

Mandatory assignment



Introduktion

I denne rapport gennemgår vi opsætningen og konfigurationen af en sikker Virtual Private Network (VPN) løsning ved hjælp af OpenVPN. Formålet med opgaven er at skabe en sikker forbindelse mellem en ekstern medarbejder og virksomhedens interne netværk. Dette opnås ved hjælp af en OpenVPN-server installeret på en Linux-distribution (Ubuntu), og en OpenVPN-klient installeret på en Windows-maskine.

Opgaven er delt op i to dele: en praktisk del, hvor der skal konfigureres en VPN-forbindelse, og en teoretisk del, som omhandler VPN-koncepter som certifikater, autentifikation og moderne angreb på SSL/TLS-protokoller. Formålet med denne rapport er at dokumentere de vigtigste trin i opsætningen af OpenVPN samt besvare teoretiske spørgsmål om sikkerhed og VPN-teknologi.

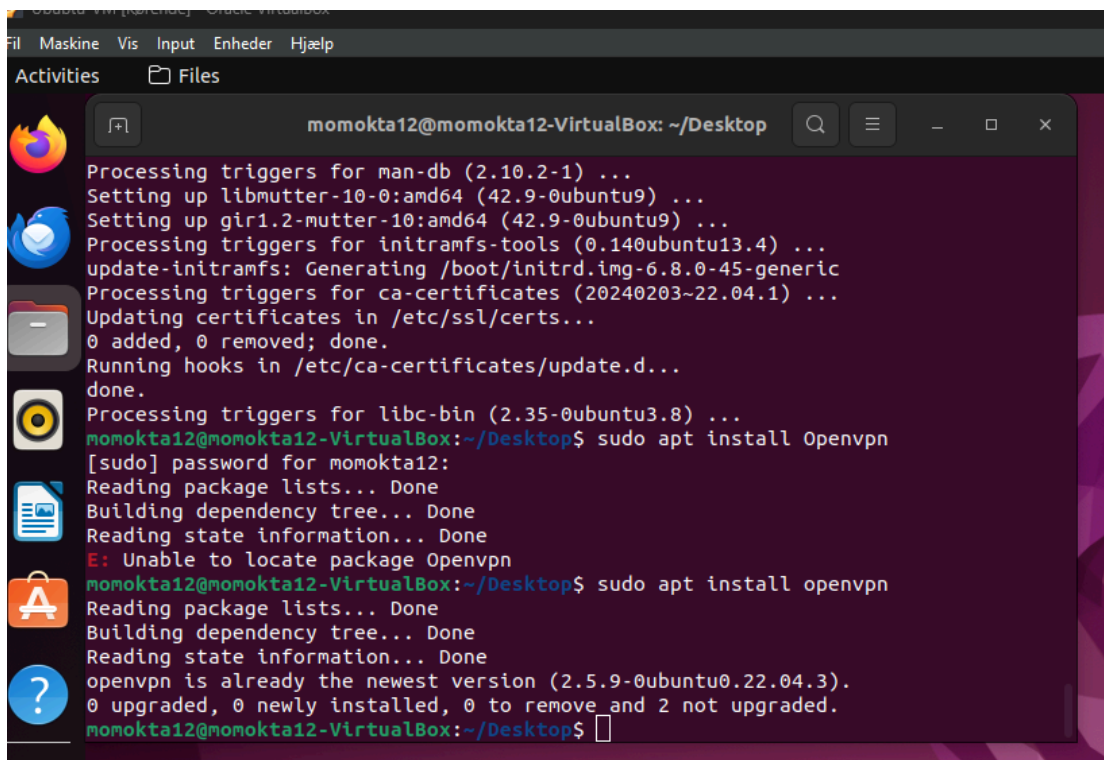
Praktisk del

1. Installation af OpenVPN Server

For at starte opsætningen af OpenVPN-serveren på Ubuntu, opdaterede jeg systemet og installerede OpenVPN. Installations Trinene blev udført som følger:

1. Systemet blev opdateret med kommandoerne "`sudo apt update`" og "`sudo apt upgrade`".
2. OpenVPN blev installeret med kommandoen "`sudo apt install openvpn`", men det viste sig, at den nyeste version allerede var installeret på systemet.

Screenshot: Installationsprocessen vises på billedet nedenfor.

