#Sonstiges | Wireguard Reverse VPN Tunnel erstellen

Inhaltsverzeichnis

- 1 Was wollen wir?
- 2 Und wie genau geht das?
  - 2.1 VPS Server & Local Work Server:
  - 2.2 VPS Server:
  - 2.3 Local Work Server:
  - 2.4 VPS Server
  - 2.5 VPS Server & Local Work Server:
  - 2.6 VPS Server
  - 2.7 Zusatz zu Firewall:
  - 2.8 Weitere Verbindungen:
  - 2.9 Probleme & Lösungen:

1 Was wollen wir?

Ersatzweise, zu dem bereits von mir erstellten WIKI #SONSTIGES | EXTERNER ZUGRIFF ÜBER IPV6 AUFS NETZWERK möchten wir eine Möglichkeit schaffen einen Tunnel über Wireguard zu realisieren.

Warum wollen wir das?

Der Tunnel mit dem Tool 6Tunnel erlaubt nur Verbindungen über TCP, ausserdem gibt es setups, wo der Benutzer keine feste IPv6 zu Hause bekommt. Dieses Szenario bietet also eine Weiterleitung von einem virtuellem Server im Internet, mit eigener IP4, zu seinem HomeLab Server zu Hause. Es können individuell nur die Ports frei gegeben werden, welche getunnelt werden sollen. Es wird aber keine Portfreigabe in der Firewall des eigenen Netzwerkes benötigt.

2 Und wie genau geht das?

Es gibt zahlreiche Möglichkeiten sich über Reverse VPN einen Zugang zu sich zu erstellen.

Diese Anleitung beschränkt sich auf dieses Szenario:

- Eine VPS Server ist im Internet vorhanden.

- Ein vorhandener DNS Provider leitet die eigene DNS zu der IP des VPS Server weiter.

- Ein Lokaler Linux Server mit Reverse Proxy Server ist für die interne Weiterleitung der Ports 80 & 443 vorhanden.

- Beide Server laufen mit Ubuntu (debian), für andere Distributionen sind evtl. andere Befehle notwendig

- Dieses Setup baut nur EINEN Tunnel auf!

Es wird also ein dauerhafter Tunnel der IP4 vom "VPS Server" zu seinem lokalem "Work Server" mit zb NGINX aufgebaut, der Reverse Proxy wandelt die ankommenden Anfragen um, erstellt evtl. ein lets encrypt Zertifikat und gibt die Anfragen an interne Server weiter.

Um sich einen virtuellen Server im Internet mit eigener IP einzurichten, schaut bitte auch im oben erwähntem anderen Wiki, dort wird der Zugang über einen IONOS Server beschrieben oder auch in meiner Anleitung zu einem kostenlosen Oracle Cloud TIER nach.

Dieser virtuelle Server ist natürlich Voraussetzung für die Einrichtung. Weiterhin benötigt Ihr einen lokalen Client, zb. einen Proxmox Server mit einem Linux, einem anderen Linux Server wie einem Raspberry Pi oder sonstigen Server.

Zur Erklärung der Angaben hier, schreibe ich immer dabei ob die Eingaben auf dem "VPS Server", dem "Local Work Server" oder beiden "VPS Server & Local Work Server" durchgeführt werden müssen!

2.1 VPS Server & Local Work Server:

Code

```
sudo apt update -y && sudo apt upgrade -y && sudo shutdown -r now
sudo apt install software-properties-common unzip -y
sudo apt install netfilter-persistent -y
sudo apt install ufw -y
sudo apt install wireguard -y
```

Erklärung: Der server wird upgedated und neu gestartet. Dann werden einige Software Pakete installiert.

Code

```
(umask 077 && printf "[Interface]\nPrivateKey=" | sudo tee /etc/wireguard/wg0.conf > /dev/null)
wg genkey | sudo tee -a /etc/wireguard/wg0.conf | wg pubkey | sudo tee /etc/wireguard/public
```

Erklärung: Die Befehle speichern den Private Key in die Datei /etc/wireguard/wg0.conf und gibt den Public Key aus -> Dieser solltet Ihr Euch zu beiden System notieren.

2.2 VPS Server:

Code

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

Folgende Daten eingeben und Parameter anpassen, Keys und einen geöffneten Port 55107

Code

```
[Interface]
PrivateKey = <Private key des VPS Server hier eingeben>
ListenPort = 55107
Address = 192.168.4.1/32
[Peer]
PublicKey = <public key vom Local Work Server hier>
AllowedIPs = 192.168.4.2/32
```

2.3 Local Work Server:

Code

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

Folgende Daten eingeben und Parameter anpassen, Keys und IP des VPS Servers, wir nutzen den Port 55107.

