

Politechnika Wrocławska, Informatyka Stosowana

Ochrona komunikacji

Cyberbezpieczeństwo, Laboratorium nr.9 - raport

Autor: Aleksander Stepaniuk
Nr. Indeksu: 272644

0. Konfiguracja VPN

Konfiguracja;

Przebieg konfiguracji na zrzutach poniżej:

```
(stud@kali-vm)-[~]
$ cat /etc/openvpn/server/server.conf
port 1194
proto udp
dev tun
ca /etc/openvpn/server/ca.crt
cert /etc/openvpn/server/server.crt
key /etc/openvpn/server/server.key
dh none
server 10.88.88.0 255.255.255.0
push "route 10.88.88.0 255.255.255.0"
keepalive 10 120
tls-crypt /etc/openvpn/ta.key 0
tls-version-min 1.2
cipher AES-256-gcm
auth sha512
tls-cipher TLS-ECDHE-ECDSA-WITH-AES-256-GCM-SHA384:TLS-ECDHE-RSA-WITH-AES-256-GCM-SHA384:TLS-DHE-RSA-WITH-AES-256-GCM-SHA384:TLS-ECDHE-ECDSA-WITH-CHACHA20-POLY1305-SHA256:TLS-ECDHE-RSA-WITH-CHACHA20-POLY1305-SHA256
user nobody
group nogroup
persist-key
persist-tun
reneg-sec 86400
status /var/log/openvpn/openvpn-status.log
```

```
(stud@kali-vm)-[~]
$ ls /etc/openvpn/server
ca.crt  server.conf  server.crt  server.key  ta.key
```

```
(stud@kali-vm)-[~]
$ sudo systemctl status openvpn-server@server.service
● openvpn-server@server.service - OpenVPN service for server
   Loaded: loaded (/usr/lib/systemd/system/openvpn-server@.service; disabled; vendor preset: enabled)
   Active: active (running) since Tue 2024-12-03 15:58:40 CET; 10s ago
     Docs: man:openvpn(8)
           https://openvpn.net/community-resources/reference-manual-for-openvpn-2-4/
           https://community.openvpn.net/openvpn/wiki/HOWTO
   Main PID: 5182 (openvpn)
   Status: "Initialization Sequence Completed"
     Tasks: 1 (limit: 4610)
```

```
student@student-ubuntu: ~  
student@student-ubuntu:~$ ls /home/student/Desktop/VPN/client  
ca.crt client1.crt client1.key client.ovpn ta.key  
student@student-ubuntu:~$ cat /home/student/Desktop/VPN/client/client.ovpn  
client  
dev tun  
proto udp  
remote 192.168.64.10 1194  
ca ca.crt  
cert client1.crt  
key client1.key  
tls-crypt ta.key 1  
#persist-key  
#persist-tun  
verb 1  
cipher AES-256-GCM  
auth SHA512  
remote-cert-tls server  
mssfix 1200  
reneg-sec 0  
#auth-nocache
```

```
GNU nano 7.2 /home/student/Desktop/VPN/client/client.ovpn  
client  
dev tun  
proto udp  
remote 172.16.96.8 1194  
ca /home/student/Desktop/VPN/client/ca.crt  
cert /home/student/Desktop/VPN/client/client2.crt  
key /home/student/Desktop/VPN/client/client2.key  
tls-crypt /home/student/Desktop/VPN/client/ta.key 1  
#persist-key  
#persist-tun  
verb 1  
cipher AES-256-GCM  
auth SHA512  
remote-cert-tls server  
mssfix 1200  
reneg-sec 0  
#auth-nocache
```

```
Dec 03 16:28:36 kali-vm openvpn[5182]: MULTI: primary virtual IP for client  
Dec 03 16:28:36 kali-vm openvpn[5182]: SENT CONTROL [client2]: 'PUSH_REPLY,  
Dec 03 16:28:37 kali-vm openvpn[5182]: client2/172.16.96.7:37258 Data Chann  
Dec 03 16:28:37 kali-vm openvpn[5182]: client2/172.16.96.7:37258 Timers: pi  
Dec 03 16:28:37 kali-vm openvpn[5182]: client2/172.16.96.7:37258 Protocol o  
Dec 03 16:30:51 kali-vm openvpn[5182]: read UDPv4 [ECONNREFUSED]: Connectio
```

```

Dec 03 16:30:55 kali-vm openvpn[5182]: MULTI: new connection by client 'clie>
Dec 03 16:30:55 kali-vm openvpn[5182]: MULTI_sva: pool returned IPv4=10.88.8>
Dec 03 16:30:55 kali-vm openvpn[5182]: MULTI: Learn: 10.88.88.6 → client1/1>
Dec 03 16:30:55 kali-vm openvpn[5182]: MULTI: primary virtual IP for client1>
Dec 03 16:30:55 kali-vm openvpn[5182]: SENT CONTROL [client1]: 'PUSH_REPLY,r>
Dec 03 16:30:56 kali-vm openvpn[5182]: client1/172.16.96.4:56726 Data Channe>
Dec 03 16:30:56 kali-vm openvpn[5182]: client1/172.16.96.4:56726 Timers: pin>
Dec 03 16:30:56 kali-vm openvpn[5182]: client1/172.16.96.4:56726 Protocol op>
lines 447-472/472 (END)

