INSTALLATION D'UN SERVEUR WIREGUARD SOUS DEBIAN 10

Debian Buster

Configuration de base

Tutoriel **WIREGUARD** – DEBIAN 10

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.

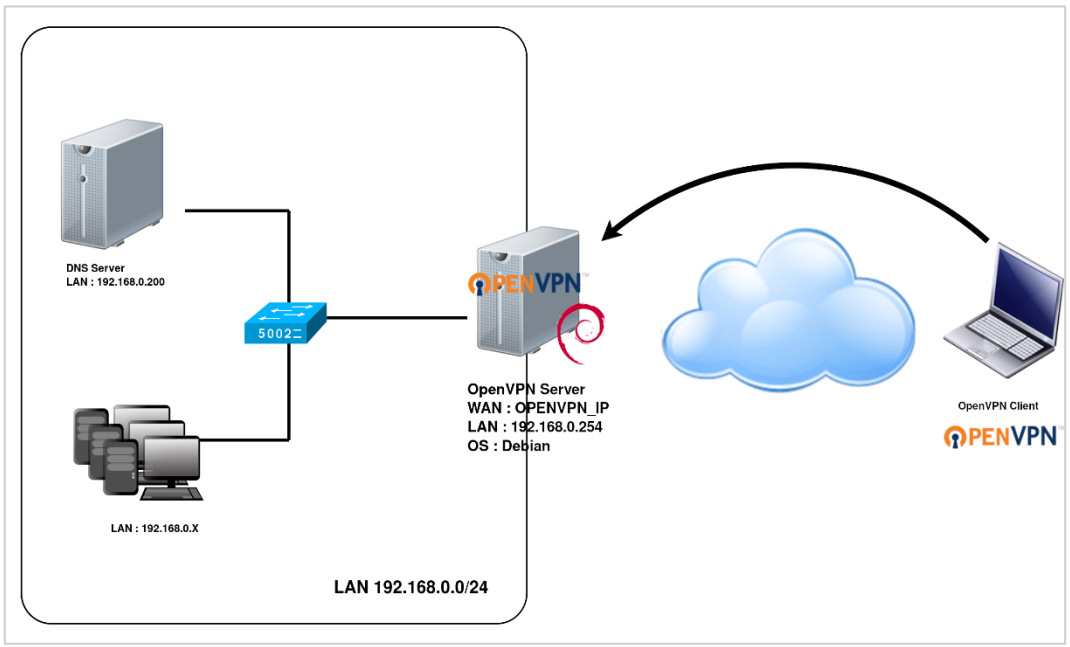
Il fournit le cryptage et l'anonymat, protège nos activités en ligne, nos achats en ligne, l'envoi d'e-mails et aide également à garder notre navigation Web anonyme.

1. Pré requis

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

- Un ou des PC client sous Windows
- Une Box (Free, Orange, Sfr…)
- Le logiciel WireGuard pour les clients
- Le logiciel Putty pour se connecter en SSH au serveur VPN
- Connaitre l'interface réseau (eth0, br0, enp0s3, ens3…) via la commande : **ip a**
  Pour notre test c'est **l'interface eth0** qui sera 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 **Debian** 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

```
# systemctl restart networking
# systemctl reboot
```

3. Paramétrage de connexion au serveur

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

- **port** : 51820
- **Protocole** : UDP

b) Activer le **SSH** sur le serveur. Pour ce faire, ouvrir le dossier **Boot**, de la carte SD du **Debian** 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 créés lors de l'installation de Linux

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. Installer le serveur WireGuard

Par défaut, le paquet WireGuard est rétroporté dans le référentiel pour Debian Buster, par conséquent.

a) Activer le dépôt de **backports**

```
# sh -c "echo 'deb http://deb.debian.org/debian buster-backports main contrib non-free'
> /etc/apt/sources.list.d/buster-backports.list"
# apt update   # Mise à jour
```

b) Rechercher le paquet Wireguard

```
# apt search wireguard   # Cherche les paquets
```

On doit obtenir la sortie suivante :

```
Sorting... Done
Full Text Search... Done
wireguard/buster-backports 1.0.20200319-1~bpo10+1 all
  fast, modern, secure kernel VPN tunnel (metapackage)

wireguard-dkms/buster-backports 0.0.20200318-1~bpo10+1 all
  fast, modern, secure kernel VPN tunnel (DKMS version)

wireguard-tools/buster-backports 1.0.20200319-1~bpo10+1 amd64
  fast, modern, secure kernel VPN tunnel (userland utilities)
```

c) Installer Wireguard VPN Server

```
# apt-get install wireguard wireguard-tools net-tools linux-headers-`uname -r`
# apt-get install wireguard-dkms wireguard-tools linux-headers-$(uname -r)
```

d) Vérifier que le module noyau Wireguard est correctement installé. Cela devrait afficher les détails du module, comme le nom de fichier, la description, l'auteur

```
# /sbin/modinfo wireguard
```

