

# Praktikum Rechnernetze

Protokoll zu Versuch 7 (OpenVPN) von Gruppe 1

---

Jakob Waibel Daniel Hiller Elia Wüstner Felicitas Pojtinger

2021-11-30

# Einführung

---

Diese Materialien basieren auf Professor Kiefers “Praktikum Rechnernetze”-Vorlesung der HdM Stuttgart.

**Sie haben einen Fehler gefunden oder haben einen Verbesserungsvorschlag?** Bitte eröffnen Sie ein Issue auf GitHub ([github.com/pojntfx/uni-netpractice-notes](https://github.com/pojntfx/uni-netpractice-notes)):



Abbildung 1: QR-Code zum Quelltext auf GitHub

Dieses Dokument und der enthaltene Quelltext ist freie Kultur bzw. freie Software.



**Abbildung 2:** Badge der AGPL-3.0-Lizenz

Uni Network Practice Notes (c) 2021 Jakob Waibel, Daniel Hiller, Elia Wüstner, Felicitas Pojtinger

SPDX-License-Identifier: AGPL-3.0

CA (=Zertifizierungsstelle) und  
Schlüssel erzeugen und signieren

---

## CA (=Zertifizierungsstelle) und Schlüssel erzeugen und signieren

Verzeichnis erstellen und betreten:

```
# mkdir openvpn  
# cd openvpn
```

Git installieren:

```
apt install git
```

Repository klonen:

```
# git clone https://github.com/OpenVPN/easy-rsa  
Cloning into 'easy-rsa'...  
remote: Enumerating objects: 2095, done.  
remote: Counting objects: 100% (13/13), done.  
remote: Compressing objects: 100% (11/11), done.  
remote: Total 2095 (delta 3), reused 4 (delta 0), pack-reused 2082  
Receiving objects: 100% (2095/2095), 11.72 MiB | 7.01 MiB/s4, 4 objects, 1.00 KiB | 0.00 MiB/s, done.
```

### Beschreiben Sie kurz den Sinn der Dateien in diesen Ordnern

Die ca.crt Datei ist öffentlich. User, Server und Client können damit beweisen, dass sie sich im selben vertrauten Netz befinden. Jeder daran beteiligte User und Server muss eine Kopie dieser Datei besitzen.

ca.key ist der private Schlüssel, mit dem die CA Zertifikate für Server und Clients signiert werden. Die ca.key Datei sollte nur auf der CA Maschine liegen, denn der Schlüssel darf nicht in die Hände eines Angreifers gelangen.

Die Private Keys liegen im Ordner private und im Ordner issued sind die signierten Zertifikate (Public Keys) für eine gegenseitige Bestätigung zwischen Server und Client.

Der Ordner certs\_by\_serial enthält alle von der CA signierten Zertifikate mit ihrer Seriennummer.

# Konfiguration von Client und Server

---



## Server konfigurieren

Analog zu der in der Versuchsanleitung geschilderten Konfigurationsdatei wird im Folgenden eine angepasste `server.conf` dargestellt:

```
# cat server.conf
proto udp
dev tun
ca pki/ca.crt
cert pki/issued/server-g1.crt
key pki/private/server-g1.key
dh pki/dh.pem
server 10.8.1.0 255.255.255.0
keepalive 10 120
comp-lzo
persist-key
persist-tun
verb 3
```

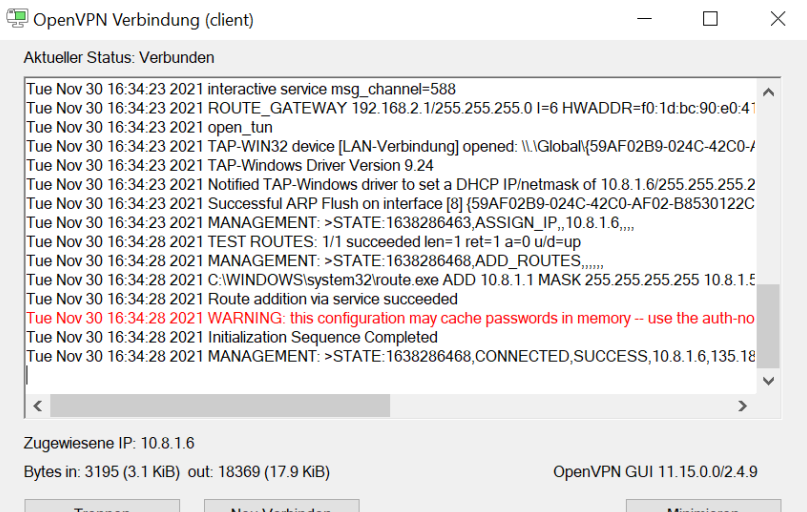
Erklären Sie die einzelnen Parameter/Optionen der „server.conf“ und der „client.conf“.

