Vortrag: VPNs

Vortrag im Linux-Cafe am 4.10. und 6.12.2023

- Fragen bzw. Ergänzungen

Was bedeutet “VPN”?

Virtual **P**rivate **N**etwork

- (einige) gängige Ziele:
  - Zugriff auf Server/Geräte, die nicht im eigenen lokalen Netzwerk sind
  - Schutz vor Schnüfflern in offenem Netz
  - Verschleiern der eigenen IP-Adresse (gegenüber dritten Servern, nicht unbdedingt vor VPN-Betreiber)

⚠️ Disclaimer ⚠️

Jegliche Verantwortung bei der Umsetzung, auch Datenverlust oder Fehlkonfiguration etc. liegt bei dir selbst. Die hier gezeigten Informationen wurden nach bestem Wissen und Gewissen zusammengestellt. Die Aktualität, Richtigkeit und Vollständigkeit kann also abweichen!
Sei dir des Risikos also bewusst und installiere und nutze VPNs auf eigene Gefahr!

Risiken beim VPN-Betrieb

- Fehlkonfiguration, v.a. Security (gute Passwörter nutzen!)
- regelmäßig updaten und auf Sicherheitslücken achten!
- Ausweichmöglichkeit bereithalten, wenn das VPN kaputtgeht: Fallback-Zugriff (zweites VPN, Cloud-Provider-Terminal, SSH o.ä.)
- nicht-technische Aspekte: informieren, wovor bestimmte VPNs schützen und wovor nicht!
- **NICHT** hier im Vortrag behandelt: Browser-“VPN”-Plugins

VPNs

in diesem Vortrag (v.a. Open-Source-VPNs*):

1. ipsec
2. Nebula
3. WireGuard
4. OpenVPN
5. Closed-Source-VPNs
6. Dienste mit VPN-Funktionen ◦ SSH

**es gibt noch eine Reihe weiterer Open-Source-VPNs, aber das würde hier den*

ipsec

- Ursprung: RFC2401 (https://datatracker.ietf.org/doc/html/rfc2401) und andere -> Zusatz-Protokoll zur Absicherung von IP-Paketen
- Erscheinungsjahr: ~1995 (v1), 1998 (v2), 2005 (v3)
    - Transport (Verbindung von zwei Geräten)
- Modi: ◦ Tunnel (Verbindung von zwei IP-Netzen)
- Implementierungen:
    - Clients (Auswahl): **StrongSwan**/**OpenSwan**/**Libreswan**, häufig in OS bzw. Netzwerk-Manager eingebaut, einige proprietäre
    - Server (Auswahl): **Fritzboxen** vor FritzOS 7.39 (einzige eingebaute Möglichkeit) bzw. seitdem optional; **StrongSwan**/**LibreSwan**

Vor- und Nachteile

- 👍 sehr viele Funktionen und Einstellungen möglich (👎 : auch unsichere)
- 👍/👎 standardisiert in über 30 (!) RFCs
- 👎 sehr kompliziert zu konfigurieren (viele verschiedene Möglichkeiten für Teilkomponenten des Gesamtkonstrukts)
- 👎 sehr viele verschiedene Implementierungen; u.U. nicht alle Clients mit allen Servern kompatibel
- 👎 in Bezug auf Durchsatz weniger effizient als Wireguard

Nebula

- Ursprung: Slack Technologies, Inc.
- Ziel: sichere Kommunikation zwischen Gruppen von Hosts in mehreren Rechenzentren; einfaches “Overlay-Netzwerk”
- Erscheinungsjahr: 2019 (als Open Source; vorher Closed Source entwickelt)
- Quellcode: https://github.com/slackhq/nebula (https://github.com/slackhq/nebula)
- Funktionsweise: Peer-to-Peer-Verbindungen, die über einen “Lighthouse”-Server hergestellt werden
- HOWTO: https://nebula.defined.net/docs/guides/quick-start/ (https://nebula.defined.net/docs/guides/quick-start/)

Vor- und Nachteile (I)

- 👍 einfaches Grundkonzept, aber auch einige fortgeschrittene Einstellungen möglich
- 👍 eingebaute Firewall-Funktionen zwischen Geräten (im Lighthouse eingestellt)
- 👍 Authentifizierung zwischen Geräten über Zertifikate
- 👍 Client für alle gängigen Betriebssysteme (auch mobil) und Architekturen verfügbar
- 👍 kann optional DNS bereitstellen

Vor- und Nachteile (II)

- 👍 Nebula-Cert-Tool zum einfachen Verwalten von Zertifikaten
- 👎 Zertifikate müssen selbst verwaltet und verteilt werden, keine Passwort-Authentifizierung möglich; Revoken von Zertifikaten nicht mit Tool möglich, sondern nur per Nebula-Config
- 👎 “Lighthouse”-Server erforderlich, der von allen Geräten erreichbar sein muss
- 👎 Konfiguration mit Bordmitteln nur über Kommandozeile bzw. mit Config-Dateien möglich

Live-Demo

Das hier gezeigte Setup ist nur für eine schnelle Demonstration gedacht. Für ein ordentliches Setup sind einige Verbesserungen sehr sinnvoll, z.B. das Nebula-CLI-Tool in /usr/local/bin zu lagern und Nebula als Service (z.B. Systemd-Unit) und nicht manuell in einem Terminal auszuführen.

Auf den nächsten Slides finden sich einige Informationen und die Kommandos aus der Live-Demo.

Diese Live-Demo basiert teilweise auf dem offiziellen Guide für Nebula (https://nebula.defined.net/docs/guides/quick-start/). Falls etwas hier nicht klappt, im Zweifel dort nachschauen.

Vorbereitung für die Demo:

- es sind ein Lighthouse-Server und mindestens ein Client nötig
- der Lighthouse-Server und die Clients können in beliebigen Netzwerken/Netzbereichen sein, solang die Clients das Lighthouse auf Port 4242 erreichen können. Hier im Beispiel ist das echte Netzwerk im Bereich 192.168.122.0/24
- das virtuelle Nebula-Netzwerk liegt hier als Beispiel im Netzbereich 192.168.110.0/24, der auf keinem der Geräte anderweitig genutzt wird

Setup

Auf einer vertrauenswürdigen Linux-Maschine (möglichst ein **anderer** Rechner als das Lighthouse!):

1. `wget https://github.com/slackhq/nebula/releases/download/v1.7.2/nebula-linux-amd64.tar.gz` ○ *(Versionsnummer ggf. anpassen)*
2. `tar -xvf nebula-linux-amd64.tar.gz`
3. `./nebula-cert ca -name "Nebula-Demo-CA"`
4. `./nebula-cert sign -name "lighthouse" -ip "192.168.110.199/24"`
5. `./nebula-cert sign -name "pc" -ip "192.168.110.1/24" -groups "pc,ssh"`
6. `./nebula-cert sign -name "example-server" -ip "192.168.110.2/24" -groups "servers,http,ssh"`

3. `wget https://raw.githubusercontent.com/slackhq/nebula/master/examples/config.yml`
9. `cp config.yml config-lighthouse.yaml`
0. Config für Lighthouse (config-lighthouse.yaml) anpassen:

```
#...
lighthouse:
  am_lighthouse: true
#...
firewall:
  #...
  outbound_action: drop
  inbound_action: drop
  #...
  inbound:
    - port: 22
      proto: any
      host: any
```

1. Auf den Lighthouse-Server kopieren: `nebula` -Binary, config-lighthouse.yaml, ca.crt, lighthouse1.crt, lighthouse1.key

