МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ПрАТ «ПВНЗ «ЗАПОРІЗЬКИЙ ІНСТИТУТ ЕКОНОМІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ»

Кафедра інформаційних технологій

ДО ЗАХИСТУ ДОПУЩЕНА

Зав. кафедри _____

д.е.н., доц. С.І. Левицький

МАГІСТЕРСЬКА ДИПЛОМНА РОБОТА

ПОРІВНЯЛЬНИЙ АНАЛІЗ ПРОТОКОЛІВ ВІРТУАЛЬНИХ ПРИВАТНИХ МЕРЕЖ

Виконав

ст.гр. КІ–111м _____ М.О. Капустін

Науковий керівник

доцент _____ Н.Р. Полуектова

Запоріжжя

2023 р.

ПРАТ «ПВНЗ «ЗАПОРІЗЬКИЙ ІНСТИТУТ ЕКОНОМІКИ
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ»
Кафедра інформаційних технологій

ЗАТВЕРДЖУЮ
Зав. кафедри
д.е.н., доцент Левицький С.І.
_____
03.10.2022 р.

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ ДИПЛОМНУ РОБОТУ
студента гр. КІ–111м, спеціальності 123 «Комп'ютерна інженерія» ОП «Комп'ютерна інженерія»
Капустіна Микити Олександровича

1.Тема: Порівняльний аналіз протоколів віртуальних приватних мереж
затверджена наказом по інституту № 02-16 від 03.10.2022 р.
2. Термін здачі студентом закінченої роботи: 12.01.2023 р.
3. Перелік питань, що підлягають розробці
1. Провести огляд предметної області, ознайомитися з літературою та інтернет-джерелами, що присвячені тематиці роботи
2. Визначити перелік протоколів віртуальних приватних мереж для порівняльного аналізу
3. Визначити .метрики та інструменти для вимірювання продуктивності
4. Провести огляд програмно-апаратного забезпечення, що дозволить виконати дослідження характеристик окремих протоколів в різних умовах
5. Провести експериментальне дослідження продуктивності протоколів VPN

6. Проаналізувати результати експерименту і порівняти продуктивність різних рішень, у різних середовищах та умовах застосування

7. Обґрунтувати рекомендації з вибору кращого протоколу

8. Оформити звіт за результатами роботи

4. Календарний графік підготовки магістерської дипломної роботи

№ етапу	Зміст	Терміни виконання	Готовність по графіку %, підпис керівника	Підпис керівника про повну готовність етапу, дата
1.	Формулювання теми магістерської дипломної роботи (збір практичного матеріалу за темою магістерської дипломної роботи)	20.10.22		
2.	І атестація І розділ магістерської дипломної роботи	27.10.22		
3.	ІІ атестація ІІ розділ магістерської дипломної роботи	17.11.22		
4.	ІІІ атестація ІІІ та IV розділ магістерської дипломної роботи, висновки та рекомендації, додатки, реферат, перевірка програмою «Антиплагіат»	29.12.22		
5.	Доопрацювання магістерської дипломної роботи, підготовка презентації, отримання відгуку керівника і рецензії	10.01.23		
6.	Попередній захист магістерської дипломної роботи	13.01.23		
7.	Подача магістерської дипломної роботи на кафедру	За 3 дні до		
8.	Захист магістерської дипломної роботи	20.01.23		

Дата видачі завдання: 03.10.2022 р.

Керівник магістерської роботи _____ Н.Р. Полуектова
(підпис) (прізвище та ініціали)

Завдання отримав до виконання _____ М.О. Капустін
(підпис студента) (прізвище та ініціали)

РЕФЕРАТ

Магістерська робота містить: 85 сторінок, 18 рисунків, 11 таблиць, 60 першоджерел та 3 додатки.

Об’єктом дослідження є протоколи віртуальних приватних мереж.

Предметом дослідження є продуктивність рішень віртуальних приватних мереж в умовах стабільного та ненадійного мережевого з’єднання.

Метою роботи є аналіз і порівняння продуктивності сучасних рішень віртуальних приватних мереж в умовах стабільного та ненадійного мережевого з’єднання, а також розробка рекомендацій для вибору найліпшого рішення.

Для досягнення мети роботи було поставлено такі завдання: дослідити проблему вибору протоколу віртуальної приватної мережі; визначити перелік протоколів для порівняльного аналізу; визначити метрики та інструменти для вимірювання продуктивності; провести експериментальне дослідження продуктивності; проаналізувати результати експерименту і порівняти продуктивність різних рішень, у різних середовищах та умовах застосування; обґрунтувати рекомендації з вибору найліпшого протоколу.

Для виконання поставлених завдань було застосовано такі методи дослідження: експеримент, вимірювання та порівняння.

Отримані результати можуть бути корисними мережевим адміністраторам для оцінки варіантів рішень при розгортанні віртуальних приватних мереж і забезпеченні віддаленого доступу, зокрема в умовах ненадійності мережевого з’єднання.

IPERF, IPSEC, L2TP, OPENVPN, VPN, WIREGUARD, ВІДДАЛЕНИЙ ДОСТУП, НЕНАДІЙНІСТЬ, ПРОДУКТИВНІСТЬ, ПРОПУСКНА ЗДАТНІСТЬ, ПРОТОКОЛ

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

Слово / словосполучення	Скорочення	Умови використання
A		
Advanced Encryption Standard	AES	В тексті
Authentication Header	AH	В тексті
B		
Bring Your Own Device	BYOD	В тексті
C		
Certification Authority	CA	В тексті
D		
Dynamic Host Configuration Protocol	DHCP	В тексті
E		
Encapsulating Security Payload	ESP	В тексті
H		
Hash-based Message Authentication Code	HMAC	В тексті
L		
Layer 2 Tunneling Protocol	L2TP	В тексті
N		
National Institute of Standards and Technology	NIST	В тексті
National Security Agency	NSA	В тексті
Network Access Server	NAS	В тексті
Network Address Translation	NAT	В тексті
Network Interface Controller	NIC	В тексті
Network Tunnel	TUN	В тексті
P		
Perfect Forward Secrecy	PFS	В тексті
Point-to-Point Tunneling Protocol	PPTP	В тексті
Слово / словосполучення	Скорочення	Умови

		використання
Pre-Shared Key	PSK	В тексті
S		
Secure Shell	SSH	В тексті
Secure Socket Tunneling Protocol	SSTP	В тексті
Security Association	SA	В тексті
T		
Transmission Control Protocol	TCP	В тексті
Transport Layer Security	TLS	В тексті
U		
User Datagram Protocol	UDP	В тексті
V		
Virtual Private Network	VPN	В тексті
Voice over IP	VoIP	В тексті
Z		
Zero Trust Network Access	ZTNA	В тексті

ВСТУП

У сучасному світі віддалена робоча сила невпинно зростає, тому забезпечення надійних технологій віддаленого доступу для працівників є як ніколи актуальним. Одним з традиційних рішень віддаленого доступу є віртуальні приватні мережі, і більшість підприємств упродовж багатьох років використовують саме це рішення для надання віддаленого доступу до своїх корпоративних мереж. Водночас мережеві адміністратори стикаються з проблемою вибору протоколу віртуальної приватної мережі, що є підґрунтям для її розгортання.

Існує низка протоколів, що пропонують різні можливості налаштування та безпеки, а також по-різному впливають на продуктивність віртуальної приватної мережі. Відсутність чітких рекомендацій або алгоритму вибору ускладнює вибір протоколу, і вимагає від мережевого адміністратора додаткового дослідження протоколів. У відкритих джерелах часто міститься інформація про застарілі та потенційно скомпрометовані протоколи, натомість дослідження сучасних протоколів зустрічається значно рідше.

До того ж, завдяки мобільним пристроям віддалені працівники у наш час можуть працювати з будь-якої точки світу, де є можливість підключення до мережі Інтернет. Це призводить до того, що віддалений доступ часто забезпечується в умовах ненадійного мережевого з’єднання. Тому важливо розуміти, як відрізняється продуктивність віртуальних приватних мереж, розгорнутих на базі різних протоколів, у ненадійному мережевому з’єднанні, і це є актуальним напрямком дослідження.

Метою роботи є аналіз і порівняння продуктивності сучасних рішень віртуальних приватних мереж в умовах стабільного та ненадійного мережевого з’єднання, а також розробка рекомендацій для вибору найліпшого рішення.

Для досягнення мети роботи було поставлено такі завдання:

– дослідити проблему вибору протоколу віртуальної приватної мережі;
– визначити перелік протоколів для порівняльного аналізу;
– визначити метрики та інструменти для вимірювання продуктивності;
– провести експериментальне дослідження продуктивності;
– проаналізувати результати експерименту і порівняти продуктивність різних рішень, у різних середовищах та умовах застосування;
– обґрунтувати рекомендації з вибору найліпшого протоколу.

Для виконання поставлених завдань було застосовано такі методи дослідження:

– метод експерименту, що дозволив дослідити продуктивність різних рішень шляхом налаштування експериментальної установки та проведення у ній необхідних вимірювань;
– метод вимірювання, що використовувався для фіксування значень показників продуктивності мережі;
– метод порівняння, що був корисним під час аналізу результатів вимірювань, для визначення найпродуктивнішого рішення в різних випадках застосування.

Об’єктом дослідження є протоколи віртуальних приватних мереж.

Предметом дослідження є продуктивність рішень віртуальних приватних мереж в умовах стабільного та ненадійного мережевого з’єднання.

Вплив рішень віртуальних приватних мереж на продуктивність мережі досліджується у роботах Ш. Нарайяна, К. Брукінга та С. де Вере [11], а також К. Навея та Ш. Ду [12]. Порівняння рішень з погляду продуктивності наводиться у роботах А. Абдулазіза, Б. Саліма, Д. Зібарі та Д. Доградмачі [17], а також Л. Оссвальда, М. Хеберле та М. Мента [18]. Вимірювання продуктивності мережі за умов ненадійності мережевого з’єднання виконується у роботах Д. Брассіла, Р. Макгіра, Р. Раджагопалана, Е. Бав’єра, Л. Робертса, Б. Марка та С. Шваба [58], а також Д. Коула та В. Тейна [59].

Проте дослідження, у якому б порівнювалась продуктивність рішень віртуальних приватних мереж в умовах стабільного та ненадійного мережевого з’єднання, знайдено не було.

Наукову новизну роботи складають: отримані експериментальним шляхом метрики продуктивності рішень віртуальних приватних мереж в умовах стабільного та ненадійного мережевого з’єднання; висновки про порівняння результатів різних рішень, в різних випадках застосування; обґрунтовані рекомендації з вибору найліпшого рішення.

Структура кваліфікаційної роботи містить: вступ, в якому описується об’єкт і конкретизується предмет дослідження, визначаються мета і комплексне завдання магістерської роботи. У першому розділі досліджується проблема вибору протоколу віртуальної приватної мережі, а також визначається перелік протоколів, що підлягають порівняльному аналізу. У другому розділі надається порівняльна характеристика обраних протоколів, описується методологія та програмно-апаратний комплекс дослідження, наводиться огляд експерименту та його результати. У третьому розділі докладно аналізуються та порівнюються результати експерименту, обґрунтовуються рекомендації з вибору найліпшого протоколу, окреслюються рекомендовані напрямки майбутніх досліджень.

Практичне значення отриманих результатів полягає в можливості визначити найпродуктивніше рішення віртуальних приватних мереж для конкретних середовищ та умов застосування. Отримані результати можуть бути корисними мережевим адміністраторам для оцінки варіантів рішень при розгортанні віртуальних приватних мереж і забезпеченні віддаленого доступу, зокрема в умовах ненадійності мережевого з’єднання.

Апробація. Основні положення магістерської роботи доповідалися на XXIV науковій конференції в Запорізькому інституті економіки та інформаційних технологій в межах секції «Інформаційні технології», і були опубліковані у збірнику тез конференції.

РОЗДІЛ 1
ДОСЛІДЖЕННЯ ПРОБЛЕМИ ВИБОРУ ПРОТОКОЛУ VPN

1.1 Проблеми забезпечення віддаленого доступу до мереж

Щороку кількість віддалених працівників у всьому світі стрімко зростає. Згідно з дослідженням Flexjobs та Global Workplace Analytics за останні 12 років віддалена робоча сила зросла на 159% (Рисунок 1.1). [1]

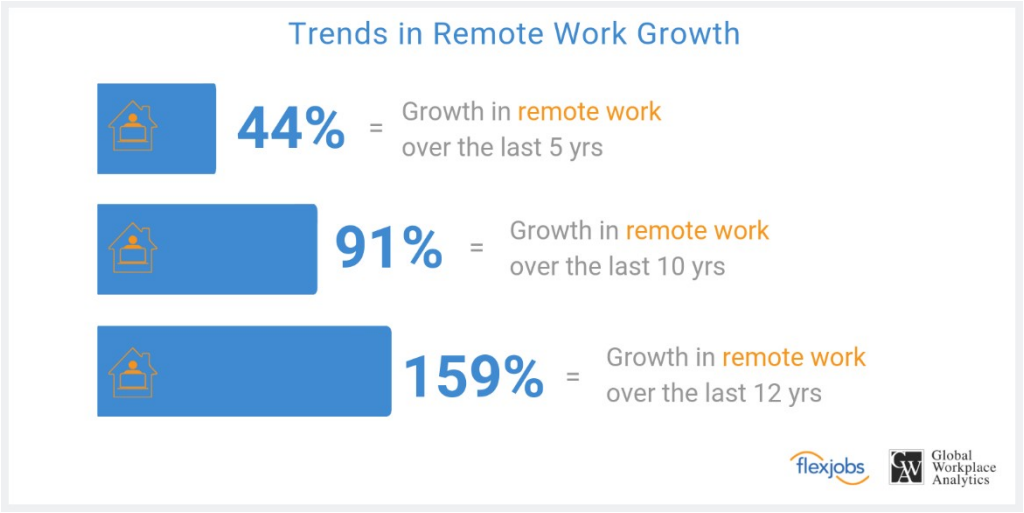


Рис. 1.1 – Тенденція до зростання віддаленої робочої сили

Це дослідження охоплює співробітників, які працюють повністю дистанційно, і співробітників, які працюють у гібридному форматі, поєднуючи дистанційну роботу з роботою в офісі. Дистанційний формат роботи приваблює гнучким тайм-менеджментом, можливістю створити індивідуальне робоче середовище, збалансованістю роботи і особистого життя. Такі переваги позитивно впливають на психічне здоров’я працівників і підвищують їх продуктивність, про що свідчить статистика зі згаданого раніше дослідження.

У 2020 році, через пандемію Covid-19, віддалена робота стала не просто зручним варіантом для деяких працівників, а необхідністю для всіх. З міркувань безпеки підприємства мусили переводити своїх співробітників на

віддалену роботу, і забезпечення надійних технологій віддаленого доступу для працівників стало критично важливим, навіть для тих підприємств, які раніше скептично ставились до роботи за межами офісу. Переконавшись у перевагах віддаленої роботи, багато з них залишили своїх співробітників працювати дистанційно й після поліпшення ситуації у країнах, де розташовані їхні офіси.

Хоча віддалена робота має багато переваг, таких як гнучкість організації роботи та безперервність робочих процесів, сам процес забезпечення віддаленого доступу до мережі компанії пов’язаний із багатьма викликами. Віддалені співробітники отримують доступ до даних і програм компанії за межами корпоративної мережі, в результаті чого вони піддаються багатьом ризикам безпеки, а також наражають дані та системи своїх роботодавців на такі ризики. Рішення віддаленого доступу оптимізують те, як компанії надають доступ до даних і програм віддаленим співробітникам, але розгортання таких систем і їх обслуговування може виявитися складним.

1.1.1 Виклики віддаленого доступу

І. Парецький, директор з маркетингу Ericom Software, у своїй статті «Рішення віддаленого доступу: подолання викликів» наводить деякі поширені проблеми, пов’язані із забезпеченням віддаленого доступу до мережі компанії, і способи їх вирішення. [2]

Системи віддаленого доступу можуть складатися з кількох рівнів: фізичних серверів або апаратних пристроїв, адміністративного програмного забезпечення та програмних клієнтів для кінцевих користувачів. Усі рівні повинні взаємодіяти і працювати злагоджено. Система також може передбачати гібридне налаштування, що включає як локальні, так і хмарні машини з різними операційними системами. Для розгортання і керування такою системою можуть знадобитись значні витрати часу і зусиль.

Вже налаштовану систему необхідно обслуговувати та адмініструвати. Компанії знадобляться спеціалісти, які будуть вирішувати проблеми з усуненням несправностей і підключенням, виправляти і оновлювати систему, а також виконувати інші поточні завдання з обслуговування. Доведеться проводити навчання, щоб внутрішній персонал міг налаштовувати та підтримувати рішення. Це призведе до додаткових експлуатаційних витрат. У разі, якщо компанія не забезпечить належну підтримку системи, це призведе до неочікуваних збоїв у доступі, простою під час періодів обслуговування та трудомістких зусиль в усуненні несправностей, коли щось піде не так.

Віддалена робоча сила стрімко зростає, тому можливість легкого та швидкого масштабування системи віддаленого доступу стала однією з головних проблем для великих і малих компаній. Масштабування може передбачати встановлення нових серверів або розширення мережевої інфраструктури, інсталяцію програмного забезпечення для керування доступом. IT-персонал має бути навчений виконувати та координувати такі завдання. Надійна система керування не повинна захлинутися при розгортанні до корпоративного рівня, вона має бути здатна масштабуватися до такої кількості клієнтських пристроїв, яка знадобиться компанії.

Рішення віддаленого доступу зазвичай представлені в одному з двох варіантів: їм або бракує функцій, необхідних для надійної підтримки корпоративних розгортань, або це рішення, які вражають своєю складністю. Клієнтські системи керування віддаленим доступом можуть мати проблеми з дозволами, несумісність з іншими програмами чи робочими процесами. Це може призвести до втрати продуктивного часу через обхідні шляхи, перенавчання або проблеми з доступом до необхідної функціональності. Останнє, чого бажає компанія, щоб її система віддаленого доступу була перешкодою для роботи кінцевих користувачів.

Система віддаленого доступу повинна гарантувати, що доступ до даних, які потрібні користувачам, є повністю безпечним. Якщо система має

пряме з’єднання з Інтернетом, ризики збільшуються. Прямий доступ до Інтернету значно розширює поверхню атаки на мережу компанії. Якщо віддалені співробітники отримують доступ до ресурсів компанії через особисті пристрої, це також ризик. Компанія не має можливості вжити всіх необхідних заходів безпеки на пристроях, якими повністю не керує. В системах віддаленого доступу з програмними клієнтами на пристрої необхідні регулярні оновлення для захисту клієнтів від вразливостей та експлойтів. Застарілі клієнти віддаленого доступу, без виправлень, це відкриті мішені для зловмисників, в разі їх злому дані та системи компанії можуть піддаватись кібератакам.

1.1.2 Застосування традиційних рішень

Віддалене робоче середовище є особливо привабливим для зловмисників з кількох причин. По-перше, середовищем домашньої мережі не керують професійно. Це означає, що набагато більше систем у домашній мережі не виправляються регулярно, а деякі з них застаріли і вже не отримають необхідні виправлення для зниження вразливості. По-друге, щоб зберегти доступ до корпоративної мережі, зловмисник, який використав систему, повинен уникати виявлення та протистояти виправленню. У домашній мережі зробити це буде простіше, адже виявлення загроз зазвичай майже відсутнє, а виправлення відбуваються випадково, наприклад, коли на комп’ютері перевстановлюють операційну систему через те, що він працює повільно. Така домашня мережа не дуже відрізняється від загальнодоступної мережі Wi-Fi де-небудь у готелі, кав’ярні чи аеропорту, її безпека може бути легко скомпрометована.

Ф. Гройс, старший аналітик мережевого захисту Інституту програмної інженерії в Координаційному центрі CERT, у своїй статті «Віддалена робота: слабкі місця та загрози підприємству» розглядає технології та пристрої

віддаленого доступу, а також їхні властивості в контексті загрозливого середовища. [3]

Одним із традиційних і найвідоміших рішень проблеми віддаленої роботи є віртуальна приватна мережа (Virtual Private Network, VPN). VPN встановлює зашифрований тунель між системою, на якій працює клієнт VPN, і сервером VPN, який потім передає трафік через тунель до решти корпоративної мережі. Система, на якій працює VPN-клієнт, фактично стає розширенням корпоративної мережі, що існує всередині цієї мережі з доступом до ресурсів, який загалом еквівалентний будь-якій іншій системі корпоративної мережі (Рисунок 1.2).

