МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

МИХАЙЛОВСЬКИЙ СЕРГІЙ МИХАЙЛОВИЧ

Допускається до захисту:
В.о. завідувача кафедри
прикладної математики та
кібербезпеки,
_____ Луценко А.В.
«__»_____20__р.

РОЗРОБКА СЕРВІСУ МАСКУВАННЯ АДРЕСАЦІЇ КОРЕСПОНДЕНТІВ У КІБЕРПРОСТОРІ

Спеціальність 125 Кібербезпека
Кваліфікаційна (бакалаврська) робота

Науковий керівник:
Загоруйко Л.В.,
к.т.н., доцент,
доцент кафедри прикладної
математики та кібербезпеки,
_____
(підпис)

Оцінка : ___/ ___ / _____
(бали/за шкалою ЄКТС/за національною шкалою)
Голова ЕК: _____
(підпис)

Вінниця – 2024

АНОТАЦІЯ

Михайловський С.М. Розробка сервісу маскування адресації кореспондентів у кіберпросторі. Спеціальність 125 «Кібербезпека». Донецький національний університет імені Василя Стуса, Вінниця, 2024.

У кваліфікаційній (бакалаврській) роботі досліджено методи маскування адресації користувачів в кіберпросторі та розроблено сервіс маскування адресації на базі хмарної платформи Amazon Web Services (AWS) з використанням протоколу OpenVPN. Проаналізовано основні методики та моделі маскування адресації, розглянуто поняття цифрового сліду та його характеристики. Здійснено практичний огляд основних інструментів, необхідних для реалізації сервісу: AWS, EC2 та OpenVPN. Наведено етапи налаштування віртуальної машини AWS EC2, конфігурування безпечного з'єднання з використанням TLS шифрування та генерації необхідних сертифікатів та ключів для автентифікації клієнтів.

Ключові слова: маскування адресації, віртуальна приватна мережа, VPN, OpenVPN, Amazon Web Services, AWS, EC2, цифровий слід, кібербезпека. 48 с., 1 табл., 13 рис., 26 джерел.

ANNOTATION

Mykhailovskyi S.M. Development of a service for masking the addressing of correspondents in cyberspace. Specialty 125 «Cybersecurity». Vasyl Stus Donetsk National University, Vinnytsia, 2024.

his qualification (bachelor's) thesis investigates methods of user address masking in cyberspace and develops an address masking service based on the Amazon Web Services (AWS) cloud platform using the OpenVPN protocol. The main techniques and models of address masking are analyzed, the concept of digital footprint and its characteristics are considered. A practical overview of the main tools necessary for the implementation of the service is carried out: AWS, EC2, and OpenVPN. The stages of configuring an AWS EC2 virtual machine, setting up a secure connection using TLS encryption, and generating the necessary certificates and keys for client authentication are presented.

Keywords: address masking, virtual private network, VPN, OpenVPN, Amazon Web Services, AWS, EC2, digital footprint, cybersecurity. 48 p., 1 table, 13 fig., 26 sources.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

VPN - Віртуальна приватна мережа (Virtual Private Network);
IP - Протокол Інтернету (Internet Protocol);
TLS - Протокол транспортного рівня безпеки (Transport Layer Security);
AWS - Хмарні сервіси Amazon (Amazon Web Services);
EC2 - Еластична служба обчислень Amazon (Amazon Elastic Compute Cloud);
CA - Сертифікаційний орган (Certification Authority);
CRL - Список відкликаних сертифікатів (Certificate Revocation List);
SSH - Захищена оболонка (Secure Shell);
HTTPS - Захищений протокол передачі гіпертексту (Hypertext Transfer Protocol Secure);
UDP - Протокол дейтаграм користувача (User Datagram Protocol);
GUI - Графічний інтерфейс користувача (Graphical User Interface);

ВСТУП

Актуальність. В сучасному інформаційному суспільстві, де обсяг цифрових даних зростає експоненціально, питання захисту приватності та безпеки в кіберпросторі набувають критичного значення. Користувачі мережі інтернет постійно взаємодіють з різноманітними онлайн-сервісами, залишаючи за собою значний цифровий слід, який може містити конфіденційну інформацію про їхні переваги, місцезнаходження, фінансові операції та інші особисті дані. Ця інформація є цінним активом для зловмисників, які можуть використовувати її для здійснення кібератак, крадіжки особистих даних, фішингу, розповсюдження шкідливих програм та інших злочинних дій.

З огляду на це, впровадження ефективних методів маскування адресації стає невід'ємною частиною комплексного підходу до забезпечення безпеки в кіберпросторі. Маскування адресації, засноване на сучасних криптографічних протоколах, таких як OpenVPN, дозволяє приховати справжню цифрову ідентичність користувача, шифрувати його мережевий трафік та захистити його від несанкціонованого доступу до його даних. Це особливо важливо в умовах використання публічних Wi-Fi мереж, які є вразливими до атак зловмисників, а також для обходу цензури та географічних обмежень, що накладаються на доступ до інформації.

Об'єкт дослідження: Забезпечення безпеки та приватності користувачів в кіберпросторі шляхом маскування адресації.

Предмет дослідження: Розробка сервісу маскування адресації кореспондентів у кіберпросторі на основі хмарної платформи Amazon Web Services (AWS) з використанням протоколу OpenVPN.

Мета роботи. Розробка та реалізація сервісу маскування адресації на базі хмарної платформи Amazon Web Services (AWS) з використанням протоколу OpenVPN.

Поставлені задачі:

- Дослідити теоретичні основи маскування в кіберпросторі, зокрема методи анонімізації ідентифікаційних даних, поняття цифрового сліду та його характеристику, а також основні методики та моделі маскування адресації.
- Провести практичний огляд основних інструментів, необхідних для розробки сервісу: хмарної платформи AWS, сервісу Amazon EC2 та протоколу OpenVPN.
- Розробити та реалізувати сервіс маскування адресації на базі AWS EC2 з використанням OpenVPN.
- Провести тестування розробленого сервісу та оцінити його ефективність в порівнянні з іншими доступними VPN сервісами.

РОЗДІЛ І. ТЕОРЕТИЧНІ ОСНОВИ МАСКУВАННЯ В КІБЕРПРОСТОРІ

1.1 Маскування як метод анонімізації ідентифікаційних даних

Маскування в кіберпросторі — це комплекс заходів та методик, спрямованих на приховування або зміну ідентифікаційних даних користувачів у цифровому середовищі для забезпечення їх анонімності та конфіденційності. Цей процес включає використання різноманітних технологічних засобів, таких як віртуальні приватні мережі (VPN), проксі-сервери, системи шифрування даних та інших механізмів забезпечення безпеки. Маскування має на меті замаскувати справжню цифрову ідентичність користувача, включаючи його IP-адресу, місцеположення, браузерні відбитки та інші унікальні атрибути, що дозволяють відслідковувати або ідентифікувати особу в Інтернеті. Основною метою маскування є захист від несанкціонованого доступу, збору, аналізу та використання особистих даних користувачів, а також забезпечення їх права на приватність у кіберпросторі.

Загалом в цій сфері можна виділити 3 основних ланки [1]:

1. ***Технічні методи маскування:***
   - Використання VPN: Забезпечує шифрування інтернет-трафіку та приховування реальної IP-адреси користувача.
   - Проксі-сервери: Діють як посередник між користувачем та веб-сайтом, змінюючи IP-адресу користувача.
   - TOR (The Onion Router): Дозволяє анонімний доступ до Інтернету шляхом маршрутизації трафіку через мережу різних серверів.
   - Шифрування даних: Застосування криптографічних методів для захисту інформації від несанкціонованого доступу [3].
2. ***Методи маскування на рівні застосунків:***
   - Використання анонімних браузерів: Спеціалізовані браузери, які не зберігають дані про користувачів та їх діяльність.

- Анонімізатори електронної пошти: Сервіси, що дозволяють відправляти електронні листи без розкриття реальної електронної адреси відправника.

3. ***Методи маскування поведінкових характеристик:***

- Зміна користувацьких агентів (user agents): Забезпечує приховування інформації про пристрій та браузер користувача. [1]
- Використання спеціалізованих розширень для браузерів: Розширення, що блокують відстеження та видаляють кукі.

За допомогою маскування користувачі можуть ефективно захиститися від різних форм кіберзлочинності, таких як фішинг, крадіжка ідентичності та інші атаки, які спрямовані на отримання конфіденційних даних. Також, маскування важливе для забезпечення безпеки в обміні даними, особливо в бізнес-середовищі, де конфіденційна інформація потребує надійного захисту.

Крім того, маскування дозволяє користувачам обходити інтернет-цензуру та географічні обмеження, надаючи доступ до інформації та ресурсів, які можуть бути обмежені в їх регіоні. Це особливо важливо у ситуаціях, де доступ до інформації обмежені.

Загалом, маскування відіграє ключову роль у забезпеченні цифрової безпеки та конфіденційності, допомагаючи користувачам захистити свої особисті дані та забезпечувати конфіденційність їх онлайн-активності.

1.2 Цифровий слід та його характеристика

Цифровий слід, відомий також як записана чи відстежувана діяльність в Інтернеті або на пристроях, представляє собою набір даних, які залишаються в результаті взаємодії з цифровим середовищем. Згідно з визначенням у словнику Malwarebytes, цей термін використовується для опису цих відстежуваних дій.

Аналізуючи це визначення, можна зазначити, що цифровий слід являє собою збір інформації, що генерується в результаті діяльності в мережі. Це може включати відвідування веб-сайтів, підписку на електронні новини, пошук товарів та інші онлайн-активності. Кожна з цих дій залишає за собою дані, які можуть

бути відстежені. Використовуючи пошукові системи чи сучасне програмне забезпечення, таке як інструменти для відстеження активності, можливо зібрати інформацію про поведінку, переваги, думки та інші аспекти, пов'язані з конкретним цифровим слідом.

Загалом можна виділити два основних види цифрових слідів [2]:

- ***Активні цифрові сліди*** формуються, коли користувач навмисно діляться інформацією про себе, здійснюючи такі дії, як публікація повідомлень або участь у соціальних мережах чи онлайн-форумах. Це відбувається, наприклад, при вході на веб-сайт за допомогою зареєстрованого імені користувача або профілю, при цьому всі публікації стають частиною активного цифрового сліду. До активних цифрових слідів також належать дії, такі як заповнення онлайн-форм (наприклад, підписка на розсилку новин) або згода на використання файлів cookie у браузері.
- ***Пасивні цифрові сліди*** утворюються, коли інформація про користувача збирається без її відома. Це може статися, коли веб-сайти реєструють інформацію про частоту відвідувань сайту, джерела переходу та IP-адреси особи. Цей процес є непомітним і часто залишається поза увагою користувачів. Прикладами пасивних цифрових слідів можуть слугувати діяльність соціальних мереж та рекламодавців, які аналізують вподобання, публікації та коментарі для створення персоналізованих профілів, націлених на розповсюдження рекламного контенту.

Цифрові сліди, які на перший погляд можуть здаватися одним із негативних аспектів онлайн-активності, насправді мають багато переваг, що сприяє покращенню взаємодії в Інтернеті. Наявність цифрового сліду може спростити онлайн-світ, яким він є сьогодні.

Сервіси, такі як Google AdSense [11], використовують цифровий слід для показу відповідної реклами. Це засновано на пасивному цифровому сліді, отриманому з дозволу користувачів, при реєстрації на вебсайті, що включає умови використання даних веб-серфінгу для показу більш релевантної реклами.

