

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“___” _____ 2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
**“Програмне забезпечення системи кібербезпеки Honey IoT-
пристроїв для навчальних атак мережі”**

Виконав здобувач вищої освіти
IV курсу, групи КБ-22-2
ОПП «Кібербезпека»
спеціальності 125 «Кібербезпека»
_____ Ткаченко О.С.
«___» _____ 2026 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Дреєв О.М.
«___» _____ 2026 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Галузь знань 12 "Інформаційні технології"
Спеціальність 125 "Кібербезпека"
Освітньо-професійна (освітньо-наукова) програма "Кібербезпека"

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

_____ Олексій СМІРНОВ

« 27 » лютого 2026 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Ткаченка Олексія Сергійовича

(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи кібербезпеки Honeypot IoT-пристроїв для навчальних атак мережі*

2. Керівник роботи *Дреєв Олександр Миколайович, канд. техн. наук, доцент*

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

затверджені наказом вищого навчального закладу № 129-02 від 27.02.2026 року

3. Строк подання студентом роботи до захисту *23.05.2026 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка програмного забезпечення системи кібербезпеки Honeypot IoT-пристроїв для навчальних атак мережі*

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

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи кібербезпеки в промислову експлуатацію.

6. Висновки

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

Структурна схема системи кібербезпеки *1 аркуш*

Функціональна схема системи кібербезпеки *1 аркуш*

Діаграма процесів *1 аркуш*

Принципова схема пристроїв *1 аркуш*

Блок-схеми алгоритмів роботи пристроїв *5 аркушів*

7. Дата видачі завдання « 27 » лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2026 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2026 р.	
3.	Розробка моделі компонента	20.03.2026 р.	
4.	Розробка структур даних	25.03.2026 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2026 р.	
6.	Програмування алгоритмів	10.04.2026 р.	
7.	Оформлення ПЗ	17.04.2026 р.	
8.	Попередній захист роботи	23.05.2026 р.	

Дата видачі завдання
« 27 » лютого 2026 р.

Підпис керівника

Дреєв О.М.
(прізвище та ініціали)

Завдання прийнято до виконання
« 27 » лютого 2026 р.

Підпис здобувача

Ткаченко О.С.
(прізвище та ініціали)

АНОТАЦІЯ

Ткаченко О.С. Програмне забезпечення системи кібербезпеки Honey IoT-пристроїв для навчальних атак мережі. 125 Кібербезпека. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи кібербезпеки Honey IoT-пристроїв для навчальних атак мережі.

Метою роботи є розробка програмного забезпечення системи кібербезпеки Honey IoT-пристроїв для навчальних атак мережі.

Результат роботи – програмно-апаратна реалізація системи кібербезпеки Honey IoT-пристроїв для навчальних атак мережі.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено програмне забезпечення автономного лабораторного комплексу. Програмну логіку реалізовано мовою MicroPython для кластера мікроконтролерів ESP32 та у вигляді CGI-скриптів (Shell) для маршрутизатора з ОС OpenWRT. Реалізовано низку ізольованих мікросервісів на базі ESP32.

Новизна роботи полягає у використанні підходу «Hardware-in-the-loop» та гейміфікації формату Capture The Flag (CTF).

Результати роботи можуть бути впроваджені в навчальний процес закладів вищої освіти для проведення лабораторних та практичних занять з дисциплін, пов'язаних із захистом комп'ютерних мереж, аудитом інформаційної безпеки та етичним хакінгом. Стенд забезпечує безпечне, економічно ефективне та ізольоване середовище для тестування кібератак без ризику для основної IT-інфраструктури.

Ключові слова: кібербезпека, навчальний стенд, інтернет речей (IoT), мікроконтролери ESP32, honeypot, етичний хакінг, CTF

ABSTRACT

Tkachenko O.S. Cybersecurity system software of Honey IoT devices for educational network attacks. 125 Cybersecurity. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

In this qualification work for the first (bachelor's) level of higher education, software has been developed, which is intended for the cybersecurity system of Honey IoT devices for educational network attacks.

The purpose of the work is the development of software for the cybersecurity system of Honey IoT devices for educational network attacks.

The result of the work is the hardware-software implementation of the cybersecurity system of Honey IoT devices for educational network attacks.

The software for the autonomous laboratory complex has been developed. The software logic is implemented in MicroPython for the ESP32 microcontroller cluster and as CGI scripts (Shell) for a router running the OpenWrt OS. A number of isolated microservices based on ESP32 have been implemented.

The novelty of the work lies in the use of the "Hardware-in-the-loop" approach and the gamification of the Capture The Flag (CTF) format.

The results of the work can be implemented into the educational process of higher education institutions for conducting laboratory and practical classes in disciplines related to computer network protection, information security auditing, and ethical hacking. The stand provides a safe, cost-effective, and isolated environment for testing cyberattacks without risk to the main IT infrastructure.

Keywords: cybersecurity, training stand, Internet of Things (IoT), ESP32 microcontrollers, honeypot, ethical hacking, CTF

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ І ОБЛАСТЬ ВИКОРИСТАННЯ	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	9
2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	9
2.2 Обґрунтування вибору засобів для побудови системи кібербезпеки та мови програмування	13
2.3 Розгорнута постановка завдання	18
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЄКТНИХ РІШЕНЬ	20
3.1 Опис функціонування системи	20
3.2 Розробка структурної схеми системи.....	22
3.3 Розробка функціональної схеми	25
3.4 Розробка діаграми процесів.....	28
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ ТА ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЄКТНИХ І ПРОГРАМНИХ РІШЕНЬ	34
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	34
4.2 Захист розробленого програмного забезпечення.....	46
5 ВПРОВАДЖЕННЯ СИСТЕМИ КІБЕРБЕЗПЕКИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	50
6 ОСНОВНІ ВИСНОВКИ.....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	58

					ВКРБ-125.26.0044.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.	Ткаченко О.С.				Програмне забезпечення системи кібербезпеки Honey IoT-пристроїв для навчальних атак мережі	Лім.	Аркуш	Аркушів
Перев.	Дреєв О. М.					Б	1	61
Н.контр.	Коваленко А.С					КБ-22-2		
Затв.	Смірнов О.А.							

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

AES	—	симетричний алгоритм блочного шифрування
API	—	програмний інтерфейс додатків
ARP	—	протокол визначення адреси
ASCII	—	стандартний код обміну інформацією
BLE	—	технологія Bluetooth з низьким енергоспоживанням
BR/EDR	—	базова/покращена швидкість передачі даних у Bluetooth
CGI	—	загальний шлюзовий інтерфейс
CoAP	—	протокол передачі даних пристроїв з обмеженими ресурсами
CTF	—	формат навчальних завдань та змагань з кібербезпеки
DAC/ЦАП	—	цифро-аналоговий перетворювач
DDoS	—	розподілена атака на відмову в обслуговуванні
DHCP	—	протокол динамічного налаштування вузла мережі
DMZ	—	демілітаризована зона (сегмент мережі)
DNS	—	система доменних імен
ECC	—	криптографія на еліптичних кривих
ESP32	—	мікроконтролер сімейства Espressif Systems
FTP	—	протокол передачі файлів
GND	—	заземлення, загальний контакт живлення
GPIO	—	універсальні входи/виходи мікроконтролера
HTML	—	мова розмітки гіпертексту
HTTP	—	протокол передачі гіпертексту
IDPS	—	система виявлення та запобігання вторгненням
IoT	—	інтернет речей
IP	—	інтернет-протокол (IP-адреса)
IPv4	—	четверта версія інтернет-протоколу
JSON	—	текстовий формат обміну даними

LAN	—	локальна обчислювальна мереж
MAC	—	апаратна адреса мережевого пристрою
MitM	—	вид кібератаки «людина посередині»
MOSFET	—	польовий транзистор (використовується як силовий ключ)
MQTT	—	спрощений мережевий протокол для IoT-пристроїв
OSI	—	базова еталонна модель взаємодії відкритих систем
PSQL	—	система керування базами даних
PWM/ШИМ	—	широтно-імпульсна модуляція, метод керування потужністю
RAM	—	оперативна пам'ять
REST	—	архітектурний стиль взаємодії компонентів додатку
REPL	—	середовище програмування для виконання команд
RSA	—	криптографічний алгоритм з відкритим ключем
RST	—	прапорець скидання у TCP-пакеті
SHA-2	—	безпечний алгоритм хешування
SPI	—	послідовний периферійний інтерфейс
SSH	—	протокол безпечного віддаленого керування
SSID	—	ідентифікатор (ім'я) бездротової мережі Wi-Fi
SVG	—	масштабована векторна графіка (формат файлу)
TCP	—	протокол передачі з гарантованою доставкою даних
TEE	—	механізм дублювання (віддзеркалення) мережевого трафіку
TLS	—	криптографічний протокол передачі даних
UART	—	універсальний асинхронний приймач-передавач
UDP	—	протокол користувацьких датаграм
USB	—	універсальна послідовна шина
UTF-8	—	формат кодування символів
VPN	—	віртуальна приватна мережа
WAN	—	глобальна комп'ютерна мережа
Wi-Fi	—	технологія бездротової локальної мережі
WPA/WPA2	—	протоколи захищеного доступу до бездротової мережі Wi-Fi

ВСТУП

Актуальність теми. У сучасному цифровому світі, де кіберзагрози стають дедалі складнішими, питання підготовки фахівців з інформаційної безпеки набувають критичного значення. Традиційні методи навчання, які базуються виключно на теоретичних матеріалах або закритих віртуальних симуляціях, часто виявляються недостатніми для формування реальних навичок протидії загрозам. Для глибокого розуміння мережевих вразливостей, принципів перехоплення трафіку та методів етичного хакінгу студентам необхідний практичний досвід роботи з реальним обладнанням.

Існуючі віртуальні платформи (CTF) частково вирішують проблему практики, проте вони не здатні повною мірою симулювати фізичний рівень мережі та специфіку взаємодії з апаратним забезпеченням. Створення фізичного навчального стенду – такого як проєкт MissionZOO, що базується на використанні доступних мікроконтролерів ESP32 та маршрутизатора з прошивкою OpenWRT – є надзвичайно актуальним завданням для сучасної ІТ-освіти.

Актуальність проєкту підкріплюється наступними аспектами:

- реалізм середовища: використання мікроконтролерів ESP32 дозволяє створити мережу «тварин» – інтерактивних сервісів (PostgreSQL, веб-сервери, Telnet-гаунтлети), що імітують реальну інфраструктуру;
- практичне опанування інструментарієм: проєкт вимагає від студентів професійного використання індустріальних засобів аналізу, таких як Wireshark для сніфінгу трафіку, Nmap/Zenmap для сканування портів та Packet Sender для маніпуляції пакетами;
- навчання методам захисту та моніторингу: завдяки інтеграції системи виявлення вторгнень (IDPS) на базі вузла Suricata та скриптів блокування на рівні маршрутизатора OpenWRT, учні навчаються не лише атакувати, а й розуміти механізми автоматичного реагування на загрози;

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

– етична складова: навчання проводиться у контрольованому середовищі, що дозволяє безпечно відпрацьовувати сценарії експлуатації сервісів без ризику для зовнішніх мереж, формуючи принципи етичного хакінгу.

Таким чином, розробка системи MissionZOO дозволяє ліквідувати прогалину між теорією та практикою, надаючи майбутнім спеціалістам унікальну можливість взаємодії з мережевим трафіком та апаратними засобами захисту.

Мета й завдання дослідження. Метою даної роботи є розробка апаратно-програмного навчального комплексу MissionZOO – безпечного середовища для практичного вивчення основ мережевого аналізу, виявлення вразливостей та принципів роботи систем захисту. Для досягнення цієї мети було поставлено наступні завдання:

а) дослідити можливості мікроконтролерів ESP32 для симуляції різних мережевих сервісів (бази даних, веб-сервери, системи виявлення вторгнень);

б) налаштувати маршрутизатор на базі прошивки OpenWRT для управління трафіком та реалізації апаратного блокування порушників;

в) програмно реалізувати 5 унікальних вузлів («тварин» зоопарку): Elephant, Falcon, Suricat, Canary та Python, кожен з яких відіграє специфічну роль у мережі;

г) розробити логіку покрокового CTF-квесту для перевірки навичок студентів.

Практична цінність отриманих результатів. Результати цієї роботи мають безпосереднє практичне застосування в освітньому процесі. Створена система MissionZOO є готовим тренувальним полігоном. Студенти отримують можливість у контрольованому, безпечному середовищі використовувати професійні інструменти аудиту (Nmap, Wireshark, Netcat, Packet Sender), не порушуючи при цьому законодавство та не наражаючи на небезпеку реальну інфраструктуру навчального закладу. Система легко масштабується та може бути використана в університетах, на курсах з кібербезпеки та для самостійного навчання.

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ І ОБЛАСТЬ ВИКОРИСТАННЯ

Призначення системи

Сучасний ландшафт кіберзагроз стрімко еволюціонує під впливом розвитку Інтернету речей (IoT), хмарних технологій та розподілених мікросервісних архітектур. Кожен новий вузол у мережі розширює поверхню атаки, що вимагає від фахівців з інформаційної безпеки не лише глибоких теоретичних знань, а й практичного розуміння апаратних і мережевих процесів. Класичні підходи до навчання, що спираються переважно на програмні симуляції або закриті віртуальні середовища, виявляються недостатніми. Вони часто не здатні достовірно відтворити специфіку фізичного рівня мережевої взаємодії. Через це майбутнім фахівцям складно відрізнити реальні аномалії від помилкових спрацьовувань систем захисту.

На ринку праці спостерігається дефіцит спеціалістів, які мають реальний досвід роботи з інструментами мережевого моніторингу та аудиту. Тривалий час створення апаратних полігонів для відпрацювання кібератак було занадто дорогим та складним завданням для більшості закладів вищої освіти. Однак поява потужних і доступних мікроконтролерів, змінила ситуацію, дозволивши перенести симуляцію складних сервісів на недороге обладнання.

Апаратно-програмний комплекс MissionZOO розроблений саме для того, щоб заповнити цю прогалину в освітньому процесі. Цей навчальний стенд делегує рутинні процеси імітації мережевої інфраструктури мікроконтролерам, забезпечуючи студентів «живою» апаратною базою. Система не просто моніторить дії користувача, а фізично реагує на них, що дозволяє максимально наблизити навчальний процес до реальних умов реагування на інциденти.

Такий підхід дозволяє відійти від сухої теорії. Студенти отримують можливість на практиці отримати навички атакуючої команди (Red Team), та

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

познайомитись з особливостями мережних сервісів та способами їх захисту (Blue Team).

1.2 Область застосування

Розроблена система MissionZOO є спеціалізованим навчально-дослідним апаратно-програмним комплексом, який призначений для моделювання складних мережових загроз та сценаріїв захисту в безпечному, повністю ізольованому середовищі. На відміну від суто віртуальних лабораторій та хмарних симуляторів, інтеграція реального апаратного забезпечення – мікроконтролерів ESP32 та маршрутизаторів із прошивкою OpenWRT – дозволяє максимально наблизити умови навчання до реальної інфраструктури. Це дає змогу охопити значно ширший спектр практичних задач, що робить систему універсальним інструментом.

Область застосування проєкту MissionZOO можна розділити на кілька ключових напрямків.

Вища та спеціальна технічна освіта

У системі академічної освіти MissionZOO має значний потенціал для впровадження як базовий лабораторний стенд. Він орієнтований на підготовку здобувачів рівнів «Бакалавр» та «Магістр» за спеціальностями «Кібербезпека», «Комп'ютерна інженерія», «Комп'ютерні науки» та «Телекомунікації». Система ліквідує розрив між сухою теорією та реальною практикою, дозволяючи студентам відпрацювати теми з таких ключових дисциплін:

- мережеві технології: глибоке вивчення рівнів моделі OSI на практиці, аналіз роботи протоколів TCP/UDP, розуміння механізмів формування бродкаст-запитів та фізичної маршрутизації пакетів;
- адміністрування мереж: отримання навичок роботи з дзеркалюванням трафіку, керування таблицями MAC-адрес на рівні маршрутизатора.
- захист інформації: практичне опанування методів виявлення вторгнень (IDPS), аналіз хибних спрацьовувань, а також вивчення принципів роботи

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

мережевих пасток (Honeypots) на прикладі взаємодії з розподіленими вузлами системи.

Професійна сертифікація та інтенсивне навчання (Bootcamps)

Сфера комерційної ІТ-освіти та курсів інтенсивної підготовки вимагає використання індустріальних стандартів та реальних інструментів аудиту. Завдяки своїй архітектурі, MissionZOO виступає ідеальним тренувальним майданчиком для фахівців, які готуються до складання сертифікацій з інформаційної безпеки. Слухачі курсів отримують можливість у легальному та безпечному середовищі відпрацьовувати сценарії мережевої розвідки за допомогою сканера Nmap, проводити глибокий аналіз перехопленого трафіку у Wireshark, а також здійснювати ручну взаємодію з вразливими сервісами через Telnet, формуючи стійкі навички етичного хакінгу.

Організація та проведення змагань формату CTF (Capture The Flag)

Традиційні змагання з кібербезпеки найчастіше проводяться на базі веб-платформ або віртуальних машин. Архітектура з розподіленими мікроконтролерами-тваринами дозволяє використовувати MissionZOO як унікальну апаратну платформу для проведення командних CTF-змагань. Наявність фізичного доступу до мережі, інтерактивний відгук обладнання (наприклад, вібро сигнали при спрацюванні пасток) та складна логіка вузлів додають змаганням інтерактивності.

Широкий спектр функціональних можливостей та модульна структура MissionZOO підкреслюють її високу універсальність як освітнього та тренувального інструменту. Використання цього комплексу у процесі підготовки майбутніх фахівців з інформаційної безпеки гарантує формування актуальних практичних навичок.

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти

У сучасній практиці підготовки фахівців з інформаційної безпеки та мережевого адміністрування склалося кілька ключових підходів до створення навчальних середовищ. Кожен із них має свої архітектурні особливості, переваги та обмеження, які важливо враховувати при розробці комплексних рішень на кшталт MissionZOO.

Віртуальні навчальні платформи (CTF та Lab-середовища)

Найбільш поширеним підходом є використання хмарних або локальних віртуалізованих лабораторій.

Hack The Box (HTB) – одна з найвідоміших платформ для тренування навичок пентесту. Користувачі отримують доступ до вразливих віртуальних машин через VPN-канал.

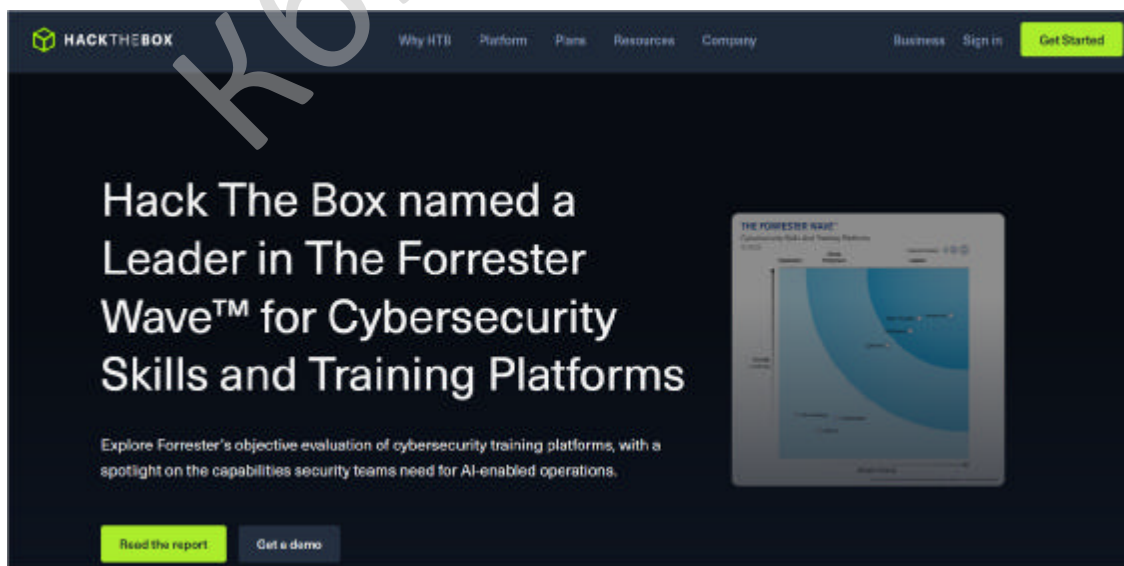


Рисунок 2.1 – Веб-сайт Hack The Box [1]

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

Ключовою особливістю платформи Hack The Box є наявність великої кількості різноманітних сценаріїв, відомих як «Boxes», які детально імітують конфігурації реальних серверів та складну архітектуру корпоративних мереж з різними рівнями вразливостей. Це дозволяє користувачам відпрацьовувати навички етичного хакінгу та пошуку прогалин у системі захисту в умовах, що максимально наближені до реальних робочих середовищ. До основних переваг такого підходу належить висока масштабованість системи, що забезпечує стабільний доступ для необмеженої кількості одночасних користувачів, а також повна відсутність потреби у значних інвестиціях у дороге фізичне мережеве обладнання чи його обслуговування. Крім того, гейміфікація процесу навчання через систему рейтингів, бейджів та змагальних завдань значно підвищує мотивацію студентів, перетворюючи складне вивчення кібербезпеки на інтерактивний та захопливий процес, що стимулює до постійного вдосконалення навичок.

TryHackMe – ця платформа орієнтована переважно на початківців, надаючи структуровані навчальні доріжки (Pathways) з покроковими інструкціями та вбудованими браузерними віртуальними машинами. Це значно знижує поріг входження, проте, як і у випадку з НТВ, доступ до мережі здійснюється через VPN-канал, що обмежує можливості вивчення низькорівневих мережевих процесів.



Рисунок 2.2 – Веб-сайт Try Hack Me [2]

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Існує і багато інших CTF майданчиків [3]. Але їх головним обмеженням є ізоляція від фізичного рівня мережі. Користувачі не мають змоги працювати з реальними радіохвилями Wi-Fi, виконувати атаки на рівні MAC-адрес у фізичному сегменті або вивчати специфіку апаратної частини IoT-пристроїв.

Програмні симулятори та емулятори мереж

Ці інструменти використовуються для проектування та налагодження мережових топологій.

Cisco Packet Tracer є комплексною програмою-симулятором, розробленою Cisco Networking Academy, яка дозволяє користувачам проектувати та моделювати мережі, використовуючи віртуальні копії обладнання Cisco. Цей інструмент ідеально підходить для академічного вивчення базових концепцій маршрутизації та комутації, оскільки він забезпечує наочну візуалізацію проходження пакетів у режимі реального часу. Проте варто враховувати, що дана платформа є лише спрощеною програмною моделлю реального світу. Її головним недоліком є неможливість запуску повноцінних зовнішніх інструментів аудиту безпеки, таких як Wireshark або Nmap, безпосередньо всередині симуляції. Це зумовлено тим, що логіка роботи мережових протоколів у Packet Tracer є значно спрощеною порівняно з реальними стеками програмного забезпечення (ПЗ), що обмежує можливості для глибокого аналізу трафіку та тестування вразливостей. [4]

GNS3 (Graphical Network Simulator-3). На відміну від симуляторів, GNS3 виступає як потужний графічний емулятор мереж, який дозволяє запускати реальні образи операційних систем, включаючи Cisco IOS та різноманітні дистрибутиви Linux, у віртуалізованому середовищі. Використання реальних образів ПЗ забезпечує надзвичайно високу точність симуляції складних мережових процесів та дозволяє інженерам тестувати конфігурації, які згодом будуть розгорнуті на фізичному обладнанні. Завдяки підтримці віртуалізації (наприклад, через QEMU або Docker), GNS3 дозволяє будувати змішані топології, які можна інтегрувати з реальною мережевою інфраструктурою, що робить цей інструмент незамінним для професійної підготовки та лабораторних досліджень. [5]

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

Але навіть просунуті емулятори часто не здатні адекватно відтворити поведінку бездротових інтерфейсів у реальних умовах (завади, потужність сигналу) та не забезпечують взаємодію з фізичним IoT-обладнанням.

Порівняльна характеристика та переваги розробленого проєкту

Розроблена система MissionZOO займає унікальну нішу апаратних навчальних стендів, пропонуючи гібридний підхід, який поєднує переваги роботи з реальним обладнанням та високу гнучкість конфігурування програмного забезпечення. На відміну від суто хмарних CTF-платформ або програмних симуляторів мережі, які пропонують ідеалізоване або віртуалізоване середовище, розроблений комплекс забезпечує глибоке занурення у фізичний рівень мережевої взаємодії.

На відміну від вищезгаданих систем, розроблена система забезпечує:

а) фізичну взаємодію (Hardware-in-the-loop) та реалізм інфраструктури: використання мікроконтролерів ESP32 дозволяє студентам працювати з реальними апаратними пристроями, що імітують повноцінні мережеві сервіси (PostgreSQL, веб-сервери, Telnet-сесії). Це не лише відтворює логіку роботи цих сервісів, але й демонструє специфіку поведінки реальних IoT-пристроїв та вбудованих систем у мережі;

б) роботу з фізичним рівнем бездротової мережі (Wi-Fi): студенти підключаються до реальної точки доступу на базі маршрутизатора OpenWRT та аналізують «живі» радіохвилі. Це критично важливо для навчання сніфінгу трафіку за допомогою Wireshark, оскільки дозволяє досліджувати реальні пакети з усіма супутніми явищами – мережевим шумом, затримками, фрагментацією та широкомовними (Broadcast/Multicast) запитами, які часто відфільтровуються або відсутні у віртуальних симуляторах;

в) апаратне та автоматизоване реагування на атаки: система реалізує складні сценарії активного захисту, де система виявлення вторгнень тісно взаємодіє з ядром мережі. Маршрутизатор здатний миттєво та фізично заблокувати MAC-адресу зловмисника при виявленні агресивного сканування портів або спроб

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

брутфорсу. Це змушує студентів не лише атакувати, але й швидко адаптуватися до протидії, застосовуючи засоби обходу блокувань, такі як підміна апаратних адрес (MAC Spoofing) за допомогою утиліт типу Technitium MAC Address Changer [20];

г) повну підтримку індустріальних інструментів аудиту безпеки: оскільки створена мережа є справжньою, а не емульованою, студенти можуть розгортати повноцінні дистрибутиви (наприклад, Kali Linux) та використовувати весь потенціал професійних утиліт (Nmap/Zenmap, Packet Sender, та інші) без жодних обмежень, які зазвичай притаманні закритим навчальним симуляторам;

д) економічну доступність: незважаючи на апаратну реалізацію, використання мікроконтролерів ESP32 робить стенд фінансово доступним для масового розгортання в аудиторіях навчального закладу (на відміну від купівлі дорогих серверних стійок);

Таким чином, MissionZOO є значно більш наочним, інтерактивним та ефективним інструментом для підготовки IT-фахівців. Він успішно долає розрив між теоретичними знаннями та практичними навичками, максимально наближуючи навчальний процес до суворих реалій експлуатації, аудиту та захисту сучасних комп'ютерних мереж.

2.2 Обґрунтування вибору засобів для побудови системи кібербезпеки та мови програмування

Для реалізації навчального проєкту MissionZOO було обрано апаратне та програмне забезпечення, що дозволяє максимально наочно продемонструвати студентам принципи мережевого аналізу, етичного хакінгу та експлуатації вразливостей у контрольованому середовищі.

Мікроконтролери ESP32 (WROOM-32 CH340C)

Для симуляції різних мережевих вузлів («тварин» у зоопарку) було обрано мікроконтролери серії ESP32. На відміну від класичних плат Arduino, ESP32 має

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

вбудований стек протоколів TCP/IP та Wi-Fi модуль, що є критично важливим для створення інтерактивного мережевого середовища.

Нове покоління мікроконтролерів серії ESP32 дозволяє створювати складні сценарії взаємодії, включаючи симуляцію баз даних, роботу з TCP/UDP сокетом та розгортання веб-серверів у мініатюрному корпусі з низьким енергоспоживанням.

Наразі лідером серед мікроконтролерів для IoT-рішень та навчальних стендів є серія ESP32 від компанії Espressif Systems. ESP32 – це серія потужних 32-бітних чіпів, що мають двоядерний дизайн, підтримку Wi-Fi та дворежимного Bluetooth. Мікроконтролер оснащений мікропроцесором Tensilica Xtensa LX6, що працює на частоті до 240 МГц. Модуль розробника побудований на базі ESP-WROOM-32 – високопродуктивного поєданого Wi-Fi + Bluetooth + BLE модуля, що призначений для широкого спектра застосувань.

Однією з ключових особливостей обраної версії є наявність інтегрованого USB-UART конвертера CH340C, що забезпечує стабільне підключення до персонального комп'ютера (ПК) та зручність програмування.

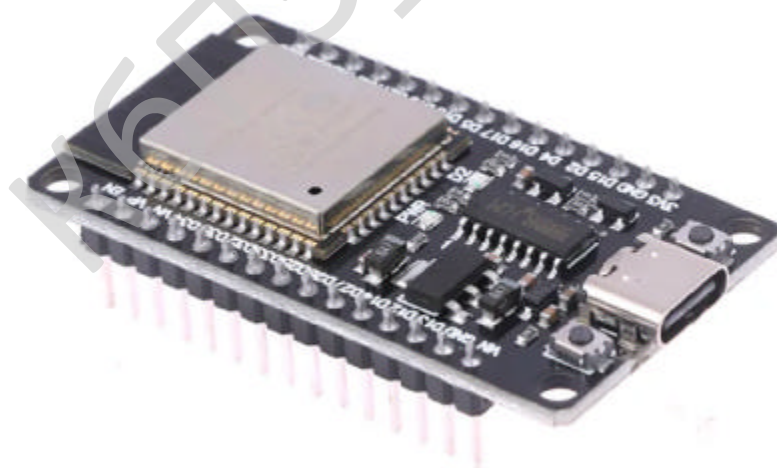


Рисунок 2.3– Зовнішній вигляд ESP32 WROOM-32 (CH340C) [6]

- підтримка MicroPython (v1.26.1), що забезпечує швидке прототипування;
- мережеві протоколи: IPv4, TCP/UDP, HTTP, FTP, MQTT;
- режими безпеки Wi-Fi: WPA/WPA2/WPA2-Enterprise.

Маршрутизатор TP-LINK TL-WR740N

Для створення мережевої інфраструктури «Зоопарку» було обрано маршрутизатор TP-LINK TL-WR740N. Це бюджетний пристрій, який завдяки підтримці кастомної прошивки OpenWRT перетворюється на керований мережевий вузол з доступом до системного рівня Linux.



Рисунок 2.5 – Маршрутизатор TP-LINK TL-WR740N [7]

Використання OpenWRT дозволяє реалізувати специфічні функції, необхідні для навчального проєкту:

- дзеркалювання трафіку (Mirroring): за допомогою пакетів iptables-mod-tee та kmod-ipt-tee роутер дублює пакети на пристрій студента для аналізу в Wireshark;

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

– керування доступом: використання утиліти `uci` та скриптів для динамічного блокування MAC-адрес «зловмисників»-студентів через API.

Технічні характеристики роутера:

- процесор: Atheros AR9331 (400 МГц);
- пам'ять: 4 МБ Flash / 32 МБ RAM;
- швидкість Wi-Fi: до 150 Мбіт/с (802.11n);
- порти: 1 WAN, 4 LAN (10/100 Мбіт/с).

Мови програмування та середовище розробки

Для даного проєкту основною мовою програмування мікроконтролерів була обрана MicroPython. На відміну від стандартної C/C++, MicroPython дозволяє студентам взаємодіяти з пристроєм у реальному часі через REPL-термінал, що значно прискорює процес навчання та налагодження мережескриптів.

Переваги MicroPython для MissionZOO:

- високорівневі бібліотеки: вбудовані модулі `network` та `usocket` дозволяють створювати TCP-сервери лише кількома рядками коду;
- асинхронність: бібліотека `uasyncio` дозволяє мікроконтролеру одночасно обробляти веб-запити та моніторити мережевий трафік (IDS функції).

Середовищем розробки обрано Visual Studio Code з розширенням Soldered MicroPython Helper. Це професійний інструмент, який забезпечує зручне завантаження файлів конфігурації на пристрої та роботу з консоллю.

Для автоматизації на боці роутера використано Bash та CGI-скрипти. Вони виступають сполучною ланкою (API) між «тваринами»-мікроконтролерами та системними функціями маршрутизатора (блокування, дзеркалювання).

Враховуючи вищесказане, поєднання ESP32, OpenWRT та MicroPython створює ідеальний технічний базис для реалізації інтерактивного навчального полігону з кібербезпеки.

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускну кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи кібербезпеки Honey IoT-пристроїв для навчальних атак мережі.

В процесі розробки випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) обґрунтувати актуальність теми розробки навчального полігону з кібербезпеки та визначити область його застосування у навчальному процесі;

б) переглянути аналогічні існуючі системи (платформи для CTF, honeypot-системи) з метою виявлення їх переваг та недоліків для побудови освітнього середовища;

в) обґрунтувати вибір засобів для побудови системи, а саме: мікроконтролерів серії ESP32, маршрутизатора TP-LINK TL-WR740N з операційною системою OpenWRT, а також мов програмування MicroPython та Bash/CGI;

г) підготувати мережеву інфраструктуру: прошити маршрутизатор операційною системою OpenWRT, створити відкриту мережу MissionZOO для можливості сніфінгу трафіку та встановити пакет iptables-mod-tee для дзеркалювання пакетів;

д) розробити та розгорнути CGI-скрипти на маршрутизаторі для забезпечення програмного управління правилами брандмауера та механізмами блокування через HTTP-запити;

е) прошити 5 мікроконтролерів ESP32 та розробити програмну логіку для 5 унікальних взаємопов'язаних сервісів («тварин»):

1) Elephant – реалізація центрального контролера та симулятора бази даних PostgreSQL;

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

2) Falcon – створення веб-панелі адміністративного моніторингу (Dashboard);

3) Suricat – розробка системи виявлення вторгнень (IDPS) для аналізу TCP-прапорів та виявлення сканування портів;

4) Canary – побудова системи-приманки (Honeyrot) з фальшивими портами та механізмами ідентифікації порушників;

5) Python – реалізація фінального шлюзу-головоломки з динамічним розподілом ролей портів;

ж) налагодити механізм API-взаємодії між усіма компонентами системи для автоматизації процесів блокування порушників та надсилання сповіщень у Telegram-бот;

з) протестувати систему шляхом симуляції мережових атак (сканування, brute-force), щоб виявити та усунути можливі помилки в логіці роботи захисних механізмів;

и) розробити та роздрукувати 3D-моделі для кожного з мікроконтролерів-тварин;

к) написати перелік умовних позначень, символів та термінів, що використовуються в області мережевої безпеки та вбудованих систем при написанні БР;

л) надати список літератури, що була використана для дослідження протоколів TCP/IP, архітектури OpenWRT та програмування на MicroPython;

м) написати анотацію на українській та англійській мовах, що коротко описує суть навчального полігону MissionZOO.

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

Опис функціонування системи

Проаналізувавши існуючі платформи для навчання кібербезпеці та визначивши їхні переваги, було прийнято рішення розробити програмно-апаратний комплекс MissionZOO. Система спрямована на вирішення проблем, пов'язаних із складністю розгортання реалістичних мережесценаріїв для студентів, забезпечуючи інтерактивне середовище на базі мікроконтролерів та спеціалізованого мережевого обладнання.

Основна функція даної системи – симуляція вразливої інфраструктури, виявлення мережесценаріїв атак та автоматичне реагування на загрози. Кожному компоненту системи відповідає певний вузол («тварина»), що виконує свою роль у сценарії.

Основними критеріями функціонування системи є:

- підключення бібліотек: використання MicroPython (v1.26.1) та модулів uasyncio для асинхронної обробки запитів, urequests для HTTP-взаємодії та network для керування WiFi-інтерфейсами;
- динамічна ініціалізація: генерація центральним вузлом (Elephant) унікальних параметрів сесії: довіреної адреси (Admin IP), адреси для перехоплення (Mirror IP) та секретних ключів;
- реєстрація вузлів: автоматична синхронізація всіх ESP32-пристроїв через HTTP API для формування єдиної мапи мережі;
- активація дзеркалювання: виклик CGI-скриптів на маршрутизаторі OpenWrt для дублювання трафіку на Mirror IP, що дозволяє студентам аналізувати весь трафік мережі через Wireshark;
- моніторинг вторгнень: сканування вхідних запитів вузлом Suricat на наявність сигнатур сканування (наприклад, RST-пакетів Nmap);

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

- робота Honeypot: симуляція вразливих сервісів на вузлі Canary з використанням пасток на портах для виявлення агресивної розвідки;
- апаратне блокування: формування запитів до маршрутизатора для миттєвого блокування MAC-адреси порушника на рівні бездротового інтерфейсу;
- синхронізація станів: періодичне оновлення даних між Falcon та Elephant для відображення актуальних логів безпеки на панелі моніторингу;
- валідація послідовності: перевірка зібраних ключів на фінальному вузлі Python у суворій відповідності до порядку, визначеного в базі даних Elephant;
- завершення циклу: видача фінального ключа при успішному проходженні всіх етапів.

Цей алгоритм слугує каркасом безпечного середовища, де фізичні пристрої взаємодіють через мережеві протоколи, а результати дій користувача (успішні чи помилкові) відображаються у реальному часі через систему сповіщень.

Враховуючи освітню мету проекту, критично важливим було забезпечити економічну ефективність системи. Використання мікроконтролерів ESP32 WROOM-32 CH340C та бюджетного маршрутизатора TP-LINK TL-WR740N з прошивкою OpenWrt дозволило значно знизити вартість розгортання комплексу порівняно з професійними тренажерами. Вибір MicroPython як мови розробки дозволяє легко масштабувати проект, додаючи нові сценарії без необхідності повної переробки архітектури.

Система MissionZOO спроектована як масштабована платформа. Якщо розглядати її як базу для комплексного навчання з кібербезпеки, можливим напрямком розвитку є додавання нових «тварин» (вузлів), що спеціалізуються на інших векторах атак:

- датчики DNS-спуфінгу: впровадження вузла для симуляції атак на систему доменних імен;
- модулі Brute-force захисту: спеціалізований вузол для відпрацювання навичок захисту від підбору паролів до SSH/FTP сервісів;

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

- IoT Vulnerability node: додавання пристроїв, що імітують вразливі розумні пристрої (MQTT, CoAP);
- Traffic Injection module: інтеграція інструментів для симуляції складних MITM-атак безпосередньо через прошивку роутера.

Інтеграція таких додаткових компонентів підвищить навчальну цінність системи та дозволить студентам отримувати досвід роботи з ширшим спектром мережових загроз. Використання гнучких CGI-скриптів на маршрутизаторі забезпечує легку інтеграцію будь-яких нових апаратних засобів у загальний алгоритм захисту мережі.

Розробка структурної схеми системи

Відповідно до теоретичних засад проектування інформаційних систем, структурна схема комплексу MissionZOO відображає сукупність елементарних апаратних та архітектурних ланок об'єкта, а також фізичні та мережеві зв'язки між ними.

Апаратно-програмний комплекс MissionZOO побудований за модульним принципом і складається з чотирьох основних архітектурних блоків:

а) робоче місце користувача (ПК/ноутбук): початкова елементарна ланка, яка виступає в ролі клієнтського термінала. З цього вузла студент здійснює підключення до цільової мережі навчального стенда. Ця ланка передбачає використання спеціалізованого програмного забезпечення (мережових сканерів Nmap, аналізаторів трафіку Wireshark, Telnet-клієнтів), за допомогою якого користувач проводить розвідку, аналізує перехоплений трафік та безпосередньо взаємодіє з імітованими сервісами;

б) маршрутизатор (OpenWRT): центральний комутаційний вузол, який виконує роль ядра локальної мережі та апаратного шлюзу. Використання пристрою з операційною системою на базі ядра Linux (OpenWRT) забезпечує не лише створення стабільної бездротової мережі, але й виконання складних

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

мережевих операцій. Маршрутизатор здійснює внутрішню комутацію пакетів та містить вбудовану підсистему фаєрволу (базовану на iptables), яка використовується для реалізації TEE-дзеркалювання трафіку та фізичного відключення клієнтів-порушників (через маніпуляції з MAC-фільтрами) шляхом виклику локальних CGI-скриптів;

в) вузли мережі (мікроконтролери ESP32): кластер з п'яти незалежних апаратних платформ (ESP32 Elephant, ESP32 Suricat, ESP32 Canary, ESP32 Falcon, ESP32 Python), побудованих на базі мікроконтролерів із вбудованими радіомодулями. Кожна плата є окремою елементарною ланкою та виконує роль виділеного мережевого сервера;

г) Telegram Bot: зовнішній логічний вузол, інтегрований в інфраструктуру через глобальну мережу. Він використовується як виділений канал (Out-of-band management) для збору телеметрії, логування інцидентів та відправки оперативних сповіщень адміністратору (викладачу). Це забезпечує можливість віддаленого моніторингу за ходом виконання завдань без безпосереднього втручання в ігровий процес студентів.

Зв'язки між цими архітектурними ланками формують комунікаційне середовище системи і реалізовані наступним чином:

– бездротове з'єднання (Wi-Fi локальна мережа): робоче місце студента підключається до маршрутизатора через стандартний бездротовий інтерфейс стандарту IEEE 802.11, формуючи клієнтську частину мережі. Аналогічним чином маршрутизатор з'єднується з усіма п'ятьма мікроконтролерами ESP32, які утворюють серверну інфраструктуру стенда. Саме через це єдине Wi-Fi середовище здійснюється весь спектр внутрішньої комунікації: обмін даними між мікроконтролерами (через внутрішні HTTP та UDP-запити), взаємодія плат із системними CGI-скриптами OpenWRT для управління доступом, а також маршрутизація трафіку студента до кінцевих вузлів-мішеней;

– транзитний вихід в Інтернет (WAN): передача даних до зовнішнього вузла Telegram Bot здійснюється за захищеним протоколом HTTPS. Оскільки

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

мікроконтролери знаходяться у внутрішній ізольованій мережі, вони формують запити, які проходять транзитом через маршрутизатор. Останній виступає шлюзом NAT (Network Address Translation), дозволяючи вихідний трафік до серверів Telegram API.

Таким чином, структурна схема наочно демонструє децентралізовану апаратну топологію зіркоподібного типу (Star Topology), де маршрутизатор є єдиною точкою комутації між користувачем, мікроконтролерами та зовнішнім світом. Така архітектура забезпечує надзвичайну масштабованість: за необхідності розширення стенда достатньо лише підключити до мережі нову плату ESP32 без втручання в конфігурацію базового ядра системи.

Структурну схему системи зображено на Рисунку 3.1.

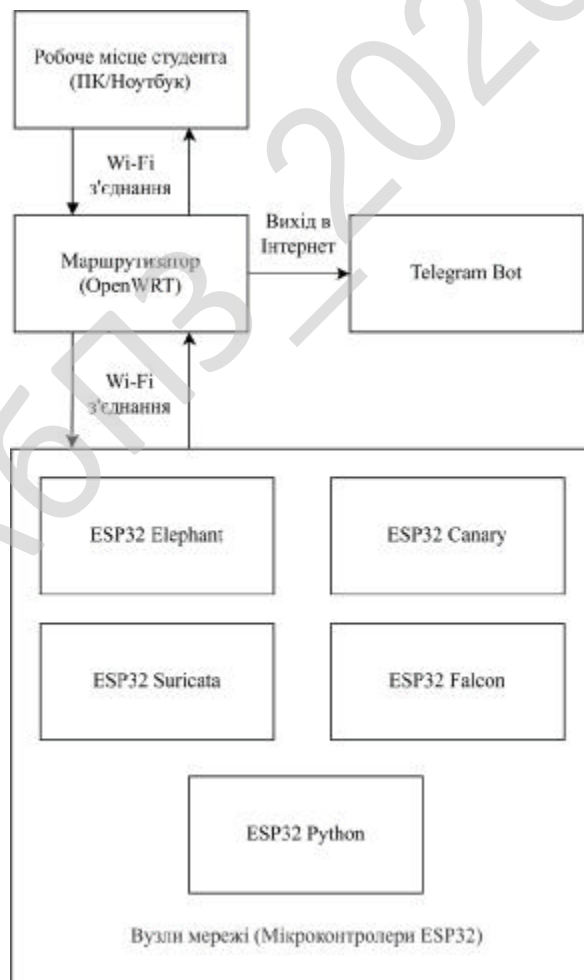


Рисунок 3.1 – Структурна схема системи

Розробка функціональної схеми

Програмна архітектура MissionZOO базується на децентралізованій моделі, де логіка розподілена між маршрутизатором та кластером мікроконтролерів ESP32. Система функціонує як набір ізольованих мікросервісів, що взаємодіють через внутрішній HTTP API та UDP-запити. Кожен апаратний вузол достовірно імітує автономний корпоративний сервіс і носить умовну назву «тварини», що вносить елемент CTF-гейміфікації. Розглянемо кожний вузол окремо.

Центральний контролер та база даних (Elephant)

Основним координаційним ядром є вузол Elephant, який ініціалізує глобальний стан мережі. Він динамічно генерує ключові параметри (Admin IP, Mirror IP) та послідовність випробувань. Вузол підтримує два сервіси: HTTP API сервер (порт 80) для реєстрації інших пристроїв і прийому звітів про порушення, та модуль симуляції PostgreSQL (порт 5432) для обробки Telnet-з'єднань. Модуль містить парсер SQL-запитів до віртуальних таблиць і логіку захисту від Bruteforce-атак, яка ініціює блокування порушника при перевищенні ліміту невірних паролів.

Система виявлення вторгнень – IDPS (Suricata)

Цей вузол виконує роль головного мережевого захисника. Його модуль sniffer_task фоновно аналізує TCP-трафік через «сирі» сокети (Raw Sockets), виявляючи патерни сканування Nmap за кількістю RST-прапорців. Внутрішній API (порт 80) приймає запити на блокування від інших вузлів, а порт 7000 видає частину ключа при отриманні кодової фрази «Give me a key, please». Suricata активно взаємодіє з маршрутизатором (через HTTP GET) для активації TEE-дзеркалювання та блокування зловмисних IP, а також розсилає широкомовні UDP-підказки студентам.

Багатопортова система-пастка Honeypot (Canary)

Вузол дезорієнтації, що прослуховує масив портів (80-84), де лише один випадковий порт є «справжнім». Звернення до портів-пасток миттєво відправляє звіт до Suricata для блокування користувача. Справжній порт містить сторінку

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

автентифікації з фрагментом ключа, прихованим за допомогою стеганографії у вихідному коді (SVG viewBox). Окремий порт 8080 генерує фальшиві дані для обману зовнішніх сканерів.

Панель моніторингу та управління (Falcon)

Модуль імітує закритий адміністративний веб-інтерфейс на порту 8080. Доступ дозволено лише зі встановленої системою Admin IP-адреси, що вимагає від студента застосування методів IP Spoofing. Вузол синхронізується з Elephant, отримуючи звіти про атаки та змінний пароль «Punch Keyword». Для отримання ключа користувач має обійти перевірку IP та правильно ввести перехоплений пароль.

Фінальне випробування (Python)

Кінцева точка сценарію, що розгортає Telnet-сервери на портах 22, 23, 2222 та 2323. Ролі портів (Trigger, Secret, Hidden, Fake) генеруються випадково при кожному старті. Спочатку студент має активувати прихований порт (Hidden), звернувшись до Trigger-порту з підробленої Admin IP, отримати дані для авторизації на Secret-порту. Після цього необхідно авторизуватися на Hidden-порту і поетапно ввести зібрані раніше фрагменти ключів у згенерованій базою Elephant послідовності.

Управління маршрутизатором (Router Scripts)

Апаратне керування мережевим рівнем здійснює маршрутизатор OpenWRT через CGI-скрипти block та mirror (написані на shell). Вони напряду працюють з підсистемою iptables та усі, забезпечуючи блокування MAC-адрес порушників на рівні Wi-Fi фільтрів та організацію TEE-дзеркалювання пакетів для сніфінгу трафіку студентами.

Кожен вузол використовує два файли: конфігураційний (config_ файл) та основний з логікою роботи (main файл) (див. Додаток Б).

Функціональну схему програмної взаємодії пристроїв системи MissionZOO можна побачити на рисунку 3.2.

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

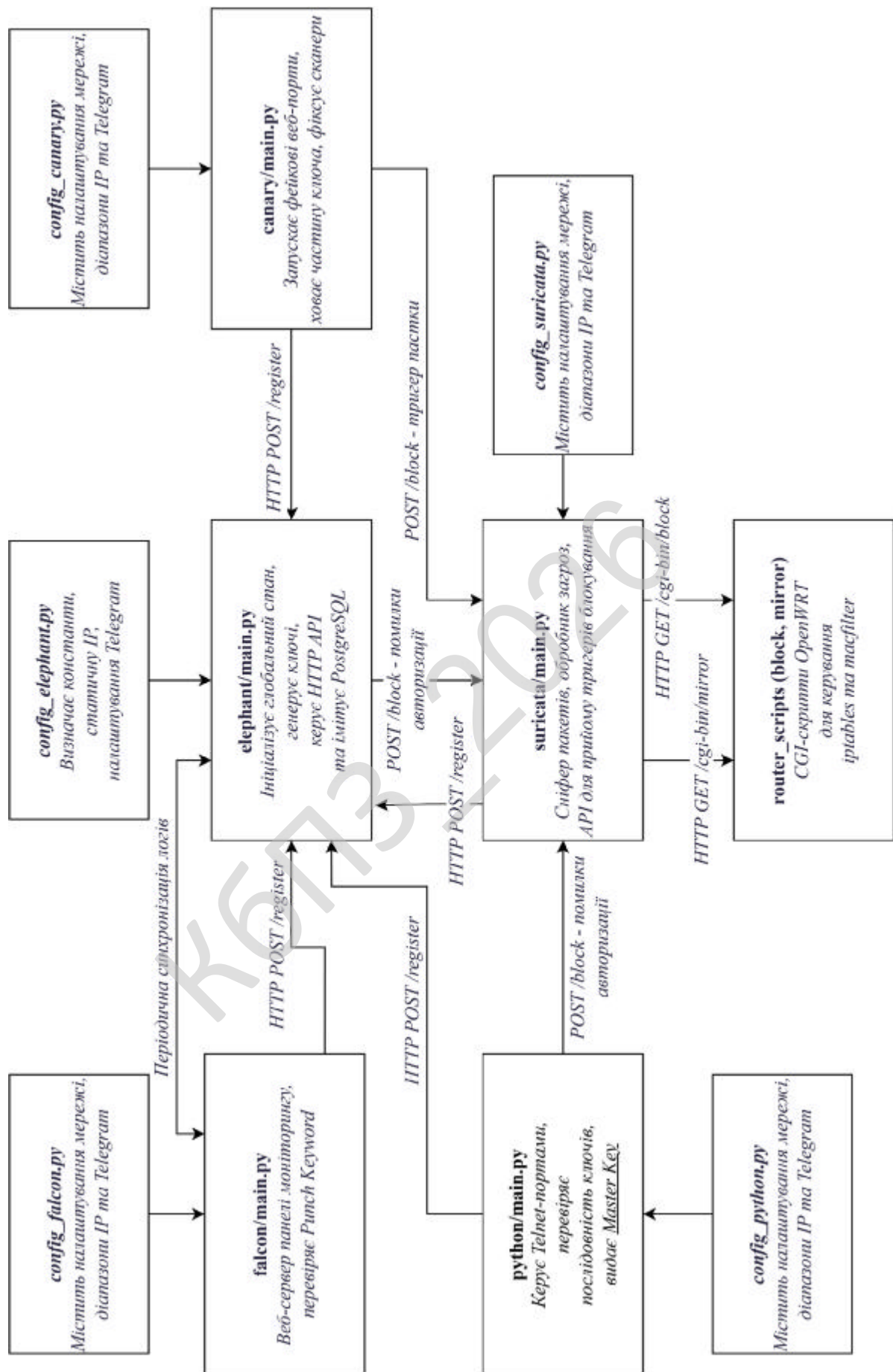


Рисунок 3.2 – Функціональна схема системи

Розробка діаграми процесів

Діаграма процесів (Рисунок 3.3) демонструє логіку взаємодії між основними вузлами інфраструктури (Elephant, Falcon, Suricata, Canary, Python, Router), користувачем (студентом) та системою сповіщень Telegram. Інформаційний обмін базується на HTTP REST запитах, протоколі Telnet, перехопленні RAW пакетів та UDP-мовності.

При подачі живлення система автоматично конфігурує робоче середовище:

- Elephant генерує випадкові IP-адреси для ролей Admin та Mirror, а також створює унікальні ключі та послідовність випробування;
- всі інші пристрої (ESP32) підключаються до мережі Wi-Fi, після чого виконують HTTP POST запит (/register) до Elephant, у відповідь вони отримують свої ключі, дозволені IP-адреси та необхідні змінні стану;
- Suricata ініціює HTTP GET запит до маршрутизатора для увімкнення дзеркалювання мережевого трафіку на створену Mirror IP.

Логіка проходження квесту студентом передбачає багатовекторну мережеву активність:

- розвідка: студент сканує мережу (Nmap) та прослуховує трафік (Wireshark), Suricata аналізує "сирі" пакети і у разі виявлення агресивного сканування (повторювані TCP RST) ініціює блокування студента, водночас студент має перехопити UDP-пакети з підказками від Suricata;
- взаємодія з сервісами: студент змінює власну IP-адресу для доступу до привілейованих панелей, наприклад, доступ до панелі Falcon (порт 8080) та тригера Python відкривається лише при маскуванні під Admin IP, з базою даних Elephant студент взаємодіє через імітований Telnet-клієнт, вводячи команди SQL для отримання схеми розташування ключів;
- фінальний етап: зібравши всі фрагменти ключів (через веб-сторінки, приховані SVG, перехоплені слова-паролі та спеціальні мережеві запити), студент

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

авторизується на прихованому Telnet-порту Python і вводить їх у вірній послідовності.

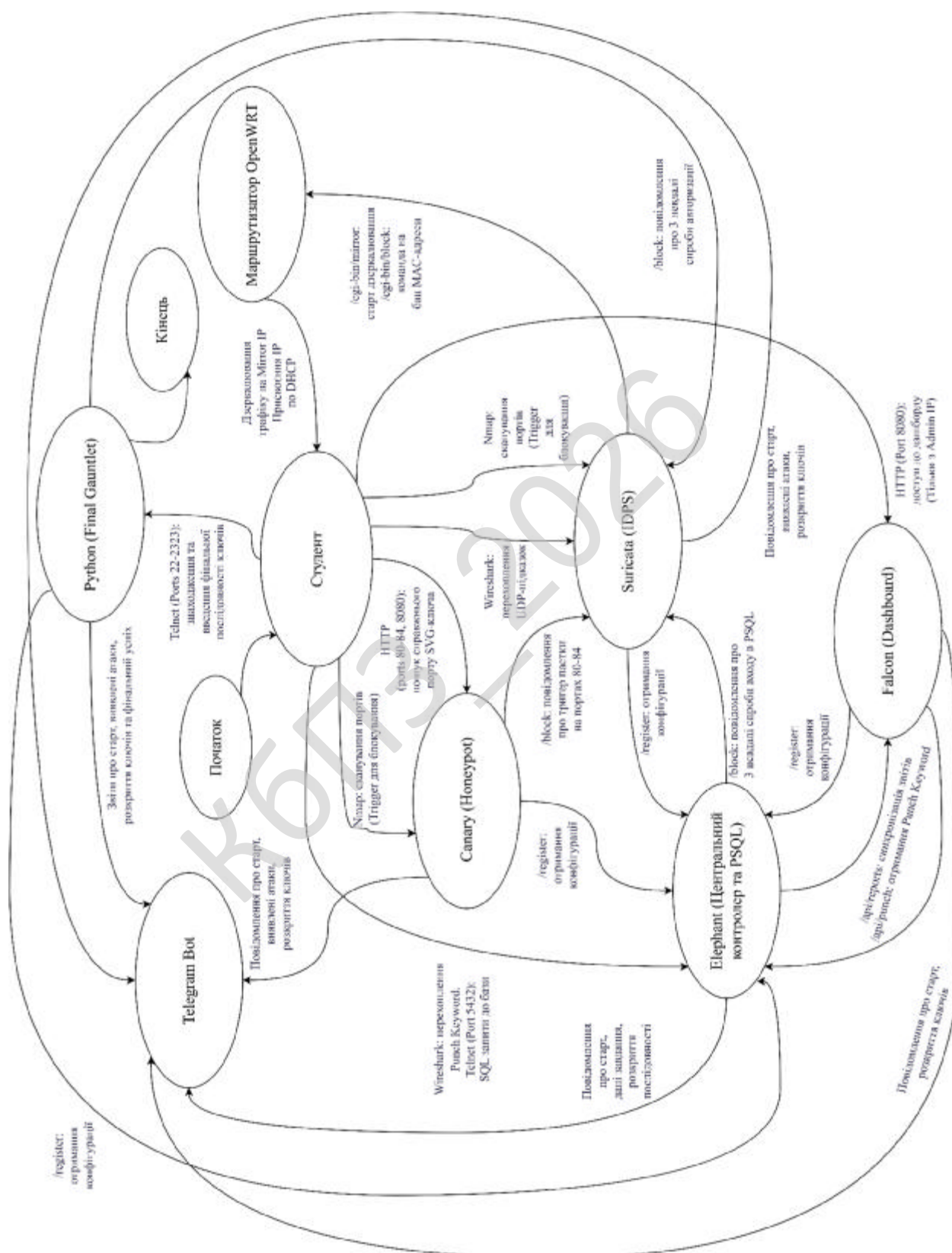


Рисунок 3.3 – Діаграма процесів

Процес обмеження доступу реалізовано у вигляді замкненого циклу:

а) якщо будь-який пристрій (Canary фіксує звернення на фейковий порт, Elephant фіксує 3 невірні паролі бази даних, Python фіксує помилки авторизації, або Suricata виявляє Nmap) фіксує порушення, він формує HTTP POST запит до API порту 80 на пристрої Suricata;

б) Suricata, як центральний вузол безпеки, відправляє HTTP-виклик до CGI-скрипта маршрутизатора (/cgi-bin/block/block?ip=...);

в) маршрутизатор за допомогою ARP-таблиці конвертує IP зломисника у MAC-адресу, додає її до "чорного списку" Wi-Fi мережі (macfilter='deny') та перезавантажує радіомодуль, фізично відключаючи студента від інфраструктури.

Для адміністративного контролю за ходом виконання лабораторної роботи кожен ESP32-мікроконтролер взаємодіє з Telegram API:

- при старті кожен вузол надсилає інформаційне зведення зі своєю IP-адресою, відкритими портами та згенерованим ключем;

- у разі виявлення загроз або порушень відправляються попередження про блокування учасників;

- після успішного завершення фінального випробування (Python), в Telegram надсилається повідомлення про успішне отримання Master Key конкретним учасником.

Розробка принципової схеми пристрою

Оскільки проєкт базується переважно на бездротових технологіях, то принципова електрична схема пристрою може бути максимально спрощеною. Система складається з 5 модулів ESP32 DevKit, які можуть живитися від звичайних USB-хабів (5V). Кожен мікроконтролер є автономним і не потребує додаткових фізичних датчиків. Їхня взаємодія здійснюється виключно через радіоканал (Wi-Fi 2.4 GHz) стандарту 802.11 b/g/n, який забезпечується роутером TP-LINK. Це робить систему надзвичайно мобільною та дешевою у розгортанні.

Але так як була можливість, було додано візуально-вібраційну взаємодію з користувачами.

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Було розроблено і роздуковано на 3D принтері моделі для кожного вузла системи, базуючись на тваринах, які є логотипом або маскотом сервіса, який імітує ESP32. Моделі було розроблено на основі кейса для ESP32 [8], сгенеровано за допомогою Gemini Nano Banana 1 [9] та Rodin Gen-1.5 (RLHF) [10], та допрацьовано і фіналізовано в Tinkercad [11].

Кожна модель вузла має однакову основу, яка складається з частини для ESP32 та кришки (Рисунок 3.4, Рисунок 3.5).

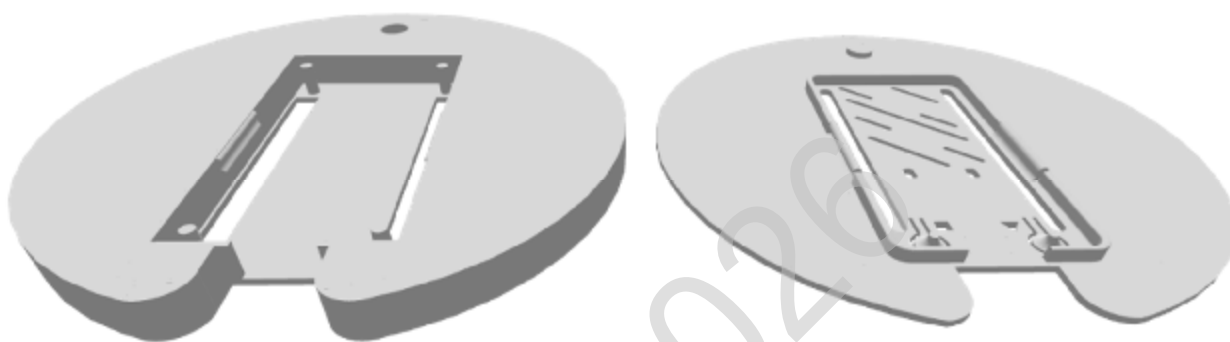


Рисунок 3.4 – Основа моделей для вузлів зверху

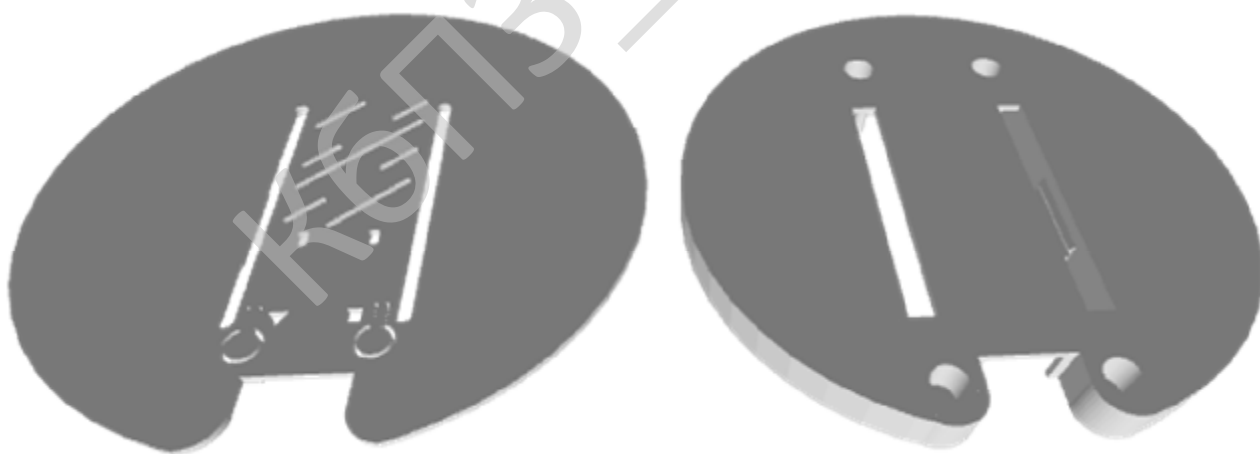


Рисунок 3.5 – Основа моделей для вузлів знизу

Верхня частина моделі являє собою голову тварини, імітацію сервіса якого виконує ESP32 (Рисунок 3.6). Ця частина з'єднується з основою за допомогою 4-х циліндричних кріплень. Всередині голова порожня з місцем для

вібромоторчика.

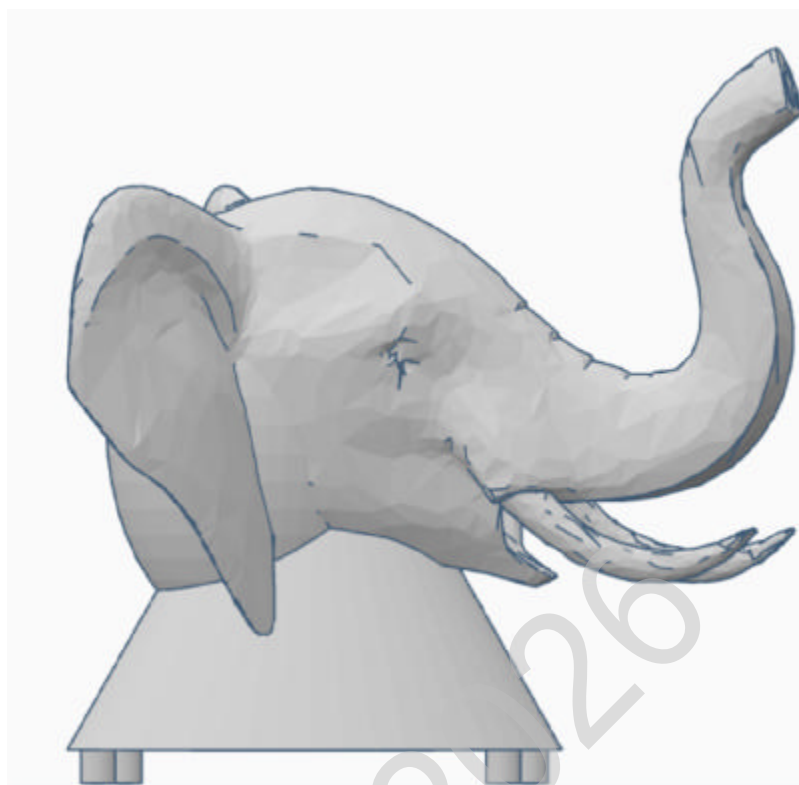


Рисунок 3.6 – Приклад верхньої частини для ESP32, яка імітує Postgres

Моделі було опубліковано на платформу Creality Cloud під назвою MissionZOO ESP32 Project [12].

Принципова схема кожного модуля на базі ESP32 у проєкті MissionZOO є достатньо простою, проте вона забезпечує необхідну гнучкість для поєднання програмної симуляції мережевих сервісів із фізичним відгуком на події в системі.

До складу кожного вузла входять:

- мікроконтролер ESP32: виступає ядром системи, що відповідає за обробку мережевих протоколів, виконання логіки «тварини» та генерацію керуючих сигналів для периферії;

- силовий комутаційний каскад: побудований на базі польового транзистора IRFZ44N [13], який виконує роль електронного ключа для керування потужним навантаженням;

– модуль фізичного відгуку (Motor/Vibrator): виконавчий механізм, що забезпечує вібро-індикацію при виявленні вразливостей або атак на сервіс.

Керування фізичним рівнем здійснюється через цифровий порт D15 мікроконтролера. Оскільки робочий струм вібромотора перевищує можливості GPIO-виводів ESP32, сигнал із порту подається на затвор (Gate) транзистора IRFZ44N. Транзистор замикає коло між негативним контактом мотора (M-) та загальною шиною (GND), тоді як позитивний контакт мотора (M+) підключений безпосередньо до лінії живлення 3V3. Це дозволяє не лише вмикати/вимикати пристрій, а й використовувати широтно-імпульсну модуляцію (PWM) для зміни інтенсивності вібрації.

Принципову схему пристрою зображено на рисунку 3.7.

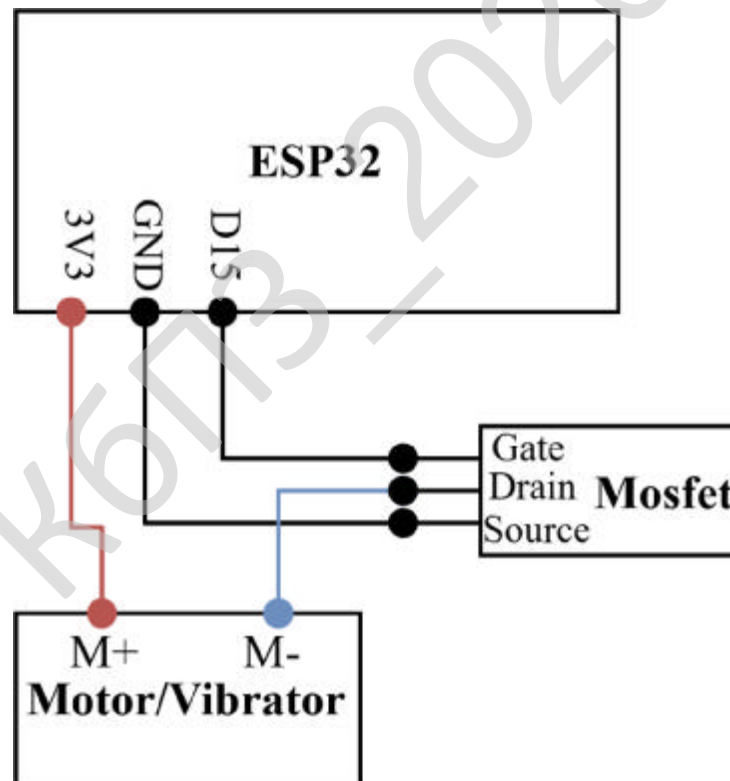


Рисунок 3.7 – Принципова схема пристрою

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ ТА ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЄКТНИХ І ПРОГРАМНИХ РІШЕНЬ

Розробка блок-схем та опис алгоритмів функціонування системи

Розглянемо етап програмної реалізації бакалаврської дипломної роботи. Для перевірки правильності реалізації проєктних рішень та логіки мережевої взаємодії мікроконтролерів було створено детальні блок-схеми роботи системи. Перед їх безпосереднім розглядом необхідно провести роз'яснення, який саме тип блок-схем використовується та за якими правилами вони побудовані.

Блок-схема – це графічне представлення алгоритму, задачі або процесу для її аналізу та розв'язування за допомогою спеціальних символів (геометричних образів), які позначають такі елементи, як операції, потік управління, дані тощо. Відповідно до стандартів оформлення, блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки даних) – прямокутником, блок прийняття рішень (розгалуження) – ромбом, а початок та кінець алгоритму – еліпсом (термінатором).

У контексті розробки інформаційних технологій функціональна схема алгоритму складається з блоків, які являють собою конструктивно відособлені частини програмного коду (функції, методи або окремі корутини), що виконують певні задачі. Основні елементи схем алгоритму – це термінатор, процес, рішення, зумовлений процес (підпрограма) та з'єднувач.

Термінатор відображає вхід із зовнішнього середовища або вихід з неї (запуск мікроконтролера або перехід у режим очікування). Процес – це виконання однієї або кількох операцій (наприклад, генерація IP-адреси, парсинг JSON-файлу). Рішення показує функцію перемикального типу з одним входом і двома або більше альтернативними виходами (наприклад, перевірка правильності

введеного пароля), з яких тільки один може бути обраний після обчислення умов. Якщо виходів два чи більше, кожен шлях позначається лінією, що виходить з відповідної вершини ромба, а поруч записуються результати обчислень («Так» або «Ні»).

Блок-схеми показують весь процес роботи системи з її підсистемами та доводять правильність обраних архітектурних рішень. Від точності деталізації блок-схеми залежить стабільність результату всієї програми.

При виборі початкової точки відліку для побудови схем було враховано, що виходячи з вибору мови програмування та технічних засобів, програма базується на асинхронній подієво-орієнтованій моделі (Event-Driven Architecture). Це вимагає глибокої декомпозиції задач на незалежні процеси, що працюють у єдиному циклі подій (Event Loop).

Програмне забезпечення навчального комплексу MissionZOO побудоване на базі мови MicroPython (v1.26.1) для мікроконтролерів ESP32 та сценаріїв командної оболонки Shell (sh) для маршрутизатора OpenWRT. Особливістю реалізації є використання асинхронної моделі програмування (бібліотека uasyncio), що дозволяє вузлам виконувати декілька мережевих завдань одночасно. Наприклад, мікроконтролер може підтримувати роботу HTTP API та паралельно, у фоновому режимі, моніторити "сирі" (Raw) TCP-пакети в мережі.

Система MissionZOO побудована за принципом розподіленої мережевої взаємодії, де кожен вузол (ESP32) виконує свою специфічну роль у загальному сценарії (симуляція сервісів, виявлення вторгнень, створення пасток тощо).

Розглянемо детально алгоритм роботи кожного вузла.

Усі пристрої мають спільний базовий етап – ініціалізацію та реєстрацію (Bootstrap), після якого алгоритми розгалужуються відповідно до призначення «тварини».

Етап ініціалізації є спільним для всіх вузлів (окрім вузла Elephant, який виступає контролером системи). При запуску кожен пристрій проходить процедуру підключення до центральної інфраструктури:

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

а) підключення до Wi-Fi: вузол підключається до визначеної точки доступу "MissionZOO".

б) реєстрація у системі: вузол надсилає HTTP POST-запит (Bootstrap) на центральний контролер (Elephant), передаючи свої метадані (назву та відкриті порти).

в) отримання контексту: у відповідь контролер надає глобальні параметри гри: Admin IP, Mirror IP (для дзеркалювання трафіку), а також частини ключів, необхідні для проходження квесту, в залежності від пристрою, з якого йде запит.

Розглянемо алгоритм роботи вузла Elephant (центральний контролер).

Вузол «Слон» (Elephant) виступає в ролі бази даних та головного API-сервера. Його основний асинхронний цикл паралельно обробляє два типи мережевих запитів:

а) HTTP API (порт 80): приймає запити «/register» для реєстрації інших вузлів та видачі їм ключів (при цьому система автоматично зберігає їхні IP- та MAC-адреси у внутрішніх реєстрах для моніторингу), а також запити «/report» для запису логів виявлених мережевих атак;

б) симуляція PostgreSQL (порт 5432): достовірно імітує роботу бази даних:

1) якщо користувач намагається підключитися, система перевіряє пароль;

2) у разі трьох послідовних невдалих спроб авторизації (Bruteforce), Elephant відправляє сигнал на вузол Suricata для негайного апаратного блокування по IP-адресі атакуючого;

3) якщо авторизація успішна, відкривається інтерактивна оболонка, яка дозволяє виконувати SQL-команди (наприклад, \l, \dt, SELECT * FROM keys) для отримання підказок та точної послідовності введення ключів. Дана послідовність формується одноразово і є статичною для поточної сесії, тому її необхідно суворо дотримуватися на фінальному вузлі Python без жодних відхилень.

Блок-схему роботи алгоритму вузла «Elephant» зображено на рисунку 4.1.

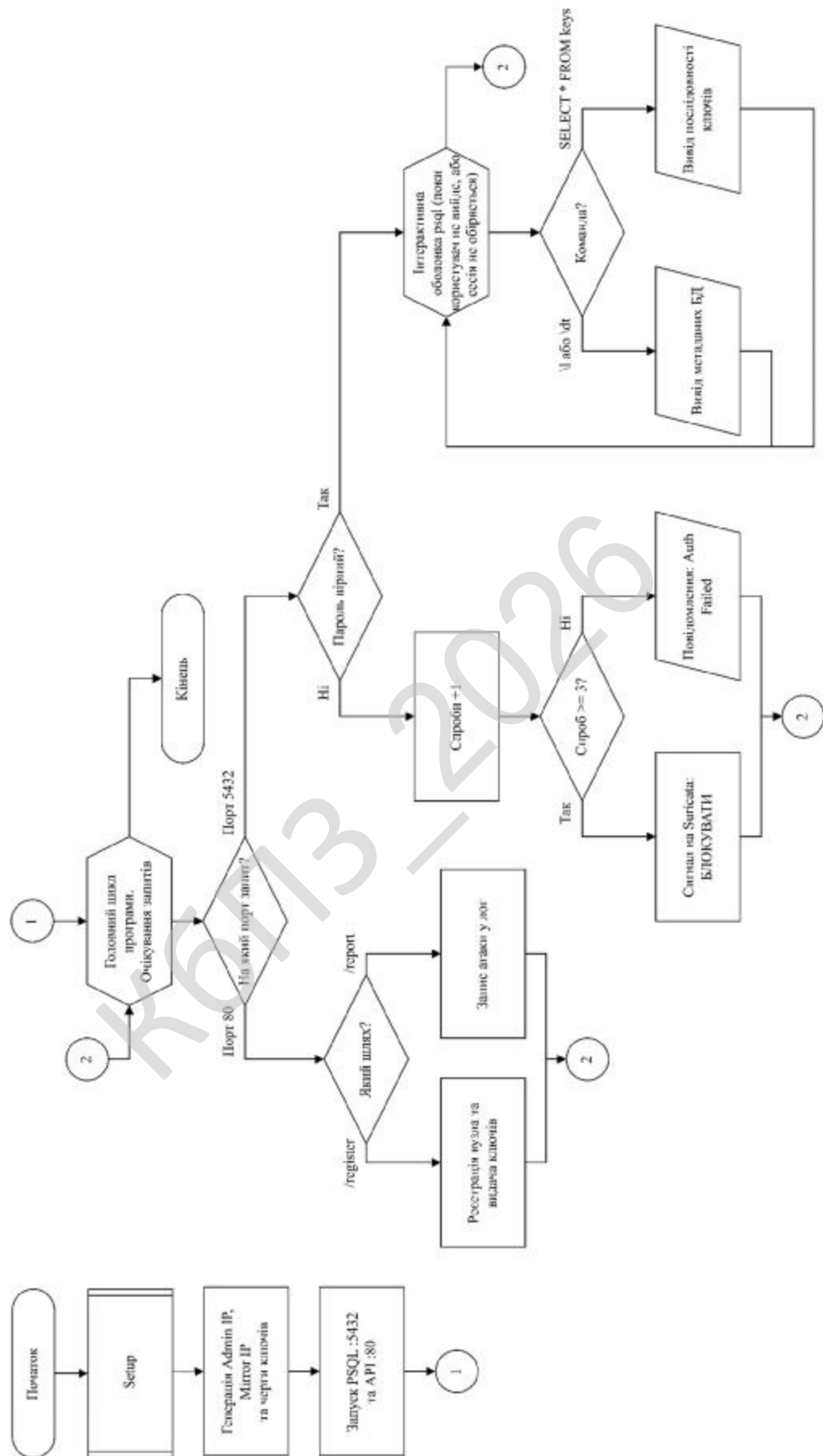


Рисунок 4.1 – Блок-схема роботи алгоритму вузла «Elephant»

Розглянемо тепер алгоритм роботи вузла Suricata (система виявлення вторгнень - IDPS).

Вузол «Сурикат» (Suricata) відповідає за моніторинг мережі та захист від несанкціонованого сканування. Після ініціалізації він надсилає команду на маршрутизатор OpenWRT для старту віддзеркалення трафіку (Mirroring). Основний цикл включає:

а) фоновий моніторинг (Raw Sockets): глибоке перехоплення всіх TCP-пакетів на мережевому рівні; якщо виявляється пакет із встановленим прапорцем RST (скидання з'єднання), лічильник спроб для цього IP збільшується. Якщо фіксується 3 або більше RST-пакетів за 5 секунд (що є типовою сигнатурою агресивного Nmap-сканування, зокрема при використанні методу SYN Stealth, коли система автоматично скидає напіввідкриті з'єднання), спрацьовує тригер блокування;

б) розсилка підказок: кожні 60 секунд таймер ініціює відправку широкомовних UDP-повідомлень (на 255.255.255.255) з інформацією про Mirror IP, а також цільових UDP-пакетів на Mirror IP з інформацією про Admin IP;

в) обробка порту 7000: якщо користувач надсилає правильну кодову фразу «Give me a key, please», йому видається частина ключа, якщо ж у запиті виявлено підпис (сигнатуру) Nmap, відбувається негайне блокування;

г) механізм блокування: при спрацюванні будь-якого тригера, Suricata спочатку перевіряє, чи не входить IP до білого списку (Whitelist), якщо ні – надсилається детальне сповіщення у Telegram-бот із чіткою вказівкою того, яка саме «тварина» ініціювала блокування, робиться відповідний запис у базу даних Elephant, і формується HTTP-запит до маршрутизатора (/cgi-bin/block). Цей запит ініціює внесення пристрою до «чорного списку» та апаратне блокування MAC-адреси порушника на рівні Wi-Fi мережі, фізично відключаючи його від інфраструктури стенда.

Блок-схему роботи алгоритму вузла «Suricata» зображено на рисунку 4.2.

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

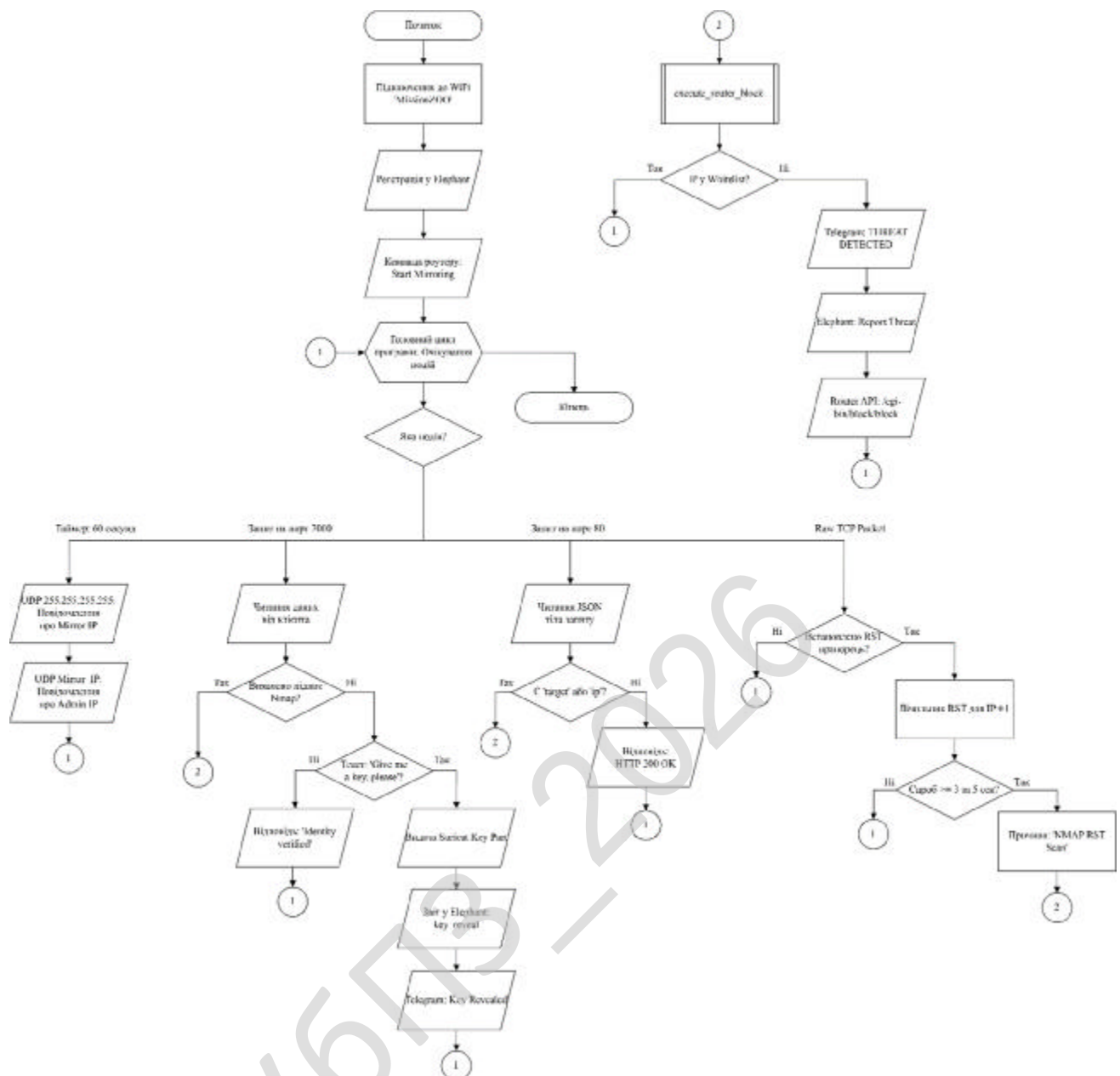


Рисунок 4.2 – Блок-схема роботи алгоритму вузла «Suricata»

Перейдемо до алгоритму роботи вузла Canary (мережева пастка).

Вузол «Канарка» (Canary) є Honeypot-системою, що дезорієнтує атакуючого. Алгоритм включає:

а) розподіл портів: при старті вузол випадковим чином обирає один «справжній» порт (True Port) у діапазоні 80-84, залишаючи інші порти як активні пастки;

б) генерація фейку (порт 8080): окремо працює порт 8080 (Decoy), який є повністю фейковим, створений для того, щоб ввести атакуючого в оману,

відображаючи хибні підказки і неіснуючу частину ключа. Це змушує студента витратити дорогий час на аналіз тупикових векторів атаки (ефект Rabbit Hole).

в) логіка пастки: якщо користувач звертається до будь-якого порту-пастки в діапазоні 80-84, Canary негайно сигналізує вузлу Suricata про вторгнення (через внутрішній HTTP API), і користувач отримує сторінку «SECURITY VIOLATION» безпосередньо перед апаратним блокуванням.

г) логіка справжнього порту: якщо користувач заходить на правильний порт, система просить логін та пароль, після трьох невдалих спроб (Bruteforce) ініціюється блокування. При правильному введенні даних користувач отримує доступ до SVG-поля, де у вихідному коді приховано ключ (текстові елементи фізично розміщено за межами видимих координат viewBox, тому їх можна виявити лише через глибоке інспектування DOM-дерева сторінки браузера), та підказок стосовно інших вузлів мережі. Блок-схему роботи алгоритму вузла «Canary» зображено на рисунку 4.3.

Розглянемо тепер алгоритм роботи вузла Falcon (панель моніторингу). Вузол «Сокіл» (Falcon) симулює адміністративний дашборд безпеки. Алгоритм включає:

а) синхронізація: кожні 60 секунд вузол асинхронно звертається до контролера Elephant (за допомогою GET-запитів) для отримання актуальних логів атак та кодового слова (Punch Keyword);

б) контроль доступу (Порт 8080): при зверненні до веб-сервера жорстко перевіряється IP-адреса клієнта на рівні обробки сокетів. Якщо вона не збігається з Admin IP (який студент має підробити за допомогою IP Spoofing, модифікуючи власні мережеві налаштування ОС), доступ негайно забороняється зі статусом помилки 403 Forbidden;

в) видача даних: при успішній IP-авторизації відображаються детальні логи атак (що дає змогу студенту побачити власні «гучні» сліди в системі та оцінити рівень своєї скритності), прогрес по знаходженню частин ключа та інтерактивне

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

поле. Якщо користувач вводить перехоплений раніше правильний Punch Keyword, йому розкривається частина ключа Falcon.

Блок-схему роботи алгоритму вузла «Falcon» зображено на рисунку 4.4.

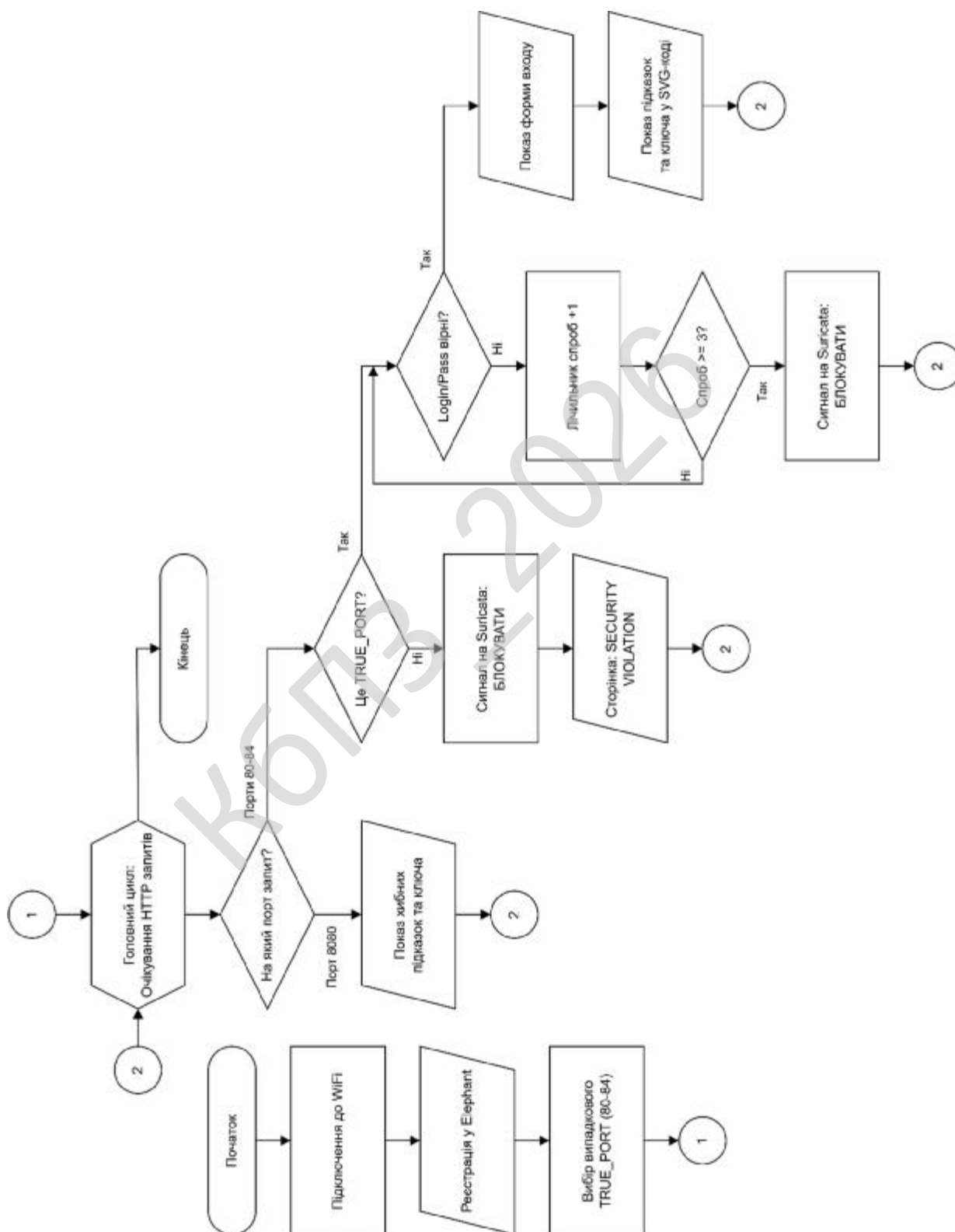


Рисунок 4.3 – Блок-схема роботи алгоритму вузла «Canary»

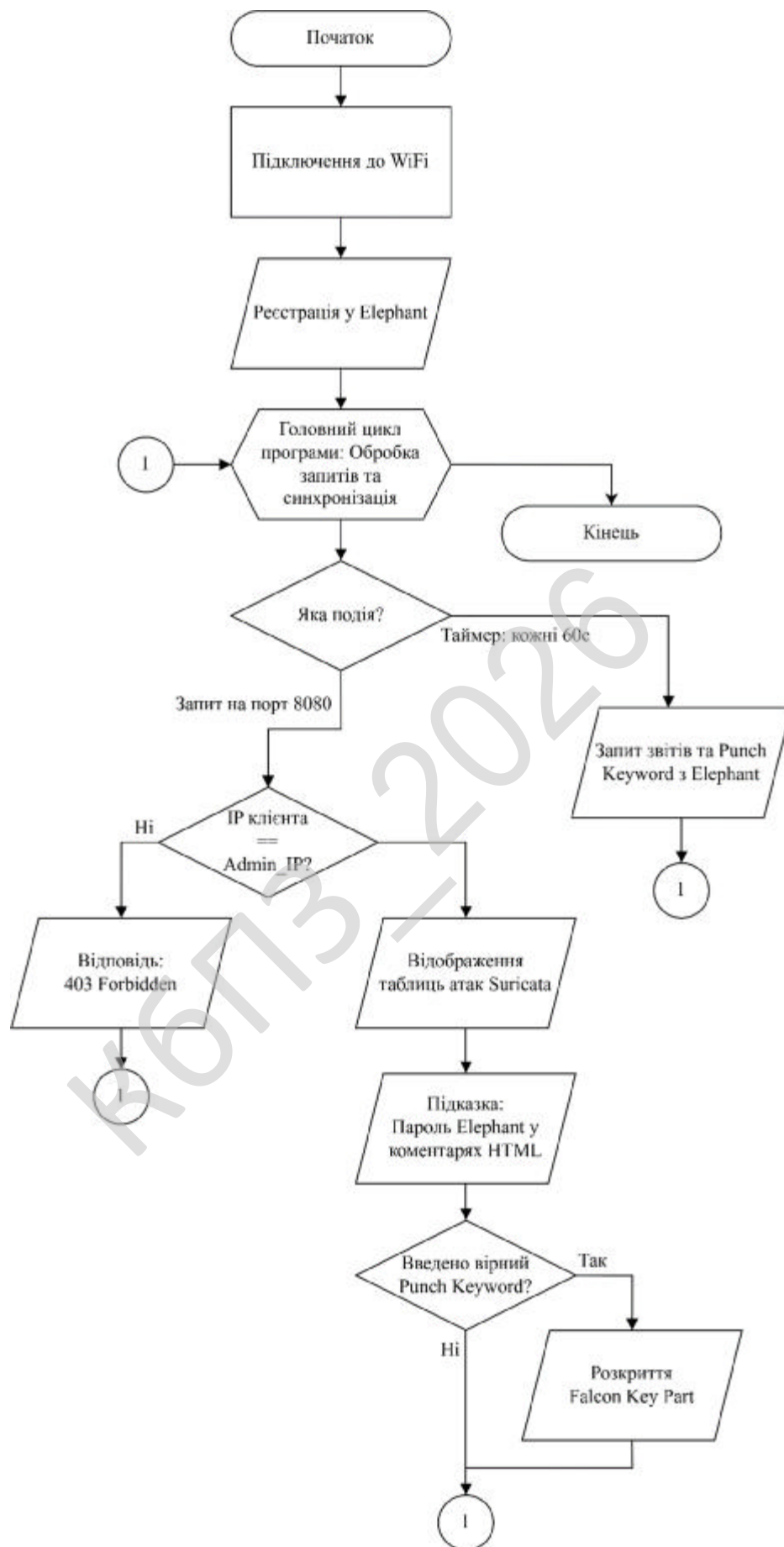


Рисунок 4.4 – Блок-схема роботи алгоритму вузла «Falcon»

Перейдемо до алгоритму роботи вузла Python (фінальний квест).

Вузол «Пітон» (Python) є кінцевою точкою всього навчального сценарію.

Алгоритм його роботи включає:

а) рандомізація ролей: при кожному запуску пристрою порти 22, 23, 2222 та 2323 випадковим чином отримують унікальні ролі: Trigger, Secret, Hidden та Fake. Такий підхід унеможливорює просте заучування маршруту атаки студентами для наступних сесій;

б) активація (Trigger та Secret): ці порти жорстко вимагають підключення виключно з Admin або Whitelisted IP-адрес. При правильному зверненні до порту Trigger, розблоковується доступ до порту Hidden. При зверненні до порту Secret – видаються облікові дані (логін та пароль) для входу до порту Hidden;

в) фінальна валідація (Hidden Port): якщо порт розблоковано, студент повинен успішно авторизуватися на ньому через Telnet-клієнт. Далі система вимагає поетапно ввести всі знайдені частини ключів від попередніх тварин. Важливо зазначити, що вузол Python не генерує послідовність перевірки динамічно; він точно і невідступно слідує еталонній послідовності, яку отримує від компонента Elephant під час старту. Будь-яке відхилення від цього точного порядку вважається хибним;

г) завершення: у разі помилки при введенні хоча б одного фрагмента, з'єднання миттєво розривається, а прогрес послідовності скидається, змушуючи починати валідацію заново. При успішному введенні всіх ключів студент перемагає: йому видається Master Key, відображений всередині візуального ASCII-арта безпосередньо в терміналі, та автоматично надсилається звітне повідомлення в Telegram викладачу про успішне виконання завдання.

Блок-схему роботи алгоритму вузла «Python» зображено на рисунку 4.5.

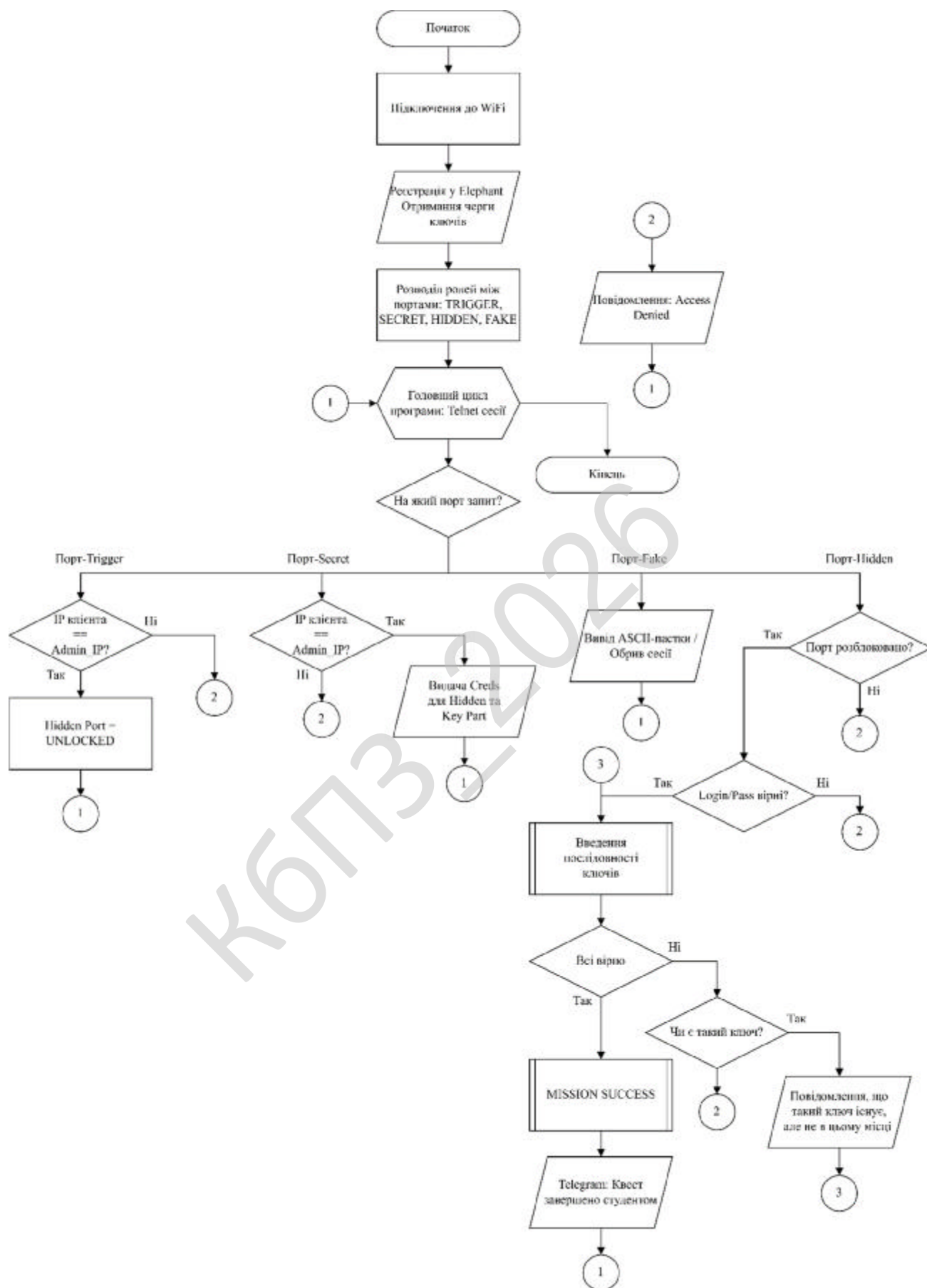


Рисунок 4.5 – блок-схема роботи алгоритму вузла «Python»

Опис бібліотечних функцій

Для стабільної роботи в умовах обмеженої пам'яті ESP32 використано наступні модулі MicroPython:

- `uasyncio`: дозволяє створювати асинхронні таски (`create_task`), наприклад це дозволяє одночасно тримати відкритими 4 Telnet-сервери на різних портах [14];
- `socket`: використовується як для стандартних TCP-з'єднань, так і для Raw-сокетів (`socket.AF_INET`, `socket.SOCK_RAW`, 6) у Suricata для низькорівневого аналізу пакетів [15];
- `network`: керує Wi-Fi стеком; функція `wlan.ifconfig()` використовується для динамічного призначення мережевих параметрів [16];
- `urequests`: забезпечує взаємодію між вузлами за протоколом HTTP – через цей модуль Elephant отримує звіти про атаки, а вузли – команди на блокування [17];
- `ujson`: використовується для обробки конфігурацій та ігрових станів, що передаються у форматі JSON [18];
- `machine`: керує «залізом»; функція `machine.Pin` активує MOSFET, що керує вібромотором, а `machine.reset()` виконує перезавантаження пристрою у разі критичних помилок [19].

Опис власних функцій

Власні функції реалізують унікальну логіку MissionZOO. Розглянемо деякі з них в залежності від вузла.

Для вузла Elephant розглянемо `handle_psql()` та `initialize_game()`:

- `handle_psql()`: повністю емулює поведінку PostgreSQL, вона обробляє команди `\l`, `\dt`, `SELECT * FROM keys` та реалізує систему авторизації за паролем, що зберігається в пам'яті (див. додаток А `main.py` (для Elephant));
- `initialize_game()`: генерує випадкові умови сесії: Admin IP, Mirror IP та «Punch Keyword» (пароль для Falcon) (див. Додаток А `config_elephant.py`).

Для вузла Suricata розглянемо `execute_router_block(target_ip, reason)` та `trigger_router_mirror()` (див. Додаток А `main.py` для Suricata):

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

– `execute_router_block(target_ip, reason)`: відправляє HTTP GET запит на шлюз OpenWRT, викликаючи CGI-скрипт блокування;

– `trigger_router_mirror()`: активує функцію віддзеркалення трафіку (TEE) на маршрутизаторі, щоб студенти могли аналізувати пакети через Wireshark.

Для вузла Canary розглянемо `handle_fake_page()` та SVG Key Masking логіку (див. Додаток А `main.py` для Canary):

– `handle_fake_page()`: якщо студент звертається до неправильного порту (пастки), функція видає сторінку "SECURITY VIOLATION" та негайно ініціює блокування через Suricat;

– SVG Key Masking: спеціальна логіка приховування ключа всередині SVG-коду з координатами поза видимою областю (`viewBox`), що змушує студента досліджувати вихідний код сторінки.

Для вузла Falcon розглянемо `handle_dashboard()` та `task_sync_punch()` (див. Додаток А `main.py` для Falcon):

– `handle_dashboard()`: генерує динамічну HTML-сторінку з таблицями атак, вона перевіряє IP відвідувача і надає доступ лише якщо IP збігається з `admin_ip`;

– `task_sync_punch()`: фонові задача, яка кожен хвилину запитує «Punch Keyword» з Elephant.

Також розглянемо скрипти для маршрутизатора (див. Додаток Б):

– `mirror (CGI)`: використовує `iptables -t mangle -A FORWARD -j TEE --gateway ...` для перенаправлення копій пакетів на вузол моніторингу.

– `block (CGI)`: використовує `uci` для керування `maclist` у налаштуваннях бездротової мережі та виконує команду `/sbin/wifi reload` для миттєвого відключення пристрою-порушника.

Захист розробленого програмного забезпечення

Відповідно до Закону України «Про авторське право і суміжні права», розроблене в межах дипломного проєкту програмне забезпечення належить до

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

об'єктів інтелектуальної власності та підлягає правовому захисту. Створена система MissionZOO є розподіленим апаратно-програмним комплексом, що складається з центрального керуючого вузла (маршрутизатора з прошивкою OpenWRT), базового сервера бази даних (Elephant) та мережі інтерактивних клієнт-серверних вузлів на базі мікроконтролерів ESP32, які виконують функції симуляції сервісів, виявлення атак та апаратного реагування.

Архітектура системи побудована за децентралізованим гібридним принципом і поєднує модель REST API, інтерактивні сесії Telnet та низькорівневий аналіз трафіку (Raw Sockets). Вузол Elephant виступає центральним інформаційним хабом, який забезпечує реєстрацію інших пристроїв, генерацію ігрових сесій та збереження логів. Інші пристрої на базі ESP32 (Suricat, Canary, Falcon, Python) виконують роль як клієнтів (отримуючи конфігурації), так і серверів, відкриваючи мережеві порти для взаємодії зі студентами.

Обмін даними між компонентами реалізовано через локальну ізольовану бездротову мережу Wi-Fi "MissionZOO". При цьому застосовано декілька механізмів передачі інформації: TCP-з'єднання для емуляції баз даних та консольного доступу, HTTP POST/GET-запити для внутрішнього API між пристроями, а також механізм широкомовних UDP-пакетів (Broadcast) для періодичної розсилки мережових підказок студентам. Взаємодія з маршрутизатором OpenWRT здійснюється через виклик системних CGI-скриптів.

Програмна частина мікроконтролерів ESP32 реалізована мовою MicroPython з використанням асинхронної моделі обробки подій на базі бібліотеки uasyncio. Асинхронна реалізація є критично важливою для проєкту, оскільки дозволяє вузлам, наприклад Suricata, одночасно виконувати перехоплення пакетів (сніфінг) у фоновому режимі, обслуговувати HTTP-запити на блокування та відправляти телеметрію без блокування основного потоку виконання.

Апаратна клієнтська частина на ESP32 реалізує логіку фізичного відгуку. Замість традиційних засобів візуалізації (дисплеїв), система використовує силовий комутаційний каскад на базі MOSFET-транзистора IRFZ44N та вібромотора. При

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

успішному проходженні завдань (введенні правильної послідовності ключів) або критичних подіях, мікроконтролер генерує керуючий сигнал на порт GPIO 15, що забезпечує тактильний та звуковий зворотний зв'язок для користувача, сигналізуючи про успішний злам або потрапляння у пастку.

Передача внутрішньосистемних даних організована у вигляді структурованих повідомлень формату JSON із використанням кодування UTF-8. Для десеріалізації та створення таких повідомлень на стороні ESP32 застосовується вбудована бібліотека `ujson`. Це дозволяє ефективно обмінюватися складними структурами даних: передавати конфігурації портів, розподіл ролей, IP-адреси адміністраторів та логіювати причини мережових блокувань.

Мережевий функціонал системи реалізовано з використанням базових модулів `network`, `socket` та `urequests`, які забезпечують повний цикл взаємодії. Особливістю розробки є використання «сирих сокетів» (`socket.SOCK_RAW`) для реалізації системи виявлення вторгнень (IDPS). Завдяки цьому мікроконтролер здатний аналізувати прапорці TCP-пакетів (зокрема RST) для виявлення спроб автоматизованого сканування мережі утилітою Nmap.

Зовнішній моніторинг інфраструктури здійснюється за допомогою інтеграції з Telegram API. Система автоматично формує та надсилає звіти до Telegram-бота інструктора про ключові події: старт стенду, успішне проходження студентом завдань або виявлення порушень (брутфорс, сканування). Передача здійснюється через HTTP POST-запити, що забезпечує безперервний контроль за процесом виконання лабораторної роботи у реальному часі.

З позиції інформаційної безпеки розроблена система є унікальною, оскільки її метою є навмисна імітація вразливостей. Проте захист самої інфраструктури стенду від критичних збоїв є багаторівневим. По-перше, система функціонує в межах закритої локальної Wi-Fi мережі без доступу до глобальної мережі Інтернет (за винятком вихідного каналу для Telegram-сповіщень). Це гарантує, що небезпечні експерименти студентів не вийдуть за межі навчального полігону і не зашкодять реальній інфраструктурі закладу.

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

По-друге, розроблено алгоритм апаратного реагування на загрози: при виявленні нелегітимних дій (потрапляння у Honeypot-пастку Canary, перебір паролів до Elephant або агресивне сканування), вузол Suricata ініціює виклик CGI-скрипта на маршрутизаторі. Маршрутизатор за допомогою утиліти `uci` та фаєрволу миттєво додає MAC-адресу порушника до списку блокування (MAC filter deny) та перезавантажує Wi-Fi інтерфейс, фізично відключаючи атакуючого від мережі.

Разом із цим, архітектура системи передбачає можливість поступового посилення безпеки адміністративної частини, для ускладнення завдання для студентів. Зокрема, у майбутньому комунікацію між панеллю управління Falcon та центральним вузлом Elephant може бути переведено на HTTPS із використанням TLS-шифрування. Доцільним також є впровадження криптографічних токенів для внутрішніх API-запитів між мікроконтролерами, щоб запобігти можливості підробки керуючих команд студентами, які застосовують техніки Man-in-the-Middle (MitM).

Результати тестування підтвердили високу стабільність програмного забезпечення в умовах активного мережевого навантаження. Система коректно обробляє одночасні Telnet-сесії, вчасно виявляє TCP-аномалії та безперебійно взаємодіє з маршрутизатором.

Програмне забезпечення побудоване за модульним принципом, що спрощує його підтримку та розгортання. Скрипти обробки мережевих підключень, логіки ігрових квестів та взаємодії з апаратною частиною розділені, що забезпечує гнучкість системи. Така архітектура дозволяє легко масштабувати MissionZOO, додаючи нові мікроконтролери-«тварини» з новими типами вразливостей без необхідності переписування ядра системи.

5 ВПРОВАДЖЕННЯ СИСТЕМИ КІБЕРБЕЗПЕКИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Впровадження розробленого апаратно-програмного навчального комплексу MissionZOO, що поєднує мережу мікроконтролерів ESP32 та маршрутизатор OpenWRT із системою сповіщень через Telegram, є логічним завершенням дипломного проєкту. Цей етап передбачає перенесення розробленого рішення з тестового середовища у реальні умови освітнього процесу (лабораторії кібербезпеки). Основна мета полягає у створенні стабільного, керованого та безпечного полігону для відпрацювання студентами навичок етичного хакінгу, виявлення вразливостей (CTF) та розуміння принципів роботи систем виявлення вторгнень (IDPS).

Процес інтеграції передбачає адаптацію системи до умов комп'ютерного класу, включаючи жорстку ізоляцію локальної бездротової мережі Wi-Fi, розгортання скриптів управління трафіком на маршрутизаторі, налаштування модулів фізичного відгуку (вібромоторів через MOSFET-ключі) та конфігурацію Telegram-сповіщень для інструктора. Особлива увага приділяється перевірці стабільності роботи під час імітації реальних мережевих атак (DDoS, Bruteforce, Nmap-сканування), коректності автоматичного блокування порушників та відновленню функціонування вузлів після завершення ігрових сесій. Такий підхід дозволяє виявити потенційні обмеження ще до повноцінного проведення лабораторних занять і забезпечити надійність стенду при одночасній роботі групи студентів.

Запропоноване рішення має гнучкий спектр застосування. У першу чергу, це освітнє середовище університетів та профільних курсів із кібербезпеки. Завдяки використанню доступних мікроконтролерів ESP32, система забезпечує реалістичний досвід роботи з мережевим трафіком без ризику пошкодження реальної IT-інфраструктури закладу. Гнучкість архітектури дозволяє адаптувати

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

MissionZOO під різні рівні складності: від базового сканування портів до складних атак типу Man-in-the-Middle. Завдяки модульній децентралізованій структурі можливе легке додавання нових «тварин» (вразливих сервісів) без суттєвих змін базової логіки контролера Elephant чи системи захисту Suricata.

В основі впровадження лежать принципи ізоляції, автоматизації та інтерактивності. Методика впровадження передбачає реалізацію комплексу послідовних етапів, спрямованих на забезпечення коректної роботи апаратної та програмної частин системи:

- підготовку апаратно-програмного середовища;
- налаштування локальної мережі Wi-Fi та маршрутизатора OpenWRT;
- конфігурацію Telegram-сповіщень;
- прошивку та налаштування розподілених ESP32-пристроїв;
- комплексне тестування мережесих атак та захисту;
- інтеграцію системи в лабораторне середовище;
- підготовку методичних матеріалів та навчання користувачів;
- моніторинг сесій та технічний супровід;
- оновлення функціоналу та масштабування зоопарку;
- аналіз освітньої ефективності впровадження.

Підготовка апаратно-програмного середовища

На початковому етапі здійснюється збірка фізичного стенда MissionZOO. Апаратна частина включає підготовку мікроконтролерів ESP32, комутацію силових каскадів на базі транзисторів IRFZ44N та підключення вібромоторів до порту GPIO 15. Виконується перевірка надійності контактів та забезпечення стабільного живлення 3.3V/5V для безперебійної роботи всіх вузлів під навантаженням. Програмна складова передбачає підготовку середовища для роботи з MicroPython (розширення у VS Code). Виконується завантаження базової прошивки MicroPython (v1.26.1) на кожен ESP32. Окремо готується маршрутизатор, на який встановлюється сумісна версія OpenWRT та

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

завантажуються розроблені CGI-скрипти (/cgi-bin/block та /cgi-bin/mirror), необхідні для управління трафіком на фізичному рівні.

Налаштування локальної мережі Wi-Fi та OpenWRT

Для гарантування безпеки навчального закладу розгортається повністю ізольована мережа "MissionZOO". На маршрутизаторі OpenWRT налаштовується SSID та DHCP-сервер. Важливою умовою є заборона маршрутизації трафіку з цієї мережі в глобальний Інтернет (за винятком конкретних дозволених портів для API Telegram). Особлива увага приділяється налаштуванню фаєрволу. Перевіряється робота утиліти iptables для реалізації функції TEE (Traffic Mirroring), яка дозволяє дублювати трафік на Mirror IP для аналізу студентами у Wireshark. Також тестується механізм macfilter, який забезпечує апаратне відключення атакуючого пристрою (через утиліту wifi reload) за командою від системи виявлення вторгнень.

Конфігурація Telegram-сповіщень

Telegram у проєкті MissionZOO виконує роль одностороннього каналу телеметрії для викладача (інструктора). На цьому етапі створюється Telegram-бот через BotFather, генерується BOT_TOKEN та визначається CHAT_ID групи управління. Ці параметри інтегруються у конфігураційні файли ESP32. Перевіряється логіка доставки сповіщень: бот повинен миттєво інформувати інструктора про старт системи, виявлення загроз (наприклад, "NMAP Scan Detected" або "Bruteforce Canary"), а також про успішне проходження фінального квесту студентом. Використання асинхронної бібліотеки urequests дозволяє надсилати звіти без блокування основних ігрових процесів мікроконтролера.

Налаштування та програмування ESP32

На цьому етапі кожен мікроконтролер отримує свою унікальну логіку (стає конкретною «твариною»). Код завантажується у відповідні пристрої: Elephant (контролер і PSQL), Suricat (IDPS), Canary (Honeypot), Falcon (Dashboard) та Python (Gauntlet). Налаштовується логіка Bootstrap-реєстрації: після підключення до Wi-Fi кожен вузол повинен автоматично знайти Elephant та отримати ігрові параметри (Admin IP, Mirror IP). Перевіряється генерація випадкових портів та ролей.

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

Налаштовується робота сирих сокетів (SOCK_RAW) для вузла Suricata, щоб він міг безперешкодно аналізувати RST-прапорці у мережі. Також тестується робота апаратного переривання: при виклику відповідної функції ESP32 має успішно активувати ШІМ-сигнал для вібромотора.

Тестування системи

Тестування є ключовим етапом, оскільки імітує реальні дії студентів-хакерів. Проводиться комплексна перевірка:

- тестування роботи вузлів: виконується сканування мережі утилітою Nmap, перевіряється, чи Suricata виявляє аномальну кількість RST-пакетів та чи коректно надсилає команду блокування на маршрутизатор, здійснюється навмисне підключення до портів-пасток вузла Canary, щоб переконатися у миттєвому блокуванні порушника та генерації сторінки SECURITY VIOLATION, та інші дії;

- тестування логіки квесту: перевіряється процес отримання ключів через SQL-ін'єкції або підбір, IP-спуфінг для доступу до панелі Falcon та фінальне послідовне введення ключів на вузлі Python;

- тестування відмовостійкості: оцінюється здатність системи продовжувати роботу при агресивному навантаженні.

Інтеграція в експлуатаційне середовище

Після тестування фізичний стенд MissionZOO встановлюється в лабораторії кібербезпеки. Система підключається до стаціонарного живлення. Маршрутизатор розміщується так, щоб забезпечити оптимальне покриття Wi-Fi для робочих місць студентів. Виконується остаточне налаштування робочих параметрів. У навчальному середовищі рішення застосовується як практична лабораторна робота. Студенти отримують завдання підключитися до мережі, провести розвідку, знайти вразливі сервіси та зібрати всі ключі, уникаючи при цьому виявлення та блокування автоматизованими системами захисту.

Навчання користувачів

Впровадження вимагає інструктажу як для викладачів, так і для студентів. Викладачам надаються інструкції з керування маршрутизатором (скидання

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

заблокованих MAC-адрес) та моніторингу прогресу через Telegram-бот. Для студентів готується методичне забезпечення. Згідно з педагогічною концепцією проекту, студентам надається не прямий алгоритм рішення, а структуроване завдання з базовими порадами (наприклад, рекомендації використовувати Packet Sender, Nmap, Wireshark або підключення до баз даних через psycopg2). Це стимулює самостійний пошук та аналіз індикаторів компрометації. Студентів також інформують про правила етичного хакінгу та межі дозволених дій у мережі MissionZOO.

Моніторинг та технічна підтримка

Під час проведення лабораторних робіт здійснюється постійний моніторинг. Основна увага приділяється Telegram-каналу, куди надходять звіти від системи. Це дозволяє викладачу бачити, хто зі студентів застосовує агресивні методи (генеруючи звіти про брутфорс), а хто діє приховано. Технічна підтримка включає оперативне перезавантаження окремих вузлів ESP32 у разі їх апаратного зависання під час інтенсивних мережевих атак. Також аналізуються логи OpenWRT для виявлення проблем із бездротовим покриттям або вичерпанням пулу DHCP-адрес.

Оновлення та масштабування системи

Завдяки модульній децентралізованій архітектурі на базі MicroPython, MissionZOO легко масштабується. Додавання нового пристрою (нової «тварини») не потребує переписування ядра системи. У майбутньому систему можна розширити, додавши вузли, що симулюють вразливі IoT-протоколи (наприклад, MQTT без шифрування), старі версії FTP-серверів або сервіси, вразливі до переповнення буфера. Також можливе вдосконалення скриптів маршрутизатора для симуляції складніших топологій мережі (VLAN, DMZ). Це дозволяє довгостроково підтримувати актуальність стенду, адаптуючи його під нові тренди в галузі інформаційної безпеки.

Оцінка ефективності впровадження

Завершальним етапом є аналіз освітньої ефективності системи.

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

Основними критеріями оцінювання є:

- час, витрачений студентами на проходження сценарію;
- відсоток успішних спроб (отримання Master Key);
- кількість виявлених системою Suricata порушень (спрацьовувань блокувань).

Отримані результати підтверджують високу практичну цінність комплексу. MissionZOO дозволяє ліквідувати прогалину між теоретичними лекціями та реальною практикою роботи з мережевим обладнанням. Використання мікроконтролерів ESP32 замість ресурсомістких віртуальних машин знижує вартість організації лабораторії, забезпечуючи при цьому високий рівень реалізму, інтерактивності та фізичного відгуку під час виконання завдань з кібербезпеки.

На рисунку 5.1 можна побачити загальний вигляд системи разом з маршрутизатором.



Рисунок 5.1 – Загальний вигляд системи

6 ОСНОВНІ ВИСНОВКИ

У ході виконання дипломної роботи було спроектовано та реалізовано апаратно-програмний навчальний комплекс MissionZOO для вивчення основ мережевого аналізу, систем виявлення вторгнень та етичного хакінгу. У процесі аналізу предметної області встановлено, що використання розподіленої інфраструктури на базі мікроконтролерів ESP32 у поєднанні з маршрутизатором OpenWRT є найбільш доцільним для освітніх цілей. Такий підхід дозволяє уникнути використання ресурсомістких віртуальних машин, забезпечує високий рівень реалізму за рахунок фізичного мережевого середовища та гарантує повну ізоляцію і безпеку під час симуляції кібератак у навчальній аудиторії.

Програмна частина системи реалізована з використанням мови MicroPython із глибоким застосуванням асинхронної бібліотеки asyncio на стороні ESP32, а також системних CGI-скриптів на рівні маршрутизатора OpenWRT. Це забезпечило ефективний логічний розподіл функціоналу між вузлами: централізоване керування та імітація баз даних (Elephant), мережева розвідка та виявлення вторгнень (Suricat), дезорієнтація порушників (Canary), моніторинг (Falcon) та фінальні випробування (Python). Реалізований функціонал включає обробку сирих сокетів (Raw Sockets) для аналізу пакетів, роботу інтерактивних Telnet-серверів, маршрутизацію API-запитів між мікроконтролерами у форматі JSON, а також передачу телеметрії викладачу через Telegram-бот.

Проведене тестування показало, що система стабільно функціонує в умовах локальної мережі під час значних навантажень, характерних для мережових атак. Підтверджено швидкість і безпомилковість реакції системи виявлення вторгнень (IDPS), яка коректно детектує сканування утилітою Nmap або спроби Bruteforce-атак і успішно взаємодіє з маршрутизатором для миттєвого апаратного блокування MAC-адрес порушників. Додатково встановлено, що реалізовані алгоритми відновлення (функція jump) дозволяють підтримувати працездатність стенду,

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

автоматично скидаючи мережеві параметри та ролі портів після успішного проходження квесту або критичних збоїв. У результаті система продемонструвала загальну надійність і готовність до багаторазового практичного використання.

Додатково підтверджено високу ефективність використання апаратних ресурсів мікроконтролера ESP32 для одночасної обробки кількох мережевих завдань. Здатність мікроконтролера паралельно підтримувати роботу 4-5 відкритих портів, здійснювати фоновий сніфінг трафіку та генерувати ШІМ-сигнали для керування фізичним відгуком (MOSFET-ключі та вібромотори) є вагомим технічним досягненням. Це критично важливо для рішень з обмеженими обчислювальними можливостями, де необхідно забезпечити баланс між продуктивністю, інтерактивністю та безперервним моніторингом загроз.

Практичне впровадження розробленої системи в навчальний процес показало її прикладну цінність, високий рівень гейміфікації та ефективність у реальних умовах підготовки ІТ-фахівців. Використання формату Capture The Flag (CTF) у поєднанні з фізичними пристроями значно підвищило зацікавленість та мотивацію студентів. Інтеграція сповіщень через Telegram дозволила викладачу дистанційно й оперативно контролювати хід виконання лабораторної роботи, фіксуючи успіхи студентів та методи їхнього впливу на систему. У результаті MissionZOO підтвердив свою ефективність як інструмент для формування практичних компетенцій роботи з мережевим аналізатором Wireshark та утилітами аудиту безпеки.

Узагальнюючи, розроблена система MissionZOO поєднує інноваційний апаратний підхід, складну розподілену архітектуру та високий рівень надійності, що підтверджує правильність обраних технічних і програмних рішень. Отримані результати свідчать про практичну придатність комплексу для використання у закладах вищої освіти. Система має великий потенціал до подальшого розвитку шляхом масштабування «зоопарку». Це гарантує її довгострокову актуальність у сфері навчання кібербезпеці.

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Hack The Box [Електронний ресурс]. – Режим доступу: <https://www.hackthebox.com/>
2. Try Hack Me [Електронний ресурс]. – <https://tryhackme.com/>
3. Перелік CTF майданчиків [Електронний ресурс]. – Режим доступу: <https://hackyourmom.com/osvita/spysok-ctf-majdanchykiv/>
4. Cisco Packet Tracer [Електронний ресурс]. – Режим доступу: <https://www.netacad.com/courses/packet-tracer>
5. GNS3 [Електронний ресурс]. – Режим доступу: <https://www.gns3.com/>
6. Плата розробника ESP32 CH340 Type-C (WiFi Bluetooth WROOM-32) [Електронний ресурс]. – Режим доступу: <https://prom.ua/ua/p1900892856-plata-razrabotchika-esp32.html>
7. Маршрутизатор TP-LINK TL-WR740N [Електронний ресурс]. – Режим доступу: <https://wintik.com.ua/router-tp-link-tl-wr740n>
8. ESP32 30pin devkit case [Електронний ресурс]. – Режим доступу: <https://www.crealitycloud.com/model-detail/esp32-30pin-devkit-case>
9. Gemini [Електронний ресурс]. – Режим доступу: <https://gemini.google.com>
10. Rodin [Електронний ресурс]. – Режим доступу: <https://hyper3d.ai>
11. Tinkercad [Електронний ресурс]. – Режим доступу: <https://www.tinkercad.com>
12. MissionZOO ESP32 Project [Електронний ресурс]. – Режим доступу: <https://www.crealitycloud.com/model-detail/69ff8995158250ad651fcf70>
13. Data Sheet – IRFZ44N [Електронний ресурс]. – Режим доступу: <https://www.infineon.com/assets/row/public/documents/24/49/infineon-irfz44n-datasheet-en.pdf>
14. uasyncio – asynchronous I/O scheduler [Електронний ресурс]. – Режим доступу: <https://docs.micropython.org/en/v1.14/library/uasyncio.html>

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

15. socket – socket module [Електронний ресурс]. – Режим доступу: <https://docs.micropython.org/en/latest/library/socket.html>

16. network – network configuration [Електронний ресурс]. – Режим доступу: <https://docs.micropython.org/en/latest/library/network.html>

17. urequests – Network Request Module [Електронний ресурс]. – Режим доступу: https://makeblock-micropython-api.readthedocs.io/en/latest/public_library/Third-party-libraries/urequests.html

18. ujson – JSON encoding and decoding [Електронний ресурс]. – Режим доступу: <https://docs.micropython.org/en/v1.15/library/ujson.html>

19. machine – functions related to the hardware [Електронний ресурс]. – Режим доступу: <https://docs.micropython.org/en/latest/library/machine.html>

20. Technitium MAC Address Changer [Електронний ресурс]. – Режим доступу: <https://technitium.com/tmac/>

21. Hack The Box vs. TryHackMe: A Professional's Guide to Choosing the Right Cybersecurity Training Platform [Електронний ресурс]. – Режим доступу: <https://flashgenius.net/blog-article/hack-the-box-vs-tryhackme-a-professionals-guide-to-choosing-the-right-cybersecurity-training-platform>

22. TryHackMe: Compare Cyber Security Training Platforms (HTB vs THM) [Електронний ресурс]. – Режим доступу: <https://tryhackme.com/compare/hackthebox>

23. TryHackMe vs Hack The Box for Beginners in 2026: Which is Better for Cyber Security Training? [Електронний ресурс]. – Режим доступу: <https://www.fusioncyber.co/blogs/tryhackme-vs-hack-the-box-for-beginners-in-2026-which-is-better-for-cyber-security-training>

24. Analysis of Networking Tools Using Cisco Packet Tracer (CPT) [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/383654067_Analysis_of_Networking_Tools_Using_Cisco_Packet_Tracer_CPT

25. GNS3 vs EVE-NG vs Packet Tracer vs VIRL vs eNSP: Which Network Emulator or Simulator is Right for You? [Електронний ресурс]. – Режим доступу:

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

<https://infosyte.com/network-simulators/>

26. Best Network Emulator in 2026: Tier-Ranked Honest Comparison [Електронний ресурс]. – Режим доступу: <https://www.netpilot.io/blog/best-network-emulator-2026>

27. A Comparison between the Use of Cisco Packet Tracer and Graphical Network Simulator 3 as Learning Media on Students' Achievement [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/352375864_A_Comparison_between_the_Use_of_Cisco_Packet_Tracer_and_Graphical_Network_Simulator_3_as_Learning_Media_on_Students'_Achievement

28. Nmap [Електронний ресурс]. – Режим доступу: <https://nmap.org/>

29. Wireshark [Електронний ресурс]. – Режим доступу: <https://www.wireshark.org>

30. Packet Sender [Електронний ресурс]. – Режим доступу: <https://packetsender.com>

31. Port Mirroring using TEE [Електронний ресурс]. – Режим доступу: <https://forum.archive.openwrt.org/viewtopic.php?id=58081>

32. Ethical Hacking Tool for Cybersecurity Testing and Exploration Device II [Електронний ресурс]. – Режим доступу: <https://ieeexplore.ieee.org/document/11186571>

33. CTF for education [Електронний ресурс]. – Режим доступу: <https://arxiv.org/pdf/2601.17543>

34. CTF as a Service: A Reproducible and Scalable Infrastructure for Cybersecurity Training [Електронний ресурс]. – Режим доступу: <https://arxiv.org/pdf/2603.22511>

35. Cybersecurity Knowledge and Skills Taught in Capture the Flag Challenges [Електронний ресурс]. – Режим доступу: <https://arxiv.org/pdf/2101.01421>

36. ESP32 Series Datasheet Version 5.2 [Електронний ресурс]. – Режим доступу: https://documentation.espressif.com/esp32_datasheet_en.pdf

					ВКРБ-125.26.0044.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

37. MicroPython documentation [Електронний ресурс]. – Режим доступу: <https://docs.micropython.org/en/latest/>
38. The UCI system [Електронний ресурс]. – Режим доступу: <https://openwrt.org/docs/guide-user/base-system/uci>
39. PostgreSQL Documentation [Електронний ресурс]. – Режим доступу: <https://www.postgresql.org/docs/current/index.html>
40. Telnet Protocol Specification [Електронний ресурс]. – Режим доступу: <https://datatracker.ietf.org/doc/html/rfc854>
41. Guide to Intrusion Detection and Prevention Systems (IDPS) [Електронний ресурс]. – Режим доступу: <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-94.pdf>
42. OpenWrt Project: TP-Link TL-WR740N [Електронний ресурс]. – Режим доступу: <https://openwrt.org/toh/tp-link/tl-wr740n>
43. Canarytokens - Free, quick, and painless detection [Електронний ресурс]. – Режим доступу: <https://canarytokens.org/>
44. awesome-honeypots: an awesome list of honeypot resources [Електронний ресурс]. – Режим доступу: <https://github.com/paralax/awesome-honeypots>
45. Suricata [Електронний ресурс]. – Режим доступу: <https://suricata.io>
46. Що таке платформа CrowdStrike Falcon [Електронний ресурс]. – Режим доступу: <https://www.dell.com/support/kbdoc/uk-ua/000126839/що-таке-платформа-crowdstrike-falcon>
47. Telegram Bot API [Електронний ресурс]. – Режим доступу: <https://core.telegram.org/bots/api>
48. What is IP spoofing? [Електронний ресурс]. – Режим доступу: <https://www.cloudflare.com/learning/ddos/glossary/ip-spoofing/>
49. Firewall overview [Електронний ресурс]. – Режим доступу: <https://openwrt.org/docs/guide-user/firewall/overview>
50. Gamification of cyber ranges in cybersecurity education [Електронний ресурс]. – Режим доступу: <https://ieeexplore.ieee.org/document/9974714>