

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

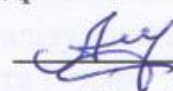
ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра
на тему: «Проект програмного сніффера для дослідження
якості та безпеки Wi-Fi мереж»

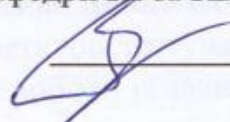
Виконала: студентка 4-го курсу групи КБ-22-2
спеціальності 125 «Кібербезпека»

 Вікторія КАПРАНЧУК

Керівник: доцент кафедри КІ та КБ, к.т.н.

 Олексій ШЕЛУХА

Рецензент: доцент кафедри КІ та КБ, к.т.н.

 Олександр ПІРОГ

Розділ	Консультант	Завдання	Підпис, дата
1	Бродський Ю.І.	02.02.2026	02.02.2026
2	Бродський Ю.І.	02.02.2026	02.02.2026
3	Бродський Ю.І.	02.02.2026	02.02.2026

Житомир – 2026

Вікторія КАПРАНЧУК
Олексій ШЕЛУХА

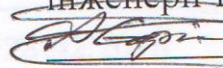
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки
Спеціальність 125 «Кібербезпека»
Освітня програма «Кібербезпека»

ЗАТВЕРДЖУЮ

Завідувач

кафедри комп'ютерної

інженерії та кібербезпеки

 Андрій Єфіменко

2 лютого 2026 р.

ЗАВДАННЯ

на випускню кваліфікаційну роботу бакалавра

Студент: Капранчук Вікторія Миколаївна

Тема роботи: «Проект програмного сніффера для дослідження якості та безпеки Wi-Fi мереж»

затверджена Наказом університету від 2 лютого 2026 р. № (58/с)

Термін здачі студентом закінченої роботи 8 червня 2026 р.

Вихідні дані роботи:

- Характеристики Wi-Fi мереж;
- Результати аналізу мережевої активності, статистика мережевого трафіку;
- Методи та програмні засоби визначення якості й безпеки Wi-Fi мереж;
- Технології мережевого моніторингу та аналізу трафіку;

Обладнання та програмне забезпечення:

1. Персональний комп'ютер або ноутбук з підтримкою віртуалізації.
2. Операційна система Windows 10/11.
3. Віртуальна машина VirtualBox.
4. Операційна система Kali Linux.
5. Мова програмування Python 3.
6. Бібліотека Scapy для захоплення та аналізу пакетів.
7. Мережеві утиліти Wireshark, tcpdump та Nmap.
8. Текстовий редактор Mousepad або Visual Studio Code.
9. Локальна мережа з доступом до Інтернету для тестування програмного засобу.

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

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Бродський Ю.Б.	02.02.2026	02.02.2026
2	Бродський Ю.Б.	02.02.2026	02.02.2026
3	Бродський Ю.Б.	02.02.2026	02.02.2026

Студент

Керівник




Вікторія КАПРАНЧУК

Олексій ШЕЛУХА

Календарний план

№ з/п	Назви етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Затвердження кваліфікаційної роботи	2.02.2026	Виконано
2	Аналіз літературних джерел та нормативної бази	15.02.2026	Виконано
3	Дослідження принципів роботи Wi-Fi мереж та загроз безпеки	28.02.2026	Виконано
4	Аналіз методів моніторингу та аналізу мережевого трафіку	10.03.2026	Виконано
5	Проектування архітектури програмного сніффера	20.03.2026	Виконано
6	Налаштування середовища розробки Kali Linux та Python	31.03.2026	Виконано
7	Реалізація модуля захоплення та фільтрації пакетів	25.04.2026	Виконано
8	Реалізація модуля аналізу якості Wi-Fi мереж	07.05.2026	Виконано
9	Реалізація модуля аналізу безпеки мереж	15.05.2026	Виконано
10	Тестування програмного засобу та аналіз результатів	25.05.2026	Виконано
11	Оформлення пояснювальної записки та підготовка до захисту	05.06.2026	Виконано

Студент Капранчук В.М.

Керівник Шелуха О.О.

ККІК-КР-Б-125-Б-ПЗ

ФІКТ, Гр. КБ-22-2

АНОТАЦІЯ

Кваліфікаційна робота бакалавра присвячена розробці програмного сніффера для дослідження якості та безпеки Wi-Fi мереж. У роботі розглянуто принципи функціонування бездротових мереж, основні показники якості Wi-Fi з'єднання та сучасні загрози безпеці бездротових мереж.

Метою роботи є розробка програмного засобу для захоплення, аналізу та обробки мережевого трафіку з метою оцінки якості та безпеки Wi-Fi мереж.

У розділі 1 проведено аналіз принципів функціонування Wi-Fi мереж, показників якості бездротового зв'язку, основних загроз безпеці, методів моніторингу мережевого трафіку та особливостей використання середовища Kali Linux для аналізу мережевої активності.

У розділі 2 висвітлено процес проєктування програмного сніффера, визначено функціональні вимоги до програмного засобу, розроблено архітектуру системи, описано методику збору мережевих даних, а також критерії оцінювання якості та безпеки Wi-Fi мереж.

У розділі 3 відображено процес розгортання програмної реалізації сніффера, налаштування середовища розробки та тестування окремих модулів. Проведено аналіз мережевої активності та оцінювання безпеки мережевого середовища. Також виконано моделювання ознак потенційно небезпечної активності, зокрема DoS-подібного навантаження, ARP-активності та аналізу DNS-запитів, що дозволило оцінити можливість використання розробленого засобу для виявлення ознак малонадійних або потенційно небезпечних Wi-Fi мереж.

Пояснювальна записка містить 85 сторінки, 51 рисунок, 14 таблиць, список використаних джерел із 30 найменувань на 3 сторінках.

Ключові слова: WI-FI, СНІФФЕР, МЕРЕЖЕВИЙ ТРАФІК, KALI LINUX, PYTHON, SCAPY, КІБЕРБЕЗПЕКА, АНАЛІЗ ТРАФІКУ, БЕЗДРОВОТА МЕРЕЖА.

					ККІК<.КР.Б – 125 – 6 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Проект програмного сніффера для дослідження якості та безпеки Wi-Fi мереж	Літ.	Арк.	Аркуші
Розроб.		Капричук В.М.		08.06				
Керівник		Шелуха О.О.		08.06			4	102
Реценз.		Пірог О.В.		08.06				
Н. контр.		Шелуха О.О.		08.06				
Затверд.		Єфіменко А.А.		08.06.26		ФІКТ, Гр. КБ-22-2		

ANNOTATION

The bachelor qualification work is devoted to the development of a software sniffer for studying the quality and security of Wi-Fi networks. The paper considers the principles of wireless network operation, the main indicators of Wi-Fi connection quality, and modern security threats to wireless networks.

The purpose of the work is to develop a software tool for capturing, analyzing, and processing network traffic in order to evaluate the quality and security of Wi-Fi networks.

Chapter 1 presents an analysis of the operating principles of Wi-Fi networks, wireless communication quality indicators, major security threats, network traffic monitoring methods, and the features of using the Kali Linux environment for network activity analysis.

Chapter 2 describes the design process of the developed packet sniffer, defines the functional requirements for the software solution, presents the system architecture, outlines the methodology for collecting network data, and specifies the criteria used for evaluating the quality and security of Wi-Fi networks.

Chapter 3 presents the deployment and implementation of the packet sniffer, the configuration of the development environment, and the testing of individual software modules. Network activity analysis and network security assessment were performed. In addition, the simulation of potentially dangerous network activity was conducted, including DoS-like traffic generation, ARP activity analysis, and DNS request monitoring. This made it possible to evaluate the applicability of the developed software tool for detecting indicators of unreliable or potentially insecure Wi-Fi networks.

The explanatory note contains 85 pages, 51 figures, 14 tables, and a list of references including 30 sources on 3 pages.

Keywords: WI-FI, SNIFFER, NETWORK TRAFFIC, KALI LINUX, PYTHON, SCAPY, CYBERSECURITY, TRAFFIC ANALYSIS, WIRELESS NETWORK.

					ККІК<.КР.Б – 125 – 6 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Капранчук В.М.			Проект програмного сніффера для дослідження якості та безпеки Wi-Fi мереж	Лім.	Арк.	Аркушів
Керівник		Шелуха О.О.						101
Реценз.		Пірог О.В.				ФІКТ, Гр. КБ-22-2		
Н. контр.		Шелуха О.О.						
Затверд.		Єфіменко А.А.						

Зміст

ВСТУП	8
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ ФУНКЦІОНУВАННЯ ТА АНАЛІЗУ WI-FI МЕРЕЖ.....	9
1.1 Принцип роботи Wi-Fi мереж.....	9
1.2 Показники якості бездротового зв'язку.....	12
1.3 Загрози безпеці Wi-Fi мереж.....	16
1.4 Методи моніторингу та аналізу мережевого трафіку.....	20
1.5 Використання Kali Linux у сфері кібербезпеки	25
1.6 Обґрунтування необхідності розробки програмного сніффера для аналізу Wi-Fi мереж.....	30
РОЗДІЛ 2 ПРОЄКТУВАННЯ ПРОГРАМНОГО СНІФФЕРА ДЛЯ АНАЛІЗУ WI-FI МЕРЕЖ	33
2.1 Аналіз вимог до програмного засобу	33
2.2 Проєктування архітектури програмного сніффера.....	35
2.3 Методика збору мережевих даних.....	38
2.4 Методика аналізу якості Wi-Fi мереж.....	41
2.5 Методика оцінювання безпеки Wi-Fi мереж.....	43
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ СНІФФЕРА ДЛЯ АНАЛІЗУ WI-FI МЕРЕЖ.....	47
3.1 Реалізація структури програмного засобу	47
3.2 Реалізація захоплення та аналізу пакетів.....	50
3.3 Реалізація дослідження мережевої активності.....	55
3.4 Реалізація аналізу мережевої інфраструктури та безпеки	64
3.5 Дослідження мережевих пакетів під час взаємодії із зовнішніми вебресурсами.....	68
3.6 Оцінювання ефективності розробленого програмного засобу.....	82
ВИСНОВКИ.....	88
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	90
ДОДАТКИ.....	93

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – Application Programming Interface

BSSID – Basic Service Set Identifier, MAC-адреса точки доступу

CLI – Command Line Interface

CSV -Comma-Separated Values

DoS – Denial of Service, атака відмови в обслуговуванні

GUI – Graphical User Interface

IEEE – Institute of Electrical and Electronics Engineers

IP – Internet Protocol

JSON – JavaScript Object Notation

Kali Linux – спеціалізований дистрибутив Linux для задач кібербезпеки

MAC – Media Access Control

MITM – Man-in-the-Middle, атака «людина посередині»

ОС – операційна система

PCAP – Packet Capture, формат збереження мережових пакетів

ПЗ – програмне забезпечення

PyShark – бібліотека Python для аналізу мережевого трафіку

QoS – Quality of Service, якість обслуговування мережі

RSSI – Received Signal Strength Indicator, рівень сигналу

Scapy – бібліотека Python для аналізу мережових пакетів

SNR – Signal-to-Noise Ratio, співвідношення сигнал/шум

SSID – Service Set Identifier, назва бездротової мережі

TCP – Transmission Control Protocol

UDP – User Datagram Protocol

WEP – Wired Equivalent Privacy

Wi-Fi – Wireless Fidelity, технологія бездротової передачі даних

WPA – Wi-Fi Protected Access

WPA2 - Wi-Fi Protected Access 2

WPA3 - Wi-Fi Protected Access 3

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

У сучасних умовах Wi-Fi мережі широко використовуються для передачі даних у домашніх, корпоративних та публічних мережах. Разом із розвитком бездротових технологій зростає кількість загроз безпеці та проблем, пов'язаних із якістю мережевого з'єднання. Недостатній рівень захисту Wi-Fi мереж може призвести до перехоплення трафіку, несанкціонованого доступу та втрати даних. Тому актуальним є створення програмних засобів для моніторингу, аналізу та оцінювання безпеки й якості бездротових мереж.

Метою кваліфікаційної роботи є розробка програмного сніффера для дослідження якості та безпеки Wi-Fi мереж із використанням мови програмування Python та середовища Kali Linux

Визначена мета обумовлює наступні завдання:

- проаналізувати основні показники якості та безпечності Wi-Fi-мереж;
- розробити проєкт методики моніторингу та аналізу мережевого трафіку, зокрема програмного сніффера;
- реалізувати модулі аналізу та безпеки мережевого трафіку;
- провести тестування програмного засобу.

Об'єктом дослідження є процес моніторингу та аналізу трафіку бездротових Wi-Fi мереж.

Предметом дослідження є методи та програмні засоби захоплення, фільтрації та аналізу мережевого трафіку Wi-Fi мереж із використанням Python та Kali Linux.

У процесі виконання кваліфікаційної роботи було розроблено проєкт програмного сніффера для аналізу мережевого трафіку Wi-Fi мереж. Програмний засіб забезпечує захоплення пакетів, аналіз показників якості мережі та виявлення потенційних загроз безпеці.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ ФУНКЦІОНУВАННЯ ТА АНАЛІЗУ WI-FI МЕРЕЖ

1.1 Принцип роботи Wi-Fi мереж

Wi-Fi – це технологія бездротового передавання даних, яка базується на стандартах IEEE 802.11 та забезпечує обмін інформацією між пристроями через радіоканал [1]. Wi-Fi використовується у домашніх, корпоративних та публічних мережах для забезпечення доступу до локальної мережі та Інтернету без використання кабельного з'єднання.

Основними компонентами Wi-Fi мережі є:

- точка доступу (Access Point);
- клієнтські пристрої;
- маршрутизатор;
- мережевий адаптер.



Рисунок 1.1 – Загальна структура Wi-Fi мережі

На рисунку 1.1 показано основні компоненти Wi-Fi мережі та взаємодію між клієнтськими пристроями, точкою доступу та мережею Інтернет. Схема демонструє принцип бездротової передачі даних у режимі Infrastructure.

Точка доступу виконує роль центрального вузла мережі та забезпечує взаємодію між клієнтами й мережею Інтернет. Клієнтськими пристроями можуть бути ноутбуки, смартфони, планшети та інші пристрої з підтримкою Wi-Fi.

Wi-Fi мережі можуть функціонувати в декількох частотних діапазонах, основними з яких є 2,4 ГГц, 5 ГГц та 6 ГГц [6, 7].

Таблиця 1.1

Порівняння характеристик діапазонів Wi-Fi

Параметр	2,4 ГГц	5 ГГц
Радіус дії	Більший	Менший
Швидкість передачі даних	Нижча	Вища
Стійкість до перешкод	Нижча	Вища
Кількість каналів	3 основні канали	Більша кількість каналів
Типові сфери використання	Домашні мережі, IoT-пристрої	Відеострімінг, онлайн-ігри, офісні мережі

Як видно з таблиці 1.1, діапазон 2,4 ГГц забезпечує більшу зону покриття, проте характеризується нижчою швидкістю передачі даних та більшою чутливістю до перешкод. Діапазон 5 ГГц забезпечує вищу пропускну здатність і кращу стійкість до завад, однак має менший радіус дії [6, 7].

Основні стандарти IEEE 802.11 наведено у таблиці 1.2.

Таблиця 1.12

Основні стандарти Wi-Fi

Стандарт	Частота	Максимальна швидкість
802.11b	2.4 ГГц	11 Мбіт/с
802.11g	2.4 ГГц	54 Мбіт/с
802.11n	2.4/5 ГГц	До 600 Мбіт/с
802.11ac	5 ГГц	До 6.9 Гбіт/с
802.11ax	2.4/5/6 ГГц	До 9.6 Гбіт/с

Основні характеристики сучасних стандартів Wi-Fi визначають максимальну пропускну здатність мережі, підтримувані частотні діапазони та ефективність використання радіоканалу [6, 7].

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

Для ідентифікації мережі використовується SSID – назва бездротової мережі. Кожна точка доступу має унікальну MAC-адресу, яка називається BSSID.

Існують два основних режими роботи Wi-Fi мереж:

- Infrastructure Mode;
- Ad-Hoc Mode.

У режимі Infrastructure усі клієнти взаємодіють через точку доступу. У режимі Ad-Hoc пристрої обмінюються даними напряму без використання маршрутизатора.

Передавання інформації здійснюється за допомогою мережевих клієнтів. Пакети містять:

- службову інформацію;
- MAC-адреси;
- IP-адреси;
- корисні дані.



Рисунок 1.3 – Структура кадру IEEE 802.11

На рисунку 1.3 зображено структуру кадру IEEE 802.11, який використовується під час передачі даних у Wi-Fi мережах. Кадр містить службові поля, адресну інформацію та корисні дані. Службові поля забезпечують керування процесом обміну інформацією між мережевими пристроями, а поля адресації використовуються для визначення відправника та отримувача даних.

Корисне навантаження кадру може містити пакети різних мережевих протоколів, зокрема IP, TCP, UDP, DNS та інших. Саме ці пакети передають інформацію між вузлами мережі та формують мережевий трафік, який може бути використаний для аналізу роботи бездротової мережі.

Для дослідження мережевого трафіку та вмісту пакетів використовуються спеціалізовані програмні засоби – сніфери. Вони дозволяють здійснювати перехоплення пакетів, аналізувати використовувані протоколи, визначати характеристики мережевих з'єднань та виявляти потенційно небезпечну активність. Отримані за допомогою сніфферів дані можуть використовуватися для оцінювання якості та безпеки функціонування Wi-Fi мереж.

Одним із важливих напрямків аналізу мережевого середовища є оцінювання параметрів якості бездротового зв'язку. Для цього використовуються спеціальні показники, які характеризують стабільність, швидкість та надійність передавання даних між мережевими пристроями.

1.2 Показники якості бездротового зв'язку

Якість Wi-Fi мережі характеризується набором параметрів, які впливають на стабільність та швидкість передавання даних.

Одним із головних показників якості бездротового зв'язку є RSSI (Received Signal Strength Indicator), який характеризує рівень потужності прийнятого сигналу [10].

Таблиця 1.3

Оцінка рівня сигналу RSSI

RSSI	Якість сигналу
-30...-50 dBm	Відмінна
-50...-67 dBm	Хороша
-67...-80 dBm	Задовільна
Нижче -80 dBm	Низька

Ще одним важливим параметром є SNR (Signal-to-Noise Ratio), що визначає співвідношення корисного сигналу до рівня шуму в каналі зв'язку [9].

До основних характеристик якості Wi-Fi належать:

- пропускна здатність;
- затримка передачі даних;
- packet loss;
- стабільність підключення;
- завантаженість каналу.

Пропускна здатність визначає максимальний обсяг даних, який може бути переданий за одиницю часу.

Під час оцінювання пропускної здатності необхідно враховувати різницю між теоретичною та реальною швидкістю передачі даних. Виробники мережевого обладнання зазвичай вказують максимальні значення швидкості, які можуть бути досягнуті лише за ідеальних умов. У реальних мережах на швидкість впливають відстань до точки доступу, рівень завад, кількість одночасно підключених клієнтів та характеристики обладнання. У результаті фактична швидкість може бути суттєво нижчою за заявлену.

У сучасних бездротових мережах швидкість передачі даних також залежить від стандарту IEEE 802.11, який використовується пристроями. Новіші стандарти підтримують ширші канали зв'язку, ефективніші алгоритми модуляції та механізми одночасного обслуговування декількох користувачів. Це дозволяє збільшувати продуктивність мережі навіть за великої кількості активних підключень.

Затримка (latency) – це час передавання пакета між пристроями. Висока затримка негативно впливає на відеозв'язок, онлайн-ігри та VoIP.

Packet Loss – це втрата пакетів під час передавання інформації. Причинами втрати пакетів можуть бути:

- перевантаження мережі;
- слабкий сигнал;
- перешкоди;
- помилки маршрутизації.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

Втрата пакетів негативно впливає на стабільність мережевого з'єднання та може призводити до погіршення якості передачі даних [12]. Навіть незначний відсоток втрат може призводити до погіршення якості сервісів реального часу. Наприклад, під час відеоконференції втрачені пакети можуть спричиняти спотворення звуку або зображення, а в мережевих іграх – затримки та нестабільність роботи клієнтських застосунків.

Крім того, втрата пакетів часто є непрямою ознакою інших проблем у мережі. Вона може свідчити про перевантаження точки доступу, конфлікти каналів, апаратні несправності або помилки налаштування мережевого обладнання. Тому даний показник використовується як один із критеріїв комплексної оцінки якості Wi-Fi мереж.

На якість Wi-Fi мережі впливають:

- відстань до точки доступу;
- кількість підключених пристроїв;
- бетонні стіни та фізичні перешкоди;
- електромагнітні завади;
- перевантаження каналів.

Для отримання достовірної інформації про стан мережі застосовуються спеціалізовані програмні засоби моніторингу. Вони дозволяють збирати статистику щодо параметрів сигналу, рівня завантаження каналів, швидкості передачі даних та характеристик мережевого трафіку. Отримані результати використовуються для оцінювання ефективності роботи мережі та пошуку причин можливих проблем.

Використання програмних засобів моніторингу особливо важливе під час дослідження бездротових мереж, оскільки значна частина параметрів не може бути визначена лише за суб'єктивними відчуттями користувача. Аналіз фактичних значень показників дозволяє отримати об'єктивну оцінку якості функціонування мережі.

Для оцінювання якості Wi-Fi використовуються спеціалізовані програмні засоби аналізу мережевого трафіку та параметрів бездротового зв'язку. Одним із най-

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		14

більш поширених інструментів є Wireshark, який забезпечує захоплення та детальний аналіз мережевих пакетів. За його допомогою можна досліджувати структуру трафіку, визначати використовувані протоколи, аналізувати затримки передачі даних та виявляти втрати пакетів.

Утиліта Tcpdump використовується для захоплення мережевого трафіку з командного рядка. Даний інструмент дозволяє здійснювати моніторинг мережевої активності в режимі реального часу та аналізувати пакети без використання графічного інтерфейсу.

Програмний комплекс Kismet призначений для моніторингу бездротових мереж. Він дозволяє виявляти точки доступу, визначати параметри бездротового середовища, аналізувати рівень сигналу та контролювати мережеву активність клієнтських пристроїв.

Бібліотека Scapy використовується для програмного аналізу та обробки мережевих пакетів. На її основі можуть створюватися власні інструменти моніторингу мережі, аналізатори трафіку та програмні сніффери. Саме дана бібліотека була використана під час реалізації програмного засобу в межах кваліфікаційної роботи.

Для оцінювання параметрів бездротового середовища також можуть використовуватися спеціалізовані засоби аналізу Wi-Fi мереж, зокрема Insider. Такі інструменти дозволяють визначати рівень сигналу, завантаженість радіоканалів, рівень перешкод та інші характеристики, які впливають на якість бездротового зв'язку.

Таким чином, сучасні засоби аналізу мереж поєднують можливості дослідження мережевого трафіку та оцінювання параметрів бездротового середовища, що дозволяє комплексно визначати рівень якості та безпеки функціонування Wi-Fi мереж.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

1.3 Загрози безпеці Wi-Fi мереж

Wi-Fi мережі є вразливими до різних типів атак через використання відкритого радіоканалу.

Основними загрозами є:

- несанкціонований доступ;
- перехоплення трафіку;
- підбір паролів;
- підробка точки доступу;
- атаки відмови в обслуговуванні.

Для захисту бездротових мереж використовуються протоколи WEP, WPA, WPA2 та WPA3, які забезпечують різні механізми автентифікації та шифрування даних [8].

Таблиця 1.4

Характеристика протоколів захисту Wi-Fi

Протокол	Особливості	Рівень безпеки
WEP	Застаріле шифрування	Низький
WPA	Покращене шифрування	Середній
WPA2	AES-шифрування	Високий
WPA3	Сучасний захист	Дуже високий

Протокол WPA3 є найбільш сучасним рішенням для захисту бездротових мереж та забезпечує підвищену стійкість до атак підбору паролів [24].

Вибір механізму шифрування безпосередньо впливає на рівень захищеності бездротової мережі. Протягом розвитку технології Wi-Fi використовувалися різні алгоритми захисту, які поступово вдосконалювалися у відповідь на появу нових методів атак. Сучасні протоколи забезпечують значно вищий рівень безпеки порівняно з попередніми рішеннями, однак навіть вони потребують правильного налаштування та використання надійних паролів доступу.

Наявність сучасного протоколу шифрування не гарантує повного захисту мережі. На практиці успішність багатьох атак залежить від помилок конфігурації, людського фактора та недостатнього контролю мережевої інфраструктури. Саме тому питання інформаційної безпеки повинно розглядатися комплексно.

WEP має значні криптографічні вразливості та може бути зламаний за короткий час. WPA2 довгий час був основним стандартом безпеки, однак також має окремі вразливості. WPA3 забезпечує підвищений рівень захисту.

Однією з найпоширеніших атак є brute force атака – підбір пароля до Wi-Fi мережі.

Атаки перебору паролів залишаються актуальними для мереж із недостатньо складними механізмами автентифікації [25].

Основні типи атак на Wi-Fi мережі наведено у таблиці 1.4.

Таблиця 1.5

Основні атаки на Wi-Fi мережі

Тип атаки	Опис
Brute Force	Підбір пароля
MITM	Перехоплення трафіку
Evil Twin	Підроблена точка доступу
Deauthentication	Відключення клієнтів
DoS	Перевантаження мережі
Packet Sniffing	Аналіз і перехоплення пакетів
WPS Attack	Атака через WPS
Rogue AP	Несанкціонована точка доступу

MITM (Man-in-the-Middle) – атака, під час якої зловмисник перехоплює трафік між клієнтом та точкою доступу.



Рисунок 1.4 – Принцип роботи MITM-атаки

На рисунку 1.4 продемонстровано схему MITM-атаки, під час якої зловмисник перехоплює мережевий трафік між користувачем та точкою доступу.

Атаки типу Man-in-the-Middle є одними з найпоширеніших способів перехоплення інформації під час передавання даних мережею [23].

Evil Twin – створення фальшивої точки доступу з аналогічним SSID для перехоплення даних користувачів.



Рисунок 1.5 – Схема атаки Evil Twin

На рисунку 1.5 показано принцип роботи атаки Evil Twin, при якій створюється підроблена точка доступу для перехоплення даних користувача.

Підроблені точки доступу можуть використовуватися для викрадення облікових даних та іншої конфіденційної інформації користувачів [16].

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

Deauthentication Attack – надсилання deauth-пакетів для примусового відключення клієнтів від мережі.

DoS (Denial of Service) – атака, спрямована на порушення працездатності мережі шляхом перевантаження.

