Diagnostic WireGuard — Linux avec NetworkManager (GNOME)

Guide de diagnostic pour les problèmes de connectivité WireGuard sur Linux Ubuntu avec NetworkManager, notamment l'absence d'accès aux services du réseau local du serveur VPN.

Symptômes typiques

- Internet fonctionne via le VPN (`ping 8.8.8.8` OK)
- Le serveur VPN est joignable (`ping 10.13.13.1` OK)
- Les services accessibles uniquement via VPN (ex: `192.168.1.x`, `10.13.13.x`) sont **inaccessibles**
- Le même profil fonctionne sur un autre appareil (téléphone, autre OS)

Étape 1 — Vérifier l'état du tunnel WireGuard

```
sudo wg show
```

Vérifier que :

- L'interface est bien active
- Un `latest handshake` récent est visible (tunnel actif)
- `allowed ips` contient bien `0.0.0.0/0, ::/0` ou les plages nécessaires

Étape 2 — Vérifier les routes appliquées

```
# Routes globales (le VPN doit apparaître ici)
ip route show

# Routes spécifiques à l'interface WireGuard (remplacer "Home" par le nom de ton interface)
ip route show dev Home
```

Problème détecté si : `ip route show dev Home` ne retourne rien malgré un tunnel actif.

Cela indique que NetworkManager n'a pas injecté les routes dans la table de routage principale — bug connu de NetworkManager avec WireGuard.

Étape 3 — Vérifier les règles de routage

```
ip rule show
```

Exemple de sortie révélant le problème :

```
0:      from all lookup local
31076:  from all lookup main suppress_prefixlength 0
31077:  not from all fwmark 0xcb8e lookup 52110
32766:  from all lookup main
32767:  from all lookup default
```

La règle `not from all fwmark 0xcb8e lookup 52110` indique que WireGuard utilise une table de routage séparée. Vérifier son contenu :

```
ip route show table 52110
```

Étape 4 — Vérifier le DNS

```
resolvectl status
```

Vérifier que l'interface WireGuard apparaît avec le bon serveur DNS (ex: `10.13.13.1`).

```
# Test de résolution DNS via le tunnel
dig @10.13.13.1 monservice.domaine.local
```

Étape 5 — Tester la connectivité par IP directe

Avant de corriger, tester si le problème est DNS ou routage :

```
# Ping du serveur VPN
ping 10.13.13.1

# Accès direct par IP à un service (sans nom de domaine)
curl -v http://IP_DU_SERVICE:PORT
```

- Si ça fonctionne par IP → problème DNS uniquement
- Si ça échoue par IP → problème de routage (continuer ci-dessous)

Correction — Ajouter les routes manquantes

Cas 1 : Le service est sur le réseau WireGuard (`10.13.13.x`)

```
sudo nmcli connection modify <NOM_CONNEXION> +ipv4.routes "10.13.13.0/24 10.13.13.1"
sudo nmcli connection reload
```

Cas 2 : Le service est sur le LAN du serveur (`192.168.1.x` par exemple)

```
sudo nmcli connection modify <NOM_CONNEXION> +ipv4.routes "192.168.1.0/24 10.13.13.1"
sudo nmcli connection reload
```

> ⚠ **Attention** : si ton PC est lui-même sur `192.168.1.x`, ajouter cette route crée un conflit. Dans ce cas, les services doivent être exposés directement sur l'IP WireGuard du serveur (`10.13.13.1`).

Cas 3 : Tout le trafic doit passer par le VPN (full tunnel)

```
sudo nmcli connection modify <NOM_CONNEXION> +ipv4.routes "0.0.0.0/0 10.13.13.1"
sudo nmcli connection reload
```

> ⚠ Cette route remplace la route par défaut. Si elle provoque une perte d'internet, la supprimer immédiatement (voir ci-dessous).

Annuler une modification (rollback)

```
# Supprimer une route ajoutée par erreur
sudo nmcli connection modify <NOM_CONNEXION> -ipv4.routes "CIDR GATEWAY"

# Exemple
sudo nmcli connection modify Home -ipv4.routes "0.0.0.0/0 10.13.13.1"

sudo nmcli connection reload
```

Puis déconnecter et reconnecter le VPN depuis GNOME.

Vérifier la configuration stockée

Les configs NetworkManager sont dans `/etc/NetworkManager/system-connections/` :

```
sudo ls /etc/NetworkManager/system-connections/
sudo cat /etc/NetworkManager/system-connections/<NOM>.nmconnection
```

Exemple de bloc `[ipv4]` correct avec routes :

```
[ipv4]
address1=10.13.13.3/32
dns=10.13.13.1;
method=manual
route1=192.168.1.0/24,10.13.13.1
```

Checklist récapitulatif

Vérification	Commande	Résultat attendu
Tunnel actif	`sudo wg show`	`latest handshake` récent
Routes VPN présentes	`ip route show dev <interface>`	Au moins une route affichée
Règles de routage	`ip rule show`	Table WireGuard présente
DNS correct	`resolvectl status`	DNS = IP du serveur VPN
Ping serveur VPN	`ping 10.13.13.1`	Réponse OK
Accès service par IP	`curl -v http://IP:PORT`	Connexion établie

Causes racines fréquentes

Symptôme	Cause probable
`ip route show dev <iface>` vide	NetworkManager n'injecte pas les routes (bug NM+WireGuard)
DNS OK mais IP inaccessible	Réseau cible absent des `AllowedIPs` ou routes manquantes
IP accessible mais domaine KO	DNS mal configuré ou non utilisé par le système
Internet coupé après modification	Route `0.0.0.0/0` en conflit avec route par défaut WiFi

Revision #1
Created 29 May 2026 09:51:57 by gpatruno
Updated 29 May 2026 09:52:12 by gpatruno