Auf dem Lighthouse-Server folgende Kommandos ausführen:

1. `sudo mkdir /etc/nebula`
2. `sudo mv config-lighthouse.yaml /etc/nebula/config.yaml`
3. `sudo mv ca.crt /etc/nebula/ca.crt`
4. `sudo mv lighthouse.crt /etc/nebula/host.crt`
5. `sudo mv lighthouse.key /etc/nebula/host.key`
6. `sudo ./nebula -config /etc/nebula/config.yaml`

1. `cp config.yml config-client.yaml`
2. Client-Config anpassen:

```
#...
static_host_map:
  # .110.1: Lighthouse-IP in Nebula; .122.178: öffentliche Lighthouse-I
  "192.168.110.1": ["192.168.122.178:4242"]
#...
lighthouse:
  hosts:
    - "192.168.110.1" # Lighthouse-IP in Nebula
#...
```

3. auf Client-Maschine(n) kopieren: `nebula` -Binary, config-client.yaml, ca.crt, client.crt client.key

Auf VPN-User-Maschinen:

1. `sudo mkdir /etc/nebula`
2. `sudo mv config-client.yaml /etc/nebula/config.yaml`
3. `sudo mv ca.crt /etc/nebula/ca.crt`
4. `sudo mv client.crt /etc/nebula/host.crt`
5. `sudo mv client.key /etc/nebula/host.key`
6. `sudo ./nebula -config /etc/nebula/config.yaml`

WireGuard

- Open-Source-Projekt
- Erscheinungsjahr: 2015 das erste Release (0.0.20161209) wurde am 09. Dezember 2016 veröffentlicht → https://git.zx2c4.com/WireGuard/tag/?h=0.0.20161209 (https://git.zx2c4.com/WireGuard/tag/?h=0.0.20161209)
- Quellcode: https://www.wireguard.com/repositories/ (https://www.wireguard.com/repositories/) // https://github.com/WireGuard (https://github.com/WireGuard)
- Whitepaper (https://www.wireguard.com/papers/wireguard.pdf)

Vor- und Nachteile

- 👍 schnelle und einfache Einrichtung auf Server bzw. Client
- 👍 Schlanker “Quell-Code” der nur aus ca. 4.000 Zeilen Code besteht. OpenVPN über 70.000 Zeilen Code, IPSec hat um die 400.000 Zeilen Code.
- 👍 Fokus auf nur wenig Kryptografietechniken dafür aber auf sehr modern und aktuelle Verfahren
- 👍 es werden alle gängigen Betriebssysteme unterstützt auch auf Routern wie die FritzBox

- 👍 direkt im Linux-Kernel integriert (seit Kernel Version 5.6)
- 👍 Bei wechsel von Verbindung WLAN → Mobilfunk → WLAN ist keine spürbare Unterbrechung vorhanden
- 👍 WireGuard etwa 50 % schneller als OpenVPN (Download wie Upload)
- 👍 Bessere Akkulaufzeit mit Handys / Tablets / Laptops

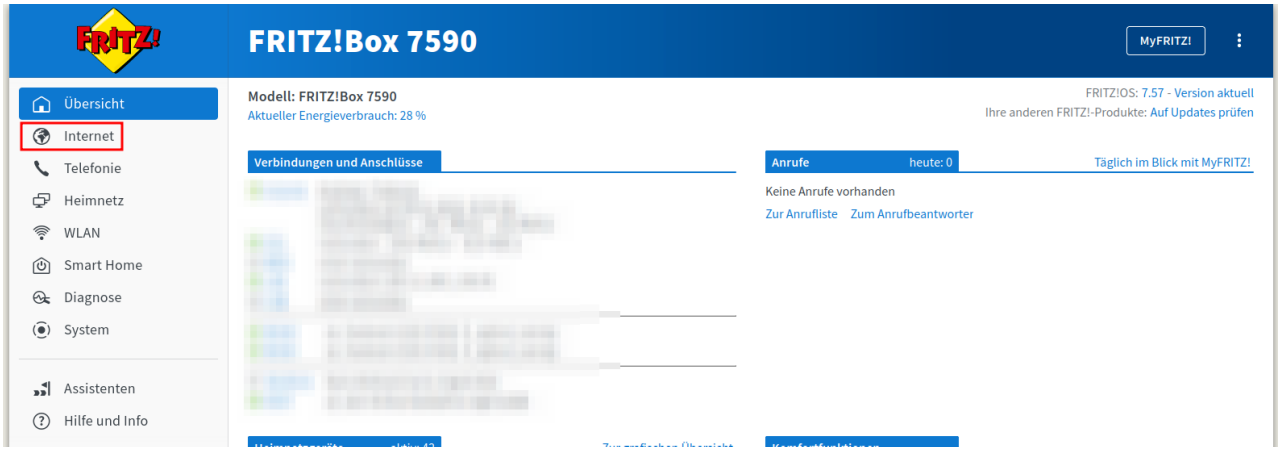
- 👎 aktuell wird nur das UDP Protokoll unterstützt, TCP nicht (Tunneln über den TCP Port 443 also nicht möglich)
  UDP (User Datagram Protocol)
  TCP (Transmission Control Protocol)
  UDP wird oft für Streaming und andere Echtzeitanwendungen bevorzugt. TCP ist etwas zuverlässiger, da es eine Überprüfung der Datenübertragung beinhaltet.
- 👎 kommerzielle VPN-Dienste setzen häufig proprietäre Anpassungen des Protokolls ein
- 👎 WireGuard ist aktuell noch in Entwicklung, kann aber dennoch ohne Bedenken eingesetzt werden
  Work in Progress (https://web.archive.org/web/20200201093649/https://www.wireguard.com/#work-in-progress) → Stand 31.03.2020

