Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук
(повна назва)
Кафедра програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів формування датасета для розпізнавання зловмисного трафіку
(тема)

Виконав:
здобувач 2 року навчання
групи ІПЗм-23-4
Максим ШУЛЬДІНЕР
(власне ім'я, прізвище)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Інженерія програмного забезпечення
(повна назва освітньої програми)

Керівник проф. Ігор ШОСТАК
(посада, власне ім'я, прізвище)

Допускається до захисту
Зав. кафедри _____ Кирило СМЕЛЯКОВ
(підпис) (власне ім'я, прізвище)
2025 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук

Кафедра програмної інженерії

Рівень вищої освіти другий (магістерський)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва)

Тип програми освітньо-наукова програма
(освітньо-професійна або освітньо-наукова)

Освітня програма Інженерія програмного забезпечення
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«___» _____20___ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Шульдінеру Максиму Павловичу
(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження методів формування датасета для розпізнавання зловмисного трафіку».
   затверджена наказом університету від 15 квітня 2025 р. № 290 Ст
2. Термін подання здобувачем роботи до екзаменаційної комісії 24 червня 2025 р.
3. Вихідні дані до роботи «*Публічні датасети, реальний мережевий трафік з атак (DoS, DDoS, Brute Force, Heartbleed, SQL Injection, Port Scanning), симулятори зловмисного трафіку, методики збору даних: лог-файли, потоки даних із мережевих пристроїв, сервіси моніторингу трафіку, методи передобробки даних: фільтрація, нормалізація, агрегування, критерії для побудови високоякісного датасету: різноманітність атак, точність збору, адаптивність до нових загроз, методики порівняння: точність збору даних, дані з датасету CIC-IDS2017 для тренування та тестування моделей, алгоритми машинного навчання для класифікації зловмисного трафіку».*

Перелік питань, що потрібно опрацювати у роботі «*аналіз предметної області та аналіз публічних датасетів для виявлення зловмисного трафіку, дослідження реального мережевого трафіку з атаками, використання симуляторів зловмисного трафіку, методики збору даних з лог-файлів та мережевих пристроїв, методи передобробки даних».*

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі і постановка задачі	21.01.25 – 01.02.25	виконано
2	Огляд й аналіз літературних, наукових джерел	02.02.25 – 04.03.25	виконано
3	Підготовка до апробації результатів дослідження. Публікація матеріалів	05.03.25 – 10.03.25	виконано
4	Теоретичне дослідження	11.03.25 – 01.05.25	виконано
5	Практична реалізація	02.05.25 – 30.05.25	виконано
6	Підготовка пояснювальної записки.	01.06.25 – 14.06.25	виконано
7	Підготовка презентації та доповіді	14.06.25 – 15.06.25	виконано
8	Перевірка на плагіат	15.06.25 – 16.06.25	виконано
9	Нормоконтроль	18.06.25 – 19.06.25	виконано
10	Рецензування	19.06.25 – 23.06.25	виконано
11	Попередній захист	23.06.25 – 24.06.25	виконано
12	Занесення диплома в електронний архів	24.06.25 – 25.06.25	виконано
13	Допуск до захисту у зав. кафедри	26.06.25	виконано

Дата видачі завдання 20 січня 2025р.

Здобувач _____
(підпис)

Керівник роботи _____ проф. Шостак І.В
(підпис) (посада, прізвище, ініціал

РЕФЕРАТ / ABSTRACT

Пояснювальна записка містить: 72 с., 19 рис., 5 лістингу. 14 джерел.

АНАЛІЗ ДАНИХ, АРХІТЕКТУРА ОБРОБКИ ЗЛОВМИСНОГО ТРАФІКУ, ДОСЛІДЖЕННЯ КІБЕРЗАГРОЗ, ГЛИБИННЕ НАВЧАННЯ, МОДЕЛІ НАПАДІВ, МОДЕЛІ ВИВЧЕННЯ ТРАФІКУ, СИСТЕМИ ВИЯВЛЕННЯ ЗЛОВМИСНОГО ТРАФІКУ.

Об’єктом дослідження є процес створення датасету для навчання моделей виявлення атак у мережевому середовищі.

Метою роботи є розробка власного датасету для аналізу мережевого трафіку з акцентом на виявлення шкідливих пакетів, зокрема SYN-флуд атак, та підготовка цього датасету для подальшого використання у процесі навчання моделей машинного навчання для виявлення атак.

Методи дослідження: системний та порівняльний аналіз, розробка програмних засобів для збору та обробки мережевого трафіку, машинне навчання для класифікації мережевих атак.

У результаті роботи були вивчені основні типи атак на мережі, зокрема DDoS, SQL-ін’єкції та фішинг, а також створено датасет для виявлення атак.

ANALYSIS OF DATA, ARCHITECTURE OF MALICIOUS TRAFFIC PROCESSING, CYBER THREAT RESEARCH, DEEP LEARNING, ATTACK MODELS, TRAFFIC LEARNING MODELS, MALICIOUS TRAFFIC DETECTION SYSTEMS.

The object of the research is the process of creating a dataset for training attack detection models in a network environment.

The goal of the work is to develop a custom dataset for network traffic analysis with an

emphasis on detecting malicious packets, specifically SYN flood attacks, and to prepare this dataset for further use in training machine learning models for attack detection.

Research methods: systems and comparative analysis, software development for collecting and processing network traffic, machine learning for classifying network attacks.

As a result of the work, the main types of network attacks, including DDoS, SQL injections, and phishing, were studied, and a dataset for attack detection was created.

Заява щодо самостійного виконання кваліфікаційної роботи та можливості її публікації в електронному архіві відкритого доступу ElArKhNURE.

Завідувачу кафедри
ПІ_____
(скорочена назва кафедри)
проф. Кирилу СМЕЛЯКОВУ_
(вчене звання, сласне ім'я, прізвище)

ЗАЯВА
щодо самостійності виконання кваліфікаційної роботи та можливості її публікації (та/або публікації анотації кваліфікаційної роботи) в електронному архіві відкритого доступу ElAr KhNURE

Я, _____Шульдінер Максим Павлович_____
(прізвище, ім'я, по батькові)

здобувач вищої освіти на другому (магістерському) рівні вищої освіти академічної групи _____ІПЗм-23-4_____

кафедра_____ програмної інженерії _____,
(повна назва кафедри)

заявляю: моя кваліфікаційна робота на тему _Дослідження методів формування датасета для розпізнавання зловмисного трафіку_____
_____
(назва роботи)

що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в репозиторії "ElArKhNURE". Погоджуюся з авторським договором, відповідно до Положення про репозиторій ХНУРЕ "ElArKhNURE". Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений (а) з вимогами академічної доброчесності, згідно з якими виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

Дата 23.06.2025 Підпис

ЗМІСТ

ВСТУП

У сучасному цифровому світі кіберзагрози постійно еволюціонують, підвищуючи ризики для цілісності та конфіденційності даних. Зловмисники використовують нові методи атак, такі як розповсюдження шкідливого програмного забезпечення через зашифровані канали, сканування портів або атаки на командні центри (C&C). Ці проблеми набувають особливої гостроти у контексті реалій 2022–2025 років, коли кібервійни стають частиною гібридних конфліктів.

Створення надійних датасетів для розпізнавання зловмисного трафіку стає ключовою основою для розробки ефективних систем виявлення вторгнень. Особливо цікавим є дослідження CIC-IDS2017 датасету[1], який детально описує різні типи атак (DoS, порт-сканування, фішинг тощо) і широко використовується в науковій спільноті. Проте, у сучасних умовах з'являється необхідність оновлення таких ресурсів та адаптації їх під новітні реалії кіберзагроз.

Наукова новизна дослідження полягає у розробці підходу до створення актуального датасету для розпізнавання зловмисного трафіку, що враховує сучасні реалії загроз. Також досліджується вплив якості датасету на точність моделей машинного навчання.

