

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**  
**КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

## **ВИПУСКНА РОБОТА**

**на тему:**

**«Віртуальне середовище для вивчення методик  
побудови безпечних VPN-з'єднань»**

**Завідувач  
випускаючої кафедри**

**Довбиш А.С.**

**Керівник роботи**

**Лаврик Т.В.**

**Студента групи КБ – 71**

**Басова М.В.**

**СУМИ 2021**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

Затверджую \_\_\_\_\_

Зав. кафедрою Довбиш А.С.

“ \_\_\_\_\_ ” \_\_\_\_\_ 2021 г.

**ЗАВДАННЯ**

**до випускної роботи**

Студента четвертого курсу, групи КБ-71 спеціальності “Кібербезпека”  
денної форми навчання Басова Максима Вікторовича.

**Тема: “Віртуальне середовище для вивчення методик побудови  
безпечних VPN-з'єднань”**

Затверджена наказом по СумДУ

№ \_\_\_\_\_ от \_\_\_\_\_ 2021 г.

**Зміст пояснювальної записки:** 1) аналітичний огляд принципів організації технології VPN; 2) опис основних протоколів для організації VPN та їх порівняльна характеристика; 3) вибір та опис основного програмного забезпечення для реалізації віртуального захищеного середовища; 4) розробка віртуального захищеного середовища та тестування коректності роботи мережі на всіх пристроях в системі.

Дата видачі завдання “ \_\_\_\_\_ ” \_\_\_\_\_ 2021 р.

Керівник випускної роботи \_\_\_\_\_ Лаврик Т.В.

Завдання прийняв до виконання \_\_\_\_\_ Басов М.В.

## РЕФЕРАТ

**Записка:** 49 стор., 41 рис., 2 табл., 1 додаток, 10 джерел.

**Об'єкт дослідження** — віртуально захищене середовище для передачі даних між приватними мережами через захищені канали трансляції інформації поверх публічних ліній передачі даних.

**Мета роботи** — розроблення віртуального захищеного середовища на основі протоколів, які задіяні в організації VPN мережі, налаштування системи клієнт-серверної архітектури та всіх допоміжних пристроїв, що задіяні в реалізації.

**Методи дослідження** — метод аналітичного огляду, метод порівняння та аналогій, метод моделювання.

**Результати** — розроблено віртуально захищене середовище із використанням операційної системи Linux, а саме дистрибутиву Centos 8 та утиліт, які були задіяні для реалізації OpenVPN сервера, клієнта та налаштування конфігураційних файлів. Це дало змогу об'єднати локальні мережі в єдину захищену мережу для безпечної передачі даних.

ВІРТУАЛЬНЕ СЕРЕДОВИЩЕ, VPN-МЕРЕЖА, ПУБЛІЧНА ТА  
ПРИВАТНА МЕРЕЖА, БРАНДМАУЕР, ЦЕНТР СЕРТИФІКАЦІЇ,  
LINUX, ШЛЮЗИ, VPN-ТУНЕЛЬ, КОНФІГУРАЦІЙНІ ФАЙЛИ,  
МЕРЕЖЕВІ НАЛАШТУВАННЯ.

## ЗМІСТ

ВСТУП .....	5
1. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	6
1.1 Основні принципи організації технології VPN .....	6
1.2 Порівняльна характеристика протоколів для організації VPN.....	8
1.2.1. Протокол PPTP .....	8
1.2.2. Протокол IPSec.....	9
1.2.3. Протокол L2TP/IPSec.....	10
1.2.4. Протокол OpenVPN .....	11
1.3 Постановка задачі .....	13
2. ВИБІР МЕТОДІВ РІШЕННЯ ЗАДАЧІ ТА ПІДГОТОВКА СИСТЕМИ ДО НАЛАШТУВАННЯ .....	14
2.1 Вибір основного програмного забезпечення для реалізації VPN.....	14
2.2 Вибір програмного забезпечення для віртуалізації .....	15
2.3 Встановлення та налаштування усіх сегментів мережі .....	16
2.4 Оновлення елементів системи та завантаження допоміжних утиліт ..	18
3. СТВОРЕННЯ ВІРТУАЛЬНОГО ЗАХИЩЕНОГО СЕРЕДОВИЩА .....	21
3.1 Налаштування брандмауера Centos 8 на шлюзах.....	21
3.2 Налаштування шлюзів у різних сегментах мережі .....	22
3.3 Налаштування серверу OpenVPN .....	23
3.4 Налаштування конфігураційного файлу OpenVPN .....	30
3.5 Налаштування OpenVPN клієнта.....	35
3.6 Тестування роботи віртуального середовища .....	39
ВИСНОВКИ.....	41
СПИСОК ЛІТЕРАТУРИ.....	42
ДОДАТОК А.....	43

## ВСТУП

Сьогодні в період діджиталізації одним із головних пунктів є надійний захист в інформаційному просторі інформаційного забезпечення та попередження спотворення, знищення, несанкціонованої модифікації, зловмисного отримання і використання інформації. У час розвитку новітніх технологій популярною тенденцією на більшості підприємств та великих компаній є перехід офісних працівників на віддалений режим роботи. Сьогодні корпоративні мережі є невід'ємною частиною сучасних компаній. За допомогою таких мереж можна безпечно передавати і отримувати інформацію. Даний підхід є дуже зручним, оскільки персонал не прив'язаний до конкретного місця, а може працювати з будь-якого місця планети. Однак канали передачі інформації та комп'ютери працівників стають більш вразливими для зловмисників, які хочуть заволодіти та скористатися інформацією та призвести до грошових втрат, заманювання клієнтів до своєї організації або взагалі до банкрутства. Для того, щоб забезпечити належний захист інформації каналів передачі інформації рекомендують використовувати так звані віртуальні приватні мережі (VPN) для захисту інтернет підключення до вашої корпоративної мережі, а також протидії кіберзагрозам.

Основними перевагами технології VPN є доволі дешеве впровадження, в подальшому основні затрати будуть зводитися до оплати послуг Інтернет провайдера, можливість підключення великої кількості абонентів, які знаходяться в різних кутках світу і звісно ж безпека передачі даних. Саме завдяки гнучкості та економічності, VPN активно витісняють LAN з ринку, оскільки до основного недоліку даної технології можна віднести високу вартість впровадження та неможливість підключення віддаленого персоналу. Затрати на використання та обслуговування VPN майже в три рази нижчі від логічних структур побудованих за технологією LAN.

# 1. ІНФОРМАЦІЙНИЙ ОГЛЯД

## 1.1 Основні принципи організації технології VPN

VPN (Virtual Private Network) — це віртуальна приватна мережа, яка є об'єднанням приватних систем та звичайних робочих комп'ютерів через публічне зовнішнє середовище трансляції інформації в єдину цілісну мережу, що забезпечує належний рівень захисту даних [1].

Створення віртуальної приватної мережі відбувається шляхом побудови віртуальних захищених каналів зв'язку, які створюються на базі відкритих каналів зв'язку загальнодоступної мережі. Ці віртуальні захищені канали зв'язку називаються тунелями VPN.

В основному технологія VPN використовується для об'єднання, наприклад, центрального офісу компанії з її філіалом, офісом бізнес-партнерів і віддалених користувачів для того, щоб безпечно обмінюватися інформацією через Інтернет. Також VPN використовується для забезпечення безпечного доступу до сервісів у внутрішній мережі. Це потрібно для того, щоб тільки користувачі внутрішньої мережі мали доступ до неї [2].

Основною метою технології тунелювання є інкапсуляція інформації, тобто упакування пакетів, що передаються разом зі службовими полями в новостворений пакет уже з іншими службовими полями. Слід відзначити, що сама технологія тунелювання не забезпечує надійних захист до закритої інформації, але саме завдяки тунелюванню з'являється можливість повного криптографічного захисту інкапсульованих пакетів. Для того, щоб забезпечити основні властивості інформаційної безпеки, користувач зашифровує пакети, інкапсулює їх у зовнішній пакет уже з новим IP-заголовком і надсилає його через допоміжну мережу (рис. 1.1).

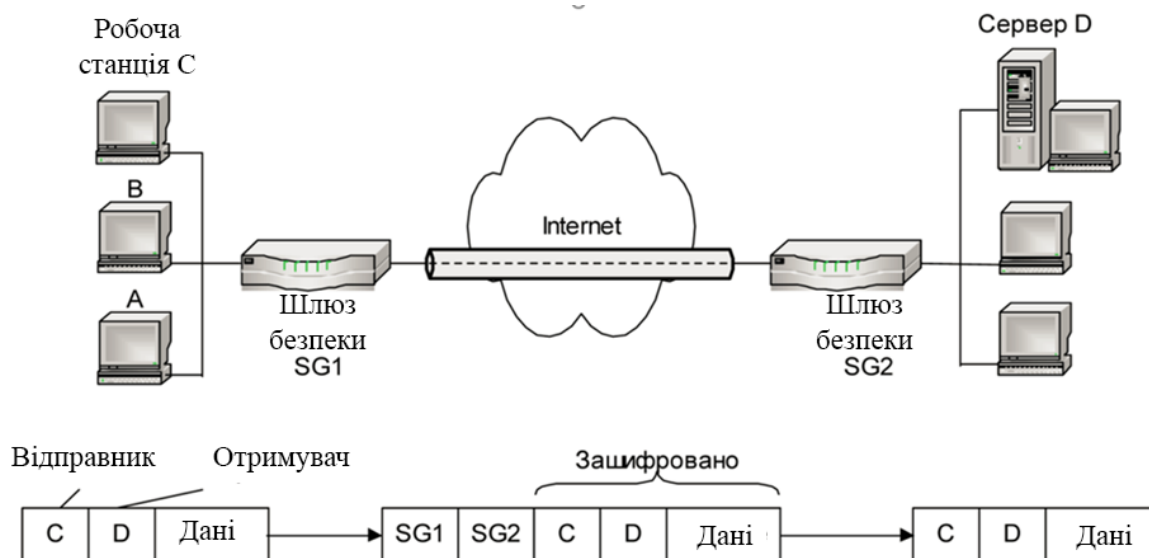


Рисунок 1.1 – Схема захищеного тунелю

Особливістю тунелювання є те, що дана технологія дозволяє зашифрувати інкапсульований пакет разом із усіма полями заголовка, а не тільки поле, в якому знаходяться дані. Шифрування є дуже важливим етапом, оскільки в заголовці пакета є поля, за допомогою яких зловмисники можуть отримати важливу інформацію. Дані з заголовку інкапсульованого пакета можуть містити інформацію про локальну структуру системи – дані про кількість підмереж і комп'ютерів та їх ір-адреси. Використовуючи такі дані зловмисник може організувати атаку на корпоративну мережу.

Пристрої VPN можуть відігравати у віртуальних приватних системах функцію VPN-клієнта, VPN-сервера або вузла безпеки VPN.

VPN-клієнт представляє собою програмний або програмно-апаратний комплекс, що реалізується в більшості випадків на базі користувацького комп'ютера. За допомогою спеціального програмного забезпечення виконується шифрування й аутентифікація трафіку в системі, це потрібно для того, щоб безпечно обмінюватися інформацією з усіма вузлами в приватній мережі.

Першочерговою метою VPN-серверу є забезпечення конфіденційності інформації компанії або звичайного користувача, а також побудові захищених з'єднань з персональними комп'ютерами і сегментами локальних систем.

Вузол безпеки VPN в свою чергу потрібен для того, щоб шифрувати та виконувати автентифікацію хостів. Розміщувати шлюз безпеки потрібно таким чином, щоб весь трафік який повинен надходити до корпоративної мережі проходив через нього[2].

## **1.2 Порівняльна характеристика протоколів для організації VPN**

### **1.2.1. Протокол PPTP**

PPTP (Point-to-Point Tunneling Protocol) – є одним із найдавніших VPN протоколів. PPTP – це тунельний протокол типу точка-точка, за рахунок якого можна реалізувати захищене з'єднання з сервером використовуючи тунелювання у звичайній локальній мережі.

Своєї популярності даний тунельний протокол набув за рахунок компанії Microsoft. Дана корпорація включила в усі версії Microsoft Windows починаючи з Windows 95 OSR2, включаючи до складу операційної системи PPTP-клієнт.

Для роботи PPTP протокол використовує два з'єднання, так само як протокол FTP, але має свої відмінності. Перше з'єднання працює за рахунок використання протоколу TCP на порту 1723, а друге з'єднання за рахунок протоколу GRE. Протокол GRE є розробкою компанії Cisco Systems, основним завданням якого є інкапсуляція пакетів мережевого рівня моделі OSI в IP-пакети. Для того, щоб користувачі, які знаходилися за технологією NAT могли з'єднуватися з сервером, протокол GRE був вдосконалений, а саме був доданий заголовок Call ID за допомогою якого маршрутизатори могли відслідковувати сеанс зв'язку між клієнтом локальної мережі до серверу який знаходився за технологією NAT і навпаки. Дана технологія використовується майже на всіх клієнтських пристроях.



З плином часу було виявлено, що протокол РРТР є не зовсім безпечним, оскільки після обриву сеансу зв'язку, для його відновлення потрібен не малий проміжок часу. Також в 2012 році була знайдена вразливість в даному протоколі, яка дозволяла розшифровувати дані за два дні, яку компанія Microsoft швидко виправила, але значною перевагою даного протоколу залишається швидкість передачі даних.

### 1.2.2. Протокол IPSec

IPSec або повна назва IP Security – це набір протоколів, основним завданням яких є шифрування, автентифікація та забезпечення цілісності даних при передачі IP-пакетів (рис. 1.2) [9].

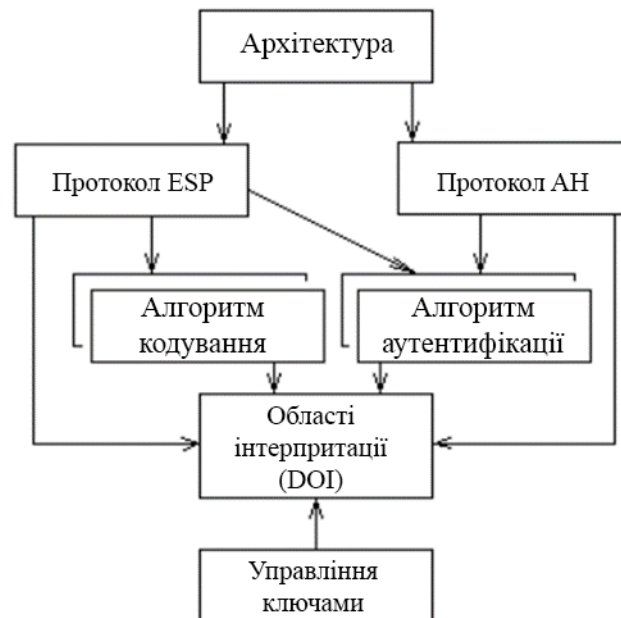


Рисунок 1.2 – Архітектура IPSec

IP Authentication Header (AH) забезпечує цілісність без встановлення з'єднання, автентифікацію та додаткову функцію захисту від повторів пакетів.

AH використовує хеш-алгоритм для обчислення значення хеша як для корисного навантаження, так і для заголовка пакета, забезпечуючи цілісність пакета. Однак це має негативний наслідок, оскільки AH не буде працювати через пристрої, в яких реалізована функція NAT, оскільки NAT змінить IP-заголовок, а сам хеш не зміниться. Пристрій, на який прийде даний пакет буде

вважати, що пакет був модифікований під час передачі, а отже його буде відхилено.

ESP (Encapsulating Security Payload) протокол також забезпечує такі три основні функції як конфіденційність, автентифікація і цілісність. Даний протокол є більш безпечним та ефективним, ніж АН оскільки хешування, яке використовується для цілісності даних, не включає в себе IP-заголовок, а отже ESP буде працювати з пристроями, які реалізують технологію NAT [3].

### **1.2.3. Протокол L2TP/IPSec**

Layer 2 Tunneling Protocol або L2TP сьогодні є майже у всіх новостворених операційних системах і працює з усіма гаджетами, які здатні працювати з технологію VPN. L2TP – протокол тунелювання другого рівня. Даний протокол є розширенням протоколу PPTP та містить в собі найкращі функції, які були в протоколі PPTP від Microsoft і L2F від Cisco Systems, оскільки ці компанії займалися розробкою L2TP протоколу.

L2TP не здатний шифрувати трафік, який проходить через нього, тому в основному його використовують в парі з IPSec. Зв'язка L2TP/IPSec вважається безпечнішою, ніж PPTP, оскільки в L2TP/IPSec використовуються два види шифрування, а саме 3DES та AES, але в більшості випадків використовується AES.

Використання протоколів у парі має свої недоліки. Одним з цих недоліків є подвійне розміщення даних в пакетах, яке погано впливає на ефективність передачі даних. Також L2TP використовує 500 UDP-порт, який часто блокується брандмауером.

Основними компонентами L2TP є L2TP Access Concentrator, завданням якого є завершення сеансу, та мережевим сервером L2TP Network Server (LNS), який є завершальним пристроєм і відповідає за автентифікацію трафіку (рис. 1.3) [7].

