Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Навчально-науковий
інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)
Кафедра інформаційних технологій та комп'ютерної інженерії
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня бакалавра

здобувача Ручки Євгенія Володимировича
(ПІБ)

академічної групи 123-22зск-1
(шифр)

спеціальності 123 Комп'ютерна інженерія
(код і назва спеціальності)

за освітньо-професійною програмою Комп'ютерна інженерія
(офіційна назва)

на тему «Розробка симулятора для аналізу мережевого трафіку та виявлення загроз у бездротових комунікаціях»
(назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	проф. Нікулін С.Л.			
спеціальної частини	проф. Нікулін С.Л.			
Рецензент				
Нормоконтролер	проф. Цвіркун Л.І.			

Дніпро
2025

ЗАТВЕРДЖЕНО:
завідувач кафедри
інформаційних технологій
та комп'ютерної інженерії
(повна назва)
_____ Гнатушенко В.В.
(підпис) (прізвище, ініціали)

«___» _____ 2025 року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня бакалавр

здобувача Ручки Є.В. **академічної групи** 123-22зск-1
(прізвище та ініціали) (шифр)

спеціальності 123 «Комп'ютерна інженерія»

за освітньо-професійною програмою «Комп'ютерна інженерія»
(офіційна назва)

на тему «Розробка симулятора для аналізу мережевого трафіку та виявлення загроз у бездротових комунікаціях»

затверджену наказом ректора НТУ «Дніпровська політехніка» від 05.05.2022 № 337-с

Розділ	Зміст	Термін виконання
Стан питання і постановка завдання	На основі матеріалів виробничих практик, інших науково-технічних джерел показати актуальність завдання, сформулювати мету та задачі виконання кваліфікаційної роботи	28.02.2025
Технічні вимоги до об'єкту вивчення	Сформулювати найменування й призначення комп'ютерної системи, висунути технічні вимоги до неї	15.04.2025
Розробка програмної частини	Розробити програму для аналізу мережевого трафіку та виявлення загроз у бездротових комунікаціях	30.05.2025

Завдання видано _____ проф. Нікулін С.Л.
(підпис керівника) (прізвище, ініціали)

Дата видачі _____

Дата подання до екзаменаційної комісії _____

Прийнято до виконання _____ Ручка Є.В.

РЕФЕРАТ

Пояснювальна записка: 51 с., 9 рис., 15 джерел, 1 додаток.

БЕЗПЕКА, ЗАГРОЗА, СИМУЛЯТОР, КОНТРОЛЬ, МЕРЕЖА, ТРАФІК, БЕЗДРОТОВІ КОМУНІКАЦІЇ

Об’єкт вивчення – симулятор для аналізу мережевого трафіку та виявлення загроз у бездротових комунікаціях.

Мета роботи − створення симулятору для аналізу мережевого трафіку та виявлення загроз у бездротових комунікаціях.

Зростання автоматизації процесів обробки, зберігання та передачі даних супроводжується загостренням проблем інформаційної безпеки. Збитки від кіберзлочинів постійно збільшуються, а кількість атак на комп'ютерні системи та мережі демонструє стійку тенденцію до зростання. У зв'язку з цим, актуальним є розробка програмного забезпечення для аналізу внутрішнього мережевого трафіку, яке б дозволило своєчасно виявляти загрози та вживати необхідних заходів для захисту пристроїв і програм від негативного впливу мережі.

Здійснено розробку симулятору для аналізу мережевого трафіку з метою виявлення загроз у бездротових мережах.

Для досягнення цієї мети була розроблена модульна архітектура симулятора, яка забезпечує чітке розділення функціоналу на логічні підсистеми, взаємодію між ними та легкість подальшого розширення. Також був створений функціонал для імітації різних типів бездротових кадрів 802.11 (маяків, кадрів даних, пробових запитів), що відображають типову активність у Wi-Fi мережах.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

API – Application Programming Interface;

CGI – Common Gateway Interface;

CRM – Customer Relationship Management;

DNS – Domain Name Service;

FTP – File Transfer Protocol;

GUI – Graphical User Interface;

LAN – Local Area Network;

TCP/IP – Transmission Control Protocol/Internet Protocol;

СВВ – системи виявлення вторгнень;

ВСТУП

Стрімке зростання популярності та відповідний розвиток комп'ютерних мереж призвели до значного ускладнення та взаємозв'язку комп'ютерних систем, що зробило їх більш вразливими до зловмисної діяльності. Підвищення рівня автоматизації процесів обробки, зберігання та передачі даних також спричинило появу нових проблем у сфері безпеки, а витрати, пов'язані з відшкодуванням збитків, завданих кіберзлочинцями, постійно зростають, [1].

Слід зазначити стійку тенденцію до збільшення кількості кібератак на комп'ютерні системи та мережі, [2]. Методи та способи дистанційного впливу безперервно вдосконалюються, а існуючі системи захисту часто не забезпечують своєчасного реагування на ці зміни. Ефективна протидія вимагає першочергового виявлення мережевої атаки для подальшого розуміння її характеру. У зв'язку з цим, повне усунення зловмисного трафіку є складним завданням. В таких умовах особлива увага приділяється розробці дієвих методів виявлення шкідливого трафіку та розробці адекватних заходів захисту даних.

Актуальність дослідження. Сьогодні комп'ютерні мережі стали невід'ємною частиною діяльності майже кожного підприємства, забезпечуючи зв'язок як між співробітниками в межах одного офісу чи будівлі, так і між командами, розташованими в різних містах і країнах. Ефективне управління сучасним підприємством неможливе без налагодженого контролю за інформаційними потоками та оперативного координування роботи всіх підрозділів і співробітників.

На сьогоднішній день активно розробляються та застосовуються різноманітні методи контролю та моніторингу мережевого трафіку, проте їхня практична ефективність не завжди є достатньою. Внаслідок цього всі існуючі технології захисту постійно піддаються дослідженню та вдосконаленню.

Дана робота спрямована на розробку симулятора для аналізу мережевого трафіку та виявлення загроз у бездротових комунікаціях. Отримані в результаті аналізу дані дозволять своєчасно виявляти потенційні загрози та вживати необхідних заходів для захисту пристроїв і програм від негативного впливу мережі.

Основною метою даної роботи є удосконалення існуючої системи збору мережевого трафіку у бездротових комунікаціях шляхом розширення її функціональних можливостей для проведення поглибленого аналізу та своєчасного виявлення несанкціонованої активності.

Для досягнення поставленої мети було визначено наступні завдання:

- визначити та аргументувати вибір оптимальних мов програмування, бібліотек та фреймворків, які забезпечать ефективність, гнучкість та масштабованість розроблюваної системи;
- розробити модульну архітектуру симулятора, яка забезпечить чітке розділення функціоналу на логічні підсистеми, взаємодію між ними та легкість подальшого розширення;
- створити функціонал для імітації різних типів бездротових кадрів 802.11 (маяків, кадрів даних, пробових запитів), що відображають типову активність у Wi-Fi мережах;
- впровадити функціонал для моделювання типових атак на бездротові мережі, зокрема DoS-атаки (Deauthentication Flood), з метою демонстрації їх впливу та перевірки механізмів виявлення;
- імплементувати базові алгоритми та правила для ідентифікації аномалій та виявлення імітованих загроз у згенерованому трафіку (наприклад, DoS-атаки, Evil Twin);
- розробити інтуїтивно зрозумілий та функціональний інтерфейс, який дозволить користувачеві керувати симуляцією (запуск, зупинка, ініціація атак), відображати захоплені дані та візуалізувати результати аналізу;

- додати функціонал для графічного представлення результатів аналізу трафіку, таких як розподіл типів кадрів, для наочної демонстрації симульованих подій;
- реалізувати механізм управління потоками для забезпечення асинхронної роботи GUI та тривалих процесів симуляції, що дозволить уникнути «зависання» інтерфейсу.

1 СТАН ПИТАННЯ І ПОСТАНОВКА ЗАВДАННЯ

У процесі обробки та зберігання даних неминучим є обмін інформацією між різними учасниками. Починаючи з кінця 1970-х років, спостерігався стрімкий розвиток комп'ютерних мереж та відповідного мережевого обладнання. Цей розвиток триває і сьогодні, охоплюючи як локальні, так і глобальні мережі, появу нових протоколів передачі даних, розширення апаратних можливостей мережевих пристроїв, збільшення кількості підключених користувачів та загального обсягу мережевого трафіку.

Такий інтенсивний розвиток породжує ряд проблем. Однією з ключових є зростання вимог до мережевого та серверного обладнання зі збільшенням кількості користувачів послуг передачі даних, що необхідно для підтримки належного рівня якості обслуговування. Важливим аспектом цього є забезпечення захисту даних, які циркулюють у мережах та зберігаються на пристроях, [3].

Для вирішення зазначених проблем широко застосовуються методи контролю та моніторингу мережевого трафіку, які допомагають ефективно виявляти та оперативно усувати виникаючі складності.

1.1 Класифікація засобів моніторингу та контролю

Засоби моніторингу та контролю мережевого трафіку можна умовно поділити на кілька основних категорій залежно від їхніх функцій та призначення, [4]. Однією з ключових є системи керування мережею — централізовані програмні комплекси, які дозволяють збирати інформацію про стан мережевого обладнання та трафіку. Окрім простого моніторингу, вони часто забезпечують напівавтоматичне або автоматичне керування мережею, наприклад, налаштування таблиць комутації або керування портами. До прикладів таких систем належать HPOpenView, SolarWinds Network Performance Monitor та IBM Tivoli NetView. Сучасні рішення цього типу

часто мають потужні можливості візуалізації, створення звітів та інтеграції з іншими інструментами.

Ще однією важливою категорією є внутрішні системи діагностики та контролю, які вбудовані безпосередньо в мережеві пристрої або операційні системи. Вони забезпечують моніторинг лише тих пристроїв, у яких встановлені, діючи, як правило, як SNMP-агенти, що передають дані до централізованої системи. Сучасні комутатори та маршрутизатори оснащені такими внутрішніми засобами, які дозволяють отримати детальну інформацію про продуктивність інтерфейсів, помилки та завантаження ресурсів.

До третьої групи належать засоби системного адміністрування, які подібні за функціональністю до систем керування мережею, однак зосереджуються на контролі за програмно-апаратними компонентами комп’ютерів у мережі, зокрема серверами та робочими станціями. Популярними прикладами є Prometheus, Grafana, Nagios Core з плагінами та Zabbix.

Четверта група представлена аналізаторами протоколів — програмними або апаратними засобами для детального аналізу мережевого трафіку. Вони здатні перехоплювати, декодувати й візуалізувати пакети даних відповідно до протоколів. Серед найвідоміших — Wireshark, tcpdump, Fiddler і CommView.

П’ята група охоплює обладнання для діагностики та сертифікації кабельної інфраструктури, включаючи кабельні тестери, аналізатори сигналу, оптичні рефлектометри (OTDR) та сертифікатори. Вони дають змогу перевірити цілісність, якість та відповідність кабельних систем стандартам.

Шоста категорія — експертні системи, які використовують знання фахівців для виявлення причин несправностей у мережі та пропонують рішення для їх усунення. Вони можуть бути як простими довідковими системами, так і складними аналітичними платформами з елементами

штучного інтелекту. У сучасних інструментах моніторингу такі можливості часто інтегровані.

Сьома група охоплює багатофункціональні портативні пристрої, які об'єднують кілька діагностичних функцій в одному корпусі, наприклад, кабельні тестери, базовий аналіз трафіку та елементи управління мережею. До таких приладів належать продукти Fluke Networks, NetAlly тощо.

Окрім зазначених засобів, використовуються також вбудовані функції моніторингу в маршрутизаторах, які можуть забезпечувати аналіз трафіку, виявлення аномалій та базові засоби безпеки без потреби у встановленні додаткового ПЗ. У деяких випадках застосовується індивідуальний підхід до побудови системи моніторингу — шляхом ретельного підбору обладнання та програмного забезпечення відповідно до потреб конкретної організації.

1.2 Системи виявлення та запобігання вторгненням

У сучасних мережевих інфраструктурах впровадження комплексних систем захисту інформації має критичне значення. Це зумовлено тим, що існують автоматизовані інструменти, зокрема пошукова система Shodan, які постійно сканують глобальну мережу з метою виявлення вразливостей у підключеному обладнанні. Особливу небезпеку становлять ті пристрої, що не мають належного захисту або використовують стандартні налаштування доступу. Такі системи легко стають об'єктами атак, що може призвести до порушення їхньої працездатності.

