CPAM Ardèche / Greta Ardèche-Drôme

VPN pour accès à distance

TP

Collet Valentin
BTS SIO-SISR / Session 2026

SOMMAIRE

Cahier des charges 2
Descriptifs de l'existant 2
Besoins 2
Contraintes 3
Ressources 3
Analyse 5
Descriptif des solutions 5
Comparaison des solutions : 7
Choix d'une solution 7
Plan d'adressage et schéma AD 8
Etude de l'impact sur le SI existant 8
Déploiement et prévision des tests 10
Mise en place 10
Rapport de tests 15
Rapport de déploiement 15
Bilan 16

Cahier des charges

Descriptifs de l'existant

Je dispose dans le cadre de ce TP d'un accès aux machines du GRETA avec un hyperviseur type 2 : VMWARE Workstation. En ce qui concerne les machines, la mienne est équipée de :

- CPU : Intel I7-10700 @ 2.90GHz
- 32 Go RAM DDR4
- 500 Go SSD
- 2 cartes réseaux

De plus, je dispose au préalable déjà de l'ISO nécessaire pour faire une installation de Debian 13, tandis que j'utiliserais un master Windows 11 pour simuler un poste de travail distant. Enfin j'utiliserais un Windows serveurs 2025 pour avoir un AD pour faire des test d'accès à distance.

Enfin je dispose d'un accès au réseau GRETA avec comme plage :

Réseau pédagogique	
Réseau	172.26.0.0/21 (255.255.248.0)
Plage DHCP	172.26.1.0 – 172.26.2.254
IP fixe Vivarais	172.26.4.1 – 172.26.4.254
Passerelle	172.26.7.254
DNS	8.8.8.8

Besoins

Le besoin principal est l'installation d'une solution VPN (Virtual Private Network) permettant un accès distant sécurisé aux ressources internes du réseau. Cette solution devra permettre aux utilisateurs nomades de se connecter de manière chiffrée au réseau interne depuis l'extérieur, tout en garantissant la sécurité et la confidentialité des données échangées.

Infrastructure nécessaire :

- Un serveur Debian 13 avec WireGuard installé :
  - 2 GB de RAM suffisant
  - 1 cœur 2 threads pour le CPU
  - Espace disque de 10 GB suffisant
- Un firewall pfSense configuré avec :
  - Règles de pare-feu adaptées pour Wire Guard
  - Port forwarding NAT du port 51820/UDP depuis le pfSense vers le serveur Debian WireGuard
- Un poste client Windows 11 avec l'application WireGuard pour tester la connexion VPN
- Un serveur Windows Server 2025 pour des tests complémentaires sur le réseau interne

Configuration réseau VPN :

- Réseau VPN : 10.0.0.0/24
- IP serveur WireGuard (interface VPN) : 10.0.0.1
- Plage IP clients VPN : 10.0.0.2 – 10.0.0.254 (/32)
- Port d'écoute WireGuard : 51820/UDP

Fonctionnalités requises :

- Chiffrement de bout en bout des communications
- Authentification par clés cryptographiques (publique/privée)
- Accès transparent aux ressources du réseau interne (172.26.0.0/21), et de maintenir un accès au WAN pour les clients nomades.
- Possibilité de désactiver/activer facilement la connexion VPN côté client
- Logs de connexion pour le suivi et la sécurité

Contraintes

Plusieurs contraintes doivent être prises en compte dans le cadre de ce déploiement :

⇨ **Contraintes temporelles :**

Je dispose de 1 TP de 8H pour réaliser l'installation complète de la solution VPN.

⇨ **Contraintes réseau :**

La configuration du port forwarding sur le pfSense pour rediriger le trafic entrant du port 51820/UDP vers le serveur WireGuard.

L'isolation du trafic VPN du reste du réseau local pour des raisons de sécurité.

⇨ **Contraintes de sécurité :**

La génération de clés cryptographiques sécurisées pour chaque client ainsi que la protection des clés privées (il ne faut jamais les transmettre en clair)

La mise en place de règles de pare-feu strictes pour limiter les accès uniquement aux ressources autorisées

⇨ **Contraintes matérielles :**

L'accès aux ordinateurs du GRETA uniquement sur site, sans possibilité de travailler à distance (sauf export/import de VM).

Ressources

Je dispose de plusieurs ressources pour mener à bien ce projet :

⇨ Ressources matérielles :

Un poste de travail avec hyperviseur VMware Workstation (type 2)

Voir descriptif de l'existant pour les caractéristiques détaillées.

⇨ Ressources humaines :

L'intervenant en cours est une personne ressource pouvant aiguiller et conseiller en cas de questionnement ou de blocage technique. De plus, mes camarades de classes peuvent aussi être ressource dans l'idée ou nous nous entraidons et pouvons discuter de caractéristique technique.

