ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ

КАФЕДРА ІНФОРМАЦІЙНОЇ ТА КІБЕРНЕТИЧНОЇ БЕЗПЕКИ

Пояснювальна записка

до магістерської роботи
на тему:

«ТЕХНОЛОГІЯ ЗАБЕЗПЕЧЕННЯ ЗАХИЩЕНОГО ОБМІНУ ДАНИМИ В МЕРЕЖІ МОБІЛЬНОГО ЗВ'ЯЗКУ ПІДПРИЄМСТВА З ВИКОРИСТАННЯМ РІШЕННЯ OPENVPN»

Виконав студент 6 курсу, групи БСДМ-62
спеціальності 125 Кібербезпека
освітньо-професійної програми«Інформаційна та кібернетична безпека»
(шифр і назва спеціальності)
Нощенко С.А.
(прізвище та ініціали)
Керівник Гайдур Г.І.
(прізвище та ініціали)
Рецензент
(прізвище та ініціали)
Нормоконтролер Чумак Н.С.
(прізвище та ініціали)

КИЇВ – 2022

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Інститут ННІЗІ
Кафедра Інформаційної та кібернетичної безпеки
Ступінь вищої освіти Магістр
Спеціальність 125 Кібербезпека
Освітньо-професійна програма Інформаційна та кібернетична безпека

ЗАТВЕРДЖУЮ
Завідувач кафедри ІКБ
_____ Гайдур Г.І.
"___"_____ 2021 року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Нощенку Станіславу Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема магістерської роботи: «Технологія забезпечення захищеного обміну даними в мережі мобільного зв'язку підприємства з використанням рішення OpenVPN»

керівник магістерської роботи Гайдур Галина Іванівна, д.техн.н., професор,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від «11» жовтня 2021 року №170.

2. Строк подання студентом магістерської роботи 15.12.2021 р.

3. Вихідні дані до магістерської роботи
концепція побудови сервісів VPN
науково-технічна література експлуатаційна
документація, нормативні документи, міжнародні стандарти.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1.Актуальність проблеми забезпечення захищеного обміну даними в корпоративній мережі.
2. Аналіз методів та засобів побудови VPN на базі технології OpenVPN.
3. Аналіз методів та засобів побудови центру сертифікації (ЦС)
4.Методологія розгортання OpenVPN на базі Linux.

5. Перелік графічного матеріалу

1.
2.
3.
5.
6.
7.
8.
9.
10.

6. Дата видачі завдання 27.09.2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ З/п	Назва етапів магістерської роботи	Строк виконання етапів магістерської роботи	Примітка
1.	Визначення актуальності проблеми забезпечення захищеного обміну даними в корпоративній мережі.	27.09.2021 р.	
2.	Аналіз наукової та технічної літератури з питань теми магістерської роботи.	12.10.2021 р.	
3.	Аналіз методів та засобів побудови VPN на базі протоколу OpenVPN	23.10.2021 р.	
4.	Розроблення варіанту розгортання OpenVPN на базі Linux	07.11.2021 р.	
5.	Розроблення рекомендацій щодо застосування OpenVPN для обміну данними в мобільній мережі підприємства	27.11.2021 р.	
6.	Оформлення результатів дослідження. Проходження плагіату	02.122021 р.	
7.	Підготовка доповіді до захисту.	15.12.2021 р.	

Студент _____ Нощенко С.А.
(підпис) прізвище та ініціали

Керівник магістерської роботи _____ Гайдур Г.І.
(підпис) прізвище та ініціали

ВІДГУК РЕЦЕНЗЕНТА

на магістерську роботу

студента Нощенка Станіслава Андрійовича

на тему: «Технологія забезпечення захищеного обміну даними в мережі мобільного зв'язку підприємства з використанням рішення OpenVPN»

Актуальність: Початок пандемії поставив всі світові компанії у складне становище. Карантин вимагав перенесення стандартного офісного режиму на роботу онлайн. До чого інформаційна безпека та технічне становище не було готовим.

В дипломній роботі розглянуто методологію побудоми комплесу OpenVPN з використанням центру сертифікації на окремому сервері. Описано та проведено налаштування OpenVPN серверу на базі операційної системи Linux. Було проаналізовано можливість використання ВПН в мобільних мережах та розроблено рекомендації щодо встановлення клієнтської частини на різні операційні системи, тому тема роботи є актуальною.

Позитивні сторони:

1. На основі проведеного аналізу, в роботі було встановлено зміст проблеми забезпечення захищеного обміну даними в корпоративній мережі мобільного зв'язку
2. Було досліджено методи та засоби побудови VPN на базі технології OpenVPN.
3. Розроблення варіанту розгортання OpenVPN на базі Linux для побудови захищеної мобільної мережі.
4. Текст викладено достатньо грамотно, послідовно. Сформульовано чіткі та змістовні висновки. Графічний матеріал оформлено якісно. Список науково-технічної літератури свідчить про вміння користуватись матеріалами за темою магістерської роботи.

Недоліки:

1. У магістерській роботі бажано було більш детально розглянути алгоритми роботи VPN протоколів.
2. В роботі доцільно було розлянути більш детально на практиці захищений обмін даними в мобільному зв'язку.

Висновок: Враховуючи недоліки, магістерська робота заслуговує оцінку «**відмінно»**, а студент **Нощенко Станіслав Андрійович** – присвоєння кваліфікації: магістр з кібербезпеки за спеціалізацією інформаційна та кібернетична безпека.

Якість роботи	
Виконано на замовлення підприємства	
Виконано за тематикою НДР	
Виконано з макетом	
Виконано з застосуванням ЕОМ та МПТ	√
Має практичну цінність	√
Проект-частина комплексної теми	

Підпис рецензента (_____)

_____

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

ПОДАННЯ
ГОЛОВІ ДЕРЖАВНОЇ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ
ЩОДО ЗАХИСТУ МАГІСТЕРСЬКОЇ РОБОТИ

Направляється студент Нощенко С.А. до захисту магістерської роботи
(прізвище та ініціали)

спеціальності 125 Кібербезпека
освітньо-професійної програми Інформаційна та кібернетична безпека
(шифр і назва спеціальності)

на тему: «Технологія забезпечення захищеного обміну даними в мережі мобільного зв'язку підприємства з використанням рішення OpenVPN».

Магістерська робота і рецензія додаються.

Директор інституту _____ _____ Савченко В.А.
(підпис) (прізвище та ініціали)

Довідка про успішність

Нощенко С.А. за період навчання в інституті
(прізвище та ініціали студента)

ННІЗІ з 2020 року по 2022 рік повністю виконав навчальний план за напрямом підготовки, спеціальністю з таким розподілом оцінок за:
національною шкалою: відмінно ____%, добре ____%, задовільно ____%;
шкалою ECTS: A ____%; B ____%; C ____%; D ____%; E ____%.

Секретар інституту _____ Журенко О.В.
(підпис) (прізвище та ініціали)

Висновок керівника магістерськоїї роботи

Студент ***Нощенко С.А.*** обрав тему роботи, метою якої було дослідити зміст технології забезпечення захищеного обміну даними в мережі мобільного зв'язку підприємства з використанням рішення OpenVPN. Перелік використаних джерел свідчить про вміння магістром розбиратись в наукових питаннях та застосовувати їх при дослідженнях. Під час виконання магістерської роботи Нощенко С.А. показав відмінну теоретичну та практичну підготовку, вміння самостійно вирішувати питання і робити висновки. Роботу виконував сумлінно, акуратно та вчасно за планом.

Керівник магістерської роботи _____ Гайдур Г.І.
(підпис) (прізвище та ініціали)
"____" _____ 2021 року

Висновок кафедри про магістерську роботу

Магістерська робота розглянута. Студент Нощенко С.А.
(прізвище та ініціали)
допускається до захисту даної магістерської роботи в Державній екзаменаційній комісії
Завідувач кафедри Інформаційної та кібернетичної безпеки
(назва)

_____ Гайдур Г.І.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина магістерської роботи: 62 сторінки, 22 рисунків, 4 таблиці, 17 джерел.

Об’єкт дослідження – інформаційна безпека корпоративних мереж.

Предмет дослідження – технологія побудови віртуальних приватних мереж на базі корпоративної організації.

Мета роботи – підвищення ефективності захисту корпоративних і публічних мереж з використанням нових технології VPN, а також визначення основних способів побудови VPN мереж.

Методи дослідження – опрацювання літератури за даною темою, аналіз технічної документації.

В даній роботі було проведено дослідження структури і принципів роботи основних VPN протоколів та ефективності їх роботи шляхом порівняння показників швидкості передачі інформації, шифрування, сумісність з ОС можливість використовувати мобільні мережі.

Досліджено методи побудови мережі з використанням протоколу OpenVPN. Визначено основні функції та компоненти мержі OpenVPN. На основі проведених досліджень в роботі було описано та проведено налаштування OpenVPN серверу на базі операційної системи Ubuntu, а також процес створення конфігурації для мобільного клієнта.

Галузь використання – кібербезпека корпоративної мережі.

КОРПОРАТИВНА МЕРЕЖА,VPN МЕРЕЖА, ПРОТОКОЛИ ШИФРУВАННЯ, TCP/IP, PPTP, L2TP, IPSEC, ТУНЕЛЬ, OPENVPN

ABSTRACT

Master’s thesis: 72 pages, 22 figures, 4 tables, 35 sources.

The object of research - information security of corporate networks.

The subject of research - the technology of building virtual private networks on the basis of corporate organization.

The aim of research - to increase the effectiveness of protection of corporate and public networks using new VPN technologies, as well as to determine the main ways to build VPN networks.

Research methods - elaboration of literature on this topic, analysis of technical documentation.

This paper is investigated the structure and principles of the main VPN protocols and their efficiency by comparing the rates of data transfer, encryption, compatibility with the OS, the ability to use mobile networks.

Methods of building a network using the OpenVPN protocol are studied. The main functions and components of the OpenVPN network are defined. Based on the research, the work is described and configured the OpenVPN server based on the Ubuntu operating system, as well as the process of creating a configuration for a mobile client.

Field of using - cybersecurity of the corporate network.

CORPORATE NETWORK, VPN NETWORK, ENCRYPTION PROTOCOLS, TCP / IP, PPTP, L2TP, IPSEC, TUNNEL, OPENVPN

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ 9

ВСТУП 10

1 АНАЛІЗ ПРОБЛЕМИ ЗАБЕЗПЕЧЕННЯ КІБЕРБЕЗПЕКИ КОРПОРАТИВНОЇ МЕРЕЖІ ПІДПРИЄМСТВА 12

1.1. Загрози інформацій безпеці корпоративної мережі підприємства 12

1.2. Аналіз проблеми забезпечення безпеки мережі з застосуванням технології VPN 16

1.3. Аналіз існуючих технологій VPN для забезпечення безпеки інформаційних мереж 21

1.3.1. VPN на основі протоколу PPTP 22

1.3.2. VPN на основі протоколу IPSec. 23

1.3.3. VPN на основі SSL 25

1.3.4. Технологія OpenVPN 25

2 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ЗАБЕЗПЕЧЕННЯ ЗАХИЩЕНОГО ОБМІНУ ДАНИМИ З ВИКОРИСТАННЯМ OPENVPN 30

2.1. Дослідження можливостей та функцій технології OpenVPN 30

2.2. Компоненти та архітектура OpenVPN 34

2.3. Центр сертифікації та утиліта EasyRSA 44

3 РОЗРОБЛЕННЯ ВАРІАНТА ТЕХНОЛОГІЇ ЗАБЕЗПЕЧЕННЯ ЗАХИЩЕНОГО ОБМІНУ ДАНИМИ В МОБІЛЬНІЙ МЕРЕЖІ ПІДПРИЄМСТВА З ВИКОРИСТАННЯМ OPENVPN COMMUNITY EDITION 49

3.1. Аналіз та проектування ВПН мережі підприємства 49

3.2. Встановлення на налаштування центру сертифікації 51

3.3. Встановлення та налаштування OpenVPN серверу 54

3.4. Аналіз клієнтської частини OpenVPN 67

ВИСНОВКИ 71

ПЕРЕЛІК ПОСИЛАНЬ 72

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ 74

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

IP – протокол міжмережевого рівня

IPSec – набір протоколів для захисту даних

OSI – мережева модель для комунікацій

L2TP – протокол тунелювання другого рівня

PKI – метод забезпечення криптозадач

RSA – криптографічний алгоритм з відкритим ключем

SHA – алгоритм безпечного хешу

SSH – мережевий протокол прикладного рівня

SSL – протокол безпеки на транспортному рівні

TCP – протокол передачі даних

TLS – криптографічний протокол

UDP – протокол в стеку TCP/IP

VPN – віртуальна приватна мережа

ВСТУП

Актуальність дослідження. На сьогоднішній день передача інформації у великих та малих підприємствах майже завжди проходить за допомогою мережевих технологій в електронному вигляді. При такому виді передачі треба мати на увазі те, що інформація у відкритих для доступу мережах може бути перехоплена та використана в корисних або деструктивних для організації цілях.

Більше того, корпоративні інформаційні структури відрізняються від звичайних своєю різноманітністю. Вони можуть складатися з різних наборів та баз даних, розподілених та локальних систем та задач. Все це відкриває багато різних можливостей для кіберзлочинців та робить технологію вразливою до атак.

З початком пандемії все більшого застосування набирає використання віддаленого доступу між територіально рознесеними інформаційними мережами. У підрозділах автоматизації підприємств це питання також важливе.

Для того, щоб вирішити проблеми, описані вище, застосовуються різні методи захисту інформації, такі як реєстрація користувачів, їх ідентифікація та аутентифікація, контроль доступу, протоколювання, створення різних мережевих екранів. На даний момент фахівці одноголосно визнають, що найнадійнішим методом захисту інформації від втручання є криптографічні методи. Даний спосіб відомий людству ще з давніх-давен, адже саме стародавнє суспільство внесло початок розвитку даного способу збереження даних. Один з нових та популярних способів, що забеспечує подібний захист – віртуальна приватна мережа (з англ. VPN Virtual Private Network).

Комп’ютерні мережі мають необхідність в наявності сервера VPN, який буде дозволяти віддаленим абонентам використовувати ресурси приватної мережі через загальнодоступні мережі. Також VPN може використовуватись для підвищення безпеки передачі інформації в локальній мережі, зменшивши можливість витоку чи крадіжки інформації, яка транспортується в ній.

Об’єкт дослідження – інформаційна безпека корпоративних мереж.

Предмет дослідження – технологія побудови віртуальних приватних мереж на базі корпоративної організації.

Мета роботи – підвищення ефективності захисту корпоративних і публічних мереж з використанням нових технології VPN, а також визначення основних способів побудови VPN мереж.

Практичне значення одержаних результатів полягає в тому , що матеріали роботи можуть бути використані керівництвом при подальшій розробці VPN мереж.

1 АНАЛІЗ ПРОБЛЕМИ ЗАБЕЗПЕЧЕННЯ КІБЕРБЕЗПЕКИ КОРПОРАТИВНОЇ МЕРЕЖІ ПІДПРИЄМСТВА

1.1. Загрози інформацій безпеці корпоративної мережі підприємства

Стрімкий розвиток комп’ютерної техніки, що розпочався з середини XX століття, неодмінно зростає по експоненті. Людство, використовуючи все більше розумної техніки, крок за кроком підходить до нового етапу розвитку, що отримало назву «інформаційного суспільства».

Масове використання розумних технологій у всіх сферах діяльності людини та обсяг інформації, що зберігається на цих носіях, виріс в мільйони разів. На сьогоднішній день, нікого не вражає той факт, що майже 80% світової ринкової продукції складає результати розумової діяльності людини та обробка інформації. Саме тому, боротьба за світовий ринок поступово переходить з матеріального світу до віртуального простору, що призводить до проблем кібербезпеки та кіберзлочинності.

На початку 2019 року у світі виникла нова проблема пандемії, яка поставила всі світові компанії у складне становище. Становище карантину вимагало перенесення стандартного офісного режиму на роботу онлайн з дому.

Сучасні технології зробили величезний крок уперед і дозволяють без проблем перевести цілу компанію на віддалену роботу: онлайн-наради, обмін документами, спільна робота у хмарі тощо.

Однак такий стиль роботи передбачає безліч нюансів, пов'язаних з кібербезпекою. За час карантину в мережі з'явилося багато нового шкідливого програмного забезпечення, частина якого націлена на крадіжку конфіденційних даних.

На сьогоднішній день експерти виділяють п’ять основних загроз інформаційній безпеці корпоративних мереж, а саме:

1) розкриття конфіденційної інформації;
2) злом (неправомірне втручання в роботу комп'ютерної системи;
3) виведення системи з ладу, зниження її працездатності;
4) перевищення повноважень непривілейованим користувача;
5) знищення та спотворення інформації.