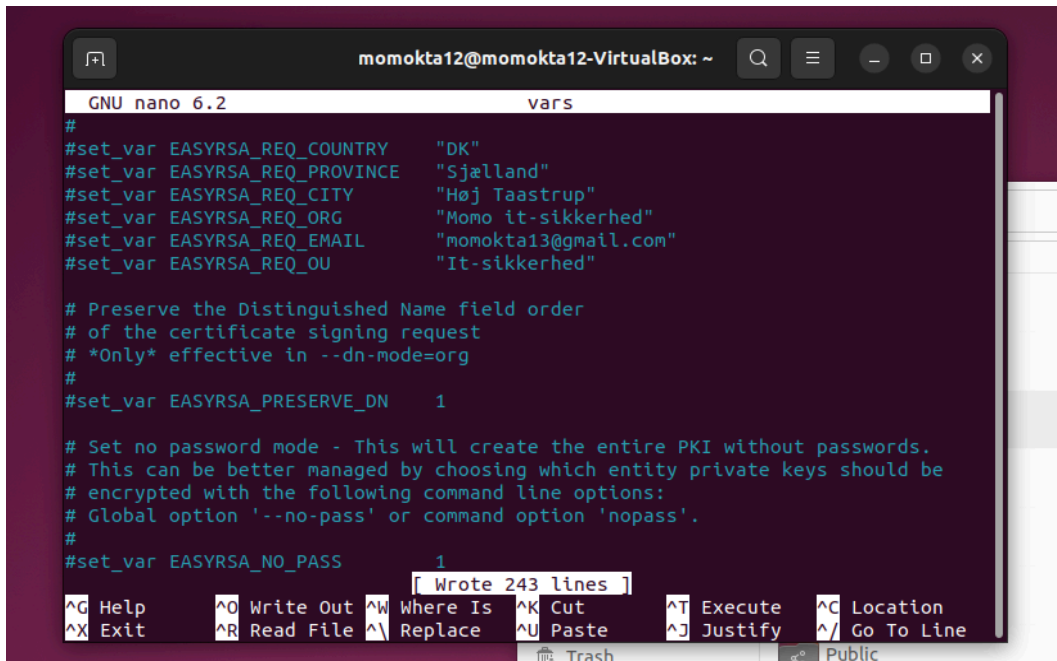


```
Processing triggers for man-db (2.10.2-1) ...
Setting up libmutter-10-0:amd64 (42.9-0ubuntu9) ...
Setting up gir1.2-mutter-10:amd64 (42.9-0ubuntu9) ...
Processing triggers for initramfs-tools (0.140ubuntu13.4) ...
update-initramfs: Generating /boot/initrd.img-6.8.0-45-generic
Processing triggers for ca-certificates (20240203-22.04.1) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...
momokta12@momokta12-VirtualBox: ~/Desktop
momokta12@momokta12-VirtualBox:~/Desktop$ sudo apt install Openvpn
[sudo] password for momokta12:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package Openvpn
momokta12@momokta12-VirtualBox:~/Desktop$ sudo apt install openvpn
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openvpn is already the newest version (2.5.9-0ubuntu0.22.04.3).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
momokta12@momokta12-VirtualBox:~/Desktop$
```

1.1 Vars

For at konfigurere certifikaterne til OpenVPN-serveren brugte jeg `Easy RSA-værktøjet`. Jeg tilpassede vars-filen med mine egne oplysninger, såsom land, by, organisation og e-mail. Dette er nødvendigt for at generere de korrekte certifikater og nøgler, som sikrer kommunikationen mellem server og klient.

