WIREGUARD

Table des matières

Auteur(s)	Relu, validé et visé par :		Date de création :	Date dernière MAJ :
Maxime Przybylo	Jérôme CHRETIENNE : **Resp. Secteur Tertiaire & Numérique**		18/04/2022	18/04/2022
	Florence CALMETTES : **Coordinatrice Filière Syst. & Réseaux**			
	Sophie POULAKOS : **Coordinatrice Filière WEBDESIGN / PPNUM**			
	Marc CECCALDI : **Coordinateur Filière Développement**			

WIREGUARD

Installation de wireguard

Le paquet wireguard est directement disponible sur les version Debian 11, cependant pour les versions antérieures il faut ajouter le rétroportage des paquets.

Allez dans le dossier /etc/apt/sources.list.d et créez le fichier buster-backports.list

Inscrivez y dedans la ligne :

```
deb http://deb.debian.org/debian buster-backports main contrib non-free
```

Méttez ensuite à jours la liste des paquets presents sur vos dépots

```
apt update
```

Ensuite, installez wireguard et les outils nécessaires :

```
apt install wireguard wireguard-tools net-tools linux-headers-`uname -r`
```

Générer les clés de serveur

Ensuite, vous devez générer les couples de clés privées et publiques qu'utilisera votre VPN. Pour cela, nous allons utiliser les outils fournis avec les paquet wireguard. Pour cette partie connectez-vous en root afin d'effectuer les commandes suivantes :

```
cd /etc/wireguard

umask 077; wg genkey | tee privatekey | wg pubkey > publickey
```

Décortiquons un peu cette commande :

Ellse se décompose enfait en 5 étapes. Tout dabord, avec l'instruction *umask 077* nous indiquons que les fichiers créés seront accessibles en lecture et écriture uniquement pour l'utilisateur propriétaire. La commande *wg genkey* permet de générer une clef privée. Nous redirigeons le résultat de cette commande, par l'utilisation du pipe (symbole "|"), à nos commandes suivantes. La commande *tee privatekey* permet d'envoyer le texte redirigé dans le fichier privatekey. La commande *wg pubkey* permet de générer une clef publique par rapport à la clef privée redirigée. Enfin, le symbole ">" permet lui aussi d'envoyer le résultat d'une commande, en l'occurrence la génération de la clef publique, dans un fichier.

Auteur(s)	Relu, validé et visé par :		Date de création :	Date dernière MAJ :
Maxime Przybylo	Jérôme CHRETIENNE : **Resp. Secteur Tertiaire & Numérique**		18/04/2022	18/04/2022
	Florence CALMETTES : **Coordinatrice Filière Syst. & Réseaux**		*Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.*	
	Sophie POULAKOS : **Coordinatrice Filière WEBDESIGN / PPNUM**			
	Marc CECCALDI : **Coordinateur Filière Développement**			

Vous pouvez visualiser les fichier créés en effectuant un *ls -al* dans le répertoire

```
root@coursdevops:/etc/wireguard# ls -al
total 16
drwx------ 2 root root 4096 avril 19 16:19 .
drwxr-xr-x 65 root root 4096 avril 19 14:40 ..
-rw------- 1 root root 45 avril 19 10:00 privatekey
-rw------- 1 root root 45 avril 19 10:00 publickey
```

Vous pouvez également visualisez vos clef avec la commande *cat* :

```
root@coursdevops:/etc/wireguard# cat privatekey
mFR97cSISl6yL9PNZjFaaNOSM6HcJX0fzqXewv7wX3k=

root@coursdevops:/etc/wireguard# cat publickey
U5s9bprJiPBRFcugaKFaA/SwI8YPW4vNAb75DYONBnU=
```