З цього можна зробити висновок, щоб інформація була у повній мірі переданою правильно, і дійшла до кінцевого пункту призначення у повному обсязі, потрібно обов'язково підтримати її характеристики.

У широкому спектрі загрози на інформаційні ресурси можна розглядати як можливі випадки природного, штучного, та антропогенного походження, що можуть привести до небажаного впливу на інформаційні ресурси, та інформацію, що знаходиться в ній. Несприятливий вплив на елементи інформаційної системи можна охарактеризувати, як уразливість системи. Саме через цей показник відбувається активізація загроз.

З різних способів класифікації загроз даних, базовою є класифікація за наслідками впливу на інформацію. Захист інформації ведеться з метою підтримки таких властивостей інформації як:

1) погроза порушення конфіденційності інформаційним ресурсам;
2) порушення цілісносності інформації;
3) порушення дієздатності інформаційно-обчислювальних систем.

Розглянемо більш конкретніше ці властивості. Загроза конфіденційності інформаційних ресурсів полягає у тому, що дані, стають розкритими, тобто персональні дані стають відомими особам, що не мають повноважень доступу до неї. Загроза порушення цілісності інформаційних ресурсів полягає в навмисному антропогенному впливі на дані, що зберігається в обчислювальній системі чи переданої з однієї системи в іншу. Порушення дієздатності інформаційно-обчислювальної системи виникає щоразу, коли в результаті навмисних дій, що вживаються іншим користувачем або зловмисником, блокується доступ до деякого ресурсу обчислювальної системи.

Компанія McAfee провела свої експертні дослідження щодо загроз, що виникли в період карантину. Підрозділ з вивчення складних загроз (McAfee Advanced Threat Research - McAfee ATR) досліджували, який вплив та масштаби мають сучасні кібератаки на суспільство та роботу компаній у розрізі карантину. Безумовно, можна сказати, що під час пандемії, поки всі працювали у віддаленому режимі, кіберзлочинці збільшували періодичність своїх атак та розробляли нові методи захоплення корпоративних даних та проникнення в корпоративні мережі для здійснення інших злочинів (Рисунок 1.1).

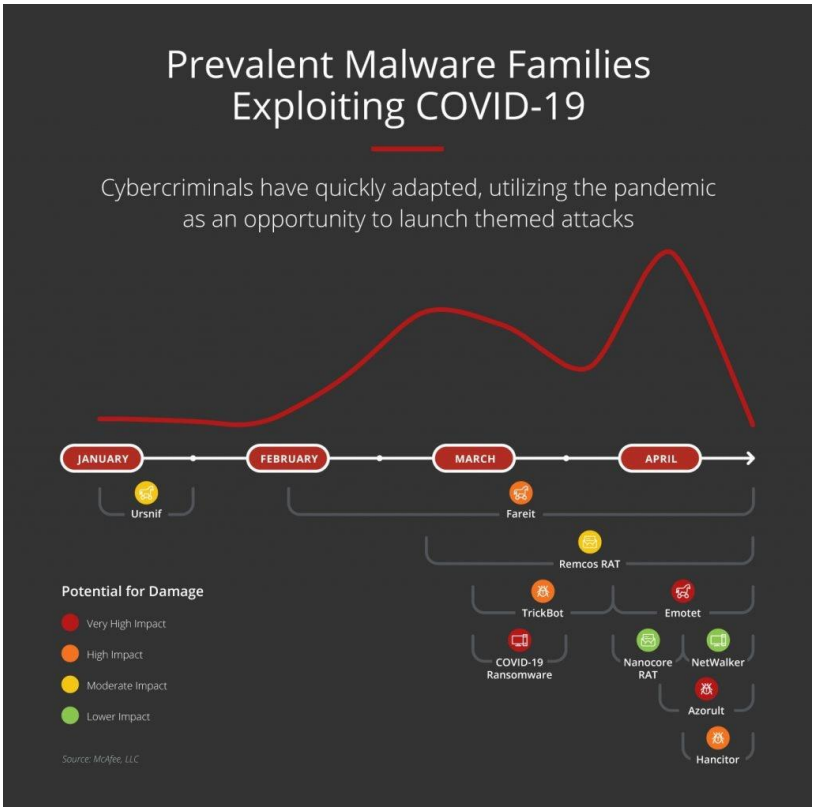

Рис. 1.1 – Динаміка росту кібератак на початку пандемії [1]

Класифікація мережевих атак:

- віддалене проникнення (англ. remote penetration) – атака яка реалізовує віддалене управління комп'ютером через мережу;
- локальне проникнення (англ. local penetration) – атака яка веде до несанкціонованого доступу до вузла;

- віддалена відмова в обслуговуванні (англ. remote denial of service) – такий тип порушує функціонування системи в рамках глобальної мережі;
- локальна відмова в обслуговуванні (англ. Local denial of service) – тип атаки який, порушує функціонування системи в рамках локальної мережі;
- атаки з використанням мережевих сканерів (від англ. Network scanners) – засновані на використанні мережних сканерів – такі програми визначають сервіси доступні для атаки;
- атаки з використанням сканерів вразливостей (від англ. Vulnerability scanners) – базуються на сканерах вразливостей – які здійснюють пошук вразливостей на вузлах, які далі будуть використані для атаки;
- атаки з використанням зломщиків паролів (від англ. Password crackers) - атаки, які базуються зломщиках паролів – програм, що підбирають паролі користувачів;
- атаки з використанням аналізаторів протоколів (від англ. Sniffers) - базуються на аналізаторах протоколів – прослуховуючих мережний трафік.

Класифікація яка буде на рис. 1.2 являється закінченою і охоплює майже всі можливі дії потенційного зловмисника. Але для запобігання атакам і цього мало, бо використовуючи її в такому виді не дає можливості визначати елементи мережі, на які може бути здійснена та чи інша атака, а також те до чого може призвести успішна атака. А це означає що організація захисту інформації в корпоративній мережі має починатися з побудови моделі загроз безпеки.

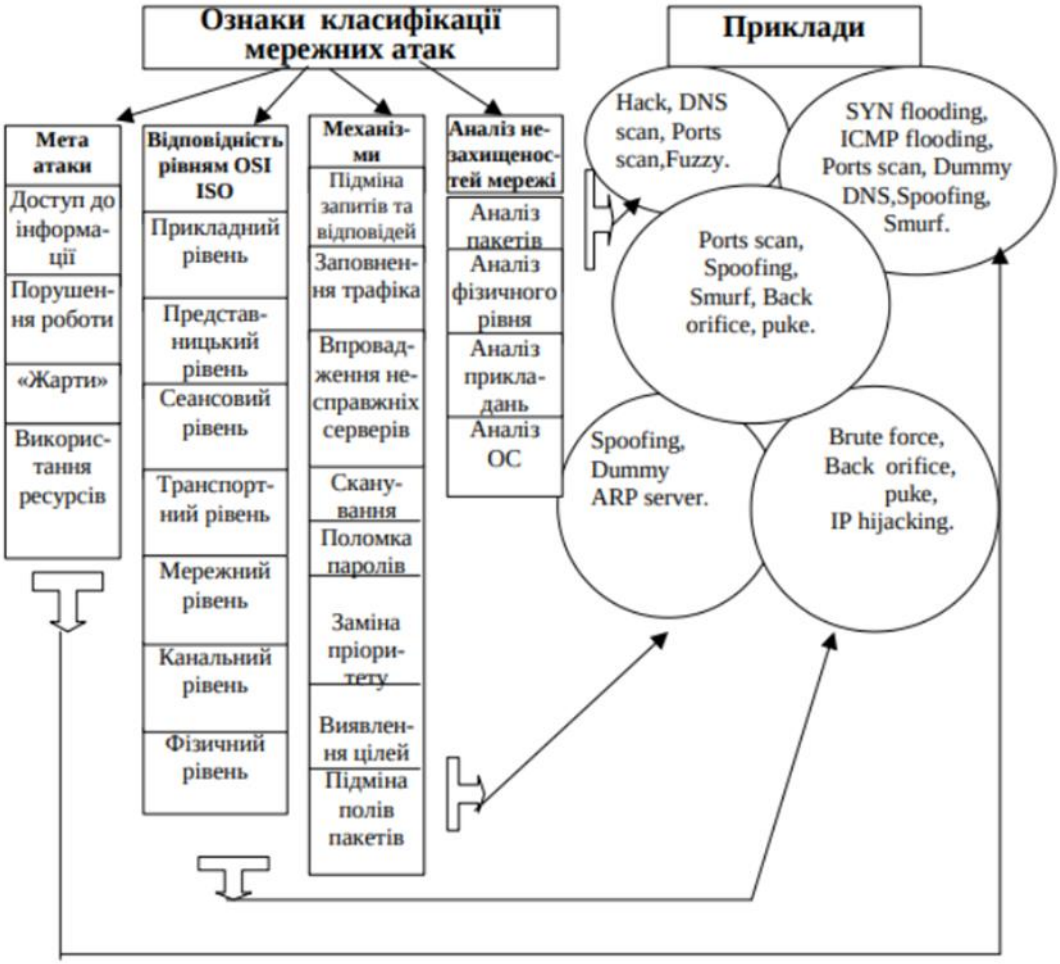

Рис. 1.2 Класифікація атак різних рівнів

1.2. Аналіз проблеми забезпечення безпеки мережі з застосуванням технології VPN

Останнім часом термін VPN з'являється в кожній розмові про захист даних в мережі. І не просто так. Не так давно технологія VPN була високотехнологічної новинкою, але сьогодні це необхідний інструмент для кожної організації, чи користувача який бажає приховати свої дані серед інших даних користувачів. По суті, технологія VPN захищає конфіденційність даних в мережі (Рисунок 1.3).

Рис. 1.3 - Загальна структура мережі VPN

Віртуальна приватна мережа (VPN) - це набір протоколів, який створює безпечне зашифроване з'єднання через менш захищену мережу, наприклад, загальнодоступний Інтернет. VPN використовує протоколи тунелювання для шифрування даних на кінці відправлення та дешифрування на кінці прийому. Щоб забезпечити додаткову безпеку, вихідні та приймаючі мережеві адреси також шифруються.

Технологія VPN може шифрувати всі дії користувача чи працівника організації в інтернеті. Всі дані, що відправляє користувач і отримуєте. Якщо користувач буду входити в мережу лише через VPN, зловмисник не зрозуміє з якої адреси підключився до джерела користувач, а бачитиме лише один із багаточисленних VPN-маршрутизаторів.

Одним з найважливіших завдань технології VPN є захист потоків корпоративних даних, що передаються по відкритих мережах. Відкриті канали можуть бути надійно захищені лише криптографічним методом. Так звані виділені лінії не мають особливих переваг перед лініями загального користування в плані інформаційної безпеки. Виділені лінії хоч би частково розташовуватимуться в неконтрольованій зоні, де їх можуть пошкодити або здійснити до них

несанкціоноване підключення. Єдина реальна перевага - це гарантована велика пропускна спроможність виділених ліній, а не підвищена захищеність.

Принцип роботи VPN не суперечить основним мережевим протоколам і технологіям. Наприклад, при установлені з'єднання віддаленого доступа, клієнт посилає серверу потік даних стандартного протокола PPP. У разі організації віртуальних виділених ліній між локальними мережами їх роутери також обмінюються пакетами PPP. Проте, принципіально новим моментом являється пересилка даних через безпечний (зашифрований) тунель, організованний в межах загальнодоступної мережі.

Якщо є потреба великої кількості ресурсів, розподілених в багатьох мережах, і проблемою являється конфіденційність переміщення інформації в цих мережах, то VPN буде потрібним вибором. В цьому заключається перевага VPN – можна просто захистити від зловмисників всю мережу.

Якщо необхідні хости, які створюють відчуття, що вони знаходяться в одній мережі, VPN - це спосіб реалізувати подібне рішення. Це дійсно зручно, якщо ви працюєте з клієнтами, яким необхідний простий доступ до головних філіалів, або якщо компанія хоче отримати повний и безпечний доступ до локальної мережі.

Використання VPN – це відносно дешевий спосіб з'єднання фізично віддалених мереж. При цьому немає необхідності в оплаті WAN-конектів, так як весь трафік між мережами передається за допомогою інтернету.

VPN базується на трьох основних методах, які застосовуються при реалізації заходів безпеки в мережах:

1) тунелювання;
2) аутентифікація;
3) шифрування.

Тунелювання це процес, у ході якого створюється захищене логічне з'єднання між двома кінцевими точками за допомогою інкапсуляції різних протоколів. Тунелювання є методом будівництва мереж, при якому один мережевий протокол інкапсулюється в інший. Від звичайних багаторівневих мережевих моделей (таких як OSI або TCP/IP) тунелювання відрізняється тим, що инкапсулюємий протокол

відноситься до того ж або більш низького рівня, ніж використаних у якості тунелю [2].

Концепція тунелювання полягає в тому, щоб «упакувати» передану порцію даних, разом зі службовими полями, у новий «конверт» для забезпечення конфіденційності та цілісності всієї переданої порції, включаючи службові поля. Тунелювання може застосовуватися на мережевому та на прикладному рівнях. Комбінація тунелювання та шифрування дозволяє реалізувати закриті віртуальні приватні мережі (VPN). Тунелювання зазвичай застосовується для узгодження транспортних протоколів або для створення захищеного з'єднання між вузлами мережі (Рисунок 1.4).

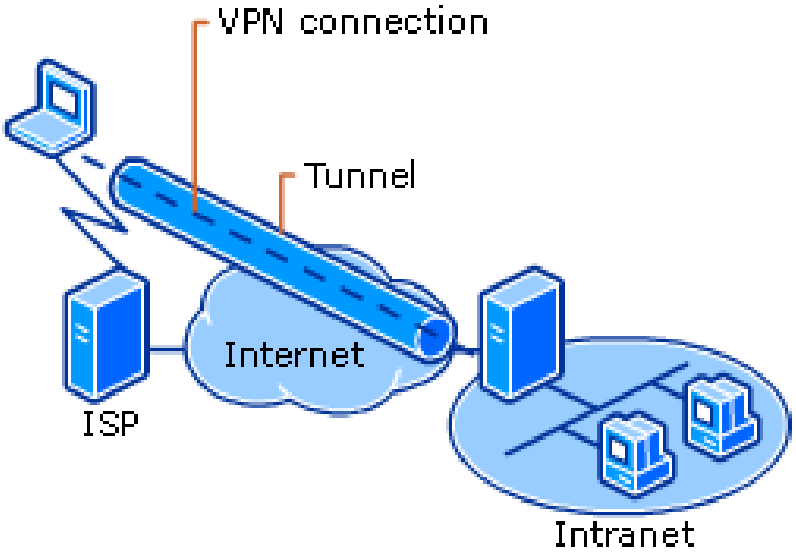

Рис. 1.4 - Модель тунельного режиму

Метод аутентифікації є основною функцією забезпечення безпеки VPN. Всі дані від комп'ютерів-клієнтів проходять через Internet до VPN-сервера. Такий сервер може знаходитися на великій відстані від клієнтського комп'ютера, і дані на шляху до мережі організації проходять через обладнання безлічі провайдерів. Як переконатися, що дані не були перехоплені або змінені? Для цього застосовуються різні методи аутентифікації і шифрування.

Для аутентифікації користувачів (PPTP) може використовувати будь-який з протоколів:

- EAP або Extensible Authentication Protocol;

- MSCHAP або Microsoft Challenge Handshake Authentication Protocol (версії 1 і 2);
- CHAP або Challenge Handshake Authentication Protocol;
- SPAP або Shiva Password Authentication Protocol;
- PAP або Password Authentication Protocol.

Шифрування гарантує, що майже ніхто не зможе отримати доступ до пакетів при пересиланні через Internet (Рисунок 1.5).

В даний час підтримуються два методи шифрування:

1) протокол MPPE шифрування або Microsoft Point-to-Point Encryption сумісний тільки з MSCHAP;
2) EAP-TLS автоматично вибирає довжину ключа шифрування при узгодженні параметрів між клієнтом і сервером.