```

1. Zadania i pytania (TLS)

Zadania 1.1-1.3;

Przebieg zadań na zrzutach ekranu poniżej:

The top screenshot shows a Wireshark capture of OpenVPN traffic. The filter is 'ssl.record.version == 0x0303'. The packet list shows several OpenVPN packets of type 'P_DATA_V2'.

The bottom screenshot shows a detailed view of a TLSv1.3 handshake. The filter is 'ssl.record.version == 0x0303 && ssl.handshake'. The packet list shows the following packets:

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|---------------|----------------|----------------|----------|--------|---|
| 166 | 305.900955614 | 34.149.100.209 | 172.16.96.4 | TLSv1.3 | 2974 | Server Hello, Change Cipher Spec |
| 180 | 306.033214856 | 34.117.188.166 | 172.16.96.4 | TLSv1.3 | 2974 | Server Hello, Change Cipher Spec |
| 259 | 306.228365986 | 34.160.144.191 | 172.16.96.4 | TLSv1.2 | 2974 | Server Hello, Certificate |
| 261 | 306.228705015 | 34.160.144.191 | 172.16.96.4 | TLSv1.2 | 139 | Server Key Exchange, Server Hello Done |
| 263 | 306.237523406 | 172.16.96.4 | 34.160.144.191 | TLSv1.2 | 147 | Client Key Exchange, Change Cipher Spec, Encr |
| 264 | 306.246237788 | 34.160.144.191 | 172.16.96.4 | TLSv1.2 | 433 | New Session Ticket, Change Cipher Spec, Encry |
| 344 | 307.016560151 | 34.107.243.93 | 172.16.96.4 | TLSv1.3 | 2974 | Server Hello, Change Cipher Spec |
| 372 | 307.213304194 | 142.251.143.42 | 172.16.96.4 | TLSv1.3 | 2974 | Server Hello, Change Cipher Spec |
| 422 | 307.244519074 | 34.117.121.53 | 172.16.96.4 | TLSv1.3 | 2974 | Server Hello, Change Cipher Spec |
| 433 | 307.252218462 | 34.36.165.17 | 172.16.96.4 | TLSv1.3 | 2974 | Server Hello, Change Cipher Spec |
| 437 | 307.252668804 | 34.36.165.17 | 172.16.96.4 | TLSv1.3 | 2974 | Server Hello, Change Cipher Spec |
| 443 | 307.255117585 | 34.36.165.17 | 172.16.96.4 | TLSv1.3 | 2974 | Server Hello, Change Cipher Spec |
| 447 | 307.257813368 | 34.117.121.53 | 172.16.96.4 | TLSv1.3 | 2974 | Server Hello, Change Cipher Spec |
| 451 | 307.258684095 | 34.117.121.53 | 172.16.96.4 | TLSv1.3 | 2974 | Server Hello, Change Cipher Spec |
| 455 | 307.263574874 | 34.107.243.93 | 172.16.96.4 | TLSv1.3 | 272 | Server Hello, Change Cipher Spec, Application |
| 463 | 307.275375426 | 34.36.165.17 | 172.16.96.4 | TLSv1.3 | 2974 | Server Hello, Change Cipher Spec |

The bottom screenshot shows the details of the selected packet (Frame 166). The details pane shows the following information:

- Frame 166: 2974 bytes on wire (23792 bits), 2974 bytes captured
- Ethernet II, Src: 52:54:00:12:35:00 (52:54:00:12:35:00), Dst: 08:00:27:1c:5b:e7
- Internet Protocol Version 4, Src: 34.149.100.209, Dst: 172.16.96.4
- Transmission Control Protocol, Src Port: 443, Dst Port: 4759
- Transport Layer Security
 - TLSv1.3 Record Layer: Handshake Protocol: Server Hello
 - Content Type: Handshake (22)
 - Version: TLS 1.2 (0x0303)
 - Length: 122
 - Handshake Protocol: Server Hello
 - TLSv1.3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
 - Content Type: Change Cipher Spec (20)
 - Version: TLS 1.2 (0x0303)
 - Length: 1
 - Change Cipher Spec Message
 - TLS segment data (2787 bytes)

Pytanie 1.4;

Oba protokoły działają w głównej mierze tak samo (służą do szyfrowania i uwierzytelniania danych, aby chronić je przed odczytem przez osoby niepożądane), z tą różnicą że TLS jest następcą SSL, a więc wprowadza różnego rodzaju ulepszenia, między innymi w zakresie bezpieczeństwa i wydajności działania protokołu.

