A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the date.

01/05/2023

A thin purple horizontal bar is at the top right. Below it is a yellow logo consisting of a stylized 'L' shape.

Mise en place d'un VPN itinérant

Atelier

Several thin, curved lines in dark blue and light grey originate from the bottom left and sweep upwards and to the right.

DEUSCHER Lucas

Table des matières

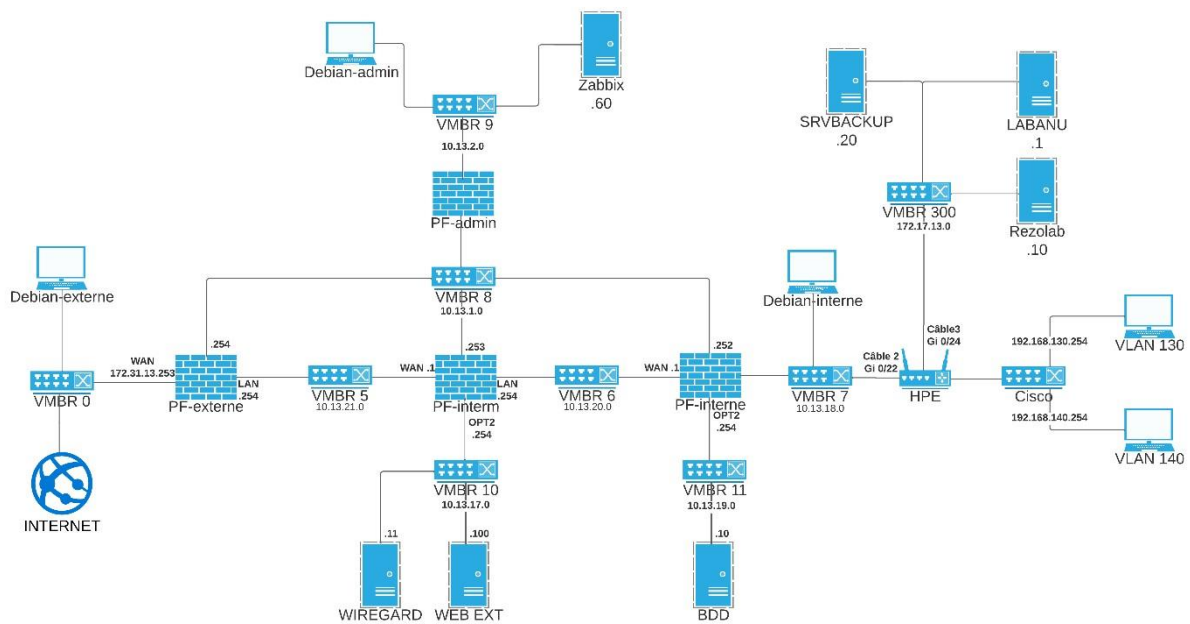
Travail à faire :	3
Installation et Configuration du serveur	
WireGuard :	4
1. Installation de WireGuard	4
2. Génération des clés	5
3. Création du fichier de configuration	5
4. Activation du routage et configuration du pare-feu	7
Configuration du Client Wireguard :	9
5. Configuration des PF-Sense	12
Tests et validation :	13

Travail à faire :

Dans ce TP, nous allons créer un VPN afin qu'un utilisateur distant puisse accéder à ses fichiers depuis chez lui.

Pour cela, nous allons mettre en place un serveur VPN WireGuard qui tourne sur du Debian 11.

Voici le schéma final :



Installation et Configuration du serveur WireGuard :

Avant de commencer, il nous faut une Debian 11 installée et configurée.

Dans notre exemple, la machine virtuelle **WireGuard** a comme IP **10.13.17.11**. Il est fortement conseillé de mettre en place un accès SSH, mais pour cela, il faudra créer les routes depuis **PF-Admin** et passer par **PF-Interm**.

Dans mon cas, les routes étaient déjà créées grâce à un autre TP. (N'oubliez pas qu'il faut également configurer les règles SSH sur **PF-Admin** et **PF-Interm**.)

