
WireGuard

Índex

1. Què és WireGuard i com funciona	1
2. Funcionalitats principals	2
3. Escenari d'aquest document	2
4. Instal·la a Ubuntu 26.04	2
4.1. Al servidor i al client	3
5. Configura el servidor	3
5.1. Genera el parell de claus	3
5.2. Habilita el reenviament de paquets IPv4	3
5.3. Identifica la interfície de sortida a Internet	4
5.4. Crea el fitxer /etc/wireguard/wg0.conf	4
5.5. Obre el firewall (UFW)	4
5.6. Activa la interfície	5
6. Configura el client	5
6.1. Genera claus al client	5
6.2. Crea /etc/wireguard/wg0.conf al client	5
6.3. Activa el túnel al client	6
7. Verifica	6
8. Afegeix més clients	7
9. Resolució de problemes	7
10. Referències i documentació	8
Índex de figures	9

WireGuard és un protocol de VPN (Virtual Private Network) modern, dissenyat per ser molt més senzill, ràpid i segur que les solucions clàssiques com OpenVPN o IPsec. Es va integrar al nucli de Linux a partir de la versió 5.6, i des d'aleshores està disponible “de sèrie” a la majoria de distribucions, incloent-hi Ubuntu.



Figura 1: WireGuard logo

A diferència d'OpenVPN o IPsec, que arrossegueu dècades d'opcions de xifratge i configuracions per motius de compatibilitat, WireGuard aposta per un únic conjunt de primitives criptogràfiques modernes, sense negociació de protocols ni PKI (infraestructura de clau pública) complexa. Això es tradueix en un codi base molt reduït (menys de 4.000 línies), fàcil d'auditar, i en un rendiment superior gràcies a la seva implementació a escala de nucli.

NOTA

Aquest document fa servir Ubuntu 26.04 LTS com a sistema de referència, tant per al servidor com per al client. Els passos són pràcticament idèntics a Ubuntu 24.04 i 22.04, amb la diferència que en versions més antigues cal instal·lar `resolvconf` o `openresolv` per gestionar el DNS del túnel.

1. Què és WireGuard i com funciona

WireGuard crea una interfície de xarxa virtual (normalment anomenada `wg0`) que estableix un túnel xifrat punt a punt entre dos o més equips (peers). No fa distinció estricta entre “client” i “servidor”: tots els extrems són peers, encara que a la pràctica sol haver-hi un equip amb IP pública fent de punt central.

El funcionament es basa en el que s'anomena **Cryptokey Routing**:

- Cada interfície té un parell de claus (privada i pública), generades amb Curve25519.
- Cada peer es configura amb la clau pública de l'altre extrem i un rang d'adreces IP permeses (`AllowedIPs`).
- El trànsit es xifra amb ChaCha20 (xifratge simètric) i s'autentica amb Poly1305, mentre que BLAKE2s s'utilitza per al hashing.
- No hi ha negociació de xifratges ni certificats: si tens la clau pública correcta, la connexió funciona; si no, els paquets simplement es descarten en silenci.

Aquest disseny minimalista és el que li dona el seu nom col·loquial de “protocol d'una sola opinió”: no cal triar entre desenes de combinacions de xifratge com a OpenVPN.

2. Funcionalitats principals

- **Rendiment:** en ser un mòdul del nucli, evita el canvi de context entre espai d'usuari i espai de nucli que pateixen OpenVPN o IPsec en mode usuari, oferint latència més baixa i més amplada de banda.
- **Configuració mínima:** un fitxer `.conf` amb dues seccions (`[Interface]` i `[Peer]`) sol ser suficient.
- **Roaming transparent:** un peer pot canviar d'IP (per exemple, en passar de WiFi a dades mòbils) i el túnel es recupera automàticament sense reiniciar la connexió.
- **Multiplataforma:** clients oficials per a Linux, Windows, macOS, Android, iOS i BSD.
- **Sense estat de connexió complex:** no manté sessions TCP; funciona sobre UDP, cosa que simplifica el NAT transversal.
- **Codi auditable:** base de codi petita, revisada formalment, cosa que en redueix la superfície d'atac.
- **Cryptokey Routing:** combina l'encaminament i l'autenticació en un sol mecanisme basat en claus públiques, eliminant la necessitat d'una PKI.

NOTA

WireGuard no inclou, per si mateix, mecanismes de gestió dinàmica d'usuaris ni portal captiu. Per a desplegaments amb molts clients es recomana complementar-lo amb eines com `wg-easy`, `HeadScale` (implementació *open source* del pla de control de Tailscale) o scripts d'aprovisionament propis.

