BTS SIO — Option SISR

DOSSIER TECHNIQUE E6

Sécurisation des accès distants par VPN SSL

Candidat : SIDICINA Nathan

Session : 2026

Option : Solutions d’Infrastructure, Systèmes et Réseaux (SISR)

Environnement : maquette de validation en pré-production (VMware ESXi)

SOMMAIRE

I. Contexte et enjeux du projet

1.1 L'environnement de l'entreprise

L'entreprise STEP dispose d'une infrastructure interne composée de plusieurs serveurs critiques : un serveur GLPI pour la gestion du parc et le helpdesk, un serveur Zabbix pour la supervision, et un pare-feu pfSense qui assure la séparation entre la zone internet (WAN) et le réseau local (LAN).

Ces serveurs hébergent des données sensibles : inventaire du parc, mots de passe, configurations réseau, état des services. Ils doivent rester accessibles uniquement depuis le réseau interne.

1.2 Le besoin

Les techniciens et administrateurs de l'entreprise ont besoin de pouvoir intervenir à distance sur l'infrastructure, notamment lors des astreintes. La question qui s'est posée était : comment permettre cet accès sans pour autant ouvrir l'infrastructure à n'importe qui sur internet ?

La contrainte était double : l'accès doit être fonctionnel (joindre GLPI depuis l'extérieur) mais aussi sécurisé (chiffrement, authentification forte, traçabilité).

1.3 Compétences mobilisées

Ce projet couvre plusieurs compétences du référentiel SISR : sécurisation des accès distants, déploiement de services VPN, gestion des certificats (PKI), administration du pare-feu pfSense, contrôle des flux réseau et maintien en condition opérationnelle.

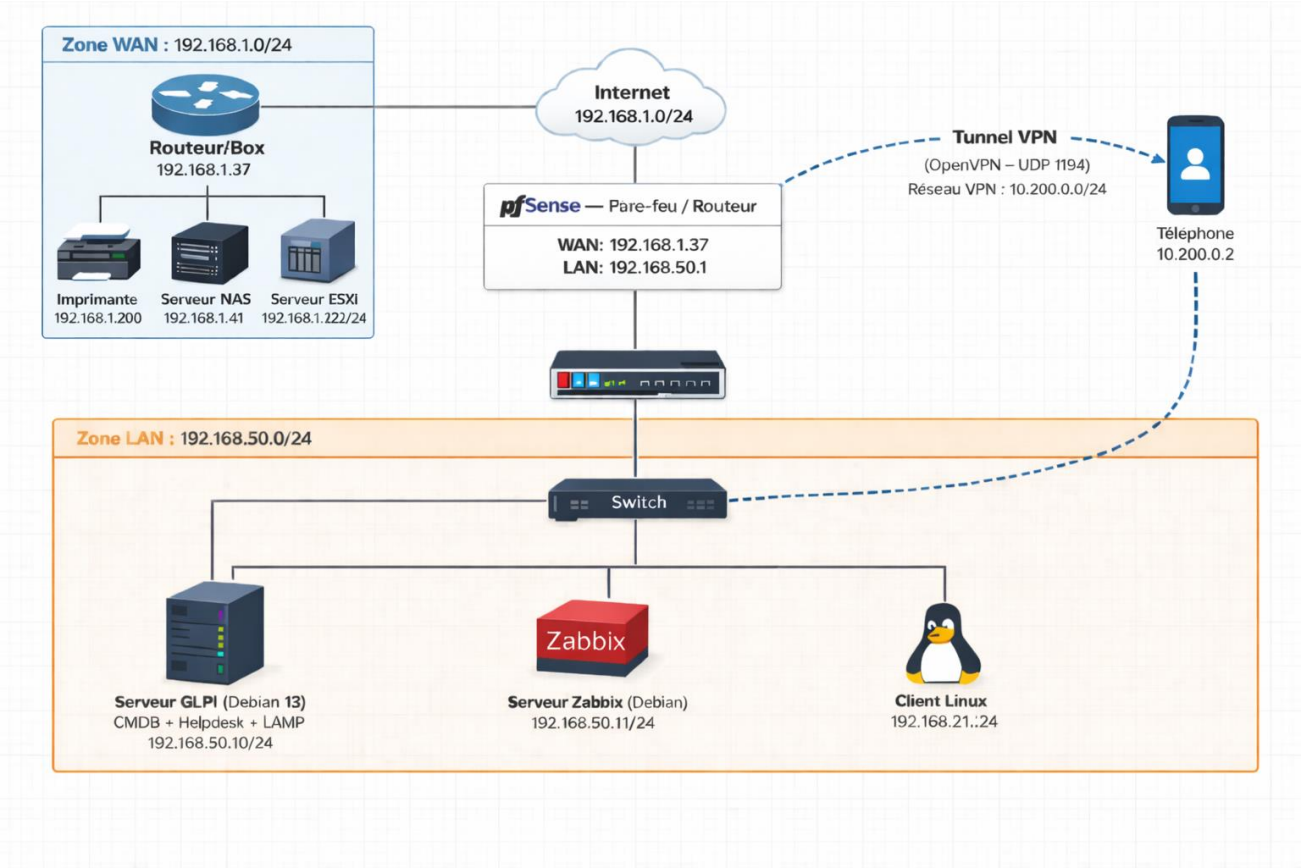
II. Architecture et choix techniques

2.1 Environnement de validation

Pour ne pas toucher à la production, j'ai reproduit l'infrastructure sur une maquette virtualisée sous VMware ESXi. Cela m'a permis de tester et d'itérer sans risque.

L'architecture comporte trois zones :

- Zone WAN : simule internet. Elle représente ce qu'un technicien voit depuis son domicile ou en déplacement. Sous-réseau : IP_RESEAU_WAN/24.
- Zone LAN : réseau interne protégé. Héberge le serveur GLPI (IP_GLPI_SERVER), le serveur Zabbix (IP_ZABBIX_SERVER) et le client Linux de test (IP_CLIENT_LINUX). Sous-réseau : IP_RESEAU_LAN/24.
- pfSense : pare-feu/routeur qui fait la jonction entre les deux zones. Interface WAN : IP_WAN_PFSENSE. Interface LAN : IP_LAN_PFSENSE.



2.2 Choix de la solution : pourquoi un VPN plutôt qu'une redirection de port ?

Avant de choisir le VPN, j'ai étudié plusieurs approches. Le tableau suivant résume la comparaison :

Critère	NAT / Redirection de port	VPN SSL (OpenVPN)	DMZ
Exposition du serveur	Publique — visible sur internet	Privée — invisible depuis l'extérieur	Partielle
Authentification	Login / mot de passe uniquement	Certificat + identifiants (double facteur)	Login / mot de passe
Chiffrement du trafic	Aucun si HTTP	AES-256 sur tout le tunnel	Dépend du service
Surface d'attaque	Maximale	Très réduite	Moyenne
Complexité de mise en place	Faible	Moyenne	Élevée

Le VPN SSL via OpenVPN a été retenu. Il permet de garder les serveurs dans le LAN protégé, d'authentifier les utilisateurs avec un certificat, et de chiffrer toute la communication. Seul le port UDP 1194 doit être ouvert sur le WAN.

III. Mise en œuvre — Phase 1 : audit et démonstration du risque

3.1 Objectif de cette phase

Avant de justifier la solution VPN, j'ai voulu montrer concrètement ce qui se passe avec une approche simpliste. J'ai donc configuré une règle NAT pour rendre GLPI accessible directement depuis le WAN, puis j'ai analysé les vulnérabilités que ça crée.

3.2 Configuration de la règle NAT sur pfSense

Étape 1 — Accès à la section NAT

Depuis l'interface d'administration pfSense, on va dans : Firewall > NAT > Port Forward, puis on clique sur Add.

Étape 2 — Paramétrage de la règle

On configure la redirection de port avec les paramètres suivants :

- Interface : WAN
- Protocol : TCP
- Destination port range : HTTP (80)
- Redirect target IP : IP_GLPI_SERVER (adresse du serveur GLPI dans le LAN)
- Redirect target port : HTTP (80)

On sauvegarde et on applique les changements.

3.3 Constat de vulnérabilité

Une fois la règle en place, le portail GLPI est accessible directement depuis n'importe quel navigateur sur le WAN. On peut s'y connecter sans aucune condition préalable.

[CAPTURE D'ÉCRAN — Navigateur depuis la zone WAN — accès direct à la page de connexion GLPI via l'IP WAN]

HTTP	TCP	Tous	Tous	80	80	192.168.1.91	☑	🔧	🗑

Mais cette facilité d'accès cache trois problèmes sérieux :

> **Vulnérabilité 1 — Trafic en clair : les identifiants GLPI circulent en HTTP. Une capture réseau permet de les lire directement.**
>
> **Vulnérabilité 2 — Exposition publique : le serveur est visible sur internet. Un scan nmap depuis le WAN détecte le port ouvert et identifie le service GLPI avec sa version.**
>
> **Vulnérabilité 3 — Authentification faible : seul un mot de passe protège l'accès. Aucune limitation des tentatives, attaque par force brute possible.**