Об’єктом дослідження є процес виявлення зловмисного трафіку в мережевих середовищах, а предметом – методи створення та використання датасетів для цього завдання. Вибір саме цієї тематики обумовлений потребою у створенні високоякісних і адаптованих до сучасних умов ресурсів, які забезпечать надійність систем виявлення атак. Для виконання дослідження застосовувалися такі методи: аналіз наукової літератури та існуючих підходів до створення датасетів, емпіричний аналіз ефективності моделей глибокого навчання при використанні різних датасетів, розробка прототипу нового датасету та його тестування в умовах, наближених до реальних.

Результати роботи можуть бути використані для покращення систем виявлення вторгнень, оптимізації процесу збору та аналізу мережевого трафіку, а також створення більш надійних систем кіберзахисту в організаціях.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1. Підходи до створення датасетів для розпізнавання зловмисного трафіку: аналіз, виклики та перспективи

Дослідження в сфері кібербезпеки, зокрема виявлення зловмисного трафіку, вимагає створення відповідних датасетів для тренування моделей машинного навчання. Якість датасету безпосередньо впливає на ефективність детекції атак, тому важливим є ретельний вибір методів для його формування. У цьому контексті важливо розглянути порівняльну характеристику різних підходів до створення датасетів для розпізнавання зловмисного трафіку.[2]

Першим кроком у створенні датасету є вибір джерела даних. Існує кілька можливих підходів: використання публічно доступних наборів даних, зібраних за допомогою власних систем моніторингу або генерування трафіку через симуляції. Публічні набори даних, такі як KDD Cup 1999, CICIDS 2017 та UNSW-NB15, широко використовуються в дослідженнях завдяки їх доступності та документації. Вони дозволяють порівнювати результати різних моделей і підходів, однак часто мають обмеження, такі як старість або недостатня різноманітність атак.

Для покращення реалістичності сучасних досліджень використовуються дані, зібрані з реальних мереж. Цей метод забезпечує більше відповідності реальним умовам, але він вимагає налаштування складних систем збору даних, зокрема застосування програмного забезпечення для моніторингу мережевого трафіку[3]. До того ж, збір таких даних може бути обмежений юридичними чи етичними аспектами, адже використання реального трафіку може порушувати конфіденційність або права користувачів.

Для створення більш різноманітних датасетів деякі дослідники застосовують симулятори, які генерують зловмисний трафік в лабораторних умовах. Ці симулятори можуть імітувати різноманітні типи атак, зокрема DDoS, SQL-ін'єкції або фішинг[4]. Використання таких інструментів дозволяє контролювати кількість, тип і складність атак, що в свою чергу покращує якість тренувальних даних для машинного навчання. Однак цей метод також має свої недоліки, оскільки

симульований трафік може не відображати всі особливості реальних атак, що знижує здатність моделей правильно класифікувати нові, невідомі типи атак.

Іншим важливим етапом є анотація даних, яка включає позначення кожного запису як «нормальний» або «зловмисний». Вибір методу анотації залежить від типу даних та джерела їх отримання. Для даних, зібраних з реальних мереж, це може включати ручну перевірку кожного запису або автоматичне маркування на основі відомих шаблонів атак. Проте автоматичні методи анотації часто схильні до помилок, тому важливо, щоб перевірка результатів здійснювалася на високому рівні.[5]

Порівняння методів створення датасетів має також враховувати різноманітність типів атак. Існує потреба в широкому спектрі атак для того, щоб моделі могли навчатися не лише на найпоширеніших загрозах, але й на рідкісних, нових методах атак. З цієї точки зору комбінування публічних і приватних джерел даних є важливим для створення всебічного набору даних, який дозволить покрити всі можливі варіанти загроз.

Проте жоден з методів не є ідеальним. Наприклад, публічні датасети, хоча і корисні, застаріли або не відображають сучасні технології атак. Реальні дані можуть бути обмеженими або застарілими. Тому важливо комбінувати ці підходи і активно адаптувати методи створення даних до нових вимог і умов. Розробка нових стратегій для збору та генерації трафіку також є важливим кроком для покращення результатів у цій галузі.

Завершення процесу створення датасету часто включає в себе етап перевірки якості даних. Оцінка якості даних допомагає виявити можливі неточності або відсутність важливих даних. Для цього можуть бути використані статистичні методи, аналіз класифікації за допомогою уже навченої моделі або інші алгоритми, які дозволяють оцінити збалансованість та репрезентативність вибірки.

1.2 Оцінка ефективності попередніх рішень та викликів у створенні датасетів для виявлення зловмисного трафіку

За результатами огляду, важливо не лише інтерпретувати масштаб проблеми, але й оцінити попередні рішення, що були застосовані для розпізнавання зловмисного трафіку. Оскільки технології постійно змінюються, ключовим є виявлення того, як попередні підходи змогли реагувати на нові типи атак, а також наскільки вони ефективні у сучасних умовах. Це включає аналіз якості попередніх датасетів, використаних методів збору даних, а також застосування моделей машинного навчання для класифікації трафіку.

Дослідження в цій області традиційно зосереджувалися на відомих типах атак, таких як DDoS або SQL-ін'єкції, але з розвитком технологій зловмисники почали використовувати нові стратегії, що ставлять під сумнів ефективність старих методів. Попередні рішення часто мали обмеження в здатності виявляти складні чи невідомі загрози. Тому одним із поточних викликів є вдосконалення моделей для обробки нових типів атак, що швидко еволюціонують.

Визначення поточних викликів є невід’ємною частиною процесу, адже саме вони допомагають сфокусувати дослідження на конкретних проблемах, які потребують вирішення. Зокрема, необхідно звернути увагу на такі аспекти, як зростаюча складність атак, високий обсяг трафіку в реальному часі та необхідність мінімізації помилкових спрацьовувань (false positives). Іншим викликом є підтримка високої точності моделей при роботі з великими, нерівномірними та неповними даними, що є характерним для реального світу.

Встановлення рівня інноваційності або практичності дослідження дозволяє оцінити, наскільки розроблені підходи можуть бути застосовані в реальних умовах. Інноваційність полягає в тому, щоб запропонувати нові методи або удосконалити існуючі підходи, які дозволяють не тільки підвищити ефективність виявлення атак, але й оптимізувати процеси збору та обробки даних. З іншого боку, практичність дослідження оцінюється через те, чи можливо втілити ці методи на реальних

інфраструктурах з обмеженими ресурсами, зокрема в умовах високого навантаження на мережу.

Сьогодні все більш важливим стає використання глибокого навчання для автоматичного виявлення аномалій у трафіку. Однак цей підхід має як сильні, так і слабкі сторони. Наприклад, глибокі нейронні мережі можуть бути надзвичайно ефективними у виявленні складних атак, але вони також потребують великих обсягів даних для тренування і значних обчислювальних ресурсів. Тому одним із викликів є знайти баланс між інноваційністю методів та їх практичною застосовуваністю в реальних умовах.

Не менш важливим є врахування інтерпретованості результатів роботи моделей машинного навчання. Хоча глибоке навчання дає вражаючі результати, його "чорний ящик" може бути проблемою для безпеки, оскільки інтерпретація рішень моделі може бути ускладнена. Інтерпретованість результатів має вирішальне значення для подальшої корекції моделей та забезпечення їх ефективності в реальних умовах, що підвищує рівень довіри до автоматизованих систем виявлення атак.

З огляду на ці фактори, інноваційні методи створення датасетів та покращення моделей для виявлення зловмисного трафіку повинні поєднувати теоретичні дослідження з практичною реалізацією, яка дозволяє їх застосування в умовах сучасних кіберзагроз.

У цьому контексті важливо розглядати використання методів, які дозволяють пояснити рішення, що приймаються моделями, наприклад, через LIME або SHAP, які намагаються візуалізувати вплив кожної ознаки на прогноз. Це дозволяє знижувати ризики помилкових рішень, підвищувати довіру до системи, а також робити її більш адаптованою до змінних умов.

