INSTALLATION D'UN SERVEUR OPENVPN SOUS RASPBERRY PI

RaspberryPi - Debian Buster
Configuration de base

Tutoriel OPENVPN - RASPBERRY

David GOÏTRÉ

Table des matières

Introduction

Un réseau privé virtuel (VPN) est un protocole utilisé pour ajouter la sécurité et la confidentialité aux réseaux privés et publics. Les VPN envoient du trafic entre deux ou plusieurs appareils sur un réseau dans un tunnel chiffré. Une fois la connexion VPN établie, tout le trafic réseau est chiffré du côté du client. Les VPN masquent votre adresse IP de sorte que nos actions en ligne sont pratiquement introuvables.

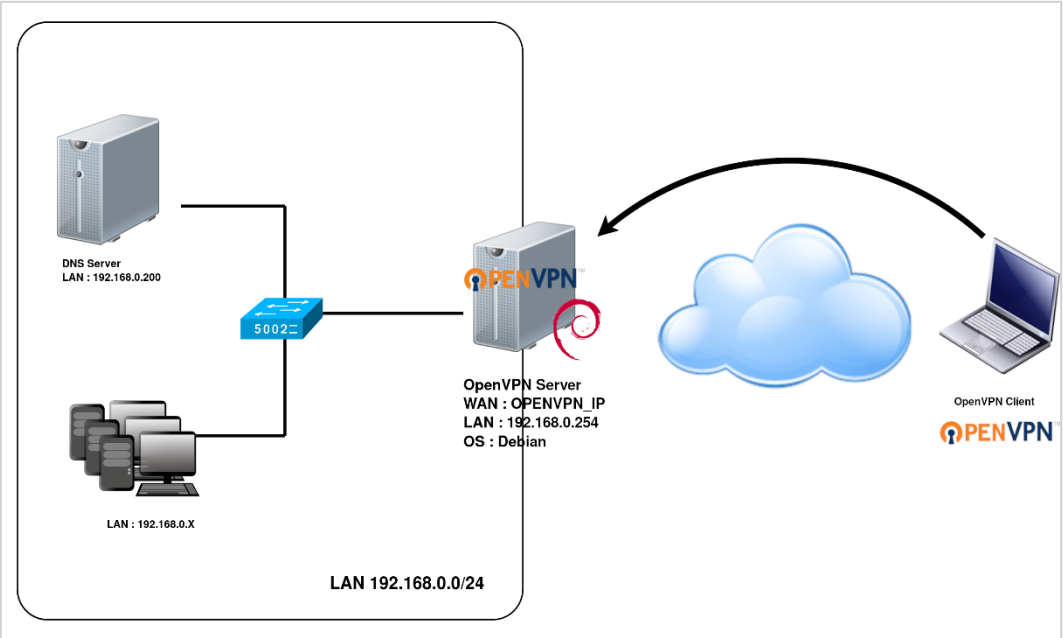
Il fournit le cryptage et l'anonymat, protège nos activités en ligne, nos achats en ligne, l'envoi d'e-mails et aide également à garder notre navigation Web anonyme.

1. Pré requis

On a besoin des différents matériels et logiciels pour la création d'un Serveur VPN avec un Raspberry.

- Un ou des PC client sous Windows
- Une Box (Free, Orange ou Sfr)
- Un Raspberry 3B+ avec l'OS Raspian Buster installé avec Etcher
- Le logiciel OpenVPN pour les clients
- Le logiciel Putty pour se connecter en SSH au serveur VPN
- Connaitre l'interface réseau (eth0, br0, ens3…) via la commande : **ip a**
  Pour notre test c'est **l'interface eth0** qui est utilisée

Voici le schéma que l'on doit obtenir une fois le serveur VPN mise en place :



Ce schéma n'est qu'un exemple. Il n'est pas essentiel de posséder une machine Serveur DNS, ni d'avoir plusieurs PC Client sur le réseau LAN.