Pytanie 1.5;

TLS/SSL handshake to process negocjacji parametrów używanych do nawiązywania połączenia między klientem a serwerem. Podczas tego procesu następuje wymiana szczegółów połączenia takich jak wersja protokołu, klucze sesji czy użyte algorytmy szyfrujące. Jest niezbędnym elementem bezpiecznego przesyłania danych zaszyfowaną drogą.

Pytanie 1.6;

Protokół TLS/SSL zapewnia ochronę transmisji danych poprzez ich szyfrowanie oraz uwierzytelnianie stron komunikacji. Zastosowania:

- Szyfrowanie połączenia między klientem a serwerem podczas przesyłania danych logowania przy rejestracji do wybranego serwisu (np. Netflix)
- Szyfrowanie połączenia między klientem a serwerem podczas prowadzenia połączenia głosowego przez aplikację typu Zoom (ochrona danych przed podsłuchem)
- Uwierzytelnianie stron internetowych HTTPS (np. facebook.com)
- Szyfrowanie e-maili (SMTP)
- Bezpieczne połączenie z serwerem VPN (ochrona prywatności)

Pytanie 1.7;

TLS 1.2 i TLS 1.3. Wersja 1.3 uznawana jest obecnie jako standard i jest powszechnie używana w większości systemów.

Pytanie 1.8;

TLS 1.3, ponieważ eliminuje przestarzałe algorytmy (uwierzytelniania i szyfrowania), redukuje liczbę podatnych funkcji i usprawnia proces handshake.

Pytanie 1.9;

TLS 1.3, ponieważ korzysta z wydajniejszych algorytmów oraz sam proces przebiega szybciej (oraz zużywa mniej energii) ze względu na:

- 1-RTT zamiast 2-RTT
- AES zamiast asymetrycznego RSA
- Krótszy handshake (każda wymiana kluczy korzysta z Diffiego-Hellmana)

Pytanie 1.10;

Wireshark przechwycił obydwie wersje protokołu: TLS 1.2 oraz TLS 1.3

Pytanie 1.11;

Zwykle z: *ClientHello*, *ServerHello*, wymiany certyfikatów i kluczy sesji. Proces ten różni się w zależności od wersji protokołu:

- **TLS 1.2:**
 1. *ClientHello*: Klient wysyła informacje o obsługiwanych algorytmach szyfrowania, wersji protokołu i danych do wygenerowania klucza.
 2. *ServerHello*: Serwer odpowiada wyborem algorytmu i wersji TLS.
 3. *Certificate*: Serwer wysyła swój certyfikat do uwierzytelnienia.
 4. *Key Exchange*: Klient i serwer wymieniają dane pozwalające na ustalenie wspólnego klucza sesji.
 5. *Finished*: Obie strony kończą proces, potwierdzając poprawność ustaleń.

- **TLS 1.3:**

1. *ClientHello*: Klient wysyła propozycję algorytmów szyfrowania i dane do wygenerowania klucza (w tym dane dla szyfrowania asymetrycznego).
2. *ServerHello*: Serwer akceptuje algorytm i generuje klucz sesji na podstawie danych klienta. Certyfikat serwera jest przesyłany w jednym kroku, skracać czas wymiany.
3. *Finished*: Po weryfikacji certyfikatu połączenie jest szyfrowane, co zmniejsza liczbę rund komunikacji.

Pytanie 1.12;

Zależy od wersji oprogramowania zarówno po stronie klienta jak i serwera (wsparcie systemowe oraz konfiguracja oprogramowania mają znaczenie). Z tego powodu warto regularnie aktualizować oprogramowanie aby mieć dostęp do najnowszych, najszybszych i najbezpieczniejszych wersji protokołów takich jak TLS. Proces komunikacji przebiegać będzie po najnowszej wspólnej wersji protokołu uzgodnionej w handshaku (obie strony muszą obsługiwać daną wersję). Dodatkowo administrator sieci może wymusić użycie konkretnej wersji protokołu. Niektóre przeglądarki lub aplikacje pozwalają na wymuszenie używania nowszych wersji. Serwery proxy lub urządzenia pośredniczące mogą wymuszać starsze wersje w przypadku braku wsparcia.

