Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

РОЗРОБКА КРОСПЛАТФОРМНОГО ЗАСТОСУНКУ ДЛЯ ВІЗУАЛІЗАЦІЇ ЛОКАЛЬНИХ КОМП'ЮТЕРНИХ МЕРЕЖ
(тема)

Виконав:
здобувач 4 року навчання,
групи ІТІНФ-21-3
Балабуха І.О.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник ас. каф. Кобилін І.О.
(посада, прізвище, ініціали)

Допускається до захисту

Завідувач кафедри інформатики _____ Кобилін О. А.
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту

Кафедра Інформатики

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Балабусі Ігорю Олексійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка кросплатформного застосунку для візуалізації локальних комп'ютерних мереж

затверджена наказом університету від 19 травня 2025 року № 381Ст
2. Термін подання здобувачем роботи до екзаменаційної комісії 27 травня 2025 р.
3. Вихідні дані до роботи Scapy, Nmap, PyQt5, матеріали бібліотек, конференцій, нвукові роботи за наближеними темами, документація Python.

4. Перелік питань, що потрібно опрацювати в роботі.
1. Аналіз існуючих засобів сканування ЛКМ.
2. Обгрунтування вибору архітектури застосунку.
3. Реалізація ARP/Nmap сканування .
4. Побудова графа на основі отриманих даних.
5. Розробка інтерфейсу, кросплатформеність .
6. Тестування застосунку, проробка різних сценаріїв використання.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) схема архітектури застосунку, скріншоти користувацького інтерфейсу, граф схематично відображаючий пристрої в мережі, таблиця виявлених пристроїв.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	07.04.2025	
2	Аналіз завдання, підбір літератури	08.04.25-10.04.25	
3	Аналіз літератури з досліджуваної проблеми	11.04.25-12.04.25	
4	Аналіз технічних засобів	12.04.25-13.04.25	
5	Розробка методу	13.04.25-25.04.25	
6	Програмна реалізація	25.04.25-11.05.25	
7	Оформлення пояснювальної записки	12.05.25-20.05.25	
8	Перевірка на нормоконтроль	21.05.25-01.06.25	
9	Перевірка на плагіат	21.05.25-01.06.25	
10	Рецензування	21.05.25-01.06.25	
11	Підготовка презентації та доповіді	21.05.25-18.06.25	
12	Занесення роботи в електронний архів	02.06.25-18.06.25	
13	Попередній захист кваліфікаційної роботи	02.06.25-18.06.25	

Дата видачі завдання 7 квітня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ _____ас. каф. Кобилін І.О._____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 42 с., 4 рис., 2 дод., 36 джерел.

Локальна мережа, топологія, ARP-сканування, Scapy, візуалізація, Python, PyQt5

Об’єктом роботи є процес візуалізації топології локальних комп’ютерних мереж. Предметом дослідження є програмні засоби збору, оброблення та відображення даних про мережеву інфраструктуру. Метою роботи є розроблення кросплатформного застосунку для автоматизованого сканування локальної мережі, визначення характеристик вузлів та побудови графічного подання топології.

Під час проходження практики було досліджено інструменти Python, зокрема бібліотеку Scapy та інструмент Nmap, реалізовано архітектуру застосунку, створено функціонал для зчитування IP-адрес, MAC-адрес, портів і побудови графа зв’язків. Практична значущість полягає у можливості використання створеного застосунку в навчальному процесі, в адміністративній роботі в офісах, закладах освіти та внутрішніх мережах організацій.

LOCAL NETWORK, TOPOLOGY, ARP SCAN, SCAPY, VISUALIZATION, PYTHON, PYQT5

The object of the work is the visualization of local computer network topology.

The aim of the work is to develop a cross-platform application for automated LAN scanning, node parameter detection and graphical topology construction. During the internship, Python-based tools were studied, including the Scapy library and the Nmap utility.

The architecture of the application was designed and implemented, functionality for reading IP addresses, MAC addresses, open ports and generating the graph of connections was developed. The practical significance lies in the possibility of using the developed tool in educational

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ЛКМ − локальна комп’ютерна мережа

GUI – графічний інтерфейс користувача

IP – унікальна адреса пристрою в мережі

MAC – фізична адреса мережевого адаптера

OS – операційна система

ARP – протокол визначення адрес (Address Resolution Protocol)

ПЗ – програмне забезпечення

VLAN – віртуальна локальна комп'ютерна мережа

Nmap – утиліта для сканування мережі

DHCP – протокол динамічної конфігурації вузла, який дозволяє пристроям у мережі запитувати та призначати їм IP-адресу

ВСТУП

У сучасному інформаційному суспільстві комп’ютерні мережі є невід’ємною складовою IT-інфраструктури підприємств, інших установ. Локальні комп’ютерні мережі забезпечують обмін даними між пристроями, спільне використання ресурсів, доступ до периферійних пристроїв і підтримку внутрішніх сервісів. Зі зростанням кількості підключених пристроїв, зокрема мобільних девайсів, підвищуються вимоги до керованості й прозорості таких мереж.

Практика виконувалась у межах теми кваліфікаційної роботи «Розробка кросплатформного застосунку для візуалізації локальних комп’ютерних мереж». У ході практики було здійснено аналіз предметної області, сформульовано функціональні вимоги до застосунку, обґрунтовано вибір інструментальних засобів, досліджено можливості бібліотеки Scapy, інструмента Nmap і фреймворку PyQt5 для реалізації графічного інтерфейсу.

Метою практики є закріплення теоретичних знань, набуття практичних навичок у сфері розробки прикладного ПЗ відповідно до кваліфікаційної роботи. У процесі виконання практики було розроблено модель застосунку, опрацьовано методи сканування ЛКМ, зчитування характеристик, побудови графа для візуалізації мережевої топології.

Актуальність роботи полягає у необхідності створення програмного засобу, що поєднує збір, обробку та структуровану подачу інформації про ЛКМ з можливістю візуалізації її структури. На відміну від існуючих рішень, що мають меншу функціональність чи моноплатформність, розроблений застосунок орієнтований на універсальність, адаптивність та доступність. Інструмент може бути використаний у навчальних проектах, щоденному моніторингу мережевої інфраструктури в корпоративних або службових мережах, тощо.

1 ОГЛЯД ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Релевантність дослідження локальних мереж

В сьогоднішньому цифровому просторі локальні комп’ютерні мережі є критично важливою складовою інформаційної інфраструктури організацій, закладів освіти, бізнесу та приватного сектору. Збільшення кількості пристроїв, поява інтернету речей, активне впровадження бездротових технологій та гібридні формати роботи зумовлюють суттєве ускладнення процесу адміністрування мереж. Для забезпечення стабільності, ефективності та безпеки необхідно мати повну й актуальну інформацію про пристрої, що функціонують у мережі.

Управління локальною мережею вимагає постійного моніторингу активності пристроїв, виявлення нових або сторонніх вузлів, аналізу конфігурації підключень та відповідності політикам безпеки. Часто адміністраторам доводиться вручну перевіряти конфігурацію маршрутизаторів, переглядати списки DHCP або використовувати складні командні утиліти для аналізу стану мережі. Це призводить до втрати часу, підвищення ймовірності помилок і загального зниження ефективності роботи [1].

Окрему загрозу становлять несанкціоновані підключення. Якщо у локальній мережі з’являється невідомий пристрій, він може бути використаний як точка атаки або витоку даних. Більшість класичних засобів безпеки, таких як фаєрволи або антивіруси, не мають можливості виявити появу нових пристроїв на рівні мережевої інфраструктури. Це створює потребу у використанні спеціалізованих інструментів, які дозволяють адміністратору бачити реальну ситуацію у мережі в реальному часі.

Особливої актуальності ця задача набуває в умовах кіберзагроз, де навіть незначна вразливість або невідомий пристрій може стати входом для

несанкціонованого доступу. Ефективне сканування мережі дає змогу оперативно реагувати на події, виявляти аномалії та будувати політики доступу на основі реального стану інфраструктури.

У зв’язку з цим актуальним є створення інструменту, який дозволяє швидко й точно ідентифікувати пристрої в локальній мережі, класифікувати їх, візуалізувати мережеву топологію та забезпечувати інтуїтивно зрозумілий інтерфейс для подальшого аналізу. Такий підхід дозволить істотно підвищити рівень контролю над інфраструктурою та зменшити ризики для безпеки.

1.2 Основи протоколу ARP та його роль у мережевому скануванні

Протокол ARP (Address Resolution Protocol) є одним із базових елементів, що забезпечують функціонування локальних комп’ютерних мереж. Його основна задача полягає у зіставленні IP-адреси з відповідною MAC-адресою на рівні канального доступу. Це дозволяє хостам, які працюють у межах однієї підмережі, визначати фізичну адресу пристрою призначення перед передачею даних [2].

Механізм роботи ARP ґрунтується на широкомовному запиті. Коли комп’ютер хоче дізнатись MAC-адресу пристрою з певною IP-адресою, він надсилає ARP-запит у мережу. Усі пристрої отримують цей запит, але лише той, чия IP-адреса збігається з вказаною у запиті, надсилає відповідь з власною MAC-адресою. Така процедура дозволяє динамічно будувати ARP-таблицю на кожному вузлі мережі.

Особливістю ARP є те, що цей протокол працює лише у межах локальної мережі. Саме це робить його корисним для сканування локальних сегментів. Надсилаючи ARP-запити до всіх IP-адрес у підмережі (наприклад, 192.168.0.0/24), можна виявити активні пристрої, які

відповідають на ці запити. Таким чином, сканування за допомогою ARP є точним і швидким способом виявлення присутності пристроїв у мережі.

Ще однією перевагою ARP-сканування є його непомітність для більшості систем безпеки. ARP-пакети вважаються типовими для стандартної роботи мережі, тому не блокуються фаєрволами і не викликають підозри.

Це дозволяє здійснювати сканування без порушення звичних умов роботи інфраструктури. У розробленому застосунку протокол ARP використовується як основний метод початкового сканування. Саме на основі відповідей на ARP-запити формується базовий список пристроїв у мережі. Далі ці пристрої аналізуються додатковими методами, зокрема через Nmap, для поглибленого визначення параметрів.

1.3 Принципи роботи утиліти Nmap

Утиліта Nmap (Network Mapper) є одним із найпоширеніших і найпотужніших інструментів для сканування комп'ютерних мереж. Вона призначена для виявлення активних хостів, аналізу відкритих портів, визначення типу операційної системи та інших параметрів пристрою. Завдяки широкому спектру підтримуваних методів Nmap дозволяє виконувати як базове сканування, так і глибокий аудит мережевої інфраструктури [3].

Основою роботи Nmap є надсилання спеціальних мережевих пакетів до пристроїв з подальшим аналізом відповідей. Залежно від обраного режиму утиліта може застосовувати різні типи сканування, серед яких: TCP SYNсканування, UDP-сканування, сканування з виявленням версій сервісів, виявлення операційної системи, traceroute, скриптове сканування за допомогою Nmap Scripting Engine. Завдяки цьому досягається висока точність і повнота результатів.

Однією з ключових переваг Nmap є підтримка агресивного режиму сканування. При його активації утиліта не тільки перевіряє порти [2], але й виконує глибокий аналіз типу пристрою, ОС, імені хоста, конфігурації сервісів. Це дозволяє максимально точно ідентифікувати пристрій навіть без повного доступу до нього.

Утиліта підтримує запуск через командний рядок, що робить її зручною для автоматизації, а також має графічну оболонку Zenmap для менш досвідчених користувачів. Проте навіть у базовому режимі користування вимагає певної підготовки та знання параметрів командного запуску.

У межах розробленого застосунку Nmap використовується як додатковий інструмент до ARP-сканування. Після виявлення активних IP-адрес програма ініціює Nmap-сканування у стандартному або агресивному режимі з ключами -sS -A -Pn -O -F, що дозволяє отримати розширені дані про пристрої. Отримана інформація обробляється і виводиться у графічному інтерфейсі для зручності перегляду та подальшої класифікації.

1.4 Бібліотека Scapy для мережевого аналізу

Scapy є потужною бібліотекою мови Python, призначеною для створення, обробки, надсилання та аналізу мережевих пакетів на низькому рівні. Вона дозволяє реалізовувати як прості, так і складні мережеві сценарії без потреби використовувати зовнішні утиліти. Завдяки підтримці великої кількості протоколів, зокрема Ethernet, ARP, IP, TCP, UDP, ICMP, DNS, HTTP та багатьох інших, Scapy [4] є універсальним інструментом для аналізу мережевого трафіку.

Головною перевагою Scapy є те, що вона не обмежується лише аналізом, а дозволяє конструювати довільні мережеві пакети і надсилати їх до обраних адресатів. Це відкриває широкі можливості для тестування,

сканування, дослідження поведінки пристроїв у мережі, а також розробки спеціалізованих інструментів.

У контексті даної роботи Scapy використовується для формування ARP-запитів та обробки відповідей. Це дозволяє виявити активні пристрої у локальній підмережі, навіть якщо вони не мають відкритих портів або ігнорують інші типи запитів. Використання бібліотеки дає можливість повністю контролювати логіку сканування, обробку помилок, часові затримки, параметри фільтрації та зберігання результатів.

Бібліотека має прийнятний синтаксис і повністю інтегрується з іншими модулями Python, що робить її придатною для створення повноцінних застосунків із графічним інтерфейсом або автоматизованими сценаріями. У межах розробленого застосунку Scapy забезпечує стабільне ARP-сканування, результат якого слугує базою для подальшого аналізу через Nmap [5].

Тому використання бібліотеки Scapy дозволяє досягти максимальної гнучкості в реалізації низькорівневого сканування і забезпечити ефективний збір інформації про мережу без потреби в зовнішніх залежностях або складних конфігураціях.

1.5 Обґрунтування вибору інструментів і методів

Для реалізації застосунку з аналізу пристроїв у локальній мережі були обрані три основні технологічні компоненти: бібліотека Scapy, утиліта Nmap та фреймворк PyQt5. Така комбінація забезпечує повноцінну функціональність: від виявлення пристроїв до зручної візуалізації результатів у графічному інтерфейсі.

Вибір Scapy обумовлений її здатністю здійснювати низькорівневу роботу з мережевими пакетами. Вона дозволяє створювати й надсилати ARP-запити, отримувати відповіді та фільтрувати їх відповідно до умов

користувача. У порівнянні з іншими бібліотеками Python, такими як socket або ipaddress, Scapy має набагато ширший функціонал і дає змогу гнучко керувати процесом сканування.

Утиліта Nmap доповнює базове ARP-сканування розширеним аналізом пристроїв. Вона дозволяє виявити відкриті порти, тип операційної системи, ім’я хоста та навіть визначити ймовірний тип пристрою. У роботі використовується агресивний режим Nmap (-sS -A -Pn -O -F), який дає змогу отримати максимально повну інформацію про мережеві вузли. [1] Аналоги Nmap або спрощені графічні сканери, такі як Advanced IP Scanner, не дозволяють досягти подібної глибини аналізу, особливо в умовах динамічної інфраструктури.

Для побудови інтерфейсу було обрано PyQt5, як один із найпотужніших фреймворків для створення графічних інтерфейсів мовою Python. На відміну від tkinter або wxPython, PyQt5 забезпечує більш професійний вигляд, підтримку іконок, анімацій, кастомних віджетів та багатомовності. У межах даного застосунку реалізовано перемикання теми оформлення, мови інтерфейсу та режимів сканування, що підвищує зручність користування і робить застосунок доступним навіть для непідготовленого користувача. Комбіноване використання цих інструментів забезпечує ефективне виконання поставлених задач: виявлення, класифікація та візуалізація пристроїв у локальній мережі. Вони взаємодіють між собою в єдиному середовищі Python, що спрощує розгортання, тестування та подальший розвиток системи [6].

1.6 Практичні сфери застосування розробленого інструменту

Розроблений застосунок має широкий спектр практичного використання в умовах реальних мережевих інфраструктур [7]. Його функціональність охоплює як виявлення пристроїв і класифікацію за

мережевими параметрами, так і візуалізацію топології, що робить інструмент придатним для застосування в кількох ключових галузях.

У корпоративному середовищі застосунок може бути використаний системними адміністраторами для моніторингу активності пристроїв у внутрішній мережі. Він дозволяє оперативно виявляти підозрілі підключення, перевіряти конфігурацію мережевої інфраструктури, забезпечувати інвентаризацію обладнання та відповідність політикам інформаційної безпеки. У локальних домашніх або малих офісних мережах інструмент допомагає виявляти сторонні пристрої, що могли підключитись до мережі без відома користувача. Це особливо актуально у випадках використання бездротових точок доступу, де сторонній пристрій може підключитися навіть без фізичного доступу до обладнання.

У сфері кібербезпеки застосунок може бути використаний на початкових етапах аудиту інфраструктури для побудови карти мережі, виявлення відкритих портів, визначення типів операційних систем та загального стану доступності вузлів. Це дозволяє сформувати модель потенційних загроз і краще зрозуміти поверхню атаки. [8, 9].

У державних структурах та органах публічного управління засіб може бути впроваджений як внутрішній інструмент моніторингу, що не потребує зовнішнього підключення або сторонніх служб. Завдяки візуалізації мережевої структури забезпечується прозорість у роботі IT-відділів, а можливість автономної роботи без залучення хмарних сервісів відповідає вимогам безпеки для критичних об’єктів.

Усі зазначені приклади доводять практичну цінність розробленого засобу, який поєднує простоту використання, інтуїтивно зрозумілий інтерфейс та широкі функціональні можливості для роботи у локальних мережах різного масштабу. Нижче (рис. 1.1) представлено таблицю пристроїв, виявлених у процесі роботи програми.

ARP сканування і показано графічну візуалізацію структури локальної мережі (рис.1.2). [10, 11].

	IP	MAC	ОС	Ім'я	Виробник	Тип
1	192.168.1.124	f4:7b:09:94:ab:0b	Microsoft Windows 10 1607 - 11 23H2	Divan.localdomain	Unknown	Computer
2	192.168.1.1	44:fe:3b:2d:c1:f5	Linux 2.6.19 - 2.6.36	o2.box	Unknown	Router
3	192.168.1.123	9e:05:b5:1d:b8:e8	Unknown	unknown9E05B51DB8E8	Unknown	Unknown
4	192.168.1.177	28:56:5a:f6:45:c5	Linux 2.6.18	unknown28565AF645C5	Unknown	Computer (Linux)
5	192.168.1.18		Microsoft Windows 10 - 11	Pixel-8a	Unknown	Computer
6	192.168.1.20	1c:91:80:dc:ee:7f	Dell PowerConnect 2708 switch	Air-von-Beni	Unknown	Unknown
7	192.168.1.22	f2:f9:2f:5d:70:8c	Unknown	iPhone	Unknown	Unknown

Рисунок 1.1 – Відображення пристроїв у таблиці застосунку [20, 21].

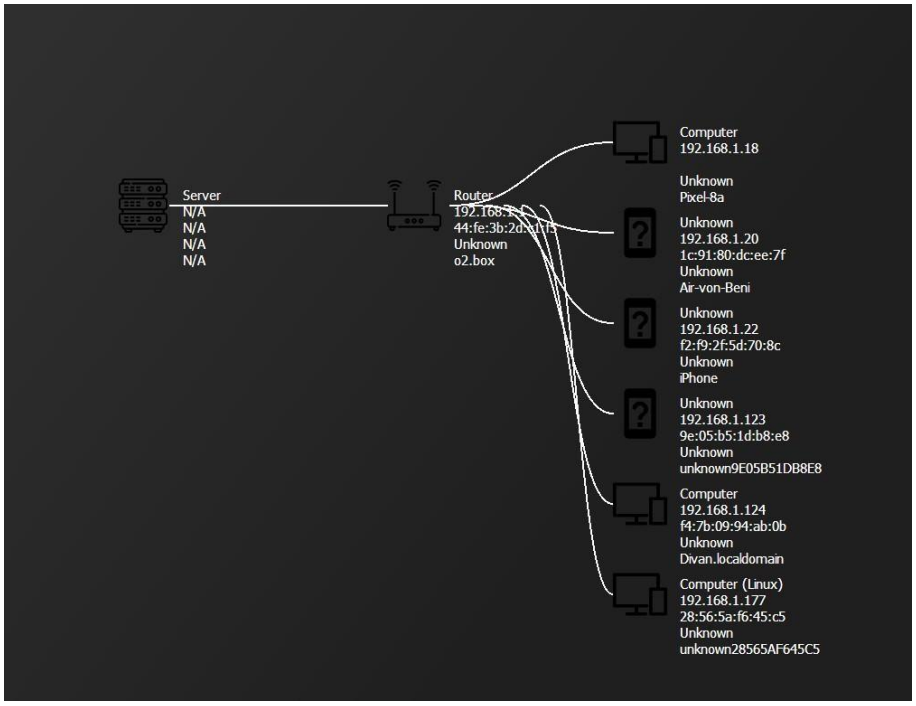

Рисунок 1.2 – Топологія локальної мережі у вигляді графа пристроїв

1.7 Постановка задачі

Метою даної кваліфікаційної роботи є створення універсального програмного застосунку для сканування, виявлення, класифікації та візуалізації пристроїв у локальній мережі з використанням сучасних методів аналізу мережевого трафіку. Такий засіб має забезпечити простоту використання, гнучкість налаштувань, візуальну наочність результатів та відповідати потребам як адміністратора локальної мережі, так і фахівця з

кібербезпеки. Для досягнення поставленої мети необхідно розв’язати наступні задачі:

– розробити графічний інтерфейс користувача з підтримкою.

– перемикання мови та теми оформлення.

– реалізувати механізм ARP-сканування для виявлення активних пристроїв у локальній мережі.

– інтегрувати засоби глибокого сканування з використанням утиліти Nmap.

– організувати автоматичну класифікацію пристроїв за MAC-адресою та типом операційної системи.

– реалізувати побудову графічної візуалізації мережевої топології.

– забезпечити зручне табличне відображення зібраних даних про пристрої.

– реалізувати можливість логування процесу сканування для подальшого аналізу .

– забезпечити стабільну роботу застосунку у різних середовищах (наприклад Windows).

Постановка задачі узгоджується з актуальними потребами в адмініструванні мереж і кіберзахисті, а також відповідає технічним можливостям сучасного стеку технологій Python.

2 АНАЛІЗ ВИМОГ ТА ПРЕДМЕТНОЇ ОБЛАСТІ

2.1 Аналіз предметної області

Локальна комп’ютерна мережа є сукупністю пристроїв, що об’єднані в межах обмеженої географічної області з метою обміну даними, доступу до ресурсів і взаємодії між вузлами. Такі мережі активно застосовуються в офісах, навчальних закладах, підприємствах і приватному секторі. Типова локальна мережа включає маршрутизатор або комутатор, клієнтські комп’ютери з різними типами ОС,а саме Windows, Mac, Linux, принтери, сервери, точки доступу Wi-Fi, IP-телефони на Android і IOS та інші різноманітні пристрої. Мережі можуть бути побудовані як дротовими, так і бездротовими каналами з’єднання. У більшості випадків використовується комбінація обох типів з розподілом адресного простору за допомогою DHCP. Передавання даних здійснюється на основі мережевої моделі TCP/IP, що забезпечує підтримку широкого спектру протоколів і сервісів. Комунікація між пристроями залежить від знання IP-адрес і MAC-ідентифікаторів, що дозволяє здійснювати маршрутизацію та адресацію пакетів [12].

Особливістю сучасних ЛКМ є постійна динаміка – пристрої підключаються, змінюють конфігурацію або залишають мережу. Через це адміністратору потрібно мати інструменти для своєчасного виявлення змін і підтримання актуального уявлення про стан інфраструктури.

Більшість стандартних засобів управління мережею не забезпечують достатньої наочності, не відображають топологію або потребують глибоких технічних знань.

У предметній області також важливо враховувати аспекти безпеки. Поява нового пристрою в мережі може свідчити як про легітимне підключення, так і про потенційну загрозу. Крім того, у великій кількості випадків адміністратор має справу не лише з виявленням пристроїв, а й з

потребою класифікувати їх, визначити тип операційної системи, відстежити активні служби або відкриті порти для подальшого прийняття рішень.

Отже предметна область охоплює взаємодію користувачів і пристроїв у межах локальної мережі, потребу в автоматизованому виявленні та аналізі її компонентів, а також візуальне представлення мережевої структури. Це створює умови для обґрунтованого формування вимог до програмного засобу, який дозволяє адміністратору швидко отримувати повну інформацію про стан мережі без використання складних систем моніторингу.

2.1.1 Структура локальної комп’ютерної мережі

Локальна комп’ютерна мережа є системою взаємозв’язаних пристроїв, розташованих у межах обмеженого фізичного простору. Вона забезпечує обмін даними, доступ до ресурсів, передавання сигналів управління та підтримку службових процесів. Найчастіше такі мережі створюються у межах офісів, навчальних аудиторій, серверних кімнат або домашніх приміщень. Основою архітектури локальної мережі є принцип прямого фізичного або логічного підключення пристроїв до одного вузла або через проміжні елементи.

До складу типової ЛКМ входять маршрутизатор, комутатор, комп’ютери, принтери, мережеві накопичувачі, точки доступу Wi-Fi та інші периферійні пристрої. Додатково можуть бути присутні сервери, медіацентри, шлюзи безпеки та системи відеоспостереження. Усі вони взаємодіють через дротові або бездротові з’єднання, які підтримуються інтерфейсами Ethernet, Wi-Fi або VLAN.

У більшості сучасних реалізацій використовується структура із центральним маршрутизатором, до якого підключені інші пристрої безпосередньо або через комутатор. Це дозволяє розділити навантаження,

створити кілька логічних сегментів і реалізувати контроль доступу до внутрішніх або зовнішніх ресурсів. У невеликих офісах структура зазвичай має одну точку виходу в Інтернет і декілька локальних клієнтів. У навчальних аудиторіях можливе підключення кількох десятків пристроїв з динамічною адресацією. У державних установах використовуються сегментовані мережі з ізоляцією критичних вузлів.

Усі електронні пристрої в локальній мережі обмінюються даними за допомогою IP-протоколу. Кожен з них має власну IP-адресу і MAC-ідентифікатор. Обслуговування таблиць адрес виконується службою DHCP, яка динамічно призначає IP-адреси пристроям під час підключення.

Усі пакети пересилаються в межах локального сегменту без участі зовнішніх маршрутизаторів, якщо не йдеться про доступ до глобальної мережі Особливістю структури локальної комп’ютерної мережі є її варіативність.

У деяких конфігураціях клієнти підключаються безпосередньо до точки доступу. В інших випадках буває так, що усі пристрої взаємодіють між собою через центральний комутатор. Це серйозно впливає на ефективність сканування, оскільки деякі пакети можуть не розповсюджуватись усьому сегменту. Саме тому точність сканування достатньо сильно залежить від конкретної топології, типу обладнання і наявності фільтрації на рівні комутаторів.

Візуалізація структури локальних комп’ютерних мереж є необхідною умовою якісного адміністрування. Вона дозволяє виявити несанкціоновані підключення до них, оцінити навантаження, локалізувати проблеми та оптимізувати розміщення обладнання.

Програмний застосунок, який будується в межах цієї роботи, має враховувати типову архітектуру локальних мереж і бути здатним адаптуватись до всіх можливих для неї варіантів.

2.1.2 Типи пристроїв і приклади конфігурацій

Локальні комп’ютерні мережі об’єднують різні типи пристроїв, кожен з яких виконує окремі функції у межах загальної інфраструктури. Для ефективного сканування та аналізу мереж необхідно враховувати різноманітність таких пристроїв, їхні характеристики, рівень взаємодії та роль у топології

Основними пристроями, які найчастіше зустрічаються в локальних мережах, є клієнтські комп’ютери. Це персональні комп’ютери або ноутбуки, які виконують роль кінцевих точок доступу. Вони взаємодіють із внутрішніми сервісами, отримують IP-адреси за допомогою DHCP та можуть працювати як у постійному, так і в тимчасовому режимі підключення.

До другої групи належать мережеві принтери та багатофункціональні пристрої. Вони зазвичай мають власний вебінтерфейс, працюють на статичних або динамічних адресах та використовуються у внутрішніх сервісах офісних установ.

У процесі сканування ці пристрої визначаються за характерними MAC-префіксами, портами або службами, що відкриті на них. Наступною групою є точки доступу Wi-Fi та маршрутизатори. Ці пристрої забезпечують підключення клієнтів у бездротовому режимі, створюють окремі підмережі, виконують NAT та обробку трафіку. Часто до них підключається велика кількість клієнтів, що динамічно змінюються. Наявність точки доступу значно впливає на загальну структуру мережі, тому вона має бути виявлена та проаналізована під час сканування.

У розширених конфігураціях до локальної мережі можуть входити сервери. Це файлові, поштові, DNS або додаткові спеціалізовані сервери. Вони мають постійні IP-адреси, відкриті порти та працюють цілодобово.

Сканування таких вузлів дозволяє визначити сервіси, що працюють у межах мережі, а також виявити потенційні вразливості. [5].

Окрему категорію становлять IoT-пристрої. Це камери відеоспостереження, датчики, системи розумного дому, енергомоніторингу та безпеки. Вони можуть працювати на нестандартних портах, не відповідати на ICMP-запити, а також мати слабкий захист. Саме тому важливо враховувати наявність таких пристроїв при проєктуванні інструментів сканування.

У типових конфігураціях ЛКМ можна виділити кілька моделей розміщення обладнання. Наприклад, у навчальній лабораторії це централізована мережа з одним маршрутизатором і великою кількістю однотипних клієнтів. У державній установі – це розділення на ізольовані сегменти, де кожен з них має свій набір критичних пристроїв. У домашній мережі – це комбінована топологія, яка включає ноутбуки, смартфони, телевізори, ігрові приставки та інші пристрої. Усі ці варіанти створюють різні умови для сканування та візуалізації, що має бути враховано в архітектурі застосунку.

Отже у локальних мережах присутні пристрої різного призначення та з різними характеристикам, тому для забезпечення точності виявлення та повноти візуалізації необхідно враховувати їхню специфіку, поведінку в мережі, рівень видимості та характерні ознаки.

2.1.3 Протоколи обміну і середовище роботи застосунку

Локальна комп’ютерна мережа функціонує завдяки використанню базових мережевих протоколів, які забезпечують адресацію, маршрутизацію, виявлення вузлів і передачу даних.

Для реалізації повноцінного сканування інфраструктури, класифікації пристроїв і візуалізації структури необхідно враховувати саме ті протоколи,

що використовуються на канальному, мережевому та транспортному рівнях моделі OSI.

Основним протоколом, який використовується для виявлення активних пристроїв, є ARP. Його роль полягає в тому, щоб зіставляти IP-адресу вузла з його MAC-ідентифікатором. ARP-запити надсилаються широкомовно в межах локального сегмента. Усі активні пристрої, що відповідають на запит, автоматично потрапляють у таблицю виявлених. Цей механізм дозволяє швидко визначити активні вузли без сканування портів або запуску додаткових служб.

Для отримання розширеної інформації про пристрої використовується утиліта Nmap, яка працює на основі протоколів TCP, UDP, ICMP та деяких специфічних механізмів ідентифікації. Вона дозволяє виявляти відкриті порти, визначати тип операційної системи, запущені служби, а також генерувати зведену інформацію про вузол. Це надає змогу сформувати повну картину мережевої активності пристрою. [13]

Також важливо враховувати особливості протоколу DHCP, який використовується для динамічного призначення IP-адрес у більшості мереж. Застосунок повинен працювати з урахуванням того, що IP-адреси вузлів можуть змінюватися під час перезапуску, перевантаження маршрутизатора або завершення терміну дії лізингу. Тому для точнішої ідентифікації доцільно використовувати MAC-адресу як сталий параметр.

У процесі побудови візуалізації важливу роль відіграють логічні протоколи, які дозволяють ідентифікувати взаємозв’язки між пристроями. Наприклад, маршрутизатор взаємодіє з іншими вузлами через протоколи NAT, DNS, DHCP або за допомогою веб-інтерфейсів. Аналіз активних портів і відкритих служб дозволяє визначити тип пристрою та його функціональне призначення Застосунок має бути здатним працювати у різних умовах. Це може бути домашня мережа з невеликою кількістю клієнтів і маршрутизатором базового рівня. Також це може бути складна офісна інфраструктура з кількома підмережами, VLAN, проксі-серверами

або внутрішніми шлюзами безпеки. Усі ці випадки потребують адаптивного підходу до роботи з протоколами та механізмами виявлення пристроїв, а ефективна робота програмного засобу – напряму залежить від знання та правильного застосування мережевих протоколів і їх використання дозволяє автоматизувати процес виявлення, класифікації та збирання інформації про вузли локальної мережі, а також забезпечити точну візуалізацію її структури.

2.1.4 Актуальні загрози та задачі виявлення пристроїв

У сучасних локальних мережах проблема безпеки стає однією з найважливіших. Незалежно від масштабу інфраструктури, будь-який неконтрольований пристрій, підключений до мережі, може становити загрозу стабільності, конфіденційності та цілісності даних. Через збільшення кількості бездротових з’єднань, використання гостьових доступів і слабкий контроль внутрішньої політики, адміністратори стикаються з необхідністю постійного моніторингу активних пристроїв.

Однією з найбільш поширених загроз є несанкціоноване підключення. Це може бути як навмисне проникнення зловмисника, так і випадкове підключення зовнішнього пристрою, наприклад, мобільного телефона або ноутбука, що не має дозволу на доступ. У ситуації, коли адміністратор не має наочних інструментів спостереження, подібне підключення може залишитись непоміченим протягом тривалого часу.

Ще однією загрозою є використання пристроїв з вразливим або застарілим програмним забезпеченням. Такі пристрої можуть бути експлуатовані сторонніми особами для доступу до внутрішніх ресурсів, створення ботнетів або збору службової інформації. Виявлення таких вузлів через сканування портів або аналіз операційної системи дозволяє своєчасно ізолювати їх або провести оновлення.

У більш складних сценаріях можливе цілеспрямоване маскування пристроїв за допомогою технік MAC-спуфінгу або ARP-спуфінгу. У такому випадку пристрій видає себе за інший, з підміною ідентифікаторів. Це може призвести до перехоплення трафіку, порушення маршрутизації або створення фальшивих вузлів. Протидія таким атакам потребує не лише пасивного моніторингу, а й активного сканування та аналізу аномалій у мережевій структурі.

У навчальних закладах, державних установах та малих офісах часто відсутня спеціалізована система контролю. У таких умовах основним інструментом залишається візуальне спостереження або ручна перевірка через інтерфейс маршрутизатора. Це не дозволяє своєчасно реагувати на зміну структури мережі, втрату вузлів або появу нових пристроїв [9].

Задача виявлення пристроїв включає кілька підетапів. По-перше, необхідно сформувати перелік активних вузлів, доступних у межах локального сегменту [13]. По-друге, потрібно зібрати інформацію про їхню IP-адресу, MAC-ідентифікатор, відкриті порти та характерні служби. По-третє, здійснюється класифікація типу пристрою – чи є це клієнтський комп’ютер, точка доступу, принтер, камера або інше обладнання. Лише після цього можливо побудувати топологію мережі, яка відображає її реальний стан. Автоматизоване виявлення пристроїв дозволяє уникнути ризиків, пов’язаних із людським фактором, затримками у реакції та помилками ідентифікації. Це особливо важливо у випадках, коли мережа є частиною критичної інфраструктури або містить конфіденційні дані.

Актуальні загрози, які виникають у локальних мережах, потребують своєчасного виявлення всіх активних пристроїв. Це завдання є базовим етапом у загальному процесі контролю, захисту і модернізації IT інфраструктури, а також ключовою функцією для застосунку, що розробляється в межах цієї роботи.

2.2 Адміністратори ЛКМ, освітні установи, держсектор

Цільова аудиторія програмного засобу, що розробляється в межах цієї роботи, охоплює широке коло користувачів, які мають потребу в контролі та аналізі локальної комп’ютерної мережі. До неї належать як фахівці у сфері інформаційних технологій, так і користувачі без глибоких технічних знань, які відповідають за підтримку стабільної роботи комп’ютерної інфраструктури в організаціях різного типу.

Першу групу користувачів становлять адміністратори локальних мереж. Це фахівці, які відповідають за технічне обслуговування, діагностику та безпеку мережевої інфраструктури у приватних компаніях, державних установах або навчальних закладах. Вони здійснюють налаштування маршрутизаторів і комутаторів, контролюють підключення нових пристроїв, відстежують інциденти, пов’язані зі збоєм з’єднання або проникненням. Для таких фахівців необхідно забезпечити повну і швидку картину стану мережі, її топології, а також можливість швидко виявити нові або підозрілі пристрої.

Наступну категорію становлять освітні заклади, зокрема технікуми, університети та профільні IT-курси. У цих середовищах створюються навчальні лабораторії з десятками підключених пристроїв. Часто вони використовують динамічну IP-адресацію, а підключення здійснюється без жорсткої авторизації. Це створює потенційні ризики проникнення ззовні або появи пристроїв, які не належать до навчального процесу. Окрім того, у таких умовах важливо мати інструмент, що дозволяє демонструвати студентам базові принципи побудови та візуалізації мережі, що робить програму корисною не лише для адміністратора, але і як навчальний засіб.

Окрему групу складають державні установи, які використовують внутрішні інформаційні системи. У цих середовищах особливо важливо дотримуватись вимог до інформаційної безпеки. Багато державних організацій не використовують складні платформи керування

інфраструктурою, а контролюють підключення вручну або за допомогою базових інструментів.

Це створює ризики, пов’язані з несанкціонованим підключенням, витоком службової інформації або непомітною активністю шкідливого програмного забезпечення. Застосунок, який може виявляти і візуалізувати мережеву активність без складної конфігурації, є особливо корисним для невеликих відділів IT-підтримки або фахівців із кіберзахисту.

Також до цільової аудиторії можна віднести технічних консультантів, викладачів IT-дисциплін, спеціалістів з мережевої безпеки та ентузіастів, які експериментують із домашніми мережами. Для всіх цих категорій важлива простота використання, мінімальні вимоги до встановлення і наявність графічного інтерфейсу, який не потребує знання командного рядка або спеціалізованих протоколів. Програмний засіб орієнтований не лише на професійне використання, але й на широку аудиторію, яка потребує базових інструментів виявлення, аналізу та візуалізації мережі.

Його гнучкість, доступність та відсутність складних налаштувань дозволяють використовувати його у багатьох сферах, незалежно від масштабу та специфіки мережевої інфраструктури.

2.2.1 Сценарії використання, технічні обмеження

У межах експлуатації програмного засобу можна виокремити кілька типових сценаріїв використання, які охоплюють як повсякденні адміністративні задачі, так і спеціалізовані ситуації, пов’язані з кібербезпекою, аудитом та навчанням. У кожному випадку застосунок має демонструвати стабільність, точність і гнучкість налаштувань, адаптованих до конкретного середовища Одним із найпоширеніших сценаріїв є щоденний моніторинг мережі у невеликих офісах. У такому середовищі адміністратор має перевіряти список підключених пристроїв, контролювати

зміни структури та виявляти неавторизовані підключення. Застосунок дає змогу запускати швидке ARP-сканування, переглядати таблицю знайдених вузлів, відстежувати MAC-адреси та їх відповідність списку дозволених пристроїв.

Другий сценарій стосується навчального середовища. У лабораторних класах із великою кількістю однотипних пристроїв важливо відслідковувати стан підключення, перевіряти роботу мережі перед практичними заняттями та забезпечити доступ до графічного інструменту для студентів. Програмний застосунок може використовуватись як демонстраційний приклад для вивчення принципів ARP-обміну, побудови топології, сканування портів або аналізу служби Nmap [14].

У державних установах основним завданням є перевірка відповідності структури мережі політикам інформаційної безпеки. Тут актуальні сценарії періодичного аудиту, коли IT-фахівець повинен отримати загальну картину всіх активних пристроїв, відфільтрувати вузли за типом, визначити критичні служби або аномальні підключення. Графічна візуалізація допомагає виявити вузли, які знаходяться поза очікуваним маршрутом або дублюють функції інших пристроїв.

Особливе значення мають сценарії швидкого реагування на підозрілі події. Наприклад, при появі невідомого пристрою, зміні кількості вузлів у підмережі або підозрілих змінах у службах доступу, адміністратор може запустити повне сканування з розширеним аналізом і отримати оновлену карту мережі. Це дає змогу вчасно виявити загрозу та локалізувати проблему.

Незважаючи на універсальність, застосунок має і певні технічні обмеження. Його ефективність безпосередньо залежить від топології мережі. Наприклад, у випадку з використанням комутаторів з ізоляцією портів або при активному фільтруванні широкомовного трафіку, ARP-запити можуть не досягати всіх вузлів. Це знижує точність виявлення пристроїв. Крім того, деякі антивірусні продукти або системи захисту

можуть блокувати сканування портів, що ускладнює використання інструментів на базі Nmap.

Іншим обмеженням є робота в розподілених мережах з кількома підмережами або VLAN. У такому випадку застосунок зможе виявити лише ті пристрої, які доступні з поточної підмережі. Для повного охоплення потрібна додаткова маршрутизація або запуск інструмента з різних точок доступу.

Важливо також враховувати права користувача. Якщо застосунок виконується з обмеженими привілеями, деякі типи запитів можуть бути заблоковані системою безпеки. Таким чином, у більшості практичних сценаріїв застосунок може бути використаний як повноцінний інструмент для аналізу та візуалізації структури ЛКМ. Але ефективність його роботи залежить від особливостей мережевого середовища, налаштувань обладнання та рівня доступу користувача.

2.3 Виявлення пристроїв і збір базової інформації

Першим і ключовим етапом у процесі аналізу локальної мережі є виявлення пристроїв, які в ній присутні. Без достовірної інформації про реальні вузли неможливо побудувати топологію, оцінити безпеку або виконати аудит. Тому функціонал, пов’язаний із виявленням активних пристроїв, є базовим для будь-якого інструменту сканування мережі.

Процес виявлення починається з визначення діапазону IP-адрес, які потрібно перевірити. У більшості випадків застосовується автоматичне визначення підмережі, що відповідає активному інтерфейсу комп’ютера. Застосунок формує список IP-адрес у межах цієї підмережі. Потім до кожної адреси надсилається ARP-запит. У разі отримання відповіді IP-адреса заноситься до таблиці виявлених вузлів разом із MAC-ідентифікатором.

ARP-запити є оптимальним інструментом для базового сканування локального сегмента мережі. Вони працюють швидко, не вимагають складної конфігурації та сумісні з більшістю мережевих середовищ. У межах розробленого застосунку для реалізації цієї функції використовується бібліотека Scapy, яка дозволяє вручну формувати ARP-пакети, надсилати їх і обробляти відповіді. Це дає гнучкість у налаштуванні параметрів запиту та розширює можливості щодо обробки результатів.

Збір базової інформації про пристрої включає не лише їх IP- і MAC-адреси, а й такі характеристики, як тип інтерфейсу, час виявлення, а також джерело відповіді.

У майбутньому ці дані використовуються для візуалізації мережі, побудови таблиць і подальшої класифікації пристроїв. Наприклад, MAC-адреса може бути використана для визначення виробника мережевого адаптера, що дозволяє зробити припущення щодо типу пристрою.

У процесі сканування важливо враховувати, що не всі пристрої обов’язково відповідають на ARP-запити. Це стосується деяких типів IoT-обладнання або пристроїв із увімкненими захисними механізмами. Тому реалізація має передбачати повторні запити, аналіз пропущених вузлів та логування всіх спроб сканування для подальшого аналізу.

Також варто зазначити, що у випадку використання в мережі статичної адресації або декількох інтерфейсів, система виявлення має бути здатна працювати з альтернативними джерелами адрес. Це може бути список, сформований вручну, або дані, отримані з DHCP-таблиці, якщо застосунок інтегрується з маршрутизатором.

Зібрані дані подаються у вигляді таблиці, яка є зручною для подальшого перегляду користувачем. Для кожного пристрою зберігається унікальний запис, який містить мінімальний набір технічних даних, потрібних для прийняття рішень. У майбутньому до цієї таблиці можуть бути додані додаткові поля, які з’являються після глибшого сканування.

Тому початкове виявлення пристроїв є фундаментом для всієї подальшої роботи застосунку. Від його якості залежить точність топології, повнота списку пристроїв і ефективність класифікації.

2.3.1 Глибоке сканування (порти, ОС)

Після початкового виявлення пристроїв у локальній мережі важливим етапом є поглиблений аналіз характеристик кожного вузла. Його метою є не лише підтвердження факту присутності пристрою в мережі, але й виявлення його функціонального призначення, аналіз рівня безпеки та ідентифікація потенційних вразливостей.

Глибоке сканування охоплює кілька напрямів – найпоширенішим є сканування відкритих портів. Кожен активний порт свідчить про те, що на пристрої працює певна служба або програма. Наприклад, відкритий порт 80 зазвичай використовується для HTTP-з'єднань, порт 443 – для HTTPS, порт 22 – для SSH, а порт 3389 – для віддаленого доступу через RDP. Аналіз набору портів дозволяє зробити висновок про роль пристрою, його операційну систему або функції в межах мережі.

Іншим важливим елементом є визначення операційної системи пристрою. Це може бути здійснено шляхом аналізу відповідей на спеціально сформовані запити або за допомогою сигнатур, що притаманні конкретним системам. Наприклад, системи на базі Windows, Linux або macOS мають різні ознаки, за якими їх можна відрізнити. Визначення ОС дозволяє не лише класифікувати пристрої, а й оцінити їх вразливість, оскільки деякі версії систем мають відомі недоліки безпеки.

У межах розробленого застосунку для виконання глибокого сканування використовується утиліта Nmap.

Вона підтримує різноманітні режими роботи, включаючи TCP SYN-сканування, UDP-сканування, визначення версій служб і операційних

систем, сканування з використанням скриптів. Nmap дозволяє здійснювати як швидкий поверхневий аналіз, так і розширене глибоке обстеження пристроїв, що знаходяться у межах заданого діапазону

У реалізації застосунку передбачено перемикання між режимами сканування. Користувач може обрати базовий або агресивний режим. У першому випадку здійснюється перевірка стандартних портів із мінімальним впливом на мережу. У другому – задіяють більш глибокі методи з повним збором інформації, включаючи спроби виявлення служб, версій ПЗ та аналіз відповідей пристроїв. Такий підхід дозволяє адаптувати інструмент до різних потреб і рівнів технічної обізнаності користувача.

Результати сканування зберігаються у внутрішніх структурах застосунку і можуть бути відображені у вигляді таблиці або візуального представлення. Для кожного пристрою додається перелік відкритих портів, виявлена ОС, ідентифіковані служби та рівень відповіді. Ці дані можуть бути використані для подальшого аналізу, формування звітів або прийняття рішень щодо ізоляції або оновлення пристрою, а глибоке сканування дозволяє зробити повний технічний портрет кожного вузла в локальній мережі. Воно і доповнює базову інформацію з ARP аналізу та є обов’язковою частиною для повноцінного інструменту візуалізації та контролю ЛКМ.

2.3.2 Класифікація, таблиця, карта мережі, логування

Після виявлення пристроїв і збору базової інформації важливо впорядкувати ці дані так, щоб користувач міг ефективно з ними працювати. У розробленому застосунку реалізовано кілька механізмів, які дозволяють структурувати, візуалізувати й зберігати інформацію про вузли локальної мережі.

Першим з таких механізмів є автоматична класифікація. Вона базується на аналізі MAC-адреси, відкритих портів, типу операційної системи та додаткових мережевих ознак. Наприклад, пристрої з MAC-префіксами виробників смартфонів можуть бути визначені як мобільні пристрої. Якщо у відповіді пристрою виявлено порти, пов’язані з віддаленим доступом або серверними службами, система класифікує його як сервер або адміністраторську станцію.

Результати класифікації подаються у вигляді інтерактивної таблиці. Кожен рядок таблиці містить дані про IP-адресу, MAC-ідентифікатор, тип пристрою, назву виробника, операційну систему, список активних портів і дату останнього виявлення. Користувач може виконати сортування або фільтрацію даних, щоб швидше знайти потрібний вузол. Наприклад, можливо відобразити лише ті пристрої, які не відповідають на запити або мають відкриті порти, пов’язані з ризиками.

Доповненням до табличного подання є графічна візуалізація топології. У застосунку реалізовано схему, яка відображає логічну структуру мережі. Кожен пристрій представлено окремим елементом із власною піктограмою, а між пристроями показано зв’язки. Така візуалізація дозволяє швидко оцінити структуру підключення, виявити нові або ізольовані вузли та зрозуміти загальну картину стану мережі.

Для збереження результатів реалізовано механізм логування. Усі події, пов’язані зі скануванням, фіксуються у текстовому журналі. Це включає дату та час запуску перевірки, кількість виявлених пристроїв, обраний режим сканування та короткий підсумок. У випадку змін у мережі адміністратор може порівняти результати сканування з попередніми перевірками.

Це особливо корисно при виявленні інцидентів безпеки або при щоденному моніторингу. Крім того, реалізовано можливість зберігати результати у форматі, придатному для експорту. Це дозволяє використовувати дані поза межами застосунку або обробляти їх за

допомогою сторонніх інструментів. Наприклад, можна створити звіт для внутрішнього аудиту або передати журнал до відділу інформаційної безпеки. Застосунок забезпечує не лише виявлення пристроїв, а й повноцінну роботу з отриманими даними. Усі ключові функції зведені в єдиний інтерфейс, що дозволяє аналізувати, зберігати й візуально контролювати мережеву інфраструктуру.

2.4 Кросплатформеність, автономність, інтерфейс

Однією з важливих вимог до сучасного програмного забезпечення є його здатність функціонувати в різних операційних середовищах без потреби в суттєвій адаптації. Це особливо актуально для інструментів мережевого аналізу, оскільки адміністратори можуть працювати на різних платформах, включаючи Windows, Linux або macOS. Саме тому доцільно розробляти програмний засіб із підтримкою кросплатформеності, яка гарантує стабільну роботу незалежно від операційної системи користувача.

У межах цього проєкту застосовується мова програмування Python, яка має реалізації для всіх популярних ОС. Для створення графічного інтерфейсу використано бібліотеку PyQt5. Вона забезпечує повноцінну підтримку графічних компонентів, однаковий вигляд застосунку на різних платформах і можливість використання стандартних елементів керування. Це дозволяє зберегти єдиний підхід до навігації, структури вікон і взаємодії з елементами інтерфейсу.

Крім кросплатформеності, важливою є автономність роботи. Багато програм потребують встановлення сторонніх компонентів, доступу до мережі або складної процедури налаштування. У цьому випадку реалізація не залежить від зовнішніх сервісів. Всі механізми сканування виконуються локально, а результати обробляються на самому пристрої користувача. Це

дозволяє запускати застосунок у середовищах із обмеженим інтернет-з’єднанням, ізольованих мережах або в умовах підвищеної безпеки.

Інтерфейс користувача розроблено з урахуванням зручності та доступності. Основне вікно містить кнопку запуску сканування, поле для вибору режиму, панель результатів і кнопку перемикання мови або теми оформлення. Всі елементи згруповано за функціональним принципом, а для кожної дії передбачено підказки. Це робить застосунок придатним для користувачів із мінімальним досвідом роботи з мережевими утилітами [15].

Особливу увагу приділено локалізації. Застосунок підтримує перемикання між українською та англійською мовами. Це дозволяє адаптувати інтерфейс до потреб користувача і застосовувати програму в освітніх закладах, де важливо наочно демонструвати принципи роботи інструментів аналізу мереж. Крім того, доступність українською мовою відповідає вимогам внутрішніх нормативів для програмного забезпечення, що використовується у державних установах. Кросплатформенність, автономність та зрозумілий інтерфейс забезпечують широке застосування розробленого інструменту. Це дозволяє інтегрувати його у будь-яке середовище, не витрачаючи додаткових ресурсів на адаптацію або перенавчання користувачів.

2.4.1 Масштабованість, надійність, безпека

У процесі розробки програмного забезпечення для роботи з локальними мережами важливо враховувати не лише базову функціональність, а й загальні якісні характеристики, зокрема масштабованість, надійність і відповідність принципам безпеки. Ці вимоги є критично важливими для інструментів, які використовуються у динамічному або захищеному середовищі.

Масштабованість забезпечує здатність системи працювати у різних конфігураціях мережі. У найпростішому випадку це може бути домашня мережа з кількома пристроями. У більш складних умовах – офісне або освітнє середовище з десятками вузлів і змінною топологією. Реалізація застосунку дозволяє автоматично визначати розмір підмережі, розраховувати кількість IP-адрес для сканування і обробляти відповіді без зниження продуктивності. Завдяки цьому програма не втрачає ефективності під час роботи в мережах різного масштабу.

Надійність визначається здатністю програми коректно функціонувати у нестабільних або частково обмежених умовах. Наприклад, якщо пристрій не відповідає на ARP-запит або блокує порти, система має продовжити роботу з іншими вузлами і коректно завершити процес. У застосунку реалізовано обробку помилок, тайм-аутів та некоректних відповідей. Результати сканування записуються поступово, що дозволяє уникнути втрати даних у разі аварійного завершення процесу [17, 18].

Особливе значення має безпека використання. Програма не передає інформацію на зовнішні сервери, не потребує доступу до інтернету і не використовує шкідливих методів сканування. Усі запити, що надсилаються до мережі, є технічно обґрунтованими і не порушують політики безпеки більшості організацій. Це дозволяє використовувати її у чутливих середовищах, зокрема в державних установах або навчальних закладах, де важливим є дотримання внутрішніх регламентів. Крім того, забезпечено захист від непередбачених дій користувача. Застосунок не дозволяє змінювати конфігурацію мережі, не вносить жодних записів у системні таблиці маршрутизації або ARP-таблиці. Усі дії є лише спостереженням і аналізом, що забезпечує безпечність використання навіть для недосвідчених користувачів. Тому, програмний засіб відповідає вимогам до якісного, стабільного та безпечного інструменту.

3 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАСОБУ

3.1 Загальна архітектура застосунку

Проєктування програмного забезпечення для аналізу локальної мережі потребує чіткого визначення його внутрішньої структури, розмежування функціональних частин та встановлення логіки взаємодії між модулями. У межах цієї роботи створено багаторівневий застосунок, який виконує сканування, обробку та візуалізацію результатів, класифікацію пристроїв і збереження зібраної інформації [19, 20].

Застосунок реалізовано мовою Python з використанням бібліотек Scapy, Nmap та PyQt5. Його архітектура побудована за модульним принципом. Основні компоненти розподілено відповідно до виконуваних функцій. Це дозволяє спростити підтримку, полегшити тестування і забезпечити можливість подальшого розширення.

Першим рівнем є взаємодія з користувачем. Тут реалізовано графічний інтерфейс, створений засобами PyQt5. Користувач має змогу вибрати режим роботи, запустити сканування, переглянути таблицю з результатами або граф мережі, а також змінити мову та тему оформлення.

Другий рівень відповідає за обробку даних. На цьому етапі відбувається підготовка запитів, запуск інструментів сканування, збір і сортування результатів. Отримані дані систематизуються і передаються до інтерфейсу. Саме тут формується логічна модель мережі, яку буде візуалізовано

На третьому рівні реалізується взаємодія з мережею та доступ до системних ресурсів. Тут відбувається безпосереднє сканування. У разі використання бібліотеки Scapy надсилаються ARP-запити. При активнішому режимі викликається утиліта Nmap, яка виконує глибокий аналіз портів, служб і операційної системи.

Взаємодія між модулями організована через функціональні виклики. Коли користувач активує сканування, інтерфейс ініціює відповідну функцію, яка обирає тип сканера, запускає перевірку, обробляє відповіді та передає їх до відображення. Такий підхід дає змогу розділити логіку обробки і візуалізацію.

За потреби застосунок може працювати без графічного інтерфейсу, наприклад у вигляді окремого скрипта.

У межах архітектури передбачено механізм логування. Після завершення сканування ключові події записуються у файл журналу. Це дає змогу зберігати історію дій, порівнювати результати, проводити аудит та виявляти зміни у мережі.

Уся обробка даних відбувається локально. Програма не потребує підключення до зовнішніх серверів або додаткових компонентів. Це підвищує рівень безпеки, пришвидшує розгортання і робить застосунок придатним для використання у захищених або автономних мережах [21].

Архітектура застосунку є зрозумілою, логічною і адаптованою до різних середовищ. Вона дозволяє ефективно реалізувати всі необхідні функції без ускладнень у підтримці або налаштуванні.

3.2 Основні функціональні блоки системи

Функціональні можливості застосунку охоплюють повний цикл аналізу локальної мережі. Реалізація охоплює виявлення пристроїв, збір технічної інформації, аналіз портів і служб, класифікацію, візуалізацію структури мережі та збереження результатів. Для ефективного виконання цих функцій застосунок поділено на кілька окремих логічних частин. Кожен з функціональних блоків відповідає за окрему групу операцій [22].

Першим є блок, який виконує виявлення пристроїв у локальній мережі. Цей модуль надсилає ARP-запити до всіх можливих IP-адрес у межах

визначеної підмережі. Відповіді обробляються, після чого формується список активних вузлів. Для реалізації цього блоку використано бібліотеку Scapy. Вона забезпечує прямий контроль над мережевими пакетами та надає гнучкість при роботі з протоколом ARP.

Другий блок відповідає за поглиблений аналіз пристроїв. Цей модуль запускає утиліту Nmap для сканування відкритих портів, виявлення запущених служб і визначення операційної системи. У налаштуваннях користувач обирає режим сканування. Програма автоматично адаптує параметри запуску, що дозволяє здійснити як базову перевірку, так і глибокий аналіз [23].

Наступний блок забезпечує класифікацію виявлених пристроїв. Програма визначає тип кожного вузла на основі MAC-адреси, відкритих портів та типу виявленої операційної системи. Наприклад, вузол із MAC-префіксом виробника мережевого обладнання та службами керування може бути класифікований як точка доступу. Аналогічно, пристрій із відкритим портом віддаленого підключення та виявленою Windows-архітектурою класифікується як клієнтська машина.

Наступним етапом є побудова топологічної структури мережі. Застосунок автоматично формує графічне представлення. У ньому кожен пристрій позначено у вигляді окремого елемента. Взаємозв’язки між пристроями визначаються на основі IP-адрес і відповіді на мережеві запити. Схема дозволяє користувачу побачити структуру мережі, взаємне розміщення вузлів і логіку підключень.

В окремому функціональному блоці реалізовано інтерфейс користувача. Усі основні дії доступні через графічні елементи PyQt5. Основне вікно містить елементи керування запуском, відображенням таблиці результатів, перемиканням режимів і мов інтерфейсу. Для кожної дії передбачено зручну навігацію. Візуальне представлення реалізовано з урахуванням потреб як досвідчених користувачів, так і початківців.

Останній блок забезпечує збереження результатів та логування. Після завершення сканування застосунок записує короткий підсумок у файл журналу. Усі основні події, включаючи параметри запуску, кількість знайдених пристроїв і час виконання, фіксуються автоматично. Це дає змогу відновити історію змін у мережі, виявити нетипову активність або підготувати технічний звіт.

Функціональна структура програми побудована таким чином, щоб забезпечити гнучкість, стабільність і розширюваність. Кожен блок працює автономно. Зв’язок між ними підтримується через чітко визначені функції, що дозволяє адаптувати логіку без необхідності зміни всієї системи.

3.3 Модель взаємодії з користувачем

Графічний інтерфейс користувача є одним із найважливіших компонентів застосунку. Саме через нього відбувається вся взаємодія з функціоналом системи. Інтерфейс реалізовано засобами бібліотеки PyQt5. Такий підхід дозволив створити багатовіконну структуру, яка об'єднує логіку керування скануванням, перегляду результатів, візуалізації та налаштувань у межах одного робочого простору. [24].

Основним елементом взаємодії є головне вікно. У ньому зосереджено ключові функції. Користувач може запустити сканування мережі, обрати тип перевірки, переглянути таблицю з результатами або відкрити вікно візуалізації. Всі дії виконуються через кнопки або випадаючі списки. Це дозволяє уникнути складних команд або ручного введення параметрів.

Далі (рис. 3.1) показано інтерфейс керування процесом сканування з перемикачами режимів і теми.

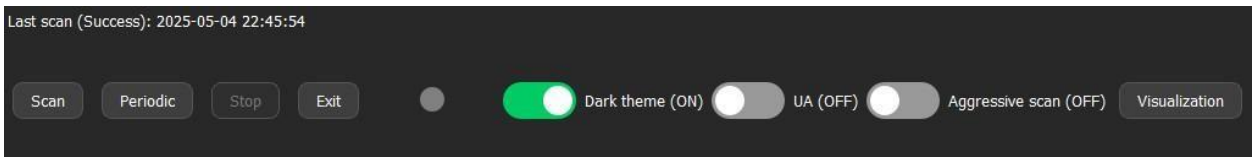

Рисунок 3.1 – Головне вікно застосунку з елементами керування

Головне вікно розділено на кілька логічних областей. У верхній частині розташовано панель керування. Вона містить кнопку запуску, вибір режиму сканування, кнопку доступу до логів і перемикач мови. Центральну частину займає таблиця результатів. У ній відображаються IP-адреси, MAC-ідентифікатори, типи пристроїв, наявність відкритих портів і виявлена операційна система. Кожен рядок таблиці можна виділити для перегляду додаткової інформації.

У нижній частині інтерфейсу розміщено статусну панель. У ній показано загальну кількість виявлених пристроїв, тривалість сканування та повідомлення про помилки або попередження. Це дозволяє користувачу оперативно оцінити стан системи та хід виконання операцій.

Окрім таблиці, застосунок містить окреме вікно для візуального представлення структури мережі. У ньому будується граф, що відображає логічні зв’язки між пристроями. Кожен вузол графа відповідає конкретному пристрою. Його тип позначається піктограмою. Колір елемента сигналізує про статус або рівень активності. Користувач може змінювати масштаб відображення, переглядати властивості окремого вузла або виконати перехід назад до таблиці [26, 27].

Для зручності у роботі з великими мережами реалізовано пошук і фільтрацію. Користувач має змогу відфільтрувати таблицю за типом пристрою, MAC адресою або операційною системою. Це спрощує роботу з великою кількістю даних та дозволяє зосередитися на окремих сегментах.

В інтерфейсі передбачено перемикання мови. Користувач може обрати українську або англійську. Усі написи, підказки, назви полів і повідомлення автоматично змінюються відповідно до обраної мови. Це дає змогу

використовувати застосунок у різних середовищах, включаючи навчальні аудиторії або компанії з міжнародною командою.

Також реалізовано перемикання теми оформлення. Користувач може обрати світлу або темну тему залежно від особистих уподобань або умов роботи. Всі елементи інтерфейсу адаптуються автоматично. Текст, фон і кнопки залишаються читабельними, незалежно від обраного режиму (рис. А.5; А3). Модель взаємодії з користувачем побудована на принципах доступності, логічності та зручності. Інтерфейс дозволяє швидко виконувати всі основні дії без потреби в додатковій підготовці або знаннях у галузі мережевих технологій. Це розширює коло потенційних користувачів та підвищує ефективність застосування програми в умовах реальної роботи [25].

3.4 Структура даних і логіка обробки

Для забезпечення ефективної роботи застосунку важливою складовою є внутрішнє представлення зібраної інформації про пристрої в локальній мережі. Всі дані, що отримуються в процесі сканування, мають бути упорядковані у зручній структурі, яка дозволяє легко здійснювати обробку, класифікацію, відображення та збереження.

Основною одиницею є запис про пристрій. Для кожного вузла, який було виявлено, створюється об’єкт або словник зі стандартним набором полів. До обов’язкових параметрів належать IP-адреса, MAC-адреса, тип пристрою, виробник мережевого адаптера, список відкритих портів, операційна система та мітка часу останнього виявлення. Ця структура дозволяє зберігати як обов’язкову інформацію, так і додаткові параметри, отримані після глибокого сканування [28].

Обробка даних виконується у кілька етапів. Спочатку застосунок збирає базову інформацію за допомогою ARP-запитів. Відповіді

перетворюються на об’єкти пристроїв і додаються до загального списку. Далі, якщо активовано розширене сканування, кожен вузол перевіряється за допомогою утиліти Nmap. Отримані результати доповнюють початкову структуру новими полями, такими як відкриті порти, типи служб та сигнатура операційної системи.

Для зберігання інформації використовується звичайний список або масив структурованих об'єктів. Це дає змогу швидко проходити по всіх елементах під час сортування, фільтрації або візуалізації. Наприклад, для побудови таблиці застосунок просто проходить по списку і витягує потрібні поля. Аналогічно, для побудови графа мережі зчитуються лише IP-адреси, статуси пристроїв і відносини між вузлами, якщо вони були виявлені [29].

У логіці обробки також передбачено оновлення даних. Якщо застосунок уже містить інформацію про пристрій з певною MAC-адресою, але отримано нову IP-адресу або змінено набір відкритих портів, запис оновлюється, а попередні значення архівуються або замінюються. Це дозволяє вести актуальний перелік вузлів і реагувати на зміни в структурі мережі.

При збереженні інформації в лог або при формуванні звіту застосунок проходить по структурі даних і серіалізує кожен запис у текстовий або табличний формат. Підтримується збереження у форматах CSV, TXT або внутрішньому форматі Python для подальшого використання. Це дозволяє легко передати результати іншому користувачу або завантажити їх повторно при наступному запуску.

У межах логіки обробки реалізовано також механізм очищення. При кожному запуску нової сесії попередні результати очищуються. Це запобігає дублюванню записів або змішуванню результатів з різних перевірок. Структура даних у застосунку є гнучкою, прозорою та придатною для виконання всіх поставлених функцій. Логіка обробки

підтримує оновлення, масштабування та збереження, що дозволяє ефективно використовувати програму в різних типах локальних мереж.

3.5 Організація логування та збереження результатів

Збереження результатів роботи є важливим елементом функціональності будь-якого інструменту аналізу мережі. Логування подій дозволяє відновити хронологію операцій, оцінити динаміку змін у мережевій інфраструктурі та здійснювати аудит у разі виникнення інцидентів. У розробленому застосунку реалізовано окрему систему яка працює автоматично без втручання користувача.

Після завершення сканування програмний засіб формує структурований звіт про виконану операцію. У цьому звіті містяться такі дані, як дата та час запуску, тривалість сканування, кількість виявлених пристроїв, тип обраного режиму, успішність відповіді на запити та наявність помилок. Усі ці параметри записуються у файл журналу, який зберігається локально [30].

Формат логів є простим і придатним для читання. Кожна сесія сканування розділена на окремий блок, який включає заголовок, перелік дій та підсумкову інформацію. Записи виконуються у форматі TXT або CSV, що дозволяє відкривати їх у стандартних текстових редакторах або електронних таблицях. Це спрощує роботу з результатами та дає змогу швидко знайти потрібний фрагмент.

Крім логування, застосунок підтримує експортування результатів сканування у вигляді таблиці. Ця функція дозволяє зберегти перелік виявлених пристроїв, разом із усіма зібраними параметрами, у файл окремо від журналу подій. Це зручно для подальшої обробки, формування звітів або передачі результатів іншим користувачам.

У випадку повторного запуску застосунку журнал попередньої сесії зберігається. Таким чином, можна створити архів логів і відстежувати історію сканувань. Це особливо корисно у середовищах, де важливо контролювати зміни у підключеннях або виявляти появу нових пристроїв у критичних сегментах мережі.

Логіка збереження даних побудована так, щоб не створювати зайвого навантаження на систему. Записи виконуються у фоновому режимі без затримок у роботі інтерфейсу. Користувач може не втручатись у процес логування, але має змогу самостійно переглянути лог-файл або експортувати дані за потреби. Реалізована система логування дозволяє забезпечити прозорість, контроль та відтворюваність процесу сканування. Це підвищує надійність інструменту, дозволяє вчасно виявляти зміни у мережевій структурі та підтримувати порядок у веденні внутрішніх журналів і звітів.

3.6 Інтернаціоналізація та налаштування зовнішнього вигляду

Для підвищення зручності користування та адаптації програмного забезпечення до різних категорій користувачів у застосунку реалізовано можливість зміни мови інтерфейсу і теми оформлення. Це дозволяє використовувати програму у навчальному середовищі, в державних установах або у змішаних багатомовних колективах без потреби у сторонній локалізації. Механізм інтернаціоналізації базується на використанні окремих мовних ресурсів [31].

У застосунку передбачено два варіанти мови інтерфейсу — українська та англійська. Зміна мови відбувається через меню налаштувань або при запуску програми. Після перемикання всі написи в інтерфейсі оновлюються автоматично. Це стосується назв кнопок, заголовків вікон, підказок, повідомлень та інструкцій.

Використання англійської мови дозволяє адаптувати програму для іноземних користувачів або використовувати її в міжнародних проєктах. Українська версія відповідає вимогам нормативних документів щодо мовної політики в державному секторі. Обидва варіанти є повноцінними і мають однакову функціональність.

Налаштування зовнішнього вигляду включає можливість перемикання між світлою та темною темою. Цей функціонал реалізовано засобами PyQt5 із використанням стилів. Усі елементи інтерфейсу автоматично змінюють колірну схему відповідно до обраної теми. Текст, тло, кнопки, таблиці та панелі мають відповідний контраст, що дозволяє зберігати зручність сприйняття незалежно від умов освітлення або особистих вподобань користувача0. Користувач може обирати тему вручну або використовувати системні налаштування. Це дає змогу легко інтегрувати застосунок у вже наявне середовище, не змінюючи звичний вигляд інтерфейсу. Теми зберігаються між сеансами, що дозволяє зберігати вибрані параметри без потреби у повторному налаштуванні [32].

Також у межах інтерфейсу реалізовано підтримку масштабування. Всі основні елементи автоматично адаптуються до розміру вікна або дисплея користувача. Це забезпечує коректну роботу на різних пристроях і дозволяє застосовувати програму у середовищах з обмеженим екранним простором. Підтримка мультимовності та гнучке налаштування інтерфейсу підвищують зручність використання програми. Це розширює сферу її застосування і забезпечує відповідність сучасним вимогам до програмних засобів, що орієнтовані на широку аудиторію. Нижче (рис. 3.2) представлено зміну мови інтерфейсу та теми оформлення у процесі використання програми.

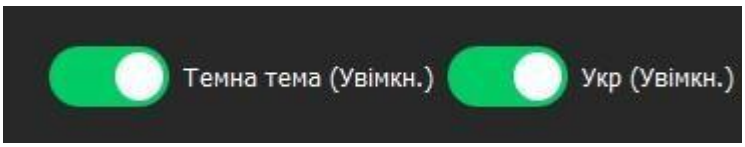

Рисунок 3.2 – Темна тема оформлення з українською мовою інтерфейсу

3.7 Тестування і верифікація роботи

З метою перевірки функціональності, стабільності та надійності розробленого програмного забезпечення було проведено комплексне тестування у різноманітних мережевих сценаріях. Основними критеріями оцінки виступали точність ідентифікації пристроїв, коректність зчитування параметрів (MAC-адрес, відкритих портів, операційних систем), продуктивність при збільшенні кількості вузлів, а також обробка нетипових ситуацій [33,34].

Сценарій 1. Домашня мережа.

Мережева інфраструктура включала маршрутизатор, декілька портативних комп’ютерів, смартфони, IP-камеру та NAS. Застосунок успішно виявив усі активні пристрої, коректно визначив MAC-адреси, ідентифікував операційні системи та відкриті порти.

Побудова графічної моделі мережі підтвердила наявність правильних зв’язків між вузлами та відповідність кожного об’єкта реальному фізичному пристрою.

Сценарій 2. Мережа з поділом на VLAN.

У тестуванні використовувалася корпоративна мережа з ізольованими широкомовними доменами. У межах доступного VLAN застосунок коректно виконав сканування та візуалізацію. Водночас пристрої, які належали до інших VLAN, не були виявлені — що повністю відповідає логіці роботи ARP-протоколу і підтверджує відсутність некоректного міжмережевого втручання.

Сценарій 3. VPN-з’єднання з внутрішньою мережею.

Під час роботи через VPN-з’єднання було виявлено обмеження доступності ARP-таблиць віддалених сегментів. Проте пристрої у межах тунельного інтерфейсу були успішно зчитані, а у випадках недоступності інформації про MAC-адресу застосунок коректно повідомляв про часткові дані. Це свідчить про відповідність обробки сценаріїв із частковою мережею.

Сценарій 4. Віртуальне середовище.

Тестування проводилося у межах локальної віртуалізованої мережі з використанням VMware Workstation. Результати засвідчили, що застосунок здатний виявляти віртуальні пристрої, ідентифікуючи їх за MAC-адресами типу «VMware Inc.». Візуалізація відображала віртуальні вузли як незалежні об’єкти, що підтверджує сумісність із віртуальними середовищами.

Сценарій 5. Робота з великою кількістю вузлів.

У межах навантажувального тесту мережу було розширено до 52 активних пристроїв. Час повного сканування не перевищував 9 секунд, побудова графічної структури займала в середньому 2–3 секунди. Усі операції виконувалися без збоїв, затримок та втрати даних.

Обробка виняткових ситуацій проходила таким чином:

Окремо перевірялись сценарії втрати мережевого з’єднання, відсутності прав доступу до інтерфейсів або некоректного запуску утиліти. У кожному випадку застосунок повідомляв про помилку, коректно завершував роботу та не спричиняв аварійного завершення процесу [35].

Підтвердження результатів проходило таким чином:

Результати сканування зберігалися у вигляді лог-файлу з фіксацією IP-адрес, MAC-ідентифікаторів, відкритих портів та типів операційних систем. Також автоматично формувалася графічна топологія з відображенням зв’язків між пристроями, що може використовуватись як засіб аудиту та візуального аналізу інфраструктури [36].

ВИСНОВКИ

У рамках кваліфікаційної роботи було проаналізовано особливості локальних комп’ютерних мереж, вивчено сучасні підходи до виявлення та класифікації пристроїв, а також розроблено кросплатформений програмний застосунок, який дозволяє візуалізувати структуру локальної мережі. Програмний засіб орієнтований на використання в умовах офісного, навчального та адміністративного середовища.

Застосунок реалізовано на мові Python із використанням бібліотек Scapy, Nmap та PyQt5. Його функціонал охоплює ARP-сканування, глибокий аналіз відкритих портів і служб, визначення операційної системи, автоматичну класифікацію пристроїв, побудову графа мережевої топології, ведення журналу подій та підтримку інтерфейсу з перемиканням мови й теми оформлення.

У ході роботи було розв’язано всі поставлені задачі. Сформульовано функціональні та нефункціональні вимоги до програмного забезпечення. Побудовано архітектуру застосунку, визначено ключові модулі, розроблено інтерфейс користувача та реалізовано всі основні сценарії використання. Особливу увагу приділено зручності, наочності та можливості використання в середовищах із різним рівнем технічної підготовки.

Результати сканування можуть бути представлені у вигляді таблиці або візуального графа, а також збережені для подальшого аналізу. Програма не потребує доступу до Інтернету, працює автономно та не порушує політики безпеки, що дозволяє застосовувати її в захищених мережах.

Розроблений програмний засіб може бути використаний у навчальному процесі, для внутрішнього аудиту, в системному адмініструванні та для попереднього аналізу мережевої інфраструктури. Його відкритість, адаптивність та простота забезпечують гнучке застосування в умовах реальної експлуатації.

Результати роботи апробовано у вигляді 1 тез доповідей під час Міжнародного молодіжного форуму «радіоелектроніка і молодь у XXI столітті».

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Tvoroshenko, I. S. (2021). *Technologies of decision-making in information systems: Textbook*. Kharkiv: KhNURE.

2. Horokhovatskyi, V. O., & Tvoroshenko, I. S. (2021). *Methods of intelligent data analysis and processing: Textbook*. Kharkiv: KhNURE.

3. Kobylin, O. A., & Tvoroshenko, I. S. (2021). *Methods of digital image processing: Textbook*. Kharkiv: KhNURE.

4. Horokhovatskyi, V. O., Tvoroshenko, I. S. (2022). Analysis of multidimensional data as a component set description. *Kharkiv: KhNURE*.

5. Kobylin, O., Horokhovatskyi, V., Tvoroshenko, I., & Peredrii, O. (2020). The application of non-parametric statistics methods in image classifiers based on structural description components. *Telecommunications and Radio Engineering*, *79*(10), 855–863.

6. Daradkeh, Y. I., Tvoroshenko, I., Horokhovatskyi, V., Latiff, L. A., & Ahmad, N. (2021). Development of effective methods for structural image recognition using fuzzy logic. *IEEE Access, 9*, 13417–13428.

7. Daradkeh, Y. I., Horokhovatskyi, V., Tvoroshenko, I., Gadetska, S., & Al-Dhaifallah, M. (2021). Classification of images based on statistical distributions for structural description. *IEEE Access, 9*, 92964–92973.

8. Horokhovatskyi, V. O., Tvoroshenko, I. S., & Peredrii, O. O. (2020). Image classification method modification based on logic processing of bit description vector. *Telecommunications and Radio Engineering, 79*(1), 59–69.

9. Daradkeh, Y. I., Horokhovatskyi, V., Tvoroshenko, I., & Zeghid, M. (2022). Cluster representation of structural image description for classification. *Computers, Materials & Continua, 73*(3), 6069–6084.

10. Tvoroshenko, I., & Yakovleva, O. (2022). Transforming image descriptors into classification features. *Indonesian Journal of Electrical Engineering and Computer Science, 33*(1), 113–125.

11. Pomazan, V., Tvoroshenko, I., & Horokhovatskyi, V. (2023). Emotion recognition app using CNNs. *International Journal of Academic Information Systems Research, 7*(7), 25–36.

12. Horokhovatskyi, V., Tvoroshenko, I., Kobylin, O., & Vlasenko, N. (2023). Cluster-based image search using structural descriptions. *Advances in Electrical and Electronic Engineering, 21*(1), 19–27.

13. Tvoroshenko, I., Pomazan, V., Horokhovatskyi, V., & Kobylin, O. (2023). CNN-based video data classification. *International Journal of Academic and Applied Research*, *7*(11), 134–145.

14. Daradkeh, Y. I., Horokhovatskyi, V., Tvoroshenko, I., Gadetska, S., & Al-Dhaifallah, M. (2023). Statistical analysis models in structural image classification. *IEEE Access, 11*, 126938–126949.

15. Tvoroshenko, I., & Gorokhovatskyi, V. (2022). Hybrid intelligent systems for dynamic data analysis. *International Journal of Engineering and Information Systems, 6*(2), 40–48.

16. Gorokhovatskyi, V., & Tvoroshenko, I. (2023). Distance matrix of structural components as an image classifier tool. *Modern Information Systems*, *7*(1), 5–13.

17. Tvoroshenko, I., & Kuznetsov, M. (2021). The role of testing in mobile application development. In *Proc. of Int. Scientific Conf.*

18. Tvoroshenko, I., & Babochkin, O. (2021). Object identification using keypoint descriptors. In *Proc. of Int. Conf.*

19. Tvoroshenko, I. S., & Zarivchatskyi, R. (2020). Object search in video streams. In *Proc. VI Int. Sci. Practic. Conf.*, Milan, pp. 500–505.

20. Gadetska, S. V., Horokhovatskyi, V. O., Stiahlyk, N. I., & Vlasenko, N. V. (2021). Binary descriptors in image classification. *Radio Electronics, Computer Science, Control, 4*, 58–68.

21. Tvoroshenko, I., & Gorokhovatskyi, V. (2022). The application of hybrid intelligence systems for dynamic data analysis. International Journal of Engineering and Information Systems, 6(2), 40–48.

22. Kobylin, I., & Lyashenko, V. (2016). Contour detection and allocation for cytological images using wavelet analysis methodology. International Journal of Advance Research in Computer Science and Management Studies, 4(6), 85–94.

23. Kobylin, I., & Lyashenko, V. (2016). Contrast modification as a tool to study the structure of blood components. Journal of Environmental Science, Computer Science and Engineering & Technology, 5(3), 150–160.

24. Kobylin, I., & Lyashenko, V. (2016). Using the methodology of wavelet analysis for processing images of cytology preparations. National Journal of Medical Research, 6(2), 98–102.

25. Kobylin, I., & Lyashenko, V. (2016). The methodology of wavelet analysis as a tool for cytology preparations image processing. Cukurova Medical Journal, 41(3), 453–463.

26. Tvoroshenko, I., & Gorokhovatskyi, V. (2021). Development of an intelligent system for monitoring and diagnosing the state of complex technical objects. Eastern-European Journal of Enterprise Technologies, 2(9), 6–14.

27. Tvoroshenko, I., & Gorokhovatskyi, V. (2020). Application of hybrid intelligent systems for the analysis of dynamic processes. Control Systems and Computers, 1, 3–10.

28. Tvoroshenko, I., & Gorokhovatskyi, V. (2019). Development of a decision support system based on hybrid intelligent technologies. Information Technologies and Learning Tools, 71(3), 1–15.

29. Kobylin, I., & Lyashenko, V. (2017). Wavelet analysis of cytological preparations image in different color systems. Open Access Library Journal, 4, 1–9.

30. Tvoroshenko, I., & Gorokhovatskyi, V. (2018). Intelligent decision support system for diagnosing the state of technical objects. Bulletin of the National Technical University «KhPI». Series: Information and Modeling Systems, 1(1245), 112–117.

31. Kobylin, I., & Lyashenko, V. (2016). Using the properties of wavelet coefficients of time series for image analysis and processing. Journal of Computer Sciences and Applications, 4(1), 27–34.

32. Tvoroshenko, I., & Gorokhovatskyi, V. (2017). Development of a hybrid intelligent system for monitoring and diagnosing the state of complex technical objects. Information Technologies and Learning Tools, 60(4), 1–15.

33. Kobylin, I., & Lyashenko, V. (2016). The use of wavelet analysis for processing images of cytological preparations. Journal of Medical Informatics and Technologies, 27, 1–6.

34. Tvoroshenko, I., & Gorokhovatskyi, V. (2016). Application of hybrid intelligent systems for analyzing the state of complex technical objects. Control Systems and Computers, 2, 3–10.

35. Kobylin, I., & Lyashenko, V. (2016). Wavelet analysis of cytological preparations image in different color systems. Open Access Library Journal, 3, 1–9.

36. Balabukha, I. O. (2025). Scanning and analyzing local network devices using ARP and Nmap. In *Proceedings of the XXIX International Youth Forum "Radio Electronics and Youth in the XXI Century"* (pp. 6–7). Kharkiv National University of Radio Electronics.