IV. Mise en œuvre — Phase 2 : mise en place de la PKI

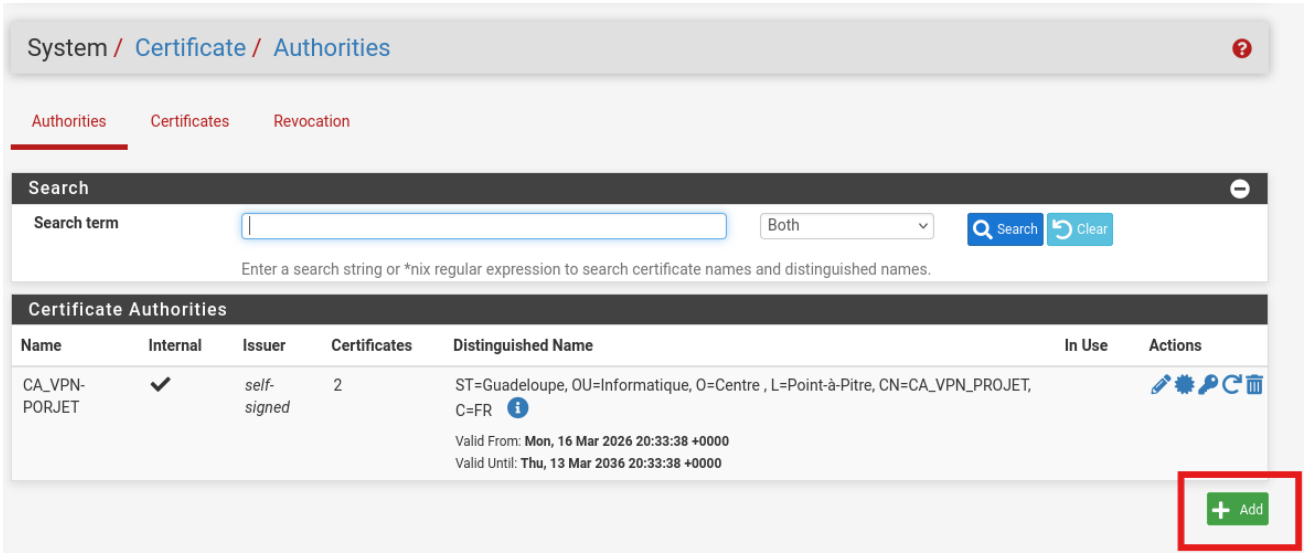
4.1 Objectif

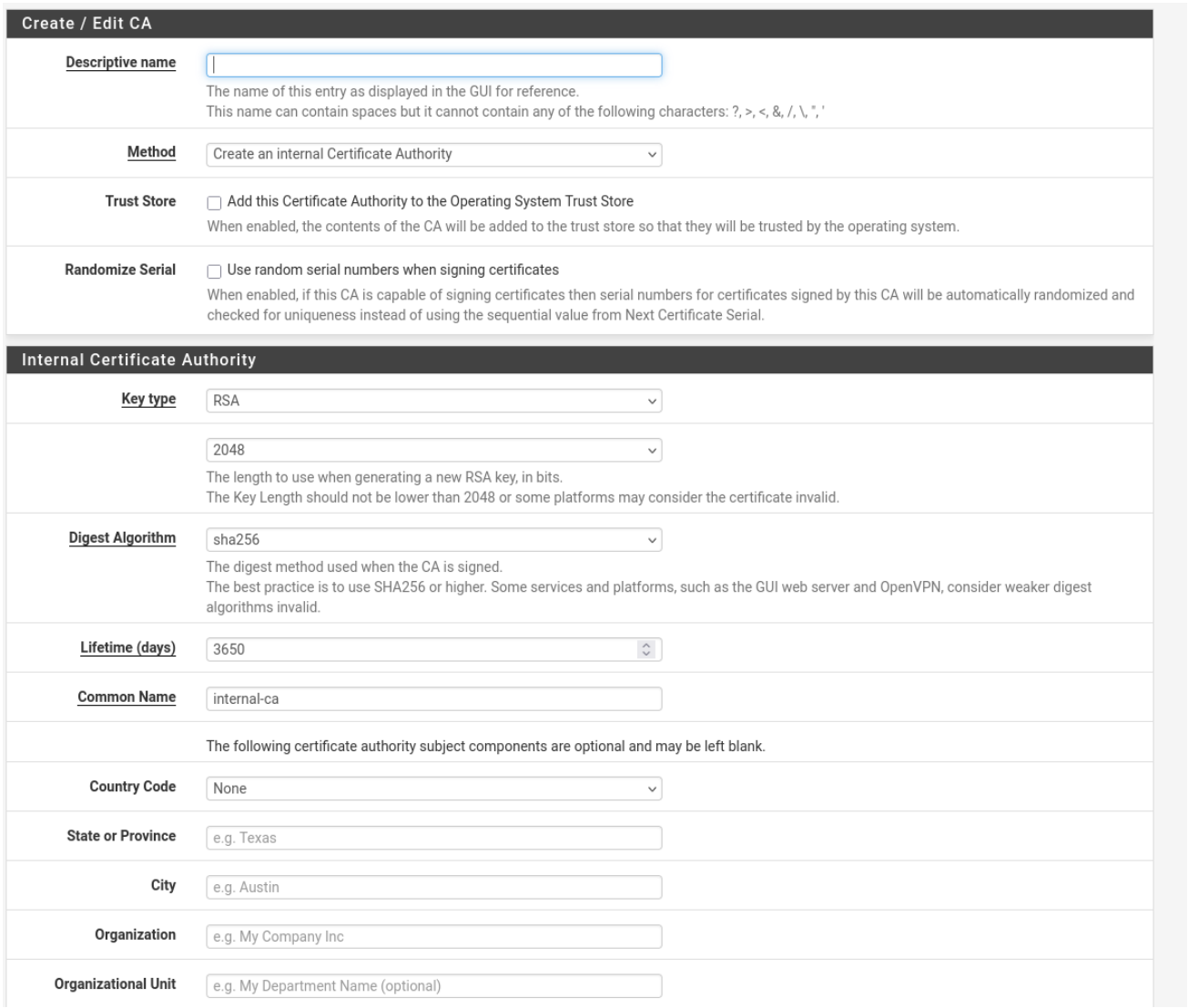
Pour que le VPN soit vraiment sécurisé, on ne peut pas se contenter d'un simple mot de passe. L'idée est d'ajouter un deuxième facteur : la possession d'un certificat numérique. Sans ce certificat, même avec les bons identifiants, la connexion est impossible.

Pour ça, il faut créer une PKI (infrastructure à clés publiques) composée de trois éléments : une autorité de certification (CA), un certificat pour le serveur VPN, et un certificat pour chaque utilisateur.

4.2 Création de l'Autorité de Certification (CA)

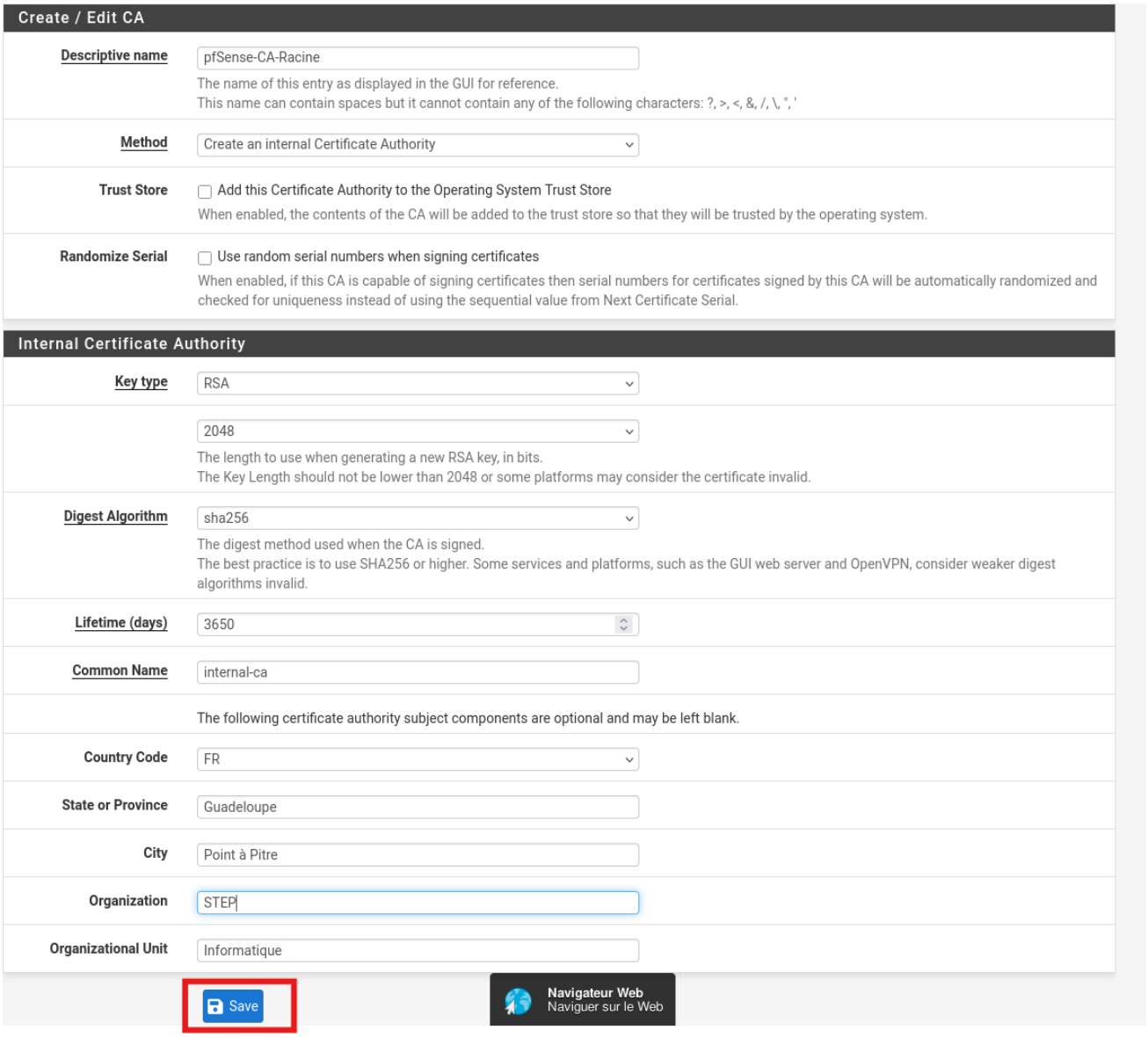
Sur pfSense, on va dans : System > Cert. Manager > CAs > Add.



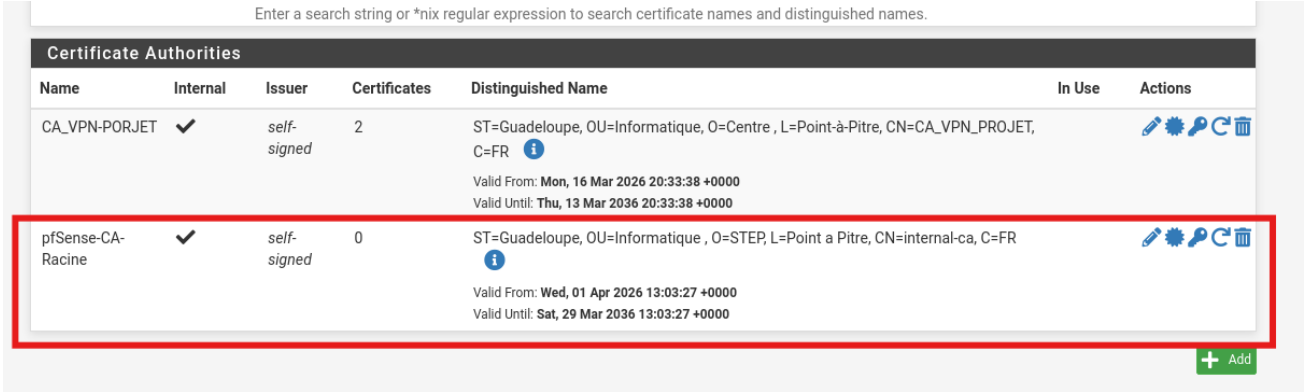


Paramètres saisis :

- Descriptive name : pfSense-CA-Racine
- Method : Create an internal Certificate Authority
- Key type : RSA, 2048 bits
- Digest Algorithm : SHA256
- Lifetime : 3650 jours (10 ans)
- Common Name : internal-ca



On sauvegarde. La CA apparaît maintenant dans la liste.

