МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему:

«Інформаційна технологія проєктування сучасних віртуальних приватних мереж»

Завідувач випускаючої кафедри — **Довбиш А.С.**

Керівник роботи — **Великодний Д.В.**

Студентка групи ІН.мз-01с — **Козолуп І.М.**

СУМИ 2021

Сумський державний університет

(назва вузу)

Факультет *ЦЗДВФН* Кафедра *Комп'ютерних наук*

Спеціальність *«Інформатика»*

Затверджую:
зав.кафедрою _____
_____
"_____" _____ 20___р.

ЗАВДАННЯ

НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТОВІ

Козолуп Ірині Миколаївні

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) *Інформаційна технологія проєктування сучасних віртуальних приватних мереж*

затверджую наказом по інституту від "____" _____ 20 ___ р. № _____

2. Термін здачі студентом закінченого проекту (роботи) _____

3. Вхідні данні до проекту (роботи) _____
_____
_____
_____

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) *1) Аналіз проблеми. Постановка задачі дослідження. 2) Інформаційний огляд. 3) Вибір методу та протоколу вирішення завдання. 4) Розробка програмного забезпечення для конфігурації VPN мережі.*

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____
_____
_____
_____
_____

6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____

Керівник _____

(підпис)

Завдання прийняв до виконання _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів дипломного проекту (роботи)	Термін виконання проекту (роботи)	Примітка
1.	*Аналіз проблеми. Постановка задачі дослідження.*		
2.	*Інформаційний огляд.*		
3.	*Вибір методу та протоколу вирішення завдання.*		
4.	*Розробка програмного забезпечення для конфігурації VPN мережі.*		
5.	*Оформлення кваліфікаційної магістерської роботи*		

Студент – дипломник _____

(підпис)

Керівник проекту _____

(підпис)

РЕФЕРАТ

Записка: 62 стор., 32 рис., 1 додаток, 32 джерела.

Об'єкт дослідження – сучасні VPN мережі.

Мета роботи – вивчення структури VPN мереж, дослідження основних методів їх побудови, визначення ключових протоколів, які використовуються при побудові такого роду мереж та розробка інформаційної системи для конфігурації VPN мереж.

Методи дослідження – метод порівняльного аналізу протоколів та методи проектування VPN мереж.

Результати – змодельовано VPN мережу за допомогою протоколу IPSec. В роботі були проаналізовані основні команди, які використовуються при її конфігурації. На основі цього аналізу було розроблено веб-інтерфейс, який генерує код налаштувань для заданих адрес роутерів. Згенерований код можна використовувати при роботі з мереживим обладнанням. Розроблена програма була написана з використанням мови програмування JavaScript та була успішно протестована.

VPN МЕРЕЖА, СКЛАДОВІ VPN МЕРЕЖ, МОДЕЛЮВАННЯ, СИМУЛЯТОР, VPN ТУНЕЛЬ, ШИФРУВАННЯ, АУТЕНТИФІКАЦІЯ, ПРОТОКОЛ, ПРОТОКОЛ IPSec, ВІРТУАЛЬНА ПРИВАТНА МЕРЕЖА.

ЗМІСТ

ВСТУП

VPN мережі завжди були корисними, та на сьогоднішній день вони стають не лише корисними а і необхідними. Віртуальні приватні мережі вже досить давно використовуються для забезпечення конфіденційності та приватності в Інтернеті.

Дана робота присвячена питанню побудови та налаштуванню сучасних VPN мереж. Актуальність роботи зумовлена необхідністю вивчення методів побудови VPN мереж, адже сьогодні все більш необхідним стає можливість віддалено підключитися до необхідних ресурсів. Корпоративні VPN мережі повинні надавати співробітникам та зацікавленим сторонам безпечний віддалений доступ до даних компанії.

VPN є одним із найважливіших інструментів захисту конфіденційної та важливої інформації організацій в мережі. Якщо співробітники працюють віддалено, їхнє з'єднання з внутрішньою мережею компанії має відбуватися через Інтернет, який є загальнодоступним. Така передача трафіку може піддатися атакам ззовні. Шифрування цього трафіку за допомогою корпоративної мережі VPN унеможливлює його розкриття.

Крім використання VPN у корпоративних мережах необхідно сказати про його важливість в особистих цілях. Він допомагає захистити особисті дані, мати безпечний доступ до власних файлів та безпечно працювати в мережі Інтернет.

Конфігурація VPN мереж займає досить багато часу, адже потрібно здійснити достатньо велику кількість налаштувань, не пропустити жодної команди, так як безпека роботи залежить саме від правильного налаштування всіх мереживих пристроїв. Рішенням цієї проблеми є програми чи веб-інтерфейси, які допомагають провести всі налаштування швидше.

Метою цієї роботи є саме розробка веб-інтерфейсу, де необхідно ввести параметри мережі і отримати готовий код налаштувань, який може бути використаний для реальних пристроїв при конфігурації VPN мережі.

1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

1.1 Поняття та цілі VPN мережі

VPN (англ. Virtual Private Network – віртуальна приватна мережа) – це безпечна мережа, яку зазвичай створюють на базі іншої мережі, наприклад такої як Інтернет. Мета VPN технології – забезпечити безпечне та надійне з'єднання між комп'ютерними мережами через існуючу загальнодоступну мережу. Хоча комунікації здійснюються через публічні мережі, де немає особливого захисту, у VPN мережах застосовують закриті канали для обміну інформацією [1].

VPN технологія надає можливість об’єднати філії, офіси, департаменти, які можуть бути розташовані на інших куточках світу однієї організації в одну мережу, використовуючи при цьому захищені канали.

Наразі зростає попит на більш економічні способи передачі даних безпечно через «небезпечні» загальнодоступні мережі, такі як Інтернет. Що і зробило VPN мережі дуже популярними сьогодні через число переваг, які вони пропонують.

Однією із важливих властивостей технології віртуальних приватних мереж є її масштабування. Провайдери мережевих послуг постійно удосконалюють та модернізують свої технології і відповідно VPN також росте, для того щоб відповідати всім критеріям. Говорячи про VPN мережі в компаніях, важливо сказати, що пристрої можуть працювати як клієнт, або як сервер VPN, адже такого роду мережі не залежать від платформ та операційних систем. Також такі віртуальні мережі мають можливість управляти службами, які в них розміщені. Важливим моментом є те, що вони при необхідності надають можливість будувати тунелі чи наскрізну передачу з шифруванням. Тунелі можуть бути збудовані до інших вузлів, як приклад, тунель від головного офісу до його філій [2, 212].

Працюючи в Інтернеті, роблячи покупки в інтернет-магазинах, перевіряючи свої профілі в соціальних мережах ми залишаємо свій цифровий слід.

Користуючись незахищеними мережами ми піддаємося небезпеці, адже в цих же мережах зловмисники можуть захопити приватну інформацію, скористатися чужими обліковими записами, щоб видавати себе за інших людей, відстежувати трафік у власних цілях. Також у відкритих мережах на сайтах при переході на рекламні банери шахраї можуть почати відстежувати трафік інших користувачів. Використовуючи VPN технологію ми зберігаємо анонімність в мережі. Приватні віртуальні мережі допомагають приховати, а саме зашифрувати всі наші дані та дії у відкритих мережах. Коли ми працюємо в мережі через VPN то дізнатися справжнє джерело підключення неможливо, адже замість нього видно лише один з маршрутизаторів, який використовує VPN мережа. Як видно на рис. 1.1 VPN є перешкодою для зловмисників, які хочуть заволодіти конфіденційною інформацією.

Рисунок 1.1 – Приклад роботи VPN мережі [22]

Використовуючи технологію VPN, підключення до Інтернету буде зашифровано, так що ніхто не побачить дані, які ми завантажуємо або відправляємо. Існує три основних види шифрування: хешування, симетричне і асиметричне шифрування. У кожного виду свої переваги і недоліки, але всі вони шифрують дані так, що в чужих руках вони будуть марними [3].

Додатковий рівень захисту, який є у більшості служб VPN – їх власна система DNS. DNS-система доменних імен – це телефонна книга Інтернету, в якій текстові URL-адреси ототожнені з відповідними IP-адресами. Система DNS дозволяє замість довгої послідовності цифр вводити назву сайту. Кіберзлочинці можуть спостерігати за запитами DNS, щоб відстежувати дії користувачів в інтернеті, але система DNS в службах VPN розроблена так, щоб за допомогою додаткового шифрування перешкодити їм.

При роботі з VPN всі дані залишаються у безпеці, адже зазвичай для маршрутизації трафіку використовується сервер, який може знаходитися у будь-якій точці світу. Такий прийом допомагає захистити інформацію від відстеження. Розвиток засобів побудови VPN спрямовується на використання мережі на основі маршрутизаторів. У цьому випадку створюється висока продуктивність, за рахунок інтеграції VPN та маршрутизації на одному пристрої [4].

VPN технологія використовується не лише для особистих цілей, а також користується великою популярністю серед великих корпорацій для того, щоб надати доступ до корпоративної мережі своїм працівникам, які працюють віддалено. При цьому користувачі мають мати права для того, щоб користуватися додатком VPN.

Вдало спроектована VPN мережа може принести організації багато переваг: розширити географію доступу співробітників до інфраструктури організації, підвищити безпеку передачі інформації, зменшити експлуатаційні витрати згідно з традиційними глобальними мережами, скоротити час передачі інформації та зменшити командні витрати, підвищити продуктивність праці, спростити топологію мережі, збільшити мобільність користувачів і дати їм більш гнучкий графік роботи.

Існує два основних види мереж VPN. Перший вид це шлюз захищеного віддаленого доступу до VPN, який зображений на рис. 1.2. Він дозволяє

користувачам підключитися до іншої мережі (до Інтернету або внутрішньої системи своєї компанії) по приватному зашифрованому тунелю.

Рисунок 1.2 – Вид мережі VPN. Шлюз захищеного віддаленого доступу до VPN [5]

Другий вид – VPN типу «мережа-мережа», який ще називають VPN між маршрутизаторами і який зображений на рис. 1.3. Цей вид мережі VPN в основному використовується в корпоративному середовищі, особливо якщо у підприємства є філіали з різним розташуванням. VPN типу «мережа-мережа» використовується для створення закритої внутрішньої мережі, де всі офіси можуть підключатися один до одного. Ця технологія відома як Інтранет.

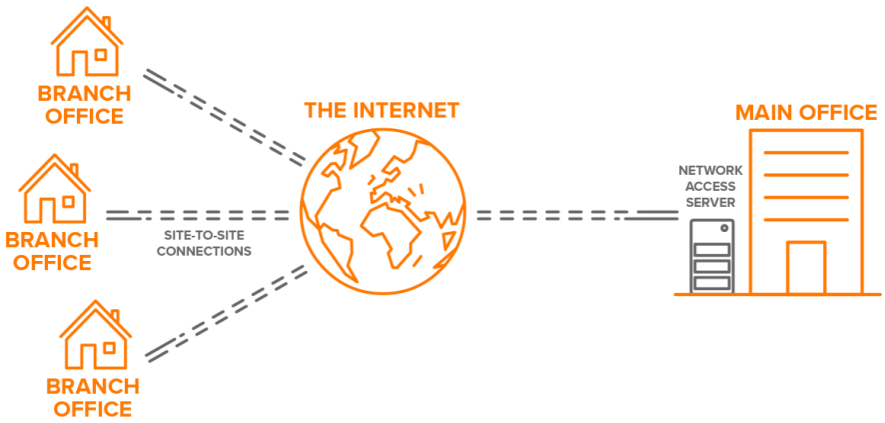

Рисунок 1.3 – Вид мережі VPN. «Мережа-мережа» [5]

VPN сьогодні є важливим інструментом для безпечної та комфортної роботи в Інтернеті. Віртуальні приватні мережі дозволяють віддаленому користувачу, який пройшов аутентифікацію, використовувати корпоративну мережу на рівні з клієнтами центральної корпоративної мережі.

1.2 Класифікація VPN мереж та їх особливості

Існує досить багато класифікацій віртуальних приватних мереж за різними ознаками. Здійснити класифікацію VPN мережі можна за п’ятьма основними параметрами, які представлені на рис.1.4.

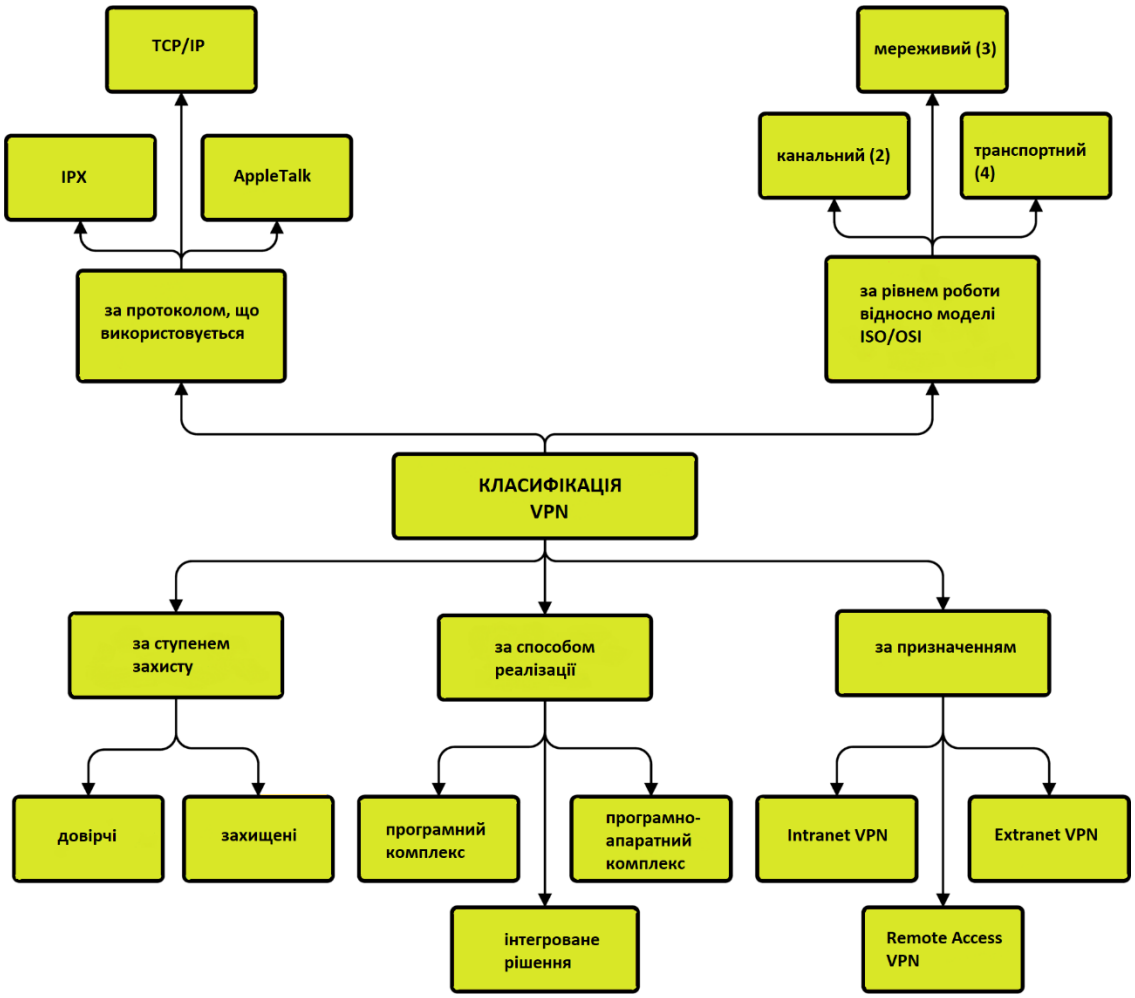

Рисунок1.4 – Класифікація VPN мереж [6]

1. За протоколом, що використовується:

1.1. IPX (англ. Internetwork packet exchange - міжмережевий обмін пакетами) – протокол моделі OSI, який знаходиться на мережевому рівні. Він працює з датаграмами [7].

