Mode Opératoire

Gestion des accès VPN collègues (WireGuard sur OPNsense)

Code :	MO-NET-007
Version :	1.1
Date :	21 septembre 2026
Auteur :	Cédric LEGRAND
Classification :	USAGE INTERNE — Équipe BTS SIO

Historique des révisions

Version	Date	Modifications
1.0	08/09/2026	Création initiale, validée par le test bout-en-bout du 08/09/2026 (client de test en 4G).
1.1	21/09/2026	Cantonment étendu au VLAN PLT : accès VPN à la plateforme docker-srv (`172.31.250.20`, ports `Ports_Plateforme_Admin`) ; note ICMP (non filtré, hors alias de ports) documentée (MO-NET-009).

1 Objet

Ce mode opératoire décrit la gestion des accès VPN distants des **collègues** sur l'instance **WireGuard** du pare-feu OPNsense : ajouter un collègue (génération du pair, du profil client et du QR code mobile), distribuer le profil par canal sûr, lister les accès en place et révoquer un collègue.

L'instance `wg-collegues` fournit un point d'entrée distant **indépendant du VPS personnel** utilisé pour l'accès historique (voir **MO-NET-001** pour le contexte pare-feu). Le tunnel est volontairement **cantonné au VLAN d'administration** (`10.0.112.0/24`, ports d'administration uniquement) : les postes pédagogiques et tout autre réseau restent inaccessibles, et les tentatives sont journalisées.

La gestion s'effectue **entièrement en ligne de commande** via le script `scripts/opnsense_wg_peer` du dépôt, qui pilote l'API REST d'OPNsense : génération des clés et du secret pré-partagé (PSK), création du pair, application de la configuration, production du fichier `.conf` client et du QR code. Aucune manipulation dans l'interface web n'est nécessaire au quotidien.

2 Champ d'application

Public concerné	Administrateurs de l'infrastructure BTS SIO
Équipement	Pare-feu OPNsense (26.7.x), instance WireGuard `wg-collegues` (interface `wg1`)
Poste d'admin	Linux avec `uv`, `wireguard-tools` (`wg`) et accès HTTPS au pare-feu (`10.0.112.1`)
Accès requis	Identifiants `root` de la WebGUI OPNsense ; coffre Vaultwarden pour la distribution des profils
Durée estimée	5 à 10 minutes par collègue
Référence appliquée	Instance déployée et validée le 08/09/2026 (handshake et cantonnement testés depuis un client 4G)

3 Rappel d'architecture

Le site est derrière un **double NAT** (OPNsense derrière la Livebox Pro). L'accès est donc exposé via une **redirection de port** sur la box, le bail DHCP du WAN OPNsense étant épinglé pour la pérenniser.

```
[Client distant]                  [Livebox Pro]                 [OPNsense]
 10.10.30.x ---- 92.154.7.236:51821 ----> 192.168.1.69 ------> wg1
   (wg-collegues)
  via Internet                       NAT UDP 51821              tunnel
   10.10.30.1/24
                                                                routage ->
   10.0.112.0/24 seul
```

Élément	**Valeur**
Instance OPNsense	`wg-collegues` (interface `wg1`), **VPN > WireGuard**
Écoute	UDP **51821**, MTU 1412, tunnel `10.10.30.1/24`
Endpoint clients	`92.154.7.236:51821` (IP publique de l'offre Pro, a priori fixe)
Redirection Livebox	règle `webui_wireguard-opnsense` : UDP 51821 → `192.168.1.69` (WAN OPNsense)
Bail DHCP Livebox	`192.168.1.69` épinglé à la MAC du WAN OPNsense (`a4:ba:db:41:a5:ca`)
Règle WAN OPNsense	*pass* UDP `any` → `wanip:51821`
Cantonement	*pass* `10.10.30.0/24` → `10.0.112.0/24` ports `Ports_VPN_Admin` ; *pass* `10.10.30.0/24` → `172.31.250.20` ports `Ports_Plateforme_Admin` (plateforme docker-srv, VLAN PLT — depuis le 21/09/2026, MO-NET-009) ; puis *block*+log du reste. ICMP non couvert par les alias de ports : le ping est bloqué par le cantonnement (normal).
Alias `Ports_VPN_Admin`	53 (DNS), 22 (SSH), 443 (GUI), 636 (LDAPS), 3389 (RDP), 5985–5986 (WinRM), 8006 (ProxMox), 10050 (Zabbix agent)
Clé privée serveur	Coffre Vaultwarden, entrée « WireGuard OPNsense wg-collegues — clé privée serveur »

Pas de second facteur sur le tunnel

WireGuard authentifie par clés uniquement (Curve25519) : il n'existe pas de MFA sur le tunnel (contrairement à OpenVPN+TOTP). Ce compromis est assumé ; il est compensé par le cantonnement réseau strict, une paire de clés plus un PSK par collègue, et la révocation immédiate en cas de doute. Le profil client (`.conf` ou QR) contient la clé privée : c'est un secret, à manipuler comme tel.