2. Paramétrage du serveur

Avant d'aller plus loin, il nous faut connaître l'interface réseau de notre serveur **RaspberryPI** et lui attribuer une adresse IP fixe.

a) Lister les interfaces

```
$ ip link | awk '{ print $2}'   # liste les interfaces
# ethtool <interface> | grep detected   # détecte l'interface connectée
```

b) Définir une adresse IP fixe

```
# nano /etc/network/interfaces   # ouvre le fichier des interfaces
```

c) Copier le texte ci-dessous dans le fichier **interfaces**

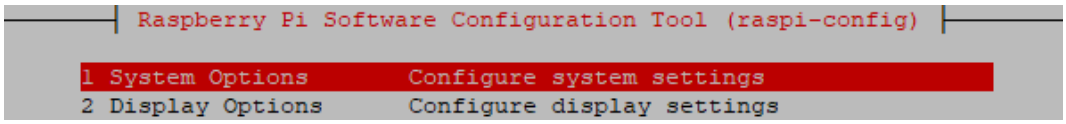
```
# Interface reseau de bouclage
auto lo
iface lo inet loopback
# Interface reseau principale
allow-hotplug eth0
iface eth0 inet static
address 192.xxx.xxx.xxx
netmask 255.255.255.0
gateway 192.xxx.xxx.xxx
```

d) Rebooter le serveur

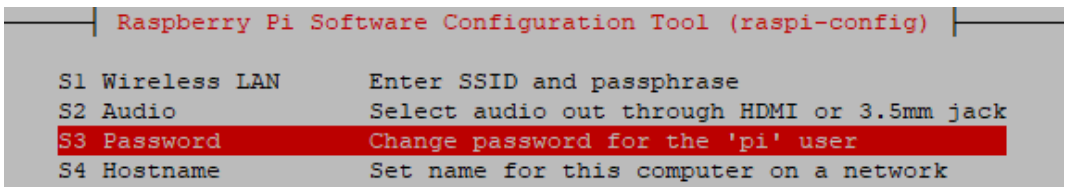
```
# /etc/init.d/networking restart
# reboot
```

e) Paramétrer le serveur

```
$ raspi-config   # ouvre l'utilitaire, sélectionner le menu System Options
```



Sélectionner le menu **S3 Password** pour modifier le mot de passe et **S4 Hostname** pour modifier le nom du serveur.



3. Paramétrage de connexion au serveur

a) Créer **une redirection de port** sur la box (Free, Orange…) vers votre serveur Raspberry.

- **port** : 1194
- **Protocole** : UDP

b) Activer le **SSH** sur le serveur. Pour ce faire, ouvrir le dossier **Boot**, de la carte SD du Raspberry via l'explorateur de Windows et créer un fichier **ssh** (sans extension) dans ce **dossier**.

c) Ouvrir **Putty** et se connecter au serveur VPN avec les identifiants (par défaut **pi/raspberry**)

b) Mettre à jour les packages du système vers la dernière version. Exécuter la commande suivante pour mettre à jour et mettre à niveau les packages de votre système :

```
# apt-get update -y
# apt-get upgrade -y
```

4. Activer le transfert IP

Certains aspects de la configuration réseau du serveur doivent être modifiés afin qu'OpenVPN puisse acheminer correctement le trafic à travers le VPN. Le premier d'entre eux est le transfert IP, une méthode permettant de déterminer où le trafic IP doit être acheminé. Ceci est essentiel pour la fonctionnalité VPN que notre serveur fournira. Editer le fichier **sysctl.conf** :

```
# nano /etc/sysctl.conf
```

Décommenter la ligne suivante :

```
net.ipv4.ip_forward = 1
```

Enregistrer le fichier lorsque l'on a terminé. Ensuite, exécuter la commande suivante pour appliquer les modifications :

```
# sysctl -p
```

5. Installer le serveur OpenVPN

Par défaut, Les paquets OpenVPN, OPenSSL et Easyrsa ne sont pas disponibles dans le référentiel par défaut sur Debian 10. Il faut les installés avec la commande suivante :

```
# apt-get install openvpn openssl easy-rsa
```

Une fois l'installation terminée, on doit également copier le répertoire **easy-rsa** pour la gestion des **certificats SSL**. Exécuter la commande suivante pour copier le répertoire **easy-rsa** du répertoire **/usr/share** vers le répertoire **/etc/openvpn**.

```
# cp -r /usr/share/easy-rsa  /etc/openvpn/
```