1.2. AppleTalk – стек протоколів, який був розроблений для встановлення зв'язку між комп'ютерами Macintosh. Як і TCP/IP, AppleTalk являє собою набір протоколів, кожен з яких відповідає за роботу певного рівня моделі ISO/OSI. На відміну від протоколів TCP/IP і IPX/SPX, пакет протоколів AppleTalk використовує власну реалізацію фізичного і канального рівнів, а не протоколи моделі ISO / OSI.

1.3. TCP/IP – це набір комунікаційних протоколів, що використовуються для об’єднання пристроїв у мережі. TCP/IP протокол складається з двох важливих протоколів: Transmission Control Protocol та Internet Protocol. TCP та IP це окремі протоколи, які працюють разом для того, щоб забезпечити, що інформація буде доставлена в необхідне місце призначення в рамках певної мережі. IP протокол отримує та визначає IP адреси пристроїв яким має бути відправлена інформація. TCP протокол відповідальний за транспортування та маршрутизацію в рамках мережі. Також даний протокол можна описати як модель передачі даних в якій інформація проходить через чотири рівні [8, 282].

Серед цих протоколів найбільш популярним є протокол TCP/IP і на сьогоднішній день спостерігається загальна тенденція до переходу на даний протокол і більшість VPN рішень підтримують саме протокол TCP/IP.

2. За рівнем роботи відносно моделі ISO/OSI:

Відповідно моделі OSI, VPN може бути встановлена на рівні каналу передачі даних, на мережевому або навіть більш високому рівні. Сьогодні зазвичай при

побудові VPN мереж використовують протоколи таких рівнів як канальний, мереживний, транспортний.

3. За призначенням:

3.1. Intranet VPN – об’єднує декілька віддалених офісів однієї компанії, які передають інформацію через відкриті канали в одну захищену мережу, як зображено на рис. 1.5. Набір усіх внутрішніх сайтів компанії, які працюють таким чином часто називають Інтранетом компанії. Інтранет VPN забезпечує той самий рівень підключення та надійності як повністю приватні мережі. Побудова Інтранет VPN за допомогою Інтернету є найбільш економічно ефективним засобом впровадження технології VPN. Intranet VPN є повністю приватними мережами, доступ до яких мають лише авторизовані користувачі в цій мережі;

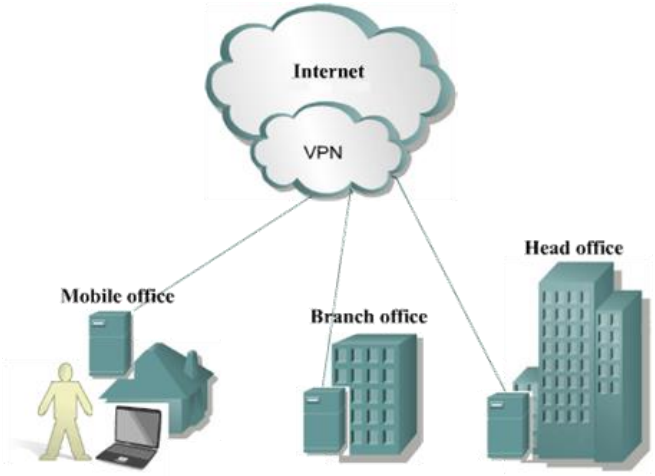

Рисунок 1.5 – Схема підключення Intranet VPN [6]

1.1. Extranet VPN – застосовується до мереж, до яких можуть підключатися користувачі поза офісом (наприклад замовники чи клієнти), які не володіють конфіденційною інформацією і в такому випадку необхідно забезпечувати високий рівень захисту. За допомогою Extranet VPN можна співпрацювати та передавати інформацію в безпечному спільному доступі, не надаючи доступу третім особам до внутрішньої мережі VPN [9];

1.2. Remote Access VPN – дозволяє користувачам, співробітникам, які працюють віддалено підключатися до корпоративної мережі, ресурсів, використовуючи власний комп’ютер чи смартфон. Таким чином створюється захищений канал між головним офісом та співробітником компанії. Remote Access VPN – це не просто спосіб для співробітників отримати віддалений доступ до приватної мережі компанії. Зараз люди використовують VPN з віддаленим доступом, які пропонують ряд служб VPN, щоб захистити та анонімізувати свою онлайн-активність і трафік [10, 892]. Схему такого підключення можна побачити на рис. 1.6.

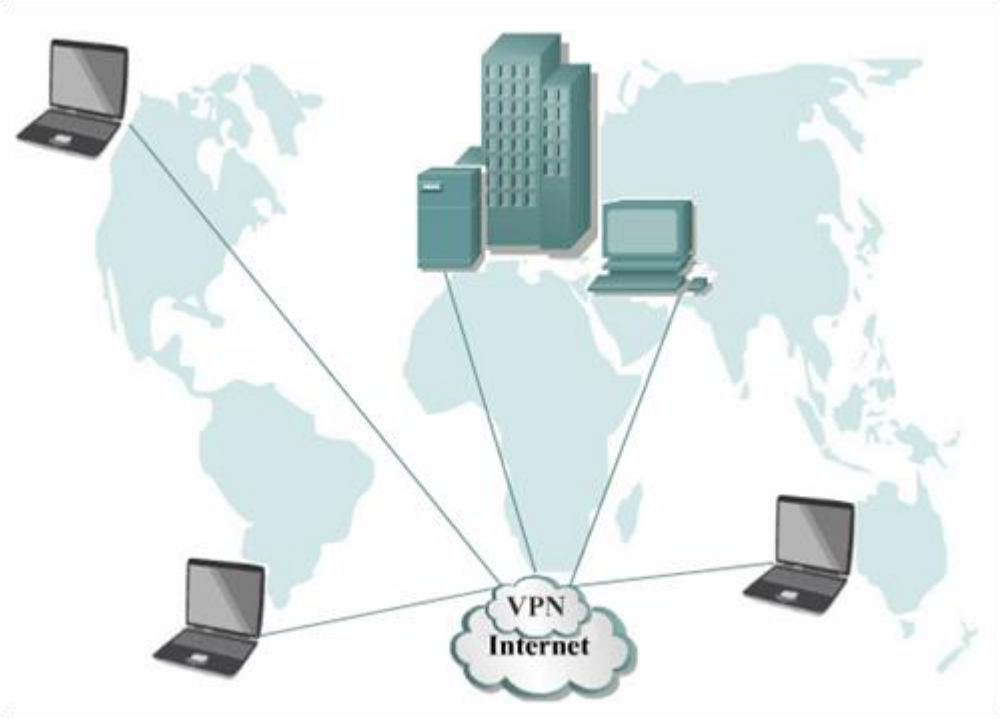

Рисунок 1.6 – Схема підключення Remote Access VPN [6]

2. За способом реалізації:

2.1. У вигляді програмного комплексу. Зазвичай використовується персональний комп'ютер, який має спеціальне програмне забезпечення, що забезпечує функціональність VPN;

2.2. У вигляді програмно-апаратного комплексу. Конфігурація VPN мережі може бути створена з використанням програмно-апаратних засобів. Такий спосіб побудови надає більше можливостей для забезпечення більшої безпеки та продуктивності;

2.3. У вигляді інтегрованого рішення. Технології VPN можуть також вирішувати питання фільтрації мережевого трафіку, забезпечення якості обслуговування та організації мережевого екрану.

3. За ступенем захисту:

3.1. Довірчі. Такого роду VPN мережі можна використовувати тоді, коли передача даних є безпечною, немає ніяких загроз для їх викрадення чи змінення. В таких випадках зазвичай застосовуються наступні VPN технології: MPLS та L2TP разом з IPSec;

3.2. Захищені. Такі VPN мережі є найбільш розповсюдженими, адже вважається, що вони забезпечують високий рівень захисту та є особливо надійними. Вони будують безпечну мережу на базі Інтернету, який вважається не досить таки захищеним. До захищених VPN мереж відносять IPSec, OpenVPN і PPTP.

Існує декілька варіантів класифікації VPN мереж, адже з кожним роком дана технологія удосконалюється і використовувані протоколи та методи побудови змінюються на інші, більш сучасні та надійні.

1.3 Основні складові VPN мереж

До основних складових, які забезпечують надійну роботу VPN мережі відносять:

- тунелювання;

- шифрування;

- аутентифікація.

Спільна взаємодія та впровадження всіх трьох складових забезпечують надійний захист для інформації, яка передається через недостатньо захищені канали зв'язку [11, 94].

Тунелювання є досить важливою складовою, адже завдяки йому середовище передачі даних між двома вузлами є прихованим, що і забезпечує захищеність інформації. Вся інформація проходить через так званий тунель. Приклад тунелювання зображений на рис. 1.7.

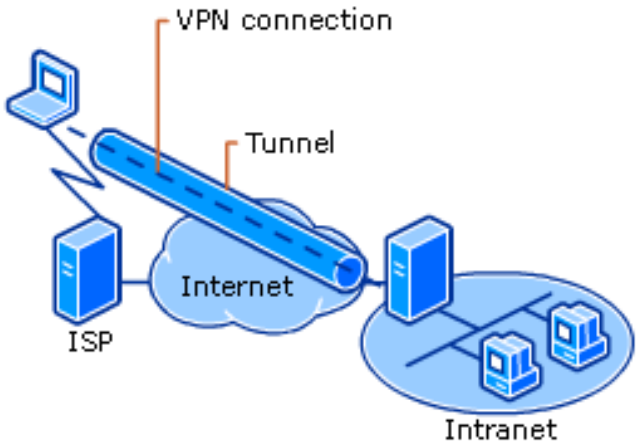

Рисунок 1.7 – Використання тунелювання у VPN мережі [12]

Пакети, які передаються від одного вузла до іншого проходять через тунель, при цьому пакети доставляються без жодних змін. Для того щоб захистити пакети всередині тунелю застосовують метод електронного цифрового підпису (ЕЦП). Його особливість в тому, що всі пакети отримують додатковий блок інформації, який є унікальним для пакету та секретного ключа електронного цифрового підпису. Аутентифікація здійснюється завдяки тому, що одержувач знає ключ ЕЦП відправника. Важливим є те, що VPN тунель маскує IP адреси, що додатково забезпечує безпеку в мережі.

Використовуючи тунелювання захист інформації під час її передачі через відкриті канали базується на створенні захищених віртуальних каналів зв'язку, так званих криптозахищених тунелів. Такі тунелі проходять через відкриту мережу і передають криптографічно захищені пакети повідомлень [13, с. 45].

Завдяки тунелюванню існує можливість передавати пакети одного протоколу засобами іншого протоколу. Таким чином вирішується проблема взаємодії декількох різнотипних мереж, де важливо слідкувати за цілісністю даних та відмінністю між протоколами та їх особливостями.

Методи аутентифікації та шифрування використовуються для забезпечення безпеки в мережі VPN. Важливо зазначити, що VPN сервер на який надходять дані від користувачів може бути розташований на іншому кінці світу і при цьому дані проходять через величезну кількість провайдерів. Тому безпеці в мережі VPN необхідно приділити особливу увагу. На рис. 1.7 зображений приклад аутентифікації у VPN мережі.

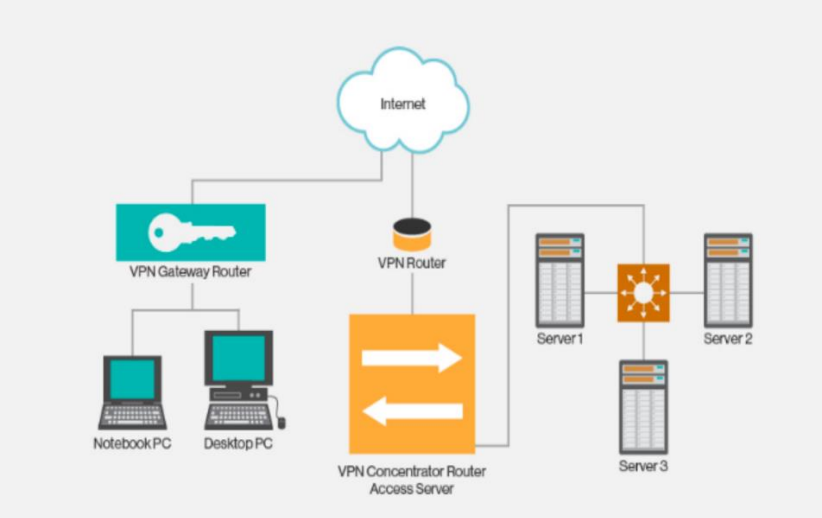

Рисунок 1.7 – Аутентифікація у VPN мережі [14]

Для аутентифікації користувачів за протоколом PPTP (Point-to-Point Tunneling Protocol – тунельний протокол типу точка-точка) може бути задіяний один із протоколів, які зображені на рис.1.8:

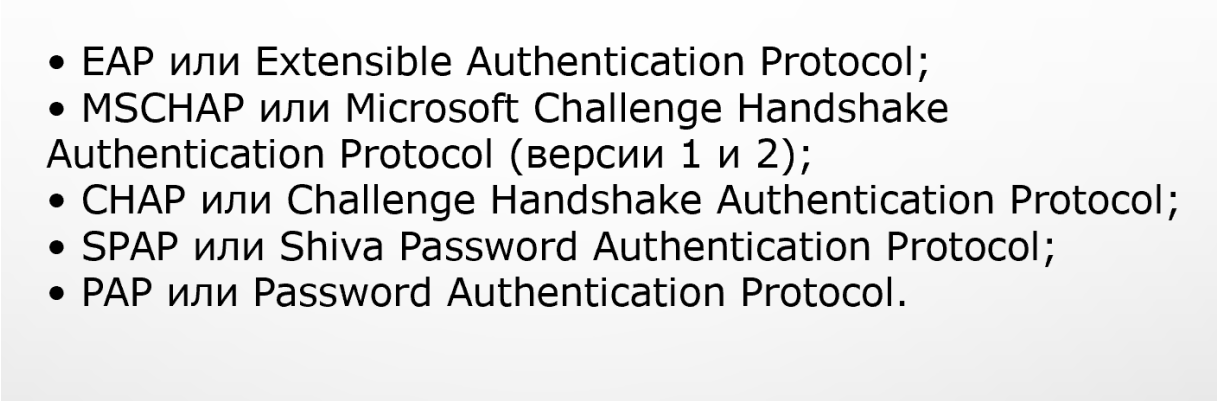

Рисунок 1.8 – Протоколи аутентифікації [15]

Серед цих протоколів найбільш надійними вважаються MSCHAP та EAP, адже вони створюють взаємну аутентифікацію, де серверу та клієнту необхідно ідентифікувати один одного на відміну від інших протоколів, де тільки сервер здійснює аутентифікацію.

Також протокол L2TP поверх IPSec є надійним варіантом, адже здійснюється аутентифікація на рівнях «користувач» та «комп'ютер», а також забезпечується шифрування даних.

Аутентифікація зазвичай створюється за допомогою відкритого тексту чи за допомогою схеми запит / відгук. У випадку якщо використовується відкритий текст, то комп’ютер-клієнт відправляє VPN серверу пароль, який порівнюється з еталоном і на основі цього приймається рішення чи дозволяти доступ чи забороняти.

У випадку схеми запит/ відгук аутентифікація використовує криптографічний протокол, який дозволяє довести, що користувач знає пароль, не розкриваючи сам пароль. Далі протокол обчислює відповідь, застосовуючи криптографічну хеш-функцію до виклику сервера в поєднанні з паролем користувача.

Якщо аутентифікація проходить з використанням протоколів L2TP поверх IPSec то спочатку використовуються локальні сертифікати. Відбувається обмін сертифікатами між клієнтом та сервером. Далі відбувається аутентифікація на рівні користувача. При цьому неважливо який протокол використовується, адже вся сесія залишається зашифрованою [16].

Аутентифікація є частиною трифазного процесу, а саме: ідентифікація, аутентифікація та авторизація. Ці три складові забезпечують безпеку в мережі.

Процес шифрування забезпечує цілісність даних під час їх пересилання в мережі та гарантує, що доступ до них закритий. Два методи шифрування, що наразі використовуються:

- MPPE – протокол шифрування або Microsoft Point-to-Point Encryption;
- EAP-TLS – що здатний автоматично обирати довжину ключа шифрування.

Приклад шифрування можна побачити на рис. 1.9.