Особливістю атак типу DoS є відсутність необхідності отримувати доступ до внутрішніх ресурсів мережі. Основною метою таких атак є створення умов, за яких легітимні користувачі не можуть використовувати мережеві сервіси. Для бездротових мереж подібні атаки є особливо небезпечними через обмеженість радіочастотного ресурсу та залежність від якості сигналу.

У корпоративних мережах наслідками DoS-атак можуть бути порушення роботи інформаційних систем, зниження продуктивності персоналу та недоступність критично важливих сервісів. Тому механізми виявлення аномальної активності є важливою складовою систем захисту.

WPS Attack – використання вразливостей механізму WPS для підбору PIN-коду та отримання доступу до мережі.

Packet Sniffing – процес перехоплення та аналізу мережевих пакетів.

Несанкціоновані точки доступу створюють додаткові ризики для безпеки мережевої інфраструктури [13].

Одним із найефективніших способів протидії більшості атак є постійний аудит мережі та контроль її параметрів безпеки. Регулярний аналіз журналів подій, перевірка підключених пристроїв та оцінювання стану мережевого обладнання дозволяють своєчасно виявляти потенційні загрози та мінімізувати ризики компрометації мережі.

Практика показує, що більшість успішних атак стають можливими через помилки конфігурації або недостатню увагу до питань інформаційної безпеки. Саме тому сучасні підходи до захисту Wi-Fi мереж поєднують технічні засоби захисту, моніторинг мережевого середовища та організаційні заходи.

Для захисту Wi-Fi мережі рекомендується:

- використовувати WPA3;
- вимикати WPS;

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

- встановлювати складні паролі;
- оновлювати прошивку маршрутизатора;
- використовувати VPN;
- контролювати підключені пристрої.

1.4 Методи моніторингу та аналізу мережевого трафіку

Моніторинг мережевого трафіку є важливим елементом забезпечення безпеки та стабільності функціонування мережі. Аналіз мережевої активності дозволяє своєчасно виявляти збої в роботі обладнання, перевантаження каналів зв'язку, а також потенційні загрози інформаційній безпеці.

Залежно від способу отримання інформації про стан мережі розрізняють пасивний та активний моніторинг.

Пасивний моніторинг передбачає спостереження за мережевим трафіком без безпосереднього втручання в роботу мережі. У цьому випадку аналізуються пакети, які вже передаються між вузлами мережі, без генерації додаткових запитів або службового навантаження. Основною перевагою пасивного моніторингу є відсутність впливу на продуктивність мережі та можливість отримання об'єктивної інформації про реальний характер мережевої активності. Для реалізації пасивного моніторингу широко використовуються програмні сніффери та аналізатори трафіку, зокрема Wireshark, Tcpdump, Kismet та Scapy.

Активний моніторинг базується на формуванні спеціальних мережевих запитів та подальшому аналізі отриманих відповідей. Для цього використовуються такі інструменти, як ping, traceroute, Nmap та інші засоби діагностики мережі. Активний моніторинг дозволяє визначати доступність мережевих вузлів, вимірювати затримку передавання даних, оцінювати пропускну здатність каналів зв'язку та виявляти відкриті мережеві служби.

Обидва підходи мають власні переваги та сфери застосування. Пасивний моніторинг забезпечує детальний аналіз фактичного мережевого трафіку, тоді як активний моніторинг дозволяє отримувати додаткову інформацію про стан мере-

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

жевої інфраструктури шляхом виконання контрольних запитів. У практичній діяльності фахівців з кібербезпеки дані методи часто використовуються комплексно для отримання максимально повної інформації про функціонування мережевого середовища.

Таблиця 1.5

Порівняння активного та пасивного моніторингу мережевого трафіку

Характеристика	Пасивний моніторинг	Активний моніторинг
Вплив на мережу	Відсутній	Створює додатковий трафік
Джерело даних	Реальний трафік	Тестові запитання
Основні інструменти	Wireshark, Tcpdump, Scapy	Ping, Traceroute, Nmap
Призначення	Аналіз мережевої активності	Перевірка доступності та продуктивності

Для аналізу мережевого трафіку використовуються спеціалізовані програмні засоби – сніфери, які дозволяють перехоплювати та досліджувати мережеві пакети [11].



Рисунок 1.6 – Загальний принцип роботи мережевого сніффера

На рисунку 1.6 наведено загальну схему роботи мережевого сніффера, який здійснює захоплення, аналіз та фільтрацію мережевого трафіку. Під час аналізу бездротових мереж можуть досліджуватися як кадри стандарту IEEE 802.11, так і пакети мережевих протоколів вищих рівнів.

Процес аналізу трафіку дозволяє отримувати інформацію не лише про технічний стан мережі, але й про особливості взаємодії між окремими вузлами. Дослідження структури пакетів дає можливість визначати використовувані мережеві сервіси, виявляти аномальні з'єднання та оцінювати характер інформаційного обміну між пристроями.

Аналіз мережевого трафіку широко використовується як у процесі адміністрування мереж, так і під час проведення аудитів інформаційної безпеки. Завдяки цьому можливо виявляти потенційні загрози ще до того, як вони призведуть до порушення роботи мережі.

Основні функції сніфферів:

- захоплення пакетів;
- фільтрація трафіку;
- аналіз заголовків пакетів;
- виявлення аномалій;
- збереження дамів пакетів.

Для роботи сніфферів може використовуватись monitor mode – спеціальний режим роботи Wi-Fi адаптера, який дозволяє перехоплювати всі кадри IEEE 802.11 у зоні бездротової мережі незалежно від їх адресата.

Основні типи Wi-Fi кадрів:

- Management Frames;
- Control Frames;
- Data Frames.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		22

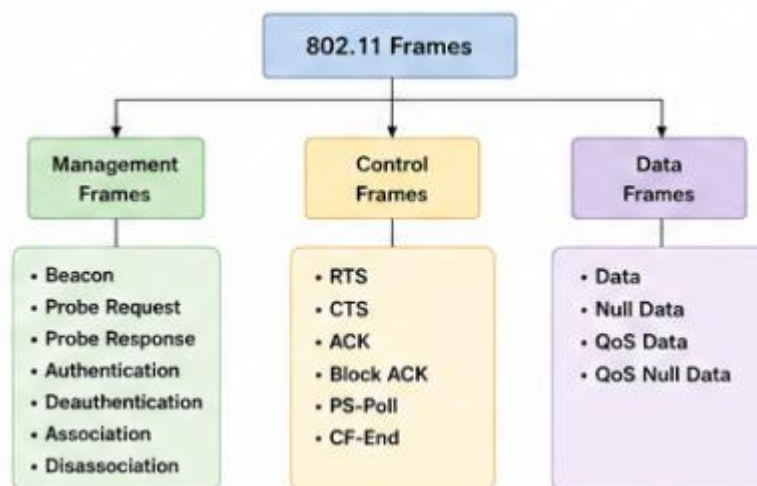


Рисунок 1.7 – Основні типи кадрів Wi-Fi мереж

На рисунку 1.7 наведено класифікацію кадрів Wi-Fi мереж відповідно до стандарту IEEE 802.11. Показано основні типи кадрів: керуючі, службові та інформаційні.

Кожен тип кадрів виконує власну функцію у процесі функціонування бездротової мережі. Керуючі кадри забезпечують встановлення та підтримання з'єднання між клієнтом і точкою доступу, службові кадри використовуються для координації передачі даних, а інформаційні кадри безпосередньо містять корисне навантаження користувачів. Аналіз співвідношення різних типів кадрів дозволяє оцінювати стан мережі та виявляти нетипову активність.

Дослідження структури кадрів є важливою складовою роботи мережевих аналізаторів. Саме на цьому рівні можуть бути виявлені окремі ознаки атак, пов'язаних із підробкою службових повідомлень або маніпулюванням процесами автентифікації.

Management Frames використовуються для керування мережею. До них належать:

- Beacon;
- Probe Request;
- Probe Response;
- Authentication;
- Deauthentication.

Data Frames містять корисні дані користувачів. Для аналізу мережевого трафіку використовуються Wireshark, tcpdump, Scapy, PyShark та Kismet.

Wireshark є одним із найпоширеніших аналізаторів мережевого трафіку та підтримує велику кількість протоколів.

Wireshark підтримує аналіз сотень мережевих протоколів та широко використовується у сфері мережевого моніторингу й кібербезпеки [4].

Tcpdump – консольний інструмент Linux для захоплення пакетів.

Scapy – Python-бібліотека для створення, аналізу та модифікації пакетів.

PyShark дозволяє працювати з пакетами засобами Python через Wireshark.

Використання програмних бібліотек для роботи з пакетами дозволяє автоматизувати значну частину процесу аналізу мережевого трафіку. На відміну від класичних аналізаторів, програмні рішення можуть виконувати обробку даних відповідно до заздалегідь визначених алгоритмів та формувати статистичні звіти без участі користувача.

Саме можливість автоматизації аналізу є однією з причин активного використання Python у сфері кібербезпеки. Поєднання бібліотек для роботи з пакетами та засобів обробки даних дозволяє створювати власні програмні інструменти моніторингу мережевого середовища.

Для збереження пакетів використовується формат PCAP.

Формат PCAP є одним із найбільш поширених форматів збереження дамів мережевого трафіку та підтримується більшістю сучасних аналізаторів пакетів [4].

Під час аналізу застосовуються фільтри:

- IP-адрес;
- MAC-адрес;
- TCP/UDP;
- DNS;
- HTTP;
- HTTPS.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				24
Змн.	Арк.	№ докум.	Підпис	Дата		

1.5 Використання Kali Linux у сфері кібербезпеки

Kali Linux є спеціалізованим дистрибутивом Linux, який містить значну кількість інструментів для аналізу захищеності інформаційних систем та мереж [2].

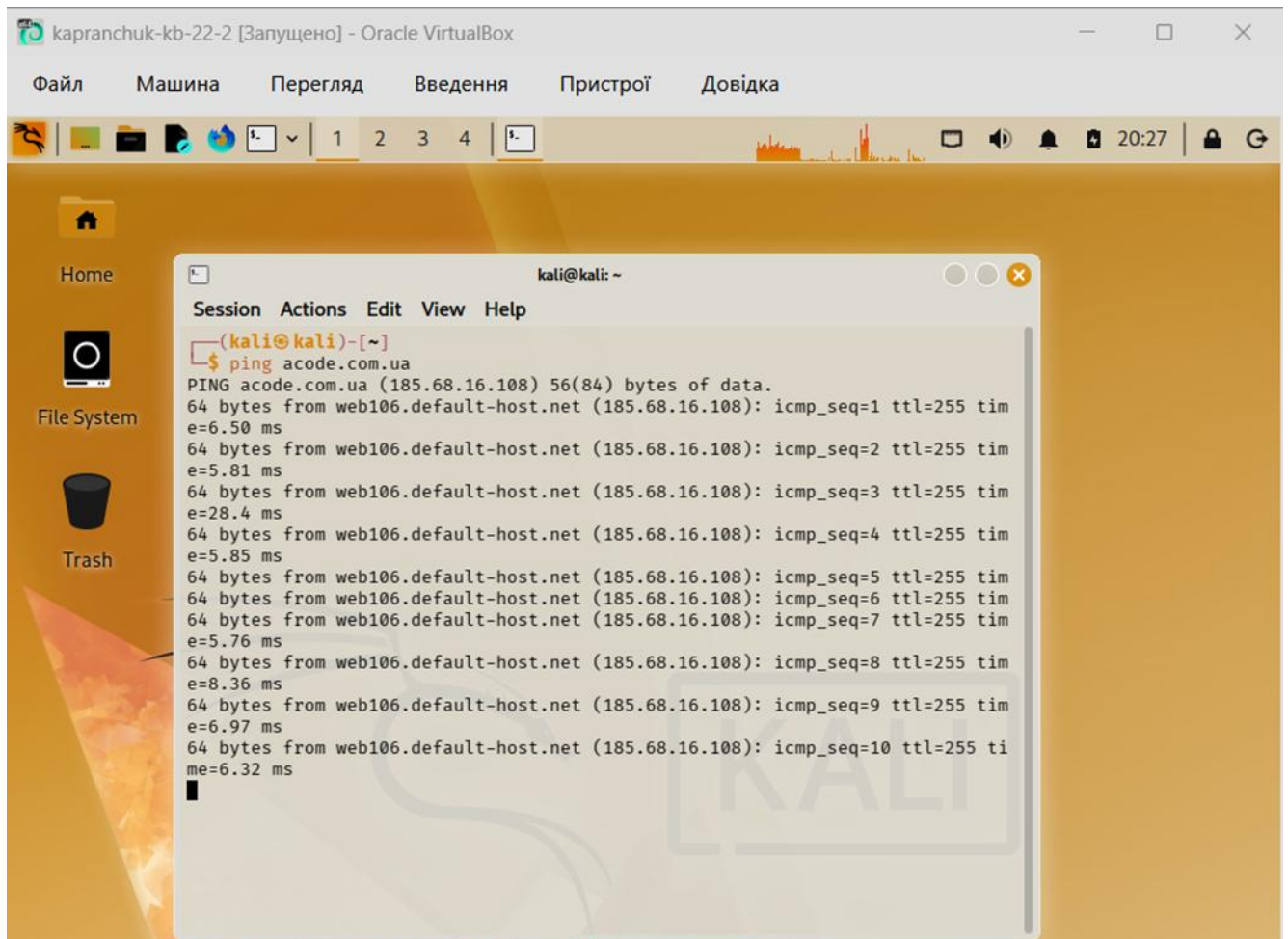


Рисунок 1.8 – Робоче середовище Kali Linux

На рисунку 1.8 представлено робоче середовище операційної системи Kali Linux, яка використовується для аналізу мережевого трафіку та тестування безпеки Wi-Fi мереж.

Kali Linux широко використовується фахівцями з кібербезпеки, системними адміністраторами та спеціалістами з мережевого аналізу.

Основні переваги Kali Linux:

- велика кількість вбудованих інструментів;
- підтримка Python;
- підтримка Wi-Fi аналізу;
- робота з мережевими пакетами;

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		25

- стабільність та гнучкість.

Основні інструменти Kali Linux наведені у таблиці 1.7:

Таблиця 1.7

Інструменти Kali Linux для аналізу мережі

Інструмент	Призначення
Wireshark	Аналіз мережевого трафіку
tcpdump	Захоплення пакетів
Aircrack-ng	Аналіз безпеки Wi-Fi
Kismet	Моніторинг бездротових мереж
Nmap	Сканування мереж
Hydra	Тестування паролів
Scapy	Аналіз пакетів у Python
Bettercap	MITM та аналіз мереж
Reaver	Атаки на WPS

Інструменти Kali Linux дозволяють виконувати аудит мережевої інфраструктури, аналіз мережевого трафіку та тестування механізмів захисту інформації [2].

Kali Linux побудована на базі Debian Linux та орієнтована на виконання завдань у сфері кібербезпеки, тестування захищеності інформаційних систем і мережевого аналізу. На відміну від універсальних операційних систем, дана платформа містить попередньо встановлений набір спеціалізованих інструментів, які дозволяють виконувати аудит мережі, дослідження мережевого трафіку та аналіз потенційних вразливостей без додаткового налаштування програмного середовища.

Однією з важливих переваг Kali Linux є підтримка великої кількості мережевих утиліт та інструментів аналізу пакетів. Операційна система забезпечує роботу з мережевими адаптерами, підтримує використання режиму моніторингу (Monitore Mode) та дозволяє виконувати захоплення мережевого трафіку в режимі реального часу. Це особливо важливо під час дослідження функціонування бездротових мереж та аналізу параметрів їх безпеки.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		26

У сфері мережевого аналізу Kali Linux часто використовується для дослідження структури пакетів, аналізу мережевих протоколів та контролю мережевої активності. За допомогою спеціалізованих засобів користувач може отримувати інформацію про джерела та отримувачів пакетів, визначати типи використовуваних протоколів, аналізувати маршрути передачі даних і виявляти аномальні мережеві події.

Важливою особливістю операційної системи є можливість використання засобів автоматизації. Підтримка мов програмування Python, Bash та інших інструментів розробки дозволяє створювати власні програмні рішення для моніторингу мережевої активності. Це значно спрощує виконання досліджень та забезпечує можливість реалізації спеціалізованих засобів аналізу мережевого трафіку.

Для дослідження мережевих протоколів у Kali Linux широко застосовуються інструменти Wireshark і tcpdump. Дані програмні засоби дозволяють перехоплювати мережеві пакети, аналізувати їх структуру та виконувати фільтрацію трафіку за різними параметрами. Отримані результати можуть використовуватися для діагностики мережевих проблем, оцінювання продуктивності мережі та виявлення потенційних загроз безпеці.

Крім аналізу трафіку, Kali Linux забезпечує можливість дослідження мережевої інфраструктури. Для цього застосовуються засоби мережевого сканування, зокрема Nmap, який дозволяє визначати активні вузли мережі, відкриті порти та доступні мережеві служби. Отримана інформація використовується під час оцінювання рівня захищеності мережевого середовища та пошуку потенційних вразливостей.

Для аналізу мережевої інфраструктури використовується утиліта Nmap, яка дозволяє виконувати сканування мереж та виявлення активних вузлів [5].

Саме поєднання інструментів моніторингу, аналізу пакетів, мережевого сканування та автоматизації робить Kali Linux однією з найбільш поширених платформ для проведення досліджень у сфері кібербезпеки. Використання даної операційної системи дозволяє створити зручне середовище для розроблення та тестування власних програм засобів аналізу мережевого трафіку.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		27

Kali Linux використовується для:

- аудиту мереж;
- тестування Wi-Fi безпеки;
- аналізу трафіку;
- пошуку вразливостей;
- навчання у сфері кібербезпеки.

У Kali Linux підтримуються:

- monitore mode;
- packet sniffing;
- робота з PCAP;
- атоматизація тестування;
- Python-розробка.

У межах даної кваліфікаційної роботи Kali Linux використовується як середовище для розробки та тестування програмного сніффера.

Python є однією з основних мов програмування у сфері кібербезпеки завдяки:

- простому синтаксису;
- великій кількості бібліотек;
- підтримці мережевого програмування.

Використання Python у сфері мережевого аналізу обумовлене наявністю великої кількості бібліотек для роботи з мережевими протоколами та автоматизації обробки даних. Завдяки простому синтаксису мова дозволяє швидко реалізовувати програмні рішення різного рівня складності та виконувати обробку великих обсягів мережевої інформації.

Під час розроблення програмних сніфферів Python часто використовується разом із бібліотекою Scapy, яка забезпечує доступ до мережесих пакетів на різних рівнях моделі TCP/IP. Бібліотека дозволяє не лише аналізувати вже отримані пакети, але й формувати власні мережеві повідомлення, що робить її універсальним інструментом для дослідження мережесих протоколів.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				28
Змн.	Арк.	№ докум.	Підпис	Дата		

Python широко застосовується у сфері кібербезпеки завдяки можливостям автоматизації аналізу мережових даних та наявності спеціалізованих бібліотек для роботи з мережевими протоколами [28].

Важливою перевагою Scapy є можливість інтеграції з іншими програмними компонентами Python. Це дозволяє реалізовувати складні системи моніторингу, автоматизувати процес збору статистики та створювати власні механізми виявлення підозрілої мережової активності.

Для реалізації сніффера використовується бібліотека Scapy, яка дозволяє:

- захоплювати пакети;
- аналізувати заголовки;
- працювати з PCAP-файлами;
- створювати власні пакети.



Рисунок 1.9 – Взаємодія бібліотеки Scapy з мережовим інтерфейсом

На рисунку 1.9 показано взаємодію бібліотеки Scapy із мережовим інтерфейсом під час захоплення та аналізу пакетів у середовищі Python.

Бібліотека Scapy забезпечує можливість створення, перехоплення, модифікації та аналізу мережових пакетів різних типів, що робить її ефективним інструментом для розроблення мережових аналізаторів [3].

1.6 Обґрунтування необхідності розробки програмного сніффера для аналізу Wi-Fi мереж

Проведений аналіз принципів функціонування Wi-Fi мереж, показників якості бездротового зв'язку, основних загроз безпеці та методів моніторингу мережевого трафіку показав необхідність використання спеціалізованих програмних засобів для контролю стану мережі. Існуючі інструменти аналізу трафіку забезпечують широкі можливості моніторингу, однак часто потребують додаткового налаштування або орієнтовані на виконання окремих завдань. У зв'язку з цим доцільною є розробка програмного сніффера, який поєднуватиме функції захоплення мережевого трафіку, аналізу показників якості Wi-Fi мереж та виявлення ознак потенційних загроз безпеці.

Бездротові мережі Wi-Fi сьогодні є одним із найпоширеніших способів організації доступу до мережі Інтернет та локальних мережевих ресурсів. Їх використовують у навчальних закладах, державних установах, підприємствах та домашніх мережах. Широке впровадження бездротових технологій обумовлене зручністю підключення, мобільністю користувачів та відсутністю необхідності прокладання кабельної інфраструктури. Разом із цим збільшується кількість пристроїв, що одночасно працюють у мережі, а також зростає кількість потенційних загроз інформаційній безпеці.

Для забезпечення стабільної роботи мережі необхідно контролювати її стан та своєчасно виявляти проблеми, які можуть негативно впливати на якість зв'язку. До таких проблем належать перевантаження мережі, висока інтенсивність трафіку, втрати пакетів, збільшення затримок передачі даних та поява невідомих мережевих вузлів. Виявлення подібних факторів дозволяє оперативно реагувати на зміни стану мережі та підтримувати необхідний рівень її працездатності.

Окрему увагу необхідно приділяти питанням інформаційної безпеки. Навіть за використання сучасних механізмів шифрування існує ризик появи підозрілої мережевої активності, несанкціонованих підключень або використання небезпечних мережевих сервісів. Для виявлення подібних ситуацій використовуються засоби

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				30
Змн.	Арк.	№ докум.	Підпис	Дата		

аналізу мережевого трафіку, які дозволяють отримувати інформацію про роботу мережі та поведінку її користувачів.

Одним із найбільш ефективних інструментів дослідження мережевої активності є програмний сніффер. Такий програмний засіб забезпечує перехоплення мережевих пакетів, їх аналіз та подальшу обробку. Отримані дані можуть використовуватися для оцінювання якості функціонування мережі, визначення типів мережевого трафіку, виявлення активних вузлів та аналізу потенційних загроз безпеці.

Як середовище розробки було обрано Kali Linux, яка містить необхідний набір інструментів для мережевого аналізу та тестування безпеки [2]. Вибір зазначених засобів обумовлений їх широким використанням у сфері мережевого аналізу, наявністю засобів захоплення пакетів та можливістю реалізації власних алгоритмів обробки мережевих даних.

Для реалізації програмного сніффера використовується бібліотека Scapy, що забезпечує роботу з мережевими пакетами на різних рівнях моделі TCP/IP[3].

Особливістю виконання проєкту є використання операційної системи Kali Linux у віртуальному середовищі. У зв'язку з відсутністю зовнішнього Wi-Fi адаптера, що підтримує режим моніторингу, дослідження виконуватиметься шляхом аналізу мережевого трафіку, доступного для віртуальної машини. Такий підхід дозволяє реалізувати основні функції сніффера та провести дослідження мережевої активності без використання додаткового апаратного забезпечення.

Основною метою проєктування є створення програмного засобу, який забезпечить автоматизований збір та аналіз мережевого трафіку для оцінювання якості функціонування мережі та виявлення потенційних загроз безпеці.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- визначити вимоги до програмного сніффера;
- розробити структуру програмного засобу;
- спроектувати алгоритм захоплення мережевих пакетів;
- визначити методику аналізу мережевого трафіку;
- розробити підхід до оцінювання показників якості мережі;
- розробити механізм оцінювання безпеки мережевого середовища;

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		31

- забезпечити можливість збереження результатів аналізу для подальшого використання.

Результатом виконання етапу проєктування повинна стати модель програмної системи, яка визначає склад основних модулів, порядок їх взаємодії та алгоритми обробки мережових даних. Розроблені рішення будуть використані під час програмної реалізації та тестування системи у наступному розділі роботи.

Висновки до розділу 1

У першому розділі було розглянуто принцип роботи Wi-Fi мереж, показники якості бездротового зв'язку та основні загрози безпеці Wi-Fi мереж. Проаналізовано типи атак, методи моніторингу мережевого трафіку та інструменти аналізу пакетів.

Також було досліджено можливості Kali Linux і бібліотеки Scapy для аналізу та моніторинг Wi-Fi мереж. Отримані результати є основою для розробки програмного сніффера.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				32
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 ПРОЄКТУВАННЯ ПРОГРАМНОГО СНІФФЕРА ДЛЯ АНАЛІЗУ WI-FI МЕРЕЖ

2.1 Аналіз вимог до програмного засобу

Після проведення теоретичного аналізу предметної області виникає необхідність визначення вимог до майбутнього програмного засобу. Формування вимог є важливим етапом розробки, оскільки саме на їх основі визначаються функціональні можливості системи, принцип її роботи та особливості взаємодії окремих програмних модулів. Вимоги дозволяють встановити перелік завдань, які повинен виконувати програмний сніффер, а також врахувати технічні обмеження та умови його використання. Сформуємо основні вимоги до програмного засобу, які визначають функціональні можливості системи, її поведінку під час роботи та обмеження, що необхідно враховувати під час розробки.

Основним призначенням програмного сніффера є отримання інформації про мережевих трафік та її подальший аналіз. Для реалізації цього завдання система повинна забезпечувати безперервне захоплення мережевих пакетів, їх обробку та відображення отриманих результатів.

До функціональних вимог належать можливості, які повинна виконувати програма під час роботи. Насамперед система повинна здійснювати захоплення пакетів із мережевого інтерфейсу. Після отримання пакета необхідно виконати його аналіз та визначити основні характеристики. До таких характеристик належать IP-адреса джерела, IP-адреса призначення, номер порту, тип мережевого протоколу та службова інформація заголовків.

Однією з ключових вимог є можливість класифікації трафіку. Програмний засіб повинен визначати типи протоколів, які використовуються в мережі. Це дозволяє оцінити структуру трафіку та виявити особливості функціонування мережі.

Для проведення подальшого аналізу система повинна накопичувати статистичні дані про мережеву активність. Отримана інформація використовується для оцінювання навантаження на мережу, інтенсивності обміну даними та загального стану мережевого середовища.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		33

Окремою вимогою є реалізація засобів аналізу безпеки. Програмний засіб повинен забезпечувати виявлення активних вузлів локальної мережі, визначення відкритих портів та мережевих сервісів. Отримані результати дозволяють оцінити рівень захищеності мережі та виявити потенційні ризики.