Ця практика дозволяє показувати рекламу, яка відповідає інтересам користувача, замість нецільової реклами, що не сприяє ані користувачеві, ані рекламодавцю.

YouTube, інший продукт Google, також використовує цифровий слід для підвищення якості послуги, показуючи відео, пов'язані з тими, що користувач переглядав раніше. Це розраховано на підвищення відповідності контенту інтересам користувача.

Facebook використовує цифровий слід для покращення користувацького досвіду у соціальних мережах, показуючи рекламу, засновану на вподобаннях користувача та його друзів, а також пропонуючи запропонований список друзів на основі спільних зв'язків та алгоритмів, заснованих на вподобаннях сторінок.

Мобільні пристрої з підтримкою GPS додають географічну інформацію до цифрового сліду, що покращує веб-сервіси, такі як Google Карти, дозволяючи точніше знаходити місцезнаходження, використовуючи історію пошуку та поточні дані GPS.

Не дивлячись на позитивну складову цифрового сліду, також існують і певні недоліки, пов'язані із збором даних про активність клієнтів в Інтернеті, який часто відбувається без їхнього відома, особливо у випадку пасивного цифрового сліду.

Сучасні технології дозволяють користувачам отримувати доступ до величезного обсягу інформації, що знаходиться на веб-сайтах. Цей процес характеризується швидким переглядом значної кількості веб-сторінок за короткий час. У цьому контексті користувачі часто стикаються з умовами використання веб-сайтів, які містять значний обсяг тексту. Через обмеженість часу та уваги користувачі зазвичай не читають ці умови детально, погоджуючись з ними без належного ознайомлення.

Ця тенденція дозволяє веб-сайтам легітимно збирати інформацію про онлайн-активність користувачів. Результатом цієї практики є формування цифрового сліду, що складається з даних про користувача, які відстежуються під час його онлайн-діяльності. Цей цифровий слід містить інформацію про поведінку користувача, його переваги, інтереси та споживчі звички.

Збільшення кількості даних, що збираються про користувачів, породжує проблему порушення їх приватного життя. Інформація, яка вважається особистою та конфіденційною, може бути доступна третім сторонам, що може мати негативні наслідки. Наприклад, використання пошукової системи Google для введення повного імені може призвести до появи несподіваних результатів, що розкривають особисту інформацію про користувача, яку він не хотів би публікувати. Крім того, фотографії, розміщені на соціальних мережах, можуть бути знайдені за допомогою пошуку зображень, що може призвести до непередбачуваних наслідків.

Соціальні мережі, такі як Facebook, Instagram, надають інструменти контролю над тим, що ділиться в Інтернеті через налаштування конфіденційності, але в їхніх правилах та умовах зазначено, що вони володіють правами на весь вміст, яким ділиться особа. Навіть після деактивації облікового запису, немає гарантій, що вся інформація, якою користувач ділився до деактивації, буде повністю знищена. Більшість соціальних мереж не видаляють інформацію про користувачів зі своїх серверів, використовуючи її для аналізу, навіть після виходу користувачів з сайту.

Таким чином, цифровий слід є двогранним явищем. З одного боку, він створює можливості для інновацій, зручності та вдосконалення послуг. З іншого - він породжує серйозні виклики, пов'язані з приватністю, безпекою та етикою в цифровому світі. Саме тому важливо підходити до цифрового сліду з обережністю та розумінням, балансуючи між перевагами, які він надає, та ризиками, які він створює.

1.3 Методики та моделі маскування адресації

В епоху, коли дані про місцезнаходження стимулюють інновації, але водночас викликають занепокоєння щодо конфіденційності, маскування адрес стало важливим засобом захисту. Навмисне приховування або модифікація адрес не дозволяє стороннім особам пов'язувати конфіденційну інформацію про

місцезнаходження з особами чи організаціями. Віртуальні приватні мережі (VPN) та складні криптографічні методи є основою сучасних стратегій маскування адрес. VPN створюють безпечні, зашифровані тунелі, які маскують справжню IP-адресу користувача, основний цифровий ідентифікатор його фізичного місцезнаходження.

Криптографія відіграє багатогранну роль у маскуванні адрес. Окрім фундаментального шифрування, що використовується у VPN, сучасні криптографічні методи дозволяють безпосередньо маніпулювати адресними даними. Такі методи, як псевдонімізація, коли адреси замінюються унікальними ідентифікаторами, і збурення геокодування, коли вносяться незначні неточності, захищають конфіденційність місцезнаходження, не ставлячи під загрозу корисність даних, що лежать в основі. Взаємодія між VPN і цими криптографічними моделями пропонує надійну основу для маскування адрес в різних додатках, забезпечуючи індивідуальну і організаційну конфіденційність в умовах мінливого цифрового ландшафту.

1.3.1 Основи криптографії та криптопротоколів

Одними із сервісів віртуальних приватних мереж, які будуть розглядатися в цьому розділі, є забезпечення конфіденційності інформації й процеса автентифікації. Ці сервіси забезпечується за допомогою використання криптографічних алгоритмів й криптопротоколів.

Криптографічний алгоритм — це математична формула, що описує процеси шифрування і розшифрування. Щоб зашифрувати відкритий текст, криптоалгоритм працює в сполученні з ключем - словом, числом або фразою.

Криптографічних алгоритмів існує безліч. В загальному вони поділяються на безключові, одноключові й двоключові алгоритми. Криптографія з ключем - це алгоритм впливу на передані дані відомий усім стороннім особам, але він залежить від деякого параметра - "ключа", яким володіють лише відправник і одержувач. Їх можна поділити на:

- Симетричні криптоалгоритми [4] використовують однаковий ключ для шифрування та розшифровки повідомлень.
- Асиметричні криптоалгоритми використовують різні ключі: "відкритий" ключ для шифрування, який доступний всім, та "закритий" ключ для розшифровки, який зберігається лише у отримувача.

Хешування, означає процес конверсії вхідного набору даних довільної довжини у вихідний бітовий рядок стандартної довжини. Цей процес, також відомий як хеш-функції або функції згортки, генерує результат, який зазвичай називається хешем, хеш-кодом або дайджестом повідомлення. Хешування знаходить застосування у порівнянні даних, оскільки відрізняючі хеш-коди однозначно вказують на різні масиви даних, тоді як однакові хеш-коди, хоч і з високою ймовірністю, але не гарантують тотожності даних. Через обмежену кількість можливих значень хеш-функцій порівняно з варіантами вхідних даних, виникають так звані колізії, що є ключовим фактором у оцінці якості хеш-функцій. Існує численні алгоритми хешування з різноманітними характеристиками, такими як розрядність, обчислювальна складність, криптостійкість тощо, вибір серед яких диктується потребами конкретного завдання. Простими прикладами хеш-функцій можуть бути контрольні суми або CRC. [4]

Електронний підпис описує електронні дані, додані або логічно пов'язані підписувачем з іншими електронними даними, які використовуються як підпис. Особливим випадком є Електронний цифровий підпис (ЕЦП), він створюється шляхом криптографічного перетворення даних, додаються до цих даних або логічно з ними асоціюються, забезпечуючи підтвердження цілісності даних та ідентифікацію підписувача. ЕЦП накладається за допомогою особистого ключа та перевіряється відкритим ключем.

Наступний рисунок 1.1 ілюструє класифікацію криптографічних алгоритмів:

Рисунок 1.1 – Класифікація криптографічних алгоритмів [8]

Протокол — опис розподіленого алгоритму, в процесі виконання якого два (чи більше) учасники послідовно виконують певні дії і обмінюються повідомленнями [2]. Послідовність кроків протоколу групується в цикли (раунди). В якості учасників (інакше – суб'єктів, сторін) протоколу можуть виступати не тільки користувачі чи абоненти, а й процеси, які виконують яку-небудь функціональну роль, наприклад клієнтські і серверні додатки. Припускається, що всі учасники виконують в ньому якусь активну роль, а пасивні спостерігачі не являються учасниками протоколу.

Комунікаційний протокол, зокрема, визначає послідовність дій учасників у процесі передачі або обміну інформацією. Типовий комунікаційний протокол включає процедури встановлення з'єднання або сесії, вибору маршруту, виявлення спотворень, відновлення передаваної інформації та інше.

Безпека протоколу полягає у забезпеченні дотримання властивостей, які характеризують безпеку, таких як доступність, конфіденційність та цілісність. Для цього у протоколах використовуються спеціальні конструкції. Протоколи, які підтримують хоча б одну з функцій безпеки, класифікуються як захищені або, більш точно, як протоколи забезпечення безпеки.

Криптографічний протокол — це специфічний тип протоколу, призначеного для реалізації функцій криптографічної системи, де учасники

використовують криптографічні алгоритми. Така система зосереджена на забезпеченні безпеки інформації за допомогою криптографічних методів, включаючи забезпечення конфіденційності, цілісності, автентифікації, запобігання відмовам та трасуванню. Складовими криптографічної системи можуть бути системи шифрування, ідентифікації, імітозахисту, цифрового підпису та інші, а також ключова система, що забезпечує функціонування решти систем [12].

1.3.2 Класифікація та основи побудови віртуальних приватних мереж

Інтеграція локальних мереж та окремих комп'ютерних систем через відкриті комунікаційні середовища у єдину віртуальну структуру, що забезпечує захист обмінюваної інформації, відома як Приватна Віртуальна Мережа (VPN). Термін "віртуальна" тут вказує на формування мережі як підмножини існуючої фізичної мережі зі симульованими комунікаційними каналами.

Ключовою характеристикою VPN є її ізоляція від основної мережі, що забезпечує достатній рівень безпеки для гарантії конфіденційності, цілісності переданої інформації, а також автентифікації сторін і недопущення відмови від авторства.

Навіть у разі використання мереж з меншим рівнем довіри, наприклад публічних мереж, захищена логічна структура VPN забезпечує високий рівень довіри завдяки застосуванню криптографічних методів.

Основні функції VPN сервісів:

1. Забезпечення конфіденційності;
2. Забезпечення цілісності;
3. Автентифікація та запобігання відмовленню від авторства.

Криптографічне шифрування є стандартним методом забезпечення конфіденційності, хоча реалізація шифрувальних алгоритмів може бути складною. Однією з ключових проблем у впровадженні VPN є ефективне

управління криптографічними ключами, особливо при збільшенні кількості користувачів.

Шифрування може призводити до зниження продуктивності. Використання апаратних рішень для шифрування дозволяє знизити навантаження на захисні пристрої та підтримувати високу швидкість передачі даних. Однак, в разі атак на VPN, існує ризик компрометації програмних компонентів, зокрема тих, що відповідають за шифрування. Несанкціонований вплив на апаратні компоненти, з іншого боку, є менш ймовірним. Для апаратного шифрування часто використовуються спеціалізовані інтегральні схеми прикладної орієнтації.

Запобігання відмовленню від авторства – це додаткова функція, що реалізується на базі автентифікації. У захищеному спілкуванні часто виникають випадки, коли крім підтвердження того, що абонент є саме тим, за кого він себе намагається видати, важливо отримати незаперечні докази того, що повідомлення одержано від конкретного користувача. Також буває необхідним доказове підтвердження того, що певний користувач дійсно одержав деяке повідомлення. Ці функції захисту у ряді випадків повинні бути невід’ємною складовою реалізації VPN

У контексті віртуальних приватних мереж, будь-який вузол цієї мережі, що є частиною створеного захищеного тунелю, може бути класифікований як кінцева або проміжна точка в потоці повідомлень, що охороняються. Це призводить до декількох можливих сценаріїв конфігурації захищеного віртуального каналу, зокрема:

- кінцеві точки тунелю співпадають з кінцевими точками потоку повідомлень;
- кінцевою точкою захищеного тунелю обирають брандмауер або граничний маршрутизатор локальної мережі, захищений тунель утворюється лише у публічній мережі;

- в якості кінцевих точок захищеного тунелю виступають засоби, що встановлені не на комп’ютерах користувачів, а на площах провайдерів Інтернет.