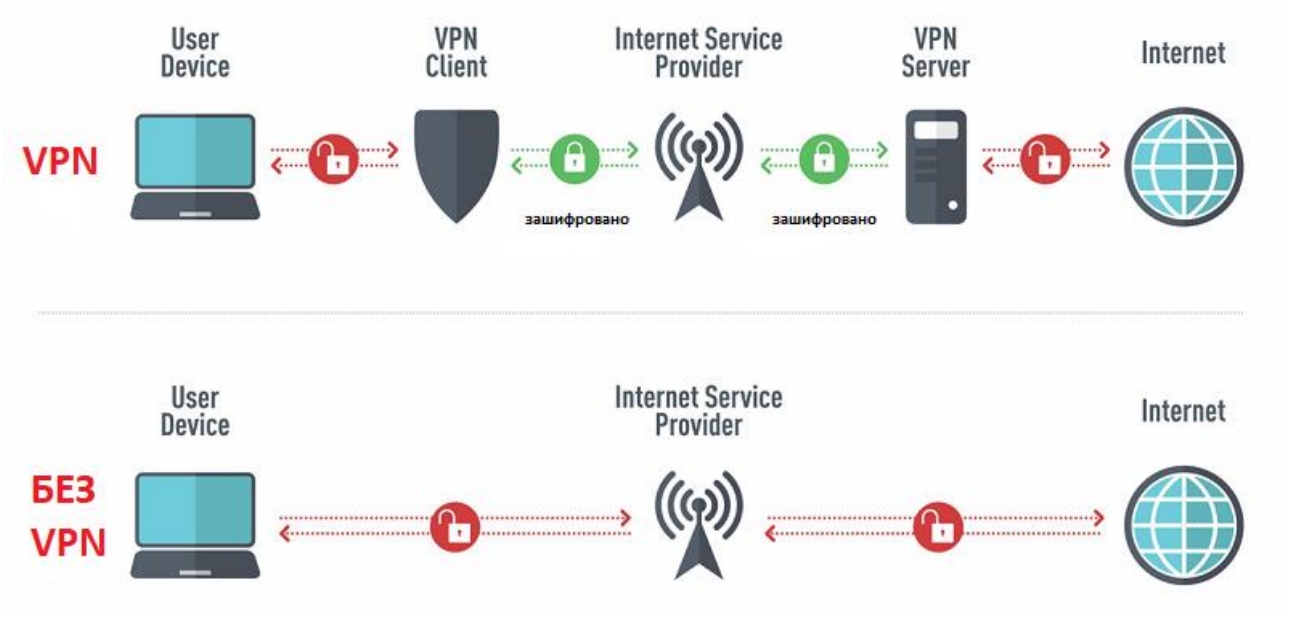

Рисунок 1.9 – Шифрування в мережі VPN [17]

Під час прийняття пакетів протокол PPTP може змінювати ключ шифрування. В мережах типу “точка-точка”, де працює Microsoft Point-to-Point Encryption протокол, пакети в основному ідуть один за одним і тому шанс не отримати інформацію дуже малий. В такому випадку ключ, а саме його величина для поточного пакету зумовлюється результатом дешифрування пакету, який був перед ним. Під час конфігурації VPN мереж взяти таку модель за основу неможливо, так як пакети відправляються в одному порядку а зазвичай приходять у зовсім іншому. Для того щоб уникнути такої ситуації і з’явилася можливість змінити ключ шифрування, PPTP протокол бере до уваги порядок проходження протоколів, а саме їхні порядкові номери. Така маніпуляція допомагає здійснити дешифрування пакету не беручи до уваги уже прийняті. Важливо враховувати, що процес шифрування може сповільнити з’єднання. Саме тому, коли необхідно прийняти рішення, які методи шифрування обрати для VPN мереж, необхідно враховувати тип даних запланованих для передачі [18].

Завдяки взаємодії трьох складових, а саме тунелюванню, аутентифікації та шифруванню виникає можливість безпечно відправляти інформацію між двома хостами через відкриту мережу. Цим самим створюючи середовище для конфігурації VPN мережі.

Процес створення віртуальних приватних мереж може складатися з чотирьох етапів. На першому етапі встановлюється VPN сервер у локальній мережі, на другому етапі завантажується клієнтське ПО користувачем та встановлюється зв’язок з сервером. На третьому етапі користувач проходить аутентифікацію, на четвертому етапі (у разі успішного завершення третього етапу) сервер та клієнтське ПО домовляються про умови безпечного з’єднання. Після вдалого завершення всіх етапів VPN мережа починає функціонувати і забезпечує безпечну передачу зашифрованих даних.

1.4 Протоколи VPN мереж: OpenVPN, IPSec , L2TP / IPSec, SSTP, IKEv2, PPTP, WireGuard

VPN-протокол – програмний фундамент, на базі якого будується будь- який VPN-сервіс. У ньому описується формат організації підключення, обміну даними всередині приватної віртуальної мережі та інші аспекти роботи ПО.

Від вибору протоколу залежить, які завдання будуть за допомогою нього вирішені, наскільки ефективно вони будуть вирішуватися, наскільки це буде безпечно та швидко. Існує кілька технологій організації VPN, тому виникають деякі розбіжності при виборі відповідних сервісів і при налаштуванні віртуальних приватних мереж [13].

У кожного протоколу свій набір характеристик, враховуючи які необхідно обирати протокол. До характеристик можна віднести саме такі:

- Підтримувані платформи – протоколи можуть бути досить специфічні та функціонувати виключно на одній-двох операційних системах. Інші ж підтримують відразу всі доступні ОС;
- Підтримувані мережі – не всі протоколи працюють в ідентичних мережах. Деякі VPN-сервіси пропонують свої послуги тільки в конкретних країнах з огляду на технологічні обмеження, введених, в тому числі, державними органами;
- Швидкість роботи – теж залежить від архітектури протоколу. Є ті, що швидше передають дані на мобільних пристроях. Є ті, що показують пікову продуктивність тільки в масштабах великих корпоративних мереж;
- Безпека – в протоколах по-різному реалізовано шифрування та інші механізми забезпечення безпеки даних. Тому, в залежності від поставлених завдань, треба вибирати технологію, яка найменш схильна до поширених для неї атака.

До найбільш популярних VPN протоколів можна віднести такі: OpenVPN, IPSec, L2TP / IPSec, SSTP, IKEv2, PPTP, WireGuard.

OpenVPN – це протокол VPN, який славиться тим, що має відкритий вихідний код і вдало пройшов безліч експертиз. Це означає, що користувачі можуть перевіряти вихідний код на наявність вразливостей або використовувати його в інших проектах. OpenVPN став одним з найважливіших протоколів VPN. Крім забезпечення надійного шифрування, OpenVPN також доступний практично для кожної платформи: Windows, MacOS, Linux, Android, iOS, маршрутизаторів і багато чого іншого [19, 32].

Також він забезпечує швидкість передачі даних вище, ніж у конкурентів. Багато сервісів, що базують свої VPN-сервери на базі OpenVPN, забезпечують передачу зашифрованого контенту на швидкості до 2000 Мбіт в секунду.

Протокол OpenVPN відповідає за підтримання комунікації між клієнтом і сервером. Як правило, він використовується для створення захищеного "тунелю" між VPN-клієнтом і VPN-сервером.

Для шифрування і аутентифікації OpenVPN використовує бібліотеку OpenSSL. Крім того, для передачі даних OpenVPN може використовувати UDP (User Datagram Protocol) або TCP (Transmission Control Protocol) протоколи.

Протоколи TCP та UDP знаходяться на транспортному рівні і використовуються для передачі даних в інтернеті. TCP вважається більш стабільним, так як пропонує функцію виправлення помилок (після відправки мережевого пакету TCP очікує підтвердження перед його повторної відправкою або відправкою нового пакета). UDP виправляє помилки, що робить його менш стабільним, але набагато швидшим.

OpenVPN найкраще працює по UDP, тому сервер доступу OpenVPN спочатку намагається встановити UDP-з'єднання. Якщо це не вдається, тільки тоді сервер

намагається створити з'єднання по протоколу TCP. Більшість VPN-сервісів за замовчуванням надають OpenVPN через UDP.

При використанні IPSec протоколу гарантується безпека передачі інформації по мережі. Завдяки йому дані зберігають свою конфіденційність, вони ніким не захоплені та цілісніть (мається на увазі, що отримані дані знаходяться в тому ж вигляді в якому були відправлені і мають той самий зміст). Також IPSec протокол доводить, що дані були відправлені саме тим чи іншим користувачем [13].

Безпечність IPSec досягається за рахунок двох основних механізмів: Authentication Header – ставить цифровий підпис кожній одиниці даних, що передається через VPN-з'єднання, Encapsulating Security Protocol – захищає цілісність інформації, що передається, та конфіденційність користувачів протоколу.

L2TP / IPSec (Layer 2 Tunneling Protocol) сам по собі не дозволяє захистити користувачів, тому його частини використовують спільно з IPSec. Протокол L2TP / IPsec відрізняється тим, що може гарантувати безпеку передачі даних на високому рівні, при цьому його легко налаштувати та може працювати з усіма ОС. Недоліком даного протоколу є те, що він інкапсулює дані, які були передані, двічі і цим самим він починає працювати повільніше у порівнянні з іншими протоколами.

Дані, які передаються через L2TP, не можуть бути змінені на шляху від відправника до одержувача. Даний протокол шифрує навіть сам процес аутентифікації, що значно ускладнює можливість проникнення і перегляд даних для будь-яких третіх осіб. Завдяки UDP інкапсуляції даних протокол L2TP можна простіше і швидше налаштувати для більшості брандмауерів [20].

Клієнти, підключені до VPN, часто запускають пряме програмне забезпечення L2TP і IPSec. Зазвичай немає необхідності встановлювати додаткове програмне забезпечення в клієнтських системах для зв'язку з сервером L2TP VPN: програмне забезпечення L2TP VPN надається з системами Windows, OS X, iOS, Android і Linux.

SSTP (Secure Socket Tunneling Protocol) – пропріетарний продукт від Microsoft. В основному він працює з Windows ОС, хоча також може бути задіяний і в Linux системах та інших операційних системах. Необхідно сказати, що не весь пакет шифрується за допомогою SSL – HTTPS-заголовок, SSTP-заголовок, PPP-заголовок і корисне навантаження шифруються, в той час як TCP-заголовок і SSL-заголовок – ні. При встановленні SSL з'єднання проходить авторизація сервера клієнтом по SSL сертифікату [21].

Він призначений для синхронного обміну даними між двома програмами і дозволяє використовувати безліч кінцевих точок застосунку по одному мережевому з'єднанню між рівноправними вузлами. Це дозволяє ефективно використовувати комунікаційні ресурси, які доступні в мережі.

Розвиток SSTP був викликаний проблемами з безпекою протоколу PPTP. SSTP встановлює з'єднання по захищеному HTTPS; це дозволяє клієнтам безпечно отримувати доступ до мереж за NAT-маршрутизаторами, брандмауерами і веб-проксі, не турбуючись про стандартні проблеми блокування портів. SSTP не призначений для VPN-підключень «сайт-сайт», але призначений для VPN-підключень «клієнт-сайт».

IKEv2 (Internet Key Exchange v2) протокол розроблено командою розробників з Microsoft і Cisco, але має кілька варіацій з відкритим вихідним кодом, написаних незалежними програмістами. IKEv2 хороший наявністю підтримки Mobility and Multi-homing Protocol. Це робить його стійким до зміни мереж і допоможе власникам смартфонів залишатися на зв'язку навіть при виході в мережу через VPN. Підключення до VPN-сервера не обривається при зміні роутера, до якого підключений гаджет або зміні точки доступу під час поїздок. Також він споживає менше ресурсів, ніж умовний OpenVPN, і від цього демонструє більш високу швидкість передачі даних [22].

IKEv2 вміє повторно підключатися в моменти тимчасової втрати інтернет з'єднання, а також під час мережевого комутатора (наприклад, від Wi-Fi до мобільних даних).

IKEv2 використовує перевірку автентичності сертифіката сервера, що означає, що він не буде виконувати ніяких дій, поки не визначить особистість запитуючої сторони. Завдяки цьому кількість атак і DoS атак значно зменшується.

Продумана архітектура і ефективна система обміну повідомленнями протоколу IKEv2 забезпечують кращу продуктивність. Завдяки вбудованому механізму NAT, який робить прохід через брандмауер і встановлює з'єднання набагато швидше, швидкість IKEv2 з'єднання також значно підвищується.

PPTP (Point-to-Point Tunneling Protocol) – це протокол тунелювання «точка-точка» і є одним з найстаріших протоколів VPN. Він все ще використовується в деяких системах, але більша кількість служб уже оновлені до більш швидких і безпечних протоколів. Але технологія VPN прогресувала, і PPTP більше не захищений. PPTP дає кращі швидкості з'єднання, саме через відсутність функцій безпеки (в порівнянні з сучасними протоколами).

Як правило протокол PPTP працює з двома з’єднаннями, одне використовується для процесу управління, а друге застосовують для інкапсуляції даних. Протокол TCP використовується з першим з’єднанням, а з другим використовується GRE протокол [22].

WireGuard — це протокол VPN, який з'явився кілька років тому. У 2020 році технологію було впроваджено в ядро Linux з номером 5.6. WireGuard безпечніше, швидше та простіше за інші VPN.

WireGuard виділяється серед конкурентів. В першу чергу, у нього більш компактний і вихідний код, що легко читається. Будь-які вразливості у VPN легко виявити та виправити. Завдяки цьому та зрозумілій документації розробникам

легше вивчати принцип роботи та імплементувати протокол у власні напрацювання [23].

WireGuard використовує найшвидші та найсучасніші криптографічні алгоритми, визнані фахівцями. Такий вузький набір знижує гнучкість шифрування, але збільшує надійність. Протокол був розроблений з прицілом на меншу складність, що робить його стійким до атак.

VPN протоколи є основою сервісу, адже містять в собі правила передачі даних та шифрування, які дають можливість легко та безпечно передавати дані з між VPN-серверами.

1.5 Методи побудови VPN мереж

Для побудови віртуальних приватних мереж можуть бути задіяні різноманітні методи та способи. При цьому важливо розуміти межу продуктивності пристроїв, які будуть використані при побудові VPN. Якщо в мережі використовується роутер, який і так перевантажений то створення на ньому тунелів та здійснення шифрування взагалі може привести до того, що вся мережа перестане працювати, адже роутер просто не справиться з навантаженням. До методів побудови VPN мережі можна віднести такі:

- VPN на основі брандмауерів

Функції шифрування та тунелювання зазвичай підтримуються великою кількістю брандмауерів, які містять модуль шифрування. У цьому випадку інформація, яка передається через брандмауер є зашифрованою. Важливо сказати, що до недоліків такого методу можна віднести те, що продуктивність такої мережі залежить від апаратного забезпечення, де брандмауер працює [24].

Приклад побудови VPN мережі на основі брандмауера зображений на рис. 1.10.

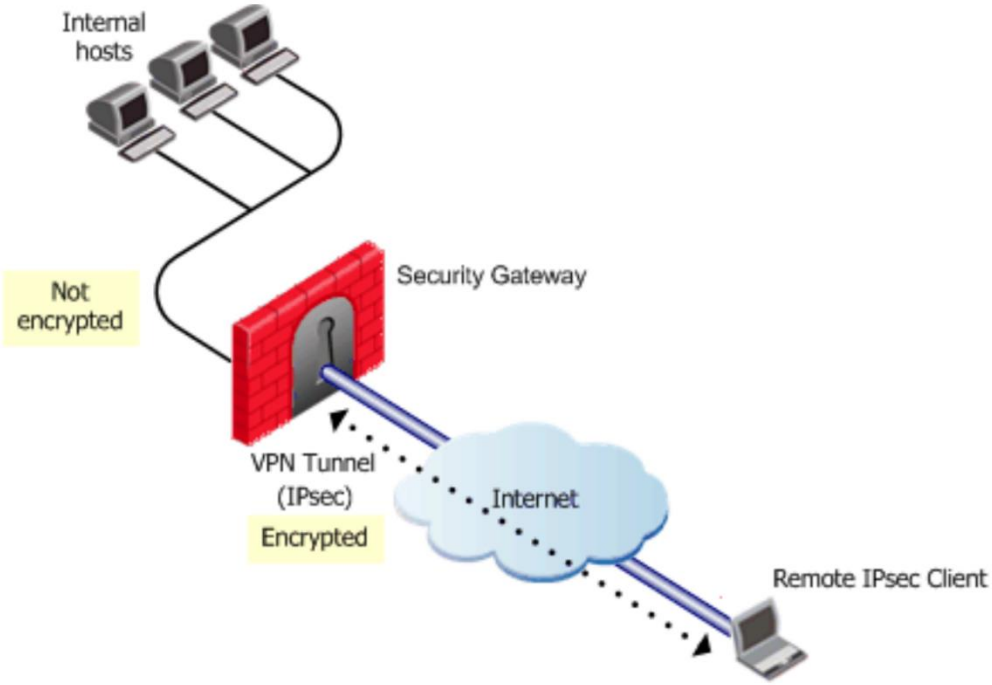

Рисунок 1.10 – VPN на основі брандмауерів [25]

Конфігурація VPN мережі з використанням брандмауерів є досить гарним рішенням для створення захисту віртуальних мереж. Якщо об’єднати методи захисту, які використовують брандмауери з VPN-шлюзом то майже всі функції, які використовуються для забезпечення безпеки мережі, опиняються на одному пристрої.