3. Escenari d'aquest document

Dues màquines amb Ubuntu 26.04 LTS:

Rol	Nom	IP pública/xarxa	IP dins del túnel
Servidor	<code>wg-server</code>	IP pública amb UDP/51820 obert	10.8.0.1/24
Client	<code>wg-client</code>	Xarxa local/Internet	10.8.0.2/24

L'objectiu és que el client es connecti al servidor, i opcionalment que tot el trànsit d'Internet del client passi pel túnel (mode "gateway complet").

4. Instal·la a Ubuntu 26.04

A Ubuntu 26.04, el nucli ja incorpora el mòdul WireGuard, de manera que només cal instal·lar les eines d'espai d'usuari (`wg`, `wg-quick`) des dels dipòsits oficials.

4.1. Al servidor i al client

Actualitza la llista de paquets:

```
sudo apt update
```

Instal·la WireGuard:

```
sudo apt install wireguard
```

Comprova la versió instal·lada:

```
wg --version
```

NOTA

El paquet wireguard és un metapaquet que instal·la wireguard-tools, el qual proporciona les ordres wg i wg-quick, així com la plantilla de servei systemd wg-quick@.service. Com que el mòdul de nucli ja hi és present a Ubuntu 26.04, no cal DKMS ni dipòsits de tercers.

5. Configura el servidor

5.1. Genera el parell de claus

```
sudo install -d -m 700 /etc/wireguard  
cd /etc/wireguard  
umask 077  
wg genkey | sudo tee server_private.key | wg pubkey | sudo tee  
↪ server_public.key
```

Això crea `server_private.key` (clau privada, que **no s'ha de compartir mai**) i `server_public.key` (clau pública, que es proporcionarà als clients).

5.2. Habilita el reenviament de paquets IPv4

Perquè el servidor pugui encaminar el trànsit dels clients cap a Internet:

```
sudo sysctl -w net.ipv4.ip_forward=1  
echo 'net.ipv4.ip_forward=1' | sudo tee  
↪ /etc/sysctl.d/99-wireguard.conf
```

5.3. Identifica la interfície de sortida a Internet

```
ip route get 1.1.1.1
```

Anota el nom de la interfície (per exemple eth0 o ens3); la necessitaràs a les regles de NAT.

5.4. Crea el fitxer /etc/wireguard/wg0.conf

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

```
[Interface]
Address = 10.8.0.1/24
ListenPort = 51820
PrivateKey = <contingut_de_server_private.key>
PostUp = iptables -A FORWARD -i wg0 -j ACCEPT; iptables -t nat -A
↳ POSTROUTING -o eth0 -j MASQUERADE
PostDown = iptables -D FORWARD -i wg0 -j ACCEPT; iptables -t nat -D
↳ POSTROUTING -o eth0 -j MASQUERADE

# Es va afegint un bloc [Peer] per cada client
[Peer]
PublicKey = <clau_pública_del_client>
AllowedIPs = 10.8.0.2/32
```

```
sudo chmod 600 /etc/wireguard/wg0.conf
sudo chown root:root /etc/wireguard/wg0.conf
```

AVÍS

Substitueix eth0 pel nom real de la teva interfície de sortida. Si t'equivoques, el NAT no funcionarà i els clients no podran sortir a Internet a través del túnel, encara que el handshake es completi correctament.

5.5. Obre el firewall (UFW)

```
sudo ufw allow 51820/udp
sudo ufw allow OpenSSH
sudo ufw enable
```

Si fessis servir UFW i necessites que reenvii trànsit, edita /etc/default/ufw i canvia DEFAULT_FORWARD_POLICY="DROP" per DEFAULT_FORWARD_POLICY="ACCEPT", i després recarrega:

```
sudo ufw reload
```

5.6. Activa la interfície

```
sudo systemctl enable --now wg-quick@wg0  
sudo systemctl status wg-quick@wg0
```

6. Configura el client

Els passos d'instal·lació són idèntics als del servidor. Actualitza la llista de paquets:

```
sudo apt update
```

Instal·la WireGuard:

```
sudo apt install wireguard
```

6.1. Genera claus al client

```
sudo install -d -m 700 /etc/wireguard  
cd /etc/wireguard  
umask 077  
wg genkey | sudo tee client_private.key | wg pubkey | sudo tee  
↪ client_public.key
```

Copia el contingut de `client_public.key` i afegeix-lo com a `PublicKey` dins del bloc `[Peer]` del `wg0.conf` del **servidor** (pas 4 anterior), i reinicia el servei al servidor:

```
sudo systemctl restart wg-quick@wg0
```

6.2. Crea `/etc/wireguard/wg0.conf` al client

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

```
[Interface]  
Address = 10.8.0.2/24  
PrivateKey = <contingut_de_client_private.key>  
DNS = 1.1.1.1  
  
[Peer]  
PublicKey = <clau_pública_del_servidor>  
Endpoint = <IP_pública_del_servidor>:51820  
AllowedIPs = 0.0.0.0/0  
PersistentKeepalive = 25
```

```
sudo chmod 600 /etc/wireguard/wg0.conf
```

NOTA

`AllowedIPs = 0.0.0.0/0` fa que **tot** el trànsit del client passi pel túnel (mode VPN complet). Si només vols accedir a la xarxa interna del servidor, posa-hi el rang concret (per exemple `10.8.0.0/24` o la subxarxa de la teva LAN).

