Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Програмно-апаратний засіб для захищеного доступу до мережі на основі персонального VPN-сервера
Назва теми

КвРКІ 220014.22.01.08 ПЗ
Шифр

Галузь знань *12 «Інформаційні технології»*
Шифр, назва

Спеціальність *123 «Комп'ютерна інженерія»*
Шифр, назва

Освітня програма *«Комп'ютерна інженерія та програмування»*
Назва

Виконав: студент III курсу, група КІ2с-22-1 _____ Олексій ЗАКРЕВСЬКИЙ
Підпис Ініціали, прізвище

Керівник _____ Василь ЯЦКІВ
Підпис, дата Ініціали, прізвище

Нормоконтролер _____ Тетяна КИСІЛЬ
Підпис, дата Ініціали, прізвище

До захисту допускаю:
зав. кафедри комп'ютерної інженерії та інформаційних систем _____ Ольга ПАВЛОВА
Підпис Ініціали, прізвище

«17» червня 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

" 10 " 01 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Олексію ЗАКРЕВСЬКОМУ
Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) програмно – апаратний засіб для захищеного доступу до мережі на основі персонального vpn-сервера

Керівник проекту (роботи) Василь ЯЦКІВ, д.ф., старший викладач
Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.03.2025 р. № 5

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

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

Програмно – апаратний засіб для захищеного доступу до мережі на основі персонального vpn-сервера та постановка задачі щодо її удосконалення

Аналіз існуючих рішень для побудови VPN-серверів: порівняння хмарних, комерційних і локальних підходів, а також оцінка доцільності створення персонального VPN на базі Raspberry Pi з використанням інструментів віртуалізації

Розгортання та конфігурація середовища для реалізації VPN-сервера

Тестування працездатності створеного VPN-рішення

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Архітектура ПЗ проєкту

Апартна частина

Встановлення та налаштваня програмного забезпечення

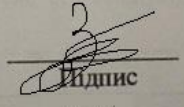
6. Консультанти розділів дипломного проекту (роботи)

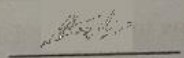
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Тетяна КИСІЛЬ, доцент кафедри КІС		
Антиплагіат	Андрій НІЧЕПОРУК, доцент кафедри КІС		

7. Дата видачі завдання « 10 » 01 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2025	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2025	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2025	виконано
4	Робота над розділом 2 – підготовка середовища для розгортання персонального vpn-сервера	01.04.2025	виконано
5	Робота над розділом 3 – встановлення та налаштування vpn-сервера на базі pivpn	29.04.2025	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2025	виконано
7	Попередній захист ВКР	26.05.2025	виконано
8	Захист ВКР на засіданні ЕК	Червень 2025 року	

Студент  Закревський ОЛЕКСІЙ
Підпис Ініціали, прізвище

Керівник роботи  Всиль ЯЦКІВ
Підпис Ініціали, прізвище

№ рядка	формат	Позначення	Найменування	Кіл. листів	№ екз	Примітка
			Текстові документи			
1		КвРКІ 022002.22.02.02 ПЗ	Пояснювальна записка	66		
			Графічні матеріали			
2		КвРКІ 022002.22.02.02 Е8	Архітектура ПЗ проєкту	1		
3		КвРКІ 022002.22.02.02 Е8	Архітектура ПЗ для кіберфізичної системи	1		
4		КвРКІ 022002.22.02.02 Е8	Встановлення та налаштваня програмного забезпечення	1		

Зм	Арк	№ докум	Підпис	Дата	КвРКІ 220014.22.01.08 ВП		
Розробив		Закревський		17.06.25	Відомість проекту	Літера	Аркуш	Аркушів
Перевір.		Яцків				У	1	1
Н. контр.		Кисіль		17.06.25		ХНУ, КІ2с-22-1		
Затв.		Павлова		17.06.25				

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмно-апаратний засіб для захищеного доступу до мережі на основі персонального VPN-сервера.».

Автор роботи: Олексій ЗАКРЕВСЬКИЙ

Керівник роботи: Василь ЯЦКІВ.

Пояснювальна записка: 66 с., 17 рис., 3 дод., 40 джерел.

Графічна частина: 3 креслення.

ПРОГРАМНО – АПАРАТНИЙ ЗАСІБ ДЛЯ ЗАХИЩЕНОГО ДОСТУПУ ДО МЕРЕЖІ НА ОСНОВІ ПЕРСОНАЛЬНОГО VPN-СЕРВЕРА.

Метою дипломної роботи є визначення умов та особливостей застосування VPN на raspberry pi, а також оцінка механізмів обробки та захисту інформації у мережі.

Об'єктом дослідження є функціонування PIVPN на raspberry pi.

Предметом дослідження є оцінка режимів застосування PIVPN на raspberry pi.

Під час проведення даного дослідження був використаний метод систематичного огляду літератури для вивчення і аналізу предметної області даного дослідження з текстових джерел інформації.

_____	30.05.2025
Підпис студента	Дата

ВСТУП

У сучасному світі інформаційних технологій питання безпеки цифрових комунікацій є одним із ключових аспектів як для приватних осіб, так і для бізнесу. Щодня мільйони користувачів передають особисті дані через Інтернет, здійснюють фінансові операції, працюють з конфіденційною інформацією, використовуючи публічні або недостатньо захищені мережі. Відповідно до зростання кіберзагроз, фішингових атак, перехоплення трафіку та зловмисного моніторингу, постає необхідність створення інструментів, що забезпечують високий рівень захисту інформації при передаванні через мережу. Одним із найефективніших рішень у цьому напрямі є використання віртуальних приватних мереж (VPN), які дозволяють створити захищений тунель між користувачем та віддаленим сервером, шифруючи весь трафік і приховуючи реальне місцезнаходження користувача.

Особливої актуальності набуває ідея створення персонального VPN-сервера, який дозволяє мати повний контроль над переданими даними, уникнути довіри до сторонніх VPN-провайдерів, а також налаштувати з'єднання відповідно до власних вимог. У цьому контексті особливу цінність становить використання одноплатних комп'ютерів, таких як Raspberry Pi, що поєднують у собі компактність, енергоефективність і достатню продуктивність для виконання серверних функцій. Raspberry Pi, працюючи під управлінням спеціалізованої операційної системи, дає змогу розгорнути повноцінний VPN-сервер у домашніх умовах з мінімальними витратами.

Метою цієї роботи є розробка програмно-апаратного засобу для захищеного доступу до мережі на основі персонального VPN-сервера з використанням програмного забезпечення з відкритим кодом PiVPN. У рамках дослідження буде розглянуто теоретичні основи VPN, проаналізовано протоколи шифрування, розроблено рішення для емуляції Raspberry Pi у віртуальному середовищі VirtualBox, налаштовано VPN-сервер з протоколом WireGuard, а також проведено

тестування з’єднання та оцінку рівня безпеки. Особливий акцент зроблено на практичній реалізації, що дозволяє розгорнути захищене з’єднання навіть у навчальних або обмежених апаратних умовах, таких як відсутність фізичного Raspberry Pi.

Результатом дипломної роботи є побудова повністю функціонального VPN-рішення, яке може використовуватись для безпечного віддаленого доступу до домашньої або офісної мережі, обходу цензури чи обмежень, створення середовища для безпечного інтернет-серфінгу. Важливою перевагою є відкритість використовуваних технологій, що забезпечує гнучкість, прозорість і можливість подальшого розвитку проєкту відповідно до зростаючих вимог інформаційної безпеки.

1 ТЕОРЕТИЧНІ ОСНОВИ ЗАХИЩЕНОГО ДОСТУПУ ДО МЕРЕЖІ ЧЕРЕЗ VPN

1.1 Поняття віртуальної приватної мережі (VPN)ъ

Віртуальна приватна мережа (VPN, Virtual Private Network) – це технологія, яка забезпечує створення захищеного з'єднання поверх публічної мережі, зазвичай Інтернету. Вона дозволяє користувачеві безпечно обмінюватися даними з віддаленим сервером або внутрішньою мережею, ніби той фізично під'єднаний до цієї мережі. Основна функція VPN полягає в тому, щоб зашифрувати весь інтернет-трафік користувача, приховати його реальну IP-адресу та забезпечити конфіденційність переданих даних.

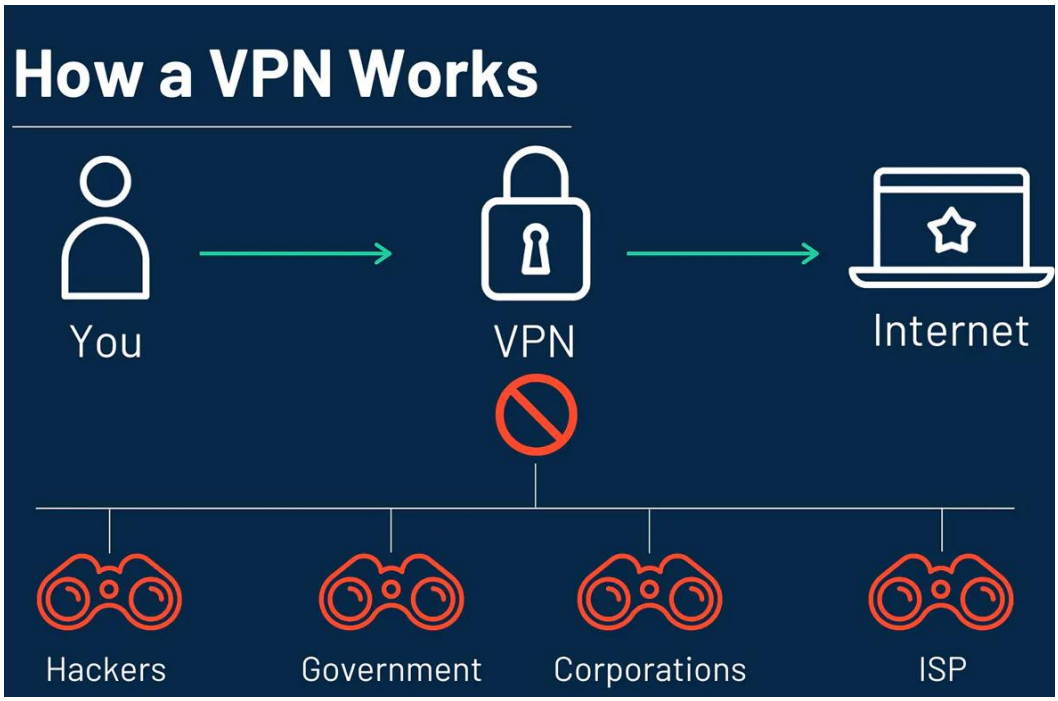


Рисунок 1.1 – Приклад роботи VPN

Принцип дії VPN базується на створенні зашифрованого тунелю, через який передається інформація. Уся комунікація між клієнтом і сервером проходить через цей тунель, де дані інкапсулюються – тобто, пакети одного протоколу

вкладаються в інші, що дозволяє передавати їх через загальнодоступні мережі без втрати безпеки. Це дає змогу не лише приховати реальне розташування користувача, але й забезпечити захист трафіку від стеження з боку зловмисників, провайдерів, або державних органів. VPN особливо корисний при використанні відкритих мереж Wi-Fi, де ризик перехоплення даних особливо високий.

У корпоративному середовищі VPN є критично важливим засобом для організації віддаленої роботи. Працівники можуть безпечно підключатися до внутрішньої мережі компанії з будь-якої точки світу, працюючи з корпоративними ресурсами, якби вони фізично перебували в офісі. Це дає змогу зберігати безперервність бізнес-процесів і одночасно захищати чутливу інформацію компанії.

Серед звичайних користувачів VPN використовується для забезпечення конфіденційності, обходу блокувань або географічних обмежень доступу до контенту, а також для уникнення інтернет-цензури. Багато людей застосовують VPN, щоб отримати доступ до ресурсів, які доступні лише в інших країнах, наприклад, стримінгових сервісів або новинних порталів.

VPN може бути реалізований у різних формах. Найпоширенішими є віддалений доступ, коли користувач під’єднується до сервера через захищене з’єднання, та міжмережеві VPN-з’єднання (сайт-до-сайту), які поєднують цілі офісні мережі. Також існують персональні VPN-рішення, що встановлюються на домашніх пристроях або серверах, забезпечуючи максимальний контроль користувача над конфігурацією та безпекою.

З технічного боку VPN працює на основі спеціальних протоколів, які визначають спосіб створення та підтримки захищеного з’єднання. До них належать, зокрема, PPTP, L2TP/IPSec, SSTP, OpenVPN, WireGuard та інші. Вони відрізняються між собою рівнем шифрування, швидкістю, стабільністю та вимогами до налаштування. З розвитком технологій все більшого поширення набувають сучасні протоколи, як-от WireGuard, які поєднують високу продуктивність із простотою використання.

Однак, незважаючи на численні переваги, VPN не є універсальним захистом від усіх загроз. Ефективність VPN значною мірою залежить від правильної конфігурації, вибору надійного провайдера (у випадку комерційного використання), рівня шифрування та політики зберігання журналів. У деяких випадках можливе зниження швидкості з’єднання через обробку трафіку на сервері. Попри це, у більшості ситуацій VPN залишається ефективним інструментом для захисту особистої інформації та забезпечення цифрової приватності в умовах сучасного інформаційного суспільства.

1.2 Основні VPN-протоколи: OpenVPN, WireGuard, інші

Один із ключових аспектів побудови VPN-з'єднання – це вибір протоколу, який визначає метод шифрування, обмін ключами, автентифікацію користувача та інші аспекти безпеки й продуктивності. Серед найбільш поширених VPN-протоколів сьогодні – OpenVPN та WireGuard. Існують також інші протоколи, кожен із яких має свої особливості, переваги та обмеження.

OpenVPN – це один із найпопулярніших і найнадійніших протоколів для створення віртуальних приватних мереж (VPN), який був розроблений у 2001 році як рішення з відкритим кодом. Його відкритість дала змогу не лише незалежно перевірити рівень безпеки, а й активно розвивати та вдосконалювати технологію зусиллями глобальної спільноти розробників. OpenVPN працює на основі бібліотеки OpenSSL, що забезпечує йому потужні можливості шифрування і дозволяє застосовувати перевірені криптографічні алгоритми, зокрема AES-256, RSA, SHA та інші.

Завдяки своїй гнучкості, OpenVPN підтримує як TCP, так і UDP протоколи транспортного рівня, що дозволяє користувачеві обирати між більш стабільним з’єднанням або вищою швидкістю передачі даних. Цей протокол чудово підходить для використання у різних середовищах — від домашніх VPN-серверів до корпоративних інфраструктур. OpenVPN може працювати через будь-який

порт, що дозволяє обійти обмеження фаєрволів і систем цензури. Завдяки можливості маскувати трафік під звичайний HTTPS, OpenVPN є особливо ефективним у країнах із жорсткою інтернет-фільтрацією.

Безпека – один із найважливіших аспектів OpenVPN. Протокол підтримує автентифікацію за допомогою сертифікатів, паролів, або ключів, що дозволяє будувати багаторівневу систему контролю доступу. Він також дозволяє використовувати TLS (Transport Layer Security) для додаткового захисту та узгодження параметрів шифрування, що унеможливлює перехоплення або підміну даних зловмисниками. Оскільки OpenVPN не має вбудованих бекдорів і не залежить від закритих стандартів, він часто обирається для чутливих задач у сфері кібербезпеки, захисту журналістів, активістів і приватних користувачів.

Ще однією перевагою OpenVPN є його кросплатформеність. Він підтримується на більшості операційних систем, включно з Windows, macOS, Linux, iOS, Android, а також на мережевому обладнанні, маршрутизаторах і одноплатних комп'ютерах, таких як Raspberry Pi. Для роботи з OpenVPN існує велика кількість клієнтів, як офіційних, так і сторонніх, що дозволяє легко налаштувати з'єднання на будь-якому пристрої.

OpenVPN часто застосовується для створення так званих «тунельних» підключень, коли весь інтернет-трафік спрямовується через VPN-сервер, забезпечуючи не тільки захист, але й можливість анонімного перегляду веб-сторінок. Крім того, його використовують для організації віддаленого доступу до корпоративних ресурсів, створення мереж між офісами, безпечного підключення IoT-пристроїв та побудови складних мережевих топологій.

Щодо налаштування, OpenVPN є більш складним у порівнянні з деякими сучасними протоколами, наприклад, WireGuard. Для його повноцінної реалізації потрібно створювати інфраструктуру відкритих ключів (PKI), генерувати сертифікати, налаштовувати конфігураційні файли як на сервері, так і на клієнтах. Проте саме така складність компенсується високим рівнем контролю і безпеки, які він забезпечує.

Завдяки своїм технічним перевагам, стабільності та широкій підтримці спільноти, OpenVPN протягом багатьох років залишається стандартом де-факто у сфері захищених мережевих з'єднань. Він використовується як в комерційних VPN-сервісах, так і в індивідуальних рішеннях, коли важливі гнучкість налаштування, шифрування, сумісність з фаєрволами та доведена на практиці ефективність.

WireGuard – це сучасний VPN-протокол, який став одним з найперспективніших у сфері захищеного передавання даних завдяки своїй простоті, ефективності та високому рівню безпеки. Він був розроблений Джейсоном Доненфельдом і вперше представлений як експериментальна технологія для ядра Linux, але швидко здобув підтримку й інтеграцію в численні платформи, включно з Android, Windows, macOS, iOS та FreeBSD. Його відкритий вихідний код і ліцензія GPL зробили його доступним для широкого використання, а ефективна архітектура та мінімалістичний дизайн дали змогу досягти високої продуктивності навіть на пристроях із обмеженими ресурсами, таких як одноплатні комп'ютери Raspberry Pi.

Однією з головних особливостей WireGuard є його компактний код – всього кілька тисяч рядків на відміну від сотень тисяч у OpenVPN або IPSec. Це не лише полегшує аудит і виявлення вразливостей, але й робить підтримку та розвиток більш прозорими. Завдяки мінімальній складності WireGuard є надзвичайно простим у впровадженні: він не потребує складної інфраструктури сертифікатів чи ручного налаштування маршрутизації. Використання сучасних криптографічних алгоритмів, таких як Curve25519 для обміну ключами, ChaCha20 для шифрування, Poly1305 для аутентифікації, BLAKE2s для хешування та HKDF для генерації ключів, дозволяє досягти високого рівня безпеки без ризику використання застарілих або скомпрометованих технологій.

WireGuard створений як «мережевий тунель» на основі UDP, що забезпечує низьку затримку та швидку передачу пакетів. Протокол працює на рівні ядра операційної системи, що дозволяє максимально ефективно використовувати

ресурси процесора. У тестах продуктивності WireGuard часто перевершує OpenVPN та IPSec, демонструючи значно вищі швидкості з'єднання при меншому споживанні енергії та нижчому навантаженні на систему. Це особливо важливо для мобільних пристроїв, де критичними є показники автономності та стабільності з'єднання при зміні мереж (наприклад, між Wi-Fi і мобільною мережею).

В основі механізму ідентифікації в WireGuard лежить обмін публічними ключами. Кожен учасник VPN-з'єднання має пару ключів: приватний та публічний. Сервер зберігає список дозволених публічних ключів клієнтів, а клієнт зберігає публічний ключ сервера. Таким чином, немає необхідності у централізованих системах сертифікації або використанні паролів, що знижує ризик витоку облікових даних. WireGuard автоматично управляє встановленням і відновленням з’єднання, використовуючи концепцію «молчунного» протоколу – він не передає жодної інформації, поки не почне фактичну передачу даних, що значно підвищує конфіденційність і ускладнює аналіз трафіку з боку зловмисників.

Завдяки своїй архітектурі WireGuard забезпечує не лише високу швидкість, але й низьку ймовірність конфігураційних помилок. Весь процес налаштування, як правило, зводиться до генерації ключів і простого заповнення конфігураційного файлу, в якому зазначаються лише базові параметри, такі як адреса VPN-інтерфейсу, порт та публічні ключі. Це дозволяє швидко розгортати як клієнтські, так і серверні рішення навіть без глибоких знань у сфері комп’ютерних мереж.

Хоча WireGuard є новішим протоколом порівняно з OpenVPN або IPSec і ще не такий широко підтримуваний у деяких корпоративних рішеннях, він стрімко набирає популярності. Багато сучасних VPN-сервісів вже інтегрували підтримку WireGuard через його здатність забезпечити кращу якість з’єднання, особливо в умовах мобільного трафіку, великих затримок або частих змін IP-адрес. У 2020

році протокол було офіційно включено до ядра Linux, що стало важливою віхою у визнанні його стабільності, ефективності та безпеки.

У контексті персональних VPN-серверів WireGuard є ідеальним вибором для домашніх користувачів, які хочуть отримати захищений доступ до своєї мережі, не витрачаючи багато часу на складне налаштування. Його легкість, висока продуктивність та підтримка сучасних криптоалгоритмів роблять його вкрай привабливим рішенням як для ентузіастів, так і для професіоналів. WireGuard демонструє приклад того, яким має бути сучасний VPN-протокол: легким, прозорим, безпечним і готовим до роботи «з коробки» на будь-якому пристрої.

Окрім OpenVPN і WireGuard, у VPN-інфраструктурах використовуються також інші протоколи. IPSec (Internet Protocol Security) — набір протоколів, який забезпечує шифрування та автентифікацію на рівні мережевого протоколу IP. Його часто застосовують у поєднанні з L2TP (Layer 2 Tunneling Protocol) або IKEv2 (Internet Key Exchange version 2). IPSec є одним із найстаріших стандартів, підтримується на більшості пристроїв «з коробки» й широко використовується в корпоративних мережах. Основна перевага – глибока інтеграція в ОС і високий рівень захисту, але через складність налаштування і сумісність із NAT він рідко використовується в домашніх рішеннях. IKEv2/IPSec є варіантом IPSec із більш швидким відновленням з’єднання після обриву, що особливо корисно на мобільних пристроях під час переходу між мережами (Wi-Fi ↔ LTE).

Іншим поширеним протоколом є SSTP (Secure Socket Tunneling Protocol) — це VPN-протокол, який був розроблений корпорацією Microsoft і вперше з’явився у Windows Vista SP1. Його головною особливістю є використання протоколу HTTPS (TCP-порт 443) для встановлення захищеного VPN-тунелю між клієнтом і сервером. Завдяки цьому SSTP може безперешкодно працювати через більшість мереж, у тому числі ті, які обмежують трафік і блокують нестандартні порти, адже порт 443 є стандартним для безпечного веб-трафіку і рідко фільтрується чи блокується.