Для протидії подібним загрозам застосовуються системи виявлення вторгнень (IDS — Intrusion Detection System) та системи запобігання вторгненням (IPS — Intrusion Prevention System), які є важливою частиною сучасної політики інформаційної безпеки, [2]. IDS — це програмні або апаратні засоби, що призначені для фіксації спроб несанкціонованого доступу до комп'ютерної системи або мережі. IPS, у свою чергу, є розширенням функціональності IDS і здатна не лише виявляти, а й оперативно блокувати шкідливу активність у реальному часі. Для цього IPS

може скинути підозріле з'єднання, обмежити мережевий трафік або повідомити оператора. Додатково такі системи можуть реалізовувати складні захисні механізми, наприклад, дефрагментацію пакетів або зміну порядку TCP-сегментів, [5].

Зазначені системи дозволяють автоматизувати процес моніторингу подій у комп'ютерних системах та мережах і проводити їхній аналіз для виявлення можливих загроз безпеці. У зв’язку з постійним зростанням як кількості, так і складності методів несанкціонованих вторгнень, СВВ стали обов’язковим компонентом безпекової інфраструктури більшості організацій. Це пояснюється як широкою доступністю інформації про існуючі вразливості, якою активно користуються зловмисники, так і появою нових, дедалі витонченіших методів проникнення в системи.

Сучасні СВВ мають різні архітектурні рішення, серед яких основними є мережеві та локальні системи. Мережеві СВВ встановлюються на спеціалізованих пристроях і аналізують трафік у локальній мережі, тоді як локальні — працюють безпосередньо на пристроях, що потребують захисту, і відстежують внутрішні дії, наприклад системні виклики чи поведінку користувачів.

Крім архітектури, СВВ також класифікуються за методом виявлення загроз. Частина з них орієнтується на пошук аномалій, що відхиляються від типових моделей поведінки, а інші застосовують сигнатурний підхід, виявляючи відомі загрози на основі характерних ознак їхньої активності, [6]. Найефективніші сучасні рішення зазвичай поєднують обидва підходи для забезпечення максимального рівня захисту.

1.2.1 Аналіз методів виявлення аномалій і зловживань

Системи виявлення аномальної поведінки (anomaly detection) функціонують на основі знання СВВ характеристик легітимної або допустимої поведінки контрольованого об'єкта. Будь-яка дія, що

відхиляється від цієї «норми» або «правильної» поведінки, розцінюється як потенційно підозріла.

На противагу цьому, системи виявлення шкідливої поведінки (виявлення зловживань) орієнтуються на розпізнавання характерних ознак, що свідчать про дії зловмисника. Поширеними прикладами професійних систем, що реалізують технологію виявлення зловживань, є Snort, Fortinet FortiGate NGFW (який включає функції IPS), Cisco Secure IPS (раніше RealSecure IDS), та IBM Security Network Protection (раніше Enterasys Advanced Dragon IDS).

Розглянемо детальніше технології, що лежать в основі цих систем (рис. 1.1), [3, 4].

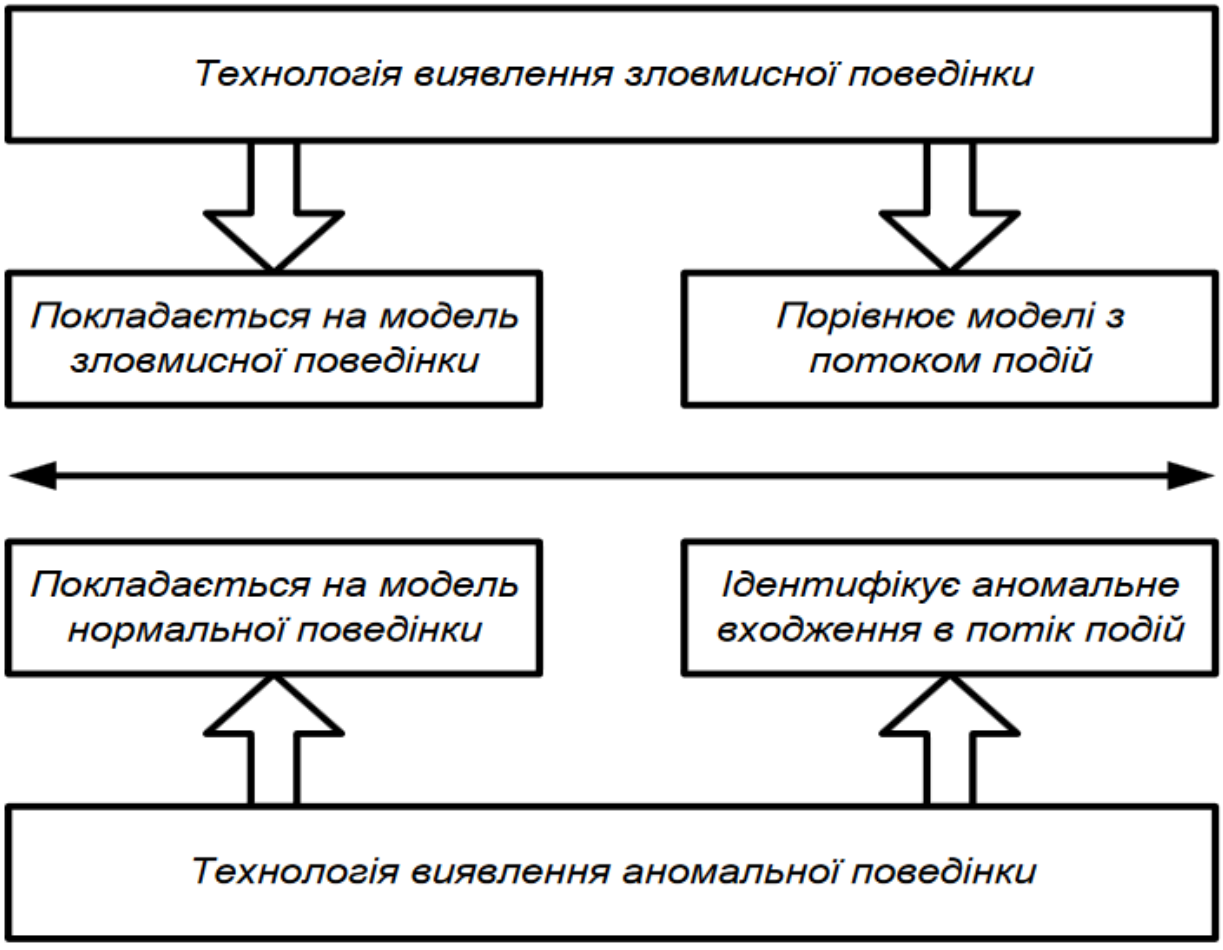

Рисунок 1.1 — Існуючі технології СВВ

Датчики аномалій призначені для виявлення нетипової поведінки, тобто аномалій, у функціонуванні певного об'єкта. Основні складнощі їх

практичного застосування пов'язані зі змінливістю поведінки об'єктів, що знаходяться під захистом, а також особливостями їхньої взаємодії із зовнішнім середовищем. Об'єктом спостереження в системах виявлення аномалій може виступати окремий комп'ютер, певна мережева служба (наприклад, FTP-сервер), обліковий запис користувача тощо. Спрацьовування датчика відбувається, коли зафіксована активність істотно відрізняється від визначеної як «нормальна» або «легітимна». Важливо зазначити, що для різних типів операцій існують індивідуальні межі допустимих відхилень від встановленої моделі поведінки, а кожен датчик моніторингу має власний налаштований «поріг чутливості». Сучасні системи виявлення аномалій для підвищення точності ідентифікації відхилень та мінімізації хибних спрацьовувань використовують широкий спектр методів, включаючи статистичний аналіз, алгоритми машинного навчання та поведінковий аналіз.

1.2.2 Технології виявлення аномальної діяльності

Системи виявлення аномалій (anomaly detection) функціонують, аналізуючи симптоми, що характеризують типову або дозволену поведінку контрольованого об'єкта СВВ. Будь-яка поведінка, що відхиляється від визначеної «нормальної» або «правильної» поведінки об'єкта, розглядається як потенційне порушення політики безпеки, [5].

На противагу цьому, системи виявлення зловмисної поведінки (виявлення зловживань) зосереджуються на розпізнаванні характерних ознак, які вказують на дії зловмисника. До поширених професійних систем, що реалізують цю технологію, належать Snort, Fortinet FortiGate NGFW (який включає функції IPS), Cisco Secure IPS (раніше RealSecure IDS), та IBM Security Network Protection (раніше Enterasys Advanced Dragon IDS).

Розглянемо детальніше технології, що лежать в основі цих систем (рис. 1.1), [7].

Датчики аномалій виявляють незвичайну поведінку, тобто аномалії, у роботі конкретного об'єкта. Об'єктом спостереження може бути окремий комп'ютер, мережева служба (наприклад, FTP-сервер), користувач тощо. Основні труднощі практичного застосування таких датчиків пов'язані зі змінливістю поведінки об'єктів, що захищаються, та їхньою взаємодією із зовнішніми системами. Датчики спрацьовують, якщо зафіксована активність відрізняється від «нормальної» (легітимної). Важливо, що для кожної операції існує визначений діапазон допустимих відхилень поведінки, а також встановлений «поріг спрацьовування» для датчика моніторингу.

Перевагами систем виявлення аномалій є:

- відсутність необхідності постійного оновлення сигнатур та правил виявлення атак;
- потенційна можливість виявлення нових, раніше невідомих видів атак, для яких ще не розроблені сигнатури;
- створення даних, які можуть бути використані в системах виявлення шкідливого програмного забезпечення (хоча це більше відноситься до аналізу поведінки, ніж до самого виявлення аномалій).

Недоліками цих систем є:

- висока ймовірність хибних спрацьовувань (помилок другого роду), коли легітимна, але нетипова поведінка помилково ідентифікується як атака;
- потреба у тривалому та ретельному процесі навчання системи для визначення «нормальної» поведінки»
- потенційно низька швидкість роботи та значні вимоги до обчислювальних ресурсів, особливо при аналізі великих обсягів даних у реальному часі.

Сучасні системи намагаються оптимізувати ці аспекти за допомогою ефективних алгоритмів та апаратного прискорення.

1.2.3 Статистичний аналіз комп'ютерних атак

Найпоширенішим підходом до виявлення аномальної поведінки є використання методів статистичного аналізу. Статистичні датчики збирають різноманітні дані про типову поведінку об'єкта спостереження та формують на їх основі профіль. Цей профіль умов являє собою набір параметрів, що описують характерну поведінку об'єкта, і генерується за допомогою методів математичної статистики, таких як метод ковзного вікна або метод зважених сум.

Процес починається з початкового періоду побудови профілю. Після цього поточна активність об'єкта постійно порівнюється з параметрами профілю, і у разі виявлення значних відхилень система сигналізує про можливу атаку. Параметри профілю можуть бути поділені на загальні групи:

1. Категоріальні параметри, які включають нечислові дані, такі як імена файлів, введені користувачем команди, відкриті мережеві порти тощо.
2. Числові параметри, які охоплюють кількісні показники, що не підлягають класифікації, наприклад, обсяг переданих даних за різними протоколами, завантаження центрального процесора, кількість отриманих файлів тощо.

Для більш точного відображення мінливої поведінки об'єктів також існують механізми динамічного оновлення профілів. Системи, що використовують статистичні методи виявлення аномалій, мають ряд переваг:

- не потребують постійного оновлення баз даних сигнатур атак, що спрощує їх обслуговування;
- здатні адаптуватися до змін у поведінці користувачів, що робить їх більш чутливими до спроб вторгнення, ніж ручні методи аналізу;
- можуть виявляти раніше невідомі атаки, для яких ще не існують сигнатури, виступаючи таким чином в якості своєрідного буфера до розробки відповідних правил для сигнатурних систем;

- потенційно здатні виявляти складніші атаки, такі як розподілені в часі або скоординовані атаки.

Однак, системи виявлення вторгнень, що базуються на статистичних методах, також мають певні недоліки:

- вищий ризик хибних спрацьовувань (помилкових повідомлень про атаку) порівняно з іншими методами;
- складність визначення оптимальних порогових значень для параметрів профілю, що вимагає глибокого розуміння поведінки контрольованої системи;
- можуть неточно фіксувати легітимні, але нетипові зміни в активності користувачів, що може призводити до помилкових спрацьовувань.

