Politechnika Białostocka

Sprawozdanie z pracowni specjalistycznej

Bezpieczeństwo Sieci Komputerowych

Ćwiczenie numer: 7

Temat: **Konfiguracja VPN**

Wykonujący ćwiczenie:

Paweł Orzel

Łukasz Hossa

Kacper Seweryn

Studia dzienne

Kierunek: Informatyka

Semestr: VI

Grupa zajęciowa: Grupa PS 10

Prowadzący ćwiczenie: mgr inż. Katarzyna Borowska

Data wykonania ćwiczenia:25.05.2022

1. Teoria

VPN jest tworzony poprzez uprzednie uwierzytelnienie klienta – komputera, smartfona lub tabletu – przez serwer VPN. Następnie serwer wykorzystuje jeden z szeregu różnych protokołów szyfrowania, aby upewnić się, że nikt nie może monitorować informacji przepływających między Tobą a Twoim celem online.

W tym miejscu należy pamiętać, że przed wysłaniem i odebraniem przez internet wszelkie dane muszą być najpierw podzielone na pakiety. Aby zapewnić bezpieczeństwo każdego pakietu danych, usługa VPN umieszcza go w zewnętrznym pakiecie, który jest następnie szyfrowany poprzez proces zwany enkapsulacją.

Ten zewnętrzny pakiet zapewnia bezpieczeństwo danych podczas przesyłania i jest podstawowym elementem tunelu VPN. Gdy dane docierają do serwera VPN, pakiet zewnętrzny jest usuwany, aby uzyskać dostęp do danych w jego obrębie, co wymaga procesu deszyfrowania.

Po ustanowieniu tunelu VPN urządzenie wysyła zaszyfrowane informacje (takie jak strona internetowa, którą chcesz odwiedzić) na serwer VPN. Odszyfrowuje ją i przekazuje informacje do wyznaczonego serwera WWW. Ukrywa również Twój prawdziwy adres IP przed wysłaniem danych. Zamiast tego będziesz mieć adres IP serwera VPN, z którym jesteś połączony.

Gdy serwer WWW odpowiada, serwer VPN szyfruje dane i wysyła je do Ciebie za pośrednictwem dostawcy usług internetowych. Twój klient VPN odszyfruje dane, gdy dotrą do Twojego urządzenia.

2. Realizacja

1) Instalacja OpenVPN na serwerze

```
student@student-VirtualBox:~$ sudo apt install
[sudo] password for student:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfwupdplugin1
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 42 not upgraded.
student@student-VirtualBox:~$ sudo apt install openvpn
Reading package lists... Done
Building dependency tree
Reading state information... Done
openvpn is already the newest version (2.4.7-1ubuntu2.20.04.4).
The following package was automatically installed and is no longer required:
  libfwupdplugin1
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 42 not upgraded.
student@student-VirtualBox:~$
```