Віртуальні приватні мережі (VPN) можна класифікувати за декількома ключовими параметрами:

1. Залежно від характеру використовуваного середовища передачі даних, VPN рішення поділяються на:
   - Захищені VPN: Це найбільш поширений тип, що дозволяє створити захищену підмережу в межах потенційно ненадійного середовища, як-от Інтернет. Приклади включають IPSec, OpenVPN [18], і PPTP.
   - Довірчі VPN: Використовуються у середовищах, які вважаються надійними. Головна мета полягає в створенні віртуальної підмережі без особливого фокусу на забезпечення безпеки. Наприклад, MPLS і L2TP, де L2TP часто використовується у поєднанні з IPSec.
2. Відповідно до способу реалізації, VPN поділяють на:
   - Апаратно-програмні VPN: Включають спеціалізований набір програмно-апаратних засобів для високої продуктивності і рівня захисту.
   - Програмні VPN: Використовують стандартний персональний комп'ютер з відповідним програмним забезпеченням для реалізації VPN.
   - Інтегровані VPN рішення: Комбінують VPN функціональність з іншими задачами, такими як фільтрація мережевого трафіку, мережеві файерволи, і гарантування якості обслуговування.
3. Залежно від призначення, розрізняють [1]:
   - Intranet VPN: Використовуються для з'єднання декількох розподілених приватних мереж у єдину захищену мережу, яка обмінюється даними через відкриті канали зв'язку. Це дозволяє організаціям забезпечити захищене підключення між різними підрозділами.

- Remote Access VPN: Для створення захищеного каналу між корпоративною мережею та індивідуальними користувачами, які підключаються до корпоративних ресурсів з дому або у відрядженні.
- Extranet VPN: Забезпечують підключення «зовнішніх» користувачів (клієнтів) до мережі організації, з особливим акцентом на забезпечення додаткових рівнів захисту для обмеження доступу до чутливої або конфіденційної інформації.

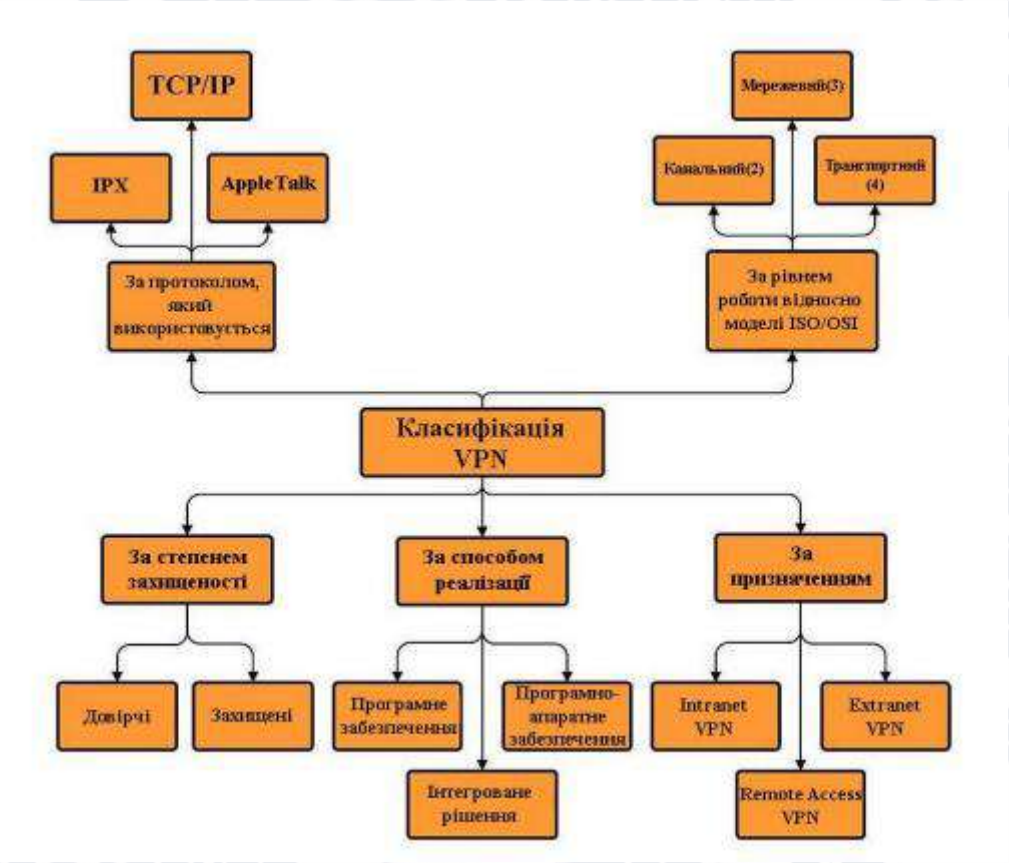

Рисунок 1.2 – Класифікація VPN

Основні компоненти VPN:

- VPN-шлюз - мережеве пристрій, підключений до декількох мереж, виконує функції шифрування, ідентифікації, аутентифікації, авторизації та тунелювання. Може бути реалізований як програмно, так і апаратно.
- VPN-клієнт (хост) реалізується програмно. Виконує функції шифрування і аутентифікації. Мережа може бути побудована без використання VPN-клієнтів.

- Тунель - логічний зв'язок між клієнтом і сервером. В процесі реалізації тунелю використовуються методи захисту інформації.
- Граничний сервер - це сервер, який є зовнішнім для корпоративної 178 мережі. В якості такого сервера може виступати, наприклад, брендмауери або система NAT.
- Забезпечення безпеки інформації VPN - ряд заходів щодо захисту трафіку корпоративної мережі при проходженні по тунелю від зовнішніх і внутрішніх загроз.

Підтримка VPN на різних рівнях моделі OSI:

- Канальний рівень: L2TP, PPTP і ін. (Авторизація і аутентифікація), Технологія MPLS (встановлення тунелю).
- Мережевий рівень: IPSec (архітектура «хост-шлюз» і «шлюз-шлюз», підтримка шифрування, авторизації та аутентифікації, проблеми з реалізацією NAT).
- Транспортний рівень: SSL / TLS (архітектура «хост-хост» з'єднання з кінця в кінець, підтримка шифрування і аутентифікації, реалізований тільки для підтримки TCP-трафіку).

1.3.3 Протокол реалізації VPN PPTP

Протокол PPTP (Point-to-Point Tunneling Protocol) [7], призначений для створення захищених віртуальних каналів при доступі віддалених користувачів до локальних мереж через Інтернет. Він передбачає створення криптозахищеного тунелю на канальному рівні моделі OSI як для випадку прямого з'єднання віддаленого комп'ютера з відкритою мережею, так і для випадку приєднання його до відкритої мережі по телефонній лінії через провайдера. При встановленні з’єднання використовується застарілий протокол двухточкового зв'язку Poinl–to–Point Protocol (далі – PPP) і є його розширенням.

Основний опис протоколу поданий в RFC2637 відкритим міжнародним співтовариством проектувальників, учених, мережних операторів і провайдерів

(Internet Engineering Task Force, далі – IETF). За замовчуванням протокол використовує передачу даних за допомогою TCP пакетів і резервує 1723 порт.

Point-to-Point Protocol (PPP), який слугує стандартом у мережі Інтернет, представляє собою комплексний набір протоколів для забезпечення двоточкового зв'язку. До цього сімейства протоколів належать:

- Link Control Protocol (LCP): Цей протокол забезпечує встановлення, тестування та управління з'єднаннями зв'язку. Він також дозволяє домовлятися про параметри використання каналів зв'язку та їх відключення, коли це необхідно. LCP підтримує як синхронні, так і асинхронні лінії, а також біт- та байт-орієнтоване кодування.
- Network Control Protocol (NCP): Цей протокол використовується для управління і конфігурації мережних з'єднань.
- Multi Link PPP (MLPPP): Цей варіант PPP дозволяє агрегувати кілька каналів зв'язку для підвищення пропускної здатності та надійності.
- Password Authentication Protocol (PAP): Протокол, що використовується для аутентифікації користувачів за допомогою пароля.
- Challenge Handshake Authentication Protocol (CHAP): Протокол для аутентифікації, який використовує процес визнання для перевірки ідентичності користувача.

Процес функціонування Point-to-Point Protocol (PPP) починається з ініціації з'єднання через прямий комунікаційний канал. На початковому етапі PPP передає пакети Link Control Protocol (LCP), що містять інформацію для конфігурації з'єднання та верифікації каналу передачі даних. Як тільки канал зв'язку успішно встановлений і LCP узгодив необхідні параметри, PPP ініціює передачу пакетів Network Control Protocol (NCP). Ця процедура має на меті вибір та налаштування конфігурації одного або декількох протоколів мережного рівня.

Після завершення налаштування протоколів мережного рівня, можлива передача дейтаграм з цих протоколів через зазначений канал. Конфігурація каналу підтримується до моменту, коли пакети LCP або NCP явно не припиняють з'єднання або поки не виникає зовнішня подія, така як закінчення терміну дії

таймера бездіяльності або втручання користувача. Цей процес забезпечує гнучке і ефективне встановлення з'єднань у різних мережевих середовищах.

Протокол PPP для достатньої універсальності і застосовності до широкої різноманітності систем включає протокол контролю каналу LCP. LCP використовується, щоб автоматично погоджувати опції формату інкапсуляції, змінювати межі розмірів пакетів, виявляти зациклення ланки і інші помилкові ситуації, пов'язані з відмінностями конфігурацій, і розривати зв'язок.

Існує три класи пакетів LCP:

- пакети для організації каналу зв'язку. Використовуються для організації і вибору конфігурації каналу;
- пакети для завершення дії каналу. Використовуються для завершення дії каналу зв'язку;
- пакети для підтримки працездатності каналу. Використовуються для підтримки і відладки каналу.

Канали Point-to-Point Protocol (PPP) стикаються з рядом складнощів, пов'язаних з використанням сімейства мережевих протоколів. Одним з прикладів цих складнощів є розподіл та керування IP-адресами, які вже є предметом уваги в локальних обчислювальних мережах, але стають ще більш складними в контексті комутованих каналів зі схемою точка-до-точки. Для вирішення цих проблем використовується стек протоколів Network Control Protocol (NCP), кожен з яких спеціалізується на конкретних функціях, необхідних для відповідних протоколів мережного рівня. Цей підхід дозволяє PPP ефективно керувати різними аспектами мережевих комунікацій, включаючи розподіл IP-адрес у складних мережевих умовах.

У контексті деяких каналів Point-to-Point Protocol (PPP), існує необхідність верифікації достовірності вузлів перед дозволом на обмін мережевими протокольними пакетами. Ця функція конфігурації дозволяє узгодити використання конкретного протоколу для забезпечення аутентифікації. За замовчуванням, аутентифікація не вимагається, але користувач має можливість

активувати різні параметри конфігурації протоколу аутентифікації через пакети "Запит конфігурації".