Для зручності подальшої роботи результати аналізу повинні зберігатися у структурованому вигляді. У межах даного проєкту передбачається використання формату CSV, який забезпечує сумісність із поширеним засобами обробки даних.

Крім функціональних вимог, важливу роль відіграють нефункціональні характеристики системи. До них належать швидкодія, стабільність роботи, зручність використання та можливість подальшого розширення функціоналу.

Програмний засіб повинен працювати в середовищі операційної системи Kali Linux та не потребувати використання додаткового спеціалізованого обладнання. Такий підхід дозволяє спростити процес розробки та забезпечити можливість відтворення результатів дослідження на інших комп'ютерних системах.

Основні вимоги до системи наведені в таблиці 2.1.

Таблиця 2.1

Вимоги до програмного сніффера

Вимога	Характеристика
Захоплення пакетів	Отримання мережевого трафіку в реальному часі
Аналіз пакетів	Обробка службової інформації пакетів
Визначення протоколів	Аналіз TCP, UDP, DNS та інших протоколів
Аналіз безпеки	Виявлення вузлів і сервісів мережі
Формування звітів	Збереження результатів аналізу
Простота використання	Робота через консольний інтерфейс
Сумісність	Підтримка можливості роботи з великою кількістю обладнання
Розширюваність	Можливість додавання нових функцій

Сформовані вимоги визначають функціональні межі проєкту та є основою для розробки архітектури програмного засобу.

2.2 Проєктування архітектури програмного сніффера

Ефективність роботи програмного засобу значною мірою залежить від правильно обраної архітектури. Для забезпечення простоти реалізації та подальшої модернізації було обрано модульний принцип побудови системи.

Модульна архітектура передбачає розподіл функцій між окремими компонентами, кожен із яких відповідає за виконання конкретного завдання. Такий підхід спрощує процес тестування програмного забезпечення, підвищує його надійність та дозволяє розширювати функціональні можливості без суттєвої зміни існуючої структури.

У межах проєкту виділено чотири основні модулі: модуль захоплення пакетів, модуль аналізу трафіку, модуль аналізу безпеки та модуль формування звітів.

Модуль захоплення пакетів виконує взаємодію з мережевим інтерфейсом операційної системи. Основним завданням цього компонента є отримання пакетів, які проходять через мережевий інтерфейс, та передача їх до наступних етапів обробки.

Модуль аналізу трафіку здійснює обробку отриманих пакетів. На цьому етапі визначаються мережеві адреси, транспортні протоколи, порти та інші параметри, які характеризують мережеве з'єднання.

Модуль аналізу безпеки призначений для оцінювання потенційних ризиків. Він виконує аналіз активних вузлів мережі, відкритих портів та сервісів, які можуть становити загрозу безпеці інформаційної системи.

Модуль формування звітів забезпечує збереження результатів аналізу та підготовку підсумкових даних для подальшого використання.

Загальна структура програмного сніффера наведена на рисунку 2.1.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		35

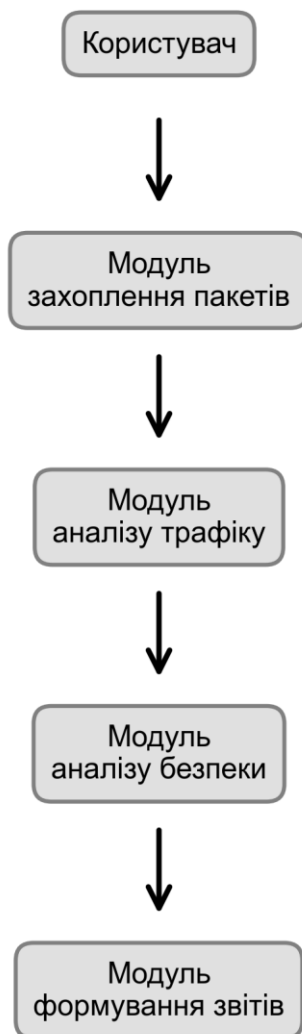


Рисунок 2.1 – Архітектура програмного сніффера

На рисунку 2.1 наведено структуру розроблюваного програмного сніффера. Система складається з модулів захоплення пакетів, аналізу трафіку, аналізу безпеки та формування звітів, які забезпечують послідовну обробку мережових даних.

Така архітектура дозволяє організувати послідовну обробку мережових даних та забезпечити виконання всіх функцій, визначених під час аналізу вимог до системи.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				36
Змн.	Арк.	№ докум.	Підпис	Дата		

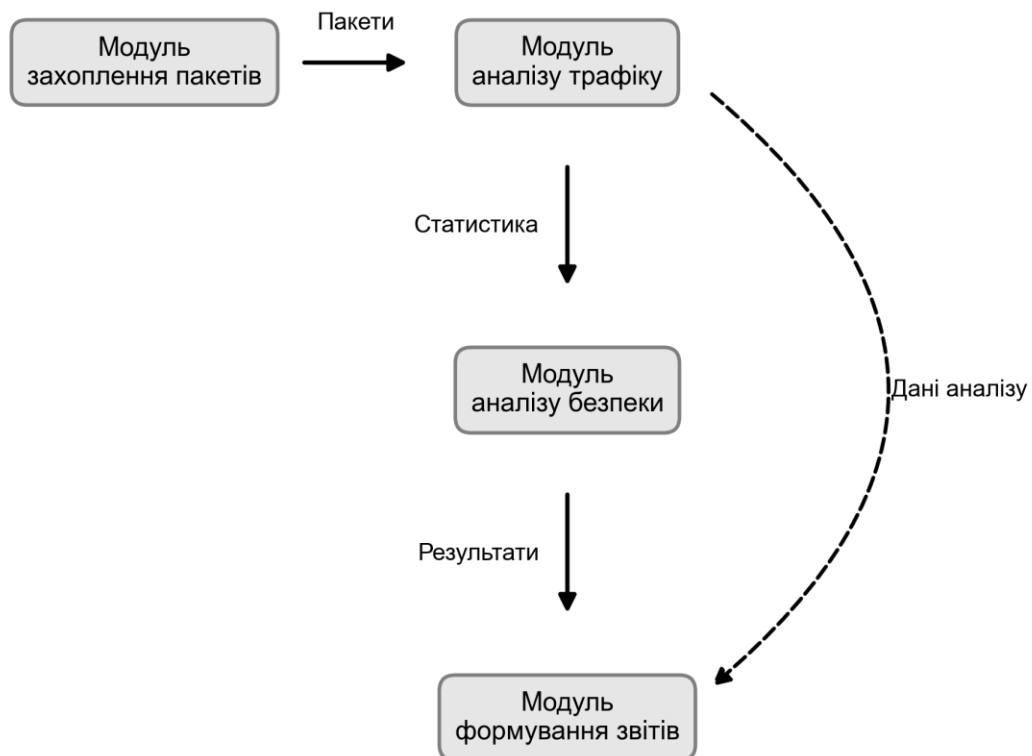


Рисунок 2.2 – Схема взаємодії модулів системи

На рисунку 2.2 показано обмін даними між основними модулями програмного сніффера. Модуль захоплення пакетів передає дані до модуля аналізу трафіку, результати якого використовуються модулем аналізу безпеки та модулем формування звітів.

Основні модулі архітектури наведені у таблиці 2.2.

Таблиця 2.2

Основні модулі системи

Модуль	Призначення
Packet Capture	Захоплення мережеских пакетів
Traffic Analyzer	Аналіз мережевого трафіку
Security Analyzer	Аналіз безпеки мережі
Report Generator	Формування та збереження звітів

Використання модульної архітектури забезпечує гнучкість системи та сво-
рює основу для подальшого розвитку програмного засобу.

2.3 Методика збору мережевих даних

Для проведення аналізу якості та безпеки мережі необхідно забезпечити отримання актуальної інформації про мережевий трафік. Саме тому одним із ключових етапів проектування програмного сніффера є розробка методики збору мережевих даних.

Під збором мережевих даних розуміється процес отримання інформації про пакети, які передаються між мережевими пристроями. Кожен пакет містить службові та користувацькі дані, що дозволяють визначити характеристики мережевого з'єднання та оцінити стан мережі.

У межах даної роботи джерело інформації виступає мережевий інтерфейс операційної системи Kali Linux. Отримання пакетів здійснюється за допомогою бібліотеки Scapy, яка надає засоби низькорівневого доступу до мережевого трафіку.

Основною перевагою такого підходу є можливість отримання інформації в режимі реального часу. Це дозволяє виконувати аналіз поточної мережевої активності без необхідності використання попередньо записаних файлів трафіку.

Процес збору даних складається з декількох послідовних етапів.

На першому етапі виконується запуск програмного засобу та ініціалізація мережевого інтерфейсу. Після запуску програма переходить у режим очікування пакетів.

Другий етап передбачає захоплення пакетів, які проходять через мережевий інтерфейс. Кожен отриманий пакет передається до модуля аналізу для подальшої обробки.

Третій етап полягає у виділенні необхідних параметрів із заголовків пакетів. Для кожного пакета визначаються:

- IP-адреса джерела;
- IP-адреса призначення;
- тип протоколу;

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		38

- номер порту;
- довжина пакета;
- час отримання пакета.

На четвертому етапі виконується класифікація пакетів за типами мережових протоколів. Це дозволяє визначити структуру мережевого трафіку та оцінити характер використання мережових ресурсів.

Останнім етапом є накопичення статистичної інформації та передача результатів до модуля аналізу якості та безпеки.

Загальний алгоритм збору мережових даних наведено на рисунку 2.2.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		39

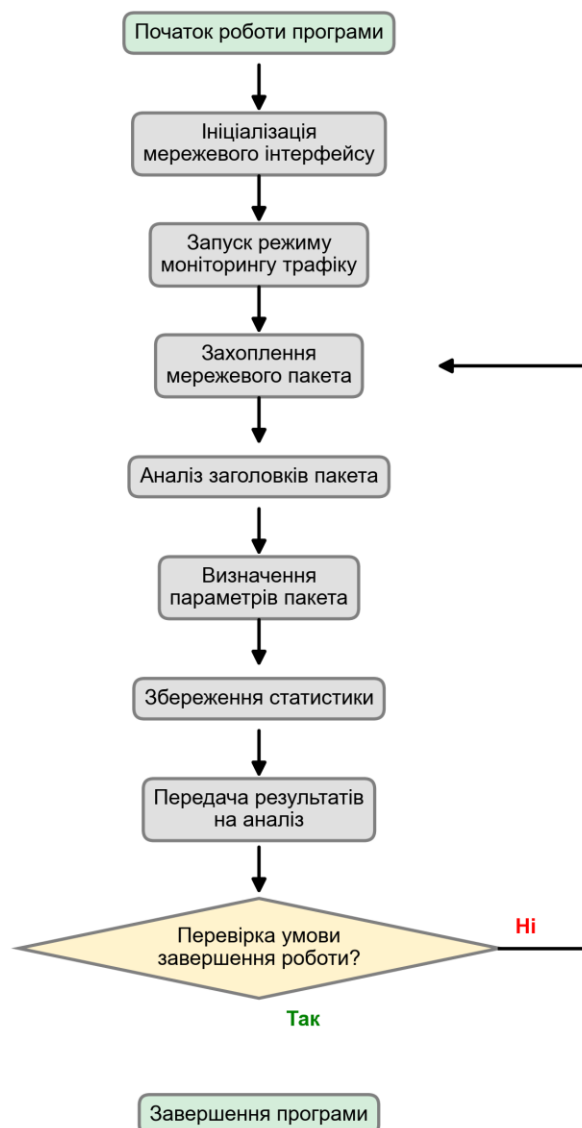


Рисунок 2.3 – Алгоритм збору мережевих даних

На рисунку 2.3 представлено алгоритм роботи програмного засобу під час збору мережевих даних. Після ініціалізації мережевого інтерфейсу програмний сніффер переходить у режим безпереривного моніторингу трафіку. Для кожного перехопленого пакета виконується аналіз заголовків, визначення основних параметрів та збереження отриманої інформації для подальшої обробки. Зазначені операції виконуються циклічно до моменту завершення роботи програмного засобу користувачем.

Під час виконання проєкту було прийнято рішення використовувати лише ті параметри пакетів, які необхідні для оцінювання якості та безпеки мережі. Це дозволяє зменшити навантаження на систему та спростити процес обробки даних.

Для збереження результатів аналізу передбачається використання структурованого формату CSV. Такий підхід забезпечує можливість подальшої обробки даних у табличних процесорах або спеціалізованих системах аналізу.

2.4 Методика аналізу якості Wi-Fi мереж

Одним із головних завдань програмного сніффера є оцінювання якості функціонування мережі. Для цього необхідно визначити набір показників, які найбільш повно характеризують стан мережевого середовища.

Якість мережі визначається сукупністю характеристик, що впливають на швидкість, стабільність та надійність передачі даних між вузлами мережі.

У межах роботи для оцінювання якості мережі було обрано такі показники:

- затримка передачі даних;
- пропускна здатність;
- інтенсивність трафіку;
- втрати пакетів;
- активність мережевих вузлів.

Першим показником є затримка передачі даних (Delay). Вона характеризує час, необхідний для доставки інформації від відправника до отримувача. Високі значення затримки можуть негативно впливати на роботу мережевих сервісів, особливо відеоконференцій, онлайн-ігор та систем дистанційного керування.

Другим показником є пропускна здатність мережі (Throughput). Ця характеристика визначає обсяг інформації, який може бути переданий через мережу за одиницю часу. Чим вищим є значення пропускної здатності, тим ефективніше використовуються мережеві ресурси.

Важливим параметром є інтенсивність мережевого трафіку. Даний показник характеризує кількість пакетів, які передаються через мережу протягом певного

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		41

часу. Зростання інтенсивного трафіку може свідчити про підвищене навантаження на мережу.

Для оцінювання надійності передачі даних використовується показник втрати пакетів (Packet Loss). Втрата пакетів може виникати через перевантаження мережі, несправності обладнання або помилки передавання даних. Значні втрати пакетів зазвичай призводять до погіршення якості мережевого з'єднання.

Окремо аналізується активність мережевих вузлів. Цей показник дозволяє визначити кількість пристроїв, які беруть участь у мережевому обміні, та оцінити рівень використання мережевої інфраструктури.

Основні метрики оцінювання якості наведено у таблиці 2.3.

Таблиця 2.3

Метрики оцінювання якості мережі

Метрика	Призначення
Delay	Визначення затримки передачі даних
Throughput	Оцінювання пропускної здатності
Packet Loss	Визначення втрат пакетів
Packet Rate	Оцінювання інтенсивності трафіку
Network Activity	Аналіз активності вузлів

Для інтерпретації отриманих результатів необхідно визначити критерії оцінювання основних мережевих показників. Значення затримки передачі даних, втрат пакетів та пропускної здатності дозволяють оцінити стабільність роботи мережі та рівень її продуктивності. Зменшення пропускної здатності, збільшення затримок або поява втрат пакетів можуть свідчити про перевантаження мережі, наявність перешкод або інші проблеми функціонування бездротового середовища.

Для подальшого аналізу отримані значення будуть порівнюватися з орієнтовними критеріями якості мережевого з'єднання, наведеними в таблиці 2.4.

Критерії оцінювання якості мережевого з'єднання

Метрика	Добре	Задовільно	Незадовільно
Delay	До 50 мс	50-100 мс	Понад 100 мс
Throughput	До 1%	1 – 3%	Понад 3%
Packet Loss	Понад 80% від максимальної швидкості	50 – 80%	Менше 50%
Packet Rate	Стабільний рівень	Помірні коливання	Різкі стрибки навантаження
Network Activity	Типова активність	Підвищена активність	Аномально висока активність

Після отримання статистичних даних виконується їх узагальнення та формування підсумкової оцінки якості мережі. Запропонована методика дозволяє отримати достатню кількість інформації для аналізу поточного стану мережі та виявлення можливих проблем її функціонування.

2.5 Методика оцінювання безпеки Wi-Fi мереж

Окрім оцінювання якості роботи мережі, важливим завданням є аналіз її безпеки. Безпека мережі визначається рівнем захищеності інформаційних ресурсів від несанкціонованого доступу та інших загроз.

Для оцінювання безпеки необхідно отримати інформацію про структуру мережі, активні вузли та мережеві сервіси.

Першим етапом аналізу є виявлення активних пристроїв локальної мережі. Під час сканування визначаються вузли, які відповідають на мережеві запити та беруть участь в обміні даними.

Отримання списку активних пристроїв дозволяє сформувавши уявлення про поточний стан мережі та виявити невідомі вузли, які можуть становити потенційну загрозу.

Наступним етапом є аналіз відкритих портів. Відкритий порт свідчить про наявність мережевого сервісу, який може приймати вхідні підключення.

Не всі відкриті порти становлять небезпеку, проте велика їх кількість збільшує потенційну поверхню атаки та може підвищувати ризик компрометації адміністрування.

Після визначення відкритих портів виконується аналіз мережевих сервісів. Особлива увага приділяється сервісами віддаленого доступу та службам адміністрування.

У межах роботи аналізуються такі поширені сервіси:

- HTTP;
- HTTPS;
- DNS;
- SSH;
- FTP.

Наступним етапом є аналіз мережевого трафіку з метою виявлення підозрілої активності.

До ознак потенційно небезпечної активності можуть належати:

- велика кількість мережевих підключень;
- часті мережеві запити;
- використання нетипових мережевих вузлів;
- поява невідомих мережевих вузлів;
- аномальне збільшення трафіку.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		44



Рисунок 2.4 – Алгоритм оцінювання безпеки мережі

На рисунку 2.4 наведено послідовність виконання аналізу безпеки мережі. Алгоритм передбачає виявлення активних вузлів, перевірку відкритих портів, аналіз мережевих сервісів, пошук підозрілої активності та формування підсумкового звіту.

Основні ознаки можливих загроз наведені у таблиці 2.5.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		45

Ознаки потенційних загроз

Ознака	Можливий ризик
Невідомий пристрій у мережі	Несанкціонований доступ
Відкритий порт	Потенційна вразливість
Підвищена мережева активність	Перевантаження мережі
Велика кількість підключень	Спроба мережевої атаки
Нетиповий протокол	Підозріла активність

Після завершення аналізу формується звіт про стан безпеки мережі. У звіті відображаються знайдені вузли, відкриті порти, виявлені сервіси та зафіксовані підозрілі події.

Розроблена методика дозволяє виконати безове оцінювання рівня безпеки мережі та виявити потенційні ризики без використання спеціалізованого дорогого обладнання.

Висновки до розділу 2

У другому розділі виконано проєктування програмного сніффера для дослідження якості та безпеки Wi-Fi мереж. Визначено функціональні вимоги до програмного засобу, розроблено його архітектуру та описано взаємодію основних модулів системи.

Запропоновано методику збору мережевих даних, яка передбачає захоплення пакетів, виділення необхідних параметрів та накопичення статистичної інформації. Розроблено підхід до оцінювання якості мережі на основі аналізу затримок, інтенсивності трафіку, пропускної здатності та втрат пакетів.

Також сформовано методику аналізу безпеки мережі, що включає виявлення активних вузлів, аналіз відкритих портів, визначення мережевих сервісів та пошук ознак потенційних загроз. Отримані результати є основою для практичної реалізації програмного сніффера та проведення експериментальних досліджень у наступному розділі роботи.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		46

РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ СНІФФЕРА ДЛЯ АНАЛІЗУ WI-FI МЕРЕЖ

3.1 Реалізація структури програмного засобу

У межах кваліфікаційної роботи було розроблено програмний сніффер для дослідження якості безпеки мережевого трафіку. Реалізація програмного засобу у середовищі Kali Linux із використанням мови програмування Python. Вибір Python обумовлений його простотою, широкими можливостями для роботи з мережевими протоколами та наявністю спеціалізованих бібліотек для аналізу трафіку.

Для створення програмного засобу було використано бібліотеку Scapy, яка надає функціонал перехоплення, формування, модифікації та аналізу мережових пакетів. Дана бібліотека широко використовується фахівцями з кібербезпеки для аналізу мережевої активності та дослідження мережових протоколів.

Розробка виконувалася у віртуальному середовищі Kali Linux, встановленому за допомогою VirtualBox. Використання віртуальної машини дозволяє створити ізольоване середовище для проведення досліджень без впливу на основну операційну систему.

Перед початком розробки було створено окрему робочу директорію проєкту:

```
mkdir ~/wifi_sniffer_diploma
```

Після створення директорії було виконано перехід до каталогу проєкту:

```
cd ~/wifi_sniffer_diploma
```

Для створення основного програмного файлу використовувався текстовий редактор nano:

```
nano sniffer.py
```

На етапі проєктування було визначено модульну структуру програмного засобу. Такий підхід спрощує подальше розширення функціоналу та забезпечує зручність супроводу програмного коду.

Структура програмного засобу складається з чотирьох основних модулів:

- модуль захоплення пакетів;
- модуль аналізу трафіку;

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		47

- модуль аналізу безпеки;
- модуль збереження результатів.

Модуль захоплення пакетів відповідає за отримання мережевих пакетів із мережевого інтерфейсу та передачу їх до інших компонентів системи.

Модуль аналізу трафіку забезпечує обробку отриманих пакетів, визначення використовуваних протоколів, аналіз IP-адрес та оцінювання мережевої якості.

Модуль аналізу безпеки призначений для виявлення формування журналу роботи програми та запис результатів аналізу у файл формату CSV.

Для реалізації зазначеної структури було створено відповідні класи:

```

GNU nano 9.0 sniffer.py *
from scapy.all import *

class PacketCapture:
    def capture_packets(self):
        print("Packet capture module")

class PacketAnalysis:
    def analyze_packets(self):
        print("Packet analysis module")

class SecurityAnalysis:
    def detect_threats(self):
        print("Security analysis module")

class SaveResults:
    def save_to_csv(self):
        print("Saving results")

def main():
    capture = PacketCapture()
    analysis = PacketAnalysis()
    security = SecurityAnalysis()
    save = SaveResults()

    capture.capture_packets()
    analysis.analyze_packets()
    security.detect_threats()
    save.save_to_csv()

if __name__ == "__main__":
    main()

```

Рисунок 3.1 – Програмна реалізація основних модулів сніффера

У наведеному фрагменті коду реалізовано базову архітектуру програмного засобу. Кожен клас відповідає за виконання окремого функціонального завдання.

Такий підхід дозволяє дотримуватись принципів модульного програмування та спрощує подальше розширення системи.

Після створення структури програми було виконано запуск програмного засобу.

Для запуску програми використовувалася команда:

```

kali@kali: ~/wifi_sniffer_diploma
Session Actions Edit View Help
(kali@kali)-[~]
$ cd ~/wifi_sniffer_diploma
(kali@kali)-[~/wifi_sniffer_diploma]
$ sudo python3 sniffer.py
[sudo] password for kali:
Packet capture module
Packet analysis module
Security analysis module
Saving results
  
```

Рисунок 3.2 – Запуск програмного сніффера у середовищі Kali Linux

На рисунку 3.2 показано запуск програмного сніффера після переходу до робочої директорії проєкту. У результаті запуску відбулося послідовне виконання всіх модулів програми та перевірка працездатності створеної архітектури.

Створена структура програмного засобу є основою для подальшої реалізації механізмів захоплення пакетів, аналізу мережевого трафіку та виявлення потенційних загроз безпеці.

Таблиця 3.1

Основні модулі програмного сніффера

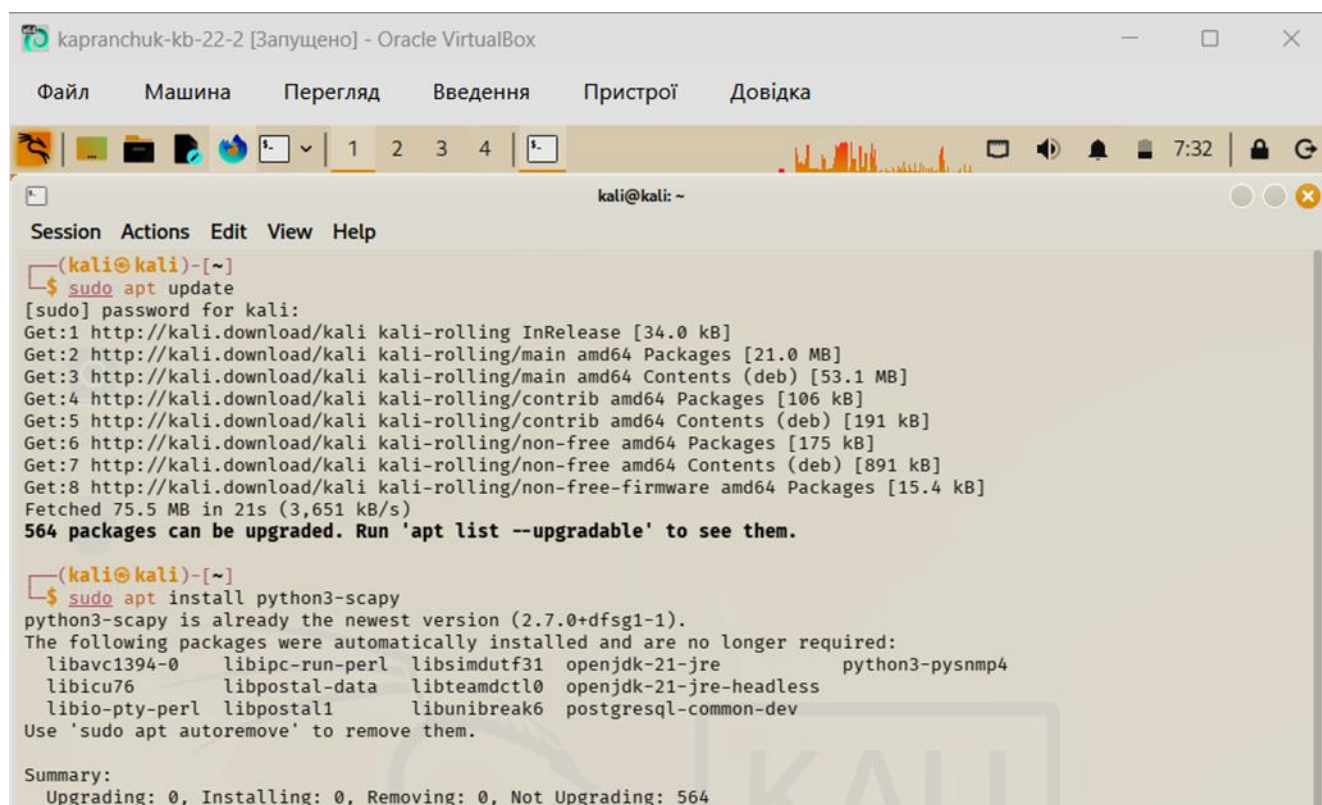
Модуль	Призначення
PacketCapture	Захоплення мережевих пакетів
PacketAnalysis	Аналіз мережевого трафіку
SecurityAnalysis	Аналізи безпеки мережі
SaveResult	Збереження результатів аналізу

3.2 Реалізація захоплення та аналізу пакетів

Основним призначенням програмного сніффера є отримання та обробка мережевого трафіку. Для реалізації даного функціоналу було використано бібліотеку Scapy, яка надає засоби для захоплення мережевих пакетів та їх подальшого аналізу.