Крім того:

- система може помилково прийняти поведінку, що насправді є атакою, за нормальну, якщо зміни в рутинній діяльності відбуваються поступово, оскільки профіль адаптується до нової поведінки;
- статистичні методи можуть бути неефективними у виявленні атак від суб'єктів, чию типову поведінку складно описати статистично;
- потребують попереднього налаштування, включаючи визначення порогових значень для кожного параметра та для кожного користувача;
- чисто статистичні системи можуть не відразу виявляти атаки від зловмисників, для яких зловмисна діяльність є типовою поведінкою, оскільки профіль будується на основі їхньої звичайної активності;
- статистичні методи, що базуються на профілях, можуть бути нечутливими до порядку подій.

Незважаючи на ці недоліки, ведуться дослідження та розробки для їхнього усунення, і чиста реалізація статистичних методів у технологіях виявлення аномалій залишається перспективним напрямком, успадковуючи

всі ключові переваги цього підходу, [8]. Сучасні підходи часто комбінують статистичні методи з іншими технологіями, такими як машинне навчання та поведінковий аналіз, для підвищення точності та зниження кількості хибних спрацьовувань.

1.3 Аналіз недоліків сучасних систем виявлення вторгнень

Зважаючи на вищезазначене, усі системи виявлення вторгнень можна узагальнено поділити на системи, що виявляють: відомі сигнатури атак, аномалії у взаємодії контрольованих об'єктів та спотворення еталонної профільної інформації.

На сьогоднішній день спостерігається недостатня кількість гібридних систем, здатних комплексно аналізувати дані, розподілені як у часі, так і в просторі. У більшості сучасних систем сигнатурний аналіз використовується окремо для розпізнавання наслідків дій зловмисників, або ж окремо для виявлення аномалій у поведінці контрольованих систем. Крім того, більшість відомих систем не мають вбудованих симуляторів атак або інших засобів перевірки ефективності системи виявлення вторгнень (СВВ), які б забезпечували простий та надійний спосіб тестування параметрів конфігурації в будь-якій комп'ютерній мережі.

Логічно, такий інструмент мав би можливість імітувати діяльність різноманітного шкідливого програмного забезпечення, атак типу «відмова в обслуговуванні», атак, спрямованих на підвищення привілеїв облікових записів (експлуатація вразливостей у мережевих службах, таких як MS SQL Server, MS IIS), а також атак, спрямованих на перенаправлення трафіку та введення неправдивої інформації. Бажано також, щоб програмне забезпечення могло генерувати розподілені атаки.

Одним із цікавих підходів до побудови симуляторів СВВ є використання багатоагентних систем, де різні типи агентів спеціалізуються на вирішенні окремих підзадач виявлення вторгнень та розгортаються на різних комп'ютерах системи. У таких архітектурах часто відсутній явний

«центр управління», оскільки будь-який агент може тимчасово виконувати функції лідера, ініціюючи співпрацю та керування залежно від поточної ситуації. Ці агенти можуть бути динамічно переміщені або відключені в мережевому та локальному середовищі. Залежно від типу та кількості атак, а також наявності обчислювальних ресурсів, може виникнути потреба у створенні кількох екземплярів агентів кожного класу. Очікується, що архітектура такої системи буде адаптуватися до конфігурації мережі, змін трафіку та появи нових типів атак, використовуючи накопичений досвід.

Хоча багатоагентні системи є перспективним напрямком, внутрішні механізми їхньої роботи часто не розкривають конкретні алгоритми виявлення атак. Крім того, існуючі емулятори можуть не працювати в режимі реального часу, що обмежує їхню практичну цінність для тестування систем захисту.

Загалом, відсутність ефективних симуляторів атак для оцінки продуктивності СВВ не є головною проблемою в цій галузі. Значно важливішими недоліками існуючих систем виявлення є надмірне покладання на простий сигнатурний аналіз, низька ефективність виявлення складних атак, розподілених у часі та просторі, а також недостатній обсяг інформації на рівні хоста та мережі для виявлення скоординованих та ізольованих атак.

З практичної точки зору, обробка всього обсягу інформації, що надходить на звичайних персональних комп'ютерах, на швидкостях сучасних мереж часто є неможливою через обмеження обчислювальних ресурсів. Аналіз трафіку та інших подій може відбуватися зі значною затримкою (в 1,5 - 2 рази повільніше за реальний час), що призводить до несвоєчасного виявлення або навіть пропуску атак на захищені дані та обчислювальні ресурси. Тому інструменти виявлення атак можуть використовуватися для фіксації всіх етапів атаки з метою подальшого детального аналізу.

Більшість сучасних СВВ не розробляються з урахуванням можливості їхньої роботи на різноманітних операційних системах та незалежних апаратних платформах. Це призводить до того, що більшість продуктів є

платформозалежними та не можуть використовувати переваги оптимізації коду для конкретних операційних систем та апаратних конфігурацій, що є суттєвим їхнім недоліком.

Крім того, більшість існуючих систем не підтримують режим «гарячої заміни», що ускладнює швидке розгортання резервного набору засобів захисту у випадку збою основного та відновлення порушеної лінії захисту периметра мережі.

Проте, у розвитку систем виявлення аномалій спостерігається позитивна тенденція до інтеграції з існуючими засобами захисту, такими як міжмережеві екрани (файрволи), системи блокування каналів та менеджери якості обслуговування (QoS).

На жаль, сучасний підхід до побудови систем виявлення вторгнень та ознак комп'ютерних атак на інформаційні системи має значні недоліки та вразливості, які дозволяють зловмисникам успішно обходити існуючі механізми захисту інформації.

На ринку представлено багато систем запобігання та виявлення вторгнень, [9], але їхнім спільним суттєвим недоліком є залежність від попередньо розроблених правил та шаблонів, що призводить до пропуску невідомих атак та нездатності «реагувати» на шкідливий трафік в реальному часі.

Крім того, слід зазначити слабку представленість продукції вітчизняних виробників на цьому ринку, а описані раніше методи часто є комерційною таємницею, [10]. Це підкреслює необхідність розробки нових систем, які можуть бути використані в наборі мережевих інструментів.

Розглянутий матеріал підтверджує актуальність проблем забезпечення безпеки мережі та необхідність подальшого дослідження та вдосконалення сучасних методів аналізу мережевого трафіку.

1.4 Мета і задачі роботи

Основною метою даної роботи є удосконалення існуючої системи збору мережевого трафіку у бездротових комунікаціях шляхом розширення її функціональних можливостей для проведення поглибленого аналізу та своєчасного виявлення несанкціонованої активності.

Для досягнення поставленої мети було визначено наступні завдання:

– визначити та аргументувати вибір оптимальних мов програмування, бібліотек та фреймворків, які забезпечать ефективність, гнучкість та масштабованість розроблюваної системи;

– розробити модульну архітектуру симулятора, яка забезпечить чітке розділення функціоналу на логічні підсистеми, взаємодію між ними та легкість подальшого розширення;

– створити функціонал для імітації різних типів бездротових кадрів 802.11 (маяків, кадрів даних, пробових запитів), що відображають типову активність у Wi-Fi мережах;

– впровадити функціонал для моделювання типових атак на бездротові мережі, зокрема DoS-атаки (Deauthentication Flood), з метою демонстрації їх впливу та перевірки механізмів виявлення;

– імплементувати базові алгоритми та правила для ідентифікації аномалій та виявлення імітованих загроз у згенерованому трафіку (наприклад, DoS-атаки, Evil Twin);

– розробити інтуїтивно зрозумілий та функціональний інтерфейс, який дозволить користувачеві керувати симуляцією (запуск, зупинка, ініціація атак), відображати захоплені дані та візуалізувати результати аналізу;

– додати функціонал для графічного представлення результатів аналізу трафіку, таких як розподіл типів кадрів, для наочної демонстрації симульованих подій;

– реалізувати механізм управління потоками для забезпечення асинхронної роботи GUI та тривалих процесів симуляції, що дозволить уникнути «зависання» інтерфейсу.

2 РОЗРОБКА СИМУЛЯТОРА ДЛЯ АНАЛІЗУ МЕРЕЖЕВОГО ТРАФІКУ ТА ВИЯВЛЕННЯ ЗАГРОЗ У БЕЗДРОТОВИХ КОМУНІКАЦІЯХ

2.1 Технічні вимоги до об’єкту професійної діяльності

2.1.1 Найменування і призначення об’єкту професійної діяльності

Симулятор для аналізу мережевого трафіку та виявлення загроз у бездротових комунікаціях, далі Симулятор, призначений для роботи фахівців з кібербезпеки, дослідників та розробників у галузі бездротових технологій. Його основна мета – надати контрольоване середовище для моделювання, аналізу та виявлення потенційних загроз у бездротових мережах, що сприятиме підвищенню їхньої безпеки.

2.1.2 Перелік підсистем, їхнє призначення й основні характеристики, вимоги до числа рівнів ієрархії

Симулятор для аналізу мережевого трафіку та виявлення загроз у бездротових комунікаціях — це комплексна програмна система, що складається з взаємопов'язаних модулів (підсистем). Кожна підсистема виконує специфічні функції, спрямовані на моделювання бездротового середовища, генерацію трафіку, імітацію атак, збір та аналіз даних, а також візуалізацію результатів. Система Симулятора має модульну архітектуру, що дозволяє гнучко додавати нові функціональні можливості та адаптуватися до різних сценаріїв моделювання.

Підсистемами Симулятора є модуль генерації трафіку, що призначений для створення різноманітного бездротового трафіку, який імітує реальні мережеві взаємодії. Його основне призначення — генерування нормального, аномального та шкідливого трафіку для тестування механізмів виявлення загроз. Серед його основних характеристик – підтримка різних протоколів (Wi-Fi, Bluetooth, Zigbee), можливість задання інтенсивності, розподілу та типів даних, а також імітація поведінки легітимних користувачів. Модуль імітації бездротового середовища (МІБС) відповідає за моделювання фізичних властивостей бездротового каналу, таких як затухання сигналу,

перешкоди, багатопроменевість, наявність стін та інших об'єктів, що впливають на поширення радіохвиль. Він створює реалістичне середовище для поширення бездротових сигналів та оцінки їхнього впливу на трафік, дозволяючи налаштовувати параметри середовища (розмір приміщення, матеріали стін, наявність перешкод) та моделювати шуми й інтерференцію.

Модуль імітації атак призначений для відтворення відомих та потенційних атак на бездротові мережі, симулюючи атаки типу DoS (Denial of Service), MitM (Man-in-the-Middle), деаутентифікація, спуфінг, інжекція пакетів та інші. Він дозволяє вибирати тип атаки, конфігурувати її параметри (інтенсивність, ціль, джерело) та підтримує створення власних сценаріїв атак. Модуль збору та попередньої обробки даних (МЗПОД) відповідає за перехоплення, фільтрацію та нормалізацію згенерованого та імітованого трафіку, збираючи необхідні дані для подальшого аналізу та виявлення загроз. Його характеристики включають фільтрацію за протоколами, IP-адресами, MAC-адресами, портами та нормалізацію даних для подальшої обробки алгоритмами машинного навчання. Модуль аналізу та виявлення загроз (МАВЗ) є ядром симулятора, що містить алгоритми для аналізу мережевого трафіку та ідентифікації аномалій і загроз. Його призначення — застосування методів статистичного, сигнатурного аналізу, машинного навчання (наприклад, кластеризація, класифікація) для виявлення шкідливої активності. Він дозволяє інтегрувати різні алгоритми виявлення, налаштовувати порогові значення та формувати правила й сигнатури для виявлення відомих загроз. Нарешті, Модуль візуалізації та звітності (МВЗ) забезпечує графічне представлення результатів симуляції, виявлених загроз та статистичних даних, надаючи користувачеві інтуїтивно зрозумілу інформацію про стан мережі, виявлені атаки та їхній вплив. Його функціонал включає графіки трафіку, таблиці виявлених загроз, теплові карти покриття та можливість експорту звітів у різні формати.