Одним із напрямків майбутніх досліджень є розробка гібридних моделей, що поєднують переваги різних підходів, таких як традиційне машинне навчання і глибоке навчання. Це дозволить створити більш стабільні моделі для виявлення зловмисного трафіку, здатні ефективно працювати в умовах великих і складних набір даних. Крім того, важливим аспектом є інтеграція новітніх підходів до

обробки і аналізу даних, таких як аналіз на основі графів або використання методів навчання з підкріпленням для постійного вдосконалення моделей без необхідності великої кількості міток для навчання.

Загалом, оптимізація процесу створення датасетів для розпізнавання зловмисного трафіку є важливим етапом для досягнення високої ефективності виявлення атак. Створення збалансованих, різноманітних та актуальних наборів даних є фундаментальним для розробки моделей, які здатні відповідати на виклики сучасного кіберпростору. Врахування різних аспектів, таких як якість даних, типи атак, методи анотації та інтерпретованість моделей, дозволить значно покращити результативність таких систем, забезпечивши надійний захист від нових і складних загроз.

1.3 Багатокритеріальний підхід до аналізу методів створення датасетів для виявлення зловмисного трафіку

У рамках даного дослідження буде здійснено порівняльний аналіз різних методів створення датасетів для розпізнавання зловмисного трафіку з використанням багатокритеріального підходу, зокрема на основі даних з датасету CIC-IDS2017[1]. Застосовуватиметься лінійне згортання з ваговими коефіцієнтами для оцінки ефективності кожного з методів.

Передбачається, що процес створення датасету для задачі виявлення зловмисного трафіку включатиме:

Аналіз методів збору даних: у дослідженні буде використовуватися мережевий трафік з реальних атак, таких як DoS, DDoS, Brute Force, Heartbleed, SQL Injection, Port Scanning та інші.[6] Оцінка таких методів, як лог-файли, потоки даних із мережевих пристроїв і сервіси моніторингу трафіку, дозволить виявити переваги та обмеження кожного підходу щодо збору даних для подальшого аналізу.

Оцінка методів передобробки даних: будуть застосовуватися методи фільтрації, нормалізації та агрегування даних, щоб знизити шуми та покращити ефективність виявлення аномалій на основі даних з датасету CIC-IDS2017.

Передобробка даних включатиме перетворення мережевих характеристик у зручний формат для застосування в алгоритмах машинного навчання.

Розробка критеріїв для побудови високоякісного датасету: з огляду на потреби моделювання глибокого навчання та інших алгоритмів для класифікації зловмисного трафіку, будуть визначені критерії для побудови датасету, що включає різноманітні типи атак.

Порівняння методів створення датасетів: на основі ключових параметрів, таких як точність збору даних, витрати на створення, здатність адаптуватися до нових видів атак, масштабованість та продуктивність, буде здійснено порівняння методів створення датасетів.

Визначення найбільш ефективних підходів для створення датасетів: на основі проведеного аналізу буде визначено, які підходи до створення датасетів є найбільш ефективними для подальшого використання в задачах машинного навчання. Це включатиме підготовку даних для тренування та тестування моделей для виявлення зловмисного трафіку, використовуючи методи, що застосовуються в датасеті CIC-IDS2017.

Метою дослідження є створення теоретичної основи для розробки методів збору та обробки даних, що дозволяють ефективно вирішувати задачу розпізнавання зловмисного трафіку з урахуванням багатокритеріальних вимог, таких як швидкість обробки, точність та адаптивність до нових загроз.

2 ОГЛЯД Й АНАЛІЗ ЛІТЕРАТУРНИХ, НАУКОВИХ ДЖЕРЕЛ

Огляд основних джерел для дослідження створення датасетів для розпізнавання зловмисного трафіку базується на кількох важливих критеріях. Авторитетність джерел була визначена через публікації в рецензованих журналах і конференціях, що мають високу репутацію у науковому середовищі, таких як IEEE та Springer. Актуальність матеріалів також була критичним фактором, оскільки дослідження з цієї теми постійно оновлюються у зв'язку зі швидким розвитком методів атак і необхідністю адаптації до нових умов. Об'єктивність була важливим аспектом, оскільки аналіз різних підходів до створення датасетів дозволяє отримати більш збалансоване уявлення про проблему, а достовірність джерел підтверджувалась високим рівнем рецензування та перевірки даних.

У результаті було обрано декілька основних тем для дослідження, серед яких методи збору даних, типи атак, анонімізація даних, синтетичні дані та використання метаданих. Методологія збору трафіку поділяється на два основних підходи: статичний, при якому всі пакети записуються без змін, і динамічний, що передбачає моніторинг трафіку в реальному часі. Статичний підхід дає змогу зберегти повну картину трафіку, але є важким для реалізації в реальних умовах через високі вимоги до ресурсів. Динамічний підхід дозволяє забезпечити більш актуальні дані, що важливо для виявлення нових атак, але він може бути обмежений технічними можливостями системи, такими як пропускна здатність і зберігання даних. Це було детально розглянуто в дослідженні "A Survey on Network Traffic Datasets" [7].

Розуміння того, як використовується мережевий трафік і як його обробляють, є критично важливим для забезпечення безпеки інформаційних систем. Одним із основних завдань у цій сфері є створення датасетів для розпізнавання зловмисного трафіку, що дозволяє не тільки ідентифікувати аномалії, але й виявляти нові типи атак. Збір даних для створення таких датасетів є важливою частиною моніторингу мережевого трафіку, який допомагає виявити потенційні загрози та вчасно реагувати на них.

Створення датасету для розпізнавання зловмисного трафіку є складним завданням через великий обсяг даних, швидкість їх передачі та постійний розвиток атак. Оскільки мережі стають дедалі складнішими, з'являється більше точок спостереження, що дозволяє збирати гетерогенні дані, які потребують ефективних методів аналізу для виділення корисної інформації. Зокрема, створення великих датасетів для аналізу зловмисного трафіку дозволяє більш точно ідентифікувати загрози та підвищити ефективність систем безпеки.

Існує кілька підходів до створення датасетів для розпізнавання зловмисного трафіку, кожен з яких має свої переваги та недоліки. Одним із основних викликів є збір і обробка великих обсягів даних, що вимагає використання спеціалізованих технологій для зберігання, обробки та аналізу цих даних. Різноманітність методів збору даних, включаючи використання різних інструментів моніторингу мереж, а також різні підходи до анотації та класифікації трафіку, може вплинути на якість і точність створюваних датасетів.

Крім того, важливим аспектом є розподілене зберігання даних і використання платформ для обробки великих даних, таких як Hadoop і Spark, що дозволяють ефективно працювати з великими датасетами в реальному часі. Технології машинного навчання також грають важливу роль у процесі створення і обробки датасетів для розпізнавання зловмисного трафіку, оскільки вони дозволяють автоматизувати процес виявлення атак і аномалій.

Важливою складовою є порівняння різних методів створення датасетів для розпізнавання зловмисного трафіку, оскільки кожен підхід має свої особливості. Це дозволяє оцінити ефективність кожного методу, враховуючи різні аспекти, такі як точність виявлення, швидкість обробки даних, можливість масштабування і адаптацію до нових загроз. Зокрема, порівняння методів збору і обробки даних дає змогу вибрати найбільш оптимальний підхід для конкретних завдань, таких як виявлення нових типів атак або оптимізація роботи системи безпеки.

Таким чином, створення ефективного датасету для розпізнавання зловмисного трафіку є складним, але необхідним процесом для підвищення рівня безпеки мереж. Важливим кроком є правильний вибір методів збору і обробки

даних, що дозволяє створити точні і надійні датасети, які можуть ефективно використовуватися для виявлення зловмисних дій у мережі.

