Tunnel Wireguard d'un serveur local à un VPS

Cette configuration permet d'établir une connexion entre votre réseau local et un serveur VPS, vous offrant la possibilité d'ouvrir les ports nécessaires sur le VPS sans exposer directement votre réseau. Vous bénéficiez également de la protection DDoS fournie par le serveur distant. Cette solution est pratique si vous ne pouvez pas ouvrir de ports sur votre routeur ou si vous disposez d'une adresse IP dynamique. L'utilisation d'un VPS comme intermédiaire vous permet de contourner les limitations de votre routeur ou de votre fournisseur d'accès Internet, tout en offrant une sécurité supplémentaire.

Prérequis

- Un minimum de connaissances en réseau, administration VPS.
- Vous disposez d'un VPS sur Debian chez IONOS, OVH, autres.
- Connaître l'adresse IP publique du VPS.
- Une heure de temps pour faire la configuration.

Installation du serveur WireGuard sur le VPS

Connectez-vous en SSH à votre machine :

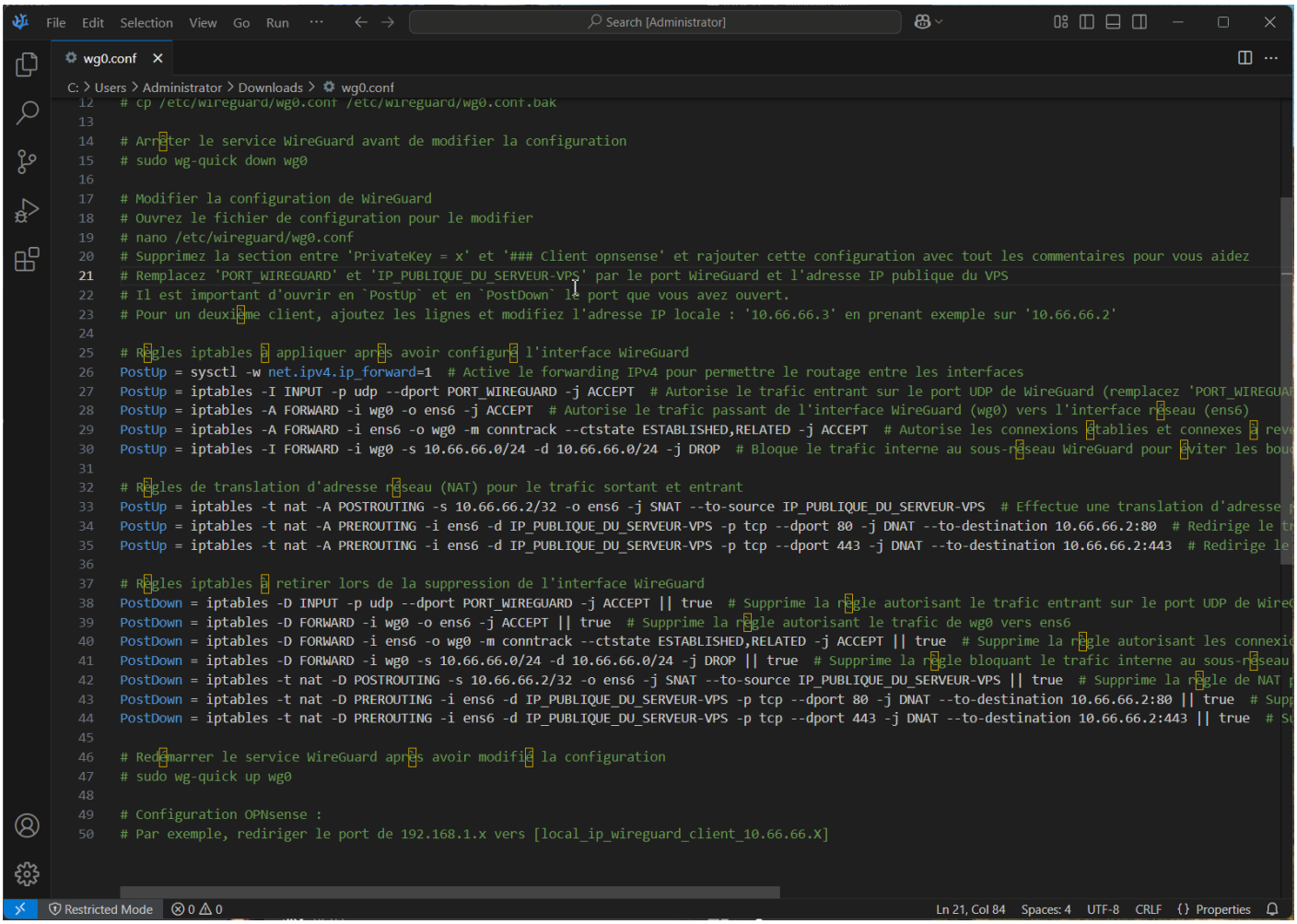
1. Connectez-vous en tant que root :

```
sudo -i
```