⇨ Ressources documentaires :

- Documentation officielle Wireguard
- Guide d'installation
- Protocole Wireguard
- Référence de la commande wg : man wg / man wg-quick
- Cours sur les VPN et les architectures réseau sécurisées dispensés en BTS SIO
- Documentation pfSense pour la configuration du NAT et des règles de pare-feu

⇨ Au niveau de Wireguard :

⇨ Sécurité :

Curve25519 : Échange de clés : est un algorithme de cryptographie. Son rôle est de permettre à deux parties d'établir un secret partagé sur un canal non sécurisé.

- Très rapide (optimisée pour les processeurs modernes)
- Niveau de sécurité équivalent à une clé RSA de 3072 bits

ChaCha20 - Chiffrement symétrique : une fois le secret partagé établi, ChaCha20 chiffre les **données réelles** qui transitent dans le tunnel VPN.

Poly1305 - Authentification des messages (MAC) : Garantir que les données n'ont pas été modifiées ou forgées pendant la transmission.

BLAKE2s - Fonction de hachage : Générer des empreintes cryptographiques et dériver des clés.

⇨ Architecture du protocole

WireGuard fonctionne en **deux phases** :

- Phase 1 : Handshake (établissement de la connexion) pour établir des clés de session partagées et authentifier mutuellement les pairs.

Le processus en 3 étapes (1-RTT handshake) : paquet d'initiation (client -> serveur) ; paquet de réponse (serveur -> client) ; données chiffrées (client <--> serveur)

- Phase 2 : transport de données : une fois le handshake réussi, toutes les données sont encapsulées

Analyse

Descriptif des solutions

⇨ WireGuard

WireGuard est une solution VPN, open-source (donc gratuite) et particulièrement performante. De base Wireguard est un protocole, il a été intégré au noyau Linux depuis, ce qui témoigne de sa maturité et de sa fiabilité et surtout facilite des performances importantes, car intégré au noyau. WireGuard se distingue par sa simplicité architecturale, ce qui facilite grandement les audits de sécurité et réduit la surface d'attaque.

La solution utilise des algorithmes cryptographiques modernes et éprouvés : Curve25519 pour l'échange de clés, ChaCha20 pour le chiffrement symétrique, Poly1305 pour l'authentification des messages, et BLAKE2s pour le hachage. Ce stack cryptographique a été choisie pour sa sécurité, mais aussi pour ses performances exceptionnelles, particulièrement sur les architectures modernes.

WireGuard fonctionne au niveau 3 (couche réseau) du modèle OSI et crée une interface réseau virtuelle (généralement wg0) qui encapsule le trafic IP. Le protocole utilise UDP pour le transport, ce qui lui confère une meilleure performance que les solutions basées sur TCP, particulièrement dans des conditions réseau difficiles. Le port par défaut est 51820/UDP, mais il peut être modifié selon les besoins.

L'un des atouts majeurs de WireGuard est sa gestion des clés : chaque pair (client ou serveur) possède une paire de clés publique/privée. L'authentification se fait par échange de clés publiques, sans nécessiter de PKI (Public Key Infrastructure) complexe comme avec IPSec. Cette approche simplifie considérablement le déploiement et la maintenance.

WireGuard implémente également un mécanisme de "roaming" transparent : si l'IP publique d'un client change (passage du Wi-Fi à la 4G par exemple), la connexion VPN se maintient automatiquement sans interruption. Le protocole est également "stealthy" par défaut : un serveur WireGuard ne répond pas aux paquets non authentifiés, ce qui le rend difficile à détecter et à attaquer.

En termes de performances, WireGuard offre des débits nettement supérieurs à OpenVPN (souvent 3 à 5 fois plus rapides) tout en consommant moins de ressources CPU grâce à son intégration. La latence est également réduite, ce qui améliore l'expérience utilisateur, particulièrement pour les applications temps réel.

Cependant, WireGuard présente quelques limitations : il n'intègre pas nativement de gestion dynamique des IP (DHCP-like), ce qui nécessite une attribution manuelle des adresses IP. Il ne dispose pas non plus de mécanisme d'authentification par utilisateur/mot de passe (uniquement par clés), ce qui peut complexifier la gestion dans de très grandes organisations. Enfin, la révocation de clés nécessite une modification de la configuration côté serveur.

⇨ OpenVPN

OpenVPN est une solution VPN open-source éprouvée. C'est l'une des solutions VPN les plus déployées au monde, particulièrement dans les environnements d'entreprise. OpenVPN se base sur la bibliothèque OpenSSL pour le chiffrement et peut fonctionner en mode TCP ou UDP.

La solution offre une flexibilité importante : elle supporte plusieurs méthodes d'authentification (certificats X.509, clés pré-partagées, authentification par login/mot de passe via LDAP/AD), peut fonctionner en mode site-à-site ou client-serveur, et propose deux modes de tunneling : TAP (niveau 2) pour simuler un réseau Ethernet complet, ou TUN (niveau 3) pour un routage IP pur.

