Netzwerk
Wireguard VPN

- GNU GPLv2 Open Source
- modern, stabil, einfach konfigurierbar, roamingfähig, schlank und schnell (höhere Übertragungsgeschwindigkeit, geringere Latenz)
- schneller als IPSec oder OpenVPN
- schlanker Code und geringe Komplexität
- energieeffizient, kaum Daten im Leerlauf, geeignet für Mobilgeräte
- kostenloser Client für alle gängigen OS (Windows (nur mit Adminkonto), Android, macOS, embedded Devices, OpenBSD, FreeBSD, NetBSD)
- seit 2020 im Linuxkernel integriert
- ähnlich wie OpenVPN ist die Konfiguration über alle Plattformen hinweg identisch
- Roamingfähig, keine Verbindungsabbrüche bei Netzwechsel
- robust gegen kurze Verbindungsausfälle
- Port: UDP, Default: UDP 51820, frei änderbar - muß manuell in Firewalls frei gegeben werden (MTU: 1420), problemlos über NAT
- jede Verbindung hat eigene Public- und Private Key (einfache Textstrings) und funktioniert ähnlich wie SSH-Keys
  Die Schlüsselpaare lassen sich meist komfortabel im Gerät erstellen, es geht aber auch offline (siehe unten).
  Damit lassen sich VPN-Verbindungen für Endgeräte konfigurieren, ohne diese in den Händen zu halten.
- jeder Public Key wird mit Liste erlaubter Netze verknüpft (Wildcard 0.0.0.0/0 bedeutet: alle Netze gehen durch Tunnel)
  - In Versandrichtung verhält sich die Liste wie eine Routing Tabelle.
  - In Empfangsrichtung dient die Liste als Access Control List.
- Clients bekommen (meist) statische IPs, Firewallregeln pro Endgerät möglich (DHCP-Pool nur über Scripts)
- eine Verbindung lässt sich auf mehreren Endgeräte nutzen, diese bekommen aber die gleiche IP und gleiche AccessRights
- nur eine Seite muß eine feste IP und einen von außen erreichbaren UDP-Port besitzen
- Server erstellt (je nach Gerät) einfache Textdatei .conf oder QR-Code, der einfach am Client eingebunden wird
- Wireguard-Router: Dreytek Vigor, MikroTik, AVM Fritzbox, GL.iNet, pfSense, OpenSense, OpenWRT

Voraussetzungen
Der Wireguard-Router als Verbindungs-Empfänger muß eine feste IP haben und der verwendete UDP-Port muss beim Router ankommen.
Wird der WG-Router nach dem Standardgateway installiert, sind im Gateway-Router folgende Einstellungen erforderlich:

- Portforwarding: UDP-Port (51820) zum WG-Router (LAN-IP)
- Routing-Tabelle: WG-Transfernetz (192.168.200.0/24) zum WG-Router (LAN-IP), Maskierung aus
- Firewall: UDP-Zielport (51820) frei geben
- Firewall: Verbindungsquelle Wireguard-Router (LAN-IP) Internet erlauben
- Firewall: Verbindungsziel WG-Transfernetz (192.168.200.0/24) erlauben (Bintec: "vollständige IP v4-Filterung" deaktivieren ->siehe Bintec)

Wireguard-Client
- exportiert 'wireguard-export.zip' (die enthält für jede Verbindung eine '[verbindung].conf'
- importiert wahlweise einzelne 'verbindung.conf' oder 'wirguard-export.zip' mit allen Verbindungen
- vorhandene Verbindungen bleiben bei einem weiteren Import erhalten

Anleitung: https://administrator.de/tutorial/merkzettel-vpn-installation-mit-wireguard-660620.html
MikroTik: siehe MikroTik Wireguard-Artikel
GL.iNet LTE: siehe GL.iNet Wireguard mit OpenVPN

Public Key und Private Key können auch unabhängig vom jeweiligen Gerät konfiguriert werden,
um eine .conf Datei für ein beliebiges Endgerät zu erstellen.

```
umask 077
wg genkey | tee server_private.key | wg pubkey > server_public.key
wg genkey | tee peer1_private.key | wg pubkey > peer1_public.key
```

Weitere VPN Client (Peer) Schlüssel generiert man dann nur noch mit wg genkey | tee peer***2***_private.key | wg pubkey > peer***2***_public.key usw.

Beispiel der **.conf Datei** für den Peer1:

```
[Interface]
# Name = wireguard-server2.example
Address = 192.168.200.3/32
PrivateKey = OMjSCv6e/iXECZwq0ZVL5Ywf/KzZvdsGpYKv1512345=
# DNS = 172.16.7.254

[Peer]
# Name = Client-peer1
PublicKey = cA+mynt84tVH1gPaUN66E8K0nfzvpsQMohrEbz54321=
Endpoint = router.myfritz.de:51820
AllowedIPs = 192.168.200.0/24, 192.168.188.0/24
# PersistentkeepAlive = 25
```

- Die importierte Verbindung heisst wie die .conf-Datei.

- #Name - Comment-Feld, nur informativ
- Address = Client: Client Adresse (192.168.200.3/32), Server: ganzes Subnetz (192.168.200.0/24), auch mehrere Subnets möglich
- PrivateKey = der jeweils eigene private.key, im Endgerät wird dann der Public-Key angezeigt
- ListenPort = UDP Port, auf den der Server hört
- Table = ggf. Routing Table
- PublicKey = gegenüberliegender public.key
- AllowedIPs = die Adressen die der Wireguard Server in den Tunnel routet. Wireguard nennt dies "Cryptokey Routing" was bewirkt das der Wireguard Server und Client das Routing für die jeweils remoten IP Netze automatisch in die Routing Tabelle übernimmt.
  AllowedIPs = 0.0.0.0/0 bewirkt, dass der gesamte Traffic durch den Tunnel geht.
- Endpoint = [öffentl.Serveradresse]:Port (nur beim Client, Port nicht vergessen!)
- PersistentkeepAlive = hält die Verbiundung offen, nur bei Clients hinter NAT

Site-to-Site VPN-Verbindung

- beide Seiten routen im Wireguard in das jeweils gegenüberliegende LAN mit einem Eintrag "AllowedIPs"
- ist der Wireguard-Router nicht das Standardgateway, muß das Standardgateway (oder der PC) ein Routing zum Zielnetz besitzen

Wireguard **Online Config Generator:**

- Tool erstellt komplette .conf-Datei: https://www.wireguardconfig.com/
- Tool erstellt Schlüsselpaare: https://wg.orz.tools/

Eindeutige ID: #1380
Verfasser: Uwe Kernchen
Letzte Änderung: 2024-04-09 19:05