Симулятор розробляється з двома основними рівнями ієрархії. Перший – це рівень управління та конфігурації, який є централізованим інтерфейсом,

що дозволяє користувачеві налаштовувати параметри симуляції, вибирати сценарії атак, керувати модулями та переглядати загальні результати. Цей рівень забезпечує єдину точку контролю над усією системою. Другий — рівень виконання та обробки, представлений децентралізованими модулями, які виконують свої специфічні функції (генерація трафіку, імітація середовища, аналіз даних) паралельно або послідовно відповідно до логіки симуляції. Взаємодія між цими модулями відбувається через стандартизовані інтерфейси. Ступінь централізації системи є високою на рівні управління, що забезпечує зручність використання та консолідований контроль. Однак, на рівні виконання, модулі можуть працювати відносно автономно, обмінюючись даними через визначені API, що підвищує гнучкість та масштабованість Симулятора. Це дозволяє в майбутньому розширювати функціонал, додаючи нові модулі або вдосконалюючи існуючі без значного перероблення всієї архітектури.

2.1.3 Вимоги до способів і засобів зв'язку для обміну інформації між підсистемами

Взаємодія між підсистемами Симулятора повинна базуватися на чітко визначених інтерфейсах і протоколах, забезпечуючи ефективний та надійний обмін даними. Для забезпечення гнучкості та масштабованості системи необхідно використовувати асинхронні механізми обміну інформацією, що дозволяють підсистемам працювати незалежно одна від одної.

Для взаємодії між модулем управління та конфігурації (що, по суті, є фронтендом або центральним контролером) та основними функціональними модулями (генерації трафіку, імітації середовища, імітації атак, аналізу). Це дозволить легко інтегрувати нові модулі та забезпечити сумісність.

Для асинхронної передачі великих обсягів даних, таких як згенерований трафік або результати аналізу, між модулями (наприклад, від модуля генерації трафіку до модуля збору даних, а потім до модуля аналізу). Це забезпечить буферизацію даних та стійкість до тимчасових затримок у

роботі окремих підсистем. Прикладами таких засобів можуть бути Kafka, RabbitMQ або подібні.

Для передачі конфігураційних файлів, великих наборів даних для навчання моделей машинного навчання або збереження логів симуляції.

2.1.4 Вимоги до характеристик взаємозв'язків створюваної мережі із суміжними мережами

Створюваний Симулятор, будучи програмним продуктом, не має «фізичних» суміжних мереж у традиційному розумінні.

Можливість імпортувати реальні зразки мережевого трафіку (наприклад, pcap-файли) для аналізу або використання як основи для генерації трафіку.

Можливість експортувати результати симуляції, виявлені загрози, статистичні дані у стандартизованих форматах (CSV, JSON, PDF) для подальшого аналізу або інтеграції з іншими системами безпеки (SIEM-системами, системами аналізу загроз).

Архітектура повинна дозволяти інтеграцію з зовнішніми бібліотеками та фреймворками для реалізації нових алгоритмів аналізу або типів атак.

2.1.5 Вимоги до режимів функціонування симулятора

Симулятор повинен підтримувати наступні режими функціонування [11]:

– можливість імітації бездротової мережі та трафіку з динамічним виявленням загроз. Цей режим дозволяє спостерігати за процесом симуляції та аналізу в інтерактивному режимі;

– можливість завантаження попередньо записаних pcap-файлів або згенерованих великих обсягів даних для подальшого аналізу. Цей режим корисний для глибокого дослідження та тестування алгоритмів виявлення на історичних даних;

– надання користувачеві інструментів для налаштування параметрів симуляції, вибору типів бездротових мереж, визначення сценаріїв атак та генерації трафіку;

– можливість використання Симулятора для навчання моделей машинного навчання на згенерованих або імпортованих даних, а також для тестування ефективності різних алгоритмів виявлення загроз.

2.1.6 Вимоги до діагностування симулятора

Діагностування працездатності Симулятора та його компонентів повинно включати:

– кожен модуль повинен генерувати детальні логи про свою роботу, включаючи інформацію про помилки, попередження, важливі події, а також про параметри та результати симуляції. Логи повинні бути доступні для перегляду та аналізу;

– симулятор повинен надавати інформацію про споживання системних ресурсів (CPU, RAM, дисковий простір) для кожного модуля, що дозволить виявляти «вузькі місця» та оптимізувати продуктивність.

– модулі повинні здійснювати перевірку цілісності даних, що передаються між ними, для запобігання поширенню некоректної інформації та забезпечення достовірності результатів;

– модуль візуалізації повинен відображати не тільки результати симуляції, але й статус роботи окремих підсистем (наприклад, активність генерації трафіку, кількість оброблених пакетів, виявлені аномалії);

– система повинна оперативно повідомляти користувача про критичні помилки або збої в роботі будь-якого модуля через інтерфейс користувача або за допомогою механізмів сповіщень.

2.1.7 Вимоги до показників призначення

Головною метою Симулятора є надання інструменту для глибокого аналізу мережевого трафіку та ефективного виявлення загроз у бездротових комунікаціях. Він повинен забезпечувати [11]:

– максимально точне відтворення реальних умов бездротового середовища та поведінки мережевого трафіку, щоб результати симуляції були релевантними;

– високий показник виявлення (True Positive Rate) та низький показник хибних спрацювань (False Positive Rate) для різних типів атак.

– здатність обробляти значні обсяги мережевого трафіку та виконувати аналіз за прийнятний час, що особливо важливо для моделювання великих та інтенсивних бездротових мереж;

– інтуїтивно зрозумілий інтерфейс та простота налаштування параметрів симуляції, що дозволяє користувачам ефективно працювати з системою;

– можливість налаштування під різні сценарії використання, протоколи, типи атак та середовища, а також можливість розширення функціоналу.

2.2 Розробка програмної частини

2.2.1 Вибір та обґрунтування засобів розробки

На етапі планування розробки Симулятора аналізу мережевого трафіку та виявлення загроз у бездротових комунікаціях було ретельно визначено та обґрунтовано ключові інструменти, мови програмування та бібліотеки, які ляжуть в основу проєкту. Цей вибір мав вирішальне значення для забезпечення ефективності розробки, гнучкості рішення та його подальшої підтримки.

Основною мовою програмування для реалізації Симулятора було обрано Python. Цей вибір обумовлений декількома значними перевагами. По-

перше, Python має широку підтримку бібліотек, що охоплюють практично всі необхідні аспекти: від мережевого аналізу та GUI-розробки до візуалізації даних, [12]. Це дозволяє значно прискорити процес розробки, оскільки не потрібно створювати функціонал з нуля. По-друге, простота та швидкість розробки на Python є беззаперечними. Висока читабельність коду та низький поріг входження дозволяють швидко прототипувати рішення та легко імплементувати складні функціональні можливості. По-третє, кросплатформність Python гарантує, що Симулятор зможе без проблем функціонувати на різних операційних системах, таких як Windows, Linux та macOS, що розширює його потенційну аудиторію. Нарешті, активна спільнота та докладна документація по Python роблять пошук рішень та навчання новим функціям надзвичайно простим.

Для забезпечення специфічного функціоналу Симулятора було інтегровано декілька ключових бібліотек та фреймворків. Scapy була обрана як основна бібліотека для маніпуляцій з мережевими пакетами. Її призначення полягає у створенні, відправленні, перехопленні та детальному аналізі мережевих пакетів, зокрема бездротових кадрів стандарту 802.11. Scapy дозволяє працювати на різних рівнях мережевої моделі, що є критично важливим для імітації як нормального мережевого трафіку, так і загроз у бездротових комунікаціях. Обґрунтуванням цього вибору є те, що Scapy є фактичним стандартом для низькорівневої роботи з пакетами в середовищі Python, надаючи розробнику безпрецедентну гнучкість, [13].

Для створення графічного інтерфейсу користувача (GUI) була використана стандартна бібліотека Python – Tkinter. Її основне призначення – розробка інтуїтивно зрозумілого інтерфейсу для керування симуляцією, відображення захоплених даних та візуалізації результатів аналізу. Tkinter є вбудованою бібліотекою Python, що означає відсутність додаткових залежностей та простоту розгортання. Вона забезпечує достатній функціонал для створення інтерактивного та зручного користувацького досвіду, [14].

Matplotlib було обрано для візуалізації даних. Ця потужна бібліотека дозволяє будувати різноманітні графіки та діаграми, зокрема кругові діаграми розподілу типів бездротових кадрів, що є важливим для наочного представлення результатів аналізу трафіку. Matplotlib є ефективним інструментом для створення високоякісної статичної, анімованої та інтерактивної візуалізації, і вона добре інтегрується з Tkinter через модуль matplotlib.backends.backend_tkagg, [15].

Нарешті, модуль Threading було включено для роботи з потоками виконання. Його призначення – забезпечити асинхронну роботу GUI та основної логіки симуляції/аналізу. Це необхідно для того, щоб інтерфейс користувача залишався чуйним і не «зависав» під час виконання тривалих операцій, таких як безперервна генерація трафіку та його аналіз. Використання потоків дозволяє виконувати обчислення у фоновому режимі, не блокуючи основний потік GUI, що є критично важливим для комфортної взаємодії з Симулятором.

2.2.2 Архітектура програмного забезпечення

Симулятор аналізу мережевого трафіку та виявлення загроз у бездротових комунікаціях побудований на модульній архітектурі, що є ключовим аспектом його дизайну. Цей підхід передбачає розділення загального функціоналу системи на незалежні, логічно відокремлені та взаємодіючі компоненти. Таке рішення забезпечує високу гнучкість системи, дозволяючи легко додавати нові функції або змінювати існуючі без впливу на інші частини програми. Крім того, модульність сприяє масштабованості, полегшує підтримку коду та робить систему простішою для розширення в майбутньому.

Модуль Графічного Інтерфейсу Користувача (GUI) є основою симулятора, надаючи користувачеві візуальний інтерфейс для взаємодії. Його основне призначення — забезпечити інтуїтивно зрозуміле керування системою, [13]. Модуль GUI включає інтерактивні елементи, такі як кнопки

для керування симуляцією (запуск, зупинка), ініціювання імітації атак, а також для відображення даних та їх візуалізації. Крім того, він містить спеціальні текстові області для виведення журналу виявлених загроз та детальних даних про пакети, що забезпечує користувачеві повну картину того, що відбувається в симульованому середовищі. Розроблений за допомогою бібліотеки Tkinter, цей модуль є чуйним та легким у використанні (рис.2.1).