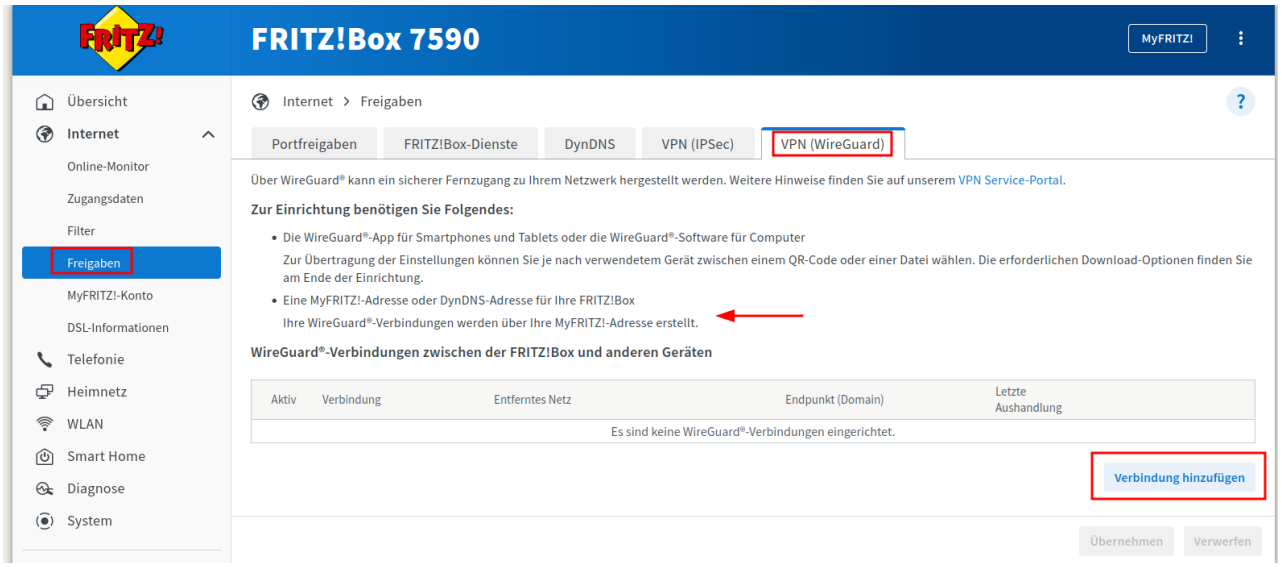
Verwendete Kryptografietechniken

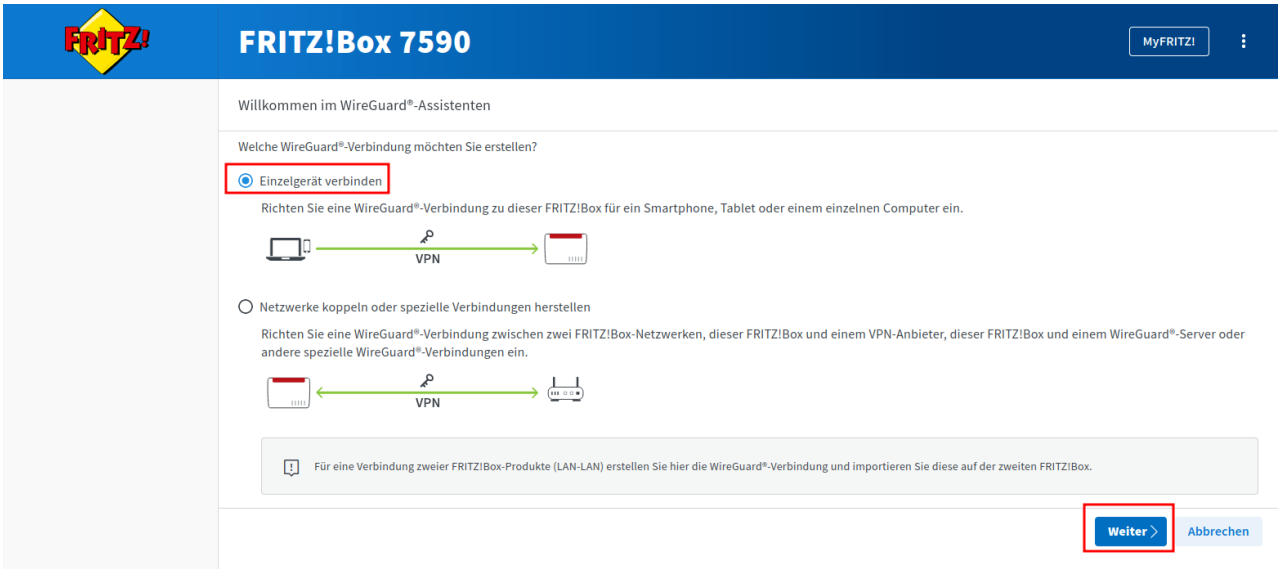
- Curve25519 mit Elliptic Curve Diffie-Hellman (ECDHE) für den Handshake (Schlüsselaustausch)
- BLAKE2s als universelle Hashfunktion (beispielsweise zum Generieren von HMAC-Codes oder für Key-Ableitungen mit HKDF)
- ChaCha20 und Poly1305 für die symmetrische Verschlüsselung und den Austausch der Daten

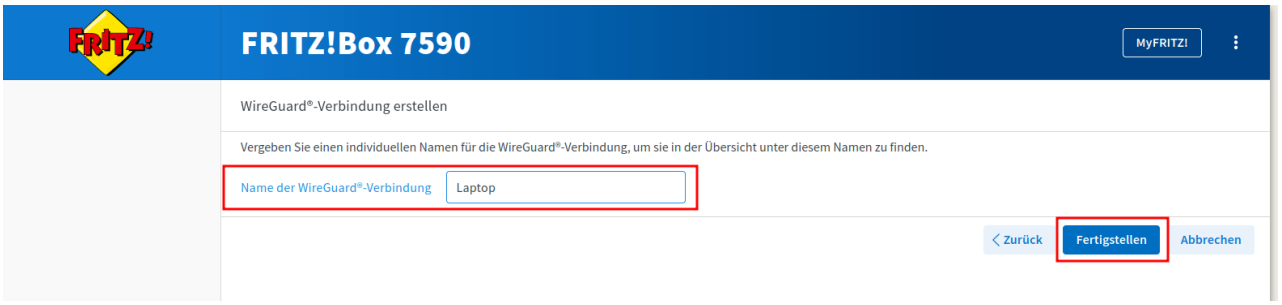
⚠ ❗ ⚠ Genauer auf die Kryptografietechniken gehen wir nicht ein, da dies zu Technisch wäre