Однією з головних переваг SSTP є глибока інтеграція з операційною системою Windows. SSTP повністю підтримується Windows без необхідності встановлення стороннього програмного забезпечення. Це робить протокол особливо зручним для корпоративних мереж, де часто використовуються системи на базі Windows Server і клієнти з Windows-пристроїв. Крім того, SSTP забезпечує високий рівень безпеки завдяки використанню протоколу SSL/TLS 3.0 або новіших версій, які підтримують шифрування сертифікатами X.509, а також перевірку справжності серверів за допомогою PKI (інфраструктури відкритих ключів). Таким чином, обидві сторони з'єднання можуть бути автентифіковані, що суттєво підвищує загальний рівень безпеки.

SSTP також захищений від атак типу «man-in-the-middle», оскільки перед встановленням тунелю клієнт перевіряє дійсність SSL-сертифіката сервера. Якщо сертифікат є недійсним або самопідписаним без довіри, з’єднання не буде встановлено. Сам процес обміну ключами та шифрування даних є аналогічним до HTTPS, що забезпечує конфіденційність і цілісність переданих даних. Використання TCP як транспортного протоколу робить SSTP надійним у мережах із високим рівнем фільтрації або у середовищах, де інші VPN-протоколи, такі як PPTP або L2TP, не працюють належним чином через обмеження мережевого середовища.

Незважаючи на переваги, SSTP має також певні обмеження. Найбільше з них – це обмежена підтримка за межами екосистеми Windows. Хоча існують деякі реалізації SSTP для Linux (наприклад, через проект sstp-client), їх підтримка є обмеженою і потребує додаткових налаштувань. Android також не має нативної підтримки SSTP, хоча деякі сторонні додатки частково реалізують протокол. Через це SSTP зазвичай не є першим вибором для кросплатформних рішень або відкритих систем, де кращим варіантом є OpenVPN або WireGuard.

Крім того, оскільки SSTP працює поверх TCP, це може призводити до так званої «TCP-over-TCP» проблеми при передаванні деяких типів трафіку. Такий підхід може погіршити продуктивність і спричинити додаткову затримку при

повторній передачі пакетів, особливо у випадках, коли одночасно використовується ще один рівень TCP-з'єднання всередині VPN-тунелю. Це робить SSTP менш ефективним у порівнянні з UDP-базованими протоколами, такими як WireGuard або OpenVPN у режимі UDP.

Ще однією потенційною слабкістю є залежність від Microsoft як єдиного основного розробника протоколу. На відміну від OpenVPN або WireGuard, які є відкритими та підтримуються спільнотою, SSTP є закритим протоколом з обмеженою документацією та відсутністю відкритої специфікації. Це обмежує можливості аналізу безпеки та розвитку альтернативних реалізацій. У корпоративних середовищах SSTP часто використовується разом з Windows Server Routing and Remote Access Service (RRAS), що забезпечує централізоване адміністрування та контроль доступу.

Незважаючи на деякі недоліки, SSTP залишається потужним інструментом для забезпечення безпечного VPN-доступу в умовах суворої фільтрації трафіку, надаючи надійний захист даних та інтеграцію з існуючими інфраструктурами Windows. Він є популярним вибором для компаній, які використовують виключно продукти Microsoft, і потребують захищеного та стабільного способу підключення віддалених працівників до корпоративної мережі.

Також іноді згадують PPTP (Point-to-Point Tunneling Protocol), який є застарілим і має відомі уразливості. Його використання сьогодні не рекомендується через низький рівень шифрування та відсутність сучасних механізмів захисту.

Таким чином, вибір VPN-протоколу повинен ґрунтуватися на співвідношенні між продуктивністю, рівнем безпеки, складністю налаштування та цільовим середовищем використання. У контексті персонального VPN-сервера на Raspberry Pi найбільш доцільним вибором є WireGuard, який забезпечує високий рівень шифрування при мінімальному навантаженні на систему, але за потреби сумісності або підтримки складніших сценаріїв підключення може бути використаний OpenVPN або IPSec.

1.3 Принципи шифрування і автентифікації у VPN

Шифрування і автентифікація є ключовими принципами, що забезпечують конфіденційність, цілісність і достовірність даних у VPN-з'єднаннях. Завдяки цим механізмам VPN дозволяє захистити передану інформацію від стороннього втручання, прослуховування, модифікації або підміни, а також гарантує, що доступ до мережі отримують лише авторизовані користувачі. Ефективна реалізація шифрування та автентифікації є запорукою безпечної роботи в публічних або недовірених мережах, таких як Інтернет.

Принцип шифрування у VPN базується на перетворенні відкритого тексту в зашифрований (ciphertext) з використанням криптографічних алгоритмів. Лише сторони, які володіють відповідними ключами, можуть розшифрувати передану інформацію. Існує два основних типи шифрування: симетричне і асиметричне. У симетричному шифруванні один і той самий ключ використовується для шифрування та дешифрування, тому безпека залежить від надійної передачі цього ключа обом сторонам. Це шифрування зазвичай використовується для самої передачі даних у VPN, оскільки воно є більш швидким. У асиметричному шифруванні застосовується пара ключів – відкритий та закритий. Один ключ використовується для шифрування, а інший – для дешифрування. Цей підхід дозволяє безпечно обмінюватися симетричними ключами та автентифікувати сторони під час встановлення VPN-з'єднання.

Класичним прикладом асиметричного шифрування у VPN є використання сертифікатів SSL/TLS або криптографічних протоколів, таких як RSA, Diffie-Hellman чи ECDH. Наприклад, при встановленні VPN-тунелю через OpenVPN або IPSec, на початку здійснюється так зване "рукостискання" (handshake), під час якого генеруються ключі сесії за допомогою асиметричної криптографії. Далі ці ключі використовуються для захищеного симетричного шифрування основного потоку даних. Протоколи, як-от TLS (Transport Layer Security), гарантують захист не лише даних, а й самого процесу обміну ключами, що унеможливлює їх

перехоплення навіть у випадку повного контролю над мережею з боку зловмисника.

Важливим елементом безпечного VPN-з'єднання є також автентифікація, тобто перевірка особи користувача або пристрою, який намагається встановити з'єднання. Автентифікація у VPN може здійснюватися різними методами: за допомогою логіна та пароля, цифрових сертифікатів, обміну ключами або мультифакторних механізмів. Найнадійнішим і найчастіше використовуваним методом у сучасних VPN-рішеннях є автентифікація за допомогою відкритого і закритого ключа. У такому випадку клієнт володіє приватним ключем, який зберігається тільки на його пристрої, а VPN-сервер перевіряє його справжність шляхом зіставлення з відповідним публічним ключем. Такий підхід забезпечує високий рівень безпеки, оскільки навіть у разі перехоплення трафіку зловмисник не зможе отримати доступ до мережі без відповідного приватного ключа.

Протокол WireGuard, наприклад, базується саме на цьому принципі. Кожен учасник мережі має пару ключів – публічний і приватний. Під час встановлення з'єднання VPN-клієнт та сервер обмінюються публічними ключами, після чого обчислюється спільний симетричний ключ для шифрування трафіку. Такий обмін відбувається без встановлення постійного з'єднання, що дозволяє зменшити час затримки, спростити конфігурацію та зробити підключення менш помітним для мережевого моніторингу. Крім того, WireGuard використовує сучасні криптографічні алгоритми, як-от ChaCha20 для симетричного шифрування та Poly1305 для перевірки цілісності, що забезпечує швидкість і безпеку навіть на слабких пристроях, таких як Raspberry Pi.

У випадку OpenVPN або IPSec, які дозволяють більш гнучке налаштування, можливе поєднання кількох методів автентифікації – наприклад, сертифікатів клієнта, облікових даних користувача, токенів або двофакторної автентифікації. Також важливим аспектом є перевірка цілісності даних — механізм, який гарантує, що дані не були змінені під час передачі. Це досягається шляхом

створення MAC (Message Authentication Code), який обчислюється для кожного пакету з використанням ключа, відомого лише сторонам з’єднання.

Ще одним важливим елементом є forward secrecy – властивість, яка гарантує, що навіть якщо зловмисник зможе зламати один сесійний ключ, він не зможе розшифрувати попередні або майбутні сесії. Це досягається за рахунок генерації нових ключів для кожної сесії або навіть кожного окремого з'єднання, що знижує ризик компрометації всієї історії переданої інформації.

Таким чином, шифрування та автентифікація у VPN-системах – це взаємопов'язані процеси, які спільно забезпечують повноцінний захист переданої інформації. Якісна реалізація цих принципів залежить від правильного вибору протоколу, налаштування криптографічних параметрів та дотримання сучасних практик безпеки. У персональних VPN-рішеннях, особливо на базі відкритого програмного забезпечення, таких як OpenVPN або WireGuard, користувач має можливість гнучко контролювати рівень захисту та адаптувати систему під власні потреби, забезпечуючи надійний і безпечний доступ до мережі з будь-якого місця у світі.

1.4 Переваги та недоліки персонального VPN-сервера

Персональний VPN-сервер має низку переваг, які роблять його привабливим рішенням для користувачів, що прагнуть максимально контролювати свою мережеву безпеку та конфіденційність. Насамперед, одним із головних плюсів є повний контроль над інфраструктурою. Користувач самостійно керує усім процесом – від налаштування апаратного забезпечення та встановлення програмного забезпечення до адміністрування ключів доступу і моніторингу підключень. Це дає змогу гарантувати, що ніякі сторонні сервіси не мають доступу до особистих даних або журналів активності, що часто є слабким місцем у комерційних VPN-рішеннях.

Ще однією важливою перевагою є можливість уникнення щомісячної абонентської плати, яка характерна для більшості платних VPN-сервісів. Після

одноразової інвестиції в обладнання, таке як Raspberry Pi, та налаштування необхідного програмного забезпечення, користувач отримує постійний доступ до власного VPN-сервера без додаткових витрат. Крім того, персональний сервер дозволяє використовувати VPN не тільки для захисту з'єднання з Інтернетом, але й для доступу до домашньої мережі з будь-якої точки світу. Це особливо корисно для організації віддаленого доступу до NAS-сховищ, камер спостереження, внутрішніх сайтів або медіасерверів.

Щодо продуктивності, персональні VPN-рішення можуть бути оптимізовані під конкретні потреби. Завдяки цьому користувач сам вирішує, які порти відкривати, які протоколи використовувати, яку політику безпеки застосовувати, а також як саме обробляти DNS-запити. Це створює гнучку та розширювану систему, яка з часом може адаптуватися до нових вимог і технологій. Також можна реалізувати додаткові механізми захисту, як-от двофакторна автентифікація, фаєрволи, обмеження доступу за IP-адресами тощо.

Однак, персональний VPN-сервер має і низку обмежень та недоліків, про які необхідно пам’ятати. По-перше, налаштування та підтримка такого рішення вимагають певного рівня технічних знань. Користувач повинен розуміти основи мережевої безпеки, роботи з операційними системами Linux, конфігурації шифрування та діагностики з’єднання. Без відповідної підготовки або досвіду неправильне налаштування може призвести до створення уразливостей у системі або навіть до її компрометації.

Також необхідно враховувати залежність від власного інтернет-з’єднання. Якщо домашній інтернет має динамічну IP-адресу, її зміна може зробити сервер тимчасово недоступним, якщо не налаштовано динамічний DNS. Крім того, у разі перебоїв з електроживленням або неполадок у роботі обладнання користувач може втратити доступ до VPN, що не трапляється з хмарними або професійними комерційними рішеннями, де забезпечується висока доступність та резервування.

Не менш важливим є питання продуктивності. Малопотужні пристрої, як-от Raspberry Pi, мають обмежену обчислювальну здатність і можуть не впоратись з

високим навантаженням або обробкою трафіку від великої кількості клієнтів. Це обмежує масштабованість персонального рішення та робить його менш придатним для великих організацій або високонавантажених сценаріїв.

Нарешті, оновлення програмного забезпечення, патчі безпеки, резервне копіювання ключів і конфігурацій також повністю лежать на відповідальності користувача. Невчасне оновлення чи втрата ключів можуть поставити під загрозу безпеку або повністю вивести систему з ладу.

Таким чином, персональний VPN-сервер — це потужне рішення для тих, хто потребує високого рівня контролю над мережею та безпекою, і готовий витрачати час на його підтримку. Його використання виправдане в ситуаціях, де важлива конфіденційність, довіра до сторонніх провайдерів відсутня або потрібен постійний доступ до локальної мережі. Але разом із тим, це рішення потребує серйозного ставлення до технічного обслуговування та відповідальності за захист власної інфраструктури.

1.5 Порівняння хмарних, комерційних та локальних VPN-рішень

Хмарні, комерційні та локальні VPN-рішення мають суттєві відмінності як у технічному аспекті, так і з точки зору призначення, рівня контролю, зручності використання, безпеки, конфіденційності та фінансових витрат. Хмарні VPN зазвичай надаються як послуги (VPN as a Service) великими провайдерами, які розміщують сервери у численних дата-центрах по всьому світу. Ці рішення створені для забезпечення масштабованості, гнучкості, швидкого доступу до географічно розподілених ресурсів та централізованого адміністрування. Вони активно використовуються у бізнесі, коли необхідно надати безпечний віддалений доступ до корпоративних систем або забезпечити резервне з’єднання у випадку перебоїв у основних каналах зв'язку. Хмарні VPN зазвичай інтегруються з іншими сервісами безпеки – такими як багатофакторна автентифікація, контроль доступу на основі ролей, моніторинг трафіку – та дають змогу централізовано керувати великою кількістю користувачів. Проте такі рішення залежать від стороннього

провайдера, що може створювати ризики витоку даних, втрати контролю над конфігураціями або підпорядкованості певним правовим юрисдикціям.

Комерційні VPN-сервіси, орієнтовані переважно на масового споживача, зазвичай пропонують прості інтерфейси для швидкого підключення до одного з багатьох доступних серверів, розташованих у різних країнах. Вони здебільшого використовуються для забезпечення приватності під час перегляду веб-ресурсів, обходу регіональних обмежень, доступу до заблокованих сайтів або захисту даних у публічних Wi-Fi-мережах. Такі сервіси працюють за моделлю підписки, мають власні мобільні й десктопні додатки, а також обіцяють високі швидкості, мінімальну затримку та відсутність збереження логів. Однак у реальності деякі з них не завжди дотримуються заявлених стандартів конфіденційності. Крім того, користувач практично не має контролю над тим, як працює сервер, де фізично розташовані дані та які саме алгоритми шифрування використовуються. Технічно просунутим користувачам може бракувати гнучкості в налаштуванні або кастомізації з'єднань.

Локальні VPN-рішення базуються на самостійній інфраструктурі, яка розгортається користувачем або організацією. Це може бути самостійно налаштований VPN-сервер, розміщений на пристрої типу Raspberry Pi, NAS або виділеному сервері. Такий підхід забезпечує найвищий рівень контролю, оскільки весь трафік проходить через власну систему, де можна повністю керувати конфігурацією, шифруванням, ключами доступу та журналюванням. Локальні VPN є ідеальними для домашніх користувачів, які бажають безпечно підключатись до своєї мережі з будь-якого місця, отримувати доступ до домашніх файлів, камер спостереження або серверів, не довіряючи стороннім компаніям. Вони також використовуються в малому бізнесі, де критично важливо уникати залежності від зовнішніх сервісів. Проте недоліками є потреба в базових технічних знаннях, відповідальність за обслуговування, оновлення програмного забезпечення та забезпечення безпеки. Локальні сервери також мають обмежену географічну доступність – якщо VPN працює на домашньому роутері або

одноплатному комп’ютері, його IP-адреса та якість інтернет-з’єднання впливають на швидкість і стабільність доступу.

У підсумку, вибір між хмарними, комерційними та локальними VPN-рішеннями залежить від конкретних потреб користувача. Якщо важливими є зручність, глобальна доступність і мінімальні технічні зусилля – комерційний VPN виглядає привабливо. У випадку корпоративної інфраструктури або великих команд доцільно використовувати хмарні рішення з підтримкою масштабування та централізованого керування. А для тих, хто цінує максимальну приватність, контроль і безпеку, оптимальним буде локальний VPN, хоча він потребує відповідної підготовки та технічної підтримки.

1.6 Висновки до першого розділу

Перший розділ дипломної роботи був присвячений аналізу теоретичних основ захищеного доступу до мережі за допомогою технології VPN. Було з’ясовано, що віртуальні приватні мережі є одним із найефективніших засобів забезпечення конфіденційності, цілісності та доступності передаваних даних у глобальних мережах, зокрема Інтернеті. VPN-технологія дозволяє створювати зашифровані тунелі між користувачем і сервером, через які передається весь мережевий трафік, що виключає можливість його перехоплення або підміни третіми сторонами. На основі цього підходу реалізується захищений канал, який забезпечує не лише конфіденційність даних, але й автентифікацію сторін, а також дозволяє приховати реальне місцезнаходження користувача, що є важливим у багатьох сценаріях використання – від роботи в публічних мережах до обходу регіональних обмежень.

У цьому розділі було детально розглянуто поняття VPN, принципи його роботи, типові сценарії застосування, а також класифікацію основних протоколів, які використовуються в реалізації VPN-з'єднань. Особливу увагу приділено двом сучасним і найпопулярнішим протоколам – OpenVPN та WireGuard. OpenVPN є

перевіреним часом і широко підтримуваним рішенням, яке дозволяє гнучко налаштовувати з'єднання з високим рівнем безпеки, але при цьому є дещо складнішим у конфігурації. WireGuard, у свою чергу, є новішим, більш легким та продуктивним протоколом, який демонструє високу ефективність та простоту використання завдяки сучасній криптографії та мінімалістичному дизайну. Ці протоколи були проаналізовані з точки зору продуктивності, сумісності, простоти налаштування, безпеки та можливостей масштабування.

Також у розділі було розкрито ключові принципи шифрування і автентифікації, що лежать в основі роботи VPN-сервісів. Зокрема, розглянуто симетричне та асиметричне шифрування, генерацію та обмін ключами, цифрові сертифікати, функції гешування і механізми автентифікації користувачів. Важливо розуміти, що надійність VPN-з’єднання напряму залежить не лише від вибору протоколу, а й від правильно налаштованої криптографічної системи, яка повинна відповідати сучасним вимогам стійкості до атак.

У межах порівняння типів VPN-рішень було розглянуто відмінності між хмарними, комерційними та локальними (персональними) VPN-серверами. Було зроблено висновок, що персональні VPN-рішення мають переваги у вигляді повного контролю над інфраструктурою, відсутності залежності від сторонніх провайдерів та можливості гнучкої адаптації під індивідуальні потреби. Водночас такі рішення вимагають базових технічних знань для налаштування, що робить їх менш доступними для пересічних користувачів. Комерційні VPN-сервіси, попри зручність та широкий функціонал, не завжди гарантують конфіденційність, оскільки користувач змушений довіряти сторонній компанії. Хмарні VPN-рішення можуть бути оптимальними для бізнесу, однак часто потребують додаткових витрат та управлінських ресурсів.

2 ПІДГОТОВКА СЕРЕДОВИЩА ДЛЯ РОЗГОРТАННЯ ПЕРСОНАЛЬНОГО VPN-СЕРВЕРА

2.1 Вибір апаратної та програмної платформи

Вибір апаратної та програмної платформи є ключовим етапом для успішної реалізації персонального VPN-сервера, адже від нього залежить стабільність роботи, продуктивність та безпека всієї мережевої інфраструктури. Для даного проєкту обрано одноплатний комп'ютер Raspberry Pi, зокрема модель, що підтримує архітектуру ARM з 64-бітною обробкою, наприклад Raspberry Pi 4, яка має до 8 ГБ оперативної пам'яті, чотириядерний процесор ARM Cortex-A72 з частотою 1.5 ГГц, а також високошвидкісний мережевий інтерфейс Gigabit Ethernet і підтримку бездротових стандартів Wi-Fi 802.11ac.

Цей апаратний вибір обумовлений оптимальним співвідношенням вартості, розміру та технічних можливостей. Raspberry Pi має низьке енергоспоживання – близько 3-7 Вт, що робить його ідеальним для роботи у цілодобовому режимі без значних затрат на електроенергію. Важливим технічним аспектом є наявність достатньої кількості USB-портів для підключення додаткових периферійних пристроїв та розширень, таких як зовнішні накопичувачі для резервного копіювання або мережеві адаптери для розширення функціоналу.

В якості операційної системи було обрано Raspberry Pi OS (раніше Raspbian), що базується на стабільній версії Debian Linux. Ця ОС оптимізована для ARM-архітектури і забезпечує підтримку широкого спектру програмного забезпечення, зокрема інструментів для мережевої взаємодії, серверних додатків і засобів безпеки. Важливо відзначити, що Raspberry Pi OS має можливість працювати як у графічному, так і у мінімальному серверному режимі (без GUI), що дозволяє максимально звільнити ресурси для роботи VPN-сервера.

Щодо програмної складової, для побудови VPN-сервера було використано PiVPN – спеціалізований інсталяційний скрипт з відкритим кодом, що значно

спрощує налаштування і керування VPN. PiVPN автоматизує процес генерації ключів, налаштування мережевого інтерфейсу, файрвола та служб VPN. Важливо, що PiVPN підтримує два основні сучасні VPN-протоколи: OpenVPN і WireGuard. З технічної точки зору WireGuard є новішим, більш легким і продуктивним протоколом, який використовує сучасні криптографічні алгоритми і мінімальний код, що зменшує площу для потенційних вразливостей.

Враховуючи потребу у гнучкому та безпечному середовищі розробки, для тестування та налагодження VPN-сервера було прийнято рішення запускати Raspberry Pi OS у віртуальному середовищі VirtualBox. VirtualBox дозволяє створити віртуальну машину з параметрами, що максимально наближені до фізичного Raspberry Pi, з можливістю налаштування мережевих адаптерів у режимах NAT або мостового підключення (bridge mode). Ці режими є критичними для тестування роботи VPN, оскільки дозволяють контролювати маршрутизацію мережевого трафіку та імітувати умови реального підключення з різних мереж.

Для емуляції апаратної архітектури ARM на x86-платформі VirtualBox, як правило, використовується спеціалізований образ операційної системи Raspberry Pi OS, адаптований під x86-архітектуру, або застосовуються додаткові засоби емуляції, наприклад QEMU. Проте для більшості навчальних і тестових задач достатньо налаштувати відповідні параметри віртуальної машини, включаючи виділення оперативної пам’яті (мінімум 2 ГБ для стабільної роботи), достатнього дискового простору (10-20 ГБ), а також коректну конфігурацію мережі.