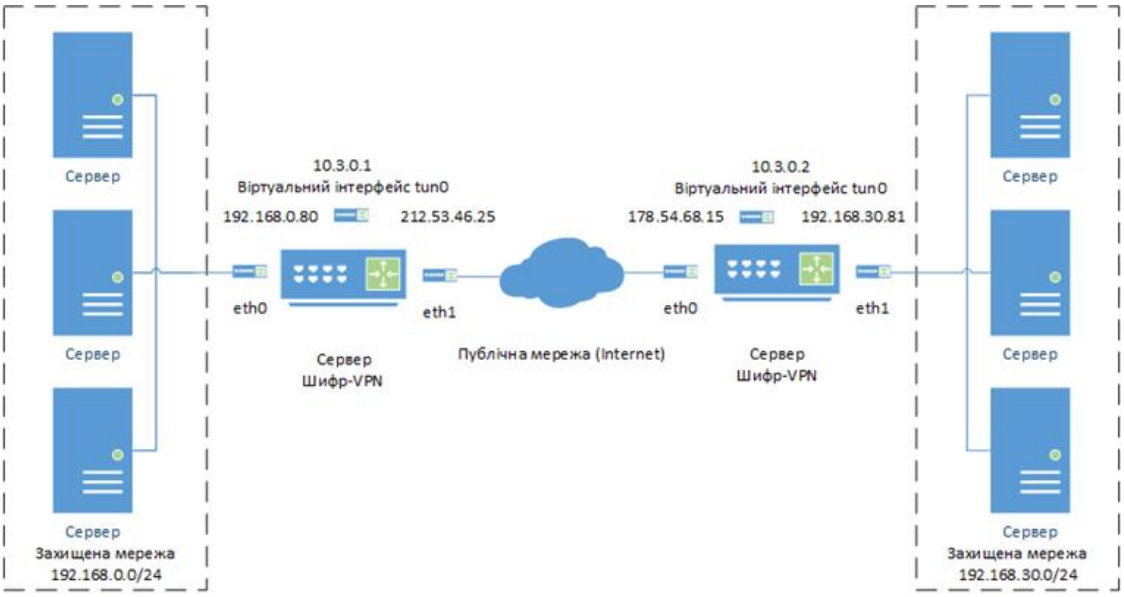

Рис. 1.5 - Модель шифрованого каналу VPN мережі

MPPE працює з ключами довжиною 32, 40, 56, 64, 72, 128 або 256 біт. Операційні системи Windows підтримують шифрування з довжиною ключа тільки 32 та 40 біт, тому в змішаному середовищі Windows слід вибирати довжину ключа яка підтримується усіма пристроями.

Протокол MMPE розроблявся для каналів зв'язку точка-точка, в яких пакети передаються послідовно, і втрата пакетів неможлива. У цій ситуації значення

ключа для чергового пакета залежить від результатів дешифрування попереднього пакета. При побудові віртуальних мереж через мережі загального доступу, цих умов дотримуватися неможливо, так як пакети даних часто приходять до одержувача не в тій послідовності, в якій були відправлені. Тому PPTP використовує для зміни ключа шифрування порядкові номери пакетів. Це дозволяє виконувати дешифрацію незалежно від попередніх прийнятих пакетів.

1.3. Аналіз існуючих технологій VPN для забезпечення безпеки інформаційних мереж

Відомо, що мережі, що використовують протокол IP, мають слабке місце, зумовлене структурою протоколу IP. Розробники IP не мали наміру забезпечувати будь-які функції безпеки на рівні IP, а гнучкість IP дозволяє хитромудро використовувати особливості даного протоколу з метою подолання контролю за трафіком, управління доступом та інших заходів безпеки. Тому дані в мережі, що використовує протокол IP, можуть бути легко підроблені або перехоплені.

При тунелюванні передачі даних по мережі протокольних пакетів мережі одного типу вони вставляються або інкапсулюються в протокольні пакети іншої мережі. Це забезпечує безпеку під час передачі даних [15].

Безліч продуктів VPN як комерційні, так і з відкритим кодом доступні на ринку. Всі продукти VPN ділиться на чотири категорії:

- VPN на основі протоколу PPTP;
- VPN на основі протоколу IPSec;
- VPN на основі SSL;
- OpenVPN.

1.3.1. VPN на основі протоколу PPTP

Одним із протоколів VPN є протокол точка-точка (PPTP) розроблений Microsoft та Ascend у 1999 році. Протокол PPTP офіційно зареєстрований як RFC263. Також PPTP клієнт був включений у Windows з 1995 року і досі включений до більшості операційних систем. В даний час протокол PPTP вважається небезпечним, оскільки надійність захищеного з'єднання безпосередньо пов'язана з надійністю самого з'єднання. Наприклад аутентифікація (пароль). Таким чином ненадійний пароль призводить до небезпечного з'єднання VPN (Рисунок 1.6).

Більшість мереж PPTP використовують протокол MSCHAPv2 для шифрування паролів. Безпека протоколу PPTP використовує сертифікати «X. 509» для захисту PPTP-з'єднання що призводить до досить безпечного з'єднання. Однак не всі клієнти PPTP підтримують EAP-TLS яка потрібна для дозволу використання сертифікатів «X. 509.». PPTP використовує два канали:

- перший канал керування для налаштування з'єднання;
- другий канал передачі даних [3].

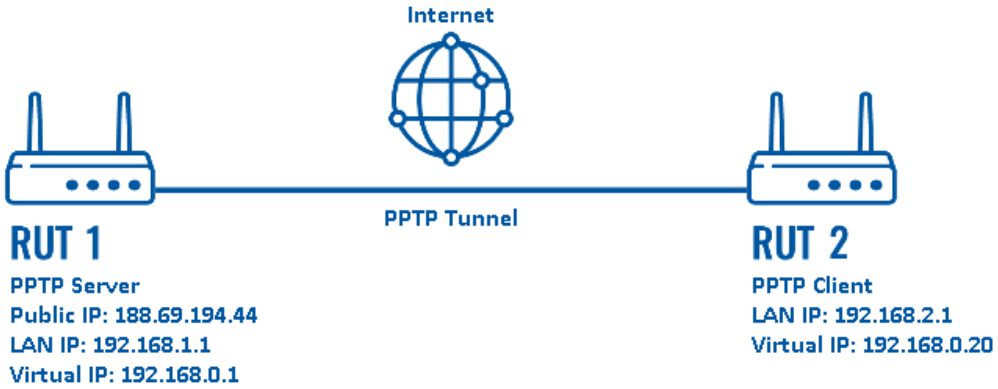

Рис. 1.6 – Модель VPN мережі протоколу точка-точка

Канал керування встановлюється через TCP-порт. Канал даних використовує загальну інкапсуляцію маршрутизації (GRE) протоколу, який є IP-протоколом.

PPTP-клієнти доступні практично на всіх операційних системах починаючи від Windows до Linux та Unix і також для пристроїв iOS та Android.

Перевагою VPN на основі PPTP є те, що клієнтське програмне забезпечення VPN вбудоване у більшість операційних систем. Крім того, час запуску для налаштування та ініціалізація PPTP VPN з'єднання відбувається дуже швидко. Недоліками VPN на основі PPTP є відсутність безпеки та параметри конфігурації як на стороні клієнта, так і стороні сервера.

EAP-TLS розширення, яке дозволяє використовувати сертифікати «X.509» повністю підтримуються лише в Microsoft Windows, але існує patch для pppd з відкритим кодом. Patch pppd включений майже в кожен дистрибутив Linux. Крім того, якщо необхідно використовувати EAP-TLS, то простота налаштування PPTP VPN значно знижується. Це тому, що EAP-TLS потрібно налаштувати інфраструктуру відкритих ключів, як IPSec та OpenVPN. Ще одним серйозним недоліком PPTP є використання протоколу GRE, який робить його несумісним із пристроями за межами NAT.

1.3.2. VPN на основі протоколу IPSec.

IPSec є офіційним стандартом IEEE/IETF для IP-безпеки. Офіційно зареєстрований як RFC2411. IPSec працює на рівні другої та третьої моделі OSI. Включає поняття політики безпеки, що робить її надзвичайно гнучкою і потужною, але також важко налаштовується. Політики безпеки дозволяють адміністратору шифрувати трафік між двома кінцевими точками на основі параметрів, таких як IP-адреса джерела та IP-адреса призначення, а також між вихідним і кінцевим портами TCP чи UDP. IPSec можна налаштувати на використання попередньо розділених ключів або сертифікатів для захисту підключення VPN. Крім того, він використовує сертифікати «X.509», одноразові паролі, протоколи імен користувача або пароль для аутентифікації VPN-з'єднання [6].

Існує два режими роботи в IPSec:

- тунельний режим;
- транспортний режим.

Транспортний режим використовується найчастіше у поєднанні з тунелюванням другого рівня (L2TP). Протокол L2TP виконує автентифікацію користувача. Клієнти IPSec, вбудовані в операційні системи, зазвичай працюють в парі IPSec та L2TP. IPSec VPN-клієнт вбудований у Microsoft Windows за замовчуванням використовує протокол IPSec-L2TP, але його можна вимкнути або оминути (Рисунок 1.7).

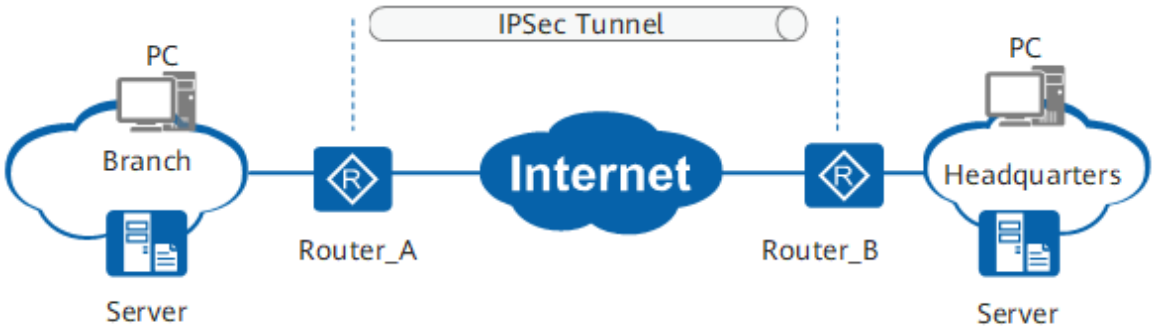

Рис. 1.7 – Можель VPN мережі протоколу L2TP з IPSec

Цілісність IPSec пакетів забезпечується за допомогою автентифікації повідомлення (HMAC) це той самий метод, який використовує OpenVPN. Одним з основних недоліків IPSec є те, що багато постачальників реалізували розширення до стандарту, що робить його більш складним щоб з'єднати дві кінцеві точки IPSec від різних постачальників. Програмне забезпечення IPSec входить до складу операційних систем, а також брандмауерів, маршрутизаторів та мікропрограмного забезпечення.

Переваги протоколу IPSec – це його безпека, хороша підтримка з боку різних постачальників та платформ включаючи маршрутизатори xDSL та Wi-Fi, а також можливість використання FineGrained політик безпеки для управління потоком трафіку. Недоліками IPSec є те, що його, як відомо, важко налаштувати і налагодити. IPSec не добре інтегрується з мережами NAT, не рекомендується, а іноді навіть неможливо запускати сервер IPSec, який знаходиться в мережі NAT.

1.3.3. VPN на основі SSL

SSL – це протокол безпеки, який забезпечує безпечне з'єднання для протоколів рівня додатків на основі TCP

SSL VPN заснований на протоколі SSL/TLS і використовує практичні переваги стандартних браузерів, які мають вбудовану підтримку SSL/TLS для розширення функцій своєї програми. На додаток до веб-доступу та додатків TCP/UDP, SSL VPN також може захищати IP-зв'язок. Зв'язок SSL VPN заснований на стандарті TCP/UDP, не обмежується NAT і може проходити через брандмауери, надаючи користувачам доступ до ресурсів інтрамережі через віртуальні шлюзи SSL VPN у будь-якому місці, роблячи віддалений доступ до безпеки більш гнучким та простим, значно скорочуючи розгортання та обслуговування підприємства

1.3.4. Технологія OpenVPN

OpenVPN це VPN на основі SSL, оскільки він використовує протоколи SSL та TLS для захищеного з'єднання. Однак OpenVPN також використовує HMAC у поєднанні алгоритму хешування для забезпечення цілісності пакетів. OpenVPN може бути налаштований для використання попередньо розділених ключів, а також сертифікатів. Ці функції зазвичай не доступні іншими VPN на основі SSL.

OpenVPN використовує віртуальний мережевий адаптер пристрій tun або tap, як інтерфейс між програмним забезпеченням OpenVPN і операційними системами.

Як правило будь-яка операційна система підтримує пристрій tun або tap та може працювати з OpenVPN. На теперішній час це Linux, NetBSD, Solaris, AIX, Windows та Mac OS, а також пристрої iOS та Android. Для всіх цих платформ клієнтське програмне забезпечення має бути встановлено, що відрізняє OpenVPN від client-less або веб-VPN.

Протокол OpenVPN не визначений у стандарті RFC, але протокол є загальнодоступним, тому що OpenVPN це частина програмного забезпечення з відкритим вихідним кодом. Факт того що він є відкритим вихідним кодом, практично робить OpenVPN більш безпечним ніж closedsource VPN так як код постійно перевіряється різними людьми. Також існує дуже мало шансів, що секретні бекдори будуть вбудовані в OpenVPN.

OpenVPN має визначення каналу управління та каналу передачі даних, обидва з яких шифруються та захищаються по-різному. Проте весь трафік проходить через одне з'єднання UDP або TCP. Канал управління шифрується та захищається з за допомогою SSL та TLS каналів, також дані шифрується за допомогою користувача протоколу шифрування. Протокол та порт за замовчуванням для OpenVPN це UDP та порт 1194 [8].

Перевагами OpenVPN є простота установки конфігурації та можливість встановлення в обмежених мережах, включаючи мережі NAT. Крім того OpenVPN включає функції безпеки, які так само сильні, як і VPN на основі IPSec, включаючи апаратний маркерний захист та підтримку для різних користувачів механізму автентифікації.

Недоліки OpenVPN знаходяться у його відсутності масштабованості та залежність у встановленні клієнтського програмного забезпечення. Ще одним недоліком є відсутність графічного інтерфейсу для налаштування та управління. Зокрема драйвер інтерфейсу tap для Microsoft Windows часто викликав проблеми розгортання, коли випускалася нова версія Windows.

Таблиця 1.1

Порівняльна характеристика протоколів PPTP, L2TP/IPSec та OpenVPN

	PPTP	L2TP/IPSec	OpenVPN
Інформація	Тунельний протокол другого рівня типу з'єднання «точка-точка» (Point-to-Point Tunneling Protocol). Розроблений компанією Microsoft. Довгий час був стандартом для VPNмереж. Володіє високою швидкістю та підтримується більшістю пристроїв які працюють з VPN.	Протокол тунелювання другого рівня (Layer 2 Tunneling Protocol) – VPN протокол який сам не забезпечує безпеку та цілісність даний, а лише дозволяє створювати тунель. Зазвичай використовується в сукупності з IPSec для забезпечення надійного шифрування трафіку. При налаштування L2TP/IPSec не виникає складності, також влаштований майже у всі ОС.	OpenVPN – рішення з відкритим кодом для побудови VPN мереж. Використовує бібліотеку OpenSSL. Гнучкий протокол, має можливість працювати на будь якому порту
Алгоритм шифрування	MPPE-128.	AES-128 для даних, та SHA256 для повідомлень з контрольними сумами.	AES-256 для даних, SHA256 для повідомлень з контрольними сумами и 2048-бітне шифрування SSL/TLS.

Продовження таблиці 1.1

Порівняльна характеристика протоколів PPTP, L2TP/IPSec та OpenVPN

Безпека	Має відомі проблеми з безпекою.	При правильну використанні не має значних вразливостей.	Надійний при використанні надійних ключів шифрування.
Переваги	- Просте налаштування; - висока швидкість роботи; - влаштований клієнт для більшості ОС.	- Просте налаштування та встановлення; - Підтримується більшістю сучасних пристроїв та ОС.	- Гнучке налаштування; - швидкий та надійний; - обходить більшість мережевих екранів та блокувань від провайдерів.
Недоліки	- Низький рівень безпеки; - Стабільність роботи залежить від якості з’єднання/зв’язку; - може бути заблокований.	- Повільніший в порівнянні з іншими протоколами; - може бути заблокований.	- Більш складне налаштування; - Потребує встановлення додаткового програмного забезпечення.

Як видно з таблиці 1.1, протоколи і стандарти мають різні характеристики, деякі пропонують високий рівень захисту, в той час як інші демонструють більшу продуктивність. Тут все залежить виключно від потреб і завдань, так як можна жертвувати продуктивністю заради отримання більшого рівня захисту і навпаки.

Таким чином, із перерахованих VPN протоколів можна виділити PPTP як самий розповсюджений та легкий в налаштуванні, L2TP/IPSec протокол який забезпечує високу безпеку, OpenVPN як найбільш універсальне та гнучке рішення для побудови VPN-мереж.

Базуючись на результатах дослідження, можемо зробити висновки, що OpenVPN спрямований на забезпечення швидкої взаємодії між мережами та елементами мережі, а також надійного захисту.