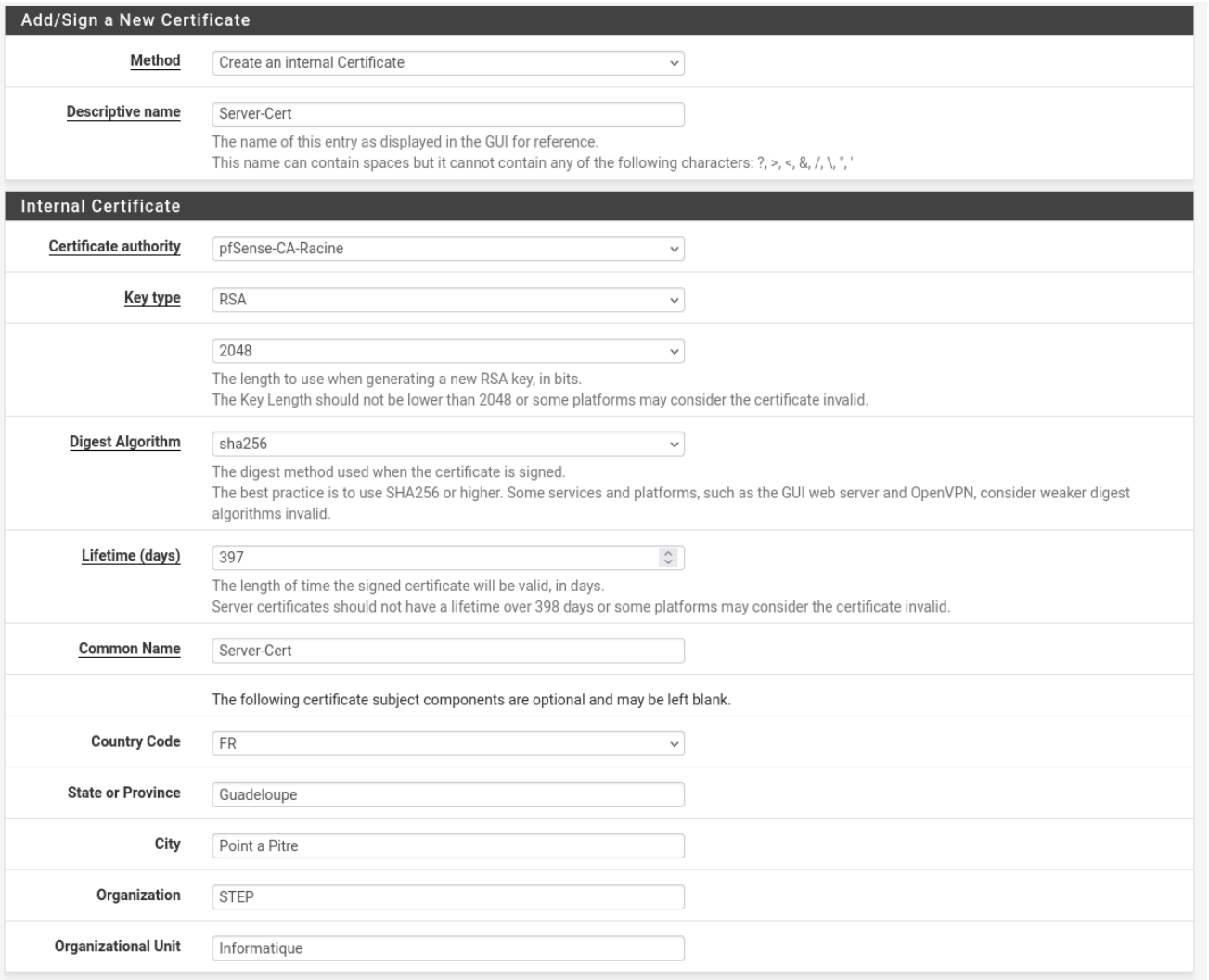


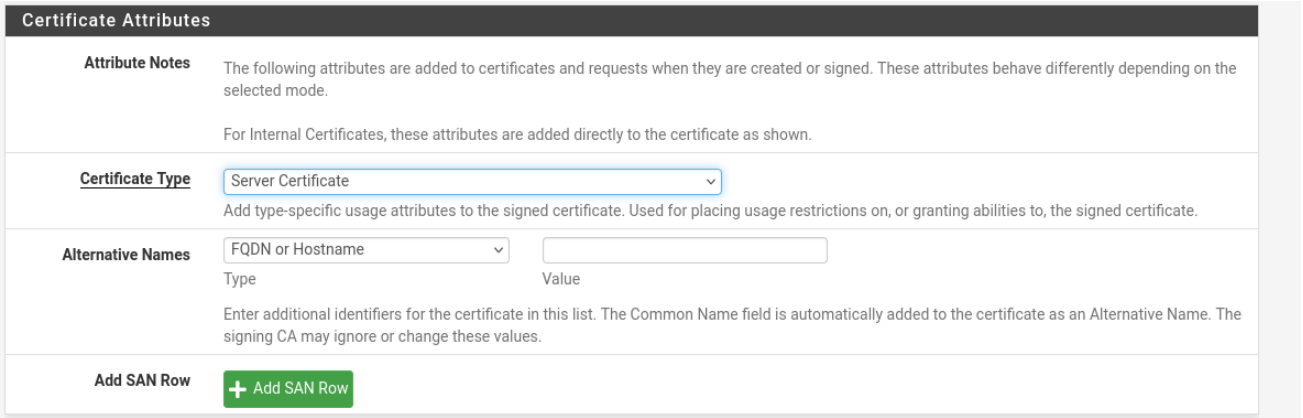
4.3 Création du certificat serveur

Ce certificat va permettre au serveur pfSense de prouver son identité aux clients VPN. On va dans : System > Cert. Manager > Certificates > Add.

Paramètres :

- Method : Create an internal Certificate
- Descriptive name : Server-Cert
- Certificate Authority : pfSense-CA-Racine
- Type : Server Certificate
- Key type : RSA 2048 / SHA256
- Lifetime : 397 jours



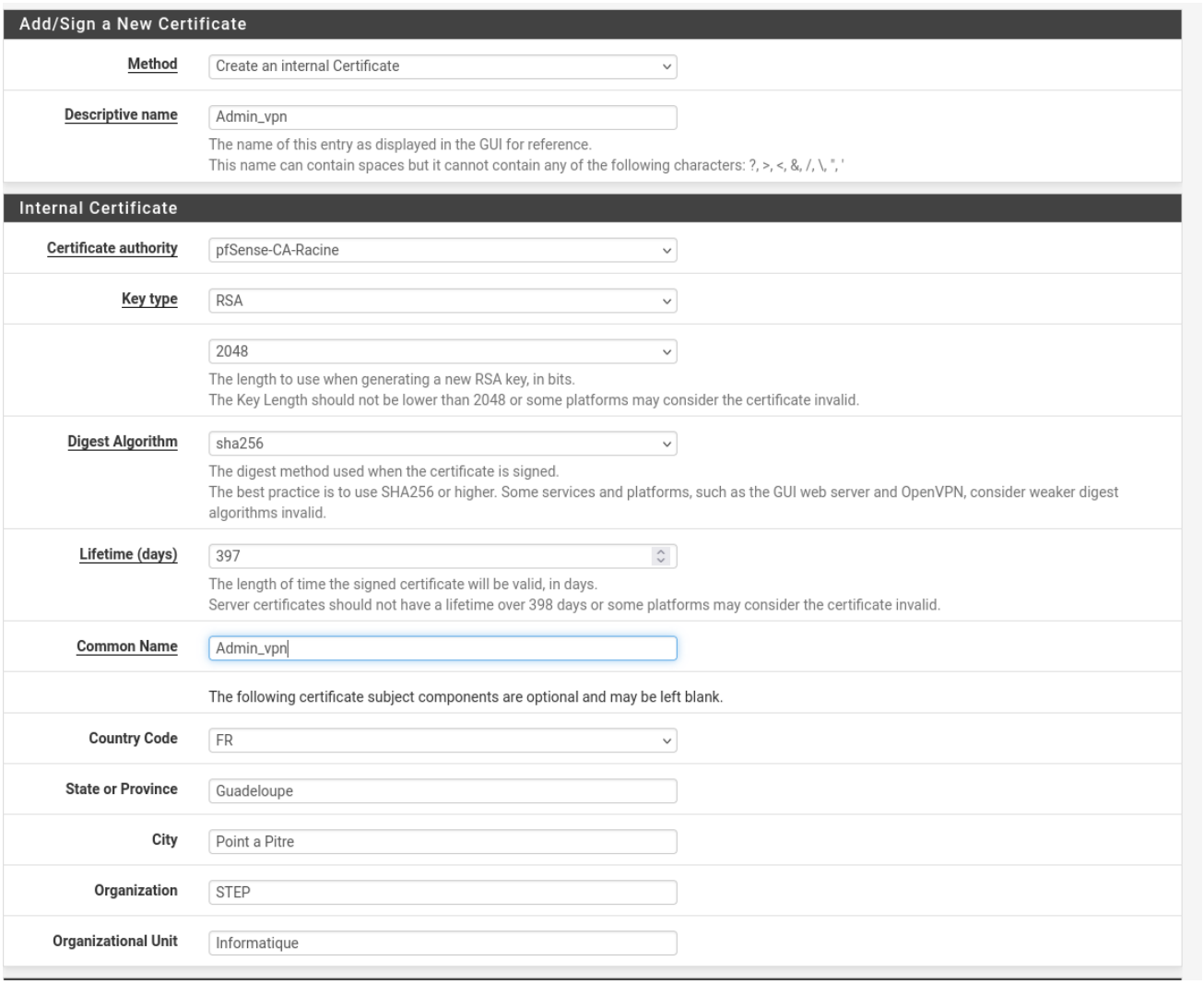


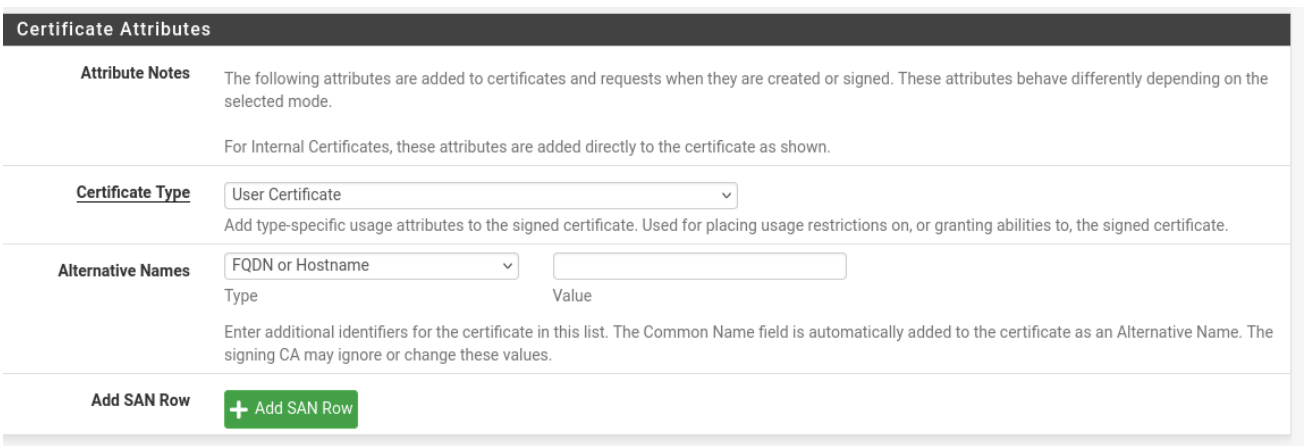
4.4 Création du certificat client (administrateur)

On crée ensuite le certificat pour l'utilisateur qui va se connecter en VPN. Même menu, mais avec le type « User Certificate » cette fois.