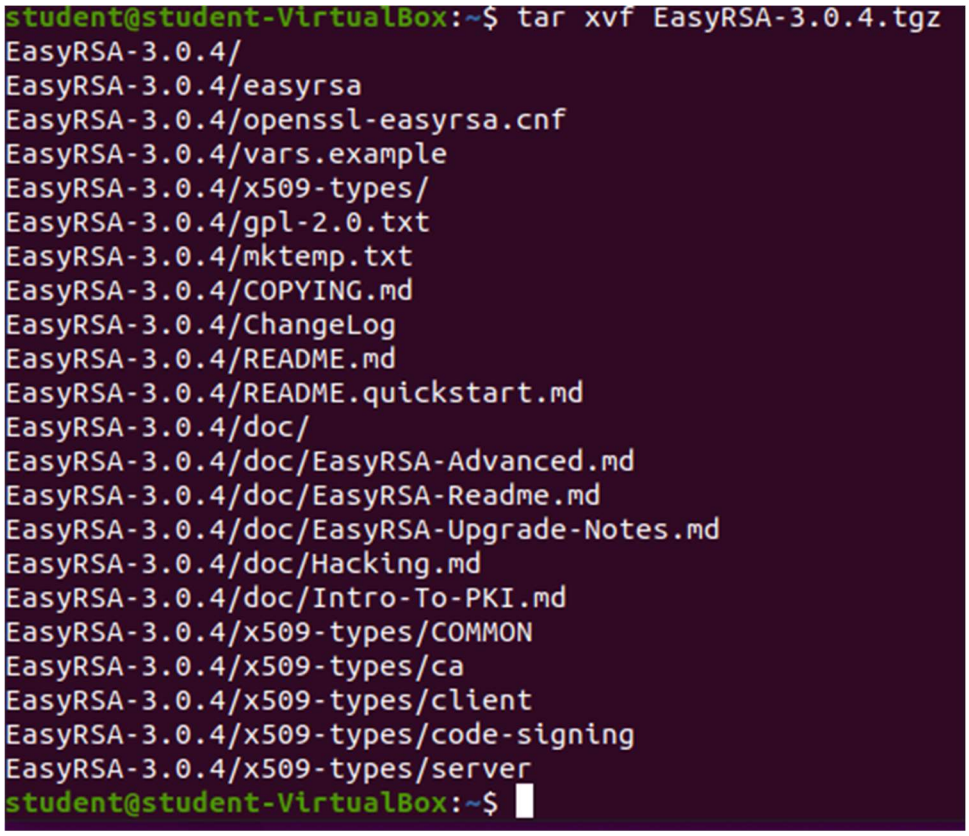
OpenVPN jest to pakiet oprogramowania wdraża techniki tworzenia bezpiecznych połączeń

2) Instalacja EasyRSA na obu maszynach hosta oraz CA

EasyRSA to narzędzie służące do tworzenia głównego urzędu certyfikacji, żądań oraz podpisywania certyfikatów

```
student@student-VirtualBox:~$ tar xvf EasyRSA-3.0.4.tgz
EasyRSA-3.0.4/
EasyRSA-3.0.4/easyrsa
EasyRSA-3.0.4/openssl-easyrsa.cnf
EasyRSA-3.0.4/vars.example
EasyRSA-3.0.4/x509-types/
EasyRSA-3.0.4/gpl-2.0.txt
EasyRSA-3.0.4/mktemp.txt
EasyRSA-3.0.4/COPYING.md
EasyRSA-3.0.4/ChangeLog
EasyRSA-3.0.4/README.md
EasyRSA-3.0.4/README.quickstart.md
EasyRSA-3.0.4/doc/
EasyRSA-3.0.4/doc/EasyRSA-Advanced.md
EasyRSA-3.0.4/doc/EasyRSA-Readme.md
EasyRSA-3.0.4/doc/EasyRSA-Upgrade-Notes.md
EasyRSA-3.0.4/doc/Hacking.md
EasyRSA-3.0.4/doc/Intro-To-PKI.md
EasyRSA-3.0.4/x509-types/COMMON
EasyRSA-3.0.4/x509-types/ca
EasyRSA-3.0.4/x509-types/client
EasyRSA-3.0.4/x509-types/code-signing
EasyRSA-3.0.4/x509-types/server
student@student-VirtualBox:~$
```