Client:

```
client # Definiert dass es sich um ein Client-Setup handelt
dev tun # Als virtuelles Netzwerkgerät
proto udp # Hier wird festgelegt, welches Protokoll verwendet wird
remote 135.181.204.42 1194 # Gibt an mit welcher Adresse und auf welchem Port die Verbindung hergestellt wird
nobind # Veranlasst OpenVPN dazu einen bind() Aufruf zu unterlassen
persist-key # Versucht Zustände über den Keyfile zu persistieren
persist-tun # Versucht Zustände über den Tun-Device zu persistieren
ca ca.crt # Gibt den Pfad zur Zertifikatsdatei an
cert issued/client-g1.crt # Gibt den Pfad zur Zertifikatsdatei an
key private/client-g1.key # Gibt den Pfad zur Key-Datei an
comp-lzo # Definiert dass keine Kompression verwendet wird
verb 3 # Definiert die Ausführlichkeit der Log-Output
```

Server:

Versuchen Sie ebenfalls mit einem Windows-Client eine Verbindung zu Ihrem Server aufzubauen. Die Client-Software können Sie von: <https://openvpn.net/index.php/open-source/downloads.html> herunterladen.



The screenshot shows the OpenVPN GUI client window titled "OpenVPN Verbindung (client)". The window has standard Windows window controls (minimize, maximize, close) in the top right corner. The main content area displays the current status as "Verbunden" (Connected). Below this is a scrollable log window showing the following text:

```
Tue Nov 30 16:34:23 2021 interactive service msg_channel=588
Tue Nov 30 16:34:23 2021 ROUTE_GATEWAY 192.168.2.1/255.255.255.0 I=6 HWADDR=f0:1d:bc:90:e0:41
Tue Nov 30 16:34:23 2021 open_tun
Tue Nov 30 16:34:23 2021 TAP-WIN32 device [LAN-Verbindung] opened: \\.\Global{59AF02B9-024C-42C0-42C0-AF02-B8530122C}
Tue Nov 30 16:34:23 2021 TAP-Windows Driver Version 9.24
Tue Nov 30 16:34:23 2021 Notified TAP-Windows driver to set a DHCP IP/netmask of 10.8.1.6/255.255.255.2
Tue Nov 30 16:34:23 2021 Successful ARP Flush on interface [8] {59AF02B9-024C-42C0-AF02-B8530122C}
Tue Nov 30 16:34:23 2021 MANAGEMENT: >STATE:1638286463,ASSIGN_IP,,10.8.1.6,,,,
Tue Nov 30 16:34:28 2021 TEST ROUTES: 1/1 succeeded len=1 ret=1 a=0 u/d=up
Tue Nov 30 16:34:28 2021 MANAGEMENT: >STATE:1638286468,ADD_ROUTES,,,,,
Tue Nov 30 16:34:28 2021 C:\WINDOWS\system32\route.exe ADD 10.8.1.1 MASK 255.255.255.255 10.8.1.5
Tue Nov 30 16:34:28 2021 Route addition via service succeeded
Tue Nov 30 16:34:28 2021 WARNING: this configuration may cache passwords in memory -- use the auth-no
Tue Nov 30 16:34:28 2021 Initialization Sequence Completed
Tue Nov 30 16:34:28 2021 MANAGEMENT: >STATE:1638286468,CONNECTED,SUCCESS,10.8.1.6,135.18
```

Below the log window, the assigned IP address is shown as "Zugewiesene IP: 10.8.1.6". At the bottom of the window, the traffic statistics are "Bytes in: 3195 (3.1 KiB) out: 18369 (17.9 KiB)". The version of the OpenVPN GUI is "OpenVPN GUI 11.15.0.0/2.4.9". At the very bottom, there are buttons for "Trennen" (Disconnect), "Neue Verbindung" (New Connection), and "Minimieren" (Minimize).