В процесі встановлення, підтримки та роз'єднання зв'язку у режимі "точка-до-точки", ланка передачі даних у PPP проходить крізь декілька відмінних стадій. Ці стадії можна зобразити у вигляді спрощеної діаграми станів, що ілюструє послідовність переходів і процесів, які мають місце під час роботи PPP.

Рисунок 1.3 – Діаграма станів протоколу PPP

Point-to-Point Tunneling Protocol (PPTP) був розроблений Microsoft за співпраці з Ascend Communications, 3Com/Primary Access, ECI-Telematics та US Robotics. PPTP є модифікацією Point-to-Point Protocol (PPP), яка впроваджує можливість створення зашифрованих тунелів на канальному рівні моделі OSI. Хоча PPTP був запропонований як проект стандарту в інтернеті, він не отримав статус офіційного стандарту.

Основна функція PPTP полягає у забезпеченні віддаленого доступу користувачів до локальних мереж, як при безпосередньому підключенні віддаленого комп'ютера до публічної мережі, так і через підключення до публічної мережі за допомогою телефонної лінії через інтернет-провайдера. Використання PPTP дозволяє віддаленому користувачу, підключеному до локальної мережі через сервер віддаленого доступу (Remote Access Server, RAS), відчувати себе так, ніби його комп'ютер фізично присутній в локальній мережі. Це досягається шляхом тунелювання повідомлень, тобто передачі даних у зашифрованому вигляді через встановлений тунель.

PPTP тунель складається з каналу управління і з каналу передачі даних. Спочатку встановлюється канал управління, а потім канал даних. Організація

каналу даних здійснюється за допомогою протоколу інкапсулювання Generic Routing Encapsulation (далі – GRE). GRE інкапсулює мережевий рівень, наприклад IPX, AppleTalk, щоб забезпечити можливість їх передачі в IP–мережах. Однак GRE не має можливості встановлювати сесії і забезпечувати захист даних від зловмисників. Для цього використовується здатність PPTP створювати з'єднання для управління тунелем. Застосування GRE в якості методу інкапсуляції обмежує поле дій PPTP тільки мережами IP.

Рисунок 1.4 – Архітектура протоколу PPTP

Застосування шифрування в рамках Point-to-Point Tunneling Protocol (PPTP) забезпечує захист даних під час їх передачі через Інтернет, перешкоджаючи несанкціонованому доступу до інформації. Сучасна реалізація PPTP включає два основних методи шифрування:

- Microsoft Point-to-Point Encryption (MPPE): Цей метод шифрування є сумісним виключно з Microsoft Challenge Handshake Authentication Protocol (MS-CHAP). MPPE пропонує зашифроване з'єднання, використовуючи алгоритми, які синхронізуються з процедурами аутентифікації MS-CHAP.
- EAP-Transport Layer Security (EAP-TLS): Цей метод здатний автоматично вибирати довжину шифрувального ключа під час процесу узгодження параметрів між клієнтом та сервером. EAP-TLS вважається більш безпечним, оскільки він забезпечує двосторонню аутентифікацію та дозволяє більш гнучке управління ключами шифрування.

В рамках Point-to-Point Tunneling Protocol (PPTP) застосовуються різноманітні механізми аутентифікації. У версії PPTP, розробленій Microsoft, підтримуються такі протоколи аутентифікації:

- Password Authentication Protocol (PAP): Цей протокол базується на передачі ідентифікаторів користувачів та їх паролів у незашифрованому вигляді. Це означає, що ідентифікаційні дані передаються через мережу без будь-якого рівня шифрування, що може створювати потенційні ризики безпеки.
- Challenge Handshake Authentication Protocol (CHAP): Цей протокол включає отримання від сервера випадково згенерованого числа, на основі якого здійснюється шифрування пароля. Цей підхід гарантує, що пароль не передається через мережу у відкритому вигляді, а зашифроване представлення пароля змінюється при кожному вході в систему, забезпечуючи вищий рівень безпеки порівняно з PAP.

1.3.4 Протокол реалізації VPN L2TP

Layer 2 Tunneling Protocol (L2TP) [7] представляє собою протокол тунелювання, який діє на канальному рівні моделі OSI. Цей протокол, розроблений Internet Engineering Task Force (IETF), є еволюцією протоколу Point-to-Point Tunneling Protocol (PPTP) і використовується постачальниками інтернет-послуг для створення віртуальних приватних мереж (VPN) через Інтернет. L2TP був створений на основі протоколів PPTP і Layer 2 Forwarding (L2F), з метою надання захищеного тунелювання трафіку PPP через універсальні мережі.

Згідно зі специфікацією L2TP, роль сервера віддаленого доступу провайдера виконує L2TP Access Concentrator (LAC), який забезпечує віддаленим користувачам мережевий доступ до їх локальних мереж через Інтернет. Натомість, як сервер віддаленого доступу для локальної мережі виступає L2TP Network Server (LNS), що підтримує сумісність з протоколом PPP.

На відміну від PPTP, L2TP не обмежений використанням протоколу IP і може використовуватися в мережах з комутацією пакетів, як-от ATM або Frame

Relay. L2TP додатково включає функції управління потоками даних та підтримує безпеку за допомогою протоколів IPSec Authentication Header (AH) та Encapsulating Security Payload (ESP).

L2TP використовує два типи пакетів: керуючі та інформаційні. Керуючі пакети застосовуються для створення, підтримки та роз'єднання тунелів, в той час як інформаційні пакети служать для інкапсуляції та передачі PPP-кадрів. Керуючі повідомлення передаються через надійний контрольний канал L2TP для забезпечення їх доставки, тоді як інформаційні повідомлення передаються через ненадійний канал даних і не повторно відсилаються у випадку їх втрати. Архітектура протоколу L2TP передбачає передачу PPP-кадрів через ненадійний канал даних, які спочатку інкапсульовані в L2TP, а потім - в транспортні пакети (наприклад, UDP, Frame Relay, ATM тощо). Керуючі повідомлення відсилаються через надійний керуючий канал L2TP у тому ж пакетному транспорті.

Рисунок 1.5 – Архітектура протоколу L2TP [12]

Layer 2 Tunneling Protocol (L2TP) передбачає формування тунелю та відповідного керуючого каналу перед ініціюванням входу чи виходу дзвінків. Сесія має бути активована для того, щоб L2TP міг передавати PPP-кадри через тунель. У межах одного тунелю можливе існування декількох сесій між тими ж самими L2TP Access Concentrator (LAC) і L2TP Network Server (LNS). Відмінно від PPTP та L2F, L2TP пропонує можливість відкриття декількох тунелів між кінцевими абонентами, кожен з яких може бути призначений для окремого застосування, забезпечуючи таким чином більшу гнучкість і безпеку у тунелюванні.

L2TP не використовує MPPE для шифрування PPP-датаграм. Замість цього, служби шифрування в L2TP базуються на IPSec у транспортному режимі. Комбінація L2TP та IPSec, відома як L2TP/IPSec, забезпечує значно вищий рівень захисту даних порівняно з PPTP, використовуючи алгоритми шифрування 3DES або AES. У випадках, коли такий високий рівень захисту не є критично важливим, можна використовувати алгоритм DES з одним 56-бітним ключем. Шифрування DES/3DES використовує ключі, отримані в процесі узгодження за протоколом Internet Key Exchange (IKE). Аутентифікація даних здійснюється за допомогою алгоритму Hash Message Authentication Code (HMAC), який генерує хеш довжиною 128 біт.

Інкапсуляція пакетів в L2TP/IPSec виконується на двох рівнях. На першому рівні до PPP-кадру (IP-датаграми) додаються заголовки L2TP і UDP. На другому рівні до повідомлення L2TP додаються заголовок і замикач протоколу ESP IPSec. Все це інкапсулюється в IP-пакет для передачі або декапсуляції. Замикач перевірки достовірності в IPSec забезпечує цілісність та аутентифікацію повідомлення. У IP-заголовку вказуються початкова та кінцева IP-адреси, що відповідають VPN-клієнту та VPN-серверу.

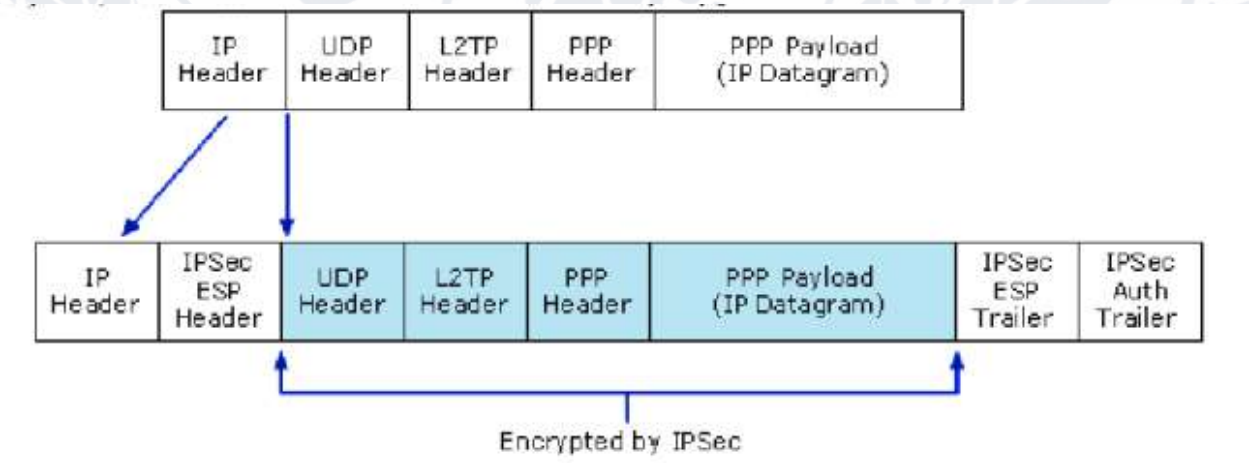

Рисунок 1.6 – Інкапсуляція пакетів L2TP/IPSec [11]

1.3.5 Протоколи SSL/TLS та його реалізація OpenVPN

Transport Layer Security (далі – TLS) [16] – криптографічний протокол, що забезпечує захищену передачу даних між вузлами в мережі Інтернет. Заснований

на SSL–протоколі версії 3.0. TLS використовує асимметричное шифрування для автентифікації, симетричне шифрування для конфіденційності та коди автентичності повідомлень для збереження цілісності повідомлення. Ядром протоколу є технологія комплексного використання асиметричних і симетричних криптосистем.

В основу протоколу, представленої на рисунку 1.7 ввійшли 2 протоколи: протокол запису та протокол діалогу, який в свою чергу, складається з трьох підпротоколів (протокол зміни специфікації шифру, протокол сповіщення та протокол рукопотискання).