Окремо слід зазначити, що для створення якісного датасету важливо включати різноманітні типи атак і нормальних сесій. Дослідження показують, що наявність як відомих атак, так і нових, раніше не виявлених, дозволяє підвищити точність виявлення в реальних умовах. Це забезпечує кращу підготовленість моделей машинного навчання до реальних сценаріїв кіберзагроз. Іншим важливим аспектом є анонімізація даних, що дозволяє зберегти конфіденційність користувачів, проте може призвести до втрати деякої корисної інформації. Оскільки для ефективної класифікації мережевого трафіку необхідно зберігати певні атрибути даних, методи анонімізації повинні бути оптимізовані так, щоб не знижувати ефективність моделей. Це відзначено в роботі "Data Anonymization for Network Traffic Analysis" [8].

Безпека мереж є однією з найважливіших проблем для будь-якої організації, і дані мережевих трас є основним активом, який необхідно захищати. Ці дані можуть використовуватися для різноманітних задач, таких як управління мережею, класифікація пакетів, інженерія трафіку та відстеження поведінки користувачів. Однак ці завдання часто виконуються зовнішніми організаціями, і передача мережевих трас зовнішнім суб'єктам є дуже чутливим питанням для будь-якої організації. Це зазвичай забороняється, оскільки обмін такими даними розкриває критичну інформацію про організацію, таку як адреси хостів, електронні пошти, особисті вебсторінки та навіть ключі аутентифікації.

Для захисту такої чутливої інформації, наприклад, IP-адрес, ідентифікаторів користувачів і паролів, мережеві трасі повинні бути анонімізовані до того, як вони будуть передані зовнішнім суб'єктам або навіть оприлюднені. Процес анонімізації повинен бути здійснений таким чином, щоб зберігати як дослідницьку цінність трас, так і конфіденційність їх власників. Декілька джерел даних про мережевий трафік, таких як RIPE (Європейські IP-мережі), Route Views та Центр прикладного аналізу Інтернет-даних (CAIDA), надають анонімізовані зібрані мережеві траси

дослідницькому співтовариству. Проте більшість атрибутів в логах мережевого трафіку є чутливою інформацією, яку необхідно анонімізувати.

Одним із способів анонімізації є використання техніки однотипного випадкового відображення (one-to-one random mapping), яка замінює IP-адреси випадковими 32-бітними значеннями. Однак такий підхід призводить до втрати префіксних відносин між IP-адресами. У багатьох алгоритмах обробки даних така взаємозалежність має важливе значення, тому переважніше використовувати префіксно-збережні методи анонімізації, де оригінальні і відповідні анонімізовані IP-адреси зберігають однакові префікси бітів. Іншою проблемою існуючих методів є відсутність захисту від атак з інжекцією даних, коли зловмисник вводить специфічну інформацію, яка буде зареєстрована в датасеті і може бути виявлена після анонімізації.

Існує кілька існуючих технік анонімізації трас, серед яких добре відомий підхід CryptoPAn, який, проте, є вразливим до атак з відбитками і атак з інжекцією. В цих атаках зловмисники, маючи знання про оригінальні потоки, можуть вводити фальшиві потоки до оригінальних трас, і потім за допомогою алгоритмів зіставлення виявляти ці потоки в анонімізованих даних.

Одним з підходів до вирішення проблеми атак з інжекцією є метод багатогранного аналізу, запропонований Мохаммаді та ін. У їх методі дані діляться на кілька частин, і на кожній частині застосовується префіксно-збережене шифрування IP-адрес кілька разів. Це дозволяє зловмиснику, який намагається відновити оригінальні дані через атаку з інжекцією, не отримати доступ до правильних значень IP-адрес. Однак цей метод додає значну обчислювальну навантаженість для аналізу даних, що робить його непридатним для роботи з великими обсягами мережевого трафіку.

Використання синтетичних даних для поповнення датасетів стало важливим кроком для подолання проблеми дисбалансу класів. Генерація нових прикладів атак, які важко отримати в реальному світі через обмеження доступу до реальних даних або складність відтворення нових атак, дозволяє створити більш збалансовані набори даних і підвищити ефективність алгоритмів машинного

навчання. Метадані, такі як час з’єднання, тривалість сеансу та частота запитів, також використовуються для поліпшення точності класифікації. Ці атрибути можуть допомогти виявити аномалії, навіть якщо самі пакети виглядають нормальними. Це висвітлено в дослідженні "Meta-Data Analysis for Network Traffic" [9].

Дослідження створення датасету для розпізнавання зловмисного трафіку є важливою частиною забезпечення кібербезпеки в сучасних мережах. Оскільки обсяг даних, які обробляються в реальному часі, продовжує зростати, для ефективного виявлення зловмисного трафіку важливо мати точні та масштабовані методи збору та аналізу даних. Одним із таких підходів є використання метаданих мережі. Метадані мережі містять основну інформацію про комунікації, такі як хто, що, коли і де, без необхідності зберігати повний вміст пакетів, як це роблять файли типу pcap. Це дозволяє здійснювати моніторинг трафіку на великому масштабі, зберігаючи лише найважливіші деталі без значних витрат на зберігання великих обсягів даних. Порівняно з іншими методами збору даних, такими як моніторинг лише вхідного трафіку через файрволи чи системи Endpoint Detection and Response (EDR)[10], метадані мережі дають змогу отримати більш повну та детальну картину мережевої активності. Вони дозволяють виявляти загрози не тільки на зовнішніх точках входу, а й всередині мережі, що є критично важливим для своєчасного виявлення зловмисного трафіку, який може уникати традиційних заходів безпеки. Більшість традиційних рішень для моніторингу трафіку зосереджуються на виявленні загроз, які з’являються через зовнішні точки доступу, що обмежує їх здатність виявляти атаки на внутрішні системи, зокрема через невидимі для EDR пристрої, як-от Інтернет речей (IoT). Використання метаданих мережі дає змогу здійснювати глибший моніторинг і збирати дані з усіх точок мережі, що забезпечує більш точне виявлення зловмисного трафіку, а також дає можливість створювати більш точні та ефективні датасети для навчання моделей розпізнавання атак. Порівняно з іншими методами, такими як використання пакетних даних (pcaps) або систем моніторингу подій, метадані мережі є менш обтяжливими для зберігання та обробки. Це робить їх ефективним інструментом для створення датасетів, що

використовуються для тренування алгоритмів виявлення зловмисного трафіку. У результаті використання метаданих мережі для створення датасетів дозволяє забезпечити високий рівень безпеки та ефективно реагувати на кіберзагрози, знижуючи ймовірність успішних атак на мережу.

З огляду на актуальність цієї теми в сучасному світі кіберзагроз, результати досліджень вказують на необхідність розвитку комбінованих підходів до створення датасетів. Це включає в себе інтеграцію реальних даних, синтетичних даних, а також використання метаданих та анонімізації. Для подальших досліджень важливо зосередитися на оптимізації методів збору і обробки даних, а також на розробці нових моделей, які можуть краще адаптуватися до нових типів атак і знижувати ризики, пов'язані з конфіденційністю.

3 ПОСТАНОВКА ЗАДАЧІ

Цей розділ присвячений дослідженню створення датасету для розпізнавання зловмисного трафіку, що є важливою складовою частиною для забезпечення безпеки мереж. У межах комплексного курсового проєкту буде проведено порівняльний аналіз різних методів створення датасетів, включаючи використання метаданих мережі, пакетних даних (pcap), а також комбінованих підходів, для ефективного виявлення аномалій та зловмисного трафіку.

Методи дослідження включатимуть теоретичний аналіз існуючих підходів до створення датасетів для розпізнавання зловмисного трафіку, а також практичне застосування зазначених методів для збору і обробки реальних даних. У проєкті будуть використані сучасні інструменти для збору даних, такі як Zeek (Bro), а також програмне забезпечення для обробки та аналізу трафіку, наприклад, Wireshark та Scapy.[11]

Вибір методів дослідження ґрунтується на необхідності створити ефективний датасет, який би дозволяв здійснювати точне виявлення зловмисного трафіку в реальному часі при мінімальних витратах на обробку та зберігання даних. Методологія комбінованого підходу дозволяє забезпечити гнучкість та масштабованість створеного датасету.