6. Configurer l'autorité de certification

Easy RSA utilise un ensemble de scripts pour générer des clés et des certificats. Tout d'abord, on doit configurer l'autorité de certification sur votre système. Pour ce faire, changer le répertoire en **/etc/openvpn/easy-rsa** et créer un nouveau fichier de configuration d'Easy RSA :

```
$ cd/etc/openvpn/easy-rsa
$ nano vars
```

Ajouter les lignes suivantes, y compris le pays, la ville et l'adresse e-mail préférée :

```
set_var EASYRSA                 "$PWD"
set_var EASYRSA_PKI             "$EASYRSA /pki"
set_var EASYRSA_DN              "cn_only"
set_var EASYRSA_REQ_COUNTRY     "FRANCE"
set_var EASYRSA_REQ_PROVINCE    "FR"
set_var EASYRSA_REQ_CITY        "Paris"
set_var EASYRSA_REQ_ORG         "Tecadmin CERTIFICATE AUTHORITY"
set_var EASYRSA_REQ_EMAIL       "admin@tecadmin.net"
set_var EASYRSA_REQ_OU          "Tecadmin EASY CA"
set_var EASYRSA_KEY_SIZE        2048
set_var EASYRSA_ALGO            rsa
set_var EASYRSA_CA_EXPIRE       7500
set_var EASYRSA_CERT_EXPIRE     365
set_var EASYRSA_NS_SUPPORT      "no"
set_var EASYRSA_NS_COMMENT      "Tecadmin CERTIFICATE AUTHORITY"
set_var EASYRSA_EXT_DIR         "$EASYRSA /x509-types"
set_var EASYRSA_SSL_CONF        "$EASYRSA /openssl-easyrsa.cnf"
set_var EASYRSA_DIGEST          "sha256"
```

Enregistrer le fichier lorsque l'on a terminé. Ensuite, exécuter la commande suivante pour initialiser le PKI.

```
$ ./easyrsa init-pki
```

On doit obtenir la sortie suivante (utilisation de la configuration Easy-RSA à partir du fichier : **./vars**) :

```
init-pki complete; vous pouvez maintenant créer une autorité de certification ou des
demandes.
Votre répertoire PKI nouvellement créé est: /etc/openvpn/easy-rsa/pki
```

Ensuite, créer les certificats CA avec la commande suivante :

```
$ ./easyrsa build-ca
```

On doit obtenir la sortie suivante :

```
Remarque: utilisation de la configuration Easy-RSA à partir de: ./vars
Utilisation de SSL: openssl OpenSSL 1.1.1c FIPS 28 mai 2019
Entrez la nouvelle phrase de passe de clé CA:
Entrez à nouveau la nouvelle phrase de passe de clé CA:
Génération de clé privée RSA, module de 2048 bits (2 nombres premiers)
.....+++++
.....+++++
e est 65537 (0x010001)
Impossible de charger /etc/openvpn/easy-rsa/pki/.rnd dans RNG
140218549745472: erreur: 2406F079: générateur de nombres aléatoires: RAND_load_file:
Impossible d'ouvrir le fichier: crypto / rand / randfile.c: 98: Nom de fichier=/etc/openvpn/easy-rsa/pki/.rnd
Vous êtes sur le point d'être invité à saisir des informations qui seront intégrées dans votre demande de certificat.
Ce que vous êtes sur le point d'entrer, c'est ce qu'on appelle un nom distinctif ou un DN.
Il y a pas mal de champs mais vous pouvez laisser certains vides
Pour certains champs, il y aura une valeur par défaut,
Si vous entrez «.», Le champ sera laissé vide.
-----
Common Name (par exemple: votre nom d'utilisateur, d'hôte ou de serveur) [Easy-RSA CA]:
La création de l'autorité de certification est terminée et vous pouvez maintenant importer et signer des demandes de certificat.
Votre nouveau fichier de certificat CA pour la publication se trouve à l'adresse suivante:
/etc/openvpn/easy-rsa/pki/ca.crt
```

La commande ci-dessus générera deux fichiers nommés **ca.key** et **ca.crt**. Ces certificats seront utilisés pour signer les certificats de notre serveur et de nos clients.

Attention si on saisit une valeur dans le champ **Common Name**, il faut aussi modifier le paramètre **nomserver** des commandes **gen-req** et **sign-req** par la valeur saisie dans la commande **build-ca.**

Le paramètre **organisation** peut être remplacé par la valeur du champ **EASYRSA_REQ_ORG** du fichier **vars**.