2. Suivez les instructions de ce dépôt pour installer votre serveur WireGuard. Laissez tout par défaut pour les questions posées.
3. À la fin de l'installation, vous aurez accès au fichier de configuration. Exécutez la commande `cat /chemindevotrefichier.conf` pour copier les informations du client. Notez ces informations quelque part afin de pouvoir les utiliser plus tard dans le processus d'installation.

4. Vous pouvez créer plusieurs clients, mais les machines ne communiqueront pas entre elles avec `iptables -I FORWARD -i wg0 -s 10.66.66.0/24 -d 10.66.66.0/24 -j DROP`, ce qui est une bonne pratique de sécurité.
5. Vous devez lire le fichier de configuration pour comprendre son fonctionnement. Vous pouvez utiliser un IDE pour changer les occurrences. Vous avez trois choses à faire : remplacer PORT_WIREGUARD par le port WireGuard généré durant l'installation du script, IP_PUBLIQUE_DU_SERVEUR-VPS par l'adresse IP publique du VPS, et ajouter en dessous les ports que vous voulez en prenant exemple sur PostUp et PostDown.

```
PostUp = iptables -t nat -A PREROUTING -i ens2 -d IP_PUBLIQUE_DU_SERVEUR-VPS -p tcp --dport 443 -j DNAT --to-destination 10.66.66.2:443  # Redirige le trafic entrant vers l'adresse IP du client WireGuard
PostDown = iptables -t nat -D PREROUTING -i ens2 -d IP_PUBLIQUE_DU_SERVEUR-VPS -p tcp --dport 443 -j DNAT --to-destination 10.66.66.2:443 || true  # Supprime la redirection du trafic entrant
```




```
# Description :
# Cette configuration vous permet d'établir une connexion entre une machine dans votre réseau local ou un routeur OPNsense et un serveur VPS. Cela vous permet d'ouvrir les ports nécessaires sur le VPS sans exposer votre réseau local, tou>

# Sauvegarder la configuration actuelle avant de la modifier
# Effectuez cette sauvegarde avant toute modification
# cp /etc/wireguard/wg0.conf /etc/wireguard/wg0.conf.bak

# Arrêter le service WireGuard avant de modifier la configuration
# sudo wg-quick down wg0

# Modifier la configuration de WireGuard
# Ouvrez le fichier de configuration pour le modifier
# nano /etc/wireguard/wg0.conf
# Supprimez la section entre 'PrivateKey = x' et '### Client opnsense' et rajouter cette configuration avec tout les commentaires pour vous aidez
# Remplacez PORT_WIREGUARD et IP_PUBLIQUE_DU_SERVEUR-VPS par le port WireGuard et l'adresse IP publique du VPS
```

```
# Remplacez ens2 par votre interface. Commande pour voir le nom : `ip link show`
# Il est important d'ouvrir en `PostUp` et en `PostDown` le port que vous avez ouvert.
# Pour un deuxième client, ajoutez les lignes et modifiez l'adresse IP locale : '10.66.66.3'
en prenant exemple sur '10.66.66.2'

# Règles iptables à appliquer après avoir configuré l'interface WireGuard
PostUp = sysctl -w net.ipv4.ip_forward=1  # Active le forwarding IPv4 pour permettre le