У межах даної роботи захоплення трафіку виконується через мережевий інтерфейс операційної системи Kali Linux. Оскільки проєкт реалізується у віртуальному середовищі VirtualBox без використання зовнішнього Wi-Fi адаптера, аналізуються пакети мережевого інтерфейсу віртуальної машини.

Перед початком було встановлено бібліотеку Scapy:



```
kapranchuk-kb-22-2 [Запущено] - Oracle VirtualBox
Файл  Машина  Перегляд  Введення  Пристрої  Довідка
1 2 3 4
kali@kali: ~
Session Actions Edit View Help
(kali@kali)-[~]
$ sudo apt update
[sudo] password for kali:
Get:1 http://kali.download/kali kali-rolling InRelease [34.0 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [21.0 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [53.1 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [106 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [191 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [175 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [891 kB]
Get:8 http://kali.download/kali kali-rolling/non-free-firmware amd64 Packages [15.4 kB]
Fetched 75.5 MB in 21s (3,651 kB/s)
564 packages can be upgraded. Run 'apt list --upgradable' to see them.

(kali@kali)-[~]
$ sudo apt install python3-scapy
python3-scapy is already the newest version (2.7.0+dfsg1-1).
The following packages were automatically installed and are no longer required:
 libavc1394-0 libipc-run-perl libsindut31 openjdk-21-jre python3-pysnmp4
 libicu76 libpostal-data libteamctl0 openjdk-21-jre-headless
 libio-pty-perl libpostal1 libunibreak6 postgresql-common-dev
Use 'sudo apt autoremove' to remove them.

Summary:
Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 564
```

Рисунок 3.3 – Встановлення та перевірка бібліотеки Scapy у середовищі Kali Linux

На рисунку 3.3 показано процес підготовки програмного середовища для розробки сніффера мережевого трафіку. За допомогою команди **sudo apt update** було виконано оновлення списку операційної системи Kali Linux. Після цього командою **sudo apt install python3-scapy** перевірено наявність бібліотеки Scapy, яка використовується для захоплення, аналізу та обробки мережевих пакетів.

Результат виконання команди свідчить, що пакет **python3-scapy** вже встановлений у системі та готовий до використання. Це підтверджує можливість реалізації програмного сніффера на мові Python без встановлення додаткових компонентів.

Для перевірки встановлення виконувалася команда (результат виконання наведено на рис. 3.4):

*python3 -c "from scapy.all import *; print('Scapy installed')"* (1)

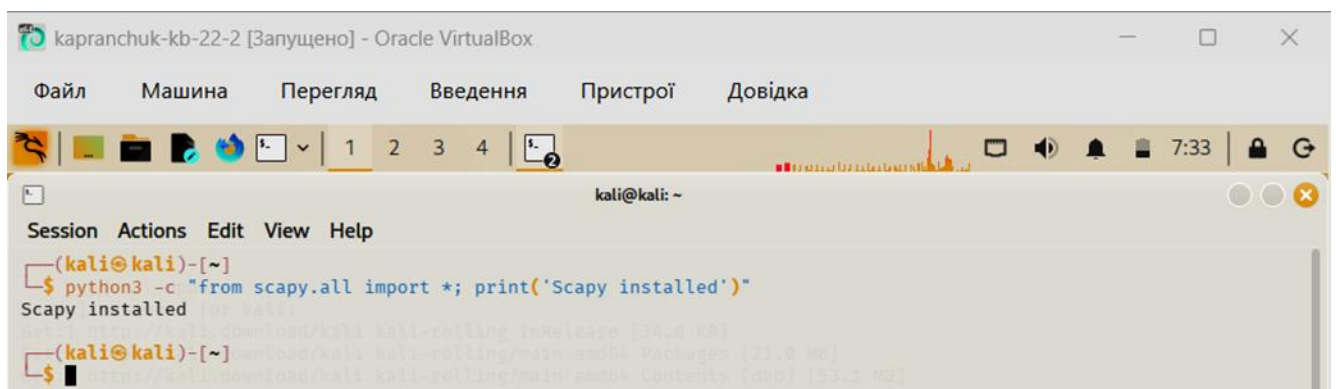


Рисунок 3.4 – Перевірка працездатності бібліотеки Scapy у середовищі Kali Linux

На рисунку 3.4 показано перевірку коректності встановлення та роботи бібліотеки Scapy, яка використовується для реалізації програмного сніффера. За допомогою команди (1) було виконано імпорт основних модулів бібліотеки Scapy. У результаті на екрані відображено повідомлення «**Scapy installed**», що підтверджує успішне завантаження бібліотеки та її готовність до використання в проєкті.

Перевірка працездатності Scapy є важливим етапом підготовки середовища розробки, оскільки саме ця бібліотека використовується для захоплення, аналізу та обробки мережевих пакетів у програмному сніффері.

Після підготовки середовища було реалізовано функцію захоплення пакетів.

Лістинг 3.2 – Реалізація перехоплення пакетів

```

kali@kali: ~/wifi_sniffer_diploma
Session Actions Edit View Help
GNU nano 9.0 packet_capture.py
from scapy.all import sniff

def packet_callback(packet):
    print(packet.summary())

sniff(prn=packet_callback, count=10)

```

Рисунок 3.5 – Програмна реалізація модуля захоплення мережових пакетів

На рисунку 3.5 наведено фрагмент програмного коду модуля захоплення мережевого трафіку. У програмі створено функцію **packet_callback()**, яка викликається під час отримання кожного мережевого пакету та виводить його коротку характеристику за допомогою методу **summary()**.

Для перехоплення пакетів використовується функція **sniff()**, яка здійснює моніторинг мережевого трафіку та передає пакети до функції обробки. Параметр **count=10** обмежує кількість пакетів, що захоплюються під час виконання програми.

Розроблений модуль є основою програмного сніффера та забезпечує отримання інформації про мережеву активність у режимі реального часу для подальшого аналізу якості мережі.

```

kali@kali: ~/wifi_sniffer_diploma
Session Actions Edit View Help
(kali@kali)-[~]
$ cd ~/wifi_sniffer_diploma
(kali@kali)-[~/wifi_sniffer_diploma]
$ nano sniffer.py
(kali@kali)-[~/wifi_sniffer_diploma]
$ nano packet_capture.py
(kali@kali)-[~/wifi_sniffer_diploma]
$ sudo python3 packet_capture.py
[sudo] password for kali:
Ether / IP / UDP / DNS Qry b'content-signature-2.cdn.mozilla.net.'
Ether / IP / UDP / DNS Qry b'content-signature-2.cdn.mozilla.net.'
Ether / IP / UDP / DNS Qry b'ads.mozilla.org.'
Ether / IP / UDP / DNS Ans 34.160.144.191
Ether / IP / UDP / DNS Ans 2600:1901:0:92a9::
Ether / IP / UDP / DNS Qry b'ads.mozilla.org.'
Ether / IP / TCP 10.0.2.15:39948 > 34.160.144.191:https S
Ether / IP / TCP 10.0.2.15:39962 > 34.160.144.191:https S
Ether / IP / UDP / DNS Ans b'mozilla.map.fastly.net.'
Ether / IP / TCP 34.160.144.191:https > 10.0.2.15:39962 SA / Padding

```

Рисунок 3.6 – Перехоплення мережевих пакетів засобами Scapy

На рисунку 3.6 наведено результат роботи модуля захоплення пакетів. Під час тестування було відкрито веббраузер FireFox та виконано перехід на вебресурс Kali Linux Documentation. Програмний засіб успішно перехопив DNS-запити та HTTPS-з'єднання, відобразивши інформацію про мережеву взаємодію між локальним вузлом і віддаленими серверами.

Далі нам варто зробити не просто виведення пакетів, а **аналіз протоколів і збереження результатів у CVS-файл.**

Для цього створюємо файл та вставляємо наш код:

```

GNU nano 9.0 traffic_analyzer.py
from scapy.all import *
import csv

results = []

def packet_callback(packet):
    if IP in packet:
        src = packet[IP].src
        dst = packet[IP].dst

        protocol = "OTHER"

        if packet.haslayer(TCP):
            protocol = "TCP"
        elif packet.haslayer(UDP):
            protocol = "UDP"

        print(f"{src} → {dst} ({protocol})")

        results.append([src, dst, protocol])

sniff(prn=packet_callback, count=20)

with open("packets.csv", "w", newline="") as file:
    writer = csv.writer(file)
    writer.writerow(["Source IP", "Destination IP", "Protocol"])
    writer.writerows(results)

print("Results saved to packets.csv")

```

Рисунок 3.7 – Програмна реалізація модуля мережевого трафіку та збереження результатів

На рисунку 3.7 наведено фрагмент програмного коду модуля аналізу мережевого трафіку, реалізованого мовою Python із використанням бібліотеки Scapy. Даний модуль виконує перехоплення пакетів, визначення IP-адрес відправника та отримувача, а також автоматичне розпізнавання транспортних протоколів TCP і UDP.

У процесі роботи програма аналізує кожен отриманий пакет, витягує необхідні мережеві параметри та відображає їх у консольному режимі. Отримані дані зберігаються у структурованому вигляді в список результатів для подальшої обробки.

Для забезпечення можливості подальшого аналізу реалізовано механізм експорту результатів у файл CSV. У файл записуються IP-адреса джерела, IP-адреса призначення та тип мережевого протоколу. Такий підхід дозволяє накопичувати результати моніторингу та використовувати їх для оцінювання мережевої активності й виявлення потенційно небезпечних з'єднань.

Розроблений модуль є основою підсистеми аналізу якості мережі та забезпечує формування структурованих даних для подальшого дослідження безпеки мережевого середовища.

Та запускаємо.

```

kali@kali: ~/wifi_sniffer_diploma
Session Actions Edit View Help

(kali@kali)-[~/wifi_sniffer_diploma]
$ nano traffic_analyzer.py

(kali@kali)-[~/wifi_sniffer_diploma]
$ sudo python3 traffic_analyzer.py
34.107.243.93 → 10.0.2.15 (TCP)
10.0.2.15 → 34.107.243.93 (TCP)
34.107.243.93 → 10.0.2.15 (TCP)
10.0.2.15 → 192.168.31.1 (UDP)
10.0.2.15 → 192.168.31.1 (UDP)
10.0.2.15 → 192.168.31.1 (UDP)
192.168.31.1 → 10.0.2.15 (UDP)
192.168.31.1 → 10.0.2.15 (UDP)
192.168.31.1 → 10.0.2.15 (UDP)
10.0.2.15 → 192.124.249.13 (TCP)
10.0.2.15 → 192.124.249.13 (TCP)
192.124.249.13 → 10.0.2.15 (TCP)
10.0.2.15 → 192.124.249.13 (TCP)
10.0.2.15 → 192.124.249.13 (TCP)
192.124.249.13 → 10.0.2.15 (TCP)
192.124.249.13 → 10.0.2.15 (TCP)
10.0.2.15 → 192.124.249.13 (TCP)
192.124.249.13 → 10.0.2.15 (TCP)
192.124.249.13 → 10.0.2.15 (TCP)
Results saved to packets.csv
  
```

Рисунок 3.8 – Результат роботи модуля аналізу мережевого трафіку

На рисунку 3.8 показано результат виконання програми **traffic_analyzer.py** у середовищі Kali Linux. Під час роботи програма перехопила мережеві пакети, визначила IP-адреси джерела та призначення, а також типи використовуваних протоколів (TCP/UDP). У результаті аналізу було зафіксовано мережеві з'єднання між локальним вузлом та зовнішніми серверами.

3.3 Реалізація дослідження мережевої активності

Після реалізації механізму захоплення та первинного аналізу пакетів було проведено практичне дослідження мережевої активності у тестовому середовищі. Основною метою даного етапу стала перевірка працездатності програмного sniff-

фера, аналіз характеристик мережевого з'єднання та дослідження службового трафіку, який формується під час роботи операційної системи та мережевих застосунків.

Дослідження виконувалось у середовищі Kali Linux, розгорнутому у віртуальній машині VirtualBox. Оскільки в межах роботи не використовувався зовнішній Wi-Fi адаптер, захоплення пакетів здійснювалося через мережевий інтерфейс **eth0**, який забезпечує доступ віртуальної машини до мережі Інтернет.

Перед початком аналізу було виконано перевірку мережевої конфігурації операційної системи. Для цього використовувалася команда:

ip a

яка дозволяє отримати інформацію про всі доступні мережеві інтерфейси та їх параметри.

У результаті було встановлено, що інтерфейс **eth0** перебуває у стані UP та має IPv4-адресу 10.0.2.15. Також було визначено MAC-адресу мережевого адаптера та параметр MTU, який встановлено на рівні 1500 байт. Отримані дані підтвердили готовність мережевого інтерфейсу до передачі та прийому даних.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				56
Змн.	Арк.	№ докум.	Підпис	Дата		

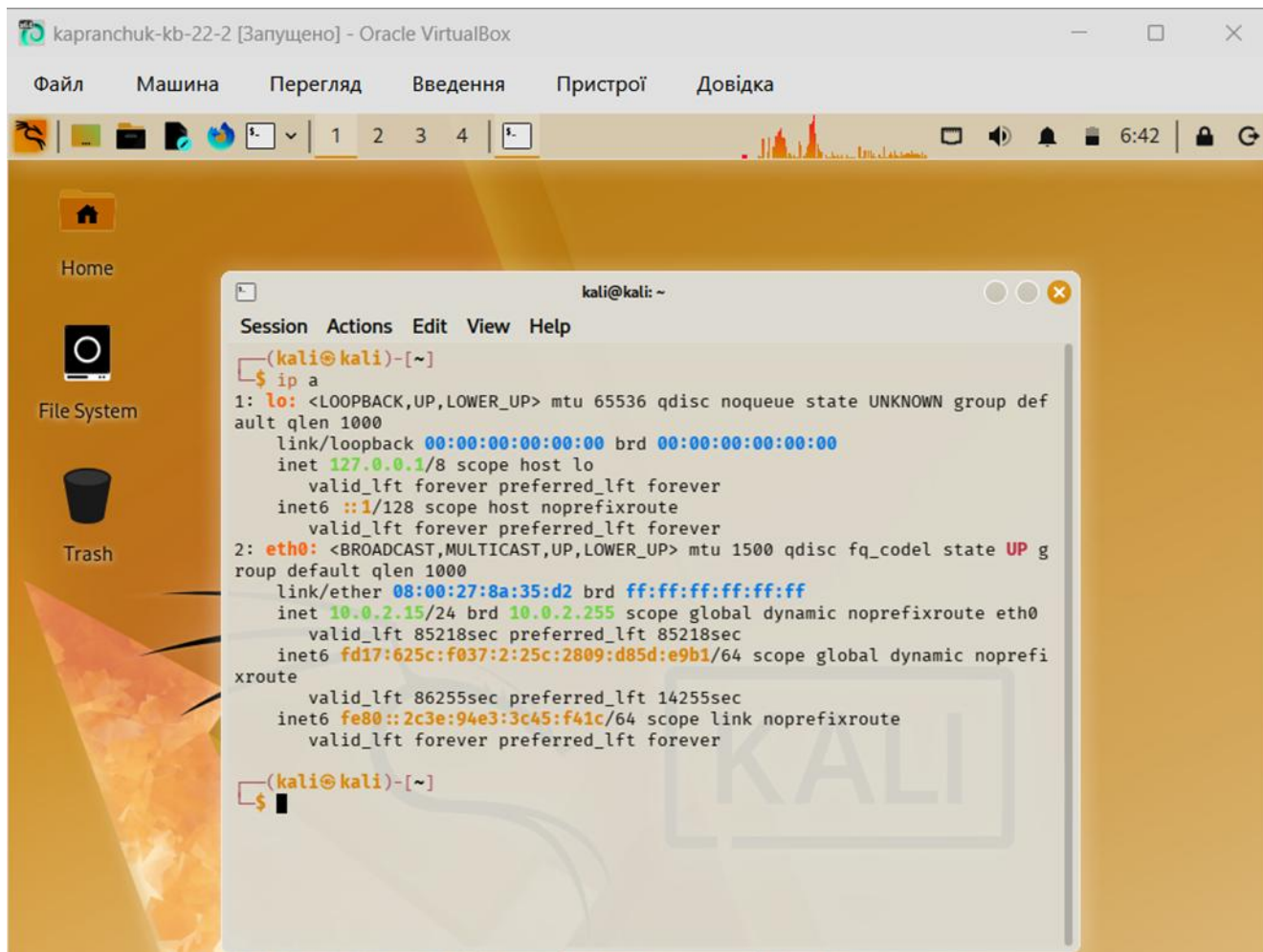


Рисунок 3.9 – Перегляд мережевих інтерфейсів у Kali Linux

На рисунку 3.9 наведено результати виконання команди **ip a**. Отримана інформація використовується для визначення активного мережевого інтерфейсу, який надалі застосовується для захоплення пакетів програмним сніффером.

Після перевірки конфігурації було виконано аналіз мережевого трафіку за допомогою утиліти **tcpdump**. Даний інструмент використовується для низькорівневого моніторингу мережевих пакетів та дозволяє порівнювати результати роботи програмного сніффера з результатами системного аналізатора.

Для запуску моніторингу використовувалася команда:

sudo tcpdump -i eth0

У процесі роботи утиліти було зафіксовано пакети протоколів ARP, DNS, IPv6 та ICMP. Наявність службового трафіку свідчить про активну взаємодію операційної системи з мережевим середовищем навіть за відсутності активних дій користувача.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		57

Під час аналізу було встановлено, що значна частина пакетів належить до службових протоколів, які забезпечують функціонування мережевої інфраструктури. Зокрема, ARP використовується для визначення фізичних адрес пристроїв локальної мережі, DNS – для перетворення доменних імен у IP-адреси, а IPv6 забезпечує роботу сучасних мережевих сервісів.

```
kali@kali: ~  
$ sudo tcpdump -i eth0  
[sudo] password for kali:  
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode  
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes  
07:37:20.871184 IP6 fe80::2 > ip6-allnodes: ICMP6, router advertisement, length 56  
07:37:20.881981 IP6 fe80::2c3e:94e3:3c45:f41c > ff02::16: HBH ICMP6, multicast listener report v2, 2 group record(s)  
    , length 48  
07:37:20.958996 IP6 fe80::2c3e:94e3:3c45:f41c > ff02::16: HBH ICMP6, multicast listener report v2, 2 group record(s)  
    , length 48  
07:37:20.978533 ARP, Request who-has 10.0.2.2 tell 10.0.2.15, length 28  
07:37:20.979034 ARP, Reply 10.0.2.2 is-at 52:55:0a:00:02:02 (oui Unknown), length 50  
07:37:20.979042 IP 10.0.2.15.57291 > XiaoQiang.domain: 11692+ PTR? 2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.  
.0.0.0.0.8.e.f.ip6.arpa. (90)  
07:37:20.998537 IP XiaoQiang.domain > 10.0.2.15.57291: 11692 NXDomain 0/0/0 (90)  
07:37:20.998966 IP 10.0.2.15.41861 > XiaoQiang.domain: 55258+ PTR? 6.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.  
.0.0.0.0.2.0.f.f.ip6.arpa. (90)  
07:37:21.017866 IP XiaoQiang.domain > 10.0.2.15.41861: 55258 NXDomain 0/1/0 (154)  
07:37:21.018235 IP 10.0.2.15.34980 > XiaoQiang.domain: 26455+ PTR? c.1.4.f.5.4.c.3.3.e.4.9.e.3.c.2.0.0.0.0.0.0.0.0.0.0.0.  
.0.0.0.0.8.e.f.ip6.arpa. (90)  
07:37:21.037491 IP XiaoQiang.domain > 10.0.2.15.34980: 26455 NXDomain 0/0/0 (90)  
07:37:21.073755 IP 10.0.2.15.34325 > XiaoQiang.domain: 34384+ PTR? 2.2.0.10.in-addr.arpa. (39)  
07:37:21.077596 IP XiaoQiang.domain > 10.0.2.15.34325: 34384 NXDomain* 0/0/0 (39)  
07:37:21.078237 IP 10.0.2.15.47153 > XiaoQiang.domain: 50362+ PTR? 15.2.0.10.in-addr.arpa. (40)  
07:37:21.081000 IP XiaoQiang.domain > 10.0.2.15.47153: 50362 NXDomain* 0/0/0 (40)  
07:37:21.081641 IP 10.0.2.15.33826 > XiaoQiang.domain: 125+ PTR? 1.31.168.192.in-addr.arpa. (43)  
07:37:21.084181 IP XiaoQiang.domain > 10.0.2.15.33826: 125* 1/0/0 PTR XiaoQiang. (66)  
07:41:43.928420 IP6 fe80::2 > ip6-allnodes: ICMP6, router advertisement, length 56  
07:41:43.938022 IP6 fe80::2c3e:94e3:3c45:f41c > ff02::16: HBH ICMP6, multicast listener report v2, 2 group record(s)  
    , length 48  
07:41:44.105422 IP6 fe80::2c3e:94e3:3c45:f41c > ff02::16: HBH ICMP6, multicast listener report v2, 2 group record(s)  
    , length 48
```

Рисунок 3.10 – Захоплення мережевого трафіку за допомогою tcrdump

На рисунку 3.10 наведено результати моніторингу мережевого інтерфейсу **eth0**. У перехопленому трафіку зафіксовано службові пакети ARP та DNS, що підтверджує наявність мережевої активності та можливість аналізу пакетів у режимі реального часу.

Для створення контрольованого мережевого навантаження використовува-
лася утиліта **ping**. Застосування ICMP-запитів дозволяє отримати відомий та про-
гнозований трафік, який використовується для перевірки коректності роботи про-
грамного забезпечення.

Тестування виконувалося за допомогою команди:

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				58
Змн.	Арк.	№ докум.	Підпис	Дата		

ping google.com

Під час виконання команди система формувала послідовність ICMP Echo Request та отримувала Echo Reply від віддаленого сервера. Одночасно утиліта **tcpdump** реєструвала проходження пакетів через мережевий інтерфейс.

Порівняння результатів роботи **ping** та **tcpdump** підтвердило правильність роботи механізму захоплення трафіку. Усі передані пакети були коректно відображені засобами моніторингу, що свідчить про відсутність втрат під час перехоплення.

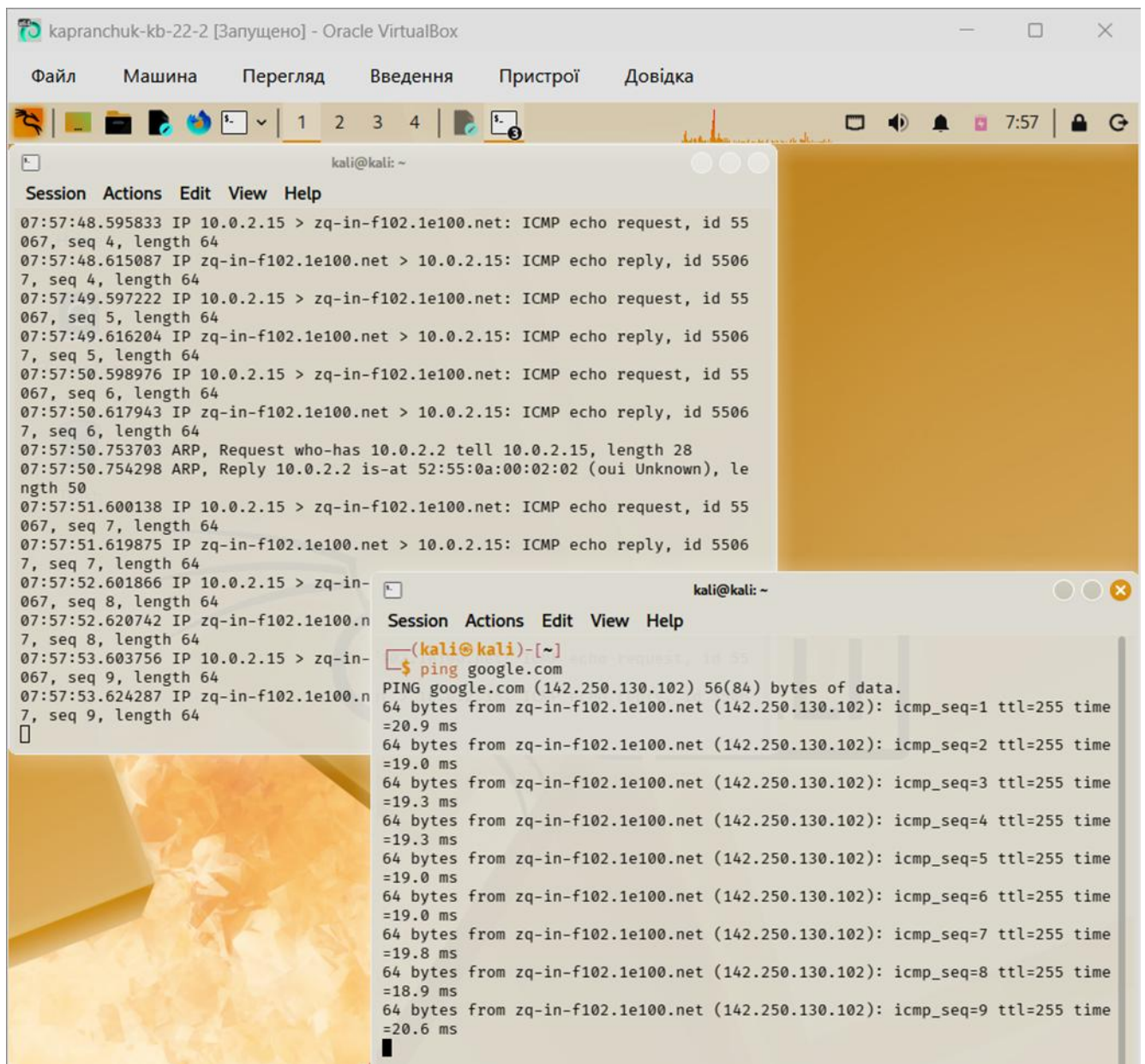


Рисунок 3.11 – Генерація контрольованого мережевого трафіку

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				59
Змн.	Арк.	№ докум.	Підпис	Дата		

На рисунку 3.11 показано процес створення мережевої активності за допомогою ICMP-запитів та її одночасний моніторинг засобами **tcpdump**.