2 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ЗАБЕЗПЕЧЕННЯ ЗАХИЩЕНОГО ОБМІНУ ДАНИМИ З ВИКОРИСТАННЯМ OPENVPN

2.1. Дослідження можливостей та функцій технології OpenVPN

Для того, щоб вирішити проблему побудови та підтримки віртуальних мереж на початку 2002 року була створена перша версія технології OpenVPN, яка будувалася як новий інструмент з відкритим вихідним кодом і використовувалась тільки для створення VPN мереж типу site-to-site що працювали на основі протоколів типу SSL / TLS або протоколу з розподіленими ключами. Технологія OpenVPN створює так званий VPN тунель, що захищає інформацію при передачі даних через один TCP / UDP порт в незахищену відкриту мережу Інтернет (Рисунок 2.1).

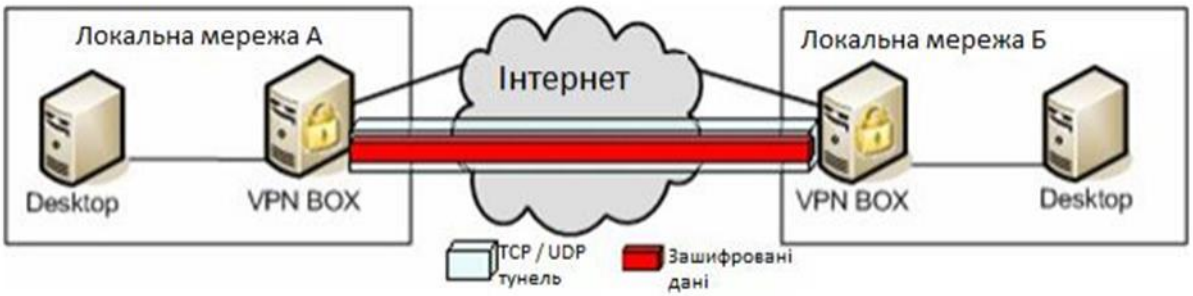

Рис. 2.1 – Модель роботи OpenVPN тунелю

Основні переваги технології OpenVPN – легкість в інсталяції, можливість встановлення в різні системи та зручне налаштування. Технологія OpenVPN може бути встановлена на такі розповсюджені платформи, як: Linux, Windows 2000/XP/ Vista, OpenBSD, FreeBSD, NetBSD, Mac OS X і Solaris, за умови шо Linux системи будуть працювати на ядрі 2.4 або новіше. Для того, щоб встановити та налаштувати програму в різних системах необхідно провести однакові маніпуляції. Технологія OpenVPN працює з архітектурою типу клієнт-сервер.

Додаток необхідно встановити на всі елементи VPN мережі, при цьому один з вузлів повинен бути сервером, а інші клієнтами. Додаток OpenVPN створює між вузлами TCP або UDP тунелі, в яких дані, що проходять між ними, зашифровуються. UDP 1194 вважається стандартним портом в OpenVPN, але додаток дає можливість змінити порт на будь-який інший, зручний для адміністратора UDP або TCP порт [9].

Додаток OpenVPN створює систему, що може працювати в двох режимах. Якщо використовувати режим розподілених ключів, буде створений один ключ, який буде зберігатись у відправляючій та приймаючій сторонах для шифрування і розшифрування. В такому випадку сам ключ повинен бути секретним, тобто для його передачі повинна використовуватись симетрична криптографія. Проблема даного режиму полягає в передачі ключа на приймаючу сторону.

Для того, щоб не ризикувати інформацією та запобігти виникненню цієї проблеми необхідно використовувати Інфраструктуру Відкритих Ключів (PKI). Суть у тому, що кожен вузол зберігає два ключі: відкритий ключ, що зберігається у всіх та закритий ключ, доступ до якого є тільки у власника. Подібна структура використовується у OpenSSL, інтегрованому в OpenVPN, для аутентифікації окремих VPN вузлів для передачі зашифрованих даних.

Таблиця 2.1

Переваги розподілених ключів та SSL

Режим OpenVPN	Розподілені ключі	SSL
Шифрування	Симетричний	Асиметричний та симетричний
Реалізація	Простіше	Складніше
Швидкість	Швидко	Повільно
Завантаження процесора	Менше	Більше
Обмін ключами	Так	Ні
Спроба оновити ключі	Ні	Так
Аутентифікація вузлів	Ні	Так

В режимі SSL (асиметричного шифрування) використання OpenVPN більш безпечне та надійне. Цей протокол SSL (Secure Sockets Layers) був розроблений компанією “Netscape” ще в 90-х роках. Загалом було створено та випущено версію протоколу v2 (1994) та v3 (1995). У 2001 IETF купила і оновила патент на цю розробку. В цей же рік назва SSL була змінена на TLS (Transport Layer Security) (RFC 2246). Сама абревіатура SSL в більшості випадків використовується при використанні і SSL і TLS протоколів.

SSL вирішує дві основні проблеми:

- аутентифікує сервер і клієнта за допомогою засобів Інфраструктури Відкритих Ключів (PKI);
- шифрує повідомлення між клієнтом і сервером та створює нові з'єднання.

SSL працює між транспортним рівнем і рівнем додатку та буде шифрувати дані на рівні програми.

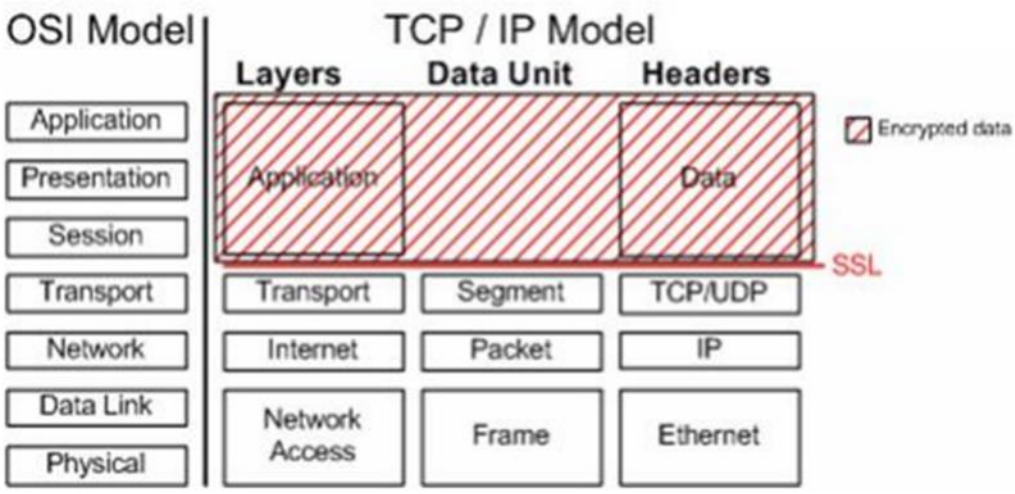

Рис. 2.2 – Розташування протоколу SSL в моделі OSI

Раніше технологію SSL використовували тільки специфічні додатки типу HTTP, однак на сьогоднішній день вона може працювати в сфері забезпечення безпечного та стабільного з'єднання через мережу Інтернет або створення та підтримки шифрованих тунелів (VPN).

Розрізняють два окремих типи VPN:

- клієнт-сервер (або віддалений доступ) VPN, де клієнт використовує стандартний web браузер такий як Firefox, Google Chrome, тощо;
- Site-to-site, створює та утримує зв'язок за допомогою спеціального програмного забезпечення, такого як OpenVPN.

Робота технології SSL відбувається в 4 основних етапи (Таблиця 2.2).

Таблиця 2.2

Чотири етапи SSL та TLS

SSL Handshake	Визначається метод шифрування для передачі даних
SSL Change Cipher Spec	Створення і передача ключа між клієнтом і сервером на цю сесію
SSL Alert	Доставка повідомлень SSL про помилки між клієнтом і сервером
SSL Record	Передача даних

Для того, щоб шифрувати дані і аутентифікувати користувачів OpenVPN використовується технологія OpenSSL, яка на даний момент являється найкращим безкоштовним рішенням і поширюється з відкритим вихідним кодом.

Технологія OpenSSL – це інструмент, що має в собі:

- бібліотеки шифрування;
- інструменти для роботи з командним рядком;
- SSL бібліотеки.

Бібліотеки шифрування створюють та обробляють велику кількість алгоритмів шифрування, таких як:

- симетричні алгоритми: DES, AES, 3DES, Blowfish і інші;
- сертифікати: x509;
- хеш-функції: MD5, HMAC.

2.2. Компоненти та архітектура OpenVPN

Одним з основних блоків OpenVPN є драйвер tun та tap. Концепція драйвера tun та tap походить з Unix та Linux, де вони часто доступні як частина операційної системи. Це віртуальні мережеві адаптери, що розглядаються операційною системою як двоточковий адаптер (в стилі tun) для трафіку IP або як повноцінного віртуального адаптера Ethernet для всіх типів трафіку (у стиль tap). На внутрішній стороні цих адаптерів знаходиться програма така як OpenVPN для обробки вхідний та вихідний пакетів трафіку. Linux, Free / Open / NetBSD, Solaris і Mac OS включають драйвер ядра tun [11].

Нещодавно аналогічний драйвер був доданий до AIX, який похідна від Unix від IBM. Для Microsoft Windows було написано спеціальний драйвер NDIS який називається адаптером TAP-WIN32. На даний момент версії NDIS5 та NDIS6 драйвера доступні, підтримуючи Windows XP та вище. Розробка цього адаптера тепер офіційно відокремлена від основного OpenVPN, але OpenVPN продовжує сильно покладатися на нього.

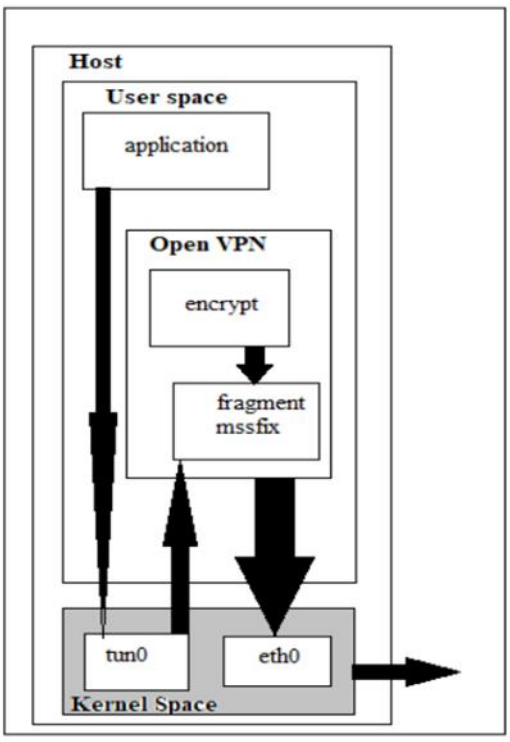

Рис. 2.3 - Потік трафіку з клієнтського додатку через OpenVPN

Потік трафіку з програми користувача через OpenVPN зображений на рис 2.3 де програма відправляє трафік на адресу, доступну через тунель OpenVPN в кілька кроків:

1. Програма передає пакет операційній системі.
2. ОС вирішує використати звичайні правила маршрутизації (пакет повинен маршрутизуватися через VPN).
3. Пакет пересилається на пристрій ядра.
4. Пристрій налаштування ядра пересилає пакет до OpenVPN.
5. Процес OpenVPN шифрує та підписує пакет та фрагментує його при необхідності, а також знову передає його до ядра (щоб відправити його адресу віддаленої кінцевої точки VPN).
6. Ядро забирає зашифрований пакет та пересилає його на віддалену кінцеву точку VPN, де відбувається зворотний процес.

Нажаль продуктивність OpenVPN завжди буде нижчою, ніж у звичайного мережного підключення. Для більшості додатків, втрата продуктивності мінімальна та допустима. Однак для швидкостей, що перевищують 1 Гбіт/с, існує слабке місце в продуктивності як з точки зору пропускної здатності так і з затримки. Слід зазначити, що продуктивність драйвера Windows набагато нижче ніж продуктивність вбудованих адаптерів tun і tap інших операційних системах. Це вірно навіть для самої реалізації драйвера TAP-Win32 в NDIS6. Для одного клієнта OpenVPN впливу досить мало. Для великомасштабного сервера OpenVPN, який обслуговує багато клієнтів, це може легко викликати проблеми з продуктивністю. Це одна з головних причин того, що спільнота розробників відкритого вихідного коду зазвичай рекомендує використовувати хост на основі Unix або Linux як сервер OpenVPN.

OpenVPN підтримує два способи зв'язку між кінцевими точками, використовуючи UDP чи TCP. UDP – це протокол без встановленого з'єднання або із втратами протоколу. Якщо пакет втрачається під час передачі, то мережеве з'єднання непомітно це виправить. TCP - це протокол орієнтований на з'єднання. Пакети відправляються та доставляються за протоколом handshake, забезпечуючи

доставку кожного пакета на інший бік. Обидва способи зв'язку мають свої переваги та недоліки. Це насправді залежить від типу трафіку, який надсилається через VPN-тунель [5].

Використання програми на основі TCP через VPN може призвести до подвійної втрати продуктивності особливо якщо є погане підключення до мережі. У цьому випадку повторюється передача втрачених пакетів, втрачених як усередині, так і зовні тунелю, що призводить до зниження продуктивність. Проте аналогічно можна стверджувати, що відправка пакетів через UDP також не є відмінною ідеєю. Якщо програми, що використовують протокол UDP для свого трафіку сприйнятливого до атак видалення чи переупорядкування, а базове зашифроване TCP з'єднання підвищить безпеку таких програм навіть більше, ніж базовий VPN на основі UDP. Якщо більша частина трафіку через VPN заснована на UDP, тоді краще використовувати з'єднання TCP між кінцевими точками VPN.

При виборі між UDP або TCP, загальне правило таке: якщо у вас працює UDP (mode udp), то використовуйте його; якщо ні, то спробуйте TCP (режим tcp-сервера та режим tcp-клієнта). Деякі комутатори та маршрутизатори неправильно пересилають трафік UDP, що може бути проблемою, особливо якщо кілька клієнтів OpenVPN підключено до одного комутатора або маршрутизатору. Так само на продуктивність OpenVPN через TCP може сильно вплинути вибір Інтернет-провайдерів (ISP), деякі провайдери використовують дивні розміри MTU або пакети, фрагментованими правилами, що призводить до вкрай низької продуктивності OpenVPN-overTCP в порівнянні з незашифрованим TCP-трафіком.

OpenVPN реалізує TLS через UDP, але спосіб OpenVPN використання TLS відрізняється від способу веб-браузера. Таким чином, коли OpenVPN запускається через TCP, трафік відрізняється від звичайного трафіку TLS. Брандмауер, який використовує глибоку перевірку пакетів (DPI), може легко відфільтрувати трафік OpenVPN. Основна різниця між OpenVPN-TLS та browser-TLS полягає у підписі пакетів. OpenVPN пропонує функції захисту від DoS-атак за допомогою підписання пакетів каналу управління за допомогою спеціального статичного ключа (--tlsauthta.key). Пакети каналу передачі даних, які передаються по тому ж UDP або

TCP з'єднанню, підписуються зовсім по-різному і легко відрізняються від трафіку HTTPS. Це також є основною причиною, чому port-sharing, де OpenVPN і безпечний веб-сервер можуть використовувати ту саму IP-адресу і номер порту, може фактично працювати.

Режим клієнт-сервер з tun і tan пристроями.

Модель розгортання OpenVPN це один сервер з декількома віддаленими клієнтами, здатними маршрутизувати трафік. Основна різниця між режимом tun і tap – це тип використовуваного адаптера. Tap адаптер забезпечує повний віртуальний інтерфейс Ethernet (Другий рівень), в той час як адаптер tun розглядається як адаптер точка-точка (Третього рівня) більшістю операційних систем. Комп'ютери, підключені за допомогою віртуальних адаптерів Ethernet, можуть утворювати єдиний широкомовний домен. Крім того, не всі операційні системи підтримують tap адаптери. Наприклад: iOS та Android підтримують лише пристрої tun. Крім того, режим tap дозволяється налаштувати міст, де звичайна мережа адаптера з'єднана мостом із віртуальним адаптером tap

Посвідчувальний центр CA.

Видає сертифікати за запитом вузлів мережі VPN, підписані сертифікатом центру, що засвідчує. Надає вузлам VPN свій власний сертифікат для перевірки засвідчувальної сторони. Керує списком відгуків сертифікатів CRL.

Сервер OpenVPN.

ПЗ сервера OpenVPN створює тунель всередині незахищеної мережі, наприклад, Інтернету. Цей тунель забезпечує безпечний зашифрований трафік між вузлами – учасниками обміну даними у мережі OpenVPN. Існує дві серверні версії OpenVPN:

- OpenVPN Community Edition;
- OpenVPN Access Server.

OpenVPN Community Edition - проект з відкритим вихідним кодом OpenVPN, також званий Community Edition (CE), є проектом віртуальної приватної мережі з відкритим вихідним кодом. Він створює безпечні VPN-з'єднання через Інтернет з використанням налаштованого протоколу безпеки, який використовує SSL/TLS.