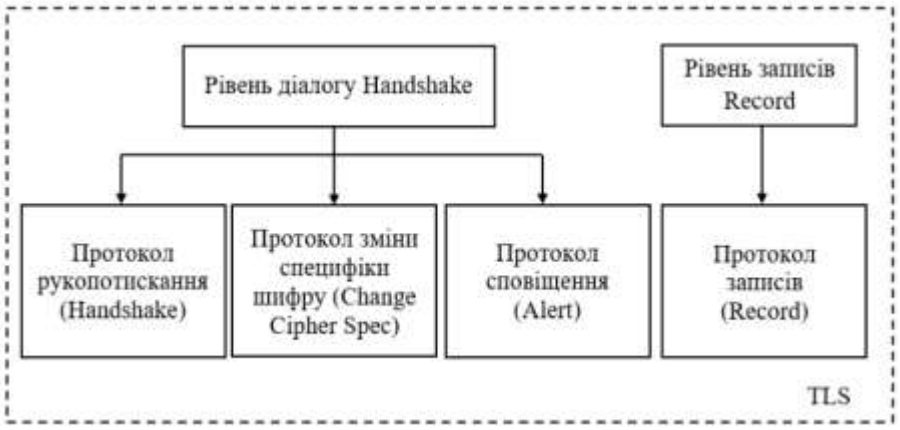

Рисунок 1.7 – Архітектура протоколу TLS

TLS VPN, базована на OpenVPN, представляє собою рішення для створення віртуальних приватних мереж (VPN), використовуючи Transport Layer Security (TLS). OpenVPN, запущений у 2002 році, є відкритим програмним забезпеченням, призначеним для розгортання site-to-site VPN мереж, що використовують TLS для забезпечення безпеки. OpenVPN функціонує як безпечний тунель для передачі даних через стандартний порт 1194, використовуючи транспортні протоколи TCP та UDP [22].

UDP є більш ефективним у цьому контексті, оскільки він підтримує передачу трафіку мережевого рівня та вище через TUN-з'єднання або трафіку канального рівня та вище через TAP-з'єднання. Оскільки OpenVPN може діяти на канальному або навіть фізичному рівні в контексті клієнта, надійність передачі даних може забезпечуватися вищими рівнями моделі OSI. Це робить UDP особливо придатним для використання з OpenVPN.

Натомість, налаштування тунелю з використанням TCP може призвести до подвійної перевірки на цілісність даних, оскільки TCP-сегменти від клієнта містяться всередині TCP-сегментів OpenVPN. Це не збільшує надійність, але може знизити швидкість з'єднання через додаткові перевірки цілісності, що в цьому випадку є нераціональними.

Архітектура мережі OpenVPN складається з наступних ключових компонентів:

- Certification Authority (CA): Цей елемент мережі відповідає за видачу цифрових сертифікатів на запити вузлів VPN-мережі. Ці сертифікати підписані сертифікатом самого CA, що дозволяє вузлам VPN-мережі перевірити автентичність CA. CA також керує Certificate Revocation List (CRL), який містить інформацію про відкликані сертифікати.
- OpenVPN Server: Програмне забезпечення сервера OpenVPN формує тунель всередині незахищеної мережі, такої як Інтернет. Цей тунель дозволяє забезпечити безпечний зашифрований обмін даними між вузлами, що беруть участь у мережі OpenVPN.
- OpenVPN Client: Клієнтське програмне забезпечення OpenVPN встановлюється на всіх вузлах, які потребують захищеного каналу передачі даних до сервера OpenVPN. За належного налаштування сервера OpenVPN, можлива захищена передача даних не тільки між клієнтами та сервером, але й між самими клієнтами OpenVPN.
- X.509 Certificates: Ці сертифікати представляють собою публічні ключі, завірені CA. Вони застосовуються для шифрування даних, а підтвердження автентичності сертифікату CA дозволяє ідентифікувати сторону, яка передає зашифровані дані.
- Приватні ключі: Ці ключі є конфіденційними і повинні генеруватися та зберігатися на кожному вузлі мережі OpenVPN. Вони призначені для розшифровування даних і ніколи не повинні передаватися через мережу.
- Certificate Revocation List (CRL): Цей список містить інформацію про сертифікати, які були відкликані та втратили довіру. CRL створюється та

оновлюється на вузлі CA і використовується для відключення вузлів від мережі шляхом внесення їх сертифікатів у список.

Для забезпечення безпеки управління каналом та передачі даних, протокол Transport Layer Security (TLS) інтегрує бібліотеку OpenSSL. Ця інтеграція дозволяє використовувати широкий спектр алгоритмів шифрування, що пропонуються цією бібліотекою. Крім того, застосування пакетної автентифікації з використанням Hash-Based Message Authentication Code (HMAC) сприяє підвищенню рівня безпеки. Апаратне прискорення шифрування також використовується для оптимізації продуктивності.

Процес інкапсуляції TLS-пакетів у контексті OpenVPN демонструється на рисунку 2.9. На ньому показано, як OpenVPN тунель інкапсулюється в сегмент UDP як частина корисного навантаження. Ця візуалізація ілюструє механізм, за допомогою якого дані передаються безпечно через відкрите середовище, таке як Інтернет.

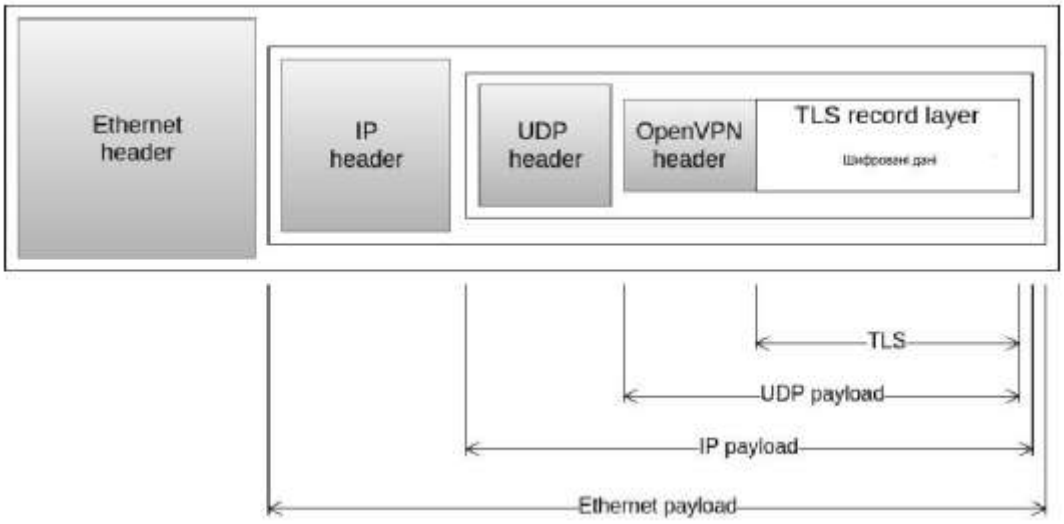

Рисунок 1.8 – Інкапсуляція TLS пакетів з використанням OpenVPN [13]

Висновки до розділу 1

В ході дослідження, представленого в даному розділі, було проведено аналіз теоретичних аспектів маскування в кіберпросторі. Було визначено, що маскування відіграє ключову роль у забезпеченні анонімності та

конфіденційності користувачів шляхом приховування їх цифрової ідентичності, що є особливо актуальним в умовах постійного зростання кількості кіберзагроз. Дослідження поняття цифрового сліду показало його двояку природу: з одного боку, він сприяє персоналізації онлайн-сервісів, а з іншого - несе потенційні ризики для приватності користувачів.

В розділі детально розглянуто різноманітні методики та моделі маскування адресації, включаючи VPN, проксі-сервери, мережу TOR та методи шифрування даних. Особлива увага приділена ролі криптографії, яка лежить в основі забезпечення конфіденційності та цілісності даних під час передачі.

Проведене дослідження дозволило сформувати теоретичну базу для подальшої розробки та реалізації сервісу маскування адресації, а також визначити основні вимоги до його функціональності та рівня безпеки.

РОЗДІЛ ІІ. РОЗРОБКА СЕРСВІСУ МАСКУВАННЯ АДРЕСАЦІЇ НА ОСНОВІ AWS ТА OPENVPN

2.1 Практичний огляд основних засобів (AWS, EC2, OpenVpn)

Хмарні сервіси - це модель надання послуг, яка дає змогу користувачам отримувати доступ до обчислювальних ресурсів, таких як сервери, сховища даних, мережі та програмне забезпечення, через Інтернет, а не володіти та керувати ними самостійно. Це означає, що користувачі платять лише за ті ресурси, які вони використовують, що може призвести до значної економії коштів.

Хмарні сервіси пропонують безліч переваг для підприємств і окремих користувачів, зокрема:

- Масштабованість: Хмарні ресурси можна легко масштабувати вгору або вниз залежно від потреб, що робить їх ідеальними для компаній, які швидко зростають або мають мінливе навантаження.
- Гнучкість: Хмарні сервіси пропонують широкий спектр послуг, які можна налаштувати відповідно до конкретних потреб користувачів.
- Надійність: Хмарні провайдери мають великі центри обробки даних з резервним копіюванням і відновленням даних, що робить їх більш надійними, ніж локальні сервери.
- Доступність: До хмарних сервісів можна отримати доступ з будь-якого місця з підключенням до Інтернету.

Щодо вибору під час розробки VPN сервісу, оптимальний варіант – Amazon Web Services (AWS). Адже він містить велику кількість функцій, які допоможуть зберегти цілісність та конфіденційність даних при використанні.

2.1.1 AWS

Amazon Web Services (AWS) - це платформа хмарних обчислень, що пропонує широкий спектр послуг, включаючи обчислення, сховище даних, мережу, бази даних, аналітику, машинне навчання, штучний інтелект та багато

іншого. AWS використовують мільйони компаній по всьому світу, включаючи стартапи, великі підприємства та державні установи.

AWS була запущена в 2006 році компанією Amazon.com, як платформа для розміщення веб-додатків. З часом AWS розширила свій спектр послуг і стала однією з найпопулярніших платформ хмарних обчислень у світі.

AWS пропонує широкий спектр продуктів, які можна розділити на наступні категорії:

- **Обчислення:** Amazon Elastic Compute Cloud (EC2) - це служба, яка надає віртуальні машини (VM), які можна використовувати для запуску будь-якого типу програмного забезпечення. Amazon Elastic Container Service (ECS) - це служба, яка використовується для запуску та керування контейнеризованими програмами. Amazon Lambda - це служба, яка використовується для запуску безсерверних функцій.
- **Сховище даних:** Amazon Simple Storage Service (S3) - це служба, яка використовується для зберігання необмеженої кількості даних. Amazon Elastic Block Store (EBS) - це служба, яка використовується для зберігання блоків даних для VM. Amazon Glacier - це служба, яка використовується для зберігання архівних даних.
- **Мережа:** Amazon Virtual Private Cloud (VPC) - це служба, яка використовується для створення власної приватної мережі в AWS. Amazon Route 53 - це служба DNS, яка використовується для маршрутизації трафіку до ваших ресурсів AWS. Amazon CloudFront - це служба CDN (Content Delivery Network), яка використовується для доставки контенту користувачам по всьому світу.
- **Бази даних:** Amazon Relational Database Service (RDS) - це служба, яка використовується для запуску керованих реляційних баз даних, таких як MySQL, PostgreSQL та Oracle. Amazon DynamoDB - це служба, яка використовується для запуску нереляційних баз даних NoSQL. Amazon Aurora - це служба, яка використовується для запуску сумісних з PostgreSQL баз даних.

- **Аналітика:** Amazon Redshift - це служба, яка використовується для запуску керованих складів даних для аналітики. Amazon QuickSight - це служба візуалізації даних, яка використовується для створення інтерактивних панелей інструментів та звітів.
- **Машинне навчання та штучний інтелект:** Amazon SageMaker - це платформа машинного навчання, яка використовується для створення, тренування та розгортання моделей машинного навчання. Amazon Rekognition - це служба, яка використовується для аналізу зображень та відео. Amazon Polly - це служба, яка використовується для перетворення тексту на мову.