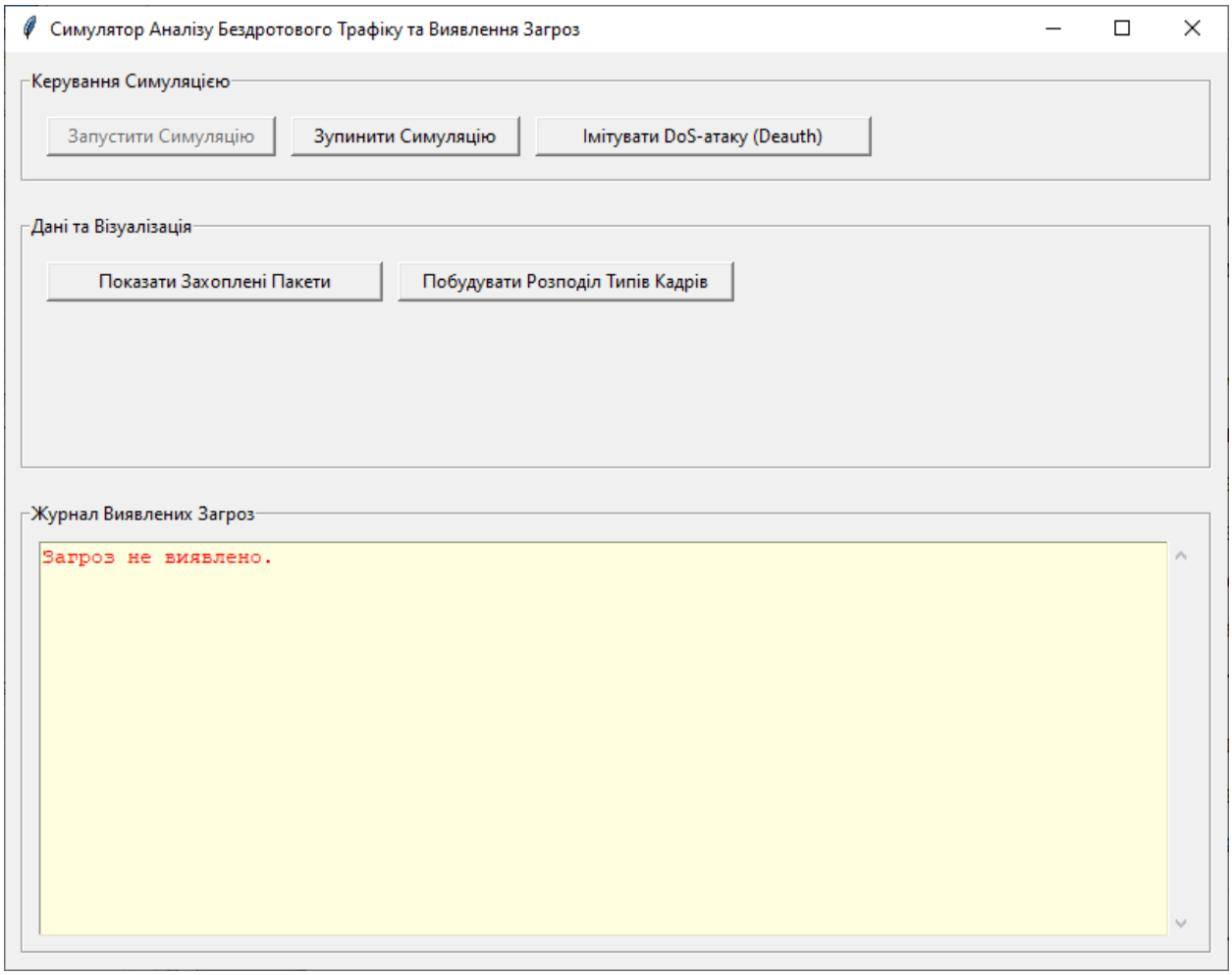

Рисунок 2.1 – Модуль Графічного Інтерфейсу Користувача

Модуль Генерації та Імітації Трафіку що відповідає за створення синтетичного бездротового трафіку. Він імітує нормальну роботу мережі, генеруючи типові бездротові кадри, такі як маяки точок доступу (AP), дані від клієнтів та пробові запити. Важливою функцією цього модуля також є генерація шкідливого трафіку, що дозволяє імітувати різні типи атак, наприклад, деаутентифікаційні фрейми, які є основою для DoS-атак у

бездротових мережах. Модуль використовує бібліотеки Scapy для конструювання пакетів, а також random та time для внесення випадковості та керування швидкістю генерації. Кожен згенерований пакет «відправляється» до Модуля Аналізу та Виявлення Загроз через внутрішню функцію process_simulated_packet(), яка діє як точка перехоплення трафіку.

```
Python 3.13.1 (tags/v3.13.1:0671451, Dec  3 2024, 19:06:28) [MSC v.1942 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> def process_simulated_packet(packet):
...     """
...     Обробляє згенерований або захоплений Scapy пакет.
...     У реальному сценарії тут би відбувався аналіз.
...     """
...     global captured_wireless_packets
...     if packet.haslayer(Dot11):
...         # Зберігаємо пакет для подальшого відображення/аналізу
...         captured_wireless_packets.append(packet)
...         # Одразу аналізуємо на загрози
...         analyze_wireless_packet_for_threats(packet)
```

Рисунок 2.2 – Модуль Генерації та Імітації Трафіку

Модуль Аналізу та Виявлення Загроз – обробляти «захоплені» пакети (тобто ті, що були згенеровані Модулем Генерації) та застосовувати набір правил для ідентифікації потенційних загроз. На поточному етапі реалізовано базові механізми виявлення аномалій, що дозволяють ідентифікувати такі атаки, як DoS-атаки (Deauthentication Flood), що базуються на надмірній кількості деаутентифікаційних кадрів, та атаки типу «Evil Twin» (підроблена точка доступу), що виявляються за невідповідністю між SSID та MAC-адресою точки доступу. Модуль використовує Scapy для парсингу отриманих пакетів та власну Python-логіку для реалізації правил виявлення. Після аналізу, Модуль Аналізу та Виявлення Загроз передає інформацію про виявлені загрози назад до Модуля GUI для негайного відображення (рис.2.3).

Модуль Візуалізації Даних перетворює сирі та агреговані дані на графічні представлення, такі як кругові діаграми розподілу типів бездротових кадрів. Це дозволяє користувачеві швидко оцінити структуру трафіку та виявити аномалії, які можуть вказувати на атаку. Технології Matplotlib та її інтеграція з Tkinter через matplotlib.backends.backend_tkagg забезпечують якісну та зручну візуалізацію. Дані для візуалізації можуть надходити як від

Модуля Аналізу (уже агреговані показники), так і безпосередньо з сховища «захоплених» пакетів.

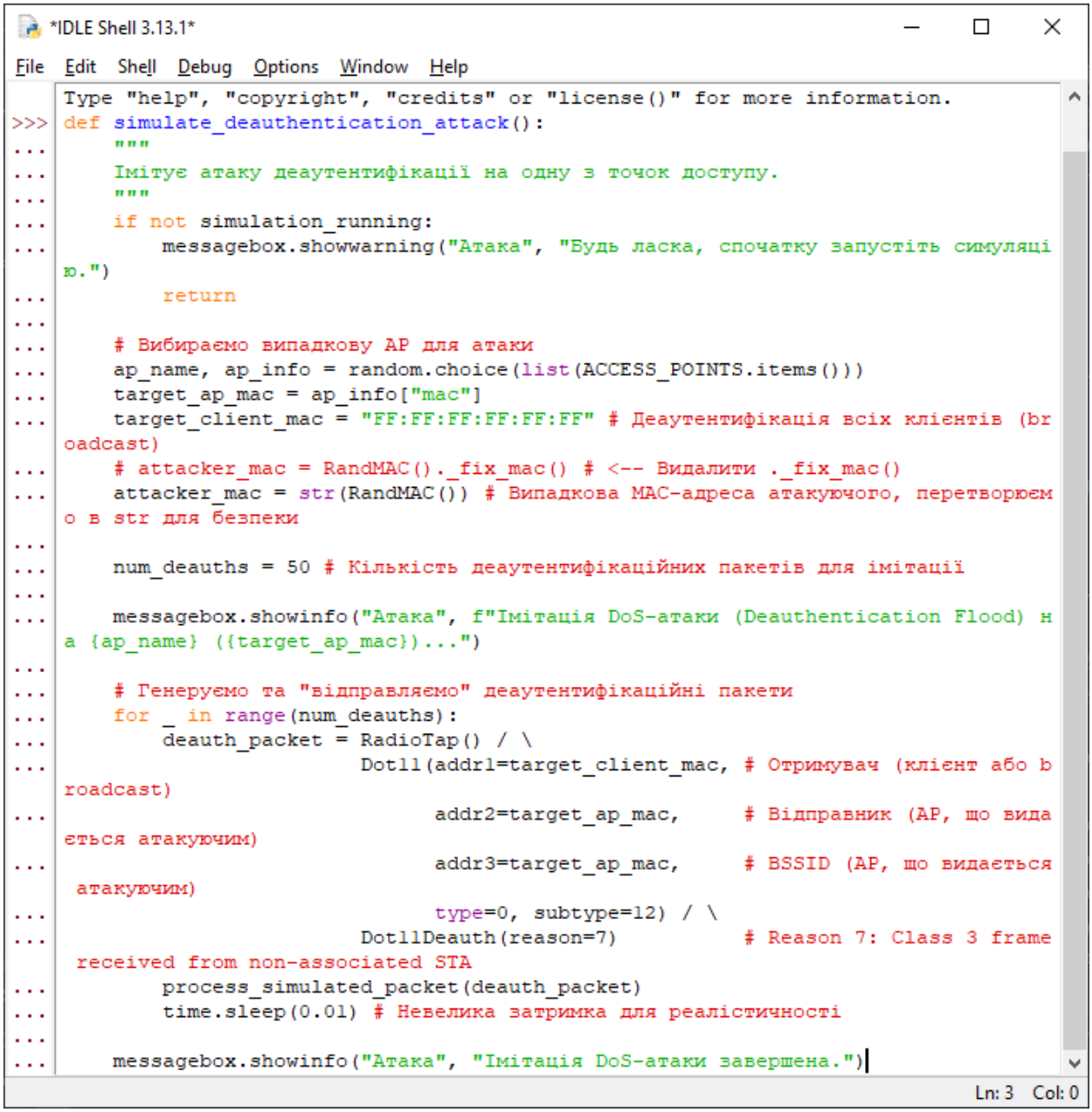

Рисунок 2.3 – Модуль Аналізу та Виявлення Загроз

Модуль Управління Потоками є інфраструктурним елементом, що забезпечує паралельне виконання операцій у Симуляторі. Його ключова функція — керувати фоновими процесами, зокрема запускати функцію генерації трафіку (simulate_and_capture_traffic()) в окремому потоці виконання. Це критично важливо, оскільки імітація трафіку є тривалою операцією, і без використання потоків основний потік GUI був би

заблокований, що призвело б до «зависання» інтерфейсу. Завдяки threading, GUI залишається чуйним, дозволяючи користувачеві взаємодіяти з програмою, поки фонові процеси виконуються.

Загальна схема взаємодії модулів у Симуляторі являє собою послідовність логічних кроків, які забезпечують комплексне функціонування системи (рис.2.4). Процес починається з ініціації роботи, коли користувач взаємодіє з Модулем Графічного Інтерфейсу Користувача (GUI). Це головний елемент, через який здійснюється все керування симулятором.

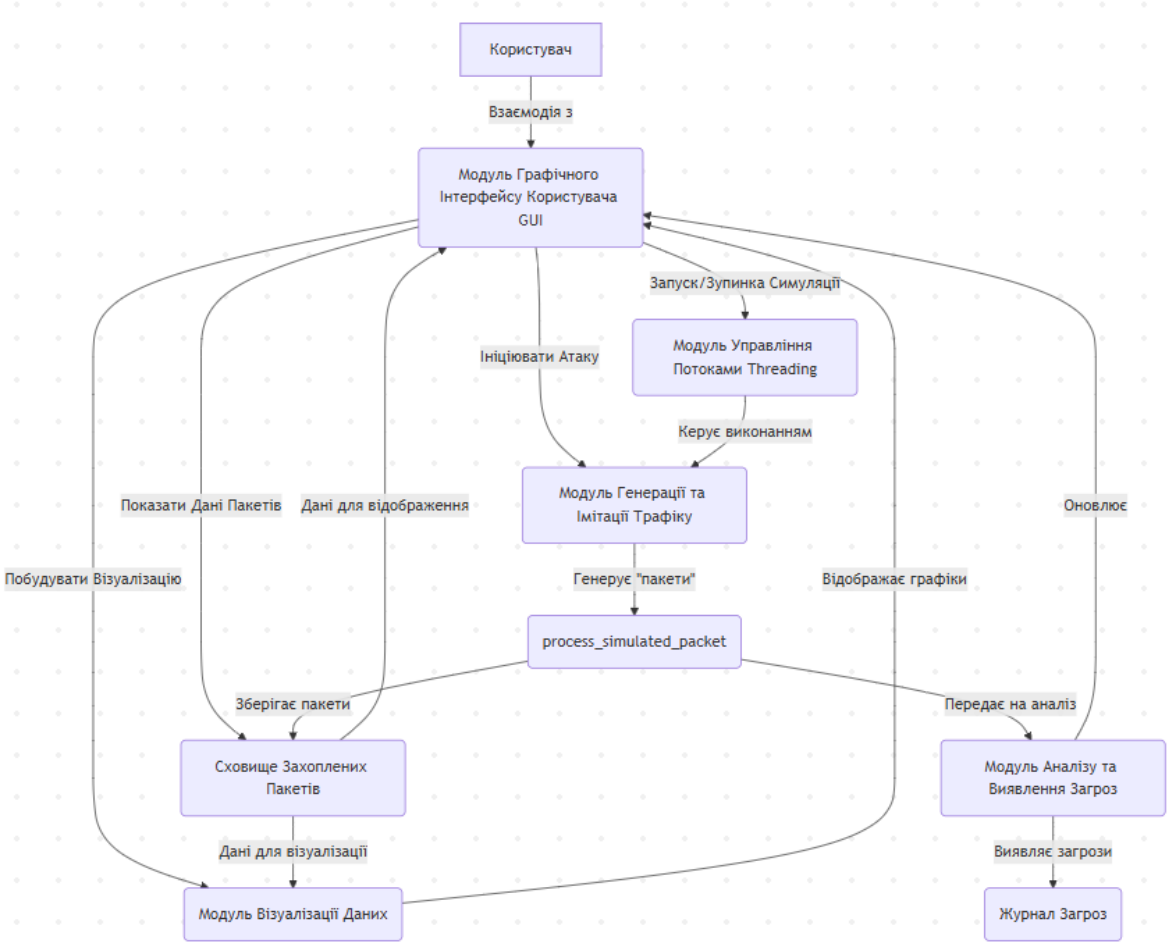

Рисунок 2.4 – Структурна схема взаємодії модулів у Симуляторі

Коли користувач натискає кнопку «Запустити Симуляцію», Модуль GUI ініціює роботу Модуля Управління Потоками (Threading). Останній, у свою чергу, запускає Модуль Генерації та Імітації Трафіку в окремому фоновому потоці. Це дозволяє основному інтерфейсу залишатися чуйним, поки у фоновому режимі відбувається генерація трафіку. Цей модуль

починає створювати синтетичні бездротові пакети, що імітують реальну мережеву активність.