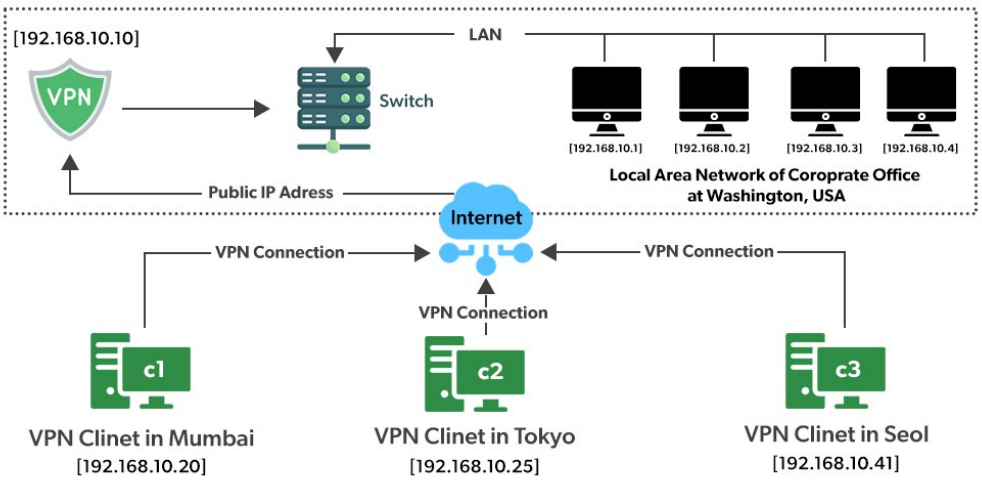


Рисунок 1.2 – Ілюстрація VPN

VPN захищаються від атак за допомогою автентифікованого контролю доступу та ізоляції. Такий підхід є ефективним за умови ефективного контролю. Хоча багато VPN налаштовано так, щоб заборонити можливість одночасного доступу до локальної фізичної мережі та віртуально підключеної корпоративної мережі, значно менше з них налаштовано так, щоб кінцева точка VPN фактично ніколи не взаємодіяла безпосередньо з локальною мережею. Якщо зловмисник постійно знаходиться у цій мережі, навіть короткочасний доступ може наразити кінцеву точку підприємства на компрометацію.

Інший підхід до віддаленого доступу полягає в тому, щоб дозволити користувачам віддалено керувати системою, яка вже знаходиться в мережі підприємства. Системи, які використовуються через віддалені робочі столи, можуть бути фізичними або віртуальними. Віртуальні системи можуть бути постійними або часто знищуватися та створюватися заново, іноді вони існують лише протягом сеансу входу користувача. Як і у випадку з VPN, віддалені робочі столи вимагають автентифікованого доступу, але практикують більш екстремальну форму ізоляції: кінцевий пристрій не є першокласним учасником корпоративної мережі; замість цього він відкриває користувачеві вікно в іншу систему. Незважаючи на те, що такий підхід додає абстракції та ускладнює роботу зловмисника, який прагне скомпрометувати підприємство, він однаково може спостерігати та навіть маніпулювати корпоративними системами з іншого боку вікна.

1.1.3 Застосування нових рішень

Цифрова трансформація спонукає компанії у всьому світі адаптувати свої мережі та стратегії безпеки. Останніми роками прискорилися дві ключові тенденції: впровадження хмарної інфраструктури та зростання розподіленої робочої сили. Разом ці тенденції призвели до реструктуризації мереж і безпеки. Тепер компаніям необхідно розгортати служби безпеки в будь-який час і в будь-якому місці на різноманітних архітектурах і кінцевих точках. Крім того, їм потрібно контролювати та захищати розподілену робочу силу, внутрішні ресурси та хмарну інфраструктуру. Оскільки традиційні проекти мережевої безпеки важко, або навіть неможливо перекласти на нові парадигми, потрібна нова модель безпеки. Компанії все частіше досліджують концепцію нульової довіри.

Національний інститут стандартів і технологій США у своїй публікації «Архітектура нульової довіри» дає визначення нульової довіри. [4]

Нульова довіра – це набір парадигм кібербезпеки, які зміщують захист зі статичних мережевих периметрів до зосередження на користувачах, активах і ресурсах. Нульова довіра передбачає відсутність прихованої довіри до активів або облікових записів користувачів виключно на основі їхнього фізичного чи мережевого розташування або на основі власності на активи. Нульова довіра є відповіддю на тенденції корпоративної мережі, серед яких: віддалені користувачі, модель залучення власних пристроїв користувачів (Bring Your Own Device, BYOD), і хмарні активи, які не розташовані в межах корпоративної мережі.

Тобто це не конкретна технологія або продукт, а саме концепція безпеки, ключовим принципом якої є усунення неявної, неперевіреної довіри для створення безпечного середовища доступу для бізнесу. Мета концепції полягає в тому, щоб надати лише точні повноваження та доступ, необхідні лише для довірених і перевірених користувачів. Нульова довіра усуває потребу у фізичних межах для розмежування довірених і ненадійних користувачів, пристроїв і мереж.

Ч. Чжан у своїй публікації «ZTNA: кращий спосіб для контролю доступу та підвищення безпеки» обґрунтовує переваги сучасних рішень на основі нульової довіри, зокрема у безпеці. [5]

На веб-ресурсі компанії Check Point у публікації «ZTNA vs VPN» детально розглянуто сучасні рішення на основі нульової довіри у контексті порівняння з традиційними VPN. [6]

Багато організацій проводили випробування та розробки на основі нульової довіри, здебільшого для вирішення проблем із керуванням ідентифікацією та контролем доступу. Мережевий доступ з нульовою довірою (Zero Trust Network Access, ZTNA) є результатом концепції нульової довіри, який має на меті замінити традиційні методи віддаленого доступу (такі як VPN) більш детальними елементами керування, більшою гнучкістю та масштабованістю, а також вищою надійністю.

Традиційні VPN припускають, що будь-який користувач, автентифікований засобами контролю периметра підприємства, або будь-який пристрій у корпоративній мережі автоматично вважається довіреним. ZTNA використовує іншу методологію: жоден користувач або пристрій не має довіреного доступу до будь-яких ресурсів, доки його особу не буде повністю перевірено та автентифіковано. Навіть тоді доступ до таких ресурсів, як сервери, програми та дані, обмежується наданими дозволами, відповідно до ролі або іншої класифікації користувача чи пристрою (Рисунок 1.3).

Рис. 1.3 – Ілюстрація складових безпеки рішення ZTNA

Ще одна причина, чому ZTNA викликає інтерес, полягає в тому, що традиційну VPN непросто масштабувати. Якщо VPN зазвичай потребують ручного налаштування для кожного користувача та пристрою, що, в свою чергу, вимагає значних витрать часу і зусиль, ZTNA надає компаніям більш гнучкий, масштабований і автоматизований спосіб контролю доступу та захисту ресурсів незалежно від фізичного розташування користувача чи пристрою (Рисунок 1.4).

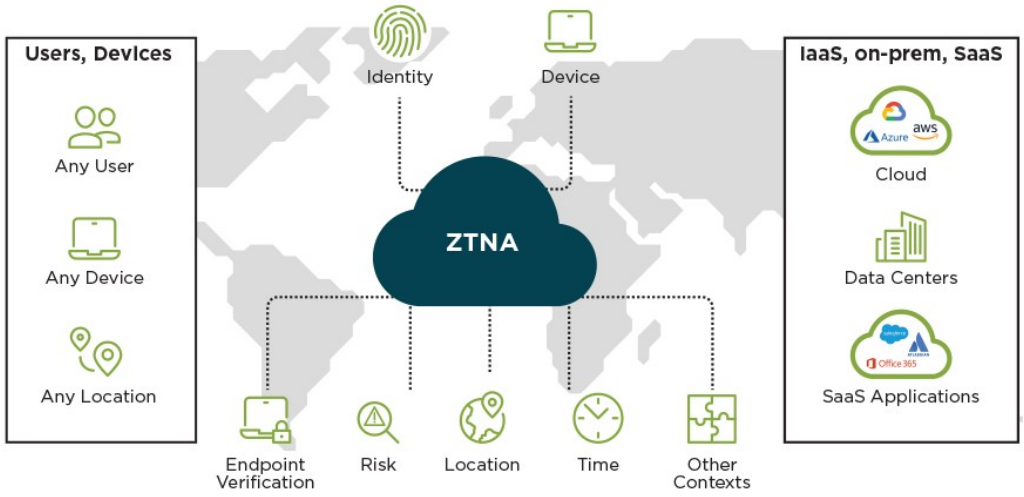


Рис. 1.4 – Ілюстрація гнучкості рішення ZTNA

Хоча рішення на основі нульової довіри мають багато переваг порівняно з традиційними рішеннями, численні компанії досі забезпечують віддалений доступ до своїх корпоративних мереж через VPN. Зважаючи на те, що концепція нульової довіри виникла нещодавно і ще розвивається, не всі компанії готові повністю відмовитись від перевіреного роками та добре знайомого VPN на користь нових рішень. Для багатьох компаній VPN залишається достатньо надійним та безпечним рішенням для організації віддаленої роботи своїх співробітників.

1.2 Забезпечення віддаленого доступу через VPN

Віртуальна приватна мережа – це спосіб розширення приватної мережі через загальнодоступну мережу, таку як Інтернет. Вона називається віртуальною, тому що залежить від використання віртуальних з’єднань, тобто тимчасових з’єднань, які не мають реальної фізичної присутності, але складаються з маршрутизованих пакетів.

Технології VPN, а також способи їх встановлення, конфігурації та використання детально описані Ч. Скоттом, П. Вулфом та М. Ервіном у їхній книзі «Віртуальні приватні мережі». [7]

Важливість безпеки при реалізації VPN, а також засоби захисту даних, які використовуються у цих технологіях, детально розглянуті С. Брауном у його книзі «Впровадження віртуальних приватних мереж». [8]

1.2.1 VPN віддаленого доступу

Компанії використовують VPN віддаленого доступу для встановлення безпечного з’єднання між корпоративною мережею та пристроями, якими користуються віддалені співробітники. Після підключення співробітники отримують доступ до ресурсів в мережі компанії так само, як якщо б їхні пристрої фізично підключили до цієї мережі.

VPN віддаленого доступу працює шляхом створення віртуального тунелю між пристроєм співробітника та мережею компанії. Цей тунель проходить через загальнодоступний Інтернет, але дані, що надсилаються туди й назад, захищені протоколами шифрування та безпеки, щоб зберегти їх конфіденційність.

Двома основними компонентами цього типу VPN є сервер доступу до мережі (Network Access Server, NAS) і клієнтське програмне забезпечення VPN (Рисунок 1.5).

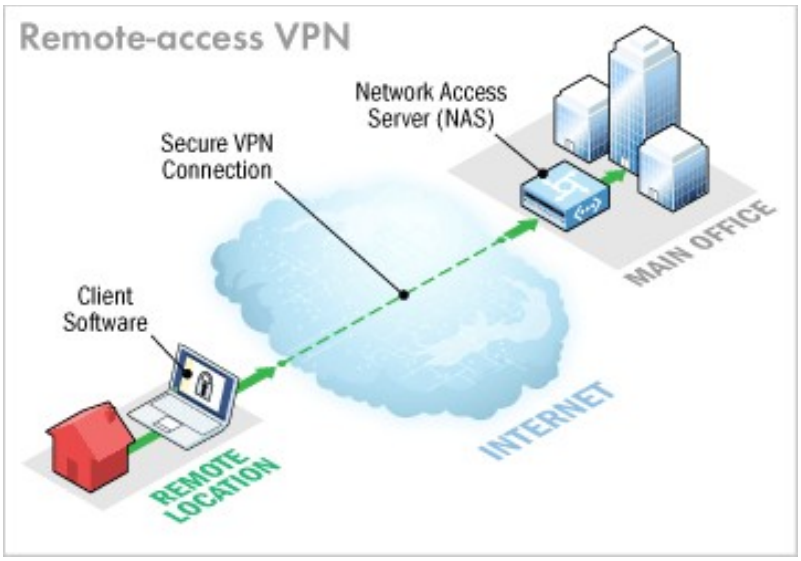


Рис. 1.5 – Ілюстрація VPN віддаленого доступу

Сервер доступу до мережі може бути виділеним сервером або програмним додатком, що працює на загальному сервері. Користувачі підключаються до NAS через Інтернет, щоб використовувати VPN віддаленого доступу. Щоб увійти в VPN, NAS вимагає від користувачів надати дійсні облікові дані. Для автентифікації цих облікових даних NAS використовує власний процес автентифікації або окремий сервер автентифікації, що працює в мережі.

Користувачі також повинні встановити клієнтське програмне забезпечення на своїх пристроях, щоб встановити та підтримувати з’єднання з VPN. Сьогодні більшість операційних систем постачаються з вбудованим програмним забезпеченням, яке може підключатися до VPN віддаленого доступу, хоча деякі служби VPN можуть вимагати від користувачів замість цього встановити певну програму. Клієнтське програмне забезпечення встановлює тунельне з’єднання з NAS і керує шифруванням, необхідним для гарантування безпеки з’єднання.

Принцип роботи VPN віддаленого доступу описує Е. Спадафора у своїй публікації «VPN віддаленого доступу: що це таке, як вони працюють і які найкращі». [9]

1.2.2 Засоби VPN для захисту даних

Існує декілька технологій, які VPN використовують для захисту даних, що передаються через Інтернет. Найважливішими з них є брандмауери, автентифікація, тунелювання і шифрування.

Брандмауер використовує такі методи, як перевірка Інтернет-адрес в пакетах або портів, які запитуються вхідними з’єднаннями, щоб вирішити, який трафік дозволено в мережі. Хоча більшість пакетів VPN безпосередньо не реалізують брандмауери, вони є невід’ємною частиною VPN. Ідея полягає в тому, аби використовувати брандмауер, щоб запобігати проникненням небажаних відвідувачів у мережу, водночас дозволивши доступ для користувачів VPN. Мережі, в яких відсутній брандмауер, наражають користувачів та їхні дані на значні ризики.

Найпоширенішими є брандмауери з фільтрацією пакетів, які блокують певні IP-сервіси, що працюють на конкретних номерах портів, від проходження через шлюз-маршрутизатор. Багато маршрутизаторів, які підтримують технології VPN, самостійно фільтрують пакети. Також поширеним методом захисту мережі є проксі-сервери, що дозволяють вхід службам VPN. Проксі-сервери зазвичай є програмними рішеннями, які працюють поверх мережевої операційної системи.

Методи автентифікації важливі для VPN, оскільки вони гарантують сторонам, які взаємодіють, що вони обмінюються даними з правильним користувачем або хостом. Автентифікація аналогічна входу в систему з іменем користувача і паролем. Однак віртуальні приватні мережі вимагають суворіших методів автентифікації для підтвердження особистості. Більшість систем автентифікації VPN ґрунтуються на системі спільного ключа. Ключі проходять через алгоритм хешування, який генерує хеш-значення. Інша сторона, яка володіє ключами, генерує власне хеш-значення, і порівнює його з тим, яке отримала з іншого кінця. Значення хеш-функції, відправлене через Інтернет, не має сенсу для спостерігача, тому навіть якщо хтось підслуховує

мережу, він не зможе підібрати пароль. Автентифікація зазвичай виконується на початку сеансу, а потім довільно упродовж сеансу, щоб гарантувати, що самозванець не прослизне у розмову.

Автентифікація також може використовуватися для забезпечення цілісності даних. Дані можуть бути відправлені за допомогою алгоритму хешування для отримання значення, яке додається у повідомлення як контрольна сума. Будь-яке відхилення в контрольній сумі, відправленій від одного партнера до іншого, означає, що дані були пошкоджені під час передачі або перехоплені і змінені на шляху.

VPN використовують тунелювання для створення приватної мережі. Вони дозволяють користувачеві підключатися до віддаленої мережі через Інтернет, що є IP-мережею. Проте є корпоративні локальні мережі, які не використовують виключно IP. Тунелювання дозволяє інкапсулювати пакет всередині пакета для підтримки несумісних протоколів. Таким чином пакет всередині пакета може належати як тому ж протоколу, так і зовсім іншому. За допомогою тунелювання також можна інкапсулювати IP-пакет в інший IP-пакет. Це означає, що можна відправити пакет з довільними вихідною та цільовою адресами через Інтернет в пакеті, що має вихідну та цільову адреси, що маршрутизуються через Інтернет. Ви можете використовувати зарезервований простір IP-адрес для приватних мереж в своїй локальній мережі, й надалі отримувати доступ до своїх хостів через Інтернет.

Шифрування є способом перетворення даних з формату, який можна прочитати, в закодований, нечитаний формат за допомогою алгоритму. Передбачуваний кодований формат може бути розшифрований лише за допомогою відповідного ключа для розшифрування.

VPN-шифрування дозволяє захистити конфіденційні дані (наприклад, номери кредитних карток, дані банківського рахунку та паролі від облікових записів) від кіберзлочинців, оскільки вони не зможуть підслуховувати інтернет-з'єднання, наприклад, при використанні загальнодоступного Wi-Fi.

VPN-шифрування також гарантує, що уряд, провайдер та рекламодавці не зможуть контролювати дії користувача в Інтернеті (Рисунок 1.6).

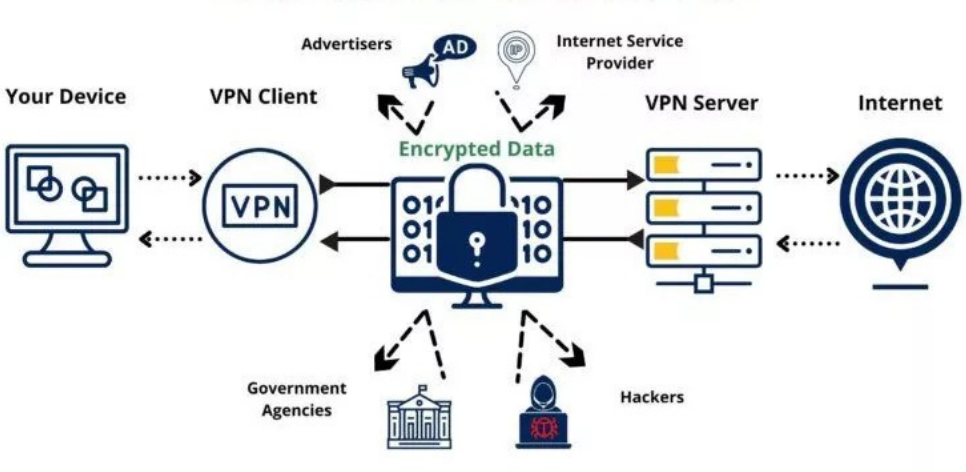


Рис. 1.6 – Ілюстрація VPN-шифрування

Коли користувач запускає VPN-клієнт і підключаєтеся до VPN-сервера, його запити шифруються перед відправкою на сервер. Потім вони розшифровуються сервером та пересилаються у відкритий Інтернет. Отримані у відповідь дані, що запитуються, знову шифруються на сервері, який потім пересилає їх на пристрій користувача. Вже на пристрої VPN-клієнт розшифровує дані і користувач може їх переглядати.

Для шифрування та розшифрування даних використовуються випадкові рядки бітів, які називають ключами. Кожен з таких ключів є унікальним. Довжина ключа шифрування обчислюється в бітах – як правило, чим довше ключ, тим вищий рівень шифрування.

У процесі шифрування та розшифрування використовуються два типи ключів: закритий та відкритий. Вони математично пов'язані, оскільки будь-яка інформація, зашифрована з допомогою відкритого ключа, може бути розшифрована лише з допомогою пов'язаного із ним закритого ключа. Крім того, відкритий ключ зазвичай знаходиться у публічному доступі, тоді як закритий ключ залишається конфіденційним і відомий лише власнику ключа.

Алгоритми шифрування поділяються на симетричні та асиметричні.

Симетричне шифрування ґрунтується на ідентичних відкритому та закритому ключах (Рисунок 1.7). Тому цей алгоритм вважають найшвидшим. Одним із прикладів симетричного шифрування є шифр AES (Advanced Encryption Standard, AES). AES може мати 128-бітові, 192-бітові та 256-бітові ключі. Цей шифр є популярним серед користувачів VPN завдяки сертифікації Національного інституту стандартів і технологій США (National Institute of Standards and Technology, NIST). Симетричними також є шифри 3DES (офіційно виведений з обігу і буде заборонений до використання після 2023 року), Blowfish і його наступник Twofish, Camellia тощо.

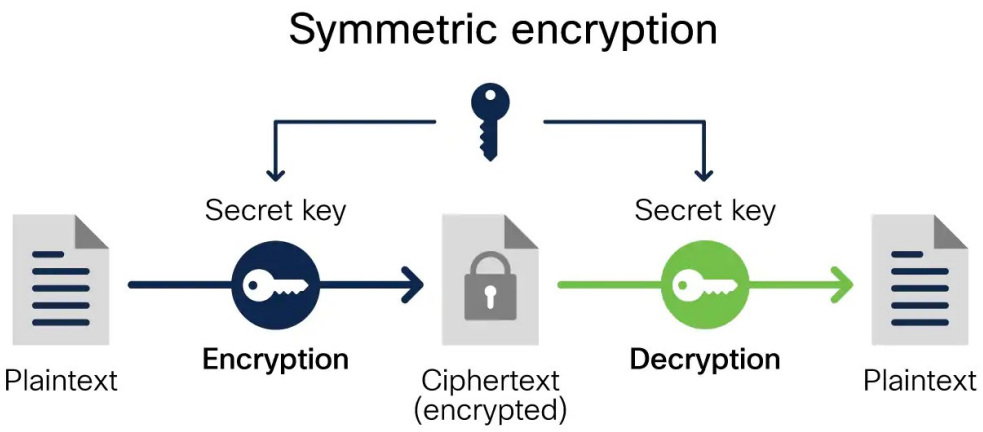


Рис. 1.7 – Симетричне шифрування

Асиметричне шифрування використовує різні ключі для процедур шифрування та розшифрування (Рисунок 1.8). Це може бути водночас зручно і ризиковано, оскільки закритий ключ неможливо відновити, якщо його буде втрачено. Гарним прикладом асиметричного шифрування є протокол RSA (абревіатура від прізвищ Rivest, Shamir та Adleman). Щоправда більшість VPN-сервісів використовують RSA виключно для встановлення з'єднання або так званого рукостискання, оскільки шифр відносно повільний. Через це RSA зазвичай не використовується для прямого шифрування даних користувача. Варто відзначити, що 1024-бітний ключ RSA більше не вважається безпечним, і рекомендовано використовувати ключі розміром у 2048 або 4096 біт.