2. Zadania (OpenVPN)

Zadania 2.1-2.4;

Przebieg zadań na zrzutach ekranu poniżej:

Capturing from tun0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|---------------|------------|--------------|----------|--------|---|
| 26 | 122.285720828 | 10.88.88.6 | 91.189.91.97 | TCP | 60 | [TCP Retransmission] 35132 → 80 [SYN] Seq=0 Win=64240 |
| 27 | 124.339624014 | 10.88.88.6 | 91.189.91.97 | TCP | 60 | [TCP Retransmission] 35132 → 80 [SYN] Seq=0 Win=64240 |
| 28 | 128.364615871 | 10.88.88.6 | 91.189.91.97 | TCP | 60 | [TCP Retransmission] 35132 → 80 [SYN] Seq=0 Win=64240 |
| 29 | 135.010599842 | 10.88.88.6 | 10.88.88.10 | TCP | 60 | [TCP Retransmission] 51734 → 7777 [SYN] Seq=0 Win=64240 |
| 30 | 136.432192828 | 10.88.88.6 | 91.189.91.97 | TCP | 60 | [TCP Retransmission] 35132 → 80 [SYN] Seq=0 Win=64240 |
| 31 | 168.311893606 | 10.88.88.6 | 10.88.88.10 | TCP | 60 | [TCP Retransmission] 51734 → 7777 [SYN] Seq=0 Win=64240 |
| 32 | 417.469391643 | 10.88.88.6 | 91.189.91.48 | TCP | 60 | 40014 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK P |
| 33 | 418.525713672 | 10.88.88.6 | 91.189.91.48 | TCP | 60 | [TCP Retransmission] 40014 → 80 [SYN] Seq=0 Win=64240 |
| 34 | 419.585091702 | 10.88.88.6 | 91.189.91.48 | TCP | 60 | [TCP Retransmission] 40014 → 80 [SYN] Seq=0 Win=64240 |
| 35 | 420.673421758 | 10.88.88.6 | 91.189.91.48 | TCP | 60 | [TCP Retransmission] 40014 → 80 [SYN] Seq=0 Win=64240 |
| 36 | 421.793365601 | 10.88.88.6 | 91.189.91.48 | TCP | 60 | [TCP Retransmission] 40014 → 80 [SYN] Seq=0 Win=64240 |
| 37 | 423.615609851 | 10.88.88.6 | 91.189.91.48 | TCP | 60 | [TCP Retransmission] 40014 → 80 [SYN] Seq=0 Win=64240 |
| 38 | 425.793555182 | 10.88.88.6 | 91.189.91.48 | TCP | 60 | [TCP Retransmission] 40014 → 80 [SYN] Seq=0 Win=64240 |
| 39 | 429.813696264 | 10.88.88.6 | 91.189.91.48 | TCP | 60 | [TCP Retransmission] 40014 → 80 [SYN] Seq=0 Win=64240 |
| 40 | 438.412046408 | 10.88.88.6 | 91.189.91.48 | TCP | 60 | [TCP Retransmission] 40014 → 80 [SYN] Seq=0 Win=64240 |

Frame 30: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface tun0
 Raw packet data
 Internet Protocol Version 4, Src: 10.88.88.6, Dst: 91.189.91.97
 Transmission Control Protocol, Src Port: 35132, Dst Port: 80

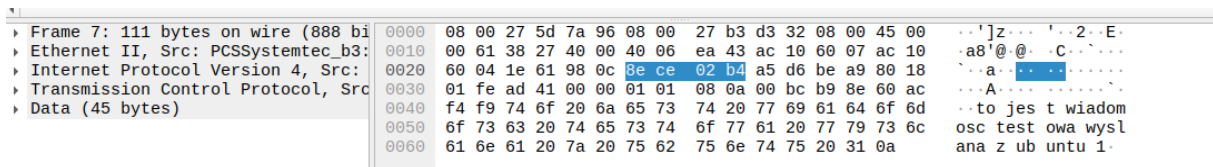
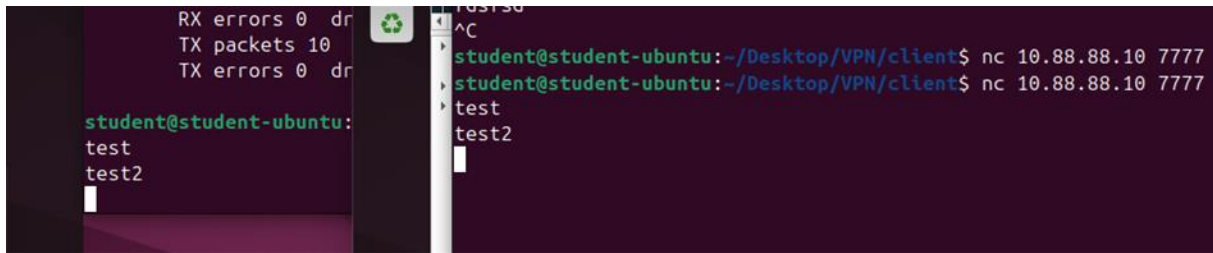
0000 45 00 00 3c b4 24 40 00 40 06 6d 1b 0a 58 58 06 E...\$.m.XX
 0010 5b bd 5b 61 89 3c 00 50 27 9a ce c8 00 00 00 00 [a.<P
 0020 a0 02 fa f0 9c 29 00 00 02 04 05 b4 04 02 08 0a
 0030 da 62 3d 17 00 00 00 00 01 03 03 07 -b=.....