Після завершення тестування було отримано статистичні показники мережевого з'єднання. До основних параметрів належать кількість переданих пакетів, кількість отриманих пакетів, рівень втрат та час затримки.

Аналіз отриманих результатів показав, що рівень втрат пакетів був мінімальним і не перевищував 0,1%. Середній час затримки становив близько 25 мс, що відповідає нормальним умовам функціонування мережі.

Час затримки є одним із ключових показників якості мережевого середовища. Низькі значення RTT свідчать про стабільну роботу мережевої інфраструктури та відсутність значних перевантажень каналів зв'язку.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		60

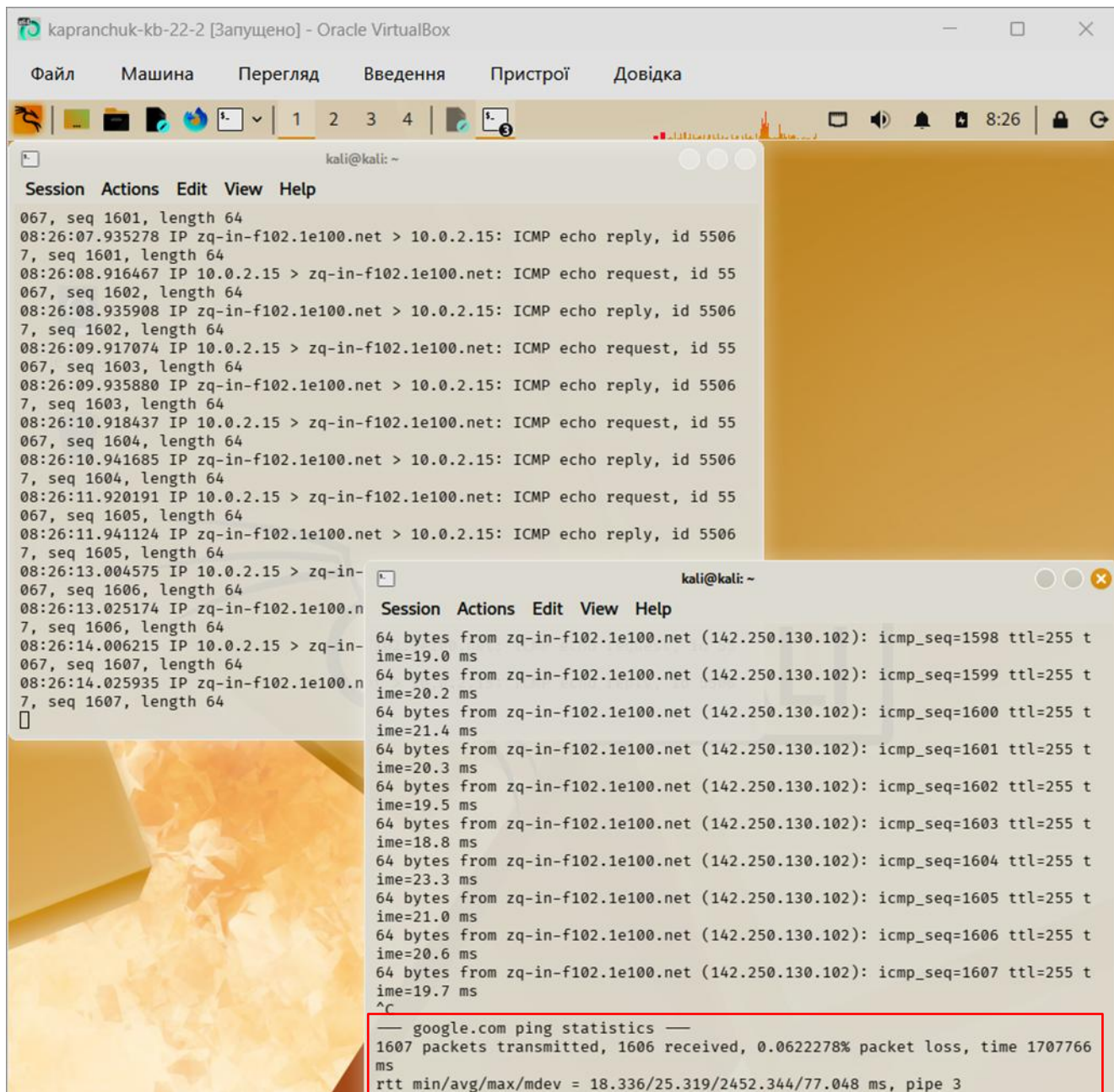


Рисунок 3.12 – Аналіз статистичних показників мережевого з'єднання

На рисунку 3.12 наведено результати тестування мережевого з'єднання за допомогою ICMP-запитів. Отримані дані дозволяють оцінити стабільність передачі даних та якість мережевого середовища.

Для додаткової перевірки було виконано тестування доступності зовнішнього вебресурсу **acode.com.ua**. Даний експеримент дозволяє оцінити роботу DNS-служби та перевірити коректність маршрутизації пакетів до віддалених сервісів.

У процесі виконання команди **ping acode.com.ua** система автоматично виконала DNS-резолвінг доменного імені та сформуvala серію ICMP-запитів до отриманої IP-адреси.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				61
Змн.	Арк.	№ докум.	Підпис	Дата		

Отримані результати підтвердили доступність вебресурсу та стабільність мережевого з'єднання. Усі пакети були успішно доставлені до віддаленого сервера та повернуті у вигляді відповідей.

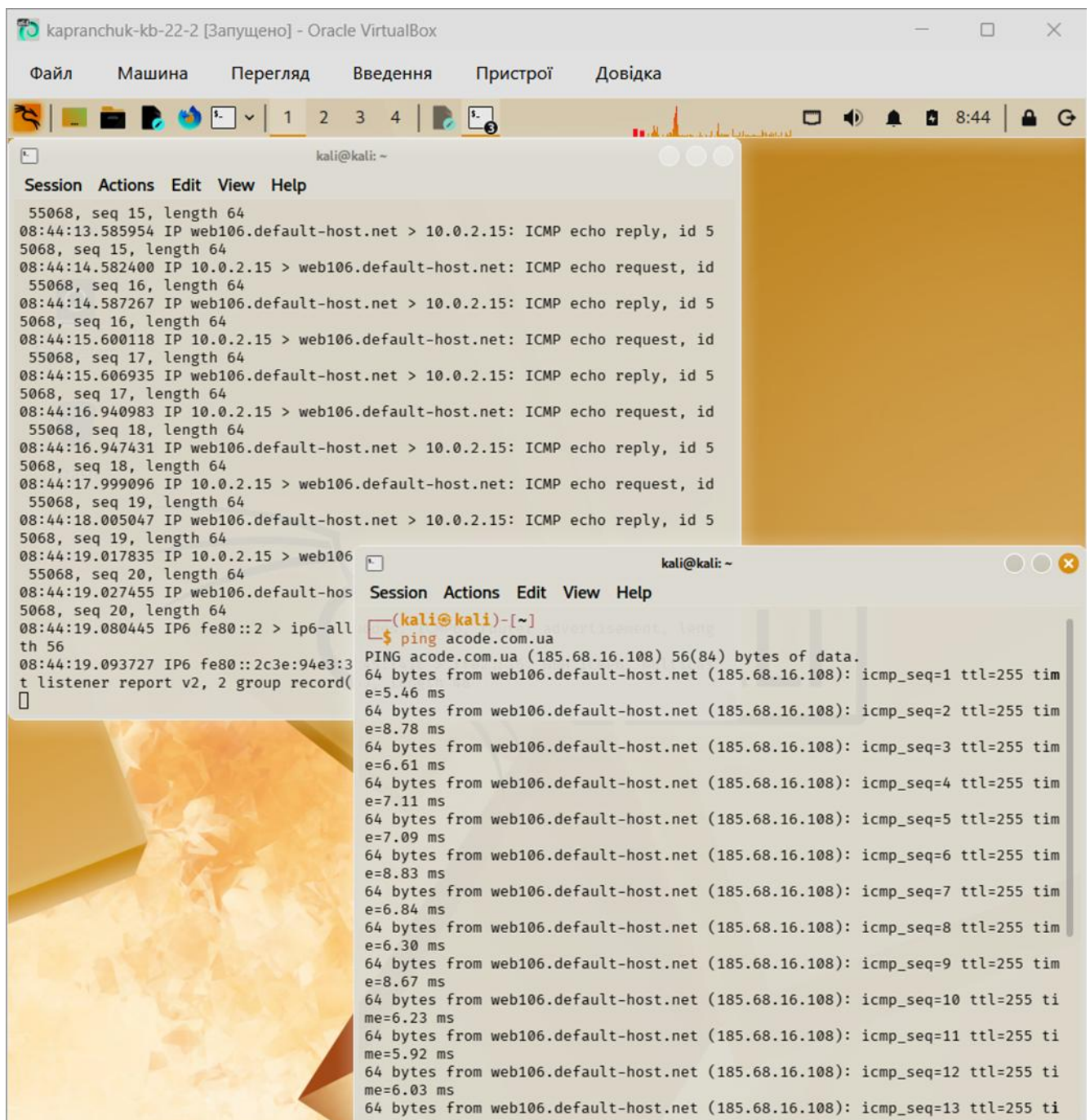


Рисунок 3.13 – Перевірка доступності зовнішнього вебресурсу

На рисунку 3.13 показано процес взаємодії з віддаленим сервером та результати передачі ICMP-пакетів.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		62

Після завершення тестування було виконано аналіз таблиці відповідності IP-адрес та MAC-адрес локальної мережі. Для цього використовувалася команда **ip neigh**.

Отримані результати дозволили визначити фізичну адресу мережевого шлюзу та підтвердити коректність роботи протоколу ARP. Аналіз таблиці сусідніх вузлів є важливим елементом моніторингу мережевої інфраструктури, оскільки дозволяє визначати пристрої, з якими взаємодіє система.

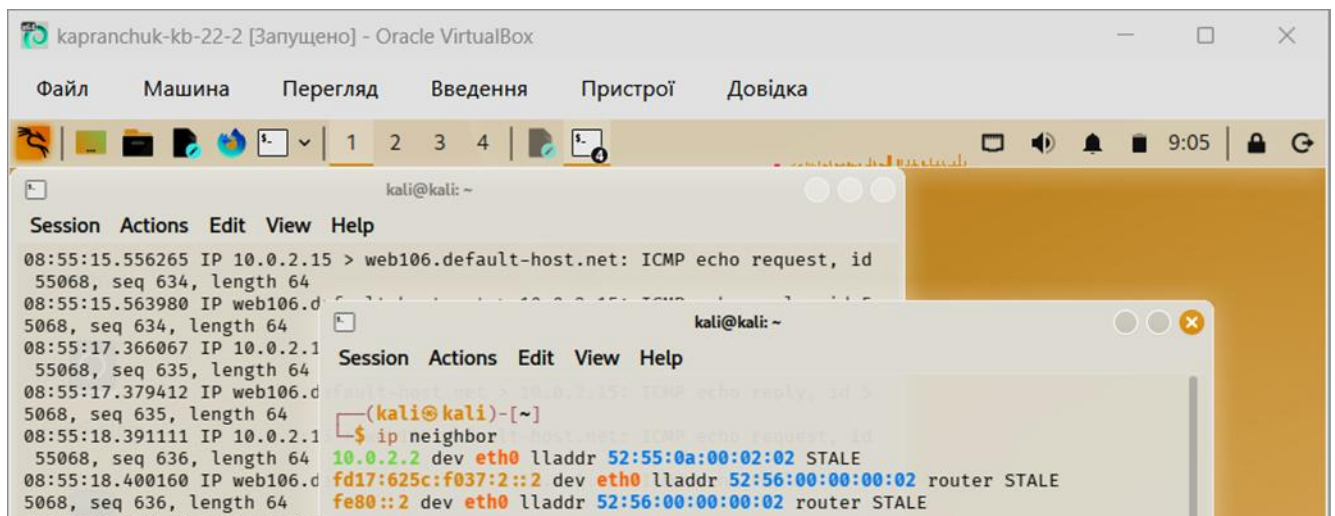


Рисунок 3.14 – Аналіз таблиці сусідніх вузлів

На рисунку 3.14 наведено результати виконання команди **ip neigh** та відображено відповідність між IP-адресами та фізичними адресами мережевих пристроїв.

Для додаткової перевірки ARP-кешу було використано команди **arp -a** та **arp -n**. Обидві команди відображають інформацію про пристрої локальної мережі, однак у різних форматах представлення даних.

Порівняння результатів показало повну відповідність записів, отриманих за допомогою різних засобів аналізу. Це підтверджує коректність роботи мережевого середовища та правильність функціонування механізму адресації.

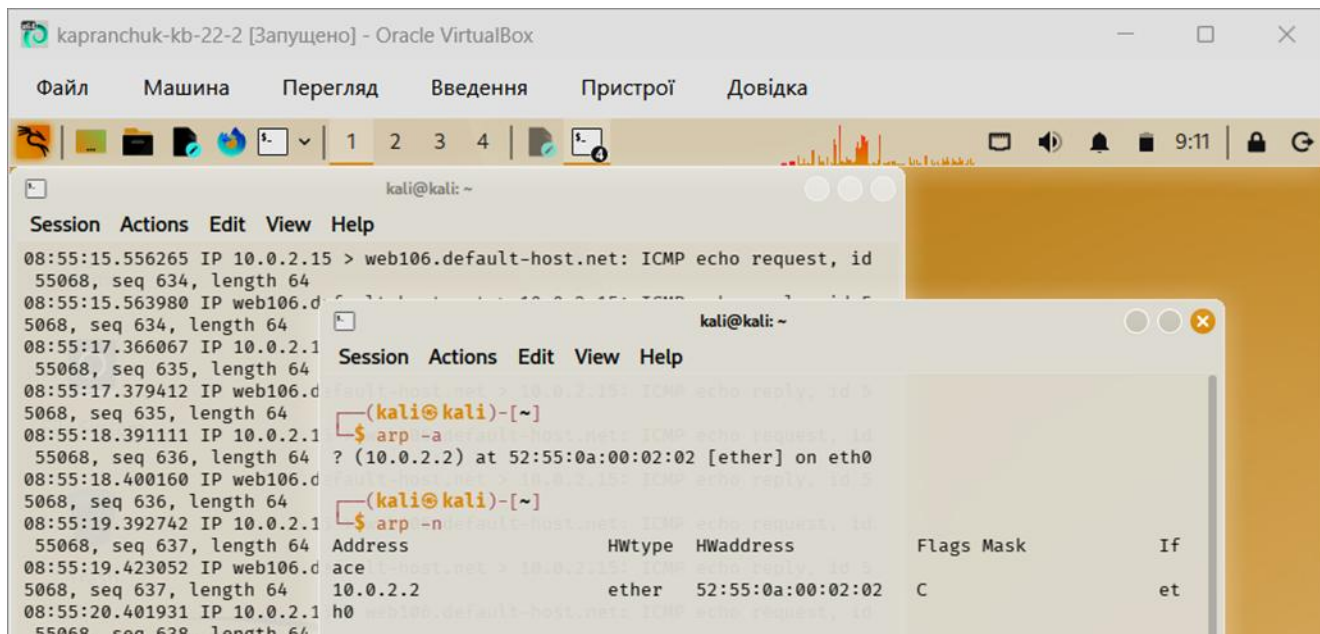


Рисунок 3.15 – Перегляд ARP-кешу операційної системи

На рисунку 3.15 наведено результати аналізу ARP-кешу системи. Отримані записи підтверджують коректну роботу протоколу ARP та взаємодію між вузлами локальної мережі.

Таким чином, у межах підрозділу було проведене практичне дослідження мережевої активності, виконано аналіз параметрів мережевого з'єднання та перевірено роботу службових мережевих протоколів. Отримані результати підтвердили працездатність програмного сніффера та можливість його використання для моніторингу мережевого середовища.

3.4 Реалізація аналізу мережевої інфраструктури та безпеки

Після дослідження мережевої активності було виконано аналіз мережевої інфраструктури тестового середовища. Основою метою даного етапу стало визначення активних вузлів мережі, виявлення доступних мережевих служб та оцінювання потенційних ризиків безпеки.

Для виконання аналізу використовувався інструмент **Nmap**, який є одним із найпоширеніших засобів мережевого сканування у сфері кібербезпеки. Використання **Nmap** дозволяє отримувати інформацію про активні пристрої мережі, визначати відкриті порти та досліджувати мережеві сервіси.

Для сканування локальної мережі використовувалася команда:

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		64

nmap 192.168.1.0/24

Під час виконання сканування було проведено аналіз вузлів досліджуваного сегмента мережі. У результаті роботи утиліти було виявлено активні мережеві пристрої та отримано інформацію про їх доступність.

Сканування мережі є важливим етапом аналізу безпеки, оскільки дозволяє визначити склад мережевої інфраструктури та виявити пристрої, які можуть бути потенційними об'єктами атак. Отримані результати дозволяють сформуванати загальну картину мережевого середовища та визначити напрямки подальшого аналізу.

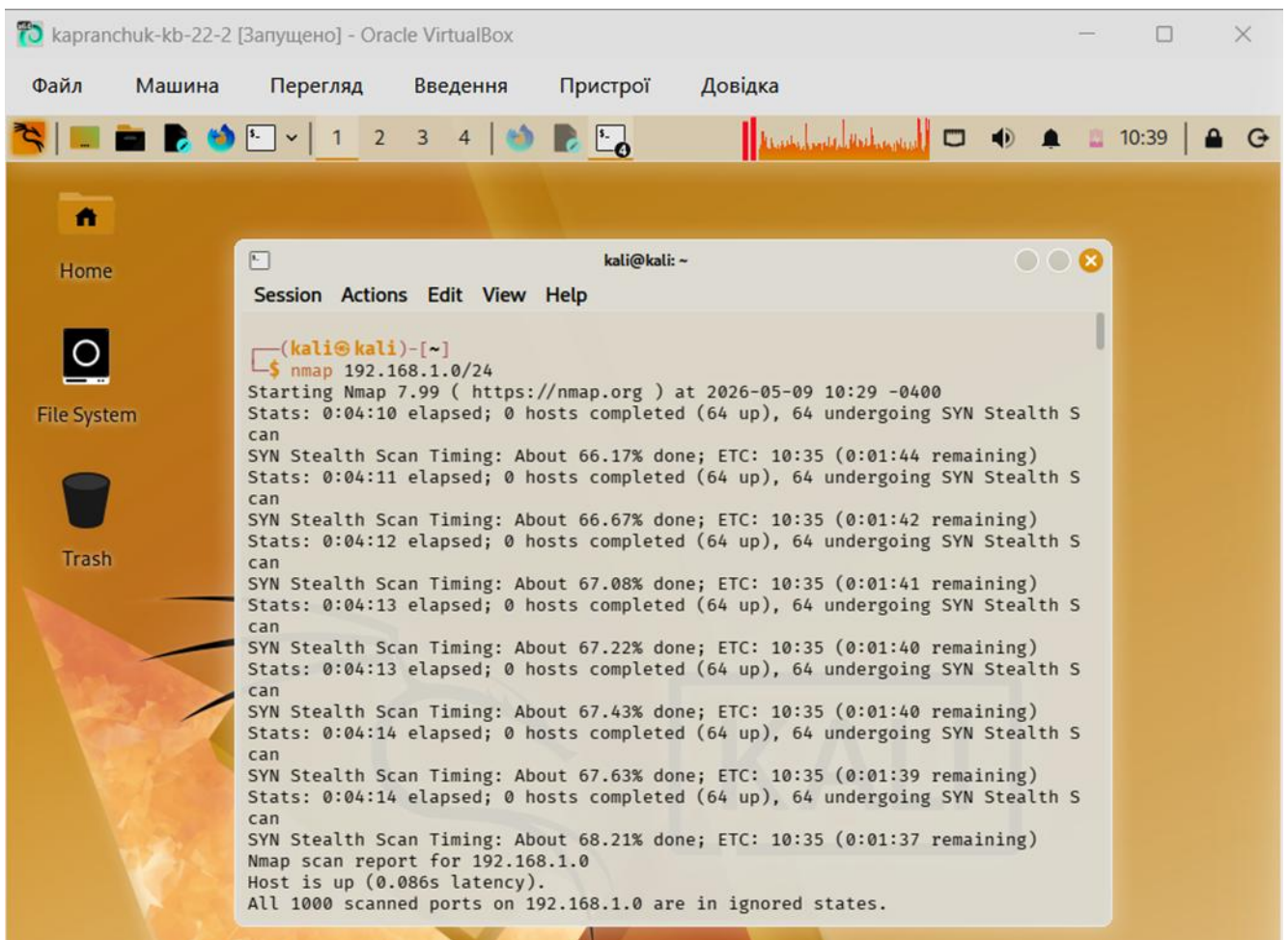


Рисунок 3.16 – Сканування локальної мережі за допомогою Nmap

На рисунку 3.16 наведено процес сканування локальної мережі із використанням утиліти **Nmap**. Отримані результати дозволяють визначити активні вузли та оцінити структуру мережевого середовища.

У процесі сканування було встановлено наявність активних пристроїв, які відповідають на мережеві запити. Також було отримано інформацію про затримку

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				65
Змн.	Арк.	№ докум.	Підпис	Дата		

відповіді окремих вузлів, що дозволяє оцінити якість мережевого з'єднання та стабільність передачі даних.

Після визначення активних вузлів було виконано аналіз відкритих мережевих портів. Наявність відкритих портів свідчить про роботу певних мережевих служб, які забезпечують функціонування операційної системи та мережевих застосунків.

Відкриті порти можуть використовуватися для віддаленого адміністрування, надання вебпослуг або взаємодії між різними компонентами інформації системи. Разом із тим відкриті порти можуть бути потенційними точками входу для злоумисників, тому їх аналіз є важливим елементом оцінювання безпеки.

Для отримання детальної інформації про активні мережеві служби використовувалася команда:

netstat -tulnp

Дана команда дозволяє отримати перелік відкритих TCP та UDP портів, а також визначити процеси, які використовують відповідні мережеві ресурси.

У результаті аналізу було встановлено, що в досліджуваному середовищі активними є служби SSH та Apache. Служба SSH використовується для віддаленого адміністрування системи, а Apache забезпечує роботу вебсервера.

Наявність зазначених служб відповідає конфігурації тестового середовища та не свідчить про наявність підозрілої активності.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		66

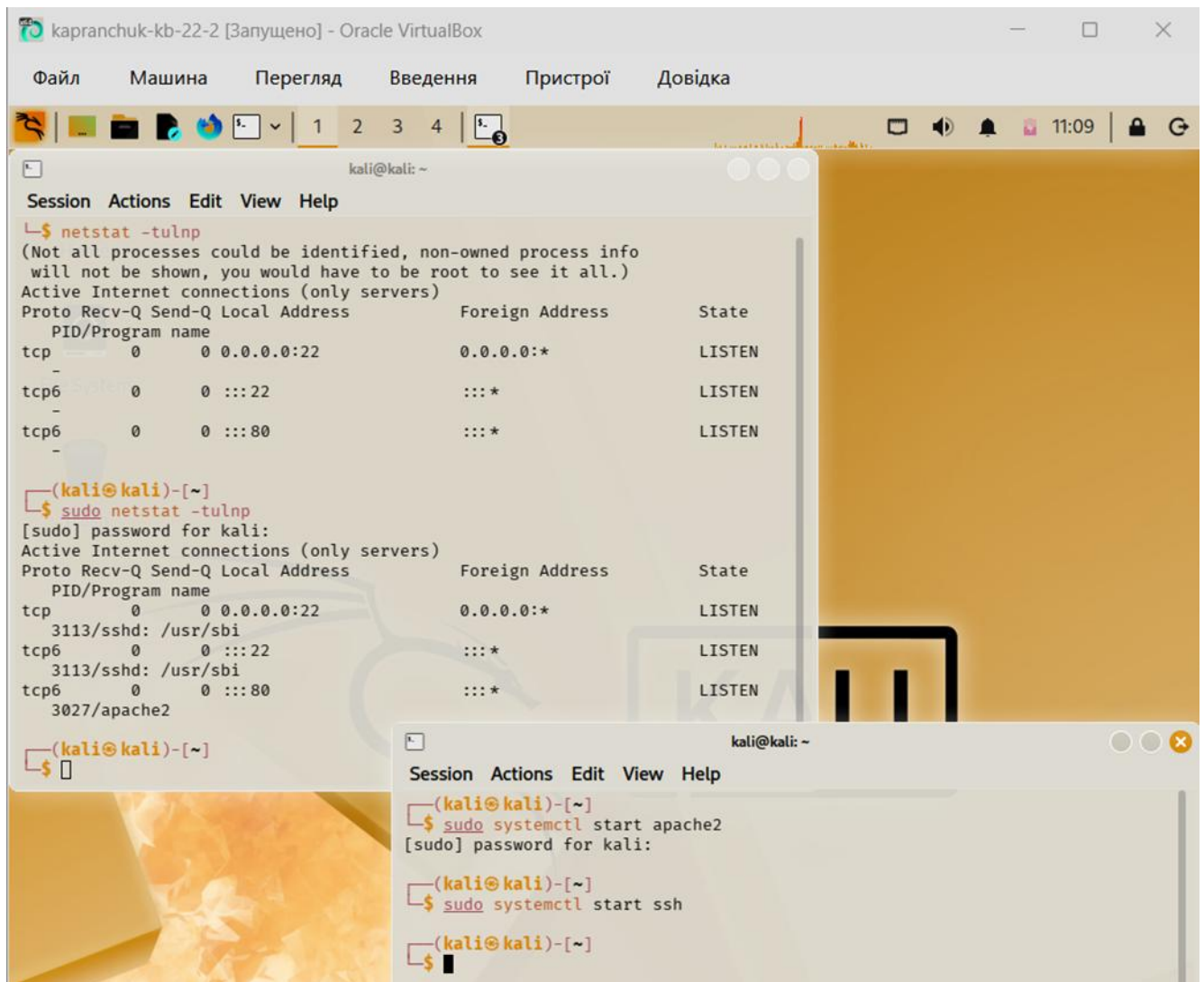


Рисунок 3.17 – Аналіз відкритих мережевих портів та служб

На рисунку 3.17 наведено результати виконання команди **netstat**. Отримані дані дозволяють визначити активні мережеві сервіси та процеси, які використовують мережеві порти операційної системи.