З точки зору безпеки, Raspberry Pi та PiVPN забезпечують можливість налаштування багатофакторної автентифікації, обмеження доступу за IP-адресами, застосування сучасних протоколів шифрування (AES-256, ChaCha20), а також автоматичне оновлення системи та служб, що є критичним для запобігання експлуатації відомих вразливостей. Використання Linux-інструментів, таких як UFW (Uncomplicated Firewall), дозволяє легко і ефективно налаштувати файрвол, обмежити доступ до мережі та логувати активність.

В результаті, вибір саме такої апаратної та програмної платформи, поєднаної з використанням віртуалізації, дає змогу не лише отримати надійний та продуктивний VPN-сервер, але й забезпечити гнучкість у процесі розробки, тестування та подальшого масштабування рішення. Цей підхід є оптимальним як для навчальних цілей, так і для впровадження у реальних домашніх чи малих офісних мережах.

VirtualBox – це потужний, кросплатформений програмний продукт для віртуалізації, який дозволяє створювати і керувати віртуальними машинами (ВМ) на фізичних комп'ютерах з різними операційними системами. Він був розроблений компанією InnoTek GmbH, а пізніше придбаний Oracle Corporation, що сприяло подальшому розвитку і стабілізації проєкту. VirtualBox є продуктом із відкритим вихідним кодом, поширюваним під ліцензією GPL, що забезпечує широке застосування як у комерційних, так і у некомерційних цілях.

З технічної точки зору VirtualBox реалізує гіпервізор типу 2, тобто працює поверх операційної системи хоста, забезпечуючи ізоляцію віртуальних машин від основної системи, але з можливістю ефективного використання ресурсів апаратного забезпечення. Його архітектура складається з кількох ключових компонентів: ядра гіпервізора, драйверів для апаратних інтерфейсів, менеджера віртуальних машин, а також графічного інтерфейсу користувача та командного рядка. Важливо, що VirtualBox підтримує широкий спектр хост-операційних систем, включно з Windows, Linux, macOS і Solaris, що робить його надзвичайно гнучким і доступним для різних користувачів.

Однією з головних переваг VirtualBox є його здатність емулювати апаратні компоненти віртуальної машини, зокрема процесори, оперативну пам'ять, мережеві адаптери, жорсткі диски, USB-порти, звукові карти і відеоадаптери. Завдяки цьому користувачі можуть запускати практично будь-яку операційну систему всередині віртуального середовища, включно з різними версіями Windows, Linux-дистрибутивами, BSD-системами і навіть менш поширеними ОС. Віртуальна машина у VirtualBox поводиться так само, як і фізичний комп'ютер,

що дозволяє тестувати програмне забезпечення, розробляти складні мережеві сценарії, проводити навчання або безпечно експериментувати без ризику пошкодження основної системи.

VirtualBox підтримує розширені можливості, такі як підтримка 64-бітних гостьових ОС, багатоядерних процесорів, апаратної віртуалізації (через Intel VT-x та AMD-V), що значно покращує продуктивність і стабільність віртуальних машин. Крім того, доступні інструменти для роботи зі знімками стану (snapshots), які дозволяють зберігати стан ВМ у певний момент часу і повертатися до нього при необхідності. Це дуже корисно для розробників і тестувальників, які можуть експериментувати з конфігурацією або програмним забезпеченням, не ризикуючи втратити стабільну версію системи.

Ще однією важливою функцією VirtualBox є можливість тонкої налаштування мережі. Підтримуються різні режими роботи мережевих адаптерів: NAT, мостовий режим (bridge), внутрішня мережа (internal network), мережа лише для хоста (host-only) і спеціалізовані варіанти. Це дає змогу моделювати складні мережеві топології, реалізовувати різноманітні варіанти підключення віртуальних машин до інтернету або локальної мережі, а також організовувати ізольовані середовища для тестування без ризику впливу на інші пристрої.

VirtualBox також інтегрується з різними інструментами автоматизації та управління, такими як Vagrant, що спрощує розгортання віртуальних середовищ у рамках DevOps-процесів. Існують командні інтерфейси (CLI), які дозволяють запускати, зупиняти, змінювати конфігурації ВМ, створювати та відновлювати знімки без необхідності відкривати графічний інтерфейс.

З точки зору продуктивності, хоча VirtualBox і є гіпервізором типу 2, він забезпечує достатньо високий рівень швидкодії завдяки підтримці апаратної віртуалізації і оптимізаціям у роботі з пам’яттю та дисковою підсистемою. Однак для найбільш вимогливих завдань або серверних навантажень частіше використовують гіпервізори типу 1 (bare-metal), такі як VMware ESXi чи Microsoft Hyper-V, які запускаються безпосередньо на апаратному рівні. Проте для

більшості освітніх, тестових, розробницьких та домашніх застосувань VirtualBox є більш ніж достатнім і дозволяє гнучко керувати віртуальними машинами на звичайних ПК або ноутбуках.

Серед недоліків VirtualBox іноді відзначають меншу продуктивність порівняно з деякими конкурентами, а також можливі проблеми сумісності з певними гостевими операційними системами або драйверами. Однак за рахунок постійних оновлень та активної спільноти користувачів більшість таких проблем вирішуються, а документація і підтримка є досить широкими.

Застосування VirtualBox дуже різноманітне: від навчальних цілей і розробки програмного забезпечення до побудови тестових середовищ для мережевих інженерів, імітації серверних конфігурацій, запуску застарілих або експериментальних операційних систем, а також для безпечного запуску програм, які можуть становити загрозу для основної системи. У випадку з реалізацією VPN-сервера на Raspberry Pi VirtualBox дає можливість емулювати всю апаратну платформу без необхідності мати фізичний пристрій під рукою, що суттєво прискорює процес розробки, тестування і налагодження.

Таким чином, VirtualBox є універсальним інструментом віртуалізації з широким спектром можливостей, що дозволяє створювати і керувати ізольованими, настроюваними віртуальними машинами, надаючи розробникам, адміністраторам і звичайним користувачам гнучкі інструменти для роботи з різними операційними системами та мережевими середовищами. Його відкритий код, кросплатформеність і постійний розвиток роблять його одним із найбільш популярних і доступних рішень у сфері віртуалізації.

2.2 Встановлення та налаштування VirtualBox

Встановлення VirtualBox є першим і надзвичайно важливим кроком для створення віртуального середовища, у якому буде емульовано платформу Raspberry Pi. VirtualBox – це кросплатформений інструмент віртуалізації, який дозволяє запускати різноманітні операційні системи у вигляді віртуальних машин на вашому основному комп'ютері (хості). Завдяки цьому забезпечується ізольоване середовище для розробки, тестування та експериментів, без ризику пошкодження основної системи.

Спочатку необхідно завантажити інсталяційний пакет VirtualBox з офіційного сайту компанії Oracle. Для цього слід перейти на сайт та обрати версію, що відповідає операційній системі вашого комп'ютера — Windows, Linux або macOS. Рекомендується завантажувати останню стабільну версію, оскільки вона містить виправлення помилок і нові функції.

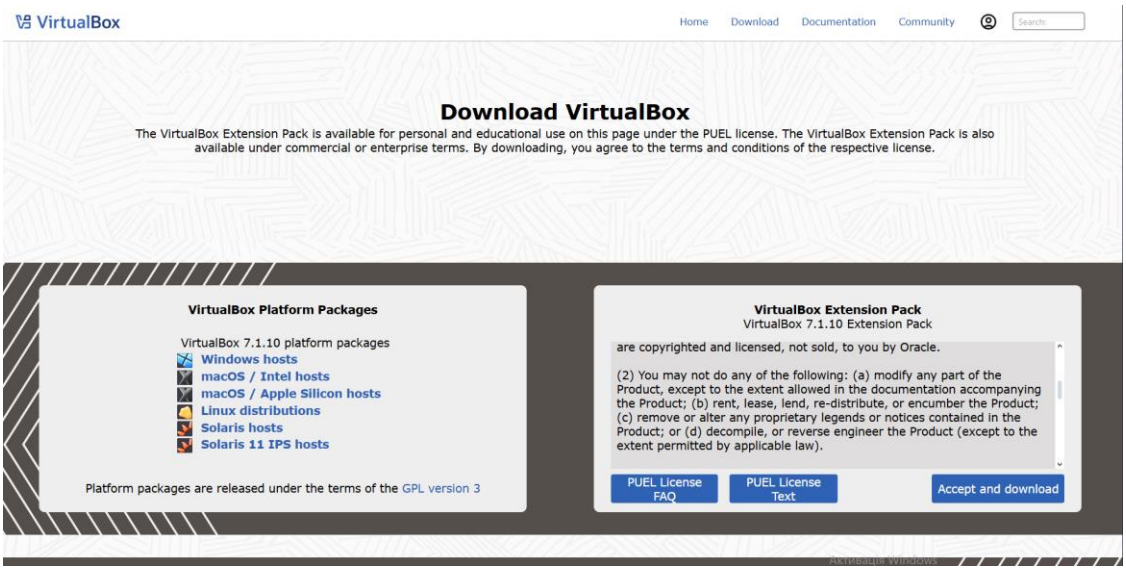


Рисунок 2.1 – Офіційний сайт компанії Oracle

Після завантаження інсталятора слід запустити його із правами адміністратора та виконати послідовність кроків інсталяції. В інсталяторі можна обрати шлях встановлення, компоненти для інсталяції (зазвичай за замовчуванням

підходять усі), а також дозволити установку мережевих драйверів, які потрібні для роботи віртуальних мереж.

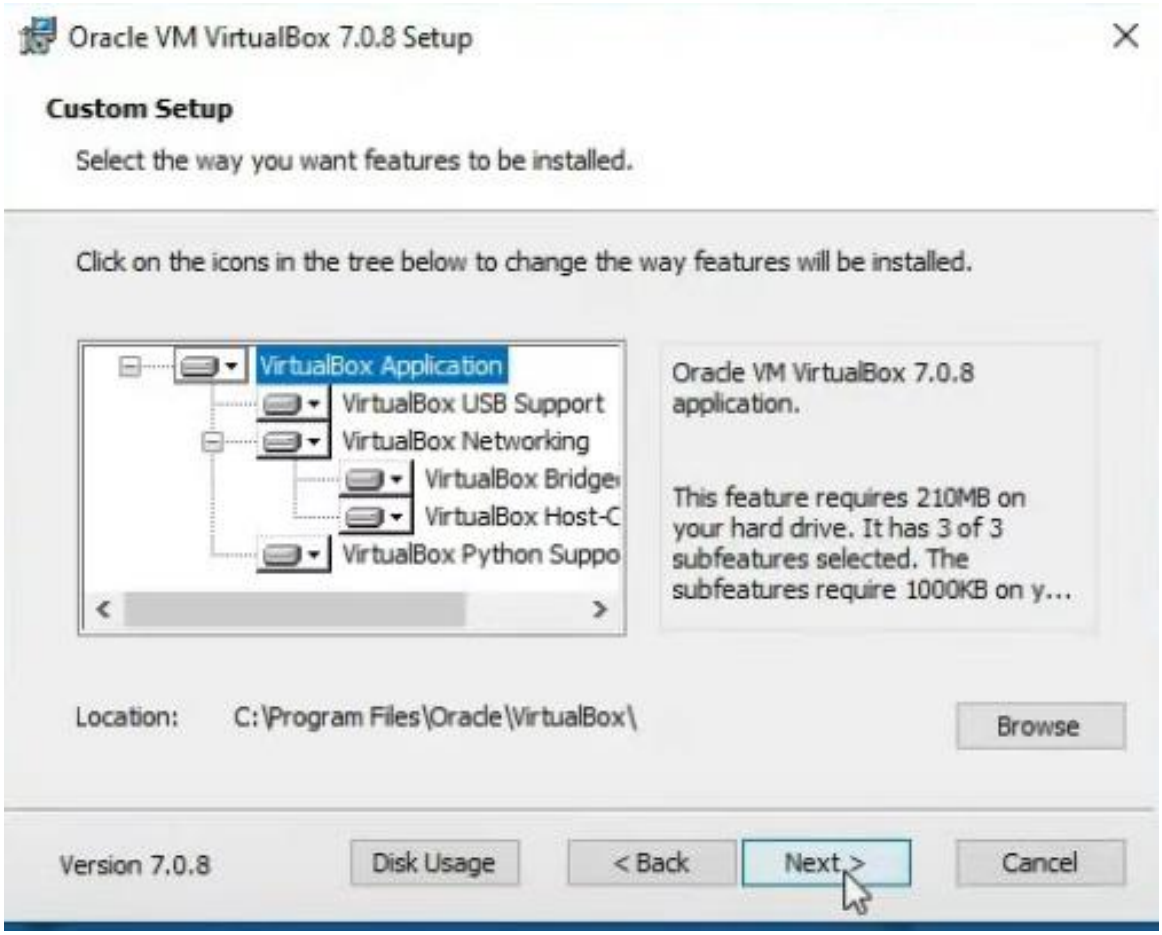


Рисунок 2.2 – Інсталяційне вікно VirtualBox

Протягом інсталяції система може попередити про тимчасове відключення мережі через інсталяцію мережевих драйверів. Це стандартна процедура, яку слід підтвердити для коректної роботи мережевих функцій VirtualBox.

Після успішного завершення інсталяції рекомендується перезавантажити комп’ютер, щоб зміни в драйверах набрали чинності.

Після запуску VirtualBox користувач потрапляє до головного вікна менеджера віртуальних машин. Тут можна створювати нові віртуальні машини, керувати вже існуючими, імпортувати або експортувати ВМ, налаштовувати мережеві параметри, додавати віртуальні диски та пристрої.

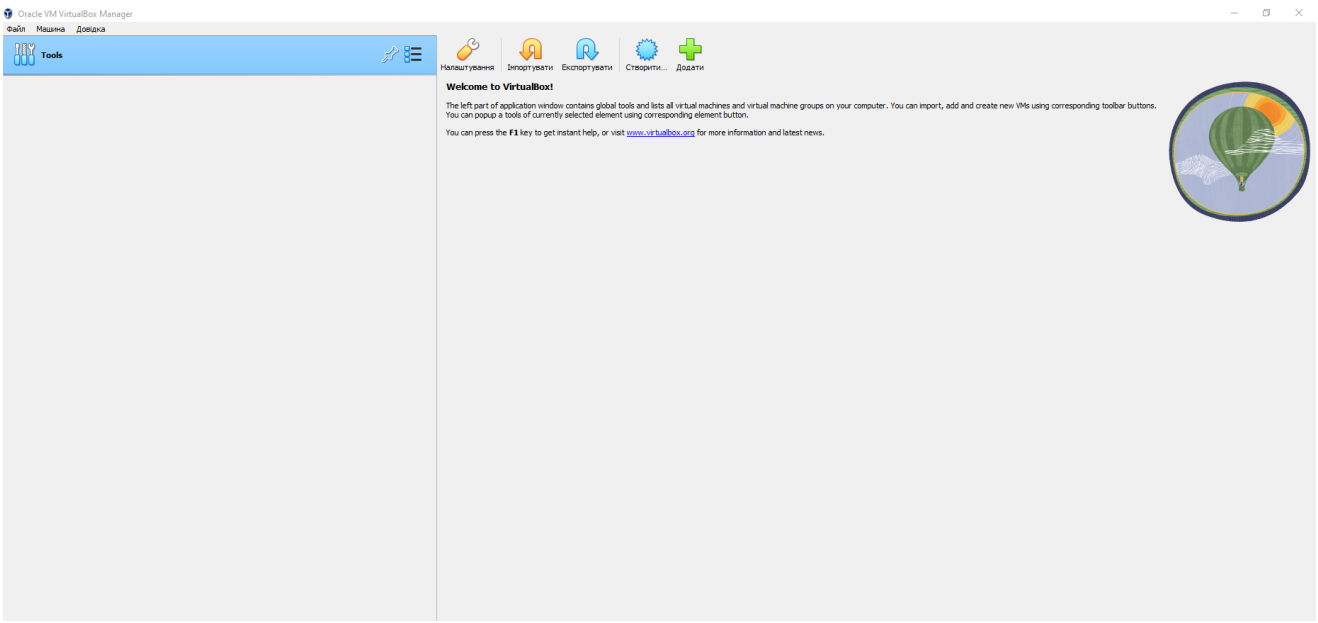


Рисунок 2.3 – Головне вікно VirtualBox

Щоб створити віртуальну машину для Raspberry Pi OS, необхідно натиснути кнопку "Створити" (Create) і задати основні параметри. Серед них — ім’я ВМ, тип операційної системи (Linux) і версія (Debian 64-bit, оскільки Raspberry Pi OS базується на Debian). Важливо правильно вказати тип і версію ОС для оптимальної роботи віртуальної машини.

Далі слід виділити необхідний обсяг оперативної пам’яті. Рекомендовано виділити не менше 1024 МБ, а для комфортної роботи — 2048 МБ або більше, залежно від ресурсів хост-машини. Недостатній обсяг пам’яті призведе до повільної роботи ВМ або навіть збоїв.

У параметрах віртуальної машини необхідно вказати обсяг оперативної пам’яті створити або підключити віртуальний жорсткий диск розміром від 8 до 20 ГБ, та налаштувати мережевий адаптер у режимі NAT або Bridge – залежно від цілей використання. Додатково важливо увімкнути підтримку апаратної віртуалізації (VT-x або AMD-V) у BIOS комп’ютера та у властивостях машини VirtualBox. Ці налаштування дозволяють віртуальній машині стабільно

працювати, підтримуючи встановлення Raspberry Pi OS та забезпечуючи повну функціональність для наступних мережевих конфігурацій.

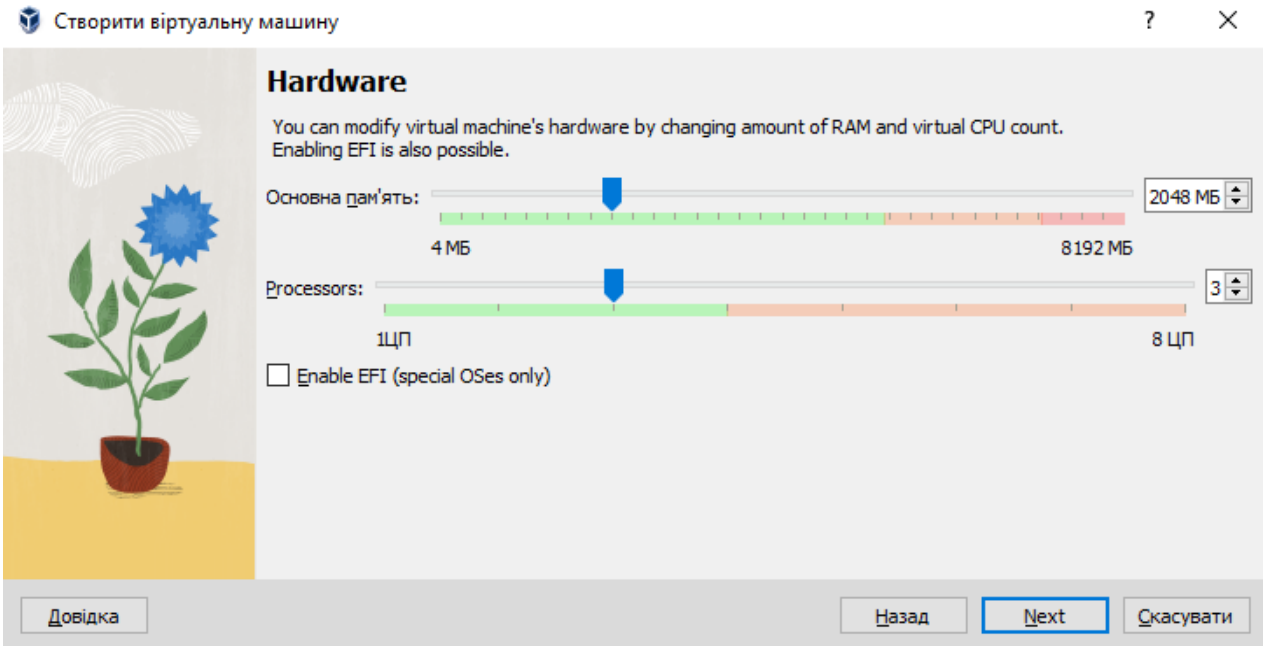


Рисунок 2.4 – Налаштування виділеної оперативної памяті та кількості ядер

Наступним кроком є створення віртуального жорсткого диска. Для Raspberry Pi OS оптимальним варіантом буде динамічно розширюваний VDI-диск розміром від 10 до 20 ГБ, що дозволяє зберігати систему та додаткові файли. Можна обрати тип зберігання – на фізичному диску або у вигляді образу.

Також важливо налаштувати інтерфейс мережі у VirtualBox. Від типу адаптера залежить доступ віртуальної машини до зовнішньої мережі або локальної інфраструктури. Режим NAT є найпростішим для підключення до Інтернету без додаткових налаштувань, тоді як режим "Міст" (Bridge Adapter) дозволяє взаємодіяти з іншими пристроями у локальній мережі, ніби віртуальна машина — це окремий фізичний комп’ютер. У рамках реалізації VPN-сервера часто обирається саме Bridge, що забезпечує стабільне з'єднання та підтримку переадресації портів для підключення ззовні. Додатково можна налаштувати

USB-контролери, аудіо, спільні папки та інші параметри, однак для роботи з серверними сервісами це не є критично необхідним.