Instalacja na maszynie CA

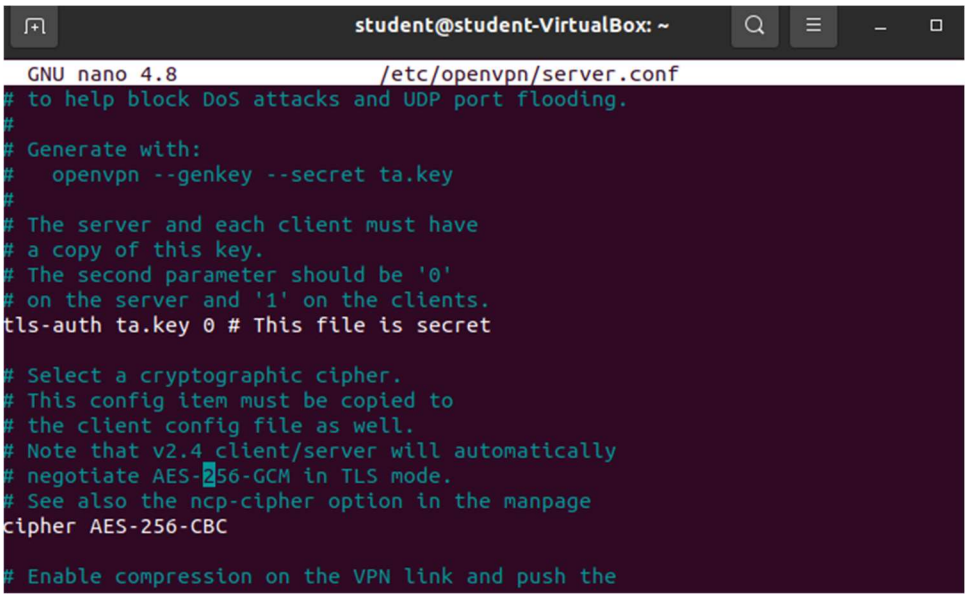


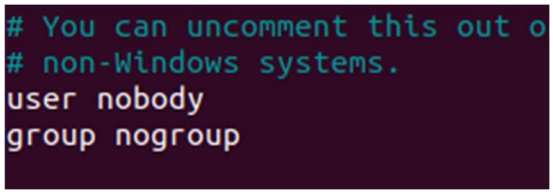
Instalacja na maszynie hosta

3) Konfiguracja OpenVPN w ramach serwera

Odkomentowywując tls-auth zapewniamy dodatkowy poziom bezpieczeństwa wykraczający poza SSL/TLS

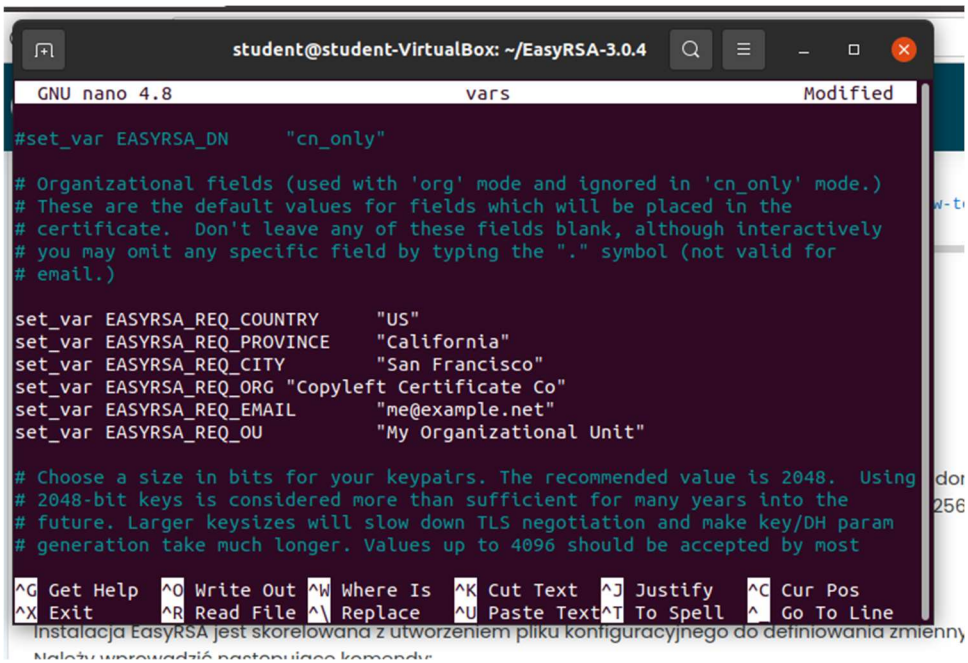
Usuwamy 2048 z sekcji dh, aby dopasować liczbę pierwszą DH do długości klucza RSA