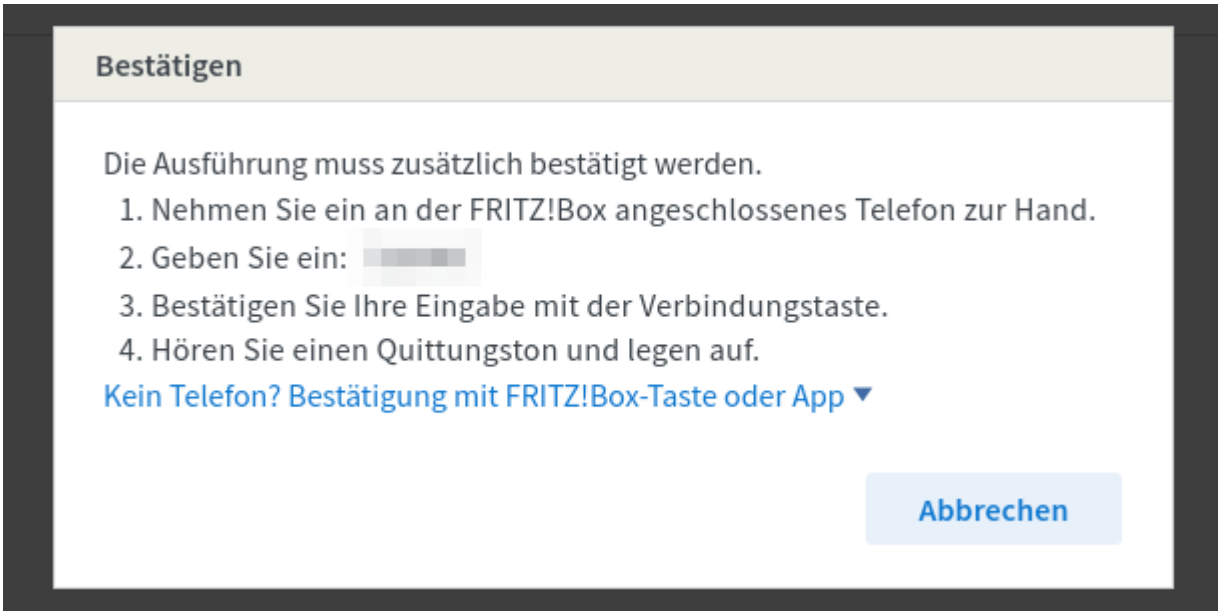
Demo Time WireGuard

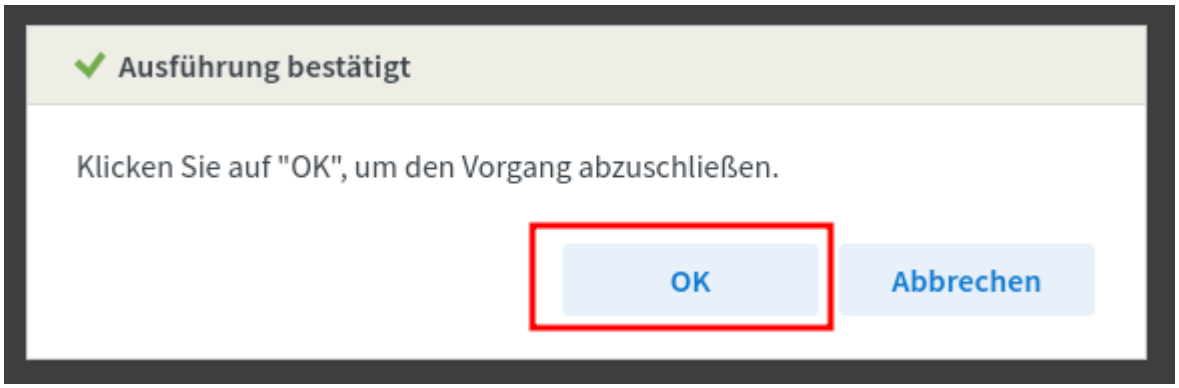


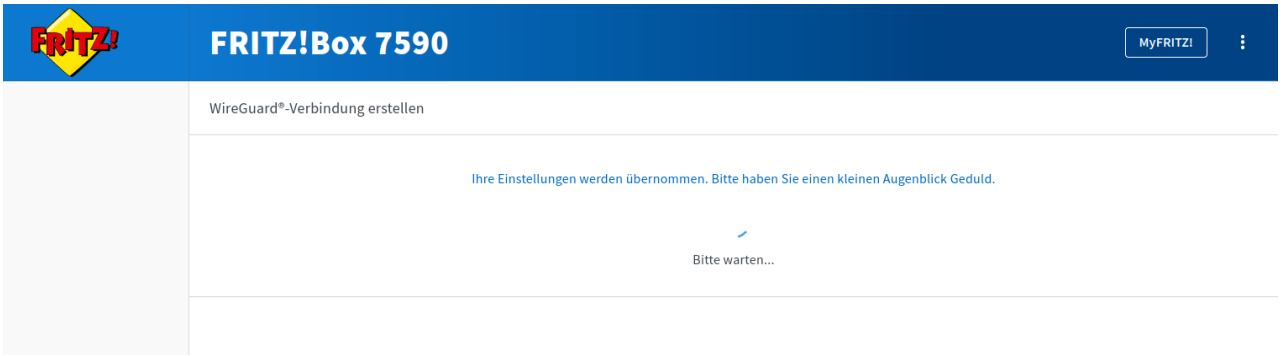

FRITZ!Box 7590
MyFRITZ!
Übersicht
Internet
Online-Monitor
Zugangsdaten
Filter
Freigaben
MyFRITZ!-Konto
DSL-Informationen
Telefonie
Heimnetz
WLAN
Smart Home
Diagnose
System
Internet > Freigaben
Portfreigaben
FRITZ!Box-Dienste
DynDNS
VPN (IPSec)
VPN (WireGuard)
Über WireGuard® kann ein sicherer Fernzugang zu Ihrem Netzwerk hergestellt werden. Weitere Hinweise finden Sie auf unserem VPN Service-Portal.
Zur Einrichtung benötigen Sie Folgendes:
Die WireGuard®-App für Smartphones und Tablets oder die WireGuard®-Software für Computer
Zur Übertragung der Einstellungen können Sie je nach verwendetem Gerät zwischen einem QR-Code oder einer Datei wählen. Die erforderlichen Download-Optionen finden Sie am Ende der Einrichtung.
Eine MyFRITZ!-Adresse oder DynDNS-Adresse für Ihre FRITZ!Box
Ihre WireGuard®-Verbindungen werden über Ihre MyFRITZ!-Adresse erstellt.
WireGuard®-Verbindungen zwischen der FRITZ!Box und anderen Geräten
Aktiv
Verbindung
Entferntes Netz
Endpunkt (Domain)
Letzte Aushandlung
Es sind keine WireGuard®-Verbindungen eingerichtet.
Verbindung hinzufügen
Übernehmen
Verwerfen


FRITZ!Box 7590
MyFRITZ!
Willkommen im WireGuard®-Assistenten
Welche WireGuard®-Verbindung möchten Sie erstellen?
Einzelgerät verbinden
Richten Sie eine WireGuard®-Verbindung zu dieser FRITZ!Box für ein Smartphone, Tablet oder einem einzelnen Computer ein.
VPN
Netzwerke koppeln oder spezielle Verbindungen herstellen
Richten Sie eine WireGuard®-Verbindung zwischen zwei FRITZ!Box-Netzwerken, dieser FRITZ!Box und einem VPN-Anbieter, dieser FRITZ!Box und einem WireGuard®-Server oder andere spezielle WireGuard®-Verbindungen ein.
VPN
Für eine Verbindung zweier FRITZ!Box-Produkte (LAN-LAN) erstellen Sie hier die WireGuard®-Verbindung und importieren Sie diese auf der zweiten FRITZ!Box.
Weiter
Abbrechen


FRITZ!Box 7590
MyFRITZ!
WireGuard®-Verbindung erstellen
Vergeben Sie einen individuellen Namen für die WireGuard®-Verbindung, um sie in der Übersicht unter diesem Namen zu finden.
Name der WireGuard®-Verbindung
Laptop
Zurück
Fertigstellen
Abbrechen


Bestätigen
Die Ausführung muss zusätzlich bestätigt werden.
1. Nehmen Sie ein an der FRITZ!Box angeschlossenes Telefon zur Hand.
2. Geben Sie ein:
3. Bestätigen Sie Ihre Eingabe mit der Verbindungstaste.
4. Hören Sie einen Quittungston und legen auf.
Kein Telefon? Bestätigung mit FRITZ!Box-Taste oder App
Abbrechen


Ausführung bestätigt
Klicken Sie auf "OK", um den Vorgang abzuschließen.
OK
Abbrechen


FRITZ!Box 7590
MyFRITZ!
WireGuard®-Verbindung erstellen
Ihre Einstellungen werden übernommen. Bitte haben Sie einen kleinen Augenblick Geduld.
Bitte warten...

FRITZ!Box 7590

MyFRITZ!

VPN (WireGuard®)

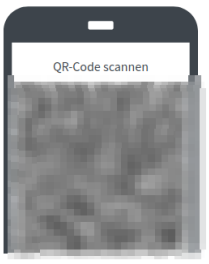
Die WireGuard®-Verbindung wurde erfolgreich erstellt.

Einstellungen auf Ihr Gerät übertragen

Sie haben nun die Möglichkeit, die Einstellungen über eine Datei auf Ihren Desktop oder Laptop zu übertragen oder über einen QR-Code an Ihr Smartphone / Tablet weiterzugeben. Nach dem Übertragen der Einstellungen auf Ihr Gerät können Sie den Fernzugriff nutzen.

Im Folgenden beschreiben wir Ihnen in kurzen Schritten, was zur Übertragung zu tun ist.

Smartphone oder Tablet



So funktioniert es:

Für die Verwendung mit einem Smartphone oder Tablet benötigen Sie die WireGuard®-App und den oben angezeigten QR-Code.