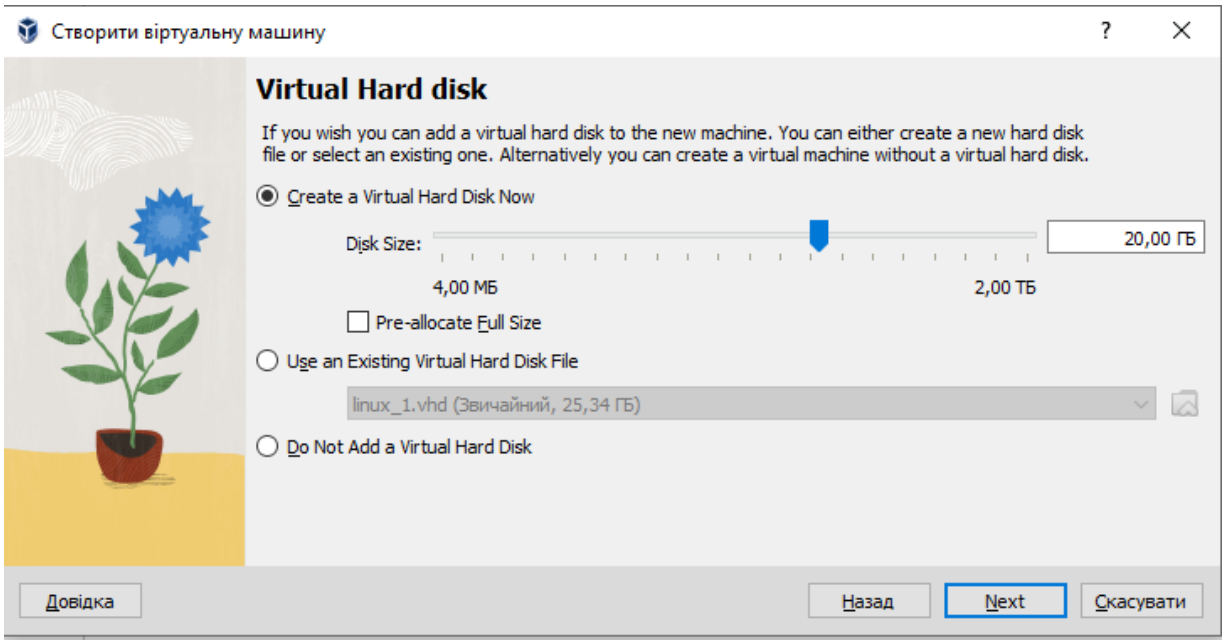


Рисунок 2.5 – Налаштування виділеної памяті

Особливо важливим аспектом є правильна конфігурація мережевого адаптера в VirtualBox. Для коректної роботи VPN-сервера слід налаштувати мережевий адаптер у режимі мостового підключення (Bridge Adapter), який дозволяє віртуальній машині отримувати власну IP-адресу у локальній мережі. Це необхідно, щоб сервер був доступний іншим пристроям у мережі і ззовні через інтернет.

В налаштуваннях мережевого адаптера потрібно вибрати фізичний мережевий інтерфейс хоста, через який буде проходити трафік. Також варто переконатися, що увімкнено апаратне прискорення мережі, якщо це підтримується.

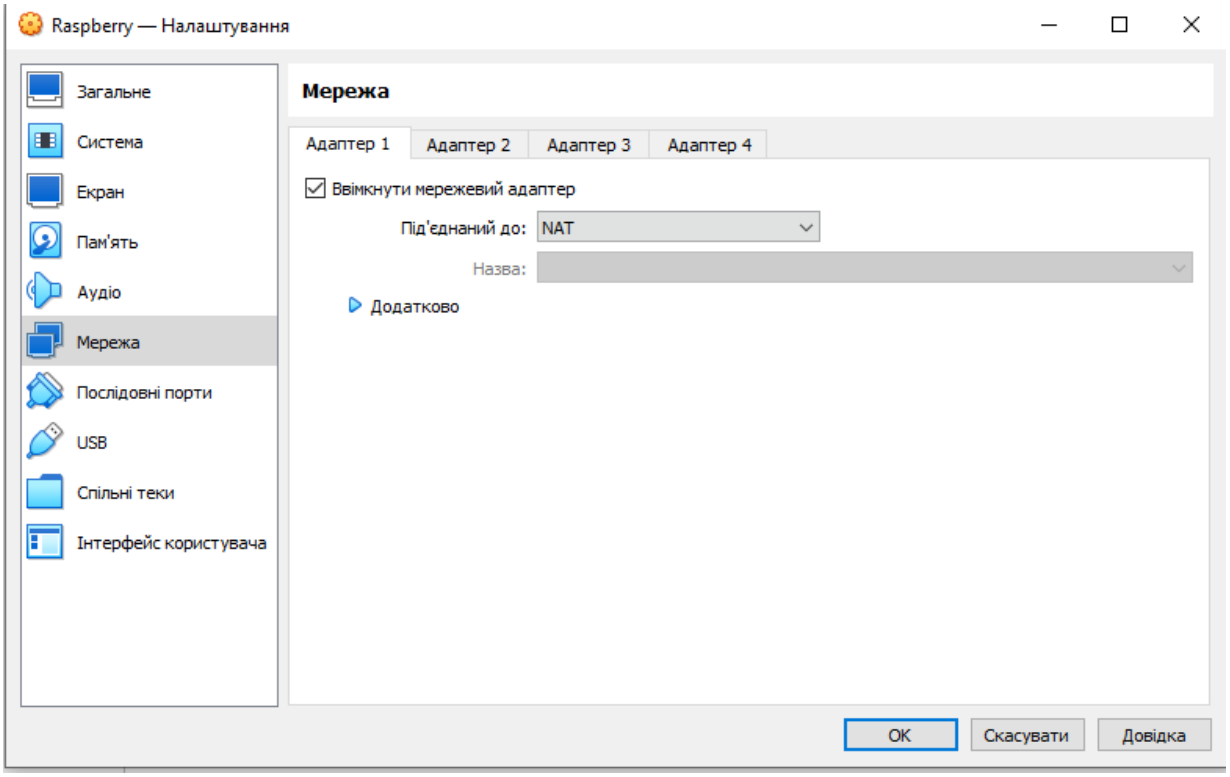


Рисунок 2.6 – Налаштування мережі

Для покращення продуктивності рекомендується активувати підтримку апаратної віртуалізації у BIOS комп'ютера (Intel VT-x або AMD-V) і перевірити, що ця опція увімкнена у VirtualBox. У налаштуваннях віртуальної машини слід перевірити, що відповідна галочка активна.

Крім того, VirtualBox дозволяє налаштовувати графічні параметри, наприклад, обсяг відеопам'яті та 3D-акселерацію, що особливо корисно при роботі з графічними інтерфейсами гостьової ОС.

Ще одним важливим параметром для підвищення стабільності роботи віртуального середовища є кількість виділених процесорних ядер та обсяг оперативної пам'яті. Рекомендується надати щонайменше 2 ядра процесора (якщо це дозволяє хостова система) та не менше 2048 МБ оперативної пам'яті для коректного функціонування Raspberry Pi OS у графічному режимі. Якщо передбачається робота лише в терміналі або з мінімалістичним інтерфейсом, допустимим буде і менший обсяг пам'яті, однак це може впливати на швидкодію

при інтенсивному навантаженні або при встановленні та конфігурації додаткових сервісів, таких як VPN.

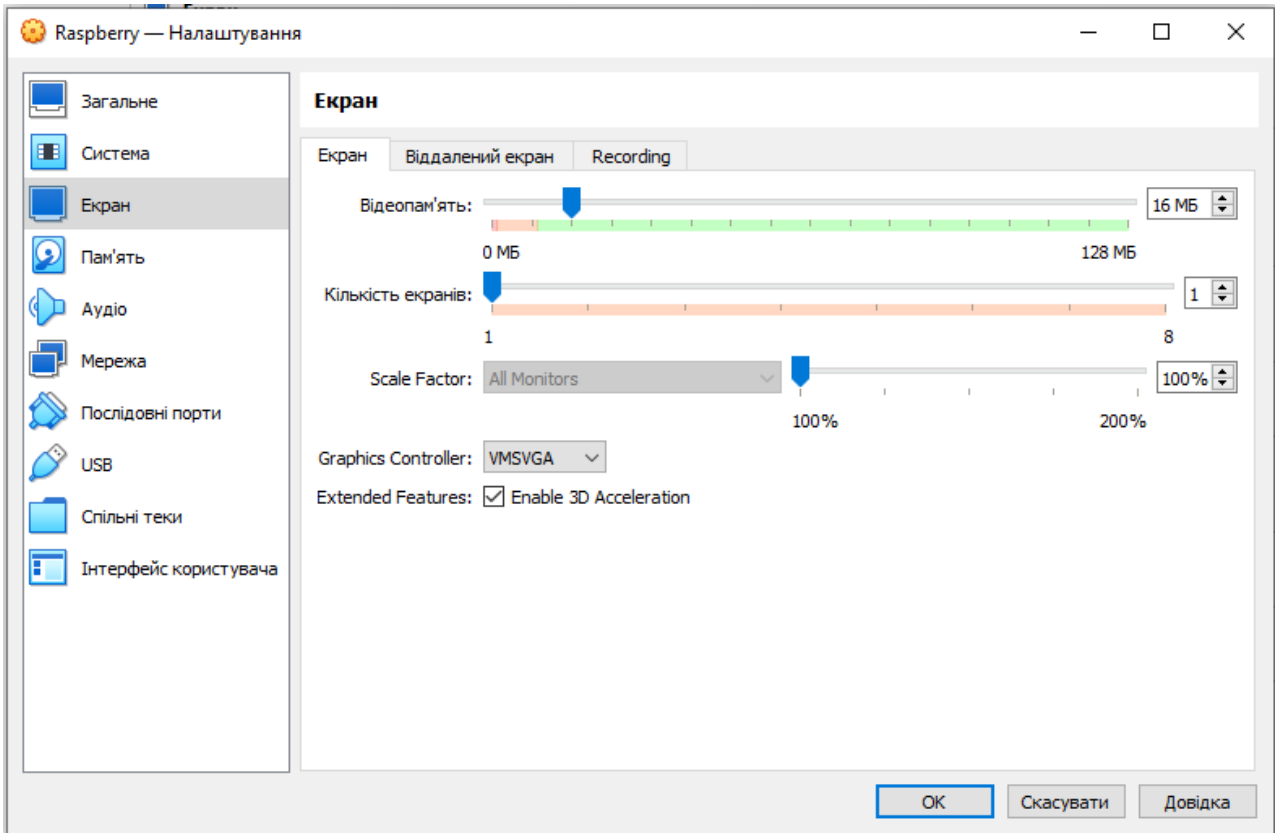


Рисунок 2.7 – Вмикання 3D-акселерацію

Після завершення всіх налаштувань віртуальну машину можна запускати та приступати до встановлення операційної системи Raspberry Pi OS. Завдяки ретельній конфігурації VirtualBox створює стабільне, продуктивне і максимально наближене до реального апаратного середовища віртуальне середовище, яке ідеально підходить для реалізації VPN-сервера.

2.3 Завантаження та підготовка Raspberry Pi OS для емуляції

Для коректної емуляції Raspberry Pi у VirtualBox необхідно використовувати сумісну версію операційної системи, призначену для архітектури x86,

оскільки стандартна Raspberry Pi OS побудована на базі ARM-процесорів, які не підтримуються VirtualBox без спеціальної емуляції, як-от через QEMU. Щоб уникнути ускладнень, використовується версія Raspberry Pi OS Desktop for PC and Mac, яка базується на Debian з графічною оболонкою PIXEL і підтримує запуск на звичайних комп'ютерах. Ця версія ідеально підходить для тестування та налаштування віртуального середовища до перенесення конфігурації на фізичний Raspberry Pi.

Образ Raspberry Pi OS для ПК можна завантажити з офіційного сайту Raspberry Pi. Після переходу на відповідну сторінку завантажень потрібно знайти секцію, що стосується операційної системи Raspberry Pi Desktop, і обрати ISO-файл. Формат ISO дозволяє завантажити цей образ напряму у VirtualBox як віртуальний оптичний диск. Це полегшує процес інсталяції, оскільки не потрібно конвертувати образ у формат VDI чи VMDK. Завантаження такого образу зазвичай займає кілька хвилин залежно від швидкості інтернет-з’єднання, а його розмір становить близько 2,5 ГБ. Після завантаження бажано перевірити хеш-суму (SHA256), щоб пересвідчитись у цілісності файлу.

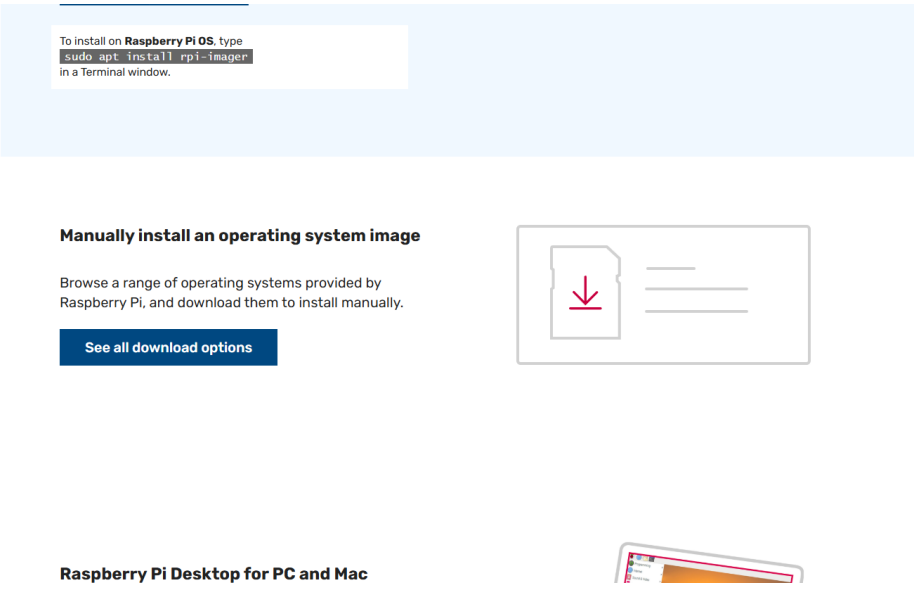


Рисунок 2.8 – завантаження Raspberry Pi OS

Після завантаження образу можна перейти до налаштування віртуальної машини у VirtualBox. У параметрах накопичувача потрібно додати новий оптичний диск та вказати шлях до завантаженого ISO-файлу. Це дозволяє запустити систему у Live-режимі або розпочати повноцінну інсталяцію на віртуальний жорсткий диск. Сама інсталяція відбувається у звичному графічному середовищі. На стартовому екрані пропонується вибір між запуском у режимі ознайомлення (Live) або інсталяцією на диск. Для збереження змін, налаштувань та встановлення додаткових компонентів потрібно обирати саме повну інсталяцію.

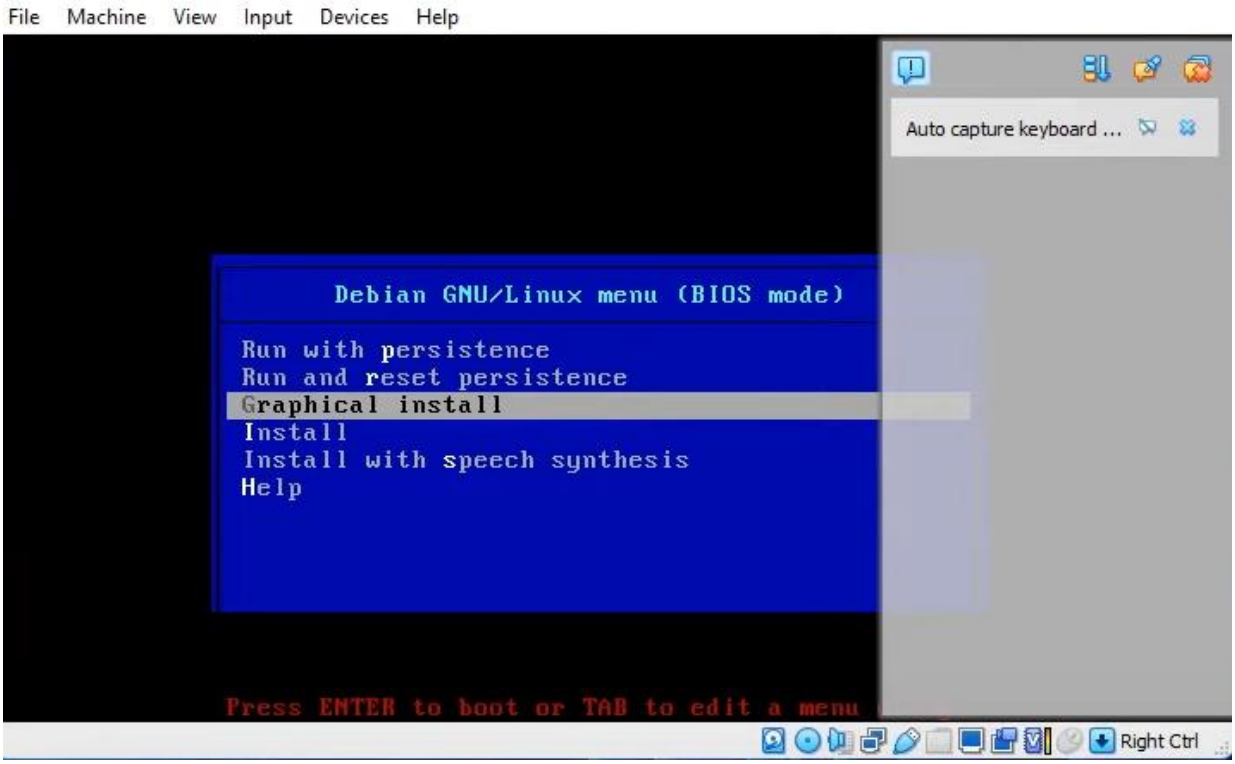


Рисунок 2.9 – Вибір режиму запуску

Після початку інсталяції система пропонує налаштувати мову, часову зону, розкладку клавіатури, ім’я користувача, пароль і параметри диска. Важливо переконатися, що інсталяція відбувається саме на віртуальний диск, створений під час налаштування віртуальної машини, і що він має достатній об’єм –

щонайменше 10 ГБ. Файлова система автоматично розбивається, і розпочинається копіювання файлів на диск. Залежно від параметрів комп'ютера процес може займати до 30 хвилин.

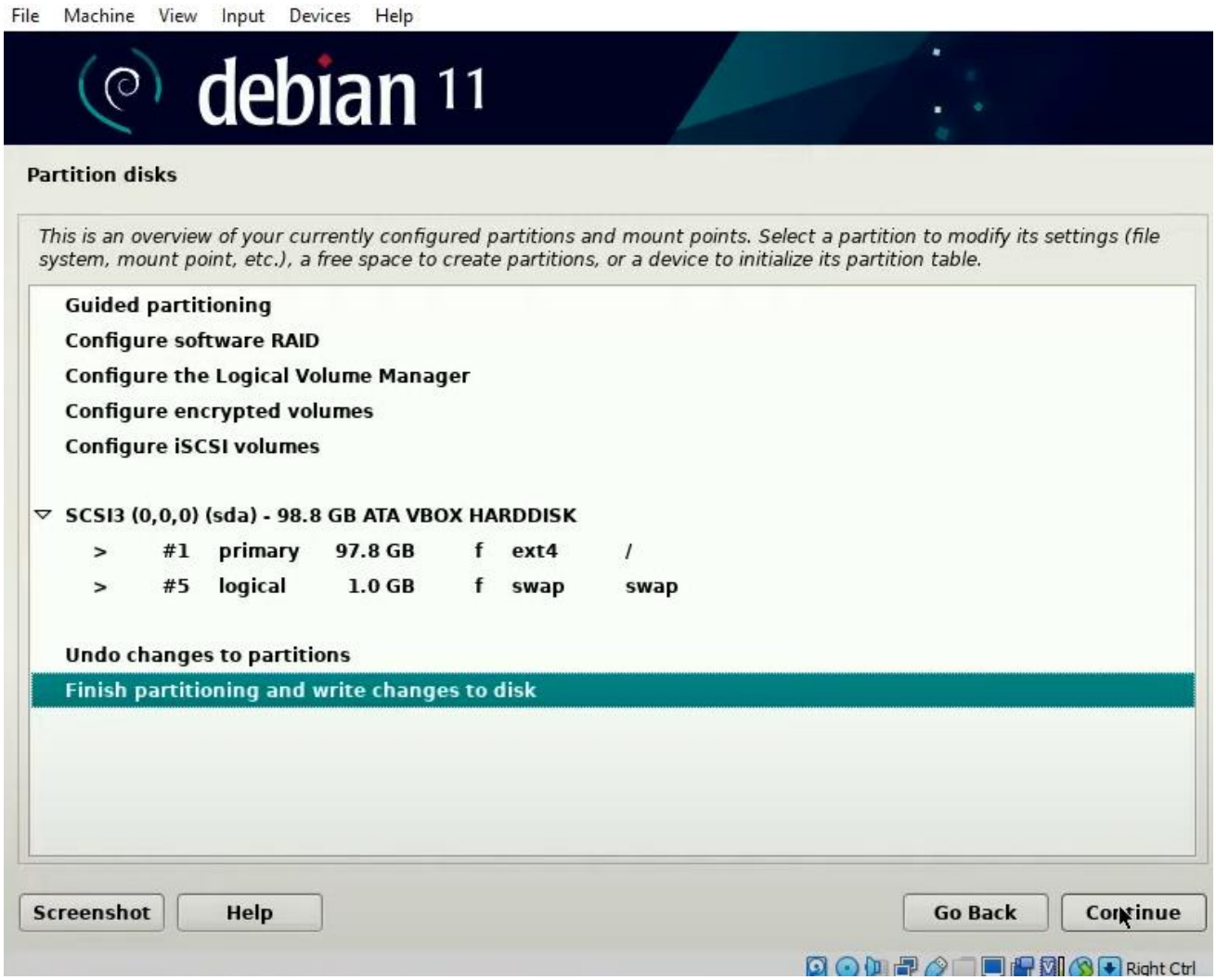


Рисунок 2.10 – Вибір диску для встановлення Raspberry Pi

Після завершення інсталяції віртуальна машина перезавантажується. Необхідно від'єднати ISO-образ із віртуального оптичного приводу, інакше система знову завантажиться у режимі Live. Після першого запуску встановленої системи відображається графічне середовище PIXEL, де можна завершити початкові налаштування. На цьому етапі доцільно перевірити мережеве підключення, увімкнути SSH для віддаленого адміністрування та оновити систему

до останньої версії за допомогою стандартної команди sudo apt update && sudo apt upgrade.