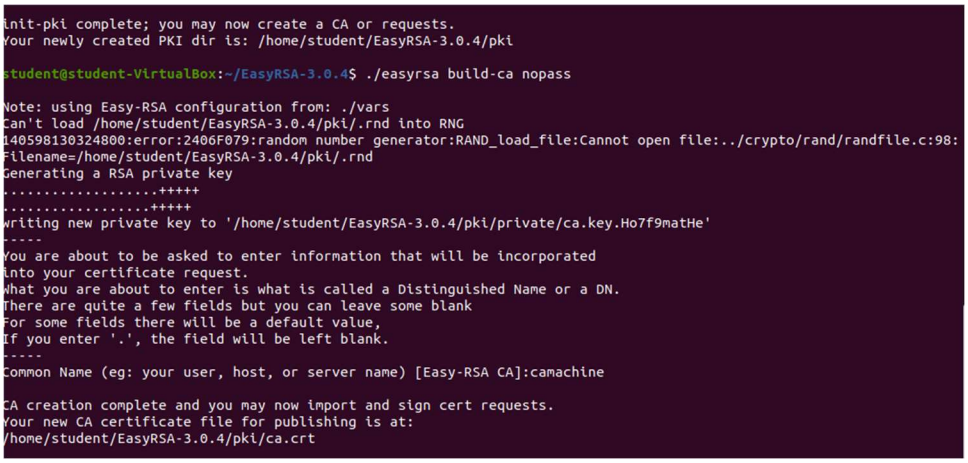




4) Konfiguracja EasyRSA na maszynie CA

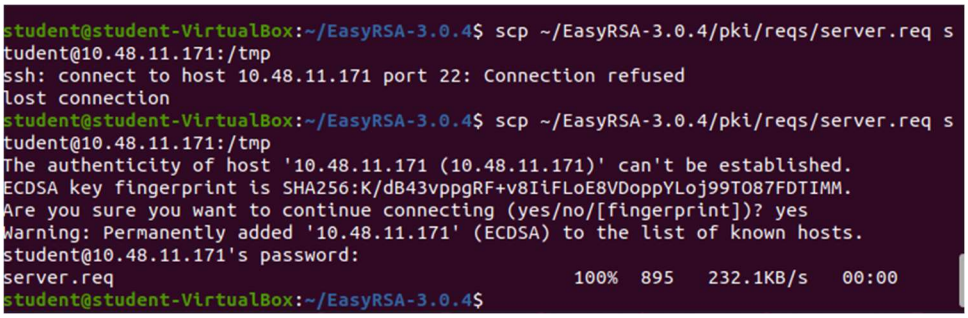
Instalacja EasyRSA jest skorelowana z utworzeniem pliku konfiguracyjnego do definiowania zmiennych, które zostaną ujęte w ramach certyfikatów. Następnie w pliku vars, odkomentować wszystkie zmienne odpowiedzialne za elementy wymagane. W ostatnim kroku należy uruchomić skrypt "easyrsa" aby zainicjować infrastrukturę klucza publicznego. Spowoduje to wygenerowanie dwóch plików (ca.crt - publiczny certyfikat CA poświadczający przynależność do zaufanej sieci; ca.key - prywatny klucz maszyny certyfikującej wykorzystywany do podpisywania kluczy oraz certyfikatów dla serwerów i klientów)





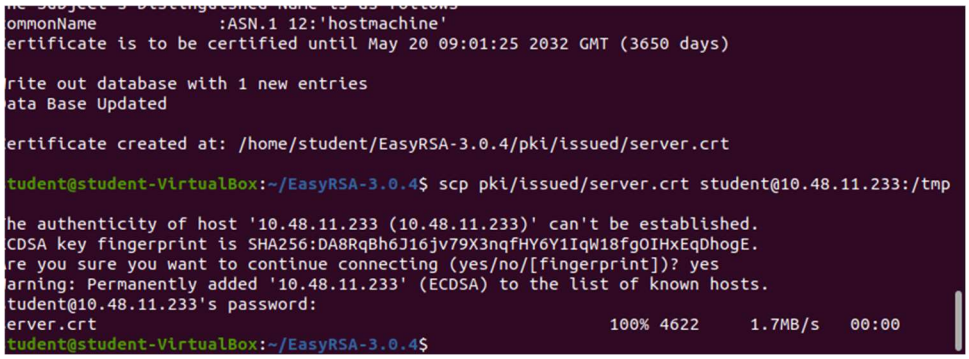
5) Żądanie certyfikatu z maszyny CA

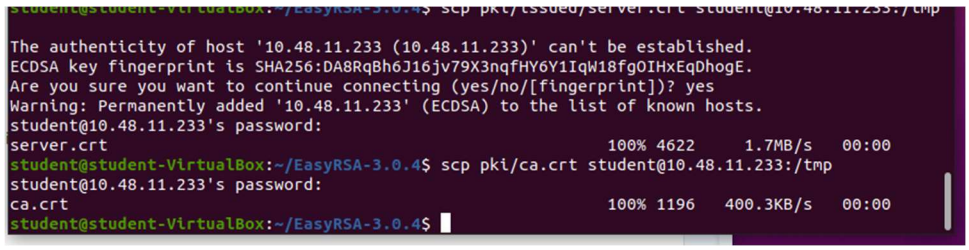
Generujemy klucz prywatny oraz żądanie certyfikatu na serwerze, a następnie przesyłamy je na maszynę certyfikującą w celu podpisania.

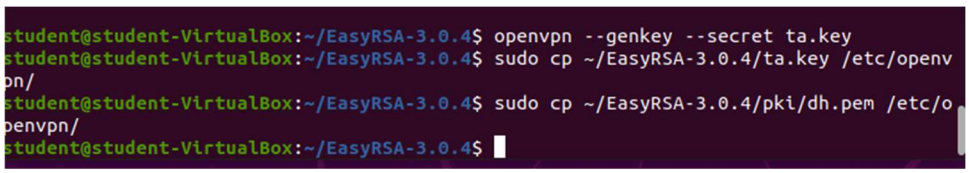


6) Generowanie i podpisywanie certyfikatu

Po stronie maszyny CA importujemy przekopiowany plik do folderu EasyRSA i go podpisujemy, a następnie przesyłamy podpisany certyfikat na serwer VPN. Następnie po stronie serwera kopiujemy pliki do odpowiednich lokalizacji. Na koniec generujemy wymianę kluczy bazując na algorytmie Diffie-Hellman. Algorytm pozwala bezpiecznie uzgodnić klucz nawet jeżeli istnieje osoba, która podsłuchuje proces uzgadniania klucza, nie chroni jednak przed atakami typu man in the middle. Algorytm nie nadaje się do szyfrowania i deszyfrowania wiadomości.