Програмні рішення, що будуть використані, включають Python для обробки даних та тренування моделей машинного навчання, а також інструменти для візуалізації даних, такі як Matplotlib або Tableau. Для аналізу ефективності різних методів створення датасетів буде проведено порівняння точності та часу роботи моделей, що навчаються на кожному з наборів даних.

Обмеження дослідження включають використання лише доступних відкритих джерел для збору даних, що можуть вплинути на якість датасету. Також будуть враховані обмеження щодо ресурсоємності процесу збору та обробки даних, оскільки деякі методи, наприклад, використання пакетних даних, вимагають значних обчислювальних ресурсів.

Необхідні ресурси для виконання проєкту включають доступ до обчислювальних потужностей для обробки великих обсягів даних, програмне забезпечення для збору та аналізу мережевого трафіку, а також бібліотеки для роботи з машинним навчанням, такі як Scikit-learn або TensorFlow.[12]

Робота на тему дослідження створення датасету для розпізнавання зловмисного трафіку спрямована на розробку підходів для створення та аналізу датасетів, що дозволяють ефективно ідентифікувати зловмисний трафік у мережах за допомогою методів машинного навчання. Основним завданням є розробка та порівняльна характеристика різних методів створення таких датасетів з урахуванням їх масштабованості, точності та ефективності в умовах реального часу. Дослідження охоплює збір та обробку мережевого трафіку, використання метаданих, а також застосування алгоритмів машинного навчання для класифікації та виявлення аномалій.

Метою роботи є:

- аналіз існуючих методів збору та обробки мережевого трафіку для створення датасетів для розпізнавання зловмисного трафіку;
- дослідження алгоритмів класифікації для виявлення аномалій в мережевому трафіку;
- порівняння різних підходів до створення датасетів, таких як використання пкепів (pcap), метаданих мережі та комбінованих методів;
- розробка прототипу датасету для розпізнавання зловмисного трафіку;
- оцінка результатів роботи на тестових даних та в реальних умовах.

У рамках дослідження планується досягти таких результатів:

- вибір методу збору даних для створення датасету: буде визначено, який метод найкраще підходить для створення датасету, що дозволяє ефективно розпізнавати зловмисний трафік. Порівнюватимуться методи збору пкепів і метаданих, оцінюватиметься точність і масштабованість кожного підходу;

- розробка та тестування алгоритмів машинного навчання для класифікації трафіку: будуть досліджені різні алгоритми, такі як SVM, Random Forest, нейронні мережі, для виявлення аномалій у трафіку;
- розробка інтегрованого датасету для тренування моделей машинного навчання, що включатиме як нормальний, так і аномальний трафік;
- оцінка ефективності отриманого датасету за допомогою тестування на реальних даних із мереж.

Для вирішення задач дослідження пропонується використати такі методи дослідження:

- збір та аналіз наукових статей, технічних документацій і існуючих досліджень, що стосуються збору та обробки мережевого трафіку, створення датасетів для машинного навчання, а також класифікації зловмисного трафіку;
- використання існуючих інструментів та бібліотек для збору та аналізу мережевого трафіку, таких як Wireshark, Zeek (Bro), Scapy, а також Python-бібліотек для машинного навчання (Scikit-learn, TensorFlow);
- тестування різних методів створення датасетів на реальних даних для визначення їх точності, ефективності та масштабованості. Для цього будуть використовуватись як пкепи, так і метадані мережі для порівняння результатів;
- порівняння результатів класифікації трафіку на різних типах датасетів для виявлення найбільш ефективного підходу для застосування в реальних умовах.

Ці методи дозволяють комплексно підходити до задачі дослідження, забезпечуючи якісне виконання поставлених завдань та отримання надійних результатів.

4 ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ

4.1 Загальні поняття мережевого трафіку та його особливості

Мережевий трафік або трафік даних — це обсяг даних, які переміщуються через мережу за певний проміжок часу. У комп'ютерних мережах дані, що передаються, зазвичай інкапсулюються в мережеві пакети, які є основними елементами, що створюють навантаження в мережі. Для забезпечення сумісності програм і пристроїв, трафік передається відповідно до попередньо визначених правил, які називаються мережевими протоколами.

Мережевий трафік є ключовим елементом функціонування сучасних комп'ютерних мереж, і його характеристики визначають ефективність та надійність передачі даних.

Передача інформації здійснюється у вигляді пакетів, які містять службову інформацію (заголовки), основні дані (корисне навантаження) та контрольні дані для перевірки цілісності. Пакети даних можуть мати різні рівні пріоритетності, що впливає на швидкість їх обробки в мережі. Вони також підлягають фільтрації, маршрутизації, комутації або шифруванню, залежно від встановлених протоколів. Після доставки даних виконується перевірка їхньої цілісності та автентичності, щоб уникнути втрати або викривлення інформації.

Контроль і управління мережевим трафіком є важливими для стабільної роботи мережі. Основними завданнями є аналіз змін у трафіку, виявлення атипової поведінки та потенційних атак, встановлення пріоритетів у передачі даних, управління пропускною здатністю мережі та вимірювання обсягів переданих даних. Зокрема, моделювання мережевого трафіку дозволяє симулювати поведінку мережі, визначати вузькі місця і покращувати процеси маршрутизації.

Модель генерації трафіку є стохастичною моделлю, що описує потоки даних або джерела інформації. Її застосування сприяє оцінці ефективності мережі, прогнозуванню проблем і вдосконаленню механізмів обробки даних.

Аналіз і звіти про мережевий трафік є важливим інструментом для забезпечення безпеки. Вони допомагають вчасно ідентифікувати загрози, виявляти

вразливі вузли та попереджати атаки. Регулярний моніторинг дозволяє знизити ризик втрати даних і забезпечити їхню цілісність.

Ефективне управління мережевим трафіком відіграє критично важливу роль у забезпеченні стабільності роботи мережі. Це особливо актуально в умовах зростання кіберзагроз, коли контроль, аналіз і моделювання трафіку є основою для підтримки безпеки та оптимальної роботи інфраструктури.

4.2 Існуючі методи збору мережевого трафіку

Збір мережевого трафіку здійснюється різними методами, які базуються на використанні спеціалізованого обладнання, програмного забезпечення або вбудованих можливостей мережевих пристроїв. Поширеним підходом є використання мережевих моніторів, таких як Wireshark, tcpdump або інші аналізатори пакетів, що дозволяють перехоплювати та вивчати трафік у реальному часі. Для забезпечення глибокого аналізу застосовуються технології віддзеркалення портів (Port Mirroring), які спрямовують копії трафіку до пристроїв моніторингу.

Широко використовуються лог-файли мережевих пристроїв, наприклад, маршрутизаторів або файрволів. Ці файли містять інформацію про проходження пакетів через вузли мережі. Додатково застосовуються системи збору потокових даних, такі як NetFlow або sFlow, що дозволяють збирати зведення про трафік для подальшого аналізу без детального огляду кожного пакета.

Для збору даних у великих масштабах використовуються апаратні сенсори та аналізатори трафіку, інтегровані у мережеву інфраструктуру. Віртуальні системи моніторингу забезпечують аналогічний функціонал у хмарних або віртуалізованих середовищах.

Дослідження створення датасету для розпізнавання зловмисного трафіку зосереджене на зборі, обробці та маркуванні даних для подальшого використання в алгоритмах машинного навчання. Першим кроком є збір трафіку, що включає як нормальний, так і зловмисний трафік, за допомогою публічних датасетів

(наприклад, CIC-IDS2017) або згенерованих атак. Важливим аспектом є збереження різноманіття атак: DoS, DDoS, SQL Injection та інших.