OpenVPN dispose d'une infrastructure PKI complète permettant de gérer des certificats, de les révoquer facilement et d'implémenter une authentification forte à deux facteurs. La configuration se fait via des fichiers texte (.ovpn) qui peuvent être générés centralement et distribués aux utilisateurs.

L'écosystème OpenVPN est très riche : clients disponibles sur toutes les plateformes (Windows, Linux, macOS, iOS, Android), outils de gestion graphique (OpenVPN Access Server pour l'édition commerciale), intégration native dans de nombreux routeurs et pare-feu.

Cependant, OpenVPN présente des inconvénients notables : la configuration est complexe, particulièrement pour les débutants. Les performances sont inférieures à WireGuard, avec une consommation CPU plus importante et des débits plus faibles.

- ⇨ **IPSec** (avec IKEv2)

IPSec (Internet Protocol Security) est une suite de protocoles standardisés par l'IETF pour sécuriser les communications IP. C'est la solution la plus ancienne et la plus standardisée, intégrée nativement dans la plupart des systèmes d'exploitation modernes. IKEv2 (Internet Key Exchange version 2) est le protocole de négociation des clés associé, définit dans la RFC 7296.

IPSec fonctionne directement au niveau de la couche réseau (niveau 3 OSI) et peut opérer en deux modes : mode transport (chiffre uniquement la charge utile) ou mode tunnel (chiffre le paquet IP complet). Il utilise deux protocoles principaux : AH (Authentication Header) pour l'authentification et l'intégrité, et ESP (Encapsulating Security Payload) pour le chiffrement et l'authentification.

L'un des atouts majeurs d'IPSec est sa standardisation : étant défini par des RFC de l'IETF, il garantit une interopérabilité entre équipements de différents constructeurs. Il est nativement supporté par Windows, macOS, iOS et la plupart des distributions Linux, sans nécessiter d'installation de logiciel tiers.

IPSec/IKEv2 offre également des fonctionnalités avancées : possibilité d'utiliser des algorithmes cryptographiques variés selon les besoins (AES, ChaCha20, etc.).

Cependant, IPSec présente plusieurs inconvénients majeurs : la configuration est complexe, nécessitant une expertise approfondie des protocoles réseau et de la cryptographie. La négociation IKE peut être lente, particulièrement lors de l'établissement initial de la connexion.

Comparaison des solutions :

Critères	WireGuard	OpenVPN	IPSec/IKEv2
Performances	Excellentes : débit élevé, faible latence, consommation CPU minimale	Moyennes : débit correct mais inférieur à WireGuard, consommation CPU plus élevée	Bonnes avec accélération matérielle, moyennes en logiciel
Sécurité	Excellente : cryptographie moderne, code minimal facilement auditable, intégré au noyau Linux	Très bonne : mature et éprouvé, nombreux audits, base de code volumineuse	Excellente : standardisé IETF, utilisé par les gouvernements, complexité élevée
Simplicité de configuration	Très simple : fichiers de configuration minimalistes, génération de clés en une commande	Complexe : nombreux paramètres, gestion PKI, fichiers de configuration verbeux	Très complexe : paramètres cryptiques, négociation IKE difficile à maîtriser
Compatibilité multi-plateformes	Excellente : clients officiels pour Windows, Linux, macOS, iOS, Android	Excellente : clients matures sur toutes les plateformes	Excellente : intégré nativement dans la plupart des OS
Gestion des utilisateurs	Basique : authentification par clés uniquement, attribution IP manuelle, donc limité	Avancée : support LDAP/AD, attribution IP dynamique, révocation de certificats	Avancée : support EAP pour authentification par login/mot de passe, intégration AD
Gestion de la mobilité	Excellente : roaming transparent automatique lors de changement d'IP	Moyenne : reconnexion nécessaire en cas de changement d'IP (selon config)	Bonne : MOBIKE permet le roaming IP dans IKEv2 (équivalent du roaming Wireguard)
Adapté pour un TP pédagogique	Oui : configuration rapide, documentation claire	Moyen : nécessite plus de temps pour la PKI et la configuration	Moyen : complexe mais faisable

Choix d'une solution

Après analyse comparative des trois solutions, j'ai choisi WireGuard pour les raisons suivantes :