5. Activer le transfert IP

Certains aspects de la configuration réseau du serveur doivent être modifiés afin que WireGuard 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
```

a) Décommenter la ligne suivante

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

b) Vérifier que **l'ip_forward** est activé

```
# sudo sysctl -p   # active l'ip_forward
# cat /proc/sys/net/ipv4/ip_forward   # affiche le résultat
```

Installer les paquets nécessaires à la gestion de Wireguard :

```
# apt-get install git
# apt-get install python3-qrcode
```

6. Générer les clés privées et publiques

a) Générer les clés privées et publiques pour le **serveur**

```
# cd /etc/wireguard
# umask 077
# wg genkey | tee server-private.key | wg pubkey > server-public.key
```

b) Générer les clés privées et publiques pour le **client**

```
# mkdir /home/user/configs
# cd /home/user/configs
# umask 077
# wg genkey | tee client1-private.key | wg pubkey > client1-public.key
```

c) Vérifier les clés privées et publiques du serveur

```
# ls -l server-private.key server-public.key   # affiche les droits sur les clés
# cat server-private.key   # affiche la clé privée
# cat server-public.key   # affiche la clé public
```

On doit obtenir la sortie suivante pour les clés serveurs

```
-rw-r--r-- 1 root root 45 Feb 24 13:37 server-private.key
-rw-r--r-- 1 root root 45 Feb 24 13:37 server-public.key
bGMzxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx=
EG20xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx=
```

d) Copier chaque clé dans les fichiers de configuration du **serveur** et du **client**

- server-private.key -> wg0.conf
- server-public.key -> client1.conf
- client-private.key -> client1.conf
- client-public.key -> wg0.conf

e) Générer une clé partagée, à copier dans les 2 fichiers de configuration, dans le champ **PresharedKey**

```
# wg genkey | tee partage-shared.key | wg pubkey > partage-shared.key
```

6. Créer le fichier de configuration du serveur

Le fichier doit impérativement être nommé **nomfichier.conf**, où **nomfichier** est un nom d'interface réseau valide sur le système. Les outils que nous allons utiliser par la suite vont en effet se servir de ce nom pour créer l'interface réseau. Donc si on nomme le fichier **wg0.conf**, l'interface s'appellera **wg0** et si on nomme le fichier **toto.conf**, l'interface réseau s'appellera **toto**.

a) Créer le fichier de configuration

```
# nano /etc/WireGuard/wg0.conf
```

b) Ajouter les lignes **PostUp**, **PostDown**. On doit obtenir un fichier comme ci-dessous (remplacer **eth0** par l'interface réseau du serveur)

[Interface]
PrivateKey = <server-private.key>
Address = 10.0.0.1/24
SaveConfig = true
PostUp = iptables -A FORWARD -i wg0 -j ACCEPT; iptables -A FORWARD -o wg0 -j ACCEPT; iptables -t nat -A POSTROUTING -o **eth0** -j MASQUERADE
PostDown = iptables -D FORWARD -i wg0 -j ACCEPT; iptables -D FORWARD -o wg0 -j ACCEPT; iptables -t nat -D POSTROUTING -o **eth0** -j MASQUERADE
ListenPort = 51820
[Peer]
PublicKey = <client-public.key>
PresharedKey = <partage-shared.key>
AllowedIPs = 10.0.0.2/32

c) Activer l'interface **wg0**

```
# ip link set up dev wg0   ou   # sudo wg-quick up ./wg0.conf
```

7. Créer le fichier de configuration du client

a) Créer le fichier **client1.conf** dans un dossier public, tel que : **/home/user/configs/** et lui donner les droits avec la commande chmod.