4 Prérequis

Prérequis

- **Le dépôt de travail est disponible localement ; le script est `scripts/opnsense_wg_peer.py` (exécutable via `uv run`, dépendances auto-installées : `requests`, `qrcode`).**
- **`wireguard-tools` est installé sur le poste (`wg genkey` / `wg pubkey` sont appelés par le script).**
- **Le poste joint `https://10.0.112.1` (sur site, ou via un accès distant existant).**
- **Le mot de passe `root` OPNsense est connu : il est fourni au script via la variable `OPN_PWD` ou saisi interactivement (jamais en clair dans une ligne de commande historisée).**
- **Le collègue dispose de l'application WireGuard (mobile ou poste) et le coffre Vaultwarden est accessible pour la distribution.**

5 Procédure

La logique d'ensemble :

```
Phase 1  Ajouter le collegue (script : cles, pair OPNsense, .conf + QR)
Phase 2  Distribuer le profil par canal sur (Vaultwarden)
Phase 3  Installer cote client (scan QR mobile / import .conf poste)
Phase 4  Lister les acces en place (inventaire)
Phase 5  Revoquer un collegue (depart, perte ou compromission)
```

5.1 Phase 1 — Ajouter un collègue

1 Lancer le script avec le nom du collègue

Depuis la racine du dépôt :

```
cd scripts
uv run opnsense_wg_peer.py manu
```

Le nom est normalisé en *slug* (minuscules ASCII, sans accents). Options utiles : `-ip 12` pour imposer le dernier octet du tunnel (`10.10.30.12`), `-dry-run` pour simuler sans rien modifier.
Le script demande le mot de passe `root` OPNsense (sauf si `OPN_PWD` est exportée), puis enchaîne : génération de la paire de clés du collègue et d'un PSK, création du pair sur l'instance `wg-collegues` via l'API REST, application de la configuration (*reconfigure*), et écriture des artefacts.

2 Relever les artefacts produits

Pour chaque collègue, le script écrit dans `~/.cache/kimi-scratch/vpn-peers-opnsense/<nom>/` (répertoire hors dépôt) :
— `<nom>.conf` — le profil client complet (clé privée, adresse

`10.10.30.x/24`, **DNS** `10.0.112.2` **et** `.3`, **MTU 1412, endpoint, PSK,** `AllowedIPs = 10.0.112.0/24` **en *split tunnel*, *keepalive* 25 s) ;**

— `<nom>_qr.png` — **le même profil encodé en QR code pour l'application mobile.**

Les deux fichiers sont en `chmod 600` **: ils contiennent la clé privée du collègue.**

Vérification immédiate côté serveur

Le pair est visible dès sa création : `uv run opnsense_wg_peer.py -list`**, ou dans la WebGUI VPN > WireGuard > Pairs. Le handshake n'apparaît (VPN > WireGuard > Statut) qu'une fois le client réellement connecté.**

5.2 Phase 2 — Distribuer le profil par canal sûr

3 Ranger le profil dans Vaultwarden

Créer (ou compléter) l'entrée du collègue dans le coffre (voir MO-PLT-005) : y joindre le contenu du `.conf` **en note sécurisée, ou le fichier en pièce jointe. Le collègue le récupère depuis son propre coffre.**

Canaux interdits

Jamais de profil dans Git, par courriel en clair ou par messagerie non chiffrée. Le QR code affiché à l'écran ne doit pas traîner : le montrer uniquement au collègue concerné, au moment de l'enrôlement.

5.3 Phase 3 — Installer côté client

4 **Mobile (Android / iOS)**

Ouvrir l'application WireGuard → + → Scanner le QR code → scanner le QR fourni → nommer le tunnel → activer. Tester avec le Wi-Fi coupé (4G) pour valider le chemin extérieur réel.

5 **Poste (Windows / macOS / Linux)**

Importer le `.conf` dans le client WireGuard (Importer un tunnel), ou en CLI Linux : `wg-quick up <fichier>`. Seul le réseau `10.0.112.0/24` transite par le tunnel (*split tunnel*) ; le reste du trafic du collègue suit sa route habituelle.