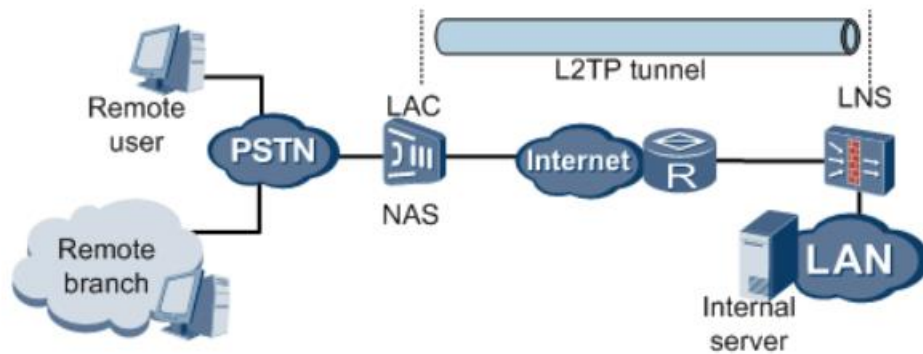


Рисунок 1.3 – Типова мережа на основі L2TP

#### 1.2.4. Протокол OpenVPN

OpenVPN – протокол з відкритим вихідним кодом, який використовує бібліотеки Open SSL, TLS і ряд інших технологій. Сьогодні даний протокол є першочерговим для використання у корпоративній сфері VPN-сервісами і реалізується на всіх доступних платформах через допоміжне програмного забезпечення.

Перевагою протоколу OpenVPN є його різноманітність в налаштуванні. Його можна налаштувати на будь-якому порту, що дає змогу замаскувати трафік наприклад під HTTPS, якщо буде обраний 443 порт для налаштування, що тільки ускладнить процес для зловмисників.

Однак гнучкість даного протоколу призводить до таких складностей, як завантаження та налаштування конфігураційних файлів, але деякі провайдери можуть забезпечити уже готовими VPN-клієнтами, що значно полегшить процес використання VPN.

Також одним із переваг протоколу VPN можна вважати регулярні аудити безпеки. На сьогоднішній момент протокол OpenVPN вважається одним із самих надійних, який забезпечує швидку передачу даних.

Розглянемо відмінності проаналізованих протоколів, які задіяні у реалізації VPN технології (табл. 1).

Таблиця 1 – Порівняльна характеристика протоколів

	PPTP	L2TP/IPSec	OpenVPN
Підтримка ОС	Windows, macOS, IOS, деякий час була підтримка GNU/Linux.	Windows, Mac OS X, Linux, iOS, Android. Більшість операційних систем не потребують додаткового ПО.	Solaris, OpenBSD, FreeBSD, NetBSD, GNU/Linux, Apple Mac OS X, QNX, Microsoft Windows, Android, iOS.
Шифрування	Microsoft Point-to-Point Encryption (MPPE), в якому реалізовано RSA RC4 з 128-бітними сеансовими ключами	3DES або AES	За замовчування AES-256-CBC
Порти	TCP-порт 1723	Для обміну ключами – UDP-порт 500, UDP-порт 1701 – для конфігурування, UDP-порт 5500 для обходу NAT	Будь-який доступний
Недоліки	RSA RS4 є вразливим до атаки Bit flipping	Шифр 3DES вразливий до атаки meet-in-the-middle	Прогалин в безпеці не було виявлено

### 1.3 Постановка задачі

Аналіз існуючих протоколів, які використовуються для побудови безпечних VPN-з'єднань дозволяє дійти висновку, що ці протоколи відрізняються між собою, мають свою специфіку та застосовуються для вирішення різних задач.

На основі цього було вирішено розробити віртуальне захищене середовище, за допомогою якого будуть розглянуті окремі методики побудови VPN-з'єднань. Це дозволить досягнути вирішення задач інформаційної безпеки, а саме забезпечення доступності, цілісності, конфіденційності особистих та корпоративних даних, до яких має доступ лише обмежене коло людей.

Метою даної роботи є розробка віртуального середовища для вивчення методик побудови безпечних VPN-з'єднань. Це віртуальне середовище може бути використане для таких задач:

- оцінка основних протоколів для реалізації VPN-з'єднань;
- вибір операційної системи та програмних додатків для реалізації VPN-з'єднань та захищеного середовища;
- коректне налаштування основних компонентів мережі для реалізації захищеного середовища в корпоративній мережі;
- вивчення основних методів побудови захищених з'єднань у мережі.

## 2. ВИБІР МЕТОДІВ РІШЕННЯ ЗАДАЧІ ТА ПІДГОТОВКА СИСТЕМИ ДО НАЛАШТУВАННЯ

### 2.1 Вибір основного програмного забезпечення для реалізації VPN

Для розробки віртуального середовища було обрано операційну систему Linux, оскільки більшість експертів вважають, що саме за допомогою неї можна забезпечити належний захист даних, файлів конфігурації та іншої важливої інформації, а також Linux найпопулярніша ОС на серверах. Належний захист ядра системи, усіх служб та програмних додатків є також одним з основних пунктів, які роблять дану систему безпечною з точки зору захисту даних[4].

Під час вибору дистрибутиву Linux, було обрано для реалізації Centos 8, тому, що за допомогою даної операційної системи можливо завантажити та налаштувати PPTP або PPTPD сервер – це реалізація серверу на основі протоколу PPTP та OpenVPN серверу, вихідний код якого є загальнодоступним для майже всіх операційних систем. Також значною перевагою даного дистрибутиву є велика кількість репозиторіїв, які містять велику кількість безкоштовних програмних додатків та уже налаштованих файлів конфігурацій.

Для того, щоб завантажити та налаштувати сервер на основі протоколу PPTP необхідно додати додатковий репозиторій та завантажити відповідні пакети:

```
yum -y install epel-release;
```

```
yum -y install ppp pptpd net-tools iptables-services;
```

Основним файлом конфігурації даного серверу є файл, який знаходиться в `/etc/pptpd.conf`.

Щоб завантажити та почати налаштування OpenVPN серверу також необхідно додати додатковий репозиторій, аналогічний що і для серверу на основі протоколу PPTP та завантажити нижче зазначені пакети:

```
yum -y install epel-release;
```

```
yum -y install openvpn;
```

Також знадобиться утиліта Easy-RSA, для того щоб згенерувати сертифікати, які потім будуть застосовані в подальшому:

```
wget https://github.com/OpenVPN/easy-rsa/archive/master.zip
```

Основними для конфігурування є такі файли:

- /etc/openvpn/server.conf – конфігураційний файл сервера;
- /etc/openvpn/ccd/client – налаштування для кожного клієнта;
- /var/log/openvpn/openvpn.log та /var/log/openvpn/openvpn.log – файли в які будуть занотовуватися основні події, які виконуються на стороні сервера та клієнта.

## 2.2 Вибір програмного забезпечення для віртуалізації

Сьогодні існує безліч програмних додатків для віртуалізації, але деякі з них є комерційним продуктом. Одним з таких додатків є VMware Workstation вихідний код якого є закритим. Даний продукт більше підходить для розробників ПЗ або інженерів, які займаються тестуванням, оскільки вона має велику кількість дрібниць, що використовуються кожного дня і яких немає в інших додатках. У нашому випадку більше підходить продукт VirtualBox від компанії Oracle. Даний додаток є безкоштовним, вихідний код якого є відкритим. Перевагами даного продукту є:

- підтримка великої кількості операційних систем, а саме Windows, Linux, Mac OS X и Solaris;
- можливість обмеження споживання ресурсів CPU і введення-виведення;
- можливість регулювання відеопам'яті;
- зрозумілий графічний інтерфейс;
- зручний редактор мережевої взаємодії на хості та інше.

### 2.3 Встановлення та налаштування усіх сегментів мережі

Для розробки віртуального середовища було використано програму VirtualBox, оскільки дана утиліта є зручною у використанні та є безкоштовною. Віртуальне середовище складається із трьох сегментів мережі, двох локальних мереж, локальна мережа за серверною частиною та локальна мережа за клієнтською частиною, а також мережа між шлюзами. В таблиці представлено імена комп'ютерів та їхні IP-адреси (Таблиця 2).

Таблиця 2 – Таблиця налаштувань IP-адрес

Комп'ютер	IP-адреса
Server	192.168.1.12/24
	10.0.9.3/24
Client	192.168.1.14/24
	10.0.10.3/24
PC1	10.0.9.4/24
PC2	10.0.10.4/24

На серверах та на звичайних комп'ютерах встановлено операційну систему Centos 8. Мінімальні системні вимоги до цієї операційної системи невеликі, а саме 1 ГБ оперативної пам'яті, одноядерний процесор та 10 ГБ вільного місця на жорсткому диску. Для роботи було створено 4 віртуальних машини (рис. 2.1). Фізична топологія мережі має такий вигляд (рис. 2.2).



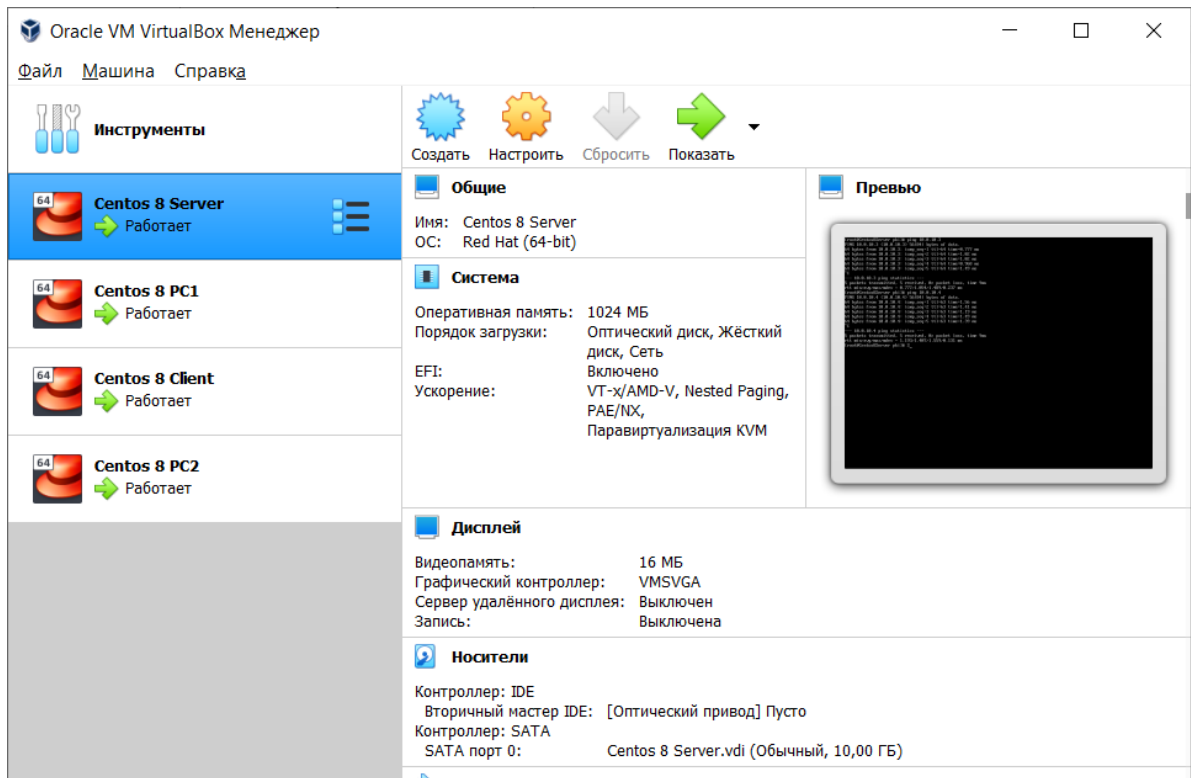


Рисунок 2.1 – Створені віртуальні машини

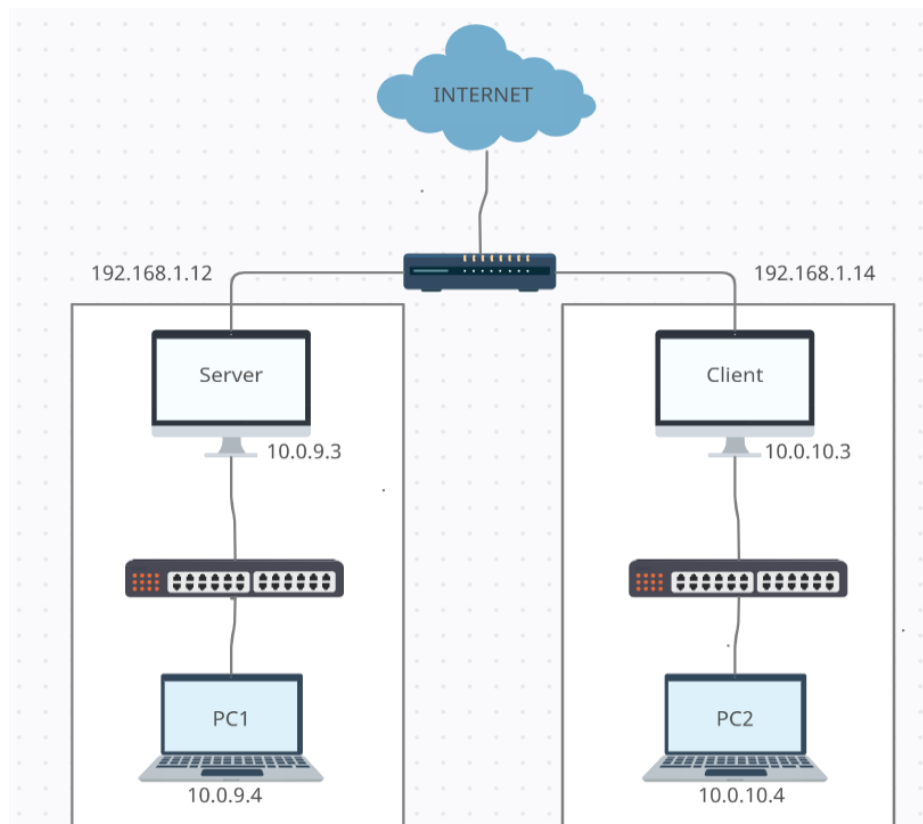


Рисунок 2.2 – Фізична топологія мережі

## 2.4 Оновлення елементів системи та завантаження допоміжних утиліт

Першим кроком після встановлення Centos 8 на наш ПК необхідно оновити систему за допомогою команди - `dnf update`. Це потрібно для того, щоб оновити системні файли, додатки, ядро системи та багато іншого важливого інструментарію(рис. 2.3, рис. 2.4) [7].

```

Centos 8 Server [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
libcurl          x86_64      7.61.1-14.el8_3.1      baseos      299 k
libldb           x86_64      2.1.3-3.el8_3         baseos      179 k
libselinux       x86_64      2.9-4.el8_3           baseos      165 k
libselinux-utils x86_64      2.9-4.el8_3           baseos      242 k
linux-firmware  noarch     20200619-101.git3890db36.el8_3 baseos      101 M
microcode_ctl   x86_64      4:20200609-2.20210216.1.el8_3 baseos      4.6 M
nettle           x86_64      3.4.1-4.el8_3         baseos      301 k
nss              x86_64      3.53.1-17.el8_3       appstream   723 k
nss-softoken    x86_64      3.53.1-17.el8_3       appstream   484 k
nss-softoken-freebl x86_64      3.53.1-17.el8_3       appstream   376 k
nss-sysinit     x86_64      3.53.1-17.el8_3       appstream    72 k
nss-util        x86_64      3.53.1-17.el8_3       appstream   136 k
openssl         x86_64      1:1.1.1g-15.el8_3     baseos      707 k
openssl-libs    x86_64      1:1.1.1g-15.el8_3     baseos      1.5 M
procps-ng       x86_64      3.3.15-3.el8_3        baseos      328 k
python3-bind    noarch     32:9.11.20-5.el8_3.1  appstream   149 k
python3-libselinux x86_64      2.9-4.el8_3           baseos      283 k
python3-perf    x86_64      4.18.0-240.22.1.el8_3 baseos      4.5 M
qemu-guest-agent x86_64      15:4.2.0-34.module.el8_3.0+755+88436ea4.5 appstream   233 k
selinux-policy  noarch     3.14.3-54.el8_3.4     baseos     622 k
selinux-policy-targeted noarch     3.14.3-54.el8_3.4     baseos      15 M
sudo            x86_64      1.8.29-6.el8_3.1      baseos     924 k
systemd         x86_64      239-41.el8_3.2        baseos     3.5 M
systemd-libs    x86_64      239-41.el8_3.2        baseos      1.1 M
systemd-pam     x86_64      239-41.el8_3.2        baseos      456 k
systemd-udev    x86_64      239-41.el8_3.2        baseos     1.3 M
tuned           noarch     2.14.0-3.el8_3.1      baseos      292 k
tzdata          noarch     2021a-1.el8            baseos      473 k
zlib            x86_64      1.2.11-16.2.el8_3     baseos      102 k
=====
-----
===== 4 =====
===== 81 =====

===== 251 M
===== ? [y/n]: y
=====
(1/85): grub2-tools-efi-2.02-90.el8_3.1.x86_64.rpm          415 kB/s | 471 kB   00:01
(2/85): kernel-4.18.0-240.22.1.el8_3.x86_64.rpm          484 kB/s | 4.4 MB   00:09
(3/85): bind-libs-9.11.20-5.el8_3.1.x86_64.rpm           300 kB/s | 172 kB   00:00
(4/85): bind-libs-lite-9.11.20-5.el8_3.1.x86_64.rpm      231 kB/s | 1.2 MB   00:05
(5/85): bind-license-9.11.20-5.el8_3.1.noarch.rpm        210 kB/s | 102 kB   00:00
(6/85): bind-utils-9.11.20-5.el8_3.1.x86_64.rpm          337 kB/s | 445 kB   00:01
(7/85): nss-3.53.1-17.el8_3.x86_64.rpm                   117 kB/s | 723 kB   00:06
(8-10/85): kernel-core-4.18.0-240.22.1.el8_3 14% [===== 1.4 MB/s | 36 MB   02:34 ETA