Screenshot: Tilpasning af Vars-filen i "Easy RSA".



```
momokta12@momokta12-VirtualBox: ~
GNU nano 6.2 vars
#
#set_var EASYRSA_REQ_COUNTRY    "DK"
#set_var EASYRSA_REQ_PROVINCE   "Sjælland"
#set_var EASYRSA_REQ_CITY       "Høj Taastrup"
#set_var EASYRSA_REQ_ORG        "Momo it-sikkerhed"
#set_var EASYRSA_REQ_EMAIL      "momokta13@gmail.com"
#set_var EASYRSA_REQ_OU         "It-sikkerhed"

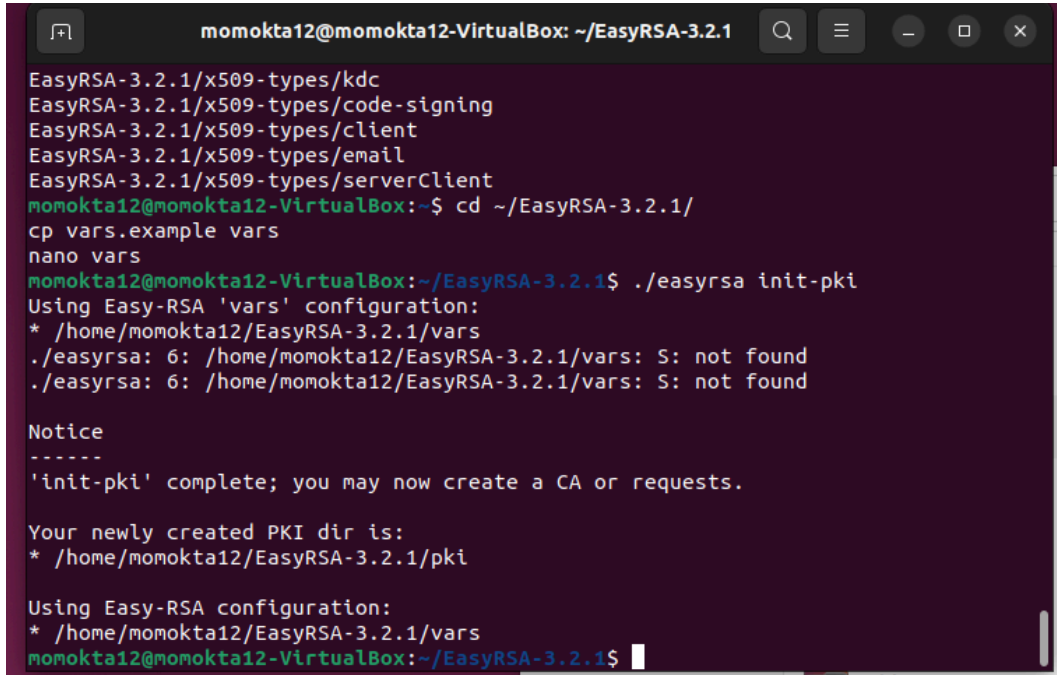
# Preserve the Distinguished Name field order
# of the certificate signing request
# *Only* effective in --dn-mode=org
#
#set_var EASYRSA_PRESERVE_DN    1

# Set no password mode - This will create the entire PKI without passwords.
# This can be better managed by choosing which entity private keys should be
# encrypted with the following command line options:
# Global option '--no-pass' or command option 'nopass'.
#
#set_var EASYRSA_NO_PASS        1
Wrote 243 lines
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^\ Replace  ^U Paste    ^J Justify  ^_ Go To Line
Trash      Public
```

1.2 PKI

Efter at have tilpasset Vars-filen, initialiserede jeg Public Key Infrastructure (PKI) ved hjælp af `Easy RSA-værktøjet`. Dette trin er nødvendigt for at kunne generere CA-certifikatet og andre nødvendige nøgler.

Screenshot: Initialisering af PKI ved hjælp af "Easy RSA".



```
momokta12@momokta12-VirtualBox: ~/EasyRSA-3.2.1
EasyRSA-3.2.1/x509-types/kdc
EasyRSA-3.2.1/x509-types/code-signing
EasyRSA-3.2.1/x509-types/client
EasyRSA-3.2.1/x509-types/email
EasyRSA-3.2.1/x509-types/serverClient
momokta12@momokta12-VirtualBox:~$ cd ~/EasyRSA-3.2.1/
cp vars.example vars
nano vars
momokta12@momokta12-VirtualBox:~/EasyRSA-3.2.1$ ./easyrsa init-pki
Using Easy-RSA 'vars' configuration:
* /home/momokta12/EasyRSA-3.2.1/vars
./easyrsa: 6: /home/momokta12/EasyRSA-3.2.1/vars: S: not found
./easyrsa: 6: /home/momokta12/EasyRSA-3.2.1/vars: S: not found

Notice
-----
'init-pki' complete; you may now create a CA or requests.

Your newly created PKI dir is:
* /home/momokta12/EasyRSA-3.2.1/pki

Using Easy-RSA configuration:
* /home/momokta12/EasyRSA-3.2.1/vars
momokta12@momokta12-VirtualBox:~/EasyRSA-3.2.1$
```

1.3 oprettelse af CA-certifikatet

Efter initialisering af PKI fortsatte jeg med at oprette CA-certifikatet (Certificate Authority) ved hjælp af følgende kommando:

```
“./easyrsa build-ca nopass”
```

Dette skabte CA-certifikatet, som er nødvendigt for at signere server- og klient-certifikaterne.