1. Installation de WireGuard

Tout d'abord, nous mettons à jour les paquets :

- `apt-get update`

Puis, nous installons WireGuard :

- `apt-get install wireguard`

La partie logicielle de WireGuard est installée, mais il faut encore générer des clés pour assurer la sécurité.

2. Génération des clés

Nous générons une clé privée et une clé publique :

- `wg genkey | sudo tee /etc/wireguard/wg-private.key | wg pubkey | sudo tee /etc/wireguard/wg-public.key`

La clé publique sera affichée dans la console. Nous devons ensuite ajouter notre clé privée dans le dossier de WireGuard :

- `sudo cat /etc/wireguard/wg-private.key`

3. Création du fichier de configuration

Nous créons un fichier de configuration :

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

Dans ce fichier il faut rajouter le contenu suivant (on viendrait le compléter par la suite)

- `[Interface]`
- `Address = 10.7.0.1`
- `SaveConfig = true`
- `ListenPort = 51820`
- `PrivateKey = <clé privée du serveur>`

La section **[Interface]** sert à déclarer la partie serveur. Voici quelques informations :

Address : Adresse IP de l'interface WireGuard au sein du tunnel VPN (différente du LAN distant).

SaveConfig : La configuration est sauvegardée automatiquement et protégée tant que l'interface est active.

ListenPort : Port d'écoute de WireGuard (par défaut **51820**, mais il est conseillé de le personnaliser).

PrivateKey : Clé privée du serveur (**wg-private.key**).

Nous démarrons maintenant l'interface avec :

- `sudo wg-quick up wg0`

Nous vérifions ensuite si tout est correct :

- `ip a`

Voici le résultat :

```
valid_lft forever preferred_lft forever
3: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noqueue state UNKNOWN
up default qlen 1000
    link/none
    inet 10.7.0.1/24 scope global wg0
        valid_lft forever preferred_lft forever
```

Nous allons vérifier la config de wg0 :

- `sudo wg show wg0`

Voici le résultat :

```
interface: wg0
  public key: byF3zSV9rMrqlgR3PFaOFAq6G8SpSit+9k7ePb9y8B8=
  private key: (hidden)
  listening port: 51820
```

Pour que l'interface wg0 soit active au démarrage, nous exécutons :

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

4.Activation du routage et configuration du pare-feu

Nous activons l'ip forwarding afin que la Debian puisse router les paquets entre les différents réseaux comme un routeur :

- `sudo nano /etc/sysctl.conf`

Nous ajoutons à la fin du fichier :

- `net.ipv4.ip_forward = 1`

Il faut activer l'IP Masquerade. Pour faire simple, cela revient à activer le NAT sur Debian (comme un pare-feu Linux). Nous allons utiliser la commande UFW, qui doit d'abord être installée :

- `apt install ufw`

Nous allons exécuter deux commandes : la première pour autoriser le SSH et la seconde pour activer le port 51820 (utilisé par WireGuard) :

- `sudo ufw allow`
`22/tcp`

Puis

- `sudo ufw allow 51820/udp`

Ensuite il nous faut le nom de l'interface de base de debian on la trouve grâce la commande ip a :

```
valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP g
rroup default qlen 1000
    link/ether 92:57:23:00:00:00 brd ff:ff:ff:ff:ff:ff
```

Grace a cette information, nous éditons le fichier suivant :

- `nano /etc/ufw/before.rules`

Et ajoutons à la fin du fichier :

- # NAT - IP masquerade
- *nat
- :POSTROUTING ACCEPT [0:0]
- -A POSTROUTING -o ens18 -j MASQUERADE
- # End each table with the 'COMMIT' line or these rules won't be processed
- COMMIT

Toujours dans le même fichier, nous allons déclarer le réseau interne de l'entreprise. Dans mon cas, je vais l'adapter pour un seul hôte, mais cela fonctionne de la même manière pour un réseau entier en adaptant le /32 en /24 ou autre.

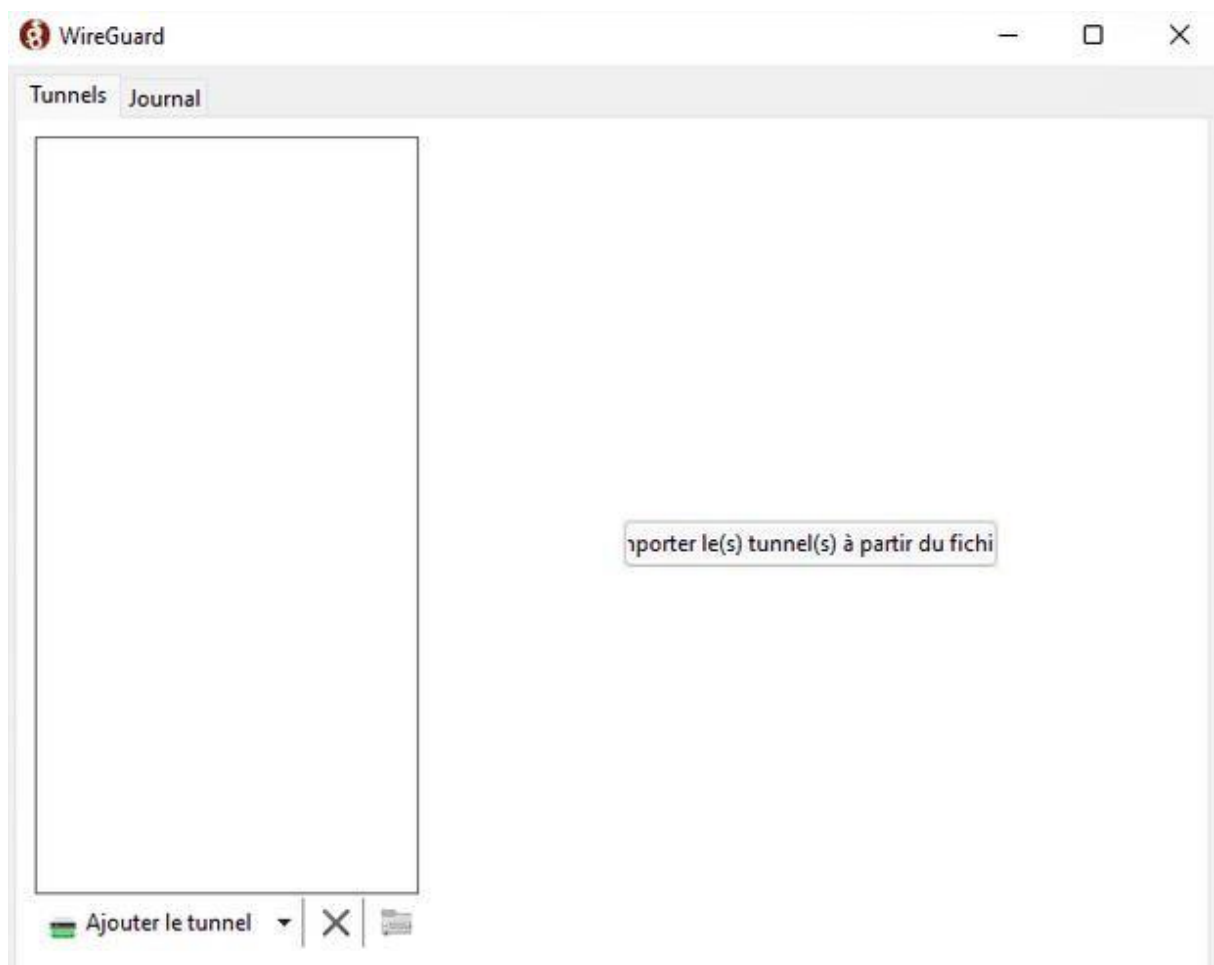
- # autoriser le forwarding pour le réseau distant de confiance (+ le réseau du VPN)
- -A ufw-before-forward -s 172.17.13.1/32 -j ACCEPT
- -A ufw-before-forward -d 172.17.13.1/32 -j ACCEPT
- -A ufw-before-forward -s 10.13.17.11/32 -j ACCEPT
- -A ufw-before-forward -d 10.13.17.11/32 -j ACCEPT
- -A ufw-before-forward -s 10.7.0.2/32 -j ACCEPT
- -A ufw-before-forward -d 10.7.0.2/32 -j ACCEPT