CE - це проект OSS (програмне забезпечення з відкритим вихідним кодом), що підтримується спільнотою. На сьогоднішній день його завантажили понад 50 мільйонів разів. Використовуючи ліцензію GPL, проект має багато розробників та учасників з OpenVPN Inc., а також більш широке співтовариство OpenVPN.

Оскільки це відкритий вихідний код, CE є повністю безкоштовним для розгортання, однак для цього потрібне глибоке розуміння Linux та використання інтерфейсу командного рядка.

OpenVPN Access Server надає корпоративне рішення VPN для підприємств у всьому світі. За допомогою цього єдиного рішення організації можуть захищати передачу даних, захищати ресурси Інтернету речей та надавати зашифрований віддалений доступ до локальних, гібридних та загальнодоступних хмарних ресурсів.

Сервер доступу надає потужний та простий у використанні веб-сайт адміністрування, який спрощує керування та налаштування VPN для всіх (зі знанням Linux або без нього). Сервер доступу об'єднує можливості сервера OpenVPN, управління доступом підприємства та пакети програмного забезпечення OpenVPN Client, які підходять для середовищ Windows, MAC, Linux та мобільних ОС (Android та iOS). Він доступний на багатьох хмарних майданчиках, таких як AWS, GCP, Azure та Oracle.

Модель ліцензування заснована на кількості активних VPN-підключень, що дозволяє OpenVPN надавати компаніям будь-якого розміру доступне та масштабоване рішення. OpenVPN Access Server можна безкоштовно встановити та використовувати для 2 одночасних VPN-підключень з метою тестування.

Всі продукти OpenVPN побудовані на протоколі OpenVPN з дотриманням найвищих стандартів безпеки, пропонуючи гнучку конфігурацію для порівняння з будь-якою мережею. Найбільша різниця полягає в тому, чи хоче системний адміністратор виконати налаштування за допомогою інтерфейсу командного рядка або використовувати онлайн-веб-інтерфейс сервера доступу.

Щодо клієнтської частини, вона встановлюється на всі вузли, яким необхідний захищений канал передачі з сервером OpenVPN. У разі відповідного

налаштування сервера OpenVPN, можлива захищена передача даних між клієнтами OpenVPN, а не тільки між клієнтами та сервером OpenVPN.

Щоб підключитися до сервера або служби VPN, вам потрібно отримати файл, який містить відомості, необхідні для підключення. Такий файл конфігурації називається профілем та має розширення .ovpn.

Якщо компанія використовує сервер доступу або OpenVPN Cloud і IT-відділ надав URL-адресу, ви можете безпосередньо імпортувати профіль, ввівши URL-адресу.

Якщо використовується Community Edition і ви отримали профіль як файл від IT-відділу або від іншої служби, сумісної з OpenVPN, ви можете імпортувати профіль, вибравши файл (Рисунок 2.4).

Рис. 2.4 - Інтерфейс клієнта OpenVPN

Сертифікат X.509.

Сертифікати X.509 являють собою публічні ключі, засвідчені центром, що засвідчує CA. Вони використовуються для зашифровування даних. Факт засвідчення сертифіката, що засвідчує центр CA, дозволяє ідентифікувати сторону, яка передає зашифровані дані.

Файл запиту на сертифікат створюється на вузлах мережі, потім він переноситься на вузол центру, що посвідчує, і там підписується. Створений в результаті підписаний сертифікат переноситься назад на вузол мережі OpenVPN [13]. Перша версія стандарту X.509 була опублікована ще 1988 року. З метою формалізації правил видачі сертифікатів Сектор стандартизації електрозв'язку МСЕ (ITU-T) розробив ієрархічну систему для визначних імен, яка дотримувалася правил служби електронних каталогів (Рисунок 2.5).

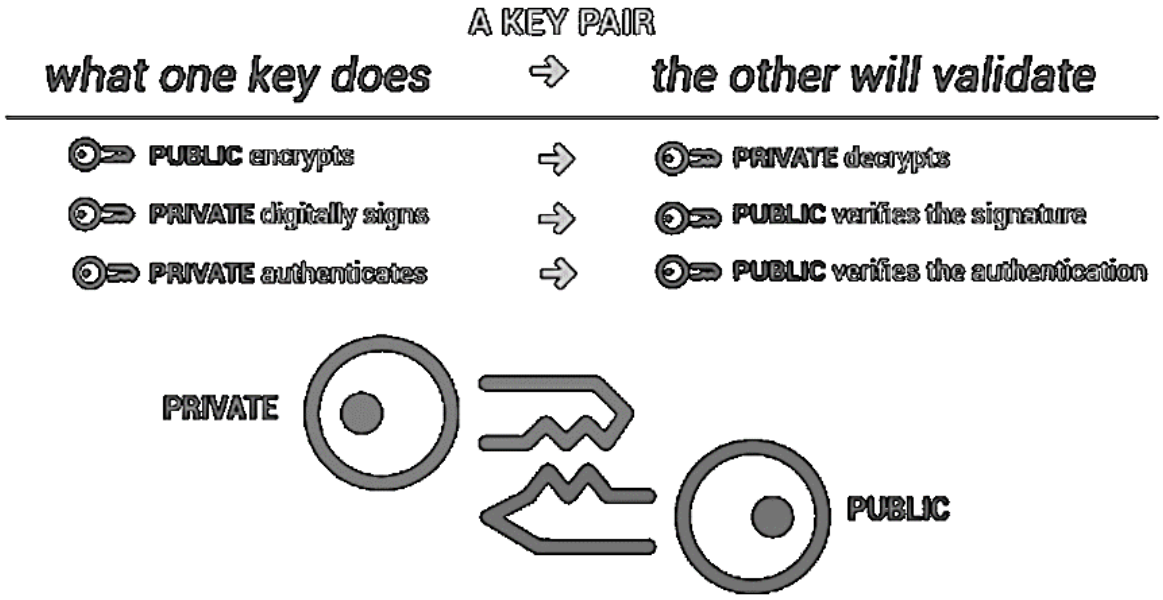

Рис. 2.5 - використання пар відкритого та закритого ключів

У 1996 році третя версія стандарту представила велике оновлення з додаванням декількох розширень, які ще використовуються сьогодні для підтримки розширення і нових додатків для використання в Інтернеті.

Тепер версія 9 – це поточна версія стандарту, яка була визначена у жовтні 2019 року.

Переваги сертифікатів X.509:

1. Довіра – цифрові сертифікати дозволяють окремим особам, організаціям і навіть пристроям встановлювати довіру у цифровому світі. В якості основи для всіх цифрових посвідчень сертифікати X.509 є повсюди і

необхідні для кожного пов'язаного процесу, від веб-сайтів до додатків, кінцевих пристроїв та онлайн-документів. Архітектура використання ключів дозволяє сертифікатам перевіряти, що:

- Відкритий ключ належить імені хоста-домена, організації або фізичній особі, що міститься в сертифікаті.
- Він був підписаний публічно довіреним центром сертифікації (CA) або самопідписаний.

2. Коли сертифікат підписується довіреним центром сертифікації, користувач сертифіката може бути впевнений, що власник сертифіката або ім'я хоста-домен були перевірені, в той час як сертифікати, що самозаверяють, можна довіряти меншою мірою, оскільки власник не проходить жодних додаткових дій.
3. Масштабованість. Додатковою перевагою цього підходу до ідентифікації на основі сертифікатів є масштабування. Архітектура PKI настільки масштабована, що може захистити мільярди повідомлень, якими організації щодня обмінюються через власні мережі та через Інтернет. Це дозволяє широко і відкрито розповсюджувати відкриті ключі, при цьому зловмисники не зможуть виявити закритий ключ, необхідний для розшифровки повідомлення.

Приватні ключі.

Приватні ключі таємні. Вони повинні створюватися та зберігатися на кожному вузлі мережі OpenVPN, призначені для розшифрування даних та ніколи не повинні передаватися по мережі.

Приватні ключі створюються на вузлах мережі OpenVPN одночасно з файлом запиту отримання сертифіката.

Список відгуків сертифікатів CRL.

Містить перелік сертифікатів, що втратили довіру. Він створюється і редагується на вузлі центру, що засвідчує CA. Щоб відключити вузол від мережі, достатньо занести його сертифікат до списку CRL.

Після створення та кожної зміни список CRL переноситься на сервери OpenVPN.

Файл Діффі-Хелмана.

Використовується, щоб у разі викрадення ключів унеможливити розшифрування трафіку, записаного ще до цього викрадення. Створюється на сервері OpenVPN.

Протокол для обміну ключами Діффі-Хеллмана (ДХ) – це метод безпечного обміну криптографічними ключами загальнодоступним каналом; він був одним із перших протоколів з відкритим ключем, спочатку концептуалізований Ральфом Мерклом і названий на честь Вітфілда Діффі та Мартіна Хеллмана. ДХ є одним із перших практичних прикладів обміну відкритими ключами, реалізованих у галузі криптографії.

Одна з фундаментальних проблем криптографії - безпечне спілкування по каналу, що прослуховується. Повідомлення потрібно зашифровувати та розшифровувати, але для цього обом сторонам потрібно мати спільний ключ. Якщо цей ключ передавати тим же каналом, то сторона, що прослуховує, теж отримає його, і сенс шифрування зникне.

Алгоритм Діффі-Хеллмана дозволяє двом сторонам отримати спільний секретний ключ, використовуючи незахищений від прослуховування, але захищений від підміни канал зв'язку. Отриманий ключ можна використовувати для обміну повідомленнями за допомогою симетричного шифрування [14].

Статичний ключ HMAC.

Служить для перевірки справжності інформації, що передається. Забезпечує захист від DoS-атак та флуду. Створюється на сервері OpenVPN.

HMAC - скорочення від англійської hash-based message authentication code, хеш-код аутентифікації повідомлень. Наявність способу перевірити цілісність інформації, що передається або зберігається в ненадійному середовищі, є невід'ємною та необхідною частиною світу відкритих обчислень та комунікацій. Механізми, які надають такі перевірки цілісності на основі секретного ключа зазвичай називають кодом автентичності повідомлення (MAC). Як правило, MAC

використовується між двома сторонами, які розділяють секретний ключ для перевірки автентичності інформації, що передається між цими сторонами. Цей стандарт визначає MAC. Механізм, який використовує криптографічні хеш-функції у поєднанні з секретним ключем називається HMAC (Рисунок 2.6).

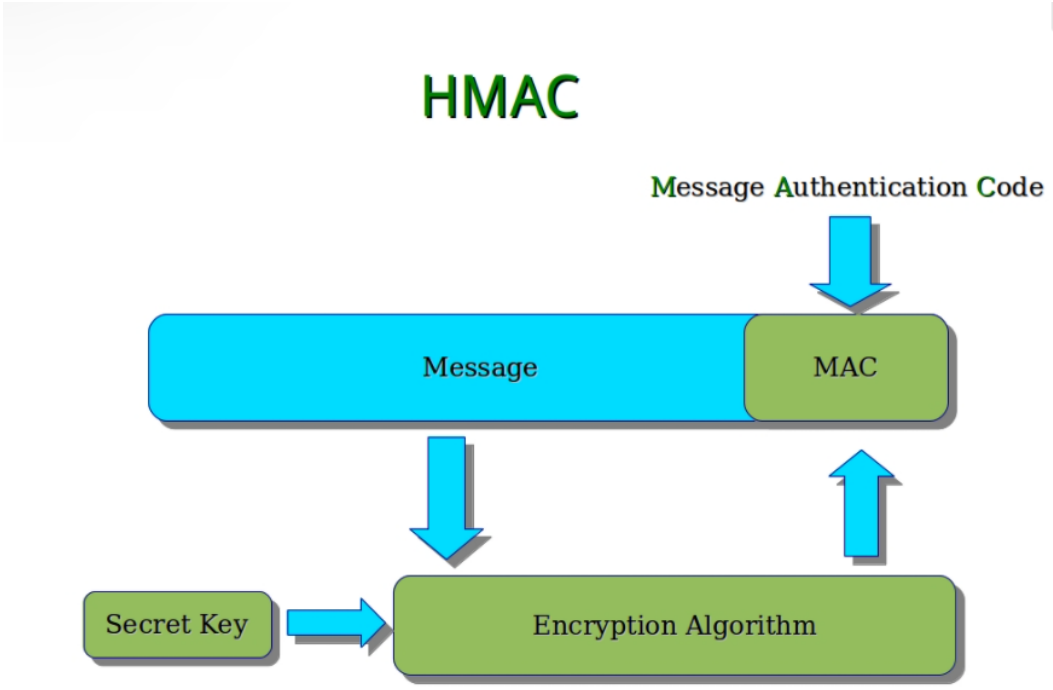

Рис. 2.6 - Принцип роботи хеш-код HMAC

Основна ціль:

1. Для того щоб можна було використовувати наявні хеш-функції без змін, зокрема, хеш-функцій, які вже є в програмному продукті, і їхній код вже доступний.
2. Щоб зберегти початкове виконання хеш-функції без жодних значних погіршень.
3. Використовувати та обробляти ключі більш простим способом.
4. Для легкої замінності базової хеш-функції в тому випадку, якщо швидша і безпечніша хеш-функція буде доступна пізніше.

Статичний ключ аутентифікації повідомлень HMAC забезпечує перевірку автентичності інформації, що передається між сторонами. Цей ключ створюється на сервері OpenVPN.

Директива tls-auth додає додатковий підпис HMAC до всіх пакетів рукостискання для перевірки цілісності. Будь-який UDP-пакет, який не має правильного HMAC-підпису, може бути відкинутий без подальшої обробки. HMAC-підпис, що встановлюється директивою tls-auth, забезпечує підвищений рівень безпеки додатково до безпеки, що надається протоколом SSL/TLS. Це може захистити від:

- DoS-атак чи флуда на UDP-порт OpenVPN.
- Сканування портів, призначеного для визначення портів, що прослуховуються сервером UDP.
- Уразливостей, пов'язаних з переповненням буфера реалізації SSL/TLS.
- Спроб ініціації SSL/TLS-рукостискання від несанкціонованої машини (хоча, зрештою, такі рукостискання не пройдуть аутентифікацію, tls-auth може відсікнути їх на більш ранній стадії).

2.3. Центр сертифікації та утиліта EasyRSA

OpenVPN користується заслуженою популярністю у системних адміністраторів, знаходячи широке застосування у найрізноманітніших сценаріях. Але часто використання OpenVPN відбувається у вигляді повторення інструкцій, без глибокого розуміння сенсу виконуваних дій. Особливо це стосується "генерації ключів", ця проста на вигляд операція має досить глибокий прихований пласт, формуючи власну інфраструктуру відкритих ключів. Відсутність розуміння структури PKI призводить до різноманітних труднощів та проблем при використанні OpenVPN.

Дослідивши продукт відомо, що для автентифікації користувачів OpenVPN використовує замість паролів сертифікати. Це дає значне підвищення безпеки. Сучасні системи не передають паролі у відкритому вигляді, використовуючи замість них досить добре захищені складними криптографічними алгоритмами

хеші. Але пароль завжди можна підібрати і це одне з найуразливіших місць. Сертифікат підібрати неможливо, його можна лише отримати від основного джерела. Але наявність сертифіката ще нічого не дає, він не є секретним і ви цілком можете його отримати, але щоб встановити захищене з'єднання з сервером вам знадобиться закритий ключ, який зберігається в надійному місці і ніколи не залишає межі клієнтського ПК .

Існує помилкова думка, що сертифікати видає саме сервер, і тому він знає, хто з клієнтів має право підключатися, а хто ні. Однак це не так і ця помилка часто призводить до різноманітних труднощів при роботі з OpenVPN.

Дослідивши це питання, визначв саме поняття інфраструктури відкритих ключів. Основна мета PKI - це забезпечення області довіри, де кожен учасник може однозначно довіряти іншому учаснику, не маючи жодної попередньої інформації про нього.

Якщо проводити аналогію, це можна порівняти з пропускною системою на прохідний. Охоронець особисто вас не знає в обличчя, ви теж, можливо, вперше його бачите, але ви пред'являєте йому перепустку встановленого зразка, і він відкриває вам турнікет. За рахунок чого це відбувається? За рахунок довіри. Перепустки видаються спеціальним відділом - бюро перепусток, авторитет та довіра, до якого незаперечні.

У нашому випадку таким «бюро перепусток» є Центр сертифікації (CA, Certification authority), навколо якого складається інфраструктура відкритих ключів. Авторитет CA є незаперечним і довіра до нього не піддається сумніву. Саме CA здійснює все управління сертифікатами як для клієнтів, так і для серверів.