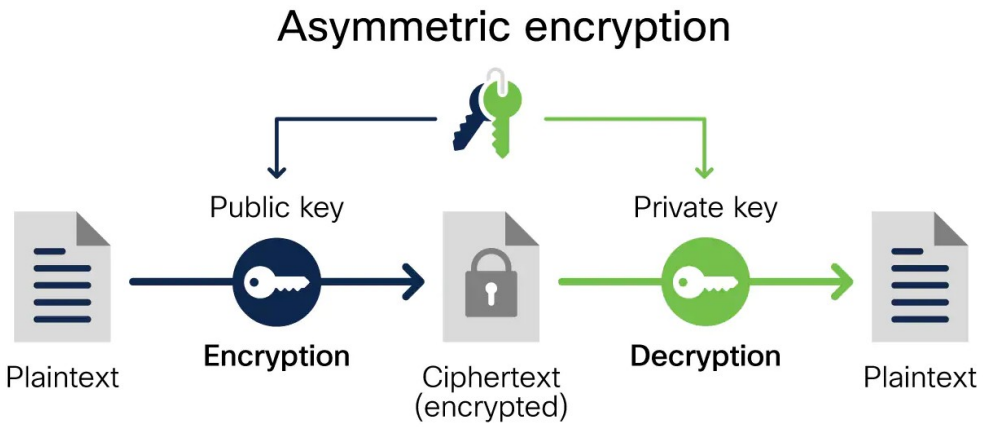


Рис. 1.8 – Асиметричне шифрування

Приклади симетричного і асиметричного шифрування наведені у публікації «Що таке шифрування?» на веб-ресурсі корпорації Cisco. [10]

1.2.3 Вплив VPN на продуктивність мережі

Використання VPN може знизити продуктивність мережевого з’єднання через те, що VPN спричиняє додаткові витрати на шифрування, а це збільшує затримку. Доводиться пожертвувати продуктивністю задля досягнення вищого рівня конфіденційності. Вивчаючи різні реалізації VPN, можна з’ясувати, як погіршується продуктивність при їх використанні.

Ш. Нарайян, К. Брукінг та С. де Вере у своїй роботі «Аналіз продуктивності мережі протоколів VPN: емпіричне порівняння в різних операційних системах» обговорили вибір реалізацій VPN і відповідного алгоритму шифрування, що впливає на продуктивність мережі. Вони довели, що протоколи VPN дають різні значення показників продуктивності мережі для різних комбінацій операційних систем, протоколів і алгоритмів. [11]

К. Навей та Ш. Ду у своїй роботі «Вплив VPN на продуктивність мережі» порівняли показники продуктивності мережі без VPN і мережі з VPN. Вони з’ясували, що VPN по різному впливає на продуктивність мережі, залежно від того, який протокол передачі даних використовується, TCP або

UDP. Також були визначені загальні показники продуктивності: пропускна здатність та затримка. [12]

З огляду результатів проведених раніше досліджень можна зробити висновок, що необхідно аналізувати вплив VPN на продуктивність мережі.

1.3 Проблема вибору протоколу VPN в умовах ненадійності мережі

VPN-протокол – це набір інструкцій, які використовуються для встановлення захищеного з'єднання між двома пристроями. У такому випадку двома захищеними пристроями будуть пристрій із запущеним VPN-клієнтом та VPN-сервер, з яким встановлюється з'єднання.

М. Хан та його колеги у їх спільній роботі «Емпіричний аналіз комерційної екосистеми VPN» розглядають поширені технології тунелювання, які використовуються у VPN-сервісах. [13]

Д. Доненфельд у своїй роботі «WireGuard: мережевий тунель ядра нового покоління» стверджує, що WireGuard вже найближчим часом може замінити провідні VPN-протоколи. [14]

1.3.1 Поширені протоколи VPN

Згідно з дослідженням М. Хана та ін., двома найпоширенішими протоколами, які сьогодні використовуються у VPN віддаленого доступу, є IPsec та OpenVPN (Рисунок 1.9).

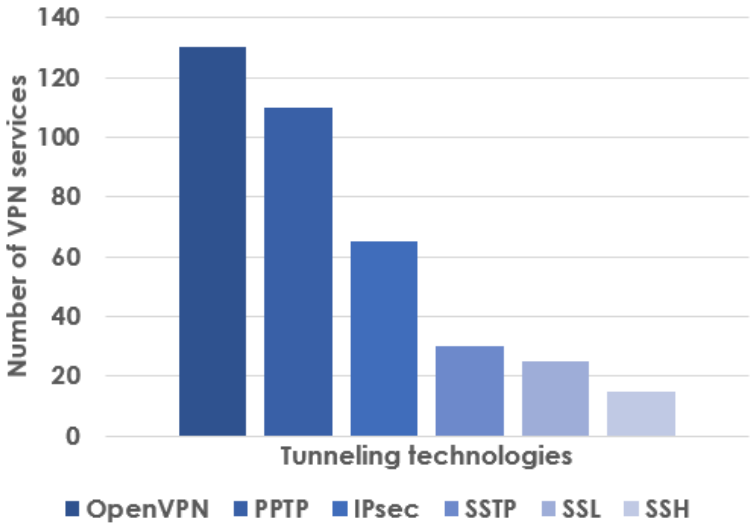


Рис. 1.9 – Поширені технології тунелювання

IPSec зазвичай використовується разом із протоколом L2TP (Layer 2 Tunneling Protocol, L2TP). L2TP забезпечує тунелювання шляхом інкапсуляції корисного навантаження з однієї точки в іншу, але не надає можливості шифрування. IPSec доповнює його, виконуючи автентифікацію і шифрування пакетів даних, і відповідно підвищує рівень безпеки. Але, через те, що дані інкапсулюються двічі, це безпосередньо впливає на швидкість передачі даних. L2TP/IPSec підтримується всіма сучасними операційними системами.

OpenVPN є універсальним протоколом, який підтримує різні алгоритми шифрування і працює на великій кількості платформ. OpenVPN пройшов декілька незалежних експертиз безпеки, що підтвердили у нього відсутність серйозних вразливостей. Головним недоліком OpenVPN вважається складність налаштування VPN за цим протоколом.

Відносно новий протокол WireGuard розроблявся з метою замінити обидва названі протоколи і забезпечити більшу продуктивність, у порівнянні з попередніми реалізаціями. Незважаючи на те, що WireGuard сьогодні займає не надто помітне місце в індустрії VPN, Д. Доненфельд називає WireGuard протоколом, який уникає складності IPsec і має ліпшу

продуктивність за OpenVPN. Спершу WireGuard був написаний виключно для Linux, але зараз цей протокол доступний для великої кількості платформ.

У літературі згадуються й інші протоколи, але після її огляду можна дійти висновку, що більшість VPN-сервісів все ж використовують один з двох протоколів – IPsec або OpenVPN, або обидва. Проте є й інші поширені протоколи, які були виключені з цього дослідження, наприклад PPTP та SSTP.

Свого часу протокол PPTP (Point-to-Point Tunneling Protocol, PPTP) був першим широкодоступним протоколом VPN, сьогодні ж це застарілий протокол, що використовує одні з найслабших алгоритмів шифрування з усіх поширених протоколів. Багато користувачів досі використовують PPTP, зокрема коли пріоритетом є не конфіденційність даних, а швидкість передачі (наприклад, для доступу до геозаблокованого вмісту), адже через застарілість протоколу сучасні комп’ютери дуже ефективно його запускають, тому PPTP швидший за інші протоколи. Розробник протоколу, корпорація Майкрософт рекомендувала використовувати іншу технологію тунелювання ще у 2012 році. [15]

На відміну від OpenVPN та WireGuard, які мають відкритий вихідний код, протокол SSTP (Secure Socket Tunneling Protocol, SSTP) недоступний дослідникам безпеки для тестування. Відомо, що розробник протоколу, корпорація Microsoft співпрацює з державними агенціями США, такими як Агентство Національної Безпеки (National Security Agency, NSA), тому багато хто підозрює, що у SSTP можуть бути бекдори. Незважаючи на те, що SSTP підтримує надійні алгоритми шифрування, багато провайдерів VPN уникають цього протоколу. [16]

1.3.2 Проблема вибору протоколу VPN

Спроби порівняльного аналізу протоколів VPN з метою визначити найліпший для реалізації VPN віддаленого доступу були і раніше. У

переважній більшості робіт протокол IPSec, зокрема в комбінації L2TP/IPSec, порівнювався з іншими, не надто новими протоколами, як, наприклад, PPTP. Набагато рідше, але все ж зустрічаються роботи, у яких порівнюються протоколи, які поширені сьогодні.

У роботі «Порівняння протоколів VPN на мережевому рівні з упором на протокол WireGuard» були детально охарактеризовані протоколи IPSec та WireGuard на мережевому рівні з точки зору дизайну, вартості, конфіденційності, автентифікації, шифрування та безпеки, цілісності, швидкості та порту. В результаті порівняння характеристик було зроблено висновки про переваги та недоліки обох протоколів. У підсумку роботи вказано на велику кількість вже проведених масштабних та ефективних випробувань IPSec, що дозволяє більшості компаній успішно використовувати його завдяки його безпеці, натомість WireGuard є дуже новим і з ним було проведено не так багато тестів, але він вже демонструє гарну продуктивність, порівняно з IPSec та іншими протоколами. Жодні експериментальні дослідження у цій роботі не проводились. [17]

У роботі «Порівняння продуктивності VPN-рішень» наведено результати аналізу продуктивності VPN-тунелів на базі трьох протоколів: IPSec, OpenVPN та WireGuard. Було проведено експериментальне дослідження, в ході якого вимірювали пропускну здатність для різної конфігурації протоколів і алгоритмів шифрування. Метою дослідження було з’ясувати, як протоколи працюють з різними алгоритмами шифрування і у яких випадках демонструють найбільшу продуктивність. Вимірювання проводились виключно на платформі Linux. [18]

З огляду наведених робіт можна зробити висновок, що продуктивність VPN на базі поширених сьогодні протоколів належним чином не була досліджена, і потребує проведення додаткового аналізу, зокрема в умовах ненадійної мережі.

1.3.3 VPN в умовах ненадійності мережі

Віддалений доступ сьогодні часто стикається з ненадійністю мереж. Звісно, що користувач, який працює в корпоративній мережевій інфраструктурі, яка контролюється та обслуговується підприємством, скоріше за все, матиме більш стабільне з’єднання, ніж віддалений працівник. Віддалені працівники можуть використовувати різні типи підключень, які не контролюються підприємством.

Користувачі стають мобільнішими, вони використовують ноутбуки, смартфони та інші портативні пристрої для роботи, а відтак можуть знаходитись у різних місцях, де неможливо забезпечити постійне стабільне з’єднання з Інтернетом. Користувач може працювати в ненадійній публічній мережі Wi-Fi або, наприклад, в поїзді, де ненадійність пов’язана з нестійким мобільним зв’язком під час руху поїзду, далеко від міст з гарною інфраструктурою.

Тому при виборі протоколу VPN важливо з’ясувати, які реалізації VPN краще долають ненадійність мереж.

1.4 Висновки за розділом

У цьому розділі було досліджено проблему вибору протоколу VPN для забезпечення віддаленого доступу, зокрема в умовах ненадійності мережі.

Той факт, що кількість віддалених працівників зростає, і що VPN залишається одним з найпоширеніших рішень для віддаленого доступу, вимагає відповіді на питання – як відрізняється продуктивність окремих реалізацій VPN і як обрати з них найліпшу для власних потреб.

Було розглянуто раніше проведені дослідження, результати цих досліджень вказують на те, що продуктивність VPN залежить від операційної системи, протоколу та алгоритму шифрування, тому необхідно провести

порівняльний аналіз продуктивності VPN на різних платформах і з різною конфігурацією.

Для порівняльного аналізу обрані два провідні на сьогодні протоколи VPN, L2TP/IPsec та OpenVPN, а також відносно новий протокол, що нарощує популярність – WireGuard. Оскільки віддалений доступ сьогодні часто стикається з ненадійністю мереж, необхідно проаналізувати продуктивність VPN як в стабільних умовах, так і в умовах ненадійності мережі.

РОЗДІЛ 2
ДОСЛІДЖЕННЯ ПРОДУКТИВНОСТІ ПРОТОКОЛІВ VPN

2.1 Порівняльна характеристика протоколів VPN

У цьому розділі представлена порівняльна характеристика протоколів L2TP/IPSec, OpenVPN та WireGuard, у якій було вирішено охарактеризувати та порівняти протоколи за такими критеріями: дизайн, сумісність з різними платформами, вбудовуваність, складність налаштування, методи автентифікації, шифрування, цілісність даних, стійкість до атак, конфіденційність, протоколи і порти з’єднання, швидкість, контекст виконання та стабільність.

Оскільки між деякими критеріями прослідковується очевидна залежність, вони були згруповані відповідно до тих аспектів протоколів, які вони характеризують. Таким чином будуть розглянуті три основні аспекти: доступність, безпека та продуктивність.

2.1.1. Загальна характеристика

Якщо порівнювати дизайн протоколів L2TP/IPsec, OpenVPN та WireGuard, можна зробити висновок, що вони суттєво відрізняються.

WireGuard покликаний бути простішим, компактнішим та зручнішим у використанні за поширені сьогодні, надзвичайно складні протоколи, про що неодноразово стверджував його розробник, Д. Доненфельд. WireGuard реалізований в менш, ніж 4000 рядках коду, що значно менше за OpenVPN та L2TP/IPsec. [19]

Менша кодова база надає протоколу WireGuard такі переваги:

– набагато простіше проводити аудит, одна людина може прочитати кодову базу WireGuard за кілька годин;

– якщо вихідний код легше перевіряти, отже легше знаходити вразливості, що допомагає зробити WireGuard безпечнішим;
– набагато менша поверхня атаки;
– краща продуктивність.

Згідно з останнім твердженням OpenVPN Technologies, кодова база OpenVPN містить приблизно 70000 рядків коду. Це більше за WireGuard, але справедливо буде відзначити, що WireGuard, яким він є сьогодні, має набагато менший набір методів автентифікації та набагато менші інтерфейси інтеграції, порівняно з OpenVPN. Це одна з причин, чому кодова база OpenVPN є більш повною. [20]

Говорити про кодову базу L2TP/IPsec важче через варіативність реалізацій, але є інформація про те, що деякі реалізації протоколу на платформі Linux сягають 400000 рядків коду (Рисунок 2.1). [21]

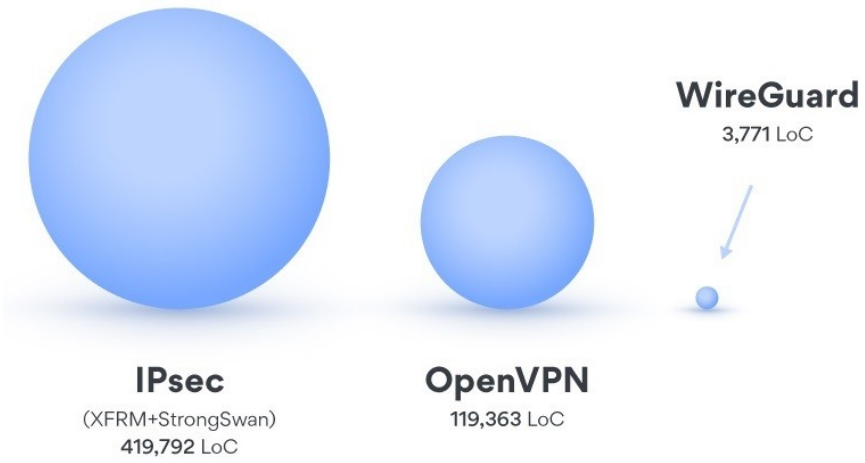


Рис. 2.1 – Порівняння кодової бази VPN-протоколів

Усі три протоколи підтримують більшість сучасних платформ: Windows, Linux, macOS, iOS, Android та інші. Деякі з них вже вбудовані в основні платформи. Проте у багатьох випадках для їх використання все ж знадобиться додаткове програмне забезпечення.

У контексті доступності вирізняється L2TP/IPsec, оскільки протокол вже вбудований у різноманітні платформи: Windows, Linux, macOS, Android, iOS тощо. Це означає, що його налаштування можна виконувати вбудованими засобами платформи, не завантажуючи для цього ніяких сторонніх програм. Реалізації L2TP/IPsec також існують в основних маршрутизаторах, таких як Cisco та Juniper. [22]

Незважаючи на той факт, що OpenVPN сьогодні вважається стандартом у галузі, його не вбудовано у жодну з платформ. OpenVPN вимагає спеціального клієнтського програмного забезпечення, яке у будь-якому разі доведеться завантажувати додатково для його використання. Компанія-розробник, OpenVPN Technologies розробляє і підтримує офіційні клієнти для свого протоколу. [23]

WireGuard також тривалий час не входив до жодної платформи. Це, насамперед, пов’язано з тим, що WireGuard – новий протокол у галузі, він був представлений 2015 року у вигляді бета-версії, перша стабільна версія протоколу була випущена набагато пізніше, аж у 2020 році. Для порівняння, OpenVPN був випущений 2001 року, а IPsec і L2TP – ще у 90-х. Очевидно, що WireGuard досі не був достатньо протестований, і це може бути однією з причин, чому його не вбудовували у будь-які платформи.

Нещодавно, а саме 2020 року відбулося те, до чого розробник WireGuard, Д. Доненфельд прагнув з моменту виходу першої бета-версії протоколу – починаючи з ядра Linux версії 5.6 WireGuard став однією з технологій, вбудованих за замовчуванням. Модуль ядра WireGuard тепер доступний у сховищах пакетів усіх основних дистрибутивів Linux і навіть деяких спеціалізованих. [24]

Щоб покращити продуктивність WireGuard у Windows, Доненфельд та інші розробники WireGuard створили новий, простіший драйвер TUN з відкритим кодом під назвою Wintun. TUN (Network TUNnel, TUN) – це віртуальний мережевий пристрій ядра системи, який повністю підтримується програмним забезпеченням, на відміну від звичайних мережевих пристроїв,

які підтримуються фізичними мережевими адаптерами. Windows не надає рідного віртуального пристрою TUN, і хоча існують деякі драйвери для досягнення цього з таких проектів, як OpenVPN або SoftEther, вони були написані давно і мають різні проблеми. Wintun можна вважати великим кроком до включення WireGuard у ядро Windows.

Налаштування L2TP/IPSec зазвичай відбувається швидко та легко, оскільки він вбудовано підтримується багатьма операційними системами. Достатньо імпортувати файли конфігурації від провайдера VPN. Проте, налаштовуючи цей протокол, можна стикнутися з проблемою блокування брандмауерами портів підключення, які використовує L2TP/IPSec. Щоб дозволити L2TP/IPSec проходити через брандмауери, необхідні додаткові кроки конфігурації, що ускладнює реалізацію безпечної VPN. [25]

OpenVPN не вбудований у жодну операційну систему, для його налаштування знадобиться додаткове програмного забезпечення. Більшість провайдерів VPN надають власні програми OpenVPN, які можна використовувати на різних операційних системах і пристроях. Використовуючи такі програми, налаштувати OpenVPN швидше й простіше, ніж виконувати конфігурацію самостійно.

OpenVPN пропонує широкий спектр конфігурованих параметрів. Протокол дозволяє обирати шифри, методи автентифікації, топологію мережі, необхідність стиснення даних та багато іншого. Завдяки цьому OpenVPN гнучкий у налаштуванні, і водночас складний для недосвідченого користувача.

OpenVPN може використовувати різні порти і протоколи з’єднання, і маскувати свій трафік під типовий веб-трафік, тому брандмауерам важче заблокувати протокол, але така можливість зберігається. [26]

WireGuard прагне бути таким же простим у налаштуванні та розгортанні, як і SSH (Secure SHell, SSH). VPN-з’єднання створюється простим обміном дуже простими відкритими ключами – так само, як обмін ключами SSH – а все інше прозоро обробляється WireGuard.

Для порівняння, інше програмне забезпечення VPN, наприклад OpenVPN і IPSec, використовує захист транспортного рівня (Transport Layer Security, TLS) і сертифікати для автентифікації та встановлення зашифрованих тунелів між системами. Різні версії TLS включають підтримку сотень різних криптографічних наборів і алгоритмів, і хоча це забезпечує велику гнучкість для підтримки різних клієнтів, це також робить налаштування VPN, що використовує TLS, більш трудомістким, складним і схильним до помилок. [27]

WireGuard має стандартний порт підключення, який він використовує за замовчуванням, але цей порт можна змінити на будь-який інший і налаштувати з’єднання через нього – у такому випадку малоймовірно, що брандмауери його блокуватимуть.