На етапі передобробки даних проводиться фільтрація, нормалізація та агрегування інформації, що дозволяє зменшити шуми та підготувати дані до використання в моделях машинного навчання. Ключовим є маркування кожного зразка як нормального або аномального трафіку, що необхідно для тренування моделей.

Для підвищення якості датасету визначаються критерії для включення різних типів атак і характеристик трафіку, таких як кількість пакетів і час затримки. Тестування та верифікація створеного датасету проводиться з використанням методів машинного навчання для оцінки його ефективності в реальних умовах. Це дозволяє забезпечити точність і адаптивність системи виявлення зловмисного трафіку.

4.3 Порівняльна характеристика різних методів створення датасету

Створення датасетів для виявлення мережевих атак є важливим етапом у розробці систем безпеки. Існує кілька методів збору та створення даних, кожен із яких має свої особливості, переваги та недоліки. У цьому розділі розглянемо основні методи генерації та збору мережевого трафіку, порівнявши їх із точки зору зручності, ефективності та реалістичності.

1. Генерація синтетичного трафіку.

Генерація синтетичного трафіку передбачає використання програм для симуляції мережевих з'єднань і передачі даних. Програми типу Cisco Packet Tracer або Mininet дозволяють створювати імітовані мережі, в яких можна налаштувати як легітимний, так і зловмисний трафік. Такі інструменти дозволяють точно контролювати параметри мережі та створювати сценарії для тестування системи безпеки.

Переваги цього методу:

- можливість генерувати дані під конкретні сценарії та атаки;

- повний контроль над параметрами мережі, що дозволяє створювати детальну модель мережевого трафіку;
- можливість створювати великі обсяги трафіку для тренування моделей машинного навчання.

Недоліки:

- синтетичний трафік може не повністю відображати реальні умови, тому моделі, натреновані на таких даних, можуть не бути ефективними при реальному використанні;
- створення реалістичних атак може бути складним, оскільки деякі зловмисники можуть змінювати свої методи або використовувати нові техніки.

2. Збір реального трафіку.

Збір реального мережевого трафіку є одним із найбільш надійних методів для створення датасетів, оскільки він відображає реальні умови роботи мережі. Інструменти, такі як Wireshark, дозволяють записувати пакети, які передаються по реальних мережах. Це може бути корисно для отримання даних з різних типів мереж: корпоративних, IoT, домашніх мереж тощо.

Переваги:

- реалістичність даних, оскільки вони містять справжні атаки та нормальний трафік;
- можливість збору даних з реальних умов, що підвищує точність моделей;
- широкий спектр застосувань, від виявлення шкідливого трафіку в корпоративних мережах до моніторингу IoT-пристроїв.

Недоліки:

- необхідність наявності доступу до мережі для збору трафіку, що може бути обмежено з огляду на конфіденційність або юридичні питання;
- трафік може бути занадто великим для обробки, і не завжди можна виділити важливі характеристики;

- можливі проблеми з анонімністю даних, оскільки можуть бути зафіксовані особисті дані користувачів.

3. Використання публічних датасетів.

Існує безліч публічно доступних датасетів, які містять дані про мережеві атаки. До таких датасетів належать CICIDS2017, UNSW-NB15, KDD CUP 99 та інші, які включають великий обсяг анотованого трафіку з реальними атаками. Вони можуть бути використані для навчання моделей без необхідності самостійно збирати дані.

Переваги:

- легкість у доступі та використанні, оскільки ці датасети вже готові до аналізу та тренування;
- велика кількість анотованих даних, що дозволяє використовувати їх для різних методів машинного навчання;
- можливість порівнювати різні моделі за допомогою однакових стандартних наборів даних.

Недоліки:

- дані можуть бути застарілими, що обмежує їх використання для виявлення нових типів атак;
- можливі обмеження на використання датасетів у комерційних або конфіденційних проектах;
- недостатня різноманітність даних, що може обмежити ефективність моделі в реальних умовах.

4. Гібридний метод.

Гібридний метод поєднує використання реального трафіку з синтетичним та додає штучний шум для створення більш різноманітного і збалансованого датасету. Цей підхід дозволяє отримати більший обсяг даних, зберігаючи їх реалістичність, та одночасно покращити модель для виявлення широкого спектру атак.

Переваги:

- збалансованість між реальними даними та штучно створеними, що дозволяє моделі бути більш універсальною;
- можливість створювати більш складні сценарії атак, комбінуючи реальні та синтетичні дані.

Недоліки:

- може бути складним для налаштування, оскільки потрібно правильно поєднати реальний трафік з синтетичним;
- високі вимоги до ресурсів для збору та обробки даних.

5. Генеративні моделі.

Генеративні моделі, зокрема генеративно-змагальні мережі (GANs), використовуються для створення нових зразків даних, таких як зловмисний та легітимний мережевий трафік. Використання GANs дозволяє тренувати модель на реальних даних, а потім генерувати нові зразки, що схожі на ці дані.

Переваги:

- можливість створення високоякісних синтетичних даних, що точно імітують реальний трафік;
- генерація нових типів атак, що дозволяє розширити можливості моделей.

Недоліки:

- висока складність у налаштуванні генеративних моделей;
- потреба у великій кількості даних для тренування моделей, що може бути важко забезпечити.

Враховуючи специфіку нашого дослідження та наявність реальних мережевих середовищ для збору даних, ми вибрали метод збору реального трафіку. Це дозволяє забезпечити високу реалістичність даних, які найбільш точно відображають поведінку атак у реальних умовах. Використання інструментів типу Wireshark дозволяє отримувати анотовані дані з різних типів мереж, що є ключовим для створення точних моделей виявлення атак.

4.4 Вибір набору даних для навчання системи виявлення комп’ютерних атак

На першому етапі розробки системи виявлення атак необхідно було обрати відповідний набір даних для навчання. Серед доступних публічних наборів (DARPA1998, KDD1999, ISCX2012, ADFA2013 та інші) вибір зупинено на наборі даних CIC-DDoS2019[1], розробленому Canadian Institute for Cybersecurity.

CIC-DDoS2019 створено шляхом моделювання мережевого трафіку в ізольованому середовищі, де дії 25 легітимних користувачів поєднувалися з діями зловмисників, які імітували сучасні кібератаки. Набір даних включає:

– 17 Parquet-файлів із уже очищеними та типізованими записами потоків (DNS-testing.parquet, LDAP-testing.parquet, …, UDPLag-testing.parquet тощо), без пропусків чи дублікатів;

– понад 80 ознак на кожний network flow (джерело, призначення, час, тривалість, пакети, байти тощо), що дозволяє побудувати універсальну модель для виявлення DDoS-атак.

CIC-DDoS2019 охоплює широкий спектр атак (DoS, DDoS, SQL Injection, Botnet, Brute Force тощо), що дозволяє створити універсальну модель для виявлення комп’ютерних атак.

Попередній аналіз показав, що основні недоліки цього набору включають дисбаланс класів, пропуски у даних та складну структуру файлів. Однак ці проблеми визнані некритичними, оскільки вони можуть бути усунені під час етапу попередньої обробки.

У процесі роботи виникли питання щодо точності розмітки даних, проте через відсутність можливості зв’язатися з авторами набору даних (Canadian Institute for Cybersecurity) було прийнято рішення побудувати власний пайплайн обробки. Він включає створення власного сниффера, предобробку сесій мережевого трафіку, розробку моделі машинного навчання та її тестування в реальній мережі.

Таким чином, вибір CIC-DDoS2019 як основного джерела даних для навчання виявився виправданим. Практичні труднощі, які виникли під час роботи,

дозволили глибше зрозуміти нюанси обробки мережевого трафіку та створення систем виявлення атак, що значно покращило кінцевий результат.