5.4 Phase 4 — Lister les accès en place

6 **Inventaire des pairs**

```
uv run opnsense_wg_peer.py --list
```

Chaque ligne donne le nom, l'adresse tunnel attribuée (`10.10.30.x/32`) et l'état activé/désactivé. Pour l'activité réelle (dernier handshake, volumes), consulter VPN > WireGuard > Statut dans la WebGUI.

5.5 Phase 5 — Révoquer un collègue

7 Retirer le pair côté OPNsense

```
uv run opnsense_wg_peer.py manu --remove
```

Le pair est supprimé de l'instance et la configuration appliquée : le tunnel du collègue ne peut plus monter (pas de handshake sans pair enregistré). WireGuard n'a pas de CRL : la révocation est ce retrait, immédiat.

8 Purger les traces locales et le coffre

Supprimer le répertoire d'artefacts `~/.cache/kimi-scratch/vpn-peers-opnsense/<nom>/` (`.conf` et QR contiennent la clé privée révoquée) et purger l'entrée Vaultwarden correspondante. Demander au collègue de supprimer le tunnel de son application si nécessaire.

6 Vérification

Vérification

À l'issue d'un ajout, valider les points suivants (le test en condition réelle se fait depuis l'extérieur, par exemple un téléphone en 4G) :

Création

- ☐ **Le pair apparaît dans `-list` avec l'adresse `10.10.30.x/32` attendue**
- ☐ **Les artefacts `.conf` et QR existent, en `chmod 600`, hors du dépôt Git**

Fonctionnement (depuis un client externe)

- ☐ **Le handshake apparaît dans VPN > WireGuard > Statut (âge en secondes, compteurs rx/tx non nuls)**

- ☐ **Une ressource d'administration est joignable (ex. RDP `10.0.112.2`, ProxMox `https://10.0.112.200:8006`)**
- ☐ **Le DNS passe (`dig @10.0.112.2`)**

Cantonement

- ☐ **Un protocole hors liste est refusé (ex. `ping 10.0.112.2` ou SMB `:445` → délai dépassé)**
- ☐ **La tentative bloquée est visible dans le journal du pare-feu (Système > Fichiers journaux > Pare-feu)**

Hygiène

- ☐ **Le profil est rangé dans Vaultwarden ; aucune copie ne subsiste dans Git ni dans un courriel**

7 Dépannage

Problème	Solution
Pas de handshake (statut vide)	Vérifier dans l'ordre : l'instance `wg-collegues` est activée et écoute (**VPN > WireGuard > Instances**) ; la redirection Livebox existe toujours (règle `webui_wireguard-opnsense`, UDP 51821 → `192.168.1.69`) ; l'IP publique n'a pas changé (comparer l'endpoint du `.conf` à l'IP réelle de la box) ; la règle WAN *pass* UDP 51821 est présente.
Handshake OK mais aucun accès	C'est le cantonnement : seuls le VLAN `10.0.112.0/24` et les ports de `Ports_VPN_Admin` sont autorisés. Vérifier que la ressource visée est bien dans ce périmètre ; sinon c'est le comportement attendu, pas une panne.
IP publique changée	L'offre Pro rend le cas improbable. Si elle change : mettre à jour l'`Endpoint` dans les `.conf` des collègues (régénérer les profils si besoin) et le tableau d'architecture de ce MO.
WAN OPNsense changé d'IP	Le bail est épinglé côté Livebox (`192.168.1.69` ↔ MAC `a4:ba:db:41:a5:ca`) ; si le pare-feu est remplacé (nouvelle MAC), réépingler le bail et vérifier que la redirection NAT pointe toujours sur le bon IP.
Script : login refusé	Le mot de passe `root` fourni (`OPN_PWD` ou saisie) est erroné, ou la WebGUI est injoignable depuis le poste. Tester `https://10.0.112.1` au navigateur.
Script : pair déjà existant	Un pair du même nom existe : choisir un autre nom, ou le retirer d'abord (`-remove`) s'il s'agit d'une régénération (l'ancien profil cesse alors de fonctionner).

8 Voir aussi

— **MO-NET-001** — Vérification post-hardening du pare-feu OPNsense (contrôles après toute modification du filtrage)
— **MO-NET-002** — Mise à jour du firmware OPNsense via l'API REST (même canal d'administration que le script de ce MO)
— **MO-PLT-005** — Ajouter un identifiant dans Vaultwarden (distribution des profils aux collègues)
— **MO-PLT-009** — Connexion à Vaultwarden (récupération du profil par le collègue)

Références documentaires