```

Рисунок 2.3 – Оновлення системи

Package Name	Version	Architecture	Download Speed	Time Remaining
(32/85): dracut-squash-049-95.git20200804.e18_3.4.x86_64.rpm	049-95.git20200804.e18_3.4.x86_64	x86_64	209 kB/s	57 kB 00:00
(33/85): bpftool-4.18.0-240.22.1.e18_3.x86_64.rpm	4.18.0-240.22.1.e18_3.x86_64	x86_64	621 kB/s	5.0 MB 00:00
(34/85): file-5.33-16.e18_3.1.x86_64.rpm	5.33-16.e18_3.1.x86_64	x86_64	244 kB/s	77 kB 00:00
(35/85): file-libs-5.33-16.e18_3.1.x86_64.rpm	5.33-16.e18_3.1.x86_64	x86_64	563 kB/s	543 kB 00:00
(36/85): freetype-2.9.1-4.e18_3.1.x86_64.rpm	2.9.1-4.e18_3.1.x86_64	x86_64	323 kB/s	394 kB 00:01
(37/85): gnutls-3.6.14-8.e18_3.x86_64.rpm	3.6.14-8.e18_3.x86_64	x86_64	531 kB/s	1.0 MB 00:01
(38/85): grub2-common-2.02-90.e18_3.1.noarch.rpm	2.02-90.e18_3.1.noarch	noarch	360 kB/s	885 kB 00:02
(39/85): grub2-efi-x64-2.02-90.e18_3.1.x86_64.rpm	2.02-90.e18_3.1.x86_64	x86_64	477 kB/s	489 kB 00:00
(40/85): grub2-tools-extra-2.02-90.e18_3.1.x86_64.rpm	2.02-90.e18_3.1.x86_64	x86_64	479 kB/s	1.1 MB 00:02
(41/85): grub2-tools-minimal-2.02-90.e18_3.1.x86_64.rpm	2.02-90.e18_3.1.x86_64	x86_64	328 kB/s	286 kB 00:00
(42/85): iptables-1.8.4-15.e18_3.3.x86_64.rpm	1.8.4-15.e18_3.3.x86_64	x86_64	452 kB/s	584 kB 00:01
(43/85): iptables-ebtables-1.8.4-15.e18_3.3.x86_64.rpm	1.8.4-15.e18_3.3.x86_64	x86_64	285 kB/s	71 kB 00:00
(44/85): iptables-libs-1.8.4-15.e18_3.3.x86_64.rpm	1.8.4-15.e18_3.3.x86_64	x86_64	338 kB/s	186 kB 00:00
(45/85): grub2-tools-2.02-90.e18_3.1.x86_64.rpm	2.02-90.e18_3.1.x86_64	x86_64	396 kB/s	2.0 MB 00:05
(46/85): iwl100-firmware-39.31.5.1-101.e18_3.1.noarch.rpm	39.31.5.1-101.e18_3.1.noarch	noarch	362 kB/s	169 kB 00:00
(47/85): iwl1000-firmware-39.31.5.1-101.e18_3.1.noarch.rpm	39.31.5.1-101.e18_3.1.noarch	noarch	443 kB/s	232 kB 00:00
(48/85): iwl105-firmware-18.168.6.1-101.e18_3.1.noarch.rpm	18.168.6.1-101.e18_3.1.noarch	noarch	367 kB/s	253 kB 00:00
(49/85): iwl135-firmware-18.168.6.1-101.e18_3.1.noarch.rpm	18.168.6.1-101.e18_3.1.noarch	noarch	448 kB/s	262 kB 00:00
(50/85): kernel-core-4.18.0-240.22.1.e18_3.x86_64.rpm	4.18.0-240.22.1.e18_3.x86_64	x86_64	568 kB/s	30 MB 00:54
(51/85): iwl1000-firmware-18.168.6.1-101.e18_3.1.noarch.rpm	18.168.6.1-101.e18_3.1.noarch	noarch	458 kB/s	256 kB 00:00
(52/85): iwl1000-firmware-18.168.6.1-101.e18_3.1.noarch.rpm	18.168.6.1-101.e18_3.1.noarch	noarch	492 kB/s	265 kB 00:00
(53/85): iwl15150-firmware-8.24.2.2-101.e18_3.1.noarch.rpm	8.24.2.2-101.e18_3.1.noarch	noarch	392 kB/s	166 kB 00:00
(54/85): iwl15000-firmware-8.83.5.1-101.e18_3.1.noarch.rpm	8.83.5.1-101.e18_3.1.noarch	noarch	479 kB/s	313 kB 00:00
(55/85): iwl16000-firmware-9.221.4.1-101.e18_3.1.noarch.rpm	9.221.4.1-101.e18_3.1.noarch	noarch	400 kB/s	186 kB 00:00
(56/85): iwl16000g2a-firmware-18.168.6.1-101.e18_3.1.noarch.rpm	18.168.6.1-101.e18_3.1.noarch	noarch	457 kB/s	329 kB 00:00
(57/85): iwl16050-firmware-41.28.5.1-101.e18_3.1.noarch.rpm	41.28.5.1-101.e18_3.1.noarch	noarch	439 kB/s	262 kB 00:00
(58/85): iwl13160-firmware-25.30.13.0-101.e18_3.1.noarch.rpm	25.30.13.0-101.e18_3.1.noarch	noarch	728 kB/s	1.7 MB 00:02
(59/85): kernel-tools-4.18.0-240.22.1.e18_3.x86_64.rpm	4.18.0-240.22.1.e18_3.x86_64	x86_64	610 kB/s	4.5 MB 00:07
(60/85): kernel-tools-libs-4.18.0-240.22.1.e18_3.x86_64.rpm	4.18.0-240.22.1.e18_3.x86_64	x86_64	566 kB/s	4.4 MB 00:07
(61/85): kexec-tools-2.0.20-34.e18_3.1.x86_64.rpm	2.0.20-34.e18_3.1.x86_64	x86_64	558 kB/s	496 kB 00:00
(62/85): libcurl-7.61.1-14.e18_3.1.x86_64.rpm	7.61.1-14.e18_3.1.x86_64	x86_64	510 kB/s	299 kB 00:00
(63/85): kmod-kvdo-6.2.3.114-74.e18_3.x86_64.rpm	6.2.3.114-74.e18_3.x86_64	x86_64	519 kB/s	335 kB 00:00
(64/85): libselinux-2.9-4.e18_3.x86_64.rpm	2.9-4.e18_3.x86_64	x86_64	466 kB/s	165 kB 00:00
(65/85): libldb-2.1.3-3.e18_3.x86_64.rpm	2.1.3-3.e18_3.x86_64	x86_64	433 kB/s	179 kB 00:00

Рисунок 2.4 – Оновлення системи

Додатково завантажимо мережеві утиліти, які можуть знадобитися пізніше, а саме для перегляду налаштувань інтерфейсів, портів, таблиць маршрутизації та інше (рис. 2.5).

```

root@Centos8Server:~
[root@Centos8Server ~]# yum install net-tools
Последняя проверка окончания срока действия метаданных: 0:51:07 назад, Вс 16 мая 2021 17:38:15.
Пакет net-tools-2.0-0.52.20160912git.e18.x86_64 уже установлен.
Зависимости разрешены.
Отсутствуют действия для выполнения
Выполнено!
[root@Centos8Server ~]# clear
[root@Centos8Server ~]# yum install bind-utils
Последняя проверка окончания срока действия метаданных: 0:52:32 назад, Вс 16 мая 2021 17:38:15.
Пакет bind-utils-32:9.11.20-5.e18_3.1.x86_64 уже установлен.
Зависимости разрешены.
Отсутствуют действия для выполнения
Выполнено!
[root@Centos8Server ~]#

```

Рисунок 2.5 – Завантаження додаткових утиліт

Тепер завантажимо пакет, за допомогою якого буде налаштовано конфігураційні файли, які відповідають за налаштування мережі та буде підключено додаткові мережеві служби для її керування (рис. 2.6) [7].

```

root@Centos8Server:~
[root@Centos8Server ~]# yum install network-scripts
Последняя проверка окончания срока действия метаданных: 1:09:13 назад, Вс 16 мая
2021 17:38:15.
Зависимости разрешены.
=====
Пакет                Архитектура  Версия                Репозиторий  Размер
=====
Установка:
network-scripts      x86_64       10.00.9-1.el8        baseos       195 k
Установка слабых зависимостей:
network-scripts-team x86_64       1.31-2.el8           baseos       28 k

Результат транзакции
=====
Установка 2 Пакета

Объем загрузки: 223 k
Объем изменений: 179 k
Продолжить? [д/Н]: y
Загрузка пакетов:
(1/2): network-scripts-team-1.31-2.el8.x86_64.r  74 kB/s | 28 kB    00:00
(2/2): network-scripts-10.00.9-1.el8.x86_64.rpm 376 kB/s | 195 kB  00:00
-----
Общий размер                184 kB/s | 223 kB    00:01
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверки транзакции
Тест транзакции проведен успешно
Выполнение транзакции
Подготовка      :                               1/1
Установка       : network-scripts-10.00.9-1.el8.x86_64 1/2
Запуск скриптлета: network-scripts-10.00.9-1.el8.x86_64 1/2
Установка       : network-scripts-team-1.31-2.el8.x86_64 2/2
Запуск скриптлета: network-scripts-team-1.31-2.el8.x86_64 2/2
Проверка        : network-scripts-10.00.9-1.el8.x86_64 1/2
Проверка        : network-scripts-team-1.31-2.el8.x86_64 2/2

Установлен:
network-scripts-10.00.9-1.el8.x86_64 network-scripts-team-1.31-2.el8.x86_64

Выполнено!
[root@Centos8Server ~]# █

```

Рисунок 2.6 – Завантаження пакету для налаштування мережі

### 3. СТВОРЕННЯ ВІРТУАЛЬНОГО ЗАХИЩЕНОГО СЕРЕДОВИЩА

Віртуальне середовище буде являти систему, де буде реалізовано безпечну передачу даних за рахунок використання VPN, а саме однієї із реалізацій даної технології – OpenVPN, а також використанні операційної системи Centos 8 та додаткових утиліт. Це дозволить об'єднати локальні мережі та передавати дані поверх Інтернет мережі.

#### 3.1 Налаштування брандмауера Centos 8 на шлюзах

Оскільки систему було вже оновлено та завантажено додаткові утиліти для роботи з мережею, почнемо налаштування firewall, який потім буде використовуватися при налаштуванні серверу та клієнту OpenVPN. Для налаштування firewall будемо використовувати утиліту iptables – це більш старий спосіб за допомогою якого можна керувати правилами для трафіку. Також можна використовувати firewalld та nftables – це більш новий інструментарій. Спочатку зупинимо роботу firewalld, який встановлений за замовчуванням, завантажимо iptables (рис. 3.1) та занесемо налаштування в конфігураційний файл /etc/sysconfig/iptables (додаток А) [8].

```

root@Centos8Server:~
[root@Centos8Server ~]# systemctl stop firewalld
[root@Centos8Server ~]# systemctl disable firewalld
Removed /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
[root@Centos8Server ~]# yum install iptables-services
Последняя проверка окончания срока действия метаданных: 2:54:02 назад, Вс 16 мая
2021 17:38:15.
Зависимости разрешены.
=====
Пакет                Архитектура  Версия          Репозиторий      Размер
=====
Установка:
iptables-services    x86_64       1.8.4-15.el8_3.3  baseos            62 k
=====
Результат транзакции
=====
Установка 1 Пакет

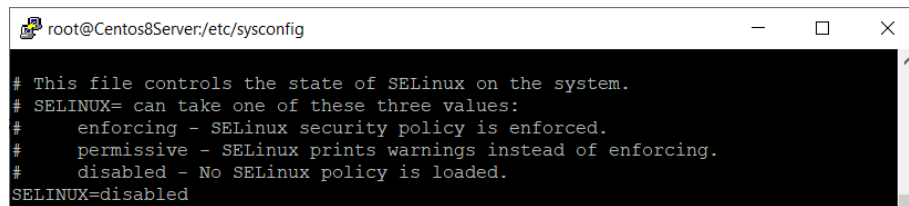
Объем загрузки: 62 k
Объем изменений: 20 k
Продолжить? [д/н]: y
Загрузка пакетов:
iptables-services-1.8.4-15.el8_3.3.x86_64.rpm 249 kB/s | 62 kB 00:00
-----
Общий размер                80 kB/s | 62 kB 00:00
-----
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно
Выполнение транзакции
  Подготовка           :
  Установка             : iptables-services-1.8.4-15.el8_3.3.x86_64 1/1
  Запуск скрипта       : iptables-services-1.8.4-15.el8_3.3.x86_64 1/1
  Проверка              : iptables-services-1.8.4-15.el8_3.3.x86_64 1/1
Установлен:
iptables-services-1.8.4-15.el8_3.3.x86_64
Выполнено!
[root@Centos8Server ~]# systemctl enable iptables
Created symlink /etc/systemd/system/multi-user.target.wants/iptables.service → /usr/lib/systemd/system/iptables.serv
[root@Centos8Server ~]#

```

Рисунок 3.1 – Завантаження та підключення сервісу iptables

### 3.2 Налаштування шлюзів у різних сегментах мережі

Також для коректної роботи необхідно відключити SE-Linux на сервері та клієнті. Для того, щоб це зробити необхідно перейти в папку `/etc/sysconfig/selinux` та змінити значення в рядку `SELINUX` на `disabled` та перезапустити систему. SE-Linux – це система контролю, яка реалізується на рівні ядра (рис. 3.2) [5].



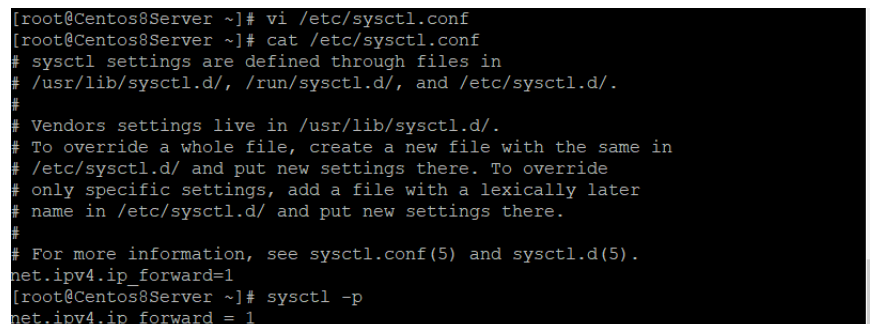
```

root@Centos8Server:/etc/sysconfig
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled

```

Рисунок 3.2 – Відключення системи контролю

Оскільки комп'ютери на яких знаходяться серверне та клієнтське програмне забезпечення є шлюзами, то необхідно налаштувати їх на проходження трафіку. Для того, щоб це зробити необхідно в конфігураційному файлі `/etc/sysctl.conf` дописати рядок `net.ipv4.ip_forward = 1` (рис. 3.3). Після цього ввести команду `sysctl -p` для того, щоб налаштування з файлу виконалися. Також для того, щоб наші шлюзи коректно працювали, необхідно додати правило в `iptables`, тобто в наш `firewall`. Правило має вигляд `-A POSTROUTING -s 10.0.9.0/24 -o enp0s8 -j MASQUERADE`(додаток А). За допомогою даного правила буде відбуватися заміна IP-адреси джерела в заголовці пакету вихідного трафіка на IP-адресу, яка знаходиться на інтерфейсі `enp0s8`, тобто буде реалізовано технологію NAT[5].



```

[root@Centos8Server ~]# vi /etc/sysctl.conf
[root@Centos8Server ~]# cat /etc/sysctl.conf
# sysctl settings are defined through files in
# /usr/lib/sysctl.d/, /run/sysctl.d/, and /etc/sysctl.d/.
#
# Vendors settings live in /usr/lib/sysctl.d/.
# To override a whole file, create a new file with the same in
# /etc/sysctl.d/ and put new settings there. To override
# only specific settings, add a file with a lexically later
# name in /etc/sysctl.d/ and put new settings there.
#
# For more information, see sysctl.conf(5) and sysctl.d(5).
net.ipv4.ip_forward=1
[root@Centos8Server ~]# sysctl -p
net.ipv4.ip_forward = 1

```

Рисунок 3.3 – Налаштування шлюзів

### 3.3 Налаштування серверу OpenVPN

Тепер безпосередньо перейдемо до завантаження та налаштування серверу OpenVPN. Першим кроком підключимо репозиторій epel-release до нашої операційної системи за допомогою команди `yum install epel-release` та завантажимо безпосередньо OpenVPN (рис. 3.4, рис. 3.5).

```

=====
Пакет                Архитектура  Версия        Репозиторий    Размер
=====
Установка:
epel-release         noarch      8-8.el8       extras          23 k
=====
Результат транзакции
=====
Установка 1 Пакет

Объем загрузки: 23 k
Объем изменений: 32 k
Продолжить? [д/н]: y
Загрузка пакетов:
epel-release-8-8.el8.noarch.rpm      103 kB/s | 23 kB    00:00
-----
Общий размер                          34 kB/s | 23 kB    00:00
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно
Выполнение транзакции
Подготовка      :                               1/1
Установка       : epel-release-8-8.el8.noarch 1/1
Запуск скрипта : epel-release-8-8.el8.noarch 1/1
Проверка        : epel-release-8-8.el8.noarch 1/1

Установлен:
epel-release-8-8.el8.noarch

Выполнено!
[root@Centos8Server ~]# █