7. Générer les fichiers de certificat de serveur

Ensuite, on doit générer une demande de pair de clés et de certificat pour notre serveur. Exécuter la commande suivante pour générer la clé serveur nommée **organisation-nomserver** :

```
$ ./easyrsa gen-req organisation-nomserver nopass
```

On doit obtenir la sortie suivante :

```
Remarque: utilisation de la configuration Easy-RSA à partir de: ./vars
Utilisation de SSL: openssl OpenSSL 1.1.1c FIPS 28 mai 2019
Générer une clé privée RSA
..... +++++
..... ..... .....
..... ..... ..... +++++
Writing new private key to '/etc/openvpn/easy-rsa/pki/private/organisation-
server.key.kOlBTwtY6a'
-----
Vous êtes sur le point d'être invité à saisir des informations qui seront intégrées
dans votre demande de certificat.
Ce que vous êtes sur le point d'entrer, c'est ce qu'on appelle un nom distinctif ou un DN.
Il y a pas mal de champs mais vous pouvez laisser certains vides
Pour certains champs, il y aura une valeur par défaut,
Si vous entrez «.», Le champ sera laissé vide.
-----
Common Name (ex: votre nom d'utilisateur, d'hôte ou de serveur) [organisation-nomserver]:
La demande de pair de clés et de certificat est terminée. Vos fichiers sont:
req: /etc/openvpn/easy-rsa/pki/reqs/organisation-nomserver.req
key: /etc/openvpn/easy-rsa/pki/private/organisation-nomserver.key
```

Attention laisser le champ **Common Name** vide**.**

8. Signer la clé du serveur à l'aide de CA

Ensuite, vous devrez signer la clé organisation-server à l'aide du certificat CA. Exécuter la commande suivante pour signer la clé du serveur :

```
$ ./easyrsa sign-req server organisation-nomserver
```

On doit obtenir la sortie suivante :

```
Remarque: utilisation de la configuration Easy-RSA à partir de: ./vars
Utilisation de SSL: openssl OpenSSL 1.1.1c FIPS 28 mai 2019
Vous êtes sur le point de signer le certificat suivant.
Veuillez vérifier l'exactitude des détails ci-dessous. Notez que cette demande
n'a pas été vérifié cryptographiquement. Veuillez vous assurer qu'il provient d'un
source ou que vous avez vérifié la somme de contrôle de la demande auprès de l'expéditeur.
Objet de la demande, à signer en tant que certificat de serveur pendant 365 jours:
sujet =
commonName = organisation-nomserver
Tapez le mot «yes» pour continuer ou toute autre entrée pour abandonner.
Confirmer les détails de la demande: yes
Utilisation de la configuration de /etc/openvpn/easy-rsa/pki/safessl-easyrsa.cnf
Entrez la phrase de passe pour /etc/openvpn/easy-rsa/pki/private/ca.key:
Vérifiez que la demande correspond à la signature
Signature ok
Le nom distinctif du sujet est le suivant
commonName: ASN.1 12: 'organisation-nomserver'
Le certificat doit être certifié jusqu'au 16 février 05:00:50 2021 GMT (365 jours)
Écrire la base de données avec 1 nouvelles entrées
Base de données mise à jour
Certificat créé à: /etc/openvpn/easy-rsa/pki/issued/organisation-nomserver.crt
```

Ensuite, vérifier le fichier de certificat généré avec la commande suivante :

```
$ openssl verify -CAfile pki/ca.crt pki/issued/organisation-nomserver.crt
```

On doit obtenir la sortie suivante :

```
pki/issued/organisation-nomserver.crt: OK
```

Ensuite, exécuter la commande suivante pour générer une **clé Diffie-Hellman** forte à utiliser pour l'échange de clés :

```
$ ./easyrsa gen-dh
```

On doit obtenir la sortie suivante :

```
Remarque: utilisation de la configuration Easy-RSA à partir de: ./vars
Utilisation de SSL: openssl OpenSSL 1.1.1c FIPS 28 mai 2019
Génération de paramètres DH, amorce sûre de 2048 bits de long, générateur 2
Cela va prendre du temps
.... + ..... + ..... ..... + ..... ++ * ++ * ++ * ++ *
Paramètres DH de taille 2048 créés dans /etc/openvpn/easy-rsa/pki/dh.pem
```

