ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «ДВОФАКТОРНА АВТЕНТИФІКАЦІЯ НА ОСНОВІ МЕСЕНДЖЕРІВ У КОРПОРАТИВНИХ VPN ДЛЯ ЗАБЕЗПЕЧЕННЯ ЗАХИЩЕНОГО ВІДДАЛЕНОГО ДОСТУПУ»

на здобуття освітнього ступеня магістра
зі спеціальності 123 Ком'ютерна інженерія
(код, найменування спеціальності)
освітньо-професійної програми Комп'ютерні системи та мережі
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Кирило ВАСИЛЕНКОВ
(підпис) *Ім'я, ПРІЗВИЩЕ здобувача*

Виконав:
здобувач вищої освіти Кирило ВАСИЛЕНКОВ
група КСДМ-61

Керівник: Артем АНТОНЕНКО
науковий ступінь, вчене звання к.т.н., доцент

Рецензент:
науковий ступінь, вчене звання _____
Ім'я, ПРІЗВИЩЕ

Київ 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

Ступінь вищої освіти Магістр

Спеціальність 123 Комп'ютерна інженерія

Освітньо-професійна програма Комп'ютерні системи та мережі

ЗАТВЕРДЖУЮ
Завідувач кафедри
Комп'ютерної інженерії

_____Наталія ЛАЩЕВСЬКА
«____ » _____2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Василенкову Кирилу Ігоровичу
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Двофакторна автентифікація на основі месенджерів у корпоративних vpn для забезпечення захищеного віддаленого доступу

керівник кваліфікаційної роботи Артем АНТОНЕНКО к.т.н., доцент,
(Ім'я, ПРІЗВИЩЕ науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «19» 10.2023р. №145

2. Строк подання кваліфікаційної роботи «29» грудня 2023р.

3. Вихідні дані до кваліфікаційної роботи: вимоги до ком'ютерної мережі

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

Дослідження принципів роботи VPN авторизації та аутентифікації

Аналіз технологій рішень на ринку для безпечного та надійного підключення

Розробка вимог до комплексу підключення до віддаленого доступу та реалізація.

5. Перелік графічного матеріалу: *презентація*

6. Дата видачі завдання «19» жовтня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Етапи виконання дипломного проекту	Термін виконання етапів	Примітка
1	Провести аналіз літератури за темою дипломної роботи та аналіз існуючих систем	20.10.23 - 26.10.23	
2	Вивчити спеціальну літературу і технічну документацію	26.10.23- 31.10.23	
3	Проаналізувати можливость налаштування віддаленого доступу за допомогою месенджерів, інструменти для реалізації та написати розділ 1	01.11.23- 09.11.23	
4	Проаналізувати вимоги та критерії до системи, порівняти та вибрати інструменти для реалізації серверної та програмної частини проекту та написати розділ 2	10.11.23- 18.11.23	
5	Реалізувати серверну та програмну частину проекту для можливості застосовувати другий фактор у вигляді мессендежу та написати розділ 3	19.11.23- 05.12.23	
6	Оформити пояснювальну записку иа супроводжувальну документацію	06.12.23- 19.12.23	
7	Підготовити графічний демонстраційний матеріал	20.12.23- 29.12.23	

Здобувач вищої освіти _____ Кирило ВАСИЛЕНКОВ
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник
кваліфікаційної роботи _____ Артем АНТОНЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 84 стор., 1 табл., 42 рис., 27 джерел.

Мета роботи – розробити та налаштувати комплекс для безпечного віддаленого підключення з використанням двофакторної авторизації.

Об'єкт дослідження – Цифрові системи аутентифікації в корпоративних віртуальних приватних мережах (VPN).

Короткий зміст роботи: У роботі проведено дослідження принципів роботи VPN, авторизації та аутентифікації, двофакторна аутентифікація, надійність та доступність схожих рішень. Створено алгоритм та саму программу для можливості підключення до віддалених мереж підтверджуючи свою особу через месенджери

КЛЮЧОВІ СЛОВА: *VPN*, *RADIUS*, АВТОРИЗАЦІЯ, АУТЕНТИФІКАЦІЯ, МЕСЕНДЖЕР

ABSTRACT

Text part of the master's qualification work: 84 pages, 42 pictures, 1 table, 27 sources.

The purpose of the work is to develop and configure a system for secure remote connection utilizing two-factor authentication.

The object of the research is Digital Authentication Systems in corporate Virtual Private Networks (VPNs).

Abstract: The study delves into the principles of VPN operation, authorization, authentication, two-factor authentication, and the reliability and accessibility of similar solutions. An algorithm and program have been created to enable connection to remote networks by verifying one's identity through messengers.

KEYWORDS*: VPN, RADIUS, AUTHORIZATION, AUTHENTICATION, MESSENGER*

ЗМІСТ

Перелік умовних позначень, скорочень, термінів

VPN	–	*Virtual private network*
LAN	–	*Local Area Network*
WAN	–	*World Area Network*
RADIUS	–	*Remote Authentication Dial-In User Service*
SSTP	–	*Secure Socket Tunneling Protocol*
ICMP	–	*Internet Control Message Protocol*
IP	–	*Internet Protocol*
COVID-19	–	*COronaVIrus Disease 2019*
UDP	–	*User Datagram Protocol*
SSL	–	*Secure Sockets Layer*
UI	–	*User Interface*
2FA	–	*Two-factor authentication*
DNS	–	*Domain Name System*
ISP	–	*Internet Service Provider*
IPsec	–	*IP Security*
L2TP	–	*Layer 2 Tunnelling Protocol*
AAA	–	*Authentication, Authorization, Accounting*
SHA	–	*Secure Hash Algorithms*
SSH	–	*Secure Shell*
AI	–	*Artificial Intelligence*

ВСТУП

Тема дипломної роботи є актуальною, оскільки вона стосується однієї з найважливіших проблем сучасної інформаційної безпеки - аутентифікації користувачів, які отримують віддалений доступ до корпоративних ресурсів через віртуальну приватну мережу (VPN). Віддалена робота стала нормою для багатьох компаній та організацій, які прагнуть забезпечити продуктивність та безпеку своїх працівників у складних умовах, таких як пандемія COVID-19, глобалізація, цифрова трансформація, екологічна свідомість тощо. Ось так за даними дослідження, проведеного компанією Cisco, кількість працівників, які працюють віддалено, зросла з 20% у 2017 році до 62% у 2020 році. А якщо вдаватися в новішу статистику від Cisco, які зазначили у дослідженні, що 82% співробітників кажуть, що можливість працювати з будь-якого місця зробила їх щасливішими, також 60% - що їхня продуктивніть зросла, але тільки 1 з 4 опитуванних сказали, що організаця, в якій вони працюють – повністю підготовлена до гібридної роботи. Отже VPN є необхідним інструментом для забезпечення захищеного з’єднання між приватною мережею (наприклад, мережею компанії) та публічною мережею (наприклад, Інтернетом), а також для отримання доступу до ресурсів, які можуть бути обмежені або цензуровані у певних регіонах.

Однак VPN не є бездоганним рішенням. VPN також має свої проблеми та виклики, які потребують постійного удосконалення та адаптації до змінних ситуацій. Однією з найбільших проблем VPN є аутентифікація користувачів, тобто перевірка того, що користувач, який намагається підключитися до VPN, є тим, за кого він себе видає. Якщо зловмисник отримає доступ до облікового запису користувача VPN, він зможе отримати доступ до всіх корпоративних ресурсів, які доступні через VPN. Це може призвести до витоку, втрати або пошкодження конфіденційної або важливої інформації, а також до порушення законодавства та нормативів щодо захисту даних.

Тому необхідно знайти ефективні та зручні способи підвищення рівня аутентифікації користувачів VPN, які зможуть запобігти або ускладнити спроби

несанкціонованого доступу до корпоративних ресурсів. Одним з таких способів є використання двофакторної аутентифікації (2FA) на основі месенджерів, таких як Telegram, які дозволяють швидко та легко отримувати та вводити одноразові коди підтвердження через смартфон або інший пристрій. Цей метод має багато переваг, таких як зниження залежності від паролів, підвищення зручності та задоволення користувачів, зменшення витрат на обслуговування та підтримку, підвищення гнучкості та сумісності тощо. Однак цей метод також має свої недоліки, такі як залежність від наявності та функціонування месенджера, ризики втрати або крадіжки смартфона, можливість обходу або підробки кодів підтвердження тощо. Тому необхідно дослідити та проаналізувати цей метод з різних точок зору, а також розробити та протестувати прототип системи двофакторної аутентифікації на основі месенджера Telegram для корпоративних VPN.

Система двофакторної аутентифікації на основі месенджерів у корпоративних VPN складається з трьох основних компонентів: VPN-сервера, RADIUS-сервера та месенджера. VPN-сервер є точкою входу для користувачів, які хочуть отримати доступ до корпоративних ресурсів через захищене з'єднання. RADIUS-сервер є центральним компонентом, який відповідає за аутентифікацію, авторизацію та облік користувачів VPN. Месенджер є додатковим фактором аутентифікації, який дозволяє користувачам отримувати та вводити одноразові коди підтвердження через свої смартфони або інші пристрої.

VPN-сервер є програмним або апаратним рішенням, яке дозволяє створювати віртуальні приватні мережі між різними пристроями та локаціями. VPN-сервер захищає передавані дані від перехоплення, зміни або підробки за допомогою шифрування, тунелювання та аутентифікації. VPN-сервер також дозволяє користувачам отримувати доступ до ресурсів, які можуть бути обмежені або цензуровані у певних регіонах. В рамках цієї дипломної роботи використовується VPN-сервер SoftEther, який є безкоштовним, відкритим та сумісним з різними операційними системами та протоколами.

RADIUS-сервер є програмним рішенням, яке виконує функції аутентифікації, авторизації та обліку користувачів VPN. RADIUS-сервер перевіряє облікові дані

користувачів, надає їм доступ до певних ресурсів та збирає статистику про їхню активність. RADIUS-сервер також взаємодіє з месенджером для генерації та відправки одноразових кодів підтвердження, які служать другим фактором аутентифікації. В рамках цієї дипломної роботи використовується RADIUS-сервер FreeRADIUS, який є безкоштовним, відкритим та популярним рішенням для управління доступом до мережі.

Месенджер є додатком для обміну повідомленнями, голосом, відео, файлами та іншими даними між користувачами. Месенджер використовується як додатковий фактор аутентифікації, який дозволяє користувачам отримувати та вводити одноразові коди підтвердження, які генеруються RADIUS-сервером. Месенджер також забезпечує шифрування та захист даних, які передаються між користувачами. В рамках цієї дипломної роботи використовується месенджер Telegram, який є безкоштовним, швидким, безпечним та сумісним з різними платформами.

Метою дипломної роботи є розробка та тестування прототипу системи двофакторної аутентифікації на основі месенджера Telegram для корпоративних VPN. Для досягнення цієї мети були поставлені наступні завдання:

- провести аналіз теоретичних основ та практичних аспектів віддаленої роботи, VPN, RADIUS, 2FA та месенджерів;
- визначити вимоги та критерії до системи двофакторної аутентифікації на основі месенджера Telegram для корпоративних VPN;
- розробити архітектуру та алгоритми роботи системи двофакторної аутентифікації на основі месенджера Telegram для корпоративних VPN;
- реалізувати прототип системи двофакторної аутентифікації на основі месенджера Telegram для корпоративних VPN за допомогою мови програмування Python та інших технологій;
- провести тестування та оцінку ефективності та зручності системи двофакторної аутентифікації на основі месенджера Telegram для корпоративних VPN.

Наукова новизна дипломної роботи полягає в тому, що вона пропонує новий підхід до підвищення рівня аутентифікації користувачів VPN за допомогою месенджера Telegram як додаткового фактора. Вона також аналізує переваги та недоліки цього підходу з різних точок зору, таких як безпека, зручність, витрати, сумісність тощо.

Практична значущість дипломної роботи полягає в тому, що вона розробляє та тестує прототип системи двофакторної аутентифікації на основі месенджера Telegram для корпоративних VPN, який може бути використаний як зразок або основа для подальшого розвитку та впровадження. Вона також демонструє, як можна використовувати сучасні технології, такі як Python, Telegram API, RADIUS, SoftEther VPN тощо, для реалізації системи двофакторної аутентифікації на основі месенджерів у корпоративних VPN.

РОЗДІЛ 1
ТЕОРЕТИЧНІ ОСНОВИ ТА ПРАКТИЧНІ АСПЕКТИ ВІДДАЛЕНОЇ РОБОТИ, VPN, RADIUS, 2FA ТА МЕСЕНДЖЕРІВ

1.1. Віддалена робота: переваги, виклики та тенденції

Віддалена робота - це форма організації роботи, яка дозволяє працівникам виконувати свої завдання та обов'язки поза традиційним офісним середовищем. Віддалена робота може здійснюватися з дому, спільних робочих просторів, кафе, готелів або будь-якого іншого місця, яке має надійне інтернет-з'єднання та підходяще обладнання. Віддалена робота може бути повночасною, неповночасною або спорадичною, залежно від потреб та уподобань роботодавця та працівника.

Віддалена робота стала більш поширеною та популярною в останні роки, особливо через пандемію COVID-19, яка змусила багато організацій та працівників прийняти віддалену роботу як спосіб впоратися з ризиками для здоров'я та безпеки, локдаунами, обмеженнями подорожей та заходами соціальної дистанції. За даними опитування, проведеного компанією Upwork, 41,8% американської робочої сили працювали віддалено в 2020 році, у порівнянні з 19,4% в 2017 році . Більше того, 36,2 мільйона американців (22% робочої сили) будуть працювати віддалено до 2025 року, що на 87% більше, ніж кількість віддалених працівників до пандемії .

Віддалена робота пропонує багато переваг для роботодавців та працівників, таких як:

- Збільшення продуктивності та ефективності: Віддалені працівники можуть працювати в більш комфортному та гнучкому середовищі, без відволікань та перерв, які властиві офісу. Вони також можуть заощадити час та гроші на дорозі, що може зменшити стрес та втому. Віддалені працівники також можуть мати більше автономії та контролю над своїм робочим графіком, темпом та методами, що може підвищити їхню мотивацію та креативність. За даними дослідження Стенфордського університету, віддалені працівники на 13% продуктивніші, ніж їхні колеги в офісі .

- Покращення балансу роботи, життя та задоволення: працівники, які працюють віддалено можуть мати більше свободи та гнучкості для управління своїми особистими та професійними обов’язками, такими як сім’я, здоров’я, хобі, освіта тощо. Вони також можуть мати більше можливостей для реалізації своїх пристрастей та інтересів, а також для подорожей та відкриття нових місць. Віддалені працівники також можуть насолоджуватися більшою приватністю та незалежністю, а також більшим почуттям влади та власності над своєю роботою.
- Зменшення витрат та впливу на навколишнє середовище: Віддалена робота може допомогти роботодавцям та працівникам заощадити гроші на різних витратах, таких як оренда офісу, комунальні послуги, обладнання, подорожі, їжа, одяг тощо. Віддалена робота також може допомогти зменшити вплив на навколишнє середовище, такий як викиди парникових газів, забруднення повітря, затори на дорогах тощо. За даними звіту Global Workplace Analytics, віддалена робота може заощадити роботодавцям 11 000 доларів на рік на одного працівника, а працівникам - від 2500 до 4000 доларів на рік . Більше того, віддалена робота може зменшити викиди парникових газів на 54 мільйони тонн на рік, що еквівалентно видаленню з доріг 10 мільйонів автомобілів .

Однак віддалена робота також ставить деякі виклики та недоліки, такі як:

- Труднощі з комунікацією та співпрацею: Віддалені працівники можуть стикатися з деякими бар’єрами та перешкодами у комунікації та співпраці зі своїми колегами, керівниками, клієнтами та іншими зацікавленими сторонами. Труднощі спілкування та співпраці: віддалені працівники можуть зіткнутися з деякими бар’єрами та перешкодами у спілкуванні та співпраці зі своїми колегами, менеджерами, клієнтами та іншими зацікавленими сторонами. У них можуть виникнути затримки, непорозуміння, неправильне тлумачення або технічні проблеми через відсутність взаємодії віч-на-віч, невербальних сигналів, зворотного зв’язку та підтримки. Вони також можуть відчувати себе ізольованими, відірваними або відчуженими від своєї команди

та організаційної культури. Згідно з опитуванням Buffer, 20% віддалених працівників повідомили, що комунікація та співпраця є для них найбільшою проблемою.

- Ризики безпеки та конфіденційності: Віддалені працівники можуть бути вразливішими перед ризиками безпеки та конфіденційності, такими як витоки даних, кібератаки, крадіжка особистості, фішингом, шкідливим програмним забезпеченням тощо. Вони також можуть мати менше контролю та захисту над своїми особистими та професійними даними, пристроями та мережами, особливо якщо вони використовують публічний або незахищений Wi-Fi, хмарні сервіси чи особисте обладнання. Вони також можуть стикатися з різними законами та нормами щодо захисту даних та конфіденційності, залежно від їхнього місця знаходження та юрисдикції. Згідно з звітом IBM, 70% віддалених працівників використовують свої особисті пристрої для роботи, але лише 39% з них мають належні заходи безпеки.
- Виклики інтеграції роботи та особистого життя: Віддалені працівники можуть мати складнощі з відокремленням та поєднанням сфери роботи та особистого життя, що може призвести до конфліктів ролей, порушень меж або ефекту переливання. Вони також можуть відчувати складнощі у керуванні своїм часом, завданнями та пріоритетами, а також у встановленні та підтримці послідовного та здорового робочого режиму. Вони можуть також стикатися з перевтомою, вигоранням чи втомою через відсутність чітких меж, очікувань та обмежень. За результатами опитування від Buffer, 18% віддалених працівників заявили, що найбільша складність для них полягає в тому, щоб відключитися після роботи.

У висновку можна сказати, що віддалена робота є складним і динамічним явищем, яке має багато переваг і викликів як для роботодавців, так і для співробітників. Віддалена робота є також тенденцією, яка, ймовірно, буде продовжувати розвиватися у майбутньому, оскільки все більше організацій і працівників приймають і впроваджують її як життєздатну і бажану опцію. Тому важливо розуміти та вирішувати різні аспекти та наслідки віддаленої роботи, а

також розвивати та впроваджувати ефективні та ефективні стратегії та практики для оптимізації та поліпшення результатів і вражень віддаленої роботи.

1.2 Віртуальна приватна мережа (VPN): принципи, цілі та технології

Віртуальна приватна мережа (VPN) - це технологія, яка створює безпечне та зашифроване з'єднання через менш безпечну мережу, наприклад, Інтернет. VPN дозволяє користувачам отримувати доступ до приватної мережі, такої як корпоративна або навчальна, віддалено та безпечно, немов вони прямо підключені до неї. VPN також дозволяє користувачам обійти географічні обмеження та цензуру, а також мати доступ до більшого обсягу контенту та послуг онлайн.

1.2.1 Основи віртуальної приватної мережі (VPN)

Щоб зрозуміти, як працює VPN, корисно знати деякі основні поняття та терміни, пов'язані з мережами та шифруванням. Ось деякі з них:

- Мережа - це група пристроїв, таких як комп'ютери, смартфони, планшети тощо, які підключені один до одного і можуть обмінюватися даними та інформацією. Мережа може бути локальною або глобальною, приватною або публічною, провідною або бездротовою тощо.
- Інтернет - це глобальна мережа мереж, яка з'єднує мільйони пристроїв та користувачів по всьому світу. Інтернет використовує набір протоколів, таких як TCP/IP, для зв'язку та передачі даних між різними мережами та пристроями.
- IP-адреса - це унікальний ідентифікатор, який надається кожному пристрою, підключеному до Інтернету. IP-адреса складається з чотирьох чисел, розділених крапками, наприклад, 192.168.1.1. IP-адреса може вказувати на місцезнаходження та ідентичність пристрою та користувача.
- Шифрування - це процес перетворення даних у незчитливу форму, використовуючи секретний ключ чи пароль. Шифрування може

захищати дані від несанкціонованого доступу, модифікації або втручання. Шифрування може бути симетричним або асиметричним, залежно від типу та кількості використовуваних ключів.

- Дешифрування - це процес перетворення зашифрованих даних назад у їхню вихідну форму, використовуючи такий самий чи інший ключ чи пароль. Дешифрування може бути виконане лише призначеним отримувачем, який має вірний ключ чи пароль.
- Авторизація - це процес надання чи відмови в доступі до мережі чи даних на основі ідентифікації та привілеїв користувача чи пристрою. Авторизація може контролювати те, що користувач чи пристрій може робити або бачити в мережі чи даних.

Використовуючи ці концепції та терміни, віртуальна приватна мережа (VPN) створює безпечне та зашифроване з'єднання між пристроєм користувача та сервером VPN, який виступає шлюзом до приватної мережі. З'єднання VPN може бути встановлене за допомогою різних методів, таких як програмне забезпечення, апаратне забезпечення або їх комбінація. Використані різні протоколи, такі як PPTP, L2TP, IPSec, SSL, OpenVPN тощо, для шифрування та тунелювання даних. Також можуть використовуватися різні режими, такі як роздільне тунелювання чи повне тунелювання, для маршрутизації даних.

Ось спрощений опис того, як працює з'єднання VPN:

- Користувач запускає програмне забезпечення VPN або додаток на своєму пристрої, вводить ім'я користувача та пароль, сканує відбиток пальця чи використовує інший метод аутентифікації.
- Програмне забезпечення VPN або додаток підключається до сервера VPN, розташованого в іншій країні чи регіоні, та відправляє запит на доступ до приватної мережі.
- Сервер VPN перевіряє ідентифікацію та облікові дані користувача та надає чи відмовляє в доступі до приватної мережі на основі правил авторизації.

- Якщо доступ наданий, сервер VPN та пристрій користувача обмінюються секретним ключем або паролем, який буде використовуватися для шифрування та дешифрування даних.
- Сервер VPN та пристрій користувача створюють безпечний та зашифрований тунель, який захищає дані від перехоплення, модифікації чи втручання.
- Пристрій користувача надсилає дані серверу VPN, який шифрує їх за допомогою секретного ключа чи пароля та додає новий заголовок з IP-адресою сервера VPN як джерелом та IP-адресою приватної мережі як призначенням.
- Сервер VPN надсилає зашифровані дані до приватної мережі, яка розшифровує їх за допомогою того самого ключа чи пароля та видаляє новий заголовок. Приватна мережа обробляє дані та надсилає відповідь назад на сервер VPN.
- Сервер VPN шифрує відповідь за допомогою того самого ключа чи пароля та додає новий заголовок з IP-адресою приватної мережі як джерелом та IP-адресою сервера VPN як призначенням.
- Сервер VPN надсилає зашифровану відповідь до пристрою користувача, який розшифровує її за допомогою того самого ключа чи пароля та видаляє новий заголовок. Пристрій користувача відображає відповідь користувачеві.

Використовуючи VPN, користувач може отримати доступ до приватної мережі та її ресурсів, таких як файли, принтери, бази даних тощо, віддалено та безпечно, ніби він прямо підключений до неї. Користувач також може отримати доступ до більшого обсягу контенту та послуг Інтернету, які можуть бути заблоковані, обмежені чи цензуровані у його регіоні чи країні, змінивши свою IP-адресу та місцезнаходження на сервер VPN. Крім того, користувач може захистити свої особисті дані та онлайн-активність від відстеження, контролю чи розкриття третіми сторонами, такими як постачальники послуг Інтернету, хакери, рекламодавці чи

урядові агентства, приховавши свою реальну IP-адресу та місцезнаходження та шифруючи свої дані.

1.2.2 Принципи та цілі використання VPN

Основні принципи VPN:

Приватність: VPN захищає особисті дані користувача та його онлайн-активність від відстеження, моніторингу або розкриття третіми сторонами, такими як постачальники інтернет-послуг (ISP), хакери, рекламодавці чи урядові агентства. VPN приховує реальну IP-адресу та місцезнаходження користувача, надаючи нові дані від сервера VPN. Також VPN шифрує дані користувача, роблячи їх незчитливими та недоступними для будь-якого, хто перехопить їх. Приватність - одна з найважливіших причин використання VPN, особливо в країнах, де свобода Інтернету та права людини перебувають під загрозою. VPN може допомогти користувачам захистити свою ідентичність, анонімність та конфіденційність онлайн, а також уникнути цензури, спостереження та збирання даних зловмисниками.

Безпека: VPN підвищує безпеку та безпечність користувача при використанні публічних або незахищених мереж, таких як Wi-Fi-точки доступу, які вразливі до атак та порушень. VPN запобігає несанкціонованому доступу до пристрою та даних користувача, а також до приватної мережі та її ресурсів. VPN також захищає користувача від шкідливих веб-сайтів, шкідливих програм, фішингу та інших онлайн-загроз. Безпека - ще одна важлива причина використання VPN, особливо, коли потрібно отримати доступ до чутливої або конфіденційної інформації, такої як банківські рахунки, електронна пошта чи робочі файли. VPN може допомогти користувачам захистити своє з'єднання та дані, а також запобігти крадіжці ідентичності, шахрайству чи кібератакам.

Свобода: VPN дозволяє користувачеві отримати доступ до більшого обсягу інтернет-контенту та послуг, які можуть бути заблоковані, обмежені чи цензуровані у їхньому регіоні чи країні з політичних, юридичних чи культурних

причин. VPN дозволяє користувачеві обходити брандмауери, фільтри та проксі-сервери, а також отримувати доступ до веб-сайтів та додатків, які можуть бути недоступні або недосяжні в інший спосіб. VPN також дозволяє користувачеві насолоджуватися більшим різноманіттям онлайн-контенту та послуг, а також порівнювати ціни та пропозиції з різних регіонів чи країн. Свобода - ще одна важлива причина, чому люди використовують VPN, особливо, коли вони хочуть отримати доступ до розважального контенту, освіти чи комунікаційних опцій, таких як стрімінг, геймінг, соціальні медіа чи месенджери. VPN може допомогти користувачам подолати географічні обмеження та цензуру, а також вивчати та відкривати нові онлайн-можливості.

Забезпечення зв'язку: VPN дозволяє користувачу з'єднуватися з офісами, роботою, віддалено та безпечно, наче вони фізично присутні. VPN дозволяє користувачеві отримати доступ до приватної мережі та її ресурсів, з будь-якого місця та в будь-який час, використовуючи будь-який пристрій. VPN також дозволяє користувачу співпрацювати та спілкуватися з колегами, керівниками, клієнтами та іншими зацікавленими сторонами за допомогою різних інструментів та платформ, таких як електронна пошта, чат, відеоконференції та інше. Забезпечення зв'язку - ще одна важлива причина використання VPN

Отже, основні принципи та цілі використання VPN - це приватність, безпека та свобода. VPN може допомогти користувачам захистити свої особисті дані та онлайн-активність, підвищити їхню безпеку та безпечність, а також отримати доступ до більшого обсягу інтернет-контенту та послуг. VPN також може надати інші переваги, такі як покращення продуктивності, швидкості та надійності користувача, а також зменшення його витрат та впливу на навколишнє середовище. Тому VPN є цінним та корисним інструментом для кожного, хто користується Інтернетом.

1.2.3. Типи VPN та їх технології

Існують різні типи VPN, які використовують різні технології для шифрування та тунелювання даних. Деякі з найпоширеніших типів VPN та їх технології:

OpenVPN: OpenVPN - це відкритий VPN-протокол, який використовує шифрування SSL/TLS для створення безпечного та гнучкого з'єднання VPN. OpenVPN може використовувати як TCP, так і UDP як протокол транспортного рівня, і може працювати на будь-якому порту. OpenVPN також може використовувати різні алгоритми шифрування, такі як AES, Blowfish або Camellia, і різні методи автентифікації, такі як сертифікати, паролі або токени. OpenVPN є одним з найпопулярніших і широко використовуваних VPN-протоколів, оскільки він пропонує високий рівень безпеки, продуктивності та сумісності. OpenVPN може працювати на різних платформах, таких як Windows, Linux, Mac, Android, iOS тощо.

IPSec/IKEv2: IPSec (Internet Protocol Security) - це VPN-протокол, який використовує криптографічні служби безпеки для захисту IP-пакетів. IPSec може працювати в двох режимах: транспортному та тунельному. Транспортний режим шифрує лише навантаження IP-пакета, тоді як тунельний режим шифрує весь IP-пакет. IPSec може використовувати різні алгоритми шифрування, такі як AES, DES або 3DES, і різні методи автентифікації, такі як сертифікати, паролі або токени. IPSec також може використовувати різні протоколи, такі як AH (Authentication Header) або ESP (Encapsulating Security Payload), щоб надавати різні служби безпеки, такі як цілісність, конфіденційність або захист від повторних атак. Часто IPSec використовується разом з IKEv2 (Internet Key Exchange version 2), який є протоколом, що встановлює та управляє асоціаціями безпеки між пірингами VPN. IKEv2 використовує обмін ключів Diffie-Hellman для створення спільного секретного ключа, а також різні алгоритми шифрування та автентифікації, такі як AES, SHA або RSA, для захисту обміну ключами. IPSec/IKEv2 - це VPN-протокол, який пропонує високий рівень безпеки, стабільності та швидкості. IPSec/IKEv2

може працювати на різних платформах, таких як Windows, Linux, Mac, Android, iOS тощо.

WireGuard: WireGuard є новим та експериментальним VPN-протоколом, який спрямований на надання простого, швидкого та безпечного VPN-з'єднання. WireGuard використовує новий криптографічний підхід, що називається Cryptokey Routing, який поєднує криптографію з відкритим ключем і симетричним шифруванням та автентифікацією. WireGuard використовує один UDP-порт для всього VPN-трафіку, а також різні алгоритми шифрування та автентифікації, такі як ChaCha20, Poly1305, або Curve25519 для захисту даних. WireGuard є VPN-протоколом, що пропонує низьку затримку, високу продуктивність та просту конфігурацію. WireGuard може працювати на різних платформах, таких як Windows, Linux, Mac, Android, iOS тощо.

SSTP: SSTP (Secure Socket Tunneling Protocol) є VPN-протоколом, який використовує шифрування SSL/TLS для створення безпечного VPN-з'єднання через TCP-порт 443. SSTP може використовувати різні алгоритми шифрування, такі як AES, RC4, або 3DES, а також різні методи автентифікації, такі як сертифікати, паролі або токени. SSTP є VPN-протоколом, який пропонує високий рівень безпеки та сумісності. SSTP може обходити більшість брандмауерів та проксі, оскільки використовує той самий порт, що і HTTPS. SSTP може працювати на різних платформах, таких як Windows, Linux, Mac, Android, iOS тощо.

L2TP/IPSec: L2TP (Layer 2 Tunneling Protocol) є VPN-протоколом, який створює тунель між піpингами VPN, але не надає жодного шифрування чи автентифікації. L2TP може використовувати як TCP, так і UDP як протокол транспортного рівня, і може працювати на різних портах. Часто L2TP використовується разом з IPSec, який надає шифрування та автентифікацію для тунелю L2TP. L2TP/IPSec може використовувати різні алгоритми шифрування, такі як AES, DES, або 3DES, а також різні методи автентифікації, такі як сертифікати, паролі або токени. L2TP/IPSec є VPN-протоколом, який пропонує помірний рівень безпеки та продуктивності. L2TP/IPSec може працювати на різних платформах, таких як Windows, Linux, Mac, Android, iOS тощо.

PPTP: PPTP (Протокол точка-точка тунелювання) - це старий та застарілий VPN-протокол, який створює тунель між пірами VPN та використовує простий механізм шифрування та аутентифікації, який називається MPPE (Microsoft Point-to-Point Encryption) та MS-CHAP (Microsoft Challenge Handshake Authentication Protocol). PPTP використовує TCP-порт 1723 для керуючого каналу та GRE (Generic Routing Encapsulation) для каналу даних. PPTP може використовувати різні алгоритми шифрування, такі як RC4, та різні методи аутентифікації, такі як паролі або токени. PPTP - це VPN-протокол, який пропонує низький рівень безпеки та продуктивності. PPTP вразливий до різних атак, таких як атаки перебору, словникові атаки чи атаки повторного відтворення. PPTP може працювати на різних платформах, таких як Windows, Linux, Mac, Android, iOS тощо, але не рекомендується для використання.

Загалом, існують різні типи VPN та їх технології, кожна з яких має свої переваги та недоліки. Вибір типу та технології VPN залежить від різних факторів, таких як потреби, вподобання та ресурси користувача, а також характеристики, вимоги та обмеження мережі.

1.2.4 Тенденції розвитку технологій VPN

Технології VPN постійно еволюціонують та вдосконалюються, оскільки вони зіткнулися з новими викликами та можливостями в змінному інтернет-ландшафті. Деякі поточні та майбутні тенденції у технологіях VPN:

Хмарні VPN: Хмарні VPN використовують хмарні обчислювальні платформи, такі як Amazon Web Services, Microsoft Azure чи Google Cloud, для розміщення та управління серверами та інфраструктурою VPN. Хмарні VPN пропонують декілька переваг, таких як масштабованість, надійність, продуктивність та ефективність витрат. Хмарні VPN також можуть інтегруватися з іншими хмарними сервісами, такими як сховища, аналітика або безпека. Хмарні VPN особливо підходять для бізнесу та організацій, які потребують доступу та обміну даними та ресурсами в

кількох місцях та на різних пристроях. Згідно з доповіддю Global Market Insights, ринок хмарних VPN очікується зростати зі складною річною зростаючою ставкою (CAGR) в 17% з 2020 по 2027 рік, досягнувши 54 мільярди доларів до 2027 року.

Мобільні VPN: Мобільні VPN призначені для мобільних пристроїв, таких як смартфони, планшети або ноутбуки, що використовують бездротові мережі, такі як Wi-Fi, мобільний зв'язок чи супутниковий зв'язок. Мобільні VPN надають безпечне та стабільне з'єднання VPN, навіть коли пристрій перемикається між різними мережами або зазнає перебоїв мережі. Мобільні VPN також можуть оптимізувати з'єднання VPN для тривалості роботи батареї пристрою, пропускної здатності та швидкості. Мобільні VPN особливо підходять для користувачів, які потребують доступу до Інтернету у русі, таких як подорожуючі, віддалені працівники чи журналісти. Згідно з доповіддю Allied Market Research, ринок мобільних VPN очікується зростати зі складною річною зростаючою ставкою (CAGR) в 19,7% з 2020 по 2027 рік, досягнувши 2,4 мільярди доларів до 2027 року.

Zero Trust VPNs: Zero Trust VPN - це VPN, які слідують принципу "ніколи не довіряйте, завжди перевіряйте", що означає, що вони не вважають, що будь-який користувач, пристрій або мережа за замовчуванням є надійними і вимагають постійної перевірки та підтвердження їхньої ідентичності та безпеки. Zero Trust VPN використовують різні технології, такі як багаторівнева аутентифікація, відбитки пристроїв, мікросегментація або штучний інтелект, щоб відслідковувати та контролювати доступ та трафік VPN та виявляти та запобігати будь-яким потенційним загрозам чи порушенням. Zero Trust VPN особливо підходять для бізнесу та організацій, які потребують захисту своїх чутливих або конфіденційних даних та ресурсів від внутрішніх та зовнішніх атак. Zero Trust VPN все ще є новим та розвиваючимся концептом, але вже привертає увагу багатьох постачальників VPN та користувачів.

Ці тенденції у технологіях VPN відображають поточні та майбутні напрями розвитку, спрямовані на забезпечення адаптивного, безпечого та оптимізованного з'єднання між пристроями та мережами.

1.3 RADIUS: Роль, Функції та Архітектура

RADIUS (Remote Authentication Dial-In User Service) - це протокол мережевої автентифікації, авторизації та обліку (AAA), який використовується для забезпечення безпеки мережі. RADIUS використовується для централізованого управління доступом до мережі, а також для обліку використання мережевих ресурсів.

Функції RADIUS включають автентифікацію користувачів, авторизацію доступу до мережі та облік використання мережевих ресурсів. RADIUS використовується для забезпечення безпеки мережі, а також для зменшення навантаження на сервери мережі.

Щодо архітектури RADIUS, вона складається з трьох основних компонентів: клієнта, сервера та бази даних. Клієнт - це пристрій, який використовується для доступу до мережі, сервер - це пристрій, який забезпечує автентифікацію користувачів та авторизацію доступу до мережі, а база даних - це місце зберігання інформації про користувачів та їх права доступу до мережі.

1.3.1 Роль RADIUS в мережевих технологіях

RADIUS є ключовим протоколом, що використовується для управління доступом до мережевих ресурсів, особливо в системах з великою кількістю користувачів, де важлива безпека та ефективність автентифікації. Основні аспекти його ролі в мережевих технологіях включають:

Захист мережевих ресурсів:

RADIUS допомагає у забезпеченні безпеки мережі, забезпечуючи контроль над тим, хто та як користується мережевими ресурсами. Він дозволяє централізовано керувати процесом автентифікації та авторизації, що робить систему менш вразливою перед неповними або несанкціонованими доступами.

Спрощення управління користувачами:

Цей протокол дозволяє адміністраторам ефективно керувати доступом користувачів до мережі, створюючи політики, які визначають, хто має доступ до яких ресурсів та які права у них є.

Підтримка мобільності:

RADIUS використовується для управління доступом до мережі для користувачів, які рухаються між різними мережами або місцями, забезпечуючи послідовність автентифікації та авторизації незалежно від місця з'єднання.

Ці аспекти ролі RADIUS в мережевих технологіях підкреслюють його важливість у забезпеченні безпеки, контролю доступу та ефективного управління користувачами у сучасних мережевих середовищах.

1.3.2 Функції RADIUS

Так як і згадувалося раніше, що це протокол мережевої автентифікації, авторизації та обліку – розглянемо кожну з цих функцій більш детально

З'ясуємо що ж таке автентифікація - це процедура встановлення належності користувачеві інформації в системі пред’явленого ним ідентифікатора. Це важливий аспект кібербезпеки, який дозволяє перевірити, що користувач є тим, за кого він себе видає. Основна мета аутентифікації полягає в перевірці правильності ідентифікаційних даних користувача. І ось як проходить сам процес аутентифікації Передача інформації для автентифікації:

- Коли користувач намагається отримати доступ до мережі (наприклад, через VPN), його пристрій або програма передає ім'я користувача та пароль на сервер RADIUS.
- Відбувається перевірка облікових даних. Сервер RADIUS отримує ці дані та порівнює їх із збереженими обліковими записами. Він також може використовувати інші методи, такі як бази даних або LDAP (Lightweight Directory Access Protocol), для перевірки ідентифікаційних даних.

- На основі результатів перевірки, сервер RADIUS повертає відповідь, яка вказує, чи були правильні ідентифікаційні дані. У разі успіху користувач має доступ, а у випадку невдачі – отримує відмову.

Повернемося до авторизації - це спосіб захисту, який визначає права користувачів таким чином, що вони будуть неоднакові. Кожен користувач може мати різні привілеї: деякі можуть увійти в акаунт, надсилати повідомлення, користуватися пристроями, а інші можуть редагувати вміст, або навіть не мати жодних спеціальних прав.

- Після успішної автентифікації сервер RADIUS вирішує, які ресурси та сервіси може використовувати користувач. Це включає визначення прав доступу до мережевих ресурсів на основі внутрішніх політик безпеки та налаштувань, збережених у його конфігурації.
- Якщо користувач успішно пройшов автентифікацію та має необхідні права, сервер RADIUS надає дозвіл на доступ до визначених мережевих ресурсів.
- RADIUS веде облік часу користування ресурсами, обсягу переданих даних та інших параметрів, що дозволяє адміністраторам контролювати та встановлювати обмеження.

1.3.3 Архітектура RADIUS

Архітектура RADIUS складається з трьох основних компонентів:
Клієнт (RADIUS-клієнт):

- Клієнт - це пристрій, який використовується для доступу до мережі (наприклад, маршрутизатор, комутатор або сервер VPN).
- Клієнт ініціює процес аутентифікації та авторизації, надсилаючи запит до сервера RADIUS.

Сервер RADIUS:

- Сервер RADIUS - це центральний компонент, який обробляє запити від клієнтів.

- Він перевіряє вірність ідентифікаційних даних користувача (аутентифікація) та визначає права доступу користувача до мережі (авторизація).
- Сервер також веде облік використання мережевих ресурсів.

База даних RADIUS:

- База даних містить інформацію про користувачів, їх ідентифікаційні дані, права доступу та інші параметри.
- Інформація зберігається у вигляді записів, які включають ім’я користувача, пароль, IP-адресу, групу доступу тощо.

Послідовність дій при підключенні (Рис.1.1):

Клієнт надсилає запит до сервера RADIUS, передаючи ідентифікаційні дані користувача.

Сервер RADIUS перевіряє вірність ідентифікаційних даних.

Якщо користувача успішно аутентифіковано, сервер визначає права доступу користувача.

Сервер також записує інформацію про використання мережевих ресурсів користувачем.

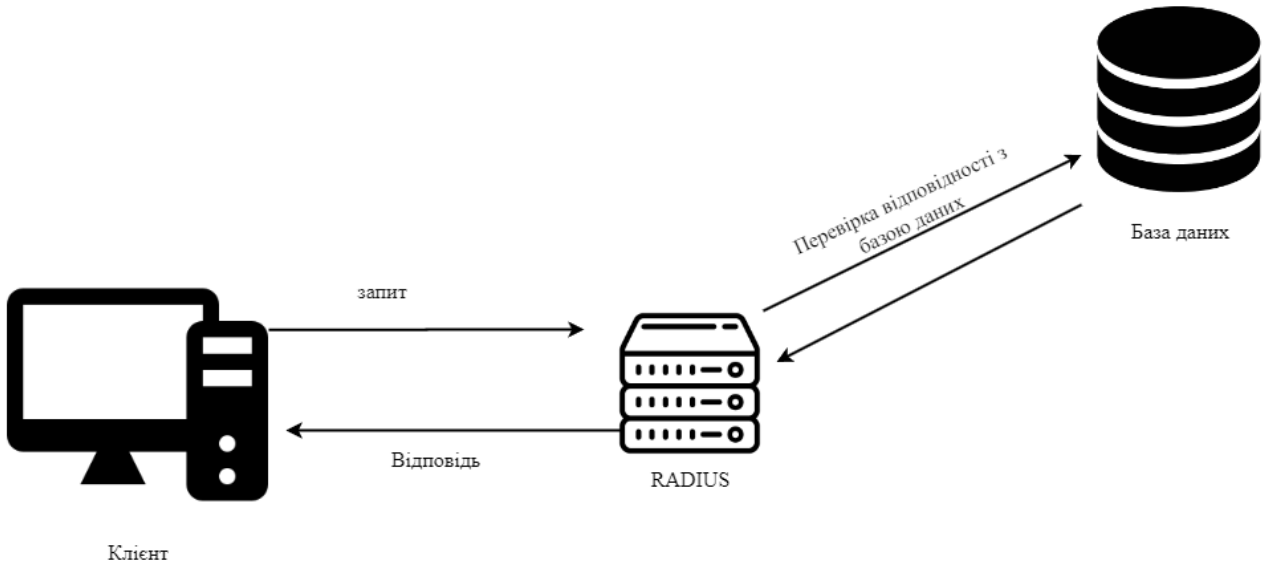

Рис.1.1 Схема роботи RADIUS

Висновки

У цьому розділі я дослідив проблематику та актуальність нагальної потреби в віддаленої роботи. Зріст кількості компаній, які потребують надійної системи для реалізації можливості надавати своїм співробітникам змогу працювати з дому, або будь якого іншого місця. Дослідив користь та недоліки, того, що люди виконують свої робочі обов'язки не знаходячись на робочому місці.

Також мною були розглянуті основні протоколи VPN та принцип роботи цієї технології взагалому. На основі цього, для реалізації свого проекту - я обрав продукт з відкритим програмним кодом та з підтримкою спільноти – SoftEther VPN, також на вибір цього рішення плинуло те, що вона надає змогу в собі налаштувати одразу протоколи SSTP, L2TP/IPSec, OpenVPN, що додає більшої гнучкості в налаштуванні.

І був розглянут протокол RADIUS. Його фундоментальні поняття та принципи роботи, за допомогою якого і буде посилюватися захист підключень у вигляді аутентифікації та авторизації.

РОЗДІЛ 2
АУТЕНТИФІКАЦІЯ.
АНАЛІЗ СУЧАСНИХ МЕТОДІВ АУТЕНТИФІКАЦІЇ

Аутентифікація – це процес перевірки і підтвердження ідентичності особи, користувача або пристрою для надання доступу до певних ресурсів, системи чи послуги. Основна мета аутентифікації полягає в забезпеченні безпеки шляхом перевірки того, що особа або пристрій, який намагається отримати доступ, дійсно є тим, за кого себе видає.

Елементи аутентифікації:

- Ідентифікація: Це перший етап, де користувач або система надає інформацію, яка дозволяє розпізнати ідентичність. Наприклад, ім'я користувача, електронна адреса, номери або будь-які інші унікальні параметри.
- Автентифікація: Це процес перевірки та підтвердження ідентичності, поданих на етапі ідентифікації. Включає в себе використання різних методів, таких як пароль, біометрія, токен або код доступу, для підтвердження особи або пристрою.
- Авторизація: Це наступний крок після успішної аутентифікації. Після підтвердження ідентичності користувача/пристрою надається доступ до визначених ресурсів, систем або послуг відповідно до прав доступу, які були встановлені.

Зобразити у вигляді блок-схеми принцип процедури автентифікації можно як на малюнку 2.1

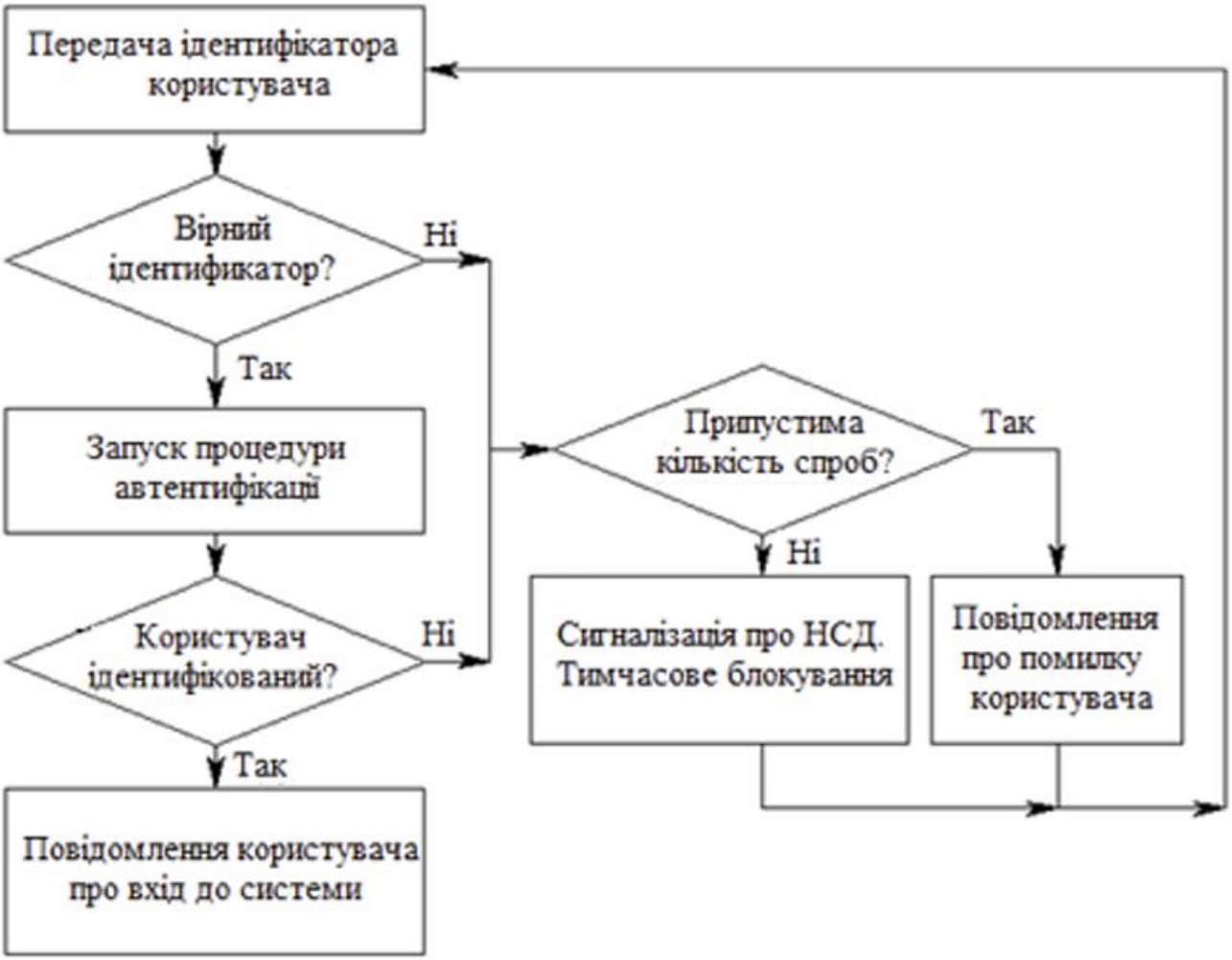

Рисунок 2.1 – Класична процедура ідентифікації та автентифікації

Традиційну автентифікацію за допомогою пароля називають ще однофакторною або слабкою. Оскільки за наявності певних ресурсів перехоплення або підбір пароля є справою часу. Таким чином часто виникає необхідність використовувати сильну або багатофакторну автентифікацію - на основі двох чи більше факторів. В цьому випадку для автентифікації використостовується не лише інформація відома користувачеві, а й додаткові фактори.

Використання багатофакторної автентифікації для підтвердження особи базується на передумові, що неавторизований користувач навряд чи зможе надати фактори необхідні для доступу. Якщо в спробі автентифікації хоча б один з компонентів відсутній або вказаний невірно, то ідентифікація користувача не встановлюється з достатнім ступенем впевненості та доступу до об’єкту

(наприклад, до будівлі або даних), захищеному багатофакторною автентифікацією, залишається заблокованим.

Доцільність використання багатофакторної автентифікації можна відстежити за таблицею 2.1

Таблиця 2.1 – Доцільність використання певного способу автентифікації

Ступінь ризику	**Приклад використання**	**Способи автентифікації**
Низький — наслідки зламу призведуть до незначної шкоди	Нова реєстрація на майданчику окремого користувача	Досить використати складний багаторазовий пароль
Середній — у разі крадіжки пароля збитки будуть відчутними, але не критичними	Здійснення покупок онлайн за допомогою банківських карток, пов'язаних з акаунтами в інтернет-магазинах	Потрібен додатковий захист у вигляді двоетапної або двофакторної автентифікації
Високий — витік інформації може призвести до колосальних негативних наслідків	Використанням банківських систем для обробки транзакцій великого обсягу грошових переказів між банками	Багатофакторна автентифікація є обов’язковою

2.1 Методи багатофакторної автентифікації

Фактори багатофакторної автентифікації можуть включати:

- фізичний об'єкт яким володіє користувач, такий як накопичувач з секретним токеном, банківська карта, ключ і т. д.
- якийсь секрет, відомий користувачеві, такий як пароль, PIK-код, і т. д.
- деякі фізичні характеристики користувача (біометричні дані), такі як відбиток пальця, голос, підпис, і т. д.
- ваше місцезнаходження, підключення до певної комп’ютерної мережі або інші способи для визначення місця розташування.

Роздивимося кожен фактор більш детально.

Фактори знання є одними з найпоширеніших методів автентифікації, де користувач повинен підтвердити знання певного секрету для входу. Пароль, що складається зі слова або рядка символів, - найчастіший засіб автентифікації, і його

часто використовують у методах багатофакторної перевірки. Існують різноманітні варіанти, від довгих кодових фраз до коротких числових ПІН-кодів, таких як для доступу до банкоматів. Зазвичай очікується, що користувачі зберігають паролі у своїй пам'яті. Проте, на практиці, це може бути викликом через потребу запам'ятовування багатьох паролів, тому деякі користувачі вдаються до використання менеджерів паролів або зручних схем для їх збереження. Такий фактор автентифікації відіграє важливу роль у забезпеченні безпеки, але якщо не використовувати надійні та унікальні паролі для кожного облікового запису, тобто один пароль для всіх - може стати слабкою ланкою в безпеці.

Фактори володіння (щось чим володіє користувач і тільки користувач) використовувалися для автентифікації протягом століть, у вигляді ключа до замків. Основний принцип полягає в тому, що ключ уособлює секрет, який поділяється між замком і ключем, цей же принцип лежить в основі автентифікації за фактором володіння в комп'ютерних системах. Токени - це приклад фактора володіння, вони бувають різних типів: Відключені токени - не мають зв’язку до клієнтського комп'ютера. Зазвичай вони використовують вбудований екран для відображення згенерованих даних автентифікації, які користувач вводить вручну. Підключенні жетони - це пристрої, які фізично підключені до використовуваного комп'ютера. Ці пристрої автоматично передають дані на комп’ютер. Існує безліч різних типів таких жетонів, включаючи пристрої читання карток, бездротові жетони. Програмний токен - це тип пристрою двофакторної автентифікації, що може використовуватися для авторизації при використання комп'ютерних сервісів. Програмні токени можуть зберігатися на будь якому електронному пристрої, такому як настільний комп'ютер, ноутбук, КПК або мобільний телефон, але, відмінно від апаратних, можуть бути дубльовані. Однак апаратні токени, зберігаючи дані на виділеному пристрої, залишаються більш безпечними. Такий підхід забезпечує високий рівень безпеки та доступності для користувачів, забезпечуючи гнучкість у виборі та застосуванні токенів залежно від потреб та умов. Ці фактори володіння є важливою складовою сучасних систем безпеки, що надають надійний шлях аутентифікації користувачів у цифровому світі.

Фактор власності. Значущий аспект у сфері аутентифікації – фактор власності, що визначається фізичними особливостями суб'єкта. Це може охоплювати портрет, відбиток пальця, долоні, голос або особливість обличчя. Цей метод автентифікації спрощує життя суб'єкта: немає необхідності запам'ятовувати паролі чи носити з собою пристрої для автентифікації. Проте біометрична система має вимогу до високої чутливості для впізнавання автентифікованого користувача та відхилення підроблення з використанням схожих біометричних параметрів. Ця технологія є витратною, але, незважаючи на це, біометрика залишається перспективною через свою унікальність та потенціал в удосконаленні систем аутентифікації.

Фактор місцезнаходження, що стосується фізичного розташування користувача, стає все більш поширеним, як четвертий елемент в аутентифікації. Коли користувач працює у власній корпоративній мережі, він може увійти, використовуючи лише пін-код. Однак, при підключенні до іншої мережі може знадобитися додатковий етап підтвердження особи. Управління доступом до мережі працює схожим чином, прив'язуючи рівень доступу до конкретної мережі, наприклад, Wi-Fi або провідного з'єднання. Також треба не забувати про ненадійниість мереж Wi-Fi, то ж краще не використовувати їх для роботи з конфедиційними даними. Це дозволяє користувачам вільно переміщатися між офісами та одержувати однаковий рівень доступу до мережі у будь-якому з них.

2.2 Двофакторна аутентифікація (2FA): методи, переваги та недоліки

Основна ідея двофакторної автентифікації полягає у використанні не тільки стандартної комбінації «логін і пароль», але і додаткового елемента безпеки, який називається другим фактором. Щоб отримати доступ до облікового запису чи інформації, потрібно підтвердити другий фактор.

Простий приклад такої системи - це використання банкомату для отримання готівки. Щоб отримати гроші, потрібна карта, яка маєте тільки ви, та PIN-код, який відомий лише вам. Якщо хтось має вашу карту, він не зможе отримати гроші без PIN-коду, і навпаки. Такий же принцип використовується при доступі до

соціальних мереж або пошти - це вимагає не тільки пароль, але й додатковий елемент підтвердження, що підвищує рівень безпеки. Багатофакторна автентифікація вимагає кількох методів перевірки особи для входу або проведення важливих операцій. Окрім паролів, другий фактор може включати унікальні токени, які змінюються кожні 30 секунд, або біометричні дані, наприклад, відбиток пальця.

2.2.1 Дослідження та порівняння методів двофакторної автентифікації

Першим фактором є введення логіна та пароля, а другим фактором може бути один із призначених двофакторних методів аутентифікації, які ми розглянемо після оцінки їх плюсів і мінусів.

Принцип роботи SMS-підтвердження простий: після введення логіна та пароля на ваш мобільний номер приходить текстове повідомлення з кодом, який необхідно ввести, щоб отримати доступ до свого облікового запису. Наступного разу ви отримаєте інший SMS-код, дійсний лише для поточного сеансу.

Переваги:

- Генерація нових кодів при кожному вході. Якщо хтось перехопить ваші логін та пароль, без коду він не зможе нічого зробити.
- Прив'язка до телефонного номеру. Вхід без вашого телефону неможливий.

Недоліки:

- Відсутність сигналу мережі може перешкодити вам у вході.
- Існує теоретична можливість підміни номера через дії оператора чи працівників салонів зв'язку.
- Якщо ви отримуєте коди на одному пристрої і з нього же намагаєтесь авторизуватися (наприклад, смартфоні), то захист перестає бути двофакторним.

Універсальний другий фактор (U2F) - це відкритий стандарт, що використовується з USB-пристроями, пристроями NFC та смарт-картками для автентифікації. Для проведення процедури автентифікації потрібно просто

підключити його (для USB-ключів), натиснути (для пристроїв NFC) або провести (для смарт-карт) біля обладнання.

Переваги:

- Ключ U2F - справжній фізичний фактор, надійний у порівнянні з SMS-кодами, які можна перехопити. Він також захищений від фішингу, оскільки працює лише з певними сайтами, де ви зареєстровані. Це один із найбільш надійних методів 2FA.

Недоліки:

- Оскільки технологія U2F ще не настільки поширена, NFC-ключі підтримуються тільки на Android-пристроях, тоді як USB-ключі працюють головним чином у браузері. Ключі U2F також мають певну вартість, часто від 10 до 20 доларів, але ціна може відрізнятися в залежності від їхньої якості.

Додатки-автентифікатори. Вони подібні до SMS-кодів, але вони не генерують коди через SMS, а використовують спеціальні додатки (наприклад, Google Authenticator, Authy). При налаштуванні ви отримуєте початковий ключ, зазвичай у вигляді QR-коду, за допомогою якого генеруються одноразові паролі криптографічними алгоритмами. Ці паролі активні від 30 до 60 секунд, забезпечуючи стабільний рівень безпеки.

Важливим аспектом цього підходу є те, що навіть якщо зловмисник перехопить десятки чи сотні попередніх кодів, йому буде важко передбачити наступний одноразовий пароль. Це забезпечує високий рівень захисту облікових записів і особистої інформації від несанкціонованого доступу.

Такий підхід є важливим кроком на шляху підвищення безпеки персональних даних, оскільки виступає додатковим бар’єром між користувачами та можливими загрозами в мережі. Його простота використання та надійність роблять його одним із найзручніших способів запобігання несанкціонованому доступу.

Переваги:

- Автентифікатор працює незалежно від мережі, вимагаючи доступ до Інтернету лише при налаштуванні.

- Підтримка декількох акаунтів в одному автентифікаторі.

Недоліки:

- Якщо хакерам вдасться отримати первинний ключ на вашому пристрої або шляхом вторгнення до сервера, вони зможуть створювати паролі у подальшому.
- Коли автентифікатор використовується на тому ж пристрої, з якого проводиться вхід, втрачається двухфакторність.

Перевірка входу через мобільні додатки. Цей метод автентифікації - своєрідна комбінація попередніх. Замість запитів кодів або одноразових паролів, ви маєте підтверджувати вхід через свій мобільний додаток, встановлений на пристрої. На пристрої зберігається особливий ключ, який перевіряється кожного разу при спробі входу. Цей метод застосовується в месенджерах та в різних онлайн-іграх. Наприклад, при вході в ваш обліковий запис через веб-версію, ви вводите логін і пароль. Після цього на ваш мобільний телефон приходить повідомлення із запитом на вхід, яке, підтвердивши, відкриває вашу сторінку у браузері.

Переваги:

- Не потрібно нічого вводити при вході.
- Підтримка декількох акаунтів в одному додатку.

Недоліки:

- Якщо зловмисники перехоплять приватний ключ, вони зможуть видавати себе за вас.
- Сенс двофакторної автентифікації втрачається при використанні одного і того ж пристрою для входу.

Апаратні токени. Найбільш безпечним методом двофакторної автентифікації є фізичні (або апаратні) токени. Вони є самостійними пристроями, які не втрачають своєї двохфакторної властивості навіть у разі збоїв, на відміну від інших способів, що були згадані раніше. Зазвичай вони мають форму USB-флешек з вбудованим процесором, який створює криптографічні ключі, які автоматично вводяться, коли

токен підключається до комп’ютера. Ключ вибирається в залежності від сервісу, до якого потрібен доступ. Наприклад, Google радить користуватися токенами стандарту FIDO U2F, які можна придбати за ціною від 6 доларів без врахування вартості доставки.

Переваги:

- Непотрібні зайві додатки та SMS-повідомлення.
- Повністю незалежний пристрій.

Недоліки:

- Потрібно купувати окремо.
- Підтримується не у всіх сервісах.
- При автентифікації на декількох акаунтів, треба мати під кожний – окремий токен

Біометричні дані. Останнім часом значно зріс інтерес до тематики біометричної аутентифікації особи. Це пов'язано із посиленням вимог до надійності аутентифікації, а також зручності аутентифікації осіб. Також зростає доступність за ціною та кількість мобільних терміналів, які можуть бути потенційно обладнані сканерами біометричних даних. Вони є доволі безпечними, тому що самим фактором для отримання доступу – стає сама людина, і вже без її фізичної присутності скомпроментувати та пройти автентифікацію – бдуде неможливо.

Елементами такої ідентифікації стають очі людини, її відбитки пальців, розрвзнпання голосу чи ходи та ще деякі фактори, які притаманні лише одній людині і які неможливо з точністю відтворити.

Переваги:

- Високий рівень безпеки, оскільки потрібно більше, ніж просто знання чи володіння чимось.
- Менше можливостей для перехоплення або втрати фактора аутентифікації.

Недоліки:

- Технологія біометрії може бути не завжди доступна або не дуже точною.
- Найбільша вартість в реалізації та інструментах виявлення ідентичності

- Проблеми з приватністю через зберігання біометричних даних.
- Можливість втрати фактора через травми, опіки, тощо.

2.3 Роль штучного інтелекту (AI) у вдосконаленні аутентифікації

Штучний інтелект (ШІ, він же artificial intelligence - AI) дедалі стає важливішим інструментом як у повсядневному житті, так і у вирішенні багатьох задач в тому числі і комерційних. Аутентифікацію це теж не оминуло і штучний інтелект вплинув на вдосконалення методів аутентифікації. AI може використовуватися для покращення різних методів аутентифікації, забезпечуючи більшу безпеку та ефективність.

Одним з методів аутентифікації, який використовує AI, є аналіз поведінки користувача. Цей метод полягає в тому, що система аналізує унікальні характеристики поведінки користувача, такі як спосіб набору тексту, рухи миші, спосіб вирішення задач тощо. Це допомагає відрізняти справжніх користувачів від зловмисників, які можуть спробувати використати чужі дані для входу в систему. Аналіз поведінки користувача може бути використаний як додатковий фактор аутентифікації або як спосіб виявлення аномалій та підозрілої діяльності.

Іншим методом аутентифікації, який використовує AI, є адаптивна аутентифікація. Цей метод полягає в тому, що система змінює рівень складності аутентифікації в залежності від обставин, таких як місцезнаходження, час, пристрій, ризик тощо. Наприклад, якщо система виявляє, що користувач намагається увійти з незвичного місця або часу, вона може запросити додаткові фактори аутентифікації, такі як одноразовий пароль, біометричний сканер, токен тощо. Адаптивна аутентифікація дозволяє забезпечити оптимальний баланс між безпекою та зручністю для користувачів.

Ще одним методом аутентифікації, який використовує AI, є удосконалення біометричних технологій. Біометрична аутентифікація - це метод, який використовує фізичні або поведінкові особливості користувача, такі як відбитки пальців, обличчя, голос, сітківка ока тощо. Ці особливості є унікальними для кожної особи і важко підробити або вкрасти. AI допомагає покращити точність та

надійність біометричних систем, використовуючи розпізнавання образів, голосу, мови тощо. Біометрична аутентифікація може бути використана як самостійний або додатковий фактор аутентифікації.

Нарешті, AI може використовуватися для моделювання загроз та покращення методів аутентифікації. За допомогою AI можна створювати моделі та симуляції потенційних загроз безпеці, щоб виявити слабкі місця в системах аутентифікації та запропонувати способи їх усунення. Штучний інтелект також може використовувати штучні нейронні мережі, які є складними математичними моделями, які можуть навчатися з даних і виконувати різні функції. Штучні нейронні мережі можна використовувати для створення одноразових паролів, шифрування даних, виявлення аномалій, класифікації користувачів і підвищення продуктивності та надійності систем автентифікації.

Підсумовуючи, штучний інтелект є потужним інструментом для покращення перевірки особи, який може забезпечити більш точну, гнучку, швидку та безпечну автентифікацію користувача. AI може використовуватися для покращення різних методів аутентифікації, таких як аналіз поведінки користувача, адаптивна аутентифікація, удосконалення біометричних технологій, моделювання загроз тощо. AI може допомогти забезпечити більш адаптивний підхід до захисту даних та ресурсів.

2.4 Прогрес та майбутні перспективи в області аутентифікації

Наразі аутентифікація активно еволюціонує, найбільше можна виділити кілька напрямів розвитку – це:

Біометрична. Все більше і більше людей користуються саме таким методом аутентифікації. Насамперед можна бачити це у мобільних телефонах, коли майже кожний смартфон обладнаний сканером відбика пальця, а також розпізнованням лиця. Зручність використання та набуття популярності у мас – мотивую розробляти більш іноваційні, надійні та продуктивні рішення для біометричної аутентифікації.

Тобто окрім вишенаведенних прикладів розпізнавання людини по її унікальних чертах – не усе, розробляються і впроваджуються такі методи як

розпізнавання вен пальців: цей метод використовує інфрачервоне світло для сканування унікального візерунку вен пальців користувача. Цей метод має переваги перед розпізнаванням відбитків пальців, оскільки він не піддається зовнішнім впливам, таким як бруд, пот, різки, опіки, тощо. Крім того, він має високий рівень безпеки, оскільки він не може бути відтворений без живого пальця.

Розпізнавання серцевого ритму: цей метод використовує електрокардіограму (ЕКГ) для вимірювання унікального електричного сигналу, який випромінює серце користувача. Цей метод має переваги перед розпізнаванням обличчя або голосу, оскільки він не залежить від освітлення, шуму, виразу обличчя, тона голосу тощо. І для нього теж потрібне тільки серце, яке б'ється.

Розпізнавання запаху: цей метод використовує спектрометр для аналізу унікального хімічного складу поту користувача.

Та це не весь спектр розвитку, куди прямує еволюція біометричної аутентифікації. Тому що в людини багато унікальних, непритаманним іншим людям, властивостей

Пасивна аутентифікація. Частково її теж можна віднести до бометричної ідентичності, але це вже більш складний процесс, який орієнтується не на якусь окрему частину або властивість людини, а на комплекс дій, якою вона керується. Здебільше схоже на ідентифікацію можливості мислити, на мозок користувача.

І ось приклади саме пасивної аутентифікації:

Наприклад Mastercard використовує технологію NuData Security, яка аналізує поведінку користувача, таку як швидкість набору, натиск на екран, кут нахилу пристрою тощо, для підтвердження його особи. Так і в наших банках, а саме Приватбанк використовуючи машинне навчання, платформа NuDetect створює профайл клієнта, в якому збираються понад 300 його унікальних параметрів. За цими параметрами платформа в режимі реального часу визначає, чи є користувач справжнім клієнтом або ж це шахраєм Платформа NuDetect була розроблена компанією NuData Security, яку придбав Mastercard у 2017 році. Mastercard розвиває свої зобов'язання щодо посилення захисту в цифровому просторі, інтегруючи NuData у свій і без того надійний набір продуктів із безпеки та захисту від

шахрайства. Це придбання також зміцнило зусилля компанії щодо забезпечення безпеки й аутентифікації на рівні пристроїв, забезпечуючи в режимі реального часу взаємодію між емітентами, торговцями та процесинговими компаніями.

Освітні та навчальні платформи: деякі освітні та навчальні платформи використовують пасивну аутентифікацію для перевірки ідентичності та академічної чесності своїх студентів. Наприклад, Proctorio використовує технологію Keystroke Dynamics, яка аналізує унікальний стиль набору тексту студента, для підтвердження його особи. Це дозволяє запобігати плагіату та шахрайству під час онлайн-тестування.

Медичні та охоронні послуги: деякі медичні та охоронні послуги використовують пасивну аутентифікацію для захисту та доступу до конфіденційної інформації своїх пацієнтів та клієнтів. Наприклад, BioConnect використовує технологію Behavioral Biometrics, яка аналізує різні аспекти поведінки користувача, такі як рухи очей, жести, постава тощо, для підтвердження його особи.

Не стоїть на місці розвиток стандартів та протоколів безбеки, які сприяють розребленню методів аутентифікації, які є найдійнішими та уніфікованішими.

Наприклад FIDO Alliance - це відкрите промислове об’єднання, яке було запущене у лютому 2013 року з метою розробки та просування стандартів аутентифікації, які допомагають зменшити залежність світу від паролів. FIDO Alliance складається з більш ніж 250 членів, серед яких великі компанії, такі як Google, Microsoft, Samsung, PayPal, Mastercard, Visa, Intel, Lenovo та інші.

FIDO Alliance розробила три основні стандарти аутентифікації: UAF (Universal Authentication Framework), U2F (Universal Second Factor) та FIDO2. Ці стандарти базуються на криптографії з відкритим ключем та біометриці, що дозволяє користувачам використовувати фізичні або поведінкові особливості, такі як обличчя, відбитки пальців, голос, сітківка ока тощо, для підтвердження своєї особи. Ці стандарти забезпечують більшу безпеку, зручність та приватність для користувачів, оскільки вони не потребують вводу, зберігання або передачі паролів.

FIDO Alliance вплинула на розвиток аутентифікації, оскільки вона запропонувала альтернативу паролям, які є основною причиною більшості

кібератак та порушень даних. FIDO Alliance також сприяла створенню більш уніфікованих та сумісних методів аутентифікації, які можуть працювати на різних пристроях, платформах та сервісах. FIDO Alliance також підтримує інновації та дослідження в галузі аутентифікації, співпрацюючи з академічними, державними та промисловими партнерами.

У майбутньому можливе поєднання цих напрямків для створення ще більш надійних та зручних методів аутентифікації. За допомогою постійних досліджень та інновацій у сфері кібербезпеки, аутентифікація продовжуватиме розвиватися, а також ставати більш інтегрованою і невидимою для кінцевого користувача, забезпечуючи водночас високий рівень захисту персональних даних та інформаційних ресурсів.

Висновки

З огляду на дослідження, проведені з автентифікації особи, можна відзначити важливі аспекти та перспективи в цій галузі. Дослідницький розділ містить аналіз різних методів аутентифікації від однофакторної до біометричної, включаючи принципи їх роботи, переваги та недоліки.

Однофакторна автентифікація, хоча й проста, має значні обмеження безпеки, оскільки вона покладається на єдиний механізм перевірки. Двофакторна автентифікація забезпечує вищий рівень захисту завдяки використанню комбінації факторів. Хоча біометрична автентифікація ефективна для ідентифікації осіб, вона має свої проблеми з конфіденційністю та ймовірністю підробки.

Одним із майбутніх напрямків у цій галузі є застосування штучного інтелекту (AI), який може покращити методи автентифікації шляхом аналізу поведінки користувачів і виявлення аномалій. Не менш важливою є розробка стандартів, таких як FIDO Alliance, який допомагає створити уніфіковану систему аутентифікації.

Отже не існує універсального методу аутентифікації, який би був ідеальним для всіх ситуацій. Різні методи мають свої переваги та обмеження, але комбінація

різних підходів може забезпечити ліпший рівень безпеки. Для подальшого розвитку області аутентифікації важливо продовжувати дослідження та впроваджувати інновації, забезпечуючи при цьому як надійність, так і зручність використання для користувачів.

РОЗДІЛ 3
ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОТОТИПУ СИСТЕМИ ДВОФАКТОРНОЇ АУТЕНТИФІКАЦІЇ НА ОСНОВІ МЕСЕНДЖЕРА TELEGRAM ДЛЯ КОРПОРАТИВНИХ VPN

Для початку треба намалювати загальний принцип роботи системи та описати її.

Ось як повинно поктапно проходити підключення до VPN з боку користувача

а) Заходить до программи за допомогою якої створюється VPN тунель

б) Вводить свої данні та данні сервера, до якого потрібно побудувати тунель

в) Починається з'єднання з сервером (очікування)

г) Приходить повідомлення до месенджеру за особистим id користувача

д) Користувач надсилає відповідь, що це саме він і бажає продовжити вводячі число з зашифрованого малюнка

е) Мессенджер відповідає, що авторизація пройшла вдало

є) Відбувається з'єдання з серевром

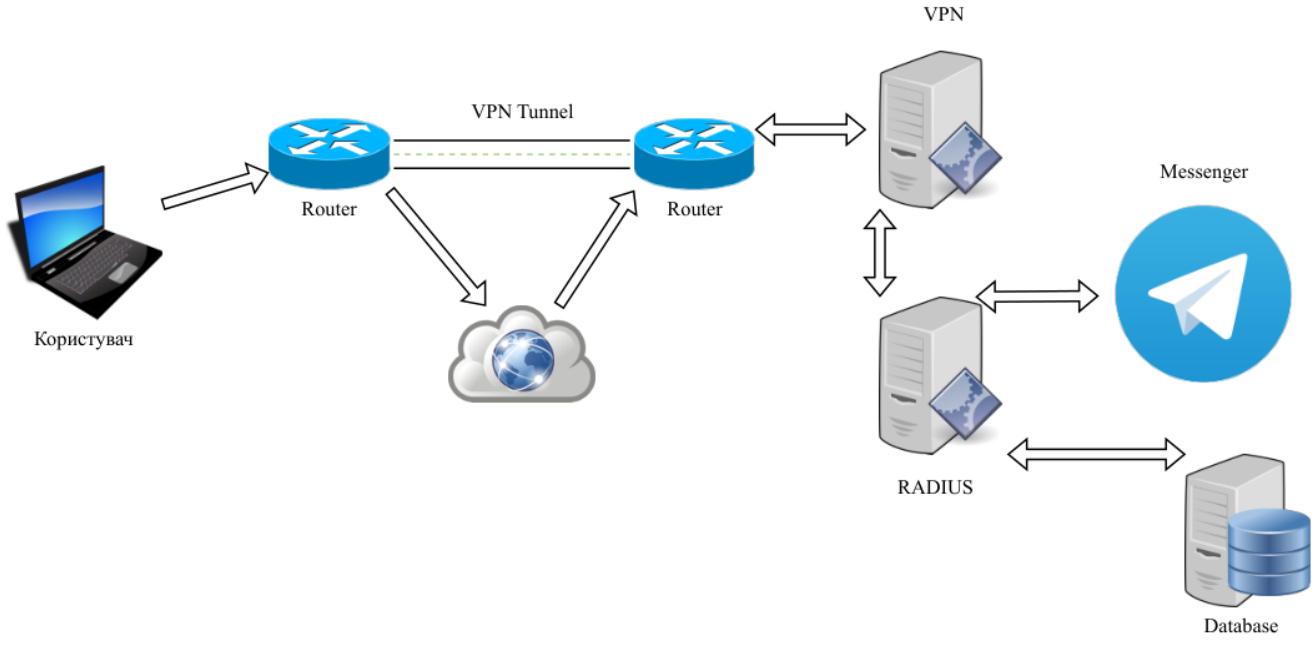

Рисунок 3.1 Схема роботи запропонованої системи

Тобто нам треба налаштувати сервери VPN та RADIUS

Спершу оберемо платформу, на якій будемо реалізовувати всі наші рішення.

Так як більшість серверів – це UNIX-подібні операційні системи, а також вони в більшості своїй безкоштовні – мій вибір теж пав на них.

Але їх теж велика кількість дистрибутивів, можна виділити 3 основні класи/родини:

- системи родини Red Hat (включно із CentOS і Fedora)
- системи родини SUSE (включно з openSUSE)
- системи родини Debian (включно з Ubuntu і Linux Mint)

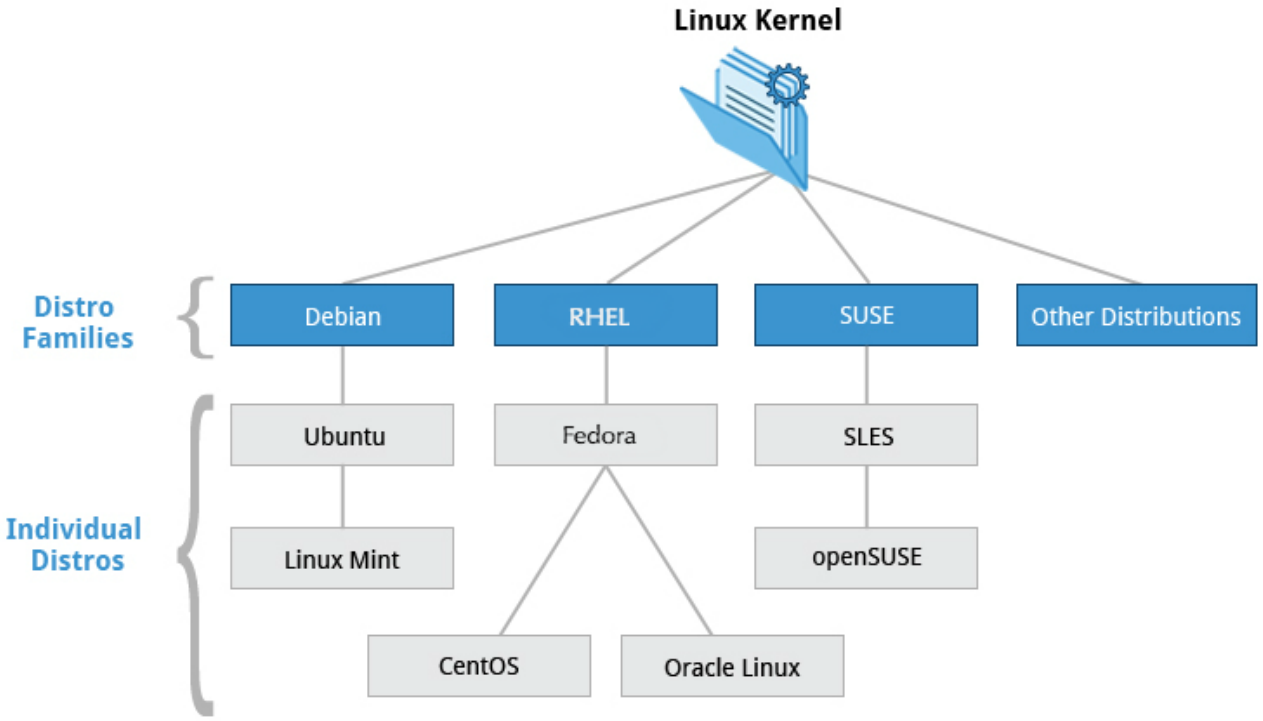

Рисунок 3.2 Основні родини UNIX

Розглянемо трохи кожний з них

Red Hat Enterprise Linux (RHEL) очолює родину, до якої входять CentOS, CentOS Stream, Fedora та Oracle Linux.

Fedora тісно співпрацює із RHEL і містить значно більше програмного забезпечення, ніж корпоративна версія Red Hat. Однією з причин цього є те, що в

розробці Fedora бере участь різноманітна спільнота, з багатьма учасниками, які не працюють на Red Hat. Крім того, він використовується як тестова платформа для майбутніх випусків RHEL.

Ми використовуватимемо CentOS Stream і CentOS частіше для демонстрацій і лабораторних робіт, оскільки вони безплатні для кінцевого користувача, а цикл випуску довший, ніж у Fedora, яка випускає нову версію приблизно кожні шість місяців.

Базова версія CentOS також практично ідентична RHEL – найпопулярнішому дистрибутиву Linux у корпоративних середовищах. Однак CentOS 8 не має запланованих оновлень після 2021 року.

Деякі ключові факти про родину дистрибутивів Red Hat:

- Fedora – передова платформа тестування для RHEL.
- CentOS є близьким клоном RHEL, тоді як Oracle Linux є переважно копією з деякими змінами (фактично CentOS є частиною Red Hat з 2014 року).
- Виправлене ядро версії 3.10 використовується в RHEL/CentOS 7, тоді як версія 4.18 використовується в RHEL/CentOS 8.
- Підтримує такі апаратні платформи, як Intel x86, Arm, Itanium, PowerPC і IBM System z.
- Використовує yum і пакети yum на основі dnf RPM (детальніше розглянемо пізніше) для інсталяції, оновлення та видалення пакетів у системі.
- RHEL широко застосовують на підприємствах, які ведуть власні системи.

Відносини між SUSE (SUSE Linux Enterprise Server, або SLES) і openSUSE подібні до тих, що описані між RHEL, CentOS і Fedora.

OpenSUSE використовується як еталонний дистрибутив для родини SUSE, оскільки він доступний для кінцевих користувачів безплатно. Два продукти надзвичайно схожі, матеріал, який охоплює openSUSE, зазвичай можна застосувати до SLES без проблем.

Деякі ключові факти про родину SUSE:

- SUSE Linux Enterprise Server (SLES) є першоджерелом для openSUSE.
- Версія ядра 4.12 використовується в openSUSE Leap 15.
- Він використовує менеджер пакетів zypper на основі RPM (детальніше розглянемо пізніше) для інсталяції, оновлення та видалення пакетів у системі.
- Він містить програму YaST (Yet Another Setup Tool) для цілей системного адміністрування.
- SLES широко застосовують у роздрібній торгівлі та багатьох інших секторах.

Дистрибутив Debian є першоджерелом для кількох інших дистрибутивів, зокрема Ubuntu. У свою чергу Ubuntu є першоджерелом для Linux Mint і ряду інших дистрибутивів. Його зазвичай застосовують як на серверах, так і на настільних комп'ютерах. Debian – це справжній проєкт спільноти з відкритим вихідним кодом (не належить жодній корпорації), який приділяє значну увагу стабільності.

Debian надає своїм користувачам найбільше та найповніше сховище програмного забезпечення з усіх дистрибутивів Linux.

Мета Ubuntu – забезпечення компромісу між довготривалою стабільністю і простотою використання. Оскільки Ubuntu отримує більшість пакетів зі стабільної гілки Debian, він також має доступ до дуже великого сховища програмного забезпечення.

Деякі ключові факти про родину Debian:

- Родина Debian є першоджерелом для Ubuntu, а Ubuntu – для Linux Mint та інших.
- Версія ядра 5.8 використовується в Ubuntu 20.04 LTS.
- Він використовує менеджер пакетів APT на основі DPKG (за допомогою apt, apt-get, apt-cache тощо, які детальніше розглянемо пізніше) для інсталяції, оновлення та видалення пакетів у системі.

- Ubuntu широко застосовують для хмарних рішень.
- Хоча Ubuntu побудовано на основі Debian і базується на GNOME під капотом, він візуально відрізняється від інтерфейсу стандартного Debian, а також інших дистрибутивів.

Зважаючи на ций короткий огляд на дистрибутиви – я вирішив використуовувати у своїй роботі диструбутив з родини Debian, а саме Ubuntu LTS (довгострокова підтримка) , так як не ній значно більше програмних рішень, вона біль безпечна, і її будуть підтримувати в актуальній формі, томущо активно підтримується спільнотою.

3.1 Встановлення та налаштування серверу

Почнемо побудову нашої системи з розгортання серверу.

Першочергово нам треба завантажити образ операційної системи з сайту Ubuntu

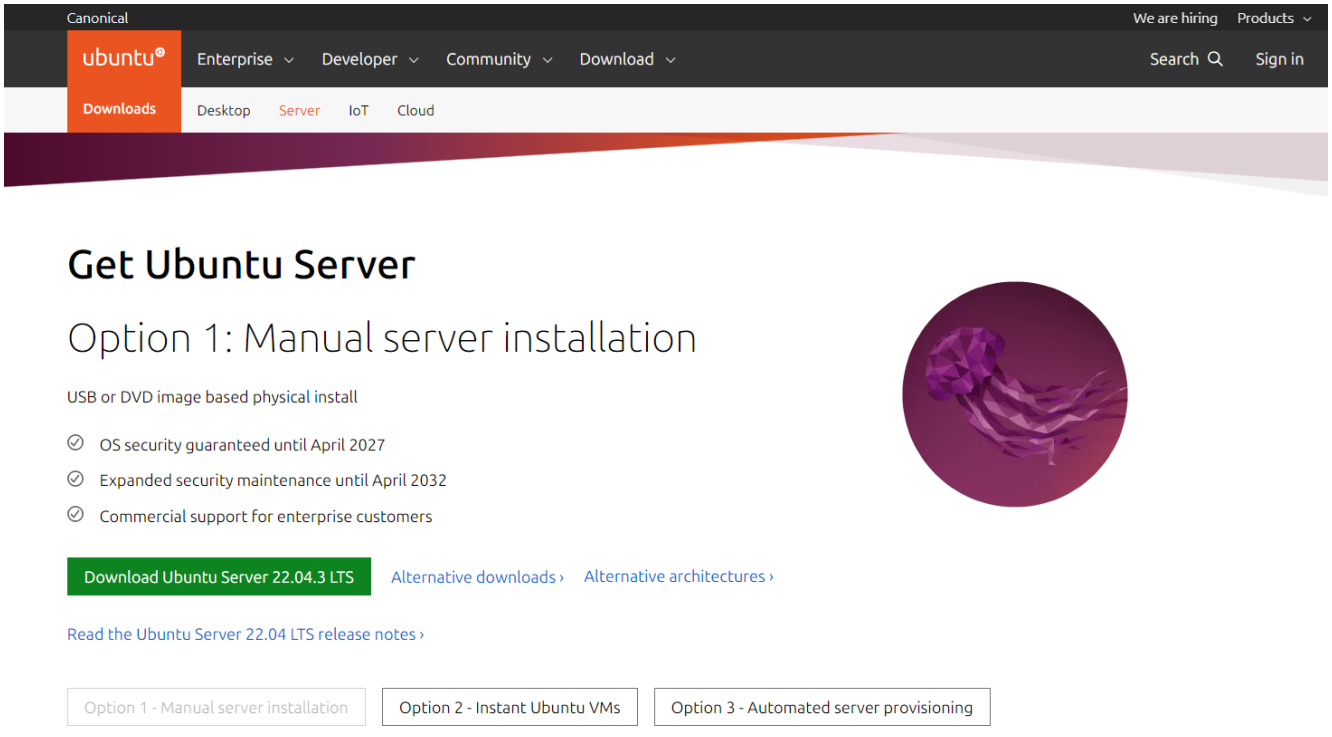

Рисунок 3.3 Офіційний сайт Ubuntu

Так як я не маю серверу фізичного, на якому мав би можливість розгорнути всі потрібні сервіси – я буду використовувати безкоштовну віртуальну машину від компанії Oracle – VirtualBox, для цього завантажимо її та встановимо.

Рисунок 3.4 Офіційний сайт VirtualBox

Вже в самому VirtualBox створюємо віртуальну машину, називаємо її VPN та вказуємо шлях до iso образу операційної системи, яку будемо встановлювати.

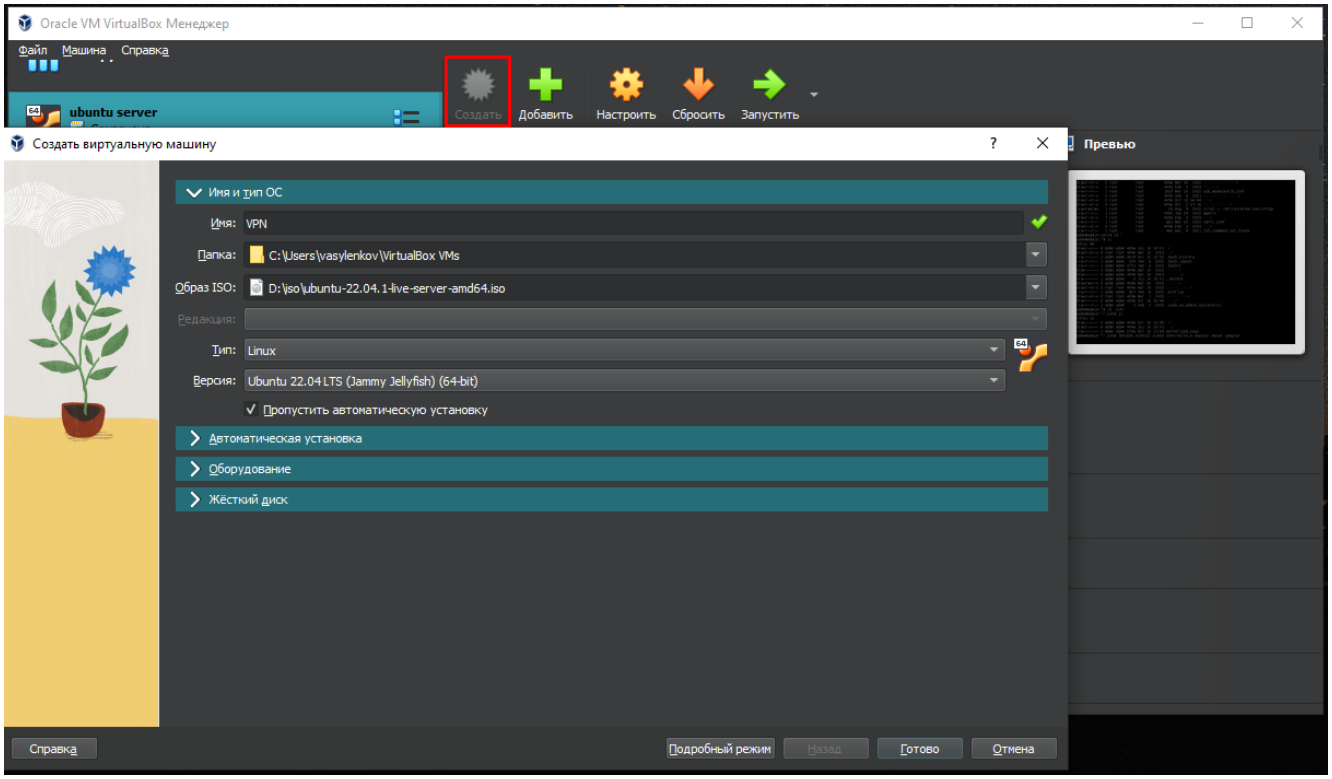

Рисунок 3.5 Вікно створення віртуальної машини в VirtualBox

І запускаємо цю віртуальну машину в нас йде встановлення операційної системи

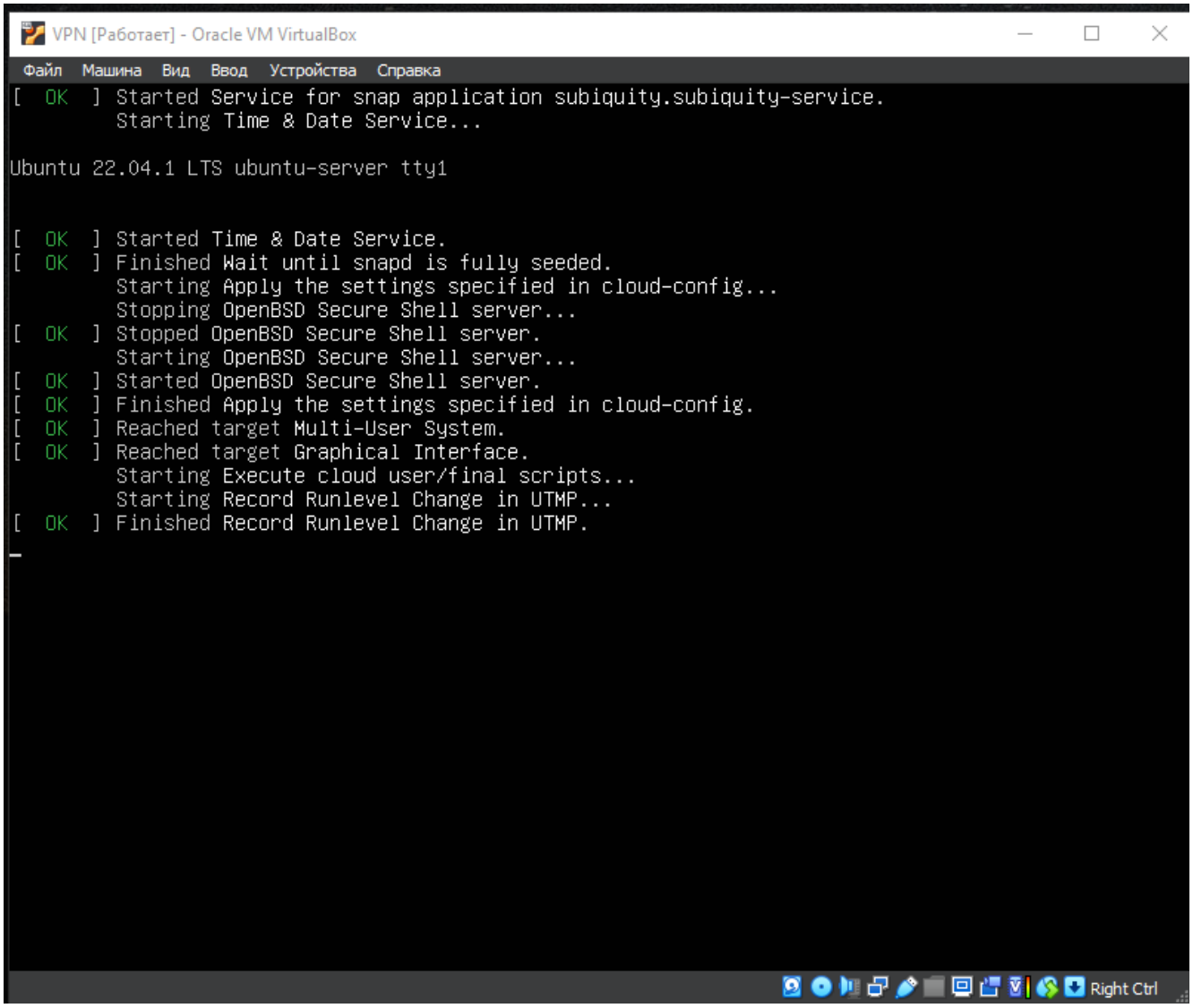

Рисунок 3.6 Вікно запуску віртуальної машини та автоматичного монтування образу

Далі автоматичний встановлючвач пропонує нам обрати мову для системи, залишаємо мову по замовченню – тобто English

Якщо є новіша версія операційної системи встановлювач – запропонує її, але я не маю необхідності в оновленні до останньої версії.

В наступному меню – пропонується обрати мову клавіатури, окрім англійскої нам інші не знадобляться, то ж продовжуємо

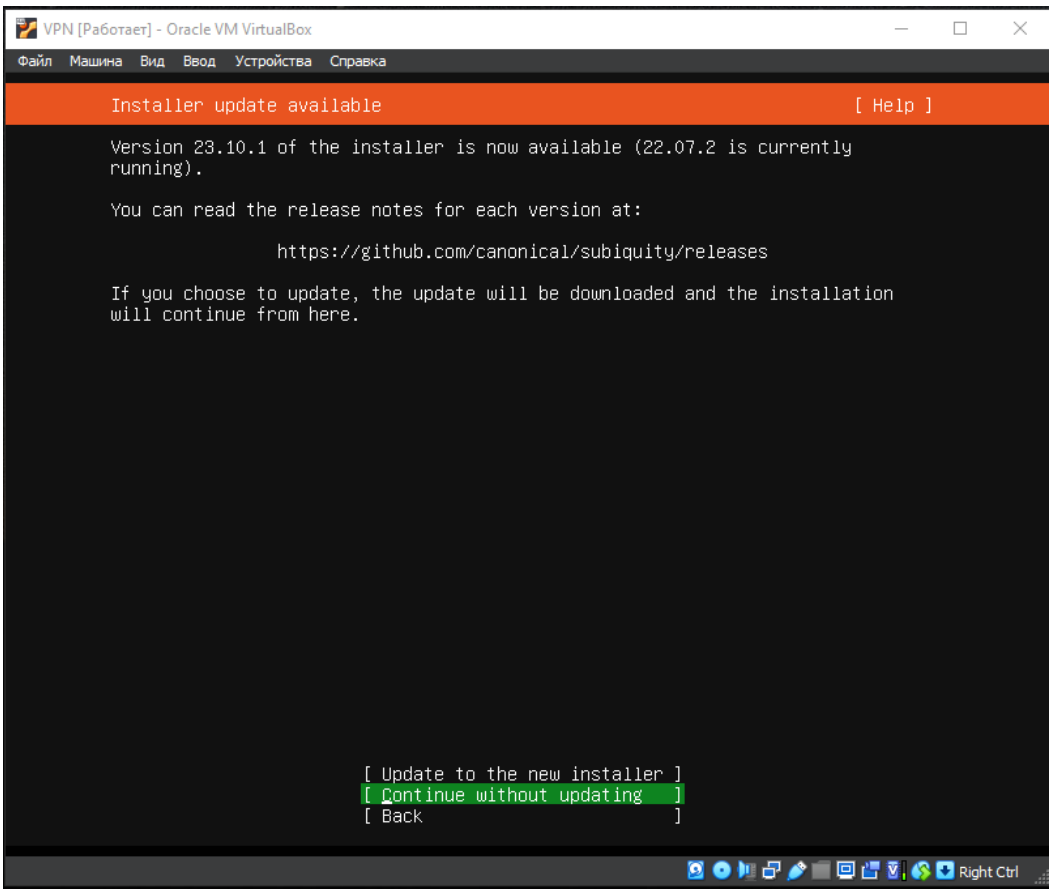

Рисунок 3.7 Вікно встановлення більш новішої версії

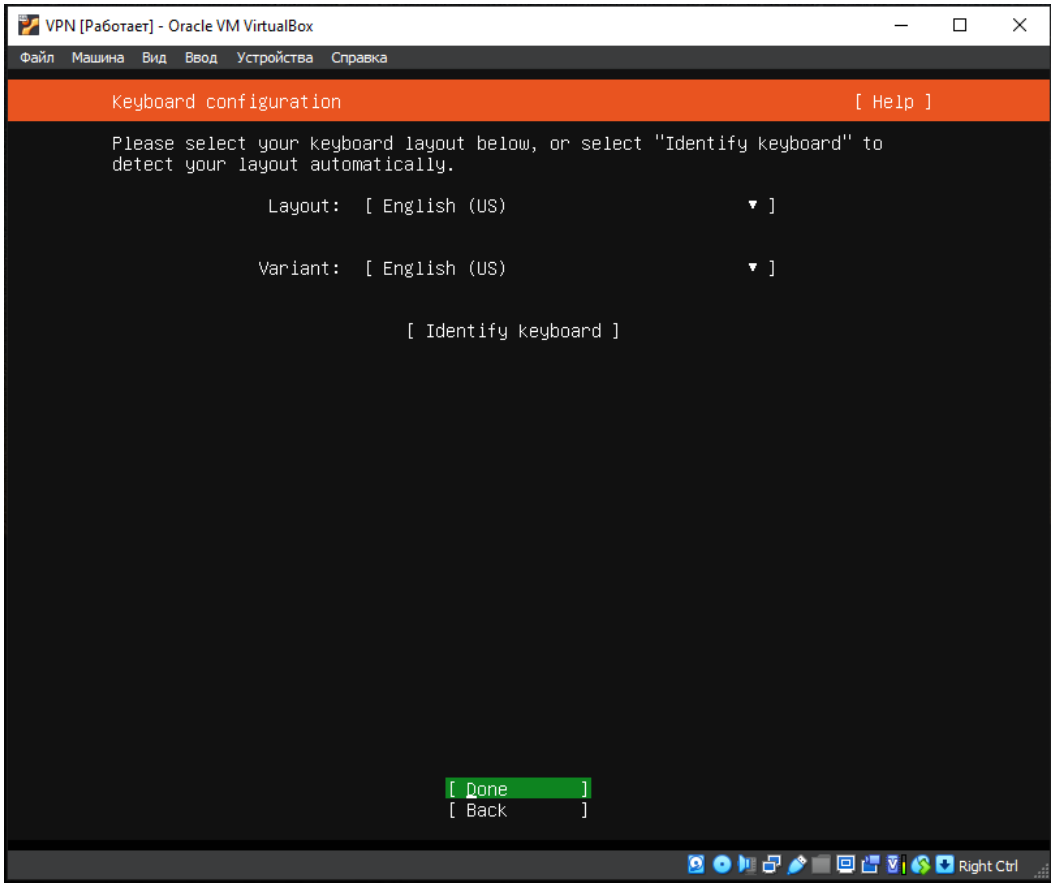

Рисунок 3.8 Вікно встановлення розкладки клавіатури

Налаштуємо одразу статичну ip адресу (за потреби завжди можна змінити), враховуючи налаштування нашої локальної мережі

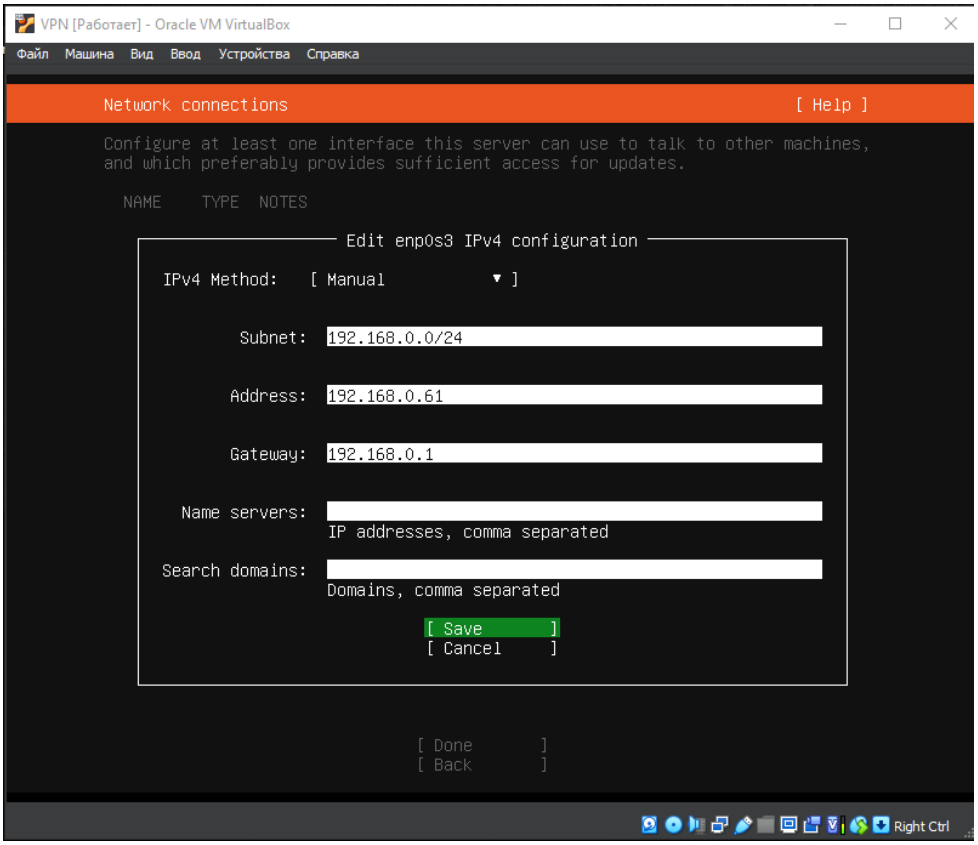

Рисунок 3.9 Встановлення ip-адреси для серверу VPN

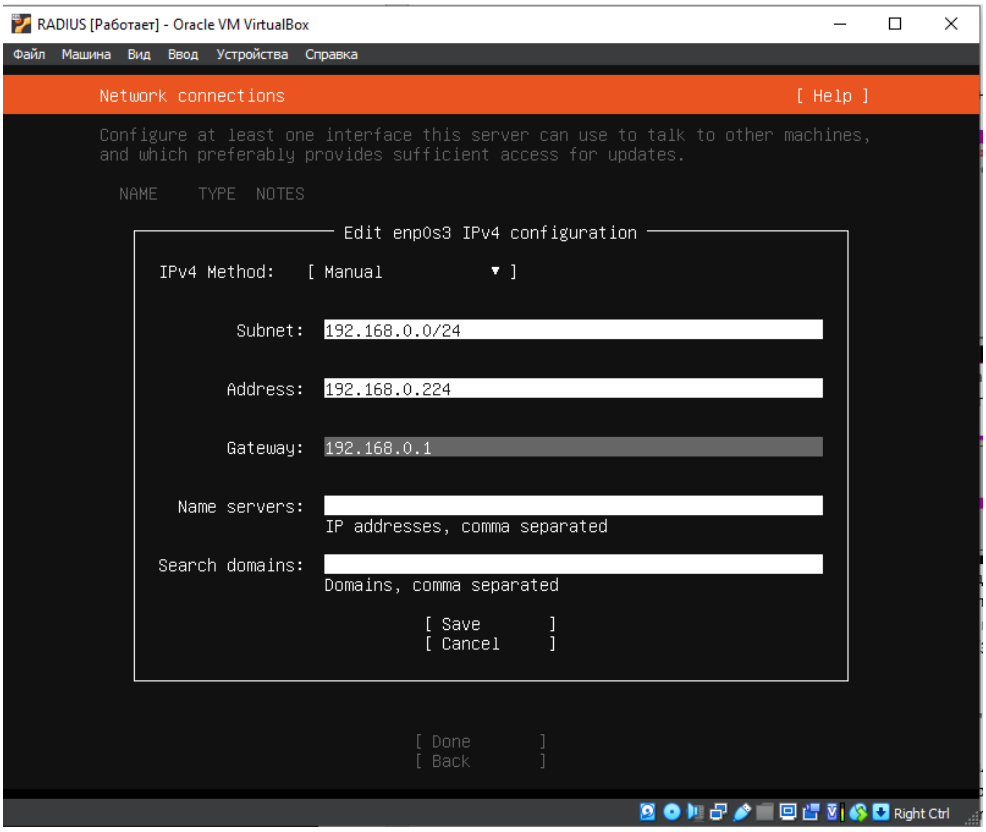

Рисунок 3.9 Встановлення ip-адреси для серверу RADIUS

Оберемо віртуальний диск, на якому і буде наша система та як саме буде розподілена логічно па'ять на диску

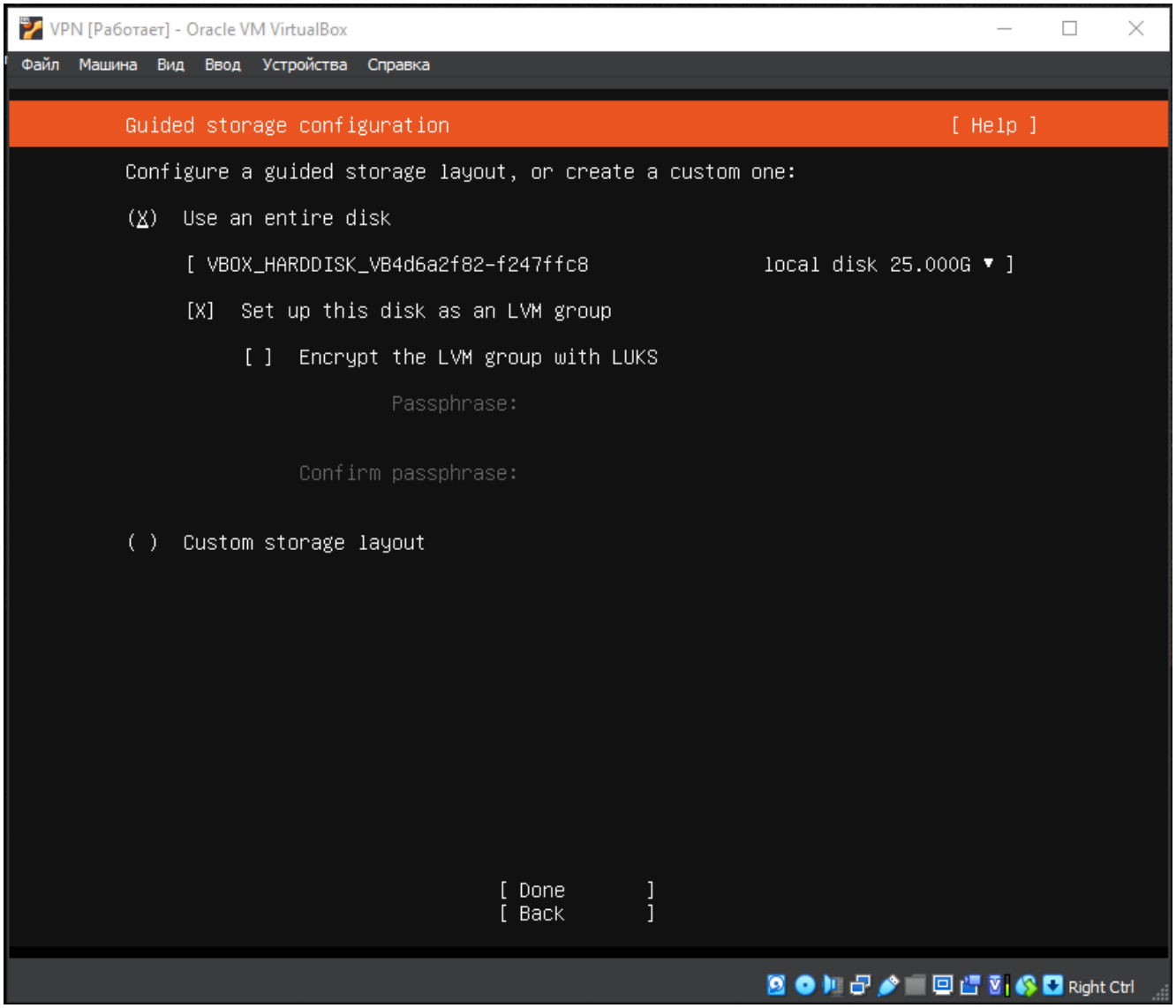

Рисунок 3.10 Вибір накопичувача для встановлення системи

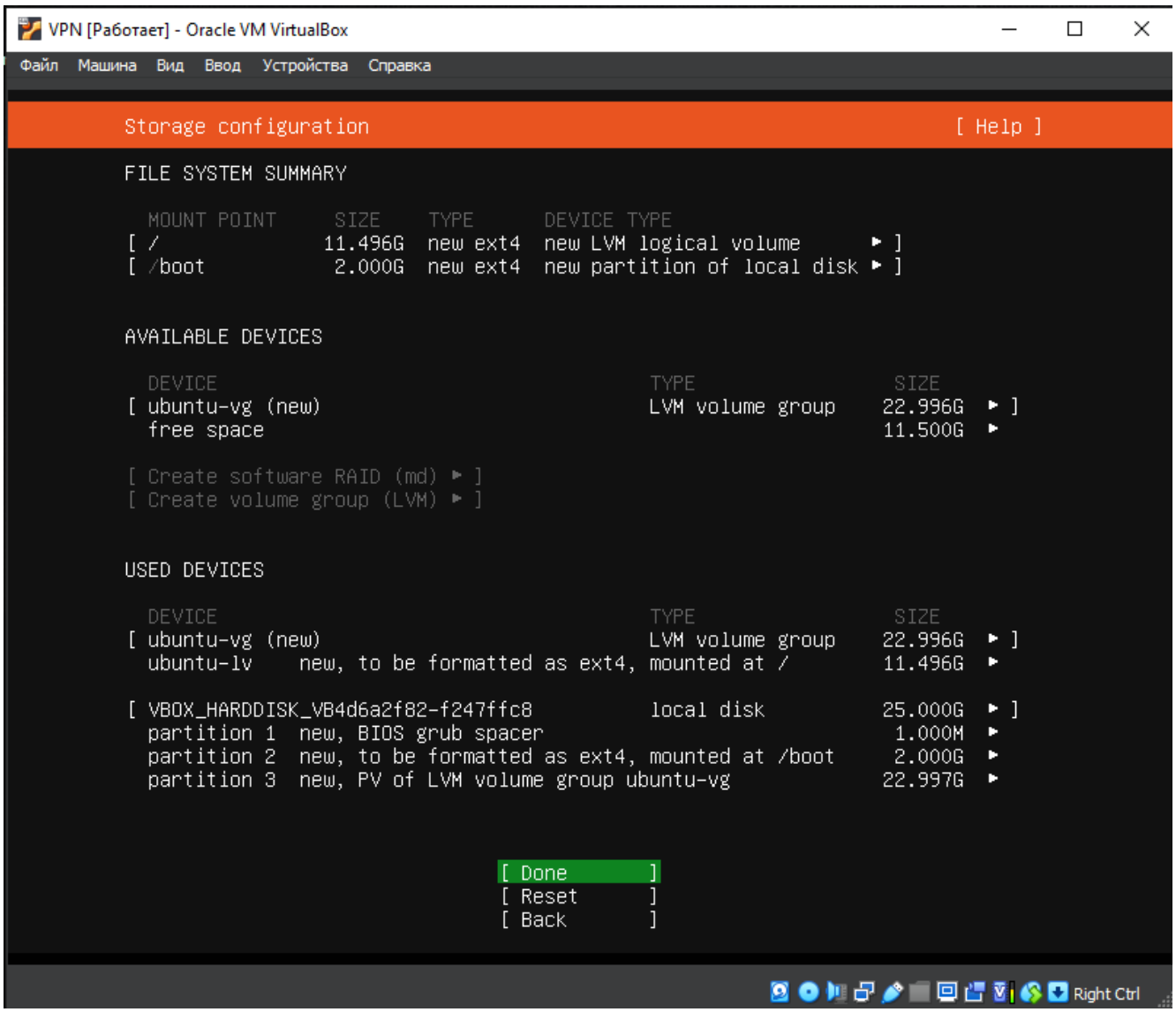

Рисунок 3.11 Логічне розподілення пам'яті диску

Погоджуємося на форматування диску

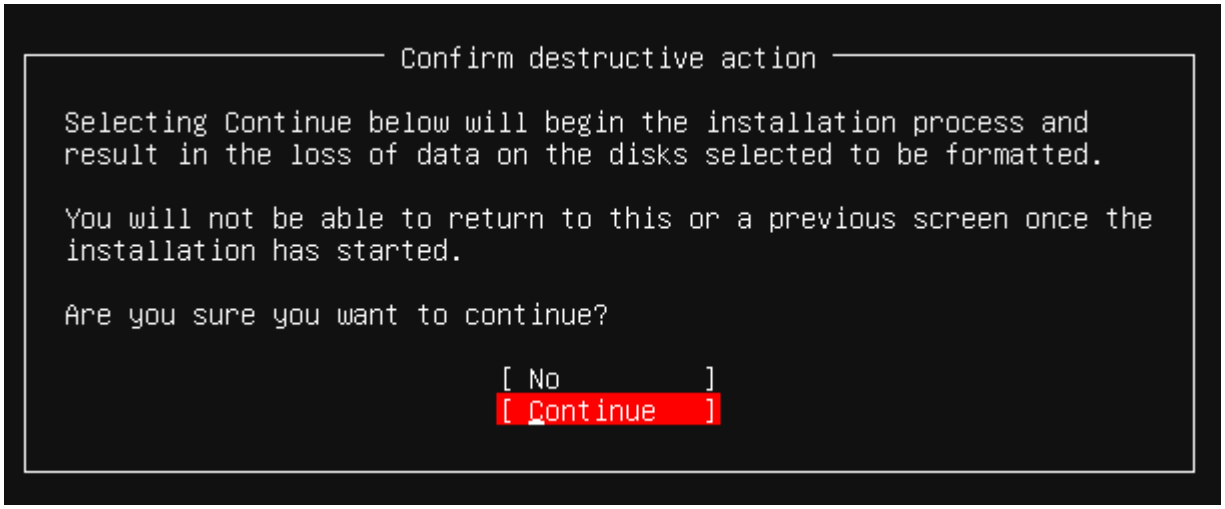

Рисунок 3.12 Форматування диску

Наступним кроком створюємо користувача для нашої системи.

Назвемо сервери наші в відповідності до їх задач

Your server`s name: VPN/RADIUS

Ім'я користувача для зручності введемо – admn.

А також пропишемо пароль для цтого користувача двічі для підтвердження вірності введення

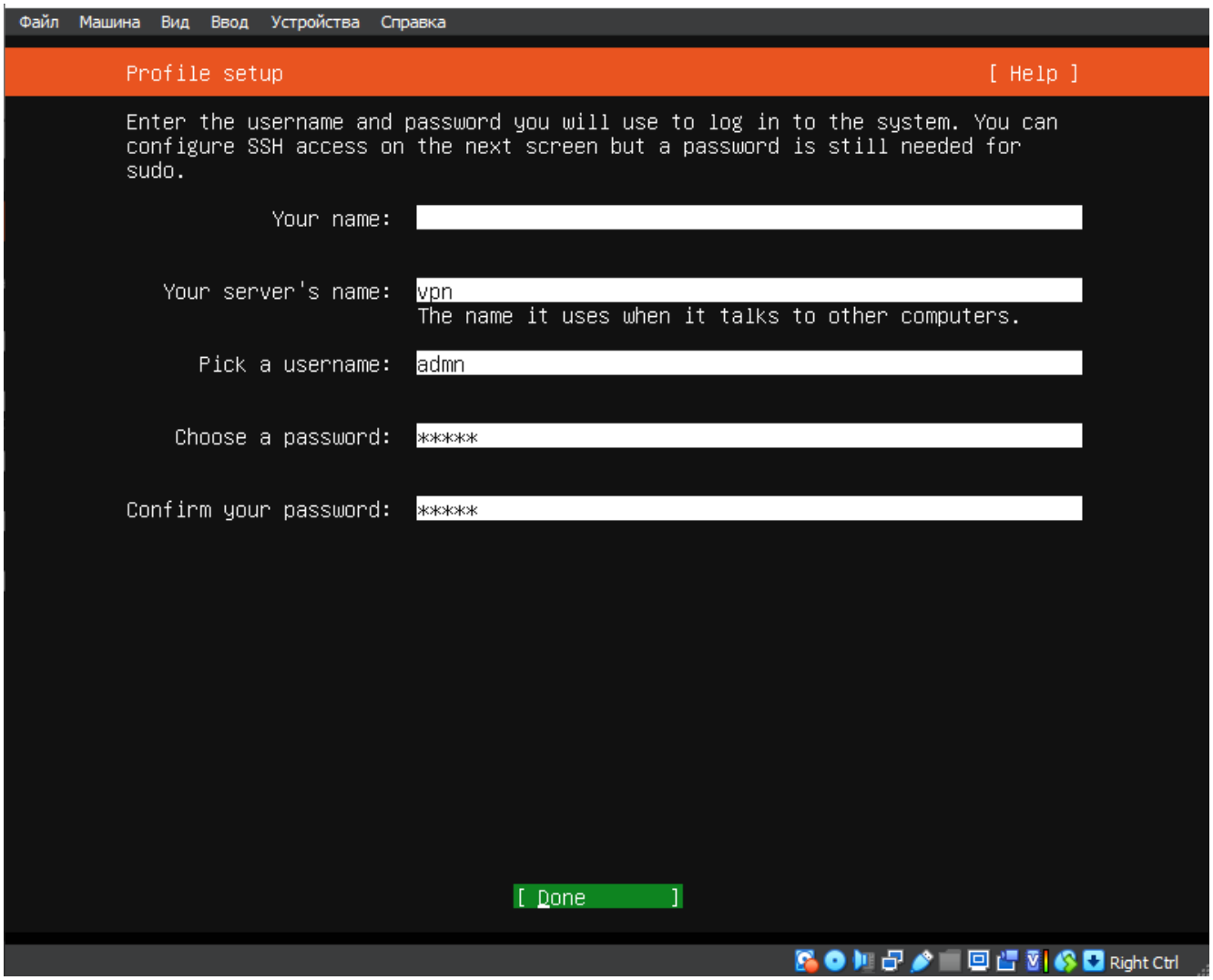

Рисунок 3.13 Встановлення даних для користувача серверу VPN

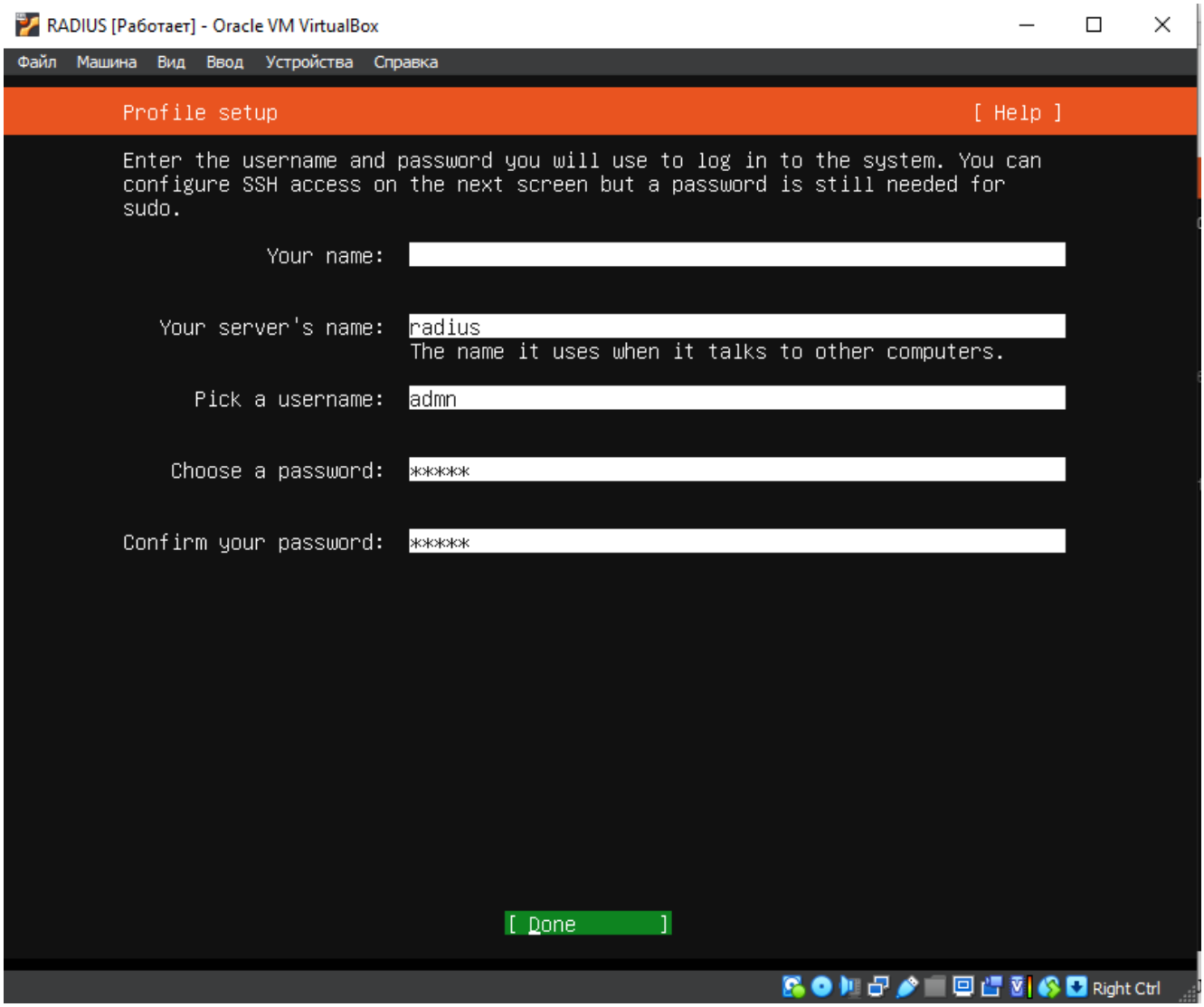

Рисунок 3.14 Встановлення даних для користувача серверу RADIUS

Далі слід вказати, що треба встановити відразу OpenSSH server, адже надалі, для більшої зручності ми будемо працювати з нашим сервером через SSH, це не тільки буде зручно, але і надасть можливість працювати так, як і з серверами, які були б встановленні ні на віртуальній машині, до якої завжди є доступ, так і з серверами, які встановленні десь, куди немає прямого фізичного доступу.

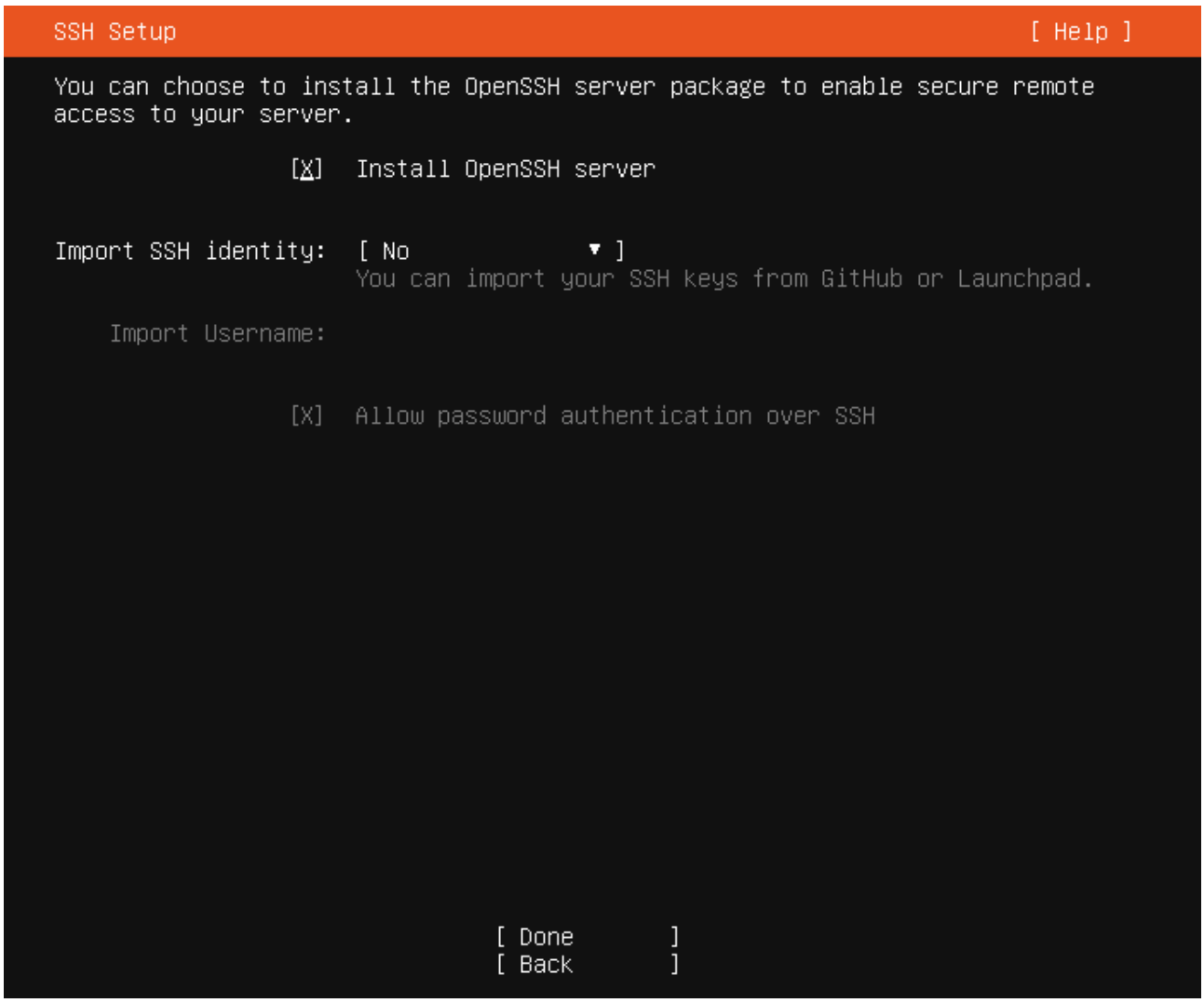

Рисунок 3.15 Встановлення OpenSSH

Продовжуємо інсталяцію Ubuntu на систему та очікуємо завершення встановлення. По закінченню - перезавантажуємо машину.

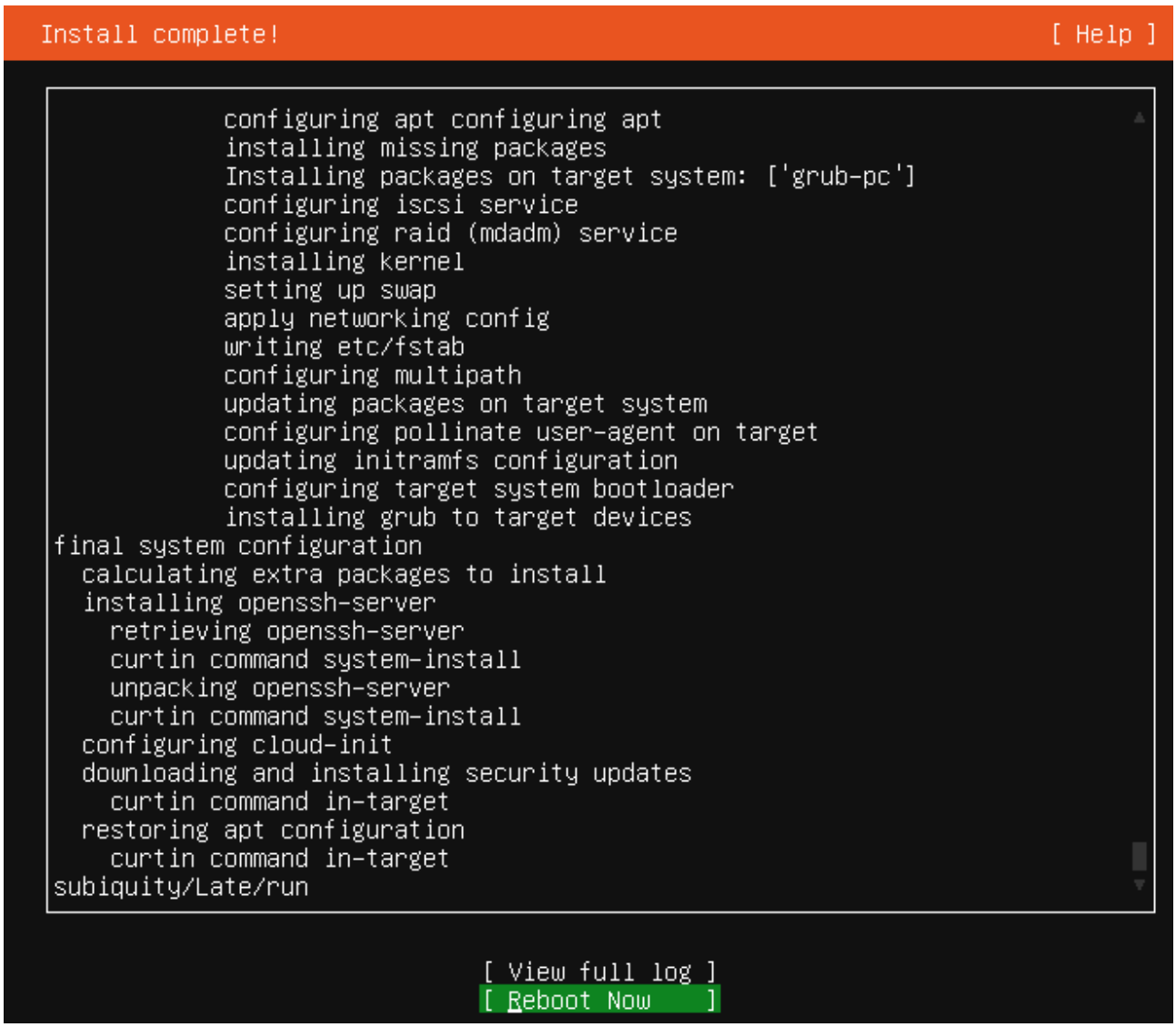

Рисунок 3.16 Вікно завершення інсталяцію та перезавантаження

3.2 Налаштування серверу VPN

Під'єднуємося до нашої машини по SSH. Тепер нам надо встановити залежності щоб працював VPN сервер

Для цього потрібно нам завантажити з сайту softetherVPN, для цього знаходимо на сайті та копіюємо посилання для завантаження. Копіюємо це посилання до нашого термінала SSH з командою wget перед посиланням щоб завантажити

Виконуючи таку комаду

wget https://www.softether-download.com/files/softether/v4.43-9799-beta-2023.08.31-tree/Linux/SoftEther_VPN_Server/64bit_-_Intel_x64_or_AMD64/softether-vpnserver-v4.43-9799-beta-2023.08.31-linux-x64-64bit.tar.gz

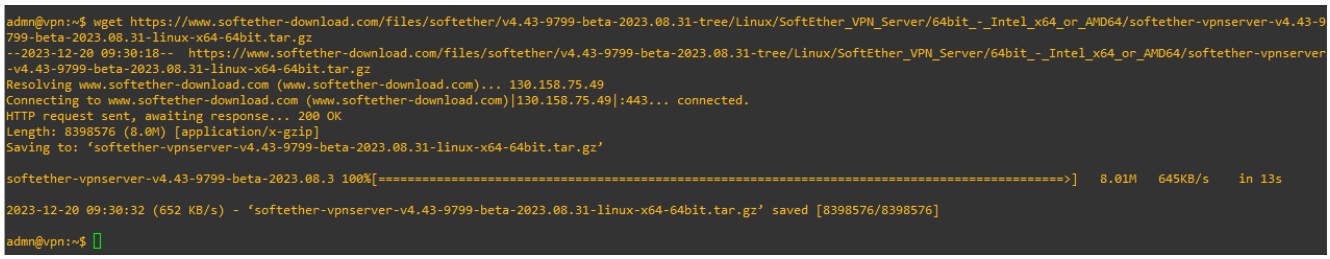

Рисунок 3.17 Завантаження пакету softetherVPN

Наступним кроком – розпакуємо архив, який ми завантажили за допомогою команди *tar*. Також для цієї команди будемо використовувати наступні ключі

- *x = «eXtract» : вилучити файли*
- *v = «verbose» : виводити інформацію в процесі виконання*
- *f = «file» : використовувати ім'я файлу архіву для розпакування*

Рисунок 3.18 Розпакування архіву softetherVPN

Переходимо до щойно розпакованного каталога VPN-серверу за допомогою команди *cd*

cd vpnserver/

Та використовуємо команду для встановлення залежностей для подальшої змоги запустити програмний продукт softetherVPN

Sudo apt install gcc binutils gzip libreadline-dev libssl-dev libncurses5-dev libncursesw5-dev libpthread-stubs0-dev

gcc: Компілятор GNU для мови програмування C.

binutils: Набір утиліт для маніпулювання об'єктними файлами та бінарними файлами.

gzip: Утиліта для стиснення файлів.

libreadline-dev: Бібліотека для роботи з рядками введення, зазвичай використовується для створення інтерактивних оболонок.

libssl-dev: Розробницький пакунок для роботи з OpenSSL (бібліотека шифрування та безпеки).

libncurses5-dev та *libncursesw5-dev:* Розробницькі файли для бібліотеки ncurses, яка надає API для створення текстових інтерфейсів користувача в терміналі.

libpthread-stubs0-dev: Розробницькі файли для бібліотеки, що надає інтерфейси для багатопотоковості в програмах.

```
admn@vpn:~$ cd vpnserver/
admn@vpn:~/vpnserver$ sudo apt install gcc binutils gzip libreadline-dev libssl-dev libncurses5-dev libncursesw5-dev libpthread-stubs0
-dev
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
binutils is already the newest version (2.38-4ubuntu2.4).
binutils set to manually installed.
The following additional packages will be installed:
  cpp cpp-11 fontconfig-config fonts-dejavu-core gcc-11 gcc-11-base libasan6 libatomic1 libc-dev-bin libc-devtools libc6-dev
  libcc1-0 libcrypt-dev libdeflate0 libfontconfig1 libgcc-11-dev libgd3 libgomp1 libisl23 libitm1 libjbig0 libjpeg-turbo8 libjpeg8
  liblsan0 libmpc3 libncurses-dev libnsl-dev libquadmath0 libtiff5 libtirpc-dev libtsan0 libubsan1 libwebp7 libxpm4 linux-libc-dev
  manpages-dev rpcsvc-proto
Suggested packages:
  cpp-doc gcc-11-locales gcc-multilib make autoconf automake libtool flex bison gdb gcc-doc gcc-11-multilib gcc-11-doc glibc-doc
  libgd-tools ncurses-doc readline-doc libssl-doc
The following NEW packages will be installed:
  cpp cpp-11 fontconfig-config fonts-dejavu-core gcc gcc-11 gcc-11-base libasan6 libatomic1 libc-dev-bin libc-devtools libc6-dev
  libcc1-0 libcrypt-dev libdeflate0 libfontconfig1 libgcc-11-dev libgd3 libgomp1 libisl23 libitm1 libjbig0 libjpeg-turbo8 libjpeg8
  liblsan0 libmpc3 libncurses-dev libncurses5-dev libncursesw5-dev libnsl-dev libpthread-stubs0-dev libquadmath0 libreadline-dev
  libssl-dev libtiff5 libtirpc-dev libtsan0 libubsan1 libwebp7 libxpm4 linux-libc-dev manpages-dev rpcsvc-proto
The following packages will be upgraded:
  gzip
1 upgraded, 43 newly installed, 0 to remove and 94 not upgraded.
Need to get 51.7 MB of archives.
After this operation, 168 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Рисунок 3.19 Встановлення залежностей

Потім, зсилаючись на видобутий каталог, запускаю процес створення SoftEther VPN, ввівши наступні команди

Команда *make* створює два двійкові файли, *vpnserver* (двійковий файл сервера) та *vpncmd* (інструмент керування терміналами VPN SoftEther), із вихідного коду.

apt-get install build-essential gnupg2 gcc make - для встановлення інструмента make для автоматизації процесу компіляції програм. Він використовує файли під назвою Makefile, щоб визначити, які частини програми потрібно компілювати та які команди виконувати для цього.

make

```
admn@vpn:~/vpnserver$ sudo apt-get install build-essential gnupg2 gcc make
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gcc is already the newest version (4:11.2.0-1ubuntu1).
The following additional packages will be installed:
  bzip2 dpkg-dev fakeroot g++ g++-11 libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libdpkg-perl
  libfakeroot libfile-fcntllock-perl libstdc++-11-dev lto-disabled-list
Suggested packages:
  bzip2-doc debian-keyring g++-multilib g++-11-multilib gcc-11-doc bzr libstdc++-11-doc make-doc
The following NEW packages will be installed:
  build-essential bzip2 dpkg-dev fakeroot g++ g++-11 gnupg2 libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl
  libdpkg-perl libfakeroot libfile-fcntllock-perl libstdc++-11-dev lto-disabled-list make
0 upgraded, 16 newly installed, 0 to remove and 94 not upgraded.
Need to get 15.1 MB of archives.
After this operation, 55.2 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Рисунок 3.20 Виконання команди *apt-get install build-essential gnupg2 gcc make*

```
admn@vpn:~/vpnserver$ make
--------------------------------------------------------------------

SoftEther VPN Server (Ver 4.43, Build 9799, Intel x64 / AMD64) for Linux Build Utility
Copyright (c) SoftEther Project at University of Tsukuba, Japan. All Rights Reserved.

--------------------------------------------------------------------


Copyright (c) all contributors on SoftEther VPN project in GitHub.
Copyright (c) Daiyuu Nobori, SoftEther Project at University of Tsukuba, and SoftEther Corporation.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.
```

Рисунок 3.21 Виконання *make*

```
*** PacketiX VPN Server HTML5 Web Administration Console (NEW) ***
This VPN Server / Bridge has the built-in HTML5 Web Administration Console.

After you start the server daemon, you can open the HTML5 Web Administration Console is available at

https://127.0.0.1:5555/
or
https://ip_address_of_the_vpn_server:5555/

This HTML5 page is obviously under construction, and your HTML5 development contribution is very appreciated.

---------------------------------------------------------------------

make[1]: Leaving directory '/home/admn/vpnserver'
```

Рисунок 3.22 Відомості за стандартне розташування серверу після запуску

Перемістемо наш softetherVPN до каталогу */opt* де розміщуються додаткові пакети програмного забеспечення

Для цього вийдемо на каталог вище та перемістимо за допомогою команди *sudo mv vpnserver /opt/softether*

Перейдемо до каталогу opt та перевіримо наявність там переміщеної директорії

```
admn@vpn:~$ cd /opt
admn@vpn:/opt$ ls
softether
admn@vpn:/opt$ cd softether/
admn@vpn:/opt/softether$ ls
Authors.txt  hamcore.se2  Makefile                              ReadMeFirst_Important_Notices_ja.txt  vpnserver
chain_certs  lang.config  ReadMeFirst_Important_Notices_cn.txt  ReadMeFirst_License.txt
code         lib          ReadMeFirst_Important_Notices_en.txt  vpncmd
admn@vpn:/opt/softether$
```

Рисунок 3.23 Переміщення до каталогу opt

Ось ми встановили, тепер перейдемо до налаштування запускаємо *./vpncmd*

```
admn@vpn:/opt/softether$ ./vpncmd
vpncmd command - SoftEther VPN Command Line Management Utility
SoftEther VPN Command Line Management Utility (vpncmd command)
Version 4.43 Build 9799   (English)
Compiled 2023/08/31 10:50:49 by buildsan at crosswin with OpenSSL 3.0.9
Copyright (c) 2012-2023 SoftEther VPN Project. All Rights Reserved.

By using vpncmd program, the following can be achieved.

1. Management of VPN Server or VPN Bridge
2. Management of VPN Client
3. Use of VPN Tools (certificate creation and Network Traffic Speed Test Tool)

Select 1, 2 or 3: 1

Specify the host name or IP address of the computer that the destination VPN Server or VPN Bridge is operating on.
By specifying according to the format 'host name:port number', you can also specify the port number.
(When the port number is unspecified, 443 is used.)
If nothing is input and the Enter key is pressed, the connection will be made to the port number 8888 of localhost (this computer).
Hostname of IP Address of Destination:

If connecting to the server by Virtual Hub Admin Mode, please input the Virtual Hub name.
If connecting by server admin mode, please press Enter without inputting anything.
Specify Virtual Hub Name:
Connection has been established with VPN Server "localhost" (port 443).

You have administrator privileges for the entire VPN Server.

VPN Server>
```

Рисунок 3.24 Управління softetherVPN за допомогою *vpncmd*

Впершу чергу встановимо пароль, щоб могли під'єднатися до нашого серверу за допомогою UI.

```
VPN Server>ServerPasswordSet
ServerPasswordSet command - Set VPN Server Administrator Password
Please enter the password. To cancel press the Ctrl+D key.

Password: *****
Confirm input: *****


The command completed successfully.

VPN Server>
```

Рисунок 3.25 Встановлення паролю адміністратора softetherVPN за допомогою *vpncmd*

Заходимо на будь яку машину, яка знаходиться у мережі. Я підключаюся з ПК, на якій в мене і розгорнуті віртуальні машини та під'єднуємося за допомогою UI, який зазделегідь був встановлен з сайту softetherVPN

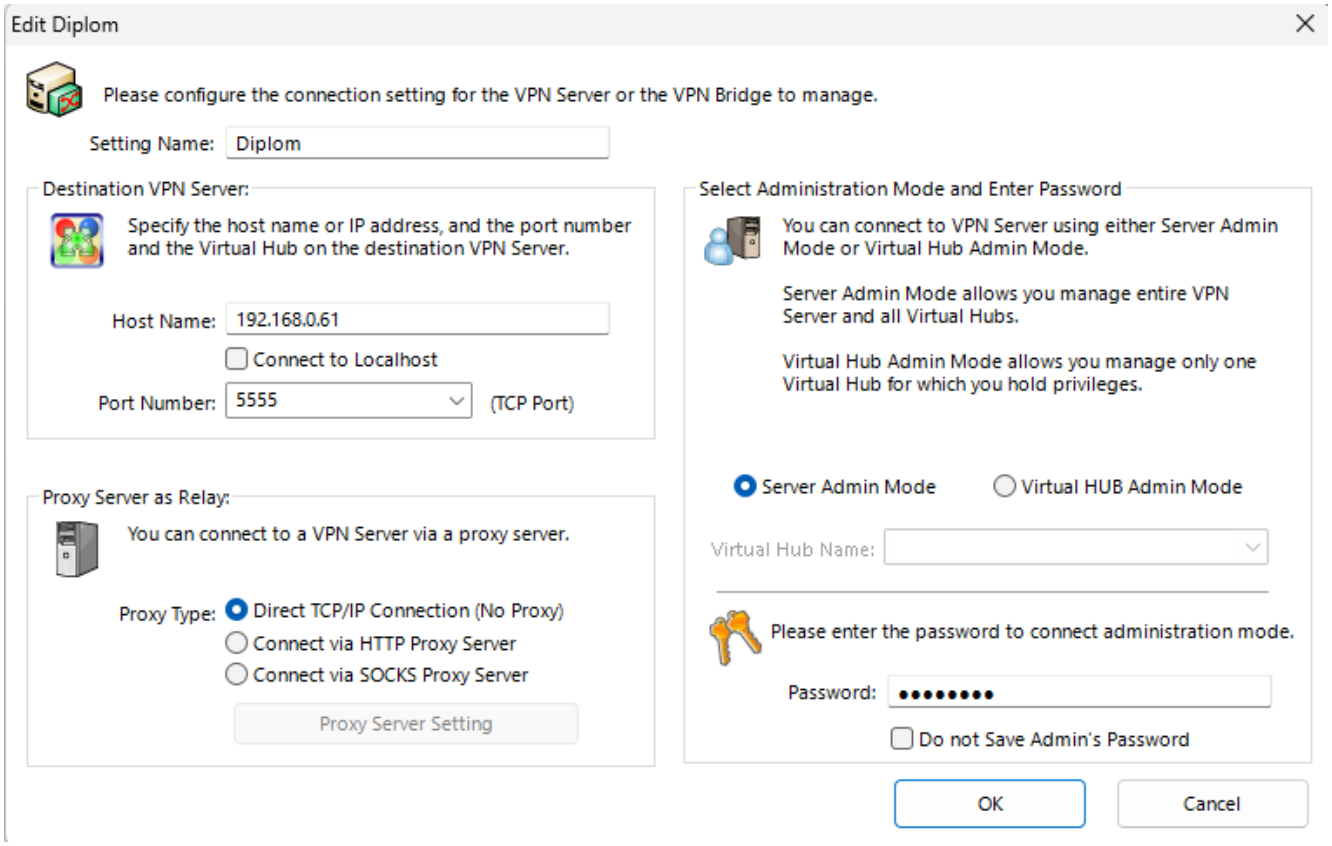

Рисунок 3.26 Підключення до серверу softetherVPN за допомогою UI

Переходимо в керування віртуального хабу та в керування користувачами, створюємо користувача з ім'ям «*», щоб можливість була у бідь кого під'єднатися та виставляємо в переліку «Auth Type» - RADIUS Authentication

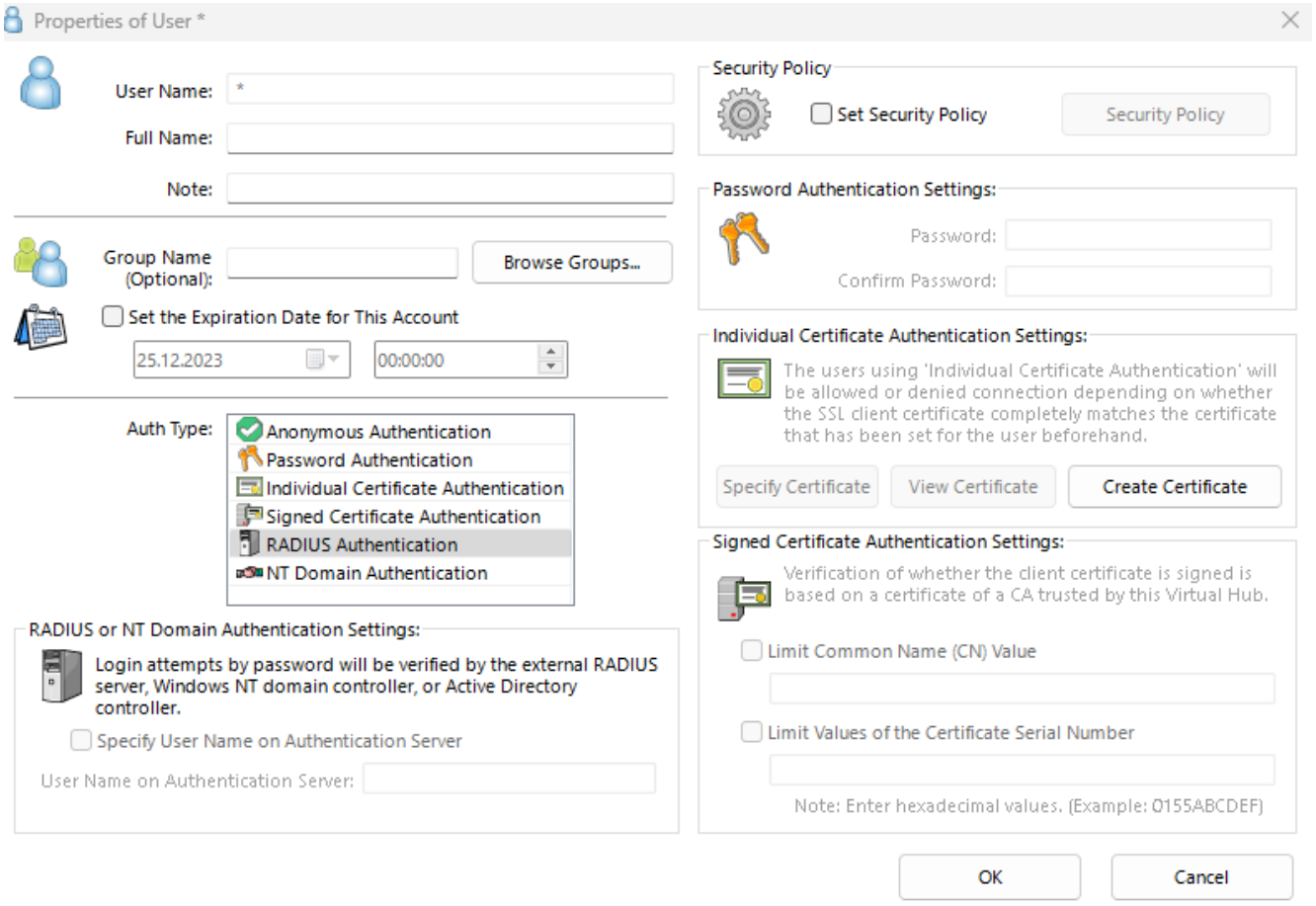

Рисунок 3.27 Створення користувача на сервері VPN

Наступним кроком ми налаштуємо Virtual NAT та VirtualDHCP на нашому VPN сервері

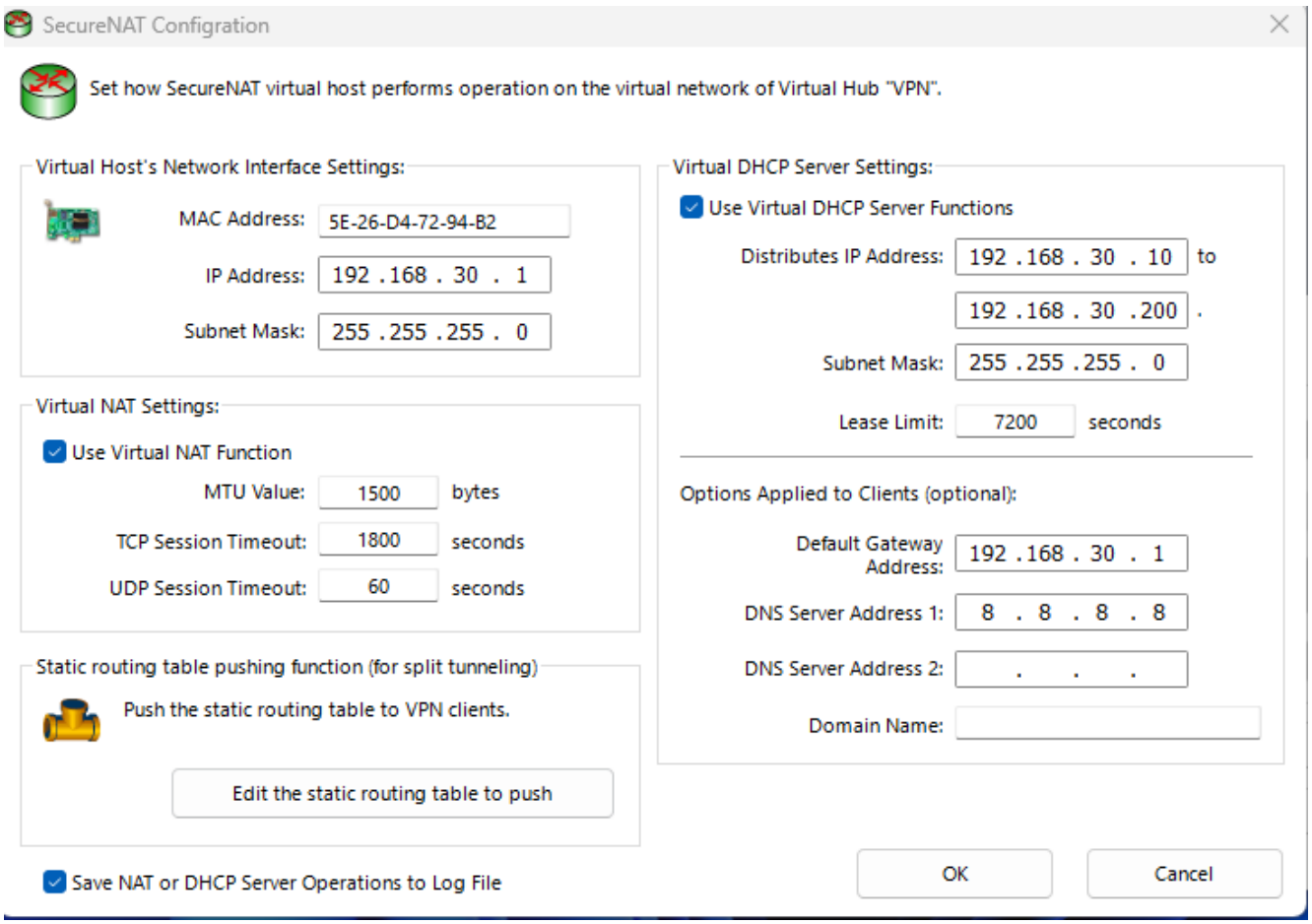

Рисунок 3.28 Налаштування Virtual NAT та VirtualDHCP

Також треба налаштувати з'єднання з RADIUS сервером, ip адреса, якого вже відома, а «Shared secret» введемо та запам'ятаємо, тому що його ще треба буде прописати в налашутваннях RADIUS сервера

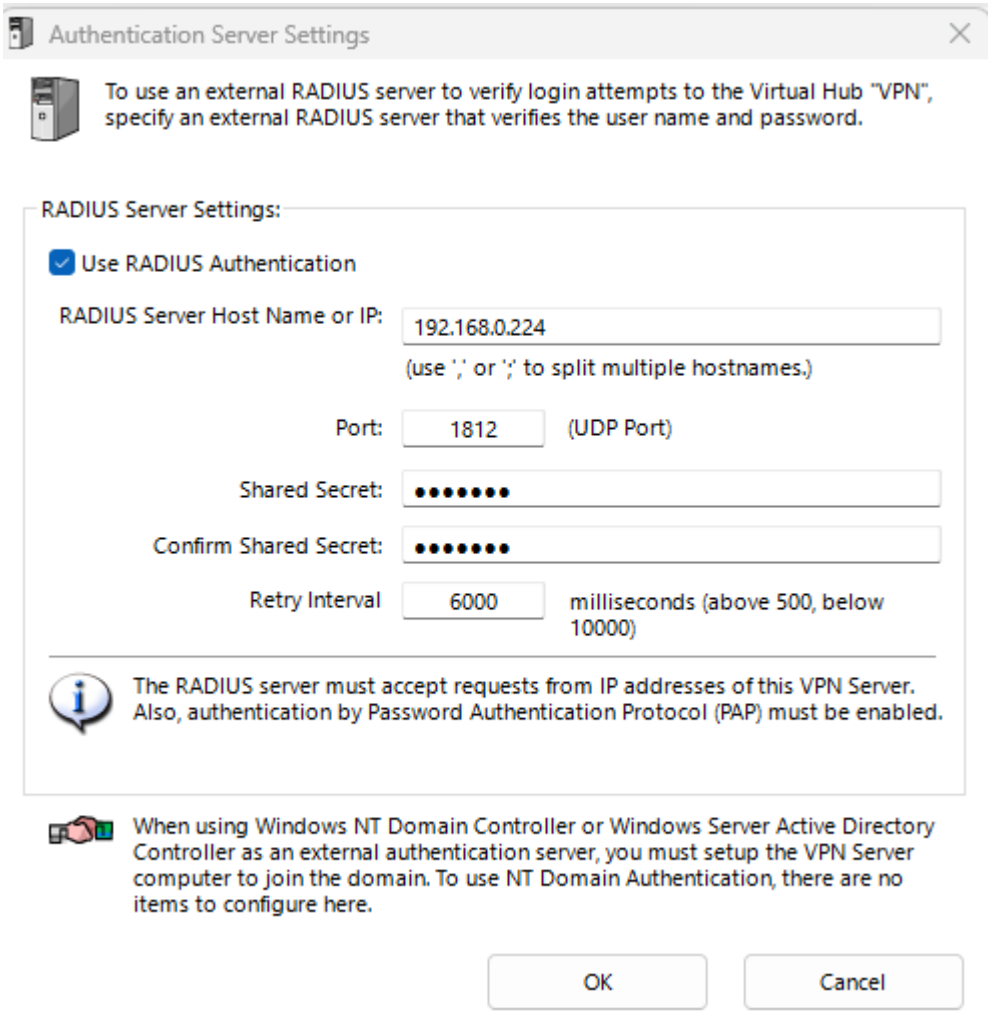

Рисунок 3.29 Налаштування з'єднання VPN з RADIUS

3.3 Налаштування серверу RADIUS

Перейдемо до встановлення freeradius на сервер. Для цього нам потрібно встановити залежності

Sudo apt-get install freeradius

```
admn@radius:~$ sudo apt-get install freeradius
[sudo] password for admn:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  freeradius-common freeradius-config freeradius-utils freetds-common libct4 libdbi-perl libfreeradius3 libtalloc2
  libtevent0 libwbclient0 make ssl-cert
Suggested packages:
  freeradius-krb5 freeradius-ldap freeradius-mysql freeradius-postgresql freeradius-python3 snmp libclone-perl
  libmldbm-perl libnet-daemon-perl libsql-statement-perl make-doc
The following NEW packages will be installed:
  freeradius freeradius-common freeradius-config freeradius-utils freetds-common libct4 libdbi-perl libfreeradius3
  libtalloc2 libtevent0 libwbclient0 make ssl-cert
0 upgraded, 13 newly installed, 0 to remove and 95 not upgraded.
Need to get 2,836 kB of archives.
After this operation, 10.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ua.archive.ubuntu.com/ubuntu jammy/main amd64 libtalloc2 amd64 2.3.3-2build1 [25.6 kB]
Get:2 http://ua.archive.ubuntu.com/ubuntu jammy/main amd64 libtevent0 amd64 0.11.0-1build1 [39.2 kB]
Get:3 http://ua.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libwbclient0 amd64 2:4.15.13+dfsg-0ubuntu1.5 [267 kB
]
Get:4 http://ua.archive.ubuntu.com/ubuntu jammy-updates/main amd64 freeradius-common all 3.0.26~dfsg~git20220223.1.00ed
0241fa-0ubuntu3.1 [234 kB]
Get:5 http://ua.archive.ubuntu.com/ubuntu jammy/main amd64 make amd64 4.3-4.1build1 [180 kB]
Get:6 http://ua.archive.ubuntu.com/ubuntu jammy/main amd64 ssl-cert all 1.1.2 [17.4 kB]
Get:7 http://ua.archive.ubuntu.com/ubuntu jammy-updates/main amd64 freeradius-config amd64 3.0.26~dfsg~git20220223.1.00
ed0241fa-0ubuntu3.1 [194 kB]
Get:8 http://ua.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libfreeradius3 amd64 3.0.26~dfsg~git20220223.1.00ed0
241fa-0ubuntu3.1 [200 kB]
Get:9 http://ua.archive.ubuntu.com/ubuntu jammy/main amd64 freetds-common all 1.3.6-1 [26.3 kB]
Get:10 http://ua.archive.ubuntu.com/ubuntu jammy/main amd64 libct4 amd64 1.3.6-1 [168 kB]
Get:11 http://ua.archive.ubuntu.com/ubuntu jammy-updates/main amd64 freeradius amd64 3.0.26~dfsg~git20220223.1.00ed0241
fa-0ubuntu3.1 [637 kB]
Get:12 http://ua.archive.ubuntu.com/ubuntu jammy-updates/main amd64 freeradius-utils amd64 3.0.26~dfsg~git20220223.1.00
ed0241fa-0ubuntu3.1 [106 kB]
Get:13 http://ua.archive.ubuntu.com/ubuntu jammy/main amd64 libdbi-perl amd64 1.643-3build3 [741 kB]
Fetched 2,836 kB in 0s (6,993 kB/s)
Preconfiguring packages ...
Selecting previously unselected package libtalloc2:amd64.
```

Рисунок 3.30 Встановлення залежності freeradius

Для більш зручного редагування файлів конфігурацій встановимо файловий менеджер Midnight Commander встановивши залежність

Sudo apt-get install mc

```
admn@radius:~$ sudo apt-get install mc
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bzip2 libssh2-1 mailcap mc-data mime-support unzip
Suggested packages:
  bzip2-doc arj catdvi | texlive-binaries dbview djvulibre-bin epub-utils genisoimage gv imagemagick libaspell-dev
  links | w3m | lynx odt2txt poppler-utils python python-boto python-tz unar wimtools xpdf | pdf-viewer zip
The following NEW packages will be installed:
  bzip2 libssh2-1 mailcap mc mc-data mime-support unzip
0 upgraded, 7 newly installed, 0 to remove and 95 not upgraded.
Need to get 2,321 kB of archives.
After this operation, 8,819 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ua.archive.ubuntu.com/ubuntu jammy/main amd64 bzip2 amd64 1.0.8-5build1 [34.8 kB]
Get:2 http://ua.archive.ubuntu.com/ubuntu jammy/universe amd64 libssh2-1 amd64 1.10.0-3 [109 kB]
Get:3 http://ua.archive.ubuntu.com/ubuntu jammy/main amd64 mailcap all 3.70+nmu1ubuntu1 [23.8 kB]
Get:4 http://ua.archive.ubuntu.com/ubuntu jammy/universe amd64 mc-data all 3:4.8.27-1 [1,427 kB]
Get:5 http://ua.archive.ubuntu.com/ubuntu jammy/universe amd64 mc amd64 3:4.8.27-1 [547 kB]
Get:6 http://ua.archive.ubuntu.com/ubuntu jammy/main amd64 mime-support all 3.66 [3,696 B]
Get:7 http://ua.archive.ubuntu.com/ubuntu jammy-updates/main amd64 unzip amd64 6.0-26ubuntu3.1 [174 kB]
Fetched 2,321 kB in 0s (16.3 MB/s)
Selecting previously unselected package bzip2.
(Reading database ... 75260 files and directories currently installed.)
Preparing to unpack .../0-bzip2_1.0.8-5build1_amd64.deb ...
Unpacking bzip2 (1.0.8-5build1) ...
Selecting previously unselected package libssh2-1:amd64.
Preparing to unpack .../1-libssh2-1_1.10.0-3_amd64.deb ...
Unpacking libssh2-1:amd64 (1.10.0-3) ...
Selecting previously unselected package mailcap.
Preparing to unpack .../2-mailcap_3.70+nmu1ubuntu1_all.deb ...
Unpacking mailcap (3.70+nmu1ubuntu1) ...
Selecting previously unselected package mc-data.
Preparing to unpack .../3-mc-data_3%3a4.8.27-1_all.deb ...
```

Рисунок 3.30 Встановлення залежності Midnight Commander

Викличемо Midnight Commander командою sudo mc та перейдемо за шляхом */etc/freeradius/3.0/*

У файлі radius.conf виправимо значення параметру *cleanup_delay = 30* , який відповідає за час відповіді аутентифікації

У файлі clients.conf додамо відомості про наш VPN сервер, а саме – назву, ip адресу так secret, який вводили при налаштуванні зв'язку VPN сервера та RADIUS сервера (на рис. 3.29)

```
client softethervpn {
	ipaddr = 192.168.0.61/24
	secret = secret1
}
```

Рисунок 3.30 Встановлення параметрів зв'язку за VPN сервером

Тепер вкажемо наш метод ауторизації, перейдемо в директорію /etc/freeradius/3.0/sites-enabled там є сімлінк на файл inner-tunnel.

Відкриємо його та в блоці authorize додамо такий блок

```
Update control {
        Auth-Type := `/usr/bin/python3 /opt/auth/main.py '%{User-Name}'`
}
```

3.4 Pyhton

Отже, я обрав мову програмування Python для реалізації ідеї двофакторної аутентифікації на базі Telegram через кілька ключових причин. По-перше, моя впевненість у знанні Python дозволить ефективно та швидко розробляти цей проект. Відкритість мови та величезна кількість доступної інформації в Інтернеті надають мені можливість швидко знаходити рішення для будь-яких технічних питань. Крім того, Python має широкий спектр бібліотек для роботи з Telegram API, що дозволяє зосередитися на функціональності проекту, а не на деталях взаємодії з API. Така гнучкість мови та її різноманітні можливості допоможуть легко впроваджувати складні функції, необхідні для двофакторної аутентифікації через Telegram. Усе це робить Python ідеальним вибором для ефективного та успішного втілення цієї ідеї.

3.4.1 Написання скрипта

Программа буде складатися з наступних функцій:

- *def CAPCTHA_make()*
- *@bot.callback_query_handler(func=lambda call: True)*
  def button_click(call):
- *@bot.message_handler(content_types=['text'])*
  def answer(message):
- *def get_keyboard(button):*
- *def timeout_bot(message):*
- *def main():*

Окрім цих функцій маються глобальні змінні та імпортування бібліотек. Розберемо все окремо, та почнемо з бібліотек.

import time # Імпорт модуля для роботи з часом

import threading # Імпорт модуля для роботи з потоками

import telebot # Імпорт модуля для роботи з Telegram Bot API

import sys # Імпорт модуля для взаємодії з системою

from multicolorcaptcha import CaptchaGenerator # Імпорт функцій для генерації CAPTCHA

Та глобальні змінні:

bot_token = 'TOKEN' # Задання токену для бота

bot = telebot.TeleBot(bot_token) # Створення об'єкту бота з використанням токену

user_name = (sys.argv[1]) # Отримання імені користувача з аргументів командного рядка

last_message_time = time.time() # Запам'ятовування часу останнього повідомлення

running = True # Прапорець для визначення стану роботи бота

db_user = {"vasylenkov": 'TELEGRAM_ID'} # Словник для зберігання користувачів та їх ідентифікаторів в системі

characters = '' # Змінна для зберігання символів CAPTCHA

Фунції опишу загалом, що вони роблять, детально код програми виклан в додатку А.

Функція *def CAPCTHA_make()* - створює CAPTCHA заданого розміру, отримує зображення та символи, що представлені на ньому, та повертає зображення CAPTCHA для подальшого використання в програмі.

@bot.callback_query_handler(func=lambda call: True) - Ця частина коду визначає обробник подій для відповідей на кнопки, що натиснуті в Telegram Bot API. І виконується функція *def button_click(call):* - реагує на натискання кнопок в чаті. Якщо ID чату співпадає з ID користувача, який взаємодіє з ботом, та дані кнопки відповідають "authorize", то відбувається процес авторизації. Якщо натиснута кнопка "recaptcha", відправляється нова CAPTCHA. Якщо натиснута кнопка "cancel", відбувається відміна аутентифікації.

@bot.message_handler(content_types=['text']) - відповідає за обробку текстових повідомлень, які отримує бот від користувачів. Після отримання тектового повідомлення викликається функція *def answer(message):* - вона обробляє текстові повідомлення, які надсилаються користувачем. Видаляє попереднє повідомлення користувача та оброблює його текст. Якщо текст повідомлення співпадає з CAPTCHA (characters), то відбувається успішна перевірка, і користувач отримує можливість кінцевої аутентифікації. У випадку невідповідності введеного тексту CAPTCHA відправляється нове зображення CAPTCHA, щоб користувач повторно ввів дані.

Функція *def get_keyboard(button):* - відповідає за створення клавіатури з кнопками для взаємодії з користувачем у Telegram. Залежно від параметру button створює клавіатуру з певними кнопками для користувача у чаті. Якщо параметр button дорівнює 1, буде створена клавіатура з однією кнопкою "Аутентифікуватися" для процесу авторизації. Якщо параметр button дорівнює 2, клавіатура матиме дві кнопки: "Відміна" та "Перегенерувати CAPTCHA" для відміни процесу аутентифікації або генерації нової CAPTCHA відповідно.

Функція *def timeout_bot(message):* - Ця функція перевіряє час останнього повідомлення користувача. Якщо пройшло більше 60 секунд з моменту останньої активності користувача, бот завершує сесію, видаляючи останнє повідомлення та зупиняючи свою роботу. Цикл продовжується, доки не виявлено взаємодію користувача або не минув визначений часовий інтервал.

Функція *def main():* - Вона розпочинає виконання основних операцій програми. Відправляє користувачу перше повідомлення з CAPTCHA та кнопками для взаємодії, створює потік для автоматичного завершення сесії, викликає метод bot.polling() для очікування вхідних повідомлень від користувача, позначає закінчення основного циклу роботи (за умови авторизації), а потім очікує завершення потоку автоматичного завершення сесії перед завершенням роботи програми.

3.4.1 Інтеграція кода в систему та встановлення бібліотек

Створимо папку /opt/auth перейшовши до директорії

cd /opt

sudo mkdir auth

Далі зайшовши до створенної папки

cd auth

Скрипт копіюємо в нього заздалегіть написану програму та зробимо його виконуванним

sudo nano main.py

sudo chmod 777 main.py

Терпер потрібно встановити бібліотеки, які буде використовувати наша програма

Для цього встановимо залежність інструменту pip за допомогою котрого будемо встановлювати бібліотеки

sudo apt install python3-pip

Бібліотеки, які нам потрібні – *multicolorcaptcha ma pyTelegramBotAPI*

pip install pyTelegramBotAPI

pip install multicolorcaptcha

```
admn@radius:/opt/auth$ pip install pyTelegramBotAPI
Defaulting to user installation because normal site-packages is not writeable
Collecting pyTelegramBotAPI
  Downloading pyTelegramBotAPI-4.14.1.tar.gz (244 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 244.2/244.2 KB 2.4 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: requests in /usr/lib/python3/dist-packages (from pyTelegramBotAPI) (2.25.1)
Building wheels for collected packages: pyTelegramBotAPI
  Building wheel for pyTelegramBotAPI (setup.py) ... done
  Created wheel for pyTelegramBotAPI: filename=pyTelegramBotAPI-4.14.1-py3-none-any.whl size=215676 sha256=7e97c07f2a4e408421e14611834fb4d75d3b9225d4552f0f39dd51f89861999b
  Stored in directory: /home/admn/.cache/pip/wheels/81/0e/bd/a68932e0d60008ec3eb11ae056e6d527af5ec9749922cd5a08
Successfully built pyTelegramBotAPI
Installing collected packages: pyTelegramBotAPI
Successfully installed pyTelegramBotAPI-4.14.1
admn@radius:/opt/auth$
```

Рисунок 3.31 Встановлення бібліотеки *pyTelegramBotAPI*

```
admn@radius:/opt/auth$ pip install multicolorcaptcha
Defaulting to user installation because normal site-packages is not writeable
Collecting multicolorcaptcha
  Downloading multicolorcaptcha-1.2.0-py3-none-any.whl (10.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 10.4/10.4 MB 8.1 MB/s eta 0:00:00
Collecting Pillow
  Downloading Pillow-10.1.0-cp310-cp310-manylinux_2_28_x86_64.whl (3.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.6/3.6 MB 52.3 MB/s eta 0:00:00
Installing collected packages: Pillow, multicolorcaptcha
Successfully installed Pillow-10.1.0 multicolorcaptcha-1.2.0
admn@radius:/opt/auth$
```

Рисунок 3.32 Встановлення бібліотеки *multicolorcaptcha*

3.5 Демонстрація роботи

Потрібно налаштувати клієнт softetherVPN вводячи в *host name* доменне ім'я, та ім'я користувача

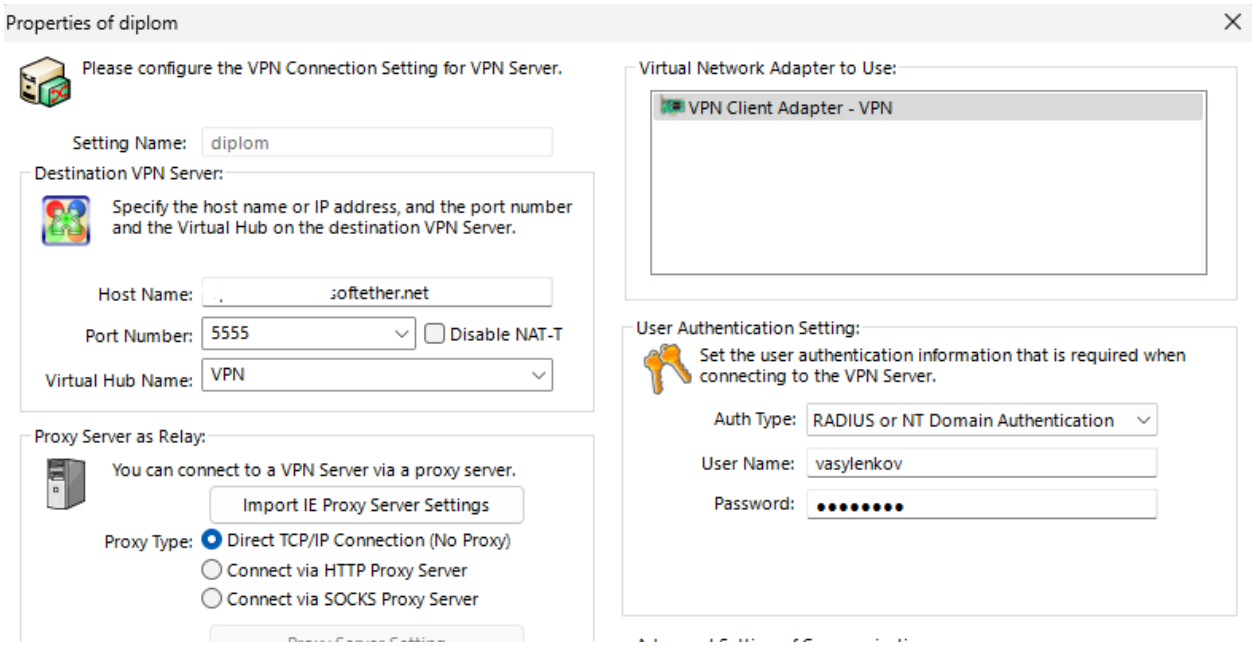

Рисунок 3.33 Налаштування підключення зі сторони клієнта

Виходимо до меню з основним списком VPN підключень обираємо те що налаштували та натискаємо «connect».

В той момент приходить до телеграму, id користувача які були налаштовані заздалегідь, повідомлення з пропозицією ввести «капчу», або відмовитися від авторизації

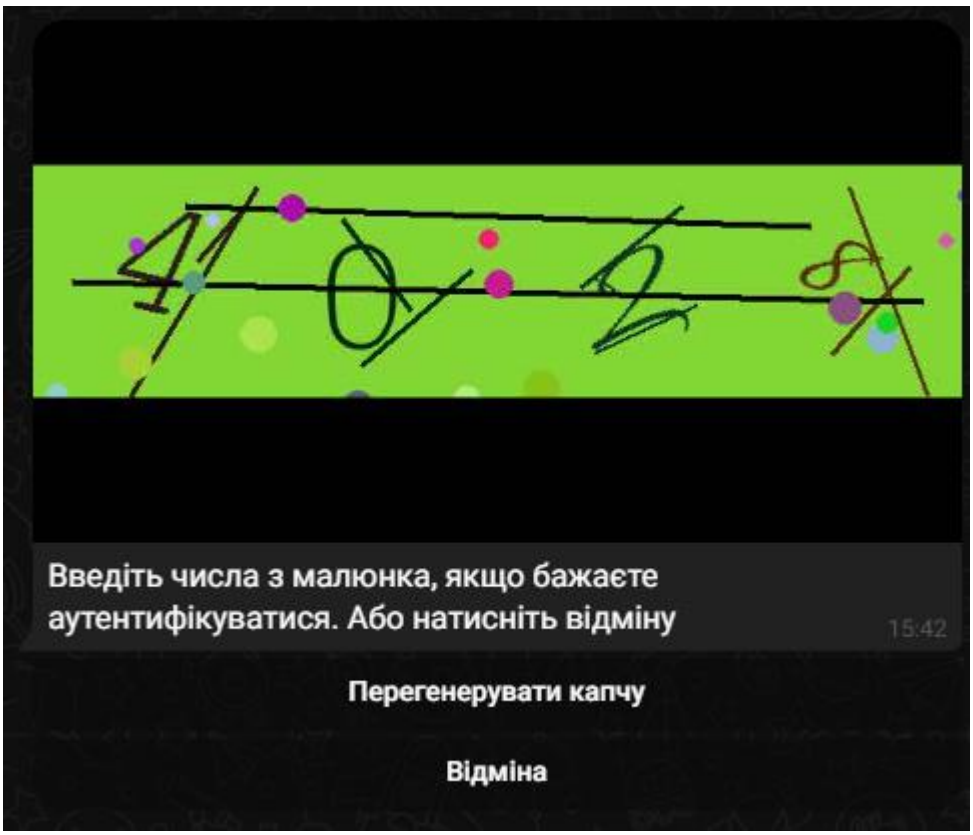

Рисунок 3.34 Перше повідомлення від бота з задачою ввести код

Якщо вона була введена вірно – прийде ще повідомлення з остаточним підтвердженням.

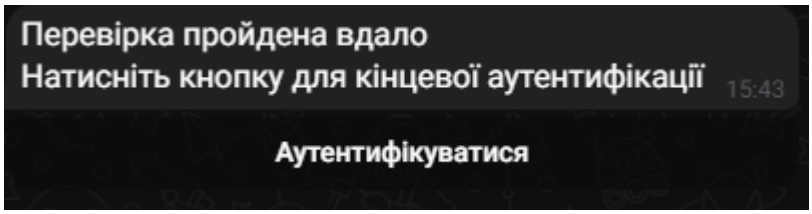

Рисунок 3.35 Підтвердження від користувача

Натискаємо кнопку і в нас вспливає інформативне повідомлення, що аутентифікація пройшла вдало.

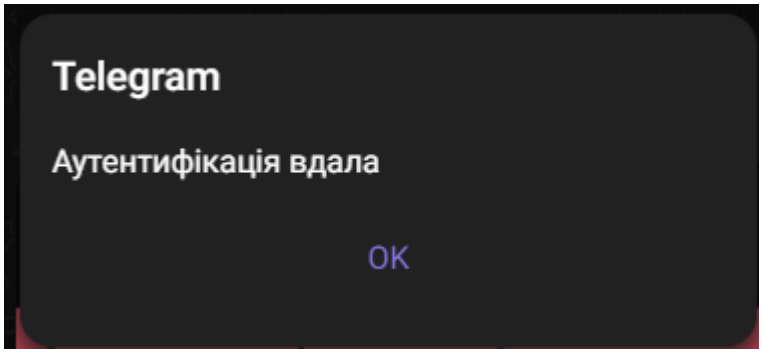

Рисунок 3.36 Інформація про вдалу аутентифікацію

Те що це так, можна подивитися в логах freeradius на сервері RADIUS

```
(2)     update control {
(2)       Executing: /usr/bin/python3 /tmp/main.py '%{User-Name}' '%{User-Password}' :
(2)       EXPAND %{User-Name}
(2)          --> vasylenkov
(2)       EXPAND %{User-Password}
(2)          -->
(2)       Program returned code (0) and output 'Accept'
(2)       Auth-Type := Accept
(2)       Session-Timeout := 60
(2)     } # update control = noop
(2)     update {
(2)       session-timeout := 60
(2)     } # update = noop
(2)     [expiration] = noop
(2)     [logintime] = noop
(2) pap: WARNING: Auth-Type already set.  Not setting to PAP
(2)     [pap] = noop
```

Рисунок 3.37 Термінал логування freeradius

Також можна побачити що VPN з'єднання встановлене в клієнті softetherVPN

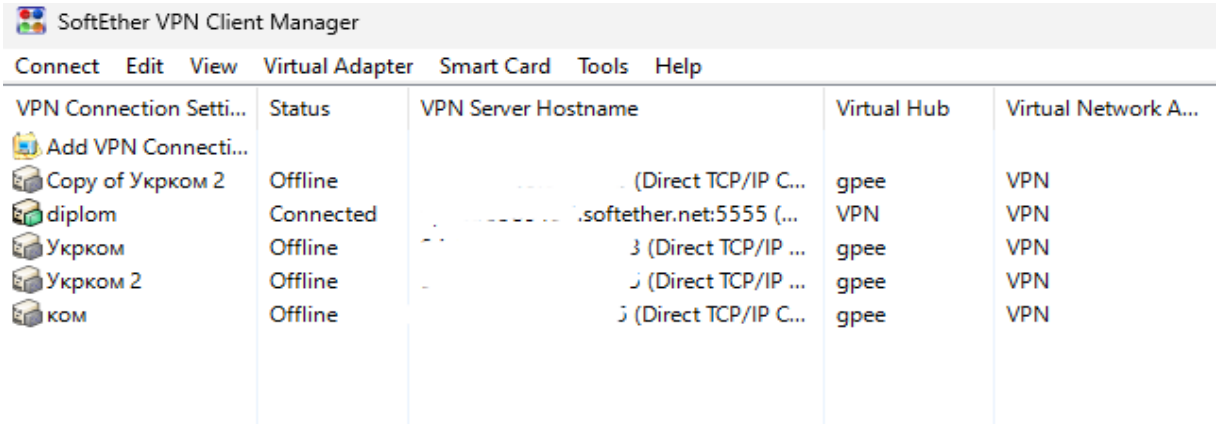

Рисунок 3.38 Встановлене підключеня до VPN

Якщо нерозбірливо те що на малюнку – є можливість перегенрувати іншу

Рисунок 3.39 Перегенерація капчі

Або якщо ви не намагалися увійти можна натиснуті «відміну» та програма поверне Reject

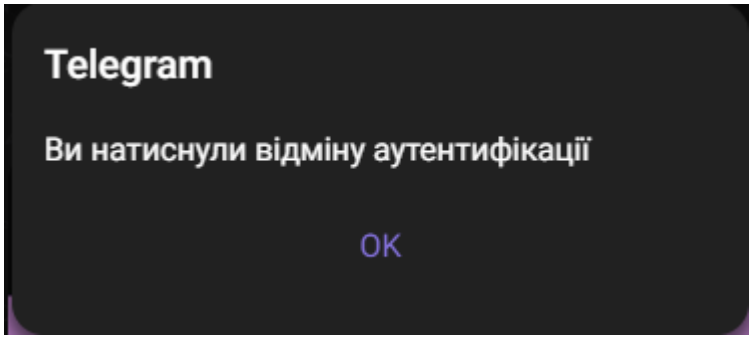

Рисунок 3.40 Відміна аутентифікації

Висновки

В розділі було спроєктованно та поетапно створено систему, яка забезпечує можливість створювати VPN-тунелі з двофакторною аутентифікацією, для якої потрібно підтвердити свої дії у месенджері Telegram.

Були розглянуті дистрибутиви, можливості unix-подібних систем, можливості freeradius, softetherVPN та варіативність створення будь якого функціоналу за допомогою мови програмування python та її вже створених бібліотек.

ВИСНОВКИ

Підводячи висновки проведеної роботи можу зазначити.

Я розглянув проблематику аутентифікації як у цілому, так і окремих випадків.

Головним чином була розглянута проблема аутентифікації при підключенні до віддаленого робочого місця та побудові VPN-з'єднання між клієнтом та сервером. З моїх досліджень я зрозумів, що важко виявити та виділити якийсь один фактор, в якомо полягає основна проблема, який можна було б виправити і все відразу стало би безпечно та надійно. Ні, є певна низка цих самих перепон, якісь полягають в людському факторі, деякі в нестійких та незахищених серверних рішеннях та багатьох інших компонентах цієї системи.

З огляду на прочитану інформацію – я зрозумів, що потрібно створювати нові методи аутентифікації, можливо вони будуть схожі на вже існуючі, але вони будуть унікальними, а чим більша кількість якихось рішень – тим більша конкурентноспроможність, з цього слідчить, що будуть розвиватися і інші методи, або модернізуватися запропоновані, так і буде зроблен вклад в розробку нової майже бездоганної системи для аутентифікації.

Отже, я запропонував рішення, а саме – вигадав принцип для аутентифікації користувачів до VPN через месенжер. Створив за цією думкою проєкт та реалізував його, використовуючи при цьому інструменти, які на мій погляд та з о'єктивних причин виявилися доволі зручними та пристасованними для вирішення поставлених задач.

Головними рішеннями, за домопогою яких було втілено в життя мій проєкт є: Softether VPN, Telegram, Freeradius, мова програмування Python.

Рішення спрямоване на підвищення рівня безпеки шляхом впровадження додаткового етапу аутентифікації, що зменшить ризики несанкціонованого доступу та підвищить відповідальність при вході до мережі. Використання месенджерів для аутентифікації надає зручність та швидкість у вході для користувачів, забезпечуючи при цьому додатковий захист.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Cisco Study: Employees say hybrid work makes them happier and more productive, but more needs to be done to make it more inclusive [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://newsroom.cisco.com/c/r/newsroom/en/us/a/y2022/m04/cisco-study-employees-say-hybrid-work-makes-them-happier-and-more-productive-but-more-needs-to-be-done-to-make-it-more-inclusive.html.

2. Future of Secure Remote Work Report [Електронний ресурс] – Режим доступу до ресурсу: https://www.cisco.com/c/en/us/products/security/future-secure-remote-work-report.html.

3. Haan K. Remote Work Statistics And Trends In 2024 [Електронний ресурс] / Katherine Haan // Forbes. – 2023. – Режим доступу до ресурсу: https://www.forbes.com/advisor/business/remote-work-statistics/.

4. Amico L. The Realities of Remote Work [Електронний ресурс] / Laura Amico. – 2021. – Режим доступу до ресурсу: https://hbr.org/2021/10/the-realities-of-remote-work.

5. Benefits and challenges of remote workers [Електронний ресурс] – Режим доступу до ресурсу: https://www.lucidchart.com/blog/benefits-and-challenges-of-remote-workers.

6. Herrity J. 10 Benefits of Working Remotely (With Challenges and Tips) [Електронний ресурс] / Jennifer Herrity. – 2023. – Режим доступу до ресурсу: https://www.indeed.com/career-advice/career-development/benefits-of-working-remotely.

7. The benefits and challenges of remote working [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: https://jobs.theguardian.com/article/the-benefits-and-challenges-of-remote-working.

8. Hasna A. What Is a VPN: How Does It Work and Should You Use It? [Електронний ресурс] / A. Hasna. – 2023. – Режим доступу до ресурсу: https://www.hostinger.in/tutorials/what-is-vpn.

9. Kochovski A. The Top 25 VPN Statistics, Facts & Trends for 2023 [Електронний ресурс] / Aleksandar Kochovski. – 2023. – Режим доступу до ресурсу: https://www.cloudwards.net/vpn-statistics/.

10. Chauncey C. VPN Statistics And Trends In 2023 [Електронний ресурс] / Crail Chauncey. – 2023. – Режим доступу до ресурсу: https://www.forbes.com/advisor/business/vpn-statistics/.

11. Types of Virtual Private Network (VPN) and its Protocols [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: https://www.geeksforgeeks.org/types-of-virtual-private-network-vpn-and-its-protocols/.

12. Broberg, J., Zeephongsekul., P., and Tari, Z. Approximating bounded general service distributions, In Proc. of IEEE Symposium on Computers and Communications, Aveiro, Protugal, Jul. 2007.

13. A Model for Content Internetworking [Електронний ресурс] / T.Gary, M. Day, S. Systems, P. Rzewski. – 2015. – Режим доступу до ресурсу: https://datatracker.ietf.org/doc/rfc3466/.

14. Architecture and Performance Models for QoS-Driven Effective Peering of Content Delivery Networks. Mukaddim Pathan, Rajkumar Buyya. 2016.

15. Буров Є. Комп’ютерні мережі / Є. Буров. – Львів: БаК, 1999. – 468 с.

16. Tanenbaum A. S. Computer Networks / Andrew S. Tanenbaum. – New Jersey: Prentice Hall PTR, 2003. – 674 с. – (Pearson Education, Inc).

17. Луцків А. М. ОГЛЯД СТАНДАРТІВ БІОМЕТРИЧНОЇ АУТЕНТИФІКАЦІЇ / А. М. Луцків, Ю. І. Брегін. // Матеріали VI Міжнародної науково-технічної конференції молодих учених та студентів. Актуальні задачі сучасних технологій. – 2017. – С. 22.

18. Sabarinath. Introduction to Two Factor Authentication and Different Types of 2FA [Електронний ресурс] / Sabarinath. – 2020. – Режим доступу до ресурсу: https://techlog360.com/two-factor-authentication-2fa/.

19. Huang H. Setup your own VPN server [Електронний ресурс] / Hugo Huang. – 2023. – Режим доступу до ресурсу: https://ubuntu.com/blog/setup-your-own-vpn-server

20. Використання штучного інтелекту у вищій освіті / [І. Драч, О. Петроє, О. Бородієнко та ін.]. // International Scientific Journal of Universities and Leadership. – 2023. – №15. – С. 66–82.

21. БАТАРЕЄВ В. В. МЕТОДИ ТА СИСТЕМИ ШТУЧНОГО ІНТЕЛЕКТУ / В. В. БАТАРЕЄВ. // Вісник Хмельницького національного університету. – 2021. – №1. – С. 17–21.

22. Форми біометричної аутентифікації [Електронний ресурс]. – 2015. – Режим доступу до ресурсу: https://worldvision.com.ua/articles/formi-biometricheskoy-autentifikatsii.

23. Мороз А. О. БІОМЕТРИЧНІ ТЕХНОЛОГІЇ ІДЕНТИФІКАЦІЇ ЛЮДИНИ. ОГЛЯД СИСТЕМ / А. О. Мороз. // Математичні машини і системи. – 2011. – №1. – С. 39–45.

24. Mastercard і ПриватБанк запускають перший в Україні проект поведінкової біометрії [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: https://privatbank.ua/news/2019/9/16/1018.

25. Devoy J. The IoT Security Foundation and FIDO Alliance Announce Collaboration to Eliminate Passwords in IoT [Електронний ресурс] / Jenny Devoy. – 2021. – Режим доступу до ресурсу: https://fidoalliance.org/the-iot-security-foundation-and-fido-alliance-announce-collaboration-to-eliminate-passwords-in-iot/.

26. Gates M. The Pernicious Problem of Passwords [Електронний ресурс] / Megan Gates. – 2022. – Режим доступу до ресурсу: https://www.asisonline.org/security-management-magazine/articles/2022/09/the-pernicious-problem-of-passwords/.

27. Антоненко А.В., Цюп’як Н.І., Коваленко А.Р., Василенков К.І., Березовський А.Ю. Інноваційні методи відображення інформації у веб-браузерах. Вчені записки Таврійського національного університету імені В.І.Вернадського. Серія «Технічні науки». 2023. Т. 34 (74). Ч. 3. № 6. С. 87-100

ДОДАТОК А

ПРОГРМАМА НА PYTHON

```
import time
import threading
import telebot
import sys
from multicolorcaptcha import CaptchaGenerator


bot_token = 'TOKEN'
bot = telebot.TeleBot(bot_token)
user_name = (sys.argv[1])
last_message_time = time.time()
running = True
db_user = {"vasylenkov": 'TELEGRAM_ID'}
characters =''




def CAPCTHA_make():
    global characters
    CAPCTHA_SIZE_NUM = 2
    generator = CaptchaGenerator(CAPCTHA_SIZE_NUM)
    captcha = generator.gen_captcha_image(difficult_level=2)
    image = captcha.image
    characters = captcha.characters

    return image
    # image.save("captcha\captcha.png", "png")

# Обработчик нажатия на кнопку
@bot.callback_query_handler(func=lambda call: True)
def button_click(call):
    # Проверяем идентификатор пользователя
    if call.message.chat.id == db_user[user_name] and call.data == "authorize":
        # bot.send_message(call.message.chat.id, "Авторизация успешна") #отправка сообщения в чат ( отпала нужда из-за алертов )
        bot.answer_callback_query(call.id, text='Аутентифікація вдала', show_alert=True)  # отправка алерта на екран
        bot.delete_message(call.message.chat.id,call.message.message_id)  # удаление сообщения с вопросом , что бы не засорять чат
        print("Accept") #сообщение о том , что авторизация успешна , которая в последствии передается радиусу
        bot.stop_bot()
        sys.exit()

    elif call.data == "recaptcha":
        bot.delete_message(call.message.chat.id, call.message.message_id)
        bot.send_photo(call.message.chat.id, CAPCTHA_make(),
                       "На ваш запит малюнок з числами було перестворено",
                       reply_markup=get_keyboard(2))

    elif call.data == "cancel":
        bot.answer_callback_query(call.id, text='Ви натиснули відміну аутентифікації', show_alert=True)
        bot.delete_message(call.message.chat.id,call.message.message_id)  # удаление сообщения с вопросом , что бы не засорять чат
```

```
        print("Reject")
        bot.stop_bot()
        sys.exit()




#Обробка вірності вводу капчі
@bot.message_handler(content_types=['text'])
def answer(message):
    # print(message)
    bot.delete_message(message.chat.id, message.message_id - 1)
    bot.delete_message(message.chat.id, message.message_id)
    if message.text== characters:
        bot.send_message(message.chat.id, 'Перевірка пройдена вдало\nНатисніть кнопку для кінцевої аутентифікації',
reply_markup=get_keyboard(1))
    else:
        bot.send_photo(message.chat.id, CAPCTHA_make(),
                       "Невірний ведені данні з малюнку. Повтіть спробу",
                       reply_markup=get_keyboard(2))



# Функция для создания клавиатуры с кнопкой для авторизации
def get_keyboard(button):
    keyboard = telebot.types.InlineKeyboardMarkup()
    if button == 1:
        button1 = telebot.types.InlineKeyboardButton(text="Аутентифікуватися", callback_data="authorize")
        keyboard.add(button1)
    if button == 2:
        button2 = telebot.types.InlineKeyboardButton(text="Відміна", callback_data="cancel")
        button3 = telebot.types.InlineKeyboardButton(text="Перегенерувати капчу", callback_data="recaptcha")
        keyboard.add(button3)
        keyboard.add(button2)
    return keyboard



# функция закрытия по времени
def timeout_bot(message):
    global last_message_time, running
    while True:
        if last_message_time is not None and (time.time() - last_message_time) >= 60: # установить время ( в сек.) через
которое выходить
            bot.delete_message(message.chat.id, message.message_id)            # удаление письма по истечению времени
            print("Reject")
            bot.stop_bot()                                      # остановка бота
            break
        elif running == False:           # условие выхода из цикла если пользователь авторизировался
            break
        time.sleep(1)



def main():
    global running
    first_message = bot.send_photo(db_user[user_name], CAPCTHA_make(),
                           "Введіть числа з малюнка, якщо бажаєте аутентифікуватися. Або натисніть відміну ",
                           reply_markup=get_keyboard(2))
```

```
    idle_thread = threading.Thread(target=timeout_bot, args=(first_message,))
    idle_thread.start()

    # Запуск бота
    bot.polling(timeout=3, long_polling_timeout=3)
    running = False
    # Ожидание завершения потока проверки бездействия
    idle_thread.join()



if __name__ == '__main__':
    main()
```