Configurer un VPN Wireguard

Pour rappel, Wireguard est un protocole VPN très jeune, et donc peut-être pas assez mature pour vos usages. Évidemment, il est impossible d'être exhaustif, et ça n'a jamais été le but de ce wiki.

Il faut aussi que j'explique rapidement les modes de fonctionnement de Wireguard. Il peut fonctionner en mode Client/Serveur, Serveur/Serveur et Mesh, tout cela ce fait de manière implicite. Pour faire simple, il est obligatoire qu'au moins une des deux machines est un port d'écoute accessible, même si il est recommandé, si possible, que les deux est un port d'écoute.
Dans le cas de Mesh, vous pouvez faire en sorte qu'une machine n'est pas de port accessible, si toutes les autres en ont un pour recevoir les communications de celle-ci.

Pour commencer, il faut évidemment installer Wireguard qui se base sur un module noyau intégré par défaut intégré dans la version 5.6, et disponible par DKMS pour les versions plus vieilles. Par exemple, sous Debian :

```
# apt install wireguard
```

Maintenant on va générer une paire de clé. Il vous faudra générer une paire de clé par VPN et par Hôte. Pour cela, lancez la commande :

```
# WGPRIV="`wg genkey`" sh -c 'echo $WGPRIV ; echo $WGPRIV | wg pubkey'
```

Cela devrait vous retourner deux lignes, la première est la clé privée, la seconde est la clé publique.

Vous pouvez si vous le souhaité aussi générer une **Pre-Shared Key** pour augmenter la sécurité. Celle-ci doit être partagée entre deux machines, mais pas réutilisée avec un autre Peer. Ce n'est pas simple à expliquer, vous avez un exemple dans la section Configuration Mesh mais si vous n'avez pas compris comment l'utiliser, ne l'utilisez pas. Si vous utilisez mal la **Pre-Shared Key**, vous rendez plus vulnérable votre session que si vous ne l'aviez pas utilisé. Pour ce faire, simplement utilisez la commande :

```
# wg genpsk
```

Ensuite on va créer pour chaque Hôte un fichier de configuration. Certaines options sont obligatoires, d'autres non. Voici ce que chaque option représentent :

wg0.conf

```
[Interface]
PrivateKey = <Clé privé précédemment généré>
ListenPort = <Port d'écoute du VPN (obligatoire sur au moins un hôte,
recommandé sur les deux)>
Address = <IP avec son masque CIDR, séparé par des virgules si
plusieurs IP (optionnel)>
DNS = <IP de serveurs DNS, séparé par des virgules (optionnel)>
MTU = <Pour surpasser la détection de MTU (optionnel)>
Table = <Nom de la table de routage où injecter les routes. Peut-être
```

```
"off" pour désactiver l'ajout des règles dans une table. Par défaut, c'est la table de routage principale (optionnel)>
PreUp = <Commande exécutée avant le lancement du VPN (optionnel)>
PostUp = <Commande exécutée après le lancement du VPN (optionnel)>
PreDown = <Commande exécutée avant l'arrêt du VPN (optionnel)>
PostDown = <Commande exécutée après l'arrêt du VPN (optionnel)>
SaveConfig = <Si "true", enregistre dans ce fichier toute modification faite "a la volé" avec la commande wg (optionnel)>


[Peer]
PublicKey = <Clé publique de l'hote>
Endpoint = <nom d'host/IP suivit du port d'écoute de la machine paire (optionnel car dépends de si la machine paire a un port d'écoute de configuré>
AllowedIPs = <IP/Réseau autorisé a transité depuis la machine paire. Cela ajoute aussi les routes dans la table de routage (optionnel)>
PresharedKey = <Pré pré-partagé (optionnel)>
PersistentKeepalive = <Envoi de données periodique pour maintenir la session ouverte (optionnel)>
```

Le nom du fichier de configuration sera le nom de l'interface VPN, il se peut donc qu'il y ait des contraintes spécifiques en fonction de l'OS que vous utilisez.

Placer votre fichier de configuration dans le dossier **/etc/wireguard/**, et lancez-la avec la commande :

```
# systemd start wg-quick@wg0
```

Vous pouvez dans faire en sorte de le lancer au démarrage comme un service classique.

Maintenant on va passer aux exemples.

Mode client / serveur

Voici la configuration serveur :

/etc/wireguard/wg0.conf

```
[Interface]
PrivateKey = kPK7kPYHxvLCizn0HQcnZaUob3paDtya2XjxF5X+QGQ=
ListenPort = 51291
Address = 10.91.0.1/24
PostUp = iptables -t nat -I POSTROUTING -i wg0 -o eth0 -j MASQUERADE
PostDown = iptables -t nat -D POSTROUTING -i wg0 -o eth0 -j MASQUERADE

[Peer]
PublicKey = xDfJp0vSPRCV0KyRTmHBi66Ve/XnCur8ysyGvS1M1i4=
```

```
AllowedIPs = 10.91.0.2/32
```

Et la configuration client :

/etc/wireguard/wg0.conf

```
[Interface]
PrivateKey = OBk5puulZn4hBycsdvxicfwXtRMgH7v6Iuz51frjGW8=
Address = 10.91.0.2/24
DNS = 8.8.8.8

[Peer]
PublicKey = TgM2pGrQnrVOYPpJLcuPmMElGDdFjMlxEYenrKNtbmo=
AllowedIPs = 0.0.0.0/24
Endpoint = myvpn.fr:51291
```

Dans le cas où vous avez besoin que la session soit toujours active, même si le client n'envoie pas de données[1)], je recommande d'ajouter l'option **PersistentKeepalive** dans sa configuration.