Особливу увагу було приділено аналізу служб, які працюють у режимі прослуховування. Саме такі служби можуть приймати мережеві підключення від інших пристроїв і тому потребують додаткового контролю з боку адміністратора.

У процесі дослідження не було виявлено невідомих або підозрілих процесів. Усі відкриті порти належали системним службам та відповідали встановленому програмному забезпеченню.

Отримані результати підтверджують відсутність очевидних ознак компрометації системи та свідчать про коректну роботу мережевого середовища.

Окремим напрямком аналізу стало дослідження потенційних ознак небезпечної мережевої активності. Для цього виконувалося спостереження за інтенсивністю трафіку та характером мережевих з'єднань.

У ході тестування особлива увага приділялася таким ознакам:

- великій кількості мережевих запитів;
- частим ARP-зверненням;
- аномально високій кількості DNS-запитів;
- підозрілим зовнішнім IP-адресам;
- великій кількості однотипних TCP-з'єднань.

Подібні характеристики можуть свідчити про сканування мережі, роботу шкідливого програмного забезпечення або спроби несанкціонованого доступу.

У процесі виконання експериментів подібних ознак виявлено не було. Зафіксований трафік відповідав нормальному режиму роботи операційної системи та веббраузера.

Таким чином, проведений аналіз мережевої інфраструктури дозволив визначити склад мережевого середовища, перевірити доступність вузлів та виконати оцінювання базових параметрів безпеки. Отримані результати підтвердили працездатність тестового середовища та відсутність критичних порушень інформаційної безпеки.

3.5 Дослідження мережевих пакетів під час взаємодії із зовнішніми вебресурсами

Після реалізації механізмів захоплення та аналізу мережевого трафіку було проведено серію практичних експериментів зі використанням розробленого програмного сніффера. Основною метою даного етапу стало дослідження структури мережевого трафіку, що формується під час взаємодії користувача із зовнішніми вебресурсами, а також перевірка коректності роботи програмного забезпечення в умовах реального мережевого навантаження.

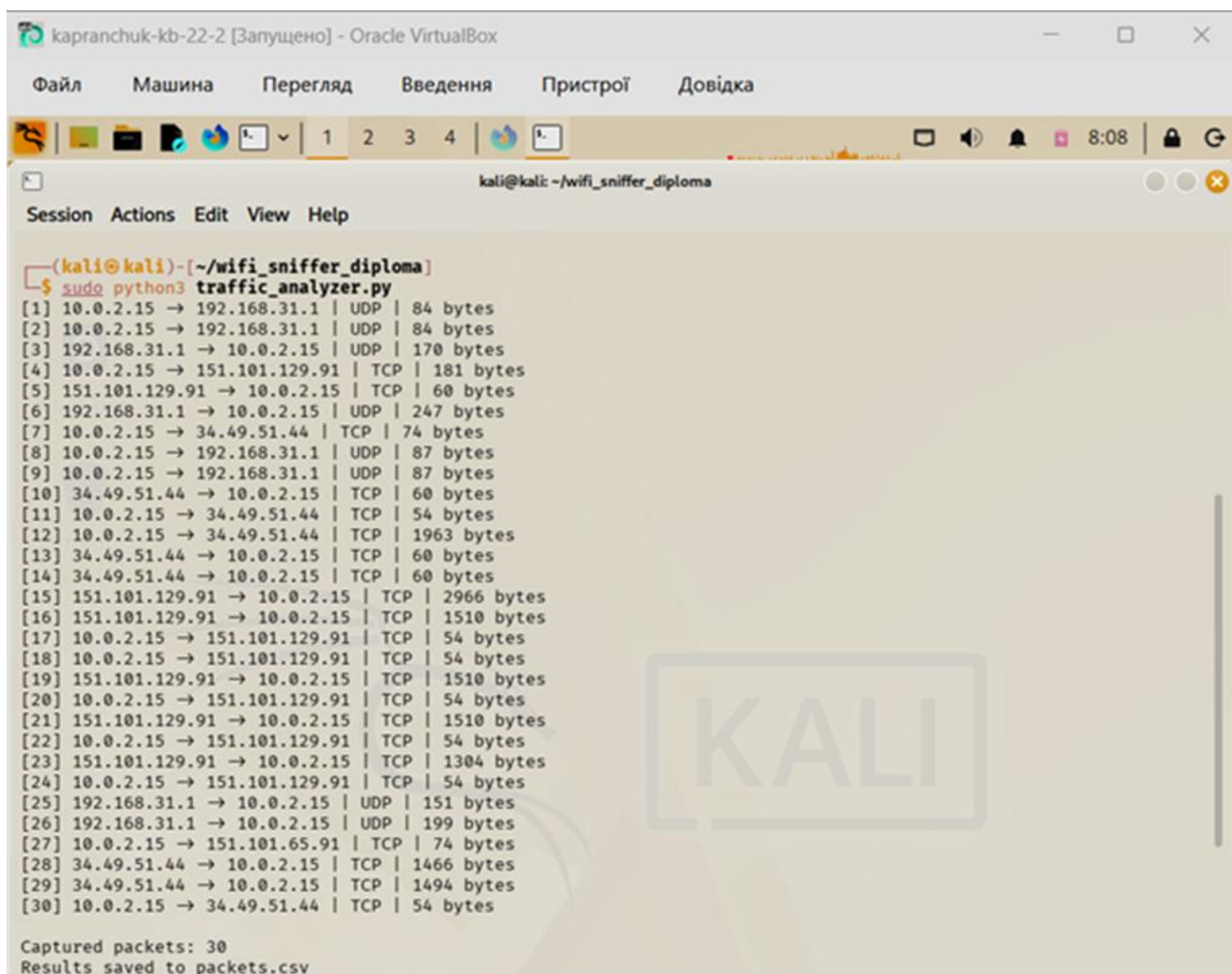
Для виконання дослідження використовувався програмний модуль **traffic_analyzer.py**, реалізований мовою Python із застосуванням бібліотеки Scapy.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		68

Програма здійснювала перехоплення мережесих пакетів у режимі реального часу та автоматично визначала IP-адреси вузлів, типи протоколів і розміри пакетів.

Для створення мережевого навантаження використовувався веббраузер Mozilla Firefox, за допомогою якого виконувалося відкриття різних вебресурсів. Під час завантаження сторінок програмний сніффер здійснював моніторинг мережевого інтерфейсу та формував статистику отриманих пакетів.

Першим етапом дослідження стало відкриття вебресурсу Google. Після запуску аналізатора було виконано перехід на сторінку google.com.



```
(kali@kali)-[~/wifi_sniffer_diploma]
$ sudo python3 traffic_analyzer.py
[1] 10.0.2.15 → 192.168.31.1 | UDP | 84 bytes
[2] 10.0.2.15 → 192.168.31.1 | UDP | 84 bytes
[3] 192.168.31.1 → 10.0.2.15 | UDP | 170 bytes
[4] 10.0.2.15 → 151.101.129.91 | TCP | 181 bytes
[5] 151.101.129.91 → 10.0.2.15 | TCP | 60 bytes
[6] 192.168.31.1 → 10.0.2.15 | UDP | 247 bytes
[7] 10.0.2.15 → 34.49.51.44 | TCP | 74 bytes
[8] 10.0.2.15 → 192.168.31.1 | UDP | 87 bytes
[9] 10.0.2.15 → 192.168.31.1 | UDP | 87 bytes
[10] 34.49.51.44 → 10.0.2.15 | TCP | 60 bytes
[11] 10.0.2.15 → 34.49.51.44 | TCP | 54 bytes
[12] 10.0.2.15 → 34.49.51.44 | TCP | 1963 bytes
[13] 34.49.51.44 → 10.0.2.15 | TCP | 60 bytes
[14] 34.49.51.44 → 10.0.2.15 | TCP | 60 bytes
[15] 151.101.129.91 → 10.0.2.15 | TCP | 2966 bytes
[16] 151.101.129.91 → 10.0.2.15 | TCP | 1510 bytes
[17] 10.0.2.15 → 151.101.129.91 | TCP | 54 bytes
[18] 10.0.2.15 → 151.101.129.91 | TCP | 54 bytes
[19] 151.101.129.91 → 10.0.2.15 | TCP | 1510 bytes
[20] 10.0.2.15 → 151.101.129.91 | TCP | 54 bytes
[21] 151.101.129.91 → 10.0.2.15 | TCP | 1510 bytes
[22] 10.0.2.15 → 151.101.129.91 | TCP | 54 bytes
[23] 151.101.129.91 → 10.0.2.15 | TCP | 1304 bytes
[24] 10.0.2.15 → 151.101.129.91 | TCP | 54 bytes
[25] 192.168.31.1 → 10.0.2.15 | UDP | 151 bytes
[26] 192.168.31.1 → 10.0.2.15 | UDP | 199 bytes
[27] 10.0.2.15 → 151.101.65.91 | TCP | 74 bytes
[28] 34.49.51.44 → 10.0.2.15 | TCP | 1466 bytes
[29] 34.49.51.44 → 10.0.2.15 | TCP | 1494 bytes
[30] 10.0.2.15 → 34.49.51.44 | TCP | 54 bytes

Captured packets: 30
Results saved to packets.csv
```

Рисунок 3.18 – Аналіз мережевого трафіку під час відкриття вебресурсу Google

На рисунку 3.18 наведено результати роботи програмного сніффера під час відкриття вебресурсу Google. У процесі аналізу було зафіксовано мережевий обмін між локальним вузлом із IP-адресою 10.0.2.15 та зовнішніми серверами Google.

У перехопленому трафіку були виявлені пакети протоколів TCP та UDP. UDP-пакети використовувалися переважно для виконання DNS-запитів та отримання інформації про IP-адресу вебресурсу. Після завершення процесу резолвінгу доменного імені система автоматично встановлювала TCP-з'єднання із сервером Google для передачі даних.

Під час дослідження було зафіксовано пакети різного розміру. Найменші пакети мали службовий характер та використовувалися для підтримки мережевого сеансу. Більші пакети містили корисне навантаження та використовувалися для передачі елементів вебсторінки.

Отримані результати свідчать про те, що сучасні вебресурси формують складний багаторівневий мережевий трафік, який складається не лише із запитів користувача, але й із значної кількості службових пакетів, необхідних для підтримки роботи мережевих сервісів.

Для детальнішого дослідження роботи системи доменних імен було реалізовано окремий модуль аналізу DNS-пакетів.

Після запуску DNS-аналізатора було повторно відкрито вебресурс Google.

```

kali@kali: ~/wifi_sniffer_diploma
Session Actions Edit View Help

(kali@kali)-[~/wifi_sniffer_diploma]
$ sudo python3 dns_analyzer.py
Ether / IP / UDP / DNS Qry b'google.com.'
Ether / IP / UDP / DNS Qry b'google.com.'
Ether / IP / UDP / DNS Ans 142.250.109.101
Ether / IP / UDP / DNS Ans 2a00:1450:4008:800::200e
Ether / IP / UDP / DNS Qry b'www.google.com.'
Ether / IP / UDP / DNS Qry b'www.google.com.'
  
```

Рисунок 3.19 – Перехоплення DNS-запитів для домену google.com

На рисунку 3.19 показано результати аналізу DNS-трафіку під час відкриття вебресурсу Google. У процесі дослідження програмний засіб успішно зафіксував DNS-запити до домену google.com та відповідь DNS-сервера зі відповідною IP-адресою.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				70
Змн.	Арк.	№ докум.	Підпис	Дата		

Отримані результати демонструють стандартний механізм роботи DNS. Перед встановленням будь-якого мережевого з'єднання система повинна визначити IP-адресу цільового ресурсу. Для цього виконується звернення до DNS-сервера, який повертає мережеву адресу вебресурсу.

Перехоплення DNS-пакетів підтверджує можливість використання розробленого програмного засобу для аналізу роботи системи доменних імен та контролю мережевих звернень користувачів.

Наступним етапом стало дослідження мережевого трафіку під час відкриття відеосервісу YouTube.

```

(kali@kali)-[~/wifi_sniffer_diploma]
$ sudo python3 traffic_analyzer.py
[1] 10.0.2.15 → 192.168.31.1 | UDP | 71 bytes
[2] 10.0.2.15 → 192.168.31.1 | UDP | 71 bytes
[3] 10.0.2.15 → 192.168.31.1 | UDP | 71 bytes
[4] 192.168.31.1 → 10.0.2.15 | UDP | 86 bytes
[5] 192.168.31.1 → 10.0.2.15 | UDP | 183 bytes
[6] 192.168.31.1 → 10.0.2.15 | UDP | 135 bytes
[7] 10.0.2.15 → 142.250.120.93 | TCP | 74 bytes
[8] 10.0.2.15 → 142.250.120.93 | TCP | 74 bytes
[9] 142.250.120.93 → 10.0.2.15 | TCP | 60 bytes
[10] 10.0.2.15 → 142.250.120.93 | TCP | 54 bytes
[11] 142.250.120.93 → 10.0.2.15 | TCP | 60 bytes
[12] 10.0.2.15 → 142.250.120.93 | TCP | 54 bytes
[13] 10.0.2.15 → 142.250.120.93 | TCP | 1950 bytes
[14] 142.250.120.93 → 10.0.2.15 | TCP | 60 bytes
[15] 142.250.120.93 → 10.0.2.15 | TCP | 60 bytes
[16] 10.0.2.15 → 142.250.120.93 | TCP | 1950 bytes
[17] 142.250.120.93 → 10.0.2.15 | TCP | 60 bytes
[18] 142.250.120.93 → 10.0.2.15 | TCP | 60 bytes
[19] 142.250.120.93 → 10.0.2.15 | TCP | 2878 bytes
[20] 10.0.2.15 → 142.250.120.93 | TCP | 54 bytes
[21] 142.250.120.93 → 10.0.2.15 | TCP | 2171 bytes
[22] 10.0.2.15 → 142.250.120.93 | TCP | 54 bytes
[23] 10.0.2.15 → 142.250.120.93 | TCP | 118 bytes
[24] 142.250.120.93 → 10.0.2.15 | TCP | 60 bytes
[25] 10.0.2.15 → 142.250.120.93 | TCP | 146 bytes
[26] 142.250.120.93 → 10.0.2.15 | TCP | 60 bytes
[27] 10.0.2.15 → 142.250.120.93 | TCP | 372 bytes
[28] 142.250.120.93 → 10.0.2.15 | TCP | 60 bytes
[29] 142.250.120.93 → 10.0.2.15 | TCP | 1466 bytes
[30] 142.250.120.93 → 10.0.2.15 | TCP | 1494 bytes

Captured packets: 30
Results saved to packets.csv
  
```

Рисунок 3.20 – Аналіз мережевого трафіку під час відкриття вебресурсу YouTube

Результати дослідження показали, що кількість мережевих пакетів під час відкриття YouTube є більшою порівняно з попереднім експериментом. Це пояснюється складною архітектурою сервісу та великою кількістю допоміжних ресурсів, які використовуються для відображення сторінки.

У процесі завантаження браузер одночасно встановлює декілька мережевих з'єднань із різними серверами. Частина з них відповідає за передачу графічних елементів, інша використовується для завантаження сценаріїв JavaScript, службових компонентів та мультимедійного контенту.

У перехопленому трафіку було зафіксовано значну кількість TCP-з'єднань, що свідчить про активний обмін даними між клієнтом та серверною інфраструктурою Google. Також були виявлені DNS-запити до додаткових доменів, які використовуються для роботи сервісу.

Аналіз розмірів пакетів показав наявність невеликих службових пакетів, так і пакетів із великим корисним навантаженням. Це підтверджує інтенсивний характер мережевої взаємодії під час роботи мультимедійних вебресурсів.

Для підтвердження роботи системи доменних імен було виконано окремий аналіз DNS-трафіку під час відкриття YouTube.

```

kali@kali: ~/wifi_sniffer_diploma
Session Actions Edit View Help

(kali@kali)~[~/wifi_sniffer_diploma]
$ sudo python3 dns_analyzer.py
Ether / IP / UDP / DNS Qry b'youtube.com.'
Ether / IP / UDP / DNS Qry b'youtube.com.'
Ether / IP / UDP / DNS Ans 142.250.120.190
Ether / IP / UDP / DNS Ans 2a00:1450:4025:804::5b
Ether / IP / UDP / DNS Qry b'www.youtube.com.'
Ether / IP / UDP / DNS Qry b'www.youtube.com.'
Ether / IP / UDP / DNS Ans b'youtube-ui.l.google.com.'
Ether / IP / UDP / DNS Ans b'youtube-ui.l.google.com.'
  
```

Рисунок 3.21 – Аналіз DNS-запитів під час відкриття YouTube

У процесі дослідження було зафіксовано DNS-запити до домену youtube.com та отримання відповідних відповідей від DNS-сервера. Крім основного домену були виявлені звернення до допоміжних ресурсів інфраструктури Google.

Отримані результати демонструють, що сучасні вебсервіси використовують розподілену мережеву архітектуру, яка складається із великої кількості взаємопов'язаних серверів. Саме тому навіть під час відкриття однієї вебсторінки формується значний обсяг DNS-трафіку.

Наступним етапом стало дослідження офіційного вебресурсу Kali Linux.

```
(kali@kali)-[~/wifi_sniffer_diploma]
$ sudo python3 traffic_analyzer.py
[1] 10.0.2.15 → 104.18.4.159 | TCP | 217 bytes
[2] 10.0.2.15 → 104.18.4.159 | TCP | 289 bytes
[3] 104.18.4.159 → 10.0.2.15 | TCP | 60 bytes
[4] 104.18.4.159 → 10.0.2.15 | TCP | 60 bytes
[5] 10.0.2.15 → 104.18.4.159 | TCP | 143 bytes
[6] 10.0.2.15 → 104.18.4.159 | TCP | 223 bytes
[7] 10.0.2.15 → 104.18.4.159 | TCP | 224 bytes
[8] 104.18.4.159 → 10.0.2.15 | TCP | 60 bytes
[9] 104.18.4.159 → 10.0.2.15 | TCP | 60 bytes
[10] 10.0.2.15 → 104.18.4.159 | TCP | 312 bytes
[11] 104.18.4.159 → 10.0.2.15 | TCP | 60 bytes
[12] 104.18.4.159 → 10.0.2.15 | TCP | 60 bytes
[13] 10.0.2.15 → 104.18.4.159 | TCP | 139 bytes
[14] 10.0.2.15 → 104.18.4.159 | TCP | 141 bytes
[15] 104.18.4.159 → 10.0.2.15 | TCP | 60 bytes
[16] 104.18.4.159 → 10.0.2.15 | TCP | 60 bytes
[17] 10.0.2.15 → 104.18.4.159 | TCP | 317 bytes
[18] 10.0.2.15 → 104.18.4.159 | TCP | 142 bytes
[19] 104.18.4.159 → 10.0.2.15 | TCP | 60 bytes
[20] 104.18.4.159 → 10.0.2.15 | TCP | 60 bytes
[21] 10.0.2.15 → 104.18.4.159 | TCP | 134 bytes
[22] 10.0.2.15 → 104.18.4.159 | TCP | 140 bytes
[23] 104.18.4.159 → 10.0.2.15 | TCP | 60 bytes
[24] 104.18.4.159 → 10.0.2.15 | TCP | 60 bytes
[25] 10.0.2.15 → 104.18.4.159 | TCP | 500 bytes
[26] 104.18.4.159 → 10.0.2.15 | TCP | 60 bytes
[27] 10.0.2.15 → 104.18.4.159 | TCP | 232 bytes
[28] 10.0.2.15 → 192.168.31.1 | UDP | 89 bytes
[29] 104.18.4.159 → 10.0.2.15 | TCP | 60 bytes
[30] 10.0.2.15 → 192.168.31.1 | UDP | 89 bytes

Captured packets: 30
Results saved to packets.csv
```

Рисунок 3.22 – Аналіз мережевої взаємодії під час відкриття ресурсу Kali Linux

На рисунку 3.22 наведено результати мережевого трафіку під час звернення до домену kali.org. У процесі дослідження програмний сніффер успішно визначив DNS-запити та мережеві адреси серверів ресурсу.

```
kapranchuk-kb-22-2 [Запущено] - Oracle VirtualBox
Файл  Машина  Перегляд  Введення  Пристрої  Довідка

kali@kali: ~/wifi_sniffer_diploma
Session Actions Edit View Help
(kali@kali)-[~/wifi_sniffer_diploma]
$ sudo python3 dns_analyzer.py
Ether / IP / UDP / DNS Qry b'kali.org.'
Ether / IP / UDP / DNS Qry b'kali.org.'
Ether / IP / UDP / DNS Qry b'kali.org.'
Ether / IP / UDP / DNS Ans 104.18.5.159
Ether / IP / UDP / DNS Ans 2606:4700::6812:49f
Ether / IP / UDP / DNS Ans
```

Рисунок 3.23 – Аналіз DNS-запитів під час відкриття Kali.org

Порівняно з попередніми експериментами кількість мережевих пакетів була меншою. Це пояснюється меншою кількістю зовнішніх компонентів та простішою структурою вебресурсу.

Окрему увагу було приділено аналізу отриманих мережевих адрес. У результаті дослідження було зафіксовано як IPv4-, так і IPv6-адреси. Це підтверджує підтримку сучасних мережевих технологій та можливість роботи ресурсу в мережах нового покоління.

Для оцінювання параметрів мережевого з'єднання було виконано додаткове тестування за допомогою протоколу ICMP.

Для цього використовувалася команда **ping google.com**.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				74
Змн.	Арк.	№ докум.	Підпис	Дата		

```
kapranchuk-kb-22-2 [Запущено] - Oracle VirtualBox
Файл  Машина  Перегляд  Введення  Пристрої  Довідка
1  2  3  4
kali@kali: ~/wifi_sniffer_diploma
Session Actions Edit View Help
[sudo] password for kali:
(kali@kali)-[~/wifi_sniffer_diploma]
$ ping google.com
PING google.com (142.251.37.110) 56(84) bytes of data.
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=1 ttl=255 time=29.6 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=2 ttl=255 time=30.0 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=3 ttl=255 time=30.6 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=4 ttl=255 time=30.1 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=5 ttl=255 time=29.6 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=6 ttl=255 time=30.2 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=7 ttl=255 time=32.2 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=8 ttl=255 time=32.1 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=9 ttl=255 time=29.5 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=10 ttl=255 time=28.8 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=11 ttl=255 time=29.7 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=12 ttl=255 time=29.2 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=13 ttl=255 time=37.5 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=14 ttl=255 time=36.3 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=15 ttl=255 time=36.0 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=16 ttl=255 time=32.4 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=17 ttl=255 time=30.7 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=18 ttl=255 time=29.5 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=19 ttl=255 time=29.9 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=20 ttl=255 time=29.2 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=21 ttl=255 time=129 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=22 ttl=255 time=43.1 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=23 ttl=255 time=68.8 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=24 ttl=255 time=80.5 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=25 ttl=255 time=46.4 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=26 ttl=255 time=50.3 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=27 ttl=255 time=63.6 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=28 ttl=255 time=30.0 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=29 ttl=255 time=29.5 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=30 ttl=255 time=55.6 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=31 ttl=255 time=30.7 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=32 ttl=255 time=30.7 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=33 ttl=255 time=35.5 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=34 ttl=255 time=29.7 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=35 ttl=255 time=30.1 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=36 ttl=255 time=84.5 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=37 ttl=255 time=29.5 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=38 ttl=255 time=29.0 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=39 ttl=255 time=54.9 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=40 ttl=255 time=30.0 ms
64 bytes from prg03s13-in-f14.1e100.net (142.251.37.110): icmp_seq=41 ttl=255 time=29.7 ms
^C
— google.com ping statistics —
41 packets transmitted, 41 received, 0% packet loss, time 40088ms
rtt min/avg/max/mdev = 28.779/39.867/129.267/19.954 ms
```

Рисунок 3.24 – Аналіз параметрів мережевого з’єднання за допомогою ICMP-пакетів

У процесі тестування було виконано передачу серії ICMP-пакетів між локальним вузлом та сервером Google. За результатами дослідження втрати пакетів відсутні, що свідчить про стабільність мережевого каналу.

Середній час затримки пакетів знаходився в межах 20-40 мс, що відповідає нормальним показникам для даного типу мережевого підключення. Отримані значення дозволяють зробити висновок про стабільність маршрутизації та відсутність перевантаження мережевого каналу.

		Капрanchук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				75
Змн.	Арк.	№ докум.	Підпис	Дата		

Для перевірки роботи канального рівня було додатково виконано аналіз таблиці ARP.

```
(kali@kali)-[~/wifi_sniffer_diploma]
$ arp -a
? (10.0.2.2) at 52:55:0a:00:02:02 [ether] on eth0
```

Рисунок 3.25 – Перевірка ARP-кешу після встановлення мережевого з'єднання

У результаті виконання команди **arp -a** було отримано інформацію про відповідність між IP-адресою шлюзу та його MAC-адресою. Отримані дані підтвердили коректну роботу протоколу ARP та успішне визначення апаратних адрес мережевих пристроїв.