Certifikatet blev oprettet under stien **/home/momokta12/EasyRSA-3.2.1/pki/ca.crt**.

Screenshot: Oprettelse af CA-certifikat ved hjælp af “Easy RSA”.



6

```
momokta12@momokta12-VirtualBox: ~/EasyRSA-3.2.1
Requested type: 'server'
Valid for: '825' days

subject=
commonName = server

Type the word 'yes' to continue, or any other input to abort.
Confirm requested details: yes

Using configuration from /home/momokta12/EasyRSA-3.2.1/pki/2d61dcea/temp.1.1
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName :ASN.1 12:'server'
Certificate is to be certified until Jan 10 13:27:02 2027 GMT (825 days)

Write out database with 1 new entries
Data Base Updated

Notice
-----
Inline file created:
* /home/momokta12/EasyRSA-3.2.1/pki/inline/private/server.inline

Notice
-----
Certificate created at:
* /home/momokta12/EasyRSA-3.2.1/pki/issued/server.crt

momokta12@momokta12-VirtualBox:~/EasyRSA-3.2.1$
```

1.5 Generering af Diffie-Hellman-nøgle

For at sikre den krypterede forbindelse genererede jeg Diffie-Hellman-parametrene ved hjælp af følgende kommando:

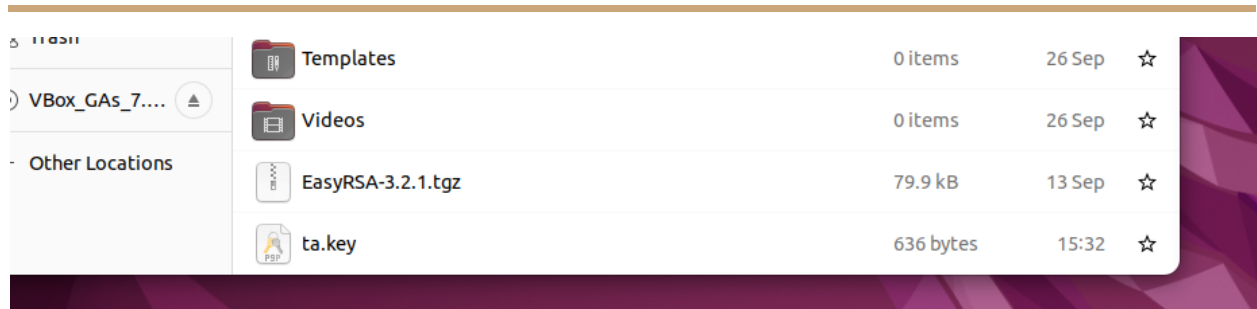
```
"./easyrsa gen-dh"
```

Dette genererede en `dh.pem`-fil, som skal bruges i OpenVPN-konfigurationen.

Screenshot: Generering af Diffie-Hellman-nøgle.

8

09/10/2024



1.7 Start og Kørsel af OpenVPN Server

Efter at have konfigureret OpenVPN-serveren blev den startet ved hjælp af systemctl. Her er processen for at starte og kontrollere serveren:

1. Start OpenVPN-serveren

- a. OpenVPN-serveren blev startet med følgende kommando:

```
"sudo systemctl start openvpn@server"
```

2. Kontrol af serverstatus

Status for OpenVPN-serveren blev kontrolleret for at sikre, at serveren kørte korrekt. Output viser, at "Initialization Sequence Completed" er nået, hvilket indikerer, at serveren er startet uden fejl.