# Analyse

---

## Analyse der Logs

Inspizieren Sie die Log-Statements des Servers und des Clients. Ist ein Tunnel etabliert?

Client-Log:

```
# sudo openvpn --config client.conf
```

```
[sudo] password for root:
```

```
2021-11-30 15:58:20 WARNING: Compression for receiving enabled
```

```
2021-11-30 15:58:20 --cipher is not set. Previous OpenVPN version
```

```
2021-11-30 15:58:20 OpenVPN 2.5.3 x86_64-suse-linux-gnu [SSL
```

```
2021-11-30 15:58:20 library versions: OpenSSL 1.1.1l
```

```
24 Aug 2021, LZO 2.10
```

```
2021-11-30 15:58:20 WARNING: No server certificate verification
```

```
See http://openvpn.net/howto.html#mitm for more info.
```

```
2021-11-30 15:58:20 TCP/UDP: Preserving recently used remote
```

```
2021-11-30 15:58:20 Socket Buffers: R=[212992->212992] S=[212
```

```
2021-11-30 15:58:20 UDP link local: (not bound)
```

# Funktionstest

Überprüfen Sie mit den Tools `ip link`, `ip address` und `ip route` die erzeugten Netzwerkkonfigurationen. Im Anschluss überprüfen Sie die Funktion des Tunnels mit einem Ping vom Client auf das `tun0` Device des Servers.

Zuerst verwenden wir `ip a`:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state U
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp2s0f0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast
    link/ether 84:a9:38:67:f2:18 brd ff:ff:ff:ff:ff:ff
3: wlp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
    link/ether c8:04:02:bd:60:52 brd ff:ff:ff:ff:ff:ff
```

## Betrachtung via Wireshark

---

# Betrachtung via Wireshark

Stellen Sie den Unterschied der Datenpakete (verschlüsselt, unverschlüsselt) mit Wireshark dar. Nutzen Sie dazu einen einfachen ping-Befehl. Beachten Sie, dass der Verkehr für Wireshark auf unterschiedlichen Interfaces stattfindet.

The screenshot shows the Wireshark interface with the following details:

- Packet List:** A table of captured packets. The first few rows show ICMP Echo (ping) requests from 192.168.1.204 to 192.168.1.1. The 'Info' column indicates the protocol and length, such as 'ICMP Echo (ping) 56(84) bytes of data'.
- Packet Details:** Expanded for a selected packet, showing:
  - Frame 12644: 80 bytes on wire (640 bits), 80 bytes captured (640 bits) on interface empy3a0, 0 s
  - Ethernet II, Src: VMXSVN014F14:08 (08:00:27:14:08), Dst: Micro-SG\_c42F:1c (08:00:27:1c:02:1c)
  - Internet Protocol Version 4, Src: 192.168.1.204, Dst: 192.168.1.23
  - User Datagram Protocol, Src Port: 55744, Dst Port: 55744
  - Source Port: 55744
  - Destination Port: 55744
  - Length: 40
  - Checksum: 9032a [unverified]
  - [Checksum Status] [Checksum] [Stream index: 18]
  - [Timeline]
  - UDP payload (38 bytes)
  - Type: 0x30 (opcode/key\_id)
  - 001: 0 ... = Opcode: P\_DATA\_V1 (0x00)
  - ... = Key ID: 0
  - Data (37 bytes)
- Packet Bytes:** A hex dump of the packet data, showing the raw bytes of the ICMP Echo request.

On the right side of the screenshot, a terminal window shows the execution of a ping command:

```
[danny@localhost g]ls ping 10.8.1.1
PING 10.8.1.1 (10.8.1.1) 56(84) bytes of data.
64 bytes from 10.8.1.1: icmp_seq=1 ttl=64 time=45.2 ms
64 bytes from 10.8.1.1: icmp_seq=2 ttl=64 time=39.1 ms
64 bytes from 10.8.1.1: icmp_seq=3 ttl=64 time=43.7 ms
64 bytes from 10.8.1.1: icmp_seq=4 ttl=64 time=43.9 ms
64 bytes from 10.8.1.1: icmp_seq=5 ttl=64 time=43.4 ms
64 bytes from 10.8.1.1: icmp_seq=6 ttl=64 time=43.1 ms
64 bytes from 10.8.1.1: icmp_seq=7 ttl=64 time=42.2 ms
64 bytes from 10.8.1.1: icmp_seq=8 ttl=64 time=43.1 ms
64 bytes from 10.8.1.1: icmp_seq=9 ttl=64 time=44.2 ms
64 bytes from 10.8.1.1: icmp_seq=10 ttl=64 time=41.8 ms
64 bytes from 10.8.1.1: icmp_seq=11 ttl=64 time=41.9 ms
^C
--- 10.8.1.1 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 39.118/42.866/45.201/1.546 ms
[danny@localhost g]ls
```

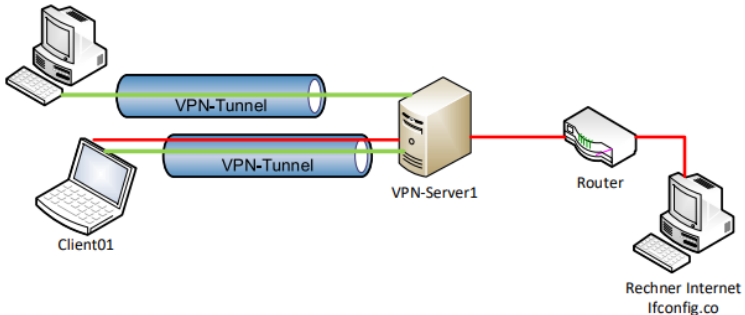


## Erweiterte Konfiguration

---

# Erweiterte Konfiguration

\*\* Bis hierher haben wir nur Datenverbindung vom Client bis zum Server realisiert (In der Grafik grün dargestellt). Der Sinn einer VPN-Verbindung ist häufig die Network-to-Network-Anbindung. Eine ähnliche Verbindung ist eine Client-Verbindung über den VPN-Server nach draußen ins Internet. Folgende Grafik veranschaulicht die gewünschte Verbindung (rot dargestellt):\*\*



## Änderung der Konfiguration

Die Datei `server.conf` muss um die IP des servers von `api.ipify.org` erweitert werden. Mit Dig können die IPs der Server verwendet werden. Wir erhalten hier mehrere IPs, da anscheinend Loadbalancing verwendet wird:

```
# dig api.ipify.org

; <<>> DiG 9.16.23-RH <<>> api.ipify.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52052
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 0

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:;, udp: 65494
;; QUESTION SECTION:
;api.ipify.org.                IN      A
```

Starten Sie den Open-VPN Client neu. Überprüfen Sie die Routen.

Nach dem Neustarten des Clients sehen die Routen wie folgt aus:

```
# ip route get 54.91.59.199
54.91.59.199 via 10.8.1.5 dev tun0 src 10.8.1.6 uid 1000
    cache

# ip route get 52.20.78.240
52.20.78.240 via 10.8.1.5 dev tun0 src 10.8.1.6 uid 1000
    cache

# ip route get 3.232.242.170
3.232.242.170 via 10.8.1.5 dev tun0 src 10.8.1.6 uid 1000
    cache

# ip route get 3.220.57.224
3.220.57.224 via 10.8.1.5 dev tun0 src 10.8.1.6 uid 1000
```

Zugriffsbeschränkung

---

**\*\*** Angenommen ein Client soll keinen Zugriff mehr über Ihren OpenVPN-Server erhalten. Wie verhindern Sie das, ohne dass Sie Zugang zum Client bekommen? Am Ende des Versuchs können sie die Methode für alle vergebenen Client-Zertifikate durchführen und testen. Können Sie diesen Vorgang wieder rückgängig machen, so das der Client wieder am VPN „teilnehmen“ kann?**\*\***

## Widerruf

Wenn wir das Zertifikat widerrufen, führt dies dazu, dass das Zertifikat ungültig wird und nicht mehr für Authentifizierungszwecke genutzt werden kann.

Dies kann mit folgendem Kommando geschehen:

```
# ./revoke -full client -g1
```

Durch das vorangegangene Kommando wurde eine CRL-Datei erstellt