Рисунок 2.1 – Перелік сервісів AWS

2.1.2 Amazon EC2

Amazon Elastic Compute Cloud (EC2) - це служба хмарних обчислень, що пропонує віртуальні машини (VM) на вимогу. EC2 можна використовувати для запуску будь-якого типу програмного забезпечення, включаючи веб-сервери, бази даних, програми для розробки програмного забезпечення та багато іншого. EC2 пропонує широкий спектр типів VM, які можна налаштувати відповідно до ваших потреб.

Взаємодіяти зі службою можливо за допомогою веб-інтерфейсу, інтерфейсу командного рядка, а також програмно за допомогою API. Для запуску екземплярів використовуються попередньо сконфігуровані образи - "образи машин Amazon" (Amazon Machine Image, AMI), що скорочує час завантаження віртуальної машини.

Служба організована згідно з принципами обчислювальної еластичності, зокрема, передплатникам доступне створення, запуск, зупинка і видалення екземплярів за запитом у міру необхідності з посекундною оплатою.

Amazon EC2 дає змогу розміщувати віртуальні машини в декількох місцях розташування. Місця розташування Amazon EC2 складаються з регіонів і зон доступності. Зони доступності - це окремі місця розташування, спроектовані так, щоб бути захищеними від збоїв в інших зонах доступності та забезпечувати недороге мережеве підключення з низькими затримками до інших зон доступності, розташованих у тому ж регіоні.

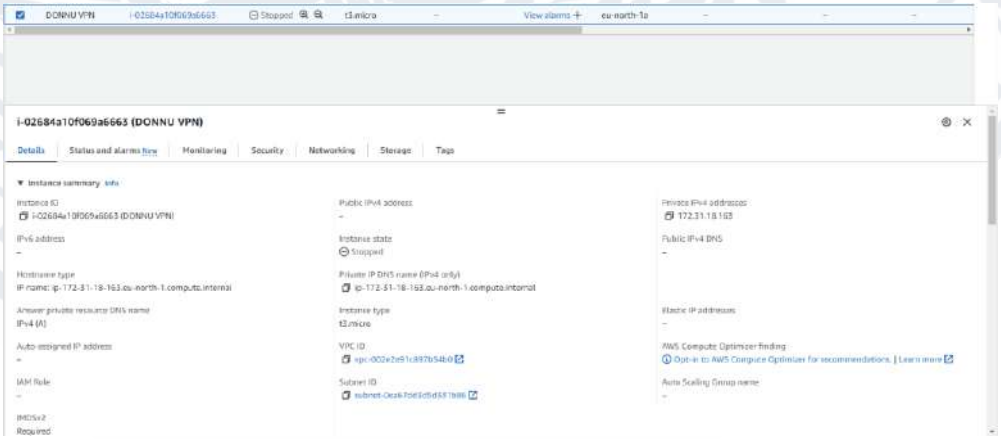

Рисунок 2.2 – Панель керування віртуальними машинами EC2.

2.1.3 OpenVPN

OpenVPN — це протокол віртуальної приватної мережі (VPN), який відомий своєю надійною безпекою та високим рівнем налаштувань. Його подвійна природа як протоколу та програмного забезпечення робить його універсальним інструментом в галузі інтернет-безпеки. Дозволяє користувачам створювати та керувати серверами або мережами VPN.

Основні технічні характеристики OpenVPN [21]:

1. **Безпека:** OpenVPN використовує OpenSSL, відому відкриту бібліотеку, для забезпечення надійної безпеки даних під час передачі. OpenSSL підтримує протоколи Secure Sockets Layer (SSL) та Transport Layer Security (TLS), надаючи OpenVPN надійний фреймворк для шифрування та дешифрування даних. OpenVPN використовує потужні шифри з 256-бітним ключем для забезпечення нерозбірливості даних після шифрування.
2. **Режими Передачі Даних:**
    - User Datagram Protocol (UDP) для швидкої, але менш надійної передачі.
    - Transmission Control Protocol (TCP) для повільної, але більш надійної доставки даних.
3. **Додаткові Можливості:**
    - Підтримка Підмереж Internet Protocol (IP).
    - Віртуальні Еталонні Адаптери.
    - Адаптивне Стиснення Посилань.
    - Формування Трафіку.
    - Можливість Безпечного Проходження Через Брандмауери.

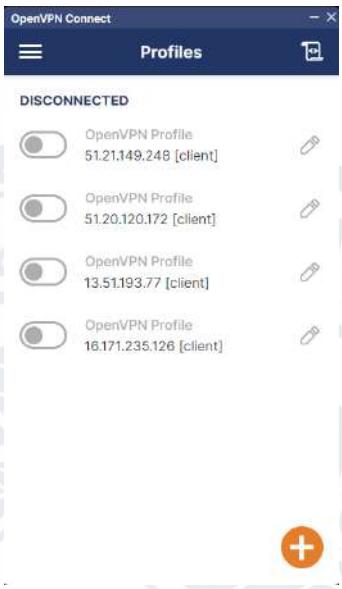

Рисунок 2.3 – Основне вікно додатку OpenVPN Connection.

2.2 Розгортання OpenVPN Server на AWS EC2

Перед початком самого процесу розгортання VPN серверу на основі EC2, потрібно зареєсттруватись на самій платформі AWS та верифікувати акаунт.

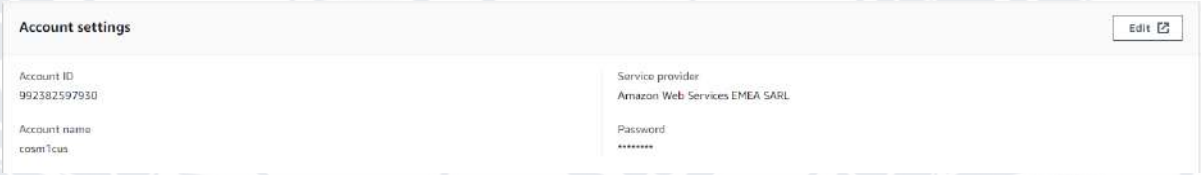

Рисунок 2.4 – Інформація про акаунт на платформі AWS.

Після чого потрібно перевірити зарахування доступу до сервісів та в разі успішного отримання фунціоналу перейти до основних процесів.

Для того щоб створити віртуальну машину потрібно запустити сторінку EC2.

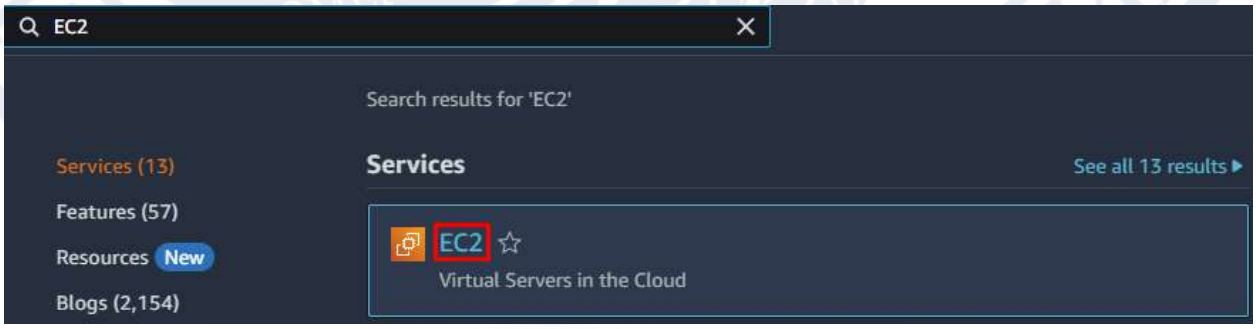

Рисунок 2.5 – Пошук EC2 сервісу.

Головна сторінка EC2 містить інформацію про доступність загальних серверів та регіонів. В разі якщо вже є створені віртуальні машини, то також можна переглянути аналітику використання. Окрім цього, тут можна і створити новий «інстанс», що дозволить нам налаштувати нову віртуальну машину.

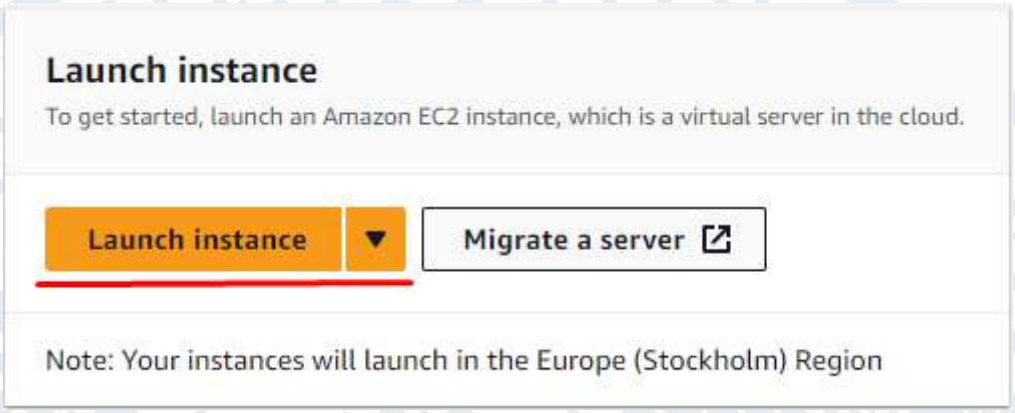

Рисунок 2.6 – Створення віртуальної машини.

Після того як було натиснуто «Launce Instance» відкривається нова сторінка з налаштуваннями, де детально можна обрати характеристики та налаштувати політику мереж.

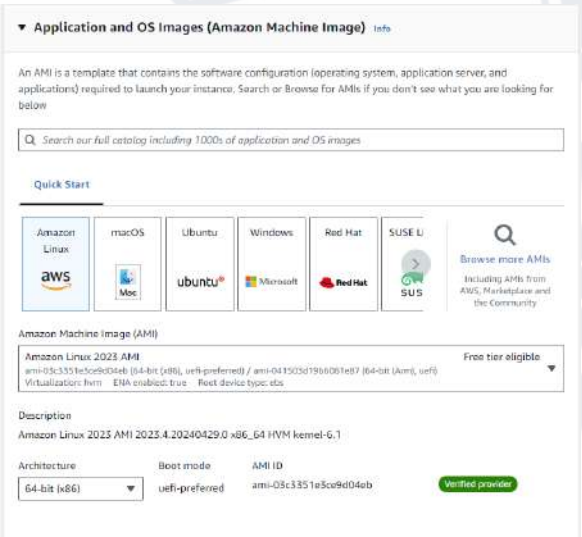

Рисунок 2.7 – Вибір ОС.

Щодо вибору операційної системи, то оптимальним варіантом для розробки VPN серівсу є Amazon Linux, адже він одразу містить налаштовані репозиторії для завантажування додатків, що допоможе майже одразу перейти до етапу налаштування віртуальної приватної мережі.