Apply a display filter ... <Ctrl-/>

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-----------------|------------------------|----------------|----------|--------|---------------------|
| 137 | 3030.1990135... | fe80::bbf3:b895:f8c... | ff02::2 | ICMPv6 | 48 | Router Solicitation |
| 1 | 0.000000000 | 10.88.88.6 | 185.125.190.17 | TCP | 60 | 37258 → 80 [|
| 2 | 1.032927497 | 10.88.88.6 | 185.125.190.17 | TCP | 60 | [TCP Retrans |
| 3 | 2.055236545 | 10.88.88.6 | 185.125.190.17 | TCP | 60 | [TCP Retrans |
| 4 | 3.096784114 | 10.88.88.6 | 185.125.190.17 | TCP | 60 | [TCP Retrans |
| 5 | 4.106677923 | 10.88.88.6 | 185.125.190.17 | TCP | 60 | [TCP Retrans |
| 6 | 5.133689440 | 10.88.88.6 | 185.125.190.17 | TCP | 60 | [TCP Retrans |
| 7 | 7.176361899 | 10.88.88.6 | 185.125.190.17 | TCP | 60 | [TCP Retrans |
| 8 | 11.213057389 | 10.88.88.6 | 185.125.190.17 | TCP | 60 | [TCP Retrans |
| 9 | 19.727502036 | 10.88.88.6 | 185.125.190.17 | TCP | 60 | [TCP Retrans |
| 10 | 67.470880540 | 10.88.88.6 | 10.88.88.10 | TCP | 60 | 44750 → 7777 |

Frame 15: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface tun0
 Raw packet data
 Internet Protocol Version 4, Src: 10.88.88.6, Dst: 10.88.88.10
 Transmission Control Protocol, Src Port: 44750, Dst Port: 7777, Seq: 51734

0000 45 00 00 3c 8b
 0010 0a 58 58 0a ae
 0020 a0 02 fa f0 63
 0030 ea b1 5d 7b 00</p>
</div>

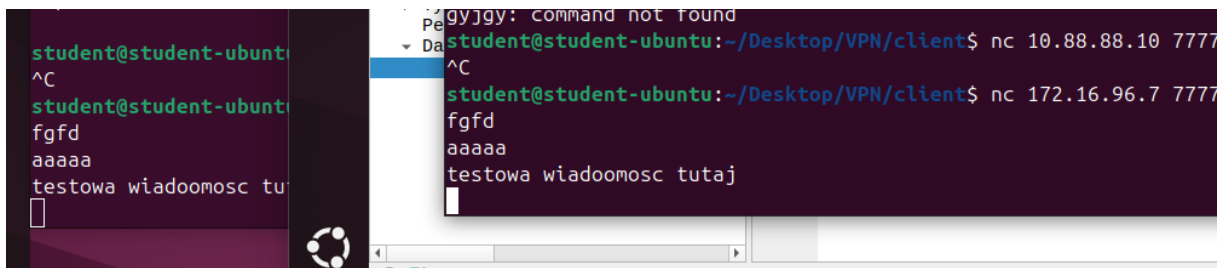
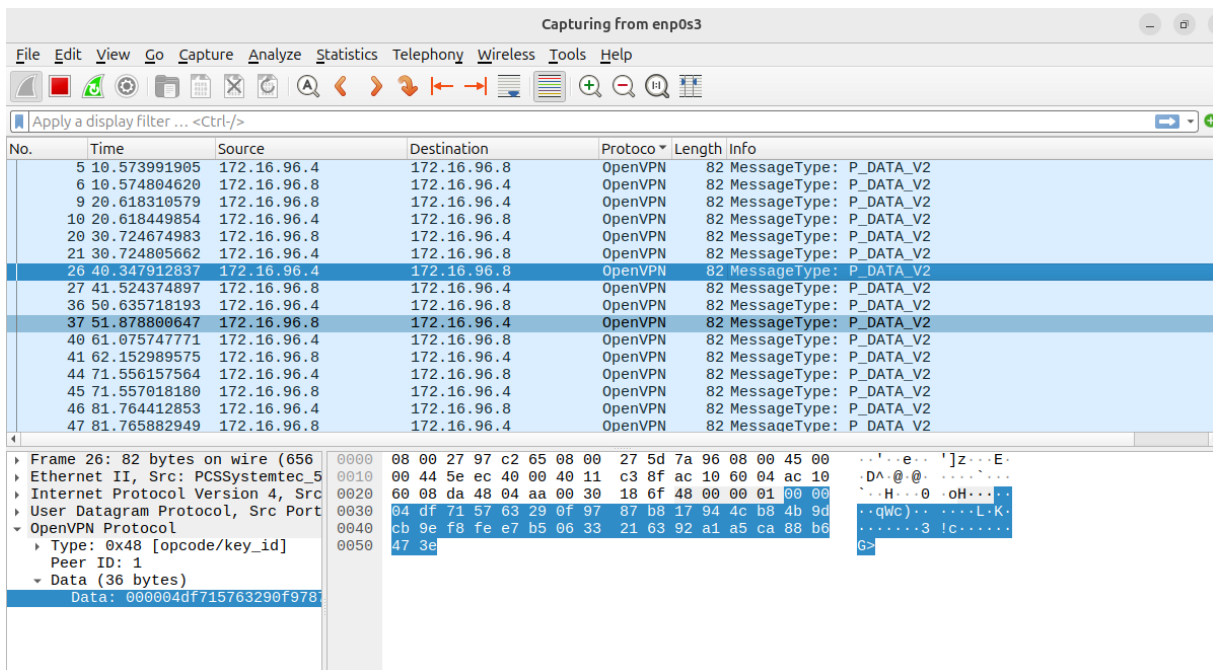


