WireGuard - VPN-Server

Quellen:

- Wireguard
- Wireguard unter Debian 11
- Wireguard: Private-Key mit GPG verschlüsselt speichern
- https://oliver-kaestner.de/posts/anleitung-wireguard-vpn-server-einrichten-internetrouting/

Server-Installation

```
sudo apt update
```

```
sudo apt install wireguard
```

Schlüssel generieren

Für den Server und für jeden Client wird je ein privater und ein daran gebundener öffentlicher Schlüssel generiert. Für die Funktionalität dürfen sie weder vertauscht noch verändert werden.
Der private Schlüssel ist sicherheitsrelevant und sollte entsprechend behandelt werden.

Die Schlüssel können prinzipiell auf einem beliebigen (Linux-)Rechner generiert werden.
Eine Kopie der Schlüssel/Keys wird in den im folgenden genannten Verzeichnissen abgelegt.
Die Generierung erfolgt hier als root (ohne sudo)

Server-Schlüssel - privat (key)

```
wg genkey | sudo tee /etc/wireguard/server.key
```

Leserechte des privaten Server-Keys auf root begrenzen

```
chmod 0400 /etc/wireguard/server.key
```

Server-Schlüssel öffentlich (pub) aus dem privaten Schlüssel generieren

```
cat /etc/wireguard/server.key | wg pubkey | sudo tee
/etc/wireguard/server.pub
```

Client-Schlüssel - privat (key)

```
mkdir -p /etc/wireguard/clients
```

```
wg genkey | tee /etc/wireguard/clients/client1.key
```

Client-Schlüssel öffentlich (pub) aus dem privaten Schlüssel generieren

```
cat /etc/wireguard/clients/client1.key | wg pubkey | tee
/etc/wireguard/clients/client1.pub
```

Client1 kann beliebig benannt werden - z.B. mit dem Namen eines Users oder eines Laptops. Eine eindeutige Bezeichnung hilft die Schlüssel zu identifizieren, wenn Freigaben erteilt oder entzogen werden müssen.

Zur Nutzung werden die Schlüssel wie folgt hinterlegt:

auf dem Server in der Server-Konfiguration[1)]

- privater Key des Servers
- öffentliche Keys aller Clients

auf jedem Client in der Client-Konfiguration[2)]:

- privater Key des Clients
- öffentlicher Key des Servers

Server einrichten

Server-Konfiguration erstellen

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

```
[Interface]
# Wireguard Server private key - server.key
PrivateKey = <PRIVATKEY-SERVER>
# Wireguard interface will be run at 10.8.0.1
Address = 10.8.0.1/24
# Clients will connect to UDP port 51820
ListenPort = 51820
# Ensure any changes will be saved to the Wireguard config file
SaveConfig = true

[Peer]
# Client public key - client1.pub
PublicKey = <PUBLIC-KEY-CLIENT1>
# clients' VPN IP addresses you allow to connect
AllowedIPs = 10.8.0.2/32

[Peer]
# weitere Clients - Laptop13.pup
PublicKey = <PUBLIC-KEY-Laptop13>
```

```
AllowedIPs = 10.8.0.3/32
```

Tunnel, Port-Forwarding

Um den gesamten Traffic über den Tunnel zu leiten (erforderlich für den aktiven Client für Internetdaten, Zugriff auf Remote-Desktops, etc.) ist Port-Forwarding erforderlich.
Ist die Netzverwaltung auf dem Server mit Netplan realisiert: siehe Tunnel mit Netplan.

```
nano /etc/sysctl.conf
```

Einfügen in die Datei

```
# Port Forwarding for IPv4
net.ipv4.ip_forward=1
# Port forwarding for IPv6
net.ipv6.conf.all.forwarding=1
```

Änderungen übernehmen

```
sysctl -p
```

Firewall

siehe Firewall

```
ufw allow OpenSSH
```

Tunnelendpunkt erreichbar machen

```
sudo ufw allow 51820/udp
```

Daten aus dem Tunnel annehmen

```
sudo ufw allow in on wg0
```

Weiterleiten von Internetverkehr

```
sudo ufw route allow in on wg0
```

```
sudo ufw route allow out on wg0
```

```
ufw reload
```

Wireguard-Server starten

```
systemctl start wg-quick@wg0.service
```

enable = in Zukunft automatisch mit dem System starten

```
systemctl enable wg-quick@wg0.service
```

```
systemctl status wg-quick@wg0.service
```

Schnittstelle wg0 überprüfen

```
ip a show wg0
```

Verbindungsinformationen anzeigen

```
sudo wg
```

Linux Client einrichten

```
sudo apt install wireguard-tools
```

Konfigurationsdatei erstellen.
Diese Datei kann auch für den Import mit einem entsprechenden Client-Programm genutzt werden - Windows oder Linux.

```
sudo nano /etc/wireguard/client1.conf
```

```
[Interface]
# Define the IP address for the client - must be matched with wg0 on
Wireguard Server
Address = 10.8.0.2/32
# specific DNS Server
DNS = 192.168.178.1
# Private key for the client
PrivateKey = <PRIVATKEY-CLIENT1>

[Peer]
# Public key of the Wireguard server - server.pub
PublicKey = <PUBLICKEY-SERVER>
# Allow all traffic to be routed via Wireguard VPN
AllowedIPs = 0.0.0.0/0
# Public IP address of the Wireguard Server
Endpoint = <IP_SERVER>:51820
# Sending Keepalive every 25 sec
PersistentKeepalive = 25
```

Unter DNS ist die IP-Adresse für den Internetzugang auf dem Server einzutragen. Da sämtliche Internetdaten durch den Tunnel geleitet werden, benötigt der Client die Route auf dem Server, wenn er parallel eine Internetverbindung nutzen möchte[3)].

Wireguard starten

Hier wird die Verwaltung über die Konsole beschrieben. Wird das Linux-System über eine GUI betrieben, ist es auch möglich, die o.a. client1.conf-Datei über die Netzwerk-Verwaltung zu importieren.

```
wg-quick up client1
```

Evtl. installieren bei Fehlermeldung beim Start VPN

```
sudo apt install openresolv
```

Schnittstelle prüfen

```
ip a show client1
```

Verbindung anzeigen » einmal auf dem Client und einmal auf dem Server

```
sudo wg show
```

Server-Zugriff vom Client testen

```
ping -c5 10.8.0.1
```

```
ping -c5 1.1.1.1
```

```
ping -c5 duckduckgo.com
```

Wireguard beenden (Server und Client)

```
wg-quick down client1
```

Windows-Client einrichten

1. Installationsdatei herunterladen
2. Installieren
3. Konfigurationsdatei über das installierte Programm importieren, siehe Linux Client einrichten einrichten
4. Verbindung aktivieren/deaktivieren

[1)]

Datei: /etc/wireguard/wg0.conf

[2)]

Datei: /etc/wireguard/client1.conf

[3)]

Webseiten aufrufen, E-Mail abrufen, etc.