Кожен згенерований пакет негайно «перехоплюється» функцією process_simulated_packet(), яка виступає в ролі точки збору даних. Далі цей пакет передається до Модуля Аналізу та Виявлення Загроз. Основне завдання цього модуля – виявлення загроз: він обробляє отримані пакети, застосовуючи заздалегідь визначені правила для ідентифікації аномалій та потенційних загроз безпеці бездротової мережі.

При виявленні будь-якої загрози, Модуль Аналізу оновлює внутрішній журнал загроз та негайно сповіщає Модуль GUI для відображення відповідних повідомлень у журналі загроз, забезпечуючи оперативну реакцію системи. Симулятор також надає можливість активної взаємодії: користувач може ініціювати атаку, наприклад, «Імітувати DoS-атаку (Deauth)», через Модуль GUI. Цей виклик надсилається безпосередньо до Модуля Генерації та Імітації Трафіку, який починає генерувати шкідливі пакети. Ці пакети, так само як і звичайний трафік, «захоплюються» та передаються на аналіз.

2.2.3 Генерація та імітація бездротового трафіку

Однією з фундаментальних можливостей Симулятора є імітація бездротового мережевого трафіку. Це реалізовано за допомогою функції simulate_and_capture_traffic(), яка виконується в окремому потоці виконання, забезпечуючи безперебійну роботу графічного інтерфейсу. Ця функція циклічно генерує різні типи бездротових кадрів стандарту 802.11, використовуючи потужні можливості бібліотеки Scapy. Серед генерованих типів кадрів:

– маяки (Beacon frames) – ці кадри імітують регулярні оголошення точок доступу (AP), що транслюють свою ідентифікаційну інформацію (SSID) та MAC-адресу. Вони формуються з використанням шарів RadioTap(), Dot11() (з параметрами type=0 для управління та subtype=8 для маяка), Dot11Beacon() та Dot11Elt() для додавання інформації про SSID;

– кадри даних (Data frames) – ці кадри моделюють безпосередній обмін даними між клієнтами та точками доступу. Для спрощення симуляції, ці кадри додатково містять IP- та TCP-шари з базовими IP-адресами та портом, що дозволяє імітувати передачу мережевих даних. Їх структура включає RadioTap(), Dot11() (з type=2 для даних та subtype=0), а також IP(), TCP() та Raw() для корисного навантаження;

– пробові запити (Probe Request) – ці кадри імітують активне сканування мережі клієнтами, які шукають доступні точки доступу. Вони генеруються за допомогою шарів RadioTap(), Dot11() (з type=0 для управління та subtype=4 для пробового запиту) та Dot11ProbeReq.

Кожен згенерований пакет передається на обробку внутрішній функції process_simulated_packet(). Ця функція виконує подвійну роль: вона додає щойно згенерований пакет до глобального списку captured_wireless_packets для подальшого відображення та аналізу, а також негайно викликає функцію аналізу загроз для перевірки пакета на наявність аномалій.

2.2.4 Імітація атаки деаутентифікації

Важливою частиною Симулятора є можливість імітувати мережеві атаки для тестування механізмів виявлення. Функція simulate_deauthentication_attack() моделює поширену атаку «Denial of Service» (DoS) у Wi-Fi мережах, відому як деаутентифікаційний флуд. Ця функція працює наступним чином. Вона випадковим чином обирає одну з попередньо визначених імітованих точок доступу, яка стане ціллю атаки. Далі функція генерує серію (50 пакетів для демонстрації) широкомовних деаутентифікаційних кадрів (Dot11Deauth) за допомогою Scapy. Ці кадри, хоча й імітуються атакуючим, виглядають так, ніби надсилаються з MAC-адреси цільової точки доступу. В результаті такого флуду, клієнти, що підключені до цільової AP, отримують ці деаутентифікаційні пакети і змушені від'єднуватися від мережі, що імітує відмову в обслуговуванні. Пакети створюються з використанням шарів RadioTap(), Dot11() (з type=0 для

управління та subtype=12 для деаутентифікації) та Dot11Deauth() з причиною 7 (Class 3 frame received from non-associated STA), що є стандартним кодом причини для таких атак. Як і звичайний трафік, кожен згенерований деаутентифікаційний кадр також «захоплюється» функцією process_simulated_packet() і передається на аналіз для перевірки ефективності механізмів виявлення загроз.

2.2.5 Аналіз та виявлення загроз

Функція analyze_wireless_packet_for_threats() є відповідальною за ідентифікацію потенційних загроз у бездротовому трафіку. Вона реалізує спрощену, але ефективну логіку виявлення, яка базується на аналізі характерних ознак шкідливої активності. Система розпізнає та реєструє потенційну атаку, якщо виявляє пакет, що є кадром деаутентифікації (type=0, subtype=12), і при цьому його отримувач є широкомовним (FF:FF:FF:FF:FF:FF), або ж відправник цього кадру не є відомою та легітимною точкою доступу. Це вказує на підозрілу активність, що може свідчити про спробу відключити клієнтів від мережі.

Для виявлення атак типу «Зла Двійник» (Evil Twin), функція аналізує маяки (type=0, subtype=8). Якщо виявлений маяк має SSID, який збігається з SSID відомої легітимної точки доступу, але MAC-адреса відправника цього маяка не збігається з MAC-адресою легітимної AP, це інтерпретується як спроба створити підроблену точку доступу для перехоплення трафіку або здійснення фішингових атак.

Усі виявлені загрози додаються до списку threat_log і негайно відображаються у відповідному текстовому полі графічного інтерфейсу користувача, забезпечуючи оперативне сповіщення про небезпеку.

2.2.6 Відображення даних та візуалізація

Симулятор надає користувачеві зручні інструменти для перегляду та аналізу зібраних даних:

– display_packet_data() – ця функція відкриває нове вікно Tkinter, оснащене віджетом scrolledtext. У цьому вікні детально відображається інформація про кожен «захоплений» бездротовий пакет. Користувач може переглянути MAC-адреси відправника та отримувача, тип і підтип кадру 802.11, а також, якщо пакет містить шари вищого рівня (наприклад, IP), відповідні IP-адреси та протоколи (наприклад, TCP, UDP);

– plot_frame_type_distribution() – ця функція використовує бібліотеку Matplotlib для побудови кругової діаграми. Діаграма наочно демонструє розподіл різних типів (Management, Data, Control) та підтипів (Beacon, Deauthentication, Data, Probe Request) кадрів стандарту 802.11 серед усіх «захоплених» пакетів. Це дозволяє швидко оцінити загальну структуру трафіку в симуляції та виявити будь-які аномалії в пропорціях різних типів кадрів, що можуть свідчити про незвичайну активність або атаку.

2.3 Взаємодія з користувачем

Графічний інтерфейс користувача (GUI) Симулятора, розроблений за допомогою бібліотеки Tkinter, є центральним елементом взаємодії між користувачем та програмним забезпеченням. Його дизайн орієнтований на інтуїтивну зрозумілість та зручність використання, забезпечуючи ефективне керування симуляцією та доступ до її результатів. Інтерфейс чітко структурований на три основні функціональні блоки: «Керування Симуляцією», «Дані та Візуалізація» та «Журнал Виявлених Загроз».

Верхня частина інтерфейсу, позначена як «Керування Симуляцією», містить основні елементи для повного контролю над процесом симуляції. Кнопка «Запустити Симуляцію» слугує початковою точкою для активації всього функціоналу симулятора. При її натисканні система ініціює складні внутрішні процеси, зокрема запуск генерації синтетичного бездротового трафіку в окремому фоновому потоці. Це стратегічне рішення дозволяє підтримувати безперервну імітацію мережевої активності без блокування основного інтерфейсу користувача, забезпечуючи плавність та чуйність

роботи програми. Після успішного запуску симуляції, ця кнопка автоматично деактивується (переходить у стан state=tk.DISABLED), щоб запобігти випадковим повторним ініціаціям, тоді як інші елементи керування, залежні від активної симуляції, відповідно активуються. Для завершення активного процесу симуляції передбачена кнопка «Зупинити Симуляцію» (рис.2.5). Її натискання надсилає коректний сигнал до фонового потоку генерації трафіку, забезпечуючи плавне припинення імітації пакетів. Після зупинки симуляції, ця кнопка також деактивується, а кнопка «Запустити Симуляцію» знову стає доступною для нового циклу. Крім того, для тестування та демонстрації можливостей виявлення загроз, інтерфейс включає кнопку «Імітувати DoS-атаку (Deauth)». Ця функція стає активною лише після запуску симуляції. При її активації Симулятор починає генерувати серію шкідливих деаутентифікаційних пакетів, які точно імітують DoS-атаку на бездротову мережу, дозволяючи користувачеві в режимі реального часу спостерігати за реакцією системи на подібні загрози.

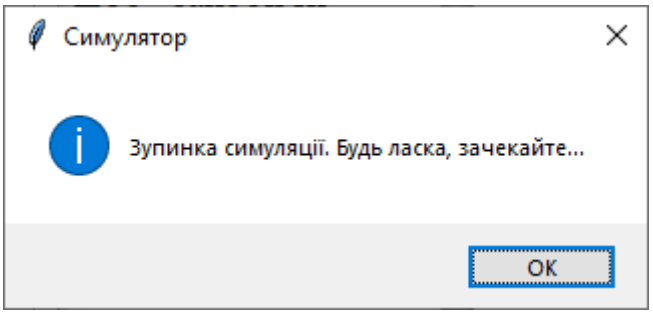

Рисунок 2.5 – Зупинка симулятора

Середня секція інтерфейсу, «Дані та Візуалізація», призначена для надання користувачеві доступу до зібраних даних та їх графічного представлення. Доступ до цих функцій активується лише після запуску симуляції, оскільки вони потребують наявності зібраних даних для обробки. Кнопка «Показати Захоплені Пакети» відкриває нове вікно, оснащене віджетом scrolledtext, що дозволяє прокручувати великі обсяги інформації. У цьому вікні детально відображаються характеристики кожного бездротового пакету, «захопленого» та обробленого симулятором, включаючи MAC-адреси відправника та отримувача, тип і підтип кадру 802.11, а також, у

випадку наявності, IP-адреси та протоколи вищих рівнів (наприклад, TCP/UDP). Це забезпечує можливість глибокого, низькорівневого аналізу трафіку. Для більш наочного аналізу даних передбачена кнопка «Побудувати Розподіл Типів Кадрів». Її активація ініціює створення графічної візуалізації за допомогою бібліотеки Matplotlib. У новому вікні з'являється кругова діаграма, яка візуально відображає пропорційний розподіл різних типів та підтипів бездротових кадрів (таких як Management, Data, Control, Beacon, Deauthentication, Probe Request) серед усіх пакетів, що були «захоплені» під час симуляції. Ця діаграма є цінним інструментом для швидкої оцінки загальної структури трафіку та виявлення аномалій, які можуть вказувати на нетипову або шкідливу активність.

Рисунок 2.6 демонструє вікно «Дані Бездротових Пакетів», що є невід'ємною частиною функціоналу Симулятора і відкривається після натискання кнопки «Показати Захоплені Пакети» в основному графічному інтерфейсі користувача. Це вікно надає деталізований, низькорівневий огляд кожного бездротового кадру, який був «захоплений» — тобто згенерований Симулятором — під час його роботи. Його ключове призначення полягає в забезпеченні прозорості мережевої активності та можливості поглибленого аналізу імітованого трафіку.

