Virtual Private Networks

By Kevin Hubbard

Have you ever:

Had to pay a bill while using public wifi?

or

Wanted to watch a movie you own away from home?

The Solution

VPN's

Three main types...

- Remote Access
- Site to Site
- Client to Internet

What is a Virtual Private Network

At its core, a VPN uses the idea of a **secure tunnel** over an untrusted network (like the internet).

- It encrypts your traffic
- Ecapsulates it within another packet
- Sends it to a trusted endpoint
- Endpoint decrypts and forwards it

VPN protocols are sets of rules determining how data is encrypted and transmitted

1. Wireguard
2. OpenVPN
3. IPSec

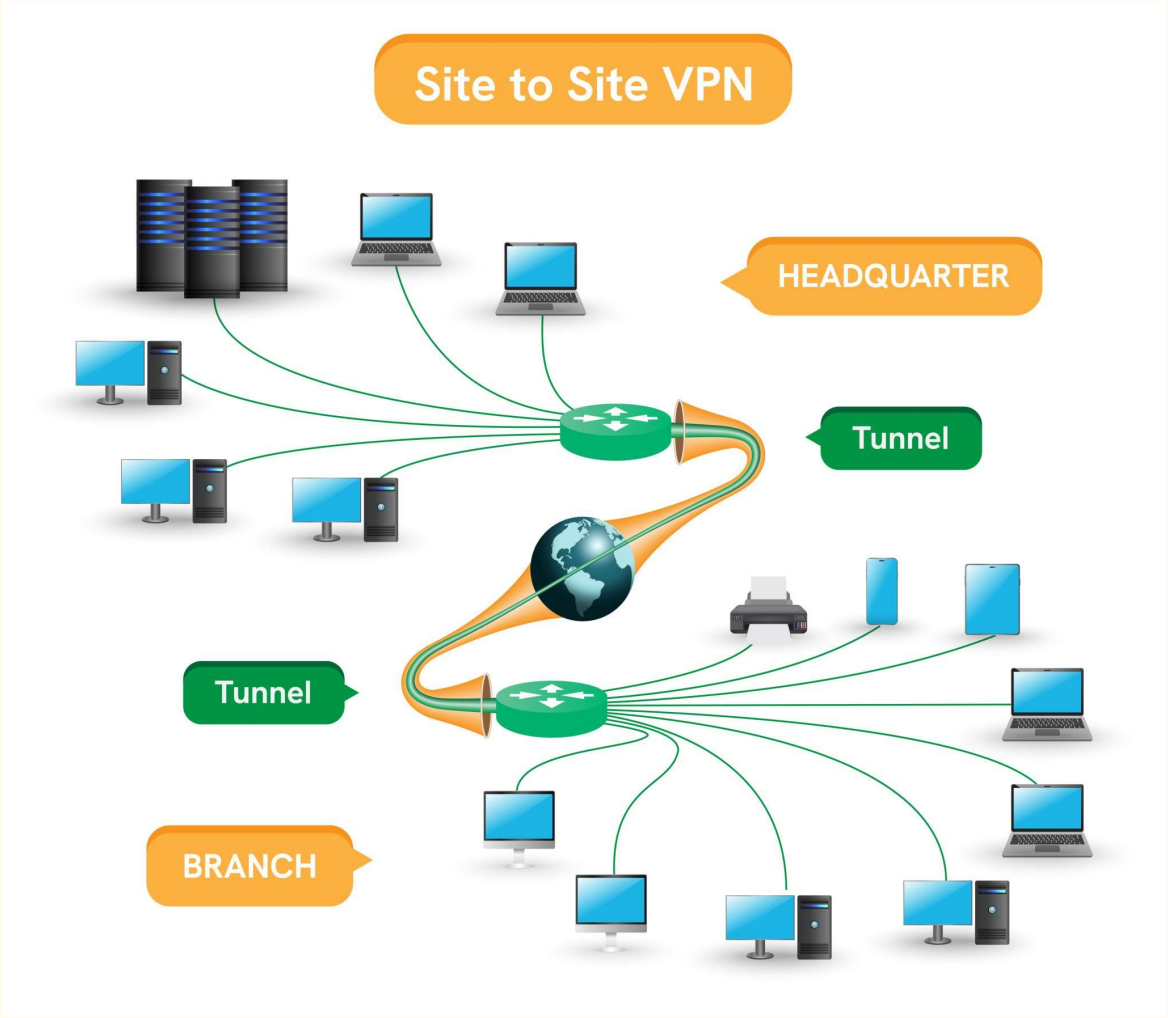
Remote Access VPN’s

Connects one device to a private network

- Used for:
  - Accessing home lab/media server
  - Remote work

Traffic Flow: You → encrypted tunnel → your home → internet

To websites you appear as your homes IP address


Site to Site VPN
HEADQUARTER
Tunnel
Tunnel
BRANCH

Client to Internet VPN's (commercial vpn)