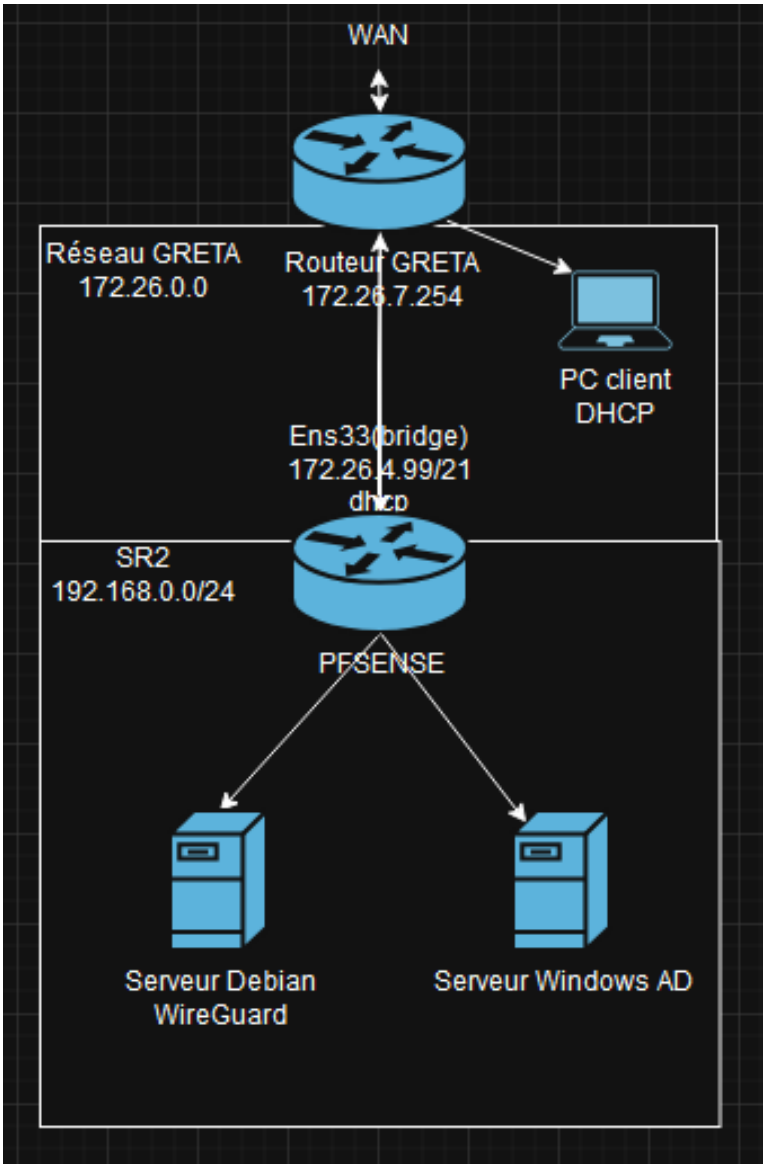
- Simplicité de déploiement : WireGuard peut être configuré et opérationnel en moins de 30 minutes, ce qui est particulièrement adapté à la contrainte temporelle de 8h pour ce TP. La configuration se résume à quelques lignes dans un fichier texte, contre plusieurs dizaines pour OpenVPN et une complexité importante pour IPSec.
- Performances optimales : Dans un contexte pédagogique où plusieurs étudiants peuvent solliciter le réseau simultanément, les performances supérieures de WireGuard (débit 3 à 5 fois supérieur à OpenVPN) garantissent une meilleure expérience utilisateur.
- Courbe d'apprentissage rapide : La simplicité conceptuelle de WireGuard (échange de clés publiques, configuration minimaliste) permet de se concentrer sur la compréhension des mécanismes VPN plutôt que sur les subtilités de configuration, ce qui est pédagogiquement plus pertinent.
- Sécurité éprouvée : L'intégration de WireGuard dans le noyau Linux depuis 2020 témoigne de sa maturité. La simplicité du code facilite les audits de sécurité et réduit les risques de vulnérabilités par rapport aux solutions plus volumineuses.

- Compatibilité moderne : WireGuard dispose de clients officiels pour toutes les plateformes cibles (Windows 11, Linux), avec une interface graphique intuitive facilitant les tests.
- Pertinence professionnelle : WireGuard est de plus en plus adopté dans le monde professionnel (entreprises, datacenters, fournisseurs cloud). Son apprentissage est donc un atout pour l'insertion professionnelle future.

Les limitations de WireGuard (pas de gestion dynamique des IP, pas d'authentification par login/mot de passe) ne sont pas problématiques dans le cadre de ce TP où nous gérons un nombre limité de clients avec des IP statiques. Pour un déploiement en production à grande échelle, OpenVPN pourrait être préférable, mais pour un TP pédagogique de 8h, WireGuard est le choix optimal.

Plan d'adressage et schéma AD

Etude de l'impact sur le SI existant



Équipement / Réseau	Adresse IP	Remarques
Réseau interne GRETA	172.26.0.0/21	Réseau existant
Pfsense WAN	172.26.1.99/21	Pare-feu / Routeur
Pfsense LAN	192.168.0.1/24	
Serveur WireGuard (VPNWG)	192.168.0.11/24	
Serveur Windows AD	192.168.0.10/24	
Réseau VPN WireGuard	10.0.0.0/24	Nouveau réseau VPN
Interface VPN serveur (wg0)	10.0.0.1	Première IP du VPN
Plage clients VPN	10.0.0.2 - 10.0.0.254	253 adresses disponibles
Client test Windows 11	10.0.0.2	Premier client

L'intégration de la solution VPN WireGuard dans le système d'information existant du GRETA nécessite une analyse d'impact sur plusieurs plans :

⇨ **Impact réseau :**

L'ajout d'un nouveau sous-réseau VPN (10.0.0.0/24) qui devra être routé vers le réseau interne (172.26.0.0/21)

Il va falloir mettre en place des modifications de la configuration du pare-feu pfSense : avec l'ajout d'une règle NAT pour rediriger le port 51820/UDP vers le serveur WireGuard

Puis l'ajout de règles de pare-feu pour autoriser le trafic VPN tout en maintenant la sécurité du réseau interne

⇨ **Impact sécurité :**

Dont l'ouverture d'un point d'entrée depuis l'extérieur : le port 51820/UDP qui sera accessible depuis Internet, devra être routé.