`PersistentKeepalive = 25` és important quan el client és darrere d'un NAT (per exemple, en una xarxa domèstica), ja que envia un paquet buit cada 25 segons per mantenir oberta la traducció d'adreces al router.

Si el client és Ubuntu 26.04 o 24.04, `systemd-resolved` ja proporciona la compatibilitat amb `resolvconf` necessària per gestionar la línia DNS =. En Ubuntu 22.04 caldria instal·lar `openresolv`:

```
which resolvconf
# si no retorna cap camí, en 22.04:
sudo apt install openresolv
```

6.3. Activa el túnel al client

```
sudo systemctl enable --now wg-quick@wg0
```

O bé, per fer una prova puntual sense deixar-lo actiu en arrencar:

```
sudo wg-quick up wg0
```

7. Verifica

Al servidor o al client, comprova l'estat de la interfície i el handshake:

```
sudo wg show
```

Una sortida correcta mostra `latest handshake` amb un temps recent i els comptadors transfer incrementant-se:

```
interface: wg0
  public key: <clau_pública_local>
  private key: (hidden)
  listening port: 51820

peer: <clau_pública_remota>
  endpoint: 203.0.113.10:51820
  allowed ips: 0.0.0.0/0
  latest handshake: 12 seconds ago
```

```
transfer: 1.24 KiB received, 2.87 KiB sent
```

Prova també la connectivitat dins del túnel:

```
ping -c 3 10.8.0.1
```

I, si has configurat el mode de gateway complet, comprova que la IP pública de sortida és la del servidor:

```
curl ifconfig.me
```

8. Afegeix més clients

Per cada nou client, cal:

1. Generar-li un parell de claus propi (mai reutilitzis claus entre dispositius).
2. Afegir un nou bloc [Peer] al `wg0.conf` del servidor, amb la seva clau pública i una IP única dins del rang (per exemple `10.8.0.3/32`).
3. Reiniciar (o recarregar) la interfície al servidor:

```
sudo wg syncconf wg0 <(wg-quick strip wg0)
```

Aquesta ordre aplica els canvis sense tallar les connexions actives dels altres peers, a diferència de `systemctl restart`.

9. Resolució de problemes

ACTIVITAT

Si el `latest handshake` no apareix mai (o mostra "(none)"), segueix aquest procés de comprovació abans de tocar res més:

1. Verifica que el port UDP 51820 estigui obert al firewall del servidor i en qualsevol firewall de xarxa intermedi (router, proveïdor cloud, grup de seguretat).
2. Confirma que `Endpoint` al client apunta a la IP i port correctes del servidor.
3. Comprova que les claus públiques estan intercanviades correctament (la clau pública del servidor va al client, i viceversa; és un error molt freqüent barrejar-les).
4. Revisa que `AllowedIPs` no s'encavalqui entre diferents peers al servidor.

Altres problemes habituals:

- **El túnel s'aixeca, però no hi ha Internet al client:** normalment és un problema de NAT/MASQUERADE al servidor. Revisa que la interfície indicada a `PostUp/PostDown` sigui la correcta i que `net.ipv4.ip_forward` estigui a 1.
- **Es perd la connexió després d'un temps:** falta `PersistentKeepalive` si el client és darrere d'un NAT.

- **DNS no resol dins del túnel:** comprova que `resolvconf` (via `systemd-resolved`) funciona correctament i que la línia DNS = és present al [Interface] del client.

10. Referències i documentació

- Lloc oficial del projecte: <https://www.wireguard.com/>
- Documentació d'instal·lació oficial: <https://www.wireguard.com/install/>
- Explicació tècnica del protocol (whitepaper): <https://www.wireguard.com/papers/wireguard.pdf>
- Manual de `wg` i `wg-quick`: `man wg`, `man wg-quick` (un cop instal·lat `wireguard-tools`)
- Documentació d'Ubuntu Server sobre WireGuard: <https://ubuntu.com/server/docs>
- Headscale (implementació *open source* d'un pla de control estil Tailscale, útil per a gestionar molts peers): <https://github.com/juanfont/headscale>

Índex de figures

1	WireGuard logo	1
---	--------------------------	---

Versions d'aquest document

- [HTML](#)
- [PDF](#)
- [ODT](#)
- [MD](#)

[Domini Públic \(CC0\)](#)