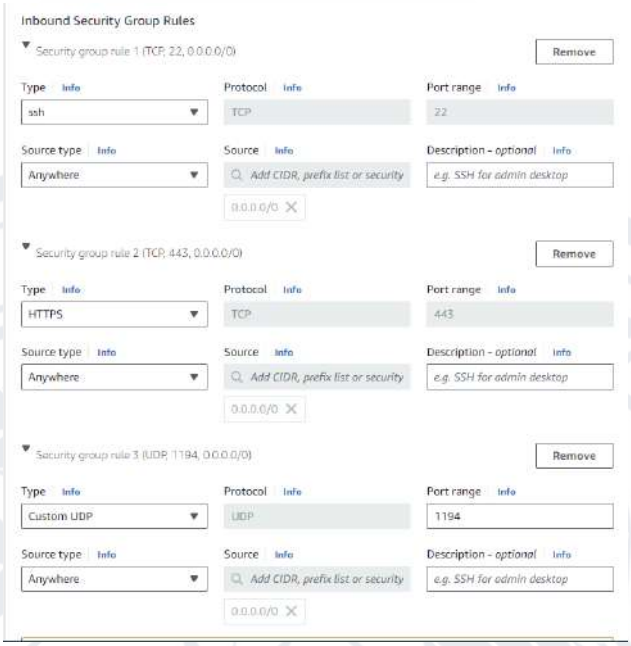

Рисунок 2.8 – Налаштування мережі.

Під час налаштування мережі було конфігуровано: два стандартних порти – 22 (SSH) який використовується для віддаленного підключення до серверу та 443 (HTTPS), що дозволяє відкрити загальний доступ до веб-сайтів, запущених на ньому. Окрім цього було додано порт 1194 (UDP), який є важливим під час запуску OpenVPN мережі.

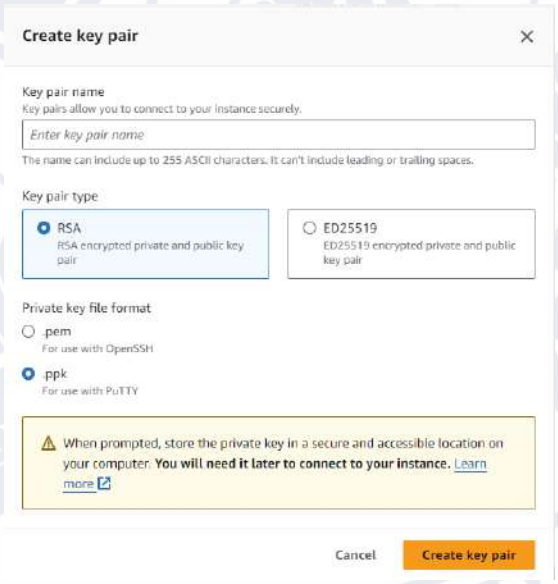

Рисунок 2.9 – Налаштування ключів доступу до віртуального сервера.

Для більшої захищеності серверу також було обрано використання ключів, щоб отримати доступ до нього. Вони були створенні RSA шифруванням, для більшої надійності та стійкості. Стосовно формату, то ключі були збережені у форматі .ppk, що дозволяє швидко під’єднуватись до SSH за допомогою PuTTy,

клієнтську програму для роботи з мережевими протоколами Telnet, SSH, SCP, SFTP, для під'єднання через COM-порт і ZModem, утиліта для генерації RSA, DSA, ECDSA, Ed25519 цифрових SSH-ключів [8].

Після чого були проведені невеликі налаштування: імені віртуальної машини, використання фізичної пам’яті тощо.

Після того, як налаштування були успішно встановлені можна під’єднуватись до серверу за допомогою SSH протокола та ключа згенерованого раніше, використовуючи утиліту PuTTy.

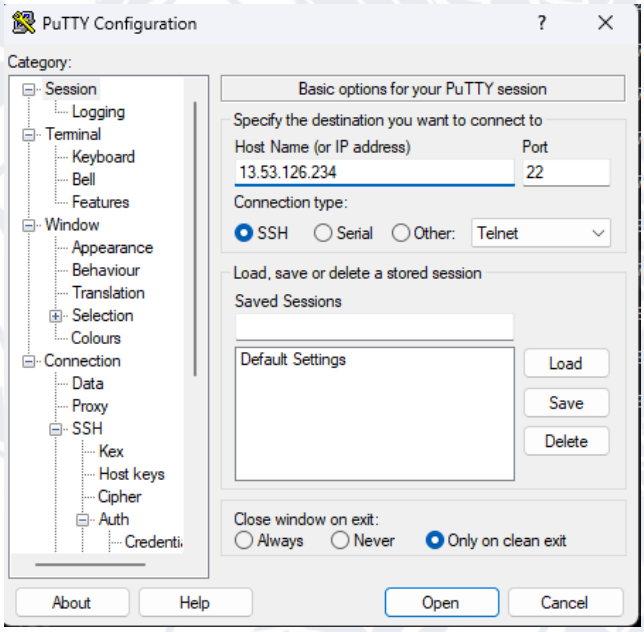

Рисунок 2.10 – Встановлення з’єднання з сервером.

Одразу після старту з’явиться діалогове вікно, в якому потрібно ввести ім’я користувача (По стандарту – ec2-user) і якщо все було зроблено вірно, то система успішно підтвердить вхід.

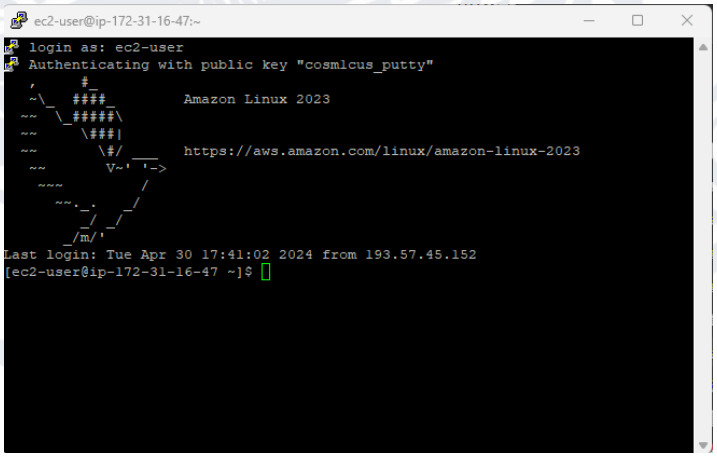

Рисунок 2.11 – Базове вікно ОС Amazon Linux.

Для налаштування OpenVPN, потрібно його завантажити з репозиторію за допомогою команди [9]:

```
sudo yum install -y openvpn easy-rsa
```

Після чого почнеться процес встановлення. Також одразу було завантажено easy-rsa – утиліта для налаштування сертифікатів та ключів підключення до самого VPN.

Наступним кроком буде створення нової директорії «easy-rsa» за розташуванням OpenVPN, для того щоб мати швидкий доступ до всіх ключів, сертифікатів та .config файлів. Для цього використовуються такі команди:

```
sudo mkdir /etc/openvpn/easy-rsa
cd /etc/openvpn/easy-rsa
sudo cp -Rv /usr/share/easy-rsa/3.0.8/ /etc/openvpn/easy-rsa
```

Після створення нової директорії потрібно створити новий каталог «PKI» де будуть зберігатись всі важливі сертифікати та ключі, окрім цього потрібно ініціалізувати CA (certificate authority) та задати PEM кодовий пароль та назву.

```
sudo ./easyrsa init-pki
sudo ./easyrsa build-ca
```

Наступним кроком буде генерація ключа Діффі-Геллмана, який забезпечує секретність вхідного та вихідного трафіку. В кінці операції буде запропоновано також задати кодовий пароль та назву.

```
sudo ./easyrsa gen-dh
```

Також, для зручності, буде прибрана функція вводу паролю під час підключення:

```
sudo ./easyrsa gen-req server nopass
sudo ./easyrsa sign-req server server
./easyrsa gen-req client nopass
sudo ./easyrsa sign-req client client
```

Після генерації та ініціалізації ключів за допомогою «easy-rsa» потрібно створити TLS ключ, за допомогою якого навіть зловмисник з доступом не зможе порушити цілісність та конфіденційність фалів.

```
cd /etc/openvpn
openvpn --genkey --secret pfs.key
```

Далі можна перейти до створення та налаштування .config файлу:

```
cd /etc/openvpn
```

```
sudo nano server.conf
```

Після цих команд відкривається текстовий редактор, в якому нам потрібно оголосити згенеровані ключі та налаштувати мережеве з'єднання з клієнтом [8-9].

```
port 1194
proto udp
dev tun
ca /etc/openvpn/easy-rsa/pki/ca.crt
cert /etc/openvpn/easy-rsa/pki/issued/server.crt
key /etc/openvpn/easy-rsa/pki/private/server.key
dh /etc/openvpn/easy-rsa/pki/dh.pem
cipher AES-256-CBC
auth SHA512
server 10.8.0.0 255.255.255.0
push "redirect-gateway def1 bypass-dhcp"
push "dhcp-option DNS 8.8.8.8"
push "dhcp-option DNS 8.8.4.4"
ifconfig-pool-persist ipp.txt
keepalive 10 120
comp-lzo
persist-key
persist-tun
status openvpn-status.log
log-append openvpn.log
verb 3
tls-server
tls-auth /etc/openvpn/pfs.key
```

Останнім кроком на серверній стороні буде запуск OpenVPN служби та ввімкнення постійного процесу, для безперервної роботи:

```
systemctl start openvpn@server.service
systemctl enable openvpn@server.service
```

Після серверних налаштувань потрібно зібрати всі згенеровані ключі, та завантажити на пристрій клієнта. Для зручності можна використати WinSCP, що дозволяє за допомогою SSH з'єднання переглядати та копіювати всі каталоги та файли в зручному GUI форматі.

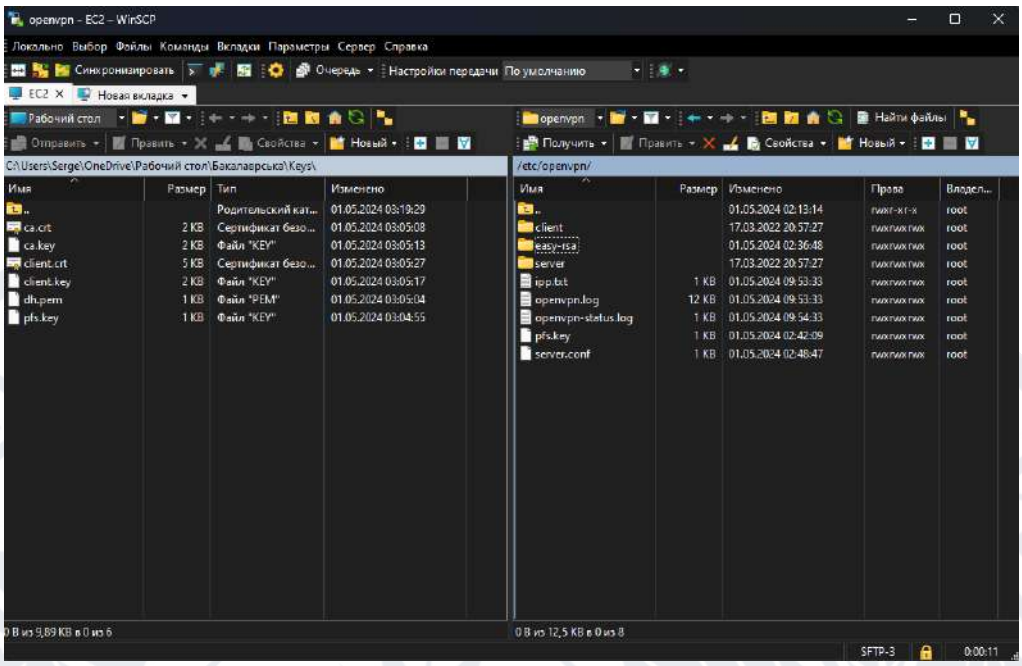

Рисунок 2.12 – Приклад роботи WinSCP.

