openvpn# ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_seq=1 ttl=64 time=0.424 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=64 time=0.473 ms
```

Je tente maintenant de ping l'adresse du client interne (192.168.1.1)

Cela fonctionne également

```
root@debian:/etc/openvpn# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=63 time=0.597 ms
```

Je peux le ping grâce a l'option push "route 192.168.1.0 255.255.255.0" mise dans le fichier de configurations du serveur pour qu'on puisse accéder au réseau 192.168.1.0

On peut également ping les adresses du client OpenVPN ou du serveur via le client du réseau interne

Pour pouvoir le copier/coller, je ssh la machine du réseau interne via le client openVPN ou le serveur

```
root@debian:~# ssh 192.168.1.1
root@debian:~# ip a s
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP group default qlen 1000
    link/ether 08:00:27:4d:80:cb brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.1/24 scope global enp0s3
       valid_lft forever preferred_lft forever
```

Je ping le client OpenVPN (192.168.10.6)

```
PING 192.168.10.6 (192.168.10.6) 56(84) bytes of data.
64 bytes from 192.168.10.6: icmp_seq=1 ttl=63 time=0.646 ms
64 bytes from 192.168.10.6: icmp_seq=2 ttl=63 time=0.748 ms
```

Le ping du serveur fonctionne également

```
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_seq=1 ttl=64 time=0.103 ms
```

Je peux toujours ping le réseau extérieur avec le client interne

```
root@debian:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=53 time=19.10 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=53 time=19.9 ms
```