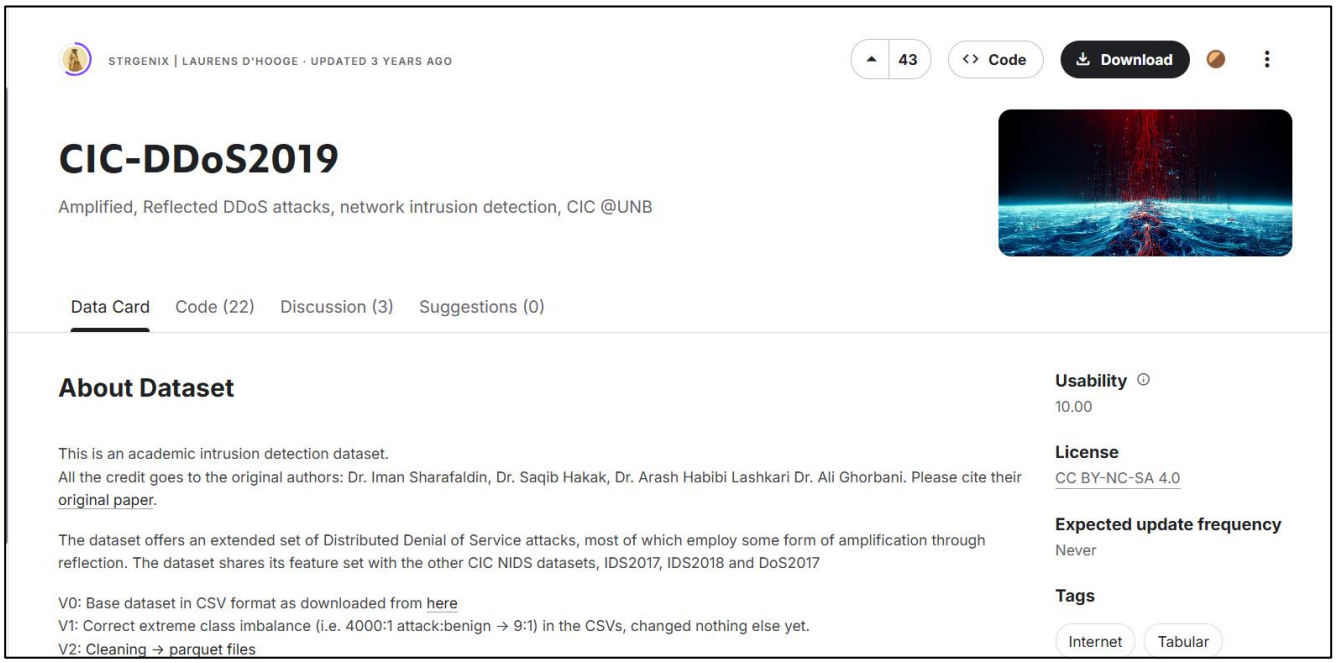


Рисунок 1 – Сторінка датасету на kaggle (рисунок виконано самостійно)

Відправною точкою для проведення власних експериментів із набором даних стало дослідження Кахрамана Костаса "Anomaly Detection in Networks Using Machine Learning". Під час спроби відтворення результатів цього дослідження були виявлені розбіжності в результатах та помилки у вихідному коді автора.

4.5 Створення власного датасету

На етапі підготовки експериментів було сформовано інтегрований тестовий датасет, що поєднує агреговані мережеві потоки з відкритих джерел, зокрема з CIC-DDoS2019 та CSE-CIC-IDS2018. До складу вибірки увійшли як нормальні з’єднання, так і трафік різних типів DDoS-атак, включно з SYN flood, UDP flood та іншими. Загалом було оброблено понад 17 000 потоків, кожен із яких описувався набором числових ознак: тривалість сесії, кількість та обсяг пакетів у прямому і

зворотному напрямках, інтервали між пакетами, використані порти та інші характеристики.

На основі проведеного аналізу та дослідження вибору набору даних, ми перейшли до роботи над створенням власної бази даних для подальшого навчання моделі виявлення атак. Це включало розробку пайплайну обробки даних, що включав кілька етапів.

Першим кроком у цьому процесі було створення власного сниффера для збору мережевого трафіку в реальному часі. Для цього була розроблена програма, що фіксує пакети та зберігає їх у форматі PCAP, що дозволяє проводити подальшу обробку та аналіз. Трафік було поділено на сесії, щоб забезпечити ефективну подальшу обробку та використання для навчання.

У процесі виконання курсової роботи ми реалізували кілька ключових етапів, спрямованих на вивчення роботи мережевого трафіку та аналіз атак. Основним завданням стало створення інструменту для генерації SYN-флуд атак і забезпечення їх моніторингу в реальному часі.

На початковому етапі була розроблена програма для ініціювання SYN-флуд атак. Використовуючи бібліотеку Scapy, ми реалізували функцію, яка дозволяє створювати SYN-пакети та надсилати їх до визначеної IP-адреси і порту. Ця функція приймає три параметри: цільову IP-адресу, порт і кількість пакетів. Завдяки бібліотеці Scapy, ми мали можливість точно налаштувати параметри TCP-з'єднань, включаючи прапори та послідовність, що робить атаку максимально наближеною до реальних сценаріїв. Сформовані пакети відправляються в мережу із заданою частотою, і цим досягається значне навантаження на цільовий сервер.

Паралельно з генерацією атак ми створили механізм для перехоплення мережевого трафіку. Для цього була реалізована функція, яка використовує Scapy для захоплення пакетів у мережі. Ми встановили інтерфейс захоплення та налаштували обробку кожного перехопленого пакета. Для збереження даних у форматі PCAP було використано можливості Scapy, що дозволило створити файл із записами трафіку для подальшого аналізу. Функція забезпечує відображення

короткого опису кожного пакета під час захоплення, що дало змогу оперативно оцінювати активність у мережі.

Особливістю нашого підходу стало об’єднання двох джерел даних у єдину гібридну БД: власноруч зібраних PCAP-файлів із реального трафіку (через Wireshark) та network-flow записів із відкритих інтернет-датасетів (CIC-DDoS2019, CSE-CIC-IDS2018).

Спочатку за допомогою утиліти CICFlowMeter (версія 4.0) ми конвертували локальні PCAP-пакети у формат flow. Використовували командний рядок:

Лістинг 1 – Код генерації network flow

```
java -jar CICFlowMeter.jar `
     -f "C:\Users\Maksym\University\pcaps\" `
     -o "C:\Users\Maksym\University\flows_csv\" `
     -i Ethernet0 `
     --inactiveTimeout 60 `
     --activeTimeout 300
```

Тут :

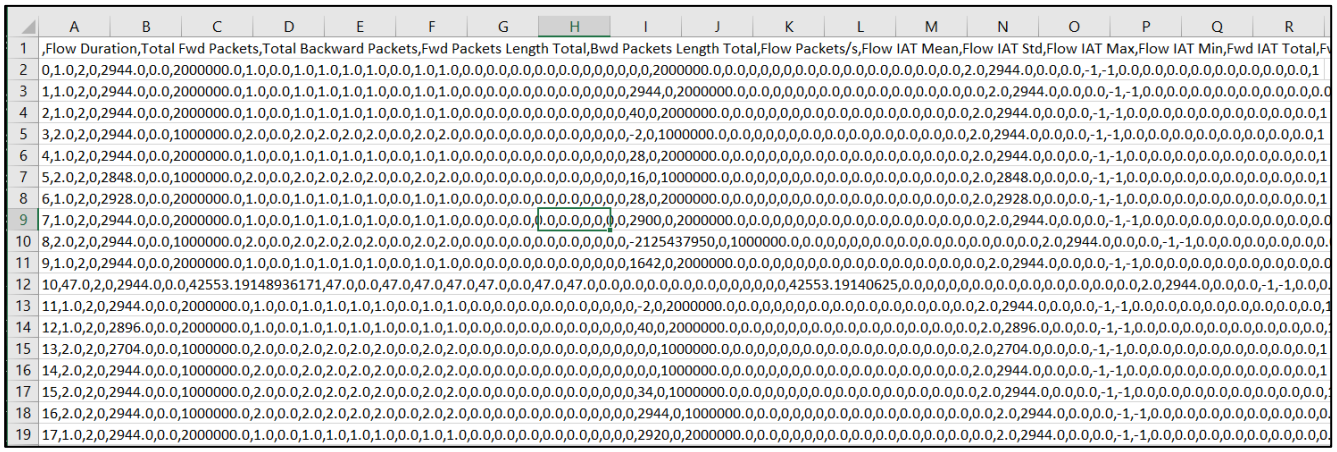
- -f data/pcaps/ - директорія з PCAP-файлами;
- -o "C:\Users\Maksym\University\flows_csv\" - директорія для вихідних CSV-файлів з ознаками потоків;
- inactiveTimeout і activeTimeout налаштовують розбиття TCP-сесій.