Dla open vpn (tun0) udało się wysłać wiadomość przez netcat, przechwycić pakiety TCP przesyłane między maszynami, dało się odczytać zawartość wiadomości z pakietu

3. Zadania i pytania (OpenVPN)

Zadania 3.1-3.4;

Przebieg zadań na zrzutach ekranu poniżej:



Dla enp0s3 ramki OpenVPN przesyłane pomiędzy klientem a serwerem mają niewidoczny adres IP adresata oraz nie można było odczytać przesyłanej wiadomości w wireshark.

Pytanie 3.5;

Tak, jeśli wiadomość została przesłana w postaci nieszyfrowanej (dla ruchu z interfejsu sieci VPN), można ją odczytać, przechwytyując ruch sieciowy. W przypadku analizy ruchu z interfejsu fizycznego widoczne pakiety OpenVPN przesyłane między klientami miały postać zaszyfrowaną i wymagały klucza dostępu.

Pytanie 3.6;

Dla ruchu na interfejsie hosta widoczny jest zarówno adres źródłowy jak i docelowy (jeśli nie użyto szyfrowania aplikacyjnego, np. TLS), natomiast na interfejsie sieci VPN pakiety posiadały jedynie adres klienta, który wysłał wiadomość oraz adres serwera, a więc odbiorca wiadomości nie mógł w tym przypadku sprawdzić z jakiego adresu wysłano wiadomość.

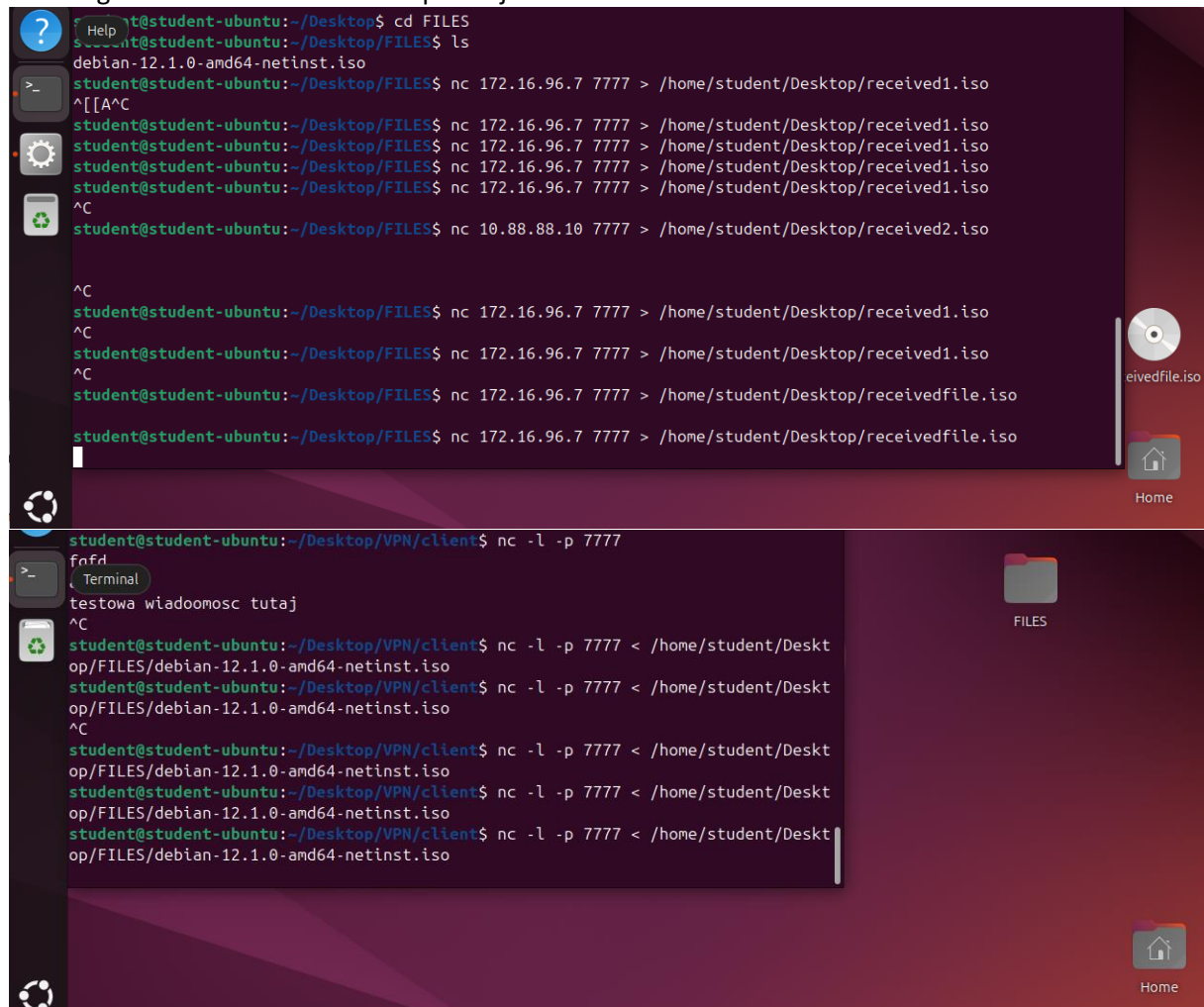
Pytanie 3.7;

