

Praktikum Rechnernetze

Protokoll zu Versuch 7 (OpenVPN) von Gruppe 1

Jakob Waibel Daniel Hiller Elia Wüstner Felix 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/poijntfx/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, Felix 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  
Receiving objects: 100% (2095/2095), 11.72 MiB | 7.01 MiB/s4,
```

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
```

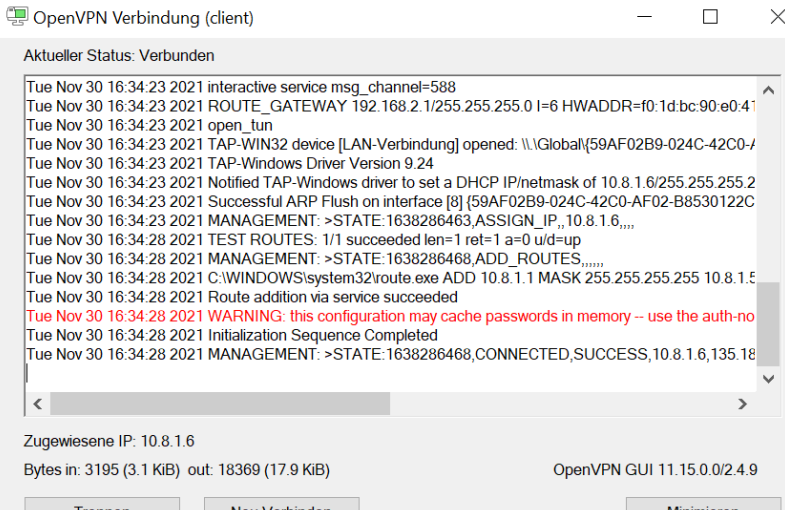
Erklären Sie die einzelnen Parameter/Optionen der „server.conf“ und der „client.conf“.

Client:

client	# Definiert dass es sich um ein
dev tun	# Als virtuelles Netzwerkgerät
proto udp	# Hier wird festgelegt, welche
remote 135.181.204.42 1194	# Gibt an mit welcher Adresse
nobind	# Veranlasst OpenVPN dazu einen
persist-key	# Versucht Zustände über den N
persist-tun	# Versucht Zustände über den N
ca ca.crt	# Gibt den Pfad zur Zertifikats
cert issued/client-g1.crt	# Gibt den Pfad zur Zertifikats
key private/client-g1.key	# Gibt den Pfad zur Key-Datei
comp-lzo	# Definiert dass keine Kompres
verb 3	# Definiert die Ausführlichkei

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.



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.

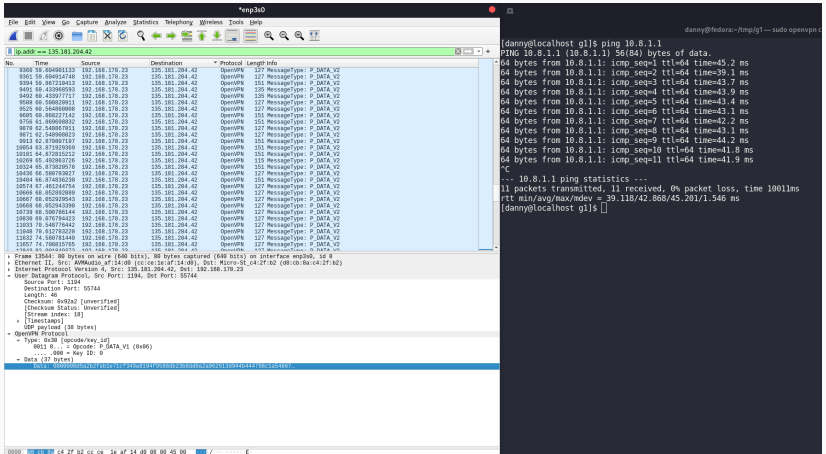
Zuerst verwenden wir `ip a`:

```
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:a9:38:67:f2:18 brd ff:ff:ff:ff:ff:ff
```

Betrachtung via Wireshark

Betrachtung via Wireshark

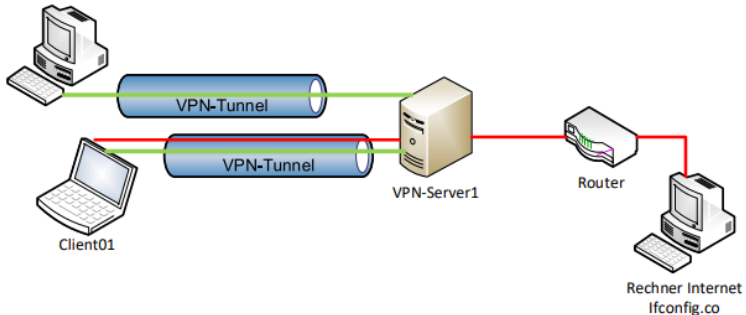
Stellen Sie den Unterschied der Datenpakete (verschlüsselt, unverschlüsselt) mit Wireshark dar. Nutzen Sie dazu einen einfachen ping-Befehl. Beachten Sie, dass der Verkehr für Wireshark auf unterschiedlichen Interfaces stattfindet.



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