Mise en place de Wireguard Client

Pour utiliser Wireguard en mode client, deux choix s'offre à vous :
1- Le premier, vous créez un conteneur Docker en mode client
2- Le deuxième vous installer le package et vous configurer Wireguard client directement sur votre machine.

1 - Via l'utilisation de l'image docker

Via l'utilisation de l'image docker

Ne définissez pas la variable d'environnement `PEERS`. Déposez votre conf client dans le dossier config en tant que `/config/wg0.conf` et démarrez le conteneur.

Si vous obtenez des erreurs liées à IPv6 dans le journal et que la connexion ne peut pas être établie, modifiez la ligne `AllowedIPs` dans votre pair/client wg0.conf pour inclure uniquement `0.0.0.0/0` et non `::/0`; et redémarrez le conteneur.

docker-compose :

```
version: "2.1"

services:

  wireguard:
    image: lscr.io/linuxserver/wireguard:latest
    container_name: wireguard
    cap_add:
      - NET_ADMIN
      # - SYS_MODULE  # optional already active
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=Europe/Paris
      - LOG_CONFS=true
    volumes:
      - /etc/container-conf/wireguard/appdata/config:/config
      - /etc/container-conf/wireguard/lib/modules:/lib/modules
    ports:
      - 51820:51820/udp
    restart: unless-stopped
```

```
    sysctls:
      - net.ipv4.conf.all.src_valid_mark=


volumes:
 wireguard:
```

2 - Via l'utilisation du package Linux

Via l'utilisation du package Wireguard

Sur le client

1/ Tout d'abord vous devez installer wireguard sur la machine :

```
sudo apt-get install wireguard
```

Puis se déplacer dans le répertoire de wireguard, par défaut le dossier est vide :

```
root@linux: cd /etc/wireguard/
root@linux: ls -al
total 28
drwx------   2 root root  4096 juin  13 12:03 ./
drwxr-xr-x 166 root root 12288 juin  13 11:24 ../
```

2/ Maintenant nous allons générer la clé privé et la clé publique :

```
wg genkey | tee privatekey | wg pubkey | tee publickey
```

Nous avons maintenant 2 fichiers dans le dossier wireguard :

```
root@linux: wg genkey | tee privatekey | wg pubkey | tee publickey
WKtF71pM6wObswaXkbxJgwPhk8a6lqrPb9oKFjaGOmM=
root@linux: ls -al
total 28
drwx------   2 root root  4096 juin  14 15:55 .
drwxr-xr-x 166 root root 12288 juin  13 11:24 ..
-rw-r--r--   1 root root    45 juin  14 15:55 privatekey
-rw-r--r--   1 root root    45 juin  14 15:55 publickey
```

3/ Pour continuer la configuration il faut noter le contenu des 2 clés générés :

```
root@linux: cat privatekey
iApdu05JqNGSp/r7PSpJ5Zqxs4kWSR6qYv9onitvsmo=

root@linux: cat publickey
WKtF71pM6wObswaXkbxJgwPhk8a6lqrPb9oKFjaGOmM=
```

> Attention ne partagez jamais votre clé privé !!

Clé privé : iApdu05JqNGSp/r7PSpJ5Zqxs4kWSR6qYv9onitvsmo=
Clé publique : WKtF71pM6wObswaXkbxJgwPhk8a6lqrPb9oKFjaGOmM=

4/a) Création du fichier de configuration Wireguard client :

```
touch wg0.conf
```

4/b) Préparer dans un bloc note la configuration suivante :

Il faut bien veiller à remplacer les variables :

- `<private_key_client>` -> Clé privé généré précédemment
- `<range_ip_vpn>` -> Adresse IP VPN du client
- `<public_key_server>` -> Clé publique du serveur (A récupérer sur le serveur)
- `<ip_server>` -> Adresse IP du serveur
- `<port_server>` -> Port écoutant sur le serveur

```
[Interface]
PrivateKey = <private_key_client>
Address = <range_ip_vpn>

[Peer]
###Public of the WireGuard VPN Server
PublicKey = <public_key_server>

### IP and Port of the WireGuard VPN Server
Endpoint = <ip_server>:<port_server>

### Allow all traffic
AllowedIPs = 0.0.0.0/0
```

4/c) Editer le fichier de configuration en collant la configuration complété :

```
nano wg0.conf
# ou
vim wg0.conf
```

Ce qui donne :

```
[Interface]
PrivateKey = iApdu05JqNGSp/r7PSpJ5Zqxs4kWSR6qYv9onitvsmo=
Address = 192.168.10.15/24

[Peer]
###Public of the WireGuard VPN Server
PublicKey = <public_key_server>

### IP and Port of the WireGuard VPN Server
Endpoint = vpn.domaine-name.com:51820

### Allow all traffic
AllowedIPs = 0.0.0.0/0
```

> Le reste de la configuration continue sur la partie serveur

Sur le serveur

> `wg0` est l'interface référençant tous les appareils pouvant se connecter au VPN.

Préparer la commande suivante :

```
wg set wg0 peer <public_key_client> allowed-ips <adress_ip_vpn_client>
```

Ce qui donne :

```
wg set wg0 peer WKtF71pM6wObswaXkbxJgwPhk8a6lqrPb9oKFjaGOmM= allowed-ips 192.168.10.15
```

Puis exécuter la commande.

Pour être sur du bon fonctionnement de la commande précédente, nous pouvons lister les clients enregistré dans le fichier de conf du serveur :

```
wg show wg0
```

Maintenant il faut confirmer et enregistrer la modification du fichier de configuration :

```
wg-quick save wg0
```

Sur le client

Nous avons créer le client et nous l'avons enregistré dans la configuration serveur. Maintenant il nous reste plus qu'a démarrer le VPN sur le client :

```
root@linux: wg-quick up wg0
[#] ip link add wg0 type wireguard
[#] wg setconf wg0 /dev/fd/63
[#] ip -4 address add 10.13.13.6/32 dev wg0
[#] ip link set mtu 1420 up dev wg0
[#] wg set wg0 fwmark 51820
[#] ip -4 route add 0.0.0.0/0 dev wg0 table 51820
[#] ip -4 rule add not fwmark 51820 table 51820
[#] ip -4 rule add table main suppress_prefixlength 0
[#] sysctl -q net.ipv4.conf.all.src_valid_mark=1
[#] nft -f /dev/fd/63
```

Vérification du fonctionnement du VPN côté client :

```
ifconfig
```

Vérification du fonctionnement du nouveau client côté serveur :

```
wg show wg0
```

3 - via le logiciel Windows

source : https://www.youtube.com/watch?v=u8w6jIdU39s&t=0s