Générer la configuration du serveur

Maintenant que vos clefs de chiffrement sont créées, vous pouvez rédiger le fichier de configuration de votre serveur. Créez maintenant le fichier wg0.conf. Dans ce fichier nous devrons indiquer la configuration de l'interface virtuel que wireguard va créer et utiliser comme pont pour les clients qui se connecterons. Cette interface aura besoin d'une adresse, d'un port d'écoute, d'une clef privée et de règles de pare-feu.

```
[Interface]
Address = 192.168.10.1/24
ListenPort = 51820
PrivateKey = mFR97cSISl6yL9PNZjFaaNOSM6HcJX0fzqXewv7wX3k=


PostUp = iptables -A FORWARD -I wg0 -j ACCEPT; iptables -t nat -A
POSTROUTING -o enp0s3 -j MASQUERADE
PostDown = iptables -D FORWARD -I wg0 -j ACCEPT; iptables -t net -D
POSTROUTING -o enp0s3 -j MASQUERADE
```

Nous avons donc défini l'adresse 192.168.10.1 pour l'interface virtuelle de wireguard. Le sous réseau 192.168.10.0/24 sera donc utilisé pour nos clients. Etant donné que ce sous réseau est réservé pour notre tunnel VPN il doit être différent de l'adresse de l'interface physique de notre serveur.