Це абсолютно окрема сутність, хоча фізично вона може бути на одному вузлі із сервером OpenVPN. Найчастіше в ролі центру сертифікації використовується Easy-RSA, хоча можна використовувати будь-яке інше програмне забезпечення, скажімо, в ролі CA цілком може виступати Mikrotik. Але загальний зміст від цього не змінюється, створивши центр сертифікації, ми разом з ним створюємо область довіри, яка називається інфраструктурою відкритих ключів.

При створенні центру сертифікації ми генеруємо ключову пару із відкритого та закритого ключів. Відкритий ключ, разом з іншою інформацією про наш CA міститься в кореневому сертифікаті ca.crt, який повинен мати найширше поширення в нашій інфраструктурі, тому що саме він становить основу довірчих відносин. Маючи на руках кореневий сертифікат, ми можемо перевірити сертифікат будь-якого іншого суб'єкта і переконатися, що він виданий нашим CA і можемо йому довіряти (Рисунок 2.7).

Рис. 2.7 - Принцип використання кореневого сертифікату ca.crt

Закритий ключ центру сертифікації є найбільшою цінністю і повинен зберігатися в надійному місці, а також ніколи не залишати межі CA. Компрометація закритого ключа фактично знищує область довіри, тому що ми не можемо бути впевнені, що пред'явлений нам сертифікат випущений саме CA, а не зловмисником, що заволодів ключем.

За допомогою центру сертифікації ми формуємо ключі та сертифікати для OpenVPN-серверів та їх клієнтів. Якщо бути суворо послідовними - це неправильно, так як закритий ключ клієнта формується на вузлі CA, а потім передається на вузол клієнта, що неправильно, так як закритий ключ не повинен залишати межі свого вузла.

Теоретично ми повинні на кожному клієнті центру сертифікації створити відкритий та закритий ключі, сформувати запит на сертифікат та надіслати його

CA, який у свою чергу випустить потрібний нам сертифікат та підпише його своїм відкритим ключем. Таким чином, жодна критично важлива для безпеки інформація по каналах зв'язку не передається.

На практиці схема, коли і ключі, і сертифікати формує CA цілком допустима, тому що все це контролюється системним адміністратором, який може вжити розумних заходів для недопущення несанкціонованого доступу до закритих ключів [12].

Таким чином, на кожному з вузлів нашої OpenVPN-мережі має бути закритий ключ, сертифікат вузла та кореневий сертифікат CA. При цьому ні клієнти, ні сервери не знають, кому саме випущені сертифікати і в якій кількості.

OpenVPN-сервер не має списку клієнтів і готовий надати доступ будь-кому, хто надасть валідний сертифікат. Тому що всі вони входять до області довіри, створеної центром сертифікації в рамках інфраструктури відкритих ключів.

Щоб отримати доступ, клієнт пред'являє серверу власний сертифікат, сервер, за допомогою кореневого сертифіката CA переконується, що даний сертифікат дійсно виданий нашим центром сертифікації і йому можна довіряти, потім перевіряється термін його дії і при успішній перевірці починається процес встановлення захищеного з'єднання.

Термін дії є одним з найважливіших параметрів сертифіката і його можна успішно використовувати на свою користь. Скажімо, для співробітника на випробувальному терміні можна випустити сертифікат на 30 днів, а потім просто перевипустити на більший термін. Це набагато безпечніше, ніж потім забути відкликати сертифікат у працівника.

Зі свого боку клієнт також перевіряє дійсність сертифіката сервера, після чого встановлює з ним зв'язок. При цьому будь-який клієнт із сертифікатом поточного CA може без проблем підключитись до будь-якого сервера в рамках інфраструктури PKI.

Що стосується CA, то зазвичай він розміщується на одному з серверів OpenVPN, але в складніших схемах має сенс винести його на окремий вузол, найкраще на віртуальну машину, яку навіть можна більшу частину часу тримати

вимкненою, включаючи тільки для того, щоб випустити або відкликати черговий сертифікат.

І якщо ми говоримо про довіру, то у якому разі суб'єкт може її втратити. Варіантів тут може бути кілька: це звільнення співробітника, і можлива компрометація його закритого ключа (наприклад, пристрій було викрадено). У цьому випадку сертифікат потрібно відкликати. У процесі відкликання сертифіката CA формує список відкликаних сертифікатів – CLR. Цей список публікується на попередньо відомому вузлі, і будь-який учасник PKI може перевірити пред'явлений йому сертифікат на предмет відкликання. Але OpenVPN так не вміє, що спричиняє деякі додаткові дії (Рисунок 2.8).

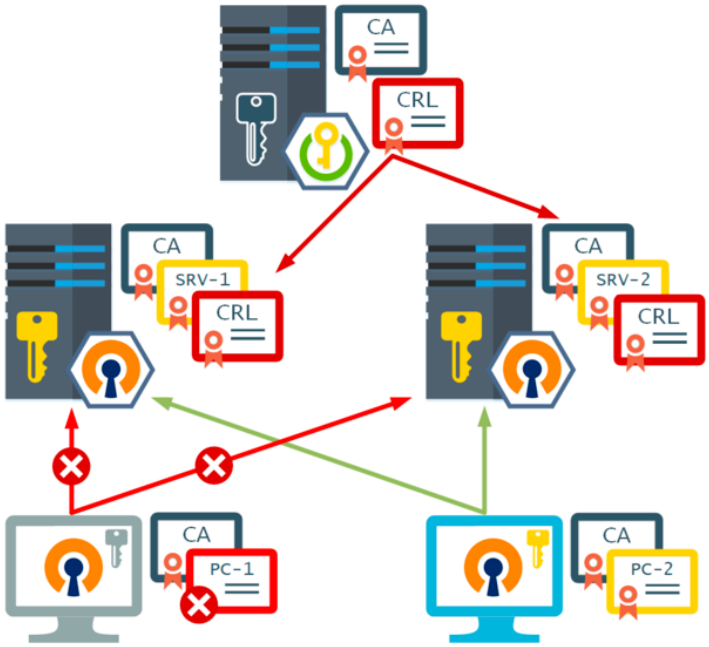

Рис. 2.8 Схема роботи CRL

Після того, як CA оновив список відкликаних сертифікатів (CRL), його потрібно поширити на всі сервери OpenVPN в нашій інфраструктурі, після чого вони будуть розуміти, що цей сертифікат відкликаний і відмовляти у з'єднанні. Якщо ми не виконаємо цю умову, то сервери, які не отримали оновлений CRL, як і раніше, дозволятимуть доступ клієнту, сертифікат якого відкликаний.

Дослідивши всі ці данні можні дійти висновку, що правильний сценарій передбачає у межах однієї організації мати один центр сертифікації.

3 РОЗРОБЛЕННЯ ВАРІАНТА ТЕХНОЛОГІЇ ЗАБЕЗПЕЧЕННЯ ЗАХИЩЕНОГО ОБМІНУ ДАНИМИ В МОБІЛЬНІЙ МЕРЕЖІ ПІДПРИЄМСТВА З ВИКОРИСТАННЯМ OPENVPN COMMUNITY EDITION

3.1. Аналіз та проектування ВПН мережі підприємства

Технологія VPN має багато переваг, однією з головних є – можливість побудови захищеної мережі використовуючи при цьому стандартні протоколи обміну інформації для побудови тунелю, обміну зашифрованою інформацією та забезпечення зв’язку для передавання зашифрованих даних.

В першу чергу для побудови мережі такого типу, потрібно визначити її архітектуру, тобто кількість підмереж які будуть створені, кількість та тип проміжного обладнання (маршрутизатори, комутатори, тип транспортних магістралей). Так як більшість працівників IT підприємст нашої країни під час карантину працюють віддалено, це питання дуже актуальне.

На практичному прикладі буде створена корпоративна мережі для підприємства ТОВ «ЕВОС ЮА». Корпоративна мережа якого поділена на дві підмережі:

- 192.168.88.0/24
- 192.168.1.0/24

За адресою 192.168.88.11 знаходиться сервер телефонії, доступ до якого надано підмережі 192.168.1.0/24 шляхом маршрутизації «iptables». Корпоративна мережа знаходиться в мережі 192.168.1.0/24, а також внутрішні веб сервіси та корпоративні додатки.

Дослідивши всю інфраструктуру підприємства, було прийнято рішення побудувати OpenVPN сервер на базі корпоративної мережі, з можливістю підміни

клієнтськи IP адрес, використовуючи тунелювання та шифрування, для доступу до сереверу телефонії.

ВПН сервер та сервер ЦС побудуємо на базі Proxmox Virtual Environment, створивши дві віртуальні машини з адресами:

- 192.168.0.146 (ЦС)
- 192.168.0.147 (ВПН сервер)

Proxmox VE - комплексне програмне забезпеченням для управління серверною віртуалізацією. Він заснований на віртуалізації KVM і віртуалізації на базі контейнерів та управляє віртуальними машинами KVM, контейнерами Linux (LXC), зберіганням, віртуальними мережами та HA-кластерами (Рисунок 3.1).

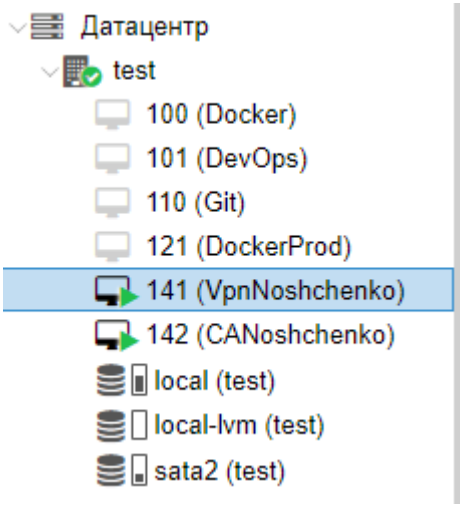

Рис. 3.1 – ВПН та ЦС сервери на базі Proxmox

За основу було вибрано безкоштовну Unix-подібну операційну систему.

Особливості встановлення OpenVPN на Unix-подібні системи на відміну від простої установки на версії Windows полягає в послідовній конфігурації файлів. Крім того, при встановленні додатку на системи Windows, OpenVPN дозволить використати графічний інтерфейс встановлюючої програми.

Не має різниці, на яку саме Unix-подібну систему ви бажаєте встановити OpenVPN – етапи встановлення мало чим відрізняються один від одного. У нашому випадку це буде Ubuntu 20.04.

3.2. Встановлення на налаштування центру сертифікації

Сервер ЦС повинен бути окремою системою. Вона буде використовуватися лише для імпорту, підписання та відкликання запитів сертифікатів. На ній не повинні працювати жодні інші служби, і в ідеальному варіанті її потрібно відключати від мережі або повністю відключати в період, коли ви активно не працюєте з ЦС [7].

Спершу потрібно встановити набір скриптів easy-rsa для запуску на сервері ЦС. Будемо використовувати інструмент управління ЦС easy-rsa для генерування приватного ключа та публічного кореневого сертифіката, за допомогою яких підпишемо запити клієнтів і серверів, що використовують ЦС.

Увійдемо на сервер ЦС від імені користувача без прав root з привілеями sudo, створеного на початковому етапі установки, та запустимо наступну команду:

- sudo apt update
- sudo apt install easy-rsa

Нам буде запропоновано завантажити пакет та встановити його.

Ми встановили easy-rsa і тепер будемо займатися створенням каркасу інфраструктури відкритих ключів (PKI) на сервері ЦС. Переконаємось, що ми увійшли в систему як користувач без прав root, і створимо директорію easy-rsa. Не використовуйте sudo для запуску наступних команд, оскільки звичайні користувачі повинні мати можливість керувати та взаємодіяти з ЦС без підвищеного рівня прав.

Створимо нову директорію easy-rsa у домашній папці.

- mkdir ~/easy-rsa

Ми використовуємо цю директорію для створення символічних посилань, що вказують на файли easy-rsa, встановлені на попередньому кроці. Ці файли знаходяться у папці /usr/share/easy-rsa на сервері ЦС.

Щоб обмежити доступ до створеної директорії PKI, використаємо команду chmod для надання доступу до неї лише власнику:

- chmod 700 /home/sammy/easy-rsa

Потім ініціалізуємо PKI у директорії easy-rsa:

- cd ~/easy-rsa
- ./easyrsa init-pki

Після цих дій у нас буде директорія, що містить усі необхідні файли для створення Центру сертифікації.

Створення Центру сертифікації.

Перш ніж створювати закритий ключ та сертифікат ЦС, необхідно створити файл з ім'ям vars та заповнити його значеннями за замовчуванням. Перейдемо в директорію easy-rsa та створимо і відредагуємо файл vars за допомогою nano або іншого текстового редактора:

- cd ~/easy-rsa
- nano vars

Відкривши файл, заповнимо його за шаблоном який нам пропонує EasyRsa (Рисунок 3.2).

```
set_var EASYRSA_REQ_COUNTRY    "UA"
set_var EASYRSA_REQ_PROVINCE   "Kyiv"
set_var EASYRSA_REQ_CITY       "Kyiv city"
set_var EASYRSA_REQ_ORG        "SUT"
set_var EASYRSA_REQ_EMAIL      "stas.noshchenko@gmail.com"
set_var EASYRSA_REQ_OU         "Community"
set_var EASYRSA_ALGO           "ec"
set_var EASYRSA_DIGEST         "sha512"
```

Рис. 3.2 - Файл конфігурації easy-rsa/vars

Для створення кореневої пари відкритого та закритого ключів для Центру сертифікації необхідно запустити команду ./easy-rsa ще раз, але вже з опцією build-ca:

- ./easyrsa build-ca

Ви побачите кілька рядків із зазначенням версії OpenSSL, а потім буде запропоновано ввести фразу-пароль для пари ключів. Використовуємо надійну фразу-пароль адже потрібно буде вводити фразу-пароль щоразу, коли нам потрібно взаємодіяти з ЦС, зокрема під час підписання або відкликання сертифіката.

Також вам буде запропоновано підтвердити просте ім'я (CN) вашого ЦС. Це ім'я, яке використовуватиметься для цього комп'ютера в контексті Центру сертифікації. Ми можемо використовувати будь-який рядок символів як звичайне ім'я ЦС, але для простоти натиснемо ENTER, щоб використовувати стандартне ім'я.

```
stas@ca:~/easy-rsa$ ./easyrsa build-ca

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

Using SSL: openssl OpenSSL 1.1.1f  31 Mar 2020

Enter New CA Key Passphrase:
Re-Enter New CA Key Passphrase:
read EC key
writing EC key
Can't load /home/stas/easy-rsa/pki/.rnd into RNG
139731598972224:error:2406F079:random number generator:RAND_load_file:Cannot open file:../crypto/rand/randfile.c:98
e=/home/stas/easy-rsa/pki/.rnd
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/stas/easy-rsa/pki/ca.crt
```

Рис. 3.3 - Створення ca.crt та ca.key

Тепер у нас є два важливі файли, ~/easy-rsa/pki/ca.crt та ~/easy-rsa/pki/private/ca.key.

Користувачі, сервери та клієнти будуть використовувати сертифікат ca.crt для підтвердження єдиної мережі довіри. Копія цього файлу повинна бути у всіх користувачів та серверів, які будуть використовувати ЦС. Усі сторони будуть використовувати публічний сертифікат для підтвердження справжності системи та запобігання атакам через посередника.

Закритий ключ ca.key буде використовуватися для підписання сертифікатів серверів і клієнтів. Якщо зловмисник отримає доступ до ЦС та файлу ca.key, вам доведеться знищити ЦС. Тому файл ca.key повинен зберігатися лише на комп'ютері ЦС, і для додаткової безпеки, комп'ютер ЦС слід вимикати, коли він не використовується для підписання запитів сертифікатів.

Ми встановили ЦС та готові використовувати його для підписання запитів сертифікатів та відкликання сертифікатів.

3.3. Встановлення та налаштування OpenVPN серверу

Для початку перейдемо на сервер який ми підготували для VPN та встановимо OpenVPN та додатково Easy-RSA, як і на сервері ЦС. Обидва пакети доступні за замовчуванням у репозиторіях Ubuntu, і тому ми можемо використовувати apt для встановлення:

- sudo apt update
- sudo apt install openvpn easy-rsa

Потрібно створити нову директорію на сервері OpenVPN від імені вашого користувача non-root user з назвою ~/easy-rsa:

- mkdir ~/easy-rsa

Переконаємось, що власником директорії є наш користувач non-root user з привілеями sudo, і обмежимо доступ за допомогою команди chmod:

- sudo chown stas ~/easy-rsa
- chmod 700 ~/easy-rsa

Після встановлення цих програм та їх переміщення у потрібні деректорії у нашій системі, наступним кроком буде створення інфраструктури відкритих ключів (PKI) на сервері OpenVPN, щоб ми могли запитувати та керувати сертифікатами TLS для клієнтів та інших серверів, які підключатимуться до нашого VPN.

Створення PKI для OpenVPN.