1. Installieren Sie die WireGuard®-App über den jeweiligen App-Store auf dem bevorzugten Gerät.

Mehr Informationen in Hilfe anzeigen

2. Starten Sie WireGuard®, tippen Sie auf das Plus „+“ und anschließend auf „aus

Desktop oder Laptop

Einstellungen herunterladen

So funktioniert es:

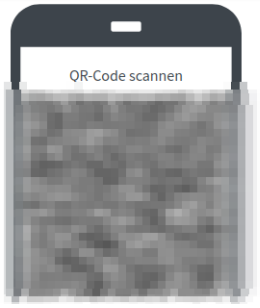
Für die Verwendung mit einem Desktop oder Laptop benötigen Sie die WireGuard®-Software und die oben bereitgestellten Einstellungen.

1. Klicken Sie auf „Einstellungen herunterladen“, um die Einstellungen für Ihre WireGuard®-Verbindung nutzen zu können.
2. Installieren Sie die WireGuard®-Software für das Betriebssystem Ihres Desktops oder Laptops.

Software auf www.wireguard.com finden

3. Starten Sie WireGuard® und klicken Sie auf „Tunnel aus Datei importieren“.

VPN (WireGuard®)



So funktioniert es:

Für die Verwendung mit einem Smartphone oder Tablet benötigen Sie die WireGuard®-App und den oben angezeigten QR-Code.

1. Installieren Sie die WireGuard®-App über den jeweiligen App-Store auf dem bevorzugten Gerät.

Mehr Informationen in Hilfe anzeigen

2. Starten Sie WireGuard®, tippen Sie auf das Plus „+“ und anschließend auf „aus QR-Code erstellen“.
3. Scannen Sie mit der Kamera Ihres Gerätes den oben angezeigten QR-Code ein.
4. Folgen Sie den weiteren Anweisungen in der WireGuard®-App.

Einstellungen herunterladen

So funktioniert es:

Für die Verwendung mit einem Desktop oder Laptop benötigen Sie die WireGuard®-Software und die oben bereitgestellten Einstellungen.

1. Klicken Sie auf „Einstellungen herunterladen“, um die Einstellungen für Ihre WireGuard®-Verbindung nutzen zu können.
2. Installieren Sie die WireGuard®-Software für das Betriebssystem Ihres Desktops oder Laptops.

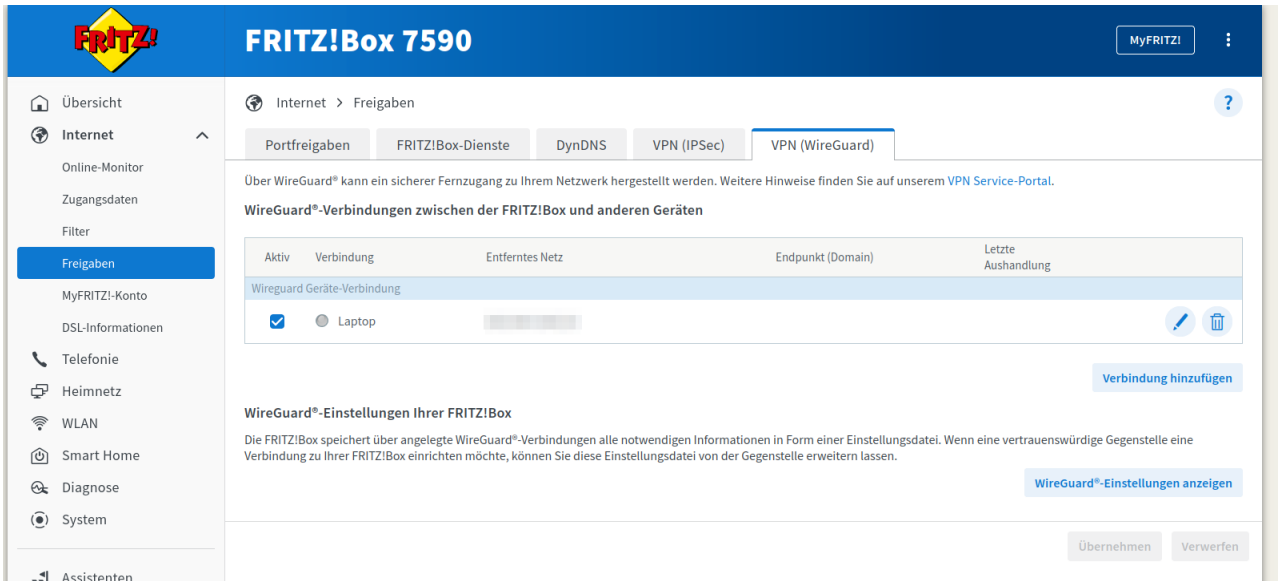
Software auf www.wireguard.com finden

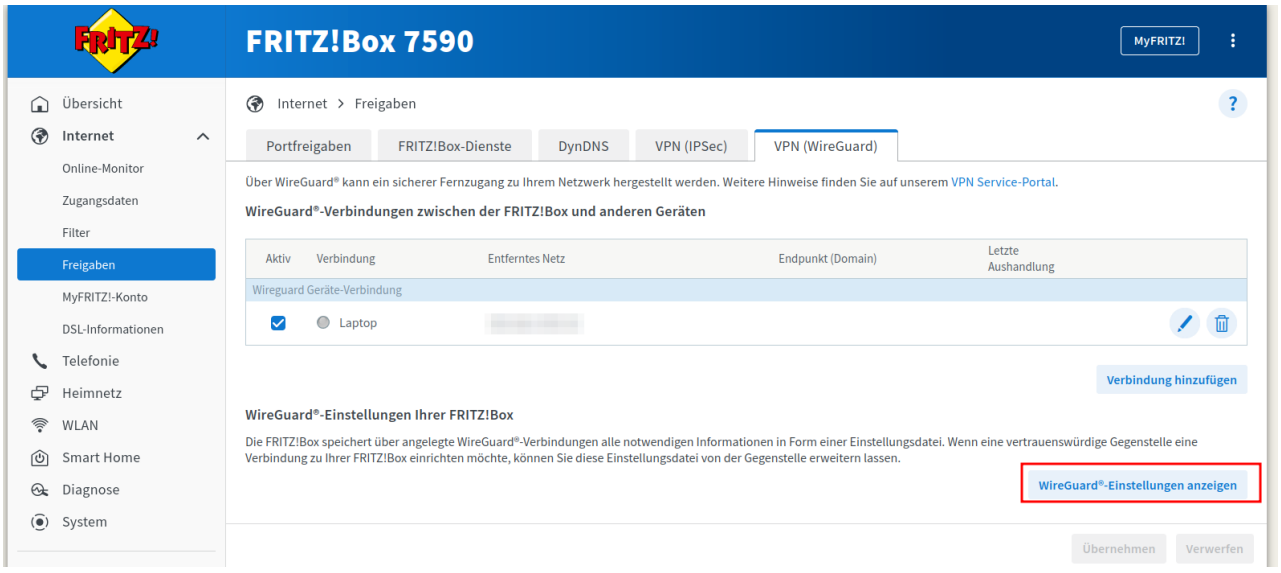
3. Starten Sie WireGuard® und klicken Sie auf „Tunnel aus Datei importieren“.
4. Importieren Sie die oben angezeigte Datei und folgen Sie den weiteren Anweisungen der Software.