Узагальнена характеристика протоколів L2TP/IPSec, OpenVPN та WireGuard представлена у таблиці (Таблиця 2.1).

Таблиця 2.1

Загальна характеристика VPN-протоколів

Критерії	L2TP/IPSec	OpenVPN	WireGuard
Рік випуску	1995 (IPSec), 1999 (L2TP)	2001	2015
Дизайн	Комплексний і складний	Складний	Простий і компактний
Доступність	Windows, Linux, macOS та ін.	Windows, Linux, macOS та ін.	Windows, Linux, macOS та ін.
Вбудовуваність	Windows, Linux, macOS та ін.	Ні	Linux
Блокування брандмауерами	Так	Малоймовірно	Малоймовірно
Налаштування	Швидке і легке	Гнучке, але складне	Негнучке, але просте

Проаналізувавши узагальнену характеристику, можна зробити такі висновки:

- Найдоступнішим з розглянутих протоколів є L2TP/IPsec, він вбудований у більшість сучасних платформ, його налаштування має бути швидким і легким, але необхідно враховувати проблему з брандмауерами;
- OpenVPN так само має високу сумісність, але налаштувати його можна тільки через додаткові програми, цей протокол гнучкий у налаштуванні, але складний;
- WireGuard вбудовано підтримується тільки на Linux, на всіх інших платформах його необхідно налаштовувати через додаткові програми, але зробити це простіше, порівняно з іншими протоколами.

2.1.2. Характеристика безпеки та шифрування

У комбінації протоколів L2TP/IPSec за безпеку з’єднання і все, що з нею пов’язано, відповідає IPSec. L2TP забезпечує виключно тунелювання, тому 2001 року в специфікації RFC 3193 було обгрунтовано, як L2TP може використовувати IPSec для заповнення прогалини в безпеці. [28]

VPN-з’єднання за протоколом IPSec починається зі встановлення асоціації безпеки (Security Association, SA) між двома хостами, що комунікують між собою. Загалом це передбачає обмін криптографічними ключами, які дозволять сторонам шифрувати та розшифровувати свою комунікацію. Між хостами автоматично узгоджується точний тип шифрування, який вони використовуватимуть, і який задовольняє їхні потреби в безпеці. Інформація про SA передається модулю IPSec на кожному хості, який використовує її для зміни пакетів, що надсилаються іншому

хосту, і для обробки подібно змінених пакетів, отриманих у відповідь. Ці зміни можуть вплинути як на заголовок пакета, так і на його корисне навантаження. [29]

Далі IPSec використовує два різні протоколи: AH (Authentication Header, AH) та ESP (Encapsulating Security Payload, ESP).

AH забезпечує механізм лише для автентифікації, але не для конфіденційності. Цей протокол перевіряє цілісність даних, виконує автентифікацію походження даних і надає додаткову службу запобігання відтворенню. Цілісність даних забезпечується за допомогою дайджесту повідомлення (хеш-суми), що генерується таким алгоритмом як HMAC-MD5 або HMAC-SHA (Hash-based Message Authentication Code, HMAC). Автентифікація походження даних забезпечується використанням спільного секретного ключа для створення дайджесту повідомлення. Запобігання відтворенню забезпечується через поле порядкового номера у заголовку AH.

ESP забезпечує і автентифікацію, і конфіденційність даних. Цей протокол можна використовувати лише для автентифікації або лише для конфіденційності, або одразу для обох цілей. Коли ESP забезпечує функції автентифікації, він використовує ті самі алгоритми, що й AH, але в інший спосіб. AH автентифікує весь пакет, включаючи зовнішній заголовок, тоді як ESP автентифікує лише частину дейтаграми пакета.

Будь-який з цих протоколів можна використовувати окремо, або комбінувати їхні функції між собою. [30]

IPSec пропонує два режими використання: транспортний та тунельний. Різниця між ними полягає у тому, як IPSec обробляє заголовки пакетів. У транспортному режимі IPSec шифрує (або автентифікує, якщо використовується лише AH) тільки корисне навантаження пакета, залишаючи існуючі дані заголовка переважно без змін. У тунельному режимі IPsec створює повністю новий пакет із новим заголовком, шифрує (або автентифікує) весь оригінальний пакет, з його заголовком, і використовує модифікований вихідний пакет як корисне навантаження для нового пакета.

IPSec, який використовує ESP в тунельному режимі для забезпечення автентифікації і шифрування – це конфігурація IPSec, що найчастіше спостерігається у VPN. Інші можливі конфігурації IPSec наведені на рисунку (Рисунок 2.2.). [31]

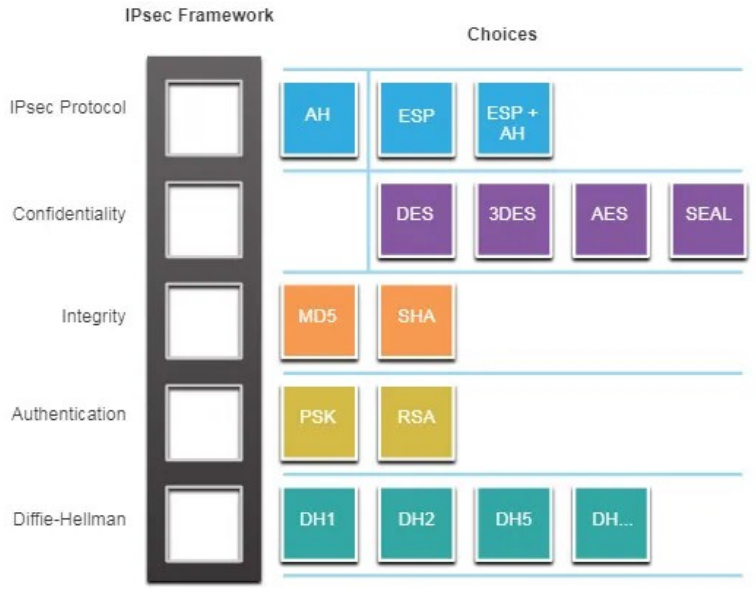


Рис. 2.2 – Конфігурації IPSec

Протокол ESP можна використовувати з різними алгоритмами шифрування, але найуживанішим залишається AES-256. Надійність такого шифрування не повинно викликати питань, оскільки AES рекомендований NSA і використовується урядом США. [32]

З іншого боку, довіра до IPSec неодноразово ставала предметом для обговорень, зокрема після так званого «витоку Сноудена» 2013 року, коли стало відомо, що NSA витрачає величезні ресурси на послаблення протоколів безпеки і, ймовірно, успішно розробляє власні методи подолання конфіденційності таких протоколів, як IPSec. Незважаючи на це, IPSec досі вважається безпечним для загального використання. [33]

Протокол AH більше не рекомендовано використовувати, відповідно до останньої спеціальної публікації NIST 2020 року, проте AH досі присутній у багатьох існуючих реалізаціях IPSec. [34]

Залежно від режиму та конфігурації OpenVPN, можливі такі методи автентифікації підключень: за допомогою пар ключів і сертифікатів або за допомогою імені користувача і пароля. [35]

У режимі TLS сервер завжди має власний ключ, виданий сертифікат та сертифікат центру сертифікації (Certification Authority, CA). Усі клієнти також повинні мати копію цього сертифіката CA. Клієнти можуть бути автентифіковані за допомогою власних сертифікатів, облікових даних користувача, або і того, й іншого, як форма двохфакторної автентифікації.

Сертифікати криптографічно підписані CA, тому вони забезпечують високий рівень безпеки та автентифікації. Натомість імена користувачів дещо менш безпечні, враховуючи типи фраз-паролів, які часто використовуються, і багаторазове повторне використання тієї самої або схожої фрази-паролю.

Зауважу, що процес автентифікації підключень доступний лише в режимі TLS. Інший базовий режим OpenVPN – це попередній спільний ключ (Pre-Shared Key, PSK), де автентифікація забезпечується лише власником симетричного ключа.

OpenVPN використовує бібліотеку OpenSSL для забезпечення шифрування. OpenSSL підтримує низку різних криптографічних алгоритмів, які наведені у таблиці (Таблиця 2.2.). [36]

Таблиця 2.2

Криптографічні алгоритми OpenSSL

Алгоритми	Призначення
AES, Blowfish, Camellia, ChaCha20, Poly1305, DES, 3DES, SM-4 та ін.	Для шифрування та автентифікації
MD5, MD4, SHA-1, SHA-2, MDC-2, BLAKE2 та ін.	Для хешування
RSA, DSA, X25519, Ed25519 та ін.	Для отримання та узгодження ключів
Perfect Forward Secrecy	Для захисту даних користувачів

Як видно з таблиці, OpenSSL пропонує гнучкі механізми забезпечення автентичності, цілісності та конфіденційності даних у VPN, реалізованих на базі протоколу OpenVPN.

Щоб попередити атаки, OpenVPN за замовчуванням забезпечує захист від повторного відтворення дейтаграм. Кожна вихідна дейтаграма позначається ідентифікатором, який гарантовано є унікальним для використовуваного ключа. Одноранговий вузол, який отримує дейтаграму, перевіряє унікальність ідентифікатора. Якщо ідентифікатор вже було отримано в попередній датаграмі, OpenVPN скидає пакет.

Захист від повторного відтворення в OpenVPN реалізується дещо по-різному, залежно від обраного режиму керування ключами: це може бути 64-розрядний унікальний ідентифікатор, який поєднує позначку часу з порядковим номером, що збільшується; або лише 32-бітний порядковий номер без мітки часу, коли OpenVPN може гарантувати унікальність цього значення для кожного ключа. [37]

Щоб бути певним, що VPN-протоколу можна довірити свої дані, його необхідно перевірити. Можливість перевірки є однією з ключових причин того, чому більшість прихильників конфіденційності віддають перевагу програмному забезпеченню з відкритим вихідним кодом. Але той факт, що код є відкритим, не полегшує аудит. OpenVPN можна перевірити, тому що це протокол з відкритим кодом, але оскільки цей код сягає сотні тисяч рядків, для його аудиту необхідна команда з експертів і багато часу.

Під час однієї з таких перевірок 2017 року у вихідному коді OpenVPN були знайдені деякі вразливості, але вони були негайно виправлені. Це свідчить про те, що протокол постійно оновлюється, і якщо ви використовуєте останню версію OpenVPN, ви не наражаєте себе на будь-які ризики. [38]

Наразі OpenVPN не має відомих вразливостей безпеки. Цей протокол неодноразово перевірявся та отримав підтримку багатьох експертів із безпеки.

WireGuard розглядає кожну кінцеву точку у VPN як одноранговий вузол. Кожен такий вузол має пару закритого і відкритого ключів, які однозначно ідентифікують цей вузол. Маючи закритий ключ, можна шляхом нетривіальних математичних розрахунків отримати відкритий ключ, але це не працює в інший бік. Тому закритий ключ має залишатися в таємниці і зберігатися надійно. Відкриті ключі використовуються для автентифікації одноранговых вузлів, коли вони з’єднуються один з одним. [39]

Початкове дуже просте рукостискання встановлює симетричні ключі, які будуть використовуватись для передачі даних. Далі це рукостискання повторюється кожні кілька хвилин, щоб забезпечити постійну зміну ключів для цілковитої прямої секретності (Perfect Forward Secrecy, PFS). Це робиться на основі часу, а не на основі вмісту попередніх пакетів, оскільки WireGuard розроблений для того, щоб обережно впоратися із втратою пакетів. WireGuard використовує розумний імпульсний механізм, який забезпечує актуальність останніх ключів та рукостискань, і автоматично виявляє, коли вони застаріли.

Якщо потрібен додатковий рівень шифрування із симетричним ключем, WireGuard також підтримує режим PSK, тобто попереднього спільного ключа, який додається до криптографії з відкритим ключем.

WireGuard усуває криптографічну гнучкість – концепцію пропонування вибору між різними алгоритмами шифрування, обміну ключами та хешування – оскільки це призводить до небезпечного розгортання з іншими технологіями. Натомість у протоколі використовується набір сучасних, ретельно перевірених і рецензованих криптографічних примітивів, що призводить до сильних криптографічних варіантів за замовчуванням, які користувачі не можуть змінити або неправильно налаштувати. Якщо будь-яка серйозна вразливість буде виявлена у

використовуваних криптографічних примітивах, буде випущена нова версія протоколу, у WireGuard також існує механізм узгодження версії протоколу між одноранговими вузлами.

Криптографічні алгоритми, які використовує WireGuard, наведені у таблиці (Таблиця 2.3). [40]

Таблиця 2.3

Криптографічні алгоритми WireGuard

Алгоритми	Призначення
ChaCha20	Для симетричного шифрування
Poly1305	Для автентифікації
Curve25519	Для еліптичної кривої Діффі-Хеллмана
BLAKE2s	Для хешування
SipHash24	Для ключів хеш-таблиці
HKDF	Для отримання ключів
Perfect Forward Secrecy	Для захисту даних користувачів

Однією з цілей розробки WireGuard було уникнути збереження будь-якого стану до автентифікації та не надсилати жодних відповідей на неавтентифіковані пакети. Не зберігаючи стан неавтентифікованих пакетів і не генеруючи відповіді, WireGuard залишається невидимим для нелегітимних однорангових вузлів і мережевих сканерів. Таким чином можна уникнути одразу кількох видів атак.

Однак ця властивість вимагає, щоб перше повідомлення, отримане відповідачем, автентифікувало ініціатора. Це, в свою чергу, наражає відповідача на атаку повторного відтворення. Зловмисник може відтворити початкові повідомлення рукостискання, щоб обманом змусити відповідача повторно згенерувати свій ключ, тим самим завершивши сеанс легітимного

ініціатора. Щоб запобігти цьому, в перше повідомлення включається 12-байтна позначка часу TAI64N. Відповідач відстежує найбільшу мітку часу, а пакети, що містять мітки, менші або рівні їй, відкидає. Навіть якщо сервер перезавантажиться, атака повторного відтворення неефективна, оскільки під час повторного підключення клієнти використовуватимуть нові мітки часу, які зроблять усі попередні мітки недійсними. [41]

Окрім очевидних переваг, WireGuard має і деякі проблеми з конфіденційністю. WireGuard пов’язує IP-адреси однорангових вузлів з їхніми відкритими ключами, і зберігає їх на VPN-сервері необмежений час або до перезавантаження VPN-сервера. Це насправді нічим не відрізняється від будь-якого іншого VPN-протоколу, адже йому потрібно знати, куди надсилати зашифровані пакети. Відмінність полягає в тому, що інші протоколи відстежують, чи активний одноранговий вузол, і коли він стає неактивним або закриває з’єднання, видаляють інформацію про нього. WireGuard цього не робить, бо не працює із концепцією з’єднання. Однорангові пристрої можуть припинити обмін даними в будь-який час і очікувати, що зможуть продовжити його у будь-який момент у майбутньому. Це зручно для користувача, але небажано для його конфіденційності. Провайдери VPN запроваджують власні способи вирішення цієї проблеми, і вони ефективні, але все ж це вважається недоліком протоколу. [42]

Ще одним недоліком вважається нездатність WireGuard призначати динамічні IP-адреси. Це означає, що щоразу, коли одноранговий вузол підключається за допомогою WireGuard до сервера, йому призначається та сама IP-адреса. Хоча це не його справжня адреса, той факт, що вона залишається незмінною під час кожного сеансу, може поставити конфіденційність вузла під загрозу, оскільки його активність стає легше відстежувати. Це не притаманно іншим протоколам, але притаманно WireGuard, і це так само доводиться враховувати провайдерам VPN.

WireGuard це протокол з відкритим вихідним кодом, а це означає, що його можна перевірити. Оскільки код цього протоколу не перевищує 4000

рядків, на пошук вразливостей знадобиться набагато менше експертів і часу, порівняно з OpenVPN.

Код протоколу WireGuard був перевірений кількома групами експертів із безпеки з приватного сектору та наукових кіл, а також був офіційно перевірений у різних обчислювальних моделях. Усі проведені перевірки підтвердили відсутність у WireGuard вразливостей безпеки. [43]

Узагальнена характеристика безпеки протоколів L2TP/IPSec, OpenVPN та WireGuard представлена у таблиці (Таблиця 2.4).

Таблиця 2.4

Характеристика безпеки VPN-протоколів

Критерії	L2TP/IPSec	OpenVPN	WireGuard
Вихідний код	У цій комбінації закритий	Відкритий	Відкритий
Автентифікація	За допомогою AH та ESP	Гнучка, TLS і сертифікати	Відкриті ключі, Poly1305 для автентифікації
Шифрування	За допомогою ESP, AES-256 або інші шифри	Гнучке, AES-256 або інші шифри	ChaCha20 для симетричного шифрування
Цілісність	Хешування, різні алгоритми	Хешування, різні алгоритми	Хешування, BLAKE2s
Запобігання відтворенню	Поля порядкових номерів	64-бітні і 32-бітні ідентифікатори	96-бітні позначки часу TAI64N
Конфіденційність	Забезпечує, але підозрюється у зламі NSA	Забезпечує	Забезпечує, але має недоліки

Проаналізувавши узагальнену характеристику, можна зробити такі висновки:

- Найсучаснішу криптографію використовує WireGuard, він суворо контролює криптографічні примітиви і гарантує надійне шифрування пакетів даних, але цей протокол має деякі проблеми з конфіденційністю, для їх усунення необхідно виконувати додаткові налаштування VPN;
- OpenVPN, навпаки, пропонує гнучкі криптографічні алгоритми для автентифікації, шифрування та забезпечення цілісності даних, це дозволяє побудувати таку структуру безпеки, яка буде відповідати конкретній моделі загроз, цей протокол також забезпечує найкращу конфіденційність з усіх розглянутих протоколів;
- L2TP/IPSec загалом вважається надійним та безпечним, він використовує криптографічні алгоритми, які заслуговують на довіру, але цей протокол потенційно скомпрометовано NSA, тому він може викликати скептичне ставлення з боку прихильників конфіденційності.

2.1.3. Порівняння продуктивності

VPN-протоколи можуть працювати на мережевому рівні, але підтримувати різні протоколи передачі даних, які працюють на транспортному рівні. Для передачі даних можна використовувати або протокол керування передачею (Transmission Control Protocol, TCP), або протокол дейтаграм користувача (User Datagram Protocol, UDP).

TCP виконує не тільки передачу пакетів, але й низку інших функцій: підтвердження доставки пакетів; повторну передачу у випадку, якщо пакети не було доставлено; затримку передачі, коли мережа перевантажена; тристороннє рукостискання для перевірки помилок передачі тощо. Таким чином TCP забезпечує надійність передачі, однак його механізми зворотного зв’язку призводять до збільшення накладних витрат на пропускну здатність.

Натомість, UDP піклується виключно про відправку пакетів, і не передбачає жодної перевірки того, чи були ці пакети доставлені. Це дозволяє зменшити затримку, але збільшується ймовірність втрати пакетів.

Отже, TCP працює повільніше, але він надійніший за UDP, тоді як UDP швидший та ефективніший, але не такий надійний, як TCP. [44]

L2TP/IPSec підтримує як UDP, так і TCP. Він використовує порт UDP 500 для початкового обміну ключами, UDP 1701 для конфігурації L2TP та UDP 4500 для обходу NAT (Network Address Translation, NAT). Через цю залежність від фіксованих портів L2TP/IPSec легше заблокувати.

OpenVPN так само підтримує UDP та TCP, але його можна використовувати на будь-якому порту. За замовчуванням він використовує порт UDP 1194 або TCP 443, щоб маскувати свій трафік під веб-трафік та уникати блокування.