Après avoir créé tous les fichiers de certificat, les copier dans le dossier **/etc/openvpn/server/**

```
# cp pki/ca.crt  /etc/openvpn/server/
# cp pki/dh.pem  /etc/openvpn/server/
# cp pki/private/organisation-nomserver.key  /etc/openvpn/server/
# cp pki/issued/organisation-nomserver.crt  /etc/openvpn/server/
```

9. Générer un certificat client et le fichier de clé

Ensuite, On doit générer la clé et le fichier de certificat pour le client. Tout d'abord, exécuter la commande suivante pour créer le fichier de clé client :

```
$ ./easyrsa gen-req client1 nopass
```

On doit obtenir la sortie suivante :

```
Remarque: utilisation de la configuration Easy-RSA à partir de: ./vars
Utilisation de SSL: openssl OpenSSL 1.1.1c FIPS 28 mai 2019
Générer une clé privée RSA
..... .... +++++
... +++++
Writing new private key to '/etc/openvpn/easyrsa/pki/private/client.key.e38GUtzHie'
-----
Vous êtes sur le point d'être invité à saisir des informations qui seront intégrées
dans votre demande de certificat.
Ce que vous êtes sur le point d'entrer, c'est ce qu'on appelle un nom distinctif ou un DN.
Il y a pas mal de champs mais vous pouvez laisser certains vides
Pour certains champs, il y aura une valeur par défaut,
Si vous entrez «.», Le champ sera laissé vide.
-----
Nom commun (par exemple: votre nom d'utilisateur, d'hôte ou de serveur) [client]:
La demande de paire de clés et de certificat est terminée. Vos fichiers sont:
req: /etc/openvpn/easy-rsa/pki/reqs/client1.req
key: /etc/openvpn/easy-rsa/pki/private/client1.key
```

Ensuite, signer la clé client à l'aide de votre certificat CA (**passphrase du serveur**) :

```
$ ./easyrsa sign-req client client1
```

On doit obtenir la sortie suivante :

```
Remarque: utilisation de la configuration Easy-RSA à partir de: ./vars
Utilisation de SSL: openssl OpenSSL 1.1.1c FIPS 28 mai 2019
Vous êtes sur le point de signer le certificat suivant.
Veuillez vérifier l'exactitude des détails ci-dessous. Notez que cette demande
n'a pas été vérifié cryptographiquement. Veuillez vous assurer qu'il provient d'une source ou que
vous avez vérifié la somme de contrôle de la demande auprès de l'expéditeur.
Objet de la demande, à signer en tant que certificat client pendant 365 jours:
sujet =
commonName = client1
Tapez le mot «yes» pour continuer ou toute autre entrée pour abandonner.
Confirmer les détails de la demande: yes
Utilisation de la configuration de /etc/openvpn/easy-rsa/pki/safessl-easyrsa.cnf
Entrez la phrase de passe pour /etc/openvpn/easy-rsa/pki/private/ca.key:
Vérifiez que la demande correspond à la signature
Signature ok
Le nom distinctif du sujet est le suivant
commonName: ASN.1 12: 'client'
Le certificat doit être certifié jusqu'au 16 février 05:11:19 2021 GMT (365 jours)
Écrire la base de données avec 1 nouvelles entrées
Base de données mise à jour
Certificat créé à: /etc/openvpn/easy-rsa/pki/issued/client1.crt
```

Ensuite, copier tout le certificat client et le fichier de clé dans le répertoire **/etc/openvpn/client/** :

```
# cp pki/ca.crt   /etc/openvpn/client/
# cp pki/issued/client1.crt   /etc/openvpn/client/
# cp pki/private/client1.key   /etc/openvpn/client/
```

10. Configurer le serveur OpenVPN

Ensuite, créer un nouveau fichier de configuration OpenVPN dans le dossier **/etc/openvpn/** :

```
# nano /etc/openvpn/server.conf
```