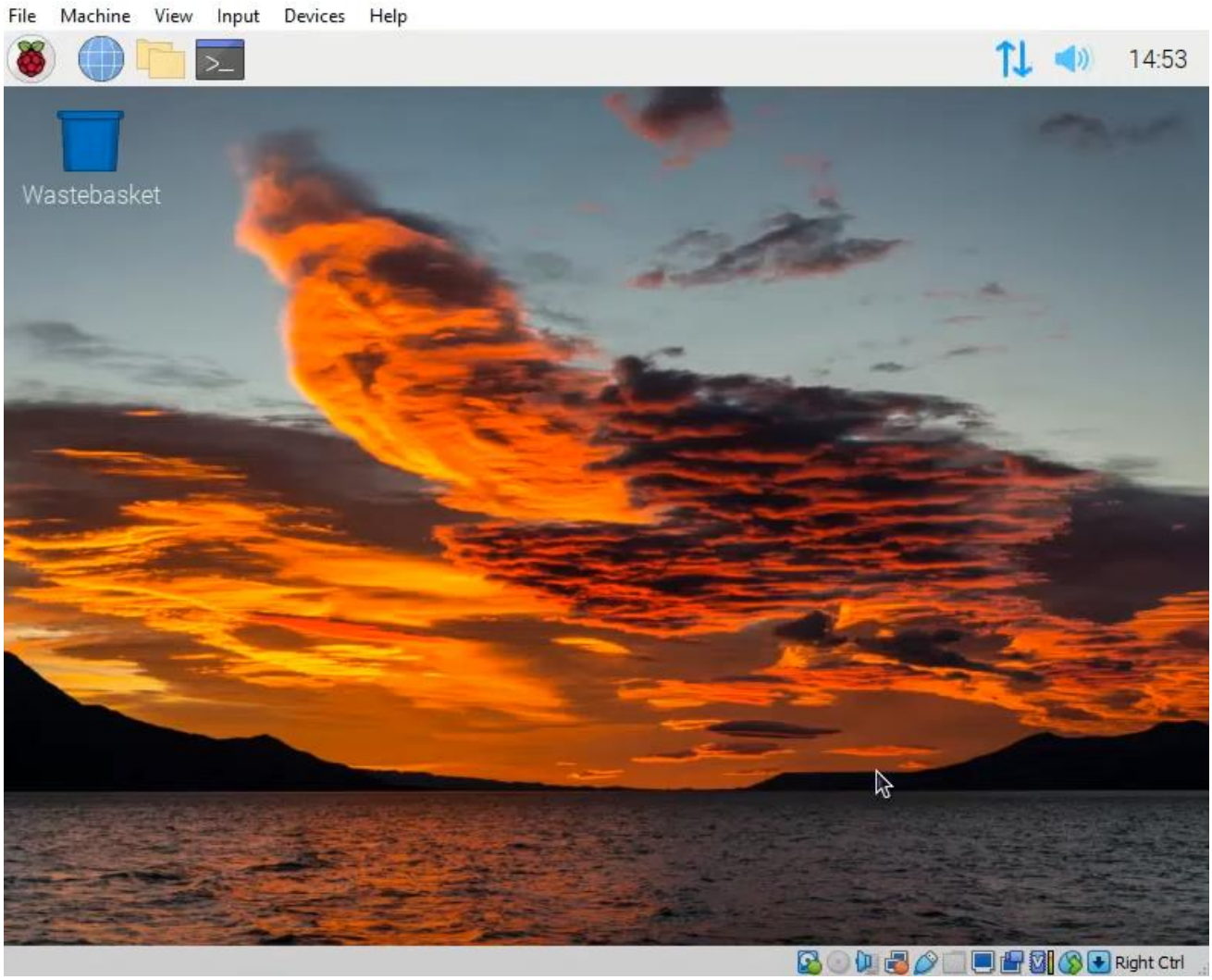


Рисунок 2.11 – Робочий стіл Raspberry Pi OS у VirtualBox

На завершення, встановлюються додаткові інструменти, необхідні для розгортання VPN-сервера: curl, git, ufw, net-tools та інші. Це забезпечує повну готовність системи до подальшого налаштування серверного ПЗ, включно з PiVPN. Таким чином, Raspberry Pi OS у віртуальному середовищі не лише повторює роботу реального Raspberry Pi, а й дозволяє безпечно тестувати всі функції, пов’язані з мережею, без ризику для основної операційної системи.

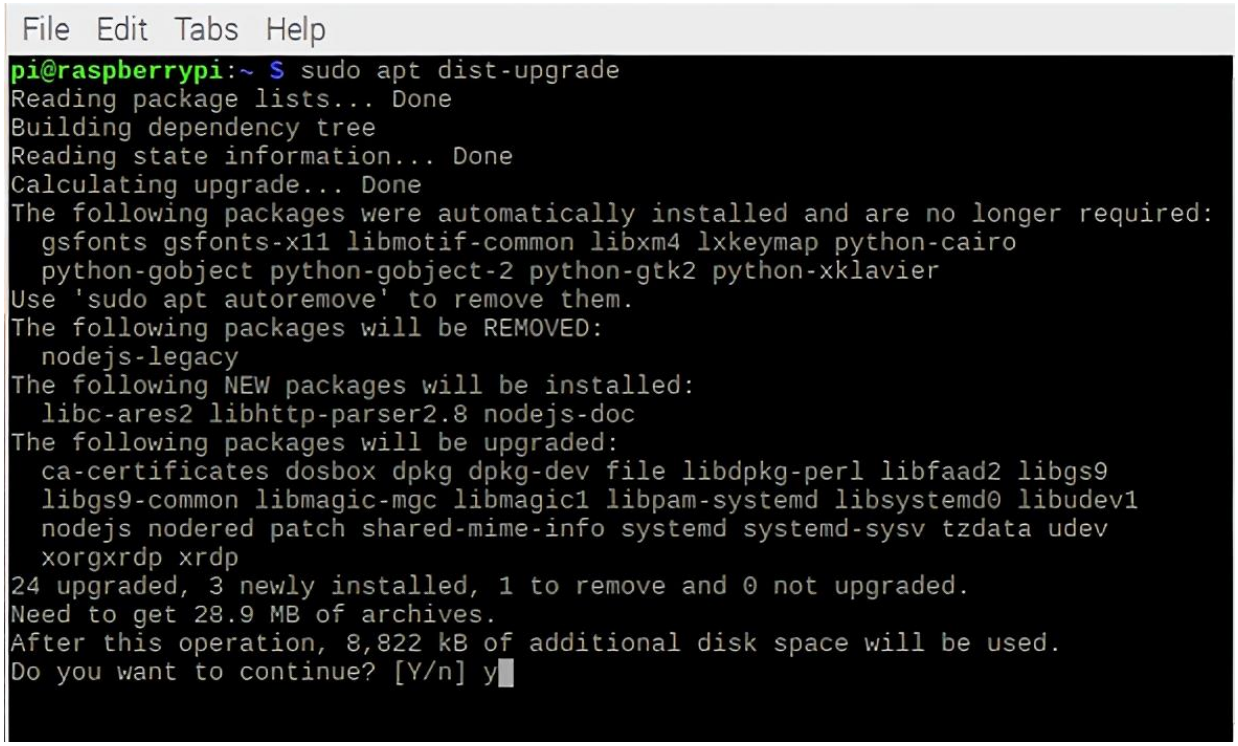


Рисунок 2.12 – Встановлення оновлень і базових утиліт у терміналі

Готовність віртуальної машини до подальшого налаштування VPN-сервера відкриває можливість перейти до наступного етапу – встановлення PiVPN та створення безпечного з’єднання.

2.4 Створення та налаштування віртуальної машини

Процес створення та налаштування віртуальної машини для емуляції Raspberry Pi у VirtualBox є ключовим етапом, що забезпечує успішне встановлення та функціонування операційної системи Raspberry Pi OS у середовищі хост-комп’ютера. Щоб створити віртуальну машину, необхідно запустити VirtualBox і скористатися майстром створення нової віртуальної машини. У полі назви доцільно вказати впізнаване ім’я, наприклад "RaspberryPi-VM", а як тип операційної системи обрати Linux. Оскільки Raspberry Pi OS базується на Debian, версію системи слід встановити як Debian (64-bit), якщо

образ, що використовується, також є 64-розрядним. Це важливо для забезпечення сумісності системних бібліотек та драйверів.

Після вибору типу системи вказується обсяг оперативної пам’яті, що буде виділений для віртуальної машини. У більшості випадків для комфортної роботи Raspberry Pi OS достатньо 2048 мегабайтів ОЗП, але при використанні графічного середовища бажано збільшити цей обсяг до 3072 мегабайтів, якщо ресурси хост-комп’ютера це дозволяють. Це забезпечить стабільну роботу системи, швидке завантаження програм і уникнення збоїв при запуску більш важких компонентів, зокрема при встановленні VPN-серверу чи графічних утиліт адміністрування.

Наступним кроком є створення віртуального жорсткого диска. Зазвичай обирається формат VDI (VirtualBox Disk Image), оскільки він найкраще підтримується VirtualBox і дозволяє гнучко керувати дисковим простором. Вибір типу збереження простору – динамічний чи фіксований – залежить від переваг користувача. Динамічний диск займає лише стільки місця, скільки фактично використовується, тоді як фіксований одразу резервує повний обсяг на фізичному диску. Для проекту із встановленням Raspberry Pi OS рекомендовано створити диск розміром щонайменше 20 гігабайтів, що забезпечить простір для оновлень, встановлення програмного забезпечення та роботи VPN.

Після створення базової структури віртуальної машини необхідно перейти до детального налаштування її параметрів. У розділі "Система" бажано вимкнути флопі-диск, налаштувати порядок завантаження, поставивши оптичний диск на перше місце, якщо буде використовуватися ISO-образ для інсталяції. Також важливо увімкнути апаратну віртуалізацію (VT-x/AMD-V), що забезпечує повну сумісність ядра Linux з віртуалізованим середовищем. У розділі "Процесор" можна додати друге ядро процесора, що покращить продуктивність, особливо при виконанні одночасних мережевих операцій або шифруванні трафіку у VPN-сервері.

У секції "Дисплей" рекомендується збільшити відеопам’ять до максимуму (наприклад, 128 МБ), а також увімкнути 3D-акселерацію, якщо графічне

середовище буде використовуватись активно. Це дозволить системі працювати плавніше, уникати візуальних затримок, і значно покращить загальний досвід взаємодії з віртуальною Raspberry Pi OS. Для налаштування накопичувача у віртуальному приводі CD/DVD потрібно вказати шлях до ISO-образу Raspberry Pi OS, який завантажено раніше. Це дозволяє системі завантажитися з цього образу для подальшої інсталяції.

Налаштування мережі також має важливе значення, оскільки VPN-сервер буде працювати з трафіком, що проходить через віртуальний інтерфейс. За замовчуванням мережевий адаптер встановлюється у режим NAT, однак для зручності тестування можна використати режим «мережевий міст» (Bridge Adapter), який надає віртуальній машині повний доступ до мережі хост-комп'ютера. Це дозволяє адресації в локальній мережі, відкриттю портів для зовнішнього доступу і реалістичному тестуванню роботи VPN.

Після збереження всіх налаштувань віртуальна машина готова до запуску. При першому старті система з ISO-образу запускає інсталяційне середовище, яке дозволяє інсталювати Raspberry Pi OS на створений віртуальний диск. Це відкриває наступний етап – безпосереднє встановлення та підготовку операційної системи до подальшої роботи у проєкті VPN-сервера.

2.5 Тестування працездатності базового середовища

Тестування працездатності базового середовища після встановлення та налаштування віртуальної машини з Raspberry Pi OS у VirtualBox є критично важливим етапом, що дозволяє впевнитися в готовності системи до подальшої інсталяції мережевого програмного забезпечення, зокрема PiVPN. Мета цього етапу – перевірити, чи всі компоненти системи функціонують належним чином, чи забезпечена стабільність віртуального середовища, чи доступна мережева взаємодія, а також чи відповідає система мінімальним технічним вимогам для запуску VPN-сервера. Передусім потрібно переконатися, що операційна система

запускається без помилок, завантажується до графічного або консольного інтерфейсу без зависань, аварійного завершення чи інших критичних повідомлень. Це свідчить про коректну установку образу Raspberry Pi OS, сумісність конфігурації VirtualBox та належну підтримку апаратної віртуалізації на хост-комп'ютері.

Одним з основних завдань є перевірка коректності підключення до мережі. Для цього використовується стандартна команда ping, наприклад, до зовнішнього ресурсу на кшталт ping google.com. Успішне отримання відповіді свідчить про активне підключення до Інтернету, що необхідне для подальшого встановлення оновлень, завантаження пакетів та роботи PiVPN. У разі відсутності відповіді потрібно перевірити налаштування мережевого адаптера у VirtualBox, переконатися, що драйвери працюють належним чином, а брандмауери хост-системи не блокують з'єднання. Окрім перевірки інтернет-з'єднання, доцільно протестувати локальну адресу IP, яка надана віртуальній машині, використовуючи команду ip a. Це дозволяє ідентифікувати IP-адресу, яка згодом буде використана для підключення до VPN-сервера з клієнтських пристроїв.

Наступним кроком є оновлення всієї системи до актуального стану, що виконується за допомогою команд sudo apt update та sudo apt upgrade. Під час цього процесу система звертається до офіційних репозиторіїв Debian, перевіряє наявність новіших версій пакетів та встановлює їх. Це особливо важливо для забезпечення сумісності ядра та бібліотек із програмним забезпеченням VPN, а також для усунення потенційних вразливостей у системі безпеки. Успішне оновлення системи є індикатором її стабільної роботи та правильної взаємодії з мережевими службами.

Крім того, потрібно перевірити наявність і функціональність базових утиліт, таких як curl, git, net-tools, які є необхідними для подальшого встановлення PiVPN. Команди на зразок curl --version або git --version дозволяють переконатися, що потрібні інструменти встановлені та працездатні. Якщо деякі пакети відсутні, їх слід інсталювати за допомогою пакетного менеджера apt. Важливо також

перевірити, чи працює служба SSH, що дозволяє віддалений доступ до віртуальної машини. Це зручно не лише для адміністрування, але й для передачі конфігураційних файлів VPN та автоматизації процесів керування сервером. У випадку, якщо SSH неактивний, його можна увімкнути через sudo systemctl enable ssh та sudo systemctl start ssh.

Ще одним аспектом тестування є оцінка завантаженості системи. За допомогою команди top або htop можна переглянути поточне використання процесора, оперативної пам’яті та інших ресурсів. Якщо система перевантажена навіть у стані простою, можливо, доведеться змінити налаштування віртуальної машини – наприклад, збільшити обсяг оперативної пам’яті чи кількість ядер процесора. Також корисно перевірити файлову систему та доступний дисковий простір за допомогою df -h, щоб переконатися, що місця вистачить для встановлення VPN та збереження логів, ключів і конфігурацій.

Загалом тестування базового середовища дозволяє переконатися, що всі технічні аспекти віртуалізації, мережевого підключення, системної стабільності та доступності необхідних компонентів відповідають вимогам для подальшого встановлення програмного VPN-рішення. Це значно знижує ризики збоїв або помилок у наступних етапах реалізації проєкту, а також дає впевненість у правильності обраної архітектури та середовища розробки.

2.6 Висновки

Після детального аналізу та реалізації середовища для розгортання персонального VPN-сервера на базі емуляції Raspberry Pi у VirtualBox можна зробити низку висновків, що підкреслюють важливість технічної підготовки та вибору інструментів на початковому етапі. Другий розділ показав, що створення повноцінного віртуального середовища, яке імітує роботу фізичного Raspberry Pi, є цілком досяжним завданням навіть у рамках обмеженого обладнання. Завдяки використанню VirtualBox як універсального і безкоштовного гіпервізора, стало

можливим швидко розгорнути тестове середовище, що повністю відтворює архітектуру та функціональність реального пристрою. Це, своєю чергою, дозволило уникнути необхідності фізичного доступу до Raspberry Pi на етапі розробки та налаштування, що є критично важливим у навчальних або дослідницьких проєктах, де бюджет або логістика можуть бути обмеженими.

Процес інсталяції та налаштування VirtualBox виявився відносно нескладним, проте вимагав уваги до деталей, зокрема в налаштуваннях мережевого адаптера, режимів віртуалізації та ресурсів віртуальної машини. Правильна конфігурація дозволила забезпечити стабільну роботу системи, гарантований доступ до Інтернету та підтримку SSH, що в сукупності створило повноцінне середовище для подальшої роботи з VPN. Завантаження та підготовка Raspberry Pi OS продемонстрували гнучкість і зручність у роботі з дистрибутивами Linux, що використовуються в системах з обмеженими ресурсами, а також довели доцільність використання саме Raspberry Pi OS як базової операційної системи для реалізації VPN-сервера.

Особливо важливою була перевірка працездатності віртуальної машини після її налаштування. Це дозволило на ранньому етапі виявити можливі проблеми з мережею, доступом до оновлень, драйверами або сумісністю програмного забезпечення, що могло би ускладнити або зробити неможливою подальшу інсталяцію VPN-сервера. Проведене тестування підтвердило, що середовище є достатньо надійним для запуску складніших сервісів, які вимагають мережевої взаємодії, роботи з криптографією та віддаленого адміністрування. Також було оцінено навантаження на ресурси хост-системи, що дало можливість оптимізувати використання пам’яті, процесора та дискового простору.

У результаті проведеної роботи було створено повноцінну віртуальну інфраструктуру, що є технічно підготовленою до встановлення та запуску VPN-сервера з використанням PiVPN. Досягнута сумісність усіх компонентів підтверджує ефективність вибраного підходу до реалізації проєкту.

3 ВСТАНОВЛЕННЯ ТА НАЛАШТУВАННЯ VPN-СЕРВЕРА НА БАЗІ PIVPN

3.1 Встановлення необхідного програмного забезпечення

PiVPN – це інструмент з відкритим кодом, який спрощує встановлення та налаштування VPN-серверів на пристроях під управлінням Linux, зокрема на Raspberry Pi. Основна мета проєкту полягає у тому, щоби надати користувачеві зручний спосіб швидко та безпечно розгорнути VPN-рішення вдома або в невеликій корпоративній мережі без необхідності глибоких технічних знань. PiVPN підтримує два основних протоколи VPN — OpenVPN і WireGuard, що робить його універсальним і адаптованим до потреб різних користувачів. Його використання знижує бар'єр входу до теми VPN, оскільки автоматизує велику кількість рутинних процесів, які в іншому випадку вимагали б ручного редагування конфігураційних файлів, керування ключами шифрування, налаштування фаєрволу та мережевої маршрутизації.

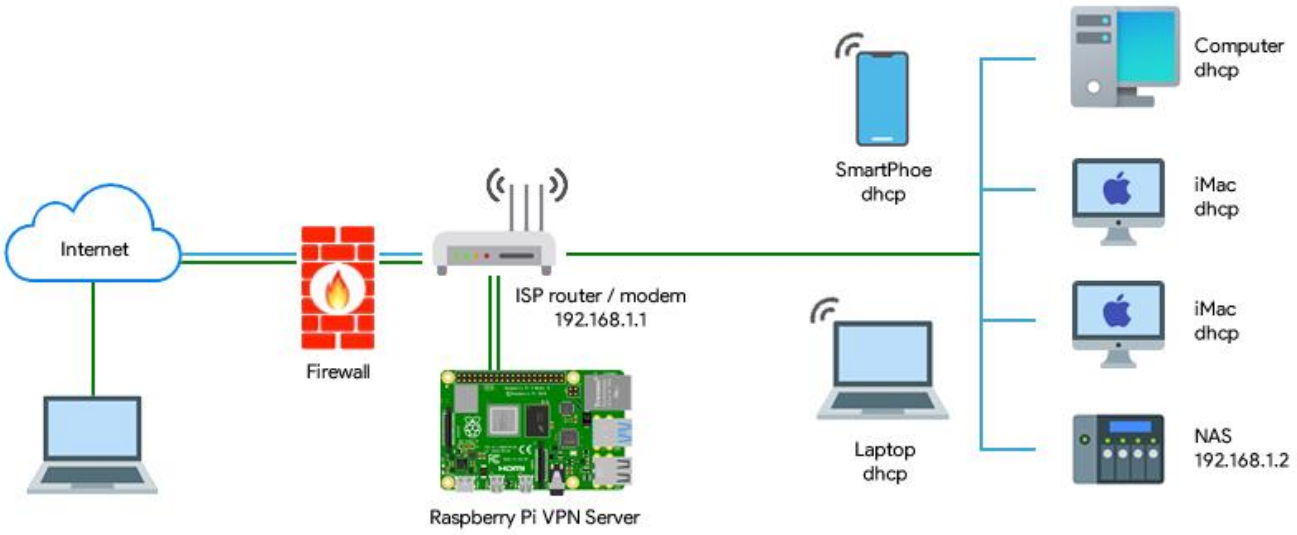


Рисунок 3.1 – Архітектура ПЗ проєкту

Перевага PiVPN полягає в його простому механізмі встановлення. Інсталяція виконується однією командою через термінал, використовуючи curl для завантаження скрипта з офіційного сайту. Після запуску відкривається текстовий інтерфейс, у якому користувач покроково вибирає налаштування майбутнього сервера: вибір VPN-протоколу, порту для з'єднання, типу IP-адресації клієнтів, інтеграція з DNS, налаштування фаєрволу та багато іншого. PiVPN також автоматично вмикає IP forwarding — важливу функцію ядра Linux, яка дозволяє пересилати пакети між інтерфейсами, що є критично важливим для роботи VPN.

Однією з головних функцій PiVPN є автоматична генерація криптографічних ключів, як для сервера, так і для кожного клієнта. Це усуває потребу у складній ручній процедурі створення ключів та сертифікатів, що особливо актуально для новачків. У разі використання OpenVPN, система створює файли .ovpn, а при виборі WireGuard – .conf, які містять усі необхідні параметри для підключення до сервера. Ці файли можна безпечно переносити на клієнтські пристрої або зчитувати через згенерований QR-код, що особливо зручно при використанні смартфонів.

Крім того, PiVPN забезпечує прості засоби керування створеними конфігураціями. За допомогою команд на кшталт pivpn add, pivpn remove, pivpn list або pivpn -qr, адміністратор може створювати нові профілі, видаляти старі, переглядати список активних користувачів або швидко виводити QR-коди для клієнтів. Це значно спрощує управління VPN-сервером у порівнянні з традиційним ручним редагуванням конфігураційних файлів. Також можливо керувати журналами з’єднань і статусом сервісів, наприклад через pivpn debug, що полегшує діагностику проблем.

Завдяки легкій інтеграції з UFW або iptables, PiVPN автоматично створює правила для обмеження доступу до системи лише через необхідні порти. Це підвищує загальний рівень безпеки і дозволяє уникнути небажаних вторгнень. За

потреби, користувач може вручну доповнювати або змінювати ці правила, але для більшості випадків PiVPN надає базову, але надійну конфігурацію фаєрволу «з коробки».

Варто також зазначити, що PiVPN постійно підтримується спільнотою і розробниками. Нові версії адаптуються під зміни в OpenVPN та WireGuard, додаються функції, виправляються баги, і оновлюється документація. Це робить проєкт надійним вибором для тих, хто хоче стабільну і захищену VPN-систему без складної ручної настройки. Попри простоту, PiVPN дозволяє гнучко масштабувати систему: з локального домашнього сервера до невеликої офісної мережі, забезпечуючи конфіденційність трафіку, захист у публічних мережах і безпечний віддалений доступ до внутрішніх ресурсів.

Таким чином, PiVPN – це потужний інструмент, який поєднує в собі простоту використання з глибокими можливостями налаштування, забезпечуючи зручну платформу для розгортання персонального VPN-сервера навіть на малопотужних пристроях на кшталт Raspberry Pi.