Перш ніж ми зможемо створювати закритий ключ і сертифікат сервера OpenVPN, нам потрібно створити локальну директорію інфраструктури відкритих ключів на сервері OpenVPN. Ми будете використовувати цю директорію для керування запитами сертифіката сервера та клієнтів замість того, щоб отримувати їх прямо на сервері ЦС [16].

Для створення директорії PKI на сервері OpenVPN потрібно вказати у файлі vars ряд значень за замовчуванням. Використаємо команду cd для переходу в

каталог easy-rsa, а потім створимо та відредагуємо файл vars за допомогою nano або іншого текстового редактора:

- cd ~/easy-rsa
- nano vars

Після відкриття файлу вставимо наступні два рядки:

- set_var EASYRSA_ALGO "ec"
- set_var EASYRSA_DIGEST "sha512"

Це потрібно для файлу vars на сервері OpenVPN, оскільки він не буде використовуватися як ЦС. Вони будуть гарантувати, що закриті ключі та запити сертифіката будуть налаштовані для використання сучасної криптографії на еліптичних кривих (Elliptic Curve Cryptography, ECC) при генерації ключів та захищених підписів для клієнтів та сервера OpenVPN.

Під використанням ECC для серверів OpenVPN і ЦС мається на увазі, що коли клієнт і сервер намагатимуться встановити загальний симетричний ключ, вони будуть використовувати алгоритми еліптичної кривої для обміну. Використання ECC для обміну ключами значно швидше, ніж використання простого алгоритму Діффі-Хеллмана з класичним алгоритмом RSA, оскільки числа набагато менші, а обчислення виконуються швидше.

Після додавання даних в файл vars, перейдемо до створення директорії PKI. Для цього скористуємось скриптом easyrsa з опцією init-pki. Ми вже використовували цю команду на сервері ЦС відповідно до попередніх вимог, необхідно запустити її тут, оскільки сервер OpenVPN та сервер ЦС мають різні директорії PKI.

- ./easyrsa init-pki

Після ініціалізації PKI на сервері OpenVPN можна перейти до наступного кроку та створити запит сертифіката та закритого ключа сервера OpenVPN.

Тепер згенеруємо закритий ключ і запит підпису сертифіката на сервері OpenVPN. Після цього передамо запит до центру сертифікації для підпису, створивши необхідний сертифікат. Після підпису сертифіката передамо його назад на сервер OpenVPN і встановимо його для використання на сервері.

Перейдимо до директорії ~/easy-rsa на сервері OpenVPN, використовуючи користувача non-root user:

- cd ~/easy-rsa

Викличемо easyrsa з опцією gen-req, за якою необхідно вказати стандартне ім'я для комп'ютера. Можемо використовувати будь-яке стандартне ім'я. Ми будемо використовуватися стандартне ім'я server. Додамо опцію nopass. Без цього файл запиту буде захищений паролем, що може призвести до проблем з дозволами.

- ./easyrsa gen-req server nopass

```
stas@vpn:~/easy-rsa$ A or requests.
A: command not found
stas@vpn:~/easy-rsa$ ./easyrsa gen-req server nopass

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

Using SSL: openssl OpenSSL 1.1.1f  31 Mar 2020
Can't load /home/stas/easy-rsa/pki/.rnd into RNG
140158871500096:error:2406F079:random number generator:RAND_load_file:Cannot open file:
e=/home/stas/easy-rsa/pki/.rnd
Generating an EC private key
writing new private key to '/home/stas/easy-rsa/pki/private/server.key.KMCe9LEjuU'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [server]:

Keypair and certificate request completed. Your files are:
req: /home/stas/easy-rsa/pki/reqs/server.req
key: /home/stas/easy-rsa/pki/private/server.key
```

Рис. 3.4 Створення server.key та server.req

В результаті буде створено закритий ключ для сервера та файл запиту сертифіката з ім'ям server.req. Скопіюємо ключ сервера в каталог /etc/openvpn/server:

- sudo cp /home/sammy/easy-rsa/pki/private/server.key /etc/openvpn/server/

В результаті виконання цих кроків було успішно створено закритий ключ для сервера OpenVPN. Також було створено запит на підпис сертифіката для сервера OpenVPN.

Тепер сервер ЦС повинен дізнатися про сертифікат server та виконати його валідацію. Після підтвердження сертифіката сервером ЦС та його надсилання назад на сервер OpenVPN клієнти, які довіряють ЦС, також зможуть довіряти серверу OpenVPN.

На сервері OpenVPN від імені користувача non-root user використаємо запит на передачу для копіювання запиту сертифіката server.req на сервер ЦС для підпису:

- scp /home/stas/easy-rsa/pki/reqs/server.req stas@192.168.0.146:/tmp

Наступним кроком повернемось на сервер ЦС. Пейдемо до каталогу ~/easy-rsa, та імпортуємо запит сертифіката за допомогою скрипта easyrsa:

- cd ~/easy-rsa
- ./easyrsa import-req /tmp/server.req server

```
stas@ca:~/easy-rsa$ ./easyrsa import-req /tmp/server.req server

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

Using SSL: openssl OpenSSL 1.1.1f  31 Mar 2020

The request has been successfully imported with a short name of: server
You may now use this name to perform signing operations on this request.
```

Рис. 3.5 - Імпортування запиту сертифіката серверу

Підпишемо запит, запустивши скрипт easyrsa з опцією sign-req та вказівкою типу запиту та стандартного імені. Як тип запиту може використовуватися client або server. Оскільки ми працюємо із запитом сертифіката сервера OpenVPN, необхідно використовувати тип запиту server:

- ./easyrsa sign-req server server

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

Request subject, to be signed as a server certificate for 1080 days:

subject=
    commonName                = server


Type the word 'yes' to continue, or any other input to abort.
  Confirm request details: yes
Using configuration from /home/stas/easy-rsa/pki/safessl-easyrsa.cnf
Enter pass phrase for /home/stas/easy-rsa/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'server'
Certificate is to be certified until Nov 30 00:47:30 2024 GMT (1080 days)

Write out database with 1 new entries
Data Base Updated

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

Рис. 3.6 - Підписання запиту сертифіката

Після виконання цих кроків успішно підписали запит сертифіката сервера OpenVPN за допомогою закритого ключа сервера ЦС [17]. Отриманий файл server.crt містить відкритий ключ шифрування сервера OpenVPN, а також новий підпис від сервера ЦС. Сенс підпису полягає в тому, щоб повідомити всіх, хто довіряє серверу ЦС, що вони можуть довіряти серверу OpenVPN при підключенні до нього.

Для завершення налаштування сертифікатів скопіюємо файли server.crt та ca.crt із сервера ЦС на сервер OpenVPN:

- scp pki/issued/server.crt sammy@your_vpn_server_ip:/tmp
- scp pki/ca.crt sammy@your_vpn_server_ip:/tmp

Повернемось на сервер OpenVPN, скопіюємо файли з /tmp в /etc/openvpn/server:

- sudo cp /tmp/{server.crt,ca.crt} /etc/openvpn/server

Як додатковий рівень безпеки ми додамо додатковий загальний секретний ключ, який використовуватиме сервер і всі клієнти, за допомогою директиви OpenVPN tls-crypt. Ця опція використовується, щоб «затемнити» сертифікат TLS, який використовується, коли сервер та клієнт спочатку підключаються один до

одного. Також вона використовується сервером OpenVPN для виконання швидких перевірок вхідних пакетів: якщо пакет підписаний за допомогою попередньо наданого ключа, сервер обробляє його, якщо підпис відсутній, сервер розуміє, що пакет отриманий з неперевіреного джерела і може відхилити його, не виконуючи додаткову роботу з розшифровки .

Ця опція допоможе переконатися, що ваш сервер OpenVPN може справлятися з незавіреним трафіком, скануванням портів та DoS-атаками, які можуть пов'язувати ресурси сервера. Вона також ускладнює виявлення мережевого трафіку OpenVPN.

Для отримання попередньо наданого ключа tls-crypt, запустимо наступну команду openvpn --genkey --secret ta.key на сервері OpenVPN у директорії ~/easy-rsa.

В результаті отримаємо файл ta.key. Скопіюємо його в директорію /etc/openvpn/server/:

- sudo cp ta.key /etc/openvpn/server

Після отримання цих файлів на сервері OpenVPN можна переходити до створення клієнтських сертифікатів та файлів ключів для користувачів, які будуть використовувати для підключення до VPN.

Створимо одну пару з ключа і сертифіката для клієнтської системи. Для кожного клієнта в скрипті потрібно вказати унікальне ім'я.

Створимо у домашній директорії структуру директорій, де зберігатимуться файли сертифікатів та ключів клієнтської системи:

- mkdir -p ~/client-configs/keys

Оскільки в цій директорії зберігатимуться пари сертифікат-ключ клієнтів та файли конфігурації, для неї слід закрити всі дозволи:

- chmod -R 700 ~/client-configs

Запустимо скрипт easyrsa з опціями gen-req та nopass, вказавши просте ім'я клієнта:

- cd ~/easy-rsa
- ./easyrsa gen-req client1 nopass

```
Using SSL: openssl OpenSSL 1.1.1f  31 Mar 2020
Generating an EC private key
writing new private key to '/home/stas/easy-rsa/pki/private/client1.key.MPvXKhnENZ'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [client1]:

Keypair and certificate request completed. Your files are:
req: /home/stas/easy-rsa/pki/reqs/client1.req
key: /home/stas/easy-rsa/pki/private/client1.key
```

Рис. 3.7 - Запит на підпис клієнтського сертифікату

Скопіюємо файл client1.key у раніше створену директорію ~/client-configs/keys/ :

- cp pki/private/client1.key ~/client-configs/keys/

Передамо файл client1.req на сервер ЦС, використовуючи безпечний метод:

- scp pki/reqs/client1.req stas@192.168.0.146:/tmp

Виконаємо вхід на сервер ЦС та перейдемо до директорії EasyRSA імпортувавши запит сертифіката:

- ./easyrsa import-req /tmp/client1.req client1

Підпишемо запит, вказавши тип запиту client:

- ./easyrsa sign-req client client1

При запиті підтверджуємо шо ми плануємо підписати запит сертифіката та що він надійшов із довіреного джерела:

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

Request subject, to be signed as a client certificate for 1080 days:

subject=
    commonName                = client1


Type the word 'yes' to continue, or any other input to abort.
  Confirm request details: yes
Using configuration from /home/stas/easy-rsa/pki/safessl-easyrsa.cnf
Enter pass phrase for /home/stas/easy-rsa/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'client1'
Certificate is to be certified until Nov 30 01:49:25 2024 GMT (1080 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /home/stas/easy-rsa/pki/issued/client1.crt
```

Рис. 3.8 Створення сертифікату клієнта

В результаті буде створено файл сертифіката клієнта з ім'ям client1.crt.

Перенесемо файл на сервер:

- scp pki/issued/client1.crt stas@192.168.0.147:/tmp

Перейдемо на сервер OpenVPN скопіювавши клієнтський сертифікат у директорію ~/client-configs/keys/ :

- cp /tmp/client1.crt ~/client-configs/keys/

Скопіюємо файли ca.crt та ta.key у директорію ~/client-configs/keys/ і надамо відповідні дозволи для користувача sudo:

- cp ~/easy-rsa/ta.key ~/client-configs/keys/
- sudo cp /etc/openvpn/server/ca.crt ~/client-configs/keys/
- sudo chown stas.stas ~/client-configs/keys/*

В результаті згенерувалися ключі та сертифікати для сервера та клієнта та збереглися у відповідних директоріях на сервері OpenVPN.

Як і в багатьох широко використовуваних інструментах з відкритим вихідним кодом, OpenVPN має безліч доступних параметрів для кастомізації вашого сервера відповідно до ваших потреб. Ми надамо інструкції з налаштування конфігурації сервера OpenVPN на основі одного з прикладів файлів конфігурації, який включено до документації для цього програмного забезпечення.

Спочатку скопіюємо файл server.conf як відправну точку для нашого власного файлу конфігурації:

- sudo cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz /etc/openvpn/server/
- sudo gunzip /etc/openvpn/server/server.conf.gz

Відкриємо новий файл для редагування в текстовому редакторі на вибір. У нашому випадку ми будемо використовувати vim:

- sudo vim /etc/openvpn/server/server.conf

Нам потрібно змінити кілька рядків у цьому файлі. Спочатку необхідно знайти розділ HMAC у конфігурації, виконавши пошук директиви tls-auth. Цей рядок має бути розкоментований. Закоментуємо ого та додамо після нього новий рядок, який містить тільки значення tls-crypt ta.key :

- ;tls-auth ta.key 0 # Цей файл є secret
- tls-crypt ta.key

Знайдемо розділ криптографічних шифрів, виконавши пошук рядків із текстом cipher. За замовчуванням встановлено значення AES-256-CBC, проте шифр AES-256-GCM забезпечує вищий рівень шифрування, продуктивності та краще підтримується сучасними клієнтами OpenVPN. Закоментуємо значення за замовчуванням, додавши «;» на початок цього рядка, а потім додамо інший рядок після неї, що містить оновлене значення AES-256-GCM:

- ;cipher AES-256-CBC
- cipher AES-256-GCM

Відразу після цього рядка додамо директиву auth для вибору алгоритму вибірки повідомлень HMAC. Для цього добре підійде SHA256:

- /etc/openvpn/server/server.conf
- auth SHA256

Знайдемо рядок із директивою dh, який визначає параметри алгоритму Діффі - Хеллмана. Оскільки ми налаштували всі сертифікати для використання криптографії на еліптичних кривих, немає необхідності використовувати файл прототипу Діффі - Хеллмана. Закоментуємо існуючий рядок dh dh2048.pem або dh dh.pem. Ім'я файлу для ключа Діффі-Хеллмана може відрізнятися від того, що наведено в прикладі файлу конфігурації сервера. Додамо рядок після неї із вмістом dh none:

- /etc/openvpn/server/server.conf
- ;dh dh2048.pem
- dh none

Далі нам потрібно запустити OpenVPN без привілеїв, з якими він запущений, тому нам потрібно зазначити необхідність запуску з користувачем nobody та групою nogroup. Щоб активувати цю можливість, розкоментуємо рядки user nobody та group nogroup, вилучивши «;» на початку кожного рядка:

- /etc/openvpn/server/server.conf
- user nobody

- group nogroup

Щоб OpenVPN міг правильно перенаправляти трафік через мережу VPN, необхідно змінити деякі параметри конфігурації мережі сервера. Насамперед потрібно змінити параметр IP forwarding, який визначає необхідність перенаправлення IP-трафіку. Це необхідно для реалізації функцій VPN, які надаються сервером.

Щоб змінити параметри IP-передачі сервера OpenVPN, що використовуються за замовчуванням, відкриємо файл /etc/sysctl.conf за допомогою nano або бажаного редактора:

- sudo nano /etc/sysctl.conf

Додамо наступний рядок до кінця файлу:

- /etc/sysctl.conf
- net.ipv4.ip_forward = 1

Щоб прочитати файл та завантажити значення для поточної сесії, введемо:

- sudo sysctl -p

Тепер сервер OpenVPN зможе перенаправляти вхідний трафік з одного мережного пристрою на інший. Це налаштування гарантує, що сервер може спрямовувати трафік від клієнтів, підключених до віртуального інтерфейсу VPN, на інші фізичні мережеві пристрої. Ця конфігурація буде передавати весь веб-трафік від клієнта через IP-адресу сервера, а відкрита IP-адреса клієнта буде фактично прихована.

OpenVPN працює як служба systemd, тому можемо використовувати systemctl для керування. Налаштуємо для OpenVPN запуск під час завантаження, щоб можна було підключатися до VPN у будь-який час, поки сервер працює. Для цього активуємо службу OpenVPN, додавши її до systemctl:

- sudo systemctl -f enable openvpn-server@server.service

Запустимо службу OpenVPN:

- sudo systemctl start openvpn-server@server.service

Перевіримо, що OpenVPN активнтй, скориставшись наступною командою.

- sudo systemctl status openvpn-server@server.service

```
stas@vpn:~$ sudo systemctl start openvpn-server@server.service
stas@vpn:~$ sudo systemctl status openvpn-server@server.service
● openvpn-server@server.service - OpenVPN service for server
     Loaded: loaded (/lib/systemd/system/openvpn-server@.service; disabled; vendor preset: enabled)
     Active: active (running) since Thu 2021-12-16 02:23:16 UTC; 3s ago
       Docs: man:openvpn(8)
             https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
             https://community.openvpn.net/openvpn/wiki/HOWTO
   Main PID: 3969 (openvpn)
     Status: "Initialization Sequence Completed"
      Tasks: 1 (limit: 2278)
     Memory: 1.3M
     CGroup: /system.slice/system-openvpn\x2dserver.slice/openvpn-server@server.service
             └─3969 /usr/sbin/openvpn --status /run/openvpn-server/status-server.log --status-version 2 --suppress-timesta