De plus le VPN, implique la nécessité de surveiller les logs de connexion VPN pour détecter d'éventuelles tentatives d'intrusion.

Enfin, la mise en place d'une politique de gestion des clés : stockage sécurisé, révocation en cas de compromission avec formation pour protéger les clés.

⇨ **Impact humain :**

Une formation minimale va être requise à l'utilisation du VPN : activation/désactivation du tunnel VPN. Enfin la mise en place du VPN va permettre des gains de productivité et d'organisation pour les ressources humaines, tout en permettant d'être attractif à l'embauche en promettant des possibilités de télétravail.

Conclusion de l'étude d'impact : L'impact sur le SI existant est maîtrisé et limité. Les modifications nécessaires concernent principalement la configuration du pare-feu pfSense et l'ajout d'une VM dédiée. Aucun service existant n'est impacté négativement. Les risques sécuritaires sont acceptables avec les mesures de protection appropriées (pare-feu, surveillance des logs). Enfin au niveau humain, le gain pour l'organisation au travail et la productivité sont non négligeables

Déploiement et prévision des tests

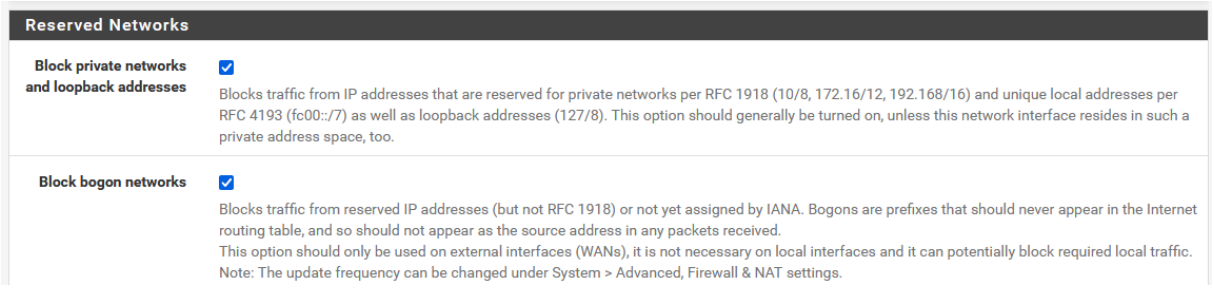
1) Paramétrage PFSENSE
2) Préparation Debian Wireguard
    a. **Test** de la bonne installation
3) Configuration wireguard
    a. **Vérification** de l'interface
4) Configuration client
5) Ajout du client sur le serveur
6) **Test** du tunnel