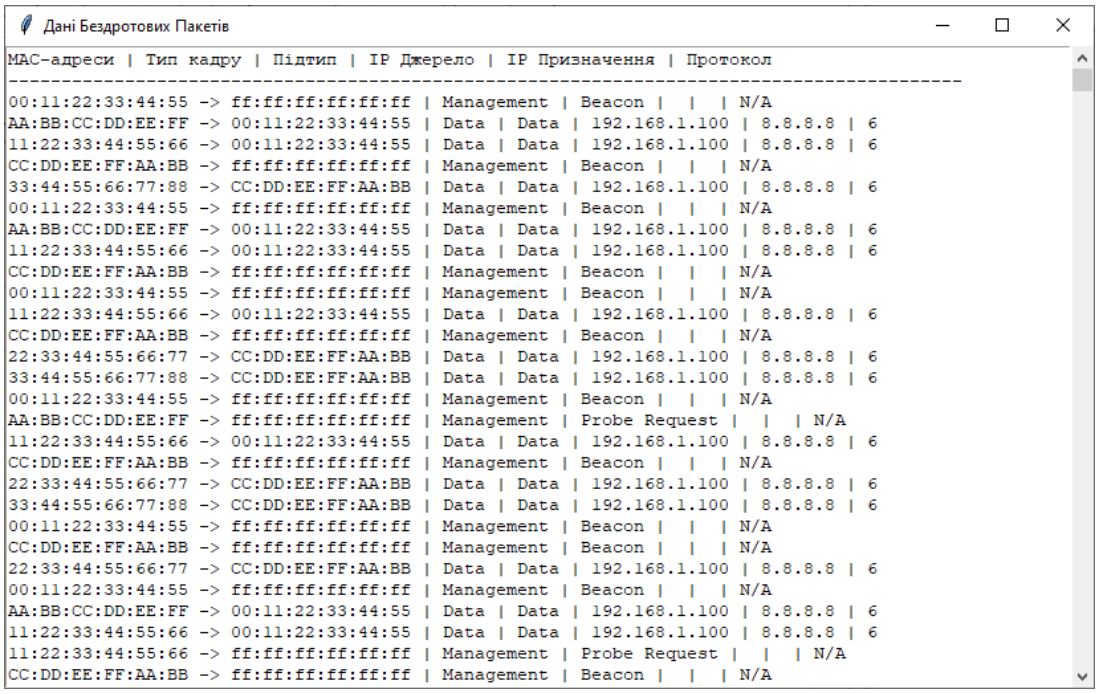

Рисунок 2.6 – Дані Бездротових Пакетів

Структура вікна виконана у вигляді таблиці, що характеризується чітко визначеними стовпцями, кожен з яких надає специфічну інформацію про пакет, сприяючи ефективному візуальному аналізу. Перший стовпець, «MAC-адреси», відображає як MAC-адресу відправника, так і отримувача бездротового кадру, використовуючи формат Source_MAC --> Destination_MAC. Такий запис дозволяє користувачеві швидко визначити напрямок передачі даних: наприклад, 00:11:22:33:44:55 --> ff:ff:ff:ff:ff:ff вказує на широкомовний кадр (коли AP або клієнт надсилає дані всім), тоді як AA:BB:CC:DD:EE:FF --> 00:11:22:33:44:55 свідчить про обмін даними між конкретним клієнтом та певною точкою доступу. Другий стовпець, «Тип кадру», класифікує бездротовий кадр 802.11 за його загальною категорією. Серед найчастіше зустрічаються такі типи, як «Management» (кадри управління, необхідні для встановлення, підтримки та розірвання бездротового зв'язку, наприклад, маяки або пробові запити) та «Data» (кадри даних, які несуть фактичне мережеве навантаження).

Третій стовпець, «Підтип», уточнює функціональне призначення кадру в межах його типу, надаючи більш детальну інформацію. Зі скріншоту видно такі підтипи, як «Beacon» (підтип Management-кадру, що використовується точками доступу для регулярного оголошення своєї присутності), «Data» (підтип Data-кадру, що позначає стандартну передачу даних) та «Probe Request» (підтип Management-кадру, який клієнти використовують для активного пошуку доступних бездротових мереж). Наступні два стовпці, «IP Джерело» та «IP Призначення», є актуальними лише для бездротових кадрів, що містять IP-пакет (характерно для кадрів типу «Data»). У цих випадках відображаються відповідні IP-адреси відправника та отримувача пакета. Наприклад, часто можна побачити 192.168.1.100, що імітує IP-адресу клієнта в локальній мережі, та 8.8.8.8, що є публічним DNS-сервером Google, використовуваним для імітації зовнішнього мережевого трафіку. Для кадрів, які не несуть IP-навантаження (таких як Management-кадри), в цих стовпцях

послідовно відображається «N/A» (Not Applicable). Завершальний стовпець, «Протокол», для кадрів, що містять IP-пакети, вказує номер протоколу на транспортному рівні. На скріншоті часто зустрічається значення 6, що відповідає протоколу TCP. Для кадрів, що не містять IP-навантаження або протоколів вищого рівня, також відображається «N/A».

Рисунок 2.7 ілюструє вікно «Розподіл Типів Кадрів», що є безпосереднім результатом реалізації функціоналу візуалізації в Симуляторі. Ця кругова діаграма візуально та інформативно демонструє пропорційну частку кожного типу та підтипу бездротових кадрів стандарту 802.11, які були «захоплені» або, точніше, згенеровані Симулятором протягом процесу симуляції. Ця візуалізація відіграє ключову роль у швидкому та ефективному виявленні будь-яких аномалій у мережевому трафіку, а також дозволяє оцінити безпосередній вплив імітованих атак на загальну структуру трафіку.

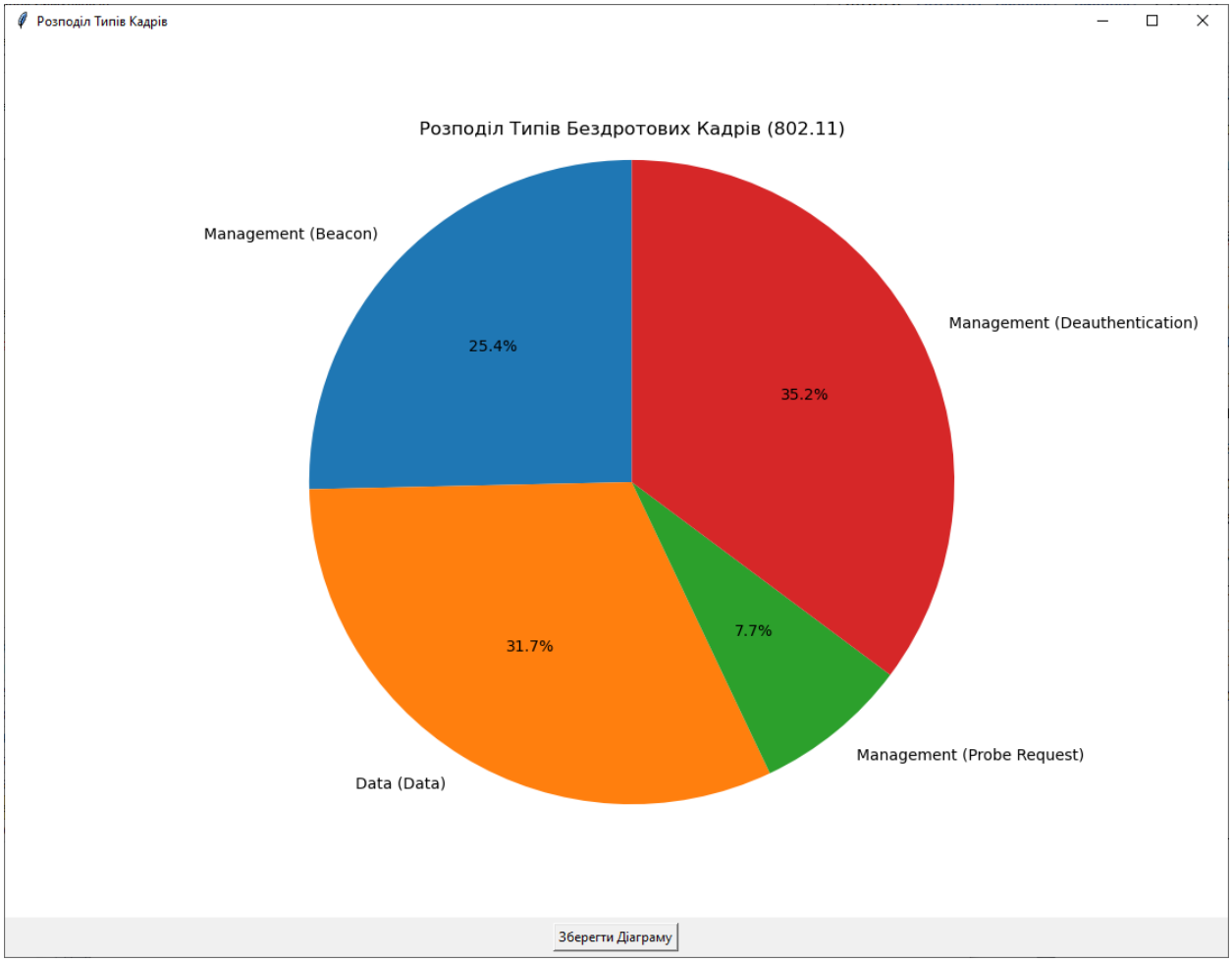

Рисунок 2.7 – Розподіл Типів Кадрів

Діаграма чітко розділена на чотири основні сегменти, кожен з яких репрезентує певну категорію бездротових кадрів та її відносну частку в загальному обсязі трафіку. Найбільший сегмент, що становить 35.2% від усього трафіку, позначений червоним кольором і відповідає кадрам управління типу «Management (Deauthentication)». Така значна частка деаутентифікаційних кадрів є надзвичайно чітким та безумовним індикатором успішної імітації DoS-атаки, а саме атаки типу «Deauthentication Flood». У типовій, нормальній бездротовій мережі кількість подібних кадрів була б вкрай незначною або взагалі відсутньою. Їхня домінуюча присутність підтверджує, що функція імітації DoS-атаки була ефективно активована і змогла успішно інтегрувати ці шкідливі пакети в симульоване мережеве середовище. Це також переконливо свідчить про те, що механізм виявлення загроз у Симуляторі, якщо він коректно налаштований на моніторинг аномально високої кількості цих кадрів, повинен був спрацювати, фіксуючи цю атаку.

Другим за величиною сегментом, що займає 31.7% від загального обсягу трафіку, є помаранчевий сегмент, який відображає кадри даних, або «Data (Data)». Ця частка відображає імітацію «корисного» мережевого навантаження в симульованій мережі, що включає обмін даними між клієнтами та точками доступу. Наявність значної кількості даних підтверджує, що симуляція генерує реалістичний фоновий трафік, який є типовим для функціонуючої бездротової мережі.

Третій сегмент, синього кольору, становить 25.4% від загального обсягу трафіку і представляє кадри управління типу «Management (Beacon)» (маяки). Маяки є фундаментальним елементом функціонування Wi-Fi мереж, оскільки вони використовуються точками доступу для регулярного оголошення своєї присутності та характеристик мережі (наприклад, SSID). Їхня значна частка на діаграмі свідчить про успішну імітацію базових функцій точок доступу, що є невід'ємною частиною реалістичної симуляції.

Нарешті, найменший сегмент, зеленого кольору, складає 7.7% трафіку і відповідає кадрам управління типу «Management (Probe Request)» (пробові запити). Ці кадри надсилаються клієнтами з метою активного пошуку доступних бездротових мереж. Ця частка відображає імітацію звичайної активності клієнтів, які сканують навколишнє середовище. Хоча у певних контекстах аномально велика кількість таких запитів може вказувати на сканування мережі потенційним зловмисником, у рамках даної симуляції ця частка розглядається як частина нормального фонового трафіку.

Нижня частина GUI, «Журнал Виявлених Загроз», є найважливішою для оцінки ефективності системи безпеки. Це спеціалізована текстова область (scrolledtext), призначена для оперативного виведення сповіщень про ідентифіковані системою потенційні загрози. Журнал має контрастний дизайн: яскраво-жовтий фон та червоний текст, що забезпечує негайне привернення уваги користувача до виявлених проблем. Повідомлення про загрози з'являються в ньому в режимі реального часу, як тільки Модуль Аналізу та Виявлення Загроз ідентифікує підозрілу активність (наприклад, ознаки DoS-атак або спроб створення «Evil Twin» точок доступу). Такий миттєвий зворотний зв'язок дозволяє користувачеві оперативно реагувати на симульовані загрози, оцінювати швидкість та точність виявлення, що є критично важливим для навчання та тестування систем безпеки бездротових мереж.