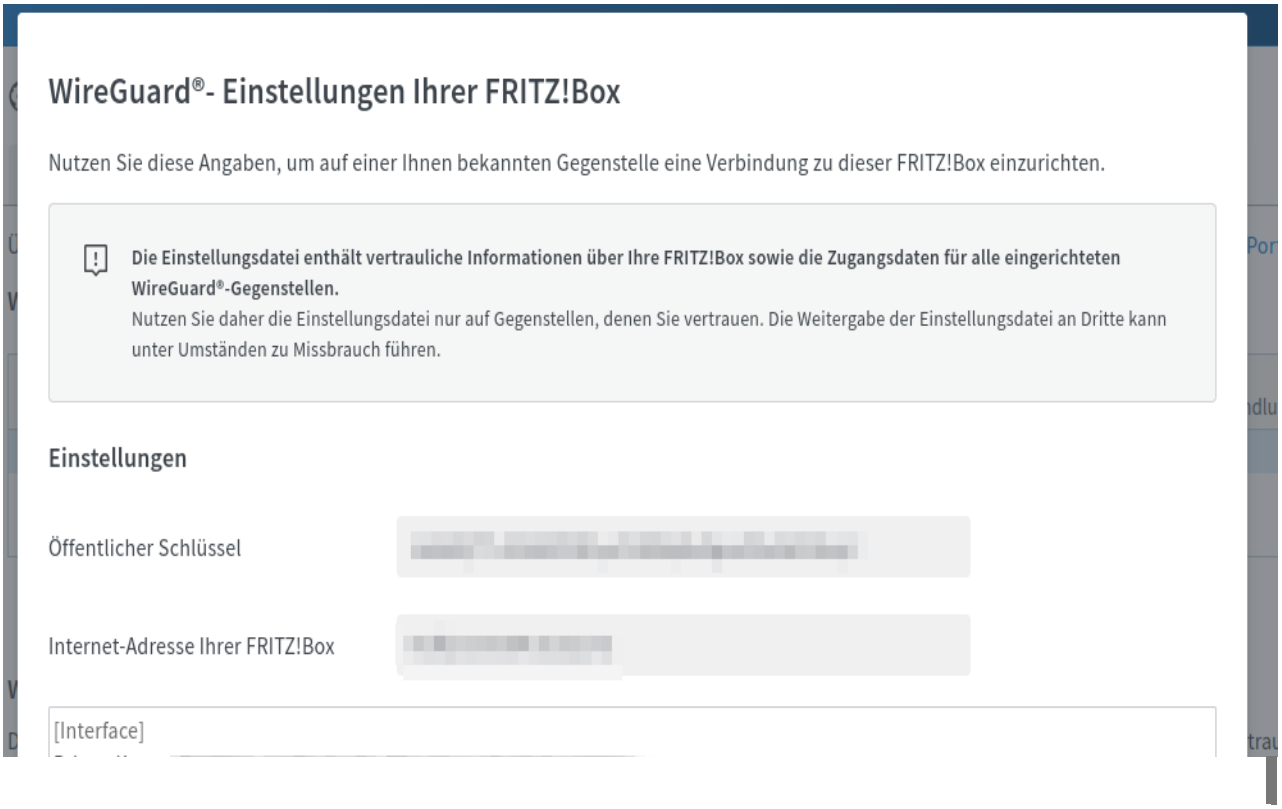
Hinweis:

Beachten Sie, dass keine weiteren Geräte diese Verbindung zeitgleich nutzen können.

Wenn Sie die Einstellungen auf Ihrem Gerät übernommen haben, können Sie diese Ansicht schließen. Die neue Verbindung wird Ihnen dann auf der Übersicht der WireGuard®-Verbindungen


FRITZ!Box 7590
MyFRITZ!
Übersicht
Internet
Online-Monitor
Zugangsdaten
Filter
Freigaben
MyFRITZ!-Konto
DSL-Informationen
Telefonie
Heimnetz
WLAN
Smart Home
Diagnose
System
Assistenten
Internet > Freigaben
Portfreigaben
FRITZ!Box-Dienste
DynDNS
VPN (IPSec)
VPN (WireGuard)
Über WireGuard® kann ein sicherer Fernzugang zu Ihrem Netzwerk hergestellt werden. Weitere Hinweise finden Sie auf unserem VPN Service-Portal.
WireGuard®-Verbindungen zwischen der FRITZ!Box und anderen Geräten
Aktiv
Verbindung
Entferntes Netz
Endpunkt (Domain)
Letzte Aushandlung
Wireguard Geräte-Verbindung
Laptop
Verbindung hinzufügen
WireGuard®-Einstellungen Ihrer FRITZ!Box
Die FRITZ!Box speichert über angelegte WireGuard®-Verbindungen alle notwendigen Informationen in Form einer Einstellungsdatei. Wenn eine vertrauenswürdige Gegenstelle eine Verbindung zu Ihrer FRITZ!Box einrichten möchte, können Sie diese Einstellungsdatei von der Gegenstelle erweitern lassen.
WireGuard®-Einstellungen anzeigen
Übernehmen
Verwerfen


FRITZ!Box 7590
MyFRITZ!
Übersicht
Internet
Online-Monitor
Zugangsdaten
Filter
Freigaben
MyFRITZ!-Konto
DSL-Informationen
Telefonie
Heimnetz
WLAN
Smart Home
Diagnose
System
Internet > Freigaben
Portfreigaben
FRITZ!Box-Dienste
DynDNS
VPN (IPSec)
VPN (WireGuard)
Über WireGuard® kann ein sicherer Fernzugang zu Ihrem Netzwerk hergestellt werden. Weitere Hinweise finden Sie auf unserem VPN Service-Portal.
WireGuard®-Verbindungen zwischen der FRITZ!Box und anderen Geräten
Aktiv
Verbindung
Entferntes Netz
Endpunkt (Domain)
Letzte Aushandlung
Wireguard Geräte-Verbindung
Laptop
Verbindung hinzufügen
WireGuard®-Einstellungen Ihrer FRITZ!Box
Die FRITZ!Box speichert über angelegte WireGuard®-Verbindungen alle notwendigen Informationen in Form einer Einstellungsdatei. Wenn eine vertrauenswürdige Gegenstelle eine Verbindung zu Ihrer FRITZ!Box einrichten möchte, können Sie diese Einstellungsdatei von der Gegenstelle erweitern lassen.
WireGuard®-Einstellungen anzeigen
Übernehmen
Verwerfen



Demo mit einer FritzBox

- Verweis auf YouTube Videos für die Umsetzung
- mit Linux
  FritzBox: WireGuard auf Linux einrichten (https://piped.adminforge.de/watch?v=hCeVj8SoAIs)
- mit Windows
  FritzBox: WireGuard auf Windows und Smartphone einrichten (https://piped.adminforge.de/watch?v=yC4S0Sk_6wA)

WireGuard App für Handys

- Android: https://play.google.com/store/apps/details?id=com.wireguard.android (https://play.google.com/store/apps/details?id=com.wireguard.android)
- Apple https://apps.apple.com/de/app/wireguard/id1441195209 (https://apps.apple.com/de/app/wireguard/id1441195209)

OpenVPN

- Open-Source-Projekt
- Erscheinungsjahr: 2002 → https://openvpn.net/community-resources/changelog-for-openvpn-2-0 (https://openvpn.net/community-resources/changelog-for-openvpn-2-0)
- Quellcode: https://github.com/OpenVPN/openvpn (https://github.com/OpenVPN/openvpn)

Vor- und Nachteile

- 👍 sehr gute Möglichkeiten die Konfiguration nach eigenen wünschen anzupassen
- 👍 Kann im Gegensatz zu WireGuard Firewalls umgehen (Tunneln über den TCP Port 443 also möglich)
- 👍 es werden alle gängigen Betriebssysteme unterstützt
- 👍 TCP wie UDP Protokoll nutzung möglich
  UDP (User Datagram Protocol)
  TCP (Transmission Control Protocol)
  UDP wird oft für Streaming und andere Echtzeitanwendungen bevorzugt. TCP ist etwas zuverlässiger, da es eine Überprüfung der Datenübertragung beinhaltet.