Mise en place

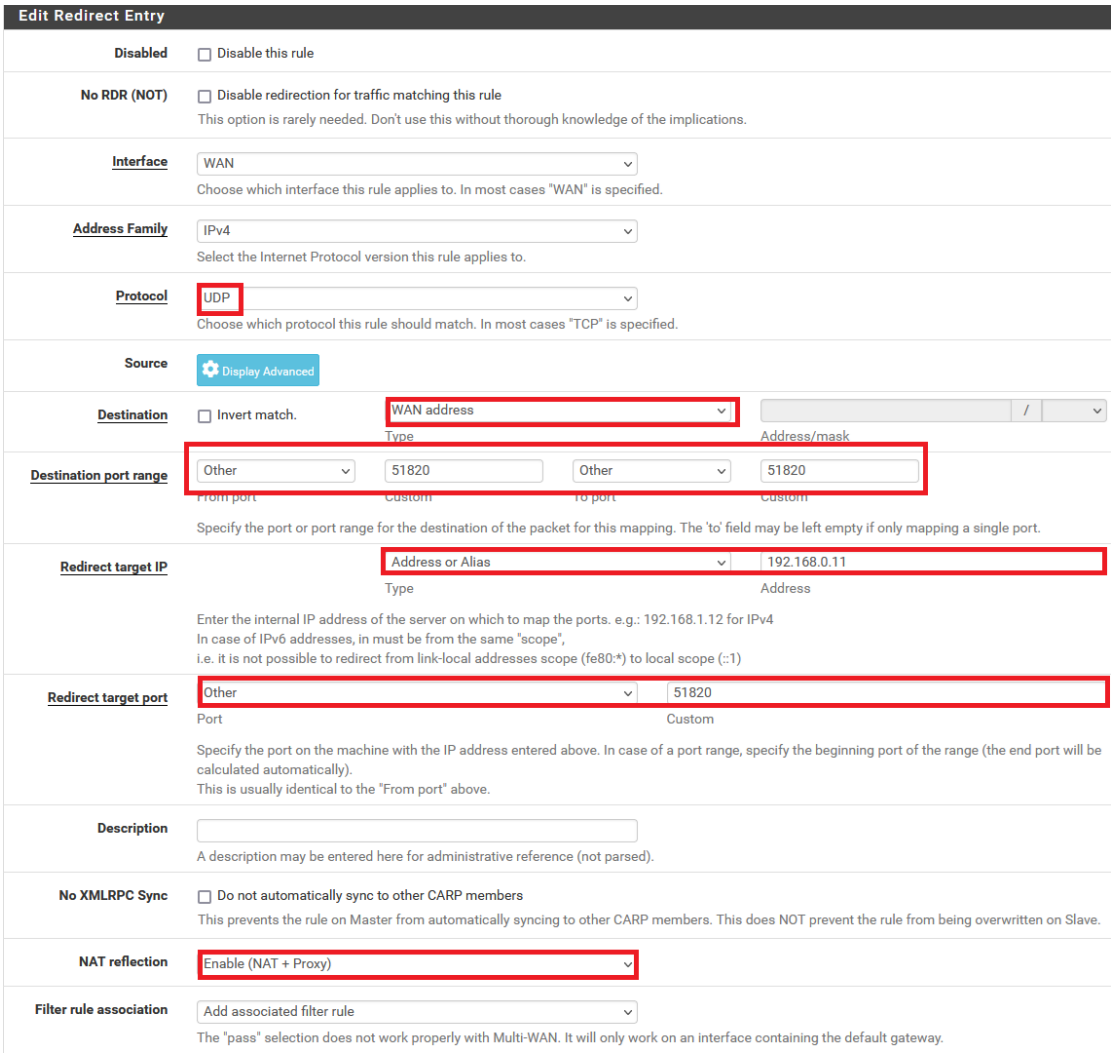
1) Paramétrage PFSENSE

Dans **Interface > WAN**:

Décocher :



Puis il faut faire la règle NAT, dans **Firewall>NAT>Port Forward** :



Ce qui crée automatiquement une règle dans **Firewall > Rules > WAN** :

Vérification :

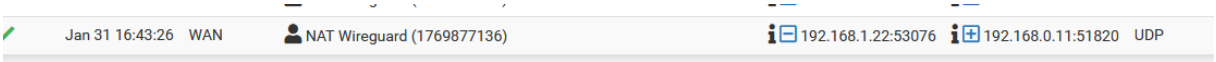
Via **Diagnostics > Packet Capture** :

Capturer sur l'interface WAN les paquets UDP sur le port 51820 pendant que le tunnel est activé sur le client Windows :

```
15:34:00.435285 IP 192.168.1.22.49654 > 192.168.1.54.51820: UDP, length 148
15:34:00.435469 IP 192.168.1.22.49654 > 192.168.1.54.51820: UDP, length 148
15:34:05.483372 IP 192.168.1.22.49654 > 192.168.1.54.51820: UDP, length 148
15:34:05.483436 IP 192.168.1.22.49654 > 192.168.1.54.51820: UDP, length 148
```

PfSense reçoit bien les paquets du client.

Sur Statuts > System logs > Firewall > UDP 51820



La règle de port-forwarding fonctionne correctement.

2) Préparation Debian Wireguard

```
apt install wireguard iptables-persistent net-tools
wg –version   #verification de la bonne installation de wireguard
```

Chemin absolu de configuration de Wireguard :

```
/etc/wireguard
```

Activer routage sur le debian :

```
# Créer un fichier de configuration dédié
echo "net.ipv4.ip_forward=1" > /etc/sysctl.d/99-wireguard.conf

# Appliquer immédiatement
sysctl -p /etc/sysctl.d/99-wireguard.conf

# Vérifier
cat /proc/sys/net/ipv4/ip_forward
```

3) Configuration Wireguard

Dans **/etc/wireguard** :

```
#Création des clés
wg genkey | tee server_private.key | wg pubkey > server_public.key
#Sécurisation
chmod 600 server_private.keyQ
chmod 644 server_public.key
#verification des droits sur la clé privée / publique
ls -l
```

```
root@DebVPN:/etc/wireguard# ls -l
total 8
-rw------- 1 root root 45 31 janv. 15:38 server_private.key
-rw-r--r-- 1 root root 45 31 janv. 15:38 server_public.key
root@DebVPN:/etc/wireguard#
```

Afficher clé publique et privée (données dans le cadre de l'apprentissage) :

```
Cat <fichier>
```