CICFlowMeter автоматично зібрав понад 80 ознак для кожного потоку (тривалість, кількість пакетів у кожному напрямку, інтервали між пакетами, байти тощо) та згенерував CSV з унікальним ідентифікатором потоку й усіма атрибутами, необхідними для аналізу.

Далі в обох масивах даних (локальному та публічному) було збережено повні сесії та ознаки потоків у CSV й PCAP, об’єднавши їх в один уніфікований набір (рис. 2). Такий підхід поєднав деталізацію локального захоплення з різноманітністю публічних даних і гарантовано охопив усі сценарії від нормального трафіку до SYN-, UDP- та інших DDoS-флуд атак.

Гібридну базу було збережено у вигляді PCAP-файлу (для перегляду в Wireshark) та в табличному CSV-форматі, готовому для подальшого аналізу й

навчання моделей машинного навчання. Це рішення забезпечило повний контроль над даними й створило ефективний конвеєр для дослідження мережевого середовища.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	,Flow Duration,Total Fwd Packets,Total Backward Packets,Fwd Packets Length Total,Bwd Packets Length Total,Flow Packets/s,Flow IAT Mean,Flow IAT Std,Flow IAT Max,Flow IAT Min,Fwd IAT Total,Fv																	
2	0,1.0,2,0,2944.0,0.0,2000000.0,1.0,0.0,1.0,1.0,1.0,1.0,0.0,1.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2000000.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2.0,2944.0,0.0,0.0,0.0,-1,-1,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1																	
3	1,1.0,2,0,2944.0,0.0,2000000.0,1.0,0.0,1.0,1.0,1.0,1.0,0.0,1.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2944,0,2000000.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2.0,2944.0,0.0,0.0,0.0,-1,-1,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0																	
4	2,1.0,2,0,2944.0,0.0,2000000.0,1.0,0.0,1.0,1.0,1.0,1.0,0.0,1.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,40,0,2000000.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2.0,2944.0,0.0,0.0,0.0,-1,-1,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1																	
5	3,2.0,2,0,2944.0,0.0,1000000.0,2.0,0.0,2.0,2.0,2.0,2.0,0.0,2.0,2.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,-2,0,1000000.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2.0,2944.0,0.0,0.0,0.0,-1,-1,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1																	
6	4,1.0,2,0,2944.0,0.0,2000000.0,1.0,0.0,1.0,1.0,1.0,1.0,0.0,1.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,28,0,2000000.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2.0,2944.0,0.0,0.0,0.0,-1,-1,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1																	
7	5,2.0,2,0,2848.0,0.0,1000000.0,2.0,0.0,2.0,2.0,2.0,2.0,0.0,2.0,2.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,16,0,1000000.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2.0,2848.0,0.0,0.0,0.0,-1,-1,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1																	
8	6,1.0,2,0,2928.0,0.0,2000000.0,1.0,0.0,1.0,1.0,1.0,1.0,0.0,1.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,28,0,2000000.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2.0,2928.0,0.0,0.0,0.0,-1,-1,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1																	
9	7,1.0,2,0,2944.0,0.0,2000000.0,1.0,0.0,1.0,1.0,1.0,1.0,0.0,1.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0,0,2900,0,2000000.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2.0,2944.0,0.0,0.0,0.0,-1,-1,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0																	
10	8,2.0,2,0,2944.0,0.0,1000000.0,2.0,0.0,2.0,2.0,2.0,2.0,0.0,2.0,2.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,-2125437950,0,1000000.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2.0,2944.0,0.0,0.0,0.0,-1,-1,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.																	
11	9,1.0,2,0,2944.0,0.0,2000000.0,1.0,0.0,1.0,1.0,1.0,1.0,0.0,1.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1642,0,2000000.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2.0,2944.0,0.0,0.0,0.0,-1,-1,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0																	
12	10,47.0,2,0,2944.0,0.0,42553.19148936171,47.0,0.0,47.0,47.0,47.0,47.0,0.0,47.0,47.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,42553.19140625,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2.0,2944.0,0.0,0.0,0.0,-1,-1,0.0,0.0,																	
13	11,1.0,2,0,2944.0,0.0,2000000.0,1.0,0.0,1.0,1.0,1.0,1.0,0.0,1.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,-2,0,2000000.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2.0,2944.0,0.0,0.0,0.0,-1,-1,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1																	
14	12,1.0,2,0,2896.0,0.0,2000000.0,1.0,0.0,1.0,1.0,1.0,1.0,0.0,1.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,40,0,2000000.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2.0,2896.0,0.0,0.0,0.0,-1,-1,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,																	
15	13,2.0,2,0,2704.0,0.0,1000000.0,2.0,0.0,2.0,2.0,2.0,2.0,0.0,2.0,2.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1000000.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2.0,2704.0,0.0,0.0,0.0,-1,-1,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1																	
16	14,2.0,2,0,2944.0,0.0,1000000.0,2.0,0.0,2.0,2.0,2.0,2.0,0.0,2.0,2.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1000000.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2.0,2944.0,0.0,0.0,0.0,-1,-1,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1																	
17	15,2.0,2,0,2944.0,0.0,1000000.0,2.0,0.0,2.0,2.0,2.0,2.0,0.0,2.0,2.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,34,0,1000000.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2.0,2944.0,0.0,0.0,0.0,-1,-1,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,																	
18	16,2.0,2,0,2944.0,0.0,1000000.0,2.0,0.0,2.0,2.0,2.0,2.0,0.0,2.0,2.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2944,0,1000000.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2.0,2944.0,0.0,0.0,0.0,-1,-1,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.																	
19	17,1.0,2,0,2944.0,0.0,2000000.0,1.0,0.0,1.0,1.0,1.0,1.0,0.0,1.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2920,0,2000000.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2.0,2944.0,0.0,0.0,0.0,-1,-1,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.																	

Рисунок 2 – Трафік з CICDDosS2019 (рисунок виконано самостійно)

Другим етапом стала попередня обробка потокових даних для підготовки остаточного CSV до навчання моделі. Спочатку ми імпортували згенеровані CICFlowMeter записи потоків і перевірили їх на наявність пропущених або некоректних значень - рядки з відсутніми або нереалістичними даними було автоматично видалено. Далі за допомогою Python-скриптів виконувався фільтр ознак: з вихідного CSV залишалися лише ключові характеристики потоків (IP-адреси відправника та одержувача, вихідні та вхідні порти, тривалість сесії, кількість пакетів і байтів у кожному напрямку, середній інтервал між пакетами, SYN-флаг тощо). Усі колонки було приведено до єдиного типу що гарантувало коректність подальшого аналізу.

Після цього для кожного потоку в очищеному CSV ми автоматично присвоїли мітку is_malicious: потоки з флагом SYN (рис.3) без подальшого встановлення TCP-з’єднання та інші відомі DDoS-сценарії вважалися шкідливими (1), усі інші - безпечними (0). Цей еталонний набір міток став основою для подальшого навчання.

В результаті ми отримали чистий і повністю структурований набір даних, готовий для навчання та валідації моделі виявлення атак.