```

Рисунок 3.4 – Підключення репозиторію

```

Выполнение транзакции
Подготовка      :                               1/1
Установка       : pkcs11-helper-1.22-7.el8.x86_64 1/2
Запуск скрипта : openvpn-2.4.11-1.el8.x86_64    2/2
Установка       : openvpn-2.4.11-1.el8.x86_64    2/2
Запуск скрипта : openvpn-2.4.11-1.el8.x86_64    2/2
Проверка        : openvpn-2.4.11-1.el8.x86_64    1/2
Проверка        : pkcs11-helper-1.22-7.el8.x86_64 2/2

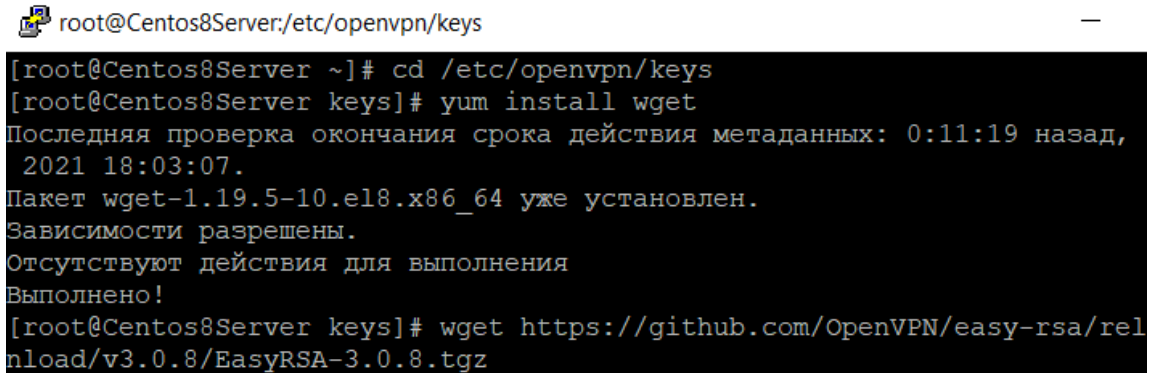
Установлен:
openvpn-2.4.11-1.el8.x86_64      pkcs11-helper-1.22-7.el8.x86_64

Выполнено!
[root@Centos8Server ~]# █

```

Рисунок 3.5 – Підключення репозиторію

Щоб працювати з OpenVPN необхідно спочатку створити сертифікати за допомогою утиліти Easy-RSA, яка є у вільному доступі. Для зберігання ключів створимо директорію `mkdir /etc/openvpn/keys`. Після цього перейдемо до створеного каталогу, завантажимо там нашу утиліту та розархівуємо завантажений архів (рис. 3.6, рис. 3.7).

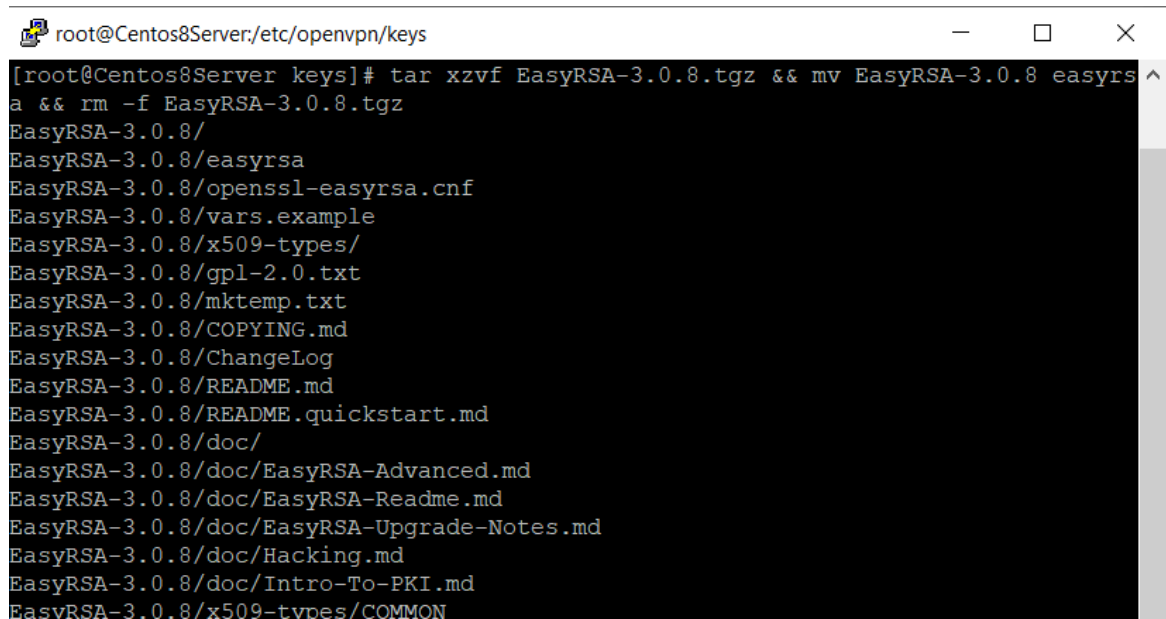


```

root@Centos8Server:/etc/openvpn/keys
[~]# cd /etc/openvpn/keys
[keys]# yum install wget
Последняя проверка окончания срока действия метаданных: 0:11:19 назад,
 2021 18:03:07.
Пакет wget-1.19.5-10.el8.x86_64 уже установлен.
Зависимости разрешены.
Отсутствуют действия для выполнения
Выполнено!
[keys]# wget https://github.com/OpenVPN/easy-rsa/rel
nload/v3.0.8/EasyRSA-3.0.8.tgz

```

Рисунок 3.6 – Створення сертифікатів



```

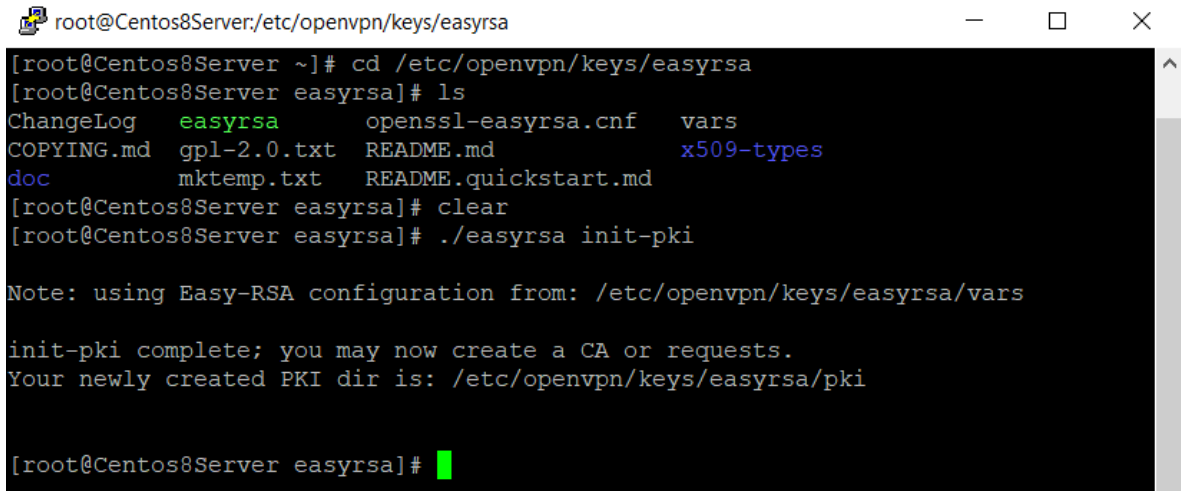
root@Centos8Server:/etc/openvpn/keys
[keys]# tar xzvf EasyRSA-3.0.8.tgz && mv EasyRSA-3.0.8 easyr
a && rm -f EasyRSA-3.0.8.tgz
EasyRSA-3.0.8/
EasyRSA-3.0.8/easyrsa
EasyRSA-3.0.8/openssl-easyrsa.cnf
EasyRSA-3.0.8/vars.example
EasyRSA-3.0.8/x509-types/
EasyRSA-3.0.8/gpl-2.0.txt
EasyRSA-3.0.8/mktemp.txt
EasyRSA-3.0.8/COPYING.md
EasyRSA-3.0.8/ChangeLog
EasyRSA-3.0.8/README.md
EasyRSA-3.0.8/README.quickstart.md
EasyRSA-3.0.8/doc/
EasyRSA-3.0.8/doc/EasyRSA-Advanced.md
EasyRSA-3.0.8/doc/EasyRSA-Readme.md
EasyRSA-3.0.8/doc/EasyRSA-Upgrade-Notes.md
EasyRSA-3.0.8/doc/Hacking.md
EasyRSA-3.0.8/doc/Intro-To-PKI.md
EasyRSA-3.0.8/x509-types/COMMON

```

Рисунок 3.7 – Створення сертифікатів



Перейдемо до каталогу `/etc/openvpn/keys/easyrsa` та створимо структуру публічних ключів за допомогою готового скрипту (рис. 3.8).



```

root@Centos8Server:/etc/openvpn/keys/easyrsa
[root@Centos8Server ~]# cd /etc/openvpn/keys/easyrsa
[root@Centos8Server easyrsa]# ls
ChangeLog  easyrsa      openssl-easyrsa.cnf  vars
COPYING.md  gpl-2.0.txt  README.md           x509-types
doc         mktemp.txt   README.quickstart.md
[root@Centos8Server easyrsa]# clear
[root@Centos8Server easyrsa]# ./easyrsa init-pki

Note: using Easy-RSA configuration from: /etc/openvpn/keys/easyrsa/vars

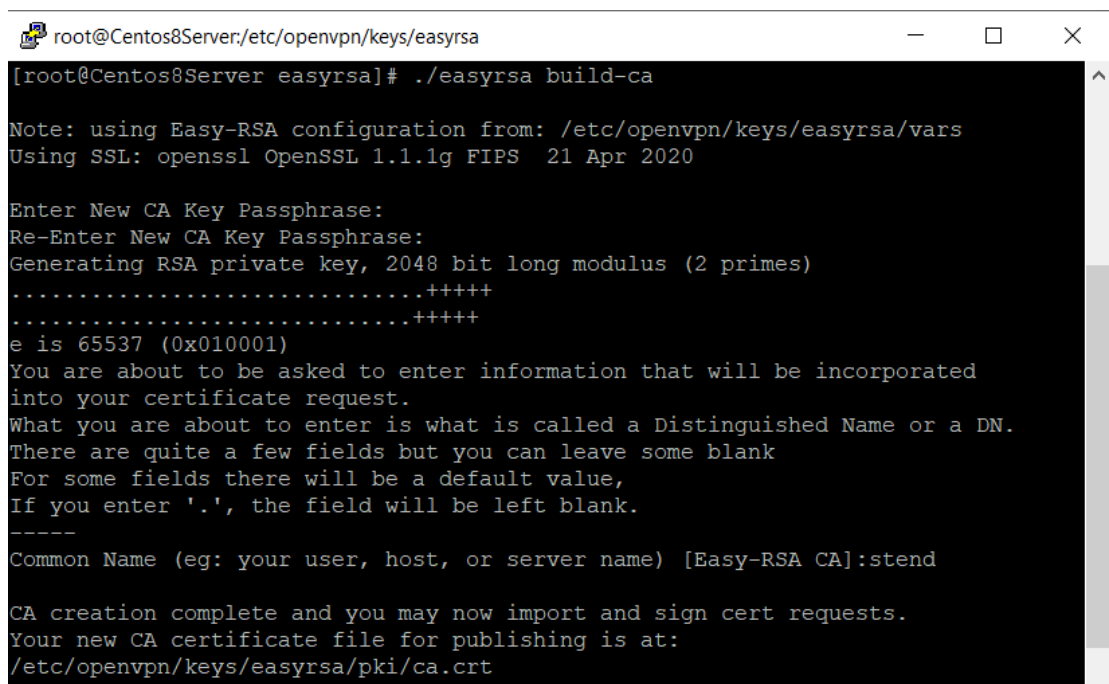
init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /etc/openvpn/keys/easyrsa/pki

[root@Centos8Server easyrsa]# █

```

Рисунок 3.8 – Створення публічних ключів

Тепер створимо центр сертифікації. Він потрібен для того, щоб можна було перевірити правдивість ключів шифрування за допомогою сертифікатів з електронним підписом. Обов'язково при створенні центру сертифікації необхідно створити пароль та запам'ятати його, оскільки він знадобиться при створенні нового сертифікату OpenVPN (рис. 3.9).



```

root@Centos8Server:/etc/openvpn/keys/easyrsa
[root@Centos8Server easyrsa]# ./easyrsa build-ca

Note: using Easy-RSA configuration from: /etc/openvpn/keys/easyrsa/vars
Using SSL: openssl OpenSSL 1.1.1g FIPS 21 Apr 2020

Enter New CA Key Passphrase:
Re-Enter New CA Key Passphrase:
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
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) [Easy-RSA CA]:stend

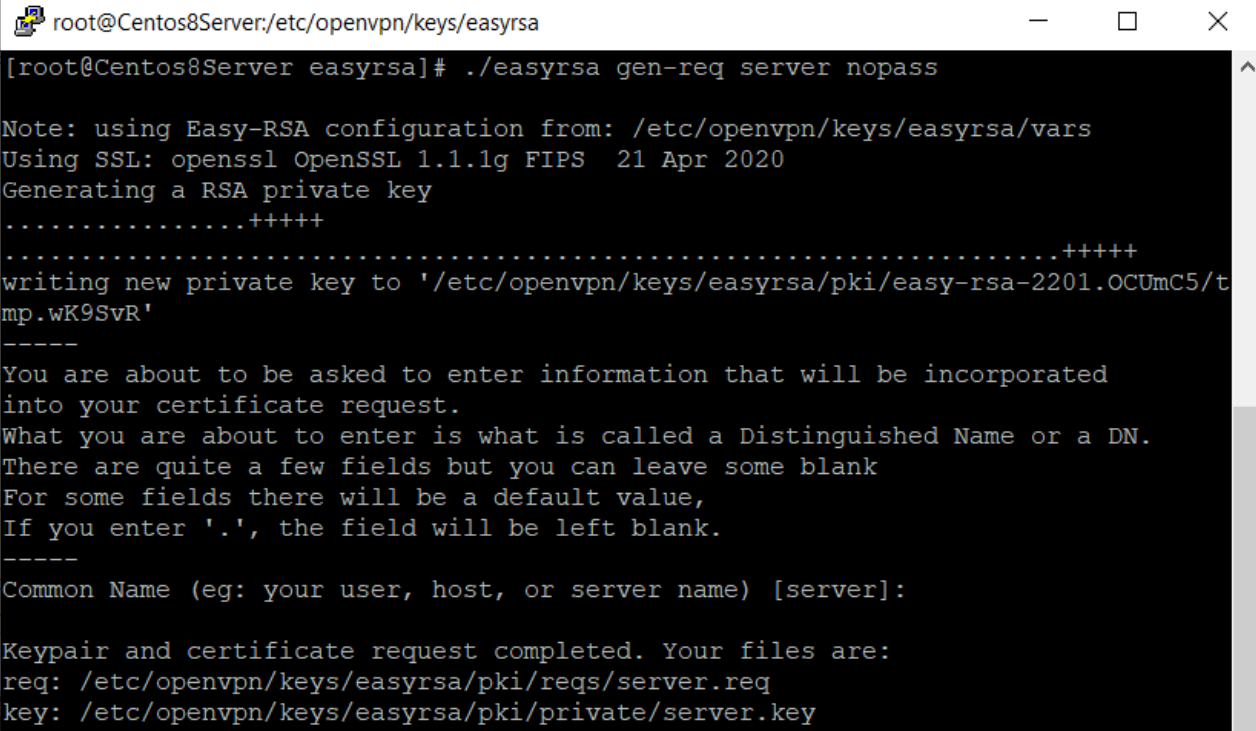
CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/etc/openvpn/keys/easyrsa/pki/ca.crt

```

Рисунок 3.9 – Створення центру сертифікації

Після виконаних дій маємо два файли. Перший файл – це приватний або секретний ключ, який завжди зберігається на сервері та нікому не передається. Він зберігається за таким шляхом `/etc/openvpn/keys/easyrsa/pki/private/ca.key`. Другий ключ – це публічний ключ, який буде передаватися клієнтам разом із призначеними сертифікатами. Він зберігається за таким шляхом `/etc/openvpn/keys/easyrsa/pki/ca.crt`.