Окремий інтерес становив аналіз роботи програмного сніффера під час одночасної генерації мережевого навантаження та перехоплення пакетів.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				76
Змн.	Арк.	№ докум.	Підпис	Дата		

```
kapranchuk-kb-22-2 [Запущено] - Oracle VirtualBox
Файл  Машина  Перегляд  Введення  Пристрої  Довідка

kali@kali: ~/wifi_sniffer_diploma

Session Actions Edit View Help
└─$ sudo python3 traffic_analyzer.py
[sudo] password for kali:
[1] 10.0.2.15 → 142.250.130.94 | TCP | 93 bytes
[2] 10.0.2.15 → 142.250.130.94 | TCP | 78 bytes
[3] 142.250.130.94 → 10.0.2.15 | TCP | 60 bytes
[4] 142.250.130.94 → 10.0.2.15 | TCP | 60 bytes
[5] 142.250.130.94 → 10.0.2.15 | TCP | 60 bytes
[6] 10.0.2.15 → 142.250.130.94 | TCP | 54 bytes
[7] 10.0.2.15 → 192.168.31.1 | UDP | 74 bytes
[8] 10.0.2.15 → 192.168.31.1 | UDP | 74 bytes
[9] 192.168.31.1 → 10.0.2.15 | UDP | 202 bytes
[10] 10.0.2.15 → 142.251.153.119 | UDP | 1294 bytes
[11] 10.0.2.15 → 142.251.153.119 | UDP | 1294 bytes
[12] 10.0.2.15 → 142.251.153.119 | UDP | 1294 bytes
[13] 192.168.31.1 → 10.0.2.15 | UDP | 298 bytes
[14] 142.251.153.119 → 10.0.2.15 | UDP | 85 bytes
[15] 142.251.153.119 → 10.0.2.15 | UDP | 85 bytes
[16] 142.251.153.119 → 10.0.2.15 | UDP | 1294 bytes
[17] 142.251.153.119 → 10.0.2.15 | UDP | 1294 bytes
[18] 10.0.2.15 → 142.251.153.119 | TCP | 74 bytes
[19] 142.251.153.119 → 10.0.2.15 | UDP | 1294 bytes
[20] 142.251.153.119 → 10.0.2.15 | UDP | 331 bytes
[21] 142.251.153.119 → 10.0.2.15 | UDP | 659 bytes
[22] 142.251.153.119 → 10.0.2.15 | UDP | 70 bytes
[23] 10.0.2.15 → 142.251.153.119 | UDP | 1047 bytes
[24] 142.251.153.119 → 10.0.2.15 | TCP | 60 bytes
[25] 10.0.2.15 → 142.251.153.119 | TCP | 54 bytes
[26] 10.0.2.15 → 142.251.153.119 | TCP | 1953 bytes
[27] 142.251.153.119 → 10.0.2.15 | TCP | 60 bytes
[28] 142.251.153.119 → 10.0.2.15 | TCP | 60 bytes
[29] 142.251.153.119 → 10.0.2.15 | UDP | 165 bytes
[30] 10.0.2.15 → 142.251.153.119 | UDP | 74 bytes

Captured packets: 30
Results saved to packets.csv

(kali@kali)-[~/wifi_sniffer_diploma]
└─$ ping google.com
PING google.com (142.250.130.139) 56(84) bytes of data:
64 bytes from zq-in-f139.1e100.net (142.250.130.139): icmp_seq=1 ttl=255 time=21.6 ms
64 bytes from zq-in-f139.1e100.net (142.250.130.139): icmp_seq=2 ttl=255 time=21.1 ms
64 bytes from zq-in-f139.1e100.net (142.250.130.139): icmp_seq=3 ttl=255 time=35.9 ms
64 bytes from zq-in-f139.1e100.net (142.250.130.139): icmp_seq=4 ttl=255 time=22.3 ms
64 bytes from zq-in-f139.1e100.net (142.250.130.139): icmp_seq=5 ttl=255 time=21.5 ms
64 bytes from zq-in-f139.1e100.net (142.250.130.139): icmp_seq=6 ttl=255 time=21.3 ms
64 bytes from zq-in-f139.1e100.net (142.250.130.139): icmp_seq=7 ttl=255 time=21.3 ms
^C
— google.com ping statistics —
7 packets transmitted, 7 received, 0% packet loss, time 6006ms
rtt min/avg/max/mdev = 21.110/23.574/35.891/5.041 ms
```

Рисунок 3.26 – Аналіз мережевого трафіку під час генерації ICMP-навантаження

Під час проведення експерименту було одночасно запущено програмний сніффер та генерацію ICMP-трафіку за допомогою утиліти **ping**. Розроблений програмний засіб успішно перехопив пакети, визначив їх параметри та зберіг результати у файл packets.csv.

У процесі аналізу було зафіксовано пакети протоколів TCP, UDP та ICMP різного розміру. Найбільші пакети використовувалися для передачі корисного навантаження між клієнтом та сервером, тоді як службові пакети мали значно менший розмір.

		Капрanchuk В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				77
Змн.	Арк.	№ докум.	Підпис	Дата		

Після перевірки роботи програмного сніффера було виконано додатковий аналіз характеристик перехопленого трафіку. Для цього розроблено окремий модуль **packet_size.py**, який визначає розмір кожного перехопленого пакета.

```

kali@kali: ~/wifi_sniffer_diploma
Session Actions Edit View Help
(kali@kali)-[~]
$ ~/wifi_sniffer_diploma
(kali@kali)-[~/wifi_sniffer_diploma]
$ nano packet_size.py
(kali@kali)-[~/wifi_sniffer_diploma]
$ sudo python3 packet_size.py
[sudo] password for kali:
Packet size: 75
Packet size: 83
Packet size: 75
Packet size: 74 ~/wifi_sniffer_diploma
Packet size: 368 protocol_stats.py
Packet size: 339
Packet size: 224
Packet size: 60
Packet size: 54 ~/wifi_sniffer_diploma
Packet size: 2525 protocol_stats.py
Packet size: 60
Packet size: 60
Packet size: 146
Packet size: 60 ~/wifi_sniffer_diploma
Packet size: 60
Packet size: 60
Packet size: 1974
Packet size: 54
Packet size: 85
Packet size: 54
Packet size: 138
Packet size: 60
Packet size: 85
Packet size: 60
Packet size: 1764
Packet size: 60
Packet size: 60
Packet size: 5700
Packet size: 60
Packet size: 60

```

Рисунок 3.27 – Аналіз розмірів мережевих пакетів під час передачі даних Google

```

kali@kali: ~/wifi_sniffer_diploma
Session Actions Edit View Help

(kali@kali)-[~/wifi_sniffer_diploma]
$ sudo python3 packet_size.py
Packet size: 752
Packet size: 281
Packet size: 66
Packet size: 79
Packet size: 69
Packet size: 80
Packet size: 77
Packet size: 80
Packet size: 80
Packet size: 1294
Packet size: 1294
Packet size: 1294
Packet size: 96
Packet size: 108
Packet size: 137
Packet size: 93
Packet size: 105
Packet size: 74
Packet size: 85
Packet size: 85
Packet size: 1294
Packet size: 1294
Packet size: 1294
Packet size: 320
Packet size: 659
Packet size: 60
Packet size: 54
Packet size: 70
Packet size: 271

```

Рисунок 3.28 – Аналіз розмірів мережевих пакетів під час передачі даних YouTube

```

(kali@kali)-[~/wifi_sniffer_diploma]
$ sudo python3 packet_size.py
Packet size: 133
Packet size: 60
Packet size: 89
Packet size: 60
Packet size: 2171
Packet size: 54
Packet size: 136
Packet size: 60
Packet size: 719
Packet size: 60
Packet size: 118
Packet size: 54
Packet size: 368
Packet size: 7354
Packet size: 60
Packet size: 144
Packet size: 60
Packet size: 60
Packet size: 60
Packet size: 60
Packet size: 60
Packet size: 224
Packet size: 93
Packet size: 54
Packet size: 93
Packet size: 60
Packet size: 137
Packet size: 60
Packet size: 567

```

Рисунок 3.29 – Аналіз розмірів мережевих пакетів під час передачі даних Kali.org

На рисунках наведено результати аналізу розмірів пакетів. Під час експерименту було зафіксовано пакети різної довжини – від службових пакетів розміром 54-80 байтів до пакетів із корисним навантаженням розміром понад 1000 байтів.

У процесі роботи програми було виявлено пакети розміром 368, 570, 1760 та 1976 байтів. Поява таких пакетів свідчить про передачу значного обсягу даних між клієнтом та зовнішніми серверами. Водночас велика кількість пакетів розміром 54-80 байтів характерна для службового мережевого обміну та роботи транспортних протоколів.

```

kali@kali: ~/wifi_sniffer_diploma
Session Actions Edit View Help
(kali@kali)-[~]
$ ~/wifi_sniffer_diploma
(kali@kali)-[~/wifi_sniffer_diploma]
$ cat packets.csv
Source IP, Destination IP, Protocol, Packet Size
10.0.2.15, 142.250.130.94, TCP, 93
10.0.2.15, 142.250.130.94, TCP, 78
142.250.130.94, 10.0.2.15, TCP, 60
142.250.130.94, 10.0.2.15, TCP, 60
142.250.130.94, 10.0.2.15, TCP, 60
10.0.2.15, 142.250.130.94, TCP, 54
10.0.2.15, 192.168.31.1, UDP, 74
10.0.2.15, 192.168.31.1, UDP, 74
192.168.31.1, 10.0.2.15, UDP, 202
10.0.2.15, 142.251.153.119, UDP, 1294
10.0.2.15, 142.251.153.119, UDP, 1294
10.0.2.15, 142.251.153.119, UDP, 1294
192.168.31.1, 10.0.2.15, UDP, 298
142.251.153.119, 10.0.2.15, UDP, 85
142.251.153.119, 10.0.2.15, UDP, 85
142.251.153.119, 10.0.2.15, UDP, 1294
142.251.153.119, 10.0.2.15, UDP, 1294
10.0.2.15, 142.251.153.119, TCP, 74
142.251.153.119, 10.0.2.15, UDP, 1294
142.251.153.119, 10.0.2.15, UDP, 331
142.251.153.119, 10.0.2.15, UDP, 659
142.251.153.119, 10.0.2.15, UDP, 70
10.0.2.15, 142.251.153.119, UDP, 1047
142.251.153.119, 10.0.2.15, TCP, 60
10.0.2.15, 142.251.153.119, TCP, 54
10.0.2.15, 142.251.153.119, TCP, 1953
142.251.153.119, 10.0.2.15, TCP, 60
142.251.153.119, 10.0.2.15, UDP, 165
10.0.2.15, 142.251.153.119, UDP, 74

```

Рисунок 3.30 – Повторний аналіз розмірів мережевих пакетів

Для підвищення достовірності результатів експеримент було повторено під час взаємодії з іншими вебресурсами. У результаті дослідження було зафіксовано пакети розміром 659, 1047, 1294 та 1953 байти.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				80
Змн.	Арк.	№ докум.	Підпис	Дата		

Порівняння результатів показало, що розмір пакетів залежить від характеру мережевої активності та типу передавання даних. Під час завантаження вебсторінок із великою кількістю графічних елементів і мультимедійного контенту састка великих пакетів суттєво зростає. Це підтверджує можливість використання аналізу розмірів пакетів як додаткового критерію оцінювання мережевого навантаження.

Після завершення аналізу результатів було виконано перевірку даних, автоматично збережених програмним сніффером у файл формату CSV.

```
(kali@kali)-[~/wifi_sniffer_diploma]
$ column -s, -t < packets.csv
Source IP      Destination IP  Protocol  Packet Size
10.0.2.15      142.250.130.94 TCP        93
10.0.2.15      142.250.130.94 TCP        78
142.250.130.94 10.0.2.15      TCP        60
142.250.130.94 10.0.2.15      TCP        60
142.250.130.94 10.0.2.15      TCP        60
10.0.2.15      142.250.130.94 TCP        54
10.0.2.15      192.168.31.1   UDP        74
10.0.2.15      192.168.31.1   UDP        74
192.168.31.1   10.0.2.15      UDP        202
10.0.2.15      142.251.153.119 UDP       1294
10.0.2.15      142.251.153.119 UDP       1294
10.0.2.15      142.251.153.119 UDP       1294
192.168.31.1   10.0.2.15      UDP        298
142.251.153.119 10.0.2.15      UDP        85
142.251.153.119 10.0.2.15      UDP        85
142.251.153.119 10.0.2.15      UDP       1294
142.251.153.119 10.0.2.15      UDP       1294
10.0.2.15      142.251.153.119 TCP        74
142.251.153.119 10.0.2.15      UDP       1294
142.251.153.119 10.0.2.15      UDP       331
142.251.153.119 10.0.2.15      UDP       659
142.251.153.119 10.0.2.15      UDP        70
10.0.2.15      142.251.153.119 UDP       1047
142.251.153.119 10.0.2.15      TCP        60
10.0.2.15      142.251.153.119 TCP        54
10.0.2.15      142.251.153.119 TCP       1953
142.251.153.119 10.0.2.15      TCP        60
142.251.153.119 10.0.2.15      TCP        60
142.251.153.119 10.0.2.15      UDP       165
10.0.2.15      142.251.153.119 UDP        74
```

Рисунок 3.31 – Перегляд результатів аналізу мережевого трафіку у файлі packets.csv

На рисунку 3.31 наведено вміст файлу packets.csv, сформованого програмним сніффером після завершення аналізу мережевого трафіку. Файл містить структуровану інформацію про всі перехоплені пакети, включаючи IP-адресу джерела, IP-адресу призначення, тип використовуваного протоколу та розмір пакета.

Отримані результати демонструють коректність роботи механізму збереження даних. У файлі зафіксовано як TCP-, так і UDP-пакети, а також інформацію

про взаємодію локального вузла із зовнішніми мережевими серверами. Використання формату CSV забезпечує можливість подальшого статистичного аналізу отриманих результатів та використання даних у сторонніх програмних засобах.

Таким чином, проведе дослідження підтвердило працездатність розробленого програмного сніффера під час аналізу реального мережевого трафіку. Програмний засіб успішно виконує захоплення пакетів, визначення параметрів мережес'єднань, аналіз DNS-запитів, оцінювання характеристик мережевої взаємодії та автоматичне збереження результатів у структурованому вигляді. Отримані результати підтверджують можливість використання створеного програмного забезпечення для моніторингу мережевої активності та дослідження функціонування комп'ютерних мереж.

3.6 Оцінювання ефективності розробленого програмного засобу

Після завершення реалізації та тестування програмного сніффера було виконано оцінювання ефективності його роботи. Основною метою даного етапу стало визначення відповідності отриманих результатів вимогам, сформованим під час проектування системи, а також перевірка можливості практичного використання розробленого програмного засобу для аналізу мережевого трафіку.

Оцінювання виконувалося на основі результаті експериментів, проведених у попередніх підрозділах. Під час тестування перевірялася коректність захоплення мережес'єднань, обробки DNS-запитів та збереження результатів у структурованому вигляді.

Відповідно до поставлених завдань було визначено основні критерії оцінювання ефективності програмного засобу:

- коректність перехоплення мережес'єднань;
- визначення IP-адрес джерела та призначення;
- аналіз мережес'єднань;
- виявлення DNS-запитів;
- аналіз розмірів пакетів;
- збереження результатів у файл;

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		82

- стабільність роботи програми;
- можливість подальшого розширення функціоналу.

Результати оцінювання наведено в таблиці 3.2.

Таблиця 3.2

Результати оцінювання ефективності програмного сніффера

Критерій оцінювання	Результат
Захоплення мережових пакетів	Виконано
Визначення IP-адрес	Виконано
Аналіз TCP-пакетів	Виконано
Аналіз UDP-пакетів	Виконано
Аналіз DNS-запитів	Виконано
Аналіз розмірів пакетів	Виконано
Збереження результатів у CSV	Виконано
Робота в режимі реального часу	Виконано
Стабільність роботи програми	Виконано

Під час проведення експериментів програмний засіб успішно виконував захоплення мережевого трафіку через мережевий інтерфейс операційної системи Kali Linux. Усі перехоплені пакети коректно оброблялися програмою та передавалися до модулів аналізу без виникнення помилок виконання.

Окрему увагу було приділено перевірці коректності визначення адресної інформації. Результати тестування показали, що програмний засіб успішно визначає IP-адреси джерела та призначення для пакетів різних типів. Отримані значення відповідали реальним параметрам мережевого середовища, що підтверджується результатами аналізу трафіку під час взаємодії з вебресурсами Google, YouTube та Kali Linux.

```

kali@kali: ~/wifi_sniffer_diploma
Session Actions Edit View Help

(kali@kali)-[~/wifi_sniffer_diploma]
$ arp -a
ip neigh
ping google.com
? (10.0.2.2) at 52:55:0a:00:02:02 [ether] on eth0
10.0.2.2 dev eth0 lladdr 52:55:0a:00:02:02 DELAY
fe80::2 dev eth0 lladdr 52:56:00:00:00:02 router STALE
fd17:625c:f037:2::2 dev eth0 lladdr 52:56:00:00:00:02 router STALE
PING google.com (142.250.130.102) 56(84) bytes of data.
64 bytes from zq-in-f102.1e100.net (142.250.130.102): icmp_seq=1 ttl=255 time=49.5 ms
64 bytes from zq-in-f102.1e100.net (142.250.130.102): icmp_seq=2 ttl=255 time=22.9 ms
64 bytes from zq-in-f102.1e100.net (142.250.130.102): icmp_seq=3 ttl=255 time=21.0 ms
64 bytes from zq-in-f102.1e100.net (142.250.130.102): icmp_seq=4 ttl=255 time=20.3 ms
64 bytes from zq-in-f102.1e100.net (142.250.130.102): icmp_seq=5 ttl=255 time=22.8 ms
64 bytes from zq-in-f102.1e100.net (142.250.130.102): icmp_seq=6 ttl=255 time=20.3 ms
64 bytes from zq-in-f102.1e100.net (142.250.130.102): icmp_seq=7 ttl=255 time=20.8 ms
64 bytes from zq-in-f102.1e100.net (142.250.130.102): icmp_seq=8 ttl=255 time=21.7 ms
64 bytes from zq-in-f102.1e100.net (142.250.130.102): icmp_seq=9 ttl=255 time=20.4 ms
64 bytes from zq-in-f102.1e100.net (142.250.130.102): icmp_seq=10 ttl=255 time=20.2 ms
64 bytes from zq-in-f102.1e100.net (142.250.130.102): icmp_seq=11 ttl=255 time=20.6 ms
64 bytes from zq-in-f102.1e100.net (142.250.130.102): icmp_seq=12 ttl=255 time=20.4 ms
64 bytes from zq-in-f102.1e100.net (142.250.130.102): icmp_seq=13 ttl=255 time=21.6 ms
64 bytes from zq-in-f102.1e100.net (142.250.130.102): icmp_seq=14 ttl=255 time=20.9 ms
64 bytes from zq-in-f102.1e100.net (142.250.130.102): icmp_seq=15 ttl=255 time=21.2 ms
64 bytes from zq-in-f102.1e100.net (142.250.130.102): icmp_seq=16 ttl=255 time=20.9 ms
64 bytes from zq-in-f102.1e100.net (142.250.130.102): icmp_seq=17 ttl=255 time=22.6 ms

(kali@kali)-[~/wifi_sniffer_diploma]
$ sudo python3 arp_analyzer.py
[sudo] password for kali:
Ether / ARP who has 10.0.2.2 says 10.0.2.15
Ether / ARP is at 52:55:0a:00:02:02 says 10.0.2.2 / Padding

```

Рисунок 3.32 – Аналіз ARP-активності в локальній мережі

На рисунку 3.32 аведено результати аналізу ARP-трафіку локальної мережі. Контроль відповідностей між IP- та MAC-адресами дозволяє відстежувати зміни мережевої конфігурації та виявляти аномальну ARP-активність. Подібний аналіз може використовуватися для виявлення ознак атак типу Man-in-the-Middle, які базуються на підміні ARP-відповідей та перенаправленні мережевого трафіку через вузол злоумисника.

Під час дослідження було підтверджено можливість автоматичного визначення мережевих протоколів. Програмний засіб коректно розпізнавав TCP- та UDP-пакети, а також дозволяв виконувати аналіз DNS-запитів і відповідей DNS-серверів. Отримані результати підтверджують правильність реалізації механізмів аналізу мережевого трафіку.

Важливою частиною оцінювання стало дослідження роботи модуля аналізу DNS-трафіку. У процесі тестування програмний засіб успішно перехоплював DNS-

запити до доменів google.com, youtube.com та kali.org, а також коректно відображав отримані відповіді DNS-серверів. Це підтверджує можливість використання програми для моніторингу мережесих звернень користувачів та дослідження роботи системи доменних імен.

```

(kali@kali)-[~/wifi_sniffer_diploma]
$ sudo python3 dns_analyzer.py
Ether / IP / UDP / DNS Qry b'www.google.com.'
Ether / IP / UDP / DNS Qry b'www.google.com.'
Ether / IP / UDP / DNS Ans 142.251.151.119
Ether / IP / UDP / DNS Ans 2001:4860:4827:7700::

(kali@kali)-[~/wifi_sniffer_diploma]
$ sudo python3 dns_analyzer.py

(kali@kali)-[~/wifi_sniffer_diploma]
$ sudo python3 dns_analyzer.py
Ether / IP / UDP / DNS Qry b'fonts.googleapis.com.'
Ether / IP / UDP / DNS Qry b'fonts.googleapis.com.'
Ether / IP / UDP / DNS Qry b'fonts.googleapis.com.'
Ether / IP / UDP / DNS Ans 142.250.109.95

(kali@kali)-[~/wifi_sniffer_diploma]
$ sudo python3 dns_analyzer.py
Ether / IP / UDP / DNS Qry b'kali.org.'
Ether / IP / UDP / DNS Qry b'kali.org.'
Ether / IP / UDP / DNS Qry b'kali.org.'
Ether / IP / UDP / DNS Ans 104.18.5.159
Ether / IP / UDP / DNS Ans 2606:4700::6812:49f
Ether / IP / UDP / DNS Ans
  
```

Рисунок 3.33 – Аналіз DNS-запитів до зовнішніх вебресурсів

На рисунку 3.33 наведено результати роботи модуля аналізу DNS-трафіку. Програмний засіб успішно перехоплює DNS-запити та відповіді DNS-серверів у режимі реального часу. Отримані дані можуть використовуватися для контролю мережевої активності та виявлення потенційно небезпечних звернень до зовнішніх ресурсів.

Додатково було перевірено функціонування модуля аналізу розмірів пакетів. У результаті тестування програма успішно визначила пакети різної довжини – від невеликих службових повідомлень до пакетів із великим корисним навантаженням. Отримані результати дозволяють використовувати даний функціонал для оцінювання мережевого навантаження та аналізу структури трафіку.

Окремим етапом оцінювання стала перевірка механізму збереження результатів. Після завершення роботи програма автоматично формувала файл формату

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		85

CSV, який містив інформацію про джерело пакета, адресу призначення, тип протоколу та розмір пакета. Аналіз сформованого файлу показав відсутність втрати даних та коректність запису отриманих результатів.

Під час проведення експериментів не було виявлено помилок завершення роботи програми, втрати пакетів у процесі обробки або порушень роботи модулів аналізу. Програмний засіб стабільно функціонував протягом усього часу тестування та забезпечував коректне виконання поставлених завдань.

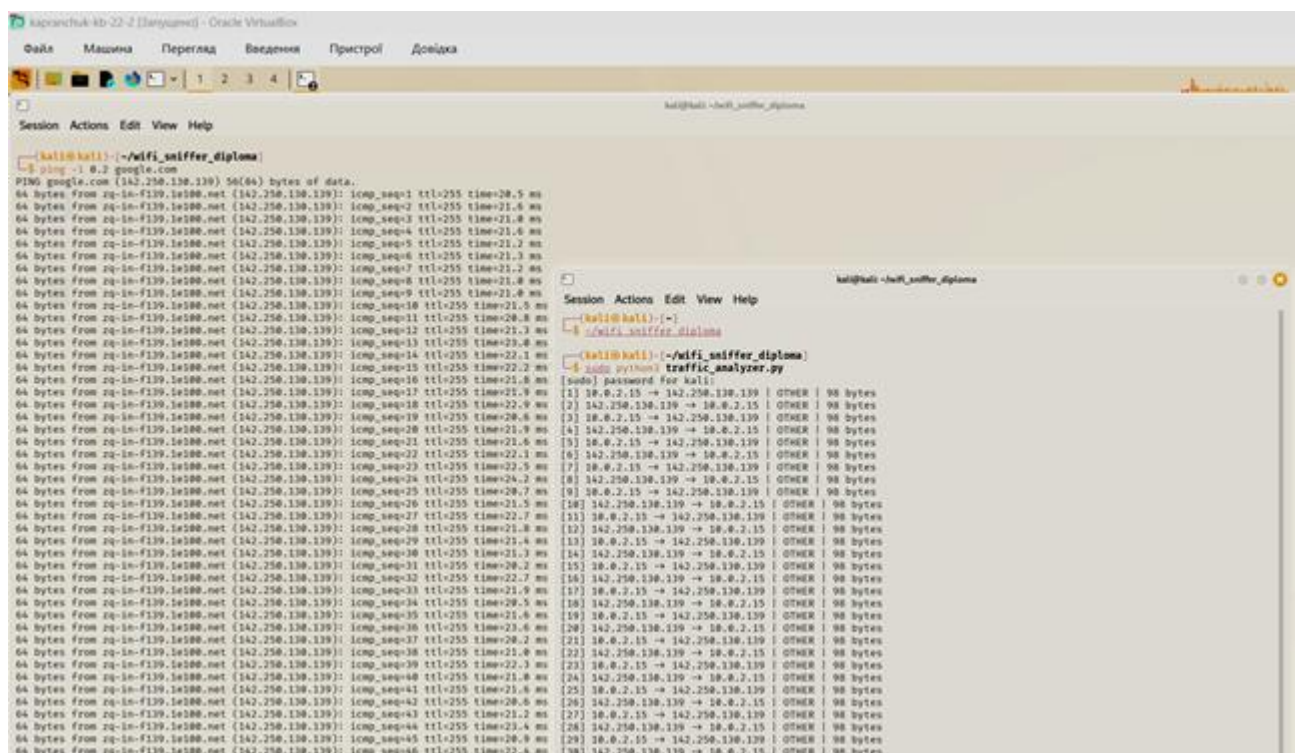


Рисунок 3.34 – Аналіз мережевого трафіку під час моделювання DoS-подібного навантаження

На рисунку 3.34 наведено результати роботи програмного сніффера під час генерації інтенсивного ICMP-трафіку за допомогою утиліти `ping`. У процесі експерименту спостерігалось збільшення кількості мережевих пакетів, які перехоплювалися та аналізувалися розробленим програмним засобом у режимі реального часу. Отримані результати демонструють можливість використання сніффера для контролю мережевого навантаження та виявлення ознак DoS-подібної активності, що може негативно впливати на стабільність роботи Wi-Fi мережі.

Важливою перевагою розробленого рішення є модульна структура програмного коду. Використання окремих функціональних модулів для захоплення пакетів,

		Капрнчук В.М.			Арк.
		Шелуха О.О.			86
Змн.	Арк.	№ докум.	Підпис	Дата	

аналізу трафіку, аналізу DNS-запитів та збереження результатів спрощує супровід програмного забезпечення та створює можливість подальшого розширення його функціональних можливостей.