WireGuard підтримує виключно UDP, оскільки його розробники віддають пріоритет швидкості, а не надійності. Він використовує порт UDP 51820 за замовчуванням, але його можна змінити на будь-який інший. [45]

Розробники WireGuard тестували і порівнювали продуктивність WireGuard з іншими поширеними протоколами. Результати тестів наведені на рисунку (Рисунок 2.3). [46]

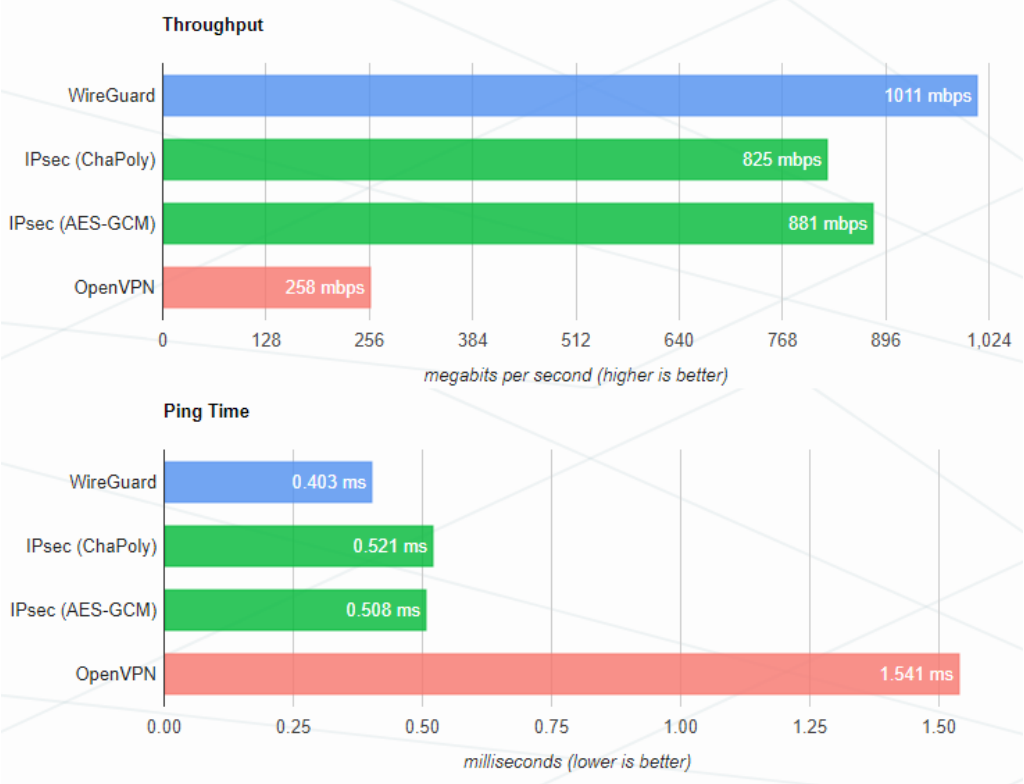


Рисунок 2.3 – Результати тестів продуктивності WireGuard

Тестування проводили за допомогою iPerf3 на платформі Linux, а результати усереднювали упродовж 30 хвилин. Результати тестів показали, що WireGuard у режимі UDP з шифруванням ChaCha20 виявився швидшим за OpenVPN у режимі UDP з шифруванням AES-256 та IPSec в обох конфігураціях, з ChaCha20 і з AES-256. Водночас OpenVPN виявився повільнішим за всі інші конфігурації.

Ось декілька можливих причин, чому OpenVPN показав такі результати:

– OpenVPN має код у 17,5 разів більший за WireGuard, на тій самій обчислювальній потужності він вимагає більше часу на виконання;
– OpenVPN використовує складні криптографічні алгоритми для шифрування та безпеки, вони також впливають на продуктивність;
– OpenVPN працює в просторі користувача, в той час, як WireGuard та L2TP/IPSec вбудовані в Linux, і працюють в просторі ядра системи.

Залежно від того, де працює VPN-протокол – у просторі користувача або у просторі ядра – він у різний спосіб отримує доступ до ресурсів системи.

Якщо протокол працює у просторі ядра, система не може перервати його виконання і перерозподілити його ресурси на користь іншого процесу. Натомість, якщо протокол працює у просторі користувача, пріоритет його виконання і розподілу доступу до ресурсів залежить від загального навантаження на систему, тому його виконання може бути перерване на користь пріоритетнішого процесу. [47]

Окрім переваги у швидкості, WireGuard забезпечує меншу кількість відключень, швидше повторне підключення у разі відключення, і на додаток пропонує нову концепцію, яку його розробники назвали маршрутизацією криптоключів. Маршрутизація криптоключів передбачає пов’язування відкритих ключів однорангових пристроїв з переліком IP-адрес тунелю, дозволених всередині тунелю. Це означає, що коли пристрій змінить свою IP-адресу, WireGuard автентифікує відкритий ключ і пов’яже його з новою IP-адресою, не вимагаючи додаткових ресурсів на повторне підключення. [48]

Це особливо ефективно для віддаленої роботи, оскільки користувачі стають мобільними і переміщуються між різними локаціями, так само їхні пристрої переміщуються між різними мережами і повинні отримувати доступ до VPN вдома, в кафе чи в офісі – і все це протягом одного робочого дня. WireGuard прилаштовується і підтримує роумінг таких пристроїв. Ані L2TP/IPSec, ані OpenVPN не підтримують схожих концепцій.

L2TP/IPSec є складнішим за OpenVPN, і може потребувати додаткової конфігурації між пристроями і маршрутизаторами NAT, але доки його налаштовано належним чином, він працюватиме стабільно.

OpenVPN вважається дуже стабільним, зокрема у бездротових, мобільних та інших ненадійних мережах, де поширені втрати пакетів і перевантаження мережі. [49]

Узагальнена характеристика продуктивності протоколів L2TP/IPSec, OpenVPN та WireGuard представлена у таблиці (Таблиця 2.5).

Таблиця 2.5

Характеристика продуктивності VPN-протоколів

Критерії	L2TP/IPSec	OpenVPN	WireGuard
Порт з’єднання	Фіксований порт UDP або TCP	Будь-який порт UDP або TCP	Будь-який порт UDP
Швидкість	Швидкий	Посередній	Дуже швидкий
Затримка	Середня	Найбільша	Найменша
Контекст виконання	У просторі ядра	У просторі користувача	У просторі ядра в Linux, у просторі користувача
Роумінг	Не підтримує	Не підтримує	Підтримує завдяки маршрутизації криптоключів
Стабільність	Стабільний	Дуже стабільний	Дуже стабільний

Проаналізувавши узагальнену характеристику, можна зробити такі висновки:

- Найшвидшим з розглянутих протоколів є WireGuard, він використовує виключно порт UDP для з’єднань і може працювати у просторі ядра, а ще цей протокол підтримує роумінг пристроїв завдяки своїй концепції маршрутизації криптоключів;
- L2TP/IPSec так само демонструє непогані показники продуктивності, коли використовує порт UDP і працює в просторі ядра, але цей протокол має деякі виклики конфігурації, що можуть вплинути на його стабільність;
- OpenVPN виявився не дуже швидким у порівнянні з іншими протоколами і єдиним, що працює виключно у просторі користувача, натомість цей протокол забезпечує найкращу стабільність, навіть в ненадійних мережах, особливо, коли використовує порт TCP.

2.2 Методологія експериментального дослідження

Представлена порівняльна характеристика дозволяє сформувати уявлення про ключові відмінності протоколів VPN, що розглядаються у цій роботі, проте цього недостатньо для того, щоб відповісти на поставлені проблемні питання. Отже, є потреба провести окреме дослідження, у якому знайти відповіді на ці питання експериментальним шляхом.

Далі у цьому розділі розглянуті метрики та інструменти, які рекомендовано використовувати для вимірювання продуктивності мережі. Наступною описана методологія експерименту, що визначає основні етапи експерименту та порядок його проведення. Насамкінець, наведені специфікації програмно-апаратного забезпечення, що використовувалось для проведення експерименту.

2.2.1 Метрики та інструменти вимірювання

Продуктивність рішень VPN можна вимірювати різними способами. Два аспекти полягають у тому, які метрики вимірювати та за допомогою яких інструментів. Згідно з дослідженнями, що були розглянуті раніше у цій роботі, під час вимірювання продуктивності мереж рекомендовано використовувати три показники: пропускну здатність, затримку та втрату пакетів. Ці поняття коротко пояснюються нижче.

Пропускна здатність – це обсяг даних, що надсилається з однієї точки в іншу протягом певного періоду часу. Пропускна здатність зазвичай вимірюється в бітах на секунду (біт/с). На цей показник впливає вся інфраструктура каналу, така як фізичне середовище та обчислювальна потужність серед інших факторів.

Затримка визначається як час, необхідний для передачі пакета в одному напрямку, наприклад, від клієнта до сервера. Під час тестування VPN затримка є значенням часу, що зазвичай вимірюється у мілісекундах (мс).

Втрата пакетів вказує на кількість пакетів, що не надійшли від джерела до місця призначення. Це може бути викликано, наприклад, перевантаженням мережі. Цей показник вимірюється як відсоток втрачених пакетів відносно надісланих пакетів (%).

Коли мова йде про подолання ненадійності, мається на увазі те, як VPN долають перешкоди. Ці перешкоди можуть бути штучними або природними, як, наприклад, висока затримка або втрата пакетів. Це можна кількісно визначити, обмеживши вищезазначені показники.

iPerf – це міжплатформний інструмент, який дозволяє виконувати стандартизовані вимірювання продуктивності для будь-якої мережі. iPerf має функціональність клієнта та сервера, і може створювати потоки даних для вимірювання пропускної здатності між двома кінцями в одному або обох напрямках. Типовий вихід iPerf містить звіт із міткою часу про кількість переданих даних і виміряну пропускну здатність.

Цей інструмент широко використовується для вимірювання продуктивності мереж, що підтверджується його використанням у джерелах, цитованих раніше у цій роботі.

Очікується, що продуктивність мережі знизиться під час використання VPN через кілька причин. Однією з причин є фізична відстань між клієнтом і сервером. Інша причина полягає в тому, який тип шифрування використовується в конфігурації VPN. Безпечніші та, відповідно, важчі алгоритми шифрування можуть використовувати більше обчислювальної потужності процесора сервера VPN, що залишає менше ресурсів для використання мережею, і тому продуктивність мережі знижується.

2.2.2 Опис основних етапів дослідження

Проблемне питання, яке підлягає дослідженню у цій роботі – як продуктивність сучасних рішень VPN відрізняється в умовах стабільної та ненадійної мережі. Очевидно, що єдиний метод дослідження, за допомогою якого можна знайти відповідь на поставлене проблемне питання, це метод експерименту.

Отже, можна виділити такі основні етапи дослідження:

1) визначити перелік рішень VPN, з якими проводитиметься експеримент, способи контролю трафіку, метрики та інструменти для вимірювань, а також дані, які необхідно зібрати;
2) протестувати мережу в експериментальній установці без будь-якого рішення VPN, щоб визначити її базову продуктивність;
3) налаштувати і протестувати рішення VPN у трьох різних операційних системах в нормальних умовах та з умовами ненадійності мережі;
4) проаналізувати та порівняти результати, щоб з’ясувати, чи є ознаки різниці в продуктивності між рішеннями VPN.

Далі кожен з цих етапів буде описаний докладніше.

На першому етапі дослідження були визначені три рішення VPN, які доцільно порівнювати в експерименті: L2TP/IPSec, OpenVPN та WireGuard.

Важливим фактом є те, що конфігурації для рішень VPN були встановлені за замовчуванням. Для всіх рішень VPN налаштування конфігурації за замовчуванням було збережено, наскільки це було можливо, замість уніфікації налаштувань щодо, наприклад, мережевих протоколів, криптографічних алгоритмів чи варіантів стиснення. Рішення не змінювати налаштування за замовчуванням було мотивовано припущенням, що розробники рішень VPN будуть найбільш кваліфікованими для забезпечення належних (безпечних) конфігурацій для своїх власних рішень VPN. Отже, конфігурації за замовчуванням використовуються в усіх випадках тестування.

Крім того, оскільки WireGuard розроблено таким чином, щоб його було просто розгортати та використовувати, він має менше параметрів конфігурації, ніж L2TP/IPSec та OpenVPN.

На основі аналізу, описаному в першому розділі цієї роботи, було визначено набір програмного та апаратного забезпечення, необхідного для проведення експерименту. Аналіз також показав, які метрики використовувати під час тестування. Налаштування експериментальної установки було виконано таким чином, щоб максимально спростити її, але мати один сервер та один клієнт для кожної операційної системи, а також мати можливість контролювати та відстежувати мережевий трафік. Це було зроблено, щоб мінімізувати зовнішній вплив на мережу, і отримати максимально достовірні результати.

Метрики мережі, що були визначені як важливі для тестування у пов’язаних роботах з першого розділу під час дослідження проблеми:

- пропускна здатність, Мбіт/с;
- затримка, 400 мс (перша ненадійність);
- втрата пакетів, 1% (друга ненадійність).

Ці метрики також є частиною змінних у цьому експерименті. К. Волін та співавтори у книзі «Експерименти в програмній інженерії» стверджують, що в експерименті є два типи змінних: залежні та незалежні. Залежні змінні – це змінні, які тестуються, а незалежні – ті, якими маніпулюють і керують. У цьому дослідженні залежною змінною є пропускна здатність, а дві ненадійності, затримка та втрата пакетів, є незалежними змінними. [50]

Згідно з документом служби підтримки Cisco, максимальна рекомендована затримка для використання VoIP (Voice over IP, VoIP) становить 400 мс, тому саме це значення затримки було вирішено використовувати. [51]

Значення втрати пакетів у 1% було вибрано, оскільки, згідно з дослідженням Ю. Полацького, прийнятний рівень втрати пакетів має бути

нижче 10% у викликах VoIP, але оскільки VPN додає накладні витрати та повторну передачу, що відбувається за такого обсягу втрачених пакетів, тестування з таким високим відсотком втрати пакетів закінчилось повним розривом VPN-з’єднання. Тому відсоток втрати пакетів було знижено до 1%, оскільки будь-яке значення, вище за це, порушувало з’єднання і не дозволяло отримати узгоджені результати. [52]

За інструмент для вимірювання продуктивності мережі було обрано застосунок iPerf3, версія 3.1.3. Ось, чим обґрунтований такий вибір:

- iPerf3 відповідає потребі тестувати пропускну здатність і містить передані дані, тобто можна відстежити, яка кількість пакетів не надійшла до місця призначення;
- iPerf3 сумісний з усіма трьома операційними системами, задіяними у цьому експерименті: Windows, Linux та macOS;
- iPerf3 використовувався в подібних дослідженнях для вимірювання подібних метрик, наприклад, для аналізу продуктивності широкосмугового бездротового зв’язку в Каліфорнії. [53]

На другому етапі дослідження було виконано тестування мережі в експериментальних налаштуваннях на всіх клієнтах без будь-якого рішення VPN. Тести проводились, щоб визначити базовий результат того, як введення модифікацій ненадійності в мережу вплине на клієнт-серверне з’єднання без VPN. З цими даними надалі порівнюватимуться результати тестування VPN.

Третій етап дослідження – це тестування VPN. Тестування було розроблено для запуску тестів по одному клієнту за раз. Один стандартний тест iPerf складається з 10-секундного безперервного тестування та звітування про пропускну здатність з інтервалом в 1 секунду. Як сказано в документації користувача, iPerf працює, записуючи масив певного обсягу байтів, певну кількість разів. За замовчуванням цей обсяг становить 128 КБ для TCP та 8 КБ для UDP. Це означає, що застосунок надсилає пакети через мережу з фіксованим розміром вікна, по суті, буфером кількості даних,

надісланих до того, як їх підтвердить одержувач. Крім того, застосунок працює в пам’яті, тому диск взагалі не задіяний під час вимірювання, а дані, що надсилаються, є шумом (випадковими даними). iPerf має дуже просту архітектуру, де одна кінцева точка є сервером, а інша – клієнтом. Будь-яка з цих точок може бути сервером або клієнтом. [54]

Тест за замовчуванням, який використовувався в цьому дослідженні, – це завантаження даних із клієнта на сервер. Кожен тест повторювався 50 разів. Перший 1-секундний інтервал тесту було видалено з набору даних, щоб дати можливість встановити належне з’єднання, оскільки тест розпочинався одразу після ввімкнення VPN.

Щоб мінімізувати ймовірні перешкоди, під час тестування однієї VPN всі інші були вимкнені, і клієнти, і сервер були перезапущені та залишені в режимі очікування на 10 хвилин, щоб дозволити будь-які автоматичні оновлення та стабілізувати роботу машин.

Для введення умов ненадійності мережі обраний маршрутизатор повинен підтримувати формування трафіку. Програмний маршрутизатор pfSense постачається з такою опцією за замовчуванням, вона називається формувачем трафіку (Traffic Shaper). У її налаштуваннях можна додати будь-яке значення затримки та відсоток втрати пакетів. За допомогою інструменту формування трафіку на маршрутизаторі pfSense було досягнуто тестування в умовах ненадійності мережі, таким чином були додані дві умови ненадійності: затримка у 400 мс та 1% втрати пакетів. Оскільки всі дані між клієнтами і сервером VPN проходять через маршрутизатор, він застосовує формування трафіку в обох напрямках. Отже, клієнти були протестовані в експериментальній установці з трьома різними налаштуваннями маршрутизатора: базовий тест, без будь-яких умов ненадійності, тест з першою ненадійністю і тест з другою ненадійністю.

Для імітації деградацій мережі використовуються затримка та втрата пакетів, оскільки основна увага приділяється VoIP. VoIP є звичайним випадком застосування для віддалених працівників, водночас чутливим до

швидкості мережі, коли використовується через VPN або загалом в мережі Інтернет. Це відрізняє VoIP від, наприклад, перегляду сторінок, оскільки на завантаження сторінок затримка та втрата пакетів впливають значно менше, в той час як потокове передавання VoIP чутливіше до таких деградацій.

На останньому, четвертому етапі дослідження, щоб відповісти на поставлене проблемне питання, необхідно зібрати дані, обробити їх, проаналізувати і дійти твердого висновку. На цьому етапі також буде розглянуто валідність експерименту, щоб переконатися у високій якості отриманих результатів.

2.2.3 Програмно-апаратне забезпечення

Далі будуть наведені специфікації апаратного забезпечення, що використовувалось для проведення експерименту. Ці специфікації також містять інформацію про програмне забезпечення, розгорнуте на цьому апаратному забезпеченні, що безпосередньо стосується експерименту та його результатів.

Специфікації клієнтів в експериментальній установці наведені у таблиці (Таблиця 2.6).

Таблиця 2.6

Специфікації апаратного забезпечення – клієнти

Специфікації	Перший клієнт	Другий клієнт	Третій клієнт
Операційна система	Microsoft Windows 10, версія 21H2	Ubuntu, версія 22.04 LTS	macOS Monterey, версія 12.3.1
Процесор	Intel Core i7-6700, 3.4 ГГц	Intel Core i7-6700, 3.4 ГГц	Intel Core i7-7820HQ, 2.9 ГГЦ
Оперативна пам’ять	DDR3 16 ГБ	DDR3 16 ГБ	DDR3 16 ГБ
Мережевий адаптер	Realtek GbE	Realtek GbE	Belkin USB-C to

	LAN, 1 Гбіт/с	LAN, 1 Гбіт/с	Gigabit Ethernet Adapter, 1 Гбіт/с
Версія OpenVPN	2.5.7	2.5.5	3.3.6
Версія WireGuard	0.5.3	1.0.20210606	1.0.15

Специфікації сервера в експериментальній установці наведені у таблиці (Таблиця 2.7).

Таблиця 2.7

Специфікації апаратного забезпечення – сервер

Специфікації	Сервер
Операційна система	Ubuntu Server, версія 20.04 LTS
Процесор	AMD Opteron X3216
Оперативна пам’ять	DDR3 16 ГБ
Версія OpenVPN	2.5.5
Версія WireGuard	1.0.20210606

Специфікації маршрутизатора в експериментальній установці наведені у таблиці (Таблиця 2.8).

Таблиця 2.8

Специфікації апаратного забезпечення – маршрутизатор

Специфікації	Маршрутизатор
Операційна система	pfSense 2.6.0-RELEASE
Процесор	Intel Xeon E3-1230 v3
Оперативна пам’ять	DDR3 16 GB
Мережевий адаптер	1) Realtek GbE LAN, 1 Гбіт/с (внутрішня мережа) 2) PCI-e Realtek RTL8125, 2.5GBASE-T (Інтернет)

Підсумовуючи, в експериментальній установці були задіяні три локальні клієнти та маршрутизатор, а також віддалений сервер. Окрім

наведеного апаратного забезпечення також використовувався комутатор, за допомогою якого всі троє клієнтів були об’єднані в локальну мережу.

2.3 Експериментальне дослідження продуктивності

Далі у цьому розділі докладніше описано налаштування експерименту, те, як проводилось тестування, і як оброблялись дані. Також пояснюються загрози валідності, пов’язані з цим дослідженням. Насамкінець, наведено узагальнену таблицю з результатами експерименту.

2.3.1 Огляд експерименту

Топологія мережі в експерименті наведена на рисунку (Рисунок 2.4). Клієнти були від’єднані від комутатора на час, поки не виконувалось тестування, з метою усунути перешкоди, що могли створюватись, наприклад, запитами DHCP (Dynamic Host Configuration Protocol, DHCP).

Експериментальна установка містить одну машину, яка працює як програмний маршрутизатор під керуванням pfSense. pfSense є спеціалізованим дистрибутивом операційної системи FreeBSD з відкритим кодом, що призначений виконувати функції мережевого екрана та мережевого маршрутизатора. [55]

Маршрутизатор має два контролери мережевого інтерфейсу (Network Interface Controller, NIC), один з яких може працювати зі швидкістю до 1 Гбіт/с, інший – до 2.5 Гбіт/с. Перший порт було підключено до комутатора, що з’єднує клієнтську мережу, другий – до сервера VPN. Комутатор між маршрутизатором та клієнтами використовувався в експериментальній установці для того, щоб спростити тестування, оскільки маршрутизатор мав лише два порти, і тому його потрібно було вручну відключати та підключати щоразу, коли виконувалось тестування на іншому клієнті. Якби комутатор

створив будь-яку затримку, вона була б однаковою для всіх клієнтів та VPN, тому її додавання є незначним для цього експерименту.