Після цього потрібно завантажити OpenVPN Client (Connect) та перенести усі завантажені ключі у папку config (./Program Files/OpenVPN/config), окрім цього потрібно створити файл підключення до серверу. Для його налаштування потрібно в цій самій директорії створити файл client.ovpn та додати туди інформацію про наш сервер та шифрування [8-9]:

```
client
dev tun
proto udp
remote YOUR.EC2.INSTANCE.IP 1194
ca ca.crt
cert client.crt
key client.key
tls-version-min 1.2
tls-cipher TLS-ECDHE-RSA-WITH-AES-128-GCM-SHA256:TLS-ECDHE-ECDSA-WITH-AES-128-
GCM-SHA256:TLS-ECDHE-RSA-WITH-AES-256-GCM-SHA384:TLS-DHE-RSA-WITH-AES-256-CBC-
SHA256
cipher AES-256-CBC
auth SHA512
resolv-retry infinite
auth-retry none
nobind
persist-key
persist-tun
ns-cert-type server
```

```
comp-lzo
verb 3
tls-client
tls-auth pfs.key
```

Останнім кроком буде встановлення з’єднання клієнта з сервером, за допомогою файлу client.ovpn.

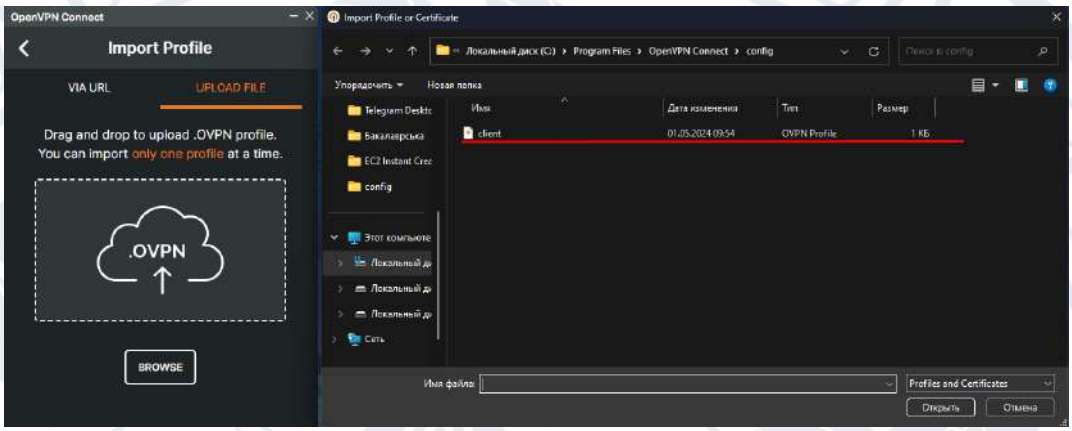

Рисунок 2.13 – Встановлення з’єднання.

2.3 Перевірка роботи OpenVPN під управлінням AWS EC2

Після запуску VPN потрібно перевірити його на працездатність. Зручним сервісом для цього буде SpeedTest by Ookla, який надає вичерпу інформацію про швидкість мережі та пінг. Для перевірки ефективності була створена таблиця з результатами, в якій відображено результати сканування мережі без VPN, Open VPN та стороннього VPN сервісу.

Таблиця 2.1 Порівняння результатів роботи

Home Network		**OpenVPN**		**Other VPN**	
Download (ping)	***Upload (ping)***	***Download (ping)***	***Upload (ping)***	***Download (ping)***	***Upload (ping)***
73.01 Mbps	63.70 Mbps	67.12 Mbps	59.38 Mbps	44.89 Mbps	42.51 Mbps
80ms	405ms	94ms	347ms	113ms	574ms

Згідно з результататами, якщо порівнювати отримані результати з вхідними даними роботи мережі без VPN, можна зробити висновок, що VPN на основі OpenVPN працює стабільніше та швидше за інші загальнодоступні сервіси.

Окрім цього, можна зазначити, що рівень захисту власної віртуальної приватної мережі набагато краще, це обумовлено більшою кількістю етапів захисту власного серверу та відсутністю загроз з точки загального серверу Amazon.

Висновки до розділу 2

В даному розділі було здійснено практичну реалізацію сервісу маскування адресації, спираючись на результати теоретичного дослідження, представленого в розділі 1. Як основа для розгортання сервісу була обрана хмарна платформа Amazon Web Services (AWS), що надає потужні інструменти для розгортання та налаштування серверів. Використання сервісу Amazon EC2 дозволило швидко створити віртуальну машину з необхідними характеристиками та налаштуваннями безпеки.

В якості протоколу VPN було обрано OpenVPN - надійне та гнучке рішення з відкритим кодом, яке забезпечує високий рівень безпеки та конфіденційності завдяки використанню сучасних криптографічних алгоритмів та можливості індивідуальних налаштувань.

В результаті проведеної роботи було успішно розгорнуто OpenVPN сервер на віртуальній машині AWS EC2, налаштовано захищене з'єднання з використанням TLS шифрування, згенеровано необхідні сертифікати та ключі для автентифікації клієнтів. Проведене тестування розробленого сервісу підтвердило його високу ефективність та стабільність роботи, а також конкурентні, в порівнянні з деякими комерційними VPN сервісами, показники швидкості та надійності.

Таким чином, в ході виконання розділу 2 було створено функціональний сервіс маскування адресації, що може бути використаний для підвищення рівня безпеки та приватності користувачів в кіберпросторі.

ВИСНОВКИ

В результаті проведеної роботи було досягнуто поставленої мети: розроблено та реалізовано сервіс маскування адресації на базі хмарної платформи Amazon Web Services (AWS) з використанням протоколу OpenVPN.

В процесі виконання роботи було проведено детальне дослідження теоретичних основ маскування в кіберпросторі, розглянуто поняття цифрового сліду та його характеристик, а також проаналізовано основні методики та моделі маскування адресації. Було проведено практичний огляд ключових інструментів, необхідних для реалізації сервісу: AWS, EC2 та OpenVPN.

В рамках практичної частини роботи було здійснено розгортання OpenVPN сервера на віртуальній машині AWS EC2, налаштовано безпечне з'єднання з використанням TLS шифрування та згенеровано необхідні сертифікати та ключі для автентифікації клієнтів.

Тестування розробленого сервісу підтвердило його ефективність та стабільність. Порівняння з іншими VPN сервісами показало, що власний VPN сервер на базі OpenVPN демонструє конкурентні показники швидкості та надійності, забезпечуючи при цьому високий рівень захисту даних завдяки використанню сучасних криптографічних алгоритмів та індивідуальних налаштувань безпеки.

Таким чином, результати роботи підтверджують актуальність та практичну цінність розробки власних VPN сервісів на базі хмарних платформ, що дозволяє користувачам підвищити рівень безпеки та приватності в кіберпросторі.

СПИСОК ВИКОРИСТАНИХ ПОСИЛАНЬ

1. А. В. Жилін, О. М. Шаповал, О. А. Успенський. Технології захисту інформації в інформаційно-телекомунікаційних системах. 2020. С. 165-206.
2. С.Г. Семенов, А.О.Подорожняк, О.І.Баленко, С.Ю.Гавриленко. Захист інформації в комп'ютерних системах та мережах. 2014. С. 111-216.
3. Борисова, К. Є. Актуальність вивчення аспектів віртуального маскування сучасним поліцейським. 2023.
4. Н.О. Щур, О.А. Покотило. Основи Криптології. 2021. С. 76-105.
5. М. В. Грайворонський, О. М. Новіков Безпека інформаційно комунікаційних систем. 2009.
6. Чим відрізняється проксі-сервер від VPN? Які між ними основні відмінності?. NordVPN. URL: https://nordvpn.com/uk/blog/proksi-proty-vpn/ (дата звернення: 24.05.2024).
7. Як вибрати найкращий vpn-протокол. Surfshark. URL: https://surfshark.com/uk/blog/vpn-protocols (дата звернення: 24.05.2024).
8. Роєцький Д. А. КРИПТОГРАФІЯ: ЗАГАЛЬНІ ВИЗНАЧЕННЯ, КЛАСИФІКАЦІЯ. АСИМЕТРИЧНІ ТА СИМЕТРИЧНІ КРИПТОАЛГОРИТМИ, ЇХ ПОРІВНЯННЯ. URL: https://royetskyyda14.wordpress.com/2014/12/15/криптографія-загальні-визначення-кл/ (дата звернення: 01.06.2024).
9. How to make your own free VPN with Amazon Web Services. Comparitech. URL: https://www.comparitech.com/blog/vpn-privacy/how-to-make-your-own-free-vpn-using-amazon-web-services/ (дата звернення: 24.05.2024).
10. Joe Luo. Building OpenVPN Server on AWS EC2. Medium. URL: https://medium.com/@sonicjoy2002/building-openvpn-server-on-aws-ec2-84b2c0a1d934 (дата звернення: 24.05.2024).
11. Kuwar Kuldeep V. V. A New Approach for the Security of VPN. URL: https://www.semanticscholar.org/paper/A-New-Approach-for-the-Security-of-VPN-Singh-Gupta/7303f5e33a05766c087867e5440af2083f787fbd%20 (дата звернення: 01.06.2024)

12. L2TP. www.wikidata.uk-ua.nina.az. URL: https://www.wikidata.ukua.nina.az/L2TP.html (дата звернення: 01.06.2024).
13. Protocol state fuzzing of an OpenVPN. URL: https://www.semanticscholar.org/paper/Protocol-state-fuzzing-of-an-OpenVPN/806072f27d4ff439624b41449c474ca0a5aa265a (дата звернення: 01.06.2024).
14. What is digital footprint? Malwarebytes. URL: https://www.malwarebytes.com/cybersecurity/basics/digital-footprint (дата звернення: 24.05.2024).
15. Anjana R. Arakerimath, Pramod Kumar Gupta. Digital Footprint: Pros, Cons, and Future. 2015. C. 52-55.
16. Beraud, P., Cruz, A., Hassell, S., & Meadows, S. Using cyber maneuver to improve network resiliency. 2011
17. Chee Keong NG, Lei Pan, Dr. Yang Xiang. Honeypot Frameworks and Their Applications: A New Framework. In Springer Briefs on Cyber Security Systems and Networks. Springer, Singapore. 2018.
18. Wang, Y., Guo, Y., Zhang, L. et al. SWIM: An Effective Method to Perceive Cyberspace Situation from Honeynet. Arabian Journal for Science and Engineering. 2018.
19. Graham Bartlett, Amjad Inamdar. IKEv2 IPsec Virtual Private Networks: Understanding and Deploying IKEv2, IPsec VPNs, and FlexVPN in Cisco IOS. 2016.
20. Vacca, John R. Network and system security. 2014.
21. Keith E. Strassberg, Richard G. Gondek, Gary Rollie. Firewalls: The Complete Reference. 2004.
22. Muhammad Iqbal, Imam Riadi. Analysis of Security Virtual Private Network (VPN) Using OpenVPN. Yogyakarta 2019.
23. Jason Liu, Yue Li, Nathanael Van Vorst, Scott Mann, Keith Hellman. A real-time network simulation infrastructure. based on OpenVPN. Florida International University 2008

24. A. Skendzic, B. Kovacic. Open source system OpenVPN in a function of Virtual Private Network. Baia Mare 2016.
25. Steven Mackey, Ivan Mihov, Alex Nosenko, Francisco Vega, Yuan Cheng. A Performance Comparison of WireGuard and OpenVPN. 2020.
26. Jan Just Keijster. OpenVPN Cookbook. Packt Publishing 2017.