На етапі налаштування персонального VPN-сервера на базі Raspberry Pi, навіть у віртуальному середовищі, ключову роль відіграє правильне встановлення необхідного програмного забезпечення. Цей процес починається з оновлення системи, що забезпечує стабільність та безпеку. Команди sudo apt update і sudo apt upgrade дозволяють завантажити актуальні пакети з репозиторіїв Raspberry Pi OS, що гарантує сумісність та знижує ризик виникнення конфліктів під час подальшої інсталяції. Після оновлення важливо переконатися, що система підтримує базові утиліти, необхідні для встановлення PiVPN та управління мережею. До них належать такі пакети, як curl, git, ufw, net-tools, resolvconf, які забезпечують взаємодію з інтернет-ресурсами, керування репозиторіями, настройку фаєрволу, перегляд мережевих інтерфейсів і правильне оновлення DNS-налаштувань.

Далі встановлюється сам PiVPN – зручний інструмент із відкритим кодом, що дозволяє автоматизувати встановлення VPN-сервера на основі OpenVPN або WireGuard. Його перевага полягає у простоті, адже інсталяція запускається однією

командою curl -L https://install.pivpn.io | bash, після чого користувач поетапно проходить через текстовий інтерфейс конфігуратора, де обирає протокол, порт, тип інтерфейсу, спосіб маршрутизації та налаштування файрволу. На цьому етапі особливо важливо, щоби в системі був доступний інтерфейс tun, який відповідає за тунелювання мережевого трафіку, а також активована підтримка IP forwarding у файлі /etc/sysctl.conf. Без цього VPN-з'єднання не зможе функціонувати коректно.

Крім того, важливо забезпечити наявність ключових криптографічних бібліотек, які потрібні для генерації ключів шифрування та безпечного обміну даними. У випадку WireGuard це модулі ядра та утиліта wg, яка забезпечує керування інтерфейсом тунелю. Якщо було обрано OpenVPN, система повинна мати встановлений пакет openvpn та відповідні бібліотеки TLS. Усі ці компоненти або встановлюються автоматично в рамках скрипта PiVPN, або, у разі помилок, потребують ручного доустановлення через APT.

На завершення налаштовується міжмережевий екран UFW, який має дозволити лише необхідні порти: порт VPN-протоколу, порт SSH для віддаленого керування та, при потребі, інші специфічні порти. Також бажано провести аудит активних служб і процесів, щоби переконатися, що в системі не працюють зайві або потенційно вразливі сервіси. Повне встановлення всіх компонентів завершується перезавантаженням системи, після чого можна переходити до створення конфігураційних файлів клієнтів і тестування підключення. Таким чином, встановлення програмного забезпечення – це не просто запуск скрипта, а комплексний процес підготовки системного середовища, що потребує розуміння мережевих, криптографічних і системних аспектів.

3.2 Інсталяція та конфігурація PiVPN

Процес інсталяції та конфігурації PiVPN на базі Raspberry Pi у віртуальному середовищі VirtualBox є ключовим етапом у побудові персонального VPN-

сервера. Після того як операційна система Raspberry Pi OS успішно встановлена і оновлена, можна переходити безпосередньо до встановлення PiVPN. Для цього першочергово потрібно переконатися, що система має доступ до Інтернету та містить усі необхідні інструменти, такі як curl, git, sudo та оновлену систему пакетів. Основна команда для запуску встановлення – це завантаження офіційного інсталяційного скрипта за допомогою команди curl -L https://install.pivpn.io | bash, яка ініціює покроковий майстер встановлення в текстовому інтерфейсі.

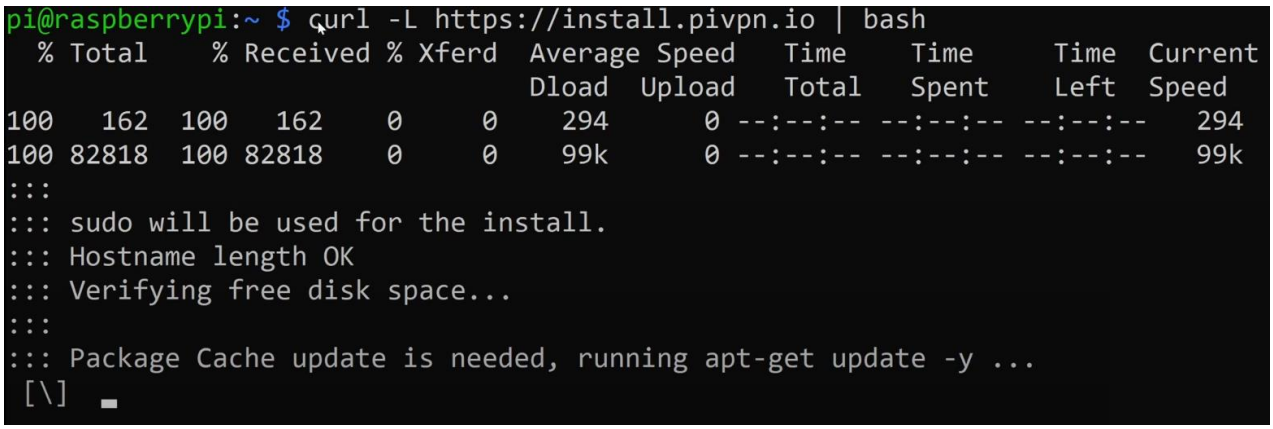


Рисунок 3.2 – Запуск інсталяційної команди в терміналі

На етапі встановлення користувач повинен вибрати VPN-протокол. Найпоширенішими є OpenVPN і WireGuard. Обидва варіанти підтримуються PiVPN, однак WireGuard часто рекомендується завдяки вищій продуктивності, меншому споживанню ресурсів і сучаснішому підходу до безпеки. Після вибору протоколу система запропонує налаштувати порт, який буде використовуватись для вхідних з'єднань, та згенерує криптографічні ключі для безпечної передачі даних. Далі активується IP forwarding, що дозволяє системі пересилати пакети між інтерфейсами. Інсталяція також пропонує автоматичну інтеграцію з фаєрволом UFW, який дозволяє відкрити лише ті порти, що необхідні для роботи VPN, що значно підвищує рівень безпеки сервера.

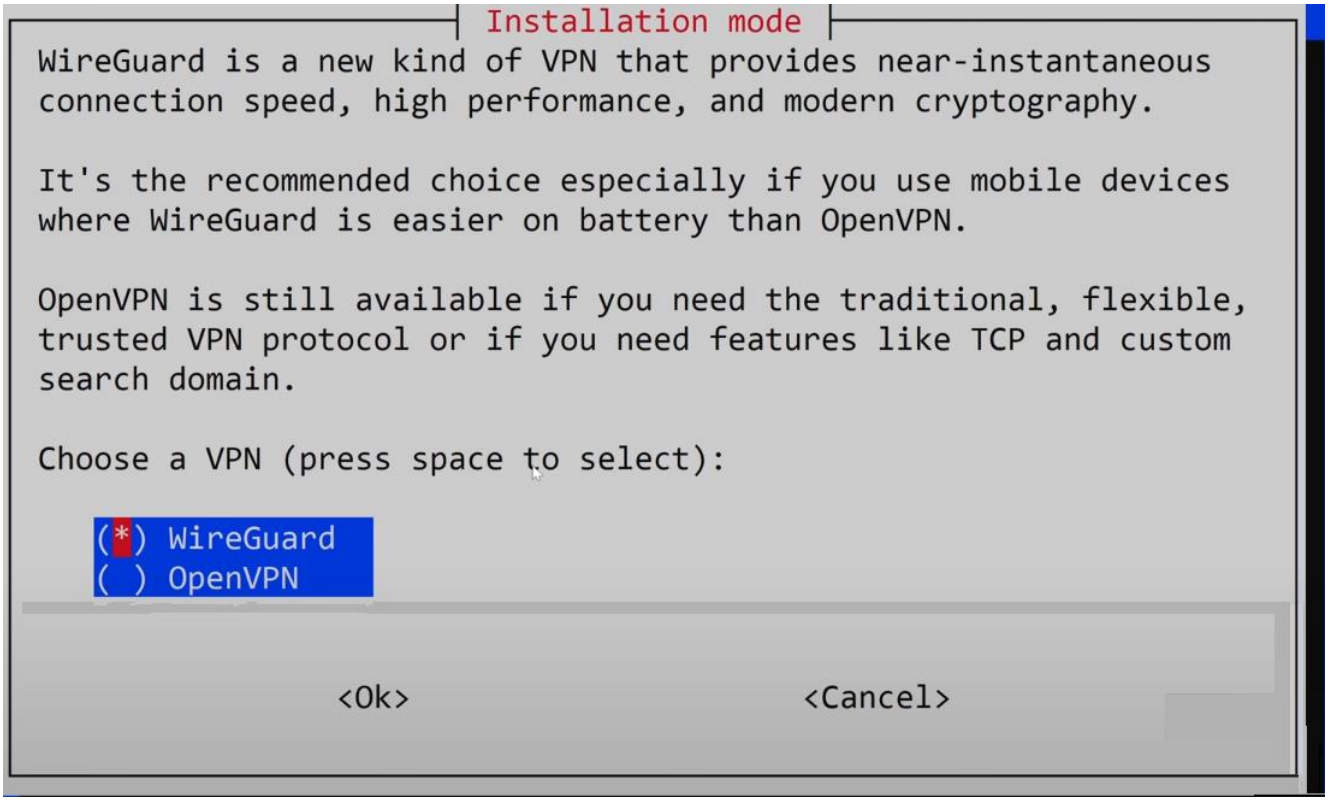


Рисунок 3.3 – Місце для скріншота: вибір протоколу VPN

Після завершення інсталяції наступним кроком є створення конфігураційних файлів для клієнтів. Це здійснюється через просту команду pivpn add, яка ініціює процес генерації ключів для нового користувача, створення відповідного профілю та збереження його у вигляді конфігураційного файлу. Для WireGuard цей файл має розширення .conf, і його можна зчитати з мобільного пристрою за допомогою QR-коду. Для зручності сканування передбачено команду pivpn -qr, яка відображає код у терміналі, що дозволяє миттєво перенести конфігурацію на мобільний клієнт.

Згенерований конфігураційний файл містить усю необхідну інформацію для встановлення захищеного з'єднання: приватний і публічний ключі клієнта, публічний ключ сервера, IP-адресу VPN-сервера, порт для підключення, а також налаштування маршрутизації та DNS. Цей файл можна зберегти локально, передати через безпечний канал або перенести на інші пристрої, що потребують доступу до VPN. У разі втрати конфігурації або компрометації ключів

адміністратор може швидко створити новий профіль і відкликати старий, забезпечуючи гнучке управління клієнтами та високий рівень безпеки підключень.

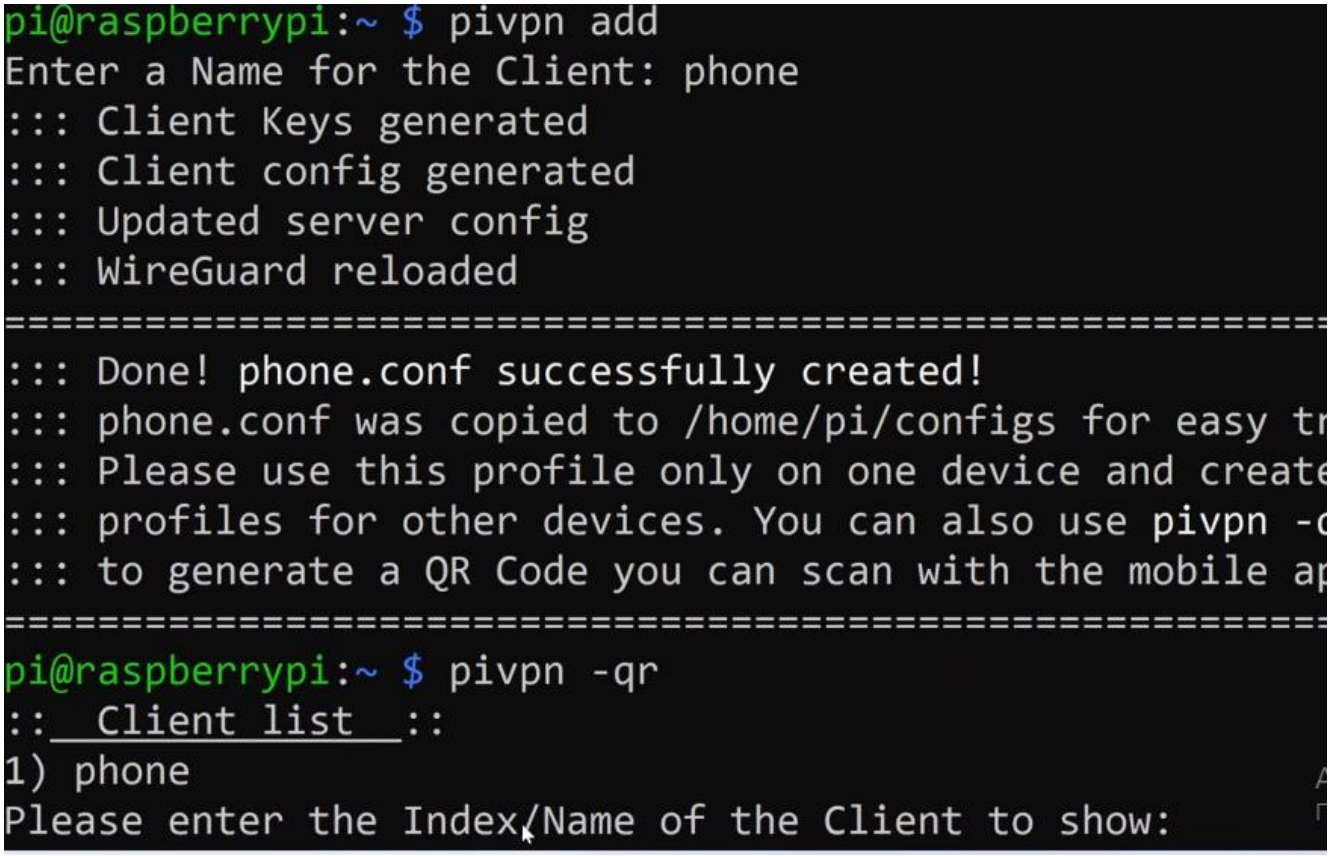


Рисунок 3.4 – Створення конфігураційного файлу клієнта через термінал

Сервер після інсталяції та створення хоча б одного користувача готовий до використання. Перевірка його роботи здійснюється через команду sudo systemctl status wg-quick@wg0 у випадку використання WireGuard, де можна побачити статус підключення, активність сервісу та можливі помилки. У разі правильного налаштування клієнт може встановити з'єднання, змінити свою IP-адресу на публічну адресу сервера і отримати захищений канал зв’язку до Інтернету або до внутрішньої мережі.

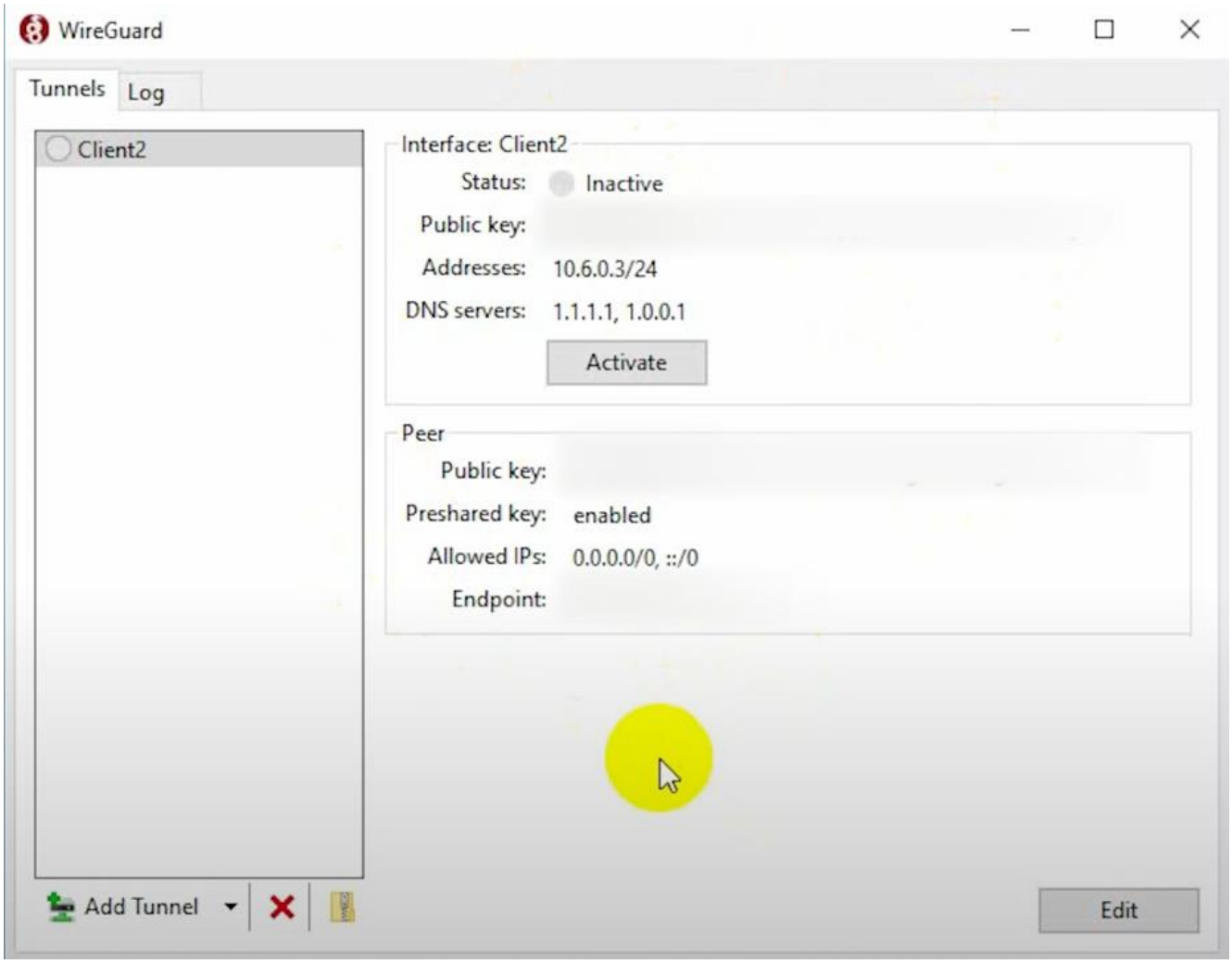


Рисунок 3.5 – Статус запущеної служби WireGuard у системі

Весь процес, попри значну кількість технічних кроків, є достатньо зручним навіть для початківців, завдяки інтерактивному інтерфейсу скрипта PiVPN. Він автоматизує більшість критичних аспектів налаштування VPN, водночас залишаючи можливість гнучкого ручного редагування у випадку потреби досвідчених користувачів. Завдяки цьому PiVPN дозволяє швидко розгорнути ефективний, надійний і безпечний VPN-сервер із використанням сучасного протоколу шифрування та простим керуванням клієнтами.

3.3 Налаштування WireGuard як основного VPN-протоколу

WireGuard є сучасним, високопродуктивним і безпечним VPN-протоколом, який набуває все більшої популярності завдяки своїй ефективності, мінімалізму в коді та простоті конфігурації. Його інтеграція в систему через PiVPN робить налаштування доступним навіть для користувачів, які не мають глибокого технічного досвіду, але при цьому забезпечує високий рівень безпеки та швидкості. Після завершення інсталяції PiVPN та вибору WireGuard як основного протоколу, система автоматично генерує всі необхідні криптографічні ключі — як для сервера, так і для клієнтів, використовуючи сучасні алгоритми на основі криптографії кривих Едвардса, зокрема Curve25519. Це дозволяє реалізувати надзвичайно ефективне асиметричне шифрування з низьким навантаженням на систему.

Налаштування WireGuard у рамках PiVPN передбачає автоматичне створення основного інтерфейсу, який зазвичай має назву wg0. Цей інтерфейс прописується в окремому конфігураційному файлі, де вказано IP-адресу, яку сервер буде використовувати у VPN-мережі, порт прослуховування, приватний ключ сервера та список дозволених клієнтів. Також у файлі конфігурації вказуються налаштування щодо маршрутизації, наприклад, чи буде трафік клієнта перенаправлятися через сервер повністю, чи лише для доступу до локальної мережі. Окрім цього, серверна конфігурація містить публічні ключі клієнтів і дозволені IP-адреси, з яких ці клієнти можуть з'єднуватися.

На клієнтській стороні для підключення до сервера використовується окремий файл конфігурації, який містить публічний ключ сервера, IP-адресу або доменне ім’я, через яке здійснюється підключення, а також порт, який було обрано під час встановлення. У цьому ж файлі зберігаються особисті ключі клієнта, IP-адреса у VPN-середовищі та правила маршрутизації. Особливістю WireGuard є його концепція peer-to-peer, де обидві сторони, клієнт і сервер,

виступають як рівноправні вузли, що дозволяє організовувати як прості конфігурації "клієнт-сервер", так і більш складні сітчасті VPN-мережі.

WireGuard не використовує традиційного механізму авторизації за допомогою логіна і пароля, покладаючись виключно на криптографічні ключі, що суттєво зменшує ризик перехоплення облікових даних і одночасно спрощує управління з’єднаннями. Щоб підключити новий пристрій до мережі, достатньо згенерувати нову пару ключів, додати публічний ключ до серверного конфігураційного файлу та створити відповідний клієнтський конфігураційний файл.

Після завершення налаштування конфігураційні файли зберігаються в каталозі /etc/wireguard/, а запуск і зупинка VPN-сервісу виконується за допомогою системного менеджера systemd, зокрема команд sudo systemctl start wg-quick@wg0 та sudo systemctl stop wg-quick@wg0. Для автоматичного запуску при старті системи використовується команда sudo systemctl enable wg-quick@wg0. Статус підключення та активність клієнтів можна контролювати через команду sudo wg, яка виводить список peer'ів, їх IP-адреси, час останнього обміну пакетами та статистику переданих даних. Це дозволяє адміністратору ефективно моніторити та керувати з’єднаннями без використання додаткових утиліт або інтерфейсів.

WireGuard демонструє високу продуктивність навіть на слабких пристроях, таких як Raspberry Pi, завдяки малій кількості коду та використанню оптимізованих алгоритмів шифрування. Завдяки цьому він забезпечує швидке з'єднання, низьку затримку та стабільну передачу даних без перевантаження системи. Простота його конфігурації в поєднанні з високою безпекою робить WireGuard ідеальним вибором для побудови персонального VPN-сервера, особливо у випадках, коли важлива швидка передача даних і захист приватності користувача.