Тепер для зручності користування сервером OpenVPN виконаємо команду `./easyrsa gen-req server nopass`. За допомогою цієї команди створимо запит сертифікату для серверу без пароля. Це дозволить запускати сервер та не вводити пароль кожного разу (рис. 3.10).



```

root@Centos8Server:/etc/openvpn/keys/easyrsa
[root@Centos8Server easyrsa]# ./easyrsa gen-req server nopass

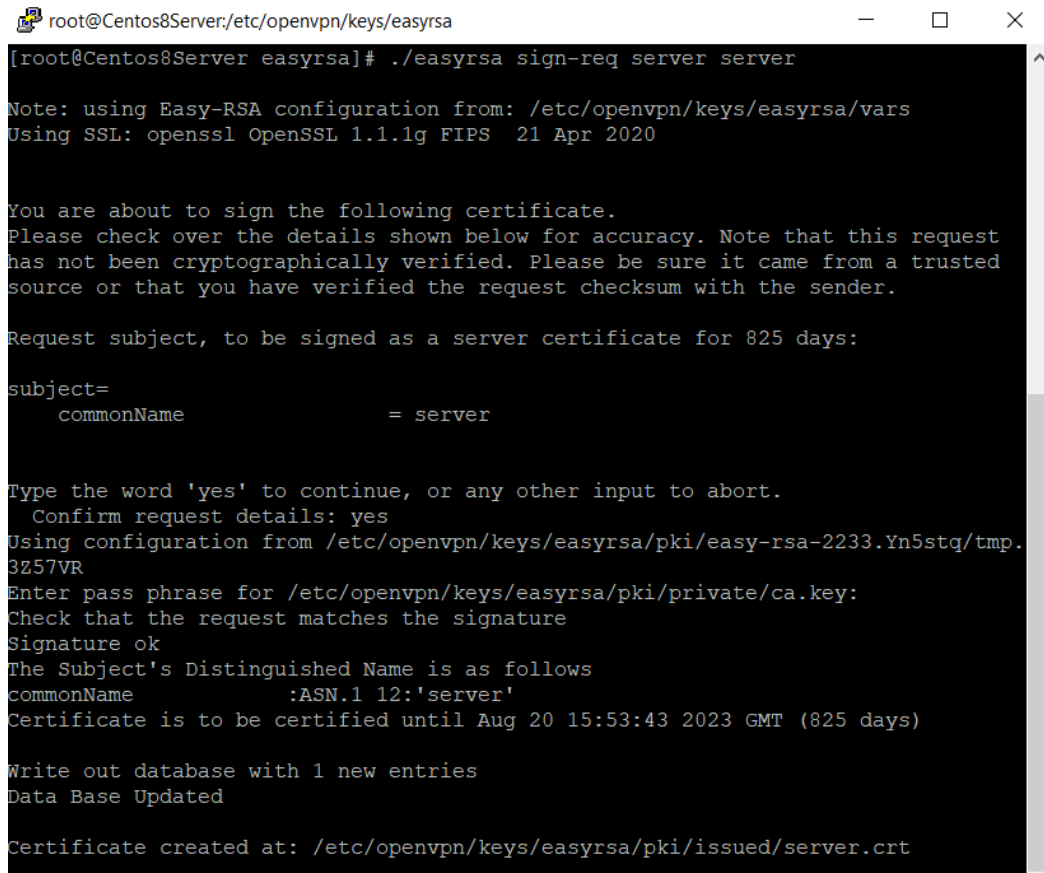
Note: using Easy-RSA configuration from: /etc/openvpn/keys/easyrsa/vars
Using SSL: openssl OpenSSL 1.1.1g FIPS 21 Apr 2020
Generating a RSA private key
.....+++++
.....+++++
writing new private key to '/etc/openvpn/keys/easyrsa/pki/easy-rsa-2201.OCUmC5/
mp.wK9SvR'
-----
You are about to be asked to enter 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) [server]:

Keypair and certificate request completed. Your files are:
req: /etc/openvpn/keys/easyrsa/pki/reqs/server.req
key: /etc/openvpn/keys/easyrsa/pki/private/server.key

```

Рисунок 3.10 – Запит сертифікату для серверу без пароля

Тепер підпишемо запит на отримання сертифікату від нашого центру сертифікації за допомогою команди - `./easyrsa sign-req server server` (рис. 3.11).



```

root@Centos8Server:/etc/openvpn/keys/easyrsa
[root@Centos8Server easyrsa]# ./easyrsa sign-req server server
Note: using Easy-RSA configuration from: /etc/openvpn/keys/easyrsa/vars
Using SSL: openssl OpenSSL 1.1.1g FIPS 21 Apr 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 825 days:

subject=
  commonName                = server

Type the word 'yes' to continue, or any other input to abort.
  Confirm request details: yes
Using configuration from /etc/openvpn/keys/easyrsa/pki/easy-rsa-2233.Yn5stq/tmp.
3Z57VR
Enter pass phrase for /etc/openvpn/keys/easyrsa/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:'server'
Certificate is to be certified until Aug 20 15:53:43 2023 GMT (825 days)

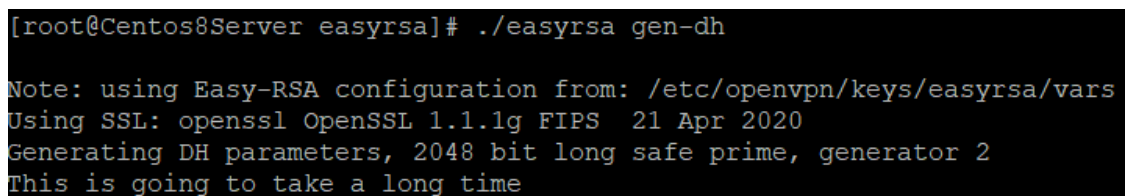
Write out database with 1 new entries
Data Base Updated

Certificate created at: /etc/openvpn/keys/easyrsa/pki/issued/server.crt

```

Рисунок 3.11 – Запит на отримання сертифікату

Також для роботи нам знадобиться створити ключ Діффі-Геллмана. Даний ключ буде використовуватися для обміну раніше створених ключів, оскільки даний метод дозволить користувачам обмінятися ключем, який в подальшому буде використовуватися для симетричного шифрування. Даний алгоритм не буде брати участі у шифруванні повідомлень, його основна мета – це розподіл ключів. Після завершення роботи скрипту, буде отримано файл - `/etc/openvpn/keys/easy-rsa/pki/dh.pem` (рис. 3.12).



```

[root@Centos8Server easyrsa]# ./easyrsa gen-dh
Note: using Easy-RSA configuration from: /etc/openvpn/keys/easyrsa/vars
Using SSL: openssl OpenSSL 1.1.1g FIPS 21 Apr 2020
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time

```

Рисунок 3.12 – Ключ Діффі-Геллмана

Тепер скопіюємо всі необхідні для роботи OpenVPN Server файли в директорію `/etc/openvpn/server/` (рис. 3.13).

```
[root@Centos8Server easyrsa]# cp pki/ca.crt /etc/openvpn/server/ca.crt
[root@Centos8Server easyrsa]# cp pki/dh.pem /etc/openvpn/server/dh.pem
[root@Centos8Server easyrsa]# cp pki/issued/server.crt /etc/openvpn/server/serve
r.crt
[root@Centos8Server easyrsa]# cp pki/private/server.key /etc/openvpn/server/serve
er.key
[root@Centos8Server easyrsa]# ls -l /etc/openvpn/server/
итого 20
-rw----- 1 root root 1180 мая 17 19:17 ca.crt
-rw----- 1 root root  424 мая 17 19:17 dh.pem
-rw----- 1 root root 4579 мая 17 19:18 server.crt
-rw----- 1 root root 1708 мая 17 19:19 server.key
[root@Centos8Server easyrsa]#
```

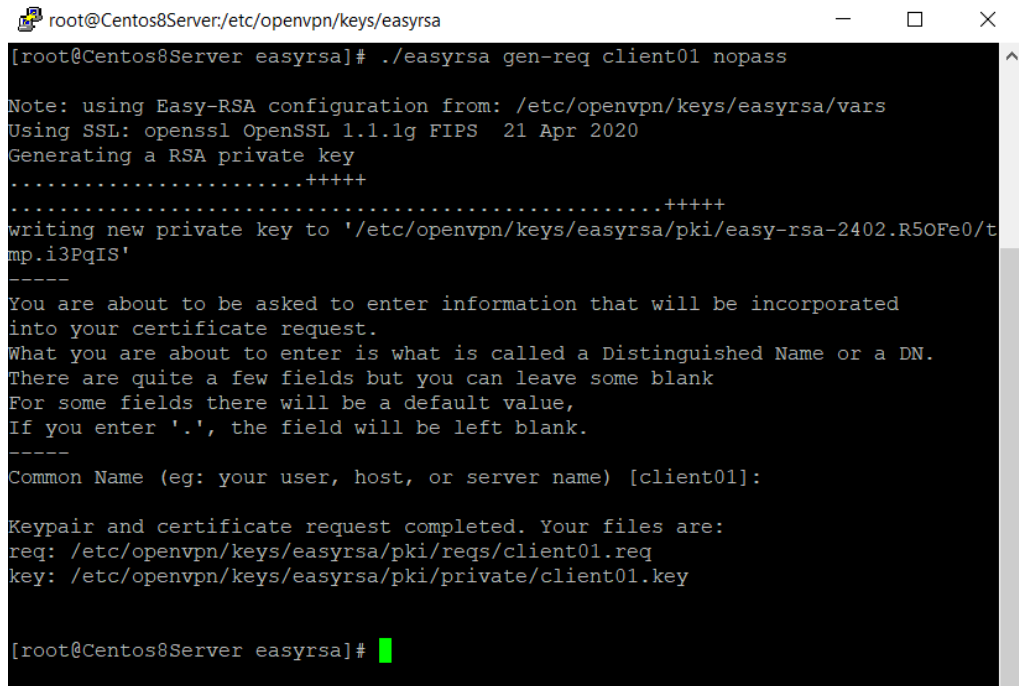
Рисунок 3.13 – Копіювання файлів

Для безпечної роботи необхідно створити також TLS ключ. За допомогою даного ключа буде відбуватися аутентифікація клієнтів, які будуть підключатися до сервера. Даний ключ створюється лише в єдиному екземплярі та повинен знаходитися на кожному комп'ютері, який є учасником VPN-мережі. Для створення TLS ключа необхідно виконати команду `openvpn --genkey --secret /etc/openvpn/server/tc.key` (рис. 3.14).

```
root@Centos8Server:/etc/openvpn/keys/easyrsa
[root@Centos8Server easyrsa]# openvpn --genkey --secret /etc/openvpn/server/tc.k
ey
[root@Centos8Server easyrsa]# ls -l /etc/openvpn/server/
итого 24
-rw----- 1 root root 1180 мая 17 19:17 ca.crt
-rw----- 1 root root  424 мая 17 19:17 dh.pem
-rw----- 1 root root 4579 мая 17 19:18 server.crt
-rw----- 1 root root 1708 мая 17 19:19 server.key
-rw----- 1 root root  636 мая 17 19:25 tc.key
[root@Centos8Server easyrsa]#
```

Рисунок 3.14 – Створення TLS ключа

Створимо ключ для клієнта OpenVPN за допомогою команд `./easyrsa gen-req client01 nopass` та `./easyrsa sign-req client client01` (рис. 3.15, рис. 3.16).



```

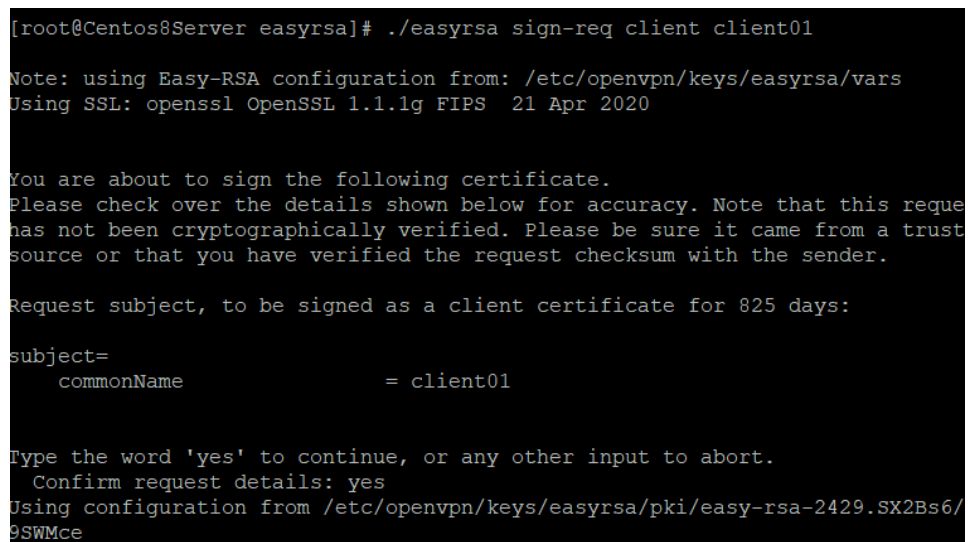
root@Centos8Server:/etc/openvpn/keys/easyrsa
[root@Centos8Server easyrsa]# ./easyrsa gen-req client01 nopass
Note: using Easy-RSA configuration from: /etc/openvpn/keys/easyrsa/vars
Using SSL: openssl OpenSSL 1.1.1g FIPS 21 Apr 2020
Generating a RSA private key
.....+++++
.....+++++
writing new private key to '/etc/openvpn/keys/easyrsa/pki/easy-rsa-2402.R50Fe0/t
mp.i3PqIS'
-----
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) [client01]:

Keypair and certificate request completed. Your files are:
req: /etc/openvpn/keys/easyrsa/pki/reqs/client01.req
key: /etc/openvpn/keys/easyrsa/pki/private/client01.key

[root@Centos8Server easyrsa]# █

```

Рисунок 3.15 – Створення ключа для клієнта



```

[root@Centos8Server easyrsa]# ./easyrsa sign-req client client01
Note: using Easy-RSA configuration from: /etc/openvpn/keys/easyrsa/vars
Using SSL: openssl OpenSSL 1.1.1g FIPS 21 Apr 2020

You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this requ
e
has not been cryptographically verified. Please be sure it came from a trust
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a client certificate for 825 days:

subject=
  commonName                = client01

Type the word 'yes' to continue, or any other input to abort.
Confirm request details: yes
Using configuration from /etc/openvpn/keys/easyrsa/pki/easy-rsa-2429.SX2Bs6/
9SWMce

```

Рисунок 3.16 – Створення ключа для клієнта

Для того, щоб клієнт став частиною мережі VPN, необхідно передати йому набір файлів, а саме `client01.crt`, `client01.key`, `ca.crt`, `tc.key` на комп'ютер на якому буде знаходитися клієнтська частина.

### 3.4 Налаштування конфігураційного файлу OpenVPN

Першим кроком створимо конфігураційний файл OpenVPN за допомогою команди `touch /etc/openvpn/server/server.conf` та занесемо в нього такі налаштування (рис. 3.17)[10]:

```
port 13555 # порт на якому буде працювати VPN
proto udp # протокол, який буде використовуватися на транспортному рівні
```

`dev tun` # у нашому випадку ми використовуємо інтерфейс, оскільки ми хочемо об'єднати дві різні локальні мережі для взаємного доступу до даних, а інакше `tap`.

```
ca /etc/openvpn/server/ca.crt
cert /etc/openvpn/server/server.crt
key /etc/openvpn/server/server.key
dh /etc/openvpn/server/dh.pem
```

```
auth SHA256 # алгоритм шифрування для аутентифікації
cipher AES-256-CBC # алгоритм за допомогою якого будуть шифруватися дані при передачі через vpn.
```

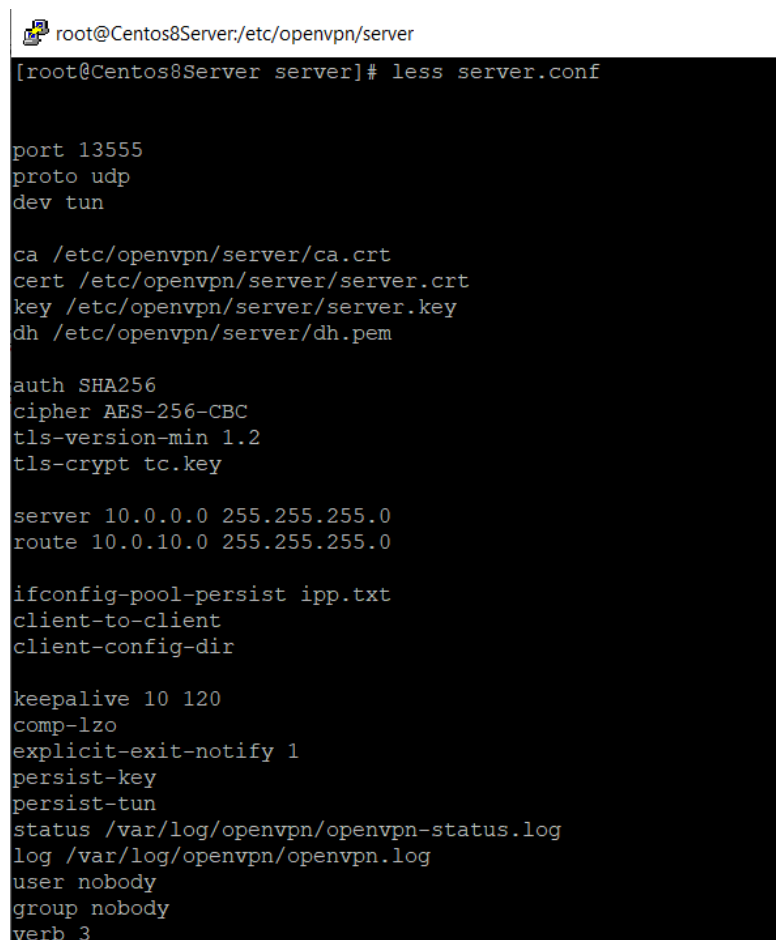
```
tls-version-min 1.2 # версія протоколу TLS.
tls-crypt tc.key # ключ для шифрування TLS.
```

```
server 10.0.0.0 255.255.255.0 # підмережа для тунелю
route 10.0.10.0 255.255.255.0 # тут вказуємо підмережу до якої будуть надходити запити через VPN.
```

```
ifconfig-pool-persist ipp.txt # файл для зберігання відповідностей client-ip.
client-to-client # даний рядок дозволяє користувачам підключатися один до одного.
```

client-config-dir /etc/openvpn/ccd # директорія для зберігання налаштувань клієнтів.

```
keepalive 10 120
comp-lzo
explicit-exit-notify
persist-key
persist-tun
status /var/log/openvpn/openvpn-status.log
log /var/log/openvpn/openvpn.log
user nobody
group nobody
verb 3
```



```
root@Centos8Server:/etc/openvpn/server
[root@Centos8Server server]# less server.conf

port 13555
proto udp
dev tun

ca /etc/openvpn/server/ca.crt
cert /etc/openvpn/server/server.crt
key /etc/openvpn/server/server.key
dh /etc/openvpn/server/dh.pem

auth SHA256
cipher AES-256-CBC
tls-version-min 1.2
tls-crypt tc.key

server 10.0.0.0 255.255.255.0
route 10.0.10.0 255.255.255.0

ifconfig-pool-persist ipp.txt
client-to-client
client-config-dir

keepalive 10 120
comp-lzo
explicit-exit-notify 1
persist-key
persist-tun
status /var/log/openvpn/openvpn-status.log
log /var/log/openvpn/openvpn.log
user nobody
group nobody
verb 3
```

