БАКАЛАВРСЬКА РОБОТА

КРБ.ІСТ-02.00.00.000 ПЗ

Група ІСТ-22-1

Владислав ВАКІВ

2026

Міністерство освіти і науки України
Івано-Франківський національний технічний університет нафти і газу
Інститут інформаційних технологій
Кафедра інформаційно-телекомунікаційних технологій та систем

Ваків Владислав Андрійович
(прізвище, ім'я, по батькові)

УДК 004.056:004.738.5
(індекс)

БАКАЛАВРСЬКА РОБОТА

Проектування системи керування доступом до мережі на основі протоколу 802.1X для IoT-інфраструктури
(назва роботи)

Інформаційні системи та технології
(назва освітньої програми)

126 − Інформаційні системи та технолгії
(шифр і назва спеціальності)

Робота містить результати власних досліджень, використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело:

Здобувач освітнього ступеня В.А. Ваків
(підпис, ініціали та прізвище здобувача)

Науковий керівник к.т.н., доцент Заміховська О. Л.
(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту

В. о. завідувача кафедри ІТТС О. Л. Заміховська
(посада) (підпис) (дата) (ініціали та прізвище)

Івано-Франківськ – 2026

Івано-Франківський національний технічний університет нафти і газу

(повне найменування закладу вищої освіти)

Інститут-факультет *Інформаційних технологій*

Кафедра *Інформаційно-телекомунікаційних технологій та систем*

Освітній рівень *бакалавр*

Спеціальність *126−Інформаційні системи та технології*

(шифр і назва)

ЗАТВЕРДЖУЮ

В. о. зав. кафедри ІТТС к.т.н.,доц.

О.Л.Заміховська

«____» 2026 року

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТОВІ

Ваківу Владиславу Андрійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Проектування системи керування доступом до мережі на основі протоколу 802.1X для IoT-інфраструктури*

керівник роботи *Заміховська Олена Леонідівна, к.т.н, доц. каф ІТТС*

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від "*7*" *квітня* 2026 року № *163/7*

2. Строк подання студентом роботи

3. Вихідні дані до роботи *концепція мережевого контролю доступу (NAC) для IoT-інфраструктури, протоколи ідентифікації IEEE 802.1X та EAP-TLS*

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

Теоретичні основи керування доступом у мережах IoT; Аналіз механізмів стандарту IEEE 802.1x та особливості його застосування в IoT-інфраструктурі; Проектування системи керування доступом до мережі; Практична реалізація та тестування розробленої системи

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

КРБ.ІСТ - 02.00.00.000 С1 Система керування доступом для IoT-інфраструктури. Структурно-логічна схема

КРБ.ІСТ - 02.00.00.000 С1 Модель розподілу довіри в інфраструктурі PKI. Схема структурна

КРБ.ІСТ - 02.00.00.000 С2 Процес автентифікації IoT-пристрою за алгоритмом EAP-TLS. Діаграма послідовності

КРБ.ІСТ - 02.00.00.000 С2 Динамічне призначення VLAN на рівні автентифікатора. Діаграма послідовності

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Заміховська О. Л., доцент		
2	Заміховська О. Л., доцент		
3	Заміховська О. Л., доцент		
4	Заміховська О. Л., доцент		

7. Дата видачі завдання *10 квітня 2026 р.*

КАЛЕНДАРНИЙ ПЛАН

№ з/п	**Назва етапів бакалаврської роботи**	**Термін виконання етапів роботи**	**Примітка**
1	*Аналіз теоретичних основ керування доступом та вразливостей в IoT*	*10.04.2026-15.04.2026*	*Виконано*
2	*Аналіз механізмів стандарту IEEE 802.1x та особливості його застосування в IoT-інфраструктурі*	*20.04.2026-27.04.2026*	*Виконано*
3	*Проектування системи керування доступом до мережі*	*30.04.2026-10.05.2026*	*Виконано*
4	*Практична реалізація та тестування розробленої системи*	*12.05.2026-20.05.2026*	*Виконано*
5	*Оформлення результатів роботи*	*21.05.2026-01.06.2026*	*Виконано*

Студент _____ Ваків В.А.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Заміховська О.Л.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Бакалаврська робота складається з 4 розділів, 25 рисунків, 2 таблиць та 1 додатка. Загальний обсяг БР 93 сторінки, в роботі використано 26 літературних джерел.

Тема бакалаврської роботи: Проектування системи керування доступом до мережі на основі протоколу 802.1X для IoT-інфраструктури.

Мета роботи: проектування системи керування доступом до мережі на основі стандарту IEEE 802.1X, оптимізованої для безінтерфейсних (headless) IoT-пристроїв, з використанням протоколу EAP-TLS та відокремленої інфраструктури відкритих ключів (PKI).

Об'єкт дослідження: процес забезпечення інформаційної безпеки та контролю доступу до мережі для автономних IoT-вузлів.

Предмет дослідження: методи та програмно-апаратні засоби реалізації мережевого контролю доступу на основі стандарту IEEE 802.1X, протоколу EAP-TLS та інструментів PKI.

Результати бакалаврської роботи: Проаналізовано вразливості IoT-мереж та обґрунтовано доцільність переходу до парадигми Zero Trust на базі стандарту IEEE 802.1X. Розроблено архітектуру безпеки з використанням EAP-TLS та відокремленого центру сертифікації (PKI) для надійної ідентифікації автономних пристроїв. Спроєктовано механізм динамічної мікросегментації мережі за RADIUS-атрибутами та налаштовано резервний режим MAC Authentication Bypass (MAB) для ізоляції застарілих вузлів. Емуляція системи в середовищі GNS3 на базі відкритого ПЗ (FreeRADIUS, MikroTik) підтвердила її стійкість до несанкціонованого фізичного проникнення та високу операційну ефективність.

Ключові слова: ІНТЕРНЕТ РЕЧЕЙ, КЕРУВАННЯ ДОСТУПОМ, IEEE 802.1X, EAP-TLS, ІНФРАСТРУКТУРА ВІДКРИТИХ КЛЮЧІВ, МІКРОСЕГМЕНТАЦІЯ, ZERO TRUST, FREERADIUS, MIKROTIK.

ABSTRACT

The bachelor's thesis consists of 4 chapters, 25 figures, 2 tables, and 1 appendix. The total volume of the thesis is 93 pages, 26 literature sources are used in the work.

Bachelor's thesis topic: Design of a network access control system based on the 802.1X protocol for IoT infrastructure.

Objective of the work: designing a network access control system based on the IEEE 802.1X standard, optimized for headless IoT devices, using the EAP-TLS protocol and a separated Public Key Infrastructure (PKI).

Object of research: the process of ensuring information security and network access control for autonomous IoT nodes.

Subject of research: methods and hardware-software tools for implementing network access control based on the IEEE 802.1X standard, EAP-TLS protocol, and PKI tools.

Results of the bachelor's thesis: The vulnerabilities of IoT networks were analyzed, justifying the transition to the Zero Trust paradigm based on the IEEE 802.1X standard. A security architecture was developed using EAP-TLS and a separated certificate authority (PKI) for reliable identification of autonomous devices. A dynamic network microsegmentation mechanism using RADIUS attributes was designed, and a fallback MAC Authentication Bypass (MAB) mode was configured to isolate legacy nodes. System emulation in the GNS3 environment based on open-source software (FreeRADIUS, MikroTik) confirmed its resilience to unauthorized physical penetration and high operational efficiency.

Keywords: INTERNET OF THINGS, NETWORK ACCESS CONTROL, IEEE 802.1X, EAP-TLS, PUBLIC KEY INFRASTRUCTURE, MICROSEGMENTATION, ZERO TRUST, FREERADIUS, MIKROTIK.

ЗМІСТ

с.

ПЕРЕЛІК ОСНОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

AAA – Authentication, Authorization, Accounting

ACL – Access Control List

CA – Certificate Authority

CRL – Certificate Revocation List

DHCP – Dynamic Host Configuration Protocol

DPI – Deep Packet Inspection

EAP – Extensible Authentication Protocol

EAPOL – Extensible Authentication Protocol over LAN

EAP-TLS – Extensible Authentication Protocol Transport Layer Security

IoT – Internet of Things

IP – Internet Protocol

ISO – International Organization for Standardization

L2 – Layer 2 (Канальний рівень базової моделі OSI)

L3 – Layer 3 (Мережевий рівень базової моделі OSI)

MAB – MAC Authentication Bypass

MAC – Media Access Control

NAC – Network Access Control

NIST – National Institute of Standards and Technology

ONU – Optical Network Unit

OSI – Open Systems Interconnection

PKI – Public Key Infrastructure (Інфраструктура відкритих ключів)

PON – Passive Optical Network (Пасивна оптична мережа)

RADIUS – Remote Authentication Dial-In User Service

VLAN – Virtual Local Area Network (Віртуальна локальна мережа)

ZTA – Zero Trust Architecture (Архітектура нульової довіри)

ВСТУП

Актуальність теми дослідження зумовлена масовою інтеграцією IoT-пристроїв у волоконно-оптичні мережі доступу (PON), що суттєво підвищує вимоги до їхнього захисту. Під час проходження переддипломної практики на базі ДП «Нетгруп-Сервіс» було проведено технічний аудит оптичних магістралей, який підтвердив високу пропускну здатність мережі, проте виявив критичну вразливість на рівні кінцевих абонентських терміналів: захист лише на фізичному рівні є недостатнім, оскільки доступ зловмисника до порту комутатора або ONU відкриває прямий вектор атаки на ядро мережі. Це обумовлює необхідність впровадження логічного контролю доступу на канальному рівні (L2).

Згідно з регламентами NIST та ISO, ідентифікація пристрою має передувати наданню мережевого доступу. Ключовим стандартом для ізоляції портів є IEEE 802.1X. Проте його впровадження для IoT ускладнене архітектурою пристроїв: сенсори та контролери працюють у режимі «headless» (без інтерфейсів введення даних), що унеможливлює ручне введення паролів. Традиційна аутентифікація PEAP-MSCHAPv2 тут неприпустима через високі ризики компрометації облікових даних, тому єдиним безпечним рішенням є метод EAP-TLS із використанням цифрових сертифікатів X.509.

Важливою проблемою є також помилкова інтеграція відкритих NAC-рішень, коли функції RADIUS-сервера (AAA-системи, наприклад, FreeRADIUS) плутають із функціями центру сертифікації (CA). Надійний захист вимагає суворого архітектурного розмежування ролей: FreeRADIUS має лише перевіряти сертифікати, тоді як їхня емісія, ротація та відкликання мають реалізовуватися окремим інструментарієм PKI (зокрема, пакетом *easy-rsa*).

Отже, розробка автоматизованої моделі контролю доступу для IoT-інфраструктури, яка усуває системні прогалини у використанні відкритого ПЗ та гарантує ідентифікацію пристроїв до моменту їх підключення, є надзвичайно

актуальною. Це дозволить ДП «Нетгруп-Сервіс» перейти від вразливого фізичного захисту до сучасної, регламентованої міжнародними стандартами системи логічної безпеки мережі.

Мета роботи – проектування системи керування доступом до мережі на основі стандарту IEEE 802.1X, оптимізованої для безінтерфейсних (headless) IoT-пристроїв, з використанням протоколу EAP-TLS та відокремленої інфраструктури відкритих ключів (PKI).

Для досягнення поставленої мети визначено такі завдання дослідження:

1. Проаналізувати вразливості сучасних IoT-мереж та існуючі механізми їхнього захисту.

2. Дослідити архітектурні компоненти та принципи функціонування стандарту IEEE 802.1X.

3. Розробити модель життєвого циклу цифрових сертифікатів (включно з їхньою генерацією засобами *easy-rsa* та відкликанням через списки CRL).

4. Спроєктувати механізм мікросегментації мережі на основі RADIUS-атрибутів.

5. Провести практичну емуляцію та тестування запропонованої системи в середовищі GNS3.

Об’єкт дослідження – процес забезпечення інформаційної безпеки та контролю доступу до мережі для автономних IoT-вузлів.

Предмет дослідження – методи та програмно-апаратні засоби реалізації мережевого контролю доступу на основі стандарту IEEE 802.1X, протоколу EAP-TLS та інструментів PKI.

Методи досліджень: У роботі використано методи системного аналізу для оцінки мережевих топологій та порівняльного аналізу алгоритмів автентифікації. Структурне моделювання застосовано для проєктування логіки ізоляції трафіку. Експериментальна перевірка працездатності запропонованого рішення здійснена методом комп’ютерної емуляції в середовищі GNS3 з використанням віртуальних машин MikroTik CHR та Linux.

Елементи наукової новизни полягають у вдосконаленні архітектури мережевого контролю доступу (NAC) для IoT-середовища шляхом оптимальної інтеграції протоколу EAP-TLS із відокремленою інфраструктурою відкритих ключів (PKI). Запропоновано методику автоматизації життєвого циклу цифрових сертифікатів, яка дозволяє ефективно керувати доступом «headless»-пристроїв без необхідності ручного втручання в конфігурацію комутаційного обладнання.

Практичне значення одержаних результатів полягає у розробці та емуляційній апробації робочої конфігурації системи контролю доступу на базі відкритого програмного забезпечення (FreeRADIUS, easy-rsa, MikroTik). Отримані рішення та інструкції можуть бути безпосередньо впроваджені в інфраструктуру ДП «Нетгруп-Сервіс» для захисту сегмента IoT-пристроїв, що дозволить автоматизувати фільтрацію підключень та мінімізувати ризики несанкціонованого доступу до ядра корпоративної мережі.

Апробація результатів дослідження. Основні положення та результати роботи було представлено на II Міжнародній науково-практичній інтернет-конференції «Synergy of Knowledge: Innovations at the Intersection of Disciplines» (11-12 червня 2026 р., м. Дніпро), де опубліковано тези доповіді у співавторстві з науковим керівником.

Структура роботи: КБР складається зі вступу, чотирьох розділів, висновків, списку використаних джерел із 26 позицій та 1 додатка. Загальний обсяг роботи становить 93 сторінки, містить 25 рисунків та 2 таблиці.

В першому розділі БР було проаналізовано теоретичні основи керування доступом в інфраструктурі Інтернету речей (IoT), базову трирівневу архітектуру, вразливості «headless»-пристроїв та актуальні загрози безпеці мереж. Обґрунтовано необхідність переходу до парадигми Network Access Control (NAC) на базі моделі «Нульової довіри» (Zero Trust) та обрано відкрите програмне забезпечення для реалізації системи.

В другому розділі БР досліджено архітектурні компоненти та механізми стандарту IEEE 802.1X. Детально проаналізовано криптографічний метод EAP-

TLS у зв'язці з інфраструктурою відкритих ключів (PKI), управління життєвим циклом сертифікатів та резервний механізм доступу MAB для застарілого обладнання. Сформовано технічні вимоги для проєктування системи безпеки.

В третьому розділі БР розроблено структурну та логічну схему захищеної мережі, а також покроковий алгоритм автентифікації IoT-пристроїв. Спроєктовано механізм динамічної мікросегментації, який дозволяє автоматично призначати ізольовані VLAN на основі RADIUS-атрибутів для розділення легітимного та неідентифікованого обладнання.

В четвертому розділі БР здійснено практичну реалізацію та комплексне тестування розробленої архітектури у середовищі емуляції GNS3. Емпірично доведено працездатність механізмів ідентифікації EAP-TLS, мікросегментації, резервного доступу MAB та відкликання сертифікатів (CRL). Оцінено стійкість системи до мережевих атак (зокрема, MAC Spoofing) та виміряно операційну ефективність розробленої моделі.

1 ТЕОРЕТИЧНІ ОСНОВИ КЕРУВАННЯ ДОСТУПОМ У МЕРЕЖАХ IOT

1.1 Архітектура та особливості сучасної IoT-інфраструктури

Розбудова сучасних підприємств неминуче вимагає перегляду підходів до організації зв'язку. Якщо ще десять років тому типова корпоративна мережа складалася переважно з комп'ютерів та ноутбуків, за якими працював персонал, то сьогодні в інформаційному обміні домінує техніка. Величезна кількість сенсорів, розумних реле, трекерів та камер здатні безперервно генерувати потоки телеметрії та самостійно реагувати на події. Таке масове використання автономної апаратури дозволяє бізнесу суттєво оптимізувати витрати, проте створює абсолютно нові, нетипові виклики для системних адміністраторів.

Щоб зрозуміти, як саме захищати ці масиви обладнання, фахівці з кібербезпеки спершу розкладають мережу на логічні складові. Найбільш зрозумілою та поширеною в інженерній практиці є концепція розподілу інфраструктури на три основні яруси. Така еталонна модель допомагає чітко розмежувати етапи збору, транспортування та кінцевого аналізу інформації [1].

Найнижчий рівень архітектури IoT, який у міжнародній практиці відомий як сенсорний рівень або рівень збору даних (Perception Layer), виконує функцію первинного збору інформації. До цього рівня належать фізичні пристрої: датчики руху, кліматичні монітори, зчитувачі ідентифікаційних карт, інтелектуальні замки та промислові контролери. Їхнє основне завдання полягає у безперервному моніторингу фізичних параметрів середовища (температури, переміщення, тиску тощо), перетворенні цих аналогових величин на електричні сигнали та формуванні цифрових пакетів даних для подальшої передачі до рівня мережевої обробки.

Проте зібрані пакети не мають жодної цінності, якщо їх безпечно не доставити за призначенням. Цю функцію виконує мережевий рівень (Network Layer), який забезпечує транспортування інформації. До його інфраструктури

входять комутатори, маршрутизатори та спеціалізовані шлюзи, що здійснюють маршрутизацію трафіку через мідні, оптоволоконні або бездротові канали зв'язку [1].

Для інженерів, які займаються питаннями мережевої безпеки, саме цей етап є найважливішим. Адже на рівні комутатора приймається критичне рішення – дозволити новому пристрою передавати свій трафік чи заблокувати його.

Після успішної маршрутизації інформація надходить на рівень застосунків (Application Layer). На цьому рівні розташовані серверні потужності та системи керування базами даних, де відбувається агрегація, систематизація та аналітична обробка отриманих даних. Результатом є формування структурованих звітів та візуалізація показників для підтримки прийняття управлінських рішень, що графічно відображено на загальній схемі архітектури IoT на рисунку 1.1.

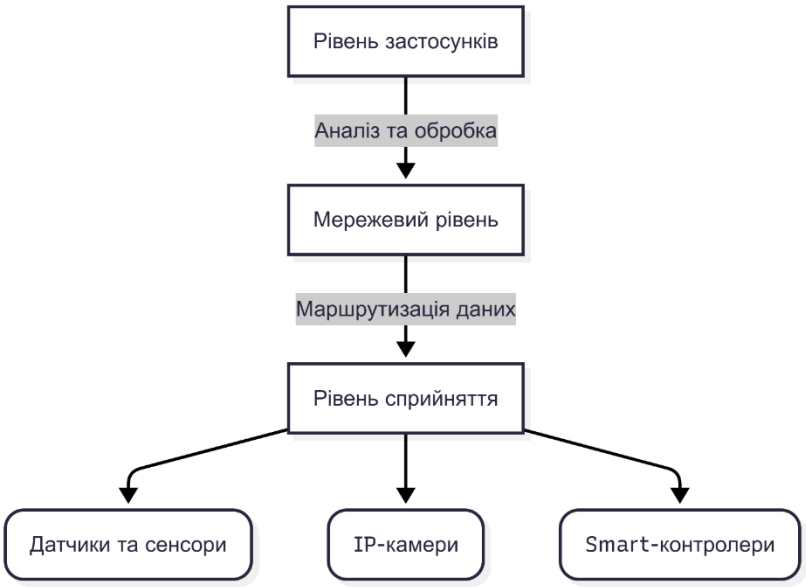

Рисунок 1.1 – Базова трирівнева архітектура мережі Інтернету речей

Спроби інтегрувати надійні механізми перевірки на межі фізичного та мережевого рівнів часто завершуються невдачею, якщо використовувати старі підходи. Причина ховається у специфічній природі самих розумних речей.

По-перше, інженери стикаються з жорсткими апаратними обмеженнями. Щоб датчик міг працювати від однієї батарейки кілька років і коштував мінімум грошей, його обладнують найпростішим мікропроцесором [2]. Отже, змусити такий пристрій генерувати складні криптографічні ключі або запускати важкі захисні алгоритми просто неможливо через брак оперативної пам'яті.

По-друге, виникає проблема відсутності інтерфейсів. Зазвичай під час підключення до робочого Wi-Fi на екрані смартфона чи комп'ютера з'являється вікно для введення логіна та пароля. Проте розумна розетка або пожежний датчик не мають екрана чи клавіатури. Через такий «headless»-формат уся процедура підтвердження легітимності повинна відбуватися приховано, на рівні протоколів, без жодного втручання людини.

По-третє, ситуацію значно ускладнює зоопарк технологій. На території одного підприємства можуть поруч працювати прилади від десятків різних брендів. Вони використовують різні мікропрограми та стандарти зв'язку. Через таку розрізненість системні адміністратори фізично не можуть встановити на всю техніку якусь одну універсальну програму для авторизації.

Окремим критичним фактором ризику є фізична доступність кінцевих пристроїв. Серверні кімнати зазвичай зачинені на замок і охороняються, тоді як вулична камера або датчик на складі доступні практично кожному. Будь-який відвідувач може непомітно витягнути мережевий кабель із приладу та вставити його у свій лептоп. Якщо порт комутатора не захищений, зловмисник миттєво опиняється у внутрішній корпоративній мережі підприємства, минаючи всі зовнішні фаєрволи [2].

Усвідомлення цих ризиків змушує фахівців відмовлятися від застарілих концепцій довіри всередині периметра. Сучасний підхід вимагає розглядати кожен підключений пристрій як потенційну загрозу. Найкращим інструментом для організації жорсткого пропускного режиму безпосередньо на рівні окремих портів є стандарт IEEE 802.1x [3].

Архітектура стандарту IEEE 802.1X базується на взаємодії трьох логічних компонентів. Апарат або програма, що ініціює підключення до локальної мережі виступає в ролі супліканта (Supplicant). Його основним завданням є запуск процесу автентифікації та пред'явлення мережевому обладнанню своїх облікових даних (наприклад, цифрових сертифікатів або унікальних ідентифікаторів пристрою) [3].

Другим ключовим компонентом архітектури виступає автентифікатор (Authenticator). Фізично цю роль зазвичай виконує керований мережевий комутатор або бездротова точка доступу. За замовчуванням логічний порт такого пристрою перебуває в неавторизованому стані. У цьому режимі порт пропускає виключно службові кадри протоколу EAP over LAN (EAPOL), необхідні для ініціації та проходження процесу автентифікації. Будь-який інший мережевий трафік, включно з доступом до ресурсів локальної інфраструктури або глобальної мережі, безумовно блокується до моменту отримання позитивного рішення від сервера автентифікації.

Третім ключовим компонентом архітектури виступає сервер автентифікації (Authentication Server), який приймає остаточне рішення щодо надання мережевого доступу. Сам комутатор (аутентифікатор) не виконує цю функцію самостійно, а діє як посередник, інкапсулює отримані від клієнта облікові дані у пакети протоколу RADIUS та пересилає їх на сервер. AAA-сервер, який зазвичай реалізовано на базі програмного забезпечення типу FreeRADIUS, верифікує надані дані, звіряючи їх із корпоративною базою даних або сховищем сертифікатів. Він аналізує запит, перевіряє права доступу і лише після успішної звірки відправляє комутатору команду – дозволити роботу пристрою чи остаточно вимкнути порт [3]. Детальну хронологію цього обміну наведено на рисунку 1.2.

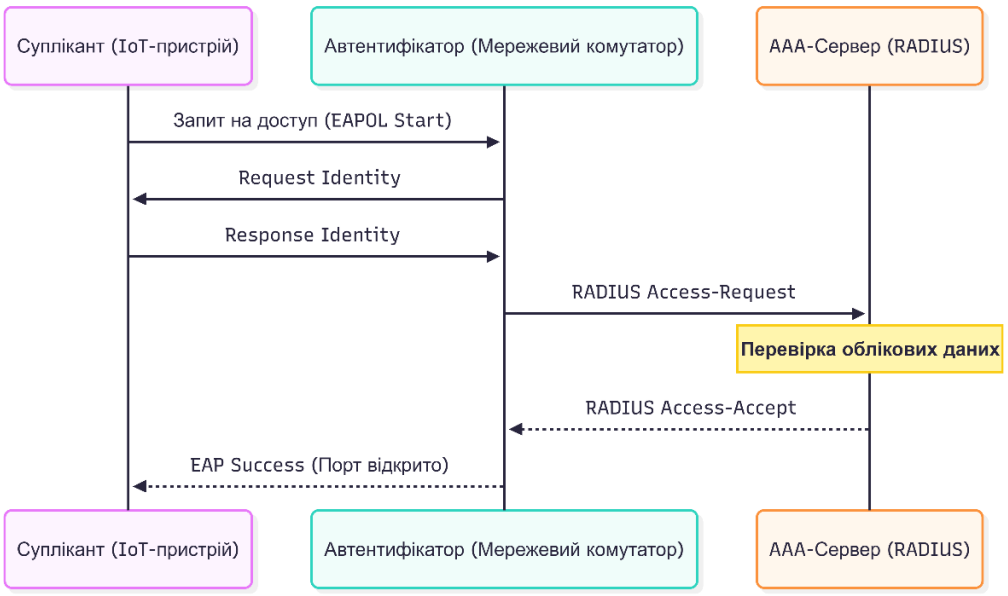

Рисунок 1.2 – Схема взаємодії компонентів під час автентифікації пристрою за стандартом IEEE 802.1x

Завдяки такому чіткому розподілу ролей, впровадження стандарту 802.1x дозволяє реалізувати модель безпеки «Нульова довіра» (Zero Trust). Згідно з цією концепцією, жоден вузол мережі не отримує ніяких привілеїв доступу до моменту успішної криптографічної ідентифікації. Ключовим інженерним завданням при проєктуванні такої системи є коректний вибір методу розширюваної автентифікації (EAP). Для IoT-середовища необхідно обирати алгоритми, які, з одного боку, забезпечують високий рівень захисту, а з іншого – залишаються обчислювально ефективними для пристроїв з обмеженими апаратними ресурсами.

1.2 Вразливості та загрози безпеці мереж IoT

Широке впровадження смарт-пристроїв кардинально змінило правила побудови корпоративних мереж. Колись адміністратори контролювали виключно комп'ютери в межах закритого офісу. Тепер кількість підключеної

техніки стрімко зростає щодня. Різноманітні кліматичні датчики, камери спостереження та розумні розетки додають нові точки входу. Так розширюється потенційна поверхня атаки підприємства. Подібна електроніка зазвичай проектується з максимальною економією апаратних ресурсів. Використовуються дешеві мікроконтролери з мінімальним обсягом оперативної пам'яті. Відповідно, інсталювати туди повноцінний антивірусний софт просто неможливо. Саме тому подібне обладнання вважається ідеальною мішенню для хакерських угруповань.

Окремою складністю є фізичне розташування розумної апаратури. Серверні шафи та головні маршрутизатори традиційно замикають у спеціальних приміщеннях. Доступ туди суворо обмежений. Натомість крайове IoT-обладнання доводиться монтувати безпосередньо в робочих зонах. Це можуть бути відкриті складські площі, паркінги, фасади будинків чи загальні коридори. Мережевий кабель часто підводиться до камери відкритим способом. Будь-яка стороння особа має фізичну можливість підійти і просто від'єднати цей шнур від приладу.

Якщо комутатор не налаштований на жорстку перевірку підключень, виникає серйозна небезпека. Зловмиснику достатньо вставити звільнений кабель у портативний комп'ютер. Більшість сучасних мереж використовують протокол DHCP для автоматичної видачі IP-адрес. Тому новий невідомий апарат майже миттєво отримує внутрішню адресу і стає повноправним учасником інформаційного обміну. Усі потужні зовнішні системи захисту та фаєрволи залишаються осторонь, адже атака розпочинається вже зсередини захищеного периметра.

Щоб зупинити несанкціоновані підключення, часто активують базову функцію Port Security. Цей механізм жорстко прив'язує заводську MAC-адресу конкретного датчика до відповідного порту комутатора. Щойно фіксується спроба передачі даних від іншої мережевої карти, транзит пакетів зупиняється, а порт автоматично блокується. Довгий час цей підхід вважався стандартом базової безпеки.

Сьогодні цей бар'єр обходиться без значних зусиль. Проблема полягає у відкритості апаратних адрес. Вони постійно передаються в ефір без найменшого натяку на шифрування. Для перехоплення не потрібно мати спеціальні навички або дороге устаткування. Достатньо запустити безкоштовний аналізатор трафіку (наприклад, Wireshark), використати пасивний перехідник або просто проаналізувати сигнали найближчої Wi-Fi мережі. Перехоплення навіть одного пакета від легітимної відеокамери одразу розкриває точну заводську адресу пристрою. Далі зловмисник відкриває налаштування операційної системи і вручну прописує скопійовані дані для атакувального мережевого інтерфейсу. Ця елементарна техніка отримала назву MAC Spoofing. Комутатор отримує пакети зі знайомою адресою і без проблем відкриває доступ до ресурсів компанії [6]. Загальну класифікацію актуальних інфраструктурних загроз представлено на рисунку 1.3.

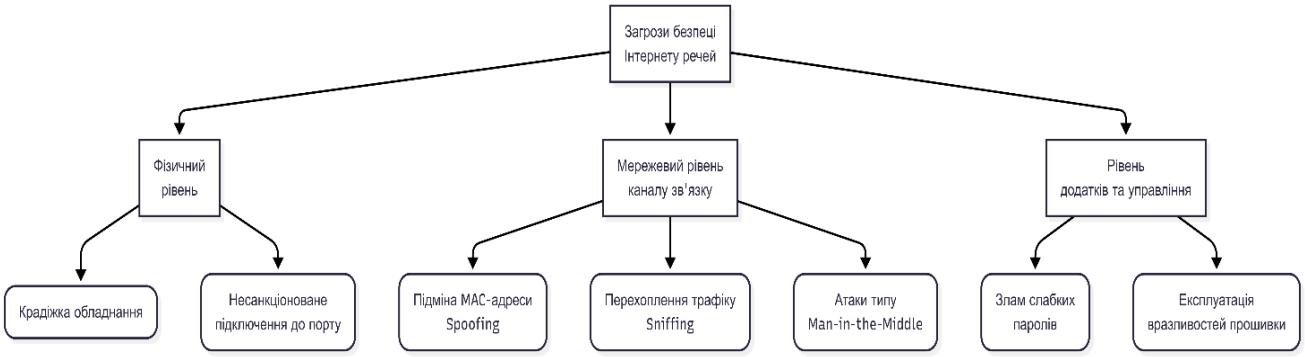

Рисунок 1.3 – Класифікація актуальних загроз в інфраструктурі Інтернету речей

Після успішного обходу первинного захисту починається етап перехоплення внутрішнього трафіку. Найпоширенішим сценарієм є атака «людина посередині» (Man-in-the-Middle). У сегменті локальних мереж вона найчастіше впроваджується шляхом отруєння ARP-кешу. Базовий протокол розв'язання адрес (ARP) створювався на зорі розвитку мережевих технологій і зовсім не містить механізмів перевірки автентичності. Атакувальний комп'ютер

починає масово генерувати фальшиві відповіді. Таким чином імітується ситуація, за якої саме вказана MAC-адреса нібито належить центральному маршрутизатору підприємства [5].

Наслідки такого втручання є вкрай руйнівними. Усі сусідні контролери, сенсори та робочі станції оновлюють таблиці маршрутизації. Відтепер вони відправляють пакети даних не на реальний шлюз, а транзитом через апарат зловмисника. Варто зазначити, що виробники бюджетної електроніки дуже рідко реалізують надійний криптографічний захист через слабкість процесорів. Інформація часто подорожує у вигляді звичайного тексту через застарілі протоколи HTTP або Telnet. Завдяки цьому зловмисник отримує змогу непомітно збирати паролі адміністраторів, аналізувати службові команди та вільно переглядати потокове відео з камер безпеки [2].

Окремо варто виділити загрози глобального масштабу. Значний відсоток техніки надходить у продаж зі стандартними налаштуваннями доступу (комбінації admin, root, 12345). Користувачі не обтяжують себе зміною базових параметрів, а виробники економлять кошти на випуску патчів безпеки. Це створює ідеальне середовище для побудови гігантських ботнетів. Найбільш відомим випадком стала епідемія шкідливого коду Mirai. Програма цілодобово сканувала інтернет-простір, виявляла відкриті порти та підбирала стандартні логіни. Заражені маршрутизатори перетворювалися на керовану армію. За командою ця мережа починала генерувати колосальний обсяг сміттєвих запитів. Такі DDoS-атаки виявилися настільки потужними, що з легкістю виводили з ладу дата-центри світових корпорацій [7].

У межах однієї організації компрометація навіть непримітного температурного датчика здатна спричинити колапс. Атакуючі рідко обмежуються лише одним зламаним приладом. Скомпрометований вузол перетворюється на зручний плацдарм для подальшого сканування інфраструктури (техніка Lateral Movement). Внутрішні захисні екрани зазвичай налаштовані на цілковиту довіру до локальних IP-адрес. Тому спроби брутфорсу

баз даних або сканування портів можуть тривати тижнями абсолютно непомітно, якщо джерелом атаки виступає розумний телевізор у кімнаті відпочинку.

Бездротові сегменти мережі утворюють ще одну критичну вразливість. Розумну апаратуру часто підключають до Wi-Fi за допомогою єдиного статичного ключа (WPA2-PSK). Якщо цей пароль стає відомим стороннім особам або звільненим співробітникам, захист усього радіосегмента перестає існувати. Змінити ключ доступу на сотнях датчиків без дисплеїв – надзвичайно складна та тривала процедура для адміністратора. Через це загальні паролі на підприємствах не оновлюються роками, створюючи величезну діру в контурі безпеки, а сценарій реалізації атаки із підміною ідентифікатора проілюстровано на рисунку 1.4.

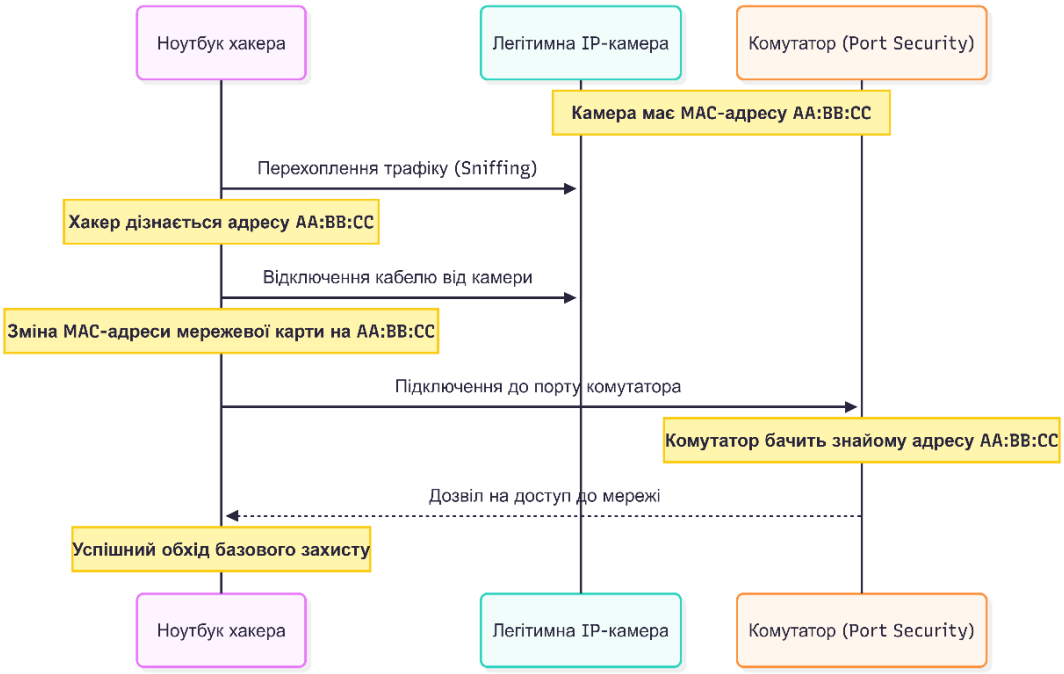

Рисунок 1.4 – Сценарій обходу базового захисту порту за допомогою підміни MAC-адреси

Усі наведені приклади доводять неспроможність застарілих інструментів безпеки в епоху Інтернету речей. Прив'язка до MAC-адрес обходиться за

хвилини, загальні паролі швидко втрачають конфіденційність, а закрити кожну вуличну камеру в сейф технічно неможливо. Надійна архітектура вимагає блокування портів на логічному рівні з обов'язковою криптографічною перевіркою апаратури ще до моменту отримання IP-адреси. Саме ці завдання ефективно вирішує технологія Network Access Control на базі стандарту IEEE 802.1x [3].

1.3 Концепція Network Access Control (NAC) та огляд рішень

Вивчення сучасних кіберзагроз чітко показує неефективність старої моделі захисту мережевого периметра. Раніше достатньо було встановити потужний фаєрвол на вході в компанію. Сьогодні ж межі корпоративної інфраструктури повністю розмиті через величезну кількість смарт-камер, бездротових сенсорів та мобільних терміналів. Загроза часто виникає вже всередині захищеного приміщення. Тому фахівці з безпеки почали масово переходити до нової парадигми захисту. Вона отримала назву Network Access Control (NAC), тобто керування доступом до мережі [8].

Системи класу NAC – це комплекс інструментів, які суворо контролюють кожне підключення. Вони визначають, який саме пристрій намагається зайти в мережу, які права йому можна надати та за яких умов. Звичайні фаєрволи просто фільтрують готовий потік даних. Натомість системи керування доступом діють набагато раніше. Вони починають перевірку в ту саму секунду, коли конектор кабелю вставляється в розетку або коли прилад надсилає перший запит до Wi-Fi роутера.

В основі таких систем лежить класична модель AAA (автентифікація, авторизація, облік). Перший крок вимагає підтвердження легітимності. Пристрій має надати правильний пароль, цифровий сертифікат або унікальний токен. Другий етап відповідає за розподіл прав. Наприклад, системний блок бухгалтера отримає доступ до фінансових баз, а камера спостереження зможе передавати

дані лише на сервер відеореєстратора. Третій крок забезпечує безперервне логування. Усі дії підключеної техніки записуються для можливих майбутніх розслідувань кіберінцидентів.

Дуже важливою складовою є функція профілювання (Device Profiling). Розумні датчики не мають екранів і не можуть самостійно ввести облікові дані. Тому мережа повинна вміти розпізнавати їх за непрямими слідами. Спеціальні аналізатори пасивно слухають трафік. Вони збирають стартові DHCP-запити, перевіряють відкриті порти та читають службові заголовки протоколів. Завдяки такому аналізу система точно розуміє, що до комутатора підключено саме оригінальний кліматичний датчик, а не ноутбук хакера з підробленою MAC-адресою [9].

Роботу системи ділять на два основні етапи. Спочатку йде фаза допуску (Pre-admission). Комутатор блокує порт і дозволяє передавати лише короткі службові повідомлення для перевірки, зокрема пакети стандарту 802.1x. Після вдалої автентифікації починається фаза після допуску (Post-admission). Сервер безпеки дає комутатору команду динамічно призначити порту відповідну віртуальну мережу (VLAN) для трафіку автентифікованого пристрою і застосувати списки контролю доступу (ACL). Слід розуміти, що сам порт фізично нікуди не «переводиться», він залишається тим самим портом, але йому динамічно призначається VLAN для цієї конкретної сесії. Якщо ж пристрій раптом почне поводитися підозріло, система може миттєво розірвати сесію та ізолювати трафік у карантинній зоні [8], що відображено на узагальненому алгоритмі прийняття рішень на рисунку 1.5.

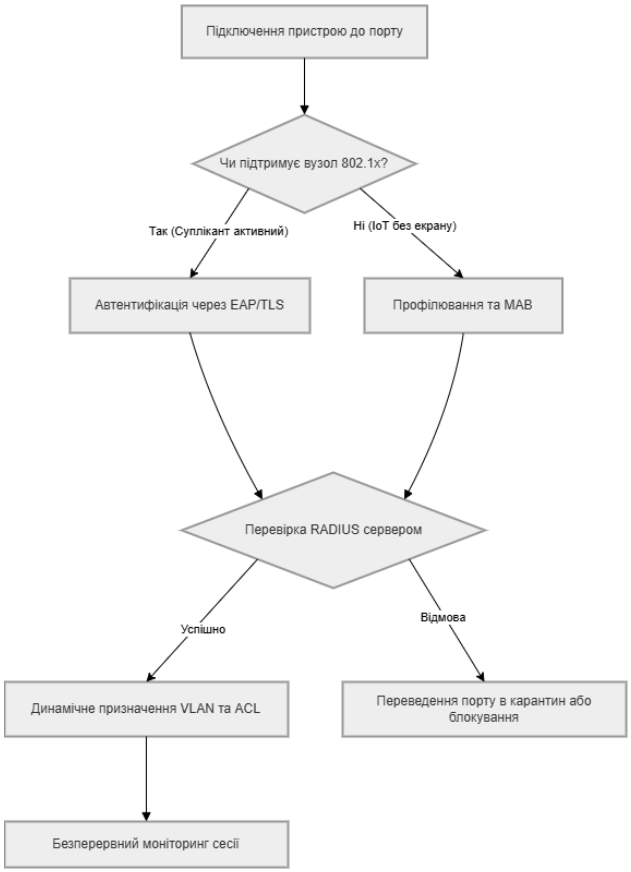

Рисунок 1.5 – Узагальнений алгоритм прийняття рішень у системі Network Access Control

Великі корпорації зазвичай використовують потужні комерційні платформи класу Enterprise NAC. Найвідомішим рішенням тут є Cisco Identity Services Engine (ISE). Це величезний комплекс, здатний профілювати тисячі типів обладнання та створювати динамічну сегментацію мережі. Іншим популярним продуктом є Aruba ClearPass. Ця система дуже гнучка і добре працює з обладнанням від різних виробників [10].

Але використання таких комерційних рішень має серйозні недоліки. Головна проблема – це надзвичайно висока вартість ліцензій. Ціна часто залежить від кількості підключених клієнтів. Для середніх компаній або для побудови лабораторних стендів такі витрати просто не мають сенсу. До того ж,

ці системи повноцінно розкривають свій потенціал лише на дуже дорогих преміальних комутаторах.

Тому фахівці часто звертаються до рішень з відкритим кодом (Open Source). Певний час дуже популярною була безкоштовна система PacketFence. Вона добре вміє ізолювати пристрої та визначати їхні типи. Проте її налаштування є досить складним, а сама архітектура вимагає багато ресурсів і специфічних версій прошивок на мережевому обладнанні.

Для створення гнучкої, надійної і при цьому доступної системи керування доступом найкраще підходить використання виділеного RADIUS-сервера. Тут абсолютним лідером є платформа FreeRADIUS. Це найвідоміший у світі сервер автентифікації з відкритим кодом. Його щодня використовують тисячі інтернет-провайдерів та компаній по всьому світу [11].

Більшість існуючих відкритих NAC-рішень орієнтовані на автентифікацію користувачів за допомогою логіна та пароля (метод PEAP-MSCHAPv2), що є абсолютно непридатним для headless IoT-пристроїв. Натомість величезна перевага FreeRADIUS – це модульність та повна підтримка методу EAP-TLS. Сервер можна налаштувати на роботу зі складними цифровими сертифікатами для автономного обладнання. Важливо зазначити, що сам FreeRADIUS не виконує функцій центру сертифікації (CA), а лише використовує створені окремим CA сертифікати для перевірки клієнтів. Паралельно він може перевіряти застарілу техніку просто за MAC-адресами (MAB). При цьому FreeRADIUS не обмежує кількість клієнтів і працює абсолютно безкоштовно [11].

Щоб протокол 802.1x працював коректно, потрібне мережеве обладнання, яке візьме на себе роль комутатора-автентифікатора. Оптимальним балансом між ціною та багатим функціоналом сьогодні є рішення від компанії MikroTik. Фірмова операційна система RouterOS має вбудовану підтримку потрібних стандартів. Вона чудово працює з VLAN та вміє динамічно керувати доступом на рівні порту. Зв'язка MikroTik та FreeRADIUS дозволяє отримати повноцінний

функціонал систем корпоративного рівня. При цьому можна використовувати досить бюджетне апаратне забезпечення або віртуальні середовища [12]. На рисунку 1.6 наведено схему взаємодії компонентів системи.

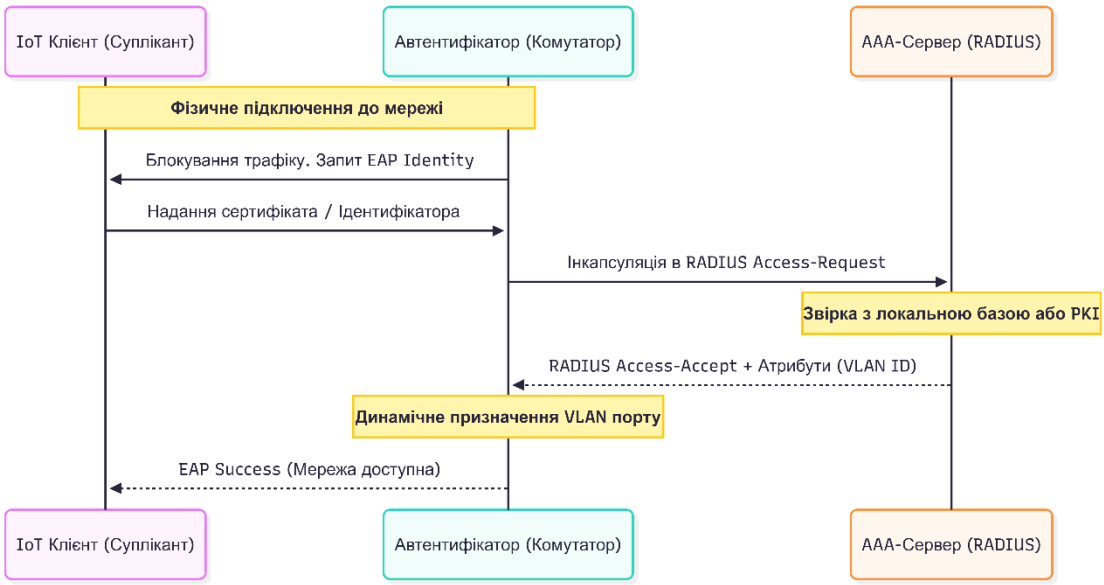

Рисунок 1.6 – Проектна схема взаємодії платформ MikroTik та FreeRADIUS під час підключення клієнта

Після фізичного підключення IoT-клієнта (супліканта) аутентифікатор (MikroTik) блокує трафік і надсилає запит EAP Identity. Отримані від клієнта дані комутатор інкапсулює у пакет RADIUS Access-Request і пересилає на AAA-сервер (FreeRADIUS) для звірки з PKI або локальною базою. У разі успіху сервер повертає RADIUS Access-Accept із атрибутами авторизації (зокрема, VLAN ID). На основі цих даних комутатор динамічно призначає порт у відповідний VLAN і надсилає клієнту EAP Success, відкриваючи мережевий доступ.

Для емпіричної перевірки моделі в середовищі GNS3 розгорнуто програмний стенд на базі Linux, FreeRADIUS та MikroTik RouterOS. Для емулювання поведінки комутатора доступу інтерфейси MikroTik CHR об'єднано в bridge-інтерфейс, на портах якого активовано 802.1X. Це дозволяє

застосовувати політики автентифікації до кожного фізичного підключення окремо, незалежно від L3-маршрутизації. Використання програмного гіпервізора GNS3 дозволяє розгорнути повноцінний стенд і наочно перевірити роботу стандарту 802.1x в умовах, максимально наближених до реальних корпоративних мереж.

Висновки до розділу 1

У першому розділі виконано аналіз теоретичних основ керування доступом в інфраструктурі Інтернету речей. Розглянуто базову трирівневу архітектуру IoT-мереж та виявлено ключові проблеми захисту «headless»-пристроїв, зумовлені їхніми жорсткими апаратними обмеженнями, відсутністю інтерфейсів введення та високою фізичною доступністю для зловмисників.

Досліджено актуальні інфраструктурні загрози безпеці, такі як підміна MAC-адреси (MAC Spoofing), атаки типу «людина посередині» (Man-in-the-Middle) та експлуатація застарілих загальних ключів доступу. Доведено, що традиційні методи захисту, зокрема базова функція Port Security на рівні комутаторів, є неефективними в сучасних реаліях і легко обходяться атакуючими.

Обґрунтовано необхідність відмови від застарілої концепції довіри всередині периметра та переходу до парадигми Network Access Control (NAC) на базі моделі «Нульової довіри» (Zero Trust). Як найефективніший стандарт логічного контролю доступу на канальному рівні визначено IEEE 802.1X.

Проведено огляд існуючих систем класу NAC та обґрунтовано вибір інструментів з відкритим вихідним кодом (Open Source). Для виконання ролі AAA-сервера обрано платформу FreeRADIUS із підтримкою криптографічного методу EAP-TLS, оскільки традиційна автентифікація за паролями для сенсорів є неприйнятною. Роль автентифікатора доручено мережевому обладнанню

MikroTik (RouterOS), яке забезпечує динамічне керування VLAN та ізоляцію портів.

Сформовано загальну проєктну модель взаємодії компонентів під час автентифікації пристрою та обґрунтовано доцільність розгортання програмного стенда в середовищі емуляції GNS3. Це формує теоретичне та архітектурне підґрунтя для подальшої практичної реалізації системи мікросегментації мережі.

2 АНАЛІЗ МЕХАНІЗМІВ СТАНДАРТУ IEEE 802.1X ТА ОСОБЛИВОСТІ ЙОГО ЗАСТОСУВАННЯ В IoT-ІНФРАСТРУКТУРІ

2.1 Принципи роботи та компоненти архітектури 802.1x

Створення стійких до кібератак корпоративних інфраструктур вимагає застосування перевірених стандартів контролю за мережевим обладнанням. Концепція «нульової довіри» сьогодні спирається на надійний технологічний фундамент – стандарт IEEE 802.1x. Історично цей інструмент розробляли для вирішення проблеми несанкціонованого доступу до дротових мереж на рівні фізичних роз'ємів комутатора. Проте закладені в нього алгоритми виявилися настільки універсальними, що згодом їх інтегрували у бездротові мережі. На даний момент ця технологія визнана найефективнішим способом ізоляції вразливих вузлів в екосистемі Інтернету речей.

Ключовий принцип функціонування стандарту IEEE 802.1X полягає у логічному блокуванні мережевого порту до моменту успішної автентифікації клієнта. Просте фізичне під'єднання патч-корду до розетки комутатора або успішна радіоасоціація з точкою доступу більше не забезпечують отримання IP-адреси. Мережевий порт залишається повністю заблокованим для стандартного трафіку. Дозволяється проходження виключно одного типу пакетів – спеціальних службових фреймів, які забезпечують процес криптографічної ідентифікації пристрою [3].

Внутрішня архітектура стандарту 802.1x утворює замкнену систему з трьох незалежних об'єктів. Кожен вузол цієї системи виконує суворо регламентовану функцію та застосовує специфічні протоколи для передачі інформації. Детальне розуміння ролі кожного учасника дозволяє безпомилково проектувати топологію безпеки підприємства.

Ініціатором процесу перевірки завжди виступає кінцеве обладнання. У нормативній документації цей об'єкт отримав назву супліканта (Supplicant).

Важливо розуміти, що суплікант – це не фізична мікросхема, а фонова програмна служба. Її призначення полягає в безперервному моніторингу стану мережевого адаптера. При виявленні нового з'єднання програма формує запит на автентифікацію та забезпечує безпечну доставку облікових даних. Класичні операційні системи зазвичай містять такі служби за замовчуванням. Проте сегмент смарт-пристроїв має іншу специфіку, оскільки там домінують рішення на базі ядра Linux. Для ініціалізації перевірки на таких пристроях, включаючи міні-комп'ютери з операційною системою Lubuntu, системні адміністратори інтегрують пакет wpa_supplicant. Ця утиліта здатна інкапсулювати цифрові сертифікати або хешовані паролі у відповідні фрейми для відправки в магістральну мережу.

Роль проміжного бар'єра відведено автентифікатору (Authenticator). Фізично це транзитне комутаційне обладнання, до якого безпосередньо підключається клієнт. Таку функцію зазвичай виконують комутатори доступу або бездротові радіомодулі. Під час побудови тестових середовищ та розгортання корпоративних систем цю задачу успішно вирішують пристрої компанії MikroTik. Їхня операційна система RouterOS підтримує необхідні мережеві стандарти «з коробки». Автентифікатор діє як суворий контрольно-пропускний вузол. Цей пристрій позбавлений права самостійно приймати рішення щодо надання доступу. Його ключова функція полягає у перехопленні запитів від клієнтської програми, їх перепакуванні у захищений формат та пересиланні до головної бази даних. До моменту надходження позитивного рішення комутатор апаратно блокує транзит користувацьких даних [12].

Програмна логіка блокування працює завдяки модулю Port Access Entity (PAE). Цей модуль програмно розділяє один фізичний роз'єм на два незалежні логічні канали. Перший канал класифікується як неконтрольований порт (Uncontrolled Port). Він постійно перебуває у відкритому стані. Однак жорсткі налаштування фільтрації дозволяють пропускати через цей інтерфейс виключно трафік протоколу автентифікації. Будь-які сторонні пакети ігноруються. Друга

логічна сутність – контрольований порт (Controlled Port). Його створено для пропуску звичайних даних, включаючи протоколи DHCP, DNS та загальний інтернет-трафік. Базовий стан цього порту завжди визначений як «Неавторизовано» (Unauthorized). Лише після успішного підтвердження криптографічних ключів модуль PAE змінює цей стан на «Авторизовано» (Authorized). Відтоді кінцеве обладнання здатне повноцінно функціонувати [3].

Кінцевим центром прийняття рішень призначається сервер автентифікації (Authentication Server). Це виділений вузол обчислювальної інфраструктури. На його накопичувачах зберігається єдина база даних працівників, апаратних ідентифікаторів та цифрових сертифікатів. Отримавши транзитний запит від комутатора, сервер здійснює розпакування даних, виконує математичну перевірку з локальними реєстрами і формує остаточний висновок. Сервер також здатний відправляти комутатору розширені мережеві атрибути. До таких атрибутів належить точний номер віртуальної локальної мережі (VLAN ID), куди необхідно ізолювати порт після успішної перевірки. Найбільш розповсюдженим рішенням для виконання таких функцій є відкрите програмне забезпечення FreeRADIUS. Ця платформа гарантує високу стабільність, підтримує модульне розширення та миттєво обробляє складні математичні алгоритми [11].

Синхронізація дій між компонентами архітектури базується на двох спеціалізованих протоколах. Комунікація між кінцевим пристроєм та транзитним комутатором забезпечується стандартом EAPOL (Extensible Authentication Protocol over LAN). Цей інструмент дозволяє транспортувати дані безпосередньо всередині кадрів Ethernet. Призначення спеціального ідентифікатора EtherType 0x888E виділяє ці кадри із загального потоку. Трансляція відбувається ще до того моменту, як пристрою буде призначена IP-адреса. Без наявності локальної адреси пристрій позбавлений можливості взаємодіяти зі стеком TCP/IP. Завдяки формату EAPOL процедура ідентифікації здійснюється на другому (канальному) рівні моделі OSI [13].

На магістральній ділянці між комутатором та центральним сервером маршрутизація даних спирається на протокол RADIUS (Remote Authentication Dial-In User Service). Мережевий комутатор витягує пакети EAP, отримані від клієнта, інкапсулює їх у магістральні RADIUS-повідомлення та транслює через захищений корпоративний канал. Зазвичай для цієї взаємодії адміністратори використовують порти UDP 1812 та 1813. Сервер обробляє отриманий масив і повертає криптографічну відповідь зворотним маршрутом, що відповідає базовій моделі функціонування логічних портів [14]. Цей процес відображено на рисунку 2.1.

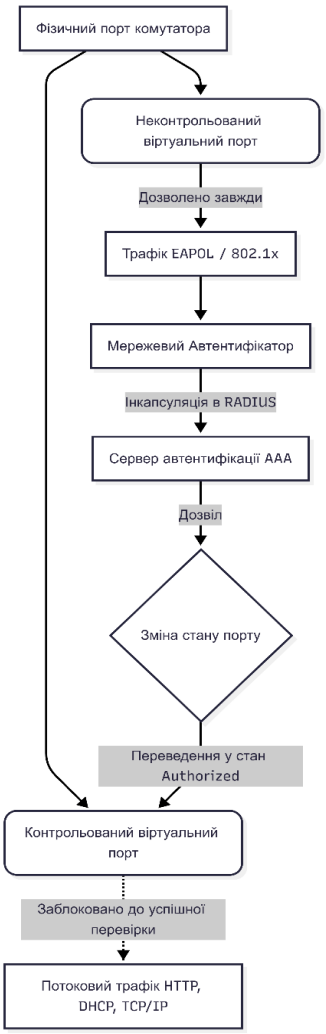

Рисунок 2.1 – Модель роботи віртуальних портів під час автентифікації

Практична реалізація підключення виглядає як багатоетапний діалог між пристроями. Стартовою точкою стає фізичне підключення патч-корду. Програмна служба wpa_supplicant реєструє появу лінку і негайно генерує кадр EAPOL-Start. Цей широкомовний імпульс сповіщає транзитне обладнання про присутність нового вузла, який готовий пройти перевірку [3].

Комутатор приймає стартовий імпульс і формує у відповідь запит EAP-Request/Identity. Суплікант опрацьовує цей пакет і відправляє відповідь EAP-Response/Identity, що містить базове мережеве ім'я. Варто зазначити, що на цьому кроці фігурує виключно відкритий текстовий ідентифікатор. З міркувань безпеки, жодні цифрові ключі чи паролі поки що в ефір не транслюються. Це унеможливлює їхнє пасивне перехоплення [13].

Мережевий комутатор не проводить аналіз цього імені. Дані розміщуються всередині службового повідомлення RADIUS Access-Request і спрямовуються на сервер FreeRADIUS. Починаючи з цієї секунди, автентифікатор перетворюється на прозорий канал зв'язку. Серверна платформа отримує пакет, виконує пошук ідентифікатора у внутрішніх реєстрах та встановлює оптимальний метод криптографічної перевірки [14].

Наступна фаза передбачає узгодження методів розширюваної автентифікації (EAP). Сервер генерує повідомлення RADIUS Access-Challenge, яке містить специфічне криптографічне завдання. Комутатор розпаковує цей контейнер і транслює завдання клієнту у формі кадру EAP-Request. Програма супліканта виконує необхідні обчислення. Відбувається шифрування контрольного рядка приватним ключем або генерація криптографічного хешу. Отриманий результат повертається комутатору як пакет EAP-Response. Залежно від вимог обраного стандарту захисту, така ітерація запитів може повторюватися декілька разів [13].

Після надходження фінальної відповіді центральний сервер проводить розрахунки. Математичний збіг підтверджує легітимність вузла. Доводиться факт наявності правильного цифрового сертифіката. Сервер формує дозвільний

сигнал RADIUS Access-Accept і надсилає його автентифікатору. У цьому ж повідомленні часто містяться динамічні атрибути, зокрема інструкції щодо цільового VLAN [14].

Отримавши такий дозвіл, комутатор сигналізує клієнту про успішне завершення перевірки пакетом EAP-Success. Одночасно програмний модуль автентифікатора знімає апаратне блокування. Логічний статус контрольованого віртуального порту змінюється на «Авторизовано». Фізичний роз'єм повністю відкривається для транспортування даних. Пристрій успешно ініціює DHCP-запит, прописує конфігурацію IP-адреси та інтегрується в захищене середовище [13]. Деталізований покроковий діалог між компонентами структури наведено на рисунку 2.2.

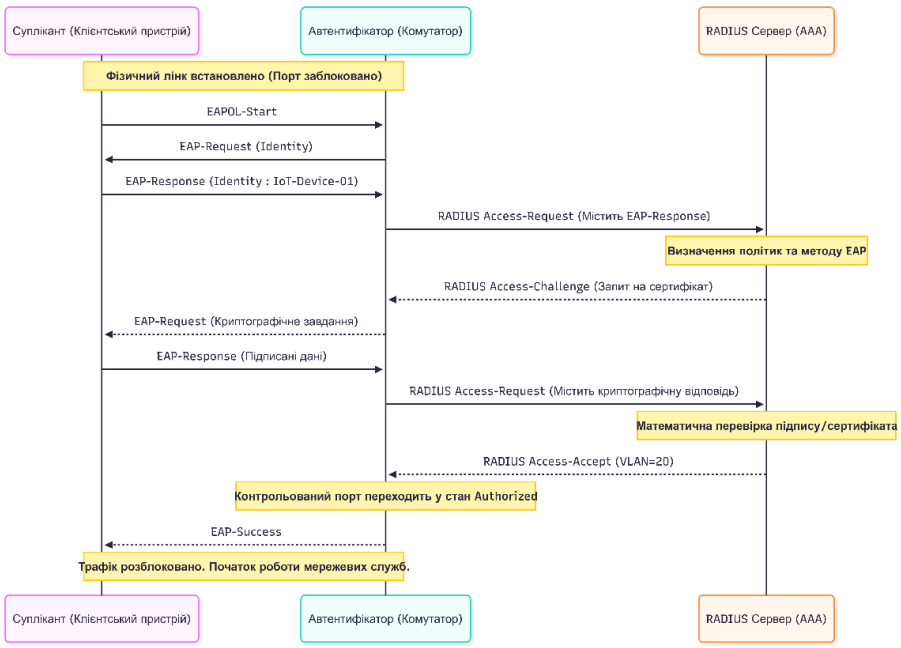

Рисунок 2.2 – Деталізована схема обміну повідомленнями під час процесу 802.1x

Описана багаторівнева взаємодія надійно блокує спроби несанкціонованого втручання в корпоративну інфраструктуру. Ефективність

даної моделі прямо пропорційна якості налаштування методу EAP. У традиційних мережах системні адміністратори віддають перевагу методам з використанням паролів (наприклад, PEAP). Проте екосистема Інтернету речей диктує інші правила. Відсутність графічних інтерфейсів унеможливлює використання ручного введення даних. Найбільш логічним і захищеним варіантом для автономного обладнання визнано впровадження цифрових сертифікатів. Аналіз алгоритмів шифрування, розгортання інфраструктури відкритих ключів (PKI) та робота механізму EAP-TLS є наступним кроком до побудови повноцінної системи керування доступом.

2.2 Аналіз методів EAP та інфраструктури відкритих ключів (PKI) - акцент на EAP-TLS

Організація базового транспортного тунелю між кінцевим обладнанням та сервером вирішує лише частину безпекових завдань. Специфікація стандарту IEEE 802.1X є абсолютно індиферентною до застосовуваних засобів криптографії. Вона виконує виключно транзитну функцію, здійснюючи комутацію службових пакетів від порту до бази даних. Завдання безпосереднього аналізу автентифікаційних маркерів, парольних фраз або сертифікатів повністю делегується протоколу розширюваної автентифікації – EAP. Інженерна архітектура цього інструмента реалізована у вигляді універсальної цифрової оболонки, всередину якої інтегруються найрізноманітніші алгоритми перевірки та шифрування. Передача криптографічних матеріалів здійснюється всередині стандартизованих мережевих кадрів, що дозволяє гнучко масштабувати рівень захисту інфраструктури залежно від апаратних та обчислювальних можливостей підключеного обладнання [13].

Глобальний реєстр методів EAP налічує десятки специфікацій, кожна з яких створювалася для подолання конкретних вразливостей свого часу.

Проєктування сучасних безпекових рішень неможливе без розуміння критичних недоліків застарілих механізмів. Найпримітивнішим історичним стандартом визнано специфікацію EAP-MD5. Її обчислювальна механіка базується на односторонньому хешуванні без використання цифрових сертифікатів. Центральний сервер автентифікації формує масив випадкових символів (challenge) і транслює його кінцевому вузлу. Апаратна частина клієнта комбінує отримане значення з локальним секретним паролем, застосовує математичну трансформацію MD5, а отриманий зліпок повертає назад. Такі обчислення практично не навантажують мікроконтролери, проте на сьогодні цей підхід вважається вразливим. Його фундаментальним недоліком є відсутність механізму зустрічної перевірки, через що автономний пристрій позбавлений можливості переконатися у справжності самого сервера. Додатково фіксується абсолютна криптографічна слабкість хешу MD5. Інтеграція цієї специфікації в нові інфраструктурні проєкти суворо заборонена.

Значно вищий рівень ізоляції пропонує алгоритм PEAP (Protected Extensible Authentication Protocol), який у корпоративному середовищі зазвичай застосовується в комбінації з протоколом MSCHAPv2. Концепція методу вимагає створення надійно зашифрованої транзитної магістралі, всередині якої відбувається прихована передача стандартних облікових даних. Ключова перевага PEAP полягає в оптимізації адміністративних процесів: складний цифровий сертифікат розгортається виключно на стороні центрального сервера, тоді як робочі станції персоналу додаткових сертифікатів не потребують. Однак спроби застосування методу PEAP в екосистемі Інтернету речей створюють парадоксальні вразливості. Смарт-апаратура працює в режимі повної автономії, і графічні інтерфейси для інтерактивного введення логіна та пароля не передбачені конструктивно. Системний інженер змушений зберігати облікові дані від мережі безпосередньо у внутрішній пам'яті контролера у вигляді відкритого тексту. Оскільки таке обладнання часто монтується за межами

охоронюваного периметра, зловмиснику достатньо фізично демонтувати пристрій та зчитати пароль із флеш-пам'яті [4].

Нейтралізація ризику викрадення паролів вимагає переходу до систем сертифікатної ідентифікації. Еталонним стандартом захисту автономного обладнання беззаперечно визнано специфікацію EAP-TLS. Цей протокол повністю відкидає використання текстових паролів – їх замінюють унікальні цифрові документи. При цьому процедура валідації завжди реалізується у двосторонньому синхронному форматі (Mutual Authentication), де сервер математично доводить свою легітимність смарт-пристрою, а пристрій криптографічно підтверджує свої права серверу [15].

2.2.1 Архітектурні принципи розгортання інфраструктури відкритих ключів (PKI)

Упровадження алгоритму автентифікації EAP-TLS передбачає обов'язкове попереднє формування інфраструктури відкритих ключів (Public Key Infrastructure, PKI). У контексті даного дослідження та загальноприйнятих вимог кібербезпеки критично важливим є архітектурне розмежування функцій емісії та перевірки криптографічних матеріалів.

Платформа FreeRADIUS, яка функціонує як ядро системи контролю доступу у розробленому стенді, не виконує функцій центру сертифікації (Certificate Authority, CA) [18]. Її призначення зводиться виключно до реалізації моделі AAA, де сервер здійснює математичну перевірку та порівняння наданих кінцевим вузлом електронних ідентифікаторів з еталонними відбитками. Сервер автентифікації не залучається до процесу безпосередньої генерації сертифікатів.

З метою забезпечення прозорості криптографічних процесів і дотримання парадигми розподіленої безпеки, роль емітента сертифікатів (CA) виділено в незалежну логічну сутність. У межах корпоративної інфраструктури цю функцію

реалізовано за допомогою спеціалізованої утиліти easy-rsa, що є стандартизованою надбудовою над криптографічною бібліотекою OpenSSL [24].

Структура інфраструктури PKI для забезпечення функціонування стандарту IEEE 802.1X передбачає створення трьох базових компонентів:

1. кореневий сертифікат (Root CA)**:** базовий еталон, який виконує функцію якоря довіри (Trust Anchor) для всього мережевого периметра та застосовується для підписання підпорядкованих документів системи [17];

2. серверний сертифікат: цифровий ідентифікатор, що розгортається на сервері FreeRADIUS і забезпечує можливість математичного підтвердження автентичності центрального сервера для IoT-пристрою на етапі встановлення з'єднання (з метою запобігання атакам типу «злий двійник») [18];

3. клієнтські сертифікати: набір індивідуальних ідентифікаторів, що використовуються як унікальні безпарольні мандати доступу для кожного автономного сенсора або контролера [18].

Для забезпечення довгострокової стійкості системи визначено життєвий цикл сертифікатів, що відповідає рекомендаціям NIST SP 800-57 та практикам Zero Trust. Узагальнені етапи життєвого циклу криптографічних матеріалів та їхню технічну реалізацію наведено в таблиці 2.1.

Таблиця 2.1 – Етапи життєвого циклу сертифікатів в IoT-інфраструктурі

Етап	**Опис**	**Технічна реалізація**
Генерація	Створення Root CA та підписання серверних/клієнтських сертифікатів	Утиліта easy-rsa + Bash-оркестрація
Розгортання	Безпечне розміщення сертифікатів на кінцевих пристроях	Виключення незахищених протоколів (HTTP/FTP); використання SCP/SFTP або фабричного впровадження
Ротація	Планова заміна сертифікатів клієнтів	Автоматизоване оновлення кожні 90 днів; підготовка до інтеграції з протоколом EST (RFC 7030)
Відкликання	Миттєве блокування скомпрометованих пристроїв	Оновлення crl.pem − перезапуск служби FreeRADIUS − автоматична відмова в Access-Accept без зміни конфігурації комутаторів

Виокремлення PKI як незалежного компоненту архітектури забезпечує масштабованість, зменшує поверхню атак на сервер автентифікації та відповідає вимогам сучасних стандартів кібербезпеки для гетерогенних IoT-середовищ (рис. 2.3).

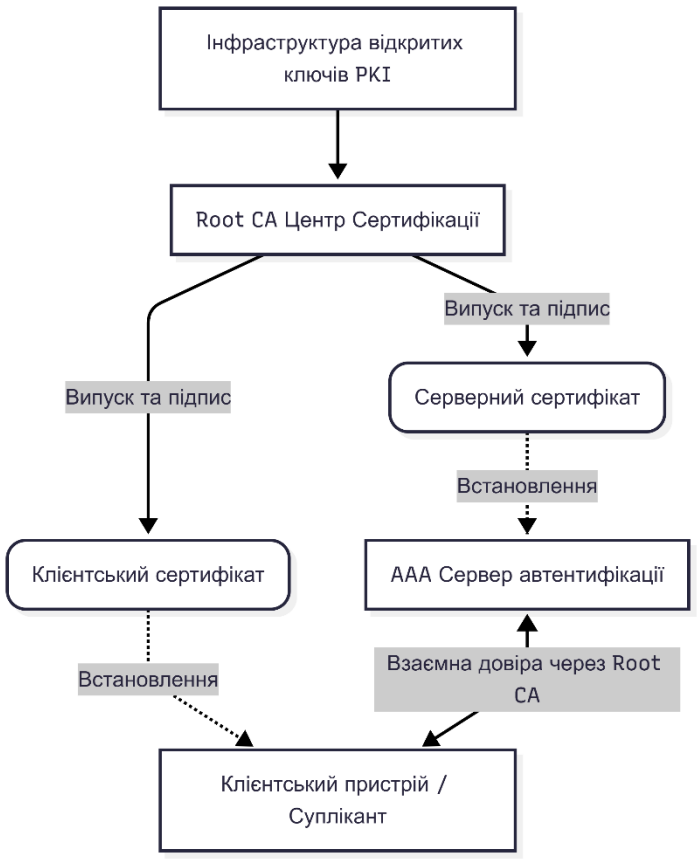

Рисунок 2.3 – Модель розподілу довіри в інфраструктурі PKI

Таким чином, логічне розмежування процесів емісії цифрових мандатів та їхньої магістральної перевірки формує надійний фундамент для побудови корпоративної системи контролю доступу. Наявність автономного центру сертифікації гарантує, що процедура криптографічного узгодження між периферійним клієнтом та сервером відбуватиметься у повністю захищеному, довіреному середовищі, що є критично необхідною умовою для безпечної реалізації протоколу EAP-TLS на наступних етапах автентифікації.

2.2.2 Роль RADIUS-сервера та етапи криптографічного узгодження EAP-TLS

Відокремивши процеси емісії криптографічних матеріалів, необхідно деталізувати безпосередню логіку роботи RADIUS-сервера. Платформа FreeRADIUS у цій топології діє як суворий контролер доступу, операційні алгоритми якого спираються на делеговану довіру [6, 11]. Під час отримання запиту сервер не генерує нових підписів. Натомість він здійснює складну математичну експертизу наданого клієнтом ланцюжка довіри, динамічно звіряючи його з локально встановленим сертифікатом Root CA та актуальним списком відкликання CRL.

Практична реалізація цієї перевірки та узгодження захищених параметрів розкривається під час процедури ініціалізації сесії. В інженерній документації ця транзакція класифікується як TLS Handshake, а деталізований алгоритм перехресного обміну повідомленнями передбачає кілька етапів. Усі математичні розрахунки відбуваються суворо всередині ізольованого тунелю, де транзитний автентифікатор MikroTik виконує функцію абсолютно прозорого ретранслятора [19].

Стадія ініціалізації розпочинається з генерації первинної ентропії. Фонова служба-суплікант компілює стартовий масив даних ClientHello. Усередині повідомлення фіксується номер підтримуваної версії протоколу TLS, формується перелік доступних математичних перетворень (Cipher Suites) та генерується блок псевдовипадкових чисел. Платформа FreeRADIUS приймає контейнер, аналізує запропоновані методи шифрування, обирає максимально стійкий алгоритм і повертає пакет ServerHello із зустрічною порцією випадкового шуму [19].

Наступна стадія передбачає перехресний обмін електронними документами. Сервер інтегрує свій сертифікат формату X.509 у блок Certificate і транслює його в локальну магістраль. Одразу після цього формується командний

пакет CertificateRequest – так центральний вузол вимагає від обладнання надати зустрічний цифровий паспорт. Завершення передачі серверних фрагментів фіксується службовим сигналом ServerHelloDone.

Операційне ядро кінцевого пристрою отримує документ, і на основі отриманих даних запускається багатоетапна процедура валідації. Насамперед перевіряються часові мітки строку дії, після чого алгоритм звертається до інстальованого раніше еталона Root CA. Здійснюється перевірка криптографічного штампа. Математичний збіг контрольних сум доводить легітимність сервера, гарантуючи клієнту відсутність перехоплення [16]. Тепер суплікант зобов'язаний верифікувати власну автентичність. Формується зворотний пакет Certificate, куди вбудовується індивідуальний електронний паспорт датчика. Проте простої відправки відкритого файлу критично недостатньо: сторонній аналізатор трафіку здатний скопіювати публічний сертифікат і здійснити спробу повторного відтворення сесії. Справжній власник апаратури повинен математично довести факт володіння секретним приватним ключем [19].

Для розв'язання цієї проблеми застосовується командний блок CertificateVerify. Клієнтська утиліта збирає дамп усіх попередніх транзакцій поточного сеансу, пропускає його через хеш-функцію, а отриманий зліпок зашифровується секретним приватним ключем. Створюється еталонний математичний підпис [15]. Одночасно генерується масив ClientKeyExchange, у який завантажується первинний криптографічний матеріал Pre-Master Secret, жорстко заблокований публічним ключем сервера. Розшифрувати цей контейнер здатна виключно оригінальна серверна платформа [12].

Сервер FreeRADIUS приймає цей потік інформації. Валідація клієнтського сертифіката реалізується через звірку з підписом Root CA та списками відкликаних сертифікатів (CRL). Якщо файл має статус дійсного, сервер витягує публічний ключ датчика і намагається розшифрувати вміст блоку CertificateVerify. Успішна розшифровка стає доказом автентичності, а втручання

стороннього комп'ютера повністю виключається. Фаза взаємної перевірки успішно закривається [15].

Фінальний етап вимагає створення симетричних робочих ключів для захисту трафіку. Сервер застосовує власний секретний ключ для розпакування матеріалу Pre-Master Secret. Тепер обидві сторони комунікації володіють ідентичним набором цифрового шуму. Використовуючи погоджені раніше математичні формули, вони розраховують головний ключ сесії (Master Secret), який перетворюється на базу для виведення специфічного ключа PMK (Pairwise Master Key). Саме ключ PMK захищатиме всі подальші користувацькі пакети від спроб перехоплення [19].

Завершення стадії рукостискання маркується пакетами ChangeCipherSpec. Кожен мережевий вузол підтверджує готовність шифрувати наступні кадри згенерованим сесійним ключем. Останнім транслюється контрольний масив Finished. Успішна розшифровка цього пакета гарантує ідеальну синхронізацію криптографічних станів обох пристроїв (рис. 2.4).

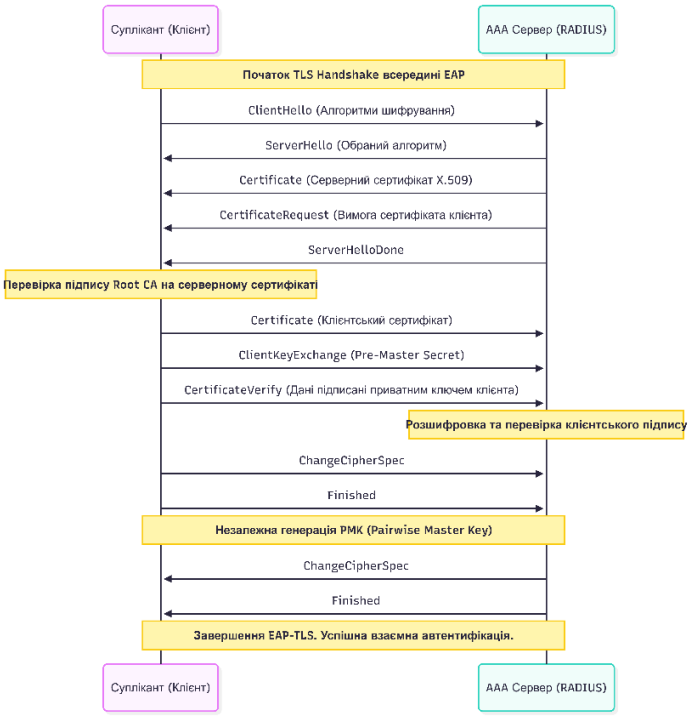

Рисунок 2.4 – Процес взаємної автентифікації та обміну ключами EAP-TLS

Успішне закриття процедури TLS Handshake ініціює відправку фінального магістрального пакета RADIUS Access-Accept, після чого транзитний автентифікатор деактивує апаратне блокування порту.

2.3 Альтернативні механізми доступу MAC Authentication Bypass (MAB)

Повноцінне розгортання архітектури EAP-TLS формує еталонний контур мережевої безпеки. Проте реальні корпоративні інфраструктури відзначаються надзвичайно високим ступенем апаратної гетерогенності. Значна частина парку обладнання позбавлена можливості обробляти складні криптографічні сертифікати. До цієї категорії належить застаріла медична апаратура, складські сканери штрих-кодів, промислові логічні контролери попередніх поколінь та прості мережеві принтери. Інтеграція фонового процесу супліканта в такі пристрої фізично неможлива через закритість прошивок або тотальний дефіцит оперативної пам'яті. Неможливість застосування протоколу 802.1x вимагає впровадження альтернативних механізмів ідентифікації для забезпечення безперервності бізнес-процесів підприємства.

Технологія MAC Authentication Bypass (MAB) розроблялася індустрією як компромісне транзитне рішення. Цей інструмент дозволяє магістральному комутатору самостійно ініціювати процедуру перевірки від імені підключеного «сліпого» клієнта. Взаємодія базується на використанні унікального апаратного ідентифікатора мережевої карти. Заводська MAC-адреса застосовується як замінник традиційного цифрового підпису або парольної фрази. Транзитний автентифікатор бере на себе функцію проксі-сервера автентифікації [20].

Сценарій активації механізму MAB запускається виключно в умовах відсутності відповіді на стартові криптографічні запити. У момент фізичного підключення кабелю комутатор діє за стандартним алгоритмом 802.1x. Порт блокується. У лінію відправляється стандартний керівний кадр EAP-

Request/Identity. Якщо кінцевий вузол не підтримує криптографічні стандарти, він апаратно ігнорує цей сигнал. Комутатор фіксує відсутність відповіді. Здійснюється серія повторних запитів. Лише після вичерпання ліміту тайм-аутів (EAP-Timeout) транзитний автентифікатор змінює алгоритм дій. Система розуміє, що на іншому кінці лінії знаходиться некерований пристрій [20].

Далі активується режим пасивного прослуховування магістралі. Комутатор залишає порт у заблокованому стані для користувацького трафіку, проте починає очікувати надходження будь-якого інформаційного кадру від невідомого обладнання. Зазвичай таким тригерним сигналом стає широкомовний запит протоколу ARP або стартовий пакет конфігурації DHCP Discover. Пристрій намагається знайти шлюз, і комутатор миттєво перехоплює цей перший пакет.

Фіксація інформаційного кадру запускає процес апаратної компіляції радіус-пакета. Комутатор витягує з отриманого кадру вихідну MAC-адресу невідомого пристрою. Далі відбувається специфічна інкапсуляція даних. Цей апаратний ідентифікатор одночасно копіюється у два базові поля службового повідомлення RADIUS Access-Request: адреса записується в атрибут User-Name і дублюється в атрибут User-Password. Додатково в пакет інтегрується спеціальний ідентифікатор Service-Type зі значенням Call-Check. Цей маркер сигналізує центральній базі даних про нестандартний формат запиту [11].

Залежно від налаштувань виробника комутаційного обладнання, синтаксичний формат запису адреси підлягає модифікації. Розділювачі у вигляді двокрапок чи дефісів часто видаляються системою, і увесь масив перетворюється на суцільний шістнадцятковий рядок. Передача цього рядка до сервера здійснюється з використанням базового протоколу PAP (Password Authentication Protocol). Сформований транзитний контейнер відправляється магістральними каналами до центрального сервера FreeRADIUS, а покроковий алгоритм обробки підключення за цим сценарієм відображено на рисунку 2.5.

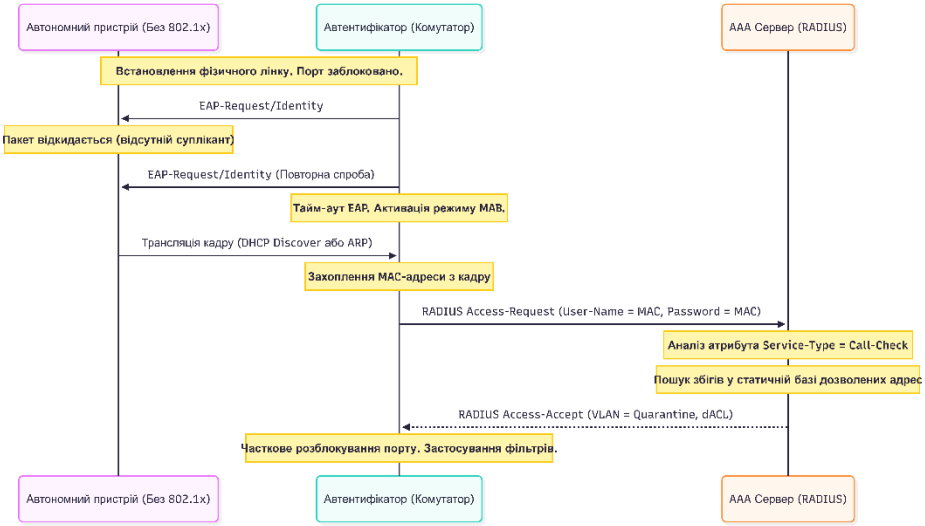

Рисунок 2.5 – Алгоритм обробки підключення за механізмом MAB

Сервер бази даних приймає запит. Модуль обробки аналізує маркер Call-Check, і система розпізнає спробу доступу за механізмом MAB. Здійснюється звернення до локальних статичних файлів або SQL-реєстрів, де виконується пошук переданої апаратної адреси у списках дозволеного обладнання. Збіг параметрів ініціює формування дозвільного пакета RADIUS Access-Accept. Транзитний автентифікатор отримує команду на розблокування інтерфейсу [11].

Варто категорично розмежовувати поняття криптографічної ідентифікації та сурогатного механізму MAB. Технологія обходу за MAC-адресою не забезпечує жодного рівня шифрування. Передача апаратних ідентифікаторів у локальному ефірі завжди здійснюється у вигляді відкритого тексту. Відсутність криптографічних примітивів та математичних сертифікатів перетворює MAB на найвразливішу ланку корпоративної безпеки. Цей протокол гарантує лише ідентифікацію, але жодним чином не підтверджує справжність вузла.

Слабкість архітектури формує ідеальні умови для хакерських втручань. Зловмиснику достатньо підключити пасивний аналізатор мережевого трафіку до будь-якого сегмента інфраструктури. Перехоплення легітимної адреси робочого принтера займає секунди. Подальша підміна власної адреси на хакерському

адаптері (вектор атаки MAC Spoofing) миттєво руйнує бар'єр. Комутатор захоплює підроблений кадр, відправляє радіус-запит і отримує схвалення від центрального сервера. База даних елементарно обманюється через нездатність перевірити дійсне джерело пакета [6].

Критична мінімізація таких ризиків вимагає неухильного застосування концепції ешелонованого захисту. Регламенти забороняють застосовувати механізм MAB як самостійний інструмент корпоративної безпеки. Платформа FreeRADIUS конфігурується за жорсткими правилами: пристрої, авторизовані виключно за апаратною адресою, ніколи не отримують доступу до центральних сегментів магістралі. Сервер генерує дозвільний сигнал, проте супроводжує його максимальним набором обмежувальних інструкцій.

Для порту транзитного комутатора динамічно призначається спеціальна ізольована віртуальна мережа (Restricted VLAN). Повноцінна маршрутизація з цієї мережі блокується на рівні ядра. Додатково застосовуються динамічні списки контролю доступу (Downloadable ACL), які фільтрують трафік на рівні транспортних протоколів. Наприклад, IP-камері дозволяється комунікувати виключно з сервером відеофіксації по порту 554, а промисловому датчику відкривається доступ лише до бази телеметрії. Будь-які інші мережеві запити жорстко відкидаються процесором комутатора [20].

Сучасні архітектури посилюють ефективність MAB технологією глибокого профілювання обладнання (Device Profiling). Центральний сервер не обмежується сухою звіркою апаратних адрес. Спеціалізовані аналітичні модулі пасивно акумулюють метадані з мережевого потоку: здійснюється аналіз специфічних опцій протоколу DHCP, скануються відкриті TCP-порти та вивчаються заголовки пакетів HTTP. Так формується цифровий відбиток пристрою [9].

Система безперервно порівнює поведінку підключеного вузла з його цифровим відбитком. Якщо пристрій із зафіксованою адресою медичного сканера раптово починає генерувати трафік до зовнішніх DNS-серверів або

сканувати сусідні підмережі за протоколом ICMP, аналізатор фіксує критичну аномалію. RADIUS-сервер негайно формує спеціальний керівний пакет відкликання авторизації (Change of Authorization, CoA). Ця команда відправляється транзитному комутатору, і для порту миттєво активується стан повного апаратного блокування. Подальший розвиток хакерської атаки зупиняється .

Інтеграція механізму MAB у лабораторне та промислове середовище потребує специфічного налаштування транзитного обладнання. Мережеві пристрої сімейства MikroTik апаратно підтримують цю технологію в межах конфігурації модуля Dot1x, де функція активується параметром MAC Authentication. Паралельне налаштування сервера FreeRADIUS вимагає редагування конфігураційних блоків та підключення статичних файлів авторизації для збереження еталонних адрес. Симбіоз криптографічного протоколу EAP-TLS для захисту сучасних систем та жорстко обмеженого механізму MAB для підтримки застарілої техніки формує комплексну гібридну модель безпеки. Глибоке практичне моделювання такої гетерогенної архітектури на базі віртуалізованого стенда розкриває повний потенціал сучасних систем динамічної сегментації та контролю доступу [12, 23].

2.4 Життєвий цикл цифрових сертифікатів та управління інфраструктурою PKI

Надійне функціонування будь-якої інфраструктури PKI неможливе без суворого контролю за життєвим циклом її криптографічних матеріалів. Якщо в класичних офісних мережах кількість кінцевих точок залишається відносно прогнозованою, то сегмент Інтернету речей відрізняється постійною динамікою та експоненційним зростанням. Тут цифровий сертифікат виступає не просто статичним файлом, а тимчасовим мандатом, який потребує безперервного нагляду. Будь-які збої в цьому ланцюжку управління неминуче призводять або

до масового блокування легітимної техніки, або до появи критичних дір у безпеці. Загалом, ефективний менеджмент PKI для автономного обладнання розбудовується навколо чотирьох ключових стадій: генерації, розгортання, планової ротації та оперативного відкликання.

Початкова стадія – емісія (або генерація) – завжди відбувається в суворо ізольованому середовищі центру сертифікації (CA). Саме тут формуються асиметричні пари ключів [17]. Практика показує, що для IoT-проєктів доцільніше застосовувати алгоритми еліптичної криптографії (ECC, зокрема криву secp256r1) замість класичного стандарту RSA. Такий вибір дозволяє суттєво зекономити обчислювальні ресурси мікроконтролерів, зберігаючи при цьому еталонний рівень криптостійкості. Під час випуску сертифіката (формат X.509v3) до нього інтегруються специфічні розширення. Наприклад, поле Subject Alternative Name (SAN) жорстко прив'язує документ до MAC-адреси конкретного датчика, а Extended Key Usage (EKU) обмежує його використання виключно клієнтською автентифікацією. Вкрай важливим нюансом для автономної техніки є генерація ключів без парольного захисту закритого ключа (формат nopass). Оскільки розумні реле чи камери позбавлені інтерфейсів вводу, вони мають самостійно зчитувати свій паспорт із пам'яті під час завантаження, щоб ініціювати сесію 802.1X без втручання оператора.

Одразу після створення ідентифікаторів виникає проблема їх безпечного транспортування на периферійне обладнання (етап розгортання). З інженерної точки зору найбільш надійним варіантом є фабричне програмування (Factory Provisioning). У цьому сценарії первинні ключі записуються в апаратні модулі довіри (TPM) ще на конвеєрі виробника, що фізично унеможливлює їх подальше вилучення навіть за умови демонтажу пристрою. Якщо ж ідентифікатор потрібно видати вже в умовах діючого підприємства, застосовуються криптографічні тунелі (SFTP/SCP) або ж концепція «нульового дотику» (Zero-Touch Provisioning). Остання реалізується завдяки протоколам SCEP або EST (RFC 7030) [21, 22]. Алгоритм їх роботи передбачає, що новий сенсор підключається

до спеціальної карантинної мережі (Provisioning VLAN), самостійно запитує свій сертифікат через захищений канал, а після його успішного збереження автоматично переводиться комутатором до робочого сегмента магістралі.

Третьою обов'язковою складовою є регулярна ротація криптографічних матеріалів. Тривале використання одних і тих самих ключів дає зловмисникам простір для накопичення перехопленого трафіку та його подальшого криптоаналізу. Щоб запобігти цьому, термін дії сертифікатів для кінцевих IoT-вузлів зазвичай обмежують рамками від 90 до 180 діб [18]. Процес ротації зводиться до автоматичної генерації нових ключів ще до того, як закінчиться термін дії поточних (як правило, за кілька тижнів до експірації). Тут інженери обов'язково застосовують принцип перекриття термінів дії (overlapping validity periods). Новий цифровий мандат набуває чинності паралельно зі старим, що гарантує безперервність роботи обладнання навіть у випадку тимчасових проблем із синхронізацією часу за протоколом NTP на підприємстві.

Фінальною ланкою життєвого циклу, яка відповідає за реагування на інциденти, є процедура відкликання. Враховуючи те, що вуличні камери чи датчики часто монтуються у фізично вразливих місцях, загроза їх крадіжки залишається стабільно високою. Щойно виявляється зникнення або злам техніки, серійний номер відповідного сертифіката негайно вноситься до реєстру недійсних ключів. На базі цього запису центр довіри компілює актуальний список відкликання (Certificate Revocation List, CRL) [16], який експортується на сервер FreeRADIUS. Відтепер будь-яка спроба підключення викраденого пристрою завершуватиметься невдачею: RADIUS-сервер виявить збіг із чорним списком, надішле команду Access-Reject, і транзитний комутатор миттєво заблокує порт. Варто зазначити, що у великомасштабних інфраструктурах постійне завантаження масивних файлів CRL створює надмірне навантаження на пам'ять сервера. Для оптимізації процесу перевірки замість локальних списків як альтернатива впроваджується протокол OCSP [23], який дає змогу точково перевіряти статус конкретного ключа в режимі реального часу.

2.5 Постановка завдання на проєктування системи керування доступом

На основі проведеного теоретичного аналізу вразливостей IoT-мереж та дослідження механізмів стандарту IEEE 802.1X, головним завданням практичної частини роботи є проєктування, розгортання та тестування комплексної системи контролю мережевого доступу (NAC).

Система повинна автоматизувати процес фільтрації підключень на рівні портів комутатора та забезпечити реалізацію концепції «нульової довіри» (Zero Trust). Для досягнення цієї мети поставлено такі технічні завдання:

– спроєктувати топологію безпеки для гетерогенного середовища, що складається з сучасних IoT-пристроїв та застарілого (legacy) обладнання;

– розгорнути відокремлену інфраструктуру відкритих ключів (PKI) для генерації та управління життєвим циклом сертифікатів;

– налаштувати сервер автентифікації для обробки запитів за протоколом EAP-TLS та резервним механізмом MAB;

– реалізувати логіку динамічної мікросегментації (призначення VLAN) на основі RADIUS-атрибутів;

– перевірити стійкість архітектури до типових атак канального рівня (зокрема, MAC Spoofing).

Оскільки розгортання такої інфраструктури на базі комерційних Enterprise-рішень потребує значних фінансових витрат на ліцензування, стратегічним рішенням є побудова системи виключно на базі відкритого програмного забезпечення (Open Source).

Для забезпечення максимальної достовірності результатів тестування було прийнято рішення відмовитися від базових пакетних симуляторів (на кшталт Cisco Packet Tracer) на користь середовища глибокої мережевої емуляції, яке дозволяє запускати повноцінні образи операційних систем. Аргументацію та

порівняльний аналіз обраного інструментарію для побудови віртуального стенда наведено в таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору інструментарію для проєктування системи

Компонент архітектури	**Обране рішення**	**Альтернативні варіанти**	**Аргументація вибору**
Середовище моделювання	GNS3	Cisco Packet Tracer, EVE-NG	Підтримка реальних образів ОС (Linux, RouterOS) через гіпервізор. Точна імітація мережевого стека.
Сервер автентифікації (AAA)	FreeRADIUS	Cisco ISE, PacketFence	Відкритий вихідний код, безкоштовність, повна підтримка складних методів (EAP-TLS, MAB).
Центр сертифікації (CA)	easy-rsa	OpenSSL (чистий), Microsoft CA	Спрощене та автоматизоване управління PKI-інфраструктурою. Ідеально підходить для Linux.
Мережевий автентифікатор	MikroTik RouterOS (CHR)	Cisco IOS (vIOS), Open vSwitch	Нативна підтримка 802.1X, MAB та динамічного VLAN Filtering. Низьке споживання ресурсів віртуальної машини.
Клієнт (Суплікант)	Lubuntu + wpa_supplicant	Windows 10, Raspberry Pi OS	Можливість запуску в «headless» режимі. Мінімальні вимоги до оперативної пам'яті (512 МБ).

Обраний стек технологій дозволяє створити повноцінний цифровий двійник корпоративного сегмента мережі. Усі налаштування, протестовані в даному середовищі, є повністю сумісними з реальним фізичним обладнанням.

Висновки до розділу 2

У другому розділі проведено детальний аналіз архітектури стандарту IEEE 802.1X та механізмів його адаптації для захисту IoT-інфраструктури. Встановлено, що концепція логічного блокування портів забезпечує надійний захист від несанкціонованого фізичного доступу.

Визначено, що для автономних пристроїв найбільш захищеним методом є EAP-TLS, який гарантує взаємну криптографічну автентифікацію на основі

цифрових сертифікатів X.509. Обґрунтовано необхідність архітектурного розмежування функцій AAA-сервера (FreeRADIUS) та центру сертифікації (easy-rsa).

Для забезпечення інтеграції застарілого обладнання досліджено механізм MAC Authentication Bypass (MAB). Доведено, що через фундаментальну вразливість до атак типу MAC Spoofing, такі клієнти повинні примусово ізолюватися в карантинних віртуальних мережах.

Підсумком теоретичного дослідження стала формалізація постановки завдання на проєктування системи керування доступом. Сформовано технічні вимоги до архітектури та здійснено порівняльний аналіз програмного інструментарію. Як оптимальне рішення для побудови моделі «Нульової довіри» обрано комплекс відкритого програмного забезпечення (Open Source): AAA-сервер FreeRADIUS, платформу MikroTik RouterOS та середовище глибокої мережевої емуляції GNS3. Це формує надійний технологічний фундамент для практичної реалізації системи у наступному розділі.

3 ПРОЕКТУВАННЯ СИСТЕМИ КЕРУВАННЯ ДОСТУПОМ ДО МЕРЕЖІ

3.1 Розробка структурної та логічної схеми мережі

Стрімке збільшення кількості різнорідних обчислювальних вузлів, що розгортаються на базі концепції Інтернету речей (IoT), вимагає від інженерів повної трансформації звичних методів захисту корпоративних мереж. На сьогодні класичний фізичний периметр безпеки практично перестав існувати, адже сучасні офісні та промислові локації насичені смарт-контролерами, IP-камерами та різноманітними сенсорами, які часто монтуються у відкритих для сторонніх осіб зонах. Аналіз типових архітектур доступу свідчить про те, що базові технології лінійного рівня (наприклад, Ethernet або пасивні оптичні мережі PON), хоч і гарантують високу швидкість передачі даних, залишаються вкрай вразливими до цілеспрямованих атак. Традиційні засоби ізоляції здебільшого розділяють середовище між користувачами, але не здатні провести глибоку верифікацію самого пристрою до моменту його повноцінного підключення до системи.

Враховуючи вищезазначені недоліки, формування системи контролю мережевого доступу (Network Access Control, NAC) на рівні логічної ізоляції стає критичною необхідністю. Основою для розробки такої захисної моделі слугує стандарт IEEE 802.1X, імплементація якого дозволяє підприємству перейти до парадигми ідентифікаційно-орієнтованого доступу (Identity-Based Access). У цій конфігурації комутатор не надає мережевих привілеїв на основі номера порту чи статичної IP-адреси; доступ відкривається виключно після того, як пристрій успішно доведе свою легітимність за допомогою криптографічних алгоритмів.

З метою практичної реалізації надійної та економічно ефективної топології в межах даного бакалаврського дослідження, було спроєктовано структурно-логічну схему на базі спеціалізованого середовища емуляції GNS3. Рішення використати цей інструментарій, а не звичайні симулятори пакетів,

обґрунтовується потребою в запуску повноцінних операційних систем (через механізми гіпервізора). Це дає змогу максимально точно відтворити реальну роботу мережевого стека, часові затримки та процеси криптографічного узгодження. Запропонована архітектура спирається на трирівневу модель безпеки, де логічні функції розподілені між наступними компонентами віртуалізованого стенда:

1. Сервер автентифікації. Цей елемент розгорнуто в ізольованому сегменті на базі відкритої ОС Ubuntu Server із закріпленою статичною IP-адресою 192.168.10.10. Обробку вхідних RADIUS-запитів, ведення журналів логування та реалізацію безпосередніх процедур перевірки доручено серверному демону FreeRADIUS. Його архітектура ідеально підходить для гетерогенних середовищ завдяки нативній підтримці складних EAP-методів (зокрема EAP-TLS). Окремо варто зазначити, що створення та керування життєвим циклом цифрових сертифікатів (Root CA) винесено в незалежний логічний домен за допомогою утиліти easy-rsa, що дозволило чітко розмежувати процеси емісії ключів від щоденного контролю доступу;

2. Автентифікатор. Транзитним вузлом, що імітує функціонал корпоративного комутатора, виступає віртуальна машина з операційною системою MikroTik RouterOS (редакція Cloud Hosted Router), якій призначено IP-адресу 192.168.10.1. Для формування необхідної L2-зв'язності фізичні інтерфейси пристрою були об'єднані у програмний міст (bridge) під назвою br-iot. У цій конфігурації магістральний інтерфейс ether1 виконує функцію транку для безпечної передачі RADIUS-пакетів до сервера. Водночас клієнтський порт ether2 переведено в режим жорсткого контролю доступу (Port Access Entity) згідно з політиками 802.1X;

3. Клієнт автентифікації.Кінцевий IoT-вузол моделюється за допомогою операційної системи Lubuntu. Щоб імітувати поведінку автономного обладнання, позбавленого засобів ручного введення (headless-формат), генерацію повідомлень EAPOL та ініціалізацію сесії на дротовому інтерфейсі

enp0s3 забезпечує системна служба wpa_supplicant. Вона функціонує у фоновому режимі та використовує для криптографічного рукостискання заздалегідь імпортовані сертифікати пристрою.

Такий розподіл обчислювальних потужностей та інтерфейсів формує стійкий бар'єр, який зупиняє будь-який абонентський трафік ще до початку перевірки ідентифікаторів. Деталізована логіка взаємодії цих вузлів проілюстрована на розробленій структурно-логічній схемі на рисунку 3.1.

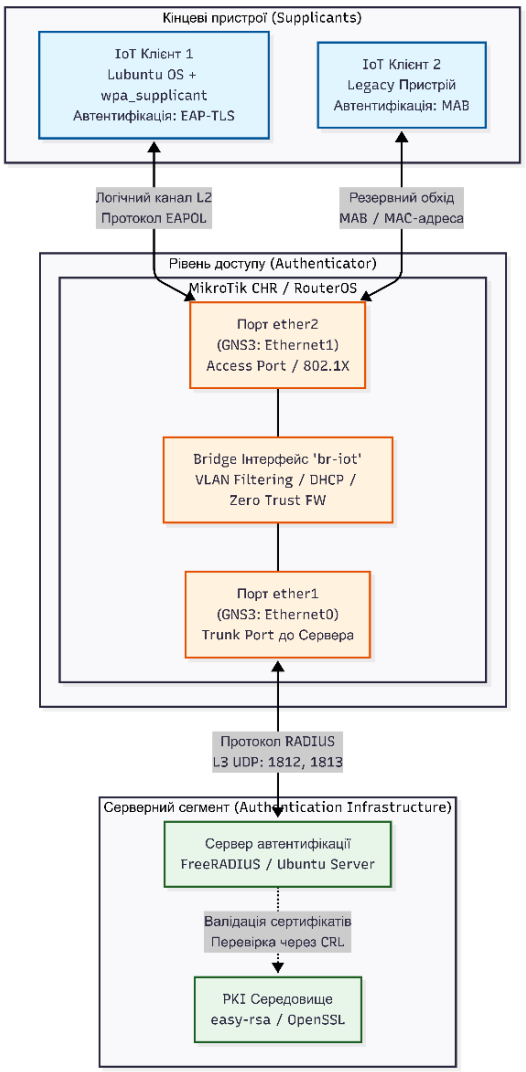

Рисунок 3.1 – Структурно-логічна схема системи керування доступом для IoT-інфраструктури у середовищі GNS3

Описана топологія виступає міцним фундаментом для впровадження архітектури нульової довіри (Zero Trust Network Access). Відповідно до її концепції, доки транзитний автентифікатор не отримає підтвердження надійності пристрою через захищений канал, будь-які спроби передачі користувацьких даних через порт ether2 (в топології GNS3 він маркується як Ethernet1) будуть блокуватися на апаратному рівні. На цій стадії комутатор здатний обробляти виключно інкапсульовані кадри EAPOL. Паралельно зберігається можливість пасивного перехоплення MAC-адреси для активації резервного механізму MAB (MAC Authentication Bypass), якщо до порту підключено обладнання застарілого типу, яке не підтримує роботу з сертифікатами.

Остаточне відкриття логічного порту відбувається лише після того, як AAA-сервер завершить математичну валідацію сертифіката та надішле комутатору дозвільний сигнал (Access-Accept). Разом із цим сигналом сервер передає специфічні атрибути, що активують механізм динамічного сегментування (VLAN Filtering). Таким чином, після проходження перевірки IoT-вузол не отримує доступ до загальної корпоративної мережі, а ізолюється у вузькоспрямованому мікросегменті. Це гарантує, що навіть у разі гіпотетичної компрометації одного датчика, жорсткі правила міжмережевого екранування не дозволять зловмиснику масштабувати атаку та дістатися до критичного ядра системи.

3.2 Розробка алгоритму автентифікації IoT-пристроїв

Ефективність функціонування спроєктованої системи контролю мережевого доступу (NAC) безпосередньо залежить від коректності логіки криптографічного обміну між кінцевим обладнанням та сервером автентифікації. Зважаючи на архітектурну специфіку інфраструктури Інтернету речей (IoT), яка характеризується відсутністю інтерфейсів для інтерактивного

введення облікових даних користувачем (так звані headless-пристрої), використання класичних парольних протоколів (наприклад, EAP-PEAP або EAP-TTLS) визнано недоцільним. Як базовий механізм трансляції даних та верифікації обрано протокол EAP-TLS (Extensible Authentication Protocol - Transport Layer Security) [15]. Цей метод забезпечує найвищий рівень захисту в 802.1X середовищах завдяки обов'язковому виконанню взаємної автентифікації (Mutual Authentication) на основі цифрових сертифікатів стандарту X.509. Такий підхід нівелює ризики атак типу Man-in-the-Middle (MitM), підміни ідентифікаторів (MAC Spoofing) та несанкціонованого підключення стороннього обладнання до корпоративного периметра.

Функціонування розробленого алгоритму базується на роботі кінцевих автоматів (State Machines) автентифікатора портів (Port Access Entity, PAE). У спроєктованому стенді на базі MikroTik CHR кожен порт, що входить до складу програмного мосту br-iot, логічно розділяється на два віртуальні канали: неконтрольований (Uncontrolled Port) та контрольований (Controlled Port). До моменту успішного проходження EAP-транзакції контрольований порт перебуває у стані жорсткого блокування (Unauthorized), що унеможливлює передачу будь-якого прикладного трафіку, включно із запитами DHCP, DNS або ARP. Обмін службовими повідомленнями EAP здійснюється виключно через неконтрольований порт, який на рівні ядра мережевої ОС налаштовано на пропуск лише кадрів типу EAPOL (EAP over LAN).

Особливу увагу під час розробки алгоритму було приділено показнику відмовостійкості кінцевого вузла. В умовах реальної експлуатації промислової інфраструктури датчики часто зазнають раптових перебоїв з електроживленням або тимчасово втрачають фізичний лінк через апаратні мікрозбої. Саме тому процес ініціалізації EAP-сесії не може залежати від ручного втручання адміністратора. У межах розробленого стенда запуск клієнтської утиліти wpa_supplicant глибоко інтегровано на рівень підсистеми ініціалізації ОС Linux (systemd) через спеціально спроєктований юніт iot-auth.service. Ця служба

налаштована з параметрами агресивного перезапуску (Restart=always) та прив'язкою до підняття мережевого стека (After=network.target). Такий інженерний підхід гарантує, що одразу після подачі живлення або відновлення L2-зв'язності пристрій миттєво та автономно розпочне процедуру доведення своєї легітимності без потреби у термінальних сесіях.

Для формалізації процесу взаємодії розроблено покроковий алгоритм автентифікації, який описує хронологію передачі службових пакетів від моменту встановлення фізичного з'єднання до отримання логічного доступу. Деталізовану послідовність обміну повідомленнями у змодельованому середовищі наведено на рисунку 3.2.

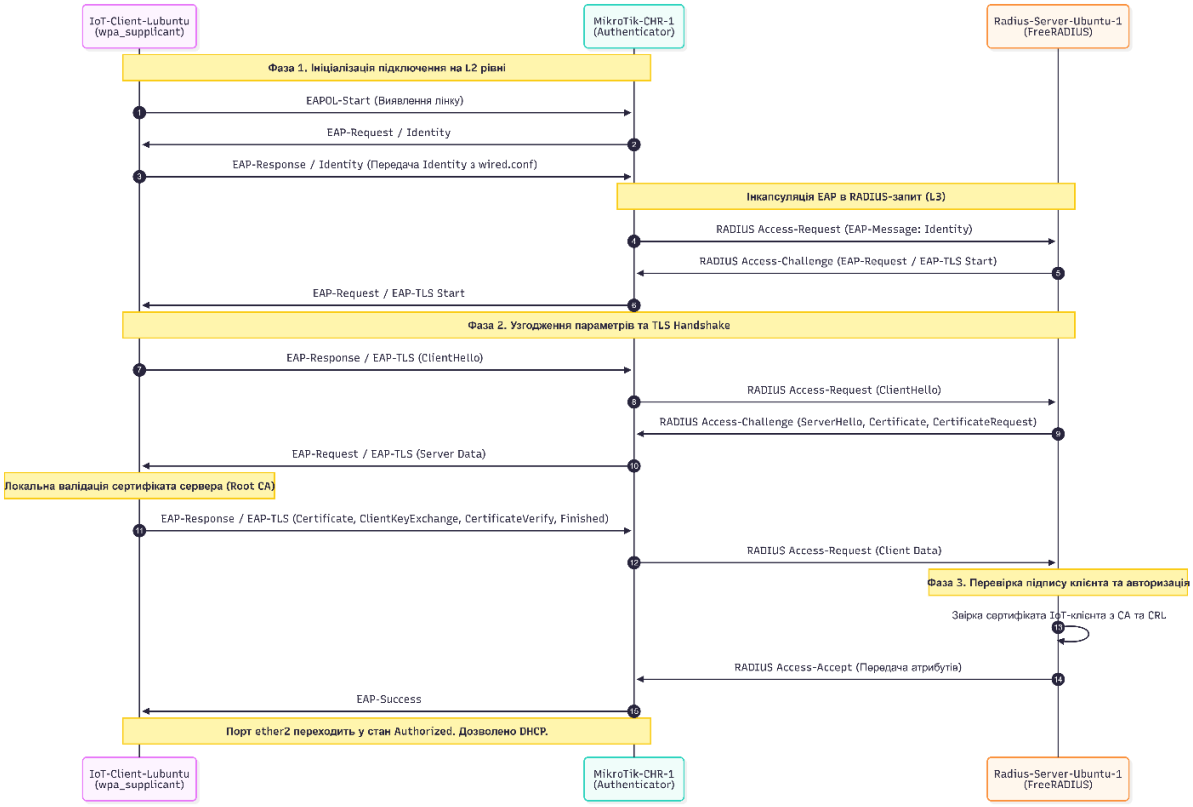

Рисунок 3.2 – Діаграма послідовності автентифікації IoT-пристрою за алгоритмом EAP-TLS

Практична перевірка розробленого алгоритму EAP-TLS здійснювалася за допомогою перехоплення трафіку в Wireshark на інтерфейсі між Lubuntu та MikroTik CHR. Аналіз дампів підтверджує строгу відповідність транзакцій протокольним стандартам:

1. Суплікант звертається до порту ether2 автентифікатора, використовуючи відкритий ідентифікатор identity=«iot-client» з конфігураційного файлу wired.conf.

2. Коли суплікант генерує первинний кадр, комутатор MikroTik виступає інтелектуальним посередником (NAS). Він вилучає корисне навантаження EAP з Ethernet-кадру та інкапсулює його в UDP-дейтаграму RADIUS Access-Request. Для забезпечення криптографічної цілісності цього транзитного каналу зв'язку між комутатором та сервером 192.168.10.10 застосовується симетричний ключ iot_secret_2026. Це дозволяє серверу відкидати будь-які запити, що надходять від неавторизованого транзитного обладнання (захист від Rogue NAS).

3. Згідно з перехопленим дампом трафіку на рисунку 3.3, сервер приймає запит та ініціює створення TLS-тунелю, надсилаючи свій публічний сертифікат radius-server.crt.

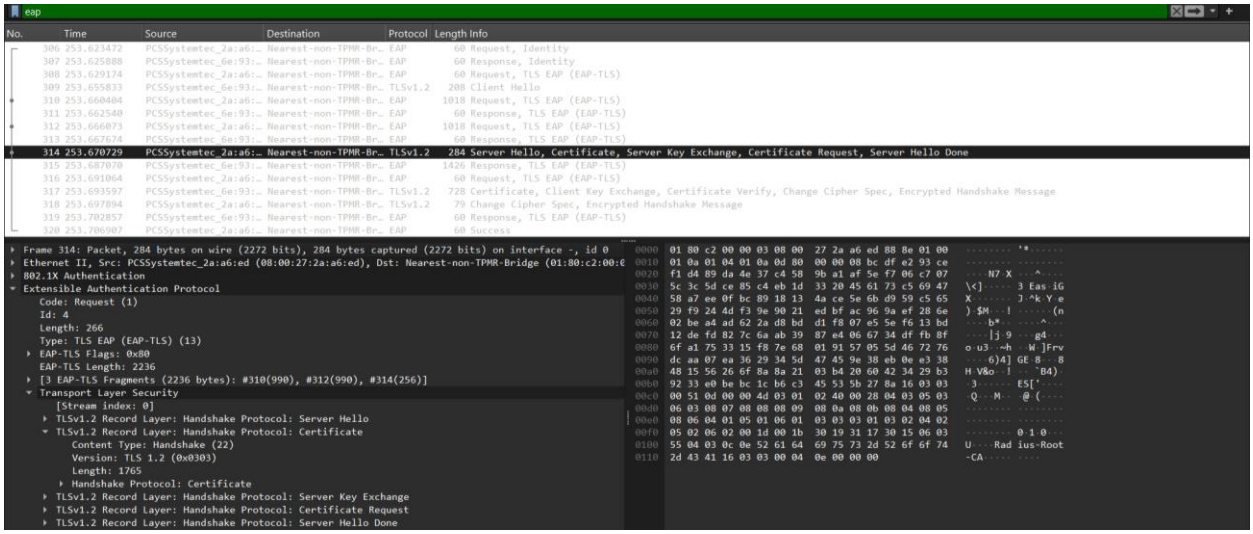

Рисунок 3.3 – Дамп мережевого трафіку процесу узгодження EAP-TLS у Wireshark

4. На цьому етапі клієнтська машина Lubuntu виконує критично важливу функцію захисту від атак типу «злий двійник» (Evil Twin). Вона математично звіряє підпис отриманого серверного сертифіката з локальним якорем довіри (Trust Anchor) – еталонним кореневим сертифікатом ca.crt, який безпечно розміщений у директорії /etc/wpa_supplicant/certs/. Тільки за умови повного підтвердження автентичності сервера, клієнт у відповідь надає свій цифровий паспорт iot-client.crt разом із криптографічним доказом володіння закритим ключем (iot-client.key).

5. Після успішної перевірки валідності клієнта, демон FreeRADIUS генерує пакет RADIUS Access-Accept. Отримавши його, MikroTik переводить порт у стан Authorized, відкриваючи шлях для передачі DHCP-запитів та корисного прикладного навантаження.

Попри високу надійність сертифікатної моделі, інфраструктура часто містить пристрої попередніх поколінь (legacy IoT), які апаратно не здатні підтримувати криптографію. Для їхньої інтеграції на порту MikroTik активовано гібридний режим MAB командою: /interface dot1x server set 0 auth-types=dot1x,mac-auth mac-auth-mode=mac-as-username-and-password retrans-timeout=2s. Якщо клієнт не відповідає на серію EAP-запитів протягом 6 секунд, комутатор використовує його MAC-адресу (наприклад, 00:11:22:33:44:55) як логін і пароль. Важливо зазначити, що алгоритм MAB є фундаментально вразливим до підміни ідентифікаторів (MAC Spoofing). З метою мінімізації цього ризику, архітектуру FreeRADIUS налаштовано таким чином, що пристрої без сертифікатів (де MAC-адреса збережена у файлі users як Cleartext-Password) жорстко маркуються як недовірені. Після їхньої авторизації MikroTik примусово призначає їм глибоко ізольований сегмент мережі (VLAN 30). Це унеможливлює компрометацію критичних серверних ресурсів навіть у випадку фізичної підміни датчика зловмисником, оскільки весь трафік з VLAN 30 жорстко фільтрується між мережевим екраном.

3.3 Політики мікросегментації та динамічне призначення VLAN

Забезпечення базової процедури автентифікації пристрою за стандартом IEEE 802.1X є необхідною, але недостатньою умовою для побудови комплексної архітектури безпеки корпоративного периметра. В умовах сучасних кіберзагроз класичний мережевий підхід, що базується на статичній прив'язці фізичних портів комутатора до визначених віртуальних локальних мереж (Static VLAN), повністю втрачає свою ефективність. Особливо критичним цей недолік є в гетерогенних інфраструктурах Інтернету речей (IoT), де топологія підключень може динамічно змінюватися, а самі пристрої часто розміщуються у фізично доступних, неконтрольованих зонах (наприклад, IP-камери на фасадах будівель, датчики в публічних холах). Якщо зловмисник відключить легітимний пристрій і під'єднає власне обладнання до статично налаштованого порту, він автоматично опиниться в довіреному сегменті мережі.

З огляду на ці ризики, в межах дипломного дослідження спроєктовано систему динамічної мікросегментації мережі, яка базується на методології архітектури нульової довіри (Zero Trust Architecture, ZTA) [25]. Згідно з цією концепцією, жоден порт комутатора за замовчуванням не належить до жодної довіреної мережі. Рішення про те, до якого ізольованого сегмента буде підключено кінцевий вузол, приймається централізовано виключно після успішного завершення процедури криптографічної перевірки.

Логіка роботи спроєктованого архітектурного механізму полягає в передачі спеціалізованих інструкцій від сервера контролю доступу (AAA) до мережевого автентифікатора (комутатора). Відповідно до стандарту RFC 2868, для реалізації такого управління використовуються стандартизовані RADIUS-атрибути [26]. Запропонована політика безпеки передбачає, що у разі успішної валідації сертифіката пристрою, сервер формує службовий пакет Access-Accept, до якого обов'язково включається набір тунельних атрибутів. Зокрема, атрибут Tunnel-Type вказує комутатору на необхідність використання технології VLAN

(значення 13), Tunnel-Medium-Type визначає середовище передачі (IEEE-802), а найважливіший атрибут Tunnel-Private-Group-Id містить ідентифікатор цільової мережі (наприклад, VLAN 20 для авторизованих IoT-датчиків або VLAN 30 для пристроїв з обмеженим доступом). Отримавши такі параметри, комутатор автоматично переконфігуровує порт і поміщає клієнта у відповідне ізольоване середовище. Алгоритм цієї взаємодії та етапність динамічного перепризначення порту відображено на діаграмі послідовності на рисунку 3.4.

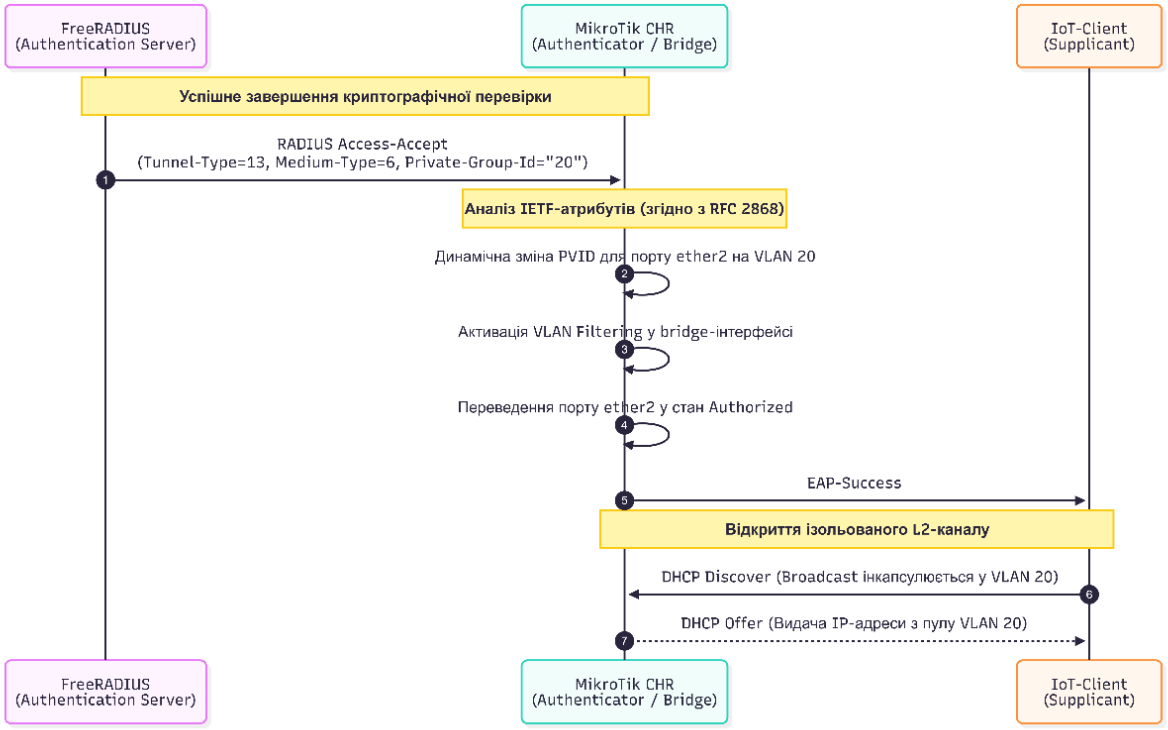

Рисунок 3.4 − Діаграма послідовності динамічного призначення VLAN на рівні автентифікатора

Архітектурна імплементація концепції Zero Trust у розробленій системі не обмежується лише розподілом на канальному рівні (L2), а підкріплюється глибокою ізоляцією на мережевому рівні (L3). Для унеможливлення доступу потенційно скомпрометованих IoT-пристроїв до ядра мережі та серверної інфраструктури, проєктом передбачено розгортання незалежних служб DHCP

для кожного мікросегмента. Додатково впроваджуються жорсткі політики міжмережевого екранування (Firewall Filter Rules) на транзитному комутаторі. Відповідно до цих проєктних правил, будь-який транзитний трафік від клієнтських підмереж безпосередньо до керуючих IP-адрес інфраструктури відкритого ключа (PKI) та RADIUS-сервера має безумовно відкидатися апаратно.

Головним результатом проєктування такого комплексного механізму є кардинальне зменшення площі потенційної атаки (Attack Surface). Навіть у випадку фізичної крадіжки пристрою або його програмної компрометації зловмисник опиниться в жорстко ізольованому середовищі без можливості ініціювати мережеві сканування, проводити атаки на відмову в обслуговуванні (DDoS) системних сервісів або отримати доступ до внутрішніх файлових ресурсів корпоративної мережі.

Висновки до розділу 3

У третьому розділі розроблено логічну та структурну архітектуру захищеної мережі, яка використовує механізми керування доступом стандарту IEEE 802.1X. Створена модель повністю відповідає концепції «Нульової довіри» (Zero Trust) та гарантує надійну сегментацію підключених пристроїв одночасно на канальному (L2) і мережевому (L3) рівнях.

Відповідно до висунутих технічних завдань здійснено вибір та розподіл ролей програмного забезпечення. Зокрема, функції обробки криптографічних запитів (AAA-сервер) покладено на платформу FreeRADIUS, роль проміжного мережевого автентифікатора виконує операційна система MikroTik CHR, а управління життєвим циклом сертифікатів винесено в окремий вузол інфраструктури відкритих ключів (PKI) на базі комплексу easy-rsa.

Запропоновано покроковий алгоритм ідентифікації обладнання IoT. Саму процедуру поділено на три базові етапи: старт логічного з'єднання через

повідомлення EAPOL, обмін параметрами шифрування з утворенням безпечного тунелю (TLS Handshake) та остаточне надання прав доступу після перевірки сервером клієнтського сертифіката.

Побудовано модель динамічної мікросегментації інфраструктури шляхом автоматичного призначення VLAN. Для цього відібрано необхідні RADIUS-атрибути (Tunnel-Type, Tunnel-Medium-Type, Tunnel-Private-Group-Id), завдяки яким комутатор здатний самостійно перемикати порт клієнта у виділену безпечну зону після проходження автентифікації.

Побудовані діаграми послідовності та схеми маршрутизації трафіку чітко описують принципи роботи мережевих компонентів. Вони слугують надійною інженерною основою для подальшого розгортання тестового стенда та емпіричної перевірки розробленої системи у наступному розділі.

4 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ РОЗРОБЛЕНОЇ СИСТЕМИ

4.1 Реалізація та тестування системи керування доступом

Перевірка дієздатності та ефективності спроєктованої архітектури контролю доступу вимагає створення повноцінного інфраструктурного стенда, здатного з високою точністю імітувати роботу гетерогенної мережі підприємства. Відповідно до обґрунтування, наведеного у підрозділі 2.5, для проведення практичних експериментів розгорнуто середовище глибокої мережевої симуляції GNS3 у зв'язці з гіпервізором VirtualBox. Формування такого закритого віртуального контуру дозволило повністю ізолювати комунікаційне середовище від непередбачуваного впливу зовнішніх процесів хостової операційної системи. Завдяки цій ізоляції досягається еталонна чистота експерименту, що є критично необхідною умовою для коректного перехоплення та подальшого аналізу цільових криптографічних пакетів.

Для забезпечення об'єктивності результатів моделювання конфігурація апаратних ресурсів віртуальних машин була свідомо адаптована до реальних індустріальних умов. Зокрема, клієнтській машині (Lubuntu), що імітує кінцевий IoT-сенсор, виділено мінімально можливий обсяг обчислювальних потужностей – 1 віртуальне ядро (vCPU) та 512 МБ оперативної пам'яті. Це дозволяє достовірно оцінити швидкість проходження важких криптографічних алгоритмів на апаратурі з обмеженими ресурсами. Водночас для серверної частини (Ubuntu Server) виділено 2 vCPU та 2 ГБ оперативної пам'яті, що забезпечує безперебійну роботу демонів автентифікації. Графічне відображення фізичних та логічних зв'язків між суплікантом, транзитним автентифікатором та AAA-сервером представлено на структурній схемі розгорнутої топології на рисунку 4.1.

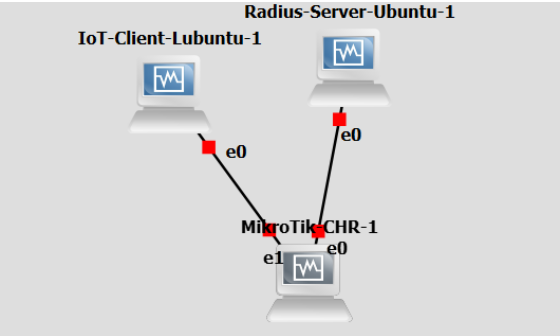

Рисунок 4.1 – Топологія віртуального тестового стенда в середовищі GNS3

Роль центрального ядра прийняття рішень (AAA-сервера) у розробленому стенді делеговано відкритій платформі FreeRADIUS версії 3.2.3. На цьому ж віртуальному сервері розгорнуто інфраструктуру відкритих ключів (PKI) за допомогою утиліти easy-rsa.

Окрему увагу під час конфігурації серверної частини було приділено стандартам криптографічного захисту. Згідно з актуальними вимогами Національного інституту стандартів і технологій США (NIST SP 800-52r2) щодо безпеки державних та корпоративних інформаційних систем, використання протоколів TLS версій, нижчих за 1.2, суворо заборонено у нових розгортаннях. Відповідно, для забезпечення максимального рівня захисту процесу обміну ключами, у конфігурації EAP-модуля FreeRADIUS було примусово зафіксовано використання криптографічного протоколу TLS версії 1.2 (як базової вимоги для зворотної сумісності з IoT-контролерами) та реалізовано підтримку сучасного стандарту TLS 1.3 для потужніших клієнтів.

Функцію мережевого автентифікатора виконує операційна система MikroTik RouterOS у редакції Cloud Hosted Router (CHR). Для емулювання поведінки класичного комутатора доступу робочі інтерфейси MikroTik CHR об'єднано в єдиний bridge-інтерфейс, на портах якого активовано протокол 802.1X. Такий підхід дозволяє застосовувати жорсткі політики автентифікації до

кожного фізичного підключення окремо і блокувати неавторизований трафік ще до моменту отримання пристроєм IP-адреси.

Найбільшим інженерним викликом під час розгортання стенда стала достовірна імітація кінцевого IoT-обладнання без екранів та клавіатур (headless-пристроїв). Для ініціалізації сесій автентифікації на вузлі Lubuntu застосовано системну утиліту wpa_supplicant, яка глибоко інтегрована в підсистему ініціалізації Linux (systemd). Її криптографічний рушій та логіка обробки X.509-сертифікатів є програмно сумісними з бібліотеками mbedTLS, які повсюдно використовуються на реальних фізичних мікроконтролерах індустріального класу (ESP32, STM32). Таким чином, розроблена модель взаємодії за протоколом EAP-TLS може бути безперешкодно перенесена на реальне мікроконтролерне обладнання.

Для верифікації криптографічних процесів та підтвердження безпеки архітектури до топології стенда інтегровано програмний аналізатор мережевого трафіку Wireshark. Його використання дозволяє здійснювати глибоку інспекцію пакетів (Deep Packet Inspection), що критично необхідно для наочної демонстрації процесу інкапсуляції кадрів EAPOL, аналізу фаз рукостискання TLS Handshake та підтвердження факту повної ізоляції парольних даних від передачі у відкритому вигляді.

4.2 Конфігурація компонентів системи

Практична реалізація спроєктованої архітектури розпочинається з підготовки та базового налаштування центрального вузла мережі – сервера автентифікації. Як базове середовище функціонування обрано консольну серверну операційну систему Ubuntu Server. Застосування цього дистрибутиву обґрунтовується наявністю довгострокової підтримки (LTS), високим рівнем відмовостійкості системного ядра та інтеграцією із сучасними механізмами управління пакунками. Перед початком розгортання цільового програмного

забезпечення здійснено налаштування статичної IP-адресації та синхронізацію системного часу за протоколом NTP, що є критично важливим для уникнення математичних помилок під час валідації терміну дії цифрових сертифікатів. Після базової конфігурації виконано оновлення системних репозиторіїв та безпосередню інсталяцію програмного комплексу FreeRADIUS спільно з утилітою easy-rsa:

```
sudo apt update
sudo apt upgrade -y
sudo apt install freeradius freeradius-utils easy-rsa -y
```

Оскільки архітектура протоколу EAP-TLS функціонує виключно на базі інфраструктури відкритих ключів (PKI), логічним продовженням розгортання стала емісія криптографічних матеріалів. Для забезпечення ізоляції процесів створено окрему робочу директорію центру сертифікації (CA) та ініціалізовано базову структуру PKI за допомогою відповідних системних команд:

```
make-cadir ~/easy-rsa
cd ~/easy-rsa
./easyrsa init-pki
```

Емісію кореневого сертифіката (Root CA), що виконує функцію незмінного якоря довіри для всієї мережі, реалізовано командою ./easyrsa build-ca. Під час генерації приватний ключ центру додатково захищено парольною фразою, а самому центру присвоєно ідентифікатор Radius-Root-CA.

На наступному етапі згенеровано цифрові документи для безпосередніх учасників мережевого обміну: сертифікат для RADIUS-сервера та унікальний клієнтський мандат для IoT-пристрою (суплікanта). Критично важливою інженерною вимогою є застосування параметра nopass під час ініціалізації генерації:

```
./easyrsa gen-req radius-server nopass
./easyrsa sign-req server radius-server
./easyrsa gen-req iot-client nopass
./easyrsa sign-req client iot-client
```

Зазначений параметр забезпечує збереження закритих клієнтських та серверних ключів без додаткового парольного шифрування. Для кінцевого автономного IoT-обладнання (яке у стенді імітується операційною системою Lubuntu) такий підхід є єдиним способом гарантувати автоматичне підключення до порту комутатора після подачі живлення, оскільки апаратні інтерфейси для ручного введення пароля адміністратором на таких датчиках відсутні.

Після успішної генерації ключів виконано найважливіший етап налаштування серверної частини – конфігурацію внутрішніх робочих файлів демона FreeRADIUS. Насамперед модифіковано файл /etc/freeradius/3.0/clients.conf, який регламентує авторизацію мережевого обладнання. До структури файлу інтегровано параметри транзитного автентифікатора (запис mikrotik-stand), зафіксовано його IP-адресу (192.168.10.1) та задано унікальний криптографічний секрет (iot_secret_2026) для шифрування транзитного радіус-обміну (рис. 4.2).

Рисунок 4.2 – Реєстрація транзитного автентифікатора у конфігурації FreeRADIUS

Для ініціалізації методів сертифікатної перевірки здійснено модифікацію модуля EAP у файлі /etc/freeradius/3.0/mods-available/eap. Глобальний параметр default_eap_type жорстко переведено в значення tls, що на рівні ядра системи забороняє використання застарілих парольних протоколів. У секції tls-config tls-common прописано системні змінні та шляхи до згенерованих раніше криптографічних матеріалів (зокрема, до закритого ключа сервера private_key_file), як це продемонстровано на рисунку 4.3. Важливим безпековим аспектом є також активація параметра check_crl = yes, який змушує сервер

автоматично звірятися зі списками відкликаних сертифікатів (CRL) під час кожної транзакції.

```
tls-config tls-common {
        private_key_password = whatever
        private_key_file = ${certdir}/radius-server.key
```

Рисунок 4.3 – Налаштування криптографічних параметрів протоколу EAP-TLS

Політики динамічної мікросегментації та резервного доступу реалізовано через редагування файлу /etc/freeradius/3.0/users. Для пристрою iot-client, що успішно ідентифікується за сертифікатом, задано правило відправки атрибутів Tunnel-Type = VLAN та Tunnel-Private-Group-Id = «20». Одночасно, з метою підготовки до тестування резервної технології MAB (MAC Authentication Bypass) для застарілих сенсорів, у конфігурацію додано еталонний тестовий запис. У ньому апаратна MAC-адреса (00:11:22:33:44:55) виступає логіном та паролем (Cleartext-Password), а порту примусово призначається карантинний ідентифікатор Tunnel-Private-Group-Id = «30» (рис. 4.4).

```
"iot-client"
        Tunnel-Type = VLAN,
        Tunnel-Medium-Type = IEEE-802,
        Tunnel-Private-Group-Id = "20"

"00:11:22:33:44:55" Cleartext-Password := "00:11:22:33:44:55"
        Tunnel-Type = VLAN,
        Tunnel-Medium-Type = IEEE-802,
        Tunnel-Private-Group-Id = "30"
```

Рисунок 4.4 – Правила динамічного призначення VLAN на базі RADIUS-атрибутів

На стороні клієнта (віртуальної машини Lubuntu) згенеровані ключі безпечно розміщено в директорії /etc/wpa_supplicant/certs/. З метою максимально

достовірної імітації поведінки автономного сенсора, утиліту EAP-супліканта wpa_supplicant інтегровано в механізм автозавантаження ОС. Замість використання нестабільних скриптів оболонки, у директорії /etc/systemd/system/ створено файл повноцінної системної служби iot-auth.service(повний лістинг конфігураційного файлу наведено у додатку А.1), а параметри конфігурації утиліти автентифікації супліканта описані у файлі wired.conf (див. додаток А.2).

Активація розробленої служби в системному ядрі Linux здійснена командами systemctl daemon-reload та systemctl enable iot-auth.service. Делегування контролю підсистемі systemd гарантує ізольоване виконання процесу у фоновому режимі та забезпечує безперервні автоматичні спроби підключення у разі збоїв.

На фінальному етапі підготовлено мережевий автентифікатор MikroTik CHR. Для емуляції L2-комутатора доступу в операційній системі RouterOS створено програмний міст (bridge), до якого включено транзитні інтерфейси, та на апаратному рівні активовано розділення трафіку (VLAN Filtering). Взаємодію з RADIUS-сервером налаштовано через системне меню /radius, куди внесено IP-адресу Ubuntu-сервера та відповідний криптографічний секрет. Безпосереднє блокування неавторизованих клієнтів реалізовано через налаштування /interface dot1x server, де для клієнтського порту ether2 ініціалізовано службу портового контролю доступу у строгому режимі 802.1X (auth-types=dot1x). Повний лістинг експортованої конфігурації автентифікатора доступу MikroTik RouterOS наведено у додатку А.3.

Після збереження конфігурацій стенд було переведено у стан повної готовності до тестування.

4.3 Тестування сценаріїв безпеки

Для підтвердження дієздатності спроєктованої архітектури контролю доступу та перевірки коректності застосованих політик безпеки було розроблено

та реалізовано комплексну програму тестування. Практичні експерименти проводилися на базі розгорнутого віртуального стенда з використанням методів глибокої інспекції пакетів (Deep Packet Inspection) за допомогою програмного аналізатора мережевого трафіку Wireshark. Запропонована методика тестування охоплює верифікацію штатної роботи криптографічних протоколів, перевірку механізмів резервування доступу для застарілого обладнання, а також аналіз реакції системи на типові вектори атак канального рівня моделі OSI.

Процедура верифікації передбачала послідовне виконання ізольованих сценаріїв, кожен з яких моделював реальні умови експлуатації корпоративної IoT-інфраструктури. Перед початком кожного етапу стан мережевого автентифікатора та супліканта приводився до базового (еталонного) вигляду з метою уникнення інтерференції результатів. Усі події безпеки, процеси інкапсуляції кадрів EAPOL у RADIUS-пакети та факти динамічного призначення віртуальних локальних мереж (VLAN) фіксувалися в режимі реального часу. Отримані мережеві дампи та системні журнали використовуються як об'єктивна доказова база ефективності розробленої моделі Zero Trust (нульової довіри).

4.3.1 Верифікація механізму криптографічної ідентифікації EAP-TLS та динамічної мікросегментації мережі

Метою першого етапу тестування є перевірка штатного процесу взаємної криптографічної ідентифікації кінцевого пристрою та сервера за стандартом IEEE 802.1X (із застосуванням методу EAP-TLS). Додатковим завданням є підтвердження коректної роботи механізму динамічної мікросегментації, який передбачає автоматичне призначення ізольованої віртуальної локальної мережі (VLAN) на порту комутатора після успішної авторизації клієнта.

Для ініціалізації процесу автентифікації на клієнтській машині (Lubuntu) було активовано системну службу iot-auth.service, що запустила фоновий процес супліканта wpa_supplicant. Програмний аналізатор Wireshark, підключений до

транзитного лінку між мережевим автентифікатором (MikroTik) та AAA-сервером (Ubuntu), зафіксував генерацію кадрів EAPOL та їх подальшу інкапсуляцію у захищені RADIUS-повідомлення. Аналіз дампів підтвердив успішне проходження всіх фаз криптографічного рукостискання (TLS Handshake), під час якого сторони обмінялися сертифікатами та згенерували спільний сесійний ключ без передачі конфіденційних даних у відкритому вигляді.

Вирішальним доказом коректної роботи логіки динамічної мікросегментації є вміст фінального пакета від сервера автентифікації. Після криптографічної валідації клієнтського сертифіката iot-client, RADIUS-сервер звернувся до локальної бази політик, сформував та відправив комутатору повідомлення Access-Accept (рис. 4.5). Детальний аналіз тіла цього пакета демонструє наявність специфічних RADIUS-атрибутів (відповідно до RFC 2868), які сервер примусово передає комутатору. Зокрема, атрибут Tunnel-Type вказує на використання технології VLAN (значення 13), а Tunnel-Private-Group-Id містить ідентифікатор цільової мережі – 20.

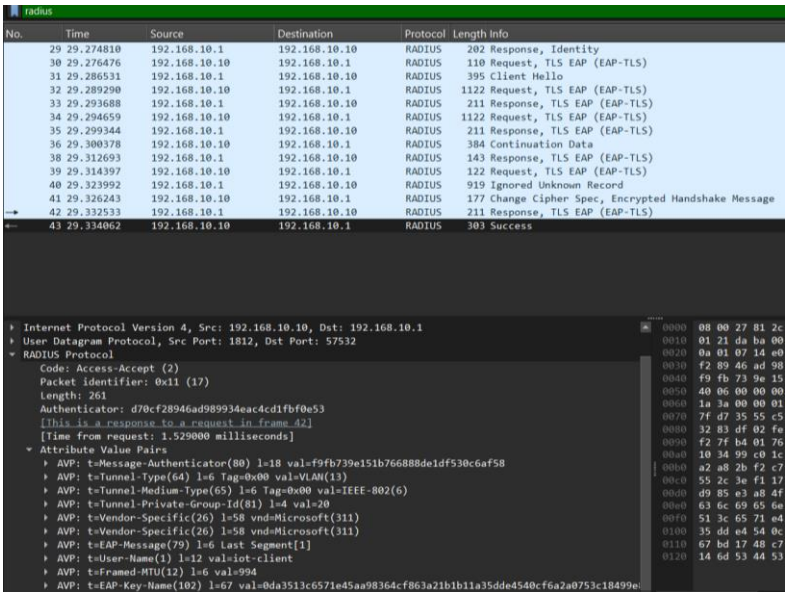

Рисунок 4.5 – Дамп RADIUS-пакета Access-Accept з атрибутами призначення VLAN 20

Отримавши та обробивши зазначені інструкції, операційна система RouterOS на автентифікаторі MikroTik автоматично змінила стан клієнтського порту ether2 на авторизований (Authorized) та логічно помістила його в межах ізольованого VLAN 20. Оскільки на даному віртуальному інтерфейсі було попередньо розгорнуто локальний DHCP-сервер, операційна система клієнта ініціювала стандартний процес отримання мережевих параметрів (DORA).

Для верифікації коректності роботи логіки призначення адрес на клієнтському вузлі було виконано перевірку мережевих інтерфейсів (команда ip a), яка підтвердила успішне отримання адреси 192.168.20.100. Одразу після цього було перевірено дієвість концепції Zero Trust: спроба клієнта ініціювати пряме ICMP-з'єднання з центральним сервером інфраструктури (192.168.10.10) була успішно заблокована правилами міжмережевого екрана (Firewall) на транзитному маршрутизаторі, що відображається як 100% втрата пакетів (рис. 4.6).

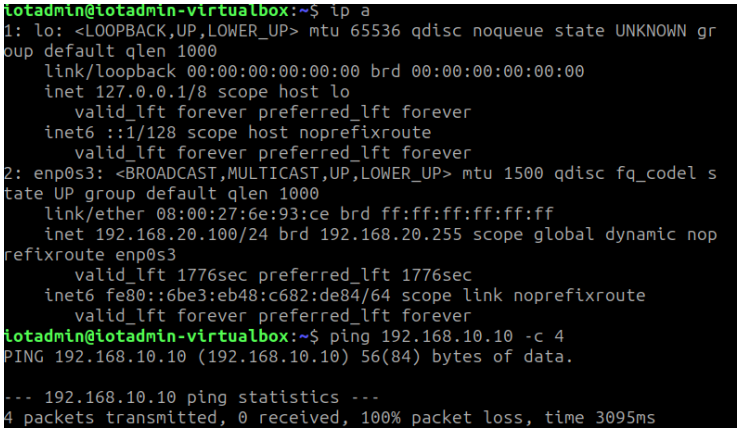

Рисунок 4.6 – Успішне отримання IP-адреси та перевірка маршрутизації у VLAN 20

Окрім базової зв'язності, у межах цього сценарію підтверджено дієвість концепції Zero Trust. Незважаючи на успішну авторизацію пристрою у мережі VLAN 20, правила міжмережевого екрана (Firewall), імплементовані на

транзитному маршрутизаторі, надійно блокують будь-які спроби IoT-сенсора ініціювати прямі мережеві з'єднання з центральним сервером інфраструктури PKI (192.168.10.10). Це унеможливлює проведення розвідки мережі або атак на AAA-сервер з боку потенційно скомпрометованого периферійного обладнання.

4.3.2 Верифікація резервного механізму доступу MAB (MAC Authentication Bypass)

Метою другого етапу тестування є перевірка здатності спроєктованої інфраструктури обробляти підключення пристроїв, які апаратно або програмно не підтримують криптографічний стек протоколу IEEE 802.1X (наприклад, застарілі індустріальні контролери, мережеві принтери або базові IoT-сенсори). Для таких вузлів єдиним можливим ідентифікатором виступає їхня унікальна апаратна MAC-адреса.

Для реалізації цього сценарію конфігурацію служби портового контролю доступу на мережевому автентифікаторі MikroTik було переведено у гібридний режим роботи. За допомогою системної команди /interface dot1x server set 0 auth-types=dot1x,mac-auth було активовано паралельну підтримку обох методів. Логіку переходу визначено параметрами mac-auth-mode=mac-as-username-and-password та retrans-timeout=2s. Згідно з цією конфігурацією, комутатор очікує пред'явлення X.509-сертифіката протягом заданого тайм-ауту (відпрацьовуючи три цикли по 2 секунди). У разі відсутності EAP-відповіді комутатор автоматично формує RADIUS-запит, використовуючи MAC-адресу клієнта одночасно як логін (User-Name) та пароль (User-Password) у відкритому вигляді (Cleartext).

Під час практичного тестування на клієнтській машині (Lubuntu) було штучно імітовано підключення застарілого пристрою. Роботу супліканта wpa_supplicant було призупинено, а апаратну MAC-адресу мережевого інтерфейсу enp0s3 було програмно змінено на заздалегідь внесену до бази

сервера (00:11:22:33:44:55). Після ініціювання мережевого з'єднання комутатор, витримавши таймаут EAP-запитів, відправив MAB-запит на сервер автентифікації.

Зафіксований аналізатором Wireshark мережевий дамп підтверджує, що RADIUS-сервер успішно ідентифікував апаратну адресу в базі даних та відповів пакетом Access-Accept (рис. 4.7).

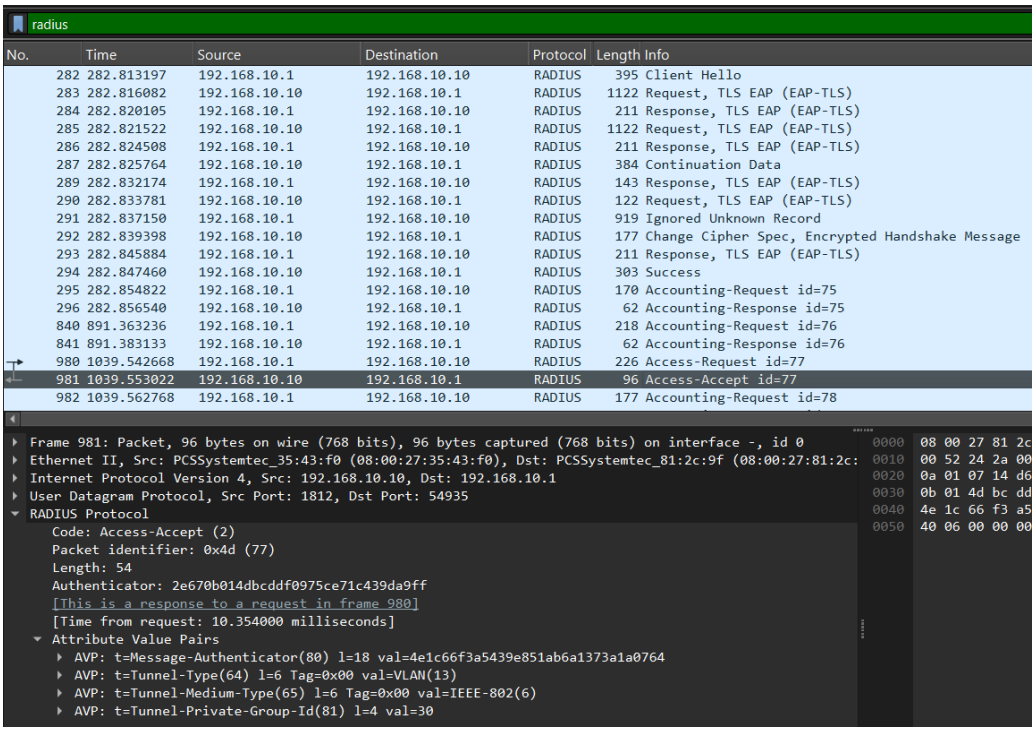

Рисунок 4.7 – Авторизація пристрою за механізмом MAB із призначенням карантинного VLAN 30

Згідно з визначеними політиками безпеки для неідентифікованого криптографічно обладнання, сервер передав комутатору атрибут Tunnel-Private-Group-Id = 30. Це забезпечило примусове поміщення даного вузла до жорстко ізольованого карантинного сегмента (VLAN 30), що мінімізує ризики латерального переміщення зловмисників у разі фізичної підміни IoT-пристрою.

4.3.3 Дослідження стійкості системи до атак із підміною апаратних ідентифікаторів (MAC Spoofing)

Оскільки розроблена гібридна модель портового контролю доступу використовує технологію MAB як резервний механізм для інтеграції застарілих периферійних пристроїв, критично важливим безпековим аспектом є перевірка її стійкості до атак типу MAC Spoofing. Цей вектор загрози передбачає, що зловмисник здійснює пасивне перехоплення трафіку в локальному сегменті мережі, виявляє легітимну апаратну адресу дозволеного пристрою (наприклад, принтера чи датчика) і програмно клонує її на власному мережевому інтерфейсі з метою несанкціонованого обходу аутентифікації.

Для проведення експерименту на базі віртуального стенда було змодельовано дії потенційного порушника. На тестовому клієнтському вузлі (Lubuntu) було тимчасово деактивовано мережевий інтерфейс enp0s3, повністю очищено поточні динамічні мережеві налаштування (команда flush) та програмно змінено оригінальну апаратну адресу на випадкову неавторизовану комбінацію 00:AA:BB:CC:DD:EE. Після повторної активації інтерфейсу та призначення статичних параметрів було ініційовано генерацію вихідного ICMP-трафіку (команда ping 192.168.10.10) з метою спровокувати мережевий автентифікатор до проведення портової перевірки.

Мережевий автентифікатор MikroTik CHR, зафіксувавши появу нового L2-кадру на порту ether2, витримав встановлений конфігурацією шестисекундний таймаут очікування EAP-сертифіката і, не отримавши відповіді, перейшов до виконання процедури MAB. Маршрутизатор сформував стандартний пакет Access-Request із порядковим ідентифікатором id=80, де як значення атрибутів User-Name та User-Password було вказано підроблену MAC-адресу 00:AA:BB:CC:DD:EE.

Отримавши зазначений запит, сервер FreeRADIUS звернувся до локального конфігураційного файлу users для звірки облікових даних. Оскільки

скомпрометована MAC-адреса була відсутня в переліку дозволених еталонних пристроїв, AAA-сервер миттєво згенерував та відправив комутатору фінальне службове повідомлення Access-Reject, заблокувавши надання мережевих ресурсів (рис. 4.8).

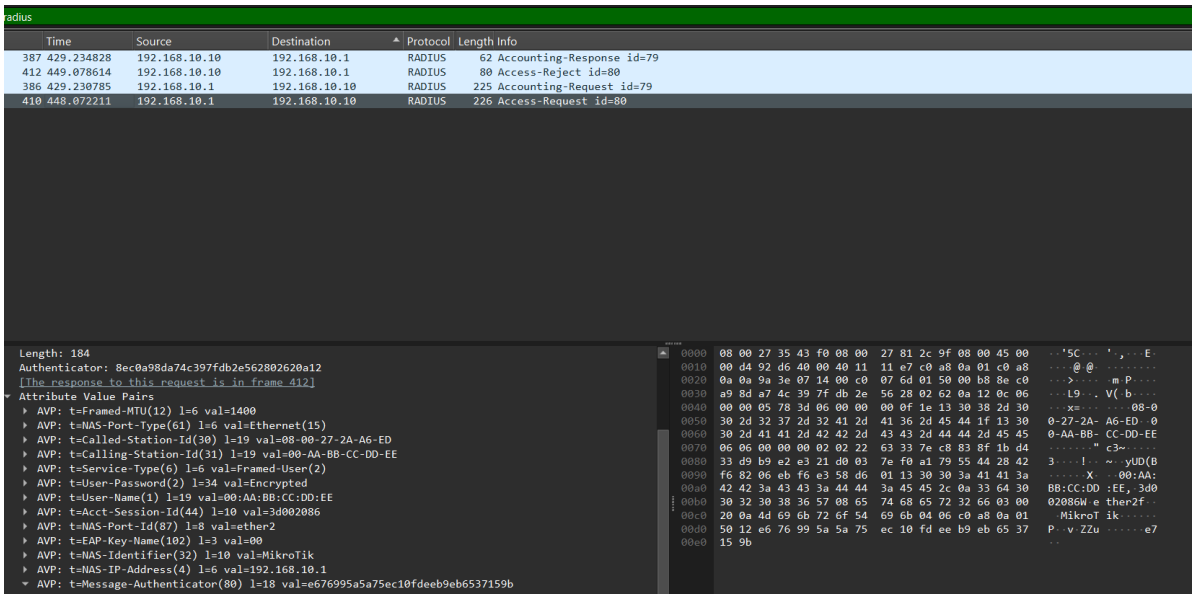

Рисунок 4.8 – Дамп RADIUS-пакета Access-Reject при спробі підміни апаратної MAC-адреси

Результати цього сценарію демонструють, що незважаючи на теоретичну вразливість самої технології MAB перед клонуванням адрес, розроблена система надійно захищає периметр мережі від підключення будь-якого випадкового або невідомого обладнання, що повністю відповідає базовим вимогам концепції Нульової довіри (Zero Trust).

4.3.4 Верифікація механізмів керування життєвим циклом цифрових мандатів та відкликання сертифікатів (CRL)

Завершальним етапом комплексної програми випробувань розгорнутого стенда стала верифікація підсистеми керування життєвим циклом цифрових

мандатів X.509. У реальних умовах експлуатації бездротових та дротових корпоративних IoT-мереж існує висока ймовірність фізичного викрадення кінцевого обладнання зловмисниками або компрометації криптографічних ключів, що зберігаються в пам'яті «headless» пристроїв. За умови використання класичного протоколу IEEE 802.1X (EAP-TLS), зловмисник, який заволодів легітимним сертифікатом пристрою, може безперешкодно обійти портовий контроль. Для нівелювання цієї загрози в архітектуру системи інтегровано механізм оперативної перевірки списків відкликаних сертифікатів.

Метою даного сценарію є перевірка швидкодії та надійності механізму блокування доступу для скомпрометованого пристрою iot-client шляхом його примусового внесення до списку відкликання центру сертифікації (CA).

Процедура тестування розпочалася на стороні сервера автентифікації Ubuntu Server. У робочій директорії локального PKI-центру за допомогою консольних утиліт пакета Easy-RSA було виконано процедуру анулювання криптографічного мандата клієнта та згенеровано оновлений файл чорного списку crl.pem:

```
cd ~/easy-rsa
./easyrsa revoke iot-client
./easyrsa gen-crl
```

Сформований бінарний файл crl.pem було безпечно скопійовано у системну директорію демона FreeRADIUS із примусовим делегуванням прав власності локальному користувачеві freerad, що необхідно для безперешкодного зчитування цього об'єкта ядром сервера. Наступним кроком у файлі конфігурації модуля EAP (/etc/freeradius/3.0/mods-available/eap) було активовано директиву check_crl = yes та чітко зафіксовано абсолютний шлях до файлу списку. Після перезапуску системної служби freeradius зміни офіційно набули чинності.

Для перевірки реакції інфраструктури на спробу несанкціонованого входу на клієнтській машині Lubuntu (яка виконувала роль скомпрометованого або

викраденого датчика) було повторно ініційовано штатну процедуру автентифікації 802.1X за допомогою утиліти wpa_supplicant.

Аналізатор мережевого трафіку Wireshark, підключений до магістрального інтерфейсу, зафіксував відправлення пакета Access-Request (порядковий номер 1377) від автентифікатора MikroTik до сервера FreeRADIUS. Під час обробки цього запиту AAA-сервер успішно зчитав сертифікат клієнта, порівняв його серійний номер із номерами, внесеними до локального оновленого списку відкликання crl.pem, і миттєво ідентифікував факт компромінації.

В результаті, замість надання доступу, сервер FreeRADIUS згенерував та надіслав у кабель фінальне службове повідомлення Access-Reject з ідентифікатором id=82 (пакет номер 1380), що зафіксовано на рисунку 4.9.

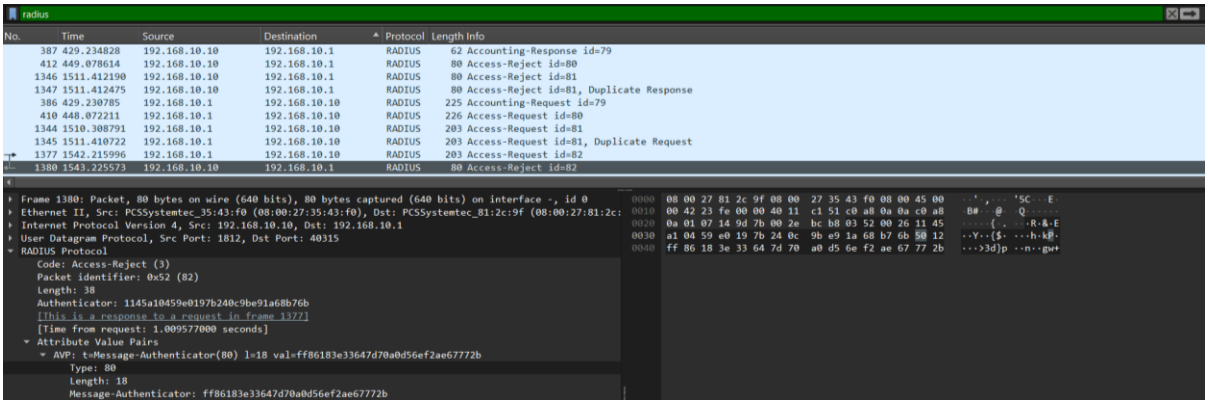

Рисунок 4.9 – Дамп RADIUS-пакета Access-Reject при спробі авторизації з використанням відкликаного сертифіката

Мережевий автентифікатор MikroTik, обробивши команду відмови, залишив порт ether2 у заблокованому стані (Unauthorized), примусово припинивши будь-яку інкапсуляцію кадрів і закривши доступ клієнту до будь-яких ресурсів корпоративної мережі. Експеримент довів, що розроблена система контролю доступу здатна оперативно реагувати на внутрішні загрози компрометації пристроїв та миттєво ізолювати периметр мережі на канальному рівні.

4.4 Аналіз результатів та мережевого трафіку. Оцінка ефективності розробленої архітектури

Після перевірки базової працездатності розробленої системи потрібн детально розібрати мережевий трафік. Головне завдання цього етапу полягало у визначенні реального навантаження, яке створюють криптографічні протоколи на комутаційне обладнання. Окрім оцінки накладних витрат, важливо було виміряти часові затримки під час підключення та зрозуміти, наскільки запропоноване рішення підходить для середовища Інтернету речей, де пристрої часто мають слабкі процесори.

Використання аналізатора Wireshark дало змогу зазирнути всередину процесу транспортування автентифікаційних даних на рівні пакетів. Як засвідчив аналіз дампів, комунікація між кінцевим IoT-вузлом та комутатором доступу відбувається суворо на канальному рівні (L2) через кадри EAPOL. Тобто до моменту успішної перевірки сертифіката пристрій фізично не має IP-адреси. Це інженерне рішення фактично зводить нанівець ризики мережевих атак (таких як сканування портів чи експлуатація вразливостей TCP/IP) з боку неавторизованого сенсора. Що стосується магістральної ділянки між комутатором і сервером RADIUS, то тут EAP-кадри загортаються в UDP-пакети. У кожному такому запиті присутній атрибут Message-Authenticator, який виконує роль цифрового підпису. Він гарантує, що транзитний трафік не був змінений під час передачі, захищаючи корпоративну мережу від атак типу «людина посередині».

Оцінюючи доцільність впровадження стандарту IEEE 802.1X для IoT, неможливо оминути питання пропускної здатності. Зібрані емпіричні дані показують, що обсяг службового трафіку під час перевірки залишається мінімальним. Звичайні стартові запити ідентифікації займають не більше 100 байт. Найбільше навантаження припадає на момент передачі відкритого ключа сервера (пакет № 314) – це приблизно 1122 байти корисного навантаження. З

огляду на ці цифри, можна стверджувати, що криптографічна перевірка не перевантажуватиме канали зв'язку навіть у моменти масового перепідключення пристроїв після перебоїв з живленням.

Швидкість встановлення захищеного з'єднання – ще один критичний маркер для автономних датчиків, які мають економити заряд батареї. Щоб з'ясувати точний час криптографічного рукостискання (TLS Handshake), було проаналізовано часові мітки еталонного підключення. Згідно з отриманим мережевим дампом на рисунку 4.10, процес складався з таких кроків:

– стартовий запит ідентифікації (пакет № 306 EAP Request, Identity) зафіксовано на позначці 253,623 с;

– обмін сертифікатами та формування ключів (пакет № 314 Server Hello, Certificate) відбулися о 253,670 с;

– повідомлення про успішну авторизацію (пакет № 320 EAP Success) сервер надіслав на позначці 253,706 с.

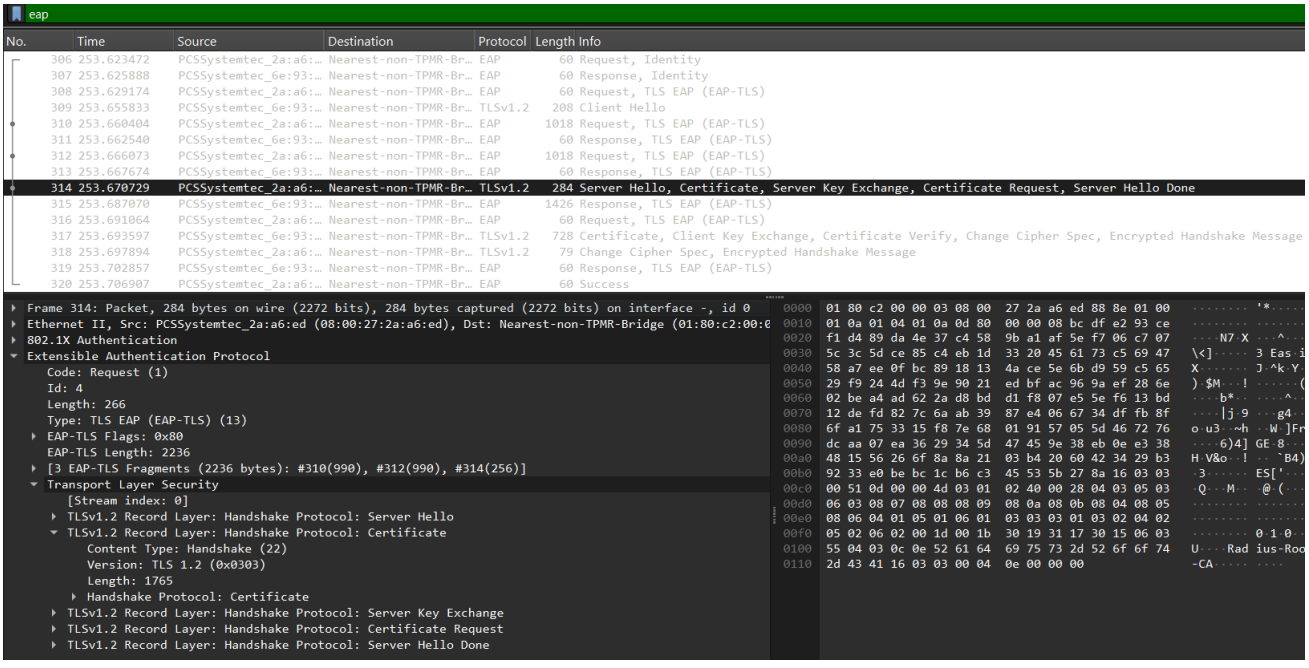

Рисунок 4.10 – Аналіз тривалості криптографічного рукостискання (TLS Handshake) за часовими мітками Wireshark

Різниця в часі між першим та останнім кроком становить рівно 0,083 секунди (83 мілісекунди). Для більшості IoT-протоколів прикладного рівня, де таймаути очікування сягають кількох секунд, така затримка є абсолютно невідчутною. Крім того, швидке завершення обчислень на стороні клієнта дозволяє мікроконтролерам миттєво повертатися в режим енергозбереження, що подовжує термін служби автономних систем.

Підсумовуючи результати мережевого аналізу, можна констатувати високу надійність запропонованої архітектури. Інтеграція інфраструктури відкритих ключів (PKI) разом із 802.1X нівелює загрози, пов'язані з використанням слабких статичних паролів. Разом із тим, наявність гібридного режиму резервування через MAB дає змогу безпечно підключати застаріле обладнання, примусово ізолюючи його у відповідних VLAN згідно з принципами Zero Trust. Практичні показники швидкодії та мінімальні мережеві витрати доводять, що розроблене рішення повністю відповідає вимогам до сучасних корпоративних мереж.

Висновки до розділу 4

У четвертому розділі здійснено практичну реалізацію та комплексне тестування спроєктованої системи контролю мережевого доступу. На базі середовища глибокої мережевої симуляції GNS3 розгорнуто повнофункціональний ізольований стенд, що включає транзитний автентифікатор (MikroTik CHR), AAA-сервер (FreeRADIUS) та IoT-суплікант (Lubuntu). Конфігурація криптографічних параметрів повністю узгоджена з актуальними вимогами стандарту NIST SP 800-52r2 щодо використання виключно безпечних версій протоколу TLS.

Емпірично підтверджено працездатність механізму динамічної мікросегментації інфраструктури. Доведено, що використання RADIUS-атрибутів дозволяє комутатору автоматично розподіляти легітимні пристрої у

відповідні віртуальні мережі (VLAN), водночас жорстко блокуючи їхній прямий доступ до критичних серверів керування на рівні міжмережевого екрана (Firewall), що повною мірою реалізує парадигму Zero Trust.

Успішно протестовано роботу гібридного режиму автентифікації. Для обладнання, позбавленого криптографічних можливостей, перевірено резервний механізм MAC Authentication Bypass (MAB) із примусовим переведенням таких вузлів у карантинний сегмент мережі. Разом із тим, імітація атаки з підміною мережевого ідентифікатора (MAC Spoofing) продемонструвала миттєве відхилення пакетів сервером, що підтверджує стійкість архітектури до несанкціонованого фізичного проникнення.

Проведено тестування сценарію компрометації кінцевого вузла та перевірено механізм відкликання цифрових мандатів. Встановлено, що інтеграція списків CRL у процес автентифікації дозволяє системі безпомилково ідентифікувати та апаратно блокувати підключення пристроїв із відкликаними сертифікатами ще на етапі рукостискання.

Аналіз часових затримок транзитного трафіку показав, що повна процедура встановлення захищеного з'єднання (TLS Handshake) займає близько 83 мілісекунд. Це доводить високу операційну ефективність розробленої архітектури та її абсолютну придатність для розгортання в індустріальних IoT-мережах з жорсткими обмеженнями обчислювальних ресурсів кінцевого обладнання.

ВИСНОВКИ

У ході виконання дипломної роботи було успішно досягнуто поставлену мету – спроєктовано систему керування доступом до мережі на основі стандарту IEEE 802.1X, оптимізовану для безінтерфейсних (headless) IoT-пристроїв, з використанням протоколу EAP-TLS та відокремленої інфраструктури відкритих ключів (PKI).

За результатами проведеного дослідження отримано такі результати:

1. Проаналізовано вразливості сучасних IoT-мереж та існуючі механізми їхнього захисту. Доведено, що традиційні підходи (базова фільтрація за заводськими MAC-адресами або Port Security) виявилися неспроможними протистояти атакам типу MAC Spoofing та перехопленню трафіку. Встановлено, що інтеграція розумної техніки у корпоративне середовище вимагає відмови від довіри всередині периметра та переходу до парадигми Zero Trust, де дозвіл на передачу даних надається виключно після суворої криптографічної перевірки.

2. Досліджено архітектурні компоненти та принципи функціонування стандарту IEEE 802.1X. Обґрунтовано відмову від застарілих парольних протоколів для автономних сенсорів, що не мають інтерфейсів вводу. Натомість доведено ефективність застосування методу EAP-TLS, який забезпечує взаємну криптографічну автентифікацію вузлів на рівні мережевого порту та унеможливлює підключення стороннього обладнання.

3. Розроблено модель життєвого циклу цифрових сертифікатів (включно з їхньою генерацією засобами easy-rsa та відкликанням через списки CRL). Створена архітектура з відокремленим центром сертифікації (CA) дозволила автоматизувати видачу криптографічних мандатів без парольного захисту (nopass) для фонових служб wpa_supplicant. Доведено надійність управління цим процесом: внесення скомпрометованого сертифіката до списку відкликання гарантує негайну заборону доступу для викраденої апаратури без переналаштування комутаторів.

4. Спроєктовано механізм мікросегментації мережі на основі RADIUS-атрибутів. Досягнуто динамічного керування доступом: після успішної перевірки X.509-сертифіката комутатор автоматично переводить порт у визначену сервером віртуальну локальну мережу (VLAN). Для інтеграції застарілого обладнання налаштовано гібридний режим доступу MAC Authentication Bypass (MAB) із примусовою ізоляцією таких вузлів у жорстко обмеженому карантинному сегменті (VLAN 30) із застосуванням політик міжмережевого екранування.

5. Проведено практичну емуляцію та тестування запропонованої системи в середовищі GNS3. На базі відкритого програмного забезпечення (FreeRADIUS, MikroTik RouterOS) розгорнуто повноцінний інфраструктурний стенд. Аналіз дампів глибокої інспекції пакетів (DPI) у Wireshark підтвердив, що повний цикл криптографічного рукостискання (TLS Handshake) триває лише 83 мілісекунди, а максимальне корисне навантаження становить 1122 байти.

Таким чином, розроблений комплекс вирішує актуальне науково-прикладне завдання усунення прогалин в безпеці headless-пристроїв і формує надійний, обчислювально ефективний фундамент для побудови сучасних корпоративних мереж, повністю захищених від несанкціонованого фізичного проникнення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications / A. Al-Fuqaha [et al.] // IEEE Communications Surveys & Tutorials. − 2015. − Vol. 17, № 4. − P. 2347–2376. − DOI: 10.1109/COMST.2015.2444095. − URL: https://ieeexplore.ieee.org/document/7123563 (дата звернення: 10.04.2026).

2. A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures / V. Hassija [et al.] // IEEE Access. − 2019. − Vol. 7. − P. 82721–82743. − DOI: 10.1109/ACCESS.2019.2924045. − URL: https://ieeexplore.ieee.org/document/8742551 (дата звернення: 10.04.2026).

3. IEEE Standard for Local and Metropolitan Area Networks − Port-Based Network Access Control : IEEE Std 802.1X-2020. − New York : IEEE, 2020. − 235 p. − DOI: 10.1109/IEEESTD.2020.9015620. − URL: https://ieeexplore.ieee.org/document/9015620 (дата звернення: 12.04.2026).

4. Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations / N. Neshenko [et al.] // IEEE Communications Surveys & Tutorials. − 2019. − Vol. 21, № 3. − P. 2702–2733. − DOI: 10.1109/COMST.2019.2910750. − URL: https://ieeexplore.ieee.org/document/8688434 (дата звернення: 15.04.2026).

5. Man-in-the-Middle Attacks in IoT Networks: Detection and Prevention / J. Whang [et al.] // 2020 IEEE International Conference on Communications (ICC). − Dublin, 2020. − P. 1–6. − DOI: 10.1109/ICC40277.2020.9148784. − URL: https://ieeexplore.ieee.org/document/9148784 (дата звернення: 18.04.2026).

6. MAC Spoofing Attacks and Defense Mechanisms / Cisco Systems. − 2021. − URL: https://www.cisco.com/c/en/us/support/docs/security/mac-spoofing.html (дата звернення: 20.04.2026).

7. Understanding the Mirai Botnet / M. Antonakakis [et al.] // 26th USENIX Security Symposium. − Vancouver, 2017. − P. 1093–1110. − URL:

https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis (дата звернення: 20.04.2026).

8. Network Access Control in IoT: Challenges, Architectures, and Future Directions / M. Ma [et al.] // IEEE Internet of Things Journal. − 2020. − Vol. 7, № 10. − P. 9636–9650. − DOI: 10.1109/JIOT.2020.2996924. − URL: https://ieeexplore.ieee.org/document/9099684 (дата звернення: 25.04.2026).

9. Device Profiling in Cisco Identity Services Engine (ISE) / Cisco Systems. − 2021. − URL: https://www.cisco.com/c/en/us/support/docs/security/identity-services-engine/profiling-design-guide.html (дата звернення: 25.04.2026).

10. Cisco ISE Architecture and Deployment / Cisco Systems. − 2023. − URL: https://www.cisco.com/c/en/us/products/security/identity-services-engine/index.html (дата звернення: 27.04.2026).

11. FreeRADIUS Technical Guide / FreeRADIUS. − 2023. − URL: https://freeradius.org/documentation/ (дата звернення: 28.04.2026).

12. MikroTik RouterOS Dot1X / MikroTik Documentation. − 2023. − URL: https://help.mikrotik.com/docs/display/ROS/Dot1X (дата звернення: 12.05.2026).

13. Extensible Authentication Protocol (EAP) : RFC 3748 / B. Aboba [et al.]. − 2004. − URL: https://datatracker.ietf.org/doc/html/rfc3748 (дата звернення: 12.05.2026).

14. Remote Authentication Dial In User Service (RADIUS) : RFC 2865 / C. Rigney [et al.]. − 2000. − URL: https://datatracker.ietf.org/doc/html/rfc2865 (дата звернення: 12.05.2026).

15. The EAP-TLS Authentication Protocol : RFC 5216 / D. Simon [et al.]. − 2008. − URL: https://datatracker.ietf.org/doc/html/rfc5216 (дата звернення: 12.05.2026).

16. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile : RFC 3280 / R. Housley [et al.]. − 2002. − URL: https://datatracker.ietf.org/doc/html/rfc3280 (дата звернення: 12.05.2026).

17. Stallings W. Cryptography and Network Security: Principles and Practice / W. Stallings. − 8th ed. − London : Pearson, 2020. − 768 p.

18. Public Key Infrastructure (PKI) / IBM. − 2023. − URL: https://www.ibm.com/think/topics/public-key-infrastructure (дата звернення: 12.05.2026).

19. Rescorla E. The Transport Layer Security (TLS) Protocol Version 1.2 : RFC 5246 / E. Rescorla. − 2008. − URL: https://datatracker.ietf.org/doc/html/rfc5246 (дата звернення: 12.05.2026).

20. MAC Authentication Bypass Deployment Guide / Cisco Systems. − 2022. − URL: https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Security/TrustSec_1-99/MAB/MAB_Dep_Guide.html (дата звернення: 12.05.2026).

21. Gutmann P. Simple Certificate Enrollment Protocol (SCEP) : Internet-Draft / P. Gutmann. − 2021. − URL: https://datatracker.ietf.org/doc/html/draft-nourse-scep (дата звернення: 12.05.2026).

22. Enrollment over Secure Transport (EST) : RFC 7030 / M. Pritikin [et al.]. − 2013. − URL: https://datatracker.ietf.org/doc/html/rfc7030 (дата звернення: 12.05.2026).

23. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol − OCSP : RFC 6960 / S. Santesson [et al.]. − 2013. − URL: https://datatracker.ietf.org/doc/html/rfc6960 (дата звернення: 12.05.2026).

24. Easy-RSA Documentation / OpenVPN. − 2023. − URL: https://easy-rsa.readthedocs.io/en/latest/ (дата звернення: 12.05.2026).

25. Zero Trust Architecture : NIST Special Publication 800-207 / S. Rose [et al.] // National Institute of Standards and Technology. − 2020. − 59 p. − DOI: 10.6028/NIST.SP.800-207. − URL: https://csrc.nist.gov/publications/detail/sp/800-207/final (дата звернення: 12.05.2026).

26. RADIUS Attributes for Tunnel Protocol Support : RFC 2868 / G. Zorn [et al.] // IETF. − 2000. − 20 p. − URL: https://datatracker.ietf.org/doc/html/rfc2868 (дата звернення: 12.05.2026).

ДОДАТОК А

Лістинги конфігураційних файлів клієнтського середовища та мережевого автентифікатора

А.1 Лістинг файлу системної служби автоматичної автентифікації (iot-auth.service)

```
[Unit]

Description=IoT 802.1X Auto Authenticator

After=network.target

[Service]

Type=simple

ExecStart=/sbin/wpa_supplicant -c /etc/wpa_supplicant/wired.conf -i
enp0s3 -D wired

Restart=always

RestartSec=3

[Install]
WantedBy=multi-user.target
```

А.2 Лістинг файлу конфігурації утиліти автентифікації супліканта (wired.conf)

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
eapol_version=2
ap_scan=0
network={
    key_mgmt=IEEE8021X
    eap=TLS
    identity="iot-client"
    ca_cert="/etc/wpa_supplicant/certs/ca.crt"
    client_cert="/etc/wpa_supplicant/certs/iot-client.crt"
    private_key="/etc/wpa_supplicant/certs/iot-client.key"
    eapol_flags=0
}
```

А.3 Лістинг конфігурації автентифікатора доступу (MikroTik RouterOS)

```
/interface bridge
add name=br-iot vlan-filtering=yes
/interface bridge port
add bridge=br-iot interface=ether1
add bridge=br-iot interface=ether2
/interface bridge vlan
add bridge=br-iot tagged=br-iot vlan-ids=20,30

/radius
add address=192.168.10.10 service=dot1x

/interface dot1x server
add interface=ether2 retrans-timeout=2s
```

Бібліографічна довідка

Тема роботи: Проектування системи керування доступом до мережі на основі протоколу 802.1X для IoT-інфраструктури.

Обсяг бакалаврської роботи 93 сторінки.

Перелік графічного матеріалу:

1. Система керування доступом для IoT-інфраструктури. Структурно-логічна схема
2. Модель розподілу довіри в інфраструктурі PKI. Схема структурна
3. Процес автентифікації IoT-пристрою за алгоритмом EAP-TLS. Діаграма послідовності
4. Динамічне призначення VLAN на рівні автентифікатора. Діаграма послідовності

Дата закінчення БР

Студент Владислав ВАКІВ