- a. "

```
sudo systemctl status openvpn@server"
```

- b. **Screenshot:** Dokumentation af, at OpenVPN-serveren kører korrekt.

```
momokta12@momokta12-VirtualBox: ~/EasyRSA-3.2.1
● openvpn@server.service - OpenVPN connection to server
   Loaded: loaded (/lib/systemd/system/openvpn@.service; disabled; vendor preset: enabled)
   Active: active (running) since Mon 2024-10-07 16:03:16 CEST; 11s ago
     Docs: man:openvpn(8)
           https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
           https://community.openvpn.net/openvpn/wiki/HOWTO
   Main PID: 36193 (openvpn)
   Status: "Initialization Sequence Completed"
     Tasks: 1 (limit: 2271)
    Memory: 3.7M
       CPU: 23ms
   CGroup: /system.slice/system-openvpn.slice/openvpn@server.service
           └─36193 /usr/sbin/openvpn --daemon ovpn-server --status /run/openvpn/server.status 10 --cd /etc/openvpn --script-s

oct 07 16:03:16 momokta12-VirtualBox ovpn-server[36193]: net_addr_v4_add: 10.8.0.1/24 dev tun0
oct 07 16:03:16 momokta12-VirtualBox ovpn-server[36193]: Could not determine IPv4/IPv6 protocol. Using AF_INET
oct 07 16:03:16 momokta12-VirtualBox ovpn-server[36193]: Socket Buffers: R=[212992->212992] S=[212992->212992]
oct 07 16:03:16 momokta12-VirtualBox ovpn-server[36193]: UDPv4 link local (bound): [AF_INET][undef]:1194
oct 07 16:03:16 momokta12-VirtualBox ovpn-server[36193]: UDPv4 link remote: [AF_UNSPEC]
oct 07 16:03:16 momokta12-VirtualBox ovpn-server[36193]: GID set to nogroup
oct 07 16:03:16 momokta12-VirtualBox ovpn-server[36193]: UID set to nobody
oct 07 16:03:16 momokta12-VirtualBox ovpn-server[36193]: MULTI: multi_init called, r=256 v=256
oct 07 16:03:16 momokta12-VirtualBox ovpn-server[36193]: IFCONFIG POOL IPv4: base=10.8.0.2 size=253
oct 07 16:03:16 momokta12-VirtualBox ovpn-server[36193]: Initialization Sequence Completed
```

2. Server Logfil

Efter at have konfigureret og startet OpenVPN-serveren, genererede jeg en logfil, som viser de forskellige hændelser og bekræfter, at serveren kører korrekt. Loggen dokumenterer en succesfuld oprettelse af forbindelse med de korrekte krypteringsalgoritmer og HMAC-autentifikation.

Uddrag fra logfil:

```
2024-10-09 00:35:29 OpenVPN 2.5.9 x86_64-pc-linux-gnu [SSL (OpenSSL)]
[LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] built on Jun 27 2024
```

```
2024-10-09 00:35:29 Incoming Control Channel Authentication: Using 256
bit message hash 'SHA256' for HMAC authentication
```

```
2024-10-09 00:35:29 TUN/TAP device tun0 opened
```

```
2024-10-09 00:35:29 Initialization Sequence Completed
```