WireGuard також має значну перевагу з точки зору коду й архітектури. На відміну від багатьох традиційних VPN-протоколів, які містять тисячі рядків коду і мають складну логіку, WireGuard реалізований у вигляді мінімалістичного

модулю ядра Linux, що містить лише кілька тисяч рядків. Такий підхід зменшує кількість потенційних вразливостей у безпеці, спрощує аудит коду й пришвидшує впровадження оновлень. Цей модуль інтегрований безпосередньо у сучасні ядра Linux, починаючи з версії 5.6, що означає підтримку на рівні системи без необхідності встановлення додаткових сторонніх компонентів. Завдяки цьому зменшується ризик конфліктів сумісності та підвищується стабільність роботи VPN-з'єднання.

З технічної точки зору, WireGuard використовує сучасні криптографічні примітиви, включаючи ChaCha20 для шифрування, Poly1305 для автентифікації повідомлень, Curve25519 для обміну ключами, а також BLAKE2s як хеш-функцію. Ці алгоритми не лише забезпечують високий рівень безпеки, а й працюють значно швидше на менш потужних процесорах, що особливо важливо у випадку використання на Raspberry Pi або віртуальних машинах. Усі ці компоненти підібрані так, щоб мінімізувати споживання ресурсів та оптимізувати швидкодію мережі.

Однією з ключових особливостей WireGuard є його статичний підхід до мережевої конфігурації. На відміну від протоколів, які динамічно встановлюють тунелі, WireGuard покладається на попередньо визначені правила маршрутизації, в яких однозначно фіксується, який IP-діапазон або маршрут доступний через певного peer'a. Такий підхід забезпечує чіткість і прогнозованість маршрутизації, що особливо важливо при налаштуванні складних мережевих топологій або інтеграції з внутрішніми сервісами.

WireGuard також має кросплатформену підтримку — окрім Linux, він повноцінно працює на Windows, macOS, Android та iOS. Клієнтські програми мають зручний інтерфейс, часто дозволяють імпортувати конфігураційний файл через QR-код, і підтримують автоматичне підключення при запуску системи. Завдяки цьому користувачі можуть легко підключатися до персонального VPN-сервера з будь-якого пристрою незалежно від операційної системи.

У порівнянні з традиційними протоколами, такими як OpenVPN або IPSec, WireGuard значно виграє у швидкості з'єднання, простоті налаштування та масштабованості. Він не потребує запуску додаткових демонів, не вимагає динамічного налаштування TLS-сертифікатів чи складної PKI-інфраструктури. Усі з'єднання визначаються лише парами публічних і приватних ключів, що забезпечує більшу прозорість та контроль над усім процесом.

Насамкінець, слід зазначити, що популярність WireGuard стрімко зростає не лише серед ентузіастів, а й у комерційному середовищі. Багато провайдерів VPN-послуг вже пропонують його як основний або альтернативний протокол. Вбудована підтримка в ядро Linux, простота інтеграції, ефективність роботи на низькопотужному обладнанні та сучасна криптографія — усе це робить WireGuard найкращим вибором для побудови сучасної, безпечної та швидкої віртуальної приватної мережі на базі персонального сервера.

WireGuard також відзначається високою ефективністю у використанні системних ресурсів, що особливо важливо при розгортанні VPN-сервера на базі пристроїв з обмеженими апаратними можливостями, таких як Raspberry Pi. Завдяки компактному коду, який становить лише кілька тисяч рядків, WireGuard легше підтримувати, аналізувати на наявність вразливостей і інтегрувати у різні платформи, зокрема в ядрі Linux, де він реалізується як модуль. Це дозволяє забезпечити не лише високу продуктивність, але й підвищену безпеку на рівні операційної системи.Завдяки своїй архітектурі WireGuard використовує сучасні криптографічні алгоритми за замовчуванням, уникаючи складних конфігураційних опцій і застарілих або небезпечних методів шифрування. Зокрема, він застосовує алгоритми Curve25519 для обміну ключами, ChaCha20 для симетричного шифрування, Poly1305 для автентифікації, BLAKE2s як хеш-функцію та HKDF для генерації ключів. Такий набір засобів дозволяє забезпечити не лише високу криптостійкість, але й сумісність з сучасними стандартами безпеки, рекомендованими для використання в державних і корпоративних мережах.

Ще однією перевагою є швидкість встановлення з'єднання. На відміну від OpenVPN, який може потребувати кількох секунд для побудови тунелю, WireGuard здатний встановити з'єднання майже миттєво, що особливо корисно при мобільному використанні або при нестабільному з'єднанні. Крім того, WireGuard автоматично визначає зміни у мережі, наприклад, зміну IP-адреси клієнта, і підтримує з'єднання без необхідності ручного перезапуску тунелю, що значно спрощує використання.

WireGuard підтримує як статичні, так і динамічні IP-адреси, що дозволяє використовувати його в широкому спектрі мережевих сценаріїв — від фіксованих серверних інсталяцій до мобільних клієнтів. Завдяки простоті та відкритості протоколу існує багато клієнтських реалізацій під різні операційні системи, включаючи Windows, macOS, Android, iOS, Linux та навіть маршрутизатори з підтримкою OpenWRT або DD-WRT. Це забезпечує гнучкість у виборі пристроїв, які можуть підключатися до VPN-серверу.

Інтеграція WireGuard у PiVPN дозволяє спростити управління VPN-користувачами — адміністратор може легко додавати або видаляти клієнтів, створювати резервні копії конфігурацій, генерувати QR-коди для зручного імпорту на мобільні пристрої та здійснювати аудит з'єднань. Простий синтаксис конфігураційних файлів дозволяє легко налаштовувати нові параметри або адаптувати налаштування до конкретних потреб мережі. Наприклад, можна задати доступ лише до певних IP-адрес, обмежити пропускну здатність або інтегрувати WireGuard у складніші схеми маршрутизації, включаючи доступ до кількох внутрішніх підмереж.

Завдяки своїм характеристикам WireGuard ідеально підходить не лише для приватного використання, а й для корпоративного середовища, де потрібна масштабована, захищена та проста у супроводі VPN-інфраструктура. Його можна використовувати для побудови site-to-site тунелів між офісами, захищених віддалених доступів для співробітників, шифрування трафіку у відкритих

мережах або як компонент більш складних систем, таких як Zero Trust архітектура або внутрішні мікросервісні взаємодії.

Таким чином, впровадження WireGuard у рамках проєкту побудови персонального VPN-сервера є обґрунтованим вибором як з технічної, так і з безпекової точки зору. Його гнучкість, простота, криптографічна стійкість та продуктивність відкривають широкі можливості для захищеного доступу до мережевих ресурсів із будь-якої точки світу без необхідності у складному адмініструванні чи дорогому обладнанні.

3.4 Генерація конфігурацій клієнтів та підключення пристроїв

Генерація конфігурацій клієнтів у PiVPN є важливим етапом налаштування персонального VPN-сервера, адже саме ці конфігурації дозволяють клієнтським пристроям встановлювати захищене з’єднання з сервером. Після інсталяції та налаштування серверної частини WireGuard за допомогою PiVPN, система надає прості інструменти для створення конфігураційних файлів, які включають усю необхідну інформацію для клієнтів. Щоб додати нового користувача, адміністратор запускає відповідну команду, яка ініціює процес створення ключів для клієнта – генеруються приватний і публічний ключі, а також відповідний профіль конфігурації у форматі .conf, що містить всі параметри підключення до VPN-сервера. Цей файл включає адресу сервера, порт, публічний ключ сервера, IP-адресу клієнта у VPN-мережі, DNS-сервер для обслуговування запитів та інші необхідні параметри. Конфігурація генерується автоматично з урахуванням всіх встановлених під час інсталяції налаштувань, тому користувачеві не потрібно вносити зміни вручну.

Після створення клієнтської конфігурації її можна передати на пристрій користувача декількома способами. Найбільш зручний – це використання QR-коду, який можна згенерувати за допомогою спеціальної команди. Це дозволяє мобільним пристроям з додатком WireGuard просто відсканувати код з екрана і

миттєво імпортувати конфігурацію. Для настільних операційних систем, таких як Windows, macOS або Linux, конфігураційний файл можна скопіювати вручну або передати через безпечний канал, наприклад SSH, USB-носій або зашифрований email. Основне завдання на цьому етапі — гарантувати, що приватні ключі не потраплять до сторонніх осіб, тому передача файлів повинна відбуватись з дотриманням усіх норм безпеки.

Після імпорту конфігурації на клієнтський пристрій, користувач може одразу підключитись до VPN. Під час підключення WireGuard автоматично здійснює обмін ключами з сервером, встановлює тунель і починає шифрувати весь мережевий трафік відповідно до вказаних правил. Для перевірки працездатності з'єднання зазвичай використовують засоби перевірки IP-адреси або прості мережеві утиліти, як-от ping або traceroute. Якщо з'єднання успішне, публічна IP-адреса клієнта змінюється на IP-адресу VPN-сервера, що свідчить про те, що весь трафік перенаправляється через захищений тунель. Завдяки властивостям WireGuard, з'єднання встановлюється дуже швидко і підтримується стабільно, навіть у разі зміни мережевого середовища на стороні клієнта.

Крім базового підключення, адміністратор може керувати клієнтськими конфігураціями, наприклад, відкликати доступ до VPN, видалити або змінити існуючі параметри. Це можна зробити за допомогою CLI-інтерфейсу PiVPN, який дозволяє переглядати список активних клієнтів, журнал підключень, а також інформацію про стан тунелю. Завдяки простій структурі конфігураційних файлів можна вручну вносити додаткові налаштування, наприклад, обмеження маршрутів, використання певних DNS або створення більш складних сценаріїв доступу.

Загалом, процес генерації клієнтських конфігурацій і підключення пристроїв у PiVPN є максимально спрощеним, безпечним та ефективним. Завдяки використанню WireGuard конфігурації є легкими, зрозумілими та не містять зайвих параметрів, що знижує ймовірність помилок. Це робить систему зручною як для домашніх користувачів, так і для адміністраторів невеликих мереж, які

прагнуть забезпечити віддалений доступ до ресурсів через надійний та швидкий VPN.

Крім основних можливостей створення клієнтських конфігурацій, PiVPN також надає інструменти для централізованого управління всіма підключеними пристроями. Це включає відображення статусу активних сесій, статистику трафіку, можливість відключення певних клієнтів або повне видалення їх з конфігурації. Адміністратор має змогу легко додати нового користувача в будь-який момент, а також забезпечити унікальність кожної конфігурації завдяки індивідуальним ключам, що запобігає дублюванню або небажаному доступу. Усі створені профілі зберігаються у визначеному каталозі сервера, тому їх резервне копіювання та архівування не викликає труднощів і може бути автоматизоване, що важливо для збереження безперервності роботи сервісу.

Крім звичайних пристроїв, таких як смартфони та ноутбуки, до VPN можна підключати інші вузли мережі — наприклад, інші сервери, IP-камери, маршрутизатори з підтримкою WireGuard, що дозволяє будувати розгалужені захищені мережі. Це відкриває широкі можливості для розгортання внутрішньої інфраструктури навіть без публічних IP-адрес, оскільки WireGuard може працювати через NAT або у складних мережевих середовищах завдяки механізмам keepalive та високій толерантності до зміни маршрутів.

Особливістю WireGuard у контексті клієнтських конфігурацій є його мінімалізм і чіткість. Конфігурація містить лише найнеобхідніше, без надлишкових параметрів, що часто зустрічаються в інших протоколах, як-от OpenVPN. Це знижує поріг входження для новачків та значно спрощує автоматизацію розгортання. Наприклад, у корпоративних сценаріях можливе автоматичне створення та поширення конфігурацій за допомогою скриптів, що базуються на PiVPN API або інтеграції з системами управління пристроями.

Завдяки підтримці сучасних криптографічних стандартів, кожна конфігурація забезпечує не тільки шифрування, а й автентифікацію на основі ключів. Це виключає необхідність використання паролів, які можуть бути

скомпрометовані або підібрані. Клієнтський пристрій може бути впізнаний виключно за своїм ключем, і навіть у разі втрати конфігурації, зловмисник не зможе з’єднатись із сервером без відповідного дозволу.

Завдяки своїй ефективності та прозорості, процес генерації конфігурацій та підключення клієнтів у PiVPN з WireGuard підходить як для одноразових, так і для масштабних розгортань. Його можна впровадити у навчальних, домашніх або навіть виробничих умовах, забезпечивши стабільний захист каналу зв’язку та мінімальні затримки. Усе це виводить персональний VPN на новий рівень доступності та зручності, поєднуючи безпеку з простотою управління.

3.5 Тестування та безпекові налаштування VPN-з'єднання

Після встановлення та базового налаштування VPN-сервера критично важливо провести комплексне тестування для підтвердження коректної роботи системи, а також впровадити додаткові заходи безпеки для захисту як самого сервера, так і мережевого трафіку користувачів. Тестування VPN-з’єднання включає перевірку працездатності VPN-протоколу, стабільності підключення, коректності маршрутизації та шифрування даних. Одним із перших кроків є перевірка активності служби VPN за допомогою системних інструментів, наприклад, перевірка статусу сервісу WireGuard або OpenVPN через команди systemctl. Це дає змогу впевнитися, що сервер працює без помилок і не припиняє свою діяльність через внутрішні збої. Далі проводиться тест відкритих портів на сервері, щоб переконатися, що обраний порт для VPN доступний і не блокується фаєрволом або провайдером. Застосовуються команди на кшталт netstat або ss для моніторингу мережевих з’єднань. Після цього слід перевірити працездатність клієнтських конфігурацій, імпортованих у відповідний VPN-клієнт на різних пристроях, включаючи смартфони, ПК та планшети. Підключення має бути стабільним, без частих розривів, із підтвердженням успішної аутентифікації та шифрування з’єднання. Особливу увагу варто звернути на зміну IP-адреси

користувача при підключенні до VPN, що свідчить про успішне тунелювання трафіку через сервер. Тестування маршрутизації передбачає перевірку доступу до локальних ресурсів, які є у внутрішній мережі, а також до інтернет-ресурсів, що підтверджує коректність налаштувань IP forwarding і NAT. Крім цього, варто провести аналіз журналів VPN-сервера для виявлення можливих помилок, несанкціонованих спроб доступу або інших аномалій, що може сигналізувати про потенційні загрози або проблеми з конфігурацією.

З безпекової точки зору важливо посилити захист VPN-сервера, впроваджуючи додаткові заходи, які мінімізують ризики атак та несанкціонованого доступу. Перш за все рекомендується змінити стандартний порт VPN-сервера на нестандартний, що ускладнить виявлення сервера автоматичними сканерами в мережі. Не менш важливим є обмеження доступу до сервера за допомогою файрвола, зокрема, налаштування правил для UFW або IPTables, які дозволяють пропускати лише необхідний трафік, блокуючи інші порти і протоколи. Для підвищення безпеки SSH-доступу варто відмовитись від автентифікації за паролем, використовуючи замість цього ключі SSH, що значно знижує ризик злому через перебір паролів. Автоматизація оновлення системи та VPN-програмного забезпечення допоможе підтримувати сервер у актуальному стані, своєчасно встановлюючи патчі безпеки та закриваючи відомі вразливості. Окремо варто розглянути використання додаткових засобів моніторингу та оповіщення про підозрілі події, які можуть сигналізувати про атаки або проблеми з роботою VPN. Усі ключі та конфігурації необхідно зберігати у захищеному вигляді, з регулярним створенням резервних копій, щоб уникнути втрати доступу через випадкове видалення або збої системи. Впровадження багаторівневої автентифікації, якщо це підтримується обраним VPN-протоколом, може ще більше підвищити рівень безпеки.

Загалом, тестування та безпекові налаштування VPN-з’єднання є невід’ємною частиною процесу розгортання захищеного персонального VPN-сервера, що гарантує надійність, стабільність та захист даних користувачів.

Регулярний моніторинг і своєчасне оновлення конфігурацій дозволяють підтримувати безпеку на високому рівні, а комплексний підхід до перевірки роботи серверу забезпечує комфортний та безпечний доступ до мережі з будь-якого місця світу.

3.6. Висновки

У третьому розділі дипломної роботи було детально розглянуто процес встановлення, конфігурації та тестування персонального VPN-сервера на базі Raspberry Pi із використанням програмного забезпечення PiVPN у віртуальному середовищі VirtualBox. Цей розділ є надзвичайно важливим з огляду на практичну реалізацію теоретичних концепцій, викладених у першому розділі, і демонструє, як за допомогою відкритих і безкоштовних інструментів можна створити надійне, гнучке та безпечне рішення для захищеного доступу до мережі.

Перш за все, було обґрунтовано вибір саме PiVPN як універсального і зручного інструменту для розгортання VPN-сервера. PiVPN значно спрощує процес встановлення і налаштування VPN-з’єднання, автоматизуючи багато рутинних операцій, що звичайно вимагають глибоких знань і значного часу. Це дозволяє навіть користувачам із базовими навичками роботи з Linux створити власний персональний VPN з мінімальними зусиллями. Особливе значення має підтримка PiVPN двох популярних VPN-протоколів — OpenVPN та WireGuard, причому в роботі було обрано WireGuard завдяки його сучасній архітектурі, високій продуктивності, простоті конфігурації та ефективним механізмам безпеки.

Процес інсталяції PiVPN, описаний у розділі, включає завантаження скрипту з офіційного сайту, запуск автоматичного інсталятору, налаштування ключових параметрів системи, таких як порт, протокол і метод маршрутизації. Особливої уваги заслуговує можливість налаштування фаєрволу та IP forwarding, що є критично важливими для забезпечення безпечної та коректної роботи VPN-

з’єднання. Докладно було розглянуто процес генерації клієнтських конфігурацій, що забезпечує простоту підключення пристроїв користувачів до VPN-сервера. Файли конфігурації містять всі необхідні параметри, включно з ключами шифрування, адресами серверів та параметрами DNS, що робить налаштування VPN на клієнтських пристроях швидким і безпомилковим.

Увага в розділі також приділялась тестуванню працездатності VPN-з’єднання. Перевірка запуску служби, моніторинг журналів, тестування підключення клієнтів із різних платформ, а також контроль зміни IP-адреси і доступу до внутрішніх мережевих ресурсів демонструють комплексний підхід до оцінки функціональності системи. Це дозволяє своєчасно виявляти й усувати можливі помилки, забезпечуючи безперебійне і безпечне функціонування VPN.

Значний акцент у розділі зроблено на безпекових налаштуваннях, що є ключовими для персональних VPN-серверів. Забезпечення захисту приватних ключів, зміна стандартних портів, обмеження доступу по SSH лише за допомогою ключів замість паролів, налаштування фаєрволу UFW, впровадження регулярного оновлення системи – усе це комплексні заходи, що мінімізують ризики злому та несанкціонованого доступу. Окремо розглянуто важливість створення резервних копій конфігураційних файлів і ключів, що дозволяє оперативно відновити працездатність VPN-сервера у випадку апаратних або програмних збоїв.

Таким чином, третій розділ демонструє, що використання PiVPN у поєднанні з WireGuard на платформі Raspberry Pi є ефективним і доступним рішенням для побудови персонального VPN-сервера. Воно поєднує у собі високу продуктивність, простоту адміністрування, гнучкість конфігурації та надійний рівень безпеки. Розроблене рішення дає змогу користувачам отримувати захищений доступ до домашньої або корпоративної мережі з будь-якої точки світу, що є особливо актуальним в умовах сучасних вимог до кібербезпеки та зростаючої кількості загроз у мережі Інтернет.

Цей розділ створює міцну основу для подальшого впровадження VPN-сервера в реальних умовах експлуатації.

ВИСНОВКИ

У ході виконання даної дипломної роботи було проведено всебічне дослідження та практичну реалізацію персонального VPN-сервера на базі апаратної платформи Raspberry Pi із використанням віртуалізації через VirtualBox та програмного забезпечення PiVPN. Робота охопила теоретичні основи функціонування віртуальних приватних мереж, особливості основних VPN-протоколів, таких як OpenVPN і WireGuard, а також принципи шифрування і автентифікації, що забезпечують безпеку передачі даних у VPN-з'єднаннях.

У першому розділі було детально розглянуто поняття VPN, основні протоколи та їх характеристики, а також порівняння різних типів VPN-рішень: хмарних, комерційних та локальних. Такий теоретичний фундамент дозволив обґрунтувати вибір персонального VPN-сервера як ефективного та гнучкого інструменту для забезпечення безпечного віддаленого доступу до мережі.

Другий розділ присвячений практичній частині роботи, зокрема вибору апаратної та програмної платформи, встановленню і налаштуванню віртуального середовища VirtualBox, завантаженню та підготовці Raspberry Pi OS для подальшої емуляції. Була проведена детальна конфігурація віртуальної машини, яка дозволила імітувати роботу фізичного Raspberry Pi з оптимальними параметрами для ефективної роботи VPN-сервера.

У третьому розділі описано процес встановлення PiVPN, вибору протоколу WireGuard, генерації конфігураційних файлів для клієнтів, налаштування мережевих параметрів та безпекових механізмів, а також комплексне тестування роботи VPN-з'єднання. Особлива увага була приділена питанням безпеки, таким як налаштування фаєрволу, захист доступу до сервера, регулярне оновлення системи та резервне копіювання.

В результаті проведеної роботи було створено стабільний, безпечний та ефективний персональний VPN-сервер, який можна використовувати для захищеного доступу до домашньої або корпоративної мережі з будь-якої точки

світу. Впроваджене рішення поєднує у собі переваги відкритого програмного забезпечення, доступної апаратної платформи і сучасних методів шифрування, що забезпечує високий рівень конфіденційності та захисту даних.