Ajouter les lignes suivantes :

```
#Configuration server
Port 1194
proto udp
dev tun
#Cles certificats
ca /etc/openvpn/server/ca.crt
cert /etc/openvpn/server/tecadmin-server.crt
key /etc/openvpn/server/tecadmin-server.key
dh /etc/openvpn/server/dh.pem
key-direction 0
data-ciphers-fallback AES-256-CBC
tls-server
tls-version-min 1.2
tls-crypt /etc/openvpn/server/tecadmin-server.key
auth SHA512
auth-nocache
#Reseau
server 10.8.0.0 255.255.255.0
push "redirect-gateway def1 bypass-dhcp"
push "DNS option dhcp 208.67.222.222"
push "DNS option dhcp 208.67.220.220"
keepalive 20 60
duplicate-cn
#Securite
persist-key
persist-tun
compress lz4
deamon
user nobody
group nogroup
#Logs
status openvpn-status.log
log-append /var/log/openvpn.log
verb 3
```

Enregistrer le fichier lorsque l'on a terminé.

11. Démarrer le service OpenVPN

OpenVPN est maintenant installé et configuré. On peut maintenant démarrer le service OpenVPN et l'activer après le redémarrage du système à l'aide de la commande suivante:

```
$ systemctl start openvpn@server
$ systemctl enable openvpn@server
```

Exécuter la commande suivante pour vérifier l'état du service OpenVPN :

```
$ systemctl status openvpn@server
```

On doit obtenir la sortie suivante :

```
● openvpn@server.service - Connexion OpenVPN au serveur
Loaded: loaded (/lib/systemd/system/openvpn@.service; enableb; préréglage du fournisseur:
enable)
Active: active (en cours d'exécution) depuis ven 2020-02-21 15:38:31 UTC; Il y a 4s
  Documents: man: openvpn (8)
      https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
      https://community.openvpn.net/openvpn/wiki/HOWTO
PID principal: 3044 (openvpn)
Statut: "Séquence d'initialisation terminée"
 Tâches: 1 (limite: 2359)
Mémoire: 1,3 M
CGroup: /system.slice/system-openvpn.slice/openvpn@server.service
      └─3044 /usr/sbin/openvpn --daemon ovpn-server --status /run/openvpn/server.status  10 --
cd /etc/openvpn --config /etc/openvpn/server.
21 février 15:38:31 debian10 systemd [1]: Démarrage de la connexion OpenVPN au serveur ...
21 février 15:38:31 debian10 systemd [1]: Démarrage de la connexion OpenVPN au serveur.
```

Une fois le service OpenVPN démarré avec succès, il créera une nouvelle interface réseau nommée tun. Vous pouvez le vérifier avec la commande suivante:

```
$ ip a show tun
```

On doit obtenir la nouvelle interface **tun** dans la sortie suivante :

```
59:  tun:  mtu 1500 qdisc pfifo_fast state UNKNOWN group default qlen 100
Link/none
inet 10.8.0.1 peer 10.8.0.2/32 scope global tun
   valid_lft forever preferred_lft forever
inet6 fe80 :: 90: e3c0: 5f1a: 27f5 / 64 scope link stable-privacy
   valid_lft forever preferred_lft forever
```

12. Générer la configuration du client

Ensuite, créer un nouveau fichier de configuration client OpenVPN nommé **client1.ovpn**. On aura besoin de ce fichier pour connecter le serveur OpenVPN à partir du système client.

```
# nano /etc/openvpn/client/client.ovpn
```

Ajouter les lignes suivantes (remplacer **xxx.xxx.xxx.xxx** par **l'ip** de la Box) :

```
#Client
client
dev tun
proto udp
remote xxx.xxx.xxx.xxx 1194
resolv-retry infinite
#Cles
ca ca.crt
cert client1.crt
key client1.key
#Securite
data-ciphers-fallback AES-256-CGM
auth SHA512
auth-nocache
tls-version-min 1.2
tls-client
nobind
persist-key
persist-tun
key-direction 1
mute-replay-warnings
verb 3
```

Enregistrer le fichier lorsque l'on a terminé.