Screenshot: Billedet nedenfor viser logfilen, som bekræfter en succesfuld serverforbindelse:

```
momokta12@momokta12-VirtualBox: ~/Desktop
cat: /var/log/openvpn.log: Permission denied
momokta12@momokta12-VirtualBox:~/Desktop$ sudo cat /var/log/openvpn.log
2024-10-09 00:35:29 DEPRECATED OPTION: --cipher set to 'AES-256-CBC' but missing in --data-ciphers (AES-256-GCM:AES-128-GCM). Future OpenVPN version will ignore --cipher for cipher negotiations. Add 'AES-256-CBC' to --data-ciphers or change --cipher 'AES-256-CBC' to --data-ciphers-fallback 'AES-256-CBC' to silence this warning.
2024-10-09 00:35:29 OpenVPN 2.5.9 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] built on Jun 27 2024
2024-10-09 00:35:29 library versions: OpenSSL 3.0.2 15 Mar 2022, LZO 2.10
2024-10-09 00:35:29 WARNING: --keepalive option is missing from server config
2024-10-09 00:35:29 net_route_v4_best_gw query: dst 0.0.0.0
2024-10-09 00:35:29 net_route_v4_best_gw result: via 192.168.1.1 dev enp0s3
2024-10-09 00:35:29 NOTE: your local LAN uses the extremely common subnet address 192.168.0.x or 192.168.1.x. Be aware that this might create routing conflicts if you connect to the VPN server from public locations such as internet cafes that use the same subnet.
2024-10-09 00:35:29 Diffie-Hellman initialized with 2048 bit key
2024-10-09 00:35:29 Outgoing Control Channel Authentication: Using 256 bit message hash 'SHA256' for HMAC authentication
2024-10-09 00:35:29 Incoming Control Channel Authentication: Using 256 bit message hash 'SHA256' for HMAC authentication
2024-10-09 00:35:29 TUN/TAP device tun0 opened
2024-10-09 00:35:29 net_iface_mtu_set: mtu 1500 for tun0
2024-10-09 00:35:29 net_iface_up: set tun0 up
2024-10-09 00:35:29 net_addr_v4_add: 10.8.0.1/24 dev tun0
2024-10-09 00:35:29 Could not determine IPv4/IPv6 protocol. Using AF_INET
2024-10-09 00:35:29 Socket Buffers: R=[212992->212992] S=[212992->212992]
2024-10-09 00:35:29 UDPv4 link local (bound): [AF_INET][undef]:1194
2024-10-09 00:35:29 UDPv4 link remote: [AF_UNSPEC]
2024-10-09 00:35:29 GID set to nogroup
2024-10-09 00:35:29 UID set to nobody
2024-10-09 00:35:29 MULTI: multi_init called, r=256 v=256
2024-10-09 00:35:29 IFCONFIG POOL IPv4: base=10.8.0.2 size=253
2024-10-09 00:35:29 Initialization Sequence Completed
momokta12@momokta12-VirtualBox:~/Desktop$
```

3. Konfiguration af OpenVPN Klient

For at sikre en stabil og sikker forbindelse mellem klienten og serveren, blev der oprettet en OpenVPN-konfigurationsfil for klienten. Denne fil blev genereret ved hjælp af "Easy RSA"-værktøjet og indeholder klientens certifikater samt information om serverforbindelsen.

3.1 Oprettelse af klient-konfigurationsfil

Filen **client1.ovpn** blev oprettet med følgende indhold:

```
"client
```

```
dev tun
```

```
proto udp
```

```
remote 192.168.1.5 1194
```

```
resolv-retry infinite
```

09/10/2024

nobind

persist-key

persist-tun

remote-cert-tls server

cipher AES-256-CBC

auth SHA256

key-direction 1

verb 3

<ca>

-----BEGIN CERTIFICATE-----

MIIDPjCCAiaGAwIBAgIUX3WxLx0a0/h7+9/uF2y...

-----END CERTIFICATE-----

</ca>

<cert>

-----BEGIN CERTIFICATE-----

MIIEwDCCBqigAwIBAgIJAKxjSY4...

-----END CERTIFICATE-----

</cert>

<key>

-----BEGIN PRIVATE KEY-----

MIIEvwIBADANBgkq...

09/10/2024

```
-----END PRIVATE KEY-----  
  
</key>  
  
<tls-auth>  
  
-----BEGIN OpenVPN Static key V1-----  
  
dd57142...  
  
-----END OpenVPN Static key V1-----  
  
</tls-auth>"
```

Forklaring af konfigurationen:

- **client:** Denne linje specificerer, at konfigurationen er for en klient.
- **dev tun:** TUN-enheden bruges til VPN-forbindelsen.
- **proto udp:** VPN bruger UDP-protokollen.
- **remote 192.168.1.5 1194:** Forbindelse til OpenVPN-serverens IP-adresse og port.
- **cipher AES-256-CBC** og **auth SHA256:** Krypteringsalgoritmerne for forbindelsen.
- **ca, cert, key** og **tls-auth:** Indlejrede certifikater og nøgler for autentifikation og sikkerhed.

3.2 Tilføjelse af klient profiler

Klientens profil blev importeret til OpenVPN-klienten, som kører på Windows.