Connects your device throught a vpn provider to the internet

- Used for:
  - Privacy
  - Hiding your IP
  - Bypassing IP Geo-restrictions

Traffic Flow: You → encrypted tunnel → VPN server (Germany, Japan, etc.) → internet

Examples: NordVPN and ExpressVPN

To Websites you appear as that servers ip/location

Creating a VPN using:

Installation Steps

1. Install WireGuard
2. Generate keys (server first)
3. Create `wg0.conf`
4. Enable IP forwarding
5. Bring up the interface with `wg-quick`
6. Add clients (peers)
7. Port forward UDP 51820

On any server:

1. Install WireGuard

```
sudo apt update
sudo apt install wireguard
```

2. Generate server keys first

```
cd /etc/wireguard
sudo umask 077
sudo wg genkey | sudo tee server_private.key | sudo wg pubkey | sudo tee server_public.key
```

```
kevin@linux-media-server:~$ sudo -i
root@linux-media-server:~# cd /etc/wireguard/
root@linux-media-server:/etc/wireguard# ls
server_private.key  server_public.key  wg0.conf
```

On any server:

3. Create the server config file

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

```
[Interface]
Address = 10.8.0.1/24
ListenPort = 51820
PrivateKey = <contents of server_private.key>
SaveConfig = true
```

(After the interface heading is complete we will add peers to the server...)

On any server:

4. Enable IP forwarding

```
echo "net.ipv4.ip_forward=1" | sudo tee /etc/sysctl.d/99-wireguard-forward.conf
sudo sysctl --system
```

5. Allow traffic forwarding (NAT)

```
sudo ufw allow 51820/udp
sudo ufw route allow in on wg0 out on eth0
```

6. Bring up WireGuard

```
kevin@linux-media-server:~$ sudo wg
[sudo] password for kevin:
interface: wg0
  public key: c2vBHVFjezO90E1iYDD9oDthbxJTZ8Gg05RvtF9e7QU=
  private key: (hidden)
  listening port: 51820
```

Creating a Peer:

1. On client or server generate keys

```
wg genkey | tee client1_private.key | wg pubkey | tee client1_public.key
```

2. Add client as peer (on server)

```
sudo wg set wg0 peer <client1_public_key> allowed-ips 10.8.0.2/32
```

Creating a Peer:

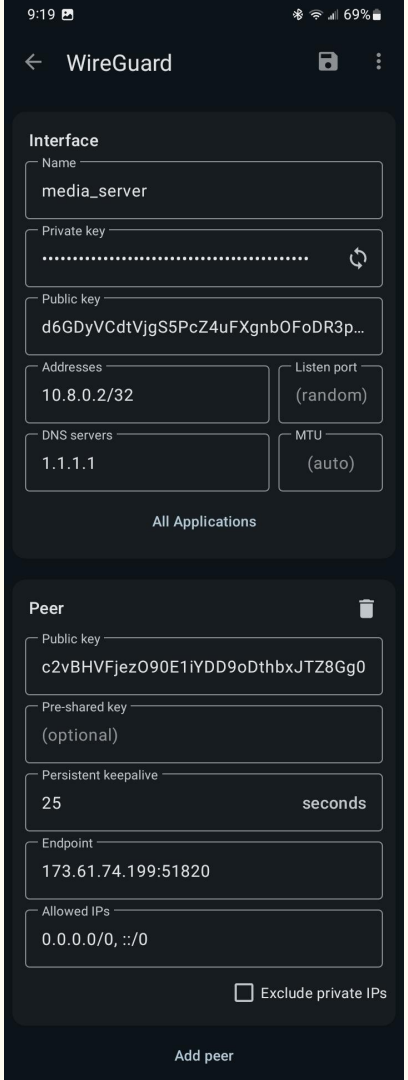
kevin@linux-media-server: ~

```
root@linux-media-server:/etc/wireguard# cat wg0.conf
[Interface]
Address = 10.8.0.1/24
SaveConfig = true
ListenPort = 51820
PrivateKey = [redacted]
=

[Peer]
PublicKey = d6GDyVCdtVjgS5PcZ4uFXgnbOFoDR3p36jvq7tAvQQY=
AllowedIPs = 10.8.0.2/32
Endpoint = 173.61.74.199:59371

[Peer]
PublicKey = YLgWKQ7by4n4jbRsm8pFiHTPpD9x467RRdr0c+gytzo=
AllowedIPs = 10.8.0.3/32
Endpoint = 173.61.74.199:52555

[Peer]
PublicKey = 0Vdx288XI/VDW+7i1kkA2k92HSpDrc9llzroHBbgvQ0=
AllowedIPs = 10.8.0.4/32
Endpoint = 130.156.128.31:59241
```