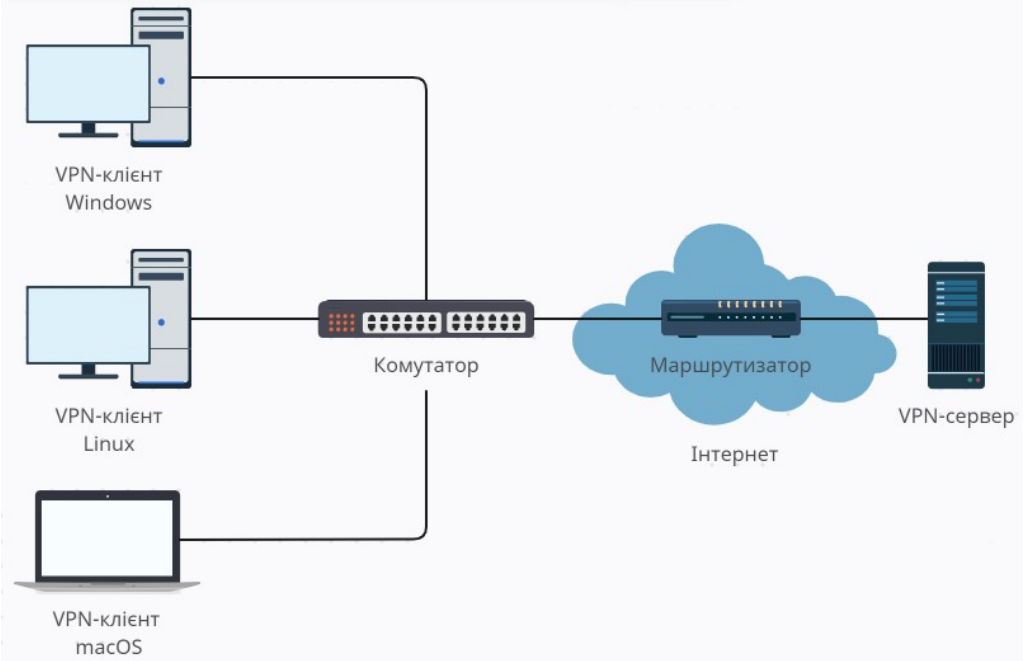


Рис. 2.4 – Топологія мережі в експерименті

Клієнти були встановлені з відповідними операційними системами: один з Windows 10, один з Ubuntu і ще один з macOS Monterey. Інструмент вимірювання iPerf був встановлений на всіх клієнтах і сервері. Докладна інформація про команди, що використовувались для запуску iPerf в кожній операційній системі, наведена у додатках.

Три рішення VPN – L2TP/IPSec, OpenVPN та WireGuard – були встановлені на всіх трьох клієнтах. Вони були протестовані один за одним на кожному клієнті без введення будь-яких варіантів ненадійності, і результати були зафіксовані. Коли ці тести були виконані, варіанти ненадійності вводилися один за одним, а результати так само фіксувалися. Windows та macOS вбудовано підтримують IPSec, і були налаштовані лише для підключення до сервера IPSec VPN, тоді як Ubuntu використовувала Libreswan. [56]

Конфігурації за замовчуванням трьох рішень VPN наведені у таблиці (Таблиця 2.9).

Таблиця 2.9

Конфігурації рішень VPN за замовчуванням

Параметри	L2TP/IPSec	OpenVPN	WireGuard
Алгоритм шифрування	AES-256	AES-256	ChaCha20
Стиснення	Так	Ні	Ні
Багатопотоковість	Так	Ні	Так

У таблиці описані важливі аспекти VPN, які необхідно розглянути у цьому дослідженні. Одним з аспектів є алгоритм шифрування даних, оскільки від того, яке шифрування використовується, залежить продуктивність VPN. Інший аспект – це стиснення корисного навантаження, яке за замовчуванням вимкнено на всіх протестованих рішеннях VPN. Останній аспект, багатопотоковість, вказує на те, чи може шифрування та дешифрування виконуватись на кількох ядрах процесора. Сучасні процесори зазвичай мають більше одного ядра, а багатопотоковість надає VPN можливість виконувати паралельне шифрування на кількох ядрах процесора. Очевидно, що VPN, яка використовує таку функціональність, швидше виконуватиме шифрування та дешифрування, а отже, матиме більшу продуктивність.

Рішення L2TP/IPSec, встановлене на сервері в експерименті, використовує xl2tpd, що є реалізацією L2TP, яка підтримується Xelerance Corporation. Реалізація IPSec, що використовувалась, це Libreswan із попереднім спільним ключем, іменем користувача та паролем. Встановлення було виконано за допомогою сценарію, рекомендованого виробником. Щоб забезпечити послідовність експерименту, усі рішення VPN було встановлено та протестовано з конфігурацією за замовчуванням.

Конфігурації інших протоколів для сервера це конфігурації за замовчуванням, що постачались з останніми версіями протоколів, доступними на момент проведення експерименту.

Налаштування мережевого трафіку виконується на програмному маршрутизаторі pfSense, за допомогою формувача трафіку (Traffic Shaper). Формувач трафіку містить обмежувачі (Limiters), а вони використовують інструмент, який має назву dummynet. [57]

За допомогою dummynet можна вводити варіанти ненадійності для тестування безпосередньо між вузлами VPN, без використання спеціальних інструментів чи програмного забезпечення на самих вузлах. Якщо використовувати додаткові інструменти або програмне забезпечення на вузлах, це означатиме, що для формування трафіку знадобиться обчислювальна потужність, яка вже не буде доступна для VPN.

Надійність застосування iPerf разом із dummynet для тестування продуктивності мережі доведено у багатьох дослідженнях, де ці інструменти використовувались подібним чином, dummynet для моделювання поганого мережевого з’єднання та iPerf для передачі даних між вузлами. [58, 59]

Перша група тестів проводилась зі звичайним стабільним мережевим з’єднанням і конфігурацією, причому всі мережеві порти та кабелі в експериментальній установці здатні працювати зі швидкістю до 1 Гбіт/с.

Щоб перевірити, як продуктивність рішень VPN відрізняється в умовах ненадійної мережі, для другої та третьої групи тестів мережу було погіршено шляхом налаштування трафіку на додавання затримки або відкидання певного відсотку пакетів під час виконання тестів iPerf.

Щоб забезпечити узгодженість середовища між різними операційними системами, мережа використовує однакові кабелі та налаштування сервера, і жодні інші служби не працюють. Під час тестування в стабільній мережі та з введеними варіантами ненадійності на тій самій операційній системі налаштування однакові, використовується те саме

обладнання, інші служби так само не працюють. Невідповідностей не очікується.

Блок-схему, що відображає порядок проведення експерименту, наведено на рисунку (Рисунок 2.5).

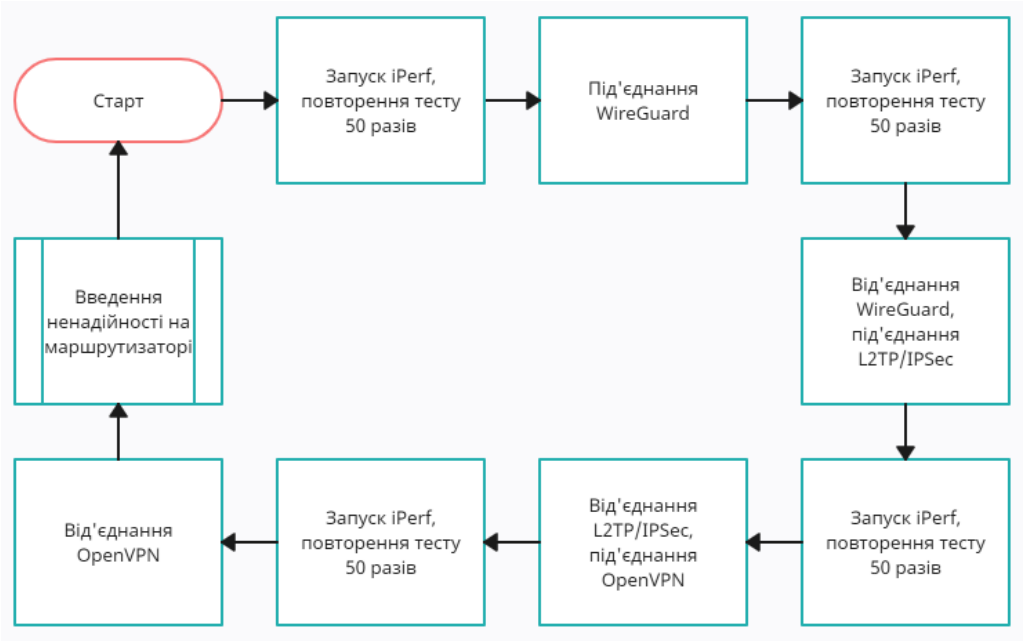


Рис. 2.5 – Блок-схема експерименту

Кроки, зображені на блок-схемі, повторювались тричі: один цикл для тестування без будь-яких деградацій мережі, один цикл для першої ненадійності та один цикл для другої ненадійності.

Автоматизація вбудована з налаштуваннями на pfSense для ненадійності. У Windows тести запускаються за допомогою сценаріїв та планувальника завдань. В Ubuntu це виконується через сценарії та crontab. У macOS для цього використовуються сценарії та робочий процес (вбудований інструмент для автоматизації, подібний до планувальника завдань Windows та crontab в Ubuntu). Докладніша інформація наведена у додатках.

Дані збираються через iPerf. Застосунок відображає всі результати як під час тестування, так і після його завершення, демонструючи підсумки

проведених тестів. Вся ця інформація зберігається в текстових документах за допомогою сценаріїв.

Усі зібрані дані фільтруються в електронну таблицю, щоб отримати середнє значення пропускної здатності для всіх VPN у всіх операційних системах. Так само обробляються дані, отримані після проведення базових тестів, і окремо з двома варіантами ненадійності.

2.3.2 Загрози валідності

Для експерименту, такого як у цьому дослідженні, є три категорії валідності, які потребують обробки відповідних загроз: внутрішня, зовнішня та валідність висновку. Зауважу, що, оскільки у цьому експерименті тестувалися лише налаштування VPN за замовчуванням, результати можуть не застосовуватися, якщо ті самі рішення VPN будуть налаштовані відповідно до певних цілей. Такими цілями може бути підвищення пропускної здатності або, наприклад, зміцнення безпеки. Якщо в конфігурацію VPN внесено будь-які зміни, результати можуть відрізнятися від результатів цього дослідження.

На додаток до загроз валідності Волін у своїй книзі пояснює, що під час проведення експериментів необхідно пам’ятати про упередження. У цьому дослідженні було виявлено та усунено два різновиди упередження, а саме монометод і моноопераційо. Упередження моноопераційї усувається шляхом додавання у тестування більше ніж однієї незалежної змінної. Упередженння монометоду усувається використанням більш ніж одного тесту. У цьому дослідженні було виконано кілька тестів для кожного випадку тестування.

Отже, загрози валідності, які були виявлені та оброблені:

1) Загрози внутрішній валідності містять все стороннє програмне забезпечення, що використовується. У найкращому випадку припускається, що використовувані вимірювальні інструменти

працювали саме так, як передбачалося. На подолання цієї загрози спрямовано дослідження інших робіт, у яких використовувалися ці інструменти та програмне забезпечення, і намагання використовувати їх подібним чином, а також слідування офіційній документації цих інструментів і програмного забезпечення. Іншим кроком для забезпечення внутрішньої валідності є порівняння результатів цього дослідження з іншими пов’язаними дослідженнями, щоб з’ясувати, чи вказують отримані результати на очікуваний напрямок.

2) У цьому дослідженні існують загрози зовнішній валідності. Якщо інше дослідження матиме на меті відтворити подібну експериментальну установку, результати можуть відрізнятися, оскільки операційні системи та програми, задіяні в цьому дослідженні, оновлюються, а разом з цим в них вносяться виправлення та вдосконалення, що може вплинути на продуктивність. Таким чином, одне рішення VPN, яке демонструє низьку пропускну здатність, може збільшити продуктивність після оновлення. І навпаки, старіші системи могли мати інші результати так само, як операційні системи та рішення VPN ймовірно мали іншу продуктивність у попередніх версіях.
3) Загроза достовірності висновку полягає в тому, що це симуляція проблеми реального світу. Ненадійність мережі в експерименті не є реальною, а моделюється маршрутизатором, і цілком можливо, що під час тестування в реальному сценарії, на реальному обладнанні та з реальною ненадійністю мережі результати можуть відрізнятися. Ще дві загрози достовірності висновків, що визначені як можливі в цьому дослідженні, це низька статистична потужність і ненадійність вимірювань. Обидві загрози прив’язані до експерименту та обробляються шляхом забезпечення того, що тести повторюються

багато разів і що будь-яка велика невідповідність вимірювань пропускається в результатах.

Загрози валідності обробляються з метою забезпечити високу якість результатів експерименту.

2.3.3 Результати експерименту

Результати експерименту представлені відповідно до рекомендацій М. Берндтссона у книзі «Дипломні проекти: посібник для студентів з комп’ютерної науки та інформаційних систем». [60]

Представлені результати є значеннями, отриманими на момент, коли iPerf надсилає пакети на сервер, а сервер отримує та відображає значення. Усі результати є усередненими значеннями 50 тестів, що проводились для кожного випадку тестування. Як було зазначено у методології експерименту, тестування проводилось у 36 різних випадках. Наприклад, тестування рішення WireGuard в операційній системі Windows без деградацій мережі – це один випадок.

Узагальнені результати тестування пропускної здатності, отримані в усіх 36 випадках, наведені у таблиці (Таблиця 2.10). Усі значення в таблиці наведені у Мбіт/с.

Таблиця 2.10

Результати тестування пропускної здатності

Операційні системи	Без VPN	WireGuard	L2TP/IPSec	OpenVPN
	Без деградацій мережі			
Windows	908,5	749,5	309,9	270,7
Ubuntu	924,6	847,9	816,7	366,4
macOS	879,4	599,3	758,3	227,7
	Перша ненадійність – затримка			

Windows	3,9	3,7	0,7	1,2
Ubuntu	51,6	48,9	49,4	4,3
macOS	27,1	5,7	16,1	22,6
	Друга ненадійність – втрата пакетів			
Windows	96,1	91,6	81,9	91,4
Ubuntu	267	171,3	141,8	91,7
macOS	106,2	79,8	63,5	69,8

Отримані результати будуть докладно проаналізовані та порівняні у третьому розділі цієї роботи.

2.4 Висновки за розділом

У цьому розділі було надано розгорнуту характеристику трьох протоколів VPN: WireGuard, L2TP/IPSec та OpenVPN. Протоколи були охарактеризовані за трьома основними аспектами: доступність, безпека та продуктивність.

Щоб відповісти на проблемне питання – як продуктивність сучасних рішень VPN відрізняється в умовах стабільної та ненадійної мережі – було вирішено провести експериментальне дослідження. На підготовчому етапі було описано методологію та програмно-апаратний комплекс дослідження.

Для експерименту була налаштована експериментальна установка, що складається з трьох локальних клієнтів і маршрутизатора, а також віддаленого сервера. Три рішення VPN тестувались у трьох різних операційних системах – Windows, Ubuntu та macOS – в стабільних умовах та з двома варіантами ненадійності, загалом це 36 випадків тестування разом із базовими тестами без VPN. Метрикою продуктивності VPN була пропускна здатність, її вимірювали за допомогою iPerf. Варіантами ненадійності були

затримка та втрата пакетів, їх налаштовували за допомогою dummynet на маршрутизаторі.

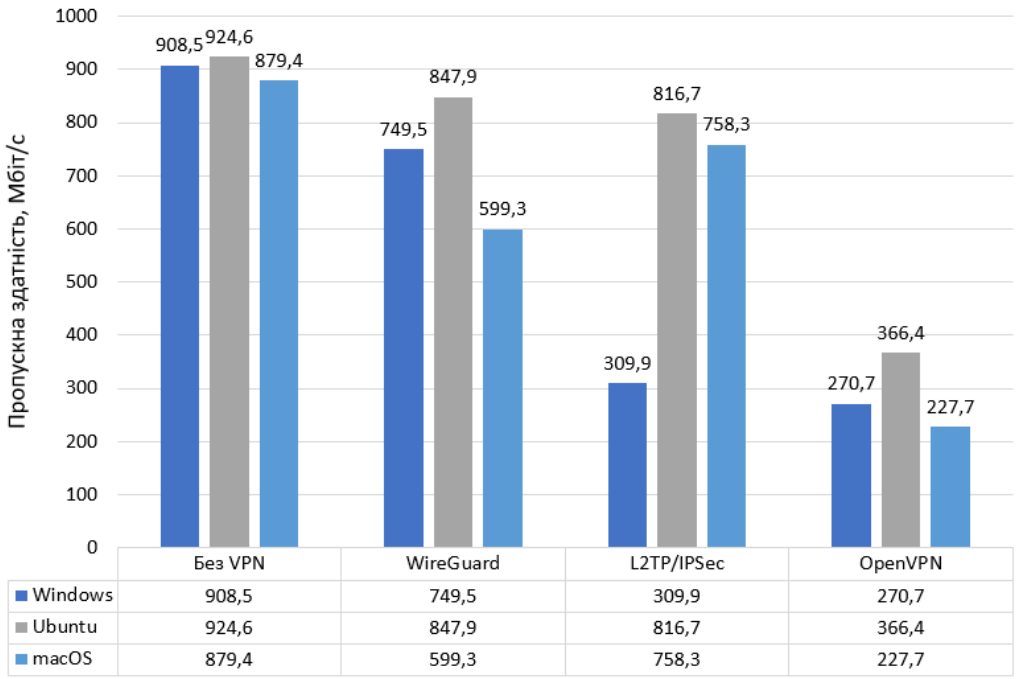
Дані, зібрані через iPerf, були усереднені та представлені у таблиці результатів експерименту. Крім того, були розглянуті та оброблені загрози валідності експерименту.

РОЗДІЛ 3
АНАЛІЗ РЕЗУЛЬТАТІВ І ОБГРУНТУВАННЯ РЕКОМЕНДАЦІЙ

3.1 Аналіз результатів експерименту

Спершу було виміряно базову продуктивність – значення пропускної здатності в стабільній мережі, без під’єднання VPN. Це значення необхідно для того, щоб оцінити та порівняти вплив різних рішень VPN на пропускну здатність мережі. Далі результати, отримані з під’єднаними VPN, будуть порівнюватися як між собою, так і з результатами базових тестів.

Усереднені результати вимірювань пропускної здатності без деградацій мережі наведені на рисунку (Рисунок 3.1).



	Без VPN	WireGuard	L2TP/IPSec	OpenVPN
Windows	908,5	749,5	309,9	270,7
Ubuntu	924,6	847,9	816,7	366,4
macOS	879,4	599,3	758,3	227,7

Рис. 3.1 – Тестування без деградацій мережі

Як і очікувалося, після під’єднання VPN пропускна здатність мережі знижувалась. Крім того, у порівнянні з базовою продуктивністю мережі, спостерігалися декілька очевидних тенденцій.

Найбільшого впливу на продуктивність мережі завдав OpenVPN, знизивши пропускну здатність в середньому на 60% в Ubuntu, на 70% у Windows та на 74% у macOS. OpenVPN виявився найповільнішим в усіх трьох операційних системах, причому майже в усіх випадках тестування пропускна здатність була у кілька разів нижчою у порівнянні з іншими рішеннями. У найгіршому випадку тестування, під час під'єднання OpenVPN у macOS, пропускна здатність впала з 879,4 Мбіт/с у базових тестах до 227,7 Мбіт/с у тестах з VPN.

L2TP/IPSec став найпродуктивнішим рішенням для macOS, разом з ним пропускна здатність знизилась на 14%, і в середньому становила 758,3 Мбіт/с. Найкраще він продемонстрував себе в Ubuntu, знизивши пропускну здатність всього на 12%, проте поступився першістю в цій операційній системі іншому рішенню. Несподівано, найменш продуктивним L2TP/IPSec виявився у Windows, пропускна здатність становила в середньому 309,9 Мбіт/с, що на 66% менше за базову продуктивність.

Найліпшу продуктивність у Windows та Ubuntu продемонстрував WireGuard, з цим рішенням середня пропускна здатність становила 749,5 Мбіт/с та 847,9 Мбіт/с відповідно. У найкращому випадку тестування, під час під'єднання WireGuard в Ubuntu, пропускна здатність знизилась лише на 8%. Найменш ефективним WireGuard виявився для macOS, знизивши пропускну здатність на 32%, в середньому до 599,3 Мбіт/с, та все ж це непоганий результат, особливо в порівнянні з OpenVPN.

Далі за допомогою маршрутизатора в експериментальну мережу було введено першу ненадійність – затримка у 400 мс. Щоб поспостерігати, як різні рішення долатимуть цей варіант ненадійності та порівняти їхню продуктивність в умовах погіршеного мережевого з'єднання, спочатку були виконані базові тести без VPN, а потім по черзі з кожним рішенням.

Усереднені результати вимірювань пропускної здатності з першою ненадійністю наведені на рисунку (Рисунок 3.2).