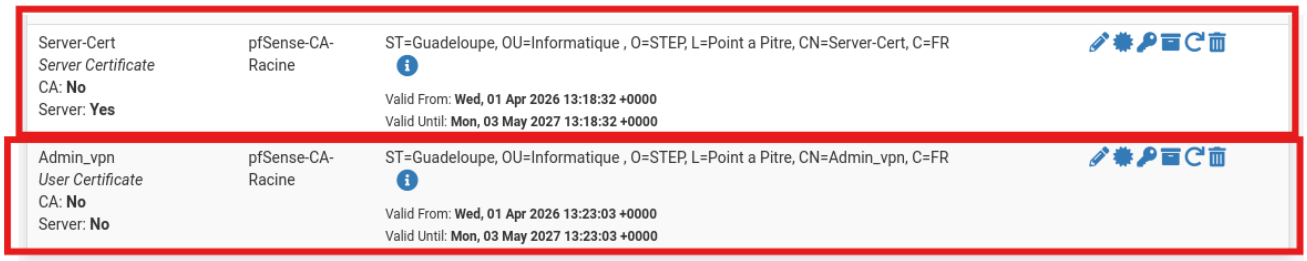
Paramètres :

- Descriptive name : Admin-vpn
- Certificate Authority : pfSense-CA-Racine
- Type : User Certificate
- Key type : RSA 2048 / SHA256
- Lifetime : 397 jours





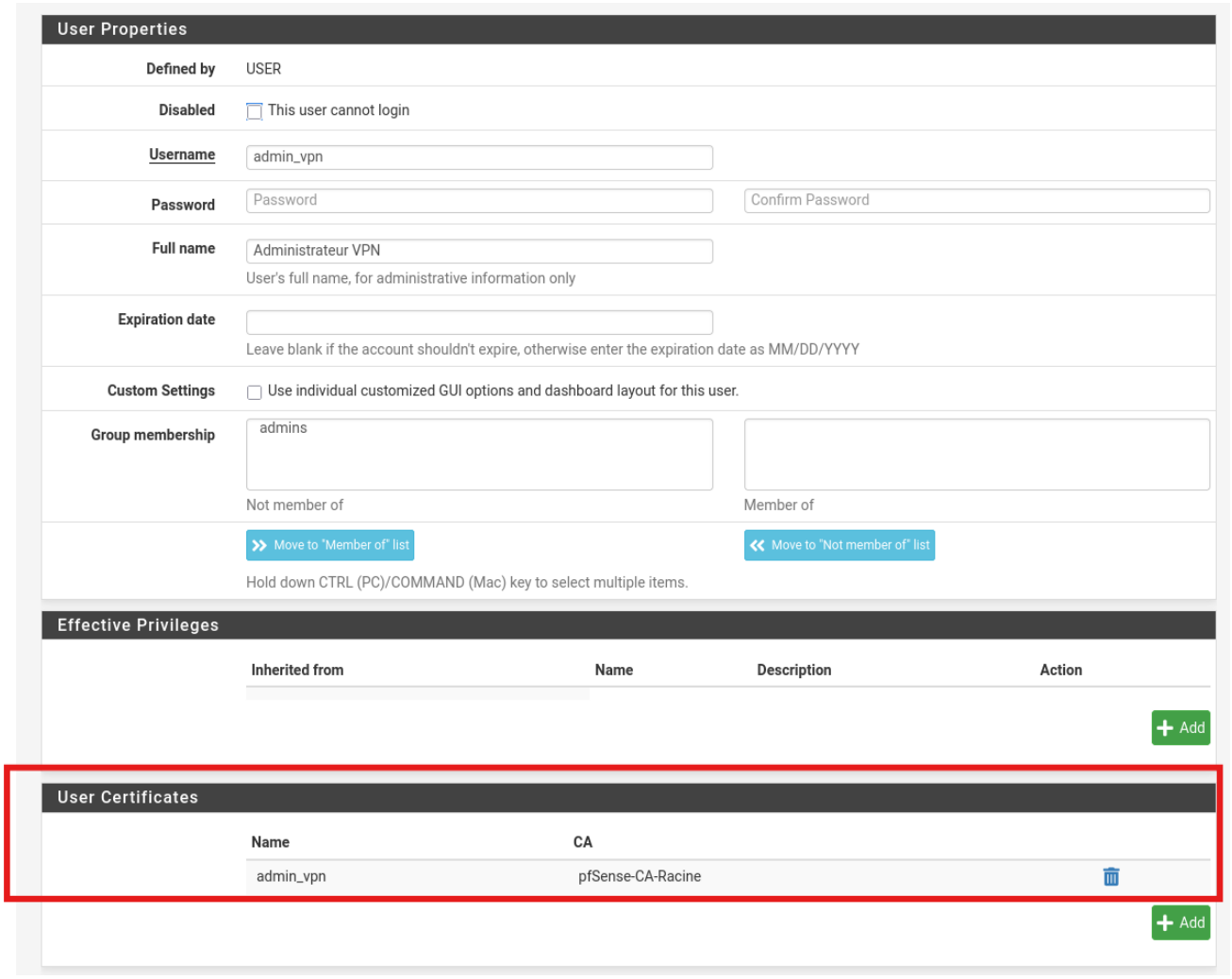
On a maintenant deux certificats dans la liste : Server-Cert et Admin-vpn, tous deux signés par notre CA.



4.5 Association du certificat à l'utilisateur pfSense

Pour que le client OpenVPN puisse exporter le profil avec le bon certificat, il faut associer Admin-vpn à un compte utilisateur pfSense. On va dans : System > User Manager > Add.

On crée l'utilisateur admin_vpn et on lui assigne le certificat admin-vpn dans la section « User Certificates ».



V. Mise en œuvre — Phase 3 : configuration du tunnel OpenVPN

5.1 Objectif

Maintenant que la PKI est en place, on peut configurer le serveur OpenVPN. L'objectif est d'établir un tunnel chiffré entre le client (côté WAN) et le réseau interne, avec un filtrage strict sur ce que le client VPN a le droit de joindre.