```
# chmod -R 755 /home/user/configs
$ nano /home/user/configs/client1.conf
```

b) Ajouter la ligne **PersistentKeepalive = 25**, pour que la connexion reste active

[Interface]
PrivateKey = <client-private.key>
Address = 10.0.0.2/24
DNS = 208.67.222.222, 208.67.220.220
[Peer]
PublicKey = <server-public.key>
PresharedKey = <partage-shared.key>
Endpoint = xxx.xxx.xxx.xxx:51820
AllowedIPs = 0.0.0.0/0
PersistentKeepalive = 25

c) Modifier la ligne **AllowedIPs** du fichier de conf. Par défaut, AllowedIPs est défini sur 0.0.0.0/0, ce qui signifie que tout le trafic passera par le réseau domestique (VPN à tunnel complet).

Pour changer cela afin que seul le trafic local soit envoyé via WireGuard, nous devons changer cette ligne par la plage IP locale. Pour la plupart, ce sera 192.168.1.0/24 ou 192.168.0.0/24. Si l'on souhaite que les clients VPN se parlent, on doit également ajouter le sous-réseau VPN (10.6.0.0/24).

d) **Indiquer au serveur**, quels clients pourront s'y connecter. Pour ce faire, on va lui indiquer pour chaque client, sa **clé publique** (à récupérer sur le poste client via le logiciel Wireguard, ainsi que l'adresse IP (ou la page d'adresses) qu'il pourra s'attribuer à l'intérieur du VPN.

```
# wg set wg0 peer <clef_publique_client> allowed-ips 10.0.0.1/32
```

Si on veut retirer l'accès à un client, il suffit de taper la commande suivante

```
# wg set wg0 peer <clef_publique_client> remove
```

8. Démarrer le service WireGuard

On peut maintenant démarrer le service WireGuard et l'activer après le redémarrage du système à l'aide de la commande suivante :

```
# systemctl enable wg-quick@wg0
```

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

```
# wg show
```

On doit obtenir la sortie suivante :

```
interface: wg0
  public key: 2huPxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx=
  private key: (hidden)
  listening port: 51820
  fwmark: 0xca6c
peer: lclUxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx=
  preshared key: (hidden)
  endpoint: xxx.xxx.xxx.xxx:43964
  allowed ips: 10.0.0.2/32
latest handshake: 40 seconds ago
transfer: 17.21 KiB received, 16.77 KiB sent
```

9. Connecter le client Windows au VPN

Il faut transférer le fichier de configuration sur le PC Client à l'aide d'un logiciel FTP.

a) Se connecter au serveur via **FileZilla** avec les mêmes identifiants utilisés dans Putty.
b) Ouvrir le dossier et récupérer le fichier

```
$ /home/user/configs/client1.conf
```

c) Copier les fichiers dans le dossier souhaité **C:\Documents\keys**
d) Ouvrir le client WireGuard
e) Importer le fichier **client1.conf**, puis cliquer sur le bouton **Activer**

10. Créer un qrcode du fichier client1.conf

a) Ouvrir le dossier des configurations clients

```
$ cd /home/user/configs
```

b) Exécuter la commande suivante

```
# qr client1.conf
```

c) Photographier le **qr code** avec l'application WireGuard du smartphone

11. 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 vos clients VPN accèdent à Internet.

a) Tout d'abord, on doit configurer UFW pour accepter les paquets transférés

```
# sudo ufw allow 51820/udp
```

b) Activer et démarrer le service Wireguard

```
# systemctl enable wg-quick@wg0
# systemctl start wg-quick@wg0
```

c) Vérifier le status de Wireguard

```
# systemctl status wg-quick@wg0
```

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

```
# sudo ufw disable
# sudo ufw enable
```

e) Vérifier que l'interface wg0 est opérationnelle

```
# ip a show wg0
```

On doit obtenir la sortie suivante

```
5: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noqueue state UNKNOWN group
default qlen 1000
  link/none
  inet 10.0.0.1/24 scope global wg0
    valid_lft forever preferred_lft forever
```

12. Commandes Linux Debian

a) Liste des commandes basique à la gestion du serveur Linux Debian

```
# wg-quick up ./wg0.conf   # activer WireGuard
# wg-quick down ./wg0.conf   # désactiver WireGuard
# systemctl start wg-quick@wg0   # démarrer WireGuard
# systemctl stop wg-quick@wg0   # arrêter WireGuard
# systemctl poweroff   # éteint le serveur en toute sécurité
# systemctl reboot   # 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
```

13. Utilisation local du VPN

Une fois connecté au VPN **via un PC local**, impossible d'accéder aux périphériques réseaux locaux sans ajouter une route spécifique.

a) Ouvrir une invite de commande

b) Saisir la commande : **route -p add 192.168.1.X/24 10.0.0.2** (@IP du périphérique / @IP du VPN)

c) Mapper les périphériques via leur adresse IP

14. Conclusion

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

Destiné à Linux Debian, **WireGuard** fonctionne aussi parfaitement sur une distribution Fedora, Ubuntu ou Mint en mode VPS ou sur un ordinateur personnel.