13. Connecter le client Windows au VPN

Il faut donner les droits à la copie, et transférer les fichiers de configuration sur le PC Client à l'aide du logiciel **FileZilla**. Lancer les commandes suivantes dans putty puis passer à la suite.

```
$ cd /etc/openvpn/client
$ mkdir ../aeff
$ cp *.* ../aeff
$ chmod -R 777 ../aeff
```

a) Se connecter au serveur via **FileZilla** avec les mêmes identifiants utilisés dans Putty.
b) Ouvrir le dossier et récupérer les fichiers du **dossier aeff**
c) Copier les fichiers dans le dossier **openvpn/config**
d) Ouvrir **OpenVPN** et se connecter
e) Ou avec **OpenVPN Connect**, importer le fichier de configuration **client1.ovpn** et se connecter

14. Connecter le client Linux au VPN

Tout d'abord, se connecter à l'ordinateur client et installer le package OpenVPN avec la commande suivante :

```
# apt-get install openvpn -y
```

Ensuite, on doit télécharger les fichiers de configuration du client OpenVPN du serveur OpenVPN vers la machine cliente.
Sur l'ordinateur client, exécuter la commande suivante pour télécharger tout le fichier de configuration client :

```
$ scp -r root @ vpn-server-ip:/etc/openvpn/client.
```

Une fois téléchargé, ouvrir le répertoire du client et exécuter la commande suivante pour vous connecter au serveur OpenVPN :

```
$ cd client
$ openvpn --config client.ovpn
```

Vous devriez voir la sortie suivante :

```
Ven 21 février 15:39:18 2020 TCP / UDP: Préservation de l'adresse distante récemment utilisée:
[AF_INET] 69.87.218.145:1194
Ven 21 février 15:39:18 Tampons de socket 2020: R = [212992-> 212992] S = [212992-> 212992]
Ven 21 février 15:39:18 2020 Lien UDP local: (non lié)
Ven 21 février 15:39:18 2020 UDP link remote: [AF_INET] 69.87.218.145:1194
Ven 21 février 15:39:18 2020 TLS: paquet initial de [AF_INET] 69.87.218.145:1194, sid =
6d27e1cb 524bd8cd
Ven 21 février 15:39:18 2020 VERIFIER OK: depth = 1, CN = Easy-RSA CA
Ven 21 février 15:39:18 2020 VERIFIER OK: depth = 0, CN = tecadmin-server
Ven 21 février 15:39:18 Canal de contrôle 2020: TLSv1.3, chiffrement TLSv1.3
TLS_AES_256_GCM_SHA384, RSA 2048 bits
Ven 21 février 15:39:18 2020 [tecadmin-server] Connexion homologue initiée avec [AF_INET]
69.87.218.145:1194
Ven 21 février 15:39:19 2020 SENT CONTROL [tecadmin-server]: 'PUSH_REQUEST' (status = 1)
Ven 21 février 15:39:19 2020 PUSH: Message de contrôle reçu: 'PUSH_REPLY, redirection-
gateway
def1, dhcp-option DNS 208.67.222.222, dhcp-option DNS 208.67.220.220, route 10.8.0.1,
topologie net30, ping 20, ping-restart 60, ifconfig 10.8.0.6 10.8.0.5, peer-id 0, chiffrement AES-
256-GCM '
Ven 21 Fév 15:39:19 2020 OPTIONS IMPORT: timers et / ou timeouts modifiés
Ven 21 février 15:39:19 2020 OPTIONS IMPORT: --ifconfig / up options modifiées
Ven 21 Fév 15:39:19 2020 OPTIONS IMPORTATION: options d'itinéraire modifiées
```

Après une connexion réussie, OpenVPN attribuera une adresse IP à notre système. On peut le vérifier avec la commande suivante :

```
$  ip a show tun
```

On doit obtenir la sortie suivante :

```
4: tun: mtu 1500 qdisc state fq_codel group UNKNOWN defaut qlen 100
    link/none
    inet 10.8.0.6 peer 10.8.0.5/32 scope global tun
        valid_lft forever preferred_lft forever
    inet6 fe80 :: 7226: 57b1: f101: 313b / 64 scope link stable-privacy
        valid_lft forever preferred _lft forever
```

On peut également consulter le journal du serveur OpenVPN pour vérifier l'état de la connexion :

```
$ tail -f /var/log/openvpn.log
```