verizon Basic Advanced Help

Network Devices

Fios Router

- Home
- Wi-Fi
- Devices
- Security & Firewall
  - General
  - Access Control
  - DMZ Host
  - IPv6 Pinholes
  - Port Forwarding
  - Port Forwarding Rules
  - Port Triggering
  - Scheduler Rules
  - SIP ALG
- Network Settings
- Diagnostics & Monitoring

Security & Firewall > Port Forwarding

Port Forwarding

Apply Changes

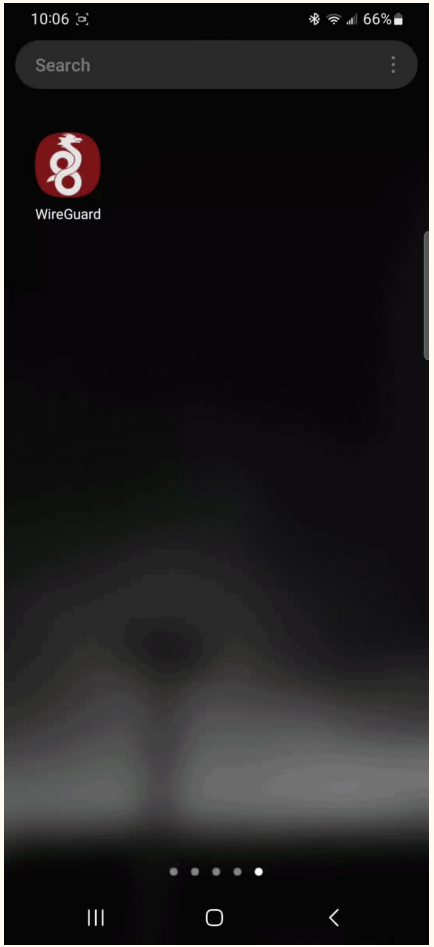
Open a tunnel between remote computers and a device port on your Home Network (LAN). Supports gaming, IoT, home security devices and more.

	35002	TCP	192.168.1.101	8082	Always	
	63146	UDP	192.168.1.101	63145	Always	
	35003	TCP	192.168.1.100	7547	Always	
	63147	UDP	192.168.1.100	63145	Always	
	35001	TCP	192.168.1.102	8082	Always	
	35000	TCP	192.168.1.103	7547	Always	
	63145	UDP	192.168.1.102	63145	Always	
KevinHTTPServer	80	TCP	192.168.1.231	80	Always	☑ ✎ 🗑 ⌄
KevinHTTPSServer	443	TCP	192.168.1.231	443	Always	☑ ✎ 🗑 ⌄
webSSH	[redacted]	TCP	192.168.1.231	22	Always	☑ ✎ 🗑 ⌄
VPN	51820	UDP	192.168.1.234	51820	Always	☑ ✎ 🗑 ⌄

Starting the VPN:

```
sudo wg-quick up laptop
```

```
kevin@debianLaptop: ~
kevin@debianLaptop:~$ vpnup
[#] ip link add debianLaptop type wireguard
[#] wg setconf debianLaptop /dev/fd/63
[#] ip -4 address add 10.8.0.4/32 dev debianLaptop
[#] ip link set mtu 1420 up dev debianLaptop
[#] resolvconf -a tun.debianLaptop -m 0 -x
[#] wg set debianLaptop fwmark 51820
[#] ip -4 route add 0.0.0.0/0 dev debianLaptop table 51820
[#] ip -4 rule add not fwmark 51820 table 51820
[#] ip -4 rule add table main suppress_prefixlength 0
[#] sysctl -q net.ipv4.conf.all.src_valid_mark=1
[#] nft -f /dev/fd/63
kevin@debianLaptop:~$
```



https://www.wireguard.com/quickstart/

Thanks!