WOJSKOWA AKADEMIA TECHNICZNA

im. Jarosława Dąbrowskiego

WYDZIAŁ CYBERNETYKI



Temat pracy: KONFIGURACJA BEZPIECZNEGO POŁĄCZENIA VPN Z WYKORZYSTANIEM OPENVPN

	KRYPTOLOGIA I CYBERBEZPIECZEŃSTWO
	(kierunek studiów)
	CYBEROBRONA
	(specjalność)
Autor:	
Łukasz I	ROZBICKI

Warszawa 2025

Spis treści

Rozdział I. Podstawowe informacje o projekcie	3
I.1. Wstęp	3
I.2. Założenia projektu	3
I.3. Cel projektu	4
I.4. Zakres prac	4
I.5. Oczekiwane wyniki	4
Rozdział II. Teoretyczne podstawy sieci VPN i kryptografii	5
II.1. Definicja i działanie sieci VPN	5
II.2. Protokoły i technologie szyfrowania stosowane w VPN	5
II.3. Bezpieczeństwo i zagrożenia w sieciach VPN	6
II.4. Infrastruktura klucza publicznego (PKI)	6
II.5. Zastosowania praktyczne VPN	7
Rozdział III. Instalacja i konfiguracja środowiska VPN	8
III.1. Przygotowanie środowiska serwera	8
III.2. Instalacja i konfiguracja OpenVPN	21
III.3. Konfiguracja serwera OpenVPN	28
III.4. Konfiguracja zapory sieciowej i routingu	32
III.5. Konfiguracja klienta OpenVPN	35
III.6. Kopiowanie plików konfiguracyjnych i zabezpieczenia	39
Rozdział IV Podsumowanie	41

Rozdział I. Podstawowe informacje o projekcie

I.1. Wstęp

W dobie dynamicznego rozwoju technologii informatycznych, coraz większą rolę odgrywa ochrona danych przesyłanych w sieciach publicznych i prywatnych. Jednym z najczęściej stosowanych rozwiązań jest wykorzystanie wirtualnych sieci prywatnych (VPN), które umożliwiają bezpieczne tunelowanie ruchu sieciowego oraz zapewnienie poufności i integralności danych. Projekt ten ma na celu skonfigurowanie środowiska VPN z wykorzystaniem OpenVPN na serwerze Ubuntu 20.04 oraz klienta Windows, co umożliwi bezpieczne przesyłanie informacji między zdalnymi lokalizacjami.

I.2. Założenia projektu

- 1. Platforma serwerowa i kliencka:
 - Serwer: Ubuntu Server 20.04 LTS skonfigurowany jako host wirtualnej maszyny.
 - Klient: system Windows z zainstalowanym oprogramowaniem OpenVPN.
- 2. Technologie i narzędzia:
 - OpenVPN jako główne oprogramowanie do zarządzania siecią VPN.
 - Easy-RSA do generowania kluczy kryptograficznych i certyfikatów.
 - Protokóły szyfrowania SSL/TLS do zabezpieczania połączeń.

3. Kroki implementacji:

- Instalacja i konfiguracja serwera Ubuntu oraz OpenVPN.
- Utworzenie infrastruktury kluczy publicznych (PKI) oraz generacja certyfikatów dla serwera i klienta.
- Konfiguracja zapory sieciowej i przekierowanie ruchu VPN.
- Weryfikacja poprawności konfiguracji i testowanie połączenia między serwerem a klientem.

4. Bezpieczeństwo:

- Użycie szyfrowania AES-256-GCM oraz funkcji skrótu SHA-256 w celu zapewnienia integralności danych.
- Ograniczenie dostępu do kluczy i certyfikatów poprzez odpowiednie uprawnienia plików.

I.3. Cel projektu

Głównym celem projektu jest praktyczne zastosowanie kryptografii w celu zapewnienia bezpiecznej komunikacji sieciowej. Wdrożenie rozwiązania VPN umożliwi:

- Zabezpieczenie przesyłanych danych przed nieautoryzowanym dostępem.
- Ochronę tożsamości użytkowników i ich aktywności w sieci.
- Demonstrację poprawnej konfiguracji systemu OpenVPN oraz możliwości jego zastosowania w środowiskach firmowych i domowych.

I.4. Zakres prac

Projekt obejmuje następujące etapy:

- 1. Instalacja serwera Ubuntu i niezbędnych narzędzi.
- 2. Konfiguracja infrastruktury kluczy publicznych (PKI) przy użyciu Easy-RSA.
- 3. Generacja certyfikatów dla serwera i klienta.
- 4. Konfiguracja plików konfiguracyjnych OpenVPN na serwerze i kliencie.
- 5. Testy połączeń oraz analiza dzienników logów w celu weryfikacji poprawności działania systemu.

I.5. Oczekiwane wyniki

Po zakończeniu projektu oczekuje się:

- Stabilnego połączenia VPN między serwerem a klientem z poprawnie uwierzytelnionymi certyfikatami.
- Przejrzystej dokumentacji zawierającej opisy kroków konfiguracji oraz wyniki testów.

Projekt ten stanowi podstawę do zrozumienia zaawansowanych aspektów bezpiecznych połączeń sieciowych oraz może być rozwinięty w przyszłości o dodatkowe funkcje, takie jak monitorowanie połączeń czy zarządzanie wieloma klientami w ramach jednego serwera VPN.

Rozdział II. Teoretyczne podstawy sieci VPN i kryptografii

II.1. Definicja i działanie sieci VPN

Wirtualna sieć prywatna (VPN) to technologia, która pozwala na bezpieczne przesyłanie danych przez publiczne sieci, tak jakby urządzenia były bezpośrednio podłączone do prywatnej sieci. Głównym celem VPN jest zapewnienie poufności, integralności i uwierzytelnienia przesyłanych danych. Użytkownicy, korzystając z tej technologii, mogą bezpiecznie łączyć się z sieciami prywatnymi, chronić swoje dane przed podsłuchiwaniem oraz ominać ograniczenia geograficzne.

Mechanizmy tunelowania danych: Tunelowanie to technika polegająca na enkapsulacji pakietów danych w bezpiecznym tunelu tworzonym przez protokoł VPN. Pakiety te są szyfrowane przed przesłaniem, co uniemożliwia ich przechwycenie przez osoby nieupoważnione. Proces tunelowania odbywa się za pomocą protokołów, takich jak IPsec czy SSL/TLS.

Rodzaje sieci VPN:

- VPN Site-to-Site: Łączy całe sieci lokalne znajdujące się w różnych lokalizacjach.
 Jest stosowany głównie w firmach, gdzie biura w różnych miejscach mogą dzielić zasoby.
- VPN Remote Access: Umożliwia indywidualnym użytkownikom łączenie się z prywatną siecią za pomocą internetu, co jest przydatne dla pracowników zdalnych lub w sytuacjach, gdy konieczny jest dostęp do zasobów sieciowych spoza biura.

II.2. Protokoły i technologie szyfrowania stosowane w VPN

Protokoły VPN:

- SSL/TLS: Zapewnia bezpieczne połączenia poprzez szyfrowanie danych na poziomie aplikacji. Jest stosowany w połączeniach HTTPS oraz w niektórych konfiguracjach VPN.
- IPsec: Działa na poziomie warstwy sieciowej i oferuje kompleksowe zabezpieczenie poprzez szyfrowanie i uwierzytelnianie każdego pakietu danych. Jest często używany w sieciach VPN typu Site-to-Site.
- OpenVPN: Otwarta implementacja VPN wykorzystująca protokoły SSL/TLS do bezpiecznego tunelowania danych.

Typy szyfrowania:

- AES-256: Zaawansowany standard szyfrowania blokowego, zapewnia wysoki poziom bezpieczeństwa.
- HMAC: Mechanizm uwierzytelniania wiadomości, który gwarantuje integralność danych i chroni przed ich modyfikacją.

• SHA-256: Funkcja skrótu stosowana do zapewnienia integralności danych i uwierzytelniania przesyłanych informacji.

Rola certyfikatów i kluczy kryptograficznych: Certyfikaty służą do uwierzytelnienia stron komunikacji oraz do negocjacji kluczy szyfrowania, co umożliwia bezpieczne nawiązanie połączenia.

II.3. Bezpieczeństwo i zagrożenia w sieciach VPN

Typowe zagrożenia:

- Ataki typu man-in-the-middle (MITM): Polegają na przechwytywaniu danych przesyłanych pomiędzy dwoma stronami komunikacji.
- Podsłuchiwanie ruchu sieciowego: Możliwość przechwytywania danych przesyłanych w niezabezpieczonej sieci.
- Ataki DDoS: Skierowane na infrastrukturę sieciową w celu jej przeciążenia i uniemożliwienia działania VPN.

Metody przeciwdziałania:

- Używanie silnych mechanizmów szyfrowania, takich jak AES-256.
- Uwierzytelnianie dwuskładnikowe (2FA).
- Weryfikacja integralności danych za pomocą funkcji skrótu (np. SHA-256).

Rola integralności danych: Zapewnia, że dane przesyłane w tunelu VPN nie zostaną zmodyfikowane. Uwierzytelnienie końców komunikacji zapobiega podszywaniu się pod legalne urządzenia.

II.4. Infrastruktura klucza publicznego (PKI)

Czym jest PKI: PKI (Public Key Infrastructure) to system zarządzania kluczami publicznymi i certyfikatami cyfrowymi, który umożliwia bezpieczne przesyłanie informacji w sieciach. Klucz publiczny jest używany do szyfrowania danych, podczas gdy klucz prywatny służy do ich odszyfrowania.

Generowanie i zarządzanie certyfikatami:

- Certyfikat urzędu certyfikacji (CA) służy do podpisywania kluczy publicznych.
- Certyfikat serwera zapewnia uwierzytelnienie po stronie serwera.
- Certyfikaty klientów umożliwiają autoryzowanym użytkownikom nawiązywanie połączenia z siecią VPN.
- Proces generowania kluczy i certyfikatów obejmuje tworzenie infrastruktury PKI, podpisywanie kluczy oraz zarządzanie ich ważnością.

II.5. Zastosowania praktyczne VPN

Bezpieczne połączenia firmowe: VPN jest szeroko stosowany w środowiskach biznesowych, gdzie zapewnia bezpieczny dostęp do zasobów sieciowych oraz umożliwia pracownikom bezpieczne połączenie z siecią firmową podczas pracy zdalnej.

Zdalny dostęp do zasobów sieciowych: Pracownicy mogą bezpiecznie łączyć się z bazami danych, plikami oraz innymi zasobami firmowymi niezależnie od lokalizacji.

Ochrona prywatności w Internecie: Użytkownicy indywidualni korzystają z VPN, aby ukryć swoje adresy IP, ominąć regionalne ograniczenia oraz zabezpieczyć połączenia w publicznych sieciach Wi-Fi.

Rozdział III. Instalacja i konfiguracja środowiska VPN

III.1. Przygotowanie środowiska serwera

ubuntu[®] releases

Ubuntu 20.04.6 LTS (Focal Fossa)

Select an image

Ubuntu is distributed on three types of images described below.

Desktop image

The desktop image allows you to try Ubuntu without changing your computer at all, and at your option to install it permanently later. This type of image is what most people will want to use. You will need at least 1024MiB of RAM to install from this image.

64-bit PC (AMD64) desktop image

Choose this if you have a computer based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). Choose this if you are at all unsure.

Server install image

The server install image allows you to install Ubuntu permanently on a computer for use as a server. It will not install a graphical user interface.

64-bit PC (AMD64) server install image

Choose this if you have a computer based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). Choose this if you are at all unsure.