Dec 16 02:23:16 vpn openvpn[3969]: Could not determine IPv4/IPv6 protocol. Using AF_INET
Dec 16 02:23:16 vpn openvpn[3969]: Socket Buffers: R=[212992->212992] S=[212992->212992]
Dec 16 02:23:16 vpn openvpn[3969]: UDPv4 link local (bound): [AF_INET][undef]:443
Dec 16 02:23:16 vpn openvpn[3969]: UDPv4 link remote: [AF_UNSPEC]
Dec 16 02:23:16 vpn openvpn[3969]: GID set to nogroup
Dec 16 02:23:16 vpn openvpn[3969]: UID set to nobody
Dec 16 02:23:16 vpn openvpn[3969]: MULTI: multi_init called, r=256 v=256
Dec 16 02:23:16 vpn openvpn[3969]: IFCONFIG POOL: base=10.8.0.4 size=62, ipv6=0
Dec 16 02:23:16 vpn openvpn[3969]: IFCONFIG POOL LIST
Dec 16 02:23:16 vpn openvpn[3969]: Initialization Sequence Completed
```

Рис. 3.9 - Стан роботи ВПН сервера

Створення інфраструктури конфігурації клієнтських систем.

Замість створення єдиного файлу конфігурації, який можна використовувати лише для одного клієнта, визначимо процес створення інфраструктури конфігурації клієнта, який ми зможемо використовувати для швидкого генерування файлів конфігурації. Спочатку створимо базовий файл конфігурації, а потім сценарій, який дозволить при необхідності генерувати унікальні файли конфігурації клієнтів, сертифікати та ключі.

Для початку створимо нову директорію для зберігання файлів конфігурації клієнтів у раніше створеній директорії client-configs:

- mkdir -p ~/client-configs/files

Скопіюємо файл із зразком конфігурації клієнта в директорію client-configs, щоб використовувати її як базову конфігурацію:

- cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf ~/client-configs/base.conf
- nano ~/client-configs/base.conf

Знайдемо у файлі директиву remote. Вона вказує клієнту адресу сервера OpenVPN, тобто публічну IP-адресу сервера OpenVPN.

Name_name/IP and port of the server.

You can have multiple remote entries

to load balance between the servers.

remote your_server_ip 443

Протокол повинен відповідати значенням, які використовуються в конфігурації сервера:

- proto udp

Розкоментуємо директиви user та group, видаливши символ «;» на початку кожного рядка:

Downgrade privileges after initalization (non-Windows only)

user nobody

group nogroup

Знайдемо директиви, які задають ca, cert та key. Поставимо знак коментаря перед рядками цих директив, оскільки додамо сертифікати та ключі до файлу:

SSL/TLS parms.

See the server config file for more

description. It's best to use

a separate .crt/.key file pair

for each client. A single ca

файл може бути використаний для всіх клієнтів.

;ca ca.crt

;cert client.crt

;key client.key

Закоментуємо директиву tls-auth, оскільки додамо ta.key прямо до файлу конфігурації клієнта:

If a tls-auth key is used on the server

then every client must also have the key.

;tls-auth ta.key 1

Створимо дзеркало налаштувань cipher та auth, заданих у файлі /etc/openvpn/server/server.conf:

cipher AES-256-GCM

auth SHA256

Додамо у файл директиву key-direction.

key-direction 1

Збережемо файл та закриємо його після завершення.

Створимо простий скрипт, який скомпілює базову конфігурацію з відповідним сертифікатом, ключем та файлами шифрування, та помістимо згенеровану конфігурацію в директорію ~/client-configs/files. Відкриємо новий файл з ім'ям make_config.sh у директорії ~/client-configs:

- nano ~/client-configs/make_config.sh

Додамо до файлу наступне:

```
#!/bin/bash
# First argument: Client identifier

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
```

Цей скрипт створює копію створеного файлу base.conf, збирає всі створені для клієнта файли сертифікатів та ключів, витягує їх вміст, додає їх у копію базового файлу конфігурації та експортує все це до нового файлу конфігурації клієнта. Це означає, що вся необхідна інформація зберігається в одному місці, і не потрібно

керувати файлами конфігурації клієнта, сертифікатами і ключами. Перевага такого методу полягає в тому, що якщо в майбутньому потрібно додати клієнт, можна просто запустити цей скрипт, щоб швидко створити файл конфігурації, а вся важлива інформація зберігатиметься в одному зручному для доступу місці.

Створимо файл конфігурації для облікових даних клієнта, перейшовши в директорію ~/client-configs і запустивши скрипт make_config.sh.

cd ~/client-configs

./make_config.sh client1

Ls -la

При цьому файл client1.ovpn буде створений у директорії ~/client-configs/files.

Цей файл потрібно буде перемістити на пристрій, який планує використовувати клієнт.

3.4. Аналіз клієнтської частини OpenVPN

Після того, як був побудований ЦС, OpenVPN сервер та створено сертифікати перейдімо до клієнтської частини, а саме можливості віддалено підключитися до корпоративної мережі з будь-якої публічної мережі.

Підключення OpenVPN матиме ім'я, яке співпадає з ім'ям файлу .ovpn. У нашому випадку це означає, що з'єднання матиме ім'я client1.ovpn, що відповідає першому згенерованому клієнтському файлу.

Windows.

Необхідно завантажити клієнтську програму OpenVPN для Windows зі сторінки завантаження OpenVPN, вибравши відповідну версію програми інсталяції для вашої версії Windows.

Після встановлення OpenVPN скопіюємо файл .ovpn до:

C:\Program Files\OpenVPN\config

При запуску OpenVPN профіль буде автоматично виявлено та зроблено основним.

Запускати OpenVPN потрібно від імені адміністратора щоразу, навіть якщо використовується обліковий запис адміністратора. Після запуску OpenVPN необхідно натиснути правою кнопкою на піктограму OpenVPN в області завдань, щоб створити з'єднання. Вибемо client1 у верхній частині меню (це профіль client1.ovpn) і натиснемо Підключитись (Рисунок 3.10).

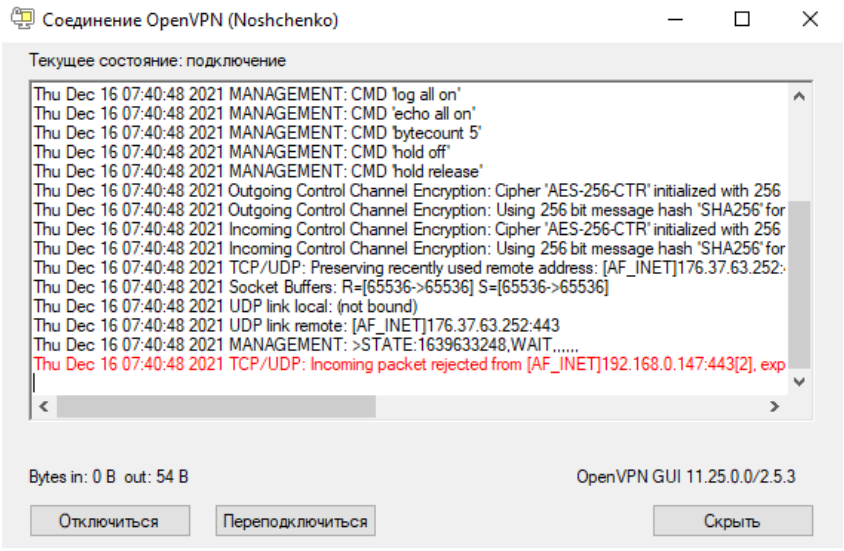

Рис. 3.10 - Графічний інтерфейс Windows OpenVPN клієнта

Операційна система iOS.

Знайти програму OpenVPN Connect, офіційний клієнт OpenVPN для iOS, можна в магазині iTunes App Store. Щоб перемістити конфігурацію клієнта iOS на пристрій, потрібне підключення до комп'ютера.

Запустимо програму OpenVPN на iPhone. Отримаємо повідомлення про те, що новий профіль готовий до імпорту. Натиснемо зелений значок плюс, щоб імпортувати його.

Програма OpenVPN готова до використання нового профілю (Рисунок 3.11).

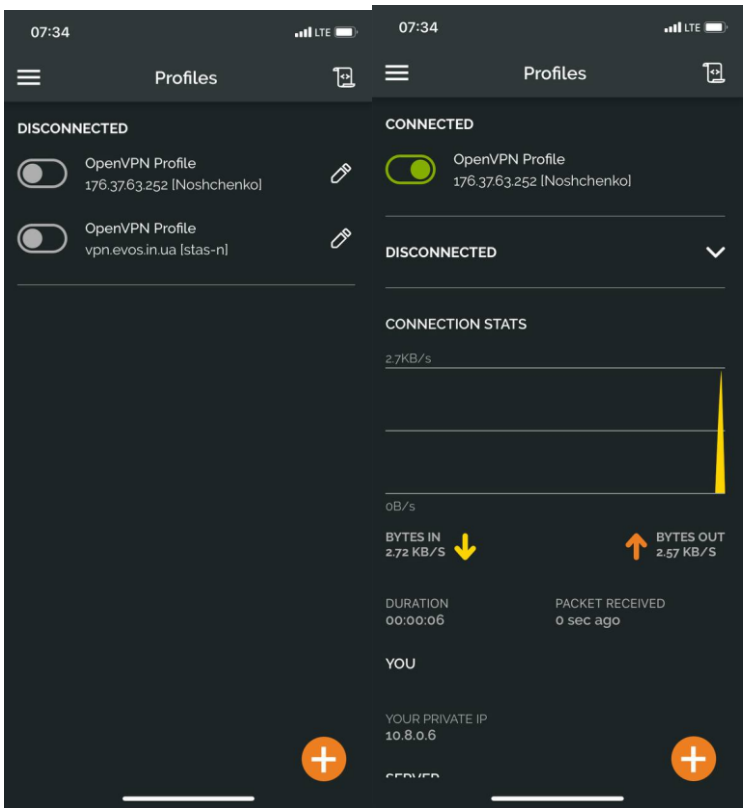

Рис. 3.11 - Графічний інтерфейс OpenVPN Connect на iOS

Перевірити статут підключення можна за допомогою сервісу 2ip.ua (Рисунок 3.12).

Рис. 3.12 - Зовнішня IP адреса до та після підключення до мережі VPN

Також клієнтам мобільної версії OpenVPN доступний зручний перегляд логіювання з самого додатку (Рисунок 3.13).

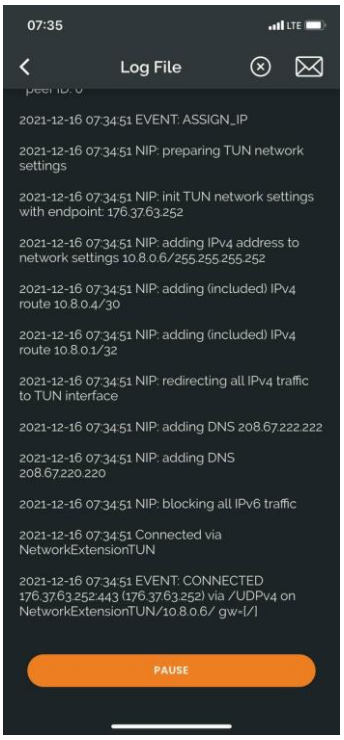

Рис. 3.13 - Розділ логіювання в OpenVPN Connect

Операційна система Android.

Аналогічна ситуація як і у користувачів Android пристроїв. Знайти програму OpenVPN Connect можна в Google Play Store.

Ми можемо перемістити профіль .ovpn, підключивши пристрій Android до комп'ютера через інтерфейс USB та скопіювавши файл. Запустимо програму OpenVPN і натиснемо меню FILE, щоб імпортувати профіль. Вибір меню імпорту профілю в OpenVPN для Android.

Операційна система macOS.

Tunnelblick — безкоштовний клієнт OpenVPN з відкритим кодом для macOS. Ми можемо завантажити останній образ цієї клієнтської програми зі сторінки завантаження Tunnelblick.

Наприкінці процесу установки Tunnelblick запитає, чи є у нас файли конфігурації. Відкриємо вікно Finder та двічі натиснемо client1.ovpn. Tunnelblick встановить профіль клієнта. Для цього потрібні привілеї адміністратора.

ВИСНОВКИ

У дипломній роботі було проведено огляд та аналіз існуючих засобів для забезпечення інформаційної безпеки локальних мереж. Проаналізовано порівняльні характеристики протоколів передачі даних та організації зв'язку між віддаленими частинами мереж з використанням технології VPN. Описано основні ризики та загрози з якими можуть стикатися користувачі в мережі Інтернет, запобігання цьому за для забезпечення інформаційної безпеки. Розглянуто сучасні проблеми сьогодення в сфері інформаційної безпеки. Обрано актуальні засоби та способи захисту інформації для забезпечення цілісності та конфіденційності даних при обміні.

На даний момент одним з варіантів для побудови мобільної мережі є OpenVPN, даючи можливість використовувати порт 443. До того ж, це один з небагатьох протоколів, підтримуваних відразу на декількох платформах.

Переваги OpenVPN:

- відносно новий протокол з відкритим вихідним кодом;
- можливість використовувати будь-які порти, що дозволяє обходити блокування більшості мобільних операторів;
- підтримує найрізноманітніші алгоритми, що дозволяє надійно захистити користувача;
- один з найшвидших протоколів.

На практичному прикладі з використанням програмного продукту Easy-rsa було створено центр сертифікації. Описано та проведено налаштування OpenVPN серверу на базі операційної системи Ubuntu 20.04 LTS.

Досліджено процес створення конфігурації для клієнта. Створено bash скрипт для автоматизації створення файлів конфігурації клієнтської частини OpenVPN. Було проаналізовано можливість використання ВПН в мобільних мережах та розроблено рекомендації щодо встановлення клієнтської частини на різні операційні системи.

ПЕРЕЛІК ПОСИЛАНЬ

1. McAfee. COVID-19 – Malware Makes Hay During a Pandemic [Електронний ресурс] – Режим доступу: https://www.mcafee.com/blogs/other-blogs/mcafee-labs/covid-19-malware-makes-hay-during-a-pandemic/.

2. Experimental performance comparison between TCP vs UDP tunnel using OpenVPN [Електронний ресурс] – Режим доступу: https://medium.com/datadriveninvestor/4-stages of-iot-architecture-explained-in-simple-words-b2ea8b4f777f.

3. Принцип Роботи Протоколу PPTP [Електронний ресурс] – Режим доступу: https://uk.wizcase.com/blog.

4. SSL VPN -крок вперед в технології VPN мереж [Електронний ресурс] – Режим доступу: https://www.anti-malware.ru/node/449.

5. Douglas Crawford. OpenVPN over TCP vs. UDP: what is the difference, and which should I choose? [Електронний ресурс] – Режим доступу: https://www.bestvpn.com/blog/7359/openvpn-tcp-vs-udp-difference-choose/.

6. IPSec — протокол захисту мережевого трафіку на IP-рівні [Електронний ресурс] – Режим доступу: https://www.ixbt.com/comm/ipsecure.shtml.

7. Public Key Infrastructure Setup [Електронний ресурс] – Режим доступу: https://ubuntu.com/server/docs/service-openvpn.

8. Навіщо потрібний OpenVPN. Області застосування [Електронний ресурс] – Режим доступу: https://hyperhost.ua/info/ru/zachem-nuzhen-openvpn-oblasti-primeneniya-dann.

9. OpenVPN 2x HOW TO [Електронний ресурс] – Режим доступу: https://openvpn.net/community-resources/how-to/.

10. Norton. VPN tunnel [Електронний ресурс] – Режим доступу: https://us.norton.com/internetsecurity-privacy-what-is-a-vpn-tunnel.html.

11. Universal TUN/TAP device driver [Електронний ресурс] – Режим доступу: https://www.kernel.org/doc/html/v5.8/networking/tuntap.html.

12. Obtaining and Using Easy-RSA [Електронний ресурс] – Режим доступу: https://easy-rsa.readthedocs.io/en/latest/#getting-started-the-basics.

13. Структура і формат SSL сертифікатів X.509 [Електронний ресурс] – Режим доступу: https://ssl.org.ua/org_x.509.html#gsc.tab=0.

14. Основи шифрування (частина 1) – Алгоритм Діффі-Хеллмана [Електронний ресурс] – Режим доступу: https://www.securitylab.ru/analytics/478912.php.

15. VPN протоколы [Електронний ресурс] – Режим доступу: https://www.cactusvpn.com/ru/beginners-guide-to-vpn/vpn-protocol/.

16. OpenVPN, про який ви так мало знаєте [Електронний ресурс] – Режим доступу: https://www.pvsm.ru/python/305082.

17. OpenVPN Part I: Setup and Configuration [Електронний ресурс] – Режим доступу: https://blog.ipeacocks.info/2017/01/openvpn-vpn-routing-bridge.html.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