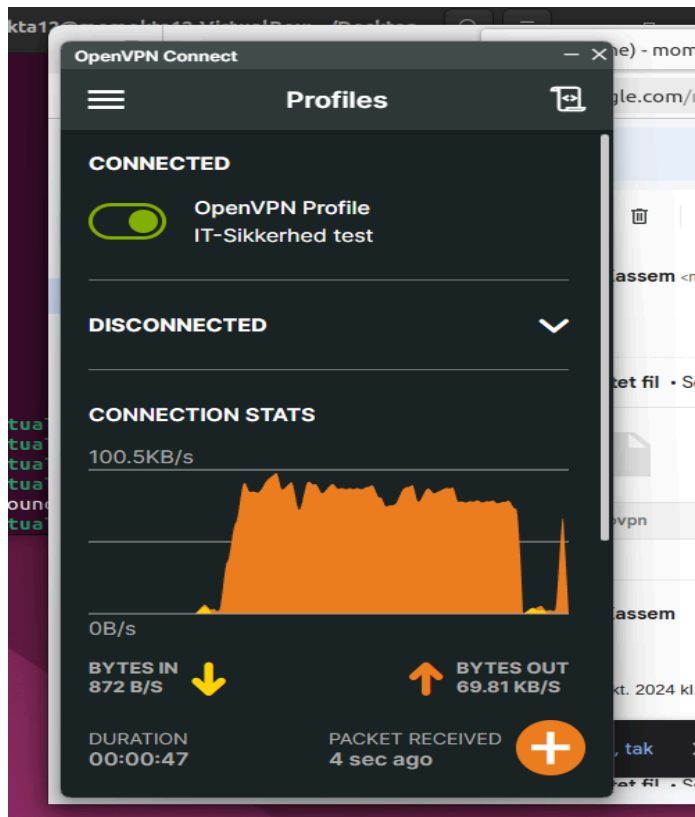
Screenshot: Her ses importen af klient profilen i OpenVPN-klienten:

3.3 Forbindelse til OpenVPN-server

Efter at have importeret klient-konfigurationsfilen, blev OpenVPN-klienten forbundet til serveren. Forbindelsen blev oprettet succesfuldt, og vi kan se statistik for forbindelsen nedenfor:

Screenshot: Her ses den succesfulde klient forbindelse til serveren:

09/10/2024



Teoretiske del:

Q1. Giv en kort beskrivelse af OpenVPN

OpenVPN er en open-source VPN-løsning, som bruger SSL/TLS til at skabe sikre punkt-til-punkt eller site-to-site forbindelser. Det bruger certifikater og nøgler til autentifikation og kryptering for at beskytte data under overførsel.

Eksempel fra vores opgave: Vi konfigurerede OpenVPN til at oprette en sikker forbindelse mellem klient og server. Ved brug af certifikater og nøgler (skabt med EasyRSA) sikrede vi, at forbindelsen kun kunne oprettes af autoriserede brugere.

Q2. Hvordan kan man erhverve/indhente et certifikat?

For at indhente et certifikat, skal en bruger generere en certifikat forespørgsel (CSR - Certificate Signing Request) og sende denne til en CA (Certificate Authority), som vil validere anmodningen og udstede et certifikat.

Eksempel fra vores opgave: Vi genererede en CSR ved hjælp af EasyRSA og signede den med vores egen CA, som vi oprettede i vores PKI-system. Dette gjorde, at både serveren og klienten fik et gyldigt certifikat.

Q3. Hvad bruges et certifikat til?

Et certifikat bruges til at autentificere en bruger eller en enhed overfor en server eller modpart. Det sikrer, at kommunikationen er mellem legitime parter og understøtter krypteret kommunikation.

Eksempel fra vores opgave: I vores opsætning brugte vi certifikater til at sikre, at klienten og serveren kunne stole på hinanden. Klienten skulle præsentere sit certifikat for serveren, og serveren brugte sit CA-signede certifikat til at bevise sin identitet.

Q4. Forklar rollen af Certificate Authority (CA)

En CA udsteder certifikater og sikrer, at de er autentiske. CA'en validerer, at de oplysninger, som er inkluderet i certifikatet, er korrekte og signerer certifikatet som bevis på legitimitet.

Eksempel fra vores opgave: Vi fungerede selv som CA og signede både server- og klientcertifikater. Dette betød, at serveren kunne stole på klientcertifikatet, fordi det var signet af vores CA.