Obraz 1. Wybór odpowiedniego obrazu instalacyjnego Ubuntu 20.04.6 LTS

Obraz przedstawia stronę internetową z sekcją wyboru obrazu instalacyjnego dla systemu operacyjnego Ubuntu 20.04.6 LTS (Focal Fossa). Użytkownik ma do wyboru dwa główne typy obrazów instalacyjnych:

1. Desktop image:

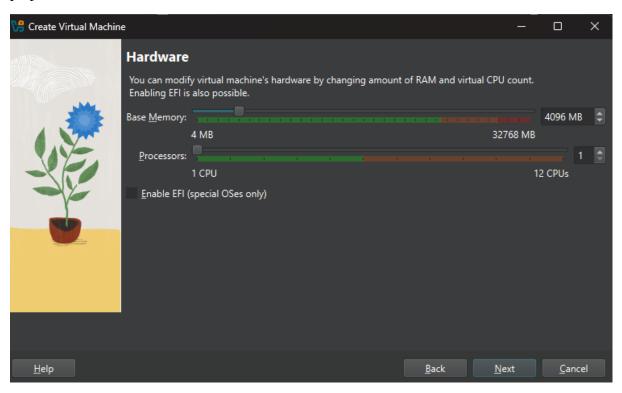
- o Przeznaczony dla komputerów stacjonarnych i laptopów.
- o Pozwala na uruchomienie Ubuntu w trybie Live bez konieczności instalacji.
- o Może być później zainstalowany na stałe.

2. Server install image:

- o Przeznaczony do instalacji na komputerach, które będą działały jako serwery.
- o Nie posiada graficznego interfejsu użytkownika (CLI).
- Zaprojektowany dla zaawansowanych użytkowników, takich jak administratorzy systemów.

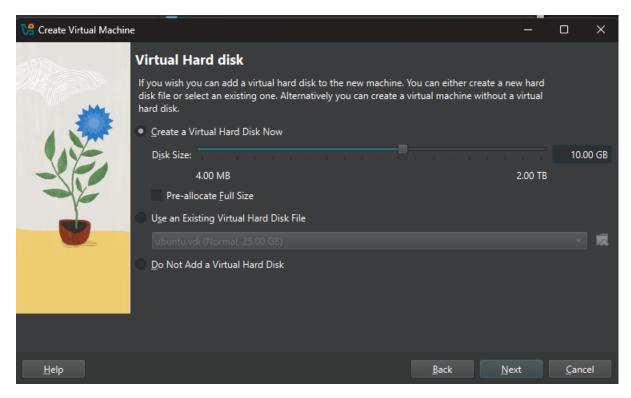
Obie opcje mają wersję dla architektury **64-bit PC (AMD64)**, co oznacza, że są przeznaczone dla komputerów opartych na nowoczesnych procesorach, takich jak Athlon64, Xeon, lub Core 2.

Obraz ten przedstawia krok wyboru odpowiedniego wariantu instalacyjnego w procesie pobierania Ubuntu, co jest istotnym elementem przygotowania środowiska serwera dla projektu.



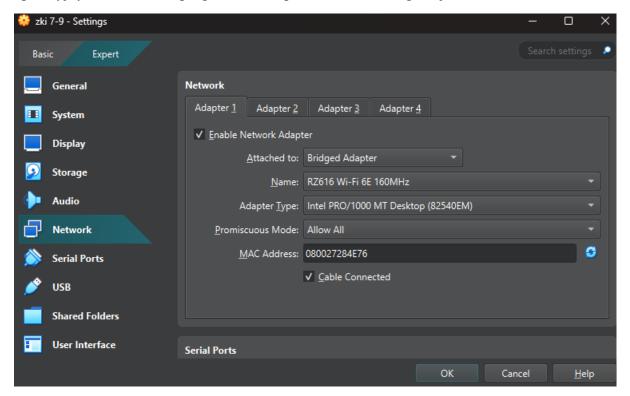
Obraz 2. Konfiguracja zasobów maszyny wirtualnej dla Ubuntu Server.

Ekran konfiguracji umożliwia dostosowanie ilości pamięci RAM (ustawionej na 4096 MB) oraz liczby wirtualnych procesorów (1 CPU) do wymagań systemu. Możliwość włączenia EFI (Extensible Firmware Interface) jest dostępna, ale zalecana tylko dla specjalnych systemów operacyjnych. Właściwe dobranie zasobów zapewnia optymalne działanie środowiska serwera podczas realizacji projektu.



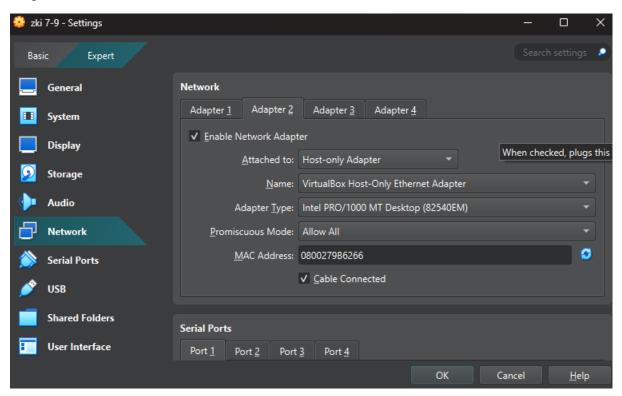
Obraz 3. Tworzenie wirtualnego dysku twardego dla maszyny Ubuntu.

Użytkownik wybiera opcję utworzenia nowego wirtualnego dysku twardego o rozmiarze 10 GB. Istnieje możliwość zmiany rozmiaru dysku w zakresie od 4 MB do 2 TB. Opcja "Preallocate Full Size" jest wyłączona, co oznacza dynamiczne przydzielanie miejsca na dysku w miarę potrzeby. Odpowiednia wielkość dysku zapewnia wystarczającą przestrzeń na system operacyjny i dodatkowe oprogramowanie potrzebne do konfiguracji serwera.



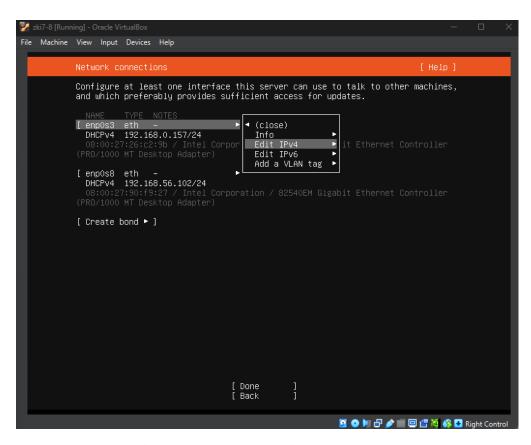
Obraz 4. Ustawienia adaptera sieciowego w trybie "Bridged Adapter".

Pierwszy adapter sieciowy jest skonfigurowany w trybie "Bridged Adapter", co oznacza, że wirtualna maszyna będzie działać jak fizyczne urządzenie w tej samej sieci co host. Interfejs sieciowy jest przypisany do fizycznego adaptera Wi-Fi (RZ616 Wi-Fi 6E). Ustawienie "Allow All" w trybie nasłuchiwania (Promiscuous Mode) umożliwia przechwytywanie całego ruchu sieciowego, co może być przydatne do monitorowania sieci i analizy bezpieczeństwa

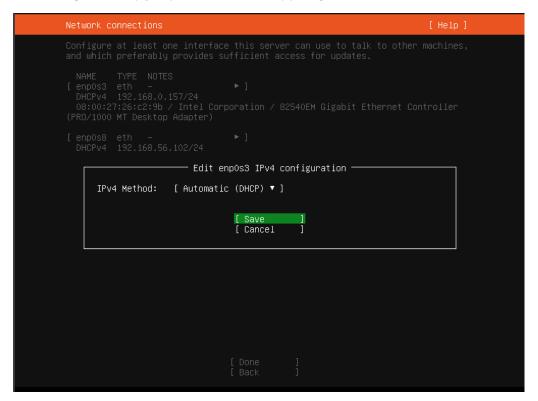


Obraz 5. Ustawienia adaptera sieciowego w trybie "Host-only Adapter".

Drugi adapter sieciowy jest skonfigurowany jako "Host-only Adapter", co pozwala na komunikację między maszyną wirtualną a hostem bez dostępu do sieci zewnętrznej. Wirtualny adapter sieciowy "VirtualBox Host-Only Ethernet Adapter" umożliwia bezpośrednie połączenie do testowania usług serwera, takich jak OpenVPN, w odizolowanym środowisku. Dzięki temu można bezpiecznie przeprowadzać testy bez ryzyka zakłócenia rzeczywistej sieci.

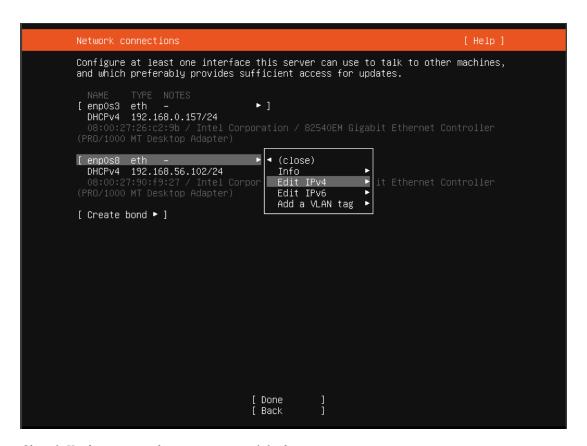


Obraz 6. Rozpoczęcie konfiguracji adresu IPv4 dla interfejsu enp0s8



Obraz 7. Konfiguracja adresu IPv4 dla interfejsu enp0s8

Interfejs enp0s3 został skonfigurowany jako karta typu "Bridged" z automatycznym przydzielaniem adresu IP poprzez DHCP (192.168.0.157/24). Pozwala to maszynie wirtualnej na bezpośrednią komunikację w tej samej sieci co host.

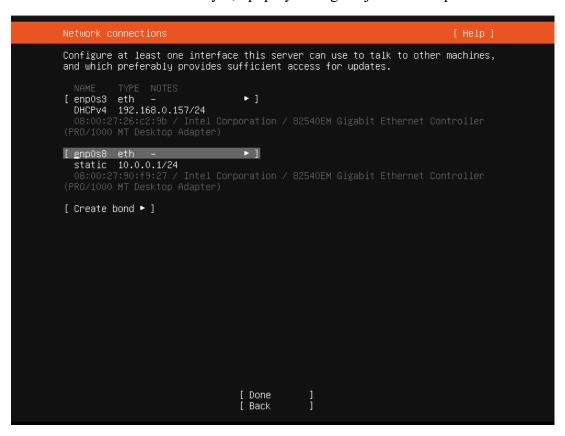


Obraz 8. Konfiguracja interfejsu sieciowego enp0s8 jako sieci wewnętrznej.

Network connections	[Help]
Configure at least one interface this server and which preferably provides sufficient acc	can use to talk to other machines, ess for updates.
NAME TYPE NOTES	
Edit enpOs8 IPv4 co	nfiguration ————————————————————————————————————
IPv4 Method: [Manual ▼]	
Subnet: 10.0.0.0/24	
Address: 10.0.0.1	
Gateway:	
Name servers: IP addresses, comma s	eparated
Search domains: Domains, comma separa	ted
[Save [Cancel	1
[Done	1
[Back	

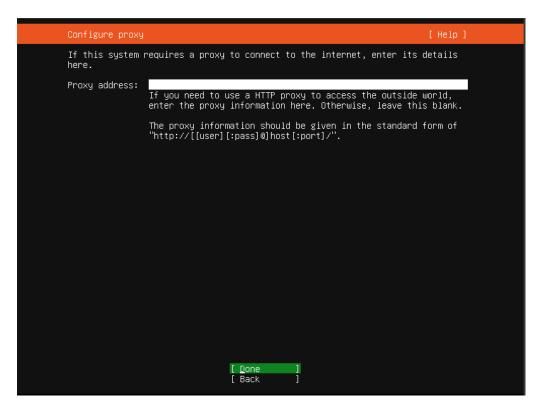
Obraz 9. Ręczna konfiguracja adresu IP dla sieci wewnętrznej enp0s8

Dla interfejsu enp0s8 skonfigurowano sieć wewnętrzną z podsiecią 10.0.0.0/24 i statycznym adresem IP 10.0.0.1. To ustawienie umożliwia izolowaną komunikację wewnętrzną, co jest kluczowe w środowisku testowym, np. przy konfiguracji serwera OpenVPN.



Obraz 10. Zakończona konfiguracja interfejsów sieciowych na serwerze.

Interfejs enp0s3 działa w trybie "Bridged Adapter" z automatycznie przydzielonym adresem IP 192.168.0.157/24, zapewniając dostęp do internetu. Interfejs enp0s8 został skonfigurowany jako statyczny z adresem IP 10.0.0.1/24, tworząc wewnętrzną sieć izolowaną, idealną do testowania usług serwera, takich jak OpenVPN.



Obraz 11. Konfiguracja połączenia bez użycia serwera proxy.

Ekran konfiguracji serwera proxy został pozostawiony pusty, co oznacza bezpośredni dostęp do internetu bez pośrednictwa dodatkowych serwerów. W takim przypadku system będzie łączyć się z siecią zewnętrzną bez żadnych ograniczeń, co jest standardowym wyborem w większości środowisk testowych.

Guided storage configuration	[Help]
Configure a guided storage layout, or create a custom one:	
(X) Use an entire disk	
[VBOX_HARDDISK_VB9eda6595-0802eba1 local disk 10.000G ▼]	
[X] Set up this disk as an LVM group	
[] Encrypt the LVM group with LUKS	
Passphrase:	
Confirm passphrase:	
() Custom storage layout	
[Done] [Back]	

Obraz 12. Automatyczne tworzenie partycji z wykorzystaniem całego dysku.

```
Storage configuration
FILE SYSTEM SUMMARY
                        8.246G new ext4 new LVM logical volume 

1.750G new ext4 new partition of local disk 

[/boot
AVAILABLE DEVICES
USED DEVICES
                                                              LVM volume group
[ ubuntu-vg (new)
                                                                                            8.246G
   ubuntu-1v
                    new, to be formatted as ext4, mounted at 7
[ VBOX_HARDDISK_VB9eda6595-0802eba1
                                                              local disk
                                                                                           10.000G ▶ ]
  partition 1 new, BIOS grub spacer
partition 2 new, to be formatted as ext4, mounted at /boot
partition 3 new, PV of LVM volume group ubuntu–vg
                                                                                            1.000M
                                                                                            1.750G
8.247G
                                             [ Done
                                             [ Reset
[ Back
```

Obraz 13. Podsumowanie konfiguracji systemu plików i partycji.

Automatycznie utworzono partycje: główną partycję systemową / o rozmiarze 8,246 GB oraz partycję rozruchową /boot o rozmiarze 1,75 GB. Partycje sformatowano w systemie plików ext4, co zapewnia stabilność i efektywność pracy systemu. LVM został skonfigurowany dla dynamicznego zarządzania przestrzenią dyskową.

Profile setup		[Help]
Enter the username and configure SSH access o sudo.	password you will use to log in t n the next screen but a password i	o the system. You can s still needed for
Your name:	Rozbicki Lukasz	
Your server's name:	serverrozbicki The name it uses when it talks t	o other computers.
Pick a username:	rozbicki	
Choose a password:	жжжж	
Confirm your password∶	жоюкк <u> </u>	
	[Done]	

Obraz 14. Tworzenie profilu użytkownika i konfiguracja serwera.

Wprowadzono dane użytkownika i serwera podczas instalacji Ubuntu. Nazwa serwera to serverrozbicki, a nazwa użytkownika to rozbicki. Użytkownik zdefiniował silne hasło do późniejszego użycia podczas logowania i wykonywania komend z uprawnieniami administratora (sudo). Te ustawienia są kluczowe dla bezpieczeństwa i identyfikacji w sieci.

SSH Setup	[Help]
You can choose to ins access to your server	stall the OpenSSH server package to enable secure remote r.
[x]	Install OpenSSH server
Import SSH identity:	[No ▼] You can import your SSH keys from GitHub or Launchpad.
Import Username:	
[x]	
	[Done] [Back]

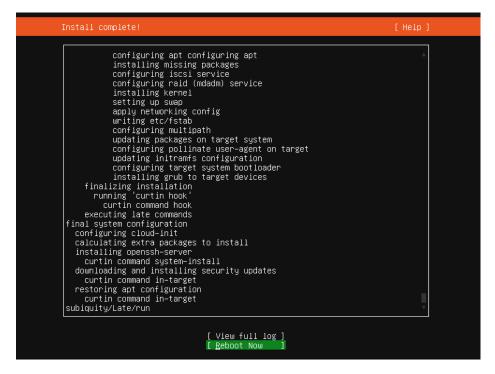
Obraz 15. Instalacja serwera OpenSSH i konfiguracja dostępu zdalnego.

Włączono instalację serwera OpenSSH, co umożliwia zdalne logowanie do maszyny za pomocą protokołu SSH. Import kluczy SSH został pominięty, a uwierzytelnianie hasłem jest dozwolone. To ustawienie jest kluczowe do zarządzania serwerem bez fizycznego dostępu do niego, zapewniając wygodę i bezpieczeństwo administracji.



Obraz 16. Pominięto instalację dodatkowych usług serwerowych.

Lista dostępnych pakietów serwerowych typu Snap, takich jak Docker, Nextcloud, PostgreSQL czy Prometheus, została pozostawiona bez wyboru. Użytkownik zdecydował się na minimalistyczną instalację, ograniczając dodatkowe oprogramowanie do niezbędnego minimum, co jest korzystne dla lepszej kontroli nad konfiguracją systemu i wydajnością.



Obraz 17. Zakończenie instalacji systemu i przygotowanie do restartu.

Instalacja Ubuntu Server została pomyślnie zakończona, obejmując konfigurację podstawowych usług, instalację jądra systemu, serwera OpenSSH oraz aktualizacje zabezpieczeń. Użytkownik może teraz zrestartować system, aby zakończyć proces i przejść do logowania na nowo skonfigurowanym serwerze.

```
23.351375] cloud-init[1568]: en_US.UTF-8... done
23.351466] cloud-init[1568]: Generation complete.
23.679333] cloud-init[1568]: Cloud-init v. 23.1.2–OubuntuO~20.04.2 running 'modules:config' at
on, 13 Jan 2025 18:07:09 +0000. Up 21.91 seconds.
ci–info: no authorized SSH keys fingerprints found for user rozbicki.
 <14>Jan 13 18:07:11 cloud—init: ----BEGIN SSH HOST KEY FINGERPRINTS-----
<14>Jan 13 18:07:11 cloud—init: 1024 SHA256:pOri+QDmusRya+y2FOqDhEwFqJuonAxd3g0gN6JD9kw root@server
 zbicki (DSA)
 :14>Jan 13 18:07:11 cloud–init: 256 SHA256:Y7c3EvIt9mAJ7yam1v//6wQxlBW4U3d7gyDfkQyJk/s root@serverr
 bicki (ECDSA)
 zbicki (ED25519)
K14>Jan 13 18:07:11 cloud–init: 3072 SHA256:20qEvmyUF/2bc9T3vJ5fEaKNzVoM2k713OB2RuA5RUo root@server
pzbicki (RSA)
 (14)Jan 13 18:07:11 cloud-init: ----END SSH HOST KEY FINGERPRINTS-
 ---BEGIN SSH HOST KEY KEYS-
ecdsa–sha2–nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBDuOtEY5at9E9p6UA4Y5p7NVuFi
iPetQrJfEgmMwfQqZkQHuzjZkUjG/1cVefVclYfgYI5j/IKDkpReayhBtSQ= root@serverrozbicki
ssh–ed25519 AAAAC3NzaC11ZDI1NTE5AAAAIJaKlc8iBr9rYHEO4tpaOObCSiAwnUFm2bVvn4QeyzqS root@serverrozbick
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDhwGVPLe/BWe99HHWrvOkLvcmaPQlioS+EZA3CWmHJQKqNWXAnQKxufFAbTKb.
GAhf+rzVUU+ZSQZCkwQ28VxZC9qVRc/Txf4++a1hDgeD695hWOyzSsxURs2ufUBOeS+hiXc+FKXd8mFqx8yrF48YrWqoAizltodq
3FantHUs6WQvKEXmxf/kN6k3XploGwvrE2NygUM2MjrMqzRPd7JhEjwnJaiOM8dOc6Gr2PAeLQ2PX4vkLQd1SexgOb/Vk8cJjkY.
_FbW/PxtXWi9CJrgNLotpo82N6lauJyYUeEHj8grBv165vEr3MbS6q2ydV2666h6IQnP2CPkxK4lICN7B2CuAPuQIIc8KfRRCVU
KtCp4ugL1tsIf6Hpb/x4Fxp5/WwqhEZR0enj6CdgETjCdG7x9OOpkCMR3Ri5TvKjyHxFxnZDqqaZGWPax8xAOyeMypLQdt7ieX3
 lofOW6LKqjt6byrp8F6IuYDCtUBs85O7WOaZzgj5G1/6AY41EOc= root@serverrozbicki
      -END SSH HOST KEY KEYS
     24.259338] cloud-init[1609]: Cloud-init v. 23.1.2-OubuntuO~20.04.2 running 'modules:final' at Mo
    13 Jan 2025 18:07:11 +0000. Up 24.07 seconds.
24.259485] cloud-init[1609]: Cloud-init v. 23.1.2-OubuntuO~20.04.2 finished at Mon, 13 Jan 2025
18:07:11 +0000. Datasource DataSourceNone. Up 24.23 seconds
[ 24.259961] cloud-init[1609]: 2025-01-13 18:07:11,994 – cc_final_message.py[WARNING]: Used fallba
    datasource
 erverrozbicki login:
```

Obraz 18. Ekran logowania do nowo zainstalowanego serwera Ubuntu.

System uruchomił się pomyślnie po instalacji, wyświetlając klucze hosta SSH oraz informacje inicjalizacyjne. Użytkownik rozbicki może teraz zalogować się, wpisując swoje wcześniej skonfigurowane dane logowania. Serwer jest gotowy do dalszej konfiguracji, np. instalacji OpenVPN lub innych usług sieciowych.

```
System information as of Mon 13 Jan 2025 06:09:32 PM UTC
                                     0.14
30.1% of 8.02GB
  Usage of /:
  Memory usage:
  Processes:
Users logged in:
  IPv4 address for enp0s3: 2a02:a31d:e0ec:c200:a00:27ff:fe26:c29b

IPv4 address for enp0s8: 10.0.0.1
 xpanded Security Maintenance for Applications is not enabled.
  updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
Gee https://ubuntu.com/esm or run: sudo pro status
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
 ndividual files in /usr/share/doc/*/copyright.
Jbuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
To run a command as administrator (user "root"), use "sudo ‹command›".
See "man sudo_root" for details.
rozbicki@serverrozbicki:~$ sudo touch /etc/cloud/cloud–init.disabled
[sudo] password for rozbicki:
rozbicki@serverrozbicki:~$ reboot
```

Obraz 19. Wyłączenie inicjowania pakietu Cloud-Init i restart systemu.

Użytkownik rozbicki wykonał polecenie sudo touch /etc/cloud/cloud-init.disabled, aby wyłączyć pakiet Cloud-Init, który jest używany do inicjalizacji systemu w chmurze. Po zakończeniu tej operacji został wykonany restart systemu (reboot), aby zmiany zaczęły obowiązywać. Wyłączenie Cloud-Init przyspiesza start serwera w środowiskach lokalnych, gdzie jego funkcje nie są potrzebne.

```
Resword:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-204-generic x86_64)

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

System information as of Mon 13 Jan 2025 06:12:34 PM UTC

System load: 0.41
Usage of /: 30.2% of 8.0268
Memory usage: 4%
Swap usage: 0%
Processes: 104
Users logged in: 0
IPv4 address for enp0s3: 192.168.0.157
IPv6 address for enp0s3: 2202:a31d:e0ec:c200:a00:27ff:fe26:c29b
IPv4 address for enp0s8: 10.0.0.1

Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
New release '22.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Jan 13 18:09:33 UTC 2025 on tty1
rozbicki@serverrozbicki:~$ sudo apt upgrade
```

Obraz 20. Aktualizacja systemu za pomocą sudo apt update i sudo apt upgrade

Po zalogowaniu na serwer Ubuntu użytkownik rozbicki uruchomił komendy sudo apt update oraz sudo apt upgrade, aby zaktualizować listę pakietów w repozytoriach i zainstalować najnowsze wersje oprogramowania. Taka procedura zapewnia bezpieczeństwo systemu oraz dostęp do najnowszych funkcji i poprawek błędów.

```
rozbicki@serverrozbicki:~$ sudo ufw allow OpenSSH
Rules updated
Rules updated (v6)
rozbicki@serverrozbicki:~$ ufw enable
ERROR: You need to be root to run this script
rozbicki@serverrozbicki:~$ sudo ufw enable
Firewall is active and enabled on system startup
rozbicki@serverrozbicki:~$ _
```

Obraz 21. Włączenie zapory sieciowej UFW i zezwolenie na połączenia SSH.

Za pomocą polecenia sudo ufw allow OpenSSH użytkownik zezwolił na połączenia zdalne poprzez protokół SSH. Następnie polecenie sudo ufw enable włączyło zaporę sieciową UFW (Uncomplicated Firewall), która jest teraz aktywna i będzie automatycznie uruchamiana przy starcie systemu. Dzięki temu serwer jest chroniony przed nieautoryzowanymi próbami dostępu.

III.2. Instalacja i konfiguracja OpenVPN.

Instalacja OpenVPN i pakietu easy-rsa:

```
Inpacking posod (1.8.26–3)
Selecting previously unselected package libpkcs11–helper1:amd64.
reparing to unpack .../3–libpkcs11–helper1_1.26–1_amd64.deb ...
Unpacking libpkcs11–helper1:amd64 (1.26–1) ...
Selecting previously unselected package opensc-pkcs11:amd64.
Preparing to unpack .../4-opensc-pkcs11_0.20.0-3_amd64.deb ...
Unpacking opensc-pkcs11:amd64 (0.20.0-3) ...
Selecting previously unselected package opensc.
Preparing to unpack .../5-opensc_0.20.0-3_amd64.deb ...
Junacking opensc (0.20.0-3) ...
Selecting previously unselected package openvpn.
reparing to unpack .../6–openvpn_2.4.12–Oubuntu0.20.04.2_amd64.deb ...
Unpacking openvpn (2.4.12–Oubuntu0.20.04.2) ...
Selecting previously unselected package easy—rsa.
Preparing to unpack .../7—easy—rsa_3.0.6—1_all.deb ...
Unpacking easy—rsa (3.0.6—1) ...
Setting up libpkcid (1.4.31—1) ...
Setting up libpkcs11—helper1:amd64 (1.26—1) ...
Setting up opensc-pkcs11:amd64 (0.20.0–3) ...
Setting up libposolite1:amd64 (1.8.26–3) ...
Setting up easy—rsa (3.0.6–1) ...
Setting up openvpn (2.4.12–0ubuntu0.20.04.2) ...
* Restarting virtual private network daemon. [ OK ]
Created symlink /etc/systemd/system/multi–user.target.wants/openvpn.service → /lib/systemd/system/op
envpn.service.
setting up opensc (0.20.0–3) ...
Getting up posod (1.8.26–3) ...
Created symlink /etc/systemd/system/sockets.target.wants/posod.socket → /lib/systemd/system/posod.so
ket.
posod.service is a disabled or a static unit, not starting it.
rocessing triggers for man–db (2.9.1–1) .
Processing triggers for mime–support (3.64ubuntu1) ...
Processing triggers for libc–bin (2.31–0ubuntu9.16) ...
Processing triggers for systemd (245.4–4ubuntu3.24) ...
 ozbicki@serverrozbicki:
```

Obraz 22. Instalacja pakietów OpenVPN i Easy-RSA.

Użytkownik wykonał polecenie sudo apt install openvpn easy-rsa, co zainstalowało OpenVPN - oprogramowanie do zarządzania wirtualnymi sieciami prywatnymi, oraz Easy-RSA - narzędzie do zarządzania infrastrukturą kluczy publicznych (PKI). Proces instalacji zakończył się pomyślnie, przygotowując serwer do konfiguracji połączeń VPN i generowania certyfikatów zabezpieczających.

```
rozbicki@serverrozbicki:~$ sudo make-cadir /etc/openvpn/easy-rsa
rozbicki@serverrozbicki:~$ sudo su
root@serverrozbicki:/home/rozbicki# cd /etc/openvpn/easy-rsa
root@serverrozbicki:/etc/openvpn/easy-rsa#
```

Obraz 23. Tworzenie katalogu dla infrastruktury CA za pomocą Easy-RSA.

Użytkownik wykonał polecenie sudo make-cadir /etc/openvpn/easy-rsa, które utworzyło dedykowany katalog do przechowywania plików związanych z infrastrukturą klucza publicznego (CA - Certificate Authority). Następnie przełączono się na konto superużytkownika (sudo su) i zmieniono bieżący katalog na /etc/openvpn/easy-rsa. Ten krok jest niezbędny do rozpoczęcia procesu generowania kluczy i certyfikatów potrzebnych do bezpiecznego działania OpenVPN.

```
Modified
  Organizational fields (used with 'org' mode and ignored in 'cn_only' mode.)
These are the default values for fields which will be placed in the
certificate. Don't leave any of these fields blank, although interactively
you may omit any specific field by typing the "." symbol (not valid for
set_var EASYRSA_REQ_COUNTRY
set_van EASYRSA_REQ_DOUNINCE
set_van EASYRSA_REQ_DITY
set_van EASYRSA_REQ_DRG
set_van EASYRSA_REQ_EMAIL
                                                                   "Mazowieckie"
                                                                 "Warszawa
                                                                 "WAT
                                                                    rozbicki@example.net"
 et_var EASYRSA_REQ_OU
  Choose a size in bits for your keypairs. The recommended value is 2048. Using 2048—bit keys is considered more than sufficient for many years into the future. Larger keysizes will slow down TLS negotiation and make key/DH param generation take much longer. Values up to 4096 should be accepted by most
   software. Only used when the crypto alg is rsa (see below.)
 set_var EASYRSA_KEY_SIZE
  The default crypto mode is rsa; ec can enable elliptic curve support. Note that not all software supports ECC, so use care when enabling it. Choices for crypto alg are: (each in lower-case)
    * rsa
    ж ес
set_var EASYRSA_ALGO
  Define the named curve, used in ec mode only:
 set_var_EASYRSA_CURVE
                                                                 secp384r1
                                                                                                                                                       Cur Pos M–U Undo
Go To Line <u>M–E</u> Redo
                            ^O Write Out
^R Read File
                                                                                                                         Justify
To Spell
    Get Help
                                                               Where Is
                                                               Replace
                                                                                             Paste Text
```

Obraz 24. Edycja pliku /etc/openvpn/easy-rsa/vars dla generowania certyfikatów..

W pliku vars skonfigurowano podstawowe parametry dla infrastruktury kluczy publicznych (PKI). Ustawiono między innymi lokalizację (Polska, Mazowieckie, Warszawa), nazwę organizacji (WAT), oraz e-mail kontaktowy rozbicki@example.net. Wybrano algorytm

krzywych eliptycznych (ECC) do generowania kluczy, co zapewnia wysoki poziom bezpieczeństwa przy mniejszej długości klucza niż w przypadku RSA.

```
GNU nano 4.8
                                                                                                                                                                        Modified
fallback to $EASYRSA for the 'x509–t
detection with an explicit dir here.
set_var EASYRSA_EXT_DIR
If you need to use a specific opensal config file, you can reference it here.

Normally this file is auto-detected from a file named opensal-easyrsa.cnf from the EASYRSA_PKI or EASYRSA dir (in that order.) NOTE that this file is Easy-RSA specific and you cannot just use a standard config file, so this is an advanced feature.
set_var EASYRSA_SSL_CONF
                                                        "$EASYRSA/openssl-easyrsa.cnf"
This is best left alone. Interactively you will set this manually, and BATCH callers are expected to set this themselves.
set_var_EASYRSA_REQ_CN
Cryptographic digest to use.
Do not change this default unless you understand the security implications.
Valid choices include: md5, sha1, sha256, sha224, sha384, sha512
et_var EASYRSA_DIGEST
                                                          "sha256"
 in batch mode without any user input, confirmation on dangerous operations, or most output. Setting this to any non-blank string enables batch mode.
set_var EASYRSA_BATCH
                        ^O Write Out
^R Read File
                                                 ^W Where Is
^∖ Replace
                                                                             ^K Cut Text
^U Paste Tex
                                                                                                             Justify
To Spell
   Get Help
Exit
                                                        Replace
                                                                                   Paste Text
```

Obraz 25. Kontynuacja konfiguracji parametrów bezpieczeństwa w pliku vars

W dalszej części pliku vars skonfigurowano kluczowe opcje bezpieczeństwa, takie jak algorytm skrótu (sha256) używany w podpisywaniu certyfikatów oraz lokalizację plików konfiguracyjnych OpenSSL. Ustawienia te są kluczowe do zapewnienia bezpiecznej i poprawnej pracy serwera VPN opartego na OpenVPN.

```
Note: using Easy-RSA configuration from: ./vars

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /etc/openvpn/easy-rsa/pki
root@serverrozbicki:/etc/openvpn/easy-rsa# ./easyrsa build-ca
Note: using Easy-RSA configuration from: ./vars
Using SSL: openssl OpenSSL 1.1.1f 31 Mar 2020
Enter New CA Key Passphrase:
Re-Enter New CA Key Passphrase:
Re-Enter New CA Key Passphrase:
willing EC key
uriting EC key
Can't load /etc/openvpn/easy-rsa/pki/.rnd into RNG
140119014077750:error:2406F079:random number generator:RAND_load_file:Cannot open file:../crypto/rand/rousereabout to be asked to enter information that will be incorporated
into your certificate request.
What you are about to the 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,
Tif you enter '.', the field will be left blank.
----
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:CArozbicki
CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/etc/openvpn/easy-rsa/pki/ca.crt
root@serverrozbicki:/etc/openvpn/easy-rsa# _
```

Obraz 26. Tworzenie infrastruktury PKI i budowa urzędu certyfikacji (CA).

Polecenie ./easyrsa init-pki zainicjowało infrastrukturę PKI, tworząc niezbędne katalogi i pliki konfiguracyjne do zarządzania kluczami i certyfikatami. Następnie polecenie ./easyrsa build-ca utworzyło urząd certyfikacji (CA), który będzie podpisywał certyfikaty dla serwera i klientów OpenVPN. Wprowadzono nazwę wspólną (CN) CAnazwisko, co identyfikuje CA w procesie uwierzytelniania. Certyfikat CA został zapisany w katalogu /etc/openvpn/easyrsa/pki/ca.crt.

```
root@serverrozbicki:/etc/openvpn/easy-rsa# ./easyrsa gen-req serverrozbicki nopass

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1f 31 Mar 2020
Generating an EC private key
writing new private key to '/etc/openvpn/easy-rsa/pki/private/serverrozbicki.key.WStUimDkl7'
----
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) [serverrozbicki]:serverrozbicki

Keypair and certificate request completed. Your files are:
req: /etc/openvpn/easy-rsa/pki/reqs/serverrozbicki.req
Key: /etc/openvpn/easy-rsa/pki/private/serverrozbicki.key

root@serverrozbicki:/etc/openvpn/easy-rsa# _
```

Obraz 27. Generowanie klucza prywatnego i żądania certyfikatu dla serwera OpenVPN

Wykonano polecenie ./easyrsa gen-req servernazwisko nopass, które wygenerowało klucz prywatny serwera oraz plik żądania certyfikatu (CSR) z nazwą wspólną (CN) servernazwisko. Klucz prywatny został zapisany w katalogu /etc/openvpn/easyrsa/pki/private/serverrozbicki.key, a żądanie certyfikatu w pliku /etc/openvpn/easyrsa/pki/reqs/serverrozbicki.req. Te pliki są niezbędne do podpisania certyfikatu przez CA i zabezpieczenia komunikacji serwera.

```
oot@serverrozbicki:/etc/openvpn/easy–rsa# ./easyrsa sign–req server serverrozbicki
Note: using Easy–RSA configuration from: ./vars
Using SSL: openssl OpenSSL 1.1.1f 31 Mar 2020
You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.
Request subject, to be signed as a server certificate for 1080 days:
subject=
    commonName
                                = serverrozbicki
Type the word 'yes' to continue, or any other input to abort.
 Confirm request details: yes
Using configuration from /etc/openvpn/easy–rsa/pki/safessl–easyrsa.cnf
Enter pass phrase for /etc/openvpn/easy–rsa/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows commonName :ASN.1 12:'serverrozbicki'
Certificate is to be certified until Dec 29 18:32:52 2027 GMT (1080 days)
Write out database with 1 new entries
Data Base Updated
Certificate created at: /etc/openvpn/easy-rsa/pki/issued/serverrozbicki.crt
root@serverrozbicki:/etc/openvpn/easy-rsa# _
```

Obraz 28. Podpisanie żądania certyfikatu serwera przez urząd certyfikacji (CA).

Wykonano polecenie ./easyrsa sign-req server servernazwisko, które podpisało żądanie certyfikatu dla serwera OpenVPN. Proces wymagał potwierdzenia poprawności danych oraz podania hasła do klucza prywatnego CA. Certyfikat serwera został pomyślnie wygenerowany i zapisany w katalogu /etc/openvpn/easy-rsa/pki/issued/serverrozbicki.crt. Certyfikat ten jest ważny przez 1080 dni i będzie używany do uwierzytelniania serwera w sieci VPN.

```
root@serverrozbicki:/etc/openvpn/easy-rsa# mkdir -p /etc/openvpn/server
root@serverrozbicki:/etc/openvpn/easy-rsa# cp /etc/openvpn/easy-rsa/pki/ca.crt /etc/openvpn/server
root@serverrozbicki:/etc/openvpn/easy-rsa# cp /etc/openvpn/easy-rsa/pki/reqs/serverrozbicki.req /etc
/openvpn/server
root@serverrozbicki:/etc/openvpn/easy-rsa# cp /etc/openvpn/easy-rsa/pki/private/serverrozbicki.key /
etc/openvpn/server
root@serverrozbicki:/etc/openvpn/easy-rsa# cp /etc/openvpn/easy-rsa/pki/issued/serverrozbicki.crt /e
tc/openvpn/server
root@serverrozbicki:/etc/openvpn/easy-rsa# _
```

Obraz 29. Kopiowanie certyfikatów i kluczy do katalogu serwera OpenVPN.

Utworzono katalog /etc/openvpn/server, a następnie skopiowano do niego wymagane pliki:

- Certyfikat urzędu certyfikacji ca.crt
- Żądanie certyfikatu serwera serverrozbicki.req
- Klucz prywatny serwera serverrozbicki.key
- Certyfikat serwera serverrozbicki.crt

```
root@serverrozbicki:/etc/openvpn/easy–rsa# cd /etc/openvpn/client
root@serverrozbicki:/etc/openvpn/client# mkdir keys
root@serverrozbicki:/etc/openvpn/client#
```

Obraz 30. Tworzenie struktury katalogów dla kluczy i certyfikatów klienta OpenVPN.

Przełączono się do katalogu /etc/openvpn/client, a następnie utworzono podkatalog keys za pomocą polecenia mkdir keys. Katalog keys będzie przechowywał klucze prywatne i certyfikaty niezbędne dla klienta VPN do nawiązywania bezpiecznego połączenia z serwerem OpenVPN.

```
root@serverrozbicki:/etc/openvpn/easy-rsa# cd /etc/openvpn/client
root@serverrozbicki:/etc/openvpn/client# mkdir keys
root@serverrozbicki:/etc/openvpn/client# cd ..
root@serverrozbicki:/etc/openvpn# chmod –R 700 client
root@serverrozbicki:/etc/openvpn#
```

Obraz 31. Nadanie praw dostępu dla katalogu klienta i jego zawartości.

Wykonano polecenie chmod -R 700 client, które przypisało katalogowi /etc/openvpn/client, jego podkatalogom oraz plikom prawa dostępu tylko dla właściciela. Właściciel ma pełne prawa do odczytu, zapisu i wykonywania, podczas gdy inni użytkownicy nie mają żadnego dostępu.

```
root@serverrozbicki:/etc/openvpn# cd /etc/openvpn/easy-rsa
root@serverrozbicki:/etc/openvpn/easy-rsa# ./easyrsa gen-req clientrozbicki nopass

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1f 31 Mar 2020
Generating an EC private key
writing new private key to '/etc/openvpn/easy-rsa/pki/private/clientrozbicki.key.bRWEYYEkod'
-----
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) [clientrozbicki]:clientrozbicki

Keypair and certificate request completed. Your files are:
req: /etc/openvpn/easy-rsa/pki/reqs/clientrozbicki.req
key: /etc/openvpn/easy-rsa/pki/private/clientrozbicki.key

root@serverrozbicki:/etc/openvpn/easy-rsa# _
```

Obraz 32. Generowanie klucza prywatnego i żądania certyfikatu dla klienta OpenVPN.

Wykonano polecenie ./easyrsa gen-req clientrozbicki nopass, które wygenerowało klucz prywatny i plik żądania certyfikatu (CSR) dla klienta OpenVPN z nazwą wspólną (CN) clientnazwisko. Klucz prywatny został zapisany w katalogu /etc/openvpn/easy-rsa/pki/private/clientrozbicki.key, a plik żądania certyfikatu w /etc/openvpn/easy-rsa/pki/reqs/clientrozbicki.req. Te pliki są niezbędne do podpisania certyfikatu przez urząd

```
oot@serverrozbicki:/etc/openvpn/easy–rsa# ./easyrsa sign–req client clientrozbicki
Note: using Easy—RSA configuration from: ./vars
Using SSL: openssl OpenSSL 1.1.1f 31 Mar 2020
You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted source or that you have verified the request checksum with the sender.
Request subject, to be signed as a client certificate for 1080 days:
subject=
     commonName
                                      = clientrozbicki
Type the word 'yes' to continue, or any other input to abort.
 Confirm request details: yes
Using configuration from /etc/openvpn/easy–rsa/pki/safessl–easyrsa.cnf
Enter pass phrase for /etc/openvpn/easy–rsa/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName :ASN.1 12:'clientrozbicki'
Certificate is to be certified until Dec 29 18:39:06 2027 GMT (1080 days)
Write out database with 1 new entries
Data Base Updated
Certificate created at: /etc/openvpn/easy-rsa/pki/issued/clientrozbicki.crt
root@serverrozbicki:/etc/openvpn/easy-rsa#
```

Obraz 33. Podpisanie żądania certyfikatu klienta przez urząd certyfikacji (CA).

Wykonano polecenie ./easyrsa sign-req client clientrozbicki, które podpisało żądanie certyfikatu klienta OpenVPN. Proces wymagał potwierdzenia poprawności danych i podania hasła do klucza prywatnego CA. Certyfikat klienta został wygenerowany i zapisany w katalogu /etc/openvpn/easy-rsa/pki/issued/clientrozbicki.crt. Certyfikat ten jest ważny przez 1080 dni i pozwala klientowi na bezpieczne łączenie się z serwerem OpenVPN.

```
root@serverrozbicki:/etc/openvpn/easy-rsa# cp /etc/openvpn/easy-rsa/pki/reqs/clientrozbicki.req /etc
/openvpn/client/keys
root@serverrozbicki:/etc/openvpn/easy-rsa# cp /etc/openvpn/easy-rsa/pki/private/clientrozbicki.key /
etc/openvpn/client/keys/
root@serverrozbicki:/etc/openvpn/easy-rsa# cp /etc/openvpn/easy-rsa/pki/issued/clientrozbicki.crt /e
tc/openvpn/client/keys/
root@serverrozbicki:/etc/openvpn/easy-rsa# cp /etc/openvpn/easy-rsa/pki/ca.crt /etc/openvpn/client/k
eys/
root@serverrozbicki:/etc/openvpn/easy-rsa#
```

Obraz 34. Kopiowanie kluczy i certyfikatów klienta oraz CA do katalogu klienta OpenVPN.

Wszystkie wymagane pliki zostały skopiowane do katalogu /etc/openvpn/client/keys:

- Żądanie certyfikatu klienta clientrozbicki.req
- Klucz prywatny klienta clientrozbicki.key
- Certyfikat klienta clientrozbicki.crt
- Certyfikat urzędu certyfikacji ca.crt

III.3. Konfiguracja serwera OpenVPN

```
root@serverrozbicki:/etc/openvpn/easy-rsa# cp /usr/share/doc/openvpn/examples/sample–config–files/se
rver.conf.gz /etc/openvpn/server/serverrozbicki.conf.gz
root@serverrozbicki:/etc/openvpn/easy–rsa# gzip –d /etc/openvpn/server/serverrozbicki.conf.gz
root@serverrozbicki:/etc/openvpn/easy–rsa#
```

Obraz 35. Skopiowanie i rozpakowanie przykładowego pliku konfiguracyjnego serwera OpenVPN.

Wykonano polecenie sudo cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz /etc/openvpn/server/serverrozbicki.conf.gz, które skopiowało przykładowy plik konfiguracyjny serwera OpenVPN do katalogu serwera. Następnie plik został rozpakowany za pomocą sudo gzip -d /etc/openvpn/server/serverrozbicki.conf.gz, przygotowując go do edycji i dostosowania do konkretnej konfiguracji serwera VPN.

```
root@serverrozbicki:/etc/openvpn/easy–rsa# openvpn ––genkey ––secret ta.key
root@serverrozbicki:/etc/openvpn/easy–rsa# cp /etc/openvpn/easy–rsa/ta.key /etc/openvpn/server
root@serverrozbicki:/etc/openvpn/easy–rsa# cp /etc/openvpn/easy–rsa/ta.key /etc/openvpn/client/keys
root@serverrozbicki:/etc/openvpn/easy–rsa# chown rozbicki.rozbicki /etc/openvpn/client/keys/*
root@serverrozbicki:/etc/openvpn/easy–rsa# _
```

Obraz 36. Generowanie i dystrybucja współdzielonego klucza TLS-crypt.

Za pomocą polecenia openvpn --genkey --secret ta.key wygenerowano współdzielony klucz TLS-crypt (ta.key), który zabezpiecza proces uzgadniania połączenia, chroniąc przed atakami typu DoS. Klucz ten został skopiowany do katalogów:

- /etc/openvpn/server dla serwera OpenVPN
- /etc/openvpn/client/keys dla klienta

Następnie za pomocą polecenia chown rozbicki.rozbicki /etc/openvpn/client/keys/* zmieniono właściciela plików w katalogu klienta, co zapewnia odpowiedni dostęp do kluczy prywatnych i certyfikatów podczas konfiguracji i użytkowania.

```
root@serverrozbicki:/etc/openvpn/easy-rsa# cd /etc/openvpn/server
root@serverrozbicki:/etc/openvpn/server# nano serverrozbicki.conf
```

Obraz 37. Edycja pliku konfiguracyjnego serwera OpenVPN.

Za pomocą polecenia nano serverrozbicki.conf otwarto do edycji plik konfiguracyjny serwera OpenVPN znajdujący się w katalogu /etc/openvpn/server. W pliku tym należy dostosować kluczowe ustawienia, takie jak:

- Lokalizacja kluczy i certyfikatów (ca.crt, server.crt, server.key, ta.key)
- Port nasłuchiwania serwera i protokół (domyślnie: UDP)
- Sieć wewnętrzna VPN i zakres adresów IP dla klientów
- Zasady routingu i przekazywania ruchu

```
GNU nano 4.8
                                        serverrozbicki.conf
# Ping every 10 seconds, assume that remote
# peer is down if no ping received during
# a 120 second time period.
keepalive 10 120
# For extra security beyond that provided
# by SSL/TLS, create an "HMAC firewall"
# to help block DoS attacks and UDP port flooding.
# Generate with:
   openvpn --genkey --secret ta.key
# The server and each client must have
# a copy of this key.
# The second parameter should be 'O'
# on the server and '1' on the clients.
tls-crypt ta.key_
# Select a cryptographic cipher.
# This config item must be copied to
# the client config file as well.
# 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
# Enable compression on the VPN link and push the
# option to the client (v2.4+ only, for earlier
# versions see below)
;compress lz4–v2
;push "compress lz4–v2"
# For compression compatible with older clients use comp–lzo
# Select a cryptographic cipher.
# This config item must be copied to
# the client config file as well.
# 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–GCM_
auth SHA256
# Enable compression on the VPN link and push the
```

option to the client (v2.4+ only, for earlier

versions see below)

;push "compress 1z4-v2"

;compress lz4–v2

```
and each of the client certificates.
# Any X509 key management system can be used.
# OpenVPN can also use a PKCS #12 formatted key file
# (see "pkcs12" directive in man page).
ca ca.crt
cert server.crt
key server.key # This file should be kept secret
# Diffie hellman parameters.
# Generate your own with:
    openss1 dhparam -out dh2048.pem 2048
|dh none_
# Network topology
# Should be subnet (addressing via IP)
# unless Windows clients v2.0.9 and lower have to
# be supported (then net30, i.e. a /30 per client)
# Defaults to net30 (not recommended)
;topology subnet
```

```
# It's a good idea to reduce the OpenVPN
# daemon's privileges after initialization.
# You can uncomment this out on
# non–Windows systems.
user nobody
group nogroup
# The persist options will try to avoid
# accessing certain resources on restart
# that may no longer be accessible because
# of the privilege downgrade.
persist-key
persist–tun
# Output a short status file showing
# current connections, truncated
# and rewritten every minute.
status /var/log/openvpn/openvpn–status.log
```

```
#
# Any X509 key management system can be used.
# OpenVPN can also use a PKCS #12 formatted key file
# (see "pkcs12" directive in man page).
ca ca.crt
cert serverrozbicki.crt
key serverrozbicki.key
# Diffie hellman parameters.
# Generate your own with:
# openssl dhparam —out dh2048.pem 2048
dh none
```

```
root@serverrozbicki:/etc# nano sysctl.conf
```

Obraz 38. Edycja pliku sysctl.conf w celu włączenia przekazywania ruchu.

Otworzono plik konfiguracyjny systemu sysctl.conf za pomocą polecenia nano sysctl.conf. W pliku dodajemy linijkę: net.ipv4.ip forward=1

```
GNU nano 4.8
                                                                                          Modified
                                              sysctl.conf
 See /etc/sysctl.d/ for additional system variables.
 See sysctl.conf (5) for information.
#kernel.domainname = example.com
# Uncomment the following to stop low–level messages on console
#kernel.printk = 3 4 1 3
# Functions previously found in netbase
# Uncomment the next two lines to enable Spoof protection (reverse–path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1
# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1
net.ipv4.ip_forward=1
  Enabling this option disables Stateless Address Autoconfiguration
#net.ipv6.conf.all.forwarding=1
^G Get Help
^X Exit
              ^O Write Out
^R Read File
                                                           Justify
                                                                      îC Cur Pos
                                                                                    M-U Undo
                              Where Is
                                             Cut Text
                                             Paste Text
```

Pozwala to na włączenie funkcji przekazywania ruchu sieciowego (IP forwarding) między interfejsami, co jest niezbędne do działania serwera OpenVPN jako bramy umożliwiającej dostęp do sieci lokalnej.

```
root@serverrozbicki:/etc# sudo sysctl -p /etc/sysctl.conf
net.ipv4.ip_forward = 1
root@serverrozbicki:/etc# _
```

Obraz 39. Przeładowanie ustawień sysctl w celu włączenia przekazywania ruchu sieciowego.

Polecenie sudo sysctl -p /etc/sysctl.conf wczytuje nowe ustawienia systemowe z pliku konfiguracyjnego sysctl.conf. W wyniku działania komendy włączono przekazywanie ruchu IPv4 (IP forwarding), co jest kluczowe dla prawidłowego działania serwera OpenVPN. Linia net.ipv4.ip_forward = 1 potwierdza, że funkcja została aktywowana.

III.4. Konfiguracja zapory sieciowej i routingu

```
root@serverrozbicki:/etc# ip route list default
default via 192.168.0.1 dev enpOs3 proto dhcp src 192.168.0.157 metric 100
root@serverrozbicki:/etc# _
```

Obraz 40. Identyfikacja interfejsu sieci publicznej dla ruchu wychodzącego.

Polecenie ip route list default pozwala na wyświetlenie domyślnej trasy routingu, wskazując interfejs sieciowy obsługujący ruch wychodzący do sieci publicznej. W wyniku działania komendy wykryto, że interfejsem publicznym jest enp0s3, przez który ruch przechodzi przez bramę 192.168.0.1. Ten interfejs będzie używany w dalszej konfiguracji zapory sieciowej (ufw).

```
root@serverrozbicki:/etc/ufw# nano before.rules
```

Obraz 41. Edycja reguł zapory przed ich załadowaniem.

```
GNU nano 4.8
                                                       before.rules
                                                                                                           Modified
 rules.before
# Rules that should be run before the ufw command line added rules. Custom
    ufw-before-output
    ufw-before-forward
# START OPENVPN RULES
# NATT table rules
∗nat
:POSTROUTING ACCEPT [0:0]
#Allow traffic from OpenVPN client ens33
-A POSTROUTING -s 10.8.0.0/8 -o ens33 -j MASQUERADE
COMMIT
# Don't delete these required lines, otherwise there will be errors
*filter
:ufw-before-input - [0:0]
:ufw-before-output – [0:0]
:ufw-before-forward – [0:0]
:ufw-not-local – [0:0]
# End required lines
# allow all on loopback
–A ufw–before–input –i lo –j ACCEPT
–A ufw–before–output –o lo –j ACCEPT
# quickly process packets for which we already have a connection
−A ufw–before–input –m conntrack ––ctstate RELATED,ESTABLISHED –j ACCEPT
                                                                                       Cur Pos M–U Undo
Go To Line M–E Redo
                   Write Out
                                    Where Is
                                                     Cut Text
                                                                      Justify
To Spell
                                                                                       Cur Pos
   Get Help
                                     Replace
                                                     Paste Text
```

Plik /etc/ufw/before.rules: Dodane zostały reguły NAT oraz przekazywanie ruchu z OpenVPN do interfejsu sieciowego. Reguły:

Dodanie reguły POSTROUTING pozwalającej na maskowanie adresów klientów VPN podczas przekazywania ruchu do sieci publicznej.

```
GNU nano 4.8
                                                                                                                                                  Modified
   /etc/default/ufw
  Set to yes to apply rules to support IPv6 (no means only IPv6 on loopback accepted). You will need to 'disable' and then 'enable' the firewall for
   the changes to take affect.
# Set the default input policy to ACCEPT, DROP, or REJECT. Please note that if
# you change this you will most likely want to adjust your rules.
DEFAULT_INPUT_POLICY="DROP"
# Set the default output policy to ACCEPT, DROP, or REJECT. Please note that if
# 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
# note that setting this to ACCEPT may be a security risk. See 'man ufw' for
# details
DEFAULT_APPLICATION_POLICY="SKIP"
# By default, ufw only touches its own chains. Set this to 'yes' to have ufw
# manage the built–in chains too. Warning: setting this to 'yes' will break
MANAGE_BUILTINS=no
   IPT backend
    Get Help
                           Write Out
                                                  Where Is
                                                                         Cut Text
                           Read File
                                                  Replace
                                                                                                                       Go To Line
                                                                                                                                               Redo
```

Plik /etc/default/ufw:

- Ustawiono politykę dla ruchu przychodzącego na DROP, co oznacza, że niezdefiniowany ruch będzie blokowany.
- Polityki OUTPUT i FORWARD są ustawione na ACCEPT, pozwalając na przekazywanie ruchu i dostęp do sieci.

```
root@serverrozbicki:/etc/default# ufw allow 1994/udp
Rule added
Rule added (v6)
root@serverrozbicki:/etc/default# ufw allow OpenSSH
Skipping adding existing rule
Skipping adding existing rule (v6)
root@serverrozbicki:/etc/default#
```

Obraz 42. Konfiguracja portów

Port 1194/UDP: Jest on domyślnie używany przez OpenVPN, więc jego dostępność umożliwi połączenia klientów z serwerem.

Port dla OpenSSH: Zapewniono dostępność dla połaczeń SSH w celu zdalnej administracji.

```
root@serverrozbicki:/etc/default# ufw disable
Firewall stopped and disabled on system startup
root@serverrozbicki:/etc/default# ufw enable
Firewall is active and enabled on system startup
root@serverrozbicki:/etc/default# _
```

Obraz 43. Reset zapory sieciowej dla zastosowania reguł

```
root@serverrozbicki:/etc/default# ufw status
Status: active
Τo
                            Action
                                        From
OpenSSH
                            ALLOW
                                        Anywhere
1994/udp
                            ALLOW
                                        Anywhere
OpenSSH (v6)
                            ALLOW
                                        Anywhere (v6)
1994/udp (v6)
                            ALLOW
                                        Anywhere (v6)
root@serverrozbicki:/etc/default# _
```

Obraz 44. Status zapory UFW z otwartymi portami OpenSSH i OpenVPN.

Widok aktywnej zapory sieciowej (UFW) na serwerze, z otwartymi portami 22 (OpenSSH) i 1194/UDP (OpenVPN), umożliwiającymi zdalny dostęp i działanie VPN.

```
root@serverrozbicki:/etc/default# systemctl –f enable openvpn—server@serverrozbicki.service
Created symlink /etc/systemd/system/multi—user.target.wants/openvpn—server@serverrozbicki.service →
/lib/systemd/system/openvpn—server@.service.
root@serverrozbicki:/etc/default# _
```

Obraz 45. Start serwera OpenVPN

Polecenie systemctl -f enable openvpn-server@serverrozbicki.service tworzy symlink i konfiguruje automatyczne uruchamianie serwera OpenVPN (instancji nazwanej) podczas startu systemu, umożliwiając stabilną obsługę połączeń VPN.

```
root@serverrozbicki:/etc/default# systemctl start openvpn–server@serverrozbicki.service
root@serverrozbicki:/etc/default#
```

Obraz 46. Uruchamianie serwera OpenVPN

Polecenie systemctl start openvpn-server@servernazwisko.service inicjuje działanie instancji serwera OpenVPN, co pozwala na natychmiastowe rozpoczęcie obsługi połączeń VPN przez skonfigurowaną usługę.

```
root@serverrozbicki:/etc/default# systemctl status openvpn-server@serverrozbicki.service

• openvpn-server@serverrozbicki.service - OpenVPN service for serverrozbicki

Loaded: loaded (/lib/systemd/system/openvpn-server@.service; enabled; vendor preset: enabled)

Active: active (running) since Mon 2025-01-13 20:02:26 UTC; 2min 50s ago

Docs: man:openvpn(8)

https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
https://community.openvpn.net/openvpn/wiki/HOWTO

Main PID: 18593 (openvpn)

Status: "Initialization Sequence Completed"

Tasks: 1 (limit: 4536)

Memory: 1.1M

CGroup: /system.slice/system-openvpn\x2dserver.slice/openvpn-server@serverrozbicki.service

18593 /usr/sbin/openvpn --status /run/openvpn-server/status-serverrozbicki.log --status

Jan 13 20:02:26 serverrozbicki openvpn[18593]: Could not determine IPv4/IPv6 protocol. Using AF_INET
Jan 13 20:02:26 serverrozbicki openvpn[18593]: Socket Buffers: R=[212992->212992] S=[212992->212992]

Jan 13 20:02:26 serverrozbicki openvpn[18593]: UDPv4 link local (bound): [AF_INET][undef]:1194

Jan 13 20:02:26 serverrozbicki openvpn[18593]: UDPv4 link remote: [AF_UNSPEC]

Jan 13 20:02:26 serverrozbicki openvpn[18593]: UDPv4 link remote: [AF_UNSPEC]

Jan 13 20:02:26 serverrozbicki openvpn[18593]: UDPv4 link remote: [AF_UNSPEC]

Jan 13 20:02:26 serverrozbicki openvpn[18593]: IDPv4 link remote: [AF_UNSPEC]

Jan 13 20:02:26 serverrozbicki openvpn[18593]: IDPv4 link remote: [AF_UNSPEC]

Jan 13 20:02:26 serverrozbicki openvpn[18593]: IFCONFIG POOL: base=10.8.0.4 size=62, ipv6=0

Jan 13 20:02:26 serverrozbicki openvpn[18593]: IFCONFIG POOL: base=10.8.0.4 size=62, ipv6=0

Jan 13 20:02:26 serverrozbicki openvpn[18593]: IFCONFIG POOL: base=10.8.0.4 size=62, ipv6=0

Jan 13 20:02:26 serverrozbicki openvpn[18593]: IFCONFIG POOL: base=10.8.0.4 size=62, ipv6=0

Jan 13 20:02:26 serverrozbicki openvpn[18593]: IFCONFIG POOL: base=10.8.0.4 size=62, ipv6=0

Jan 13 20:02:26 serverrozbicki openvpn[18593]: IFCONFIG POOL: base=10.8.0.4 size=62, ipv6=0

Jan 13 20:02:26 serverrozbicki ope
```

Obraz 47. Sprawdzanie statusu serwera OpenVPN

Polecenie systemctl status openvpn-server@serverrozbicki.service wyświetla szczegóły działania usługi OpenVPN, takie jak jej bieżący status, czas działania oraz ewentualne komunikaty diagnostyczne, co pozwala zweryfikować, czy serwer VPN działa poprawnie. W tym przypadku usługa jest aktywna i działa (status: active (running)).

III.5. Konfiguracja klienta OpenVPN

```
root@serverrozbicki:/etc/default# cd /etc/openvpn/client
root@serverrozbicki:/etc/openvpn/client# mkdir files
root@serverrozbicki:/etc/openvpn/client# _
```

Obraz 48. Tworzenie katalogu dla plików konfiguracyjnych klienta OpenVPN

W ramach konfiguracji klienta OpenVPN, w katalogu /etc/openvpn/client utworzono podkatalog files przy użyciu polecenia mkdir files. Katalog ten będzie przechowywać niezbędne pliki konfiguracyjne i certyfikaty potrzebne do ustanowienia połączenia VPN.

```
root@serverrozbicki:/etc/openvpn/client# cp /usr/share/doc/openvpn/examples/sample-config-files/clie
nt.conf /etc/openvpn/client/baserozbicki.conf
root@serverrozbicki:/etc/openvpn/client# _
```

Obraz 49. Kopiowanie domyślnego pliku konfiguracyjnego klienta OpenVPN

Skopiowano domyślny plik konfiguracyjny klienta OpenVPN z lokalizacji /usr/share/doc/openvpn/examples/sample-config-files/client.conf do katalogu /etc/openvpn/client pod nową nazwą baserozbicki.conf. Plik ten posłuży jako podstawa do konfiguracji klienta VPN.

```
root@serverrozbicki:/etc/openvpn/client# nano baserozbicki.conf
```

Obraz 50. Edycja pliku baserozbicki.conf

Edycja pliku konfiguracyjnym klienta OpenVPN:

```
# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
remote your–server–ip 1194
```

```
# Are we connecting to a TCP or
# UDP server? Use the same setting as
# on the server.
proto udp_
```

```
# SSL/TLS parms.
# See the server config file for more
# description. It's best to use
# a separate .crt/.key file pair
# for each client. A single ca
# file can be used for all clients.
;ca ca.crt
;cert client.crt
;key client.key
```

```
# 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-GCM
auth SHA256

# 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
key-direction 1_
```

W edytowanym pliku konfiguracyjnym klienta OpenVPN basenazwisko.conf wprowadzono następujące kluczowe ustawienia:

1. Adres zdalny serwera i port (remote your-server-ip 1194):

 Ustalono IP lub nazwę hosta serwera VPN, na który klient ma się łączyć. Port ustawiono na domyślny 1194 dla protokołu UDP.

2. Protokół komunikacji (proto udp):

 Określono użycie protokołu UDP, który jest szybszy i bardziej efektywny niż TCP w połączeniach VPN.

3. Ścieżki do certyfikatów i klucza prywatnego:

- W tej sekcji pliku (po odkomentowaniu) klient użyje klucza prywatnego i certyfikatów do uwierzytelniania.
- 4. **Kryptografia** (cipher AES-256-GCM oraz auth SHA256):
 - Zastosowano szyfrowanie AES-256-GCM, zapewniające wysoki poziom bezpieczeństwa.
 - Weryfikacja integralności danych odbywa się za pomocą algorytmu SHA256.

5. **Kierunek klucza** (key-direction 1):

 Konfiguracja odpowiada kierunkowi klucza TLS, wymaganym do poprawnego działania funkcji tls-auth lub tls-crypt.

6. **Dziennik i logi** (verb 3, mute 20):

 Ustawienia poziomu szczegółowości logów (verb 3) oraz wyciszenie powtarzających się wiadomości logów (mute 20).

Podsumowując, konfiguracja ta umożliwia bezpieczne połączenie klienta z serwerem OpenVPN, uwierzytelnienie za pomocą klucza i certyfikatów oraz zapewnia zaawansowaną ochronę danych dzięki protokołowi UDP i szyfrowaniu AES-256.

root@serverrozbicki:/etc/openvpn/client# touch make_config_rozbicki.sh root@serverrozbicki:/etc/openvpn/client# nano make_config_rozbicki.sh _

Obraz 51. Utworzenie i edycja pliku konfiguracyjnego skryptu klienta OpenVPN

Opis: Skrypt automatycznie generuje plik konfiguracyjny .ovpn klienta OpenVPN, łącząc bazowy plik konfiguracyjny (baserozbicki.conf) z odpowiednimi certyfikatami, kluczami i ustawieniami TLS.

Obraz 52. Struktura skryptu make config rozbicki.sh

Główne elementy:

- KEY_DIR: Katalog przechowujący klucze i certyfikaty.
- OUTPUT DIR: Katalog, w którym wygenerowane zostaną pliki .ovpn.
- BASE_CONFIG: Ścieżka do bazowej konfiguracji klienta.
- cat: Komenda służąca do odczytu i łączenia zawartości plików konfiguracyjnych i certyfikatów w jeden plik wyjściowy .ovpn.

```
root@serverrozbicki:/etc/openvpn/client# chmod 700 /etc/openvpn/client/make_config_rozbicki.sh
root@serverrozbicki:/etc/openvpn/client# _
```

Obraz 53. Nadanie odpowiednich praw dostępu do skryptu make_config_rozbicki.sh

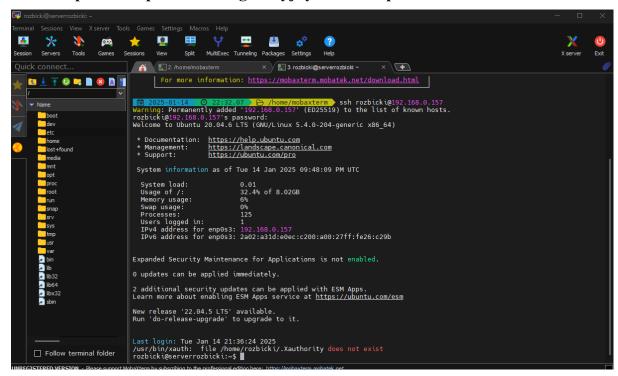
To ustawienie gwarantuje, że tylko właściciel (administrator) może modyfikować lub uruchamiać ten plik, co zwiększa bezpieczeństwo wrażliwych operacji na kluczach i konfiguracjach klienta OpenVPN.

```
root@serverrozbicki:/etc/openvpn/client# ./make_config_rozbicki.sh clientrozbicki
root@serverrozbicki:/etc/openvpn/client# _
```

Obraz 54. Uruchomienie skryptu make config rozbicki.sh

To polecenie powoduje wygenerowanie pliku konfiguracyjnego .ovpn dla klienta o nazwie clientrozbicki w katalogu docelowym (np. /etc/openvpn/client/files). Skrypt ten łączy bazowy plik konfiguracyjny z odpowiednimi certyfikatami i kluczami, tworząc kompletną konfigurację VPN gotową do użycia przez klienta.

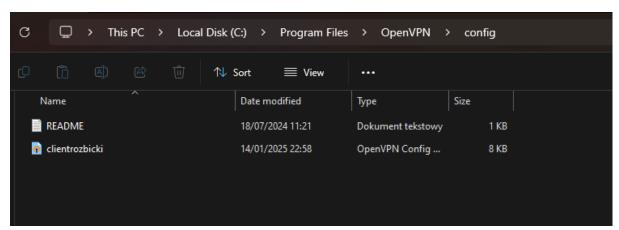
III.6. Kopiowanie plików konfiguracyjnych i zabezpieczenia



Obraz 55. Pobranie pliku konfiguracyjnego za pomocą programu MobaXterm

Otworzono MobaXterm i nawiązano połączenie SSH z serwerem 192.168.0.157. Połączenie zostało ustanowione, co umożliwia bezpośrednią interakcję z serwerem. Otworzono zakładkę SFTP/Explorer po lewej stronie interfejsu MobaXterm. Przejrzano katalogi serwera, przechodząc do lokalizacji, w której zapisano wygenerowany plik konfiguracyjny .ovpn (np. /etc/openvpn/client/files).

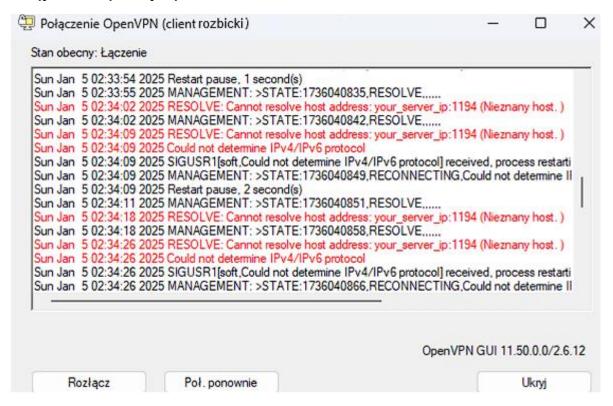
Skopiowano plik .ovpn na lokalną maszynę Windows, przeciągając go z eksploratora serwera lub używając opcji Download dostępnej w menu kontekstowym. Plik jest teraz dostępny na komputerze lokalnym do zaimportowania w kliencie OpenVPN.



Obraz 56. Umieszczenie pliku konfiguracyjnego do folderu OpenVPN

kopiowano plik **clientrozbicki.ovpn** do folderu konfiguracyjnego OpenVPN na Windowsie: **C:\Program Files\OpenVPN\config.** Plik jest teraz gotowy do użycia w kliencie OpenVPN.

Następnym krokiem jest uruchomienie OpenVPN GUI, wybranie odpowiedniego profilu, a następnie nawiązanie połączenia z serwerem VPN.



Komunikat błędu wskazuje na problem z rozpoznaniem adresu serwera VPN, co jest spowodowane błędnym lub nieprawidłowym wpisem w sekcji remote pliku clientrozbicki.ovpn.

Sun Jan	5 02:42:00	2025 MAN	AGEMENT: CN	AD 'hold release'	C.				T
				ng recently used		ss: [AF INET]	192.16	8.100.106	
				penVPN Data Cl					
			link local: (not						
Sun Jan	5 02:42:00	2025 UDP	link remote: [Al	F_INET]192.168	.100.106:119	4			
Sun Jan	5 02:42:00	2025 MAN	AGEMENT: >S	TATE:17360413	320,WAIT				
				TATE:17360413					
Sun Jan	5 02:42:00	2025 TLS:	Initial packet fr	om [AF_INET]19	2.168.100.10	6:1194, sid=0	14914f2	2 456d2e!	
oun Jan	5 02:42:00	2025 VERI	FY OK: depth=	1. CN=CA					
Sun Jan	5 02:42:00	2025 VERI	FY KU OK	Control of March 1985					
Sim Inn	5 02:42:00	2025 Valida		extended key u	sage				
Sun Jan	5 02:42:00	2025 ++ Ce		KU (str) TLS Wel	Server Author	entication, exp	ects TI	LS Web S	
Sun Jan Sun Jan	5 02:42:00 5 02:42:00	2025 ++ Ce 2025 VERI	FY EKU OK	68668	Server Auth	entication, exp	ects Tl	LS Web S	
Sun Jan Sun Jan Sun Jan	5 02:42:00 5 02:42:00 5 02:42:00	2025 ++ Ce 2025 VERI 2025 VERI	FY EKU OK FY OK: depth=	0. CN=server					
Sun Jan Sun Jan Sun Jan Sun Jan	5 02:42:00 5 02:42:00 5 02:42:00 5 02:42:00	2025 ++ Ce 2025 VERI 2025 VERI 2025 Contr	FY EKU OK FY OK: depth= ol Channel: TL	0, CN=server Sv1.3, cipher TL	Sv1.3 TLS_A	ES_256_GCM	_SHA3	384, peer	
Sun Jan Sun Jan Sun Jan Sun Jan	5 02:42:00 5 02:42:00 5 02:42:00 5 02:42:00	2025 ++ Ce 2025 VERI 2025 VERI 2025 Contr	FY EKU OK FY OK: depth= ol Channel: TL	0, CN=server Sv1.3, cipher TL	Sv1.3 TLS_A	ES_256_GCM	_SHA3	384, peer	
Sun Jan Sun Jan Sun Jan Sun Jan	5 02:42:00 5 02:42:00 5 02:42:00 5 02:42:00	2025 ++ Ce 2025 VERI 2025 VERI 2025 Contr	FY EKU OK FY OK: depth= ol Channel: TL	0, CN=server Sv1.3, cipher TL	Sv1.3 TLS_A	ES_256_GCM	_SHA3	384, peer	
Sun Jan Sun Jan Sun Jan Sun Jan Sun Jan	5 02:42:00 5 02:42:00 5 02:42:00 5 02:42:00 5 02:42:00) 2025 ++ Ce) 2025 VERI) 2025 VERI) 2025 Contr) 2025 [serve	FY EKU OK FY OK: depth= ol Channel: TL	0, CN=server Sv1.3, cipher TL	Sv1.3 TLS_A	ES_256_GCM	_SHA3	384, peer	
Sun Jan Sun Jan Sun Jan Sun Jan Sun Jan	5 02:42:00 5 02:42:00 5 02:42:00 5 02:42:00 5 02:42:00) 2025 ++ Ce) 2025 VERI) 2025 VERI) 2025 Contr) 2025 [serve	FY EKU OK FY OK: depth= ol Channel: TL	0, CN=server Sv1.3, cipher TL	Sv1.3 TLS_A	ES_256_GCM	_SHA3	384, peer	
Sun Jan Sun Jan Sun Jan Sun Jan	5 02:42:00 5 02:42:00 5 02:42:00 5 02:42:00	2025 ++ Ce 2025 VERI 2025 VERI 2025 Contr	FY EKU OK FY OK: depth= ol Channel: TL	0, CN=server Sv1.3, cipher TL	Sv1.3 TLS_A	ES_2	56_GCM	56_GCM_SHA	tion, expects TLS Web S 56_GCM_SHA384, peer 192.168.100.106:1194
an an an	5 02:42:00 5 02:42:00 5 02:42:00 5 02:42:00 5 02:42:00 y IP: 10.8.0) 2025 ++ Ce) 2025 VERI) 2025 VERI) 2025 Contr) 2025 [serve	FY EKU OK FY OK: depth= ol Channel: TL erserafin] Peer	0, CN=server Sv1.3, cipher TL	Sv1.3 TLS_A ted with [AF_	ES_256_GCM	1_SHA3 .100.10	384	peer 194
5 02:42:00 5 02:42:00 5 02:42:00 5 02:42:00 5 02:42:00 y IP: 10.8.0.6	0	2025 ++ Ce 2025 VERI 2025 VERI 2025 Contr 2025 [serve	FY EKU OK FY OK: depth= ol Channel: TL erserafin] Peer	0, CN=server Sv1.3, cipher TL Connection Initia	Sv1.3 TLS_A ted with [AF_	ES_256_GCM INET]192.168	1_SHA3 .100.10	384, peer 06:1194	

Po zmianie remote 192.168.0.157 1194 na obrazie widzimy okno połączenia OpenVPN z komunikatem Stan obecny: Połączony, co oznacza, że tunel VPN został pomyślnie zestawiony. Ważne wpisy w logach:

- VERIFY OK dla certyfikatów: CN=caserozbicki oraz CN=serverrozbicki weryfikacja certyfikatów przebiegła pomyślnie.
- TLSv1.3, cipher TLS_AES_256_GCM_SHA384 Połączenie korzysta z szyfrowania AES-256-GCM w ramach protokołu TLS 1.3, zapewniając wysoki poziom bezpieczeństwa.
- Przyznany IP: 10.8.0.6 Klientowi przydzielono adres IP z puli sieci VPN.
- Bajtów pobranych/wysłanych Wskazuje na aktywną wymianę danych.

Podsumowanie: VPN działa poprawnie, a wszystkie kluczowe elementy (szyfrowanie, certyfikaty i przydział IP) zostały skonfigurowane bez błędów.

Rozdział IV. Podsumowanie

Cel projektu:

Celem tego projektu było skonfigurowanie w pełni działającego środowiska sieciowego opartego na technologii OpenVPN, zapewniającego bezpieczny zdalny dostęp do sieci poprzez szyfrowane połączenie VPN. W projekcie przeprowadzono szczegółowe etapy instalacji, konfiguracji i weryfikacji poprawności działania zarówno po stronie serwera, jak i klienta, co pozwoliło na stworzenie stabilnego i bezpiecznego połączenia.

I. Przygotowanie środowiska

Pierwszym krokiem było zainstalowanie serwera OpenVPN na maszynie z systemem Ubuntu. W tym celu zainstalowano wymagane pakiety, takie jak **openvpn** i **easy-rsa**, które posłużyły do generowania materiałów kryptograficznych niezbędnych do utworzenia certyfikatów bezpieczeństwa.

II. Utworzenie CA i materiałów kryptograficznych

Jednym z kluczowych etapów projektu było utworzenie urzędu certyfikacji (CA – Certificate Authority) oraz wygenerowanie materiałów kryptograficznych, które zapewniają integralność i poufność przesyłanych danych:

- **Utworzenie CA**: Wygenerowano główny certyfikat CA przy użyciu narzędzia Easy-RSA.
- **Certyfikat serwera**: Wygenerowano żądanie certyfikatu serwera, które zostało podpisane przez CA, aby zagwarantować zaufanie i autentyczność.

- Certyfikat klienta: Analogicznie wygenerowano i podpisano certyfikat klienta, co umożliwia jego autoryzację podczas nawiązywania połączenia.
- **Klucz TLS-crypt**: Wygenerowano dodatkowy klucz TLS-crypt, który zabezpiecza proces negocjacji połączenia VPN przed atakami typu DoS i MitM (Man-in-the-Middle).

III. Konfiguracja serwera OpenVPN

Serwer OpenVPN został skonfigurowany w oparciu o dostarczony szablon konfiguracyjny, który odpowiednio dostosowano do potrzeb projektu. Najważniejsze ustawienia obejmowały:

- Port i protokół: OpenVPN został skonfigurowany na porcie UDP (domyślnie 1194).
- Certyfikaty: Wskazano ścieżki do wygenerowanych certyfikatów i kluczy.
- Ustawienia sieciowe: Zdefiniowano zakres adresów IP przydzielanych klientom VPN oraz włączono przekierowanie ruchu (IP forwarding).
- Ochrona TLS: Włączono dodatkową warstwę bezpieczeństwa przy użyciu klucza TLS-crypt.

IV. Konfiguracja zapory sieciowej i routingu

W celu zapewnienia bezpiecznego dostępu do serwera OpenVPN i prawidłowego routingu ruchu sieciowego skonfigurowano zaporę sieciową UFW (Uncomplicated Firewall):

- 1. **Otwarcie odpowiednich portów**: Umożliwiono ruch na porcie 1194/UDP oraz dostęp SSH do serwera.
- 2. **Przekierowanie ruchu**: W pliku /etc/ufw/before.rules skonfigurowano reguły NAT, aby umożliwić przekazywanie ruchu z klientów VPN do zewnętrznej sieci.
- 3. **Włączenie i aktywacja UFW**: Zapora została włączona i ustawiona na automatyczne uruchamianie wraz ze startem systemu.

V. Konfiguracja klienta OpenVPN

Kolejnym etapem była konfiguracja klienta VPN. W tym celu:

- 1. **Skopiowano przykładowy plik konfiguracyjny klienta** do odpowiedniego katalogu na serwerze.
- 2. **Dostosowano konfigurację**: Wskazano adres serwera, protokół, certyfikaty i klucze oraz zabezpieczenia TLS.

- 3. **Stworzono skrypt automatyzujący generację plików klienckich**: Skrypt w prosty sposób generował pliki konfiguracyjne **.ovpn** dla każdego klienta, łącząc wszystkie niezbędne klucze i certyfikaty w jednym pliku.
- 4. Przeniesiono plik konfiguracyjny klienta na maszynę z systemem Windows w celu jego uruchomienia.

VI. Weryfikacja poprawności konfiguracji

Weryfikacja poprawności została przeprowadzona poprzez:

- Uruchomienie usługi OpenVPN na serwerze i sprawdzenie jej statusu.
- Skopiowanie pliku konfiguracyjnego klienta na maszynę kliencką.
- **Uruchomienie połączenia VPN na kliencie** i monitorowanie logów w poszukiwaniu ewentualnych błędów.
- **Potwierdzenie poprawnego zestawienia połączenia** adres IP 10.8.0.0/24 przydzielony klientowi potwierdził, że tunel VPN działa poprawnie.

Wnioski

Projekt zakończył się sukcesem, a jego rezultat to w pełni funkcjonalne środowisko OpenVPN zapewniające bezpieczny zdalny dostęp do zasobów sieciowych. Dzięki zastosowaniu certyfikatów i kluczy szyfrujących oraz dodatkowych zabezpieczeń TLS-crypt, połączenie jest dobrze zabezpieczone przed nieautoryzowanym dostępem i potencjalnymi atakami sieciowymi.

Podczas realizacji projektu można było zaobserwować znaczenie dokładnej konfiguracji sieciowej i zapory UFW, ponieważ nawet drobne błędy mogły prowadzić do problemów z połączeniem. Ostateczna weryfikacja potwierdziła poprawność wszystkich etapów konfiguracji.