

Basis für Client Config

Es müssen alle Felder der Art `<Text>` geändert werden. Die Zeichen `<` und `>` sind zu entfernen.

Was wird benötigt

Allgemeine Informationen (erhaltet ihr von uns):

- Sentinel IP
- Sentinel Wireguard Public Key
- Secret für udp2raw

An euch angepasst:

- Private Key
 - Option 1: Ihr generiert eure Keys (Private/Public) und gebt uns den Public Key.
 - Zum Erstellen von einem Keypair im Terminal: `wg genkey | tee private | wg pubkey`
`> public`
 - Option 2: Wir generieren die Keys und geben euch den Private Key
- Von uns euch zugewiesene Client IP (für jedes Netz eine)

Die MTU muss für udp2raw herunter gesetzt werden. In den unteren Configs auf jeden Fall die MTU lassen und nicht verändern! Das gilt auch für Verbindungen ohne udp2raw

Wireguard Nativ (UDP)

```
# Client configuration
[Interface]
Address = <zugewiesene Client IP>
DNS = 10.128.0.3
PrivateKey = <Private Key>
MTU=1342

# Server configuration
[Peer]
```

```
PublicKey = <Sentinel Public Key>
AllowedIPs = 0.0.0.0/0
Endpoint = <Sentinel IP>:<UDP Port>
```

Kill Switch

Der folgende von Mullvad erstellte Code implementiert einen Kill Switch, welcher verhindert das Traffic am VPN Tunnel vorbei geht. Insbesondere relevant für IPv6! Er kann **auf Linux** der Config hinzugefügt.

```
PostUp = iptables -I OUTPUT ! -o %i -m mark ! --mark $(wg show %i fwmark) -m addrtype ! --dst-type LOCAL -j
REJECT && ip6tables -I OUTPUT ! -o %i -m mark ! --mark $(wg show %i fwmark) -m addrtype ! --dst-type LOCAL -j
REJECT
PreDown = iptables -D OUTPUT ! -o %i -m mark ! --mark $(wg show %i fwmark) -m addrtype ! --dst-type LOCAL -j
REJECT && ip6tables -D OUTPUT ! -o %i -m mark ! --mark $(wg show %i fwmark) -m addrtype ! --dst-type LOCAL
-j REJECT
```

Getunnelt mit TCP

Use case: Öffentliche Netzwerke wie bspw. an einer Universität oder in der Bahn blockieren häufig UDP Verkehr. Hierbei wird [udp2raw](#) verwendet, um einen verschlüsselten Fake TCP Tunnel zwischen dem Client und Server zu erstellen. Über diesen Tunnel wird dann die eigentliche VPN Verbindung aufgebaut. Wir verwenden gängige Ports, wie bspw. 443/HTTPS, um Blockierungen in Firewalls zu umgehen.

Erfordert [udp2raw](#) auf dem Client Gerät.

Anleitung für Android Geräte [mit root](#), bzw. Workaround für [ohne root](#).

```
# Client configuration
[Interface]
Address = <zugewiesene Client IP>
DNS = 10.128.0.3
PrivateKey = <Private Key>
MTU=1342

# Verhindert dass udp2raw traffic auch über den VPN Tunnel gesendet wird
```

```
PostUp = ip rule add to <Sentinel IP> lookup main
PostDown = ip rule del to <Sentinel IP> lookup main
```

```
# Start/Stop von udp2raw
```

```
PreUp = udp2raw -c -l 127.0.0.1:50001 -r <Sentinel IP>:<TCP Port> -k "<udp2raw Secret>" -a
>/var/log/udp2raw.log 2>&1 &
PostDown = killall udp2raw || true
```

```
# Server configuration
```

```
[Peer]
```

```
PublicKey = <Sentinel Public Key>
```

```
AllowedIPs = 0.0.0.0/0
```

```
Endpoint = 127.0.0.1:50001
```

Kill Switch

Der folgende von Mullvad erstellte Code implementiert einen Kill Switch, welcher verhindert das Traffic nicht durch den VPN Tunnel geht. Insbesondere relevant für IPv6! Wir haben ihn etwas erweitert um Traffic zu Sentinel zuzulassen, da sonst die udp2raw Verbindung bricht. Er kann **auf Linux** der Config hinzugefügt.

```
PostUp = iptables -I OUTPUT ! -d <Sentinel IP> ! -o %i -m mark ! --mark $(wg show %i fwmark) -m addrtype ! --dst-type LOCAL -j REJECT && ip6tables -I OUTPUT ! -o %i -m mark ! --mark $(wg show %i fwmark) -m addrtype ! --dst-type LOCAL -j REJECT
PreDown = iptables -D OUTPUT ! -d <Sentinel IP> ! -o %i -m mark ! --mark $(wg show %i fwmark) -m addrtype ! --dst-type LOCAL -j REJECT && ip6tables -D OUTPUT ! -o %i -m mark ! --mark $(wg show %i fwmark) -m addrtype ! --dst-type LOCAL -j REJECT
```

Zwischen Netzen wechseln

Zum Wechseln der Netze genügt es in der Config die **zugewiesene Client IP**, bzw. die Netzadresse und den **Port** zu ändern. Daten der Netze können der [Tabelle](#) entnommen werden.

Hinweise zu den Keys

Alle Netze verwenden dieselben Keys (Wireguard und udp2raw). Auch euer Public Key wird in allen Netzen hinterlegt. Das bedeutet dass zu Verschiedenen Netzen Verbindungen gleichzeitig genutzt

werden können. Wer mehrere Connections möchte, dem empfehlen wir allerdings einfach mehrere Keys zu hinterlegen. Somit bleibt nämlich die Möglichkeit eines einfachen Netzwechsels (durch ändern von zwei Parametern) erhalten. Die Idee ist also so viel Flexibilität wie möglich mit so wenig Informationen (Bsp. Keys) wie nötig.

Revision #31

Created 17 May 2024 06:41:49 by Mike

Updated 27 May 2024 18:18:31 by Mike