Nous appliquons les modifications et redémarrons les services :

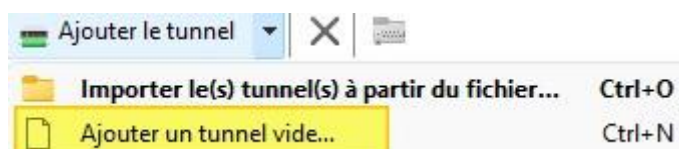
- sudo ufw enable
- sudo
- systemctl restart
- ufw

Configuration du Client Wireguard :

Le client wireguard doit être téléchargé sur site officiel et installé :



Nous allons créer un nouveau tunnel vide il faut faire « Ajouter le tunnel » puis « Ajouter un tunnel vide... » :



Nous allons créer un peer (pair) sur un serveur distant. Ce qu'il faut savoir, c'est que WireGuard recherche deux blocs de configuration : l'un nommé "[Interface]" et l'autre "[Peer]". Les crochets sont très importants. Je vais vous fournir ces deux blocs ; vous n'aurez qu'à remplacer les valeurs par les vôtres.

- [Interface]
- PrivateKey =
OP6dceP4+C5QwSFXg0uXcQ2PiLG9gJpgTW1Hte+4q2s=
- Address = 10.7.0.2/24
- DNS = 8.8.8.8

- [Peer]
- PublicKey =
byF3zSV9rMrglgR3PFaOFAq6G8SpSit+9k7ePb9y8B8=
- AllowedIPs = 10.7.0.2/24, 10.13.17.11/32, 172.17.13.1/32
- Endpoint = 172.31.13.253:51820

Explication des paramètres :

PublicKey : Il s'agit de la clé publique du serveur WireGuard sous Debian 11 (vous pouvez l'obtenir avec la commande `sudo wg`).

AllowedIPs : Liste des adresses IP ou sous-réseaux accessibles via ce réseau VPN WireGuard. Ici, il s'agit du sous-réseau du VPN WireGuard (**10.7.0.2/24**) et du LAN distant (**172.17.13.1/32**).

Endpoint : Adresse IP de l'hôte Debian 11, qui sert de point de liaison WireGuard (il faudra préciser son adresse IP publique).

Nous devons maintenant ajouter ce client au serveur Debian.
Pour cela, nous stoppons l'interface wg0 :

- `sudo wg-quick down /etc/wireguard/wg0.conf`

Nous modifions le fichier wg0.conf du serveur :

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

Nous ajoutons le bloc suivant :

- `[Peer]`
- `PublicKey =`
`PbkwKFpLaqXINQWeu7ycaWz0dRsA3OCyu4j5p6EGNTA=`
- `AllowedIPs = 10.7.0.2/32`

Ce bloc contient la clé publique de win 10 ainsi que l'adresse ip de son interface allouée à lui.

Il ne reste plus qu'à sauvegarder le fichier et à relancer l'interface « wg0 » :

- `wg-quick up /etc/wireguard/wg0.conf`

Afin de garantir le bon fonctionnement on procède à une vérification :

```
interface: wg0
  public key: byF3zSV9rMrqlgR3PFaOFAq6G8SpSit+9k7ePb9y8B8=
  private key: (hidden)
  listening port: 51820

peer: PbkwKFpLaqXINQWeu7ycaWz0dRsA3OCyu4j5p6EGNTA=
  preshared key: (hidden)
  endpoint: 172.31.13.110:55532
  allowed ips: 10.7.0.2/32
```




Enfin nous sécurisons les fichiers de configuration :