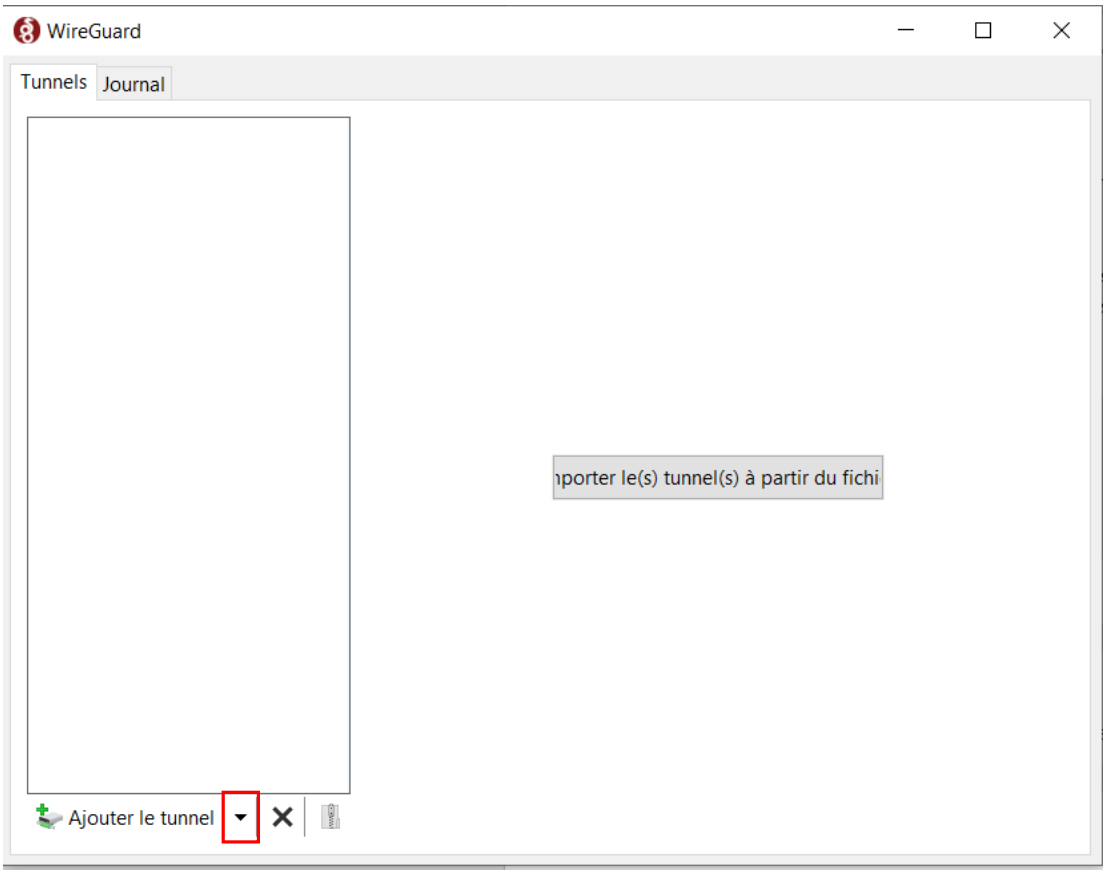
WIREGUARD

Gestion des clients

La configuration de nos clients VPN se passe en 2 étapes : tout d'abord installer le logiciel qui va permettre à nos clients de se connecter. Ce logiciel (ou paquet pour les clients linux) va nous permettre de générer les clefs privée et publiques pour les clients. Une fois que le couple de clefs client est mis en place nous pouvons démarrer la deuxième étape : inscrire les clients et leur clef publique dans la configuration du serveur.

Client Windows

Sur windows vous pouvez télécharger le client Wireguard. Une fois installé il vous faut cliquer sur la flèche à côté du bouton "ajouter le tunnel" en bas de la fenêtre et sélectionner ensuite "Ajouter un tunnel vide".



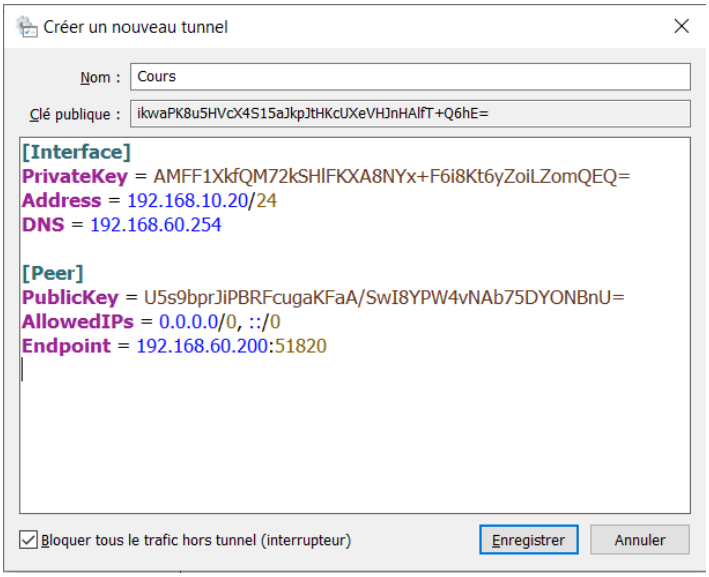
Sur la nouvelle fenêtre apparait un début de configuration pour votre client, notamment la clef publique et la clef privée. Sauvegardez la clef publique, vous en aurez besoin par la suite. La section Interface est déjà commencé, elle correspond à la configuration réseau du client sur le réseau VPN. Une autre section Peer sera à remplir et correspondra à la configuration réseau du serveur VPN. Dans la section Interface doit figurer la clef privée du client, son adresse sur le réseau VPN et l'adresse du serveur DNS.

```
[Interface]
PrivateKey = AMFF1XkfQM72kSHlFKXA8NYx+F6i8Kt6yZoiLZomQEQ=
Address = 192.168.10.20/24
DNS = 192.168.60.254
```

Dans la section Peer doit figurer la clef publique du serveur VPN, la liste des sous réseaux qu'on veut router par le tunnel, en l'occurrence tous, et l'adresse et le port du serveur VPN (dans notre cas nous mettrons l'adresse de la carte éthernet physique de notre serveur mais en réalité nous devrions indiquer l'adresse publique du réseau de notre serveur VPN).

```
[Peer]
PublicKey = U5s9bprJiPBRFcugaKFaA/SwI8YPW4vNAb75DYONBnU=
AllowedIPs = 0.0.0.0/0, ::/0
Endpoint = 192.168.60.200:51820
```

Quand vous remplissez entièrement votre section, un bouton "bloquer tous le traffic hors tunnel (interrupteur)". Lorsqu'il est coché ce bouton fait en sorte que tout le traffic du client passe uniquement par le tunnel VPN.