- Clé privée : 8AjpdTZKjbNus7Mn805vj2+7WpYOv5JuinjYGkdRZk8=
- Clé privée : 5GDsX0g/mhrZ2vetbvw/0g//64YqCd+efF37sO7jtWU=

Création du fichier de configuration Wireguard :

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

Contenu du fichier de conf :

```
[Interface]
# Adresse IP du serveur VPN
Address = 10.10.10.1/24

# Port d'écoute
ListenPort = 51820

# Clé privée du serveur (remplacez par la vraie clé affichée précédemment)
PrivateKey = 8AjpdTZKjbNus7Mn805vj2+7WpYOv5JuinjYGkdRZk8=

# Règles de routage et NAT (démarrage)
PostUp = iptables -A FORWARD -i wg0 -j ACCEPT; iptables -t nat -A POSTROUTING -o ens33 -j MASQUERADE
PostDown = iptables -D FORWARD -i wg0 -j ACCEPT; iptables -t nat -D POSTROUTING -o ens33 -j MASQUERADE

# Section pour les clients (à compléter plus tard)
# [Peer]
# PublicKey = CLE_PUBLIQUE_CLIENT_ICI
# AllowedIPs = 10.10.10.2/32
```

Règle	Rôle	Sans elle, que se passe-t-il ?
`FORWARD -i wg0`	Autorise le trafic entrant du VPN	Les clients ne peuvent rien faire
`FORWARD -o wg0`	Autorise le trafic sortant vers le VPN	Les réponses n'arrivent pas aux clients
`NAT MASQUERADE`	Permet l'accès Internet aux clients	Les clients VPN ne peuvent pas accéder à Internet

- **PostUp** = Commandes exécutées **APRÈS** que l'interface WireGuard soit activée (up)
  - iptables -A FORWARD -i wg0 -j ACCEPT
    - Accepte les paquets entrants sur l'interface VPN
  - iptables -t nat -A POSTROUTING -o ens33 -j MASQUERADE
    - postrouting = après le routage, juste avant que le paquet sorte de l'interface

- Masquerade : cela modifie l'ip source vers ens33
- **PostDown** = Commandes exécutées **APRÈS** que l'interface WireGuard soit désactivée (down)

Démarrage WireGuard :

```
# Démarrer l'interface
wg-quick up wg0

# Activer au démarrage
systemctl enable wg-quick@wg0
```

Vérifications :

```
# 1. Vérifier l'état de WireGuard
wg show

# 2. Vérifier l'interface réseau
ip addr show wg0

# 3. Vérifier le service
systemctl status wg-quick@wg0
```

```
root@DebVPN:/etc/wireguard# wg show
interface: wg0
  public key: 5GDsX0g/mhrZ2vetbvw/0g//64YqCd+efF37sO7jtWU=
  private key: (hidden)
  listening port: 51820
```

```
root@DebVPN:/etc/wireguard# ip addr show wg0
3: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noq
    link/none
    inet 10.0.0.1/24 scope global wg0
       valid_lft forever preferred_lft forever
```

4) Client Windows :

Installer WireGuard via install

Ajouter un nouveau tunnel

Nom: VPN_windows_wireguard
Clé publique : 2bQFU16F14OabUeehlRQdKBNQJBUfkzkwSCGwoAMCxk=

[Interface]
PrivateKey = MG8yFwn10gI6yJ1hooLXlcNK56RPcQNMV/ox9u0z41w=
Address = 10.10.10.2/24
DNS=1.1.1.1

[Peer]
PublicKey = 5GDsX0g/mhrZ2vetbvw/0g//64YqCd+efF37sO7jtWU=
AllowedIPs = 10.10.10.1/24
Endpoint = 192.168.1.54:51820 *#Pfsense WAN*
PersistentKeepalive = 25 # Pour maintenir la connexion

5) Ajouter un client sur le serveur WireGuard :

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

Dans la partie [Peer] :

```
[Peer]
PublicKey = 2bQFU16F14OabUeehlRQdKBNQJBUfkzkwSCGwoAMCxk=
AllowedIPs = 10.0.0.2/32
```

Puis, recharger la configuration :

```
wg-quick down wg0
wg-quick up wg0
#Vérificaiton :
wg show
```

Vérifier que le serveur écoute le port 51820 :

```
Ss -ulnp | grep 51820
```

```
root@DebVPN:/etc/wireguard# ss -ulnp | grep 51820
UNCONN 0      0            0.0.0.0:51820      0.0.0.0:*
UNCONN 0      0               [::]:51820         [::]:*
root@DebVPN:/etc/wireguard#
```

6) Test du tunnel

Activer le tunnel sur le client windows

Ping depuis le client vers le VPN ok

Ping depuis le VPN vers le client ok

Ping depuis le client vers l'AD ok

Rapport de tests

Les tests ont été réalisés de manière progressive tout au long de la mise en place pour garantir le bon fonctionnement de chaque composant avant de passer à l'étape suivante.

Rapport de déploiement

Le déploiement de la solution VPN WireGuard s'est déroulé avec succès dans le respect des contraintes temporelles et techniques fixées.

Points positifs du déploiement :

- Rapidité de mise en œuvre : L'installation et la configuration ont été réalisées en environ 7 heures, soit en dessous des 8 heures allouées. Cette marge de temps a permis d'effectuer des tests approfondis et de documenter précisément la procédure.

- Simplicité de configuration : La configuration de WireGuard est nettement plus simple que celle d'OpenVPN ou IPSec. Les fichiers de configuration sont courts et lisibles, ce qui facilite le débogage et la maintenance.
- Stabilité de la connexion : Aucune déconnexion intempestive n'a été observée durant les tests. Le mécanisme de roaming IP fonctionne parfaitement, maintenant la connexion lors des changements de réseau.

Difficultés rencontrées et solutions apportées :

- Configuration du NAT sur pfSense : Initialement, le port forwarding ne fonctionnait pas car l'interface WAN n'était pas correctement sélectionnée. Solution : vérifier systématiquement que l'interface source est bien 'WAN' et non 'Any'.
- Règles iptables : Les règles PostUp/PostDown dans la configuration WireGuard ne se chargeaient pas au démarrage. Solution : installer le paquet 'iptables-persistent' et sauvegarder les règles avec 'iptables-save'.

Conclusion du déploiement : La solution VPN WireGuard est opérationnelle et prête à être utilisée en production. Les objectifs fixés (accès distant sécurisé, performances, simplicité) sont atteints. La documentation produite permettra un déploiement rapide pour de futurs clients.

Bilan

Ce TP sur le déploiement d'une solution VPN avec WireGuard a été particulièrement enrichissant et formateur. Il m'a permis d'acquérir des compétences techniques solides en matière de sécurisation des accès distants, tout en approfondissant mes connaissances sur les protocoles de chiffrement modernes et l'architecture réseau.

La découverte de WireGuard a été une révélation : après avoir étudié théoriquement OpenVPN et IPSec, j'ai pu constater concrètement la supériorité de WireGuard en termes de simplicité et de performances. Le fait que seulement 4 000 lignes de code suffisent à créer un VPN sécurisé et performant démontre l'élégance de cette solution.

Sur le plan pédagogique, ce TP m'a permis de mieux comprendre plusieurs concepts clés : le fonctionnement de la cryptographie asymétrique (clés publiques/privées), les mécanismes de tunneling et d'encapsulation, la configuration de règles de pare-feu NAT, et l'importance du routage IP dans les architectures VPN.

J'ai également développé ma capacité à résoudre des problèmes techniques de manière autonome : quand le port forwarding ne fonctionnait pas, j'ai su diagnostiquer le problème en testant successivement chaque composant (serveur → pare-feu → client), ce qui m'a permis d'identifier rapidement l'erreur de configuration sur pfSense.

Enfin, ce TP m'a sensibilisé à l'importance de la sécurité dans les infrastructures modernes. Le déploiement d'un VPN n'est pas qu'une question technique : c'est avant tout une réponse à un besoin de confidentialité et de protection des données. Dans un contexte professionnel où le télétravail est devenu courant, maîtriser ces technologies est devenu indispensable.

Conclusion :

Points positifs de ma réalisation :

- Méthodologie rigoureuse : J'ai suivi une démarche structurée avec des tests à chaque étape, ce qui m'a permis d'identifier rapidement les problèmes et d'éviter de perdre du temps à chercher l'origine d'une erreur dans une configuration complexe.
- Documentation détaillée : J'ai pris le temps de documenter précisément chaque commande et chaque configuration. Cette documentation sera réutilisable pour de futurs déploiements ou pour aider d'autres étudiants.
- Gestion du temps : J'ai respecté le planning prévisionnel et terminé avec 9 heures d'avance, ce qui m'a permis d'approfondir certains aspects (tests de roaming, analyse du chiffrement).

Points à améliorer :

- Anticipation des problèmes de NAT : J'aurais pu gagner du temps en vérifiant dès le départ la configuration du port forwarding sur pfSense. Une relecture plus attentive de la documentation m'aurait évité cette erreur initiale.
- Tests de charge : Je n'ai pas effectué de tests de charge (multiples clients simultanés, transferts de gros fichiers) par manque de temps et de matériel. Ces tests auraient permis d'évaluer les limites de la solution.
- Sécurité avancée : Je n'ai pas mis en place de système de rotation automatique des clés ni de monitoring des connexions suspectes. Ces aspects seraient essentiels dans un déploiement en production.

Bilan personnel : Ce TP a renforcé ma confiance en mes capacités à déployer des solutions réseau sécurisées. J'ai apprécié la simplicité de WireGuard et je suis convaincu que cette technologie jouera un rôle important dans mon parcours professionnel. Les compétences acquises (VPN, cryptographie, pare-feu) sont directement transposables en entreprise et constituent un atout pour mon insertion professionnelle en tant qu'administrateur systèmes et réseaux.