- `sudo chmod 600 /etc/wireguard/ -R`

5. Configuration des PF-Sense

Nous devons maintenant configurer **PF-Externe** et **PF-Interm** pour permettre le passage des paquets WireGuard. Deux règles de NAT ont été créés :

Pour la **PF-externe** :

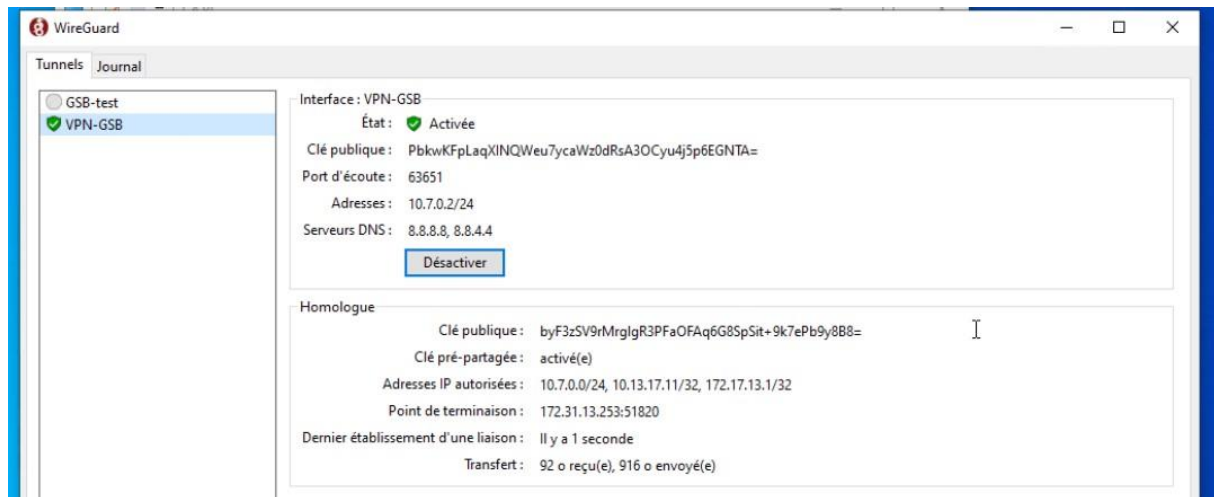
	Interface	Protocol	Source Address	Source Ports	Dest. Address	Dest. Ports	NAT IP	NAT Ports	Description	Actions
<input type="checkbox"/>	WAN	TCP/UDP	*	*	WAN address	51820	10.13.21.1	51820	WIREGUARD	  

Pour la **PF-interm** :

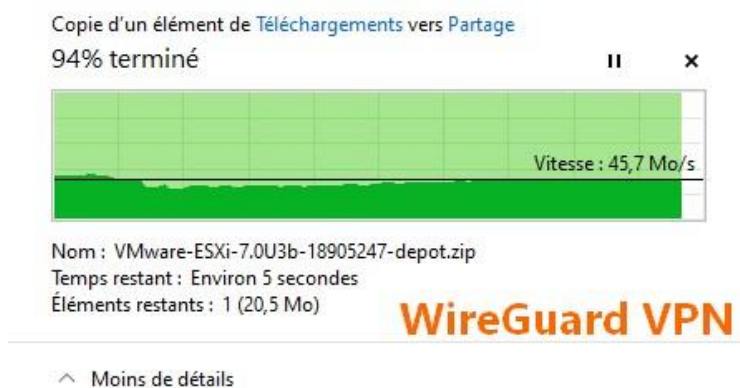
	Interface	Protocol	Source Address	Source Ports	Dest. Address	Dest. Ports	NAT IP	NAT Ports	Description	Actions
<input type="checkbox"/>	WAN	TCP/UDP	*	*	WAN address	51820	10.13.17.11	51820	WIREGUARD	  

Tests et validation :

Nous activons le VPN et vérifions la connectivité :



Un test de transfert de fichiers permet d'évaluer les performances :



Une comparaison avec OpenVPN :