Рисунок 3.17 – Створення конфігураційного файлу OpenVPN

Створимо необхідні директорії для зберігання налаштувань користувачів та для зберігання log файлів (рис. 3.18).

```
[root@Centos8Server server]# mkdir /etc/openvpn/ccd
[root@Centos8Server server]# mkdir /var/log/openvpn
[root@Centos8Server server]#
```

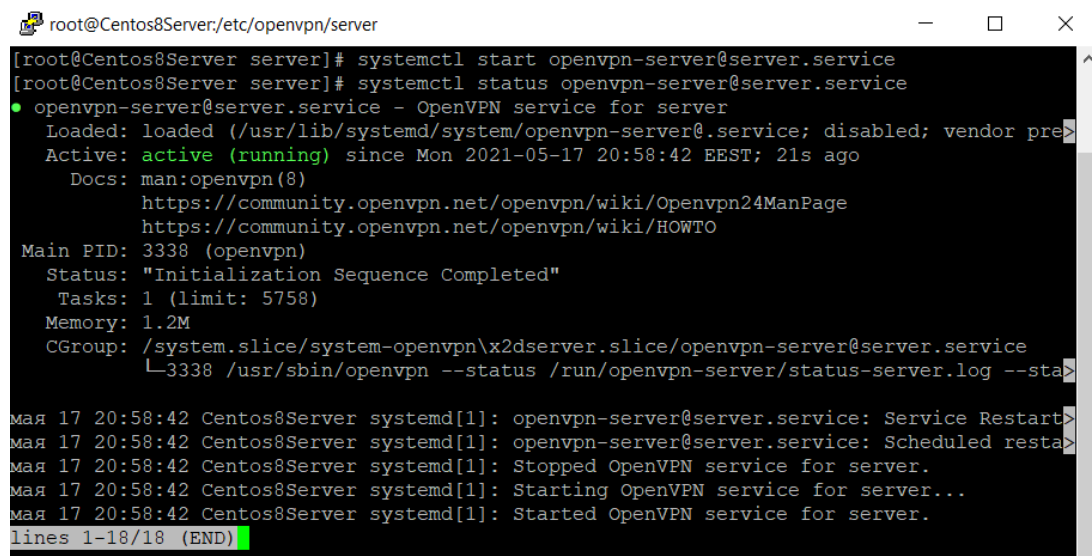
Рисунок 3.18 – Створення додаткових директорій

Тепер в папці де будуть зберігатися конфігураційні файли клієнтів, створимо файл client01 та занесемо в нього параметр iroute 10.0.10.0 255.255.255.0. Даний параметр потрібен для того, щоб співвіднести клієнта та підмережу за яку він відповідає. Зробимо це за допомогою команди touch /etc/openvpn/ccd/client01 (рис. 3.19).

```
[root@Centos8Server server]# touch /etc/openvpn/ccd/client01
[root@Centos8Server server]# vi /etc/openvpn/ccd/client01
[root@Centos8Server server]# less /etc/openvpn/ccd/client01
[root@Centos8Server server]# cat /etc/openvpn/ccd/client01
iroute 10.0.10.0 255.255.255.0
[root@Centos8Server server]#
```

Рисунок 3.19 – Занесення параметрів у файл client01

Після цього можемо запускати наш сервер за допомогою команди `systemctl start openvpn-server@server.service` та перевіримо статус за допомогою команди `systemctl status openvpn-server@server.service` (рис. 3.20).

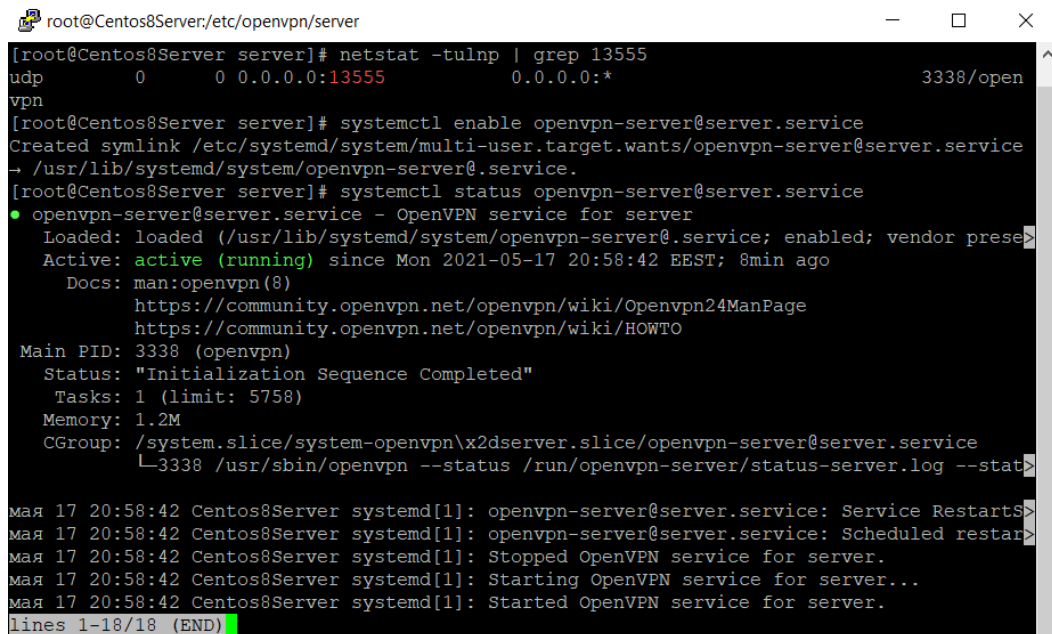


```
root@Centos8Server:/etc/openvpn/server
[root@Centos8Server server]# systemctl start openvpn-server@server.service
[root@Centos8Server server]# systemctl status openvpn-server@server.service
● openvpn-server@server.service - OpenVPN service for server
   Loaded: loaded (/usr/lib/systemd/system/openvpn-server@.service; disabled; vendor pre
   Active: active (running) since Mon 2021-05-17 20:58:42 EEST; 21s ago
     Docs: man:openvpn(8)
           https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
           https://community.openvpn.net/openvpn/wiki/HOWTO
   Main PID: 3338 (openvpn)
   Status: "Initialization Sequence Completed"
     Tasks: 1 (limit: 5758)
    Memory: 1.2M
   CGroup: /system.slice/system-openvpn\x2dserverslice/openvpn-server@server.service
           └─3338 /usr/sbin/openvpn --status /run/openvpn-server/status-server.log --sta
мая 17 20:58:42 Centos8Server systemd[1]: openvpn-server@server.service: Service Restart
мая 17 20:58:42 Centos8Server systemd[1]: openvpn-server@server.service: Scheduled resta
мая 17 20:58:42 Centos8Server systemd[1]: Stopped OpenVPN service for server.
мая 17 20:58:42 Centos8Server systemd[1]: Starting OpenVPN service for server...
мая 17 20:58:42 Centos8Server systemd[1]: Started OpenVPN service for server.
lines 1-18/18 (END)
```

Рисунок 3.20 – Запуск сервера



Перевіримо також чи запустився наш сервер на тому порті, на якому ми зазначили за допомогою команди `netstat -tulnp | grep 13555` та додамо наш сервер до автозапуску (рис. 3.21).



```

root@Centos8Server/etc/openvpn/server
[root@Centos8Server server]# netstat -tulnp | grep 13555
udp        0      0 0.0.0.0:13555      0.0.0.0:*          3338/openvpn

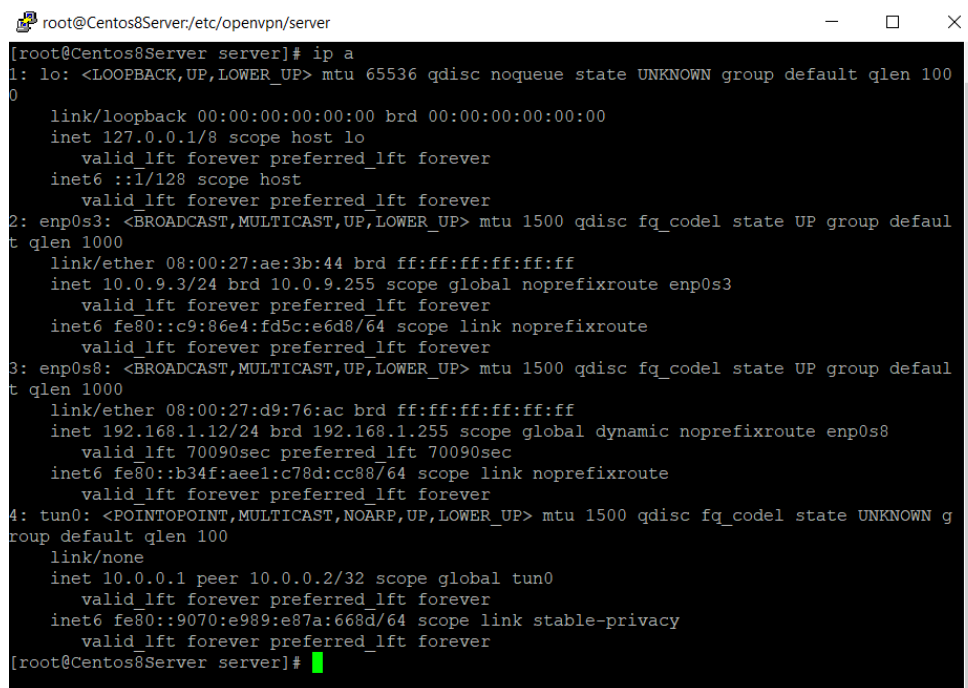
[root@Centos8Server server]# systemctl enable openvpn-server@server.service
Created symlink /etc/systemd/system/multi-user.target.wants/openvpn-server@server.service
→ /usr/lib/systemd/system/openvpn-server@.service.
[root@Centos8Server server]# systemctl status openvpn-server@server.service
● openvpn-server@server.service - OpenVPN service for server
   Loaded: loaded (/usr/lib/systemd/system/openvpn-server@.service; enabled; vendor prese
   Active: active (running) since Mon 2021-05-17 20:58:42 EEST; 8min ago
     Docs: man:openvpn(8)
           https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
           https://community.openvpn.net/openvpn/wiki/HOWTO
   Main PID: 3338 (openvpn)
   Status: "Initialization Sequence Completed"
     Tasks: 1 (limit: 5750)
    Memory: 1.2M
   CGroup: /system.slice/system-openvpn\x2dserver.slice/openvpn-server@server.service
           └─3338 /usr/sbin/openvpn --status /run/openvpn-server/status-server.log --stat

мая 17 20:58:42 Centos8Server systemd[1]: openvpn-server@server.service: Service Restarts
мая 17 20:58:42 Centos8Server systemd[1]: openvpn-server@server.service: Scheduled restar
мая 17 20:58:42 Centos8Server systemd[1]: Stopped OpenVPN service for server.
мая 17 20:58:42 Centos8Server systemd[1]: Starting OpenVPN service for server...
мая 17 20:58:42 Centos8Server systemd[1]: Started OpenVPN service for server.
lines 1-18/18 (END)

```

Рисунок 3.21 – Перевірка порту сервера

Тепер перевіримо налаштування мережевих налаштувань, а саме інтерфейсів та таблицю маршрутизації(рис. 3.22).



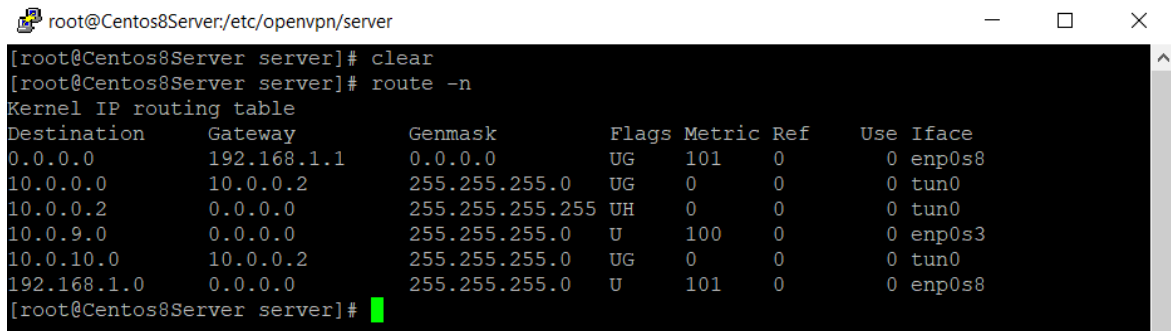
```

root@Centos8Server/etc/openvpn/server
[root@Centos8Server server]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ae:3b:44 brd ff:ff:ff:ff:ff:ff
    inet 10.0.9.3/24 brd 10.0.9.255 scope global noprefixroute enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::c9:86e4:fd5c:e6d8/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:d9:76:ac brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.12/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s8
        valid_lft 70090sec preferred_lft 70090sec
    inet6 fe80::b34f:aeel:c78d:cc88/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
4: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 100
    link/none
    inet 10.0.0.1 peer 10.0.0.2/32 scope global tun0
        valid_lft forever preferred_lft forever
    inet6 fe80::9070:e989:e87a:668d/64 scope link stable-privacy
        valid_lft forever preferred_lft forever
[root@Centos8Server server]#

```

Рисунок 3.22 – Перевірка мережевих налаштувань

В налаштуваннях мережі з'явився новий інтерфейс, а саме tun0, який буде брати участь в утворенні VPN тунелю. Перевіримо таблицю маршрутизації (рис. 3.23).



```

root@Centos8Server/etc/openvpn/server
[root@Centos8Server server]# clear
[root@Centos8Server server]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.1    0.0.0.0         UG    101   0      0   enp0s8
10.0.0.0        10.0.0.2       255.255.255.0   UG    0     0      0   tun0
10.0.0.2        0.0.0.0        255.255.255.255 UH    0     0      0   tun0
10.0.9.0        0.0.0.0        255.255.255.0   U     100   0      0   enp0s3
10.0.10.0       10.0.0.2       255.255.255.0   UG    0     0      0   tun0
192.168.1.0     0.0.0.0        255.255.255.0   U     101   0      0   enp0s8
[root@Centos8Server server]#

```

Рисунок 3.23 – Перевірка таблиці маршрутизації

З рисунку бачимо, що трафік мережі 10.0.10.0/24 буде маршрутизуватися в тунель. Важливим пунктом для правильної роботи VPN є налаштування iptables, необхідно додати такі правила, щоб трафік VPN був дозволений в мережі:

```
iptables -A INPUT -i enp0s8 -p udp --dport 13555 -j ACCEPT
```

```
iptables -A INPUT -i tun+ -j ACCEPT
```

```
iptables -A OUTPUT -o tun+ -j ACCEPT
```

```
iptables -A FORWARD -i tun+ -j ACCEPT
```

```
$IPT -t nat -A POSTROUTING -s 10.0.0.0/24 -j MASQUERADE
```

### 3.5 Налаштування OpenVPN клієнта

Оскільки раніше вже було налаштовано клієнтський комп'ютер як шлюз, правильно налаштовано iptables (додаток А) та відключено SeLinux, то відразу переходимо до завантаження OpenVPN ( рис. 3.24).

```

root@Centos8Client:~
=====
Пакет                Архитектура  Версия           Репозиторий  Размер
=====
Установка:
  openvpn           x86_64       2.4.11-1.e18    epel         543 k
Установка зависимостей:
  pkcs11-helper     x86_64       1.22-7.e18      epel         64 k
=====
Результат транзакции
=====
Установка 2 Пакета

Объем загрузки: 608 к
Объем изменений: 1.4 М
Продолжить? [д/н]: у
Загрузка пакетов:
(1/2): pkcs11-helper-1.22-7.e18.x86_64.rpm   143 kB/s | 64 kB   00:00
(2/2): openvpn-2.4.11-1.e18.x86_64.rpm     728 kB/s | 543 kB  00:00
=====
Общий размер: 608 к

```

Рисунок 3.24 – Завантаження OpenVPN

Створимо конфігураційний файл в директорії `/etc/openvpn/client/` та занесемо в нього такі параметри[10]:

```

dev tun
proto udp
remote 192.168.1.12 13555
client
resolv-retry infinite
ca /etc/openvpn/client/ca.crt
cert /etc/openvpn/client/client01.crt
key /etc/openvpn/client/client01.key
tls-crypt /etc/openvpn/client/tc.key
route 10.0.9.0 255.255.255.0
remote-cert-tls server
auth SHA256

```

```

cipher AES-256-CBC
persist-key
persist-tun
resolv-retry infinite
nobind
comp-lzo
verb 3
status /var/log/openvpn/openvpn-status.log 1
status-version 3
log-append /var/log/openvpn/openvpn-client.log

```

Також скопіюємо в `/etc/openvpn/client/` ключі, які ми згенерували на нашому сервері. Зробимо це за допомогою утиліти WinSCP (рис. 3.25).