routage entre les interfaces
PostUp = iptables -I INPUT -p udp --dport PORT_WIREGUARD -j ACCEPT  # Autorise le trafic
entrant sur le port UDP de WireGuard (remplacez 'PORT_WIREGUARD' par le numéro de port réel)
PostUp = iptables -A FORWARD -i wg0 -o ens2 -j ACCEPT  # Autorise le trafic passant de
l'interface WireGuard (wg0) vers l'interface réseau (ens2)
PostUp = iptables -A FORWARD -i ens2 -o wg0 -m conntrack --ctstate ESTABLISHED,RELATED -j
ACCEPT  # Autorise les connexions établies et connexes à revenir sur l'interface WireGuard
PostUp = iptables -I FORWARD -i wg0 -s 10.66.66.0/24 -d 10.66.66.0/24 -j DROP  # Bloque le
trafic interne au sous-réseau WireGuard pour éviter les boucles ou les attaques internes

# Règles de translation d'adresse réseau (NAT) pour le trafic sortant et entrant
PostUp = iptables -t nat -A POSTROUTING -s 10.66.66.2/32 -o ens2 -j SNAT --to-source
IP_PUBLIQUE_DU_SERVEUR-VPS  # Effectue une translation d'adresse pour le trafic sortant du
client vers l'IP publique
PostUp = iptables -t nat -A PREROUTING -i ens2 -d IP_PUBLIQUE_DU_SERVEUR-VPS -p tcp --dport 80
-j DNAT --to-destination 10.66.66.2:80  # Redirige le trafic entrant vers l'adresse IP du
client WireGuard
PostUp = iptables -t nat -A PREROUTING -i ens2 -d IP_PUBLIQUE_DU_SERVEUR-VPS -p tcp --dport
443 -j DNAT --to-destination 10.66.66.2:443  # Redirige le trafic entrant vers l'adresse IP du
client WireGuard

# Règles iptables à retirer lors de la suppression de l'interface WireGuard
PostDown = iptables -D INPUT -p udp --dport PORT_WIREGUARD -j ACCEPT || true  # Supprime la
règle autorisant le trafic entrant sur le port UDP de WireGuard
PostDown = iptables -D FORWARD -i wg0 -o ens2 -j ACCEPT || true  # Supprime la règle
autorisant le trafic de wg0 vers ens2
PostDown = iptables -D FORWARD -i ens2 -o wg0 -m conntrack --ctstate ESTABLISHED,RELATED -j
ACCEPT || true  # Supprime la règle autorisant les connexions établies à retourner vers wg0
PostDown = iptables -D FORWARD -i wg0 -s 10.66.66.0/24 -d 10.66.66.0/24 -j DROP || true  #
Supprime la règle bloquant le trafic interne au sous-réseau WireGuard
PostDown = iptables -t nat -D POSTROUTING -s 10.66.66.2/32 -o ens2 -j SNAT --to-source
IP_PUBLIQUE_DU_SERVEUR-VPS || true  # Supprime la règle de NAT pour le trafic sortant du
client
```

```
PostDown = iptables -t nat -D PREROUTING -i ens2 -d IP_PUBLIQUE_DU_SERVEUR-VPS -p tcp --dport 80 -j DNAT --to-destination 10.66.66.2:80 || true  # Supprime la redirection du trafic entrant
PostDown = iptables -t nat -D PREROUTING -i ens2 -d IP_PUBLIQUE_DU_SERVEUR-VPS -p tcp --dport 443 -j DNAT --to-destination 10.66.66.2:443 || true  # Supprime la redirection du trafic entrant

# Redémarrer le service WireGuard après avoir modifié la configuration :
# sudo wg-quick up wg0

# Vérifiez l'état de WireGuard :
# sudo wg
```