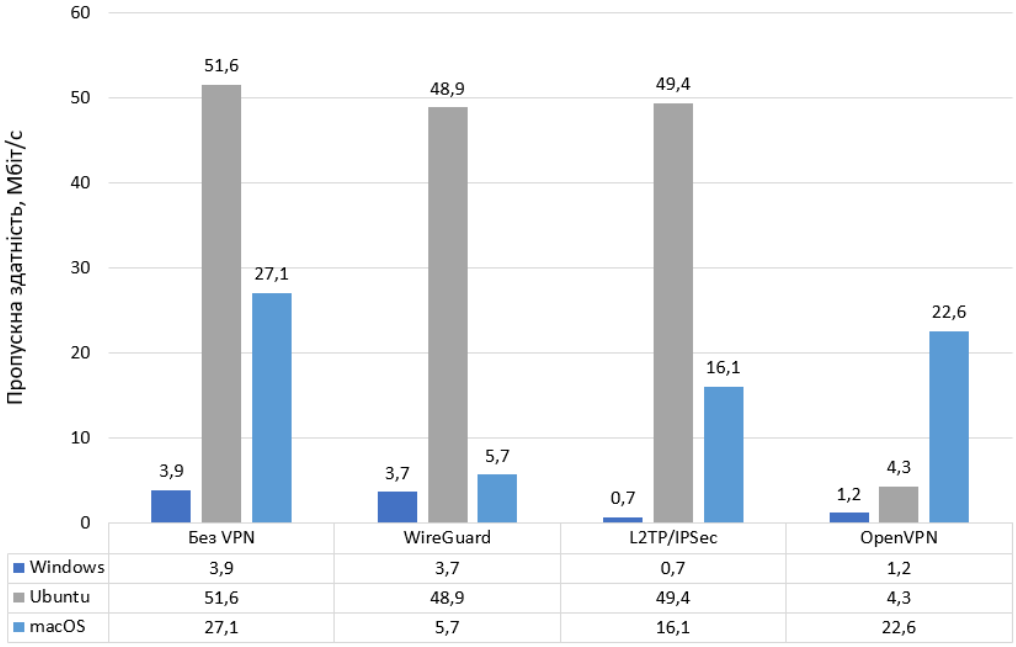


Рис. 3.2 – Тестування з першою ненадійністю

Після введення затримки істотне зниження пропускної здатності спостерігалося в усіх трьох операційних системах. Найкращі результати у тестах без VPN з цим варіантом ненадійності продемонструвала Ubuntu – в середньому 51,6 Мбіт/с проти 27,1 Мбіт/с у macOS та 3,9 Мбіт/с у Windows. MacOS мала посередні результати як у тестах без VPN, так і у тестах з будь-яким рішенням VPN. Windows мала найгірші результати в усіх випадках тестування.

L2TP/IPSec найліпше впорався із затримкою в Ubuntu, разом з ним середня пропускна здатність була трохи меншою, ніж у тестах без VPN, і становила 49,4 Мбіт/с. У macOS він мав посередні результати, знизивши пропускну здатність на 41%, в середньому до 16,1 Мбіт/с, але інше рішення у цій операційній системі виявилось кращим.

Найпродуктивнішим під час тестування із затримкою у macOS виявився OpenVPN, середня пропускна здатність з цим рішенням знизилась лише на 17%, порівняно з тестами без VPN, і становила 22,6 Мбіт/с. Проте у Windows та Ubuntu результати були невтішні, пропускна здатність разом з OpenVPN знизилась на 69% та 92% відповідно.

WireGuard впорався із затримкою в Ubuntu так само добре, як і L2TP/IPSec. Пропускна здатність становила в середньому 48,9 Мбіт/с, тобто різниця із тестами без VPN становила близько 5%. У macOS пропускна здатність була на 79% нижчою, у порівнянні з тестами без VPN, і становила в середньому 5,7 Мбіт/с, що є найнижчим результатом у цій операційній системі. Зауважу, що WireGuard продемонстрував кращі результати у Windows, ніж OpenVPN та L2TP/IPSec, знизивши пропускну здатність лише на 5%, та попри це вона була дуже низькою.

Далі за допомогою маршрутизатора в експериментальній мережі було усунено першу ненадійність, і додано другу – 1% втрати пакетів. Порядок тестування залишився таким самим, як і у попередньому випадку, спочатку проводились тести без VPN, а потім окремо з кожним рішенням.

Усереднені результати вимірювань пропускної здатності з другою ненадійністю наведені на рисунку (Рисунок 3.3).

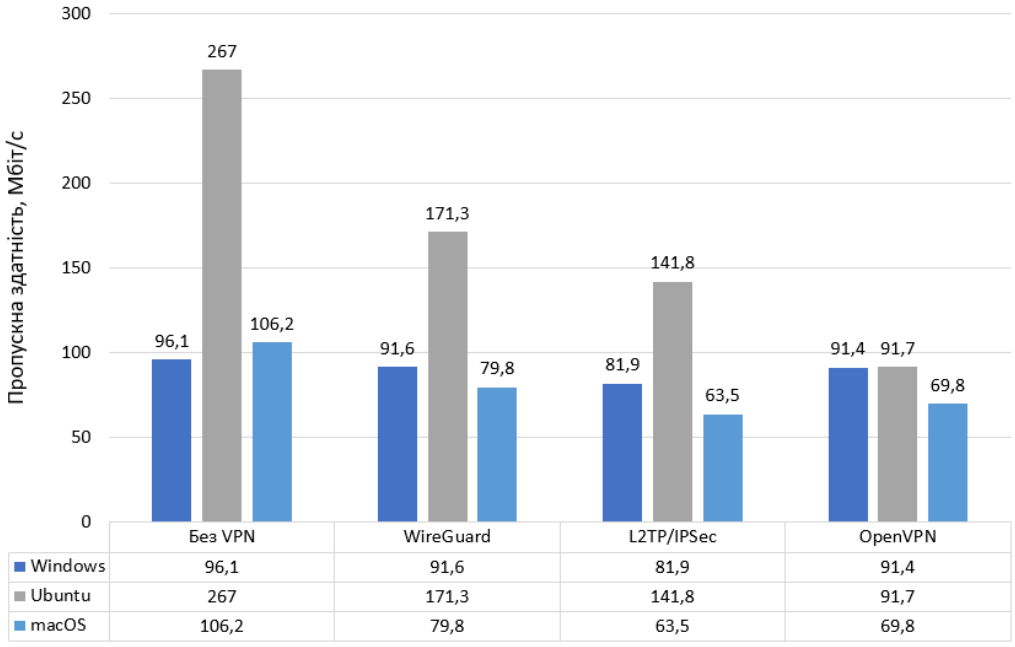


Рис. 3.3 – Тестування з другою ненадійністю

Після додавання втрати пакетів так само спостерігалося зниження пропускної здатності в усіх трьох операційних системах, щоправда воно було

не таким значним, як у випадку із затримкою. У тестах без VPN з цим варіантом ненадійності найкращі результати знову продемонструвала Ubuntu, пропускна здатність становила в середньому 267 Мбіт/с. Для порівняння, результати у macOS та Windows були у кілька разів меншими – 106,2 Мбіт/с та 96,1 Мбіт/с відповідно.

Найліпше з втратою пакетів впорався WireGuard, причому в усіх трьох операційних системах. У Windows пропускна здатність знизилась на несуттєві 5%, в середньому до 91,6 Мбіт/с, хоча справедливо буде відзначити, що у Windows всі три рішення продемонстрували майже однакові результати. У macOS перевага WireGuard була помітнішою, пропускна здатність знизилась на 25%, і становила в середньому 79,8 Мбіт/с, в той час, як з іншими рішеннями відсоток зниження був більшим. В Ubuntu середня пропускна здатність становила 171,3 Мбіт/с, що на 36% менше, у порівнянні з тестами без VPN, і це найкращий результат у цій операційній системі.

OpenVPN, як і WireGuard, непогано впорався з втратою пакетів у Windows, різниця з тестами без VPN не перевищила 5%. Він мав посередні результати в macOS, знизивши пропускну здатність на 34%. В Ubuntu OpenVPN виявився найменш продуктивним, пропускна здатність становила в середньому 91,7 Мбіт/с, у порівнянні з тестами без VPN це на 66% менше.

L2TP/IPSec продемонстрував найнижчі результати у Windows та macOS, знизивши пропускну здатність в середньому до 81,9 Мбіт/с та 63,5 Мбіт/с відповідно. Для Windows різниця з тестами без VPN склала 15%, для macOS – близько 40%. В Ubuntu результати були посередніми, пропускна здатність знизилась на 47%, але зауважу, що загалом різниця з іншими рішеннями була невелика.

Проаналізувавши результати експерименту, можна зробити декілька обґрунтованих висновків:

– Найефективнішими рішеннями VPN в стабільній мережі були WireGuard та L2TP/IPSec. WireGuard забезпечив найліпшу продуктивність у Windows, це пояснюється низкою факторів, одним

з яких може бути те, що це рішення у конфігурації за замовчуванням застосовує багатопотоковість. L2TP/IPSec був найліпшим у macOS, це пов’язано з тим, що він підтримується вбудовано, тому його можна вважати добре інтегрованим та оптимізованим для цієї операційної системи. Крім того, L2TP/IPSec за замовчуванням застосовує стиснення даних та багатопотоковість. В Ubuntu вони мали наближені результати, з невеликою перевагою з боку WireGuard.

– У стабільній мережі як у базових тестах, так і з будь-яким рішенням VPN найкращі результати було зафіксовано в Ubuntu. Це можна пояснити тим, що мережеві протоколи Ubuntu краще обробляють розмір вікна тестів iPerf у цьому конкретному експерименті, порівняно з іншими операційними системами.
– Затримка спричинила істотне зниження продуктивності в усіх випадках тестування. Втім, якщо порівнювати продуктивність рішень VPN, найліпше із затримкою в Ubuntu впорався L2TP/IPSec, трохи нижчими були результати WireGuard. У Windows в усіх випадках тестування із затримкою продуктивність була дуже низькою, але серед інших найліпше із затримкою в ній впорався WireGuard. OpenVPN мав найкращі результати із затримкою у macOS.
– У всіх випадках тестування із затримкою, окрім тестів з OpenVPN, найліпша продуктивність спостерігалась в Ubuntu. Це пов’язано з тим, як операційні системи обробляють розмір вікна та повторну передачу при затримці, і вочевидь, мережевий стек Ubuntu робить це краще за macOS, і набагато краще за Windows.
– WireGuard найкраще впорався із втратою пакетів в усіх трьох операційних системах. Хоча у Windows та macOS всі три рішення мали дуже близькі результати, незначна перевага була за WireGuard. Під час тестування із втратою пакетів в Ubuntu ця різниця була

помітнішою. Це пояснюється тим, що WireGuard застосовує механізми, які дозволяють йому обережно долати втрату пакетів.

- У тестах із втратою пакетів усі три рішення VPN були найбільш продуктивними в Ubuntu, і приблизно однаково продуктивними у Windows та macOS. Причини такої тенденції ті самі, що й у випадку із затримкою.
- OpenVPN не став лідером у жодному випадку тестування, окрім тестів із затримкою у macOS. Це очікуваний результат. OpenVPN за замовчуванням не застосовує багатопотоковість, на відміну від двох інших рішень. Крім того, OpenVPN завжди працює у просторі користувача, що менш продуктивно, ніж якби він працював у просторі ядра.
- WireGuard виявився найпродуктивнішим рішенням у більшості випадків тестування. Він був дуже ефективним у Windows та Ubuntu, і дещо поступався іншим рішенням у macOS. Загалом WireGuard продемонстрував саме такі результати, які від нього й очікувались.

3.2. Рекомендації з вибору протоколу VPN

На основі результатів проведеного дослідження були обґрунтовані рекомендації з вибору найліпшого з погляду продуктивності протоколу VPN для забезпечення віддаленого доступу з різних операційних систем, за різних умов надійності мережевого з’єднання.

Якщо сервіс VPN розгортається у надійному мережевому з’єднанні в середовищі Windows, відповідно до результатів цього дослідження, рекомендовано використовувати WireGuard.

Якщо це середовище на базі Linux (наприклад, Ubuntu), з надійним мережевим з’єднання рекомендовано використовувати WireGuard або

L2TP/IPSec. Обидва рішення є ефективними на Linux, для найкращих результатів рекомендується WireGuard.

Якщо середовище складається переважно з пристроїв під керуванням macOS, за умов надійного мережевого з’єднання рекомендовано використовувати L2TP/IPSec.

Якщо розгортати VPN у ненадійному мережевому з’єднанні, де очікується затримка, у середовищі Windows, рекомендовано використовувати WireGuard.

Якщо середовище базується на Linux, з ненадійним мережевим з’єднанням, схильним до затримки, рекомендовано використовувати L2TP/IPSec або WireGuard. Ці два рішення майже однаково ефективно впорались із затримкою на Linux у цьому дослідженні, для найкращих результатів рекомендується L2TP/IPSec.

Якщо середовище переважно під керуванням macOS, а мережеве з’єднання так само ненадійне і схильне до затримки, рекомендовано використовувати OpenVPN або L2TP/IPSec. Для найкращих результатів рекомендується OpenVPN, оскільки він дещо краще впорався із затримкою на macOS у цьому дослідженні.

Якщо розгортання VPN виконується у ненадійному мережевому з’єднанні, де очікується втрата пакетів, у всіх трьох середовищах рекомендовано використовувати WireGuard для найкращих результатів.

Якщо це середовище Windows, OpenVPN буде так само ефективним, як і WireGuard, тому він також рекомендований до використання.

Узагальнені рекомендації наведені у таблиці (Таблиця 3.1).

Таблиця 3.1

Рекомендації з вибору протоколу VPN

Мережеве з’єднання	Windows	Linux	macOS
Надійне	WireGuard	WireGuard	L2TP/IPSec
Затримка	WireGuard	L2TP/IPSec	OpenVPN

Втрата пакетів	WireGuard	WireGuard	WireGuard

Щоб зробити вибір зручнішим, було вирішено побудувати дерево рішень на основі обґрунтованих рекомендацій.

Дерево рішень для вибору найліпшого протоколу VPN з погляду продуктивності, наведено на рисунку (Рисунок 3.4).

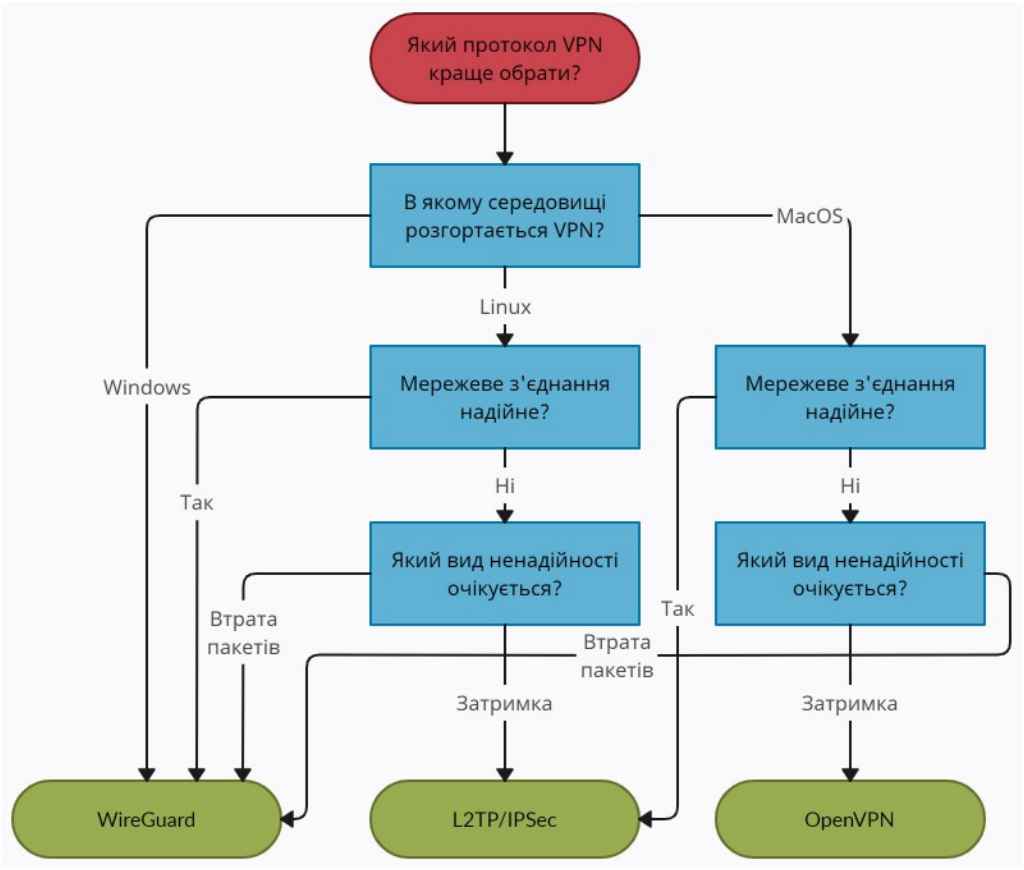


Рис. 3.4 – Дерево рішень

Обґрунтовані рекомендації можуть бути корисними мережевим адміністраторам для оцінки варіантів рішень при розгортанні VPN і забезпеченні віддаленого доступу в умовах ненадійності мережі.

Отримані результати також можуть бути використані як база для проведення інших експериментів із VPN.

Якщо мережевий адміністратор має намір використовувати ту саму конфігурацію, що й у цьому дослідженні, то результати можуть бути

корисними без необхідності проведення власного тестування.

3.3. Напрямки майбутніх досліджень

У майбутніх дослідженнях рекомендується протестувати рішення VPN на іншому апаратному забезпеченні для сервера, маршрутизатора та клієнтів, виміряти та порівняти використання обчислювальних ресурсів під час виконання тестів, розмістити VPN-сервер на іншій операційній системі, наприклад, Windows Server.

У тестах з ненадійностями мережі рекомендується використовувати різні значення керованих метрик, таких як затримка та втрата пакетів. Так, наприклад, можна протестувати, як рішення VPN впораються з меншою або більшою затримкою, або з вищим відсотком втрати пакетів, якщо налаштувати експериментальну установку таким чином, щоб не розривати з’єднання при підвищених втратах.

Щоб краще проаналізувати продуктивність рішень VPN, було б корисно ввести більше залежних та незалежних змінних. Це дозволить отримати докладніші і точніші результати.

Крім того, в тестах можна використовувати не тільки інші метрики, але й інші інструменти вимірювання, а потім порівняти отримані результаті з результатами тестів iPerf.

У майбутніх дослідженнях важливо протестувати рішення VPN з різними конфігураціями. У цьому дослідженні використовувались конфігурації за замовчуванням, але якщо застосувати інші налаштування для шифрування та решти параметрів, результати будуть відрізнятись. Було б корисно протестувати рішення VPN з уніфікованим набором криптографічних алгоритмів, щоб вирівняти ігрове поле.

Ще один з варіантів майбутнього дослідження полягає у тому, щоб замість симуляції провести тестування в реальній компанії, на реальному апаратному забезпеченні, з реальним мережевим трафіком між сервером і

клієнтами. За реальних умов і налаштувань результати тестів будуть більш переконливими.

3.4. Висновки за розділом

У цьому розділі було докладно проаналізовано результати експерименту. Було описано тенденції, що спостерігались під час порівняння результатів різних рішень VPN, в різних середовищах та умовах тестування. Було визначено найбільш та найменш продуктивні рішення у кожному випадку тестування.

Було обґрунтовано рекомендації з вибору протоколу VPN для забезпечення віддаленого доступу за різних умов надійності мережевого з’єднання. Для того, щоб зробити вибір зручнішим, на основі рекомендацій було побудовано дерево рішень для вибору найліпшого протоколу VPN з погляду продуктивності.

Також було розглянуто приклади практичного застосування результатів, отриманих у цьому дослідженні.

Насамкінець, було окреслено напрямки майбутніх досліджень, що можуть бути корисними для отримання докладніших і точніших результатів.

ВИСНОВКИ

У даній дипломній роботі було досліджено проблему вибору протоколу VPN для забезпечення віддаленого доступу до корпоративної мережі. В той час як одні підприємства прагнуть застосовувати найновіші рішення з найкращими гарантіями безпеки, як-от нульова довіра, інші віддають перевагу традиційним рішенням. Найпоширенішим для віддаленого доступу сьогодні залишається застосування VPN.

Під час аналізу раніше проведених досліджень було з’ясовано, що продуктивність VPN залежить від обраного протоколу, конфігурації та середовища застосування. Було вирішено протестувати та порівняти продуктивність різних рішень VPN у різних випадках застосування. Щоб на основі результатів тестування обґрунтувати рекомендації з вибору найліпшого рішення.

Для порівняльного аналізу було обрано два провідні на сьогодні протоколи VPN, L2TP/IPsec та OpenVPN, а також відносно новий протокол, що нарощує популярність – WireGuard. Оскільки віддалений доступ сьогодні часто стикається з ненадійністю мереж, було вирішено проаналізувати продуктивність VPN як в стабільних умовах, так і в умовах ненадійності мережі. У розглянутих дослідженнях продуктивності VPN тестування в подібних умовах не проводилось.