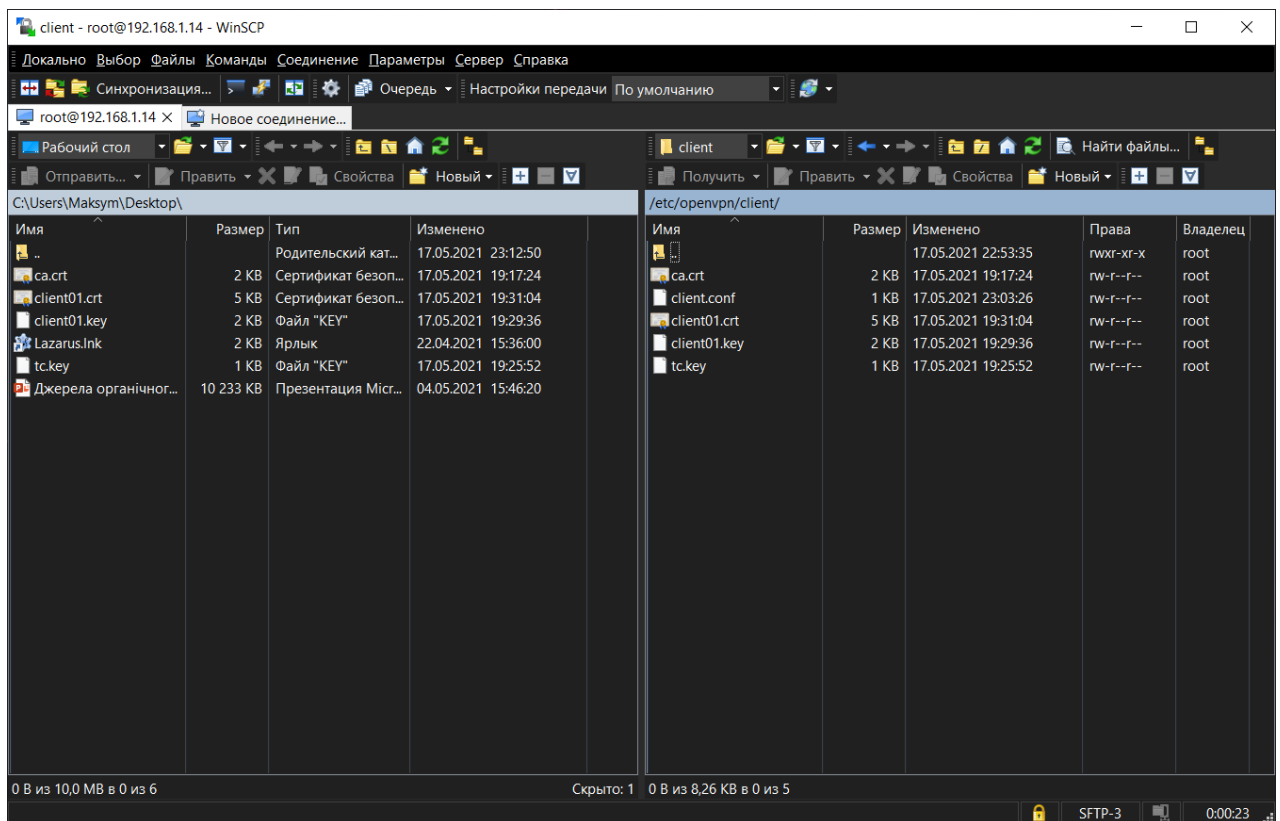


Рисунок 3.25 – Копіювання файлів за допомогою утиліти WinSCP

Створимо директорію для логів /var/log/openvpn, запускаємо клієнтську службу за допомогою команди `systemctl start openvpn-client@client.service` та перевіримо статус `systemctl status openvpn-client@client.service` (рис. 3.26).

```
[root@Centos8Client client]# mkdir /var/log/openvpn
[root@Centos8Client client]# systemctl start openvpn-client@client.service
[root@Centos8Client client]# systemctl enable openvpn-client@client.service
Created symlink /etc/systemd/system/multi-user.target.wants/openvpn-client@client.service.
[root@Centos8Client client]# systemctl status openvpn-client@client.service
● openvpn-client@client.service - OpenVPN tunnel for client
   Loaded: loaded (/usr/lib/systemd/system/openvpn-client@client.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2021-05-17 23:44:44 EEST; 37s ago
     Docs: man:openvpn(8)
           https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
           https://community.openvpn.net/openvpn/wiki/HOWTO
  Main PID: 1689 (openvpn)
    Status: "Initialization Sequence Completed"
     Tasks: 1 (limit: 5758)
    Memory: 1.3M
   CGroup: /system.slice/system-openvpn\x2dclient.slice/openvpn-client@client.service
           └─1689 /usr/sbin/openvpn --suppress-timestamps --nobind --config client.conf

мая 17 23:44:44 Centos8Client systemd[1]: Starting OpenVPN tunnel for client...
мая 17 23:44:44 Centos8Client systemd[1]: Started OpenVPN tunnel for client.
[root@Centos8Client client]#
```

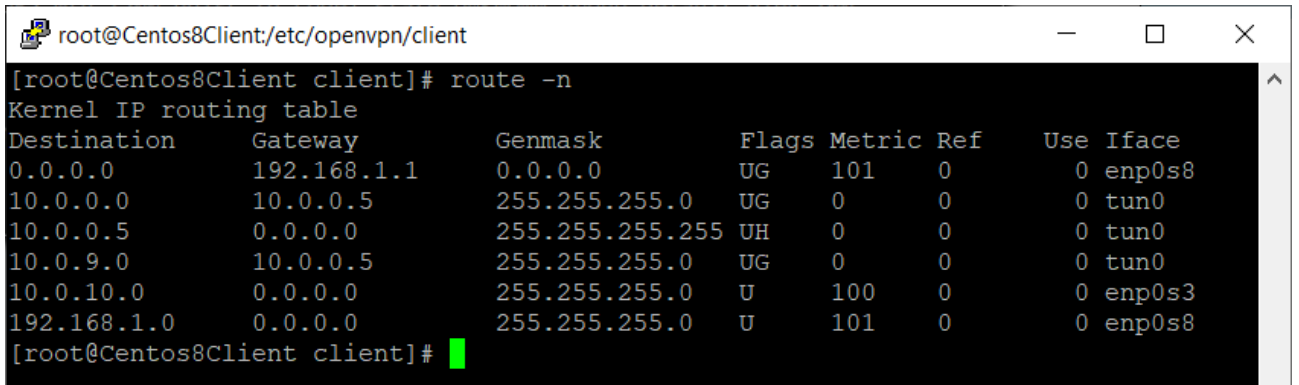
Рисунок 3.26 – Запуск клієнтської служби

Тепер перевіримо налаштування мережевих налаштувань, а саме інтерфейсів та таблицю маршрутизації (рис. 3.27).

```
root@Centos8Client/etc/openvpn/client
    valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 08:00:27:a3:a6:5f brd ff:ff:ff:ff:ff:ff
   inet 10.0.10.3/24 brd 10.0.10.255 scope global noprefixroute enp0s3
      valid_lft forever preferred_lft forever
   inet6 fe80::c41:e877:49f8:9bd8/64 scope link noprefixroute
      valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 08:00:27:1b:c2:bb brd ff:ff:ff:ff:ff:ff
   inet 192.168.1.14/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s8
      valid_lft 81953sec preferred_lft 81953sec
   inet6 fe80::b3a9:d1f8:64e:d4ec/64 scope link noprefixroute
      valid_lft forever preferred_lft forever
4: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 100
   link/none
   inet 10.0.0.6 peer 10.0.0.5/32 scope global tun0
      valid_lft forever preferred_lft forever
   inet6 fe80::81aa:6c81:f748:66df/64 scope link stable-privacy
      valid_lft forever preferred_lft forever
[root@Centos8Client client]#
```

Рисунок 3.27 – Перевірка мережевих налаштувань

В налаштуваннях мережі з'явився новий інтерфейс, а саме tun0, який буде брати участь в утворенні VPN тунелю. Перевіримо таблицю маршрутизації (рис. 3.28).



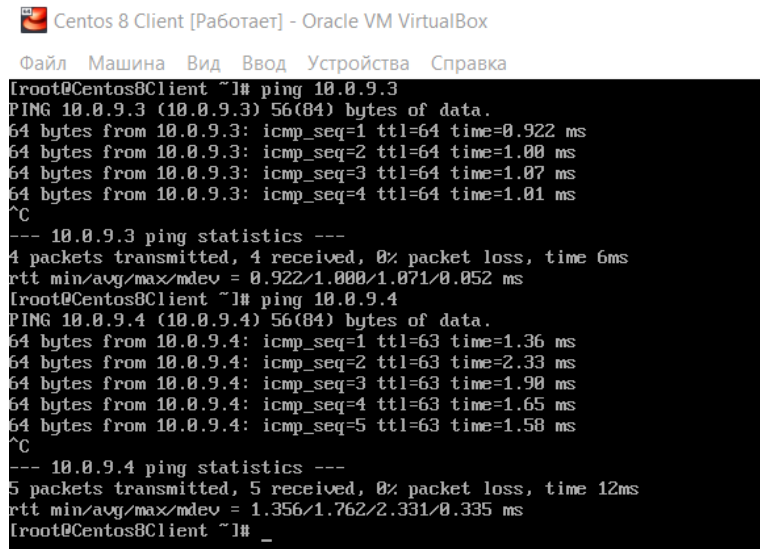
```
root@Centos8Client:/etc/openvpn/client
[root@Centos8Client client]# route -n
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0          192.168.1.1    0.0.0.0         UG    101    0      0   enp0s8
10.0.0.0         10.0.0.5       255.255.255.0   UG    0      0      0   tun0
10.0.0.5         0.0.0.0        255.255.255.255 UH    0      0      0   tun0
10.0.9.0         10.0.0.5       255.255.255.0   UG    0      0      0   tun0
10.0.10.0        0.0.0.0        255.255.255.0   U     100    0      0   enp0s3
192.168.1.0     0.0.0.0        255.255.255.0   U     101    0      0   enp0s8
[root@Centos8Client client]#
```

Рисунок 3.28 – Перевірка таблиці маршрутизації

Четверте правило в таблиці чітко демонструє, що трафік до віддаленої мережі 10.0.9.0 буде проходити через VPN-тунель.

### 3.6 Тестування роботи віртуального середовища

Оскільки комп'ютери в приватній мережі уже налаштовані (додаток А), то перевіримо як працює наша VPN мережа. Зайдемо на комп'ютер на якому встановлено клієнтську частину та пропінгуємо сервер – 10.0.9.3 та комп'ютер PC1 – 10.0.9.4 (рис. 3.29).



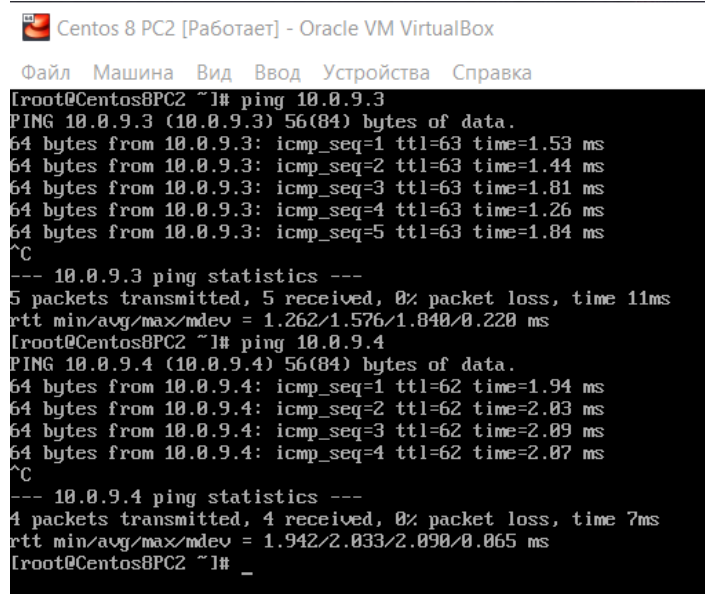
```

Centos 8 Client [Работает] - Oracle VM VirtualBox
Файл Машина Вид Ввод Устройства Справка
[root@Centos8Client ~]# ping 10.0.9.3
PING 10.0.9.3 (10.0.9.3) 56(84) bytes of data.
64 bytes from 10.0.9.3: icmp_seq=1 ttl=64 time=0.922 ms
64 bytes from 10.0.9.3: icmp_seq=2 ttl=64 time=1.00 ms
64 bytes from 10.0.9.3: icmp_seq=3 ttl=64 time=1.07 ms
64 bytes from 10.0.9.3: icmp_seq=4 ttl=64 time=1.01 ms
^C
--- 10.0.9.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 6ms
rtt min/avg/max/mdev = 0.922/1.000/1.071/0.052 ms
[root@Centos8Client ~]# ping 10.0.9.4
PING 10.0.9.4 (10.0.9.4) 56(84) bytes of data.
64 bytes from 10.0.9.4: icmp_seq=1 ttl=63 time=1.36 ms
64 bytes from 10.0.9.4: icmp_seq=2 ttl=63 time=2.33 ms
64 bytes from 10.0.9.4: icmp_seq=3 ttl=63 time=1.90 ms
64 bytes from 10.0.9.4: icmp_seq=4 ttl=63 time=1.65 ms
64 bytes from 10.0.9.4: icmp_seq=5 ttl=63 time=1.58 ms
^C
--- 10.0.9.4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 12ms
rtt min/avg/max/mdev = 1.356/1.762/2.331/0.335 ms
[root@Centos8Client ~]# _

```

Рисунок 3.29 – Перевірка роботи VPN на Client

Зайдемо на комп'ютер PC2 та пропінгуємо сервер – 10.0.9.3 та комп'ютер PC1 – 10.0.9.4 (рис. 3.30).



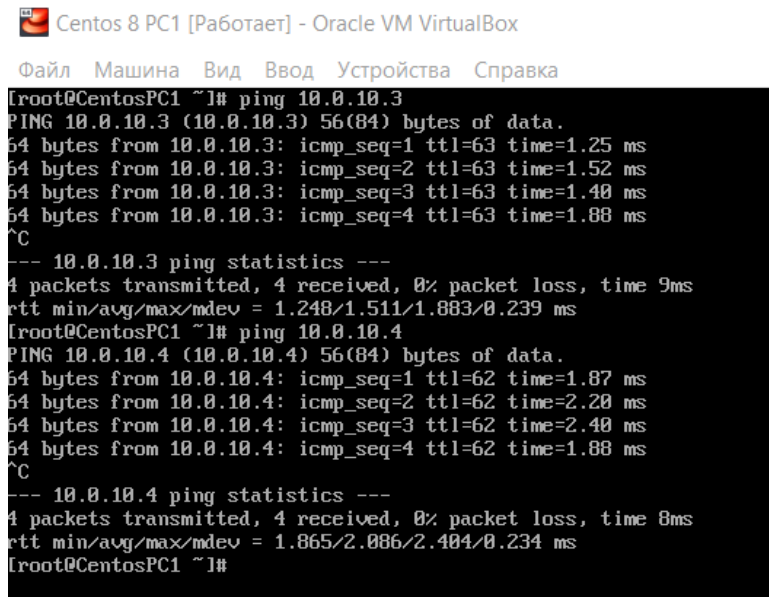
```

Centos 8 PC2 [Работает] - Oracle VM VirtualBox
Файл Машина Вид Ввод Устройства Справка
[root@Centos8PC2 ~]# ping 10.0.9.3
PING 10.0.9.3 (10.0.9.3) 56(84) bytes of data.
64 bytes from 10.0.9.3: icmp_seq=1 ttl=63 time=1.53 ms
64 bytes from 10.0.9.3: icmp_seq=2 ttl=63 time=1.44 ms
64 bytes from 10.0.9.3: icmp_seq=3 ttl=63 time=1.81 ms
64 bytes from 10.0.9.3: icmp_seq=4 ttl=63 time=1.26 ms
64 bytes from 10.0.9.3: icmp_seq=5 ttl=63 time=1.84 ms
^C
--- 10.0.9.3 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 11ms
rtt min/avg/max/mdev = 1.262/1.576/1.840/0.220 ms
[root@Centos8PC2 ~]# ping 10.0.9.4
PING 10.0.9.4 (10.0.9.4) 56(84) bytes of data.
64 bytes from 10.0.9.4: icmp_seq=1 ttl=62 time=1.94 ms
64 bytes from 10.0.9.4: icmp_seq=2 ttl=62 time=2.03 ms
64 bytes from 10.0.9.4: icmp_seq=3 ttl=62 time=2.09 ms
64 bytes from 10.0.9.4: icmp_seq=4 ttl=62 time=2.07 ms
^C
--- 10.0.9.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 7ms
rtt min/avg/max/mdev = 1.942/2.033/2.090/0.065 ms
[root@Centos8PC2 ~]# _

```

Рисунок 3.30 – Перевірка роботи VPN на PC2

І тепер навпаки заїдемо на комп'ютер PC1 та пропінгуємо комп'ютер, на якому встановлено клієнтську частину – 10.0.10.3, та звичайний комп'ютер PC2 – 10.0.10.4 (рис. 3.31).