Таким чином, реалізований VPN-сервер є надійним і гнучким інструментом, який відповідає сучасним вимогам безпеки інформації та може бути рекомендований до використання як у побутових, так і в малих корпоративних мережах. Результати роботи відкривають можливості для подальшого розвитку і масштабування системи, інтеграції з іншими мережевими сервісами, а також удосконалення функціоналу для задоволення специфічних потреб користувачів. Загалом, проведена робота підтверджує актуальність та практичну цінність персональних VPN-рішень на основі Raspberry Pi та PiVPN.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. PiVPN – The Simplest VPN Installer for Linux and Raspberry Pi URL: https://www.pivpn.io (дата звернення 26.05.25).
2. WireGuard – Fast, Modern, Secure VPN Tunnel URL: https://www.wireguard.com (дата звернення 26.05.25).
3. Raspberry Pi OS Documentation URL: https://www.raspberrypi.com/documentation (дата звернення 26.05.25).
4. Oracle VirtualBox – Virtualization Software URL: https://www.virtualbox.org (дата звернення 26.05.25).
5. OpenVPN – Secure VPN Service URL: https://openvpn.net (дата звернення 26.05.25).
6. Raspberry Pi Imager – Install Raspberry Pi OS URL: https://www.raspberrypi.com/software/ (дата звернення 26.05.25).
7. PiVPN GitHub Repository URL: https://github.com/pivpn/pivpn (дата звернення 26.05.25).
8. WireGuard Quick Start Guide URL: https://www.wireguard.com/quickstart/
9. Ubuntu Docs – WireGuard VPN Setup URL: https://ubuntu.com/server/docs/service-wireguard (дата звернення 26.05.25).
10. Debian Wiki – WireGuard Setup URL: https://wiki.debian.org/WireGuard (дата звернення 26.05.25).
11. How to Set Up a VPN Server on Raspberry Pi (Pi My Life Up) URL: https://pimylifeup.com/raspberry-pi-vpn-server/ (дата звернення 26.05.25).
12. How to Install WireGuard VPN on Linux (DigitalOcean) URL: https://www.digitalocean.com/community/tutorials/how-to-set-up-wireguard-on-ubuntu-20-04 (дата звернення 26.05.25).
13. What Is a VPN? (Cloudflare Learning Center) URL: https://www.cloudflare.com/learning/network-layer/what-is-a-vpn/ (дата звернення 26.05.25).
14. DNS Leak Test URL: https://dnsleaktest.com (дата звернення 26.05.25).

15. WhatIsMyIPAddress – IP Checker URL: https://whatismyipaddress.com (дата звернення 26.05.25).

16. ExpressVPN – Official VPN Provider URL: https://www.expressvpn.com (дата звернення 26.05.25).

17. NordVPN – WireGuard Implementation URL: https://nordvpn.com/features/wireguard/ (дата звернення 26.05.25).

18. ProtonVPN – WireGuard Info and Setup URL: https://protonvpn.com/support/what-is-wireguard/ (дата звернення 26.05.25).

19. WireGuard on OpenWRT (OpenWRT Wiki) URL: https://openwrt.org/docs/guide-user/services/vpn/wireguard/start (дата звернення 26.05.25).

20. Performance Analysis of VPN Protocols (IEEE Xplore) URL: https://ieeexplore.ieee.org/document/9242921 (дата звернення 26.05.25).

21. Comparative Study of VPN Protocols (ScienceDirect) URL: https://www.sciencedirect.com/science/article/pii/S1877050920316481 (дата звернення 26.05.25).

22. WireGuard vs OpenVPN (ResearchGate) URL: https://www.researchgate.net/publication/348700084 (дата звернення 26.05.25).

23. Install WireGuard VPN on Debian (Linode Docs) URL: https://www.linode.com/docs/guides/install-wireguard-on-debian/ (дата звернення 26.05.25).

24. Raspberry Pi Blog URL: https://www.raspberrypi.com/news/ (дата звернення 26.05.25).

25. Raspberry Pi Stack Exchange URL: https://raspberrypi.stackexchange.com/ (дата звернення 26.05.25).

26. UFW – Uncomplicated Firewall Setup URL: https://wiki.ubuntu.com/UncomplicatedFirewall (дата звернення 26.05.25).

27. Netgate pfSense Docs – WireGuard URL: https://docs.netgate.com/pfsense/en/latest/vpn/wireguard/index.html (дата звернення 26.05.25).

28. WireGuard Tools GitHub URL: https://github.com/WireGuard (дата звернення 26.05.25).

29. OpenVPN GitHub URL: https://github.com/OpenVPN (дата звернення 26.05.25).

30. Arch Linux Wiki – WireGuard Setup URL: https://wiki.archlinux.org/title/WireGuard (дата звернення 26.05.25).

31. Learn Raspberry Pi (Udemy Course) URL: https://www.udemy.com/course/learn-raspberry-pi/

32. Introduction to Cybersecurity (Coursera – IBM) URL: https://www.coursera.org/learn/ibm-cybersecurity-basics (дата звернення 26.05.25).

33. How to Use PiVPN to Create a Secure Home VPN (Tom's Hardware) URL: https://www.tomshardware.com/how-to/setup-vpn-server-pivpn (дата звернення 26.05.25).

34. Secure VPNs for Beginners (How-To Geek) URL: https://www.howtogeek.com/221001/what-is-a-vpn-and-why-would-i-need-one/ (дата звернення 26.05.25).

35. VPN Explained (Mozilla Foundation) URL: https://foundation.mozilla.org/en/privacynotincluded/articles/what-is-a-vpn/ (дата звернення 26.05.25).

36. Use VirtualBox to Emulate Raspberry Pi URL: https://medium.com/@josephtaylor/raspberry-pi-emulation-using-virtualbox-cb8b6f8c74c3 (дата звернення 26.05.25).

37. Kali Linux on Raspberry Pi Setup URL: https://www.kali.org/get-kali/#kali-arm (дата звернення 26.05.25).

38. Raspberry Pi Network Configuration URL: https://www.raspberrypi.com/documentation/computers/configuration.html (дата звернення 26.05.25).

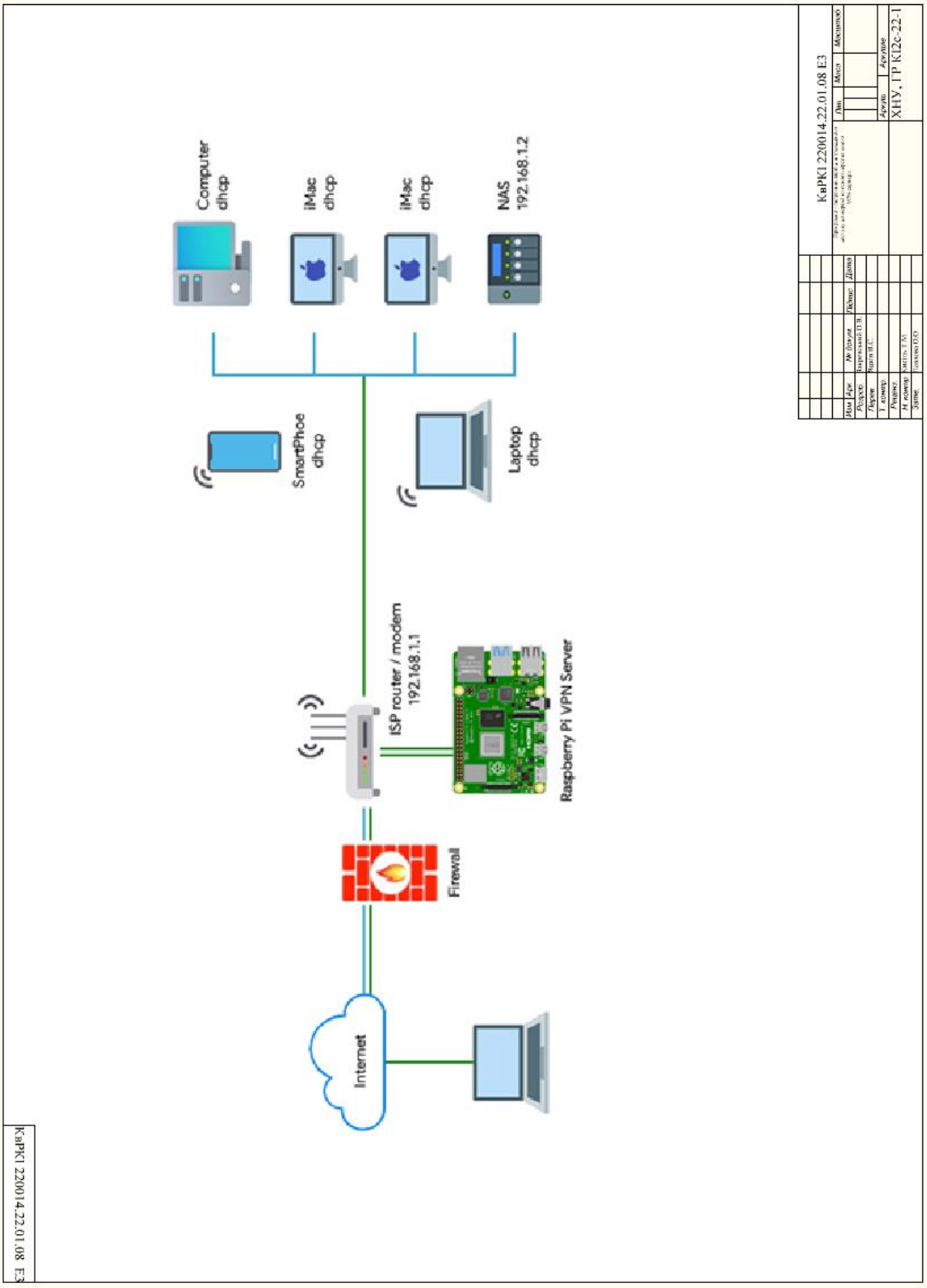
39. VirtualBox Guest Additions Installation URL: https://www.virtualbox.org/manual/ch04.html (дата звернення 26.05.25).

40. Setting Up Networking in VirtualBox URL: https://www.virtualbox.org/manual/ch06.html (дата звернення 26.05.25).

Додаток А

(обов’язковий)

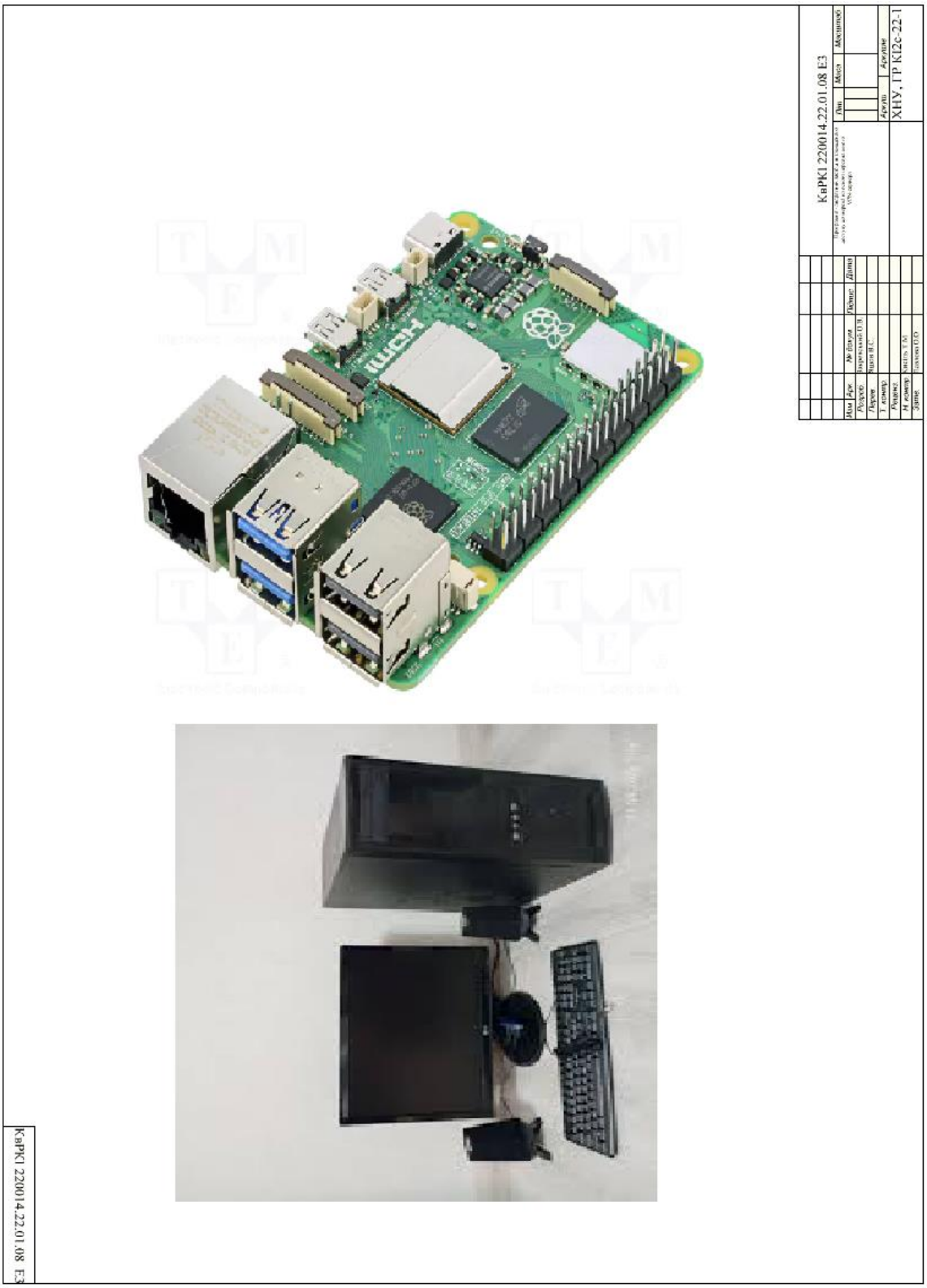
КОПІЯ КРЕСЛЕННЯ «АРХІТЕКТУРА ПЗ ПРОЄКТУ»



Додаток Б

(обов’язковий)

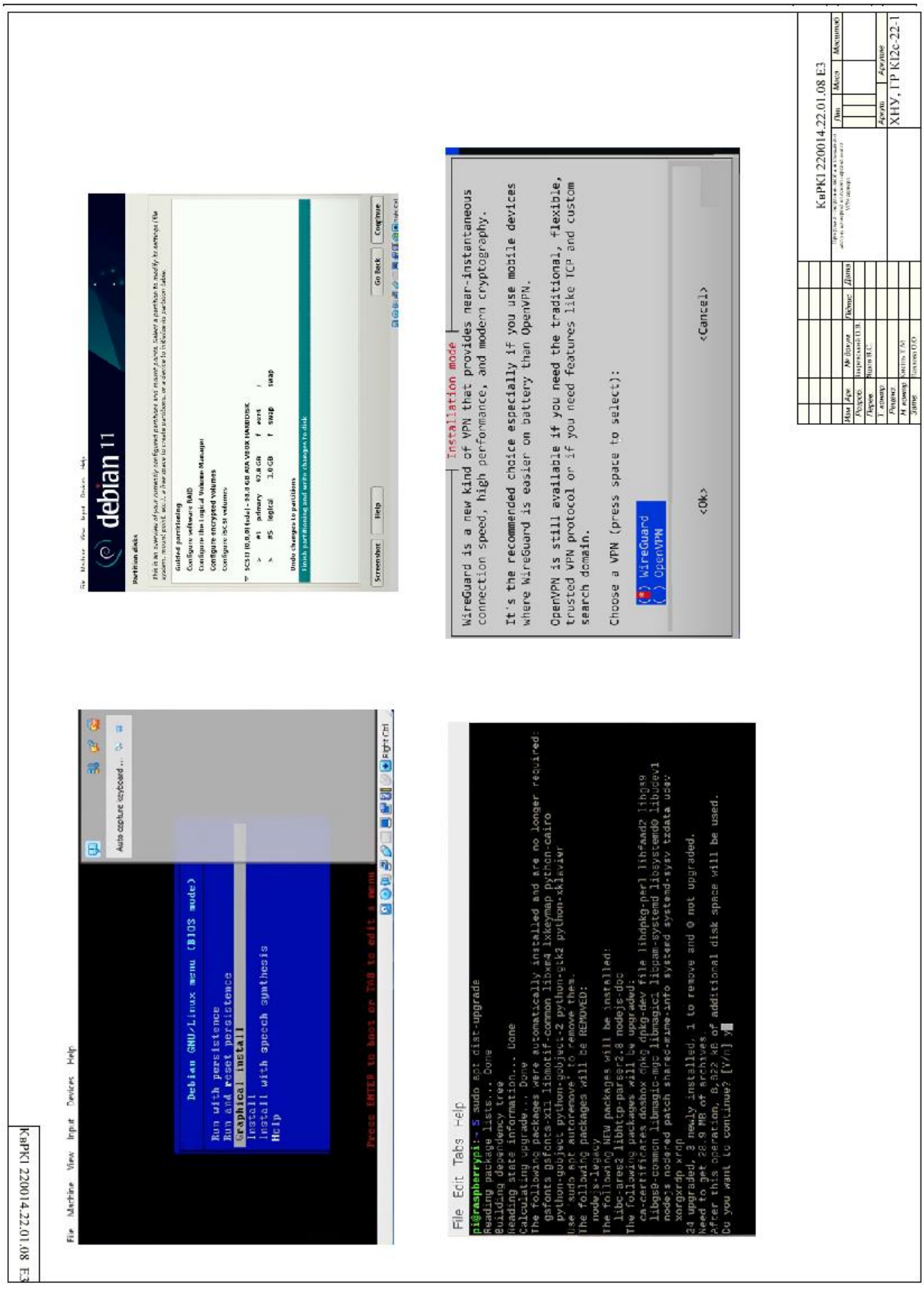
КОПІЯ КРЕСЛЕННЯ «АПАРАТНА ЧАСТИНА»



Додаток В

(обов'язковий)

КОПІЯ КРЕСЛЕННЯ «ВСТАНОВЛЕННЯ ТА НАЛАШТВАНЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»



РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Програмно-апаратний засіб для захищеного доступу до мережі на основі персонального VPN-сервера

Автор: Закревський Олексій

Спеціальність: 123– Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Василь Яцків д.т.н.

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) Запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи.;
2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
3) Окремі збіги представлені загальновживаними фразами, наприклад: «на рисунку зображено», «загальна структура системи», «висновки до розділу» тощо.
4) Якість запозичень відповідає технічним особливостям дослідження: виявлено збіги в кодах, формулах і термінах, які є вихідними даними до великої кількості задач і не можуть вважатися авторськими порушеннями.
5) Система зафіксувала технічні модифікації тексту, зокрема: заміну окремих символів ,скорочення індексів у формулах, зміну розміщення символів. Це є наслідком форматування або експорту документа, а не цілеспрямованого уникнення перевірки.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості StrikePlagiarism, складає 3.4% і адресується до 27 першоджерела; та системою Anti-Plagiarism складає 0%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ Василь ЯЦКІВ

Гарант ОП _____ Андрій НІЧЕПОРУК

Завідувач кафедри КІС _____ Ольга ПАВЛОВА

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Закревський ОЛЕКСІЙ

Співавтор:

Назва: Закревський_Програмно-апаратний засіб для захищеного доступу до мережі на основі персо-нального VPN-сервера

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:1.5%

Коефіцієнт подібності 2:0%

Мікропробіли: 6

Заміна букв: 0

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2025-06-12 21:55:03.0

Після аналізу Звіту подібності констатую наступне:

☒ **Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.**

☐ **Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.**

☐ **Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.**

Обгрунтування:

2025-06-13 Доцент Андрій Нічепорук

Дата експерт

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Закревський Олексій Володимирович

Тема: Програмно-апаратний засіб для захищеного доступу до мережі на основі персонального VPN-сервера

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 66

1.Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є розробка програмно-апаратного засобу для забезпечення захищеного доступу до мережі на основі персонального VPN-сервера. У межах роботи реалізовано повний цикл створення VPN-рішення, починаючи з вибору апаратної та програмної платформи, емуляції Raspberry Pi у VirtualBox, встановлення та налаштування системи, а також тестування працездатності. Було прийнято рішення використовувати технологію PiVPN з протоколом WireGuard, що дало змогу забезпечити високу продуктивність, простоту розгортання і сучасні засоби шифрування.

2.Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає завданню, поставленому у дипломному проєкті. Усі етапи, передбачені технічним завданням, були реалізовані: вибір архітектури, моделювання та налаштування середовища, встановлення VPN-сервера, конфігурація клієнтів, а також тестування з'єднання з урахуванням безпекових параметрів.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі досліджено теоретичні основи побудови віртуальних приватних мереж, класифікацію VPN-протоколів, принципи шифрування та автентифікації, а також розглянуто переваги та недоліки персонального VPN-сервера. Окрема увага приділена порівнянню хмарних, комерційних та локальних рішень. У другому

розділі реалізовано вибір програмної та апаратної платформи для створення тестового середовища, встановлення VirtualBox, завантаження та налаштування Raspberry Pi OS, створення віртуальної машини та перевірка її працездатності. Технічна частина включає точні параметри конфігурації та практичні аспекти віртуалізації. У третьому розділі виконано повну установку та налаштування PiVPN, реалізовано протокол WireGuard як основний, згенеровано клієнтські конфігурації, проведено тестування з'єднання та налаштування безпекових параметрів. Робота базується на сучасних підходах до розгортання VPN, використовуючи легкий та ефективний протокол WireGuard, який є однією з найсучасніших розробок у сфері мережевої безпеки.

4. Позитивні сторони роботи: Робота відзначається високим рівнем практичної реалізації, використанням актуальних інструментів віртуалізації, сучасних VPN-протоколів і гнучких засобів керування сервером. Вибір PiVPN і WireGuard забезпечив баланс між безпекою, швидкістю та простотою експлуатації. Була продемонстрована здатність системи до масштабування, розширення та перенесення у реальне середовище Raspberry Pi.

5. Негативні сторони роботи: У роботі обмежено висвітлено аспекти масштабування на великі корпоративні інфраструктури та інтеграцію з централізованими системами управління. Також залишилися поза увагою деякі додаткові методи моніторингу й автоматичного оновлення системи.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

8. Інші зауваження: _____

9. Оцінка дипломної роботи: добре

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Бездрабчик І.П. Зав.каф. ІПЗ, ХНУ

"17" червня 2025 р. _____ (підпис)

Завідувачу кафедри КІІС
д-р. філософії, доц. Ользі ПАВЛОВІЙ
Олексій ЗАКРЕВСЬКИЙ
ПІБ здобувача вищої освіти

ФІТ, 3 курсу, групи КІ2с-22-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Strike-Plagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

15.06 2025 року

Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 0.0%

Dictionaries check: en_US, ru_RU, ua_UA. **Errors in the documents: 13%**

	Document		Sum coincidence on the DB	
ID: 245500 Title: БКР Програмно-апаратний засіб для захищеного доступу до мережі на основі персо-нального VPN-сервера Added in a DB: 2025-06-12 Authors: Закревський ОЛЕКСІЙ Heads: Василь ЯЦКІВ Consultants: Opponents:	Symbols	Lexemes	Symbols	Lexemes
	105442	690	676 (1%)	8 (1%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes