

Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра
на тему: «Проект підсистеми багатофакторної
автентифікації та обліку користувачів
відкритої кампусної мережі»

Виконав:

студент 4-го курсу групи КБ-21-1
спеціальності 125 «Кібербезпека»

 Юрій ЛІТВІНСЬКИЙ

Керівник:

Доцент кафедри КІ та КБ, к.т.н.

 Олексій ШЕЛУХА

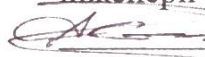
Рецензент:

Завідувач кафедри КІ та КБ, к.т.н., доцент

 Андрій ЄФІМЕНКО

Житомир – 2025

Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки
Спеціальність 125 «Кібербезпека»
Освітня програма «Кібербезпека»

ЗАТВЕРДЖУЮ
Завідувач
кафедри комп'ютерної
інженерії та кібербезпеки
 Андрій ЄФІМЕНКО
03 березня 2025 р.

ЗАВДАННЯ

на випускню кваліфікаційну роботу бакалавра

Студент: Літвінський Юрій Олегович

Тема роботи: «Проект підсистеми багатофакторної автентифікації та обліку користувачів відкритої кампусної мережі»

затверджена Наказом університету від 03 березня 2025 р. № 80/с

Термін здачі студентом закінченої роботи 24 червня 2025 р.

Вихідні дані роботи (зазначається предмет і об'єкт дослідження):

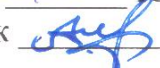
- процеси функціонування кампусної мережі;
- вимоги до централізованого обліку користувачів;
- методи, засоби та технології багатофакторної авторизації.

Обладнання та програмне забезпечення:

1. GNS3 (емулятор)
2. Ubuntu Server

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1			
2			

Студент  Юрій ЛІТВІНСЬКИЙ
Керівник  Олексій ШЕЛУХА

Календарний план

№ з/п	Назви етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка задачі	03.03.2025	Виконано
2	Пошук, огляд та аналіз аналогічних розробок	17.03.2025	Виконано
3	Формулювання технічного завдання	24.03.2025	Виконано
4	Опрацювання літературних джерел	21.04.2025	Виконано
5	Проектування мережі	05.05.2025	Виконано
6	Налагодження мережі та захисту	19.05.2025	Виконано
7	Перевірка працездатності мережі та захисту	26.05.2025	Виконано
8	Оформлення пояснювальної записки	09.06.2025	Виконано
9	Захист		

Студент  Юрій ЛІТВІНСЬКИЙ

Керівник  Олексій ШЕЛУХА

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи на тему «Проект підсистеми багатфакторної автентифікації та обліку користувачів відкритої кампусної мережі» містить вступ, технічне завдання, три основних розділи, висновки, список використаних джерел та додатки.

У першому розділі наведено постановку технічного завдання, в якій обґрунтовано необхідність впровадження багатфакторної автентифікації в умовах відкритої кампусної мережі. Проаналізовано архітектуру типової мережі університету, визначено вхідні дані, основні вимоги до системи та складові її реалізації. Узагальнено роль інформаційної безпеки та виділено ключові технології, що будуть використані.

Другий розділ присвячено аналітичному та проєктному етапу створення підсистеми. Детально описано принципи побудови компонентів багатфакторної авторизації, проведено обґрунтування вибору програмних засобів з відкритим кодом, проєктну структуру серверної частини та логіку взаємодії між компонентами системи. Особливу увагу приділено сумісності, масштабованості та безпечності запропонованого рішення.

У третьому розділі розглянуто процес практичної реалізації підсистеми в тестовому середовищі. Надано покрокове налаштування компонентів: OpenLDAP, FreeRADIUS, поштового сервера, Telegram-бота, механізму OTP та вебінтерфейсу для користувачів. Проведено перевірку працездатності шляхом моделювання сесій автентифікації та оцінено відповідність функціоналу поставленим вимогам.

Загальний обсяг пояснювальної записки становить 57 сторінок, включаючи додатки. У роботі представлено 62 рисунка. Список використаних джерел містить 19 найменувань.

Ключові слова: КОМП'ЮТЕРНА МЕРЕЖА, БАГАТОФАКТОРНА АУТЕНТИФІКАЦІЯ, LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL, ONE-TIME PASSWORD, КАМПУС, ІНФОРМАЦІЙНА БЕЗПЕКА.

					ККІКБ.КР.Б – 125 – 16 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Проект підсистеми багатофакторної автентифікації та обліку користувачів відкритої кампусної мережі	Літ.	Арк.	Аркушів
Розроб.		Літвінський Ю.О.		24.06				
Керівник		Шелуха О.О.		24.06			4	57
Реценз.		Єфіменко А.А.		27.06		ФІКТ, Гр. КБ-21-1		
Н. контр.		Шелуха О.О.		27.06				
Затверд.		Єфіменко А.А.		27.06				

ANNOTATION

The explanatory note for the bachelor's qualification project titled “Design of a Multi-Factor Authentication and User Accounting Subsystem for an Open Campus Network” includes an introduction, technical assignment, three main chapters, conclusions, a list of references, and appendices.

The first chapter presents the technical problem statement, substantiating the necessity of implementing multi-factor authentication in the context of an open campus network. It analyzes the architecture of a typical university network, defines the input data, outlines system requirements, and describes the key components and technologies involved. The role of information security is emphasized, and the core implementation tools are identified.

The second chapter focuses on analytical and design stages of the subsystem development. It provides a detailed description of the principles for constructing multi-factor authentication components, justifies the use of open-source software, and presents the logical structure of the server infrastructure. The interaction between system elements is described with an emphasis on compatibility, scalability, and security of the proposed solution.

The third chapter covers the practical implementation process within a virtual test environment. It includes the step-by-step configuration of components such as OpenLDAP, FreeRADIUS, a mail server, Telegram bot, OTP mechanism, and a user-facing web interface. Authentication sessions are simulated to verify functionality, and the system's compliance with the outlined requirements is assessed.

The explanatory note consists of 57 pages including appendices. The work contains 62 figures, and the list of references includes 19 sources.

Keywords: COMPUTER NETWORK, MULTI-FACTOR AUTHENTICATION, LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL, ONE-TIME PASSWORD, CAMPUS, INFORMATION SECURITY.

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ БАГАТОФАКТОРНОЇ АВТЕНТИФІКАЦІЇ ТА ОБЛІКУ КОРИСТУВАЧІВ	10
1.1. Аналіз сучасних загроз інформаційній безпеці у відкритих мережах	10
1.2. Сучасні підходи до автентифікації користувачів	13
1.3. Аналіз методів обліку користувачів у кампусних мережах	15
1.4. Постановка технічного завдання	17
Висновки до першого розділу	18
РОЗДІЛ 2. ПРОЄКТУВАННЯ КАМПУСНОЇ МЕРЕЖІ ТА ПІДСИСТЕМИ АВТЕНТИФІКАЦІЇ КОРИСТУВАЧІВ	20
2.1. Розробка архітектури кампусної мережі	20
2.2. Вибір методів та засобів реалізації проєкту кампусної мережі	23
2.3. Розробка структура підсистеми автентифікації	28
Висновки до другого розділу	31
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОЄКТУ СИСТЕМИ	32
3.1. Налаштування моделі середовища	32
3.1.1. Вибір обладнання	32
3.1.2. Налаштування	34
3.2. Тестування та оцінка ефективності розробленої моделі кампусної мережі	46
3.3. Аналіз роботи підсистеми автентифікації	49
Висновки до третього розділу	51
ВИСНОВКИ	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	54
ДОДАТКИ	56

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – Application Programming Interface — інтерфейс прикладного програмування

DNS – Domain Name System — система доменних імен

FreeRADIUS – Система автентифікації та авторизації, заснована на протоколі RADIUS

HOTP – HMAC-based One-Time Password — одноразовий пароль на основі хеш-функції

HTTPS – Hypertext Transfer Protocol Secure — захищений протокол передачі даних

IP – Internet Protocol — інтернет-протокол

LDAP – Lightweight Directory Access Protocol — протокол доступу до каталогів

MFA – Multi-Factor Authentication — багатофакторна автентифікація

OpenLDAP – Реалізація LDAP з відкритим вихідним кодом

OTP – One-Time Password — одноразовий пароль

QR – Quick Response — двовимірний штрихкод, зчитуваний камерою

RADIUS – Remote Authentication Dial-In User Service — віддалена служба автентифікації

RBAC – Role-Based Access Control — управління доступом на основі ролей

SMTP – Simple Mail Transfer Protocol — протокол передавання електронної пошти

TOTP – Time-based One-Time Password — одноразовий пароль на основі часу

UI – User Interface — користувацький інтерфейс

VPN – Virtual Private Network — віртуальна приватна мережа

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність розробки підсистеми багатофакторної автентифікації зумовлена стрімким зростанням кількості загроз, пов'язаних із несанкціонованим доступом до ресурсів освітніх мереж. Кампусні мережі, які є відкритими за своєю природою, обслуговують широке коло користувачів з різними правами доступу та потребами: від студентів до запрошених відвідувачів. Відсутність належного рівня захисту призводить до втрати конфіденційності, компрометації облікових записів, а також до порушення цілісності та доступності сервісів.

Проведений аналіз типових загроз, архітектури кампусної мережі та можливостей сучасних засобів автентифікації показує, що найефективнішим підходом є поєднання централізованої перевірки облікових даних із додатковим підтвердженням особи через одноразовий код. Такий підхід не лише підвищує рівень захисту, а й забезпечує гнучкість у виборі засобів автентифікації, що важливо в умовах різно-рідного користувацького середовища.

У зв'язку з цим постає необхідність створення універсальної, надійної та масштабованої підсистеми багатофакторної автентифікації, що базується на відкритих технологіях, відповідає вимогам безпеки та може бути впроваджена в реальну ІТ-інфраструктуру навчального закладу.

Метою роботи є розробка проєкту підсистеми багатофакторної автентифікації та обліку користувачів у відкритій кампусній мережі з урахуванням сучасних вимог до інформаційної безпеки, забезпеченням контролю доступу та надійного зберігання облікових даних.

Для досягнення поставленої мети в роботі необхідно вирішити такі основні **завдання**:

1. Провести аналіз актуальних загроз інформаційній безпеці в контексті відкритих кампусних мереж та виявити вразливості, пов'язані з автентифікацією користувачів.

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

2. Дослідити існуючі методи та технології автентифікації, зокрема багатofакторні підходи, з метою вибору оптимальних рішень для освітнього середовища.

3. Розробити проєкт підсистеми багатofакторної автентифікації та обліку користувачів із урахуванням обраних технологій і вимог до безпеки, масштабованості та сумісності.

4. Реалізувати та протестувати функціональну модель підсистеми в тестовому середовищі з подальшим аналізом її ефективності.

Об’єктом дослідження є процеси автентифікації та обліку користувачів у комп’ютерних мережах з відкритою архітектурою, зокрема в умовах кампусної мережі.

Предметом дослідження є технології та методи реалізації багатofакторної автентифікації (MFA) та обліку користувачів у відкритому мережевому середовищі, з урахуванням вимог інформаційної безпеки.

Таким чином, дана робота є актуальною з точки зору забезпечення інформаційної безпеки в умовах зростання кількості користувачів та пристроїв у відкритих кампусних мережах.

Практичне значення роботи полягає в тому, що розроблена підсистема багатofакторної автентифікації може бути використана фахівцями з кібербезпеки та системними адміністраторами для побудови захищених інформаційних середовищ в межах кампусних мереж із відкритим доступом.

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ БАГАТОФАКТОРНОЇ АВТЕНТИФІКАЦІЇ ТА ОБЛІКУ КОРИСТУВАЧІВ

1.1. Аналіз сучасних загроз інформаційній безпеці у відкритих мережах

У сучасному цифровому середовищі відкриті мережі, зокрема кампусні, стають об'єктами численних кіберзагроз. Зважаючи на зростаючу залежність освітніх закладів від інформаційних технологій, забезпечення інформаційної безпеки в кампусних мережах стає одним із ключових завдань адміністрацій та технічного персоналу. Такі мережі зазвичай охоплюють великі території, мають багато підключених пристроїв та часто не мають достатніх засобів централізованого контролю, що створює сприятливі умови для реалізації атак зловмисників.

Однією з найбільш небезпечних загроз є фішинг. Ця атака заснована на соціальній інженерії — тобто маніпуляції довірою користувача. Зловмисник надсилає електронний лист, який імітує офіційне повідомлення (наприклад, від адміністрації вишу або технічної підтримки), де користувача просять перейти за посиланням для зміни паролю або підтвердження облікового запису (Рис. 1.1). Посилання веде на фальшивий сайт, зовні схожий на справжній. Якщо користувач вводить свої облікові дані, вони потрапляють прямо до рук зловмисника. Після цього останній може отримати доступ до внутрішніх ресурсів мережі, включаючи особисту інформацію, журнали відвідувань, навчальні матеріали тощо [1]

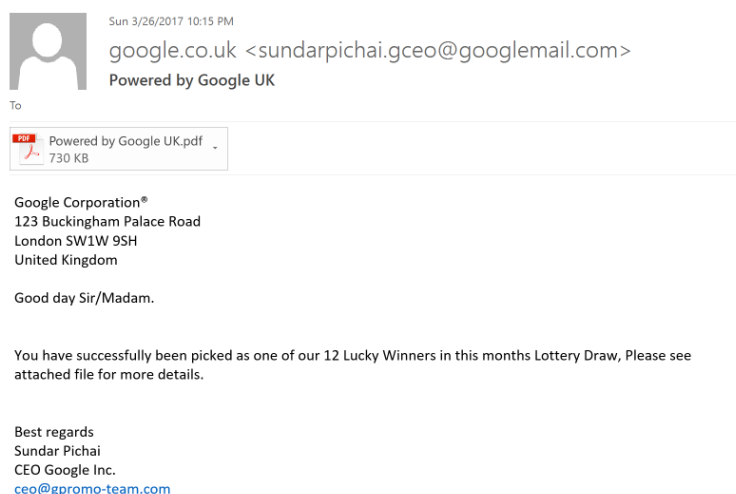


Рисунок 1.1. - Приклад фішингового листа

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

Інша серйозна загроза — витік даних. Це може статися через слабкий захист баз даних або помилки персоналу. Наприклад, відкритий доступ до адміністративних інтерфейсів або зберігання конфіденційної інформації на незахищених носіях. Витік може бути спричинений також діями зловмисників, які шляхом сканування вразливих вузлів виявляють відкриті порти або використовують відомі вразливості в програмному забезпеченні. Наслідками витоку можуть бути як репутаційні втрати для закладу, так і юридична відповідальність згідно з законодавством про захист персональних даних. Програми-вимагачі (ransomware) — це шкідливе ПЗ, яке шифрує файли на пристрої жертви та вимагає викуп за розшифрування (Рис. 1.2). У навчальному середовищі така атака може заблокувати доступ до розкладів, електронних журналів, систем дистанційного навчання. Часто програми-вимагачі потрапляють до системи через шкідливі вкладення в електронній пошті або через застаріле ПЗ без оновлень. Навіть після оплати викупу, немає гарантій, що файли буде відновлено [1]

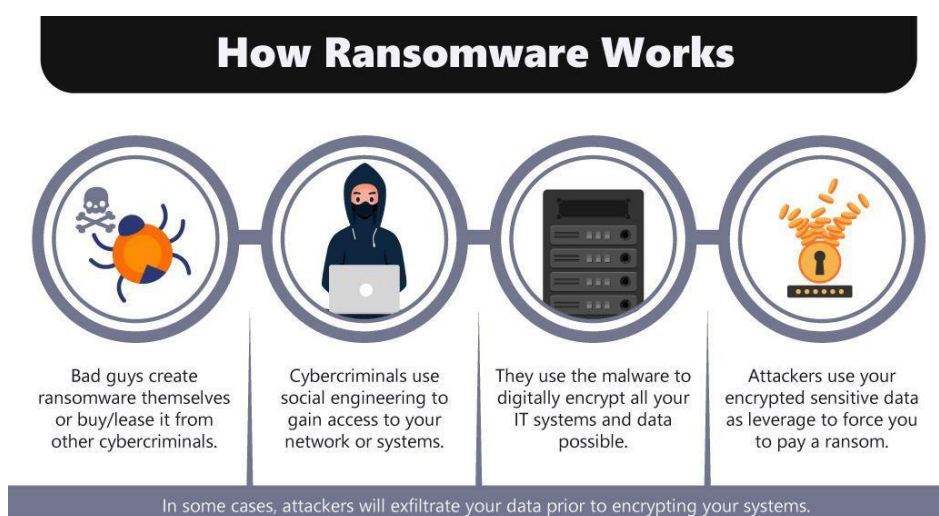


Рисунок 1.2. - Як працює програма-вимагач

Атаки типу DoS (відмова в обслуговуванні) та DDoS (розподілена відмова в обслуговуванні) спрямовані на виведення з ладу вебсайтів та онлайн-сервісів шляхом надсилення великої кількості запитів, які перевантажують сервер (Рис. 1.3). У кампусних умовах це може призвести до недоступності освітніх платформ у період

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

складання іспитів або подачі заяв. DDoS атаки зазвичай реалізуються за допомогою ботнетів — мереж заражених пристроїв, які діють одночасно. [1]

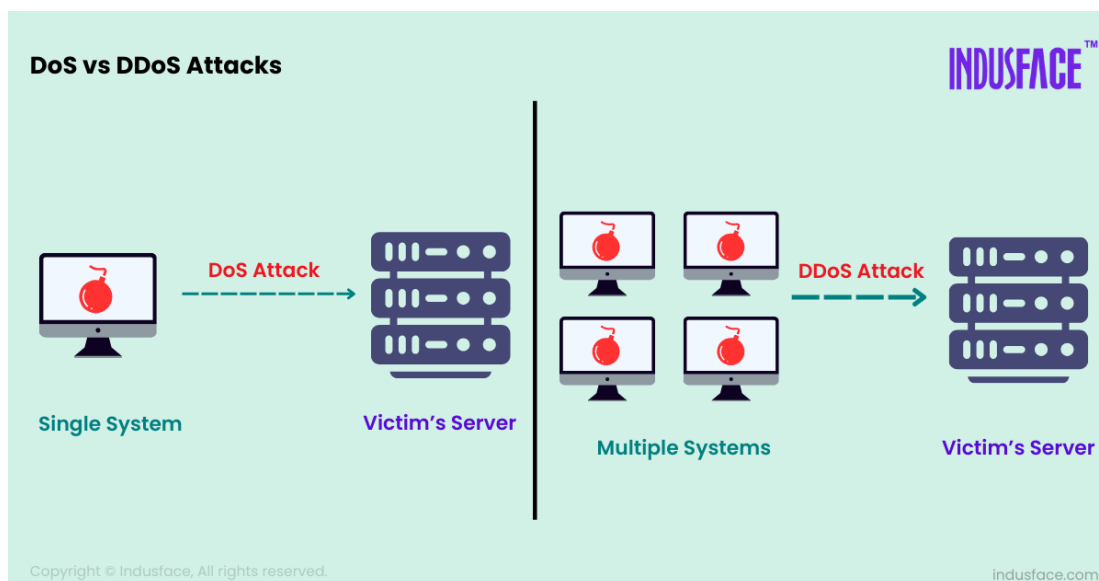


Рисунок 1.3. - Принцип роботи DoS/DDoS

Атаки типу MitM (man-in-the-middle, "людина посередині") реалізуються шляхом перехоплення даних між користувачем і сервером. Наприклад, у відкритій Wi-Fi мережі гуртожитку зловмисник може створити фальшиву точку доступу з ідентичною назвою. Коли користувач підключається до неї, всі його дії (включно з введенням паролів) проходять через пристрій зловмисника. Якщо не використовується захищене з'єднання (HTTPS, VPN), інформація може бути прочитана, змінена або записана. [2]

Соціальна інженерія — це техніка, при якій зловмисники використовують людський фактор для обходу технічних заходів безпеки. Наприклад, особа може зателефонувати співробітнику IT-відділу, представившись керівником і вимагаючи надати пароль або адміністративний доступ. Часто такі атаки виявляються ефективнішими за технічні методи, особливо в середовищах із низьким рівнем кіберграмотності. [3]

Застаріле ПЗ — один з найбільш недооцінених ризиків. Навіть відома і добре протестована операційна система може мати критичні вразливості, які виправляються лише в оновленнях. Якщо на навчальному сервері не встановлено оновлень

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

безпеки, це дає змогу атакувальникам легко скористатися цими дірами. Аналогічно, незахищені вебдодатки, створені студентами чи викладачами без належного контролю, також можуть бути вразливими. Внутрішні загрози часто залишаються поза увагою. Це можуть бути співробітники або студенти, які навмисно або випадково порушують політики безпеки. Наприклад, вони можуть завантажити конфіденційні дані на персональні пристрої, залишити комп'ютер без блокування або використовувати один пароль для всіх сервісів.

Усі ці загрози демонструють необхідність впровадження багаторівневої системи захисту, яка включає не лише технічні, але й організаційні заходи. Одним із ключових елементів такої системи є багатофакторна автентифікація, яка зменшує ризик несанкціонованого доступу навіть у випадку крадіжки пароля.

1.2. Сучасні підходи до автентифікації користувачів

Автентифікація є одним із ключових процесів в інформаційній безпеці, що забезпечує контроль доступу до систем і ресурсів. Під автентифікацією розуміють процес перевірки достовірності особи або об'єкта, який намагається отримати доступ до інформаційної системи. У найпростішому випадку це перевірка логіна та пароля, однак у сучасних умовах цього недостатньо для забезпечення належного рівня безпеки. Метою автентифікації є гарантія того, що суб'єкт доступу є саме тим, за кого себе видає. Вона лежить в основі багатьох моделей контролю доступу, включаючи моделі з ролями (RBAC), політиками (ABAC) тощо. Процес автентифікації передуює авторизації — тобто наданню прав доступу — і є обов'язковою частиною практично кожної інформаційної системи. Типи автентифікації класифікуються за тим, яку інформацію або дію користувач має надати для підтвердження своєї особи [4]:

1. Щось, що користувач знає — найпоширеніший тип, що включає паролі, PIN-коди, відповіді на контрольні запитання. Його основний недолік полягає у вразливості до фішингу, перехоплення та соціальної інженерії.

2. Щось, що користувач має — наприклад, смарт-картка, токен, телефон з додатком для генерації кодів (OTP), USB-ключ (наприклад, YubiKey). Такі засоби забезпечують вищий рівень безпеки, але можуть бути втрачені або викрадені.

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

3. Щось, чим користувач є — біометричні дані: відбитки пальців, скан обличчя, райдужної оболонки ока. Біометрія вважається зручною для користувача, проте має ризики, пов'язані з неможливістю змінити дані у випадку компрометації.

4. Контекстна (поведінкова) автентифікація — автентифікація на основі звичних для користувача дій або параметрів (місцезнаходження, пристрій, час доби, спосіб введення тексту тощо) [4].

Сучасні системи безпеки зазвичай використовують багатофакторну автентифікацію (Multi-Factor Authentication, MFA) — поєднання щонайменше двох із зазначених типів. Наприклад, введення пароля (знання) + підтвердження через додаток Google Authenticator (володіння). Це значно ускладнює дії зломисника, який повинен мати доступ до кількох джерел одночасно. [4]

Принципи побудови ефективної системи автентифікації включають:

- Унікальність — кожен користувач має отримувати доступ лише під своїм унікальним обліковим записом.
- Надійність — застосування складних паролів, криптографічного захисту, безпечного зберігання даних.
- Доступність — автентифікація не повинна створювати надмірні труднощі для легітимного користувача.
- Масштабованість — здатність системи працювати з великою кількістю користувачів і підключених пристроїв.
- Захист від підміни — мінімізація ризику імітації або крадіжки даних автентифікації. [5]

Системи автентифікації можуть бути централізованими (наприклад, на основі LDAP, RADIUS, Active Directory) або децентралізованими (наприклад, OAuth 2.0, OpenID Connect). Вибір залежить від потреб конкретної мережі та масштабів її використання.

У відкритих кампусних мережах особливу увагу приділяють адаптивним системам автентифікації, які автоматично враховують зміну поведінки користувача

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

або середовища та в разі виявлення аномалій — запитують додаткове підтвердження особи. Це дозволяє поєднати зручність користування із високим рівнем безпеки.

1.3. Аналіз методів обліку користувачів у кампусних мережах

Облік користувачів у кампусних мережах є критично важливою складовою забезпечення інформаційної безпеки та управління доступом. Кампусна мережа, яка обслуговує велику кількість студентів, викладачів, адміністративного персоналу та гостей, потребує надійної системи обліку, яка дозволяє не лише ідентифікувати користувачів, але й вести журнал їхньої активності, контролювати права доступу та забезпечувати аудит дій у мережі.

Одним із найпоширеніших підходів до обліку є використання технології AAA (Authentication, Authorization, Accounting). У цьому контексті "Accounting" або "облік" стосується процесу збору інформації про використання ресурсів: хто, коли, з якого пристрою підключився, які сервіси використовував, скільки часу був активним тощо. [6]

Для реалізації обліку широко застосовуються такі протоколи як RADIUS (Remote Authentication Dial-In User Service) та TACACS+ (Terminal Access Controller Access-Control System Plus). Протокол RADIUS дозволяє централізовано вести облік підключень користувачів до мережі, зокрема через Wi-Fi або VPN. Кожна спроба підключення супроводжується записом у журнал, який можна аналізувати в режимі реального часу або зберігати для подальшого аудиту. У середовищі відкритих кампусних мереж особливо важливою є інтеграція обліку з LDAP-сервісами (Lightweight Directory Access Protocol), які забезпечують централізоване зберігання даних про користувачів. Наприклад, FreeRADIUS може використовуватись разом з OpenLDAP для автентифікації та обліку студентів за допомогою єдиної бази даних. Це дозволяє адміністраторам контролювати доступ до ресурсів залежно від ролі користувача (студент, викладач, гість) та обмежувати можливості підключення.

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

Ще одним ефективним методом обліку є використання журналювання подій (log management). Наприклад, системи на базі syslog-ng або Graylog дозволяють зберігати та аналізувати журнали з усіх мережевих пристроїв, точок доступу, шлюзів тощо. Це дає змогу оперативно реагувати на аномалії та несанкціоновану активність у мережі.

У сучасних умовах набирають популярності SIEM-системи (Security Information and Event Management), які об'єднують облік, моніторинг і аналіз подій безпеки в єдину платформу (Рис. 1.4). Такі системи як Splunk, Wazuh або IBM QRadar дозволяють не лише збирати інформацію про активність користувачів, але й виявляти підозрілу поведінку на основі кореляції подій з різних джерел. [7]

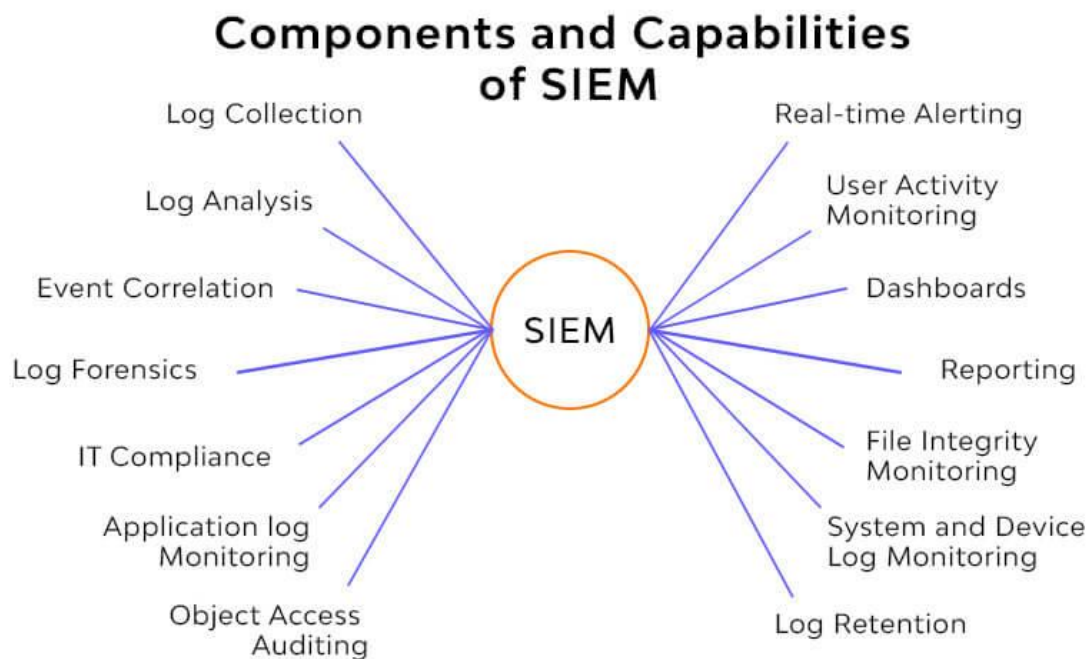


Рисунок 1.4. - Компоненти SIEM

Крім технічних засобів, важливо також забезпечити регламент обліку — тобто документовані правила зберігання журналів, період їх ротації, обмеження доступу до логів та порядок їхнього використання у випадках розслідування інцидентів.

Таким чином, ефективний облік користувачів у кампусних мережах має поєднувати:

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				16
Змн.	Арк.	№ докум.	Підпис	Дата		

- централізовану автентифікацію та авторизацію,
- детальне журналювання подій,
- системи аналізу та реагування на інциденти,
- відповідні політики безпеки та доступу до логів.

Тільки комплексний підхід до обліку дозволяє забезпечити прозорість, контроль і безпеку в умовах відкритої динамічної мережі.

1.4. Постановка технічного завдання

У сучасних умовах цифрової трансформації та зростання загроз інформаційній безпеці освітні установи дедалі частіше стикаються з потребою впровадження ефективних засобів захисту користувацького доступу. Кампусні мережі, що є відкритими за своєю природою та охоплюють значні території навчальних закладів, потребують впровадження надійних рішень для автентифікації користувачів та обліку їх дій. Одним із пріоритетних напрямів у цьому контексті є реалізація багатфакторної автентифікації (MFA) із централізованим управлінням обліковими записами.

Підсистема повинна функціонувати у межах загальної мережевої архітектури навчального закладу, включаючи користувацький сегмент (студенти, викладачі), серверний сегмент (служби автентифікації, обліку, моніторингу), а також адміністративні вузли доступу. Користувач, що намагається отримати доступ до мережних сервісів, має бути ідентифікований з використанням логіна та пароля, а також одноразового коду, надісланого на електронну пошту або через месенджер Telegram.

Основні вимоги до системи включають:

- забезпечення централізованого зберігання облікових записів із використанням служби каталогів OpenLDAP;
- реалізацію первинної автентифікації на основі облікових даних з LDAP та вторинної перевірки одноразовим кодом (OTP), який надсилається користувачу через вибраний канал;

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

- інтеграцію служби автентифікації FreeRADIUS для обробки запитів доступу та обліку активності користувачів;
- можливість вибору способу доставки OTP (email або Telegram) через зручний вебінтерфейс;
- ведення журналу подій автентифікації, фіксація часу, IP-адреси, методу доступу та результату перевірки;
- налаштування тестового середовища у віртуалізованій інфраструктурі (VirtualBox, GNS3) для перевірки працездатності підсистеми без необхідності використання фізичного обладнання;
- використання вільного програмного забезпечення та відкритих протоколів, що забезпечить гнучкість, доступність та можливість подальшої модифікації системи.

Загалом, створювана підсистема повинна забезпечувати надійний рівень автентифікації з мінімальним навантаженням на користувача, відповідати принципам безпечного зберігання облікових даних, бути адаптивною до умов змінного середовища, а також підтримувати розширення та модернізацію в разі зміни вимог до інформаційної безпеки організації. Результатом реалізації має стати прототип системи, що підтверджує функціональність, ефективність і перспективність запропонованого підходу.

Висновки до першого розділу

В ході написання першого розділу було проведено комплексний аналіз теоретичних засад, пов'язаних із забезпеченням автентифікації та обліку користувачів у відкритих мережах, зокрема в контексті кампусного середовища. Розглянуто актуальні загрози інформаційній безпеці, які найчастіше реалізуються у відкритих та недостатньо захищених мережах. Серед них — фішинг, атаки типу MitM, шкідливе програмне забезпечення, соціальна інженерія та внутрішні загрози, що свідчить про необхідність багаторівневого захисту.

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

Окрему увагу приділено поняттю автентифікації, її видам та принципам реалізації. Наголошено, що багатофакторна автентифікація (MFA) є однією з найефективніших практик сучасної кібербезпеки, що дозволяє значно зменшити ризик несанкціонованого доступу навіть у випадках компрометації облікових даних.

Також було проаналізовано методи обліку користувачів, які реалізуються в кампусних мережах із використанням RADIUS, LDAP, SIEM-систем та механізмів журналювання. Встановлено, що ефективний облік забезпечує прозорість дій користувачів, аудит доступу та моніторинг аномальної активності.

Загалом, теоретичні основи, викладені в цьому розділі, формують базу для подальшого проєктування та реалізації підсистеми багатофакторної автентифікації і обліку користувачів, що буде розглянуто у наступних розділах.

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЄКТУВАННЯ КАМПУСНОЇ МЕРЕЖІ ТА ПІДСИСТЕМИ АВТЕНТИФІКАЦІЇ КОРИСТУВАЧІВ

2.1. Розробка архітектури кампусної мережі

Кампусна мережа — це телекомунікаційна інфраструктура навчального закладу, що охоплює кілька будівель, включаючи навчальні корпуси, бібліотеки, лабораторії, гуртожитки та адміністративні приміщення (Рис. 2.1). Вона створюється для забезпечення стабільного, масштабованого та безпечного доступу до освітніх, інформаційних і сервісних ресурсів університету як для постійних користувачів (студенти, викладачі, персонал), так і для тимчасових (гості, партнери, абітурієнти).

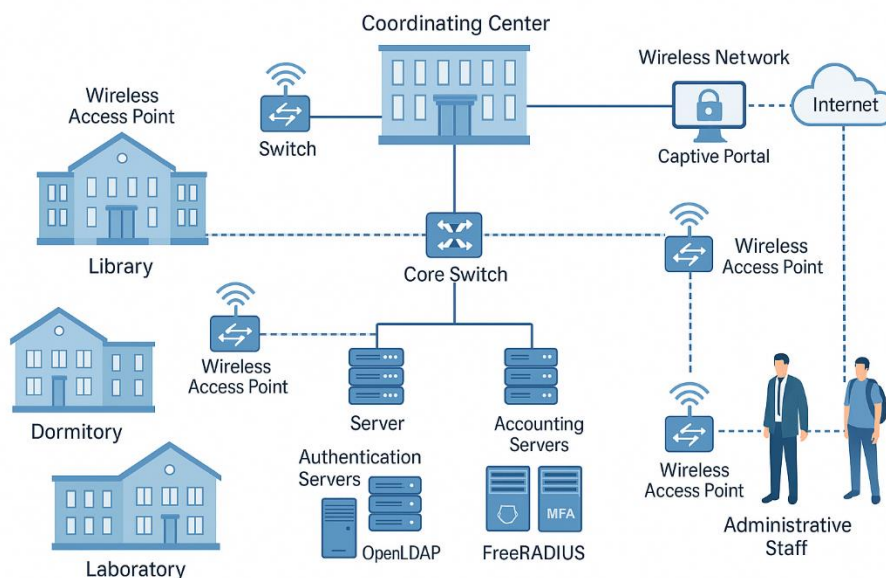


Рисунок 2.1. - Візуальне представлення кампусної мережі

Особливістю відкритих кампусних мереж є їх публічність та широкий діапазон пристроїв, що підключаються, зокрема ноутбуки, смартфони, планшети, розумні пристрої тощо. Це створює серйозні виклики з точки зору інформаційної безпеки, управління ідентифікацією користувачів, запобігання несанкціонованому доступу та ведення обліку дій у мережі.

Загальна архітектура такої мережі базується на багаторівневій структурі та включає:

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

- ядро мережі: комутатори та маршрутизатори, що забезпечують високу пропускну здатність і надійність;
- серверна інфраструктура: центральні сервери аутентифікації, каталогу, обліку та моніторингу;
- точки доступу Wi-Fi: встановлені у зонах загального користування, навчальних аудиторіях та гуртожитках;
- інструменти безпеки: міжмережеві екрани, системи виявлення вторгнень (IDS), сегментація мережі за допомогою VLAN;
- користувацька частина: пристрої, що належать студентам, викладачам, адміністрації.

Для управління доступом до ресурсів у такій архітектурі важливе значення має впровадження механізмів автентифікації та обліку. Базова перевірка логіна і пароля вже не є достатньою, особливо у відкритому середовищі. Тому все більш актуальним стає використання багатофакторної автентифікації (MFA), яка передбачає перевірку двох або більше факторів: знання (пароль), володіння (смартфон, токен), біометричні ознаки (відбиток пальця, обличчя).

Крім того, сучасна мережева архітектура передбачає централізоване управління обліковими записами користувачів, використання LDAP-серверів (наприклад, OpenLDAP або Microsoft Active Directory), а також контроль доступу за допомогою RADIUS-серверів (наприклад, FreeRADIUS), що дозволяє збирати статистику, контролювати спроби входу та оперативно реагувати на інциденти.

Запропонована архітектура дозволяє змодельовати основні виклики відкритого середовища — безпечне підключення користувачів, захист від несанкціонованого доступу, облік активності та підтримку масштабованості.

При проектуванні була використана ієрархічна модель, яка включає в себе три рівня:

- рівень доступу (access layer);
- рівень розподілу / агрегації (distribution layer);
- рівень ядра (core layer).

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				21
Змн.	Арк.	№ докум.	Підпис	Дата		

Для кожного рівня визначені свої функції. Три рівня не припускають наявності 3 різних пристроїв. Може бути як кілька пристроїв, так і один пристрій, яке виконує всі функції, визначені на двох сусідніх рівнях.

Рівень ядра знаходиться на самому верху ієрархії і відповідає за надійну і швидку передачу великих обсягів даних. Трафік, що передається через ядро, загальний для більшості користувачів. Призначені для користувача дані обробляються на рівні розподілу. Він при необхідності, пересилає запити до ядра.

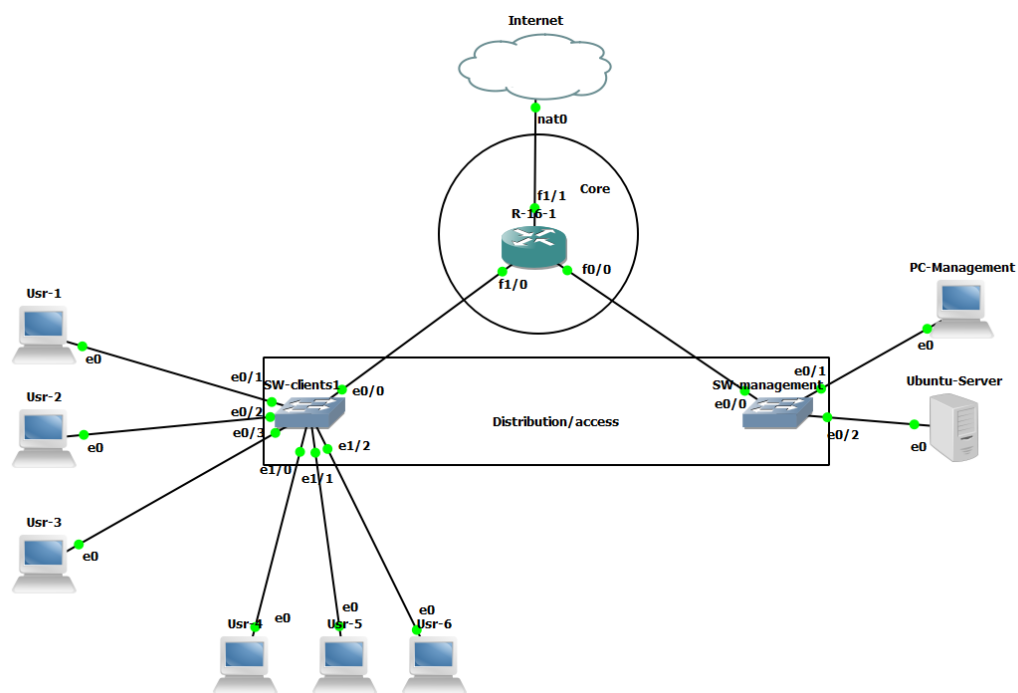


Рисунок 2.2. - Топологія мережі

У даній топології кампусної мережі виділено два логічні сегменти:

1. Сегмент користувачів (SW-clients1):

- Включає робочі станції Usr-1 до Usr-6;
- Підключені до комутатора SW-clients1;
- Використовується для моделювання поведінки студентів або працівників кампусу;
- Передбачена VLAN 10 (опційно).

2. Сегмент адміністрування та сервісів (SW-management):

- Включає Ubuntu-Server-1 та PC-Management;

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

- Підключені до комутатора SW-management;
- Ubuntu-Server-1 працює як LDAP+RADIUS+SMTP сервер;
- PC-Management використовується для моніторингу і адміністрування;
- Передбачена VLAN 20 (опційно).

Ці два сегменти з'єднані через маршрутизатор R-16-1, який також має доступ до зовнішньої мережі (Internet).

Таблиця 2. 1.

IP-адресація

Пристрій	Інтерфейс	IP-адреса	Мережа / VLAN
Usr-1	Мережевий адаптер	DHCP	192.168.10.0/24 (VLAN 10)
Usr-2	Мережевий адаптер		192.168.10.0/24 (VLAN 10)
Usr-3	Мережевий адаптер		192.168.10.0/24 (VLAN 10)
Usr-4	Мережевий адаптер		192.168.10.0/24 (VLAN 10)
Usr-5	Мережевий адаптер		192.168.10.0/24 (VLAN 10)
Usr-6	Мережевий адаптер		192.168.10.0/24 (VLAN 10)
SW-clients1	VLAN1	192.168.10.2	192.168.10.0/24 (VLAN 10)
R-16-1	Fa1/0	192.168.10.1	192.168.10.0/24 (VLAN 10)
	Fa0/0	192.168.20.1	192.168.20.0/24 (VLAN 20)
SW-management	VLAN1	192.168.20.2	192.168.20.0/24 (VLAN 20)
Ubuntu-Server	Мережевий адаптер	192.168.20.10	192.168.20.0/24 (VLAN 20)
PC-Management	Мережевий адаптер	192.168.20.11	192.168.20.0/24 (VLAN 20)
Internet (Cloud)	-	DHCP/ NAT	Зовнішня мережа

2.2. Вибір методів та засобів реалізації проєкту кампусної мережі

Для розробки підсистеми багатофакторної автентифікації та обліку користувачів відкритої кампусної мережі було прийнято рішення реалізувати найпростішу,

стабільну та доступну конфігурацію з мінімально необхідним функціоналом. Основною метою є забезпечення базової перевірки особи користувача (логін, пароль) та реалізація додаткового підтвердження через одноразовий код, надісланий електронною поштою. У фокусі — відкриті технології та простота налаштування.

1. OpenLDAP — сервер каталогів

OpenLDAP — це реалізація протоколу LDAP з відкритим вихідним кодом, яка широко використовується в Linux-середовищах як централізоване сховище інформації про користувачів. Він дозволяє створювати ієрархічні структури, зберігати атрибути акаунтів (паролі, групи, контактні дані), та ефективно працює у зв'язці з іншими сервісами автентифікації, зокрема FreeRADIUS, Kerberos, PAM тощо.

Хоча OpenLDAP є гнучким і розширюваним, він вимагає ручного налаштування, добре знання LDIF-структур, та не має зручного графічного інтерфейсу для адміністрування. У багатьох випадках адміністратори створюють або використовують окремі веб-інтерфейси, як-от phpLDAPadmin або FusionDirectory.

У порівнянні з Microsoft Active Directory (AD), OpenLDAP виграє у відкритості, однак поступається в частині зручності адміністрування. AD інтегрується з Windows-мережею «з коробки», має централізовані інструменти управління через RSAT (Remote Server Administration Tools) та підтримує складні політики групової безпеки (GPO). Водночас, AD вимагає використання платних серверних ОС та обмежений в роботі з Linux-клієнтами без додаткової настройки (наприклад, через SSSD, realmd).

FreeIPA є ще одним поширеним варіантом, який комбінує LDAP-сервер, Kerberos, DNS та сервіси управління сертифікатами (PKI). Це повноцінна система ідентифікації та управління доменом для Linux/Unix-мереж, яка має зручну WebUI і автоматичну інтеграцію компонентів. FreeIPA особливо вигідна в середовищах з великою кількістю Linux-клієнтів, де потрібна централізована автентифікація та керування доступом.

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		24

Таблиця 2. 2.

Порівняння ключових характеристик серверів каталогів

Характеристика	OpenLDAP	Microsoft AD	FreeIPA
Тип ліцензії	Відкрита (Open Source)	Пропрієтарна	Відкрита (GPL)
Платформа	Linux/Unix	Windows Server	Linux (RHEL, Fedora)
Інтеграція з Linux	Відмінна	Обмежена	Відмінна
Підтримка схеми LDAP	Так	Так	Так
Інтеграція з Kerberos	Так	Так (вбудовано)	Так (вбудовано)
Підтримка Windows клієнтів	Обмежена	Відмінна	Обмежена
Адміністративний інтерфейс	CLI/LDAP Tools	GUI (MMC, RSAT)	WebUI, CLI
Простота налаштування	Середня	Висока	Висока
Підтримка MFA	Через PAM/RADIUS	Вбудовано	Вбудовано (OTP)

2. FreeRADIUS — сервер автентифікації та обліку

FreeRADIUS — це провідна реалізація протоколу RADIUS з відкритим кодом, яка підтримує інтеграцію з LDAP, Active Directory, базами даних, PAM-модулями та OTP. Він надзвичайно масштабований, підтримує тисячі запитів на секунду, а також широко застосовується для автентифікації у Wi-Fi, VPN, корпоративних мережах і університетських кампусах.

На відміну від платних аналогів, FreeRADIUS не має веб-інтерфейсу за замовчуванням — налаштування здійснюється через конфігураційні файли. Це робить його більш гнучким, але потребує вищого рівня підготовки адміністратора. Логування є гнучким і повним: можна фіксувати спроби входу, авторизації, час сесії та облік трафіку.

Cisco ISE — це корпоративне рішення для контролю доступу, яке підтримує RADIUS і TACACS+, має зручну WebUI, розширену аналітику і вбудовану

підтримку інтеграції з MFA (через DUO, SAML). Однак воно є комерційним продуктом, що вимагає апаратної або віртуальної платформи Cisco, має високу вартість ліцензування та орієнтоване на великі корпоративні середовища.

Microsoft NPS — це компонент Windows Server, що реалізує RADIUS-сервер для автентифікації користувачів AD. Він має базову інтеграцію з Windows-інфраструктурою, але слабші можливості масштабування та логування. Крім того, NPS складно інтегрується з Linux або відкритими системами.

Таблиця 2. 3.

Порівняння ключових характеристик серверів автентифікації та обліку

Характеристика	FreeRADIUS	Cisco ISE (TACACS+)	Microsoft NPS
Тип ліцензії	Відкрита (GPL)	Пропрієтарна (Cisco)	Пропрієтарна (MS)
Підтримка протоколів (RADIUS/TACACS+)	RADIUS	RADIUS, TACACS+	RADIUS
Інтеграція з LDAP/AD	Так (OpenLDAP, AD)	Так (через AD/LDAP)	Так (Active Directory)
Підтримка MFA	Так (через PAM, OTP)	Так (через DUO, SAML)	Обмежено (через сторонні сервіси)
Масштабованість	Висока	Висока	Середня
Інтерфейс керування	CLI / файли конфігурації	Web GUI	GUI (MMC)
Можливості логування	Повна (доступ, помилки, accounting)	Повна + аналітика	Обмежена
Сумісність з Wi-Fi/VPN	Так	Так	Так
Інсталяція на Linux	Так (рідна підтримка)	Ні (треба окрема апаратна платформа)	Ні (потрібен Windows Server)

3. Email OTP / Статичний код — другий фактор автентифікації

У спрощеній реалізації другий фактор представлений у вигляді одноразового коду, що:

- або зберігається у внутрішньому полі LDAP-облікового запису (наприклад, userOTP), і перевіряється через policy.conf;

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				26
Змн.	Арк.	№ докум.	Підпис	Дата		

- або генерується вручну і надсилається на електронну пошту користувача перед автентифікацією.

Email OTP — це один із найпростіших способів реалізації другого фактору. Код генерується на сервері (випадковий рядок, наприклад 6 цифр) і надсилається користувачу на заздалегідь визначену адресу. Основною перевагою є простота реалізації — достатньо скрипта на bash або Python, що викликає `mailx`, `ssmtp` або `sendmail`. Недоліком є залежність від інтернет-з'єднання та потенційна вразливість каналу доставки (email може бути перехоплено).

Telegram OTP реалізується через бота, що надсилає код користувачу в чат. Перевагою є швидкість доставки та простота для користувача. Недоліки — потреба в постійному інтернеті, залежність від стороннього сервісу (Telegram), а також необхідність підтримувати бот-сервер.

Google Authenticator базується на TOTP (часовий код) і є одним із найбільш безпечних і популярних рішень. Реалізується через PAM-модуль у Linux. Коди генеруються локально, не потребують інтернету, однак потрібен мобільний додаток і складніша початкова конфігурація (QR-код, секрети).

YubiKey — це апаратний ключ, який реалізує OTP, FIDO2/WebAuthn та інші стандарти. Найвищий рівень безпеки, однак вартість одного ключа — від 40\$. Підходить для організацій з підвищеними вимогами до захисту.

Статичний код — найпростіший метод, що базується на збереженні пароля в LDAP або конфігураційному файлі. Не змінюється автоматично, не є одноразовим і не має захисту від повторного використання. Його можна використовувати як тестовий або тимчасовий варіант MFA.

Таблиця 2. 4.

Порівняння ключових характеристик різних способів реалізації одноразових паролів

Характеристика	Email OTP	Telegram OTP	Google Authenticator	YubiKey	Статичний код
Тип методу	Код на email	Через бот	TOTP на мобільному	Апаратний ключ	Фіксований пароль

Вартість впровадження	Низька	Середня	Низька	Висока	Дуже низька
Простота реалізації	Висока	Середня	Середня	Низька	Висока
Надійність/Безпека	Середня	Середня	Висока	Дуже висока	Низька
Потреба в інтернеті	Так	Так	Ні	Ні	Ні
Можливість інтеграції з Linux	Так (через mailx/sendmail)	Так (через API-бота)	Так (libram-google-authenticator)	Так (через PAM)	Так (файл/LDAP)
Потреба у додатку/обладнанні	Ні	Ні	Так (моб. додаток)	Так (обладнання)	Ні
Підтримка одноразовості	Так (часовий/одноразовий)	Так	Так (часовий код)	Так	Ні
Складність для користувача	Низька	Середня	Середня	Низька	Низька

4. Серверне середовище

Ubuntu Server 22.04 LTS було обрано як базову платформу через:

- стабільність, регулярні оновлення безпеки;
- доступність у вигляді ISO-образів для емуляції в GNS3/VirtualBox;
- підтримку OpenLDAP і FreeRADIUS у стандартних репозиторіях.

Це забезпечує швидку інсталяцію, можливість автоматизації налаштувань і широке використання документації спільноти.

2.3. Розробка структура підсистеми автентифікації

Розроблена підсистема автентифікації та обліку користувачів базується на трьох ключових компонентах: OpenLDAP, FreeRADIUS та механізмі підтвердження другого фактора у вигляді одноразового або статичного коду (Рис. 2.3).

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				28
Змн.	Арк.	№ докум.	Підпис	Дата		

Структура побудована з урахуванням спрощеного розгортання у віртуальному середовищі та забезпечує базову модель багатфакторної автентифікації з можливістю масштабування.

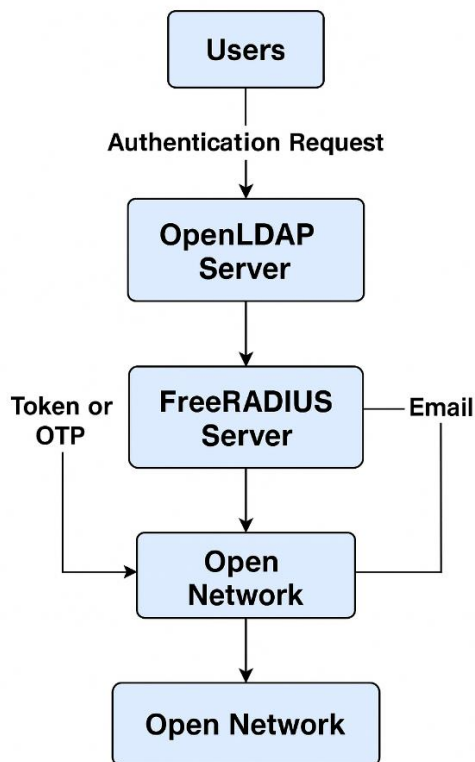


Рисунок 2.3. - Взаємодія компонентів

Основні компоненти підсистеми:

1. OpenLDAP — сервер каталогів, що зберігає облікові записи користувачів, а також додаткові атрибути (наприклад, поле для OTP-коду). LDAP виступає джерелом достовірної інформації про особу користувача.

2. FreeRADIUS — основний механізм обробки запитів автентифікації. Він взаємодіє з OpenLDAP для перевірки логіна та пароля. Крім цього, FreeRADIUS налаштовано на додаткову перевірку поля `userOTP`, `otpPassword` або іншого атрибута, де зберігається другий фактор автентифікації (наприклад, код, який надсилається користувачу електронною поштою).

3. Email-сервер (локальний або хмарний) — використовується для надси- лання OTP-кодів на пошту користувача перед спробою входу. У спрощеній версії

цей етап може бути замінено ручним додаванням значення OTP у поле LDAP-акаунта.

4. Користувач — взаємодіє з системою через клієнтський пристрій (наприклад, термінал, Captive Portal, VPN або будь-який RADIUS-сумісний сервіс). Користувач вводить логін, пароль і код підтвердження.

Алгоритм взаємодії компонентів:

1. Користувач надсилає запит на автентифікацію (через radtest, Captive Portal, VPN тощо).

2. FreeRADIUS перевіряє логін і пароль користувача через LDAP.

3. Після цього виконується додаткова перевірка — збірка введеного коду з OTP-полем у LDAP.

4. У разі успішного збігу — автентифікація проходить, а доступ дозволяється.

5. Всі дії логуються для подальшого аудиту.

Безпека підсистеми:

- LDAP-з'єднання може бути захищене за допомогою TLS (LDAPS), що запобігає перехопленню облікових даних.

- FreeRADIUS підтримує перевірку клієнтів за IP та спільним ключем (`shared secret`), що знижує ризик несанкціонованих запитів.

- Email-повідомлення з OTP можна обмежити в часі (наприклад, термін дії коду 5 хвилин) або одноразовістю.

- Логування включає збереження спроб автентифікації, часу, IP-адреси клієнта та результату перевірки, що дозволяє ефективно виявляти підозрілу активність.

Архітектура є модульною: можлива заміна другого фактора на інший (Telegram, Google Authenticator, SMS), або впровадження бази даних замість LDAP у масштабованих версіях. Також може бути додано веб-інтерфейс для адміністрування облікових записів.

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				30
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновки до другого розділу

У цьому розділі було здійснено поетапне проектування підсистеми багатофакторної автентифікації та обліку користувачів відкритої кампусної мережі. Зокрема, описано загальну архітектуру мережі, що охоплює взаємодію між сервером автентифікації, сховищем облікових даних (OpenLDAP), та клієнтськими пристроями.

Обґрунтовано вибір відкритих технологій: OpenLDAP для централізованого зберігання облікових записів, FreeRADIUS як потужного AAA-сервера та Email OTP або статичного коду як простого способу реалізації другого фактору автентифікації. Проведено порівняння із сучасними аналогами кожного компонента (Active Directory, Cisco ISE, FreeIPA, Google Authenticator тощо), що дало змогу переконатися в доцільності обраної конфігурації з огляду на простоту, відкритість, гнучкість та можливість впровадження в навчальному чи тестовому середовищі.

У результаті спроектовано модульну, масштабовану структуру підсистеми, яка легко адаптується під потреби конкретної організації та може слугувати базою для подальшої розширеної реалізації в рамках повноцінного проєкту. Розроблене проєктне рішення є оптимальним компромісом між функціональністю, безпекою та простотою впровадження.

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				31
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОЄКТУ СИСТЕМИ

3.1. Налаштування моделі середовища

3.1.1. Вибір обладнання

Маршрутизатор

У сучасному світі комп'ютерних мережі стають все більш складними та потужними. Мережеві пристрої, такі як маршрутизатори, відіграють важливу роль у забезпеченні ефективного передачі даних і забезпеченні з'єднання між різними підмережами. Один з таких потужних маршрутизаторів - Cisco 7206VXR, що відноситься до серії маршрутизаторів Cisco 7200 (Рис. 3.1).



Рисунок 3.1. - Маршрутизатор Cisco 7206VXR

Cisco 7206VXR має високу продуктивність і здатний обробляти великі обсяги трафіку. Він оснащений потужним процесором і може пропускати до 2 мільйонів пакетів на секунду. Цей маршрутизатор підтримує широкий спектр інтерфейсів зв'язку, таких як Ethernet, Fast Ethernet, Gigabit Ethernet, серійний, Token Ring і інші. Cisco 7206VXR може бути розширений для задоволення зростаючих потреб мережі. Він підтримує велику кількість маршрутів і може бути конфігурований для маршрутизації на основі різних протоколів, таких як OSPF, EIGRP, BGP та інші. Крім того, він може бути використаний як центральний маршрутизатор або в кластерному режимі для балансування навантаження.

Однак, незважаючи на свою популярність у минулому, Cisco 7206VXR є застарілим обладнанням, яке не рекомендовано до використання в сучасних розгортаннях через обмежену продуктивність, енергоспоживання та відсутність підтри-

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		32

мки нових протоколів. В реальному середовищі більш доцільним було б використання сучасніших пристроїв, таких як серії Cisco ISR 4000 або Catalyst 8000, які мають вищу продуктивність, підтримку віртуалізації, інтеграцію з хмарними сервісами та оновлену систему безпеки.

Використання Cisco 7206VXR у межах проєкту зумовлене зручністю емуляції, що дозволяє ефективно імітувати її роботу в тестовому середовищі без потреби у фізичному обладнанні. Це дає можливість розгорнути реалістичну інфраструктуру з маршрутизацією, NAT, ACL та іншими функціями, необхідними для перевірки роботи підсистеми автентифікації.

Комутатори

Cisco WS-C2960-24TT-L комутатор фіксованої конфігурації, здатний функціонувати в мережах компаній та організацій малого та середнього розміру (Рис. 3.2).



Рисунок 3.2. - Комутатор Cisco WS-C2960-24TT-L

Великий набір інтелектуальних технологій і механізмів, реалізований у цьому пристрої, включає розширену якість обслуговування, вбудовану безпеку, регулювання ширини каналу, а також надійне програмне забезпечення. Воно, у свою чергу, включає і Network Admission Control, а також здатне автоматично проводити оновлення та встановлення налаштувань за замовчуванням. Комутатор легкий в установці та управлінні. Для керування використовується консольний порт або веб-інтерфейс. Для початку роботи не потрібні спеціальні налаштування.

Безпека здійснюється також за допомогою автентифікації користувачів та контролю мережі. У оснащенні моделі Cisco WS-C2960-24TT-L знаходиться 24 порти, що працюють на швидкості 10/100 Мбіт/с, 2 гігабітні порти для високошвидкісного з'єднання. Загальна пропускна здатність сягає 16 Гбіт/с.

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		33

Робочі станції та сервери

Для моделювання клієнтських пристроїв було обрано дистрибутив Kali Linux, що дозволяє ефективно перевірити на практиці процес автентифікації, використовуючи вбудовані утиліти командного рядка (наприклад, radtest, nmap, curl) та графічні інтерфейси. Kali Linux є зручною платформою для тестування безпеки, а також містить готові інструменти для взаємодії з сервером FreeRADIUS.

У ролі сервера було використано Ubuntu Server 22.04 LTS – стабільний, широко підтримуваний дистрибутив Linux, який ідеально підходить для розгортання служб автентифікації, серверів каталогів (OpenLDAP) та поштових рішень (mailutils, Postfix). Він дозволяє налаштувати повноцінне середовище для реалізації централізованої автентифікації користувачів з використанням сучасних інструментів з відкритим вихідним кодом.

3.1.2. Налаштування

Проведемо базове налаштування мережевого обладнання:

```
R-16-l#enable
R-16-l#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R-16-l(config)#
R-16-l(config)#interface f0/0
R-16-l(config-if)# ip address 192.168.20.1 255.255.255.0
R-16-l(config-if)# no shutdown
R-16-l(config-if)#
R-16-l(config-if)#interface f1/0
R-16-l(config-if)# ip address 192.168.10.1 255.255.255.0
R-16-l(config-if)# no shutdown
R-16-l(config-if)#
R-16-l(config-if)#interface f1/1
R-16-l(config-if)# ip address dhcp
R-16-l(config-if)# no shutdown
R-16-l(config-if)#
R-16-l(config-if)#ip dhcp excluded-address 192.168.10.1 192.168.10.9
R-16-l(config)#ip dhcp pool USERS
R-16-l(dhcp-config)# network 192.168.10.0 255.255.255.0
R-16-l(dhcp-config)# default-router 192.168.10.1
R-16-l(dhcp-config)# dns-server 192.168.10.1
R-16-l(dhcp-config)# lease 0 4
R-16-l(dhcp-config)#
R-16-l(dhcp-config)#interface f1/1
R-16-l(config-if)# ip nat outside

*Jun  5 00:43:05.235: %LINEPROTO-5-UPDOWN: Line protocol on Interface NVI0, chan
*Jun  5 00:43:05.643: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state t
*Jun  5 00:43:05.847: %LINK-3-UPDOWN: Interface FastEthernet1/0, changed state t
*Jun  5 00:43:06.003: %LINK-3-UPDOWN: Interface FastEthernet1/1, changed state t
*Jun  5 00:43:06.643: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
*Jun  5 00:43:06.847: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
*Jun  5 00:43:07.027: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
R-16-l(config-if)#interface f0/0
R-16-l(config-if)# ip nat inside
R-16-l(config-if)#interface f1/0
R-16-l(config-if)# ip nat inside
R-16-l(config-if)#
R-16-l(config-if)#access-list 1 permit 192.168.0.0 0.0.255.255
R-16-l(config)#ip nat inside source list 1 interface f1/1 overload
R-16-l(config)#
R-16-l(config)#end
R-16-l#
*Jun  5 00:43:11.795: %SYS-5-CONFIG_I: Configured from console by console
R-16-l#
*Jun  5 00:43:19.391: %DHCP-6-ADDRESS_ASSIGN: Interface FastEthernet1/1 assigned
DHCP address 192.168.43.130, mask 255.255.255.0, hostname R-16-l
```

Рисунок 3.3 - Налаштування адрес та VLAN

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				34
Змн.	Арк.	№ докум.	Підпис	Дата		

```

SW-clients1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW-clients1(config)#
SW-clients1(config)#interface vlan1
SW-clients1(config-if)# ip address 192.168.10.2 255.255.255.0
SW-clients1(config-if)# no shutdown
SW-clients1(config-if)#
SW-clients1(config-if)#line vty 0 4
SW-clients1(config-line)# login
% Login disabled on line 2, until 'password' is set
% Login disabled on line 3, until 'password' is set
% Login disabled on line 4, until 'password' is set
% Login disabled on line 5, until 'password' is set
% Login disabled on line 6, until 'password' is set
SW-clients1(config-line)# password ci$col6
SW-clients1(config-line)# transport input ssh
SW-clients1(config-line)#
SW-clients1(config-line)#end
*Jun  4 22:21:46.045: %LINK-3-UPDOWN: Interface Vlan1, changed state to up
*Jun  4 22:21:47.045: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to up
SW-clients1(config-line)#end
SW-clients1#
*Jun  4 22:21:48.814: %SYS-5-CONFIG_I: Configured from console by console
SW-clients1#

```

Рисунок 3.4 - Налаштування адрес та VLAN

```

Usr-1> ip dhcp
DDORA IP 192.168.10.11/24 GW 192.168.10.1

Usr-1>

```

```

(kali@kali)-[~]
$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b4:a1:05 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.10/24 brd 192.168.10.255 scope global dynamic noprefixroute eth0
        valid_lft 14023sec preferred_lft 14023sec
    inet6 fe80::67b2:9b7b:bc01:36db/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
(kali@kali)-[~]

```

```

student@serv16:/etc/netplan$ cat 00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  version: 2
  ethernet:
    enp0s3:
      dhcp4: no
      addresses:
        - 192.168.20.10/24
      gateway4: 192.168.20.1
      nameservers:
        addresses: [192.168.20.1, 8.8.8.8, 8.8.4.4]
student@serv16:/etc/netplan$

```

Рисунок 3.5 – 3.7 - Налаштування адресації на станціях та сервері

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				35
Змн.	Арк.	№ докум.	Підпис	Дата		

Встановлення та налаштування OpenLDAP:

```
student@serv16:~$ sudo apt install slapd ldap-utils -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libldap-2.5-0 libltdl7 libodbc2
Suggested packages:
  libsasl2-modules-gssapi-mit | libsasl2-modules-gssapi-heimdal odbc-postgresql tdsodbc
The following NEW packages will be installed:
  ldap-utils libltdl7 libodbc2 slapd
The following packages will be upgraded:
  libldap-2.5-0
1 upgraded, 4 newly installed, 0 to remove and 44 not upgraded.
Need to get 2,065 kB of archives.
After this operation, 6,470 kB of additional disk space will be used.
Ign:1 http://ua.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libldap-2.5-0 amd64 2.5.16+dfsg-
ubuntu0.22.04.1
Get:2 http://ua.archive.ubuntu.com/ubuntu jammy/main amd64 libltdl7 amd64 2.4.6-15build2 [39.6 kB]
Get:3 http://ua.archive.ubuntu.com/ubuntu jammy/main amd64 libodbc2 amd64 2.3.9-5 [159 kB]
7% [3 libodbc2 42.1 kB/159 kB 26%]
```

Рисунок 3.8 - Встановлення пакунків

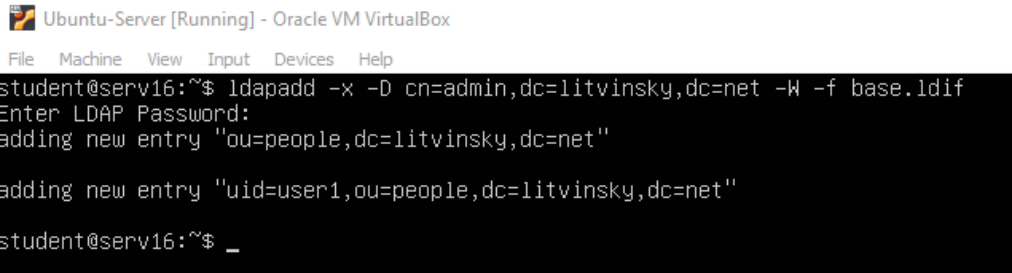
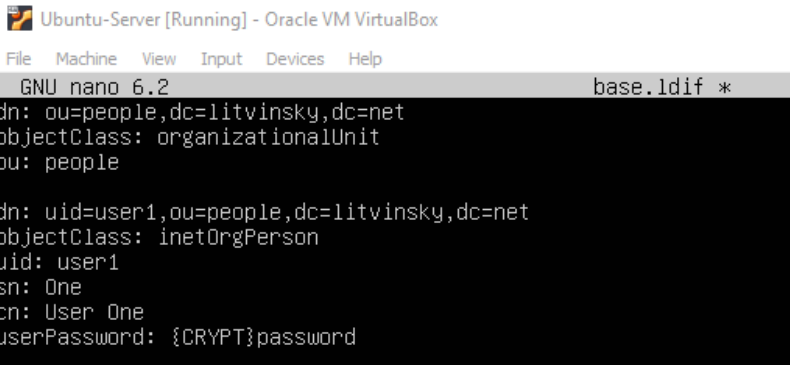
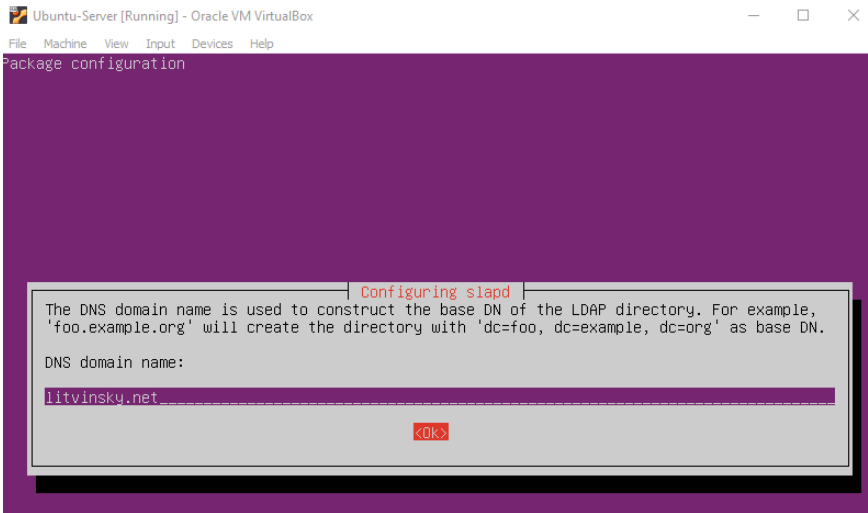
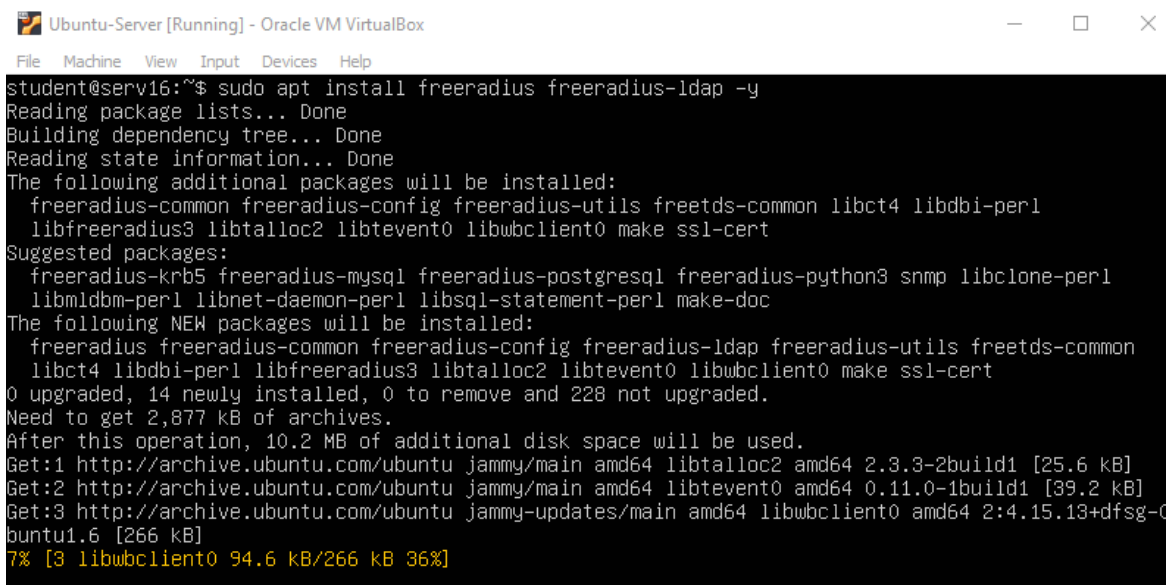


Рисунок 3.9 – 3.11 – Конфігурація LDAP, створення користувача

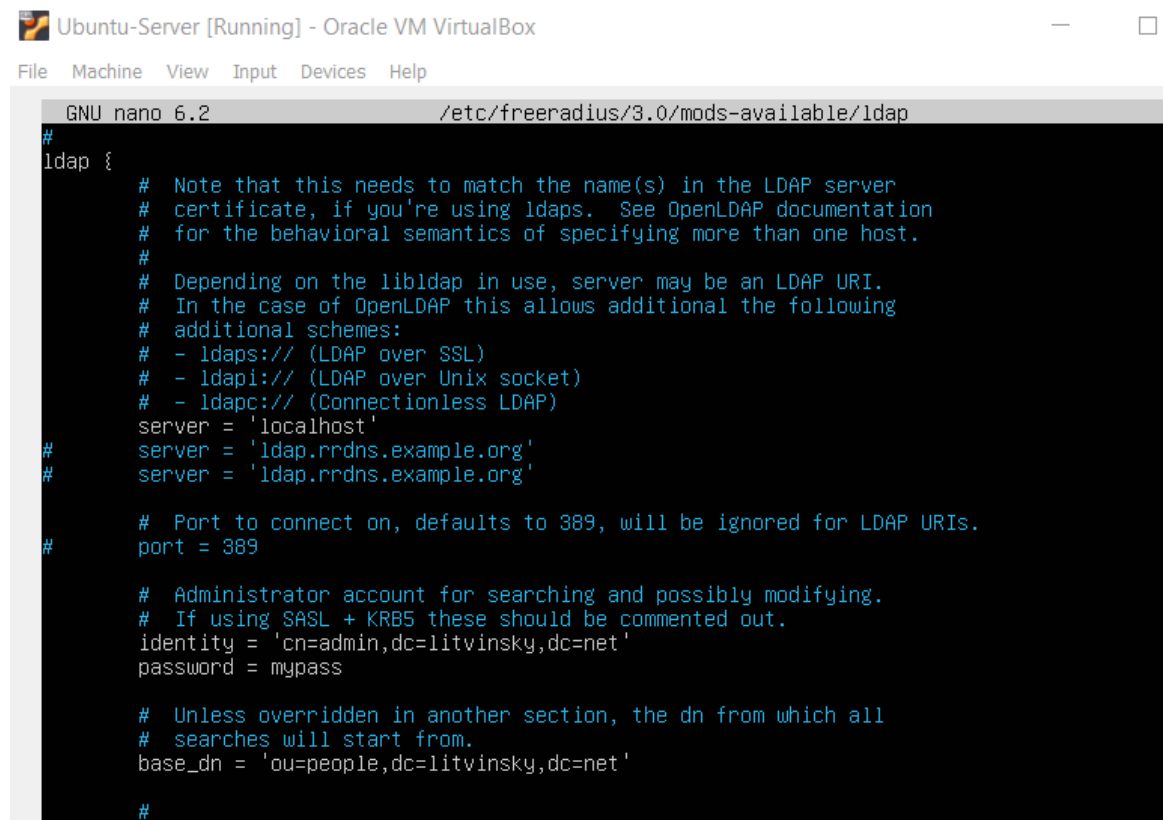
Встановлення FreeRADIUS:



```
Ubuntu-Server [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
student@serv16:~$ sudo apt install freeradius freeradius-ldap -y
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-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-ldap freeradius-utils freetds-common
  libct4 libdbi-perl libfreeradius3 libtalloc2 libtevent0 libwbclient0 make ssl-cert
0 upgraded, 14 newly installed, 0 to remove and 228 not upgraded.
Need to get 2,877 kB of archives.
After this operation, 10.2 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 libtalloc2 amd64 2.3.3-2build1 [25.6 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/main amd64 libtevent0 amd64 0.11.0-1build1 [39.2 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libwbclient0 amd64 2:4.15.13+dfsg-0
ubuntu1.6 [266 kB]
7% [3 libwbclient0 94.6 kB/266 kB 36%]
```

Рисунок 3.12 - Встановлення пакунків

Включення LDAP у FreeRADIUS:



```
Ubuntu-Server [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
GNU nano 6.2 /etc/freeradius/3.0/mods-available/ldap
#
ldap {
    # Note that this needs to match the name(s) in the LDAP server
    # certificate, if you're using ldaps. See OpenLDAP documentation
    # for the behavioral semantics of specifying more than one host.
    #
    # Depending on the libldap in use, server may be an LDAP URI.
    # In the case of OpenLDAP this allows additional the following
    # additional schemes:
    # - ldaps:// (LDAP over SSL)
    # - ldapi:// (LDAP over Unix socket)
    # - ldapc:// (Connectionless LDAP)
    server = 'localhost'
    #
    # server = 'ldap.rndns.example.org'
    #
    # server = 'ldap.rndns.example.org'
    #
    # Port to connect on, defaults to 389, will be ignored for LDAP URIs.
    #
    port = 389
    #
    # Administrator account for searching and possibly modifying.
    # If using SASL + KRB5 these should be commented out.
    identity = 'cn=admin,dc=litvinsky,dc=net'
    password = mypass
    #
    # Unless overridden in another section, the dn from which all
    # searches will start from.
    base_dn = 'ou=people,dc=litvinsky,dc=net'
    #
}
```

Рисунок 3.13 - Редагування конфігураційних файлів FreeRADIUS

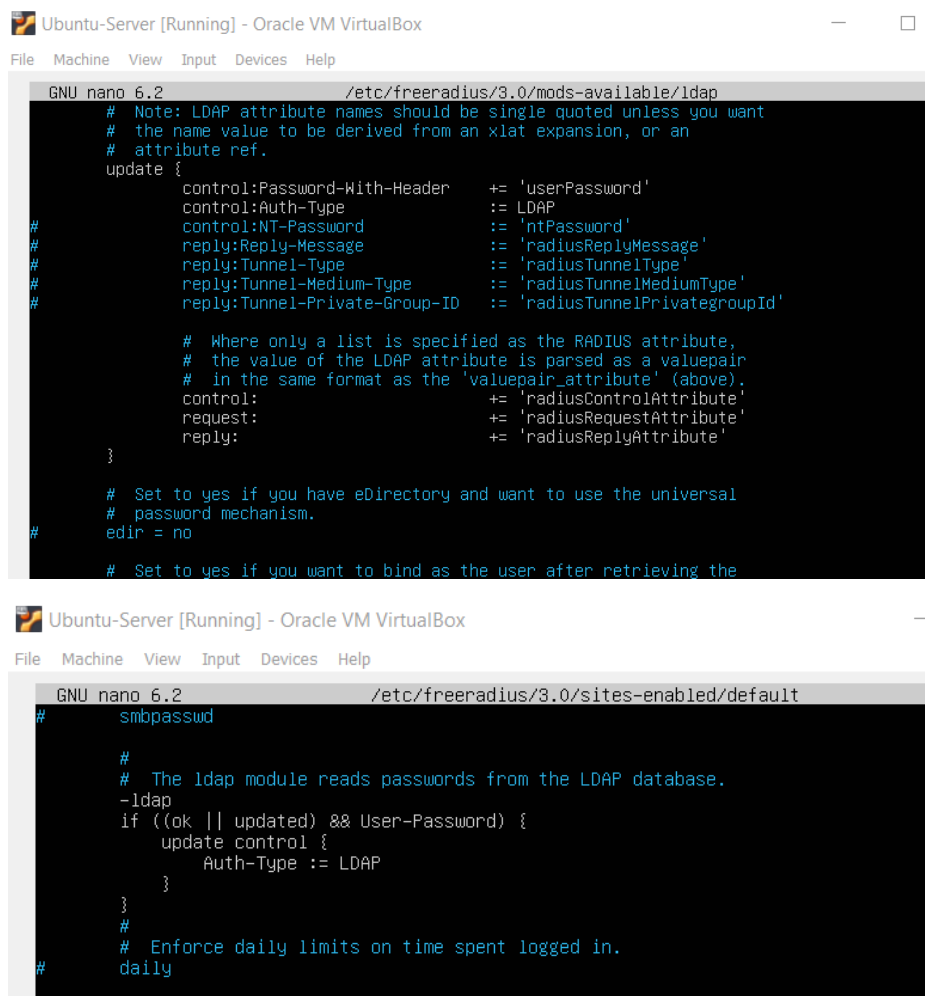


Рисунок 3.14 – 3.15 - Редагування конфігураційних файлів FreeRADIUS

Одноразові коди (One-Time Passwords, OTP) є ключовим елементом сучасної багатофакторної автентифікації. У цьому проєкті було реалізовано підтримку TOTP (Time-based One-Time Password) та HOTP (HMAC-based One-Time Password) для додаткового захисту при доступі до ресурсів відкритої мережі. Система використовує FreeRADIUS як центральний сервер автентифікації, OpenLDAP — як джерело облікових записів, а OTP реалізовані через зовнішні скрипти.

Для створення одноразових кодів розглядалися два основних підходи:

TOTP — часозалежний код, змінюється кожні 30 секунд.

HOTP — залежить від лічильника, що зростає при кожному вході.

В межах проєкту було реалізовано TOTP як основний метод через його універсальність і сумісність із мобільними застосунками, зокрема Google

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		38

Authenticator, FreeOTP, Aegis тощо. Також додатково передбачена можливість реалізації HOTP або OTP через електронну пошту/Telegram.

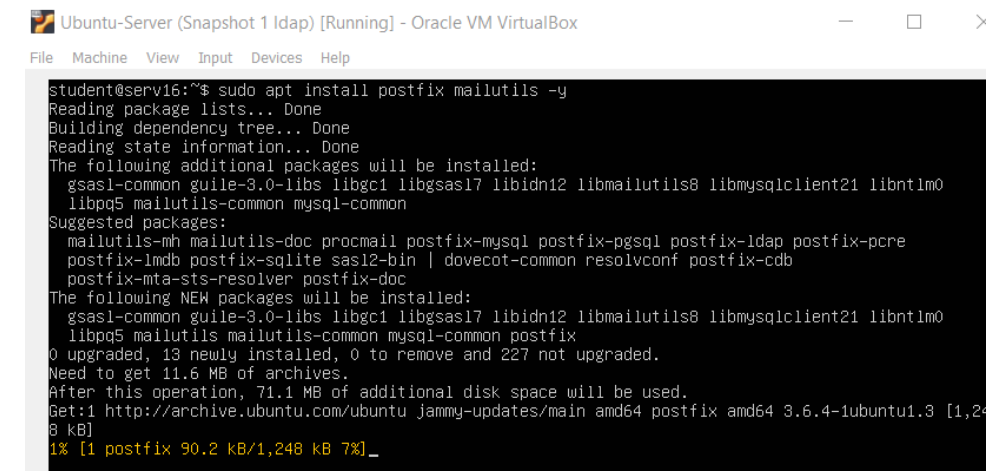


Рисунок 3.16 - Встановлення пакунків

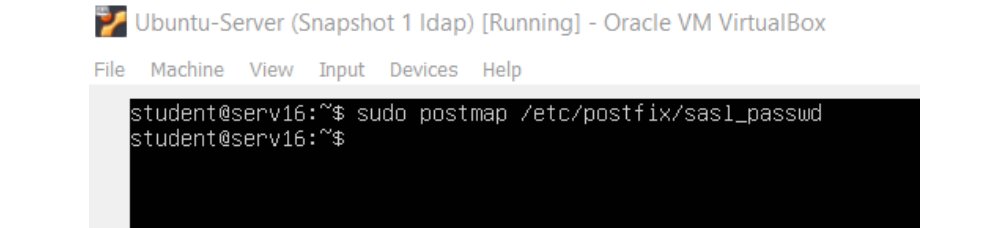
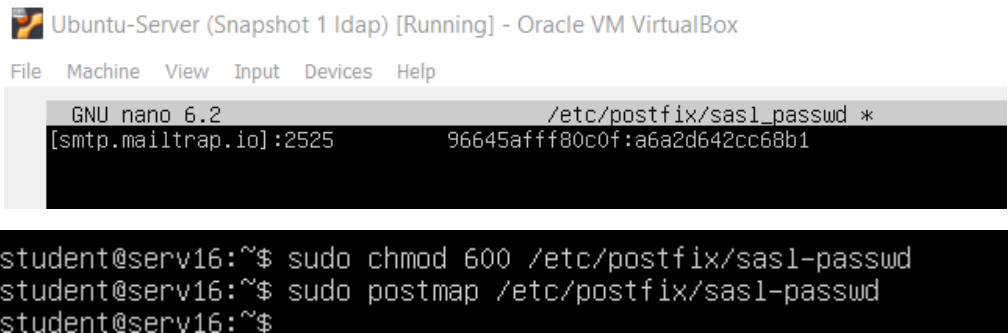
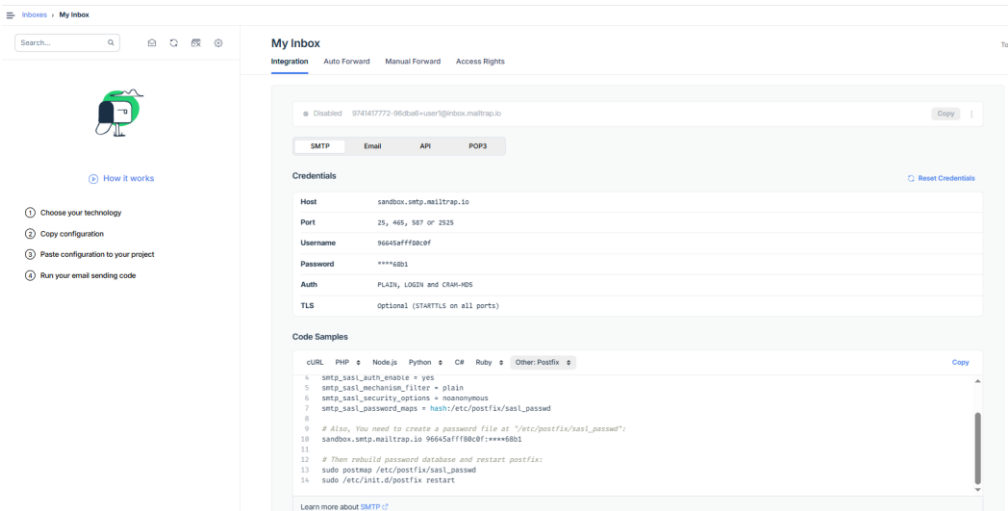


Рисунок 3.17 – 3.20 - Налаштування надсилання OTP на email

```
Ubuntu-Server (Snapshot 1 Idap) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
GNU nano 6.2 /etc/postfix/main.cf *
# fresh installs.
compatibility_level = 3.6

# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_tls_security_level=may

smtp_tls_CApath=/etc/ssl/certs
smtp_tls_security_level=may
smtp_tls_session_cache_database = btree:${data_directory}/smtpd_scache

smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_unauth_destination
myhostname = serv16
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = $myhostname, litvinsky.net, serv16, localhost.localdomain, localhost
relayhost = [smtp.mailtrap.io]:2525
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = ipv4
smtp_use_tls = yes
smtp_sasl_auth_enable = yes
smtp_sasl_security_options = noanonymous
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt
```

Рисунок 3.21 - Налаштування надсилання OTP на email

Скрипт генерації й надсилання OTP:

```
Ubuntu-Server (Snapshot 1 Idap) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
GNU nano 6.2 /usr/local/bin/send_otp.sh *
#!/bin/bash

USERNAME=$1
EMAIL=$(ldapsearch -x -LLL -b "ou=people,dc=litvinsky,dc=net" "(uid=$USERNAME)" mail | grep '^mail:|
OTP=$(shuf -i 100000-999999 -n 1)

echo "$OTP" > "/tmp/${USERNAME}_otp"

echo "Your one-time password to access litvinsky.net is: $OTP" | mail -s "OTP for login" "$EMAIL"

exit 0_
```

```
Ubuntu-Server (Snapshot 1 Idap) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
GNU nano 6.2 /usr/local/bin/check_otp.sh
#!/bin/bash

USERNAME="$1"
USER_INPUT="$2"

OTP_FILE="/tmp/${USERNAME}_otp"

if [[ ! -f "$OTP_FILE" ]]; then
    echo "OTP file not found"
    exit 2
fi

REAL_OTP=$(tr -d '[:space:]' < "$OTP_FILE" 2>/dev/null)

if [[ "$USER_INPUT" == "$REAL_OTP" ]]; then
    echo "OTP is correct"
    exit 0
else
    echo "OTP is incorrect"
    exit 1
fi
```

Рисунок 3.22 – 3.23 - Написання скриптів на надсилання та перевірку OTP

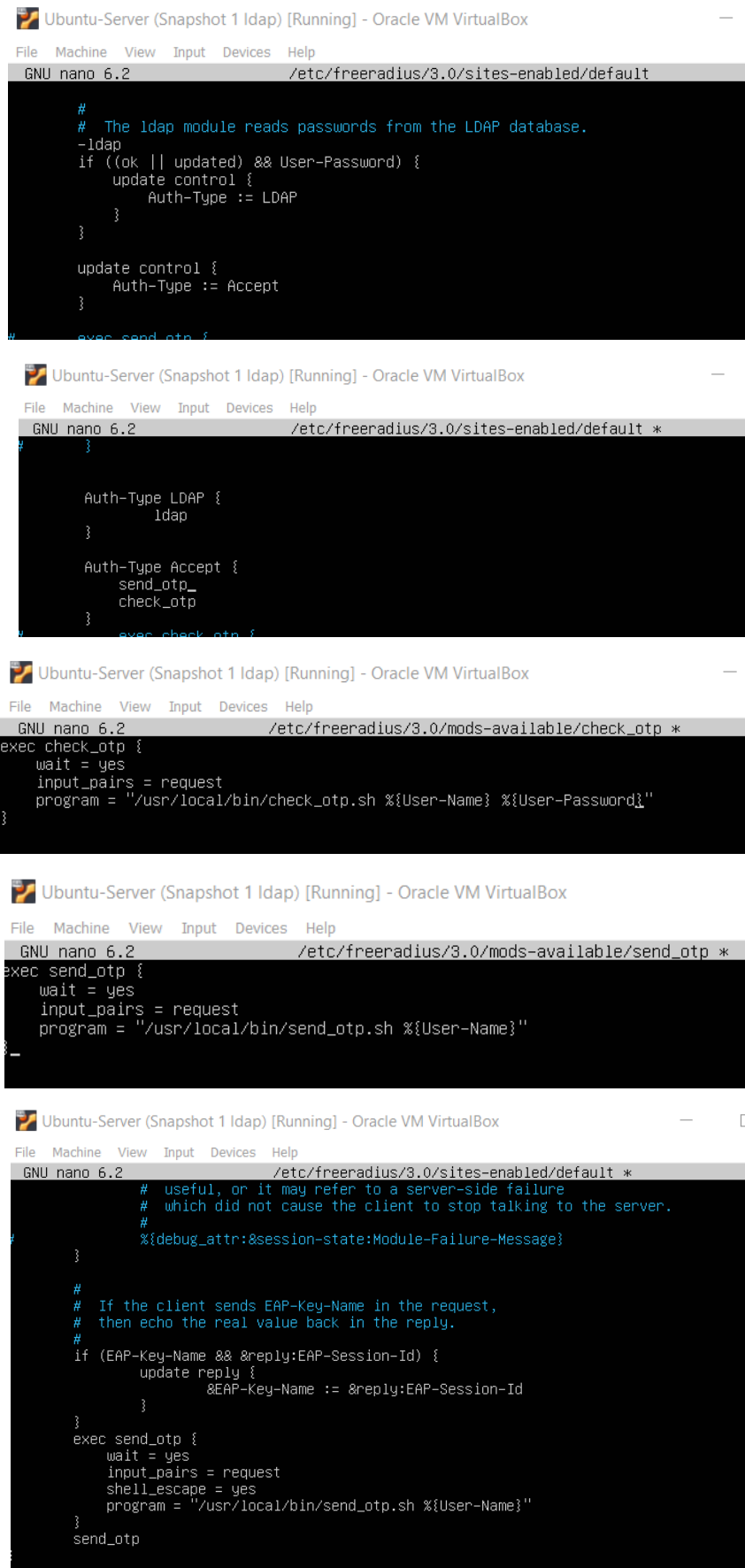


Рисунок 3.24 – 3.28 - Редагування файлів FreeRADIUS для надсилання та перевірки OTP

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				41
Змн.	Арк.	№ докум.	Підпис	Дата		

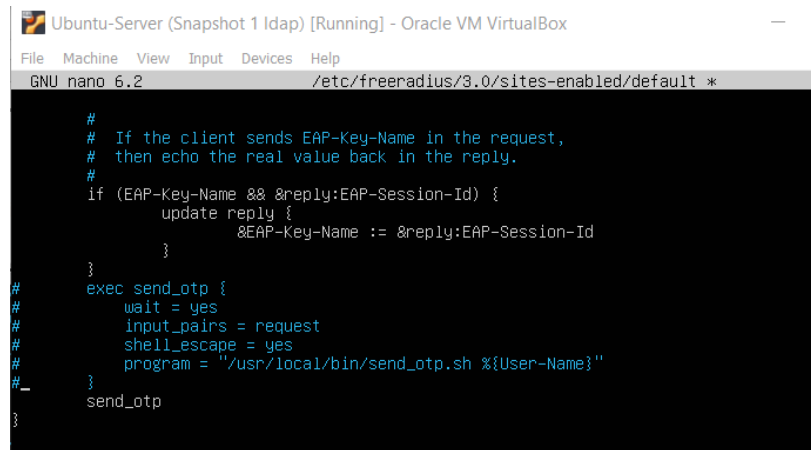


Рисунок 3.29 - Редагування файлів FreeRADIUS для надсилання та перевірки OTP

Для покращення зручності, гнучкості та універсальності процесу багатофакторної автентифікації було реалізовано вебінтерфейс, який дозволяє користувачам самостійно обирати бажаний метод отримання одноразового коду підтвердження (OTP): через електронну пошту або месенджер Telegram. Це розширення забезпечує кращу користувацьку взаємодію, адаптовану до індивідуальних потреб.

У базовій реалізації OTP надсилався виключно через електронну пошту, що не завжди є зручним або доступним для користувача. Telegram, як поширений і швидкий канал комунікації, дає можливість пришвидшити доставку коду та спростити його використання. З огляду на це, було вирішено додати підтримку двох альтернативних каналів доставки OTP та надати користувачеві можливість вибору.

Для реалізації обрано фреймворк Flask, який є легким і простим у налаштуванні вебсервером на базі Python. Для надсилання поштових повідомлень використано бібліотеку smtplib, а для Telegram — офіційний Telegram Bot API через HTTP-запити.

Інтерактивний вебінтерфейс реалізовано у вигляді Flask-застосунку, що доступний за внутрішньою IP-адресою сервера. Користувач відкриває сторінку автентифікації, вводить логін та пароль, після чого обирає бажаний метод отримання OTP. Після надсилання форми сервер:

1. генерує одноразовий код (6-значне випадкове число),
2. надсилає його через вибраний канал (Email або Telegram),
3. зберігає OTP тимчасово в сесії користувача,

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		42

4. очікує введення коду на наступній сторінці,
5. порівнює введений код із збереженим,
6. дозволяє або забороняє доступ.

```

student@serv16: ~
GNU nano 6.2 /usr/local/bin/send_telegram_otp.py *
import requests
import random

# Дани
bot_token = '7675911722:AAE5rC03VfYRKG1OH_RrQ5CdNHdR-mUBsJc'
chat_id = '6160974864'
otp = str(random.randint(100000, 999999))

# Збережи OTP у файл або базу (опційно)
with open('/tmp/otp.txt', 'w') as f:
    f.write(otp)

# Відправити через Telegram API
message = f"Ваш одноразовий код доступу: {otp}"
url = f"https://api.telegram.org/bot{bot_token}/sendMessage"
payload = {'chat_id': chat_id, 'text': message}

response = requests.post(url, data=payload)

if response.status_code == 200:
    print("OTP sent successfully.")
else:
    print("Error sending OTP:", response.text)

```

```

student@serv16: ~
GNU nano 6.2 /etc/freeradius/3.0/sites-enabled/default *
#
#       lifetime = 0
#       max_connections = 0
#
#
# Authorization. First preprocess (hints and huntgroups files),
# then realms, and finally look in the "users" file.
#
# Any changes made here should also be made to the "inner-tunnel"
# virtual server.
#
# The order of the realm modules will determine the order that
# we try to find a matching realm.
#
# Make 'sure' that 'preprocess' comes before any realm if you
# need to setup hints for the remote radius server
authorize {
    #
    # Take a User-Name, and perform some checks on it, for spaces and other
    # invalid characters. If the User-Name appears invalid, reject the
    # request.
    #
    # See policy.d/filter for the definition of the filter_username policy.
}

exec send_otp_telegram {
    wait = yes
    input_pairs = request
    program = "/usr/local/bin/send_telegram_otp.py"
}

```

Рисунок 3.30 – 3.31 - Додавання можливості отримати OTP через Телеграм

```

student@serv16:~$ sudo apt install python3-flask python3-ldap3 python3-requests postfix mailutils
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
mailutils is already the newest version (1:3.14-1).
postfix is already the newest version (3.6.4-ubuntu1.3).
python3-flask is already the newest version (2.0.1-2ubuntu1.1).
python3-requests is already the newest version (2.25.1+dfsg-2ubuntu0.1).
python3-requests set to manually installed.
The following packages were automatically installed and are no longer required:
  libexpat1-dev libjs-sphinxdoc libjs-underscore zlib1g-dev
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  python3-ldap3
0 upgraded, 1 newly installed, 0 to remove and 218 not upgraded.
Need to get 229 kB of archives.
After this operation, 2,296 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu/universe amd64 python3-ldap3 all 2.8.1-1 [229 kB]
54% [1 python3-ldap3 153 kB/229 kB 67%]

```

Рисунок 3.32 - Встановлення пакунків

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				43
Змн.	Арк.	№ докум.	Підпис	Дата		

```

student@serv16: ~
GNU nano 6.2 /usr/local/bin/check_otp.sh *
#!/bin/bash

USERNAME="$1"
USER_INPUT="$2"

OTP_FILE="/tmp/${USERNAME}_otp"
BOT_TOKEN="7675911722:AAE5rC03VfYRKG1OH_RrQ5CdNHdR-mUBsjc"
CHAT_ID="6160974864"

if [[ ! -f "$OTP_FILE" ]]; then
    echo "OTP file not found"
    exit 2
fi

REAL_OTP=$(tr -d '[:space:]' < "$OTP_FILE" 2>/dev/null)

if [[ "$USER_INPUT" == "$REAL_OTP" ]]; then
    echo "OTP is correct"

    # Повідомлення в Telegram про успішний вхід
    MESSAGE="✅ Успішна автентифікація користувача $USERNAME"
    curl -s -X POST "https://api.telegram.org/bot$BOT_TOKEN/sendMessage" \
        -d chat_id="$CHAT_ID" \
        -d text="$MESSAGE" > /dev/null

    # Очистити OTP після перевірки (одноразовість)
    rm -f "$OTP_FILE"

    exit 0
else
    echo "OTP is incorrect"

    # Повідомлення в Telegram про невдалу спробу
    MESSAGE="❌ НЕВІРНИЙ OTP-код для користувача $USERNAME"
    curl -s -X POST "https://api.telegram.org/bot$BOT_TOKEN/sendMessage" \
        -d chat_id="$CHAT_ID" \
        -d text="$MESSAGE" > /dev/null

    exit 1
fi

```

```

student@serv16: /opt/mfa_web
GNU nano 6.2 /opt/mfa_web/app.py *
from flask import Flask, render_template, request, redirect, session
from otp.email_sender import send_email
from otp.telegram_sender import send_telegram
from otp.otp_manager import generate_otp

app = Flask(__name__)
app.secret_key = 'lllllll'

@app.route('/')
def login():
    return render_template('login.html')

@app.route('/send_otp', methods=['POST'])
def send_otp():
    username = request.form['username']
    method = request.form['method']
    otp = generate_otp(username)
    session['username'] = username
    session['otp'] = otp
    if method == 'email':
        send_email(username, otp)
    else:
        send_telegram(username, otp)
    return render_template('otp.html')

@app.route('/verify_otp', methods=['POST'])
def verify_otp():
    otp_input = request.form['otp']
    if otp_input == session.get('otp'):
        return "✅ Доступ дозволено"
    return "❌ Невірний код", 403

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)

```

Рисунок 3.33 – 3.34 - Налаштування конф. Файлів (налаштування OTP Телеграм та інтерфейсу)

```

student@serv16: /opt/mfa_web

GNU nano 6.2 /opt/mfa_web/templates/login.html *
<h2>Вхід</h2>
<form action="/send_otp" method="post">
  <input type="text" name="username" placeholder="Логін" required><br>
  <input type="password" name="password" placeholder="Пароль" required><br>
  <label><input type="radio" name="method" value="email" checked> Отримати OTP на Email</label><br>
  <label><input type="radio" name="method" value="telegram"> Отримати OTP в Telegram</label><br>
  <button type="submit">Продовжити</button>
</form>

student@serv16: /opt/mfa_web

GNU nano 6.2 /opt/mfa_web/templates/otp.html *
<h2>Підтвердження OTP</h2>
<form action="/verify_otp" method="post">
  <input type="text" name="otp" placeholder="Введіть OTP" required><br>
  <button type="submit">Перевірити</button>
</form>

student@serv16: /opt/mfa_web

GNU nano 6.2 /opt/mfa_web/otp/email_sender.py *
import smtplib
from email.message import EmailMessage

def send_email(username, otp):
    recipient = f"{username}@litvinsky.net"
    msg = EmailMessage()
    msg.set_content(f"Вам OTP-код: {otp}")
    msg['Subject'] = "Підтвердження доступу"
    msg['From'] = "noreply@litvinsky.net"
    msg['To'] = recipient

    with smtplib.SMTP('localhost') as s:
        s.send_message(msg)

student@serv16: /opt/mfa_web

GNU nano 6.2 /opt/mfa_web/otp/telegram_sender.py *
import requests

TOKEN = "7675911722:AAE5rC03VfYRKGI0H_RrQ5CdNHdR-mUBsjc"
CHAT_ID_MAP = {
    'user1': '6160974864'
}

def send_telegram(username, otp):
    chat_id = CHAT_ID_MAP.get(username)
    if chat_id:
        url = f"https://api.telegram.org/bot{TOKEN}/sendMessage"
        data = {"chat_id": chat_id, "text": f"Вам OTP: {otp}"}
        requests.post(url, data=data)

student@serv16: /opt/mfa_web

GNU nano 6.2 /opt/mfa_web/otp/otp_manager.py *
import random

def generate_otp(username):
    return str(random.randint(100000, 999999))

```

Рисунок 3.35 – 3.39 - Налаштування конф. Файлів (налаштування OTP Телеграм та інтерфейсу)

3.2. Тестування та оцінка ефективності розробленої моделі кампусної мережі

В попередньому розділі було виконано розгортання проєкту мережевого середовища, в цьому розділі буде проведено перевірку працездатності всіх його компонентів.

```
student@serv16:~$ ping 192.168.10.10
PING 192.168.10.10 (192.168.10.10) 56(84) bytes of data.
64 bytes from 192.168.10.10: icmp_seq=3 ttl=63 time=20.6 ms
64 bytes from 192.168.10.10: icmp_seq=4 ttl=63 time=15.5 ms
64 bytes from 192.168.10.10: icmp_seq=5 ttl=63 time=21.8 ms
^C
--- 192.168.10.10 ping statistics ---
5 packets transmitted, 3 received, 40% packet loss, time 4032ms
rtt min/avg/max/mdev = 15.539/19.324/21.796/2.717 ms
student@serv16:~$ ping 192.168.20.1
PING 192.168.20.1 (192.168.20.1) 56(84) bytes of data.
64 bytes from 192.168.20.1: icmp_seq=6 ttl=255 time=60.8 ms
64 bytes from 192.168.20.1: icmp_seq=7 ttl=255 time=6.98 ms
64 bytes from 192.168.20.1: icmp_seq=8 ttl=255 time=15.1 ms
^C
--- 192.168.20.1 ping statistics ---
8 packets transmitted, 3 received, 62.5% packet loss, time 7116ms
rtt min/avg/max/mdev = 6.982/27.632/60.830/23.706 ms
student@serv16:~$ ping 192.168.10.11
PING 192.168.10.11 (192.168.10.11) 56(84) bytes of data.
64 bytes from 192.168.10.11: icmp_seq=12 ttl=63 time=22.1 ms
64 bytes from 192.168.10.11: icmp_seq=13 ttl=63 time=28.2 ms
64 bytes from 192.168.10.11: icmp_seq=14 ttl=63 time=24.8 ms
^C
--- 192.168.10.11 ping statistics ---
14 packets transmitted, 13 received, 7.1% packet loss, time 1000ms
rtt min/avg/max/mdev = 22.100/25.000/28.200/3.050 ms
```

Рисунок 3.40 - Перевірка зв'язку

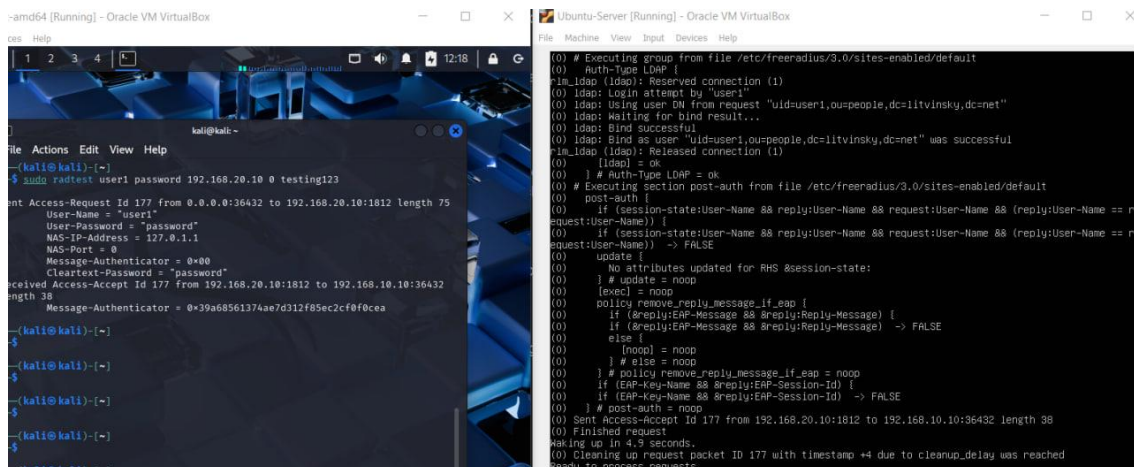


Рисунок 3.41 - Перевірка аутентифікації (без OTP)

```
student@serv16:~$ sudo systemctl restart postfix
student@serv16:~$ echo "Your OTP is 123456" | mail -s "OTP test" [redacted]@gmail.com
student@serv16:~$
```

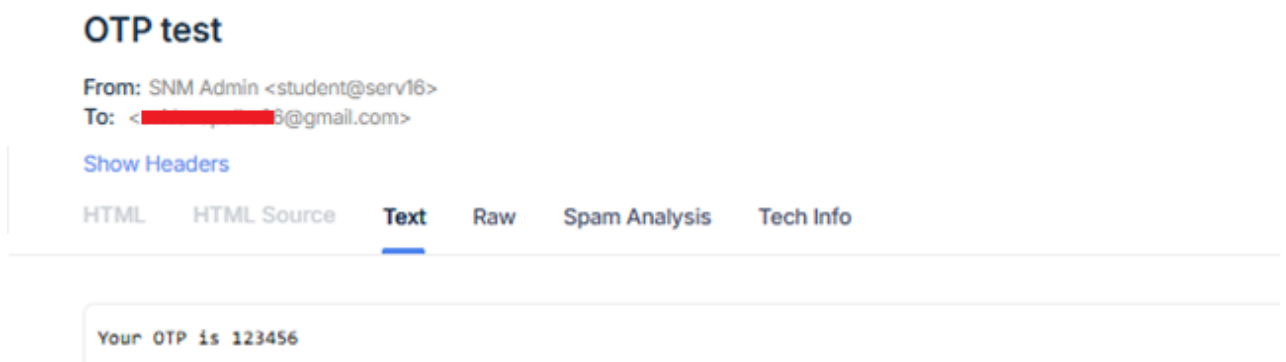


Рисунок 3.42 - Перевірка отримання OTP (email)

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				46
Змн.	Арк.	№ докум.	Підпис	Дата		

```
student@serv16:~$ cat /tmp/user1_otp
356155
student@serv16:~$ /usr/local/bin/check_otp.sh user1 356155
OTP is correct
student@serv16:~$ /usr/local/bin/check_otp.sh user1 186574
OTP is incorrect
student@serv16:~$
```

Рисунок 3.43 - Перевірка правильності OTP

```
student@serv16:~$ sudo chmod +x /usr/local/bin/send_otp.sh
student@serv16:~$ /usr/local/bin/send_otp.sh [REDACTED] 56@gmail.com user1
student@serv16:~$ _
```

OTP for user1

From: SNM Admin <student@serv16>
To: <[REDACTED]56@gmail.com>

Show Headers

HTML HTML Source Text Raw Spam Analysis Tech Info

Your one-time password to access litvinsky.net is: 356155

Рисунок 3.44 – 3.45 - Перевірка генерації та отримання OTP (email)

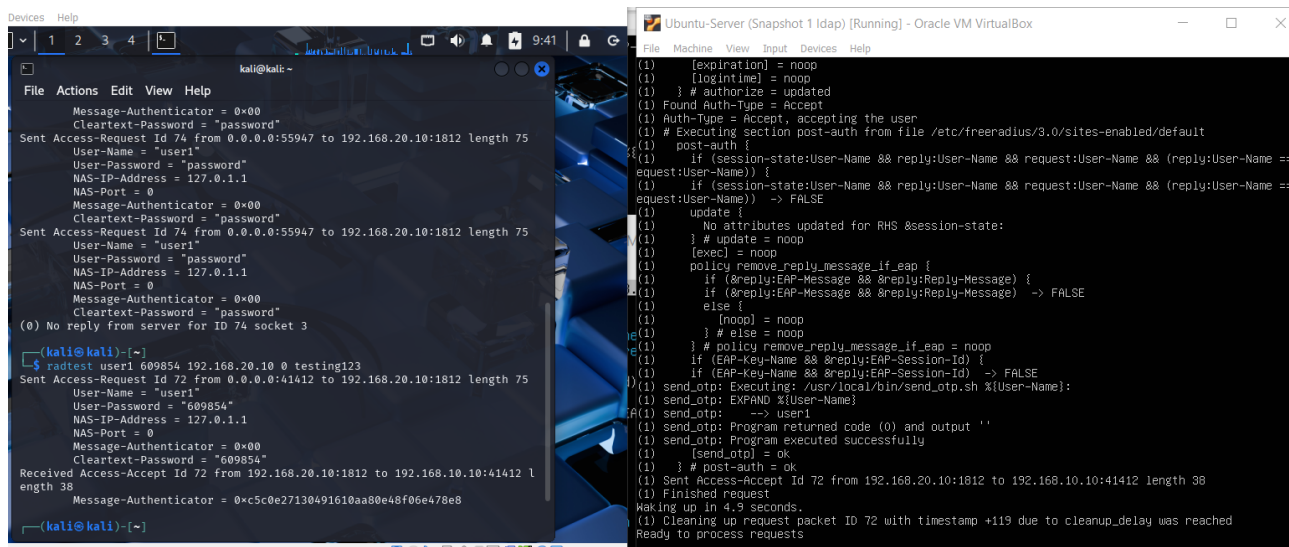
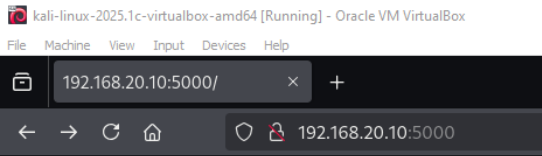


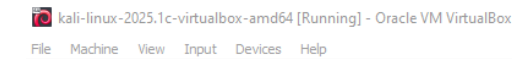
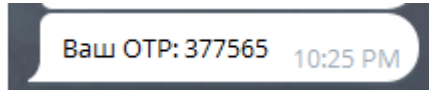
Рисунок 3.46 - Перевірка аутентифікації (введення отриманого OTP)

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				47
Змн.	Арк.	№ докум.	Підпис	Дата		

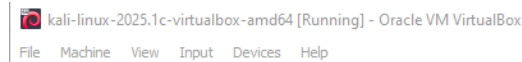


Вхід

user1
.....
☐ Отримати OTP на Email
☒ Отримати OTP в Telegram
Продовжити



✓ Доступ дозволено



✗ Невірний код

```
smtplib.SMTPRecipientsRefused: {'': (501, b'5.1.3 Bad recipient address syntax')}\n192.168.10.10 - - [19/Jun/2025 19:23:37] "POST /send_otp HTTP/1.1" 500 -\n192.168.10.10 - - [19/Jun/2025 19:23:50] "POST /send_otp HTTP/1.1" 200 -\n192.168.10.10 - - [19/Jun/2025 19:24:00] "POST /verify_otp HTTP/1.1" 200 -\n192.168.10.10 - - [19/Jun/2025 19:25:24] "POST /send_otp HTTP/1.1" 200 -\n192.168.10.10 - - [19/Jun/2025 19:26:00] "POST /verify_otp HTTP/1.1" 200 -\n192.168.10.10 - - [19/Jun/2025 19:26:21] "POST /verify_otp HTTP/1.1" 403 -\n192.168.10.10 - - [19/Jun/2025 19:27:01] "POST /send_otp HTTP/1.1" 200 -
```

Рисунок 3.47 – 3.51 - Отримання OTP через Телеграм, введення правильного та неправильного коду

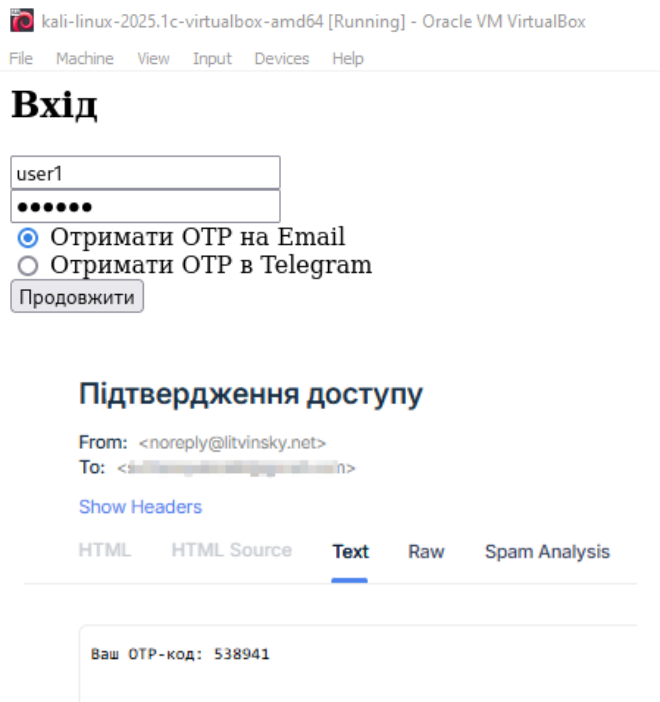
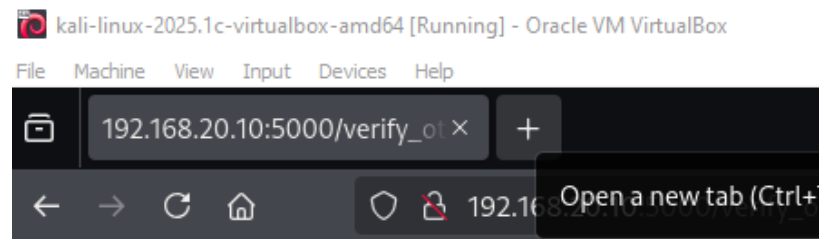
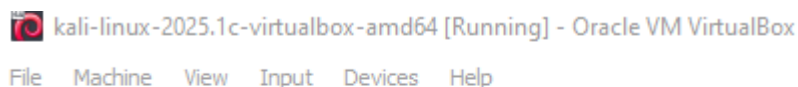


Рисунок 3.52 - Отримання OTP через email

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				48
Змн.	Арк.	№ докум.	Підпис	Дата		



✓ Доступ дозволено



✗ Невірний код

```

Use a production WSGI server instead.
* Debug mode: off
* Running on all addresses.
  WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://192.168.20.10:5000/ (Press CTRL+C to quit)
192.168.10.10 - - [19/Jun/2025 19:30:34] "POST /send_otp HTTP/1.1" 200 -
192.168.10.10 - - [19/Jun/2025 19:31:30] "POST /verify_otp HTTP/1.1" 200 -
192.168.10.10 - - [19/Jun/2025 19:32:41] "POST /send_otp HTTP/1.1" 200 -
192.168.10.10 - - [19/Jun/2025 19:32:45] "POST /verify_otp HTTP/1.1" 403 -

```

Рисунок 3.53 – 3.55 - Отримання OTP через email, введення правильного та неправильного коду

3.3. Аналіз роботи підсистеми автентифікації

У ході реалізації проєкту було створено працездатну підсистему багатфакторної автентифікації, що базується на зв'язці FreeRADIUS, OpenLDAP та власноруч реалізованих механізмах доставки одноразових паролів (OTP) через електронну пошту або месенджер Telegram. Реалізація включає базу користувачів у LDAP, перевірку паролів першого рівня автентифікації, генерацію OTP з обмеженим терміном дії, можливість вибору каналу доставки коду, а також вебінтерфейс для взаємодії користувача з системою.

Уся логіка реалізована у вигляді незалежних модулів, що забезпечують:

- генерацію одноразових кодів за алгоритмом TOTP,
- надсилання повідомлень засобами SMTP або Telegram Bot API,
- інтеграцію з RADIUS-сервером через зовнішні скрипти (exes),
- збереження OTP у тимчасових структурах (сесії або проміжних файлах),
- перевірку введених кодів перед остаточним дозволом доступу.

Система протестована в умовах лабораторної мережі та демонструє стабільну роботу при правильній конфігурації компонентів.

Разом з тим, під час реалізації було виявлено низку обмежень, які відкривають простір для подальшого вдосконалення функціоналу. Зокрема:

- Відсутність системи ролей (RBAC). На даному етапі всі користувачі мають однакові права доступу. Реалізація системи ролей дозволить встановлювати різні рівні доступу для студентів, викладачів, адміністративного персоналу тощо. Це забезпечить більш гнучке управління політиками безпеки та дозволить автоматизувати налаштування доступу до ресурсів.

- Обмежена підтримка реєстрації нових користувачів. Поточна система орієнтована на попередньо доданих користувачів у каталозі LDAP. Проте доцільним є впровадження механізму реєстрації нових облікових записів через вебінтерфейс з перевіркою введеної інформації (email або Telegram) та подальшим автоматичним створенням користувача в LDAP.

- Обмежена безпека вебінтерфейсу. Застосунок реалізований через Flask із базовим захистом сесії, однак відсутня підтримка HTTPS, журналювання дій, а також захист від повторного використання OTP, підбору паролів та зловмисного доступу. Надалі слід реалізувати шифрування трафіку, обмеження по IP/таймауту, CAPTCHA, обмеження кількості спроб автентифікації.

- Обмежена кількість способів автентифікації. Поточна реалізація підтримує лише OTP через Telegram або email, однак потенційно можлива інтеграція з Google Authenticator, Authy, Duo, а також реалізація механізмів QR-автентифікації, WebAuthn або push-нотифікацій.

- Відсутність централізованої панелі адміністрування та логування. Поточна система не передбачає окремого інтерфейсу для адміністратора, а також не виконує аудит активності користувачів. У подальшому варто реалізувати централізовану панель, що дозволить керувати обліковими записами, скидання OTP, моніторинг спроб входу, логування IP-адрес, часу, успішності авторизацій тощо.

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				50
Змн.	Арк.	№ докум.	Підпис	Дата		

У перспективі дана система може стати основою для побудови більш масштабною та надійною платформи автентифікації, інтегрованої з внутрішніми сервісами кампусу, VPN-доступом, Wi-Fi captive portal, тощо.

Висновки до третього розділу

В ході написання третього розділу було проведено повну технічну інтеграцію компонентів — налаштовано середовище на базі Ubuntu Server, встановлено та сконфігуровано сервери OpenLDAP і FreeRADIUS, а також створено допоміжну інфраструктуру для надсилання одноразових паролів (OTP) через поштову систему Postfix та месенджер Telegram.

Особлива увага приділялась безпеці та адаптивності системи: реалізовано подвійний етап перевірки автентичності користувача, при якому перший фактор підтверджується за допомогою облікових даних у LDAP, а другий — через OTP, який надсилається за вибором користувача. Вебінтерфейс, реалізований у Flask, забезпечує інтерактивну взаємодію з користувачем та дозволяє вибрати канал отримання коду — електронну пошту або Telegram, що робить систему зручною для використання в умовах різного технічного рівня аудиторії.

Суттєвим досягненням є впровадження зовнішніх скриптів OTP (send_otp, check_otp), які забезпечують гнучкість і масштабованість рішення, дозволяючи легко додавати нові методи доставки кодів або змінювати логіку автентифікації. Система була протестована як у командному рядку (через radtest), так і через вебінтерфейс, що підтвердило її стабільність, правильну інтеграцію компонентів та відповідність вимогам безпеки.

Окремо варто зазначити перспективи подальшого розвитку системи. Серед рекомендованих покращень — впровадження ролей користувачів (RBAC), реалізація механізму реєстрації нових користувачів через вебінтерфейс, розширення переліку методів автентифікації (наприклад, підтримка Google Authenticator, біометрії або апаратних токенів), підвищення рівня захисту переданих даних, а також розробка централізованої адміністративної панелі з функціями логування та моніторингу.

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				51
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У процесі виконання дипломної роботи було реалізовано повноцінний комплекс заходів, спрямованих на підвищення рівня інформаційної безпеки в комп'ютерних мережах закладів вищої освіти. Робота поєднує теоретичні дослідження, інженерне проєктування та практичну реалізацію ключових елементів системи автентифікації на основі сучасних стандартів і технологій.

У першому розділі було здійснено детальний огляд існуючих загроз безпеці мережевих ресурсів та обґрунтовано доцільність впровадження багатофакторної автентифікації (MFA) як одного з найефективніших засобів захисту. Наведено класифікацію факторів автентифікації (знання, володіння, біометрія), проаналізовано популярні алгоритми OTP (одноразових паролів), такі як TOTP (Time-based One-Time Password) та HOTP (HMAC-based One-Time Password), а також обрано технологічні засоби, придатні для впровадження у навчальному середовищі з обмеженими ресурсами.

Другий розділ було присвячено загальній характеристиці об'єкта дослідження, постановці практичних завдань, вибору апаратного та програмного забезпечення, аналізу можливостей емуляторів та тестового середовища. Було проведено аналіз вимог до системи, вибрано відповідну серверну платформу (Ubuntu Server), а також визначено оптимальну архітектуру взаємодії між компонентами — FreeRADIUS, OpenLDAP, Flask, Telegram Bot API та Postfix SMTP. Додатково проаналізовано альтернативні рішення, доступні на ринку, зокрема Duo Security, Keycloak, FreeIPA, Google Authenticator, з висвітленням причин відмови від них у межах даного проєкту.

У третьому розділі реалізовано ключову функціональність підсистеми автентифікації. Було створено інтегроване середовище, в якому користувач проходить первинну автентифікацію через LDAP, а далі — другий етап перевірки через OTP, який надсилається на електронну пошту або через Telegram. Запроваджено зручний вебінтерфейс для взаємодії з користувачем,

реалізовано генерацію, доставку та перевірку OTP-кодів через власні Python-скрипти, інтегровані з FreeRADIUS через модуль ehex. Проведено всебічне

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				52
Змн.	Арк.	№ докум.	Підпис	Дата		

тестування системи, виявлено та усунуто ключові помилки, підтверджено стабільність роботи підсистеми в тестовому середовищі. Аналіз результатів показав ефективність використаного підходу як з погляду функціональності, так і з огляду на простоту впровадження в освітні організації.

Загалом, запропоноване рішення відповідає сучасним вимогам до інформаційної безпеки, є достатньо гнучким, масштабованим і придатним для подальшого розвитку. До основних переваг реалізованої підсистеми належать:

- підтримка кількох каналів доставки OTP (email, Telegram);
- інтеграція з LDAP та RADIUS, що дозволяє легко вбудувати MFA в існуючу мережеву інфраструктуру;
- використання відкритих технологій без залежності від комерційних платформ;
- зручний інтерфейс користувача, реалізований через вебдодаток;
- можливість розширення архітектури (додавання нових методів автентифікації, реєстрація користувачів, рольова модель доступу тощо).

Перспективи розвитку системи включають впровадження повноцінної панелі адміністратора, реалізацію централізованого журналювання та аудиту подій, підтримку сучасних стандартів автентифікації (WebAuthn, FIDO2), автоматизацію управління користувачами, інтеграцію з електронними журналами та системами управління навчанням (LMS).

Результати дипломної роботи можуть бути використані як основа для побудови промислової системи безпечної автентифікації у відкритих або корпоративних мережах, з урахуванням вимог GDPR, ISO/IEC 27001 та інших стандартів у сфері інформаційної безпеки.

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				53
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Prey Project. Cybersecurity Threats IT Professionals in Education Face.
URL: <https://preyproject.com/blog/cyber-security-threats-it-professionals-in-education-face>
2. Kaspersky. The Risks of Using Public Wi-Fi.
<https://usa.kaspersky.com/resource-center/preemptive-safety/public-wifi-risks>
3. Wikipedia. Social Engineering (Security).
[https://en.wikipedia.org/wiki/Social_engineering_\(security\)](https://en.wikipedia.org/wiki/Social_engineering_(security))
4. Microsoft Learn. Authentication fundamentals. <https://learn.microsoft.com/en-us/azure/active-directory/fundamentals/authentication-overview>
5. OWASP Foundation. Authentication Cheat Sheet. https://cheatsheet-series.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html
6. Cisco. AAA Access Control. <https://www.cisco.com/c/en/us/about/security-center/aaa.html>
7. IBM. What is SIEM? <https://www.ibm.com/topics/siem>
8. Lucas Augusto Meyer et al. "How effective is multifactor authentication at deterring cyberattacks?" arXiv preprint arXiv:2305.00945 (2023).
<https://arxiv.org/abs/2305.00945>
9. Кампусна мережа бізнес центру. <https://www.knuba.edu.ua/wp-content/uploads/2022/10/Кампусна-мережа-бізнес-центру.pdf>
10. Чурілов А.Ю. Інформаційна безпека комп'ютерних систем: підручник. — Харків: ХНУРЕ, 2020. — 328 с.
11. Mayer A., Smith J. Multi-Factor Authentication for Secure Access: A Practical Guide. — Springer, 2021. — 178 p.
12. Романов В.О. Основи побудови захищених інформаційних систем. — Львів: ЛНУ, 2021. — 302 с.
13. Kizza J.M. A Guide to Computer Network Security. — Springer, 2020. — 398 p.

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				54
Змн.	Арк.	№ докум.	Підпис	Дата		

14. Гнатенко А.В. Протоколи авторизації та безпечного обміну даними. — Дніпро: ДНУ, 2022. — 168 с.
15. Gerald Carter. LDAP System Administration. — O'Reilly Media, 2003. — 312 p.
16. The FreeRADIUS Technical Guide — <https://wiki.freeradius.org/>
17. Jonathan Hassell. RADIUS: Securing Public Access to Private Resources. — O'Reilly Media, 2002. — 240 p.
18. Noah Gift, Kennedy Behrman. Python for DevOps: Learn Ruthlessly Effective Automation. — O'Reilly Media, 2019. — 416 p.
19. Miguel Grinberg. Flask Web Development. — 2nd ed. — O'Reilly Media, 2018. — 258 p.

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		55

ДОДАТКИ

		Літвінський Ю.О.			ККІКБ.КР.Б – 125 – 16 – ПЗ	Арк.
		Шелуха О.О.				56
Змн.	Арк.	№ докум.	Підпис	Дата		

Топологія