Було надано розгорнуту характеристику обраних протоколів VPN. Протоколи було охарактеризовано за трьома основними аспектами: доступність, безпека та продуктивність. Узагальнену характеристику було наведено у порівняльних таблицях.

Щоб порівняти продуктивність різних рішень VPN у різних середовищах та умовах застосування, було вирішено провести експериментальне дослідження. Для оцінки продуктивності VPN вимірювалась пропускна здатність. Ненадійність мережі досягалась шляхом введення затримки та втрати пакетів.

Для проведення експерименту було налаштовано експериментальну установку, що складалась з трьох клієнтів та маршрутизатора, а також віддаленого сервера. Три рішення VPN тестувались у трьох різних операційних системах – Windows, Ubuntu та macOS – в стабільних умовах та з двома варіантами ненадійності. Для вимірювання пропускної здатності на кожному з клієнтів був встановлений спеціальний інструмент iPerf, варіанти ненадійності вводились за допомогою програмного маршрутизатора pfSense, що вбудовано підтримує можливість формування трафіку.

З метою забезпечити високу якість результатів було розглянуто та оброблено загрози валідності експерименту. Результати експерименту були зафіксовані, усереднені та представлені у порівняльній таблиці. Після чого їх було докладно проаналізовано.

На основі проведеного порівняльного аналізу було обґрунтовано рекомендації з вибору протоколу VPN для забезпечення віддаленого доступу за різних умов надійності мережевого з’єднання. У межах цих рекомендацій для кожного розглянутого випадку застосування було визначено найліпший протокол VPN з погляду продуктивності. Також було окреслено рекомендовані напрямки майбутніх досліджень.

Отримані результати можуть бути корисними мережевим адміністраторам для оцінки варіантів рішень при розгортанні VPN і забезпеченні віддаленого доступу в умовах ненадійності мережі.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Remote Work Statistics: 9 Stats That Show Telecommuting is the Future [Електронний ресурс] / Режим доступу: www. URL: https://miro.com/guides/remote-work/statistics – 03.10.2022

2. Remote Access Solutions: Overcoming the Challenges [Електронний ресурс] / Режим доступу: www. URL: https://readwrite.com/remote-access-solutions-overcoming-the-challenges/ – 03.10.2022

3. Remote Work: Vulnerabilities and Threats to the Enterprise [Електронний ресурс] / Режим доступу: www. URL: https://insights.sei.cmu.edu/blog/remote-work-vulnerabilities-and-threats-to-the-enterprise/ – 05.10.2022

4. Zero Trust Architecture [Електронний ресурс] / Режим доступу: www. URL: https://www.nist.gov/publications/zero-trust-architecture – 07.10.2022

5. ZTNA: A Better Way to Control Access, Boost Security [Електронний ресурс] / Режим доступу: www. URL: https://www.hillstonenet.com/blog/ztna-a-better-way-to-control-access-boost-security/ – 07.10.2022

6. ZTNA vs VPN [Електронний ресурс] / Режим доступу: www. URL: https://www.checkpoint.com/cyber-hub/network-security/what-is-zero-trust-network-access-ztna/ztna-vs-vpn/ – 07.10.2022

7. Scott C. Wolfe P. Erwin M. Virtual Private Networks, Second Edition [Текст] / O'Reilly, 1999. 225 p. – 10.10.2022

8. Brown S. Implementing Virtual Private Networks [Текст] / McGraw-Hill, 1999. 594 p. – 12.10.2022

9. Remote access VPN: what are they, how do they work and which are the best [Електронний ресурс] / Режим доступу: www. URL: https://www.techradar.com/vpn/remote-access-vpn – 14.10.2022

10. What Is Encryption? Explanation and Types [Електронний ресурс] / Режим доступу: www. URL: https://www.cisco.com/c/en/us/products/security/encryption-explained.html – 14.10.2022

11. Network Performance Analysis of VPN Protocols: An Empirical Comparison on Different Operating Systems [Електронний ресурс] / Режим доступу: www. URL: https://ieeexplore.ieee.org/document/4908347 – 15.10.2022

12. Virtual Private Network's Impact on Network Performance [Електронний ресурс] / Режим доступу: www. URL: https://ieeexplore.ieee.org/document/8601281 – 15.10.2022

13. An Empirical Analysis of the Commercial VPN Ecosystem [Електронний ресурс] / Режим доступу: www. URL: https://dl.acm.org/doi/pdf/10.1145/3278532.3278570 – 17.10.2022

14. WireGuard: Next Generation Kernel Network Tunnel [Електронний ресурс] / Режим доступу: www. URL: https://www.wireguard.com/papers/wireguard.pdf – 17.10.2022

15. Microsoft Security Advisory 2743314 [Електронний ресурс] / Режим доступу: www. URL: https://learn.microsoft.com/en-us/security-updates/SecurityAdvisories/2012/2743314?redirectedfrom=MSDN – 17.10.2022

16. Microsoft handed the NSA access to encrypted messages [Електронний ресурс] / Режим доступу: www. URL: https://www.theguardian.com/world/2013/jul/11/ microsoft-nsa-collaboration-user-data – 17.10.2022

17. Comparison of VPN Protocols at Network Layer Focusing on WireGuard Protocol [Електронний ресурс] / Режим доступу: www. URL: https://www.researchgate.net/publication/345681297_Paper-Comparison_of_VPN_Protocols_at_Network_Layer_Focusing_on_Wire_Guard_Protocol_Comparison_of_VPN_Protocols_at_Network_Layer_Focusing_on_Wire_Guard_Protocol – 20.10.2022

18. Performance Comparison of VPN Solutions [Електронний ресурс] / Режим доступу: www. URL: https://core.ac.uk/download/pdf/322886318.pdf – 20.10.2022

19. WireGuard – Next Generation Secure Network Tunnel [Електронний ресурс] / Режим доступу: www. URL: https://www.sstic.org/2018/presentation/WireGuard/ – 31.10.2022

20. The New Cloudflare VPN: What It Is And Is Not [Електронний ресурс] / Режим доступу: www. URL: https://openvpn.net/blog/what-is-cloudflare-vpn/ – 31.10.2022

21. WireGuard VPN review: A new type of VPN offers serious advantages [Електронний ресурс] / Режим доступу: www. URL: https://arstechnica.com/gadgets/2018/08/wireguard-vpn-review-fast-connections-amaze-but-windows-support-needs-to-happen/ – 31.10.2022

22. VPN Protocol Comparison: PPTP vs OpenVPN vs L2TP vs SSTP [Електронний ресурс] / Режим доступу: www. URL: https://www.vpnuniversity.com/learn/vpn-protocols-compared-pptp-vs-openvpn-vs-l2tp-vs-sstp – 31.10.2022

23. OpenVPN Connect Client [Електронний ресурс] / Режим доступу: www. URL: https://openvpn.net/vpn-client/ – 31.10.2022

24. What is WireGuard? Secure, simple VPN now part of Linux [Електронний ресурс] / Режим доступу: www. URL: https://www.csoonline.com/article/3434788/what-is-wireguard-secure-simple-vpn-still-in-development.html – 31.10.2022

25. Fix: L2TP VPN issues (blocked / not responding) [Електронний ресурс] / Режим доступу: www. URL: https://windowsreport.com/l2tp-vpn-blocked/ – 31.10.2022

26. What is OpenVPN? Is OpenVPN safe? [Електронний ресурс] / Режим доступу: www. URL: https://www.comparitech.com/blog/vpn-privacy/what-is-openvpn/ – 31.10.2022

27. How To Set Up WireGuard on Ubuntu 20.04 [Електронний ресурс] / Режим доступу: www. URL: https://www.digitalocean.com/community/tutorials/how-to-set-up-wireguard-on-ubuntu-20-04 – 31.10.2022

28. RFC 3193: Securing L2TP using IPsec [Електронний ресурс] / Режим доступу: www. URL: https://www.rfc-editor.org/rfc/rfc3193 – 02.11.2022

29. How IPsec works, it’s components and purpose [Електронний ресурс] / Режим доступу: www. URL: https://www.csoonline.com/article/2117067/how-ipsec-works.html – 02.11.2022

30. AH and ESP protocols [Електронний ресурс] / Режим доступу: www. URL: https://www.ibm.com/docs/en/zos/2.4.0?topic=ipsec-ah-esp-protocols – 02.11.2022

31. What is IPsec and how does it work? [Електронний ресурс] / Режим доступу: www. URL: https://www.comparitech.com/blog/information-security/ipsec-encryption/ – 02.11.2022

32. National Security Agency (NSA) Suite B Cryptography [Електронний ресурс] / Режим доступу: www. URL: https://www.ibm.com/docs/en/ibm-mq/7.5?topic=ssl-national-security-agency-suite-b-cryptography – 02.11.2022

33. Revealed: how US and UK spy agencies defeat internet privacy and security [Електронний ресурс] / Режим доступу: www. URL: https://www.theguardian.com/world/2013/sep/05/nsa-gchq-encryption-codes-security – 02.11.2022

34. Guide to IPsec VPNs [Електронний ресурс] / Режим доступу: www. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-77r1.pdf – 02.11.2022

35. Authentication [Електронний ресурс] / Режим доступу: www. URL: https://community.openvpn.net/openvpn/wiki/Concepts-Authentication – 03.11.2022

36. WireGuard vs OpenVPN in 2022: 7 Big Differences [Електронний ресурс] / Режим доступу: www. URL: https://restoreprivacy.com/vpn/wireguard-vs-openvpn/ – 03.11.2022

37. Reference manual for OpenVPN 2.4 [Електронний ресурс] / Режим доступу: www. URL: https://openvpn.net/community-resources/reference-manual-for-openvpn-2-4/ – 03.11.2022

38. The OpenVPN 2.4.0 Audit by OSTIF and QuarksLab Results [Електронний ресурс] / Режим доступу: www. URL: https://ostif.org/the-openvpn-2-4-0-audit-by-ostif-and-quarkslab-results/ – 03.11.2022

39. WireGuard VPN for Remote Working [Електронний ресурс] / Режим доступу: www. URL: https://www.scalefactory.com/blog/2020/12/16/wireguard-vpn-for-remote-working/ – 04.11.2022

40. WireGuard VPN: Secure and Fast, But Bad for Privacy? [Електронний ресурс] / Режим доступу: www. URL: https://restoreprivacy.com/vpn/wireguard/ – 04.11.2022

41. Protocol & Cryptography – WireGuard [Електронний ресурс] / Режим доступу: www. URL: https://www.wireguard.com/protocol/ – 04.11.2022

42. Using WireGuard for Privacy Protection [Електронний ресурс] / Режим доступу: www. URL: https://www.ivpn.net/knowledgebase/general/using-wireguard-for-privacy-protection/ – 04.11.2022

43. Formal Verification – WireGuard [Електронний ресурс] / Режим доступу: www. URL: https://www.wireguard.com/formal-verification/ – 04.11.2022

44. TCP vs. UDP: Understanding 10 Key Differences [Електронний ресурс] / Режим доступу: www. URL: https://www.spiceworks.com/tech/networking/articles/tcp-vs-udp/ – 05.11.2022

45. VPN Protocols: OpenVPN vs IPSec, WireGuard, L2TP, & IKEv2 [Електронний ресурс] / Режим доступу: www. URL: https://restoreprivacy.com/vpn/openvpn-ipsec-wireguard-l2tp-ikev2-protocols/ – 05.11.2022

46. Performance – WireGuard [Електронний ресурс] / Режим доступу: www. URL: https://www.wireguard.com/performance/ – 05.11.2022

47. Kernel Space Definition [Електронний ресурс] / Режим доступу: www. URL: http://www.linfo.org/kernel_space.html – 05.11.2022

48. WireGuard: fast, modern, secure VPN tunnel [Електронний ресурс] / Режим доступу: www. URL: https://www.wireguard.com/ – 05.11.2022

49. Comparison of VPN protocols [Електронний ресурс] / Режим доступу: www. URL: https://www.ivpn.net/pptp-vs-ipsec-ikev2-vs-openvpn-vs-wireguard/ – 05.11.2022

50. Wohlin C. Runeson P. Host M. Ohlsson M. Regnell B. Wesslen A. Experimentation in Software Engineering [Текст] / Springer, 2012. 236 p. – 09.11.2022

51. Understanding Delay in Packet Voice Networks [Електронний ресурс] / Режим доступу: www. URL: https://www.cisco.com/c/en/us/support/docs/voice/voice-quality/5125-delay-details.html – 09.11.2022

52. Influence of packet loss on a speaker verification system over IP network [Електронний ресурс] / Режим доступу: www. URL: https://ieeexplore.ieee.org/document/7477365 – 09.11.2022

53. Wireless Broadband Measurement in California [Електронний ресурс] / Режим доступу: www. URL: https://ieeexplore.ieee.org/document/6614357 – 10.11.2022

54. iPerf – iPerf3 and iPerf2 user documentation [Електронний ресурс] / Режим доступу: www. URL: https://iperf.fr/iperf-doc.php – 10.11.2022

55. Introduction – pfSense Documentation [Електронний ресурс] / Режим доступу: www. URL: https://docs.netgate.com/pfsense/en/latest/general/index.html – 14.11.2022

56. Libreswan VPN software [Електронний ресурс] / Режим доступу: www. URL: https://libreswan.org/ – 14.11.2022

57. Traffic Shaper – Limiters – pfSense Documentation [Електронний ресурс] / Режим доступу: www. URL: https://docs.netgate.com/pfsense/en/latest/trafficshaper/limiters.html – 14.11.2022

58. Improving VPN performance over multiple access links [Електронний ресурс] / Режим доступу: www. URL:

https://ieeexplore.ieee.org/document/4769158 – 16.11.2022

59. A small network for modeling MPLS [Електронний ресурс] / Режим доступу: www. URL: https://ieeexplore.ieee.org/document/7506760 – 16.11.2022

60. Berndtsson M. Hansson J. Olsson B. Lundell B. Thesis Projects, A Guide for Students in Computer Science and Information Systems, Second Edition [Текст] / Springer, 2008. 158 p. – 19.11.2022

ДОДАТКИ

Додаток А

Сценарії тестів iPerf для ОС Ubuntu

У цьому додатку наведено сценарії, що використовувались для запуску тестів iPerf клієнтом під керуванням операційної системи Ubuntu. Вони були заплановані за допомогою Crontab, лістинги налаштування якого також наведено тут.

```
// Сценарії для запуску тестів
// Запуск тестів без VPN
#!/bin/bash
iperf3 -c 10.1.1.101 -t 10 >> /home/ubuntu-client/iperftests/
lognoVPNUb.txt
// Запуск тестів з WireGuard
#!/bin/bash
iperf3 -c 10.10.1.1 -t 10 >> /home/ubuntu-client/iperftests/
logWireguardUb.txt
// Запуск тестів з L2TP/IPSec
#!/bin/bash
iperf3 -c 10.9.1.1 -t 10 >> /home/ubuntu-client/iperftests/
logL2tpIpsecUb.txt
// Запуск тестів з OpenVPN
#!/bin/bash
iperf3 -c 10.8.1.1 -t 10 >> /home/ubuntu-client/iperftests/
logOpenvpnUb.txt

// Налаштування Crontab
1-50 0,4,8 * * * /home/ubuntu-client/iperftests/testnoVPN.sh
55 0,4,8 * * * /home/ubuntu-client/iperftests/runWireguard.sh
1-50 1,5,9 * * * /home/ubuntu-client/iperftests/testWireGuard.sh
52 1,5,9 * * * /home/ubuntu-client/iperftests/killWireguard.sh
55 1,5,9 * * * /home/ubuntu-client/iperftests/runL2tpIpsec.sh
1-50 2,6,10 * * * /home/ubuntu-client/iperftests/
testL2tpIpsec.sh
```

```
52 2,6,10 * * * /home/ubuntu-client/iperftests/killL2tpIpsec.sh
55 2,6,10 * * * /home/ubuntu-client/iperftests/runOpenvpn.sh
1-50 3,7,11 * * * /home/ubuntu-client/iperftests/testOpenvpn.sh
52 3,7,11 * * * /home/ubuntu-client/iperftests/killOpenvpn.sh

// Сценарії для підключення та відключення VPN
// Підключення WireGuard
#!/bin/bash
sudo wg-quick up wg0
// Відключення WireGuard
#!/bin/bash
sudo wg-quick down wg0
// Підключення L2TP/IPSec
#!/bin/bash
sudo nmcli connection up IPSec
// Відключення L2TP/IPSec
#!/bin/bash
sudo nmcli connection down IPSec
// Підключення OpenVPN
#!/bin/bash
sudo nmcli connection up HPEOVPN
// Відключення OpenVPN
#!/bin/bash
sudo nmcli connection down HPEOVPN
```

Додаток Б

Сценарії тестів iPerf для ОС Windows

У цьому додатку наведено сценарії, що використовувались для запуску тестів iPerf клієнтом під керуванням операційної системи Windows. Вони були заплановані за допомогою планувальника завдань Windows, так само як це було зроблено із Crontab в Ubuntu, лістинги налаштування також наведено тут.

```
// Сценарії для запуску тестів
// Запуск тестів без VPN
for /l %%x in (1, 1, 50) do (
C:\Users\windows-client\Desktop\Programs\iperf-3.1.3-win64\
iperf3.exe -c
10.1.1.101 -t 10 --logfile
C:\Users\windows-client\Desktop\Programs\lognoVPNWin.txt
timeout /t 50 /nobreak)
// Запуск тестів з WireGuard
for /l %%x in (1, 1, 50) do (
C:\Users\windows-client\Desktop\Programs\iperf-3.1.3-win64\
iperf3.exe -c
10.10.1.1 -t 10 --logfile
C:\Users\windows-client\Desktop\Programs\logWireguardWin.txt
timeout /t 50 /nobreak)
// Запуск тестів з L2TP/IPSec
for /l %%x in (1, 1, 50) do (
C:\Users\windows-client\Desktop\Programs\iperf-3.1.3-win64\
iperf3.exe -c
10.9.1.1 -t 10 --logfile
C:\Users\windows-client\Desktop\Programs\logL2tpIpsecWin.txt
timeout /t 50 /nobreak)
// Запуск тестів з OpenVPN
for /l %%x in (1, 1, 50) do (
```

```
C:\Users\windows-client\Desktop\Programs\iperf-3.1.3-win64\
iperf3.exe -c
10.8.1.1 -t 10 --logfile
C:\Users\windows-client\Desktop\Programs\logOpenvpnWin.txt
timeout /t 50 /nobreak)

// Підключення та відключення VPN
// Підключення WireGuard
param ([switch]$Elevated)
function Check-Admin {
$currentUser = New-Object Security.Principal.WindowsPrincipal $
([Security.Principal.WindowsIdentity]::GetCurrent ())
$currentUser.IsInRole
([Security.Principal.WindowsBuiltinRole]::Administrator)
}
if ( (Check-Admin) -eq $false) {
if ($elevated)
{
# could not elevate, quit
}
else {
Start-Process  powershell.exe  -Verb  RunAs  -ArgumentList  ('-
noprofile
-noexit -file "{0}" -elevated' -f
($myinvocation.MyCommand.Definition))
}
exit
}
C:\"Program Files"\WireGuard\wireguard.exe
$wshell = New-Object -ComObject wscript.shell;
$wshell.AppActivate ('WireGuard')
Sleep 2
$wshell.SendKeys ('~')
exit
// Відключення WireGuard
```

```
TASKKILL /F /IM wireguard.exe
// Підключення L2TP/IPSec
rasdial L2TP_IPSec vpnclient MvkstJpoieyG7mhc
// Відключення L2TP/IPSec
rasdial L2TP_IPSec /disconnect
// Підключення OpenVPN
start openvpn-gui.exe --connect config.ovpn
// Відключення OpenVPN
TASKKILL /F /IM openVPN-gui.exe
TASKKILL /F /IM ovpnagent.exe
TASKKILL /F /IM openvpn.exe
```

Додаток В

Сценарії тестів iPerf для ОС macOS

У цьому додатку наведено сценарії, що використовувались для запуску тестів iPerf клієнтом під керуванням операційної системи macOS. Планування виконувалось аналогічно до того, як це було зроблено в інших операційних системах, за допомогою вбудованого інструменту планування – Automator Workflows.

```
// Сценарії для запуску тестів
// Запуск тестів без VPN
#!/bin/bash
sudo /Users/macos-client/Desktop/iperf3 -c 10.1.1.101 -t 10
>> /Users/macos-client/Desktop/iperftests/lognoVPNMac.txt
// Запуск тестів з WireGuard
#!/bin/bash
sudo /Users/macos-client/Desktop/iperf3 -c 10.10.1.1 -t 10 >>
/Users/macos-client/Desktop/iperftests/logWireguardMac.txt
// Запуск тестів з L2TP/IPSec
#!/bin/bash
sudo /Users/macos-client/Desktop/iperf3 -c 10.9.1.1 -t 10 >>
/Users/macos-client/Desktop/iperftests/logL2tpIpsecMac.txt
// Запуск тестів з OpenVPN
#!/bin/bash
sudo /Users/macos-client/Desktop/iperf3 -c 10.8.1.1 -t 10 >>
/Users/macos-client/Desktop/iperftests/logOpenvpnMac.txt
```