- VPN на основі маршрутизаторів

Не менш ефективним методом для створення VPN мережі є використання маршрутизаторів. Оскільки, дані, які знаходилися в локальній мережі все рівно проходять через маршрутизатор, то важливо використовувати і шифрування для забезпечення безпеки інформації. VPN мережі дають можливість підключення

клієнтів через маршрутизатор із використанням приватних, а не публічних адрес, як це реалізується у варіанті IP-мережі.

Шифрування пакетів не є основним завданням для маршрутизаторів, а лише додатковим, що в свою чергу забирає більшу кількість ресурсів маршрутизатора. Тож, якщо маршрутизатору вистачає продуктивності конфігурити VPN мережу то він з такою задачею легко справиться, а якщо він і так перегружений то він не лише не зможе справитися з VPN мережею а і з власними основними задачами.

Основною проблемою, що виникає при реалізації маршрутизації між VPN клієнтами на єдиному сервері, є їх ізоляція трафіку при довільних топологіях маршрутизації та комутації. Різні VPN послуги використовують віртуальні інтерфейси для маршрутизації трафіку VPN клієнтів.

При використанні методу побудови мережі на основі маршрутизаторів важливо сказати про те, що лише цей спосіб не допоможе захистити корпоративну мережу в повній мірі, адже корпоративні дані все ще залишаються вразливими. Тому разом з цим маршрутизаторами використовують і брандмауери для більш надійного захисту [26].

Приклад побудови VPN мережі на основі маршрутизатора зображений на рис. 1.11.

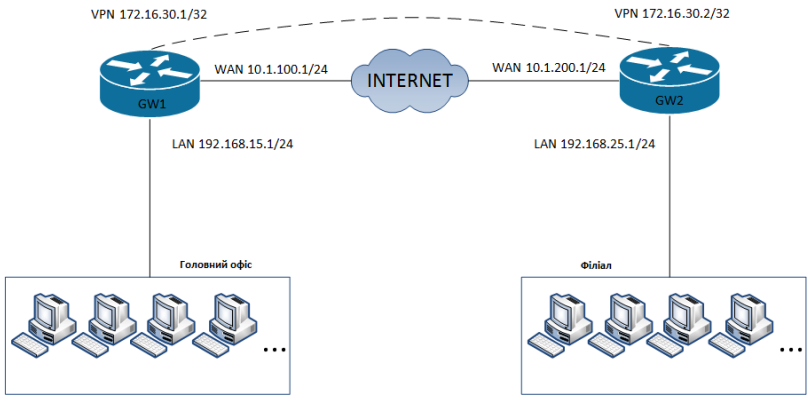

Рисунок 1.11 – VPN на основі маршрутизаторів [27]

- VPN на основі програмного забезпечення

Ще одним способом конфігурації VPN мережі є використання програмного забезпечення. У цьому випадку програмні рішення установлюються на окремий сервер, який в основному працює як proxy-сервер. При застосуванні такого підходу виділений комп'ютер знаходиться за брандмауером.

Приклад побудови VPN мережі на основі програмного забезпечення зображений на рис. 1.12.

Рисунок 1.12 – VPN на основі програмного забезпечення [28]

При виборі методу побудови віртуальних приватних мереж потрібно звертати увагу на велику кількість особливостей від яких залежить успішність та ефективність побудови.

1.6 Постановка задачі

Провівши аналіз літератури та з’ясувавши важливість VPN мереж на сьогоднішній день, метою магістерської кваліфікаційної роботи є розробка веб-орієнтованої системи, де можна було б автоматично налаштовувати маршрутизатори за певним протоколом. Також важливим елементом такої програми є можливість перенести зконфігур овані налаштування на симулятори та реальне обладнання.

Система повинна бути інтуїтивно зрозумілою та легкою у використанні навіть для недосвідчених користувачів, які мають небагато досвіду у налаштуванні мереж.

Веб-сторінка повинна містити форми для введення IP адрес маршрутизаторів, кнопку для генерації налаштувань та кнопку для копіювання згенерованих налаштувань для перенесення їх у симулятори для перевірки.

Постановка задачі:

1. Конфігурація VPN мережі у симуляторі Packet Tracer;
2. Розробка веб-орієнтованої системи для налаштування маршрутизаторів VPN мережі;
3. Тестування зконфігурованого налаштування за допомогою програми у симуляторі GNS3.

2 МОДЕЛЮВАННЯ VPN МЕРЕЖІ З ВИКОРИСТАННЯМ СИМУЛЯТОРА PACKET TRACER

2.1 Конфігурація мережі з використанням симулятора PACKET TRACER

При проектуванні комп’ютерних мереж чи їх перевірки застосовуються різного роду інструменти. Такі інструменти називаються мережевими симуляторами. Вони допомагають системним адміністраторам працювати у разі необхідності не з реальними мережами а із змодельованими.

До найбільш популярних інструментів для моделювання можна віднести такі: Cisco Packet Tracer, GNS3, Boson NetSim, VIRL, EVE-NG.

Cisco Packet Tracer – це кросплатформовий інструмент візуального моделювання, розроблений Cisco Systems. Це дає можливість створювати мережеві топології та імітувати сучасні комп'ютерні мережі.

Дане програмне забезпечення дозволяє імітувати конфігурацію маршрутизаторів та комутаторів Cisco за допомогою інтерфейсу моделюючої лінії.

GNS3 (Graphical Network Simulator) – це одна з найпопулярніших програм емуляції мережі, яка дозволяє спостерігати взаємодію мережевих пристроїв в різних топологіях мереж, дозволяє змоделювати віртуальну мережу з маршрутизаторів і віртуальних машин. GNS3 підтримує великий обсяг віртуальних мережевих пристроїв від різних постачальників мережевого обладнання за рахунок використання пристроїв, які є простими в імпорті шаблонами. У GNS3 кожний віртуальний мережевий пристрій можна запускати і зупиняти незалежно від інших віртуальних пристроїв [29].

Boson NetSim – інструмент фактично імітує мережевий трафік реальної мережі, яку користувачі можуть створювати самостійно. Інструмент пропонує

технологію віртуальних пакетів: програмно створені пакети, які маршрутизуються та комутуються через моделюючу мережу. Якщо мереживий пристрій можна налаштувати шляхом додавання модулів, програма уточнить, які модулі необхідно вставити в пристрій при його додаванні в топологію мережі. NetSim визначає тип інтерфейсів, які додає кожен модуль.

VIRL (Virtual Internet Routing Lab) – це емулятор віртуальної мережі Cisco, призначений для освітніх установ і приватних осіб. VIRL працює в клієнт-серверній моделі, яка аналогічна GNS3. Сервер VIRL встановлюється або на окремому сервері, або в якості віртуальної машини, потім створюються мережеві топології, і вони взаємодіють з сервером за допомогою клієнтської програми VM Maestro. Емулятор включає в себе функцію AutoNetKit, яка дозволяє автоматично заповнювати базову конфігурацію функцій на вузлах по всій топології мережі [30].

EVE-NG (Emulated Virtual Environment Next Generation) – це емулятор віртуальної мережі, який, подібно до VIRL Personal Edition, був розроблений для приватних осіб та невеликих підприємств. Клієнт EVE-NG – це ключова особливість, яка відрізняє даний емулятор від VIRL і GNS3. У EVE проектування, підключення і управління мережевими топологіями виконується через браузер. Не потрібно завантажувати і встановлювати окремий додаток як доповнення до сервера для віртуалізації, підключення та налаштування мережевих пристроїв. Необхідно просто розгорнути сервер через налаштування або віртуальну машину, а все інше зробити за допомогою браузера [31].

За допомогою Cisco Packet Tracer інструменту налаштуємо VPN мережу, яка зображена на рис. 2.1, використовуючи IPsec протокол. Спершу збираємо схему, розподіляємо IP адреси та налаштовуємо маршрутизацію.

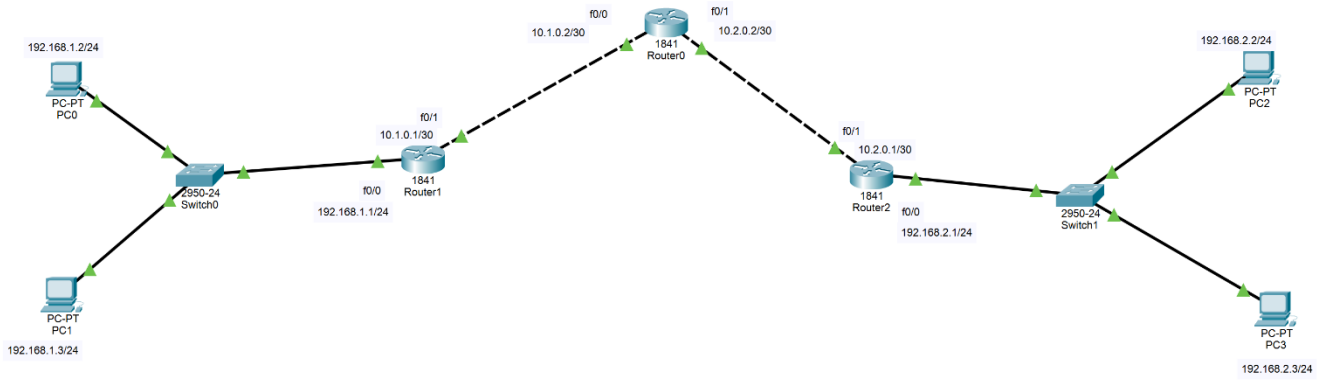

Рисунок 2.1 – Схема VPN мережі в Packet Tracer

Для схематизації використовувались роутери 1841, комутатори та комп’ютери. Далі налаштуємо маршрутизатор Router1 як зображено на рис. 2.2.

Рисунок 2.2 – Налаштування VPN на роутері Router 1

Командою ip nat outside ми вказали, що інтерфейс f0/1 обслуговує зовнішній елемент мережі, а командою ip nat inside ми аналогічно показали, що інтерфейс f0/0 обслуговує внутрішній елемент мережі. Далі командою ip access-list extended FOR-NAT створюємо список доступу для визначення трафіку, який випускається в Інтернет. Командою deny ip 192.168.1.0.0.0.0.255 192.168.2.0.0.0.0.255 забороняємо трафік, який не потрібно піддавати обробці NAT. Використовуємо команду permit ip 192.168.1.0 0.0.0.255 any для надання дозволу на доступ до мережі 0.0.0.255. Командою ip nat inside source list FOR-NAT interface f0/1 overload прив'язуємо внутрішні адреси до зовнішнього інтерфейсу. Далі налаштовуємо VPN і спершу задаємо політику з рівнем пріоритету командою crypto isakmp policy 1. Командою encryption 3des задаємо алгоритм шифрування повідомлень. Використовуємо команду hash md5 для задання алгоритму хешування. Командою authentication pre-share визначемо метод аутентифікації сторін з попередньо узгодженими ключами та вкажемо ідентифікатор групи Діффі-Хеллмана з 1024-бітним ключом шифрування за допомогою команди group 2. Встановлюємо значення загального ключа та ip адресу віддаленої сторони, використовуючи команду crypto isakmp key cisco address 10.2.0.1.Командою crypto ipsec transform-set tunnel esp-3des esp-md5-hmac задаємо набір перетворень під ім'ям тунель із зазначенням двох перетворень, які визначають протоколи та алгоритми захисту IPsec. Далі створюємо розширений список доступу з назвою VOR-VPN, який визначає який трафік буде проходити по VPN тунелю – ip access-list extended FOR-VPN та дозволяємо передачу даних від одної мережі до іншої – permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255. Використовуючи команду crypto map CMAP 10 ipsec-isakmp створюємо крипто карту під ім'ям CMAP та порядковим номером 10. Далі ідентифікуємо ipsec партнерів – set peer 10.2.0.1 та встановлюємо попередньо створений набір перетворень – set transform-set tunnel. Опираємося на список доступу, який визначає вибір трафіку, який підлягає захисту засобами ipsec – match address FOR-VPN. Далі

прив’язуємо крипто карту до інтерфейсу f0/1. Налаштування маршрутизатора Router 1 завершена.

Подібним способом налаштовуємо маршрутизатор Router 2 як зображено на рис. 2.3.

Рисунок 2.3 – Налаштування VPN на роутері Router 2

Після того, як були налаштовані обидва роутери можна перевірити працездатність створеного тунелю можна за допомогою команди ping з одного комп’ютера однієї підмережі до комп’ютера іншої підмережі як це показано на рис. 2.4.

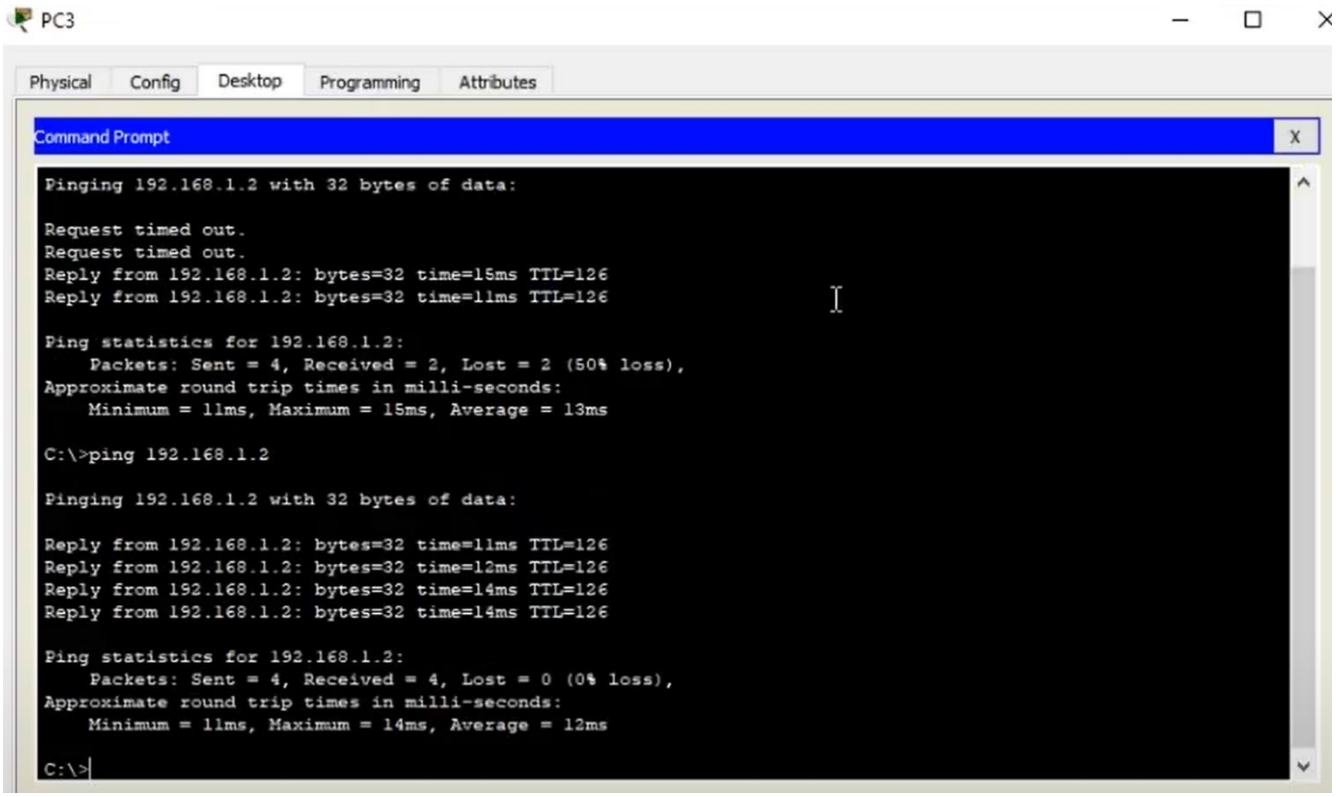

Рисунок 2.4 – Перевірка VPN тунелю за допомогою команди ping

Також можна здійснити перевірку за допомогою введенням на одному із роутерів команди show crypto isakmp sa як показано на рис. 2.5.

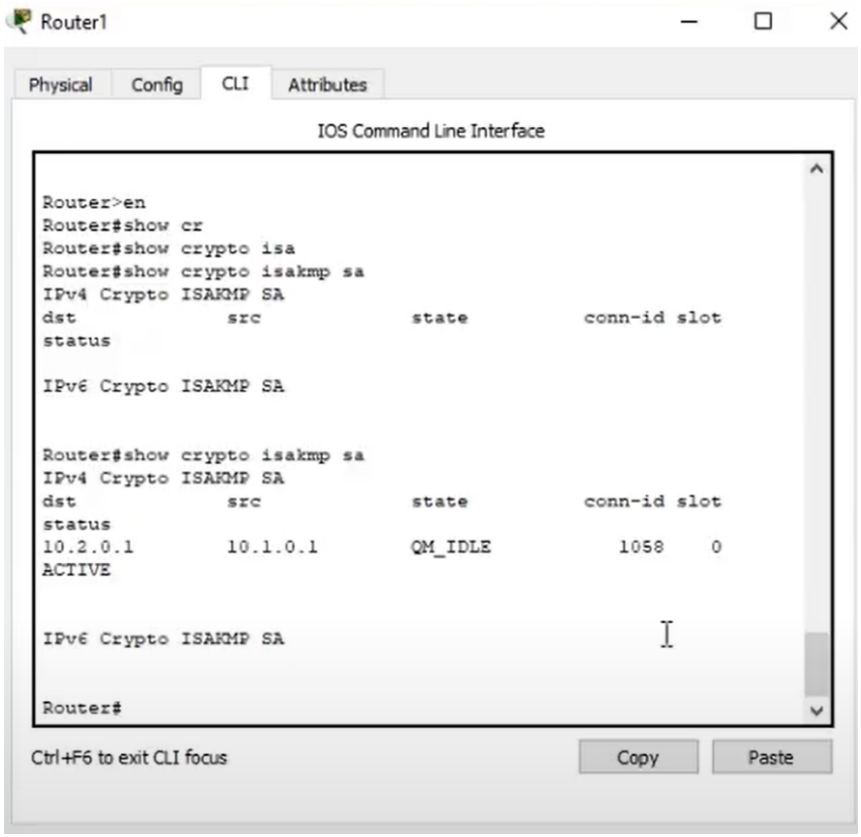

Рисунок 2.5 – Перевірка VPN тунелю за допомогою команди show crypto isakmp sa

Як бачимо тунель має статус Active. Отже, всі налаштування були виконані вірно. Packet Tracer є досить корисним інструментом, адже допомагає мереживим адміністраторам проектувати мережі заздалегідь, продумати всі нюанси та елементи а вже потім переносити в реальне середовище.

2.2 Розробка веб-орієнтованої системи для створення VPN мережі

Розробка програми була здійснена за допомогою JavaScript. JavaScript це об’єктно-орієнтована мова програмування, яка робить веб-сторінки інтерактивними.

JavaScript в браузері може добавляти новий HTML-код на сторінку, змінювати уже існуючий зміст, змінювати стилі. Вся візуальна частина програми – це поєднання HTML+CSS. Тобто кнопки, діаграми, повзунки це статичні елементи. За допомогою JS все анімується.

Програми в JavaScript називаються скриптами. Скрипти надаються та виконуються у вигляді простого тексту. Вони не потребують спеціальної підготовки чи компіляції для запуску. Сьогодні JavaScript може виконуватися не тільки в браузері, але і в серверній частині [32].

До основних особливостей JavaScript програмування належать:

1. Динамічна типізація. Тобто тип даних визначатиметься лише тоді, коли змінною чи const буде присвоюватися її значення.
2. Гнучка робота із функціями. У JS функції можна не тільки виконувати, але ще й повертати функції з функцій, передавати функції як параметри іншим функціям і надавати функції значення змінних.
3. JavaScript підтримується усіма сучасними браузерами.
4. Об'єктно-орієнтоване програмування. Тобто це така методологія програмування, у якій вся програма представляється у вигляді сукупності об'єктів.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ГРАФІЧНОГО ІНТЕРФЕЙСУ НАЛАШТУВАННЯ VPN МЕРЕЖІ

3.1 Розробка графічного інтерфейсу налаштування VPN мережі

В симуляторі Packet Tracer були налаштовані роутери та маршрутизація для створення VPN мережі за допомогою протоколу IPSec. Як бачимо головним недоліком такого роду симуляторів є відсутність зрозумілого графічного інтерфейсу для конфігурації складних мереж. Конфігурація мережі забирає багато часу і якщо таких мереж необхідно зконфігурити декілька десятків за день то легко щось наплутати. Розробка графічного інтерфейсу для налаштування VPN мережі є досить актуальним завданням.

Програма була реалізована за допомогою мови програмування JavaScript. Інтерфейс застосунку є простим та зрозумілим. Для наглядності до веб-сторінки окрім IPSec протоколу був доданий також протокол IKEv2.

Відкриваючи онлайн сторінку ми бачимо схему мережі, IPSec, IKEv2 протоколи з radio button та кнопки [Generate] та [Copy] як зображено на рис. 3.1.

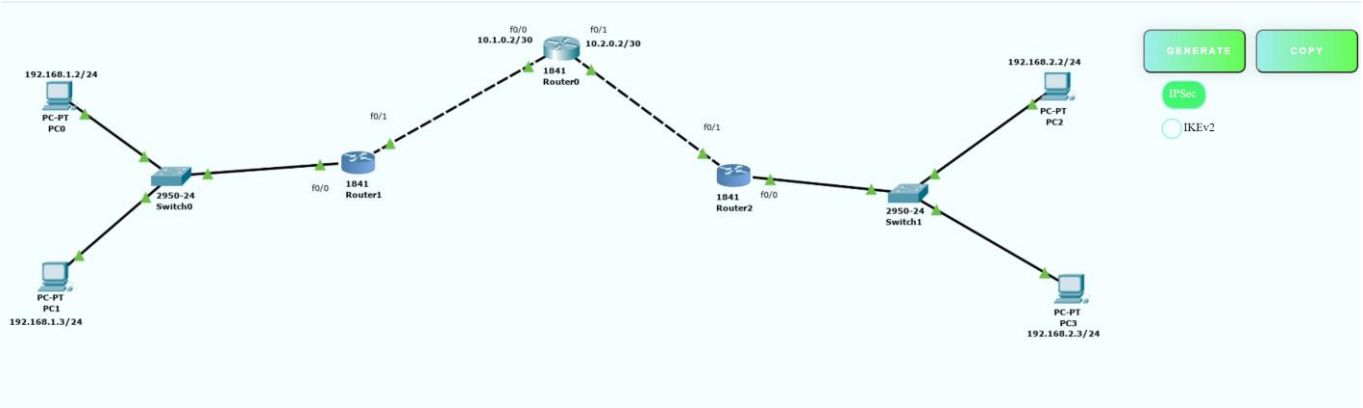

Рисунок 3.1 – Інтерфейс веб-сторінки для налаштування VPN мережі

Для того, щоб почати конфігурацію потрібно обрати необхідний роутер, та ввести IP адресу та маску його інтерфейсів. Для введення IP адреси та маски була

також застосована валідація, при введенні даних невірного формату відповідні поля підсвічуються червоним кольором як зображено на рис. 3.2.

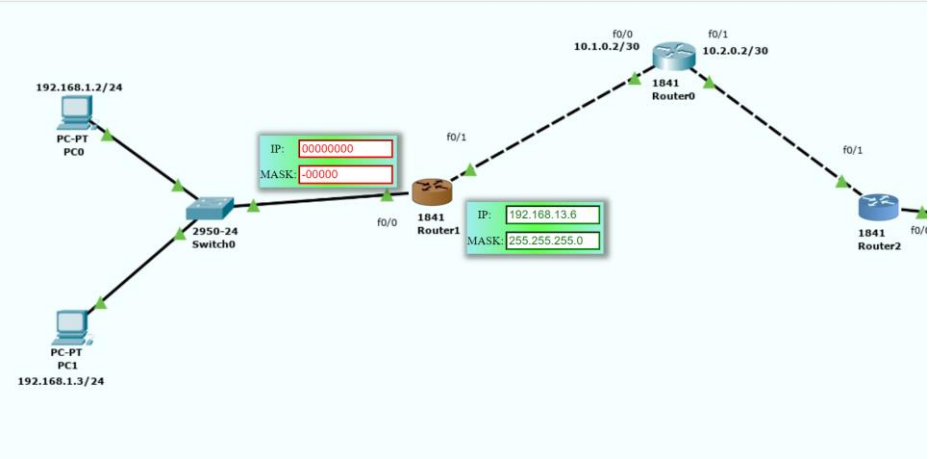

Рисунок 3.2 – Валідація на введення даних невірного формату

Після того як були введені вірні дані для обох роутерів (вірні дані підсвічуються зеленим кольором на екрані з’явиться згенерований код для роутерів за вибраним протоколом. Для того щоб подивитися код для певного роутера потрібно його вибрати як зображено на рис. 3.3 та рис. 3.4.

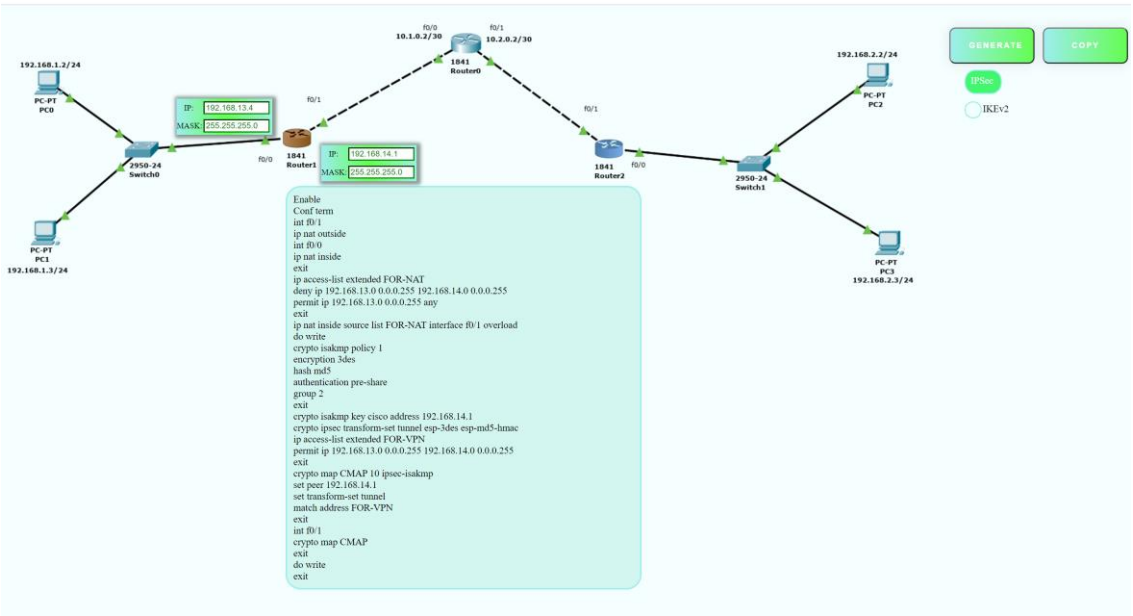

Рисунок 3.3 – Згенерований код налаштувань для Router 1

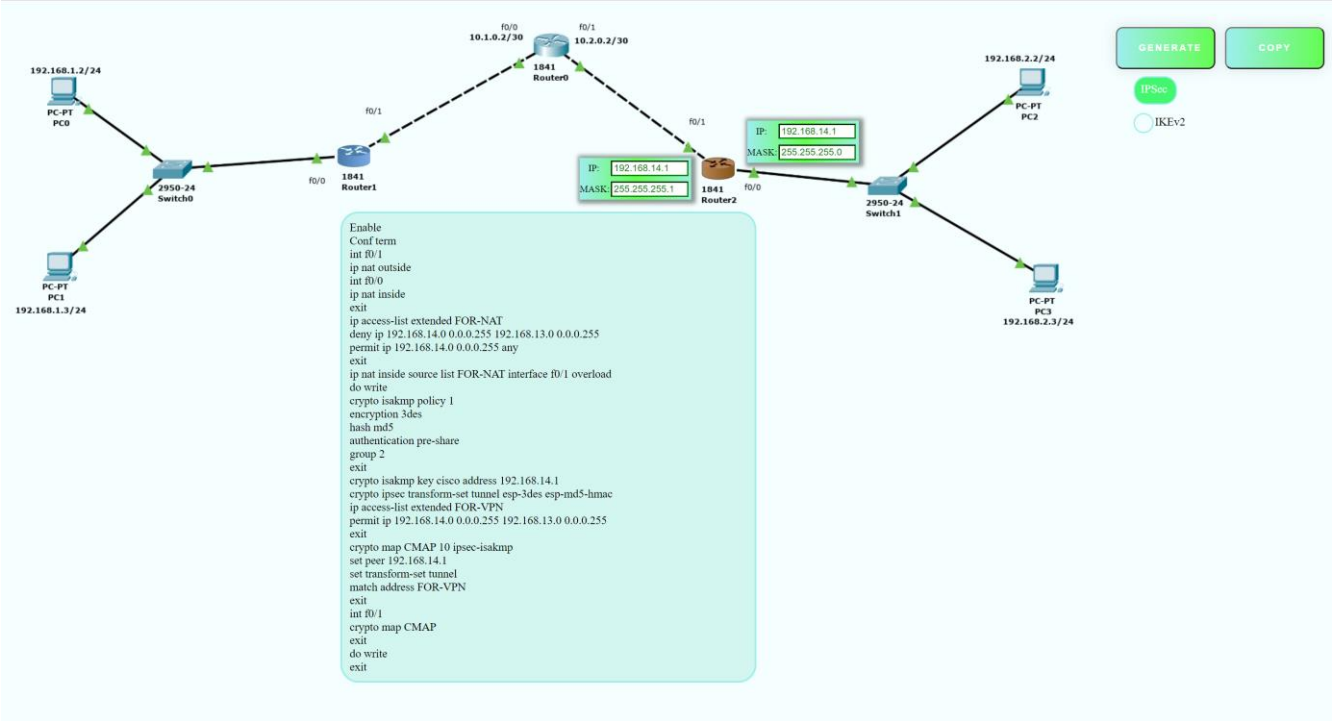

Рисунок 3.4 – Згенерований код налаштувань для Router 2

Якщо потрібно згенерувати код налаштувань з цими же даними за іншим протоколом, необхідно вибрати протокол IKEv2 та натиснути кнопку [Generate] як зображено на рис. 3.5.

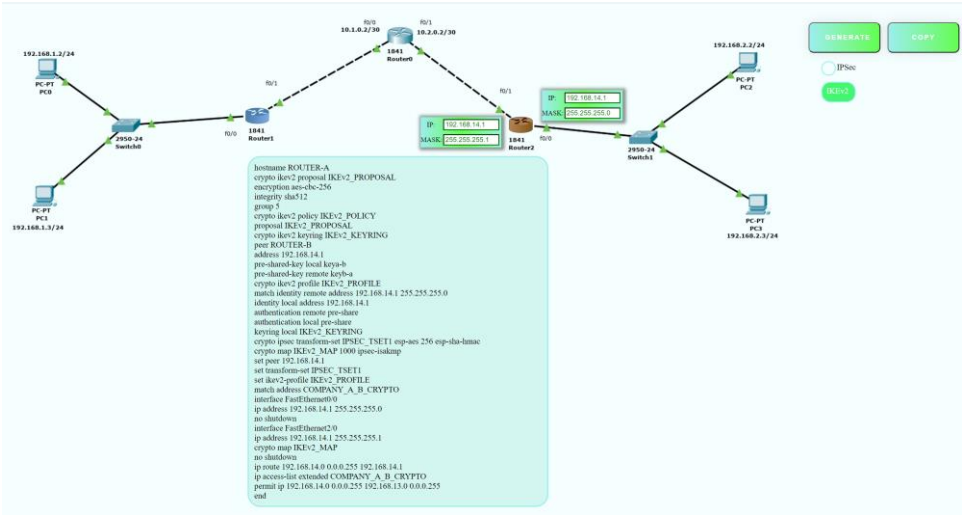

Рисунок 3.5 – Згенерований код налаштувань за протоколом IKEv2

Для того, щоб налаштувати VPN мережу на реальних пристроях необхідно скопіювати згенерований код за допомогою кнопки [Copy] та вставити його в поле налаштувань відповідного роутера.

3.2 Тестування створеної системи для налаштування VPN мережі в симуляторі GNS3

Для того, щоб протестувати розроблену програму, необхідно перевірити згенерований код налаштувань у симуляторі GNS 3. Перевірка буде здійснена за протоколом IPSec. Спочатку необхідно ввести IP адресу та маску для роутерів Router 1 та Router 2 в розробленому інтерфейсі, вибрати протокол IPSec та згенерувати код налаштувань як зображено на рис. 3.6 та рис. 3.7.

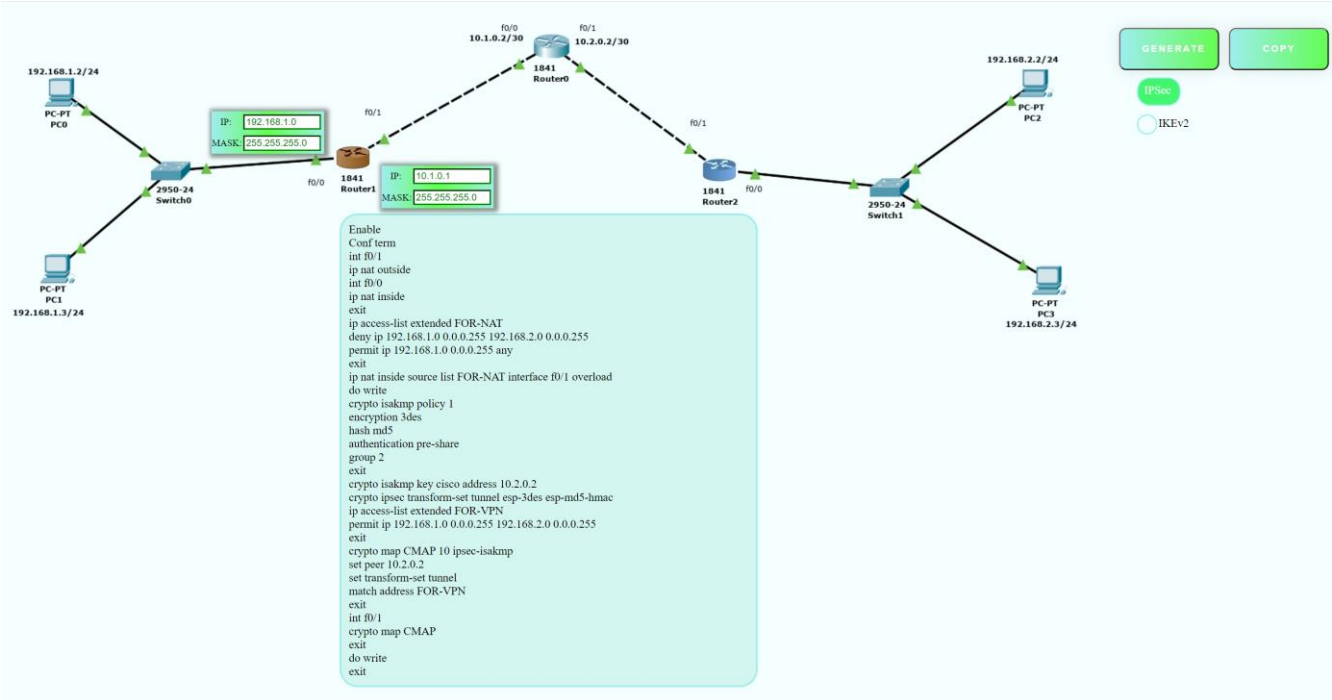

Рисунок 3.6 – Згенерований код налаштувань для Router 1

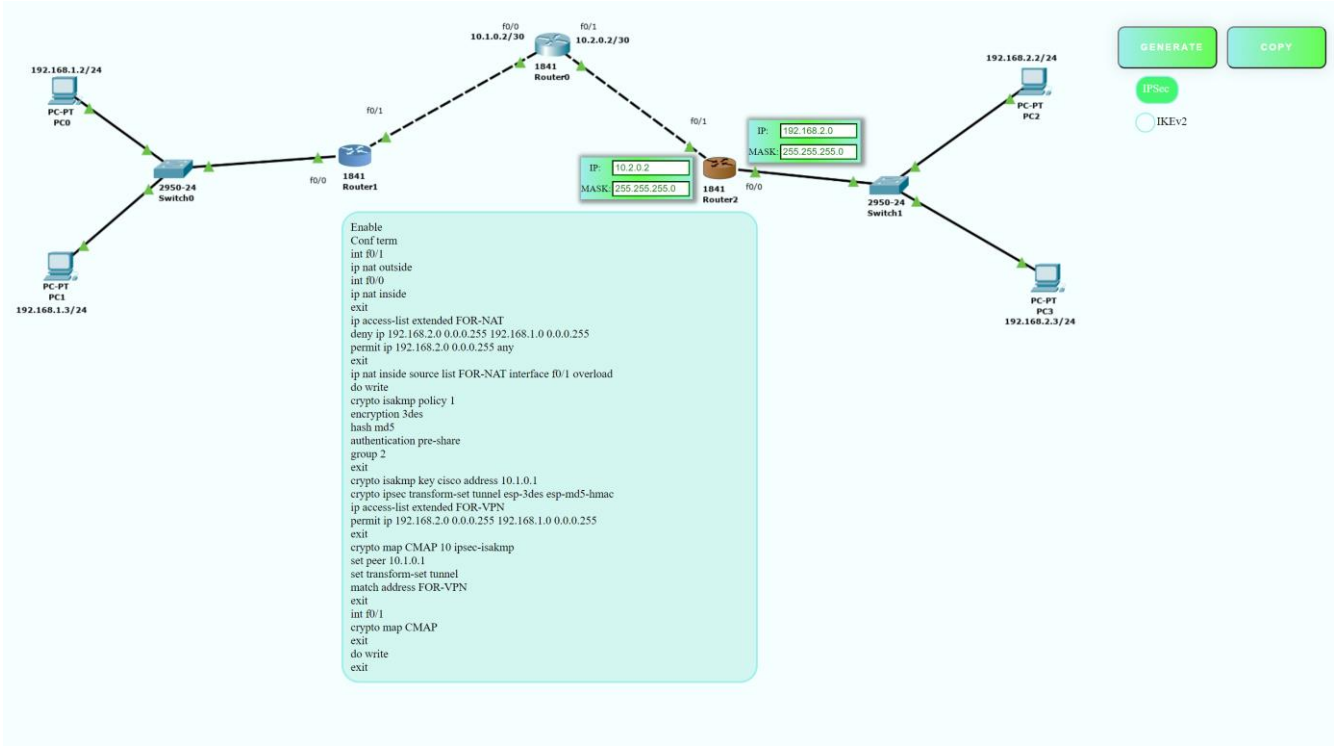

Рисунок 3.7 – Згенерований код налаштувань для Router 2

Змоделюємо схему мережі у GNS 3, як зображено на рис. 3.8.

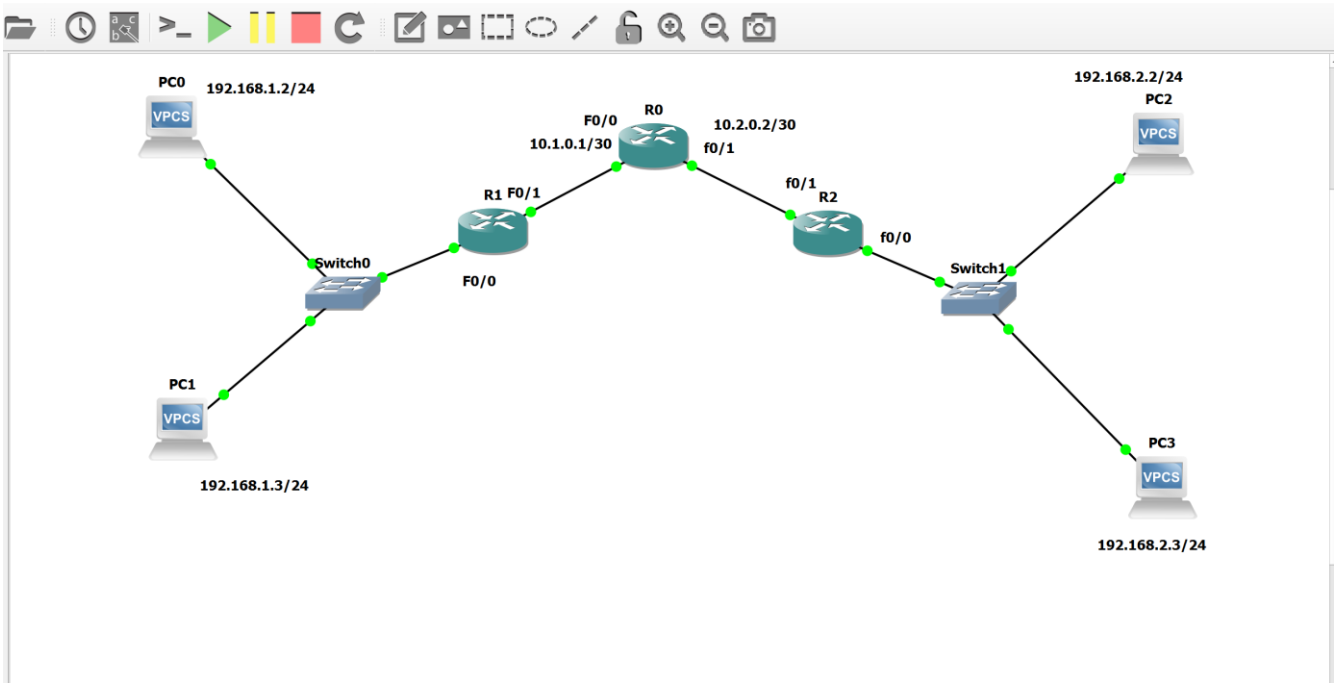

Рисунок 3.8 – Модель VPN мережі у GNS 3

Згенеровані налаштування копіюємо, використовуючи кнопку [Copy] та вставляємо у консоль роутерів 1 та 3 відповідно як зображено на рис. 3.9 та рис. 3.10.

```
r1#Enable
r1#Conf term
Enter configuration commands, one per line. End with CNTL/Z.
r1(config)#int f0/1
r1(config-if)#ip nat outside
r1(config-if)#int f0/0
r1(config-if)#ip nat inside
r1(config-if)#exit
r1(config)#ip access-list extended FOR-NAT
r1(config-ext-nacl)#deny ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
r1(config-ext-nacl)#permit ip 192.168.1.0 0.0.0.255 any
r1(config-ext-nacl)#exit
r1(config)#ip nat inside source list FOR-NAT interface f0/1 overload
r1(config)#do write
Building configuration...
[OK]
r1(config)#crypto isakmp policy 1
r1(config-isakmp)#encryption 3des
r1(config-isakmp)#hash md5
r1(config-isakmp)#authentication pre-share
r1(config-isakmp)#group 2
r1(config-isakmp)#exit
r1(config)#crypto isakmp key cisco address 10.2.0.2
r1(config)#crypto ipsec transform-set tunnel esp-3des esp-md5-hmac
r1(cfg-crypto-trans)#ip access-list extended  FOR-VPN
r1(config-ext-nacl)#permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
r1(config-ext-nacl)#exit
r1(config)crypto map CMAP 10 ipsec-isakmp
r1(config-crypto-map)set peer 10.2.0.2
r1(config-crypto-map)set transform-set tunnel
r1(config-crypto-map)match address FOR-VPN
r1(config-crypto-map)exit
r1(config)int f0/1
r1(config)crypto map CMAP
*Nov  4 17:16:08.786: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
r1(config)exit
r1(config)do write
Building configuration...
[OK]
r1(config)exit
```

Рисунок 3.9 – Скопійований код налаштувань у Router 1

```
r2#Enable
r2#Conf term
Enter configuration commands, one per line. End with CNTL/Z.
r2(config)#int f0/1
r2(config-if)#ip nat outside
r2(config-if)#int f0/0
r2(config-if)#ip nat inside
r2(config-if)#exit
r2(config)#ip access-list extended FOR-NAT
r2(config-ext-nacl)#deny ip 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255
r2(config-ext-nacl)#permit ip 192.168.2.0 0.0.0.255 any
r2(config-ext-nacl)#exit
r2(config)#ip nat inside source list FOR-NAT interface f0/1 overload
r2(config)#do write
Building configuration...
[OK]
r2(config)#crypto isakmp policy 1
r2(config-isakmp)#encryption 3des
r2(config-isakmp)#hash md5
r2(config-isakmp)#authentication pre-share
r2(config-isakmp)#group 2
r2(config-isakmp)#exit
r2(config)#crypto isakmp key cisco address 10.1.0.1
r2(config)#crypto ipsec transform-set tunnel esp-3des esp-md5-hmac
r2(cfg-crypto-trans)#ip access-list extended  FOR-VPN
r2(config-ext-nacl)#permit ip 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255
r2(config-ext-nacl)#exit
r2(config)crypto map CMAP 10 ipsec-isakmp
r2(config-crypto-map)set peer 10.1.0.1
r2(config-crypto-map)set transform-set tunnel
r2(config-crypto-map)match address FOR-VPN
r2(config-crypto-map)exit
r2(config)int f0/1
r2(config)crypto map CMAP
*Nov  4 17:19:08.786: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
r2(config)exit
r2(config)do write
Building configuration...
[OK]
r2(config)exit
```

Рисунок 3.10 – Скопійований код налаштувань у Router 2

Для перевірки налаштувань буде використана команда show crypto ipsec sa. Спочатку відправимо команду ping з комп’ютера PC-3 на комп’ютер PC-0. Як бачимо на рис. 3.11 пінг проходить.

```
PC-3> ping 192.168.1.2
84 bytes from 192.168.1.2 icmp_seq=1 ttl=62 time=97.939 ms
84 bytes from 192.168.1.2 icmp_seq=2 ttl=62 time=65.963 ms
84 bytes from 192.168.1.2 icmp_seq=3 ttl=62 time=85.946 ms
84 bytes from 192.168.1.2 icmp_seq=4 ttl=62 time=101.937 ms
84 bytes from 192.168.1.2 icmp_seq=5 ttl=62 time=88.944 ms

PC-3>
```

Рисунок 3.11 – Відправка ping команди

Далі запускаємо команду show crypto ipsec sa. У цій команді представлені зіставлення безпеки IPSec між вузлами. Між вузлами 10.1.0.1 та 10.2.0.2 створюється зашифрований тунель, що забезпечує передачу трафіку між мережами 192.168.1.0 та 192.168.2.0. Поля pkts encrypt та pkts decrypt відображають проходження трафіку через зашифрований тунель і повинні збільшуватись як зображено на рис. 3.12.

```
R1#show crypto ipsec sa

interface FastEthernet0/1
        Crypto map tag: CMAP, local addr 10.1.0.1

        protected vrf: (none)
        local  ident (addr/mask/prot/port): (192.168.1.0/255.255.255.0/0/0)
        remote ident (addr/mask/prot/port): (192.168.2.0/255.255.255.0/0/0)
        current_peer 10.2.0.2 port 500
          PERMIT, flags={origin_is_acl,}
         #pkts encaps: 9, #pkts encrypt: 9, #pkts digest: 9
         #pkts decaps: 9, #pkts denrypt: 9, #pkts verify: 9
         #pkts compressed: 0, #pkts decompressed: 0
         #pkts not compressed: 0, #pkts compr. failed: 0
         #pkts not decompressed: 0, #pkts decompress failed: 0
         #send errors 0, #recv errors 0

          local crypto endpt.: 10.1.0.1, remote crypto endpt.: 10.2.0.2
          path mtu 1500, ip mtu 1500, ip mtu idb FastEthernet0/1
          current outbound spi: 0x46BBC30F(1186710187)

          inbound esp sas:
           spi: 0xBF3FF23B(3208639035)
             transform: esp-aes esp-sha-hmac ,
 --More--
```

Рисунок 3.12 – перша відправка команди show crypto ipsec sa

Тож відправимо команду ping з комп'ютера PC-3 на комп'ютер PC-0 ще один раз як на рис. 3.13.

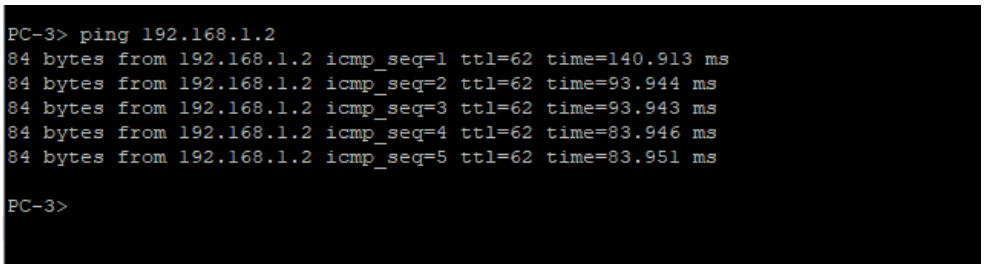

Рисунок 3.13 – Відправка ping команди

Далі знову запускаємо команду show crypto ipsec sa і перевіримо, що значення полів pkts encrypt та pkts decrypt збільшились як на рис. 3.14.

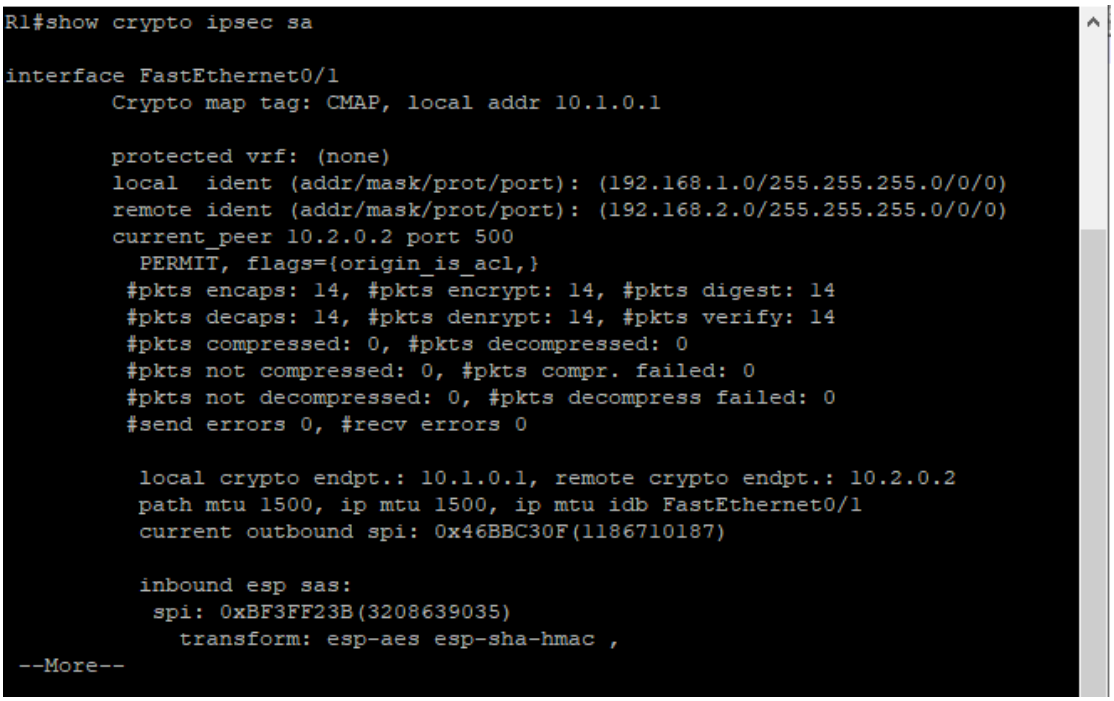

Рисунок 3.14 – друга відправка команди show crypto ipsec sa

Скориставшись розробленою системою можна зробити висновок, що вона спрощує та пришвидшує налаштування VPN мереж, особливо коли таких мереж необхідно зконфігурити декілька десятків в день.

ВИСНОВКИ

У ході роботи було з’ясовано, що VPN мережі стають все популярнішими та необхіднішими у сучасному світі Інтернету, адже саме вони забезпечують конфіденційність та приватність. У роботі було розкрито поняття VPN мережі, її особливості та цілі використання такого роду мереж. Також було розглянуто основну класифікацію віртуальних приватних мереж та її основні складові. Значну увагу було приділено вивченню протоколів VPN мереж, адже вони визначають яким способом мережа буде реалізована.

Було досліджено принципи роботи основних симуляторів мереж, для роботи були використані такі симулятори як: Cisco Packet Tracer та GNS 3. Вони дозволяють конфігурувати мережі, використовуючи маршрутизатори, комутатори та інше необхідне обладнання. В ході дослідження було виявлено, що головним недоліком таких сиимуляторів є незручний графічний інтерфейс, що робить процес налаштування довшим та енергозатратнішим.

У ході роботи був розроблений графічний інтерфейс, який надає можливість конфігурувати VPN мережу за протоколами IPSec та IKEv2. Для генерації необхідно лише ввести IP адресу та маску інтерфейсів роутерів і код для налаштування буде згенерований. Також важливим елементом системи є можливість скопіювати згенерований код на реальне обладнання. Така система допоможе системним адміністраторам проектувати комп'ютерні мережі швидше та з меншою кількістю помилок. Для того, щоб не експериментувати на реальному обладнанні, моделювання можна виконувати в цій системі.

Як висновок можна сказати, що мета кваліфікаційної магістерської роботи була досягнута, поставлені цілі та завдання були виконані. Веб-орієнтована система була розроблена та доводить, що конфігурація VPN мережі може бути не такою складною та довготривалою.

СПИСОК ЛІТЕРАТУРИ

1. Брайко В. В. Дослідження принципів роботи технологій VPN [Електронний ресурс] – https://core.ac.uk/download/pdf/84825462.pdf
2. Блинков Ю. В. Изучение информационных сетей и сетевых технологий на виртуальных машинах / Ю. В. Блинков.–П.: ПГУАС, 2016. – 344 с.
3. Корепанова Н. Л. Защита канала передачи данных на базе технологии VPN в системах экологического мониторинга [Электронный ресурс] / Н. Л. Корепанова, М. А. Лебедева. – Режим доступа: https://msoe.ru/wp-content/uploads/2019/03/33-05.pdf
4. Кулмамиров С. Преимущества протоколов частных VPN сетей [Электронный ресурс] / С. Кулмамиров, Ж. Изтаев, Г. Ордабаева. – Режим доступа: https://pps.kaznu.kz/ru/Main/FileShow2/132973/600/3/16445/2018//
5. Что такое VPN и как это работает: базовое руководство Avast [Электронный ресурс] – https://blog.avast.com/ru/chto-takoe-vpn-i-kak-eto-rabotaet-bazovoe-rukovodstvo-avast
6. Классификация VPN сетей [Электронный ресурс] – https://studbooks.net/2239927/informatika/klassifikatsiya_setey
7. Олифер Н. Компьютерные сети. Принципы, технологии, протоколы / В. Олифер, Н. Олифер. – Спб.: Питер, 2020. – 1008 с.
8. Wang S. P. Computer Architecture and Organization: Fundamentals and Architecture Security / S. P. Wang. – B.: Springer, 2021. – 344 p.
9. Virtual Private Networks and Remote Access [Electronic resource] – https://www.sciencedirect.com/topics/computer-science/extranet
10. Woland A. Integrated Security Technologies and Solutions / A. Woland, V. Santuka, J. Sanbower, C. Mitchell. – I.: Cisco Press, 2019. – 1148 p.

11. Bazzell M. Hiding from the Internet: Eliminating Personal Online Information / M. Bazzell. – Scotts Valley: CreateSpace Independent Publishing Platform, 2016. – 334 p.
12. Vpn Tunnel [Electronic resource] – https://hackercombat.com/ip-protection-in-2019-cope-with-weak-points/vpn-tunnel/
13. Платунова С. М. Реализация комплексной безопасности в корпоративных сетях. Шлюз безопасности как универсальное средство для обеспечения защиты данных и предотвращения вторжений / С. М. Платунова, И. В. Елисеев, Е. Ю. Авксеньева. – СПб.: Университет ИТМО, 2020. – 64 с.
14. VPN issues and technical problems to overcome [Electronic resource] – https://www.techtarget.com/searchnetworking/photostory/4500270545/The-best-VPNs-for-enterprise-use/5/VPN-issues-and-technical-problems-to-overcome
15. Технология построения виртуальной частной сети — протоколы IPSec, SSL [Электронный ресурс] – https://seti.ucoz.ru/index/lekcija_5_tekhnologija_postroenija_virtualnoj_chastnoj_seti_protokoly_ipsec_ssl/0-23
16. Норман Р. Выбираем протокол VPN [Электронный ресурс] / Р. Норман. – Режим доступа: http://www.infocity.kiev.ua/os/content/os206.phtml
17. VPN Protocols – Different Types Compared [Electronic resource] – https://privacycanada.net/best-vpn-protocols/
18. Николахин А. Ю. Использование технологии VPN для обеспечения информационной безопасности [Электронный ресурс] / А. Ю. Николахин. – Режим доступа: http://nirit.org/wp-content/uploads/2018/12/60-68.pdf
19. Keijser J. J. OpenVPN Cookbook / J. J. Keijser. – B.: Packt Publishing, 2017. – 467 p.

20. Пархоменко І. І. Способи захисту каналів корпоративних мереж на базі VPN-рішень [Електронний ресурс] – http://journals.dut.edu.ua/index.php/dataprotect/article/view/1245
21. Обзор Secure Socket Tunneling Protocol [Электронный ресурс] – https://vc.ru/dev/119780-obzor-secure-socket-tunneling-protocol
22. Выбираем протокол для VPN. Сравнение OpenVPN, PPTP, L2TP/IPsec и IPsec IKEv2 [Электронный ресурс] – https://book.cyberyozh.com/ru/vyibiraem-protokol-dlya-vpn-sravnenie-openvpn-pptp-l2tp-ipsec-i-ipsec-ikev2/
23. WireGuard VPN [Электронный ресурс] – https://help.keenetic.com/hc/ru/articles/360010592379-WireGuard-VPN
24. Программное обеспечение виртуальных частных сетей (VPN) [Электронный ресурс] – https://seti.ucoz.ru/index/lekcija_4_programmnoe_obespechenie_virtualnykh_chastnykh_setej_vpn_struktura_vpn_klassifikacija_vpn/0-22
25. Introduction to VPN [Electronic resource] – https://sc1.checkpoint.com/documents/R76/CP_R76_VPN_AdminGuide/13894.htm
26. Ушаков Ю. А. Создание мультисервисной многоточечной VPN сети с динамической автонастройкой [Электронный ресурс] / Ю. А. Ушаков, П. Н. Полежаев, А. Е. Шухман. – Режим доступа: http://vestnik.osu.ru/2015_9/31.pdf
27. Gre туннель mikrotik ipsec [Electronic resource] – https://dudom.ru/kompjutery/gre-tunnel-mikrotik-ipsec/
28. Інструкція з використання послуги VPN в хмарі GigaCloud [Електронний ресурс] – https://tqm.com.ua/ua/likbez/chapy/vykorystannja-poslugy-vpn-v-hmari-gigaCloud
29. Колесник В. В. Огляд програмних емуляторів та симуляторів для побудови працездатних моделей мережі [Електронний ресурс] / В. В. Колесник, Т. А.

Вакалюк. – Режим доступу: https://conf.ztu.edu.ua/wp-content/uploads/2021/01/7.pdf

30. Золотухин М. С. Сетевые симуляторы и эмуляторы оборудования CISCO [Электронный ресурс] / М. С. Золотухин, Е. С. Симонова. – Режим доступа: https://top-technologies.ru/ru/article/view?id=38134
31. Краткое описание симулятора EVE-NG [Электронный ресурс] – https://www.codetd.com/ru/article/11864623
32. An Introduction to JavaScript [Electronic resource] – https://javascript.info/intro#summary

ДОДАТОК

Додаток А

```
<!DOCTYPE html>
<html><head><title>VPN Configuration</title>

<script src="jquery-3.6.0.min.js" type="text/javascript"></script>
<script src="clipboard.js"></script>
<script src='ip-subnet-calculator.js'></script>

<style>
.block {
display: none;
}

.main-img {
    position: absolute;
}

.content {
    margin-top: 260px;
    text-align: center;
    width: 1000px;
}

.left-content {
width: 70%;
float: left;
}

.right-content{
    position: absolute;
    left: 1565px;
}

.blow .router{
    display: none;
}

.router{
        width: 50px;
        position: absolute;
        cursor: pointer;
}

.configs-block{
    margin-left: calc(100px + 100px);
}

.ip-label {
    float: left;
    font-size: 10pt;
    width: 40px;
    text-transform: uppercase;
    padding-top: 7px;
}
```

```
.ip-mask{
    position: absolute;
    background: linear-gradient(to right, #9eeeef 0%, rgba(87, 255, 66, 0.92) 51%, rgb(158, 239, 225) 100%);
    box-shadow: 0 0 0 4px rgba(173, 220, 221, 0.17), 2px 1px 6px 4px rgba(10, 10, 0, 0.5);
}

.ip-mask input {
    margin: 4px;
    width: 101px;
}

.ip-mask input.error{
    border-color: red;
    color: red;
}

.ip-mask input.success{
    border-color: green;
    color: green;
}

.code-block {
    margin-top: 10px;
    margin-left: 470px;
    text-align: left;
}

.r1 .ip-mask-1{
    top: 221px;
    left: calc(100px + 715px);
}
.r1 .ip-mask-2{
    top: 169px;
    left: calc(100px + 951px);
}

.r2 .ip-mask-1{
    top: 230px;
    left: calc(100px + 437px);
}
.r2 .ip-mask-2{
    top: 154px;
    left: calc(100px + 199px);
}

img.router.r1 {
    left: calc(100px + 885px);
    top: 219px;
}

img.router.r2 {
    left: calc(100px + 372px);
    top: 201px;
```

```
}

.generated-code {
    background: rgba(124, 222, 205, 0.27);
    width: 560px;
    border-radius: 25px;
    border: 3px solid;
    padding: 10px;
    border-color: #9defec;
}

.ctrl-btn {
    cursor: pointer;
    width: 140px;
    text-decoration: none;
    font-weight: bold;
    display: inline-block;
    color: white;
    padding: 20px 30px;
    margin: 10px 5px;
    border-radius: 10px;
    font-family: 'Montserrat', sans-serif;
    text-transform: uppercase;
    letter-spacing: 2px;
    background-image: linear-gradient(to right, #9eeeef 0%, rgba(87, 255, 66, 0.92) 51%, rgb(158, 239, 225) 100%);
    background-size: 200% auto;
    box-shadow: 0 0 20px rgba(0,0,0,.1);
    transition: .5s;
}
.ctrl-btn:hover {background-position: right center;}

.radio-buttons{
    margin-left: 30px;
}

input[type="radio"] {
  position: absolute;
  opacity: 0;
  z-index: -1;
}
.radio-buttons label {
  position: relative;
  display: inline-block;
  margin-right: 10px;
  margin-bottom: 10px;
  padding-left: 30px;
  padding-right: 10px;
  line-height: 36px;
  cursor: pointer;
}
```

```
.radio-buttons label::before {
  content: " ";
  position: absolute;
  top: 6px;
  left: 0;
  display: block;
  width: 24px;
  height: 24px;
  border: 2px solid rgb(157, 239, 236);
  border-radius: 4px;
  z-index: -1;
}
.radio-buttons input[type="radio"] + label::before {
  border-radius: 18px;
}
/* Checked */
.radio-buttons input[type="radio"]:checked + label {
  padding-left: 10px;
  color: #fff;
}

.radio-buttons input[type="radio"]:checked + label::before {
  top: 0;
  width: 100%;
  height: 100%;
  background: rgb(67, 247, 112) 51%;
}
/* Transition */
.radio-buttons label,
.radio-buttons label::before {
  -webkit-transition: .25s all ease;
  -o-transition: .25s all ease;
  transition: .25s all ease;
}

</style></head>

<body style="color: rgb(0, 0, 0); background-color: rgb(244, 255, 255);" alink="#000099" link="#000099" vlink="#990099" width="1000px">

<div class="schema">
        <img class="main-img" alt="sheme" style="width: 1516px; height: 456px;" src="base.png"><br>
    <div class="routers">
        <div class="regular">
            <img src="router-S.png" class="router r1">
            <img src="router-S.png" class="router r2">
        </div>
        <div class="blow">
            <img src="router.png" class="router r1">
            <img src="router.png" class="router r2">
        </div>
    </div>
</div>
```

```
<div class="right-content">
    <input value="generate" class="generete ctrl-btn" type="button">
    <button value="copy" class="copy-btn ctrl-btn" data-clipboard-target="#generated-code-for-copy">Copy</button>
    <div class="radio-buttons">
        <div>
            <input type="radio" name="net-type" id="rb1" value="IPSec" checked/>
            <label for="rb1">IPSec</label>
        </div>
        <div>
            <input type="radio" name="net-type" id="rb2" value="IKEv2"/>
            <label for="rb2">IKEv2</label>
        </div>
    </div>

</div>

<div class="content">

<div class="left-content">
    <div class="inputs-block">
        <div class="block r1">

            <div class="configs-block">
                <div class="ip-mask ip-mask-1">
                    <div><label class="ip-label">ip: </label><input placeholder="xxx.xxx.xxx.xxx" class="ip" type="text"></div>
                    <div><label class="ip-label">mask: </label><input placeholder="xxx.xxx.xxx.xxx" class="mask" type="text"></div>
                </div>
                <div class="ip-mask ip-mask-2">
                    <div><label class="ip-label">ip: </label><input placeholder="xxx.xxx.xxx.xxx" class="ip" type="text"></div>
                    <div><label class="ip-label">mask: </label><input placeholder="xxx.xxx.xxx.xxx" class="mask" type="text"></div>
                </div>
            </div>
        </div>


        <div class="block r2">

            <div class="configs-block">

                <div class="ip-mask ip-mask-1">
                        <div><label class="ip-label">ip: </label><input placeholder="xxx.xxx.xxx.xxx" class="ip" type="text"></div>
                        <div><label class="ip-label">mask: </label><input placeholder="xxx.xxx.xxx.xxx" class="mask" type="text"></div>
                </div>
                <div class="ip-mask ip-mask-2">
                        <div><label class="ip-label">ip: </label><input placeholder="xxx.xxx.xxx.xxx" class="ip" type="text"></div>
                        <div><label class="ip-label">mask: </label><input placeholder="xxx.xxx.xxx.xxx" class="mask" type="text"></div>
                </div>
            </div>
        </div>
    </div>
    <div class="code-block" style="display: none;">
        <div class="generated-code" id="generated-code">
```

```
        </div>
        <textarea id="generated-code-for-copy" style="position:  absolute;top: 0;right: 9000px;"></textarea>
    </div>
</div>

</div>

<script>

var clipboard = new ClipboardJS('.copy-btn');


$('.blow .router').click(function(e){
    var classes = $(e.currentTarget)[0].classList
    var tabClass = '.'+classes[1]

    hideAllBlocks();
    hideAllRouters();
});

$('.regular .router').click(function(e){
    $('.right-content').show();

    var classes = $(e.currentTarget)[0].classList
    var tabClass = '.'+classes[1]

    hideAllBlocks();
    hideAllRouters();

    showBlock(tabClass);
    showRouter(tabClass);
});

function showBlock(block){

$(block).css('display', 'block')
$(block).addClass('active')
}

function showRouter(router){
    $('.blow '+router).css('display', 'block')

    showCodeBlockIfNeeded();
}

function hideAllRouters(){
    $('.blow .router').css('display', 'none')
}

function hideAllBlocks(){
$('.block').css('display', 'none')
$('.block').removeClass('active')
}
```

```
$('.generete').click(showCodeBlockIfNeeded)

function showCodeBlockIfNeeded(){
    var isValid = validateIPaddress();

    if (isValid) {
        showCodeBlock();
    } else {
        hideCodeBlock();
    }
}

function inputData(){
    var ipMasksBlocks = $('.r1 .ip-mask');
    var wroteIPs = []

    wroteIPs.push({'l-ip-in': $('.r2 .ip-mask-2 ').find('.ip').val(), 'l-mask-in':$('.r2 .ip-mask-2 ').find('.mask').val(),
    'l-network-in': IpSubnetCalculator.calculateCIDRPrefix( $('.r2 .ip-mask-2 ').find('.ip').val(), $('.r2 .ip-mask-2 ').find('.mask').val() ).ipLowStr,
    'l-invertMask-in': IpSubnetCalculator.calculateCIDRPrefix( $('.r2 .ip-mask-2 ').find('.ip').val(), $('.r2 .ip-mask-2 ').find('.mask').val() ).invertedMaskStr});
    wroteIPs.push({'l-ip-out': $('.r2 .ip-mask-1 ').find('.ip').val(), 'l-mask-out':$('.r2 .ip-mask-1 ').find('.mask').val(),
    'l-network-out': IpSubnetCalculator.calculateCIDRPrefix( $('.r2 .ip-mask-1 ').find('.ip').val(), $('.r2 .ip-mask-1 ').find('.mask').val() ).ipLowStr,
    'l-invertMask-out': IpSubnetCalculator.calculateCIDRPrefix( $('.r2 .ip-mask-1 ').find('.ip').val(), $('.r2 .ip-mask-1 ').find('.mask').val() ).invertedMaskStr});

    wroteIPs.push({'r-ip-in': $('.r1 .ip-mask-2 ').find('.ip').val(), 'r-mask-in':$('.r1 .ip-mask-2 ').find('.mask').val(),
    'r-network-in': IpSubnetCalculator.calculateCIDRPrefix( $('.r1 .ip-mask-2 ').find('.ip').val(), $('.r1 .ip-mask-2 ').find('.mask').val() ).ipLowStr,
    'r-invertMask-in': IpSubnetCalculator.calculateCIDRPrefix( $('.r1 .ip-mask-2 ').find('.ip').val(), $('.r1 .ip-mask-2 ').find('.mask').val() ).invertedMaskStr});
    wroteIPs.push({'r-ip-out': $('.r1 .ip-mask-1 ').find('.ip').val(), 'r-mask-out':$('.r1 .ip-mask-1 ').find('.mask').val(),
    'r-network-out': IpSubnetCalculator.calculateCIDRPrefix( $('.r1 .ip-mask-1 ').find('.ip').val(), $('.r1 .ip-mask-1 ').find('.mask').val() ).ipLowStr,
    'r-invertMask-out': IpSubnetCalculator.calculateCIDRPrefix( $('.r1 .ip-mask-1 ').find('.ip').val(), $('.r1 .ip-mask-1 ').find('.mask').val() ).invertedMaskStr});

    return wroteIPs;
}

 function showCodeBlock()  {
    $('.code-block').show();
    generateCode();
 }

 function hideCodeBlock()  {
    $('.code-block').hide();
 }
```

```
function validateIPaddress()  {
        var ipformat = /^(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$/;

        var inputs = $('.active .ip-mask input')

        $('.error-title').hide();

        var isValid = inputs.length > 0;

        inputs.each(function (i, el){

            $(el).removeClass('error');
            $(el).removeClass('success');

            if (!$(el).val().match(ipformat)){
                isValid = false;
                $('.error-title').show();
                console.log($(el).val()+"dont valid")
                $(el).addClass('error');
            } else {
                $(el).addClass('success');
            }
        });
        return isValid;
}
```

```
var codes = {
    "r1": {
        'IPSec': [
            "Enable",
            "Conf term",
            "int f0/1",
            "ip nat outside",
            "int f0/0",
            "ip nat inside",
            "exit",
            "ip access-list extended FOR-NAT",
            "deny ip __r-network-in__ __r-invertMask-in__ __l-network-in__ __l-invertMask-in__",
            "permit ip __r-network-in__ __r-invertMask-in__ any",
            "exit",
            "ip nat inside source list FOR-NAT interface f0/1 overload",
            "do write",
            "crypto isakmp policy 1",
            "encryption 3des",
            "hash md5",
            "authentication pre-share",
            "group 2",
            "exit",
            "crypto isakmp key cisco address __l-ip-out__",
            "crypto ipsec transform-set tunnel esp-3des esp-md5-hmac",
            "ip access-list extended  FOR-VPN",
            "permit ip __r-network-in__ __r-invertMask-in__ __l-network-in__ __l-invertMask-in__",
            "exit",
            "crypto map CMAP 10 ipsec-isakmp",
            "set peer __l-ip-out__",
            "set transform-set tunnel",
            "match address FOR-VPN",
            "exit",
            "int f0/1",
            "crypto map CMAP",
            "exit",
            "do write",
            "exit"
```

```
    ],
    'IKEv2' : [
        "hostname ROUTER-A",
        "crypto ikev2 proposal IKEv2_PROPOSAL",
        "encryption aes-cbc-256",
        "integrity sha512",
        "group 5",
        "crypto ikev2 policy IKEv2_POLICY",
        "proposal IKEv2_PROPOSAL",
        "crypto ikev2 keyring IKEv2_KEYRING",
        "peer ROUTER-B",
        "address __l-ip-out__",
        "pre-shared-key local keya-b",
        "pre-shared-key remote keyb-a",
        "crypto ikev2 profile IKEv2_PROFILE",
        "match identity remote address __l-ip-out__ __l-mask-out__",
        "identity local address __r-ip-out__",
        "authentication remote pre-share",
        "authentication local pre-share",
        "keyring local IKEv2_KEYRING",
        "crypto ipsec transform-set IPSEC_TSET1 esp-aes 256 esp-sha-hmac",
        "crypto map IKEv2_MAP 1000 ipsec-isakmp",
        "set peer __l-ip-out__",
        "set transform-set IPSEC_TSET1",
        "set ikev2-profile IKEv2_PROFILE",
        "match address COMPANY_A_B_CRYPTO",
        "interface FastEthernet0/0",
        "ip address __r-ip-in__ __r-mask-in__",
        "no shutdown",
        "interface FastEthernet2/0",
        "ip address __r-ip-out__ __r-mask-out__",
        "crypto map IKEv2_MAP",
        "no shutdown",
        "ip route __r-network-in__ __r-invertMask-in__ __l-ip-out__",
        "ip access-list extended COMPANY_A_B_CRYPTO",
        "permit ip __r-network-in__ __r-invertMask-in__ __l-network-in__ __l-invertMask-in__",
        "end"
```

```
    ]},
"r2": {
    'IPSec': [
        "Enable",
        "Conf term",
        "int f0/1",
        "ip nat outside",
        "int f0/0",
        "ip nat inside",
        "exit",
        "ip access-list extended FOR-NAT",
        "deny ip __l-network-in__ __l-invertMask-in__ __r-network-in__ __r-invertMask-in__",
        "permit ip __l-network-in__ __l-invertMask-in__ any",
        "exit",
        "ip nat inside source list FOR-NAT interface f0/1 overload",
        "do write",
        "crypto isakmp policy 1",
        "encryption 3des",
        "hash md5",
        "authentication pre-share",
        "group 2",
        "exit",
        "crypto isakmp key cisco address __r-ip-out__",
        "crypto ipsec transform-set tunnel esp-3des esp-md5-hmac",
        "ip access-list extended  FOR-VPN",
        "permit ip __l-network-in__ __l-invertMask-in__ __r-network-in__ __r-invertMask-in__",
        "exit",
        "crypto map CMAP 10 ipsec-isakmp",
        "set peer __r-ip-out__",
        "set transform-set tunnel",
        "match address FOR-VPN",
        "exit",
        "int f0/1",
        "crypto map CMAP",
        "exit",
        "do write",
        "exit"
```

```
        ],
        'IKEv2' : [
            "hostname ROUTER-B",
            "crypto ikev2 proposal IKEv2_PROPOSAL",
            "encryption aes-cbc-256",
            "integrity sha512",
            "group 5",
            "crypto ikev2 policy IKEv2_POLICY",
            "proposal IKEv2_PROPOSAL",
            "crypto ikev2 keyring IKEv2_KEYRING",
            "peer ROUTER-A",
            "address __r-ip-out__",
            "pre-shared-key local keya-b",
            "pre-shared-key remote keyb-a",
            "crypto ikev2 profile IKEv2_PROFILE",
            "match identity remote address __r-ip-out__ __r-mask-out__",
            "identity local address __l-ip-out__",
            "authentication remote pre-share",
            "authentication local pre-share",
            "keyring local IKEv2_KEYRING",
            "crypto ipsec transform-set IPSEC_TSET1 esp-aes 256 esp-sha-hmac",
            "crypto map IKEv2_MAP 1000 ipsec-isakmp",
            "set peer __r-ip-out__",
            "set transform-set IPSEC_TSET1",
            "set ikev2-profile IKEv2_PROFILE",
            "match address COMPANY_A_B_CRYPTO",
            "interface FastEthernet0/0",
            "ip address __l-ip-in__ __l-mask-in__",
            "no shutdown",
            "interface FastEthernet2/0",
            "ip address __l-ip-out__ __l-mask-out__",
            "crypto map IKEv2_MAP",
            "no shutdown",
            "ip route __l-network-in__ __l-invertMask-in__ __r-ip-out__",
            "ip access-list extended COMPANY_A_B_CRYPTO",
            "permit ip __l-network-in__ __l-invertMask-in__ __r-network-in__ __r-invertMask-in__",
            "end"
        ]}
}
```

```
}

function generateCode(){
    var data = inputData()
    var dataIndex = -1
    var netType = $('[name="net-type"]:checked').val()
    var tabCodes = codes[$('.block.active')[0].className.split(' ')[1]][netType]

    var codesWithData = ""

    for (var i in tabCodes){
        var val = tabCodes[i]

        for (var dat in data) {
            for (var key in data[dat]) {
                val = val.replace('__'+key+'__', data[dat][key])
            }
        }
        codesWithData +=val + '<br>'
    }

    $('.generated-code')[0].innerHTML = codesWithData
    $('#generated-code-for-copy')[0].value = codesWithData.replace(/<br>/g, '\n')
}

function getNetwork(ip, mask){
    var result = ""

    var ipNumbers = ip.split('.')
    var maskNumbers = mask.split('.')

    for (var i in ipNumbers){
        var val = ipNumbers[i]

        if (maskNumbers[i] === "0"){
            val = "0"
        }
        if (i != 3){
            val += '.'
        }
        result += val
    }
    return result
}
```

```
function invertMask(mask){
 var list = mask.split('.')
 for (var i in list){
  if (list[i] == '0'){
   list[i] = '255'
  } else if (list[i] == '255'){
   list[i] = '0'
  }
 }
 return list.join('.')
}

function getIpNetwork(ip, mask){
    var ipList = ip.split('.')
    var maskList = mask.split('.')

    for (var i in maskList){
        if (maskList[i] == '0'){
            ipList[i] = '0'
        }
    }

    return ipList.join('.')
}
</script>
</body>
</html>
```