Q5. Beskriv forskellige felter i et X.509-certifikat

De grundlæggende felter i et X.509-certifikat inkluderer:

-
- **Subject:** Ejeren af certifikatet (f.eks. en person eller en server).
 - **Issuer:** Den autoritet, som har udstedt certifikatet (CA).
 - **Public Key:** Den offentlige nøgle, som svarer til en privat nøgle.
 - **Validity Period:** Den periode, hvor certifikatet er gyldigt.
 - **Signature:** CA'ens signatur for at bekræfte certifikatets ægthed.

Eksempel fra vores opgave: Da vi oprettede vores certifikater, definerede vi bl.a. serverens identitet i feltet "Common Name" og satte en udløbsdato.

Q6. Forklar de følgende emner:

1. **OpenVPN Static Key Mode:** Denne metode bruger en præ-shared nøgle til at etablere en krypteret forbindelse. Den er enklere, men mindre sikker, fordi den ikke bruger certifikater.
2. **OpenVPN TLS Mode:** Dette er den mest almindelige tilstand, hvor OpenVPN bruger SSL/TLS til kryptering. Her bruges certifikater til at etablere en sikker forbindelse, som vi gjorde i vores opsætning.
3. **Recent Attacks on SSL/TLS:** Moderne angreb som f.eks. man-in-the-middle, hvor en angriber forsøger at afbryde kommunikationen mellem to parter. For at beskytte mod sådanne angreb brugte vi HMAC-signering i vores opsætning.

Q7. Forklar koncepterne:

1. **Confidentiality:** Data er krypteret og kan kun læses af de autoriserede parter. Vores brug af AES-256 sikrer dette.
2. **Data Integrity:** Sikrer, at data ikke er blevet ændret under overførsel. HMAC-signeringen, vi implementerede, bekræfter dette.
3. **Authentication:** Bekræfter identiteten af parterne i kommunikationen. Vi brugte certifikater til at bekræfte både klienten og serveren.
4. **Non-repudiation:** Garanterer, at en afsender ikke kan benægte at have sendt data. Med vores PKI og signering sikrer vi, at afsenderens identitet er valideret.

-
5. **Availability:** At sikre, at VPN-forbindelsen altid er tilgængelig for autoriserede brugere. Ved korrekt serveropsætning sikrede vi stabilitet og tilgængelighed.

Q8. Forklaring af TLS-handshake

TLS-handshaken består af flere trin:

1. **Client Hello:** Klienten sender en forespørgsel til serveren og foreslår krypteringsalgoritmer.
2. **Server Hello:** Serveren vælger krypteringsmetoden og sender sit certifikat til klienten.
3. **Key Exchange:** Klienten og serveren udveksler nøgler til krypteringen.
4. **Completion:** Når nøglerne er udvekslet, starter krypteret kommunikation.

Eksempel fra vores opgave: Vi så disse trin, da vi brugte TLS til at sikre forbindelsen mellem klient og server. Certifikaterne og nøglerne udveksles, og en sikker session etableres.

Konklusion:

I den teoretiske del har vi gennemgået de grundlæggende koncepter og processer, der understøtter VPN-sikkerhed. Certifikater, PKI, kryptering og TLS-handshaken er alle afgørende for at sikre, at forbindelsen er autentificeret, krypteret og modstandsdygtig over for moderne angreb. Vores praktiske opsætning med OpenVPN illustrerede, hvordan disse teknologier bruges i praksis for at skabe en sikker VPN-forbindelse.

Kildeliste (APA-stil):

1. OpenVPN. (n.d.). *OpenVPN Manual*. Hentet fra <https://openvpn.net>
2. Rescorla, E. (2001). *SSL and TLS: Designing and Building Secure Systems*. Addison-Wesley.
3. Ferreira, P. (2015). *Understanding Public Key Infrastructure (PKI)*. Wiley.
4. Cooper, D. A., Santesson, S., Farrell, S., Boeyen, S., Housley, R., & Polk, W. (2008). *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. IETF. Hentet fra <https://tools.ietf.org/html/rfc5280>

-
5. Oppliger, R. (2016). *SSL and TLS: Theory and Practice, Second Edition*. Artech House.
 6. Diffie, W., & Hellman, M. E. (1976). *New directions in cryptography*. IEEE Transactions on Information Theory, 22(6), 644-654.
 7. Dierks, T., & Rescorla, E. (2008). *The Transport Layer Security (TLS) Protocol Version 1.2*. IETF. Hentet fra <https://tools.ietf.org/html/rfc5246>