On doit obtenir la sortie suivante :

```
Ven 21 février 15:39:18 2020 45.58.34.83:37445 Canal de contrôle: TLSv1.3, chiffrement TLSv1.3
TLS_AES_256_GCM_SHA384, RSA 2048 bits
Ven 21 février 15:39:18 2020 45.58.34.83:37445 [client] Connexion homologue initiée avec
[AF_INET] 45.58.34.83:37445
Ven 21 février 15:39:18 2020 client/45.58.34.83:37445 MULTI_sva: le pool a renvoyé IPv4 =
10.8.0.6, IPv6 = (Non activé)
Ven 21 février 15:39:18 2020 client/45.58.34.83:37445 MULTI: Learn: 10.8.0.6 -> client /
45.58.34.83: 37445
Ven 21 février 15:39:18 2020 client/45.58.34.83:37445 MULTI: IP virtuelle principale pour client /
45.58.34.83: 37445: 10.8.0.6
Ven 21 février 15:39:19 2020 client/45.58.34.83: 37445 PUSH: Message de contrôle reçu:
'PUSH_REQUEST'
Ven 21 février 15:39:19 2020 client/45.58.34.83: 37445 SENT CONTROL [client]: 'PUSH_REPLY,
redirect-gateway def1, dhcp-option DNS 208.67.222.222, dhcp-option DNS 208.67.220.220,
route 10.8.0.1 , topologie net30, ping 20, ping-restart 60, ifconfig 10.8.0.6 10.8.0.5, peer-id 0,
chiffrement
AES-256-GCM '(état = 1)
Ven 21 février 15:39:19 2020 client/45.58.34.83:37445 Data Channel: en utilisant le chiffrement
négocié 'AES-256-GCM'
Ven 21 février 15:39:19 2020 client/45.58.34.83:37445 Canal de données sortant: chiffrement
'AES-256-GCM' initialisé avec une clé de 256 bits
Ven 21 février 15:39:19 2020 client/45.58.34.83:37445 Canal de données entrant: chiffrement
'AES-256-GCM' initialisé avec une clé de 256 bits
```

15. Configurer le routage à l'aide de UFW

Par défaut, le pare-feu UFW n'est pas installé dans Debian 10. On peut l'installer avec la commande suivante :

```
# apt-get install ufw -y
```

Après avoir installé le pare-feu UFW, vous devrez ajouter des règles de pare-feu pour activer le masquage afin que les clients VPN accèdent à Internet. Ouvrir le fichier **ufw**.

```
# nano /etc/default/ufw
```

Modifier la ligne suivante :

```
DEFAULT_FORWARD_POLICY = "ACCEPT"
```

Enregistrer et fermer le fichier. Ensuite, ouvrer le fichier **before.rules** :

```
# nano /etc/ufw/before.rules
```

Ajouter les lignes suivantes à la fin du fichier avant le COMMIT :

```
*nat
:POSTROUTING ACCEPT  [0: 0]
-A POSTROUTING -s 10.8.0.0/24 -o tun -j MASQUERADE
```

Enregistrer le fichier lorsque vous avez terminé. Ensuite, autoriser le port OpenVPN par défaut 1194 et OpenSSH avec la commande suivante :

```
# ufw allow 1194 /udp
# ufw allow OpenSSH
```

Ensuite, recharger le pare-feu UFW à l'aide de la commande suivante:

```
$ ufw disable
$ ufw enable
```

Si on ne veut pas installer le pare-feu, exécuter la commande suivante :

```
# iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o tun -j MASQUERADE
```

16. Commandes RaspberryPi et OpenVPN

Liste des commandes utiles à la gestion du serveur **OpenVPN**.

```
# /etc/init.d/openvpn restart   # redémarre OpenVPN
# systemctl restart openvpn@server.service   # recharge le service
# shutdown -h now   # éteint le serveur en toute sécurité
# shutdown -r now   # redémarre le serveur en toute sécurité
# apt install xrdp   # install le bureau à distance RDP
# systemctl enable xrdp   # active xrdp en tant que service système
# apt install openssh-server  # installe le SSH
# systemctl enable sshd.service  # active le service SSH au démarrage
##### Désactive la mise en veille #####
# systemctl mask sleep.target suspend.target hibernate.target hybrid-sleep.target
```

17. Conclusion

OpenVPN est installé et configuré avec succès sur le serveur **RaspberryPi Debian 10**. On peut désormais accéder à Internet en toute sécurité et protéger son identité.

OpenVPN peut aussi très facilement s'adapter à une autre plateforme Linux.

Informations :

- Le mode **TAP** correspond à du Open SSL
- Le mode **TUN**, bridgé, correspond à un tunnel IPsec, c'est du site à site