Code

```
[Interface]
PrivateKey = <private key des Local Work Server hier eingeben>
Address = 192.168.4.2/32
[Peer]
PublicKey = <public key vom VPS Server hier>
AllowedIPs = 192.168.4.1/32
Endpoint = <öffentliche ipv4 adresse des VPS Server hier>:55107
PersistentKeepalive = 25
```

2.4 VPS Server

Eine evtl. vorhandene Firewall auf dem VPS Server für den Port 55107, 80 und 443 öffnen, dann:

Code

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

Dort unten anhängen:

Code

```
net.ipv4.ip_forward=1
net.ipv6.conf.all.forwarding=1
```

Mit diesem Befehl die Weiterleitung aktivieren:

Code

```
sudo sysctl -p
```

2.5 VPS Server & Local Work Server:

Code

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

2.6 VPS Server

*Achtung: Ersetze **eth0** mit dem Namen deiner Netzwerkarte. Finde diese mit folgendem Befehle heraus:*

Code

```
ip -c a
```

Gebe dann die folgenden (korrigierten) Befehle ein:

Code

```
sudo iptables -P FORWARD DROP
sudoiptables-AFORWARD-ieth0-owg0-ptcp--syn--dport80-mconntrack--ctstateNEW-jACCEPT
sudoiptables-AFORWARD-ieth0-owg0-ptcp--syn--dport443-mconntrack--ctstateNEW-jACCEPT
sudoiptables-AFORWARD-iwg0-oeth0-mconntrack--ctstateESTABLISHED,RELATED-jACCEPT
sudoiptables-AFORWARD-iwg0-oeth0-mconntrack--ctstateESTABLISHED,RELATED-jACCEPT
sudoiptables-tnat-APREROUTING-ieth0-ptcp--dport80-jDNAT--to-destination192.168.4.2
sudoiptables-tnat-APREROUTING-ieth0-ptcp--dport443-jDNAT--to-destination192.168.4.2
sudoiptables-tnat-APOSTROUTING-owg0-ptcp--dport80-d192.168.4.2-jSNAT--to-source192.168.4.1
sudo iptables -t nat -A POSTROUTING -o wg0 -p tcp --dport 443 -d 192.168.4.2 -j SNAT --to-so
```

Hier wird das Routing eingestellt, evtl. angepasste IP Adressen aus der Konfiguration hier auch anpassen. Falls Ports hinzukommen sollen, muss das auch mit diesen Befehlen weiter geleitet werden.

Diese Regeln können mit den folgenden Befehlen dauerhaft gespeichert werden:

Code

```
sudo apt install netfilter-persistent -y
sudo netfilter-persistent save
sudo systemctl enable netfilter-persistent
sudo apt install iptables-persistent
```

Nach dem letztem Befehl zwei mal "Yes" mit der Eingabetaste wählen.

Firewall aktivieren mit:

Code

```
sudo ufw allow 22
sudo ufw allow 55107
sudo ufw allow 80
sudo ufw allow 443
sudo ufw enable
sudo ufw status
```

ACHTUNG: der Befehl sudo allow 22 gibt den Port 22 frei um per ssh noch auf den Server zu kommen, falls man das nicht macht, sperrt man sich aus. Hier evtl. auch weitere Ports mit allow öffnen, welche man benötigt.

JETZT SOLLTE BEREITS DIE VERBINDUNG BESTEHEN!!

Falls noch nicht geschehen installiert auf dem Local Work Server zb. docker, docker.compose, Portainer und Nginx reverse Proxy. Wenn Ihr nun eine öffentliche DNS auf Euren VPS Server umleitet kann dann der NGINX direkt auf Port 80 & 443 die Anfragen aus dem Internet intern weiterleiten.

2.7 Zusatz zu Firewall:

In diesem Szenario ist im eigenem Netzwerk KEINE Freigabe in der Firewall nötig, der Work Server baut immer aktiv ein Verbindung zum VPS Server auf. Im VPS Server müssen die PORTS: 80 (TCP), 443 (TCP), 55107 (UDP) und evtl. 22 für den SSH Zugang frei gegeben sein (Wichtig: Bei IONOS muss dies in der Server Verwaltung eingestellt werden)

2.8 Weitere Verbindungen:

Man kann auch weitere Server über einen Tunnel mit dem Wireguard VPS Server verbinden, aber nur mit anderen Ports. Dies erreicht man einfach dadurch, dass man die Anleitung von oben erneut ausführt, aber diesmal anstatt den Namen wg0.conf einfach wg1.conf benutzt und dieses an allen entsprechenden Stellen austauscht.

2.9 Probleme & Lösungen:

Die Verbindung ist äusserst stabil und performant.

Ich hatte aber am Anfang folgendes Problem: Die linux Firewall UFW wird eigentlich mit "sudo UFW enable" dauerhaft gestartet, auch nach einem Neustart muss die Firewall automatisch aktiviert werden. Aus irgendeinem Grunde hat sie das bei mir nicht richtig gemacht. Das Problem habe ich gelöst in dem ich einfach unter "sudo crontab -e" die Firewall nach einem Neustart wieder aktivieren mit "@reboot sudo UFW enable" -> Das muss man aber nur machen wenn die Firewall aus geht nach einem Neustart.

Disclaimer:

Alle Anleitungen/Tutorials sind nach bestem Wissen und Gewissen verfasst, gehen immer von den definierten Software/Firmware-Versionen aus und sind auf das englische GUI ausgelegt.

Es gibt keine Garantien auf Erfolg. Im Falle eines Misserfolges hilft aber sicherlich die Community hier immer weiter.

Keiner der Autoren oder der Betreiber des Forums ist für die aus der Nutzung resultierenden Probleme/Herausforderungen verantwortlich.

Jegliche hier beschriebenen Schritte erfolgen ausnahmslos in eigener Verantwortung des Durchführenden. Eltern haften für ihre Kinder.

Auswählen:

Gültige Software-Version Keine Firmware-Relevanz!