Configuration serveur / serveur

Configuration du serveur 1 :

/etc/wireguard/wg0.conf

```
[Interface]
PrivateKey = IAlQf25Ccdsk/BEjE48fPFknPsJsSLl9PURUSzrYLkk=
ListenPort = 51291
Address = 10.91.0.1/24 , fc00::1/64

[Peer]
PublicKey = MhAAHTiL9Yr/etc9T3n4rnyE8EFg5pTd7GCKQo24S1Q=
AllowedIPs = 10.91.0.2, 10.20.0.0/16 , fc00::2/64 , fd00:20::/64
Endpoint = server2.fr:51291
```

Et celle du serveur 2 :

/etc/wireguard/wg0.conf

```
[Interface]
PrivateKey = CMjWIFTj4ElIrkrh29GTb5nfQXmYZujNB8iPKZHUnFc=
ListenPort = 51291
Address = 10.91.0.2/24 , fc00::2/64

[Peer]
PublicKey = YBXY1gGwWw9TWLRECykN0YZ+LWRjhWYq+CB9akowAUU=
```

```
AllowedIPs = 10.91.0.1, 10.10.0.0/16 , fc00::1/64 , fd00:10::/64
Endpoint = server1.fr:51291
```

Configuration Mesh

Pour l'exemple, je vais montrer avec 3 serveurs, dont 1 qui ne peut pas avoir de ports d'écoute, ce qui n'est pas la situation recommandée. Essayer d'avoir un port d'écoute sur chaque serveur.

Configuration du serveur 1 :

/etc/wireguard/wg0.conf

```
[Interface]
PrivateKey = 8P4uWq+h/VHX7Snah8YgdL1enK1J/BvjKoOgjZV1MXg=
ListenPort = 51291
Address = 10.91.0.1/24

[Peer]
# Serveur 2
PublicKey = jvLEQMgGaozqnMZY7GVfg5buRiHFFVcZdCEyDQ/AyQM=
AllowedIPs = 10.91.0.2 , 10.20.0.0/24
PresharedKey = SbqgoqmmVFhxI4b8ytv0oYFufty9s+5dYogsVf6ymzU=
Endpoint = server2.fr:51291

[Peer]
# Serveur 3
PublicKey = MhAAHTiL9Yr/etc9T3n4rnyE8EFg5pTd7GCKQo24S1Q=
AllowedIPs = 10.91.0.3 , 10.30.0.0/24
PresharedKey = yjyqKqgB5Le4jxFzglFEvzaR/J/c4e61KZ3Bp07j8G8=
```

Configuration du serveur 2 :

/etc/wireguard/wg0.conf

```
[Interface]
PrivateKey = 4BNkcX2zQ1Q19XsAa6uWHJZzEZ5oHAvOQR5pIU44vWk=
ListenPort = 51291
Address = 10.91.0.2/24

[Peer]
# Serveur 1
PublicKey = xOE4qzHEWyRwObpMc15USyYjGBUMx1/VGLgTC3tNC28=
AllowedIPs = 10.91.0.1 , 10.10.0.0/24
PresharedKey = SbqgoqmmVFhxI4b8ytv0oYFufty9s+5dYogsVf6ymzU=
Endpoint = server2.fr:51291

[Peer]
```

```
# Serveur 3
PublicKey = MhAAHTiL9Yr/etc9T3n4rnyE8EFg5pTd7GCKQo24S1Q=
AllowedIPs = 10.91.0.3 , 10.30.0.0/24
PresharedKey = MJ78z/bOlq/UB8lsxDoUTUZOLxK2BXe50IItRGBqpLc=
```

Et la configuration du serveur 3, celui qui n'a pas de port d'écoute :

/etc/wireguard/wg0.conf

```
[Interface]
PrivateKey = 4BNkcX2zQ1Q19XsAa6uWHJZzEZ5oHAvOQR5pIU44vWk=
Address = 10.91.0.3/24

[Peer]
# Serveur 1
PublicKey = xOE4qzHEWyRwObpMc15USyYjGBUMx1/VGLgTC3tNC28=
AllowedIPs = 10.91.0.1 , 10.10.0.0/24
PresharedKey = yjyqKqgB5Le4jxFzglFEvzaR/J/c4e61KZ3Bp07j8G8=
Endpoint = server1.fr:51291
PersistentKeepalive = 25

[Peer]
# Serveur 2
PublicKey = jvLEQMgGaozqnMZY7GVfg5buRiHFFVcZdCEyDQ/AyQM=
AllowedIPs = 10.91.0.2 , 10.20.0.0/24
PresharedKey = MJ78z/bOlq/UB8lsxDoUTUZOLxK2BXe50IItRGBqpLc=
Endpoint = server2.fr:51291
PersistentKeepalive = 25
```

Notez l'utilisation des **PresharedKey**, ainsi que celle de **PersistentKeepalive** .

[1)]
exemple si vous utilisez cette méthode pour passer au travers d'un firewall, et que vous avez besoin que des usagers cotés serveurs puissent accéder à des ressources cotées du client

From:
https://wiki.virtit.fr/ - **VirtIT**

Permanent link:
https://wiki.virtit.fr/doku.php/kb:linux:generalites:configurer_un_vpn_wireguard?rev=1598004729

Last update: **2020/08/21 10:12**