Client Linux

Sur linux, en tant que client, il faut installer les paquets wireguard et resolvconf.

```
sudo apt update
sudo apt install wireguard resolvconf
```

Une fois le paquet installé il faudra, comme pour la configuration du serveur créer les clefs privées et publiques ainsi que le fichier de configuration de l'interface virtuelle.

```
cd /etc/wireguard
umask 077; wg genkey | tee privatekey | wg pubkey > publickey
```

Les fichiers contenants les clefs publiques et privées sont donc créés dans le dossier wireguard. Nous pouvons maintenant créer le fichier wg0.conf pour y indiquer une configuration similaire à celle faite sur le client windows.

```
[Interface]
PrivateKey = PFDI8eu932JGFHSOCB83ejdEKG+skjfdsLDOZUsl99L=
Address = 192.168.10.30/24
DNS = 192.168.60.254

[Peer]
PublicKey = U5s9bprJiPBRFcugaKFaA/SwI8YPW4vNAb75DYONBnU=
AllowedIPs = 0.0.0.0/0, ::/0
Endpoint = 192.168.60.200:51820
```

Avec ce fichier de configuration notre client Linux est prêt à se connecter mais il reste encore à les renseigner dans la configuration de notre serveur.

Serveur Wireguard

Pour ajouter nos clients à la configuration de notre serveur il faut rajouter un section Peer par client dans le fichier de configuration de l'interface de notre serveur /etc/wireguard/wg0.conf. Pour chaque client, ou chaque peer, il faudra renseigner sa clef publique ainsi que l'adresse ip qui lui sera attribué sur le sous réseau VPN.

```
[Peer]
PublicKey = ikwaPKK8u5HVcX4S15aJkpJtHKcUXeVHJnHAlfT+Q6hE=
AllowedIPs = 192.168.10.20/32

[Peer]
PublicKey = kjdjfhsLKJLKJDGDSSjhfdQ498dfhsklkjlkjfsdrç+f=
AllowedIPs = 192.168.10.30/32
```

Avec ces deux enregistrements votre configuration est terminée. Vous pouvez maintenant démarrer votre interface avec la commande :

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

Cette même commande est à lancer sur le client Linux pour se connecter. Pour le client windows il suffit de cliquer sur le bouton activer pour se connecter.

Sur le serveur, vous pouvez tester que vos clients sont bien connectés avec la commande wg. Cette commande vous affiche un résumé de votre configuration wireguard. Si des clients sont connectés vous voyez de quand date la dernière vérification de connexion (le handshake) et quelle quantité de donnée est en train d'être transféré.

Client Mobiles

Pour les clients mobiles (téléphones, tablettes) wireguard dispose d'une application gratuite téléchargeable sur les stores. Comme pour les clients linux et windows il y aura besoin d'un jeu de clés privés/publiques et d'un fichier de configuration d'interface virtuelle.

Sur le serveur VPN créez dans votre répertoire personnel un dossier appelé clients_vpn. Une fois dedans générez, comme vous l'avez fait pour le client linux, les éléments nécessaires à la création du fichier d'interface que vous appellerez mobile.conf . N'oubliez pas par la suite de rajouter votre nouveau client au fichier de configuration de l'interface virtuelle de votre serveur.

Sur l'application mobile wireguard vous avez trois possibilités, soit chargé un fichier de configuration. Soit rédiger le fichier de configuration. Soit scanner un QR code. Les deux premières méthodes sont assez simples à mettre en œuvre. Linux possède un paquet permettant de convertir de fichiers en QR codes. Pour se faire vous avez besoin du paquet qrencode.

```
sudo apt install qrencode
```

Vous pouvez maintenant taper la commande qui permet de générer un QR code

```
qrencode -t ansiutf8 < mobile.conf
```

Votre QR code est généré et apparait sur votre écran. Vous pouvez sauvegarder ce QR code dans un fichier png en spécifiant l'option –o <nom_de_fichier>.