7) Konfiguracja maszyny klienta

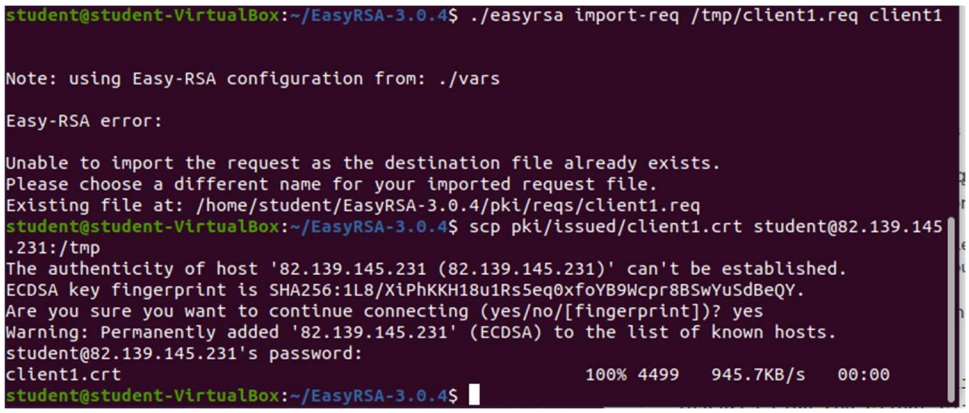
Po stronie klienta tworzymy folder do przechowywania certyfikatów oraz kluczy.

```
student@student-VirtualBox:~/EasyRSA-3.0.4$ ./easyrsa gen-req client1 nopass
Can't load /home/student/EasyRSA-3.0.4/pki/.rnd into RNG
140676262704448:error:2406F079:random number generator:RAND_load_file:Cannot op
en file:../crypto/rand/randfile.c:98:Filename=/home/student/EasyRSA-3.0.4/pki/.
rnd
Generating a RSA private key
.....................................................................
.....................................................+++++
.........................+++++
writing new private key to '/home/student/EasyRSA-3.0.4/pki/private/client1.key
.2Xp3ZTuZH8'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [client1]:clientmachine

Keypair and certificate request completed. Your files are:
req: /home/student/EasyRSA-3.0.4/pki/reqs/client1.req
key: /home/student/EasyRSA-3.0.4/pki/private/client1.key
```

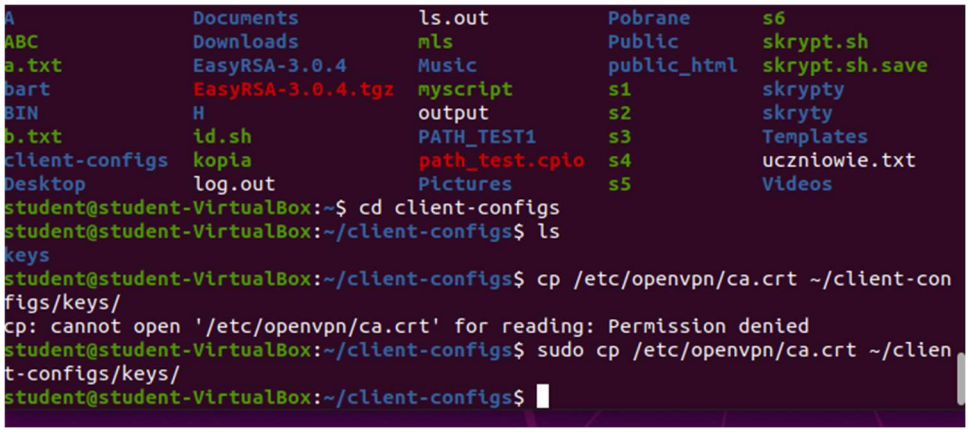
```
student@student-VirtualBox:~/EasyRSA-3.0.4$ cp pki/private/client1.key ~/client
-configs/keys/
student@student-VirtualBox:~/EasyRSA-3.0.4$ scp pki/reqs/client1.req student@10
.48.11.171:/tmp
The authenticity of host '10.48.11.171 (10.48.11.171)' can't be established.
ECDSA key fingerprint is SHA256:K/dB43vppgRF+v8IiFLoE8VDoppYLoj99TO87FDTIMM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.48.11.171' (ECDSA) to the list of known hosts.
student@10.48.11.171's password:
client1.req                                   100%  895   162.3KB/s   00:00
student@student-VirtualBox:~/EasyRSA-3.0.4$
```

Kopiujemy klucz klienta do utworzonego wcześniej folderu i przesyłamy plik z rozszerzeniem req na maszynę CA.

```
Request subject, to be signed as a client certificate for 3650 days:

subject=
    commonName                = clientmachine


Type the word 'yes' to continue, or any other input to abort.
  Confirm request details: yes
Using configuration from ./openssl-easyrsa.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'clientmachine'
Certificate is to be certified until May 20 09:27:07 2032 GMT (3650 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /home/student/EasyRSA-3.0.4/pki/issued/client1.crt
```