Dla interfejsu hosta widać szczegóły transmisji (adresy IP źródłowy i docelowy, porty, dane aplikacyjne). Na interfejsie VPN widoczne są tylko zaszyfrowane pakiety i dane typu metadane (np. adres serwera VPN).

4. Zadania i pytania

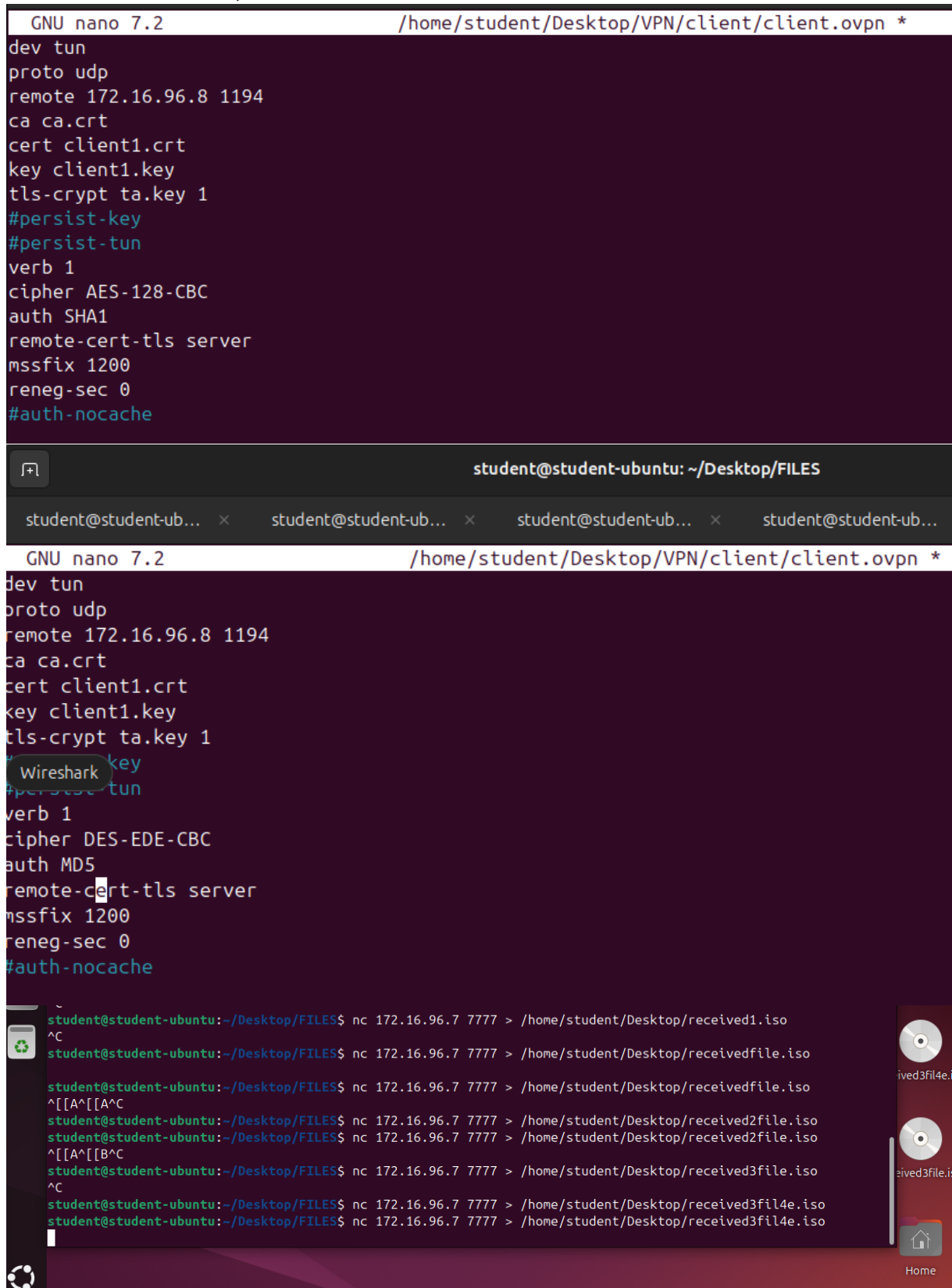
Zadania 4.1-4.2;

Przebieg zadań na zrzutach ekranu poniżej:



Zadanie 4.3;

- Ustawienia 1: 27,73 sekund
- Ustawienia 2: 35,69 sekund
- Ustawienia 3: 42,11 sekund
- Ustawienia 4: 39,19 sekund



```
GNU nano 7.2 /home/student/Desktop/VPN/client/client.ovpn *
dev tun
proto udp
remote 172.16.96.8 1194
ca ca.crt
cert client1.crt
key client1.key
tls-crypt ta.key 1
#persist-key
#persist-tun
verb 1
cipher AES-128-CBC
auth SHA1
remote-cert-tls server
mssfix 1200
reneg-sec 0
#auth-nocache

student@student-ubuntu: ~/Desktop/FILES

student@student-ub... x student@student-ub... x student@student-ub... x student@student-ub...

GNU nano 7.2 /home/student/Desktop/VPN/client/client.ovpn *
dev tun
proto udp
remote 172.16.96.8 1194
ca ca.crt
cert client1.crt
key client1.key
tls-crypt ta.key 1
#persist-key
#persist-tun
verb 1
cipher DES-EDE-CBC
auth MD5
remote-cert-tls server
mssfix 1200
reneg-sec 0
#auth-nocache

student@student-ubuntu:~/Desktop/FILES$ nc 172.16.96.7 7777 > /home/student/Desktop/received1.iso
^C
student@student-ubuntu:~/Desktop/FILES$ nc 172.16.96.7 7777 > /home/student/Desktop/receivedfile.iso
student@student-ubuntu:~/Desktop/FILES$ nc 172.16.96.7 7777 > /home/student/Desktop/receivedfile.iso
^[[A^[[A^C
student@student-ubuntu:~/Desktop/FILES$ nc 172.16.96.7 7777 > /home/student/Desktop/received2file.iso
student@student-ubuntu:~/Desktop/FILES$ nc 172.16.96.7 7777 > /home/student/Desktop/received2file.iso
^[[A^[[B^C
student@student-ubuntu:~/Desktop/FILES$ nc 172.16.96.7 7777 > /home/student/Desktop/received3file.iso
^C
student@student-ubuntu:~/Desktop/FILES$ nc 172.16.96.7 7777 > /home/student/Desktop/received3fil4e.iso
student@student-ubuntu:~/Desktop/FILES$ nc 172.16.96.7 7777 > /home/student/Desktop/received3fil4e.iso
```

Pytanie 4.4;

Tak, szyfrowanie i dodatkowe trasy przez serwer VPN zwiększają opóźnienia przy przesyłaniu pliku, jednak sama różnica jest raczej na tyle niewielka, że można założyć, że wybór algorytmu nie ma większego znaczenia przy czasie przesyłania pliku. Te różnice mogą być spowodowane wieloma innymi czynnikami (jak to, że im dłużej pracuje się na wirtualnych maszynach tym bardziej zaczyna im brakować ramy i po godzinie pracy potrafią się zacinać przy 3 wirtualnych środowiskach (działają wolniej)).

Pytanie 4.5;

Algorytmy silne (np. AES-256-GCM) zapewniają lepsze bezpieczeństwo, ale mogą być wolniejsze w użytku. DES-EDE-CBC jest uznawany za bezpieczny, jednak niezalecany, bo zaleca się korzystanie z lepszego w każdym względzie AESa. Algorytm MD5 jest zdecydowanie przestarzały i nie gwarantuje współcześnie żadnego bezpieczeństwa w związku z tym nie zaleca się korzystanie z niego.

Pytanie 4.6;

Za najgorszy zestaw można uznać ten z algorytmem DES-CBC oraz MD5, ponieważ nie korzysta się z tych algorytmów we współczesnej kryptografii (posiadają wady pozwalające je prościej złamać) i zalecane jest zastąpienie je przez współczesne metody takie jak AES czy SHA256. Z kolei za najbezpieczniejszy zestaw można z kolei uznać użycie AES-256-GCM oraz SHA512, ponieważ długości klucza są długie i zapewniają bezpieczne szyfrowanie danych.