6. Activez et démarrez le service resolvconf :

```
systemctl start resolvconf
systemctl enable resolvconf
systemctl status resolvconf
```

7. Mettez à jour le fichier /etc/resolvconf/resolv.conf.d/head pour qu'il contienne les lignes suivantes :

```
nameserver 9.9.9.9
nameserver 9.9.9.10
```

Exécutez la commande :

```
resolvconf --enable-updates
resolvconf -u
```

8. Pour que les changements prennent effet, vous devez redémarrer les services correspondants :

```
sudo systemctl restart resolvconf.service
sudo systemctl status resolvconf.service
```

Installation du client Wireguard sur le serveur dans votre réseau local

Ce guide fournit les instructions étape par étape pour installer le client généré précédemment sur un système Debian.

1. Mise à jour du Système

Mettez à jour les paquets du système :

```
sudo apt update && sudo apt upgrade -y
```

2. Installer WireGuard

Installez WireGuard à partir des dépôts de Debian :

```
sudo apt install wireguard curl resolvconf iptables -y
```

3. Configurer WireGuard

Créez un fichier de configuration pour l'interface WireGuard :

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

Collez la configuration que vous avez précédemment généré dans le serveur dans le fichier.

C'est optionnel, mais vous pouvez ajouter des règles pour empêcher les machines locales de communiquer avec la machine. Par exemple, dans l'exemple ci-dessous, j'ai ouvert les ports 22, 80 et 443 sur le réseau local tout en bloquant tout autre trafic entrant :

```
[Interface]
PrivateKey = VOTRE_CLE_PRIVEE
Address = 10.66.66.2/24     # IP locale pour l'interface WireGuard
ListenPort = 5678 # Le port WireGuard
DNS = 9.9.9.9,9.9.9.10

# Autoriser le trafic
PostUp = iptables -A INPUT -s 192.168.0.0/16 -p udp -m multiport --dports 22,80,443 -j ACCEPT
PostUp = iptables -A INPUT -s 192.168.0.0/16 -p tcp -m multiport --dports 22,80,443 -j ACCEPT

# Bloquer tout autre trafic entrant depuis le réseau local
PostUp = iptables -A INPUT -s 192.168.0.0/16 -j DROP

# Supprimer la règle autorisant le trafic
PostDown = iptables -D INPUT -s 192.168.0.0/16 -p udp -m multiport --dports 22,80,443 -j
ACCEPT
```

```
PostDown = iptables -D INPUT -s 192.168.0.0/16 -p tcp -m multiport --dports 22,80,443 -j ACCEPT

# Supprimer la règle bloquant tout autre trafic entrant depuis le réseau local
PostDown = iptables -D INPUT -s 192.168.0.0/16 -j DROP

[Peer]
PublicKey = CLE_PUBLIQUE_DU_SERVEUR
Endpoint = IP_PUBLIQUE_DU_SERVEUR:51820 # Le port WireGuard
AllowedIPs = 0.0.0.0/0
```

Ces règles visent à créer un kill switch afin d'empêcher la machine de communiquer sans VPN en cas de coupure, tout en autorisant l'accès à l'ensemble du réseau local :

```
PostUp = iptables -I OUTPUT ! -o %i -m mark ! --mark $(wg show %i fwmark) -m addrtype ! --dst-type LOCAL ! -d 192.168.0.0/16 -j REJECT && ip6tables -I OUTPUT ! -o %i -m mark ! --mark $(wg show %i fwmark) -m addrtype ! --dst-type LOCAL -j REJECT
PreDown = iptables -D OUTPUT ! -o %i -m mark ! --mark $(wg show %i fwmark) -m addrtype ! --dst-type LOCAL ! -d 192.168.0.0/16 -j REJECT && ip6tables -D OUTPUT ! -o %i -m mark ! --mark $(wg show %i fwmark) -m addrtype ! --dst-type LOCAL -j REJECT
```

4. Démarrer WireGuard

Mettez en route l'interface WireGuard :

```
sudo wg-quick up wg0
```

Activez WireGuard au démarrage :

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

5. Vérifier la Connexion

Vérifiez l'état de WireGuard :

```
sudo wg
```

6. Tester l'IP Publique

Vérifiez votre adresse IP publique en utilisant `curl` :

```
curl ifconfig.me
```

7. Arrêter WireGuard

Fermez l'interface :

```
sudo wg-quick down wg0
```

Astuce

Si vous avez besoin, dans le futur, d'ouvrir un port et que vous ne pouvez pas vous permettre d'éteindre le service WireGuard qui coupe tout le traffic, modifiez le fichier de configuration pour ajouter vos nouvelles règles pour ouvrir un port ou supprimez `PostUp =` ou `PostDown =` pour fermer le port. Ensuite, il suffit d'ouvrir le port en retirant `PostUp =` ou `PostDown =` (pour fermer le port), ce qui permet de ne pas avoir d'interruption et de faire la commande dans le terminal, par exemple :

```
iptables -t nat -D PREROUTING -i ens2 -d IP_PUBLIQUE_DU_SERVEUR-VPS -p tcp --dport 443 -j DNAT
--to-destination 10.66.66.2:443 || true  # Supprime la redirection du trafic entrant
```

Révision #33
Créé 2025-03-27 15:37:58 CET par Philippe Favre
Mis à jour 2025-10-03 18:29:08 CEST par Philippe Favre