- 👎 schwerer Einzurichten / zu Konfigurieren
- 👎 keine Umsetzung per FritzBox möglich, somit wird immer eine Separate Hardware wie z.B. ein RaspberryPi benötigt.
- 👎 Download / Upload ist etwa 50% langsamer
- 👎 keine direkte Linux-Kernel Integration
- 👎 schlechtere Akkulaufzeit mit Handys / Tablets / Laptops

Verwendete Kryptografietechniken

- AES, Blowfish, Camellia, DES, Poly1305 …

⚠ ❗ ⚠ Genauer auf die Kryptografietechniken gehen wir nicht ein, da dies zu Technisch wäre und je nach individueller Konfiguration variiert.

Demo Time OpenVPN

...

- Installations-Script (https://github.com/angristan/openvpn-install)

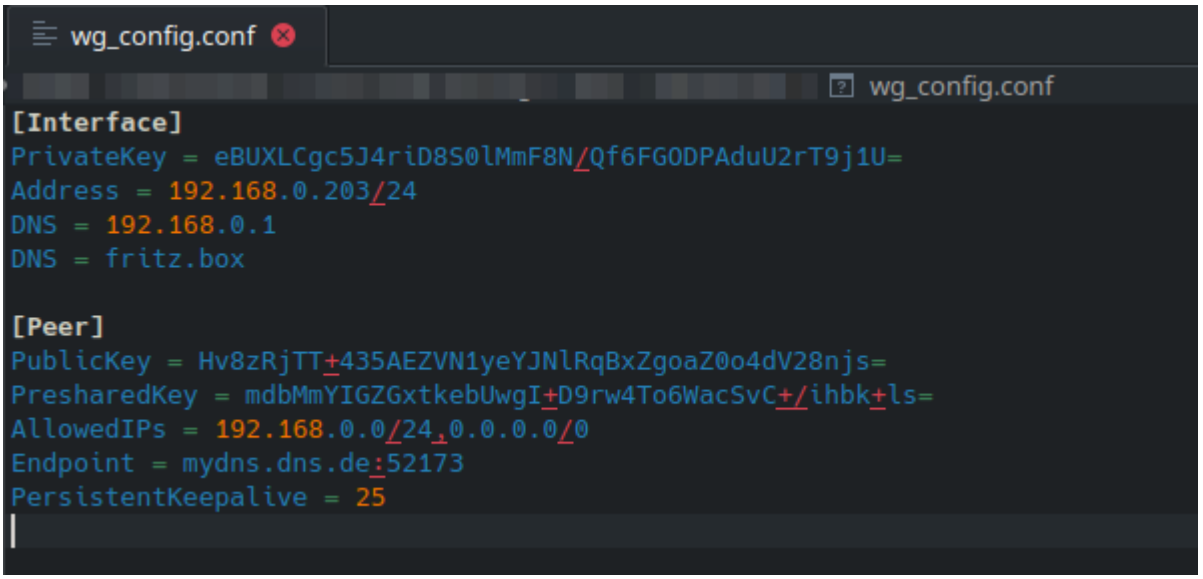
```
curl -O https://raw.githubusercontent.com/angristan/openvpn-install/
chmod +x openvpn-install.sh

su -
./openvpn-install.sh
```

OpenVPN App für Handys

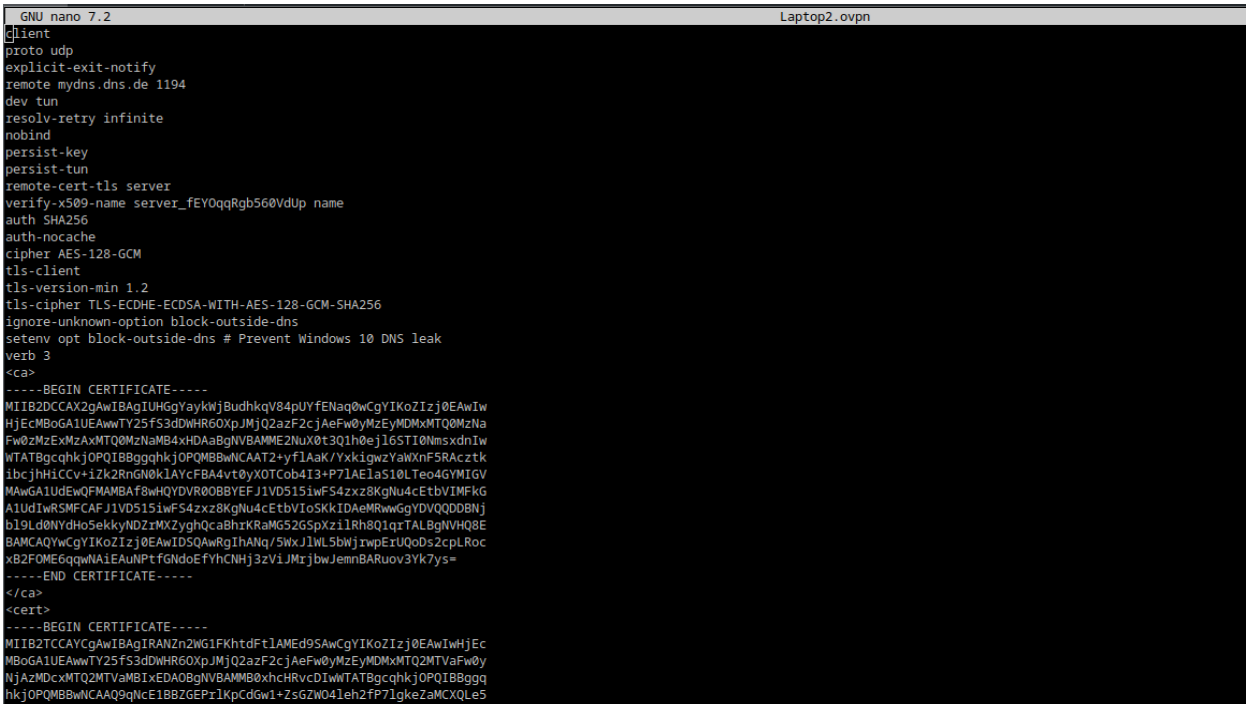
- Android: https://f-droid.org/de/packages/de.blinkt.openvpn/ (https://f-droid.org/de/packages/de.blinkt.openvpn/)
- Apple: https://apps.apple.com/us/app/openvpn-connect-openvpn-app/id590379981 (https://apps.apple.com/us/app/openvpn-connect-openvpn-app/id590379981)

vergleich WireGuard Config mit OpenVPN Config

- WireGuard:



- OpenVPN:



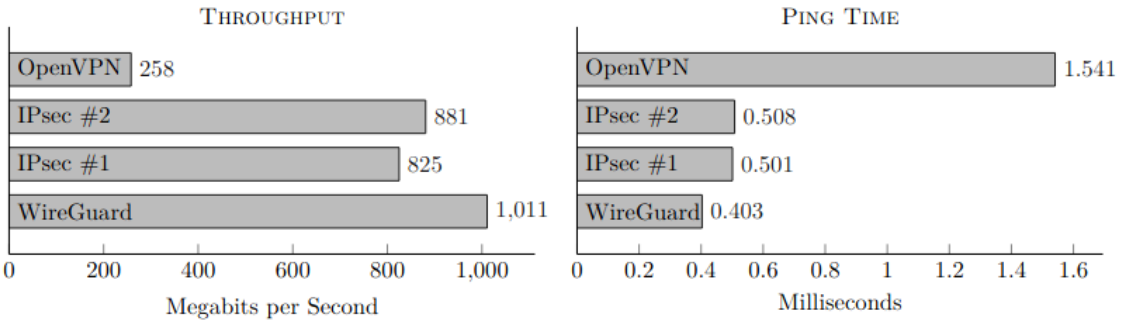
Server.config

Server.config

```
port 1194
proto udp
dev tun
user nobody
group nogroup
persist-key
persist-tun
keepalive 10 120
topology subnet
server 10.8.0.0 255.255.255.0
ifconfig-pool-persist ipp.txt
push "dhcp-option DNS 192.168.0.1"
push "redirect-gateway def1 bypass-dhcp"
dh none
ecdh-curve prime256v1
tls-crypt tls-crypt.key
crl-verify crl.pem
ca ca.crt
```

WireGuard vs. OpenVPN / IPsec

Protocol	Configuration
WireGuard	256-bit ChaCha20, 128-bit Poly1305
IPsec #1	256-bit ChaCha20, 128-bit Poly1305
IPsec #2	256-bit AES, 128-bit GCM
OpenVPN	256-bit AES, HMAC-SHA2-256, UDP mode



- Quelle - Seite 18 (https://www.wireguard.com/papers/wireguard.pdf)

- Webseiten im Bezug auf WireGuard oder OpenVPN

Video #1 (https://piped.adminforge.de/watch?v=hPQxGcOTkDs)

Video #2 (https://piped.adminforge.de/watch?v=ug-Dm3_tsLQ)

Webseite #1 (https://www.privacyaffairs.com/de/wireguard-vs-openvpn/)

Webseite #2 (https://www.bleib-virenfrei.de/it-sicherheit/vpn/artikel/die-wichtigsten-vpn-protokolle/)

Webseite #3 (https://schroederdennis.de/allgemein/wireguard-vs-openvpn-vs-ipsec-ikev2-der-vpn-vergleich/)

Webseite #4 (https://vpnoverview.com/de/vpn-informationen/vpn-protokolle-im-vergleich/)

Andere Tools mit VPN-Funktionen

- **Meshtastic (https://meshtastic.org/)**: dezentralisiertes Mesh-Netzwerk für kleine, stromsparende Geräte, Ziel: von Stromzufuhr unabhängig sein
- **tinc (https://tinc-vpn.org/)**: Verbindung/Tunnel zwischen einzelnen Geräten, die bereits eine Verbindung zueinander haben
- ***Tunneling-Dienste***: können Ports/Dienste auf einem lokalen Rechner vom Internet aus erreichbar machen. Eine Sammlung einiger solcher Dienste findet sich hier (https://github.com/anderspitman/awesome-tunneling)
- **SSH**: hat ein paar Port-Forwarding-Features

SSH

Möglichkeit: SSH für Port-Weiterleitungen nutzen

- Möglichkeiten:
  - auf der Kommandozeile
  - in der SSH-Config
  - Socks-Proxy
  - *Bonus:* Port-Forwarding in laufender SSH-Session anpassen

Quellen: `man ssh` , `man ssh_config` , einige Vorträge (siehe Quellen)

Local-Forward

Anfragen an Client-Port werden über SSH zu Port auf Serverseite weitergeleitet

- CLI: `ssh -L [bind_address:]port:host:hostport my-remote-host.example.com`
  - bind_address auf 127.0.0.1 setzen, wenn nur lokal benötigt, ansonsten ist der Port auf allen Interfaces verfügbar!
  - statt Host auch andere Adressen möglich
  - auch möglich: Sockets durchreichen

Local-Forward in SSH-Config (`~/.ssh/config`):

```
Host my-remote-host.example.com
  LocalForward localhost:8080 127.0.0.1:8080
  LocalForward localhost:8081 host2.example.com:8080

```

Remote-Forward

Anfragen an Server-Port werden über SSH zu Port auf Client-Seite weitergeleitet

- CLI: `ssh -R [bind_address:]port:host:hostport my-remote-host.example.com`
- Remote-Forward in SSH-Config (`~/.ssh/config`):

```
Host my-remote-host.example.com
  RemoteForward localhost:8080 127.0.0.1:8080
  RemoteForward localhost:8081 host2.example.com:8080
```

SOCKS-Proxy/“Dynamic Forward” mit SSH

- ein lokaler Port wird von SSH geöffnet und SSH proxiet damit über das SOCKS4/5-Protkokoll beliebige Anfragen
  - SOCKS-Proxy wird von gängigen Browsern unterstützt
- CLI: `ssh -D 1234 my-remote-host.example.com`
  - *umgekehrte Richtung*: `ssh -R localhost:1234 my-remote-host.example.com`
  - ⚠️ ❗ ⚠️ Remote-Dynamic-Forward kann von allen genutzt werden, die den Port erreichen, und erlaubt Zugriff auf beliebige lokale Systeme! Unbedingt `PermitRemoteOpen` in `man ssh_config` nutzen!

SOCKS-Proxy/DynamicForward in SSH-Config (`~/.ssh/config`):

```
Host my-remote-host.example.com
  DynamicForward localhost:8080
```

Bonus: Escape Characters

Quelle: `ESCAPE CHARACTERS` in `man ssh`

Einfach in einer laufenden SSH-Session Enter und danach eine dieser Tastenfolgen drücken:

- `~?` : verfügbare Escape-Sequenzen auflisten
- `~#` : weitergeleitete Verbindungen auflisten

- ***Wenn*** *`PermitLocalCommand` in der SSH-Config auf Client-Seite aktiviert ist:* `~C` und danach z.B. `-L [bind_address:]port:host:hostport` : fügt die entsprechende Port-Weiterleitung der laufenden SSH-Session hinzu
  - funktioniert auch mit `-R ...` und `-D ...`
  - kann auch Port-Weiterleitungen entfernen mit `-K...` , also z.B. `-KL127.0.0.1:1234`

closed-source-VPNs

- **Tailscale** (https://tailscale.com/): Wireguard-basiert; Funktionsweise ähnlich zu Nebula mit noch ein paar Zusatz-Features (z.B. RBAC, SSO), aber nur der Client ist Open Source, nicht der Koordinations-Server
- **Zerotier** (https://www.zerotier.com/): ähnlich wie Tailscale und Nebula, aber mit eigenem Protokoll statt Wireguard; ist komplett Open Source, aber unter BSL 1.1, ein Controller darf nur für nicht-kommerzielle Zwecke selbst gehostet werden

Quellen zum Weiterlesen

- Nebula: https://medium.com/several-people-are-coding/introducing-nebula-the-open-source-global-overlay-network-from-slack-884110a5579 (https://medium.com/several-people-are-coding/introducing-nebula-the-open-source-global-overlay-network-from-slack-884110a5579)
- Noise Protocol Framework: https://noiseprotocol.org (https://noiseprotocol.org)
- Vorträge zu SSH: Besser leben mit SSH (https://media.ccc.de/v/gpn20-8-besser-leben-mit-ssh), Noch besser leben mit SSH (https://media.ccc.de/v/gpn21-28-noch-besser-leben-mit-ssh), SSH Configuration, Intermediate Level (https://media.ccc.de/v/mch2022-170-ssh-configuration-intermediate-level)

Feedback

- Fragen bzw. Ergänzungen

Klick (https://doc.adminforge.de/voso-gCZSQe1gMiBF8T3zw#)