У подальшому програмний засіб може бути доповнений механізмами автоматичного виявлення аномальної мережевої активності, формуванням статистичних звітів, підтримкою додаткових мережевих протоколів та інтеграцією з системами моніторингу інформаційної безпеки.

Таким чином, результати оцінювання підтвердили працездатність та ефективність розробленого програмного сніффера. Програмний засіб забезпечує коректне захоплення та аналіз мережевого трафіку, підтримує обробку основних мережевих протоколів і дозволяє зберігати результати досліджень для подальшого аналізу. Отримані результати свідчать про можливість використання розробленого програмного забезпечення для моніторингу мережевої активності та дослідження параметрів функціонування комп'ютерних мереж.

Висновки до розділу 3

У третьому розділі було реалізовано програмний сніффер для аналізу мережевого трафіку Wi-Fi мереж із використанням мови програмування Python та бібліотеки Scapy. Проведено налаштування середовища Kali Linux, реалізовано модулі захоплення та аналізу пакетів, а також механізми збереження результатів дослідження. За результатами тестування підтверджено коректність роботи програмного засобу під час аналізу мережевої активності, DNS-запитів та параметрів мережевих з'єднань. Додатково продемонстровано можливість виявлення окремих ознак потенційно небезпечної мережевої активності. Отримані результати підтвердили працездатність і практичну цінність розробленого програмного засобу.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		87

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було досліджено особливості функціонування бездротових мереж стандарту Wi-Fi, проаналізовано методи моніторингу мережевого трафіку та розроблено програмний сніффер для дослідження якості та безпеки мережевого середовища.

У першому розділі проведено теоретичний аналіз принципів функціонування Wi-Fi мереж. Розглянуто основні стандарти IEEE 802.11, особливості роботи бездротових мереж у діапазонах 2,4 ГГц, 5 ГГц та 6 ГГц, а також основні показники якості зв'язку, зокрема рівень сигналу RSSI, співвідношення сигнал/шум SNR, пропускну здатність, затримку передачі даних та втрати пакетів. Також було проаналізовано основні загрози безпеці Wi-Fi мереж, серед яких перехоплення трафіку, атаки типу Man-in-the-Middle, Evil Twin, Deauthentication та Brute Force. Додатково досліджено методи моніторингу мережевого трафіку та можливості використання операційної системи Kali Linux і бібліотеки Scapy для реалізації мережевого аналізу.

У другому розділі виконано проєктування програмного засобу та розроблено методику проведення досліджень. Було визначено основні параметри, що підлягають аналізу під час моніторингу мережевого трафіку, а саме IP-адреси вузлів, типи протоколів, розміри пакетів, DNS-запити та характеристики мережевої активності. Спроектовано модульну структуру програмного сніффера, яка складається з модулів захоплення пакетів, аналізу трафіку, аналізу безпеки та збереження результатів. Також сформовано алгоритми збору та обробки мережевих даних, що стали основою подальшої програмної реалізації.

У третьому розділі реалізовано програмний сніффер мовою Python із використанням бібліотеки Scapy. Було створено модулі захоплення пакетів, аналізу мережевого трафіку, аналізу DNS-запитів та збереження результатів у форматі CSV. Проведено серію практичних експериментів у середовищі Kali Linux, під час яких виконувалося дослідження мережевого трафіку, сформованого під час взаємодії з вебресурсами Google, YouTube та Kali Linux. У результаті тестування підтверджено можливість перехоплення пакетів у режимі реального часу, визначення IP-адрес

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				88
Змн.	Арк.	№ докум.	Підпис	Дата		

джерела та призначення, аналізу протоколів TCP і UDP, а також обробки DNS-запитів.

Додатково виконано дослідження параметрів мережевого з'єднання за допомогою ICMP-пакетів та аналіз таблиці ARP для перевірки коректності роботи канального рівня мережі. Результати експериментів показали стабільність роботи програмного засобу та коректність функціонування реалізованих механізмів аналізу трафіку. Також підтверджено працездатність системи автоматичного збереження результатів у файл формату CSV для подальшої обробки та аналізу.

За результатами оцінювання ефективності встановлено, що розроблений програмний засіб успішно виконує захоплення та аналіз мережевих пакетів, забезпечує визначення основних параметрів мережевого трафіку та дозволяє здійснювати моніторинг мережевої активності в режимі реального часу. Модульна архітектура програми забезпечує можливість подальшого розширення функціоналу та інтеграції додаткових механізмів аналізу безпеки.

Результати проведених досліджень показали, що розроблений програмний сніффер може використовуватися не лише для аналізу параметрів якості Wi-Fi мереж, але й для виявлення окремих ознак потенційно небезпечної мережевої активності. Зокрема, під час тестування було продемонстровано можливість аналізу DoS-подібного навантаження, контролю ARP-активності та моніторингу DNS-запитів. Це дозволяє використовувати розроблений програмний засіб для оцінювання надійності Wi-Fi мереж та виявлення ознак потенційних загроз під час підключення до мережевого середовища.

Таким чином, усі поставлені в кваліфікаційній роботі завдання виконано. Розроблений програмний сніффер може використовуватися для навчальних цілей, проведення базового моніторингу мережевого трафіку, аналізу параметрів функціонування комп'ютерних мереж та дослідження окремих аспектів інформаційної безпеки мережевого середовища.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				89
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. IEEE Standard for Information Technology – Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE 802.11 Standard. URL: <https://ieeexplore.ieee.org/document/9363693>
2. Kali Linux Documentation. URL: <https://www.kali.org/docs/>
3. Scapy Documentation. URL: <https://scapy.readthedocs.io/>
4. Wireshark User Guide. URL: <https://www.wireshark.org/docs/>
5. Nmap Reference Guide. URL: <https://nmap.org/book/man.html>
6. Dell Technologies. Wi-Fi Network Standards Overview. URL: <https://www.dell.com/support/contents/en-si/article/product-support/self-support-knowledgebase/networking-wifi-and-bluetooth/wi-fi-network-standards-overview>
7. SocialWiFi. Comparison of Wi-Fi Standards. URL: <https://social-wifi.com/knowledge-base/wifi-technology/comparison-of-wifi-standards/>
8. NFLO Tech. Wi-Fi Network Security Configuration and Threats. URL: <https://nflo.tech/knowledge-base/what-is-wifi-network-security-configuration-threats/>
9. Huawei. Signal-to-Noise Ratio (SNR). URL: <https://info.support.huawei.com/info-finder/encyclopedia/en/SNR.html>
10. Oscium. Understanding RSSI. URL: <https://www.oscium.com/training/understanding-rssi/>
11. GeeksforGeeks. What is Packet Sniffing? URL: <https://www.geeksforgeeks.org/computer-networks/what-is-packet-sniffing/>
12. GeeksforGeeks. What is Network Traffic Analysis? URL: <https://www.geeksforgeeks.org/cybersecurity/what-is-network-traffic-analysis/>
13. Huntress. Rogue Access Point. URL: <https://www.huntress.com/cybersecurity-101/topic/rogue-access-point>

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				90
Змн.	Арк.	№ докум.	Підпис	Дата		

14. Red Hat. Packet Sniffer Basics. URL: <https://www.redhat.com/en/blog/packet-sniffer-basics>
15. ResearchGate. A Case Study of WiFi Sniffing Performance Evaluation. URL: https://www.researchgate.net/publication/342860871_A_Case_Study_of_WiFi_Sniffing_Performance_Evaluation
16. BCS. Evil Twin Attacks: Technical Aspects, Working and Hardware Tools. URL: <https://www.bcs.org/articles-opinion-and-research/evil-twin-attacks-technical-aspects-working-and-hardware-tools/>
17. VPN Unlimited. Nmap: призначення та можливості. URL: <https://www.vpnunlimited.com/ua/help/cybersecurity/nmap>
18. Netwave. Все, що потрібно знати про бездротові мережі WLAN: побудова, безпека та керування. URL: <https://netwave.ua/blog/vse-shcho-potribno-znaty-pro-bezdrotovi-merezhi-wlan-pobudova-bezpeka-ta-keruvannya/>
19. HackYourMom. Посібник з Nmap. URL: <https://hackyourmom.com/kibervijna/posibnyk-z-nmap/>
20. HackYourMom. Основи та застосування протоколів TCP/IP. URL: <https://hackyourmom.com/kibervijna/osnovy-ta-zastosuvannya-protokoliv-tcp-ip-povnyj-putivnyk/>
21. HackYourMom. Pentest Wi-Fi: інструменти для захисту мережі. URL: <https://hackyourmom.com/privatnist/pentest-wifi-instrumenty-dlya-zahystu-vashoyi-merezhi/>
22. HackYourMom. Практичний посібник із використання Wireshark і tcpdump у локальних мережах. URL: <https://hackyourmom.com/kibervijna/chastyna-6-praktychnyj-posibnyk-iz-vykorystannya-wireshark-i-tcpdump-u-lokalnyh-merezhah-analiz-paketiv-iz-komandnogo-ryadka/>
23. HackYourMom. Шпаргалка MITM. URL: <https://hackyourmom.com/kibervijna/shpargalka-mitm/>

24. HackYourMom. WPA2 Enterprise для домашньої мережі. URL: <https://hackyourmom.com/kibervijna/chy-var-to-zastosovuvaty-wpa2-enterprise-dlya-vashoyi-domashnoyi-merezhi/>
25. Foxminded. Brute Force: принцип роботи та захист. URL: <https://foxminded.ua/brute-force/>
26. Medium. Understanding the Benefits of Using Kali Linux for Penetration Testing. URL: <https://medium.com/@techlatest.net/understanding-the-benefits-of-using-kali-linux-for-penetration-testing-bfc89b8fab39>
27. Oracle. Oracle VM VirtualBox Documentation. URL: <https://www.virtualbox.org/wiki/Documentation>
28. Python Software Foundation. Python Documentation. URL: <https://docs.python.org/3/>
29. Mozilla Foundation. Mozilla Firefox Documentation. URL: <https://support.mozilla.org/>
30. RFC 1035. Domain Names – Implementation and Specification. URL: <https://www.rfc-editor.org/info/rfc1035/>

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				92
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				93
Змн.	Арк.	№ докум.	Підпис	Дата		

Вихідний код базової структури програмного сніффера

The screenshot shows a Kali Linux virtual machine environment. On the left, a terminal window displays the following commands and their outputs:

```

(kali@kali)-[~]
$ mkdir ~/wifi_sniffer_diploma
(kali@kali)-[~]
$ cd ~/wifi_sniffer_diploma
(kali@kali)-[~/wifi_sniffer_diploma]
$ nano sniffer.py
(kali@kali)-[~/wifi_sniffer_diploma]
$ mousepad sniffer.py
  
```

On the right, the Mousepad text editor shows the Python code for the sniffer program:

```

1 from scapy.all import *
2
3 class PacketCapture:
4     def capture_packets(self):
5         print("Packet capture module")
6
7 class PacketAnalysis:
8     def analyze_packets(self):
9         print("Packet analysis module")
10
11 class SecurityAnalysis:
12     def detect_threats(self):
13         print("Security analysis module")
14
15 class SaveResults:
16     def save_to_csv(self):
17         print("Saving results")
18
19 def main():
20     capture = PacketCapture()
21     analysis = PacketAnalysis()
22     security = SecurityAnalysis()
23     save = SaveResults()
24
25     capture.capture_packets()
26     analysis.analyze_packets()
27     security.detect_threats()
28     save.save_to_csv()
29
30 if __name__ == "__main__":
31     main()
32
  
```

Рисунок 4.1 – Реалізація базової архітектури програмного сніффера

У лістингу наведено базову структуру програмного сніффера, реалізовану мовою Python. Архітектура програмного засобу побудована за модульним принципом та містить модулі захоплення пакетів, аналізу мережевого трафіку, аналізу безпеки та збереження результатів. Такий підхід забезпечує зручність подальшого розширення функціональних можливостей системи та спрощує супровід програмного коду.

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				94
Змн.	Арк.	№ докум.	Підпис	Дата		

```

kpranchuk-kb-22-2 [Запущено] - Oracle VirtualBox
Файл  Машина  Перегляд  Введення  Пристрої  Довідка
kali@kali: ~/wifi_sniffer_diploma
Session Actions Edit View Help
GNU nano 9.0 dns_analyzer.py *
from scapy.all import *

def packet_callback(packet):
    if packet.haslayer(DNS):
        print(packet.summary())

sniff(prn=packet_callback,count=20)

```

Рисунок 4.2 – Вихідний код модуля аналізу DNS-трафіку

У лістингу наведено програмну реалізацію модуля аналізу DNS-трафіку, створеного мовою Python із використанням бібліотеки Scapy. Модуль призначений для перехоплення мережових пакетів та виявлення DNS-запитів і DNS-відповідей у мережевому потоці.

```

kpranchuk-kb-22-2 [Запущено] - Oracle VirtualBox
Файл  Машина  Перегляд  Введення  Пристрої  Довідка
kali@kali: ~/wifi_sniffer_diploma
Session Actions Edit View Help
GNU nano 9.0 arp_analyzer.py *
from scapy.all import *

def packet_callback(packet):
    if packet.haslayer(ARP):
        print(packet.summary())

sniff(prn=packet_callback,count=20)

```

Рисунок 4.3 – Вихідний код модуля аналізу ARP-трафіку

У лістингу наведено програмну реалізацію модуля аналізу ARP-трафіку, створеного мовою Python із використанням бібліотеки Scapy. Основним призначенням даного модуля є перехоплення та дослідження пакетів протоколу ARP (Address Resolution Protocol), який використовується для визначення відповідності між IP-адресами та фізичними MAC-адресами пристроїв у локальній мережі. програмну реалізацію модуля аналізу DNS-трафіку, створеного мовою Python із використанням бібліотеки Scapy. Модуль призначений для перехоплення мережових пакетів та виявлення DNS-запитів і DNS-відповідей у мережевому потоці.

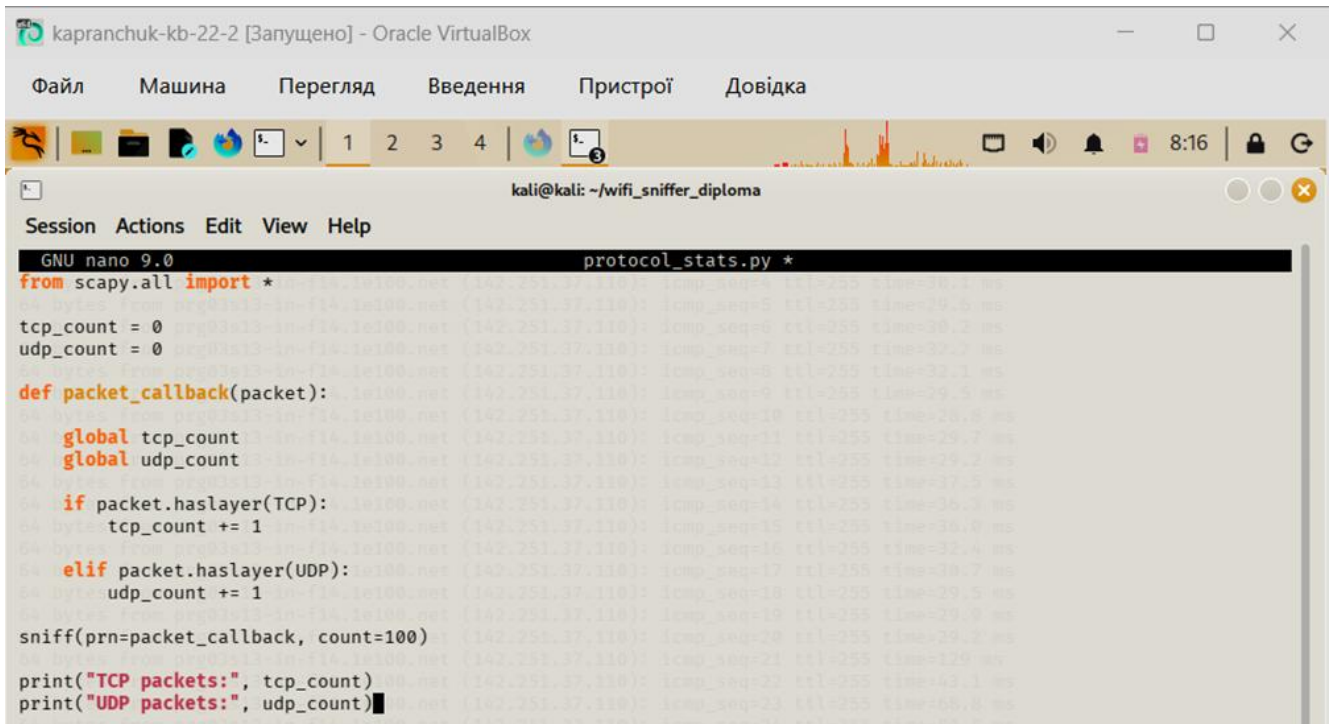


Рисунок 4.4 – Вихідний код модуля статистичного аналізу мережевих протоколів

У лістингу наведено програмну реалізацію модуля статистичного аналізу мережевого трафіку. Основним призначенням даного модуля є визначення кількості пакетів транспортних протоколів TCP та UDP, що були зафіксовані під час моніторингу мережевого середовища.

Лістинг програмної реалізації побудови архітектури програмного сніффера

```

import os
import matplotlib.pyplot as plt

print("Загальна структура програмного сніффера наведена на рисунку 2.1.")
print("Користувач")
print("  ↓")
print("Модуль захоплення пакетів")
print("  ↓")
print("Модуль аналізу трафіку")
print("  ↓")
print("Модуль аналізу безпеки")
print("  ↓")
print("Модуль формування звітів")
print("\nТака архітектура дозволяє організувати послідовну обробку мережевих да-
них")
print("та забезпечити виконання всіх функцій, визначених під час аналізу вимог до
системи.\n")

modules = [
    "Користувач",

```

		Капрanchук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				96
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    "Модуль\нзахоплення пакетів",
    "Модуль\наналізу трафіку",
    "Модуль\наналізу безпеки",
    "Модуль\нформування звітів"
]

fig, ax = plt.subplots(figsize=(6, 8))
y_positions = [8, 6, 4, 2, 0]
x_pos = 5

for i, text in enumerate(modules):

    ax.text(
        x_pos, y_positions[i], text,
        ha="center", va="center", color="black", fontsize=11, fontname="Arial",
        bbox=dict(boxstyle="round,pad=0.6", facecolor="#E0E0E0", edgecolor="gray",
lw=1.5)
    )

    if i < len(modules) - 1:
        ax.annotate(
            "",
            xy=(x_pos, y_positions[i+1] + 0.6),
            xytext=(x_pos, y_positions[i] - 0.6),
            arrowprops=dict(arrowstyle="->", color="black", lw=2,
mutation_scale=20)
        )

ax.set_xlim(2, 8)
ax.set_ylim(-1, 9)
ax.axis("off")

output_filename = "architecture_diagram.png"
plt.savefig(output_filename, bbox_inches="tight", dpi=300)
plt.close(fig)

```

Лістинг програмної реалізації блок-схеми обробки мережевих пакетів

```

import os
import matplotlib.pyplot as plt

print("Текстовий вигляд структури:")
print("Початок роботи програми")
print("  ↓")
print("Ініціалізація мережевого інтерфейсу")
print("  ↓")
print("Захоплення пакета")
print("  ↓")
print("Аналіз заголовків")
print("  ↓")
print("Визначення параметрів пакета")
print("  ↓")
print("Збереження статистики")
print("  ↓")
print("Передача результатів на аналіз")
print("  ↓")
print("Кінець циклу\n")

steps = [
    "Початок роботи програми",

```

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				97
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        "Ініціалізація\нмережевого інтерфейсу",
        "Захоплення пакета",
        "Аналіз заголовків",
        "Визначення\нпараметрів пакета",
        "Збереження статистики",
        "Передача результатів\нна аналіз",
        "Кінець циклу"
    ]

fig, ax = plt.subplots(figsize=(6, 10))

y_positions = [14, 12, 10, 8, 6, 4, 2, 0]
x_pos = 5

for i, text in enumerate(steps):

    box_color = "#E0E0E0" if i not in [0, len(steps) - 1] else "#D4EDDA"

    ax.text(
        x_pos, y_positions[i], text,
        ha="center", va="center", color="black", fontsize=11, fontname="Arial",
        bbox=dict(boxstyle="round,pad=0.6", facecolor=box_color, edgecolor="gray",
lw=1.5)
    )

    if i < len(steps) - 1:
        ax.annotate(
            "",
            xy=(x_pos, y_positions[i + 1] + 0.5),
            xytext=(x_pos, y_positions[i] - 0.5),
            arrowprops=dict(arrowstyle="->", color="black", lw=2,
mutation_scale=20)
        )

ax.set_xlim(2, 8)
ax.set_ylim(-1, 15)
ax.axis("off")

output_filename = "flowchart_diagram.png"
plt.savefig(output_filename, bbox_inches="tight", dpi=300)
plt.close(fig)

```

Лістинг програмної реалізації схеми взаємодії функціональних модулів

```

import os
import matplotlib.pyplot as plt
from matplotlib.patches import Polygon

print("Генерується блок-схема алгоритму з циклом та умовою...")

fig, ax = plt.subplots(figsize=(8, 11))

steps = [
    "Початок роботи програми",
    "Ініціалізація\нмережевого інтерфейсу",

```

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				98
Змн.	Арк.	№ докум.	Підпис	Дата		

```

"Запуск режиму\нмоніторингу трафіку",
"Захоплення\нмережевого пакета",
"Аналіз заголовків пакета",
"Визначення\нпараметрів пакета",
"Збереження статистики",
"Передача результатів\нна аналіз",
"Перевірка умови\нзавершення роботи?",
"Завершення програми"
]

y_positions = [18, 16, 14, 12, 10, 8, 6, 4, 1.5, -1.5]
x_pos = 5

for i, text in enumerate(steps):
    if i == 8:
        continue

    box_color = "#E0E0E0" if i not in [0, 9] else "#D4EDDA"

    ax.text(
        x_pos, y_positions[i], text,
        ha="center", va="center", color="black", fontsize=10, fontname="Arial",
        bbox=dict(boxstyle="round,pad=0.5", facecolor=box_color,
edgecolor="#7F7F7F", lw=1.5)
    )

y_rhomb = y_positions[8]

diamond_vertices = [
    [x_pos - 1.8, y_rhomb],
    [x_pos, y_rhomb + 0.9],
    [x_pos + 1.8, y_rhomb],
    [x_pos, y_rhomb - 0.9]
]
diamond = Polygon(diamond_vertices, facecolor="#FFF3CD", edgecolor="#7F7F7F",
lw=1.5)
ax.add_patch(diamond)

ax.text(x_pos, y_rhomb, steps[8], ha="center", va="center", color="black",
fontsize=10, fontname="Arial")

for i in range(len(steps) - 1):
    if i != 8:
        y_start = y_positions[i] - 0.5 if i != 7 else y_positions[i] - 0.6
        y_end = y_positions[i + 1] + 0.5 if i + 1 != 8 else y_positions[i + 1] +
0.9

        ax.annotate("", xy=(x_pos, y_end), xytext=(x_pos, y_start),
arrowprops=dict(arrowstyle="->", color="black", lw=1.6, muta-
tion_scale=14))

ax.text(x_pos + 0.2, 0.2, "Так", ha="left", va="center", fontsize=10, font-
name="Arial", color="green",
fontweight="bold")

```

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				99
Змн.	Арк.	№ докум.	Підпис	Дата		

```

ax.plot([x_pos + 1.8, x_pos + 2.5], [y_rhomb, y_rhomb], color="black", lw=1.6)
ax.plot([x_pos + 2.5, x_pos + 2.5], [y_rhomb, y_positions[3]], color="black",
lw=1.6)
ax.annotate("", xy=(x_pos + 1.3, y_positions[3]), xytext=(x_pos + 2.5, y_posi-
tions[3]),
            arrowprops=dict(arrowstyle="->", color="black", lw=1.6, muta-
tion_scale=14))

ax.text(x_pos + 2.0, y_rhomb + 0.2, "Hi", ha="center", va="bottom", fontsize=10,
fontname="Arial", color="red",
        fontweight="bold")

ax.set_xlim(1, 8)
ax.set_ylim(-3, 20)
ax.axis("off")

output_filename = "monitoring_algorithm.png"
plt.savefig(output_filename, bbox_inches="tight", dpi=300)
plt.close(fig)

```

Лістинг програмної реалізації алгоритму аналізу безпеки мережевого се- редовища

```

import os
import matplotlib.pyplot as plt

print("Рисунок 2.4 – Алгоритм оцінювання безпеки мережі")
print("Початок аналізу")
print("  ↓")
print("Виявлення активних вузлів")
print("  ↓")
print("Аналіз відкритих портів")
print("  ↓")
print("Визначення мережевих сервісів")
print("  ↓")
print("Пошук ознак підозрілої активності")
print("  ↓")
print("Формування звіту безпеки")
print("  ↓")
print("Завершення\n")

steps = [
    "Початок аналізу",
    "Виявлення\nактивних вузлів",
    "Аналіз\nвідкритих портів",
    "Визначення\nмережевих сервісів",
    "Пошук ознак\nпідозрілої активності",
    "Формування\nзвіту безпеки",
    "Завершення"
]

fig, ax = plt.subplots(figsize=(6, 9))

y_positions = [12, 10, 8, 6, 4, 2, 0]
x_pos = 5

```

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		100

```

for i, text in enumerate(steps):

    box_color = "#E0E0E0" if i not in [0, len(steps) - 1] else "#D4EDDA"

    ax.text(
        x_pos, y_positions[i], text,
        ha="center", va="center", color="black", fontsize=11, fontname="Arial",
        bbox=dict(boxstyle="round,pad=0.6", facecolor=box_color,
edgecolor="#7F7F7F", lw=1.5)
    )

    if i < len(steps) - 1:
        ax.annotate(
            "",
            xy=(x_pos, y_positions[i + 1] + 0.5),
            xytext=(x_pos, y_positions[i] - 0.5),
            arrowprops=dict(arrowstyle="->", color="black", lw=1.8,
mutation_scale=15)
        )

ax.set_xlim(2, 8)
ax.set_ylim(-1, 13)
ax.axis("off")

output_filename = "security_algorithm.png"
plt.savefig(output_filename, bbox_inches="tight", dpi=300)
plt.close(fig)

```

		Капранчук В.М.			ККІКБ.КР.Б – 125 – 6 – ПЗ	Арк.
		Шелуха О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		101