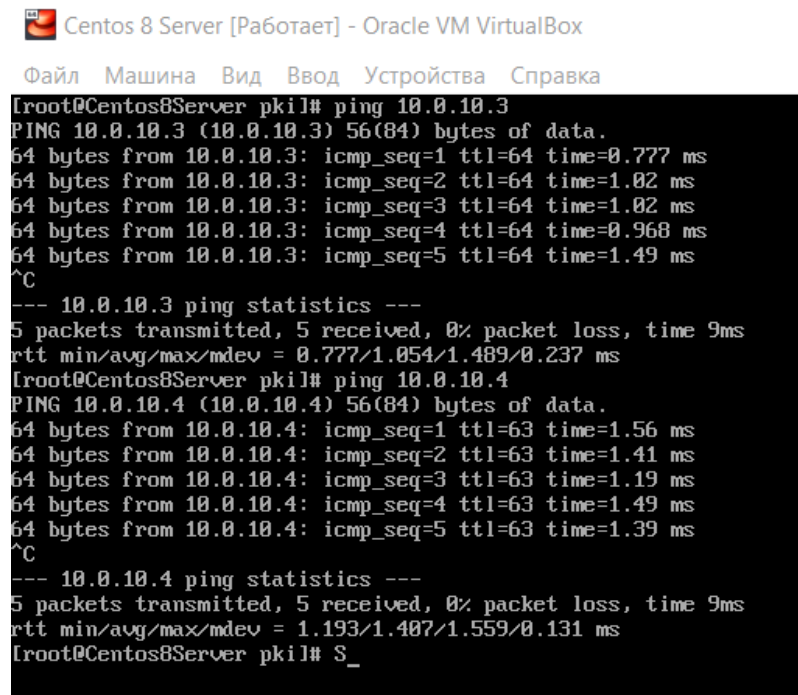
```

Centos 8 PC1 [Работает] - Oracle VM VirtualBox
Файл Машина Вид Ввод Устройства Справка
[root@CentosPC1 ~]# ping 10.0.10.3
PING 10.0.10.3 (10.0.10.3) 56(84) bytes of data.
64 bytes from 10.0.10.3: icmp_seq=1 ttl=63 time=1.25 ms
64 bytes from 10.0.10.3: icmp_seq=2 ttl=63 time=1.52 ms
64 bytes from 10.0.10.3: icmp_seq=3 ttl=63 time=1.40 ms
64 bytes from 10.0.10.3: icmp_seq=4 ttl=63 time=1.88 ms
^C
--- 10.0.10.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 9ms
rtt min/avg/max/mdev = 1.248/1.511/1.883/0.239 ms
[root@CentosPC1 ~]# ping 10.0.10.4
PING 10.0.10.4 (10.0.10.4) 56(84) bytes of data.
64 bytes from 10.0.10.4: icmp_seq=1 ttl=62 time=1.87 ms
64 bytes from 10.0.10.4: icmp_seq=2 ttl=62 time=2.20 ms
64 bytes from 10.0.10.4: icmp_seq=3 ttl=62 time=2.40 ms
64 bytes from 10.0.10.4: icmp_seq=4 ttl=62 time=1.88 ms
^C
--- 10.0.10.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 8ms
rtt min/avg/max/mdev = 1.865/2.086/2.404/0.234 ms
[root@CentosPC1 ~]#

```

Рисунок 3.31 – Перевірка роботи VPN на PC1

І тепер заїдемо на сервер та пропінгуємо комп'ютери – 10.0.10.3 та 10.0.10.4 (рис. 3.32).



```

Centos 8 Server [Работает] - Oracle VM VirtualBox
Файл Машина Вид Ввод Устройства Справка
[root@Centos8Server pkil# ping 10.0.10.3
PING 10.0.10.3 (10.0.10.3) 56(84) bytes of data.
64 bytes from 10.0.10.3: icmp_seq=1 ttl=64 time=0.777 ms
64 bytes from 10.0.10.3: icmp_seq=2 ttl=64 time=1.02 ms
64 bytes from 10.0.10.3: icmp_seq=3 ttl=64 time=1.02 ms
64 bytes from 10.0.10.3: icmp_seq=4 ttl=64 time=0.968 ms
64 bytes from 10.0.10.3: icmp_seq=5 ttl=64 time=1.49 ms
^C
--- 10.0.10.3 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 9ms
rtt min/avg/max/mdev = 0.777/1.054/1.489/0.237 ms
[root@Centos8Server pkil# ping 10.0.10.4
PING 10.0.10.4 (10.0.10.4) 56(84) bytes of data.
64 bytes from 10.0.10.4: icmp_seq=1 ttl=63 time=1.56 ms
64 bytes from 10.0.10.4: icmp_seq=2 ttl=63 time=1.41 ms
64 bytes from 10.0.10.4: icmp_seq=3 ttl=63 time=1.19 ms
64 bytes from 10.0.10.4: icmp_seq=4 ttl=63 time=1.49 ms
64 bytes from 10.0.10.4: icmp_seq=5 ttl=63 time=1.39 ms
^C
--- 10.0.10.4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 9ms
rtt min/avg/max/mdev = 1.193/1.407/1.559/0.131 ms
[root@Centos8Server pkil# S_

```

Рисунок 3.32 – Перевірка роботи VPN на Server



## ВИСНОВКИ

У роботі було розглянуто технологію VPN, а саме, переваги використання даної технології в корпоративній мережі, вартість впровадження на підприємстві та зростання популярності на експлуатацію даної технології в майбутньому. Також було розглянуто технологію тунелювання та шифрування пакетів, які задіяні в VPN. Основними компонентами системи VPN, які задіяні в її реалізації, виокремлюють такі: VPN-клієнт, VPN-сервер та шлюз безпеки VPN. Було розглянуто основні протоколи, які задіяні в реалізації технології VPN, такі як PPTP, IPSec, L2TP/IPSec та OpenVPN, а також їх переваги та недоліки при використанні.

Для реалізації поставленої мети було здійснено вибір програмного продукту, який буде задіяний для віртуалізації, операційної системи для побудови віртуального середовища, а також проаналізовані основні програмні додатки, які будуть задіяні при розробці, додаткові репозиторії та конфігураційні файли.

На практичному прикладі було реалізовано віртуальне середовище, за допомогою якого можна детально зрозуміти всі аспекти технології VPN. Було налаштовано усі сегменти мережі, а саме дві локальні мережі та мережа між шлюзами. Під час роботи було завантажено та використано додаткові утиліти, а також було налаштовано брандмауер операційної системи Centos 8, за допомогою якого відбувається керування трафіком. Також було налаштовано серверну та клієнтську частину OpenVPN в різних сегментах мережі, що дало змогу приєднати дві локальні мережі та протестувати правильність роботи створеної системи на кожному пристрої в системі. Дане віртуальне середовище дозволяє безпечно передавати інформацію поверх публічних мереж, з легкістю приєднувати мережі до захищеного середовища, а отже може використовуватися в компаніях, для під'єднання працівників до корпоративної мережі, а також може бути використане для вивчення основних методів побудови захищених з'єднань у мережі.

## СПИСОК ЛІТЕРАТУРИ

1. Олифер В., Олифер Н. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 5-е изд. — СПб.: Питер, 2016. — 996 с.
2. Организация корпоративных сетей на основе VPN построение, управление, безопасность [Электронный ресурс] // Офіційний сайт проекту kp.ru. — 2018 рік — <https://www.kp.ru/guide/korporativnaja-set.html>.
3. IPSec — протокол защиты сетевого трафика на IP-уровне [Электронный ресурс] // Офіційний сайт проекту ixbt.com. — 2001 рік - <https://www.ixbt.com/comm/ipsecure.shtml>.
4. Системы защиты Linux [Электронный ресурс] // Офіційний сайт блогу Habr. — 2020 рік — <https://habr.com/ru/company/ruvds/blog/523872>.
5. Настройка шлюза на CentOS 7 [Электронный ресурс] // Офіційний сайт блогу serveradmin.ru. — 2019 рік — <https://serveradmin.ru/nastroyka-shlyuza-dlya-lokalnoy-seti-na-centos-7/>.
6. CentOS 7 и 8 настройка сервера после установки [Электронный ресурс] // Офіційний сайт блогу serveradmin.ru. — 2020 рік — <https://serveradmin.ru/centos-nastroyka-servera/>.
7. Типичная сеть с L2TP [Электронный ресурс] // Офіційний сайт проекту lanmarket.ua. — 2021 рік - <https://lanmarket.ua/entsiklopediya/telekommunikatsionnye-tekhnologii/l2tp.html>.
8. Use iptables with CentOS 7 [Электронный ресурс] // Офіційний сайт проекту rackspace technology — 2019 рік — <https://docs.rackspace.com/support/how-to/use-iptables-with-centos-7/>.
9. Шаньгин В.Ф. Информационная безопасность — Москва, 2014. — 702 с.
10. Jan Just Keijser. OpenVPN Cookbook: Second Edition: Birmingham, 2017. — 395 с.

## ДОДАТОК А

### **Мережеві налаштування PC1.**

Налаштування файлу /etc/sysconfig/network-scripts/ifcfg-enp0s3

TYPE=Ethernet

PROXY\_METHOD=none

BROWSER\_ONLY=no

BOOTPROTO=NONE

DEFROUTE=yes

IPADDR=10.0.9.4

PREFIX=24

GATEWAY=192.168.1.12

DNS1=8.8.8.8

IPV4\_FAILURE\_FATAL=no

NAME=enp0s3

DEVICE=enp0s3

ONBOOT=yes

### **Мережеві налаштування PC2.**

Налаштування файлу /etc/sysconfig/network-scripts/ifcfg-enp0s3

TYPE=Ethernet

PROXY\_METHOD=none

BROWSER\_ONLY=no

BOOTPROTO=NONE

DEFROUTE=yes

IPADDR=10.0.10.4

PREFIX=24

GATEWAY=192.168.1.14

DNS1=8.8.8.8

IPV4\_FAILURE\_FATAL=no

```
NAME=enp0s3  
DEVICE=enp0s3  
ONBOOT=yes
```

### **Налаштування серверу.**

Налаштування файлу /etc/sysconfig/network-scripts/ifcfg-enp0s3

```
TYPE=Ethernet  
PROXY_METHOD=none  
BROWSER_ONLY=no  
BOOTPROTO=none  
DEFROUTE=yes  
IPADDR=10.0.9.3  
PREFIX=24  
IPV4_FAILURE_FATAL=no
```

```
NAME=enp0s3  
DEVICE=enp0s3  
ONBOOT=yes
```

Налаштування файлу /etc/sysconfig/network-scripts/ifcfg-enp0s8

```
TYPE=Ethernet  
PROXY_METHOD=none  
BROWSER_ONLY=no  
BOOTPROTO=dhcp  
DEFROUTE=yes  
IPV4_FAILURE_FATAL=no
```

```
NAME=enp0s8  
DEVICE=enp0s8  
ONBOOT=yes
```

Налаштування брандмауера сервера.

\*raw

:PREROUTING ACCEPT

:OUTPUT ACCEPT

COMMIT

\*filter

:INPUT DROP

:FORWARD DROP

:OUTPUT DROP

-A INPUT -i lo -j ACCEPT

-A INPUT -i enp0s3 -j ACCEPT

-A INPUT -p icmp -m icmp --icmp-type 0 -j ACCEPT

-A INPUT -p icmp -m icmp --icmp-type 3 -j ACCEPT

-A INPUT -p icmp -m icmp --icmp-type 11 -j ACCEPT

-A INPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT

-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

-A INPUT -m state --state INVALID -j DROP

-A INPUT -p tcp -m tcp --tcp-flags FIN,SYN,RST,PSH,ACK,URG NONE -j

DROP

-A INPUT -p tcp -m tcp ! --tcp-flags FIN,SYN,RST,ACK SYN -m state --  
state NEW -j DROP

-A INPUT -i tun+ -j ACCEPT

-A INPUT -i enp0s8 -p tcp -m tcp --dport 22 -j ACCEPT

-A INPUT -i enp0s8 -p udp -m udp --dport 13555 -j ACCEPT

-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT

-A FORWARD -m state --state INVALID -j DROP

-A FORWARD -i enp0s3 -o enp0s8 -j ACCEPT

-A FORWARD -i enp0s8 -o enp0s3 -j REJECT --reject-with icmp-port-  
unreachable

-A FORWARD -i tun+ -j ACCEPT

-A FORWARD -i enp0s3 -o tun+ -j ACCEPT

```

-A OUTPUT -o lo -j ACCEPT
-A OUTPUT -o enp0s3 -j ACCEPT
-A OUTPUT -o enp0s8 -j ACCEPT
-A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A OUTPUT -p tcp -m tcp ! --tcp-flags FIN,SYN,RST,ACK SYN -m state --
state NEW -j DROP
-A OUTPUT -o tun+ -j ACCEPT
COMMIT
*nat
:PREROUTING ACCEPT
:INPUT ACCEPT
:POSTROUTING ACCEPT
:OUTPUT ACCEP
-A POSTROUTING -s 10.0.0.0/24 -j MASQUERADE
-A POSTROUTING -s 10.0.9.0/24 -o enp0s8 -j MASQUERADE
COMMIT
*mangle
:PREROUTING ACCEPT
:INPUT ACCEPT
:FORWARD ACCEPT
:OUTPUT ACCEPT
:POSTROUTING ACCEPT
COMMIT

```

### **Налаштування ПК з клієнтською частиною.**

```

Налаштування файлу /etc/sysconfig/network-scripts/ifcfg-enp0s3
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no

```

BOOTPROTO=none

DEFROUTE=yes

IPADDR=10.0.10.3

PREFIX=24

IPV4\_FAILURE\_FATAL=no

NAME=enp0s3

DEVICE=enp0s3

ONBOOT=yes

Налаштування файлу /etc/sysconfig/network-scripts/ifcfg-enp0s8

TYPE=Ethernet

PROXY\_METHOD=none

BROWSER\_ONLY=no

BOOTPROTO=dhcp

DEFROUTE=yes

IPV4\_FAILURE\_FATAL=no

NAME=enp0s8

DEVICE=enp0s8

ONBOOT=yes

Налаштування брандмауера на ПК з клієнтською частиною.

\*filter

:INPUT DROP

:FORWARD DROP

:OUTPUT DROP

-A INPUT -i lo -j ACCEPT

-A INPUT -i enp0s3 -j ACCEPT

-A INPUT -p icmp -m icmp --icmp-type 0 -j ACCEPT

-A INPUT -p icmp -m icmp --icmp-type 3 -j ACCEPT

-A INPUT -p icmp -m icmp --icmp-type 11 -j ACCEPT

-A INPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT

```

-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -m state --state INVALID -j DROP
-A INPUT -p tcp -m tcp --tcp-flags FIN,SYN,RST,PSH,ACK,URG NONE -j
DROP
-A INPUT -p tcp -m tcp ! --tcp-flags FIN,SYN,RST,ACK SYN -m state --
state NEW -j DROP
-A INPUT -i tun+ -j ACCEPT
-A INPUT -i enp0s8 -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -i enp0s8 -p udp -m udp --dport 13555 -j ACCEPT
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -m state --state INVALID -j DROP
-A FORWARD -i enp0s3 -o enp0s8 -j ACCEPT
-A FORWARD -i enp0s8 -o enp0s3 -j REJECT --reject-with icmp-port-
unreachable
-A FORWARD -i tun+ -j ACCEPT
-A FORWARD -i enp0s3 -o tun+ -j ACCEPT
-A OUTPUT -o lo -j ACCEPT
-A OUTPUT -o enp0s3 -j ACCEPT
-A OUTPUT -o enp0s8 -j ACCEPT
-A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A OUTPUT -p tcp -m tcp ! --tcp-flags FIN,SYN,RST,ACK SYN -m state --
state NEW -j DROP
-A OUTPUT -o tun+ -j ACCEPT
COMMIT
*nat
:PREROUTING ACCEPT
:INPUT ACCEPT
:POSTROUTING ACCEPT
:OUTPUT ACCEPT

```



```
-A POSTROUTING -s 10.0.0.0/24 -j MASQUERADE
-A POSTROUTING -s 10.0.10.0/24 -o enp0s8 -j MASQUERADE
COMMIT
*mangle
:PREROUTING ACCEPT
:INPUT ACCEPT
:FORWARD ACCEPT
:OUTPUT ACCEPT
:POSTROUTING ACCEPT
COMMIT
```