Інформаційне вікно із заголовком «Атака», яке з'являється в Симуляторі під час ініціації відповідної загрози. Це вікно є невід'ємною частиною системи зворотного зв'язку з користувачем, його основне призначення — підтвердити, що команда на імітацію DoS-атаки була успішно прийнята та процес її виконання розпочато (рис.2.8).

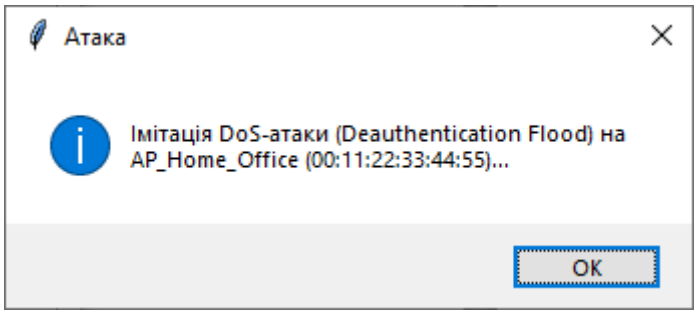

Рисунок 2.8 – Атака

Вікно має стандартний інформаційний дизайн, що забезпечує чітке та швидке розуміння його призначення. Його заголовок «Атака» прямо вказує на контекст поточної дії — імітацію шкідливої мережевої активності. Поруч розташована стандартна синя іконка «і» (information), яка візуально підтверджує, що перед користувачем знаходиться інформаційне повідомлення, а не попередження про помилку чи критичну ситуацію, що дозволяє уникнути непотрібної тривоги.

Центральним елементом вікна є його текстове повідомлення: «Імітація DoS-атаки (Deauthentication Flood) на AP_Home_Office (00:11:22:33:44:55)...». Цей текст є вичерпним і надає користувачеві всю необхідну інформацію про поточну атаку. Фраза «DoS-атаки (Deauthentication Flood)» чітко інформує про конкретний тип імітованої загрози – це атака «відмова в обслуговуванні», яка реалізується шляхом надмірної генерації деаутентифікаційних кадрів, що є характерною ознакою подібних атак у бездротових мережах. Далі, вказівка «на AP_Home_Office» зазначає назву імітованої точки доступу, яка була випадковим чином обрана як ціль для цієї атаки. Це дозволяє користувачеві легко орієнтуватися, яка саме частина симульованої мережі піддається впливу. Додатково, MAC-адреса «(00:11:22:33:44:55)» цільової точки доступу надається для забезпечення максимальної конкретики. Ця інформація дозволяє користувачеві зіставити подію з деталями пакетів, які відображаються у вікні «Дані Бездротових Пакетів», і побачити, як саме шкідливі пакети спрямовані на цю конкретну MAC-адресу. Три крапки «...» в кінці повідомлення слугують візуальним

індикатором того, що процес імітації атаки успішно розпочато і наразі триває.

У нижній частині вікна розташована стандартна кнопка «ОК». Її функція полягає в тому, щоб дозволити користувачеві закрити це інформаційне вікно після ознайомлення з повідомленням. Важливо підкреслити, що закриття цього вікна не впливає на хід імітації атаки; вона продовжується у фоновому режимі, незалежно від стану відображення цього повідомлення.

ВИСНОВКИ

У рамках даної роботи було успішно розроблено та реалізовано Симулятор для аналізу мережевого трафіку та виявлення загроз у бездротових комунікаціях. Цей програмний комплекс є важливим інструментом, який дозволяє ефективно змоделювати функціонування бездротової мережі, імітувати як звичайний мережевий трафік, так і типові атаки, що зустрічаються в Wi-Fi середовищах (наприклад, DoS-атаки та атаки «Evil Twin»). Крім того, він забезпечує демонстрацію механізмів виявлення потенційних загроз у змодельованому трафіку.

Вибір мови програмування Python, разом з використанням потужних і гнучких бібліотек, таких як Scapy для мережевого аналізу, Tkinter для розробки графічного інтерфейсу, Matplotlib для візуалізації даних та Threading для забезпечення багатопоточності, виявився повністю обґрунтованим і раціональним. Це рішення дозволило створити систему, яка відзначається високою гнучкістю, масштабованістю та легкістю у читанні та підтримці коду. Особливу роль у досягненні цих якостей відіграла модульна архітектура симулятора, яка забезпечує чітке розділення функціоналу на взаємодіючі, незалежні підсистеми: Модуль Графічного Інтерфейсу Користувача, Модуль Генерації та Імітації Трафіку, Модуль Аналізу та Виявлення Загроз, Модуль Візуалізації Даних та Модуль Управління Потоками. Такий архітектурний підхід значно спростив процеси розробки, налагодження та заклав міцну основу для подальшого розширення та модернізації системи.

Ключові функціональні можливості симулятора були успішно реалізовані та ефективно продемонстровані в роботі. Генерація та імітація бездротового трафіку дозволяє створювати різноманітні та реалістичні сценарії мережевої активності, включаючи імітацію маяків точок доступу, кадрів даних та пробових запитів, що є фундаментальним для достовірного моделювання поведінки бездротової мережі. Функція імітації DoS-атаки (Deauthentication Flood) успішно відтворює одну з найпоширеніших і

найдеструктивніших загроз у Wi-Fi мережах, що має критичне значення для тестування та демонстрації механізмів безпеки. Аналіз та виявлення загроз ефективно ідентифікує імітовані атаки (DoS-атаки та атаки типу «Evil Twin») на основі заздалегідь визначених правил та логіки, що підкреслює здатність системи реагувати на аномальну та потенційно шкідливу активність. Крім того, можливості відображення даних та візуалізації надають користувачеві наочні та інтуїтивно зрозумілі інструменти для детального аналізу «захоплених» пакетів, а також для статистичного представлення розподілу типів кадрів, що значно підвищує зрозумілість симульованих подій та сприяє швидкій інтерпретації результатів.

Розроблений графічний інтерфейс користувача є інтуїтивно зрозумілим, функціональним та забезпечує повний контроль над усім процесом симуляції — від запуску та зупинки до ініціації конкретних атак та відображення всіх отриманих результатів. Особливе значення має інтегрований журнал виявлених загроз, який надає сповіщення про ідентифіковані аномалії в режимі реального часу, слугуючи важливим елементом зворотного зв'язку та дозволяючи оперативно реагувати на виявлені небезпеки.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Machine Learning Algorithms for Network Intrusion Detection [Електронний ресурс]: Режим доступу: https://link.springer.com/chapter/10.1007/978-3-319-98842-9_6. Дата доступу 01.04.2025

2. What is the difference between functional and non functional requirement? [Електронний ресурс]: Режим доступу: https://stackoverflow.com/questions/16475979/what-is-the-difference-between-functional-and-non-functional-requirement. Дата доступу 01.4.2025

3. Abdi H., Williams L.J. (2010). Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics, 2: 433–459.

4. Korniyenko B., Yudin A., Galata L. Risk estimation of information system.Wschodnioeuropejskie Czasopismo Naukowe. 2016. № 5. P. 35 - 40.

5. H. Alaidaros i M. Mahmuddin, «Flow-Based approach on Bro Intru-Sion Detection» Journal of Telecommunication, Electronic and Computer Engineering (JTEC), vol. 9, № 2-2, С. 139-145, 2017

6. Support-vector networks [Електронний ресурс]: Режим доступу: https://link.springer.com/article/10.1007/BF00994018/. Дата доступу 01.04.2025

7. Bilwaj Gaonkar, Christos Davatzikos «Analytic estimation of statistical significance maps for support vector machine based multi-variate image analysis and classification», April 2013.

8. Ben-Hur, Asa, Horn, David, Siegelmann, Hava, and Vapnik, Vladimir; «Support vector clustering» (2001) Journal of Machine Learning Research, 2: 125–137

9. Statnikov, A., Hardin, D., & Aliferis, C., «Using SVM weight-based methods to identify causally relevant and non-causally relevant variables». 2006.

10. Piryonesi S. Madeh; El-Diraby Tamer E. (2020-06-01). «Role of Data Analytics in Infrastructure Asset Management: Overcoming Data Size and Quality Problems». Journal of Transportation Engineering, Part B: Pavements.

11. Комп’ютерні мережі: [навчальний посібник] / А. Г. Микитишин, М. М. Митник, П. Д. Стухляк, В. В. Пасічник. — Львів: «Магнолія 2006», 2013. — 256 с. ISBN 978-617-574-087-3

12. Олексій В. Програмування мовою Python. Навчальна книга – Богдан, 2019. 504 с.

13. Ерік М. Пришвидшений курс Python. Практичний, проєктно-орієнтований вступ до програмування. Видавництво Старого Лева, 2021. 600 с.

14. Robbins P. Python Programming for Beginners. Independently published, 2023. 114 p.

15. Barry P. Head First Python: A Brain-Friendly Guide Paperback. O'Reilly Media, 2016. 584 p.

ДОДАТОК А

Текст програмного коду

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ДНІПРОВСЬКА ПОЛІТЕХНІКА»

ПРОГРАМНИЙ КОД

Текст програми

804.02070743.24002-01 12 01

Листів 5

2025

АНОТАЦІЯ

Дана програма містить в собі програмний код створення Симулятор, для аналізу мережевого трафіку та виявлення загроз у бездротових комунікаціях. Цей програмний комплекс є важливим інструментом, який дозволяє ефективно змоделювати функціонування бездротової мережі, імітувати як звичайний мережевий трафік, так і типові атаки, що зустрічаються в Wi-Fi середовищах (наприклад, DoS-атаки та атаки «Evil Twin»). Крім того, він забезпечує демонстрацію механізмів виявлення потенційних загроз у змодельованому трафіку.

ЗМІСТ

Стор.

Фрагмент коду програми

```
import tkinter as tk
from tkinter import scrolledtext
import matplotlib.pyplot as plt
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
from scapy.all import *

# Global variables
packet_data = []  # To store captured packet data

# Function to capture and analyze packets
def capture_packets():
    global packet_data
    packet_data.clear()  # Clear previous packet data
    interface = «eth0»  # Specify the network interface to capture packets (e.g., eth0, wlan0)

    def packet_handler(packet):
        # Analyze packet and extract relevant information
        src_ip = packet[IP].src
        dst_ip = packet[IP].dst
        protocol = packet[IP].proto
        packet_data.append((src_ip, dst_ip, protocol))

    # Start capturing packets on the specified interface
    sniff(iface=interface, prn=packet_handler, store=0)

# Function to display packet data in the GUI
def display_packet_data():
```

```
    # Create a new window for displaying packet data
    data_window = tk.Toplevel(root)
    data_window.title(«Packet Data»)

    # Create a scrolled text widget to show packet data
    text_area = scrolledtext.ScrolledText(data_window, width=80, height=20)
    text_area.pack(expand=True, fill='both')

    # Insert packet data into the text area
    for packet in packet_data:
        text_area.insert(tk.END, f»Source IP: {packet[0]}, Destination IP: {packet[1]}, Protocol: {packet[2]}\n»)

# Function to plot a simple pie chart of protocol distribution
def plot_protocol_distribution():
    protocols = [packet[2] for packet in packet_data]
    unique_protocols = list(set(protocols))
    protocol_counts = [protocols.count(proto) for proto in unique_protocols]

    # Plotting the pie chart
    fig, ax = plt.subplots()
    ax.pie(protocol_counts, labels=unique_protocols, autopct='%1.1f%%')
    ax.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
    ax.set_title(«Protocol Distribution»)

    # Display the plot in the GUI
    chart_window = tk.Toplevel(root)
    chart_window.title(«Protocol Distribution»)
    chart_canvas = FigureCanvasTkAgg(fig, master=chart_window)
    chart_canvas.draw()
```

```
    chart_canvas.get_tk_widget().pack()

# Create the main GUI window
root = tk.Tk()
root.title(«Network Traffic Analysis Tool»)

# Create buttons for capturing packets, displaying packet data, and plotting protocol distribution
capture_button = tk.Button(root, text=«Capture Packets», command=capture_packets)
capture_button.pack(pady=10)

display_button = tk.Button(root, text=«Display Packet Data», command=display_packet_data)
display_button.pack(pady=10)

plot_button = tk.Button(root, text=«Plot Protocol Distribution», command=plot_protocol_distribution)
plot_button.pack(pady=10)

# Start the main event loop
root.mainloop()
```