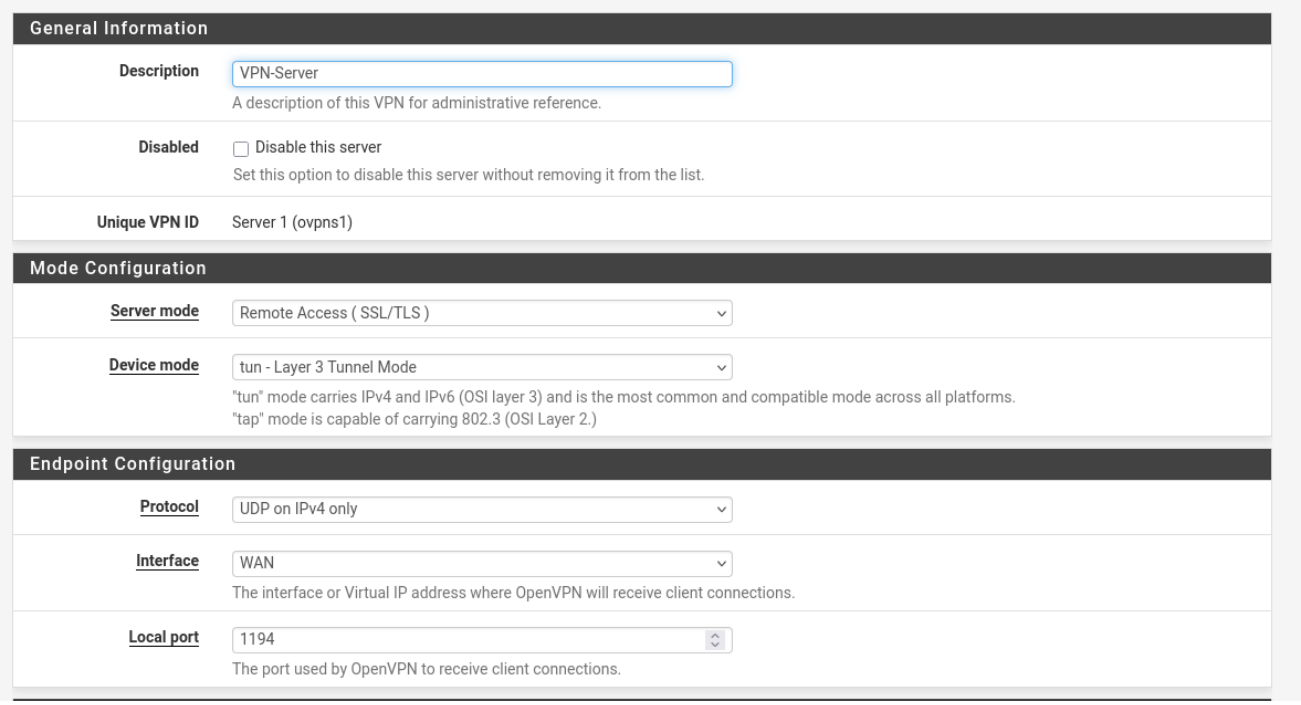
5.2 Configuration du serveur OpenVPN

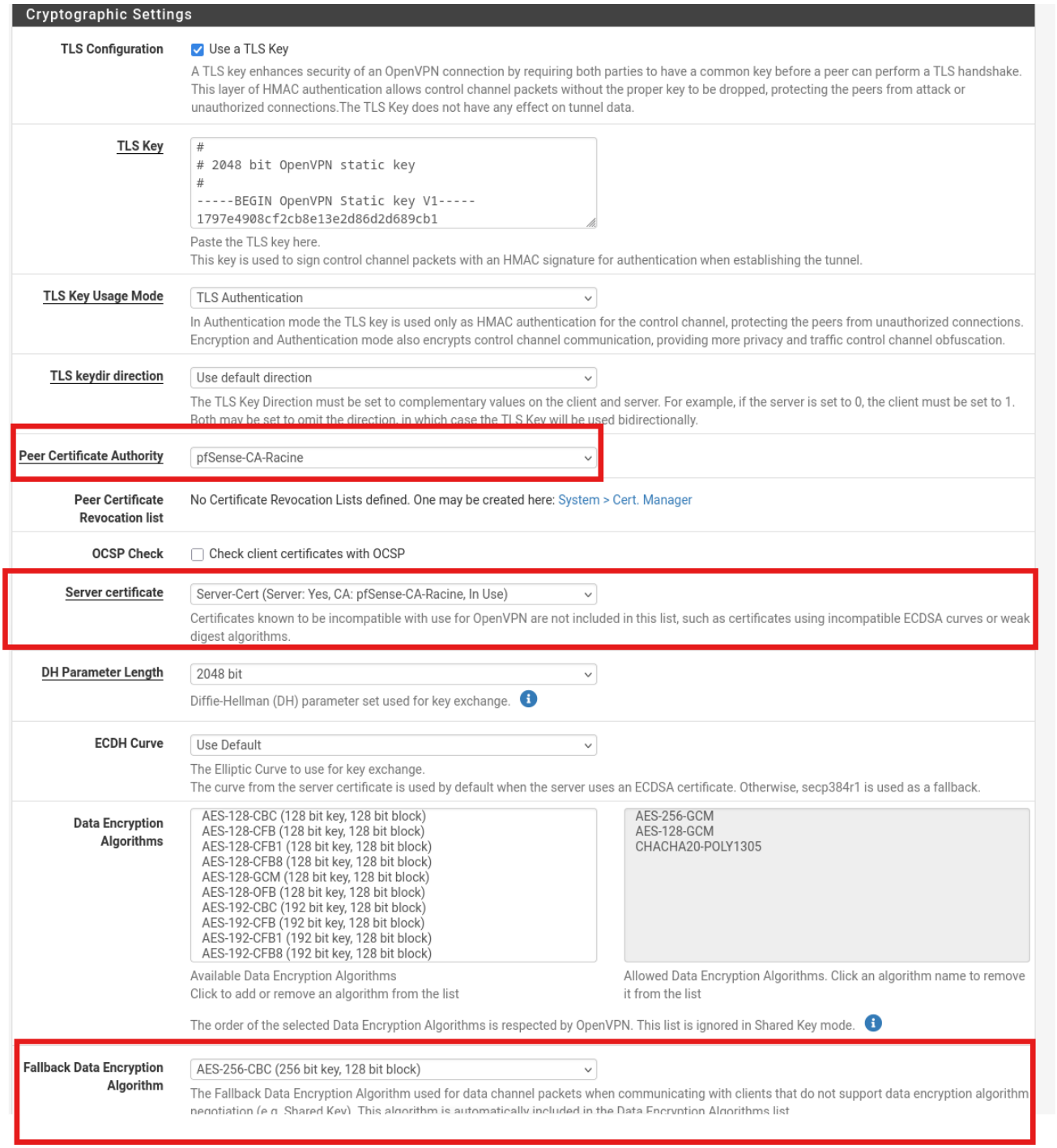
Étape 1 — Création du serveur VPN

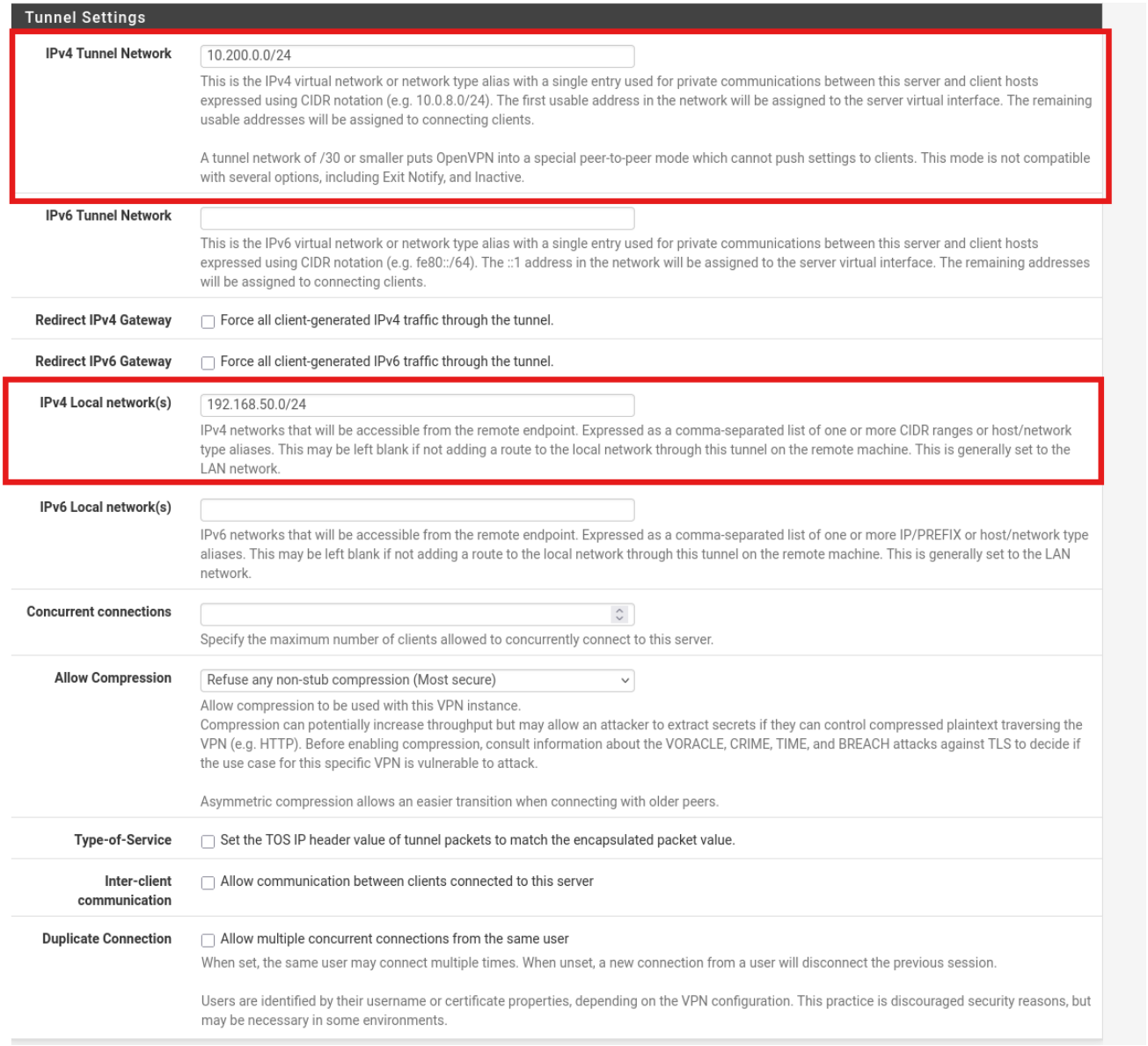
On va dans : VPN > OpenVPN > Servers > Add.

Paramètres principaux :

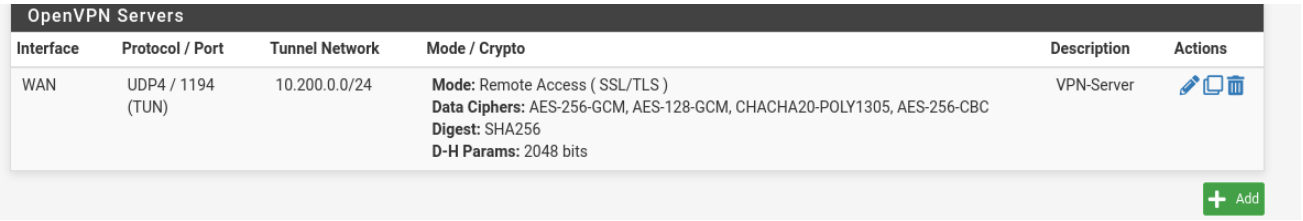
- Server mode : Remote Access (SSL/TLS + User Auth)
- Protocol : UDP on IPv4 only
- Device mode : tun — Layer 3 Tunnel Mode
- Interface : WAN
- Local port : 1194
- TLS Configuration : activé (génération d'une clé TLS automatique)
- Peer Certificate Authority : pfSense-CA-Racine
- Server certificate : Server-Cert
- Encryption Algorithm : AES-256-CBC
- Tunnel Network : 10.200.0.0/24 (zone isolée pour les clients VPN)
- Local Network : 192.168.50.0/24 (pour le push de la route vers le client)




Cryptographic Settings
TLS Configuration
Use a TLS Key
A TLS key enhances security of an OpenVPN connection by requiring both parties to have a common key before a peer can perform a TLS handshake. This layer of HMAC authentication allows control channel packets without the proper key to be dropped, protecting the peers from attack or unauthorized connections.The TLS Key does not have any effect on tunnel data.
TLS Key
#
2048 bit OpenVPN static key
#
-----BEGIN OpenVPN Static key V1-----
1797e4908cf2cb8e13e2d86d2d689cb1
Paste the TLS key here.
This key is used to sign control channel packets with an HMAC signature for authentication when establishing the tunnel.
TLS Key Usage Mode
TLS Authentication
In Authentication mode the TLS key is used only as HMAC authentication for the control channel, protecting the peers from unauthorized connections. Encryption and Authentication mode also encrypts control channel communication, providing more privacy and traffic control channel obfuscation.
TLS keydir direction
Use default direction
The TLS Key Direction must be set to complementary values on the client and server. For example, if the server is set to 0, the client must be set to 1. Both may be set to omit the direction, in which case the TLS Key will be used bidirectionally.
Peer Certificate Authority
pfSense-CA-Racine
Peer Certificate Revocation list
No Certificate Revocation Lists defined. One may be created here: System > Cert. Manager
OCSP Check
Check client certificates with OCSP
Server certificate
Server-Cert (Server: Yes, CA: pfSense-CA-Racine, In Use)
Certificates known to be incompatible with use for OpenVPN are not included in this list, such as certificates using incompatible ECDSA curves or weak digest algorithms.
DH Parameter Length
2048 bit
Diffie-Hellman (DH) parameter set used for key exchange.
ECDH Curve
Use Default
The Elliptic Curve to use for key exchange.
The curve from the server certificate is used by default when the server uses an ECDSA certificate. Otherwise, secp384r1 is used as a fallback.
Data Encryption Algorithms
AES-128-CBC (128 bit key, 128 bit block)
AES-128-CFB (128 bit key, 128 bit block)
AES-128-CFB1 (128 bit key, 128 bit block)
AES-128-CFB8 (128 bit key, 128 bit block)
AES-128-GCM (128 bit key, 128 bit block)
AES-128-OFB (128 bit key, 128 bit block)
AES-192-CBC (192 bit key, 128 bit block)
AES-192-CFB (192 bit key, 128 bit block)
AES-192-CFB1 (192 bit key, 128 bit block)
AES-192-CFB8 (192 bit key, 128 bit block)
Available Data Encryption Algorithms
Click to add or remove an algorithm from the list
AES-256-GCM
AES-128-GCM
CHACHA20-POLY1305
Allowed Data Encryption Algorithms. Click an algorithm name to remove it from the list
The order of the selected Data Encryption Algorithms is respected by OpenVPN. This list is ignored in Shared Key mode.
Fallback Data Encryption Algorithm
AES-256-CBC (256 bit key, 128 bit block)
The Fallback Data Encryption Algorithm used for data channel packets when communicating with clients that do not support data encryption algorithm negotiation (e.g. Shared Key). This algorithm is automatically included in the Data Encryption Algorithms list.



On sauvegarde. Le serveur OpenVPN apparaît dans la liste avec le statut actif.



> *UDP a été choisi plutôt que TCP pour éviter le problème de TCP over TCP qui dégrade les performances. Le mode tun (couche 3) est adapté aux accès distants : le client reçoit une IP dans le sous-réseau tunnel et le routage vers le LAN est poussé automatiquement (Local Network).*

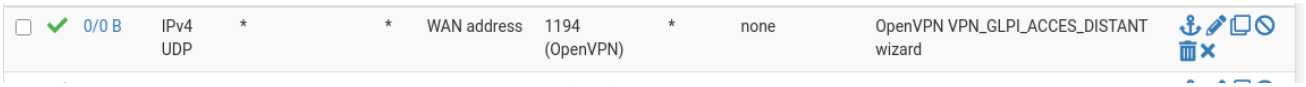
5.3 Règles de pare-feu

Étape 1 — Règle WAN : autoriser le port 1194

Sans règle WAN, pfSense bloque tout le trafic entrant par défaut. Il faut autoriser explicitement les connexions sur le port 1194 UDP.

On va dans : Firewall > Rules > WAN > Add.