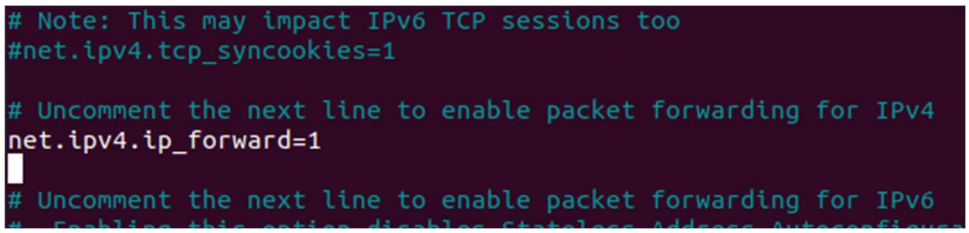


Importujemy żądani certyfikatu CA i wykonujemy autoryzację podpisu.



Na koniec kopiujemy odpowiednie pliki do odpowiadającym im folderów na serwer.

8) Konfiguracja routingu IP oraz firewall



Ustawiamy forwardowanie IP ustawiając 1, forwardowanie byłoby wyłączone dla wartości 0.

```
#
# rules.before
#
# Rules that should be run before the ufw command line added rules. Custom
# rules should be added to one of these chains:
#   ufw-before-input
#   ufw-before-output
#   ufw-before-forward
#
# START OPENVPN RULES

# NAT table rules

*nat

:POSTROUTING ACCEPT [0:0]&amp;nbsp;

# Allow traffic from OpenVPN client to eth0&amp;nbsp;

-A POSTROUTING -s 10.8.0.0/8 -o eth0 -j MASQUERADE
                              [ Read 89 lines ]
```

```
# you change this you will most likely want to adjust your rules.
DEFAULT_OUTPUT_POLICY="ACCEPT"

# Set the default forward policy to ACCEPT, DROP or REJECT.  Please note that
# if you change this you will most likely want to adjust your rules
DEFAULT_FORWARD_POLICY="ACCEPT"

# Set the default application policy to ACCEPT, DROP, REJECT or SKIP. Please
```

Dodając port 1194 do ruchu UDP udostępniamy mechanizm identyfikacji różnych punktów końcowych na jednym hoście dzięki portom. UDP zajmuje się dostarczaniem pojedynczych pakietów, udostępnionych przez IP, na którym się opiera.

```
Rules updated
Rules updated (v6)
student@student-VirtualBox:~/client-configs$ sudo ufw allow 1194/udp $ sudo ufw
allow OpenSSH
ERROR: Need 'to' or 'from' clause
student@student-VirtualBox:~/client-configs$ sudo ufw allow 1194/udp
Rules updated
Rules updated (v6)
student@student-VirtualBox:~/client-configs$ sudo ufw allow OpenSSH
Rules updated
Rules updated (v6)
student@student-VirtualBox:~/client-configs$
```

```
Rules updated
Rules updated (v6)
student@student-VirtualBox:~/client-configs$ sudo ufw disable
Firewall stopped and disabled on system startup
student@student-VirtualBox:~/client-configs$ sudo ufw enable
Firewall is active and enabled on system startup
student@student-VirtualBox:~/client-configs$ sudo systemctl start openvpn
student@student-VirtualBox:~/client-configs$ sudo systemctl status openvpn
● openvpn.service - OpenVPN service
     Loaded: loaded (/lib/systemd/system/openvpn.service; enabled; vendor prese>
     Active: active (exited) since Wed 2022-05-25 10:18:59 CEST; 53min ago
    Process: 730 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 730 (code=exited, status=0/SUCCESS)

maj 25 10:18:59 student-VirtualBox systemd[1]: Starting OpenVPN service...
maj 25 10:18:59 student-VirtualBox systemd[1]: Finished OpenVPN service.
lines 1-8/8 (END)
```

Po zresetowaniu firewalla, uruchomieniu usługi vpn, sprawdzamy status usługi vpn

```
# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
remote my-server-1 1194
;remote my-server-2 1194
```

```
# EasyRSA can do this for you.
remote-cert-tls server

# If a tls-auth key is used on the server
# then every client must also have the key.
tls-auth ta.key 1

# Select a cryptographic cipher.
# If the cipher option is used on the server
# then you must also specify it here.
# Note that v2.4 client/server will automatically
# negotiate AES-256-GCM in TLS mode.
# See also the ncp-cipher option in the manpage
cipher AES-256-CBC
key-direction 1
script-security 2

up /etc/openvpn/update-resolv-conf

down /etc/openvpn/update-resolv-conf
# Enable compression on the VPN link.
# Don't enable this unless it is also
# enabled in the server config file.
#comp-lzo

# Set log file verbosity.
verb 3

# Silence repeating messages
;mute 20
```

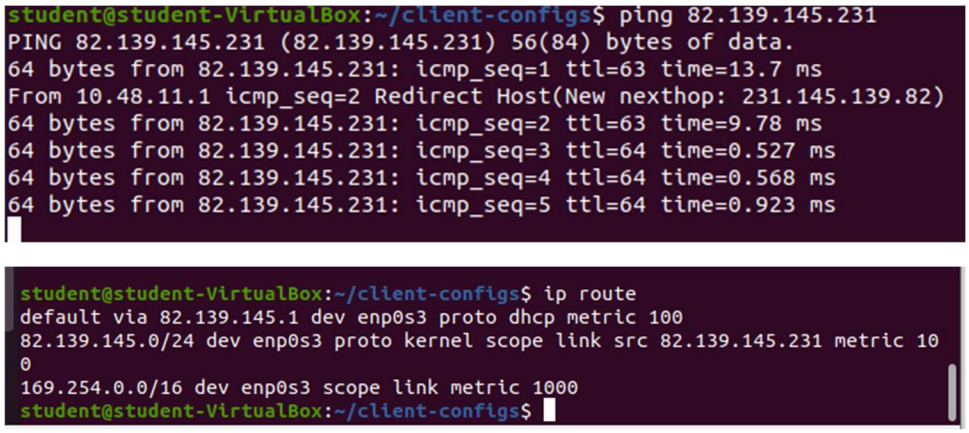
9) Generowanie konfiguracji dla klientów

```
student@student-VirtualBox:~/client-configs$ nano ~/client-configs/make_config.sh
```

Stworzyliśmy skrypt do automatycznej kompilacji konfiguracji klienta zgodnie z wytycznymi na platformie cez2

10) Przekierowanie całego ruchu DNS przez VPN

```
# in order for this to work properly).
push "redirect-gateway def1 bypass-dhcp"

# Certain Windows-specific network settings
```



Wykonanie komendy *ping* oraz *ip route*

3. Wnioski

- W zadaniu 9 wystąpił problem z wygenerowaniem konfiguracji dla klienta. Próba stworzenia pliku clientmachine.ovpn kończyła się niepowodzeniem co przeszkodziło w osiągnieciu poprawnych wyników na końcu zadania. Reszta zadania przebiegła pomyślnie.
- Instrukcja dotycząca zadania była przejrzysta i bardzo pomogła w realizacji.
- Na każdej maszynie wymagany był zainstalowany serwer SSH oraz OpenVPN. Wymagana była także instalacja pakietu EasyRSA do generowania certyfikatów.
- Skonfigurowany został także routing oraz firewall
- VPN jest szeroko stosowanym rozwiązaniem na świecie
- VPN zapewnił ukrycie adresu IP co pozwala klientowi zachować anonimowość w sieci.