GUIDE DE DEPLOIEMENT WIREGUARD + UNBOUND SUR VPS UBUNTU

Index

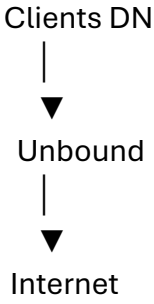
Introduction

Ce document décrit le déploiement d'un résolveur DNS récursif **Unbound** dont le fonctionnement dépend d'un tunnel **WireGuard**, avec contrôle du démarrage via **systemd**.

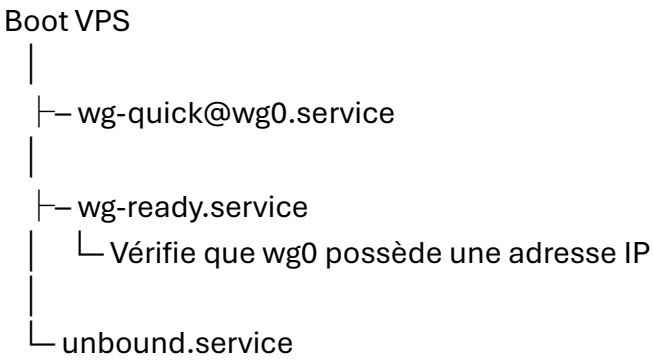
Objectifs :

- Tunnel WireGuard démarré automatiquement au boot.
- Unbound démarré uniquement lorsque le tunnel est opérationnel.
- Validation DNSSEC.
- Support DNS-over-TLS (DoT).
- Procédure de tests et d'audit.
- Vérifications post-déploiement.

Architecture



Séquence de démarrage :



Prérequis

Serveur :

- Ubuntu 24.04 LTS
- Accès root ou sudo
- VPS avec connectivité Internet

Paquets :

sudo apt update
sudo apt install wireguard unbound dnsutils tcpdump -y

Étape 1 : Configuration WireGuard

Créer le fichier :

sudo nano /etc/wireguard/wg0.conf

Exemple :

[Interface]
PrivateKey = <CLE_PRIVEE>
Address = 10.10.10.2/24

[Peer]
PublicKey = <CLE_PUBLIQUE_SERVEUR_DISTANT>
Endpoint = vpn.example.net:51820
AllowedIPs = 0.0.0.0/0, ::/0
PersistentKeepalive = 25

Sécuriser :

sudo chmod 600 /etc/wireguard/wg0.conf

Activation :

sudo systemctl enable wg-quick@wg0
sudo systemctl start wg-quick@wg0

Validation :

sudo wg show

Vérifier l'IP :

ip addr show wg0

Étape 2: Configuration Unbound

Fichier principal :

sudo nano /etc/unbound/unbound.conf.d/recursive.conf

Exemple :

```
server:
  interface: 0.0.0.0
  interface: ::0

  port: 53

  do-ip4: yes
  do-ip6: yes

  do-udp: yes
  do-tcp: yes

  qname-minimisation: yes

  hide-identity: yes
  hide-version: yes

  aggressive-nsec: yes

  prefetch: yes

  harden-glue: yes
  harden-dnssec-stripped: yes

  cache-min-ttl: 300
  cache-max-ttl: 86400

  edns-buffer-size: 1232

remote-control:
  control-enable: yes
```

Vérification :

sudo unbound-checkconf

Démarrage :

sudo systemctl enable unbound
sudo systemctl restart unbound

Étape 3 : Dépendance WireGuard → Unbound

Création du script de contrôle

Créer :

sudo nano /usr/local/bin/wg-ready.sh

Contenu :

```
#!/bin/bash

WG_IF="wg0"
MAX_RETRIES=10
SLEEP_TIME=2

for i in $(seq 1 $MAX_RETRIES); do
  if ip addr show dev $WG_IF | grep -q "inet "; then
    echo "WireGuard $WG_IF est prêt avec IP"
    exit 0
  fi

  sleep $SLEEP_TIME
done

echo "WireGuard $WG_IF n'a pas obtenu d'IP"
exit 1
```

Permissions :

sudo chmod +x /usr/local/bin/wg-ready.sh

Création du service systemd

Créer :

sudo nano /etc/systemd/system/wg-ready.service

Contenu :

```
[Unit]
Description=Wait until WireGuard wg0 has an IP
Requires=wg-quick@wg0.service
After=wg-quick@wg0.service

[Service]
Type=oneshot
ExecStart=/usr/local/bin/wg-ready.sh
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target
```

Modification d'Unbound

Créer un override :

sudo systemctl edit unbound

Contenu :

```
[Unit]
After=wg-ready.service
Requires=wg-ready.service
```

Recharger systemd :

sudo systemctl daemon-reload

Activation :

```
sudo systemctl enable wg-ready.service
sudo systemctl enable unbound
```

Étape 4 : Correction du hostname

Après modification du hostname, vérifier :

hostnamectl

Vérifier :

cat /etc/hostname

Exemple :

srv-btsdns02

Modifier :

sudo nano /etc/hosts

Contenu recommandé :

127.0.0.1 localhost
127.0.1.1 srv-btsdns02

::1 localhost ip6-localhost ip6-loopback

Validation :

sudo hostname

Aucune erreur ne doit apparaître.

Procédure de tests

Test WireGuard

Vérifier :

sudo wg show

Vérifier l'adresse :

ip addr show wg0

Test du service intermédiaire

systemctl status wg-ready.service

Résultat attendu :

Active: active (exited)

Test Unbound

systemctl status unbound

Résultat attendu :

Active: active (running)

Validation de la résolution DNS

dig example.com @127.0.0.1

Résultat attendu :

status: NOERROR

Validation DNSSEC

dig dnssec-failed.org @127.0.0.1

Résultat attendu :

SERVFAIL

Puis :

dig cloudflare.com @127.0.0.1

Le flag :

ad

doit apparaître.

Validation DNS-over-TLS

dig example.com @127.0.0.1 +tls

Résultat attendu :

SERVER: 127.0.0.1#853 (TLS)

Validation après reboot

Redémarrer :

sudo reboot

Contrôler :

systemctl status wg-quick@wg0
systemctl status wg-ready.service
systemctl status unbound

Les trois services doivent être actifs.

Vérification du trafic DNS dans le tunnel

Lancer :

sudo tcpdump -i wg0 port 53 (pour le traffic en clair)

Dans un second terminal :

dig google.com @127.0.0.1

Des paquets DNS doivent apparaître sur l'interface WireGuard.

Vérifications d'audit

Configuration

sudo unbound-checkconf

État WireGuard

sudo wg show

État des services

systemctl status wg-quick@wg0
systemctl status wg-ready.service
systemctl status unbound

Vérification des dépendances

systemctl show unbound | grep -E "After=|Requires="

Résultat attendu :

After=wg-ready.service
Requires=wg-ready.service

Critères de conformité

Contrôle	Résultat attendu
WireGuard actif	Oui
Adresse IP sur wg0	Oui
wg-ready actif	Oui
Unbound actif	Oui
DNS récursif fonctionnel	Oui
DNSSEC valide	Oui
DNS-over-TLS actif	Oui
Démarrage automatique après reboot	Oui
Unbound dépendant de WireGuard	Oui
Hostname résolu localement	Oui

Ce document constitue une base d'exploitation et d'audit adaptée à un environnement de production.