- Action : Pass
- Interface : WAN
- Protocol : UDP
- Destination port : 1194 (OpenVPN)
-

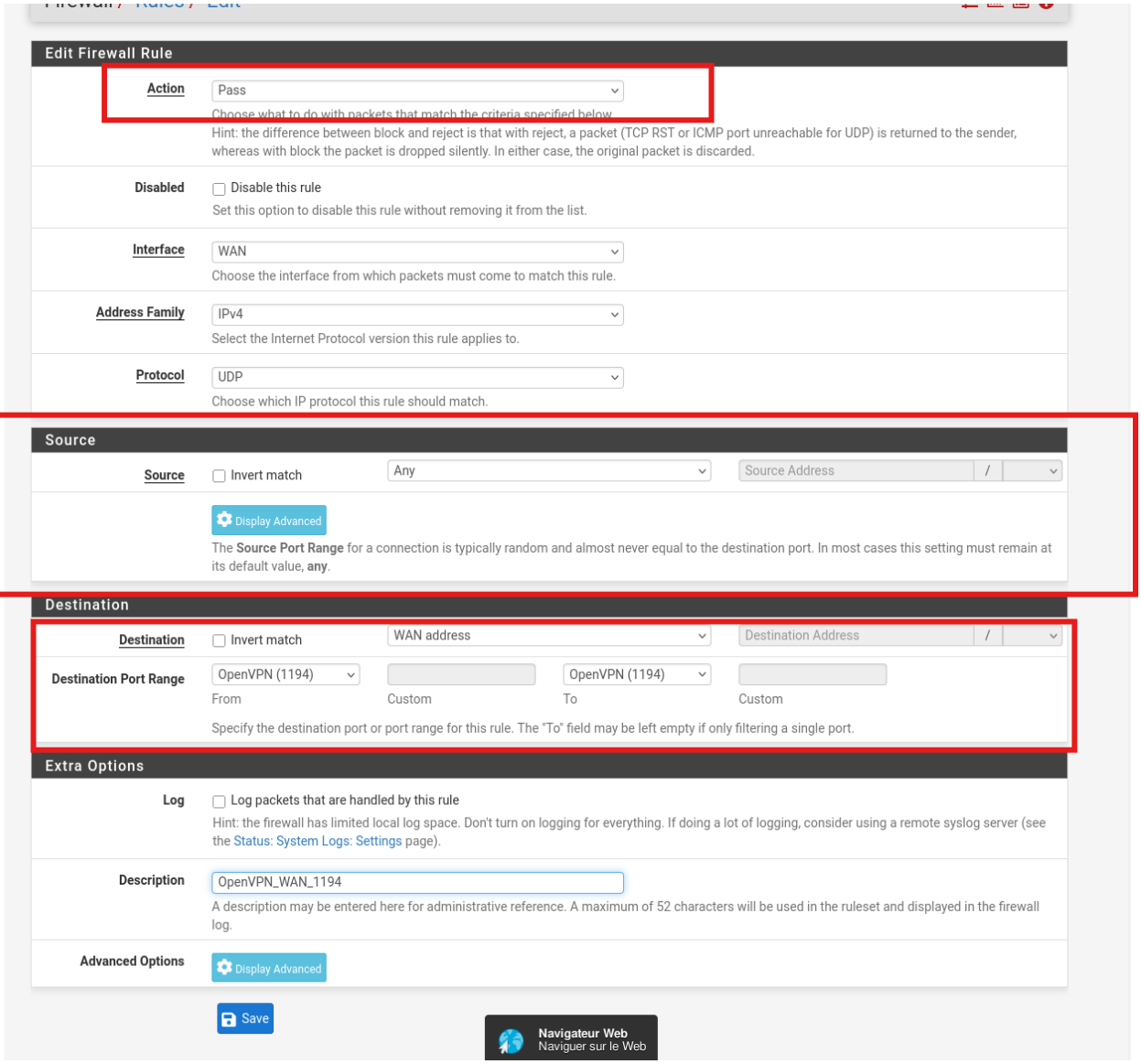


Étape 2 — Règle OpenVPN : filtrage des flux du tunnel

Une fois connecté au VPN, le client doit être limité à ce dont il a vraiment besoin. On applique le principe du moindre privilège : on n'autorise que les flux vers GLPI sur les ports 80 et 443, tout le reste est bloqué.

On va dans : Firewall > Rules > OpenVPN > Add.

- Action : Pass
- Interface : OpenVPN
- Protocol : TCP
- Source : 10.200.0.0/24 (réseau des clients VPN)
- Destination : 192.168.50.10/24
- Destination port : HTTP (80) et HTTPS (443)



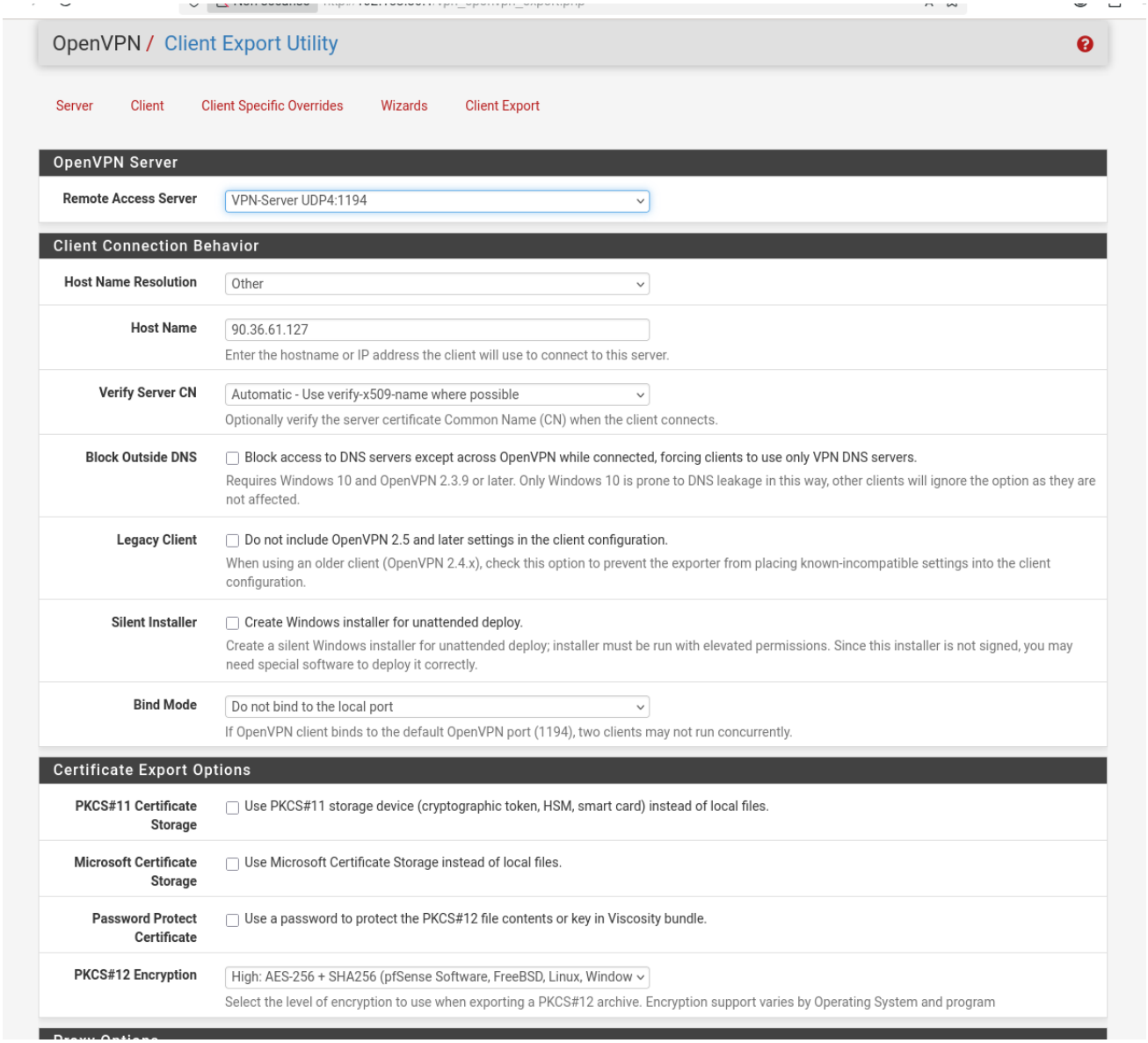
Tout ce qui n'est pas explicitement autorisé (SSH, RDP, accès Zabbix...) est bloqué par la règle implicite de refus par défaut de pfSense.

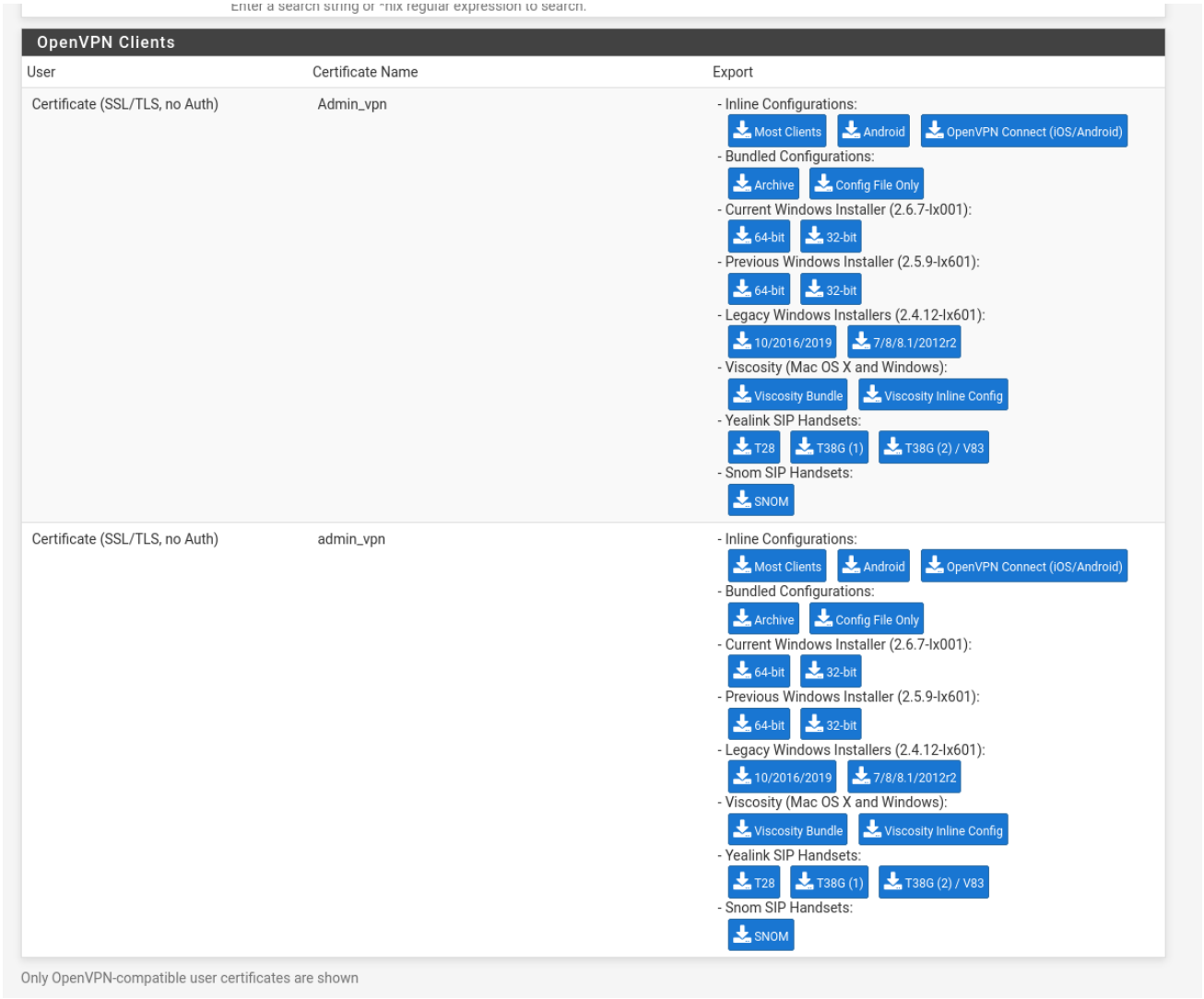
5.4 Export du profil client et test de connexion

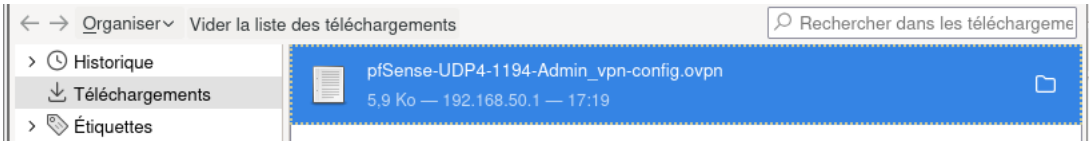
Étape 1 — Export du profil .ovpn

pfSense intègre un outil d'export qui génère un fichier .ovpn contenant toute la configuration : adresse du serveur, certificat CA, certificat client, clé privée. Le client n'a rien à configurer manuellement.

