Table des matières

Wireguard VPN 2
Installation 2
Générer les clés 2
Serveur 2
Client 2
Configuration 3
Serveur 3
Restriction 3
Client 4
Port Forward 5
Activer l'interface 5
Désactiver l'interface 5
Contôler iptables 6
Gestion du service 6
Configuration Client 6
Android 6
Ubuntu 7
Injecter la conf 7
Dépannage 8

Wireguard VPN

Installation

```
apt install wireguard raspberrypi-kernel-headers wireguard-tools resolvconf
```

Générer les clés

```
cd /etc/wireguard ; umask 077
```

Serveur

Générer la clé privé:

```
wg genkey > server_private.key
```

Générer la clé publique:

```
wg pubkey > server_public.key < server_private.key
```

Client

Généré une clé privé:

```
wg genkey > client1_private.key
```

Généré une clé public:

```
wg pubkey > client1_public.key < client1_private.key
```

Généré une clé partagé:

```
wg genpsk >> client1_preshared.key
```

Configuration

Serveur

```
nano /etc/wireguard/wg0.conf
```

```
[Interface]
# plage d'adresse privée pr le réseau vpn
Address = 10.9.0.1/24
MTU = 1420
# port d'écoute du serveur vpn
ListenPort = 55135
# @ip du resolveur dns , ici c'est l'@ip de mon pihole
# mais on peut aussi mettre celle d'un public,par ex:1.1.1.1
DNS = 192.168.1.2 # ip du resolveur
# mettre la clé privée de notre serveur (fichier server_private_key)
PrivateKey = VJUUasN+z6hYn4C6xxxxxxxxxxxxxxxx=
# règles de routage
# penser à adapter avec le nom de votre interface reseau (ici eth0)
# au besoin faire un "ip a" pour trouver votre interface
PostUp = iptables -A FORWARD -i %i -j ACCEPT; iptables -A FORWARD -o %i -j
ACCEPT; iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
PostDown = iptables -D FORWARD -i %i -j ACCEPT; iptables -D FORWARD -o %i -j
ACCEPT; iptables -t nat -D POSTROUTING -o eth0 -j MASQUERADE
[Peer]
# client1 - telephone android
# contenu de la clé publique (fichier client1_public.key)
PublicKey = bLuMB4ed7txxxxxxxxxxxxxxxxxxxxxxxx
PresharedKey = vQw81695uC1R3Y01cvCN56CvinQqDrZRQuLavSne+AE==
# @ip privée du vpn client
AllowedIPs = 10.9.0.2/32
PersistentkeepAlive = 60
```

Restriction

Si vous souhaitez restreindre l'accès a certain de vos réseaux à vos client. (Dans le cas ou votre serveur peut initialement communiquer avec).
Voici comment procéder via les règles PostUp et PostDown de iptables:

```
PostUp = iptables -A FORWARD -i %i -j ACCEPT; \
         iptables -A FORWARD -o %i -j ACCEPT; \
```

```
         iptables -t nat -A POSTROUTING -o enxb827eb0659e0 -j MASQUERADE; \
         iptables -I FORWARD 1 -s 10.9.0.8/30 -d 192.168.0.0/24 -j DROP; \
         iptables -I FORWARD 1 -s 10.9.0.8/30 -d 10.200.144.0/27 -j DROP; \
         iptables -I FORWARD 1 -s 10.9.0.8/30 -d 10.200.166.0/25 -j DROP

PostDown = iptables -D FORWARD -i %i -j ACCEPT; \
           iptables -D FORWARD -o %i -j ACCEPT; \
           iptables -t nat -D POSTROUTING -o enxb827eb0659e0 -j MASQUERADE;
\
           iptables -D FORWARD -s 10.9.0.8/30 -d 192.168.0.0/24 -j DROP; \
           iptables -D FORWARD -s 10.9.0.8/30 -d 10.200.144.0/27 -j DROP; \
           iptables -D FORWARD -s 10.9.0.8/30 -d 10.200.166.0/25 -j DROP
```

Le paramètre **“iptables -I FORWARD 1”** positionne les règles avant toutes les autres dans iptables.
Le segment **“10.9.0.8/30”** correspond au ip (0.8 à 0.11) du range !!

Client

```
Interface]
#client 1
# @ip du client
Address = 10.9.0.2/24
DNS = 9.9.9.9, 149.112.112.112
# contenu du fichier client1_private_key
PrivateKey = aI2cUEOwaMNjgfdxxxxxxxxxxxxxxxxxxxxxxxxx (Clé privé généré pour
le client)

[Peer]
#client 1
# contenu du fichier server_public_key
PublicKey = Tzq/nLB07p8GKxxxxxxxxxxxxxxxxxxxxxxxxxxxx (Celle du serveur VPN)
PresharedKey = hggrdmqjdml5kcn99Pklbhdczocizasc9A=  (celle du device)

# @ipPubliqueBox:port_vpn,
Endpoint = 82.xx.xx.xxx:51820 (ou Nom de domaine si redirect)
AllowedIPs = 0.0.0.0/0, ::0/0
#(si 0.0.0.0/0 alors tout le trafic est routé dans le vpn)
#AllowedIPs = 192.168.1.0/24, 192.168.5.0/28
#(si 192.168.1.0/24, 192.168.5.0/28 alors seulement le trafic à destination
de ces réseaux sera routé dans le tunnel vpn)
PersistentKeepalive = 25 (pas obligatoire)
```

Port Forward

dé-commenter la ligne dans le fichier de configuration.

```
nano /etc/sysctl.conf
net.ipv4.ip_forward=1
```

Recharger les paramètres du noyau:

```
sysctl -p /etc/sysctl.conf
```

Activer l'interface

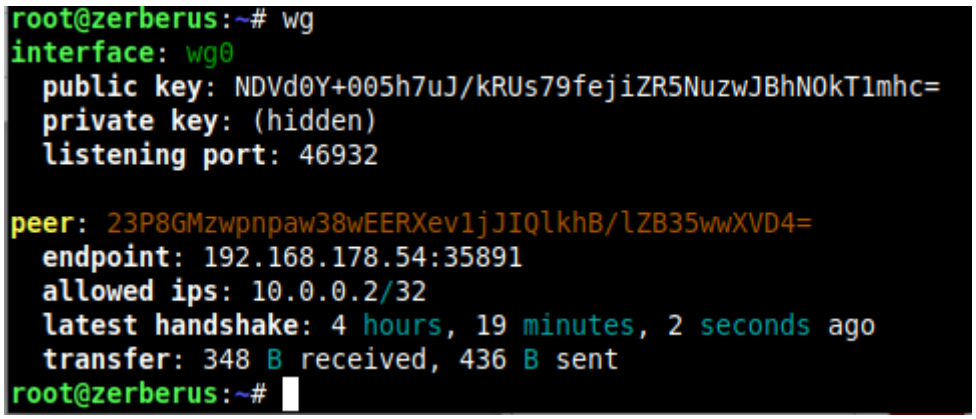
```
wg-quick up wg0
```

Vérifier le tunnel.

```
ip a
```

Vérifier le status.

```
sudo wg
```



Activer l'interface pour chaque redémarrage*

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

Désactiver l'interface

```
wg-quick down wg0
```

Contôler iptables

```
sudo iptables -L FORWARD -v -n --line-numbers
```

Gestion du service

```
systemctl status wg-quick@wg0
systemctl start wg-quick@wg0
systemctl stop wg-quick@wg0
systemctl enable wg-quick@wg0
systemctl disable wg-quick@wg0
```

Configuration Client

Android


```
sudo apt install qrencode
```

```
qrencode -t ansiutf8 < /etc/wireguard/wg0-client1.conf
```

Il ne vous restera plus qu'a flasher le QRCode avec votre mobile depuis le shell ! 😎

Ubuntu

Installer Wireguard:

```
sudo apt install wireguard-tools
```

Monter le tunnel:

```
sudo wg-quick up "votre-fichier.conf"
```

Vérifier le tunnel :

```
sudo wg show
```

```
interface: Mon-vpnCNX
  public key: obvzuobazvrevTEzefzdFDJFZVZDADBFE=
  private key: (hidden)
  listening port: 50833
  fwmark: 0xcc7a

peer: frTGZfonfz834/L3rFTJD457dvkhzpdvzosdcp=
  preshared key: (hidden)
  endpoint: 77.45.143.123:55135
  allowed ips: 0.0.0.0/0, ::/0
  latest handshake: 38 seconds ago
  transfer: 32.26 KiB received, 36.04 KiB sent
```

Have Fun !! 😎

Injecter la conf

Vous pouvez injecter la configuration directement dans l'environnement graphique de Gnome ou KDE Plasma.
Ceci permettra d'avoir l'option VPN déjà configuré et accessible depuis la GUI.

1. Ouvrir un terminal
2. Saisir la commande

```
nmcli connection import type wireguard file MaConfWG.conf
```

Dépannage

Si vous avez cette erreur après montage du tunnel:

```
wg-quick up wg0
```

/usr/bin/wg-quick: line 31: resolvconf: command not found

Faire ceci:

```
ln -s /usr/bin/resolvectl /usr/local/bin/resolvconf
```

```
wg-quick up wg0
```

```
dbus-daemon[798]: [system] Activating via systemd: service
name='org.freedesktop.resolve1'......
```

Passer par le resolver de systemd:

```
systemctl enable systemd-resolved.service
```

```
systemctl start systemd-resolved.service
```

Noter qu'après chaque ajout de client dans la configuration il faut relancer le tunnel vpn.

```
wg-quick down wg0 / wg-quick up wg0
```