On va dans : VPN > OpenVPN > Client Export. On sélectionne le serveur VPN-Server et l'utilisateur admin_vpn, puis on télécharge le fichier « Inline Configurations — Most Clients ».

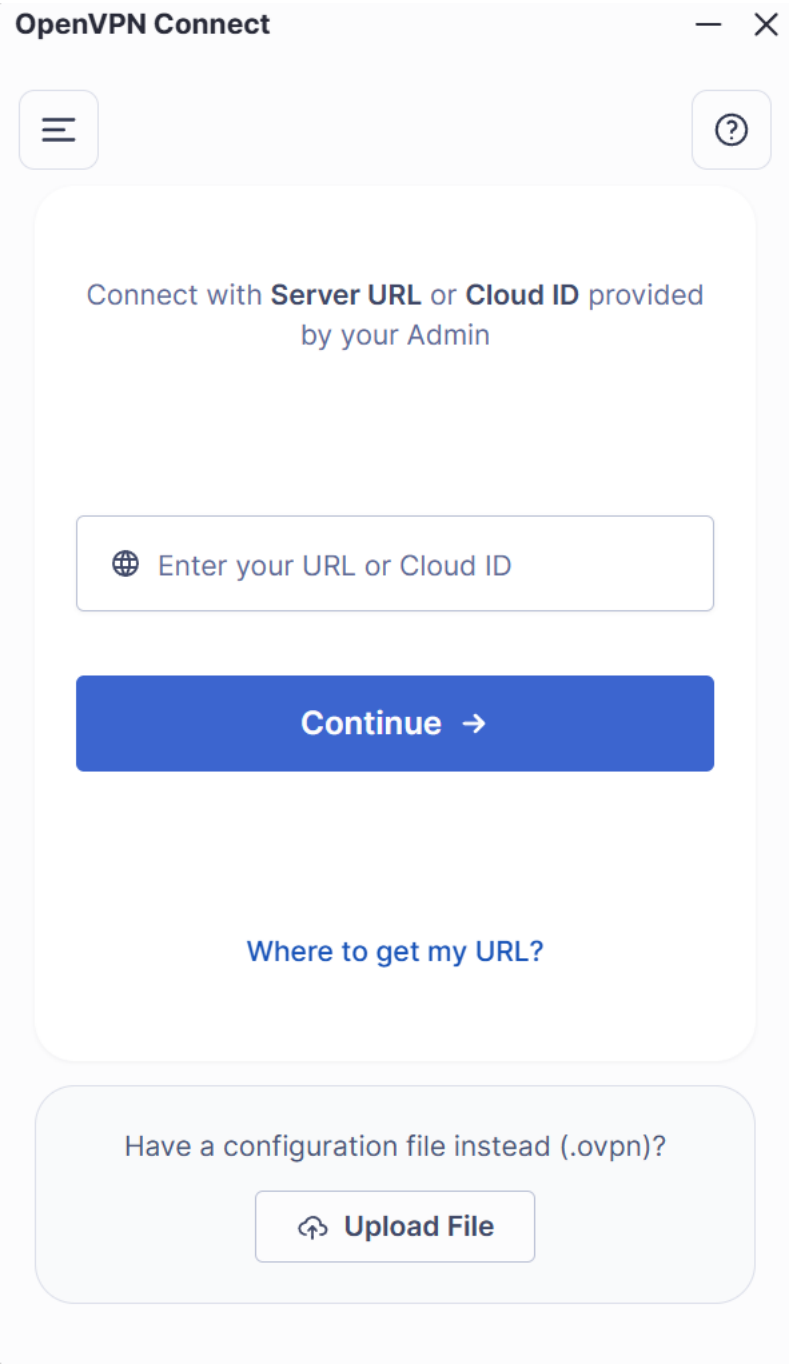


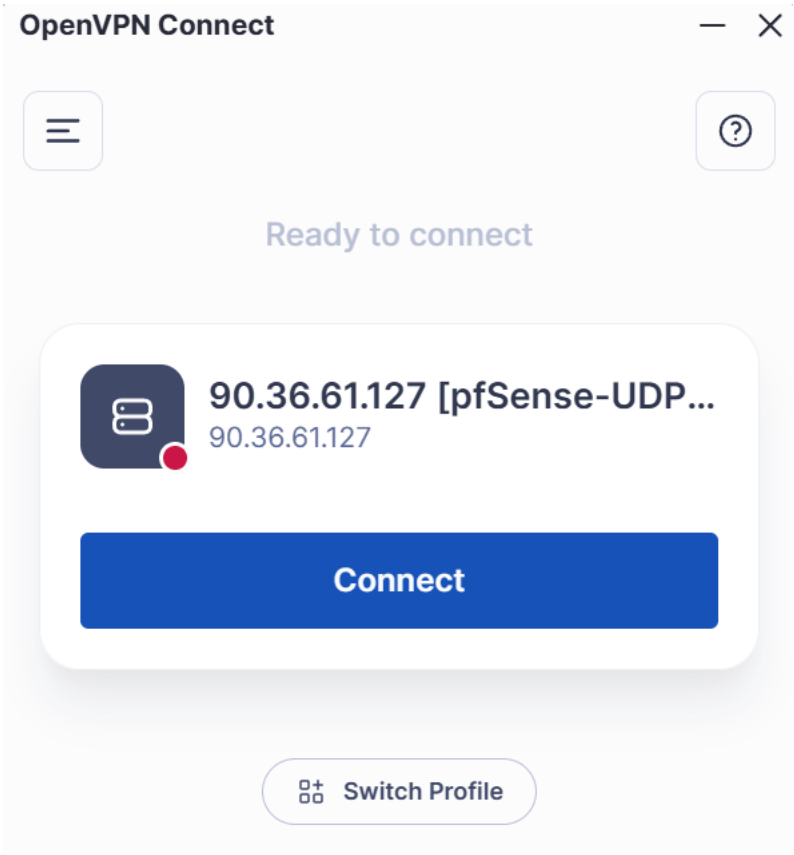


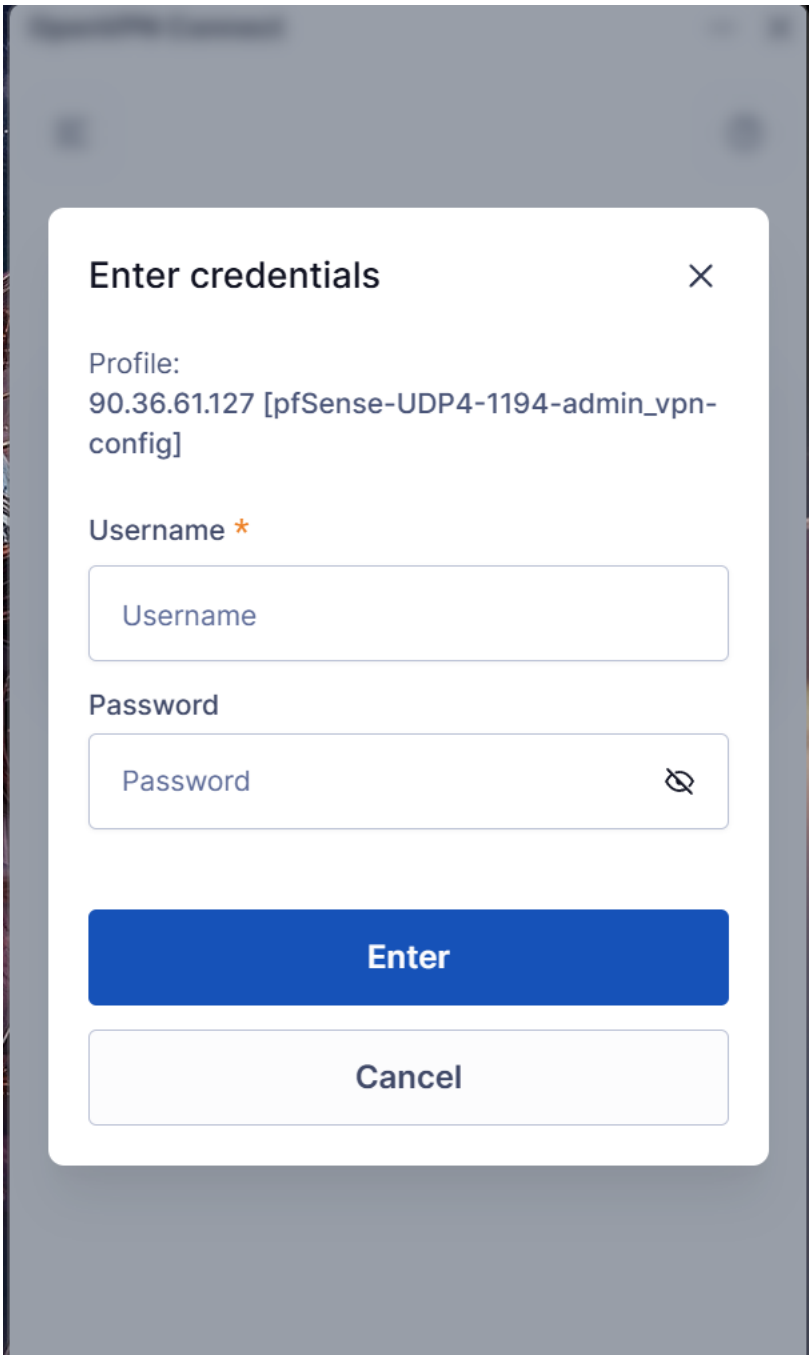


Étape 2 — Import et connexion depuis le WAN

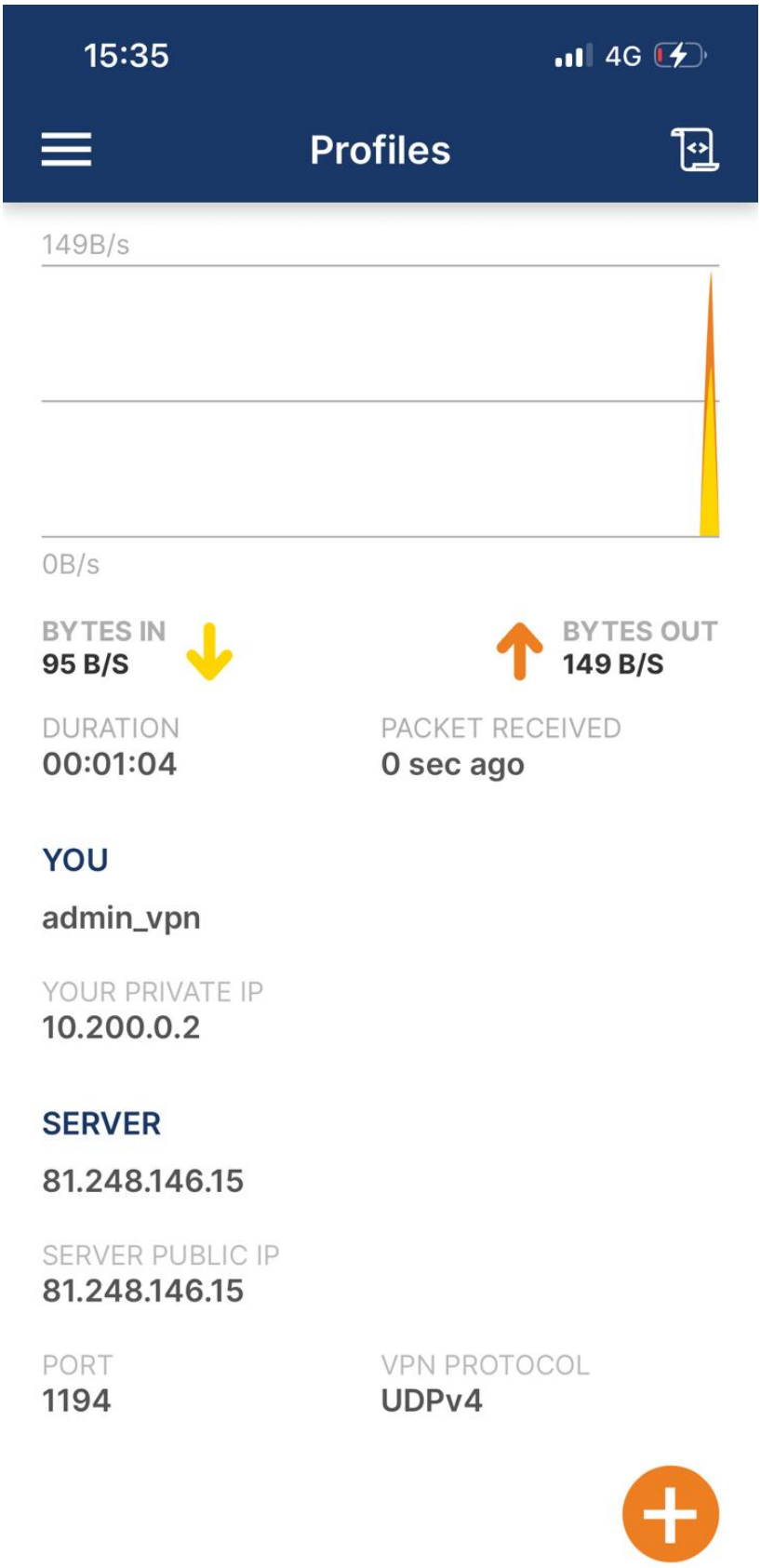
Sur la machine Windows dans la zone WAN, on importe le fichier .ovpn dans le client OpenVPN Connect. On lance la connexion et on saisit les identifiants de l'utilisateur admin_vpn.


OpenVPN Connect
Connect with Server URL or Cloud ID provided by your Admin
Enter your URL or Cloud ID
Continue →
Where to get my URL?
Have a configuration file instead (.ovpn)?
Upload File


OpenVPN Connect
Ready to connect
90.36.61.127 [pfSense-UDP...
90.36.61.127
Connect
Switch Profile

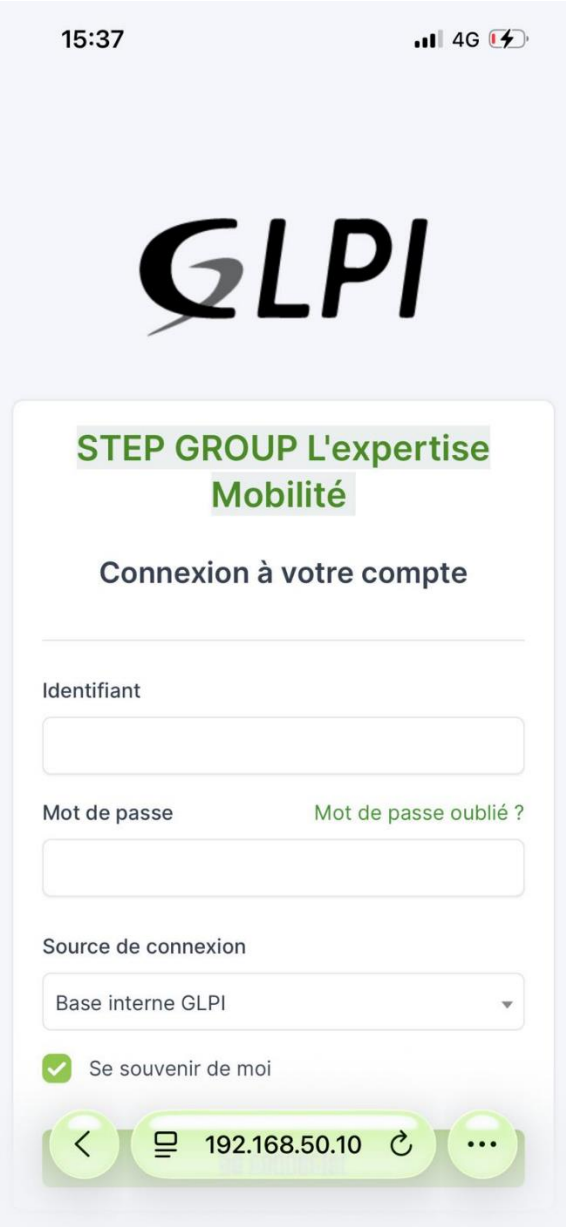


La connexion s'établit. Le client reçoit une IP dans le réseau tunnel (IP_CLIENT_VPN) et la route vers IP_RESEAU_LAN/24 est ajoutée automatiquement à sa table de routage.



Étape 3 — Vérification de l'accès à GLPI

Depuis la machine WAN connectée au VPN, on ouvre un navigateur et on accède à http://IP_GLPI_SERVER/glpi. La page de connexion GLPI s'affiche.



On vérifie aussi que le ping vers IP_GLPI_SERVER fonctionne depuis le client VPN.

VI. Validation et recette

Le tableau ci-dessous présente l'ensemble des tests effectués pour valider le passage de l'architecture vulnérable à l'architecture sécurisée.

ID	Description du test	Résultat attendu	Résultat obtenu	Statut
T01	Accès GLPI via IP WAN sans VPN (NAT désactivé)	Échec — Time Out	Connexion refusée	OK

ID	Description du test	Résultat attendu	Résultat obtenu	Statut
T02	Connexion VPN sans certificat valide	Rejet d'authentification	Accès refusé	OK
T03	Connexion VPN avec certificat + identifiants	Tunnel établi	IP IP_CLIENT_VPN obtenue	OK
T04	Ping vers le serveur GLPI via tunnel	Réponse TTL=64	4/4 paquets reçus	OK
T05	Accès portail GLPI via tunnel	Page de login	Affichage confirmé	OK
T06	Tentative SSH vers serveur (port 22) via tunnel	Blocage pare-feu	Time Out	OK

VII. Maintien en condition opérationnelle (MCO)

La solution ne s'arrête pas à la mise en place. Pour qu'elle reste fiable dans le temps, j'ai défini plusieurs procédures :

- Surveillance des logs VPN : contrôle hebdomadaire des journaux OpenVPN pour détecter des tentatives de connexion inhabituelles. Chemin pfSense : Status > System Logs > VPN.
- Gestion des départs : en cas de départ d'un employé, on révoque son certificat via System > Cert. Manager > Certificates. Le certificat révoqué est ajouté à la CRL et pfSense refuse automatiquement les nouvelles connexions avec ce certificat.
- Sauvegarde de la configuration : export XML de la configuration pfSense avant toute mise à jour via Diagnostics > Backup & Restore. Cela permet de restaurer rapidement en cas de problème.
- Renouvellement des certificats : les certificats ont une durée de 397 jours. Un rappel doit être planifié pour les renouveler avant expiration.

> *Évolution future envisagée : l'ajout d'une authentification TOTP (code temporaire) via FreeRADIUS couplé à pfSense renforcerait encore la sécurité en ajoutant un troisième facteur. La solution pourrait également être étendue pour sécuriser l'accès à d'autres services internes comme le NAS ou l'interface Zabbix.*

VIII. Bilan et analyse critique

8.1 Ce que la solution apporte

Critère	Avant (NAT)	Après (VPN)	Verdict
Visibilité du serveur	Exposé sur internet	Invisible depuis l'extérieur	Amélioré
Authentification	Login + mot de passe	Certificat + login + mot de passe	Sécurisé
Chiffrement	Aucun (HTTP en clair)	AES-256 sur tout le tunnel	Chiffré

Critère	Avant (NAT)	Après (VPN)	Verdict
Surface d'attaque	Maximale	Un seul port UDP 1194	Réduite
Traçabilité	Inexistante	Logs OpenVPN + pare-feu	Assurée

8.2 Développement durable

Ce projet prend en compte les enjeux écoresponsables. La virtualisation sous VMware ESXi regroupe plusieurs machines sur un seul serveur physique, réduisant la consommation énergétique. Le choix de pfSense (open source) et d'OpenVPN (open source) évite toute dépendance à des licences propriétaires coûteuses. La solution de contrôle d'accès par certificats permet de gérer les accès distants sans déplacement physique des techniciens, réduisant ainsi l'empreinte carbone liée aux interventions sur site.

8.3 Limites identifiées

Cette maquette a quelques limites qu'il faudrait adresser avant une mise en production réelle :

- HTTPS non activé sur GLPI : dans la maquette, GLPI tourne en HTTP. Le tunnel chiffre le transport, mais activer HTTPS sur GLPI ajouterait une couche de sécurité supplémentaire.
- Absence de haute disponibilité : si pfSense tombe, l'accès VPN est coupé. En production, une configuration en cluster CARP serait nécessaire.
- Gestion manuelle des certificats : acceptable pour une petite équipe, mais à grande échelle il faudrait automatiser via un système PKI dédié.

8.4 Conclusion

Ce projet m'a permis de comprendre concrètement pourquoi on ne peut pas se contenter d'une redirection de port pour un accès distant sécurisé. En partant d'une démonstration du risque, puis en construisant une solution par étapes (PKI, tunnel, filtrage), j'ai mis en place une architecture qui répond aux exigences de sécurité professionnelle. La solution est fonctionnelle, documentée et maintenue dans le temps.

IX. Annexes

Annexe 1 : Tableau comparatif des solutions étudiées

Critère	NAT Port Forwarding	VPN SSL OpenVPN	Verdict
Visibilité serveur	Publique — exposé à tout internet	Privée — invisible derrière le VPN	Optimisé
Authentification	Login/MDP uniquement	Certificat + Login/MDP (2 facteurs)	Sécurisé
Confidentialité des flux	Nulle — HTTP en clair	Totale — AES-256 (illisible)	Chiffré
Surface d'attaque	Maximale — brute force, scans	Minimale — seul UDP 1194 ouvert	Réduit

Annexe 2 : Plan de tests détaillé

ID	Description	Résultat attendu	Résultat obtenu	Conclusion
T01	Accès HTTP via IP publique (NAT désactivé)	Time Out	Échec de connexion	Brèche comblée
T02	Connexion VPN sans certificat valide	Rejet auth	Accès refusé	PKI protège l'accès
T03	Connexion VPN avec certificat + MDP	Tunnel établi	Connecté, IP tunnel obtenue	Fonctionnel
T04	Accès GLPI au travers du tunnel	Page de login	Succès	Routage OK
T05	Tentative SSH port 22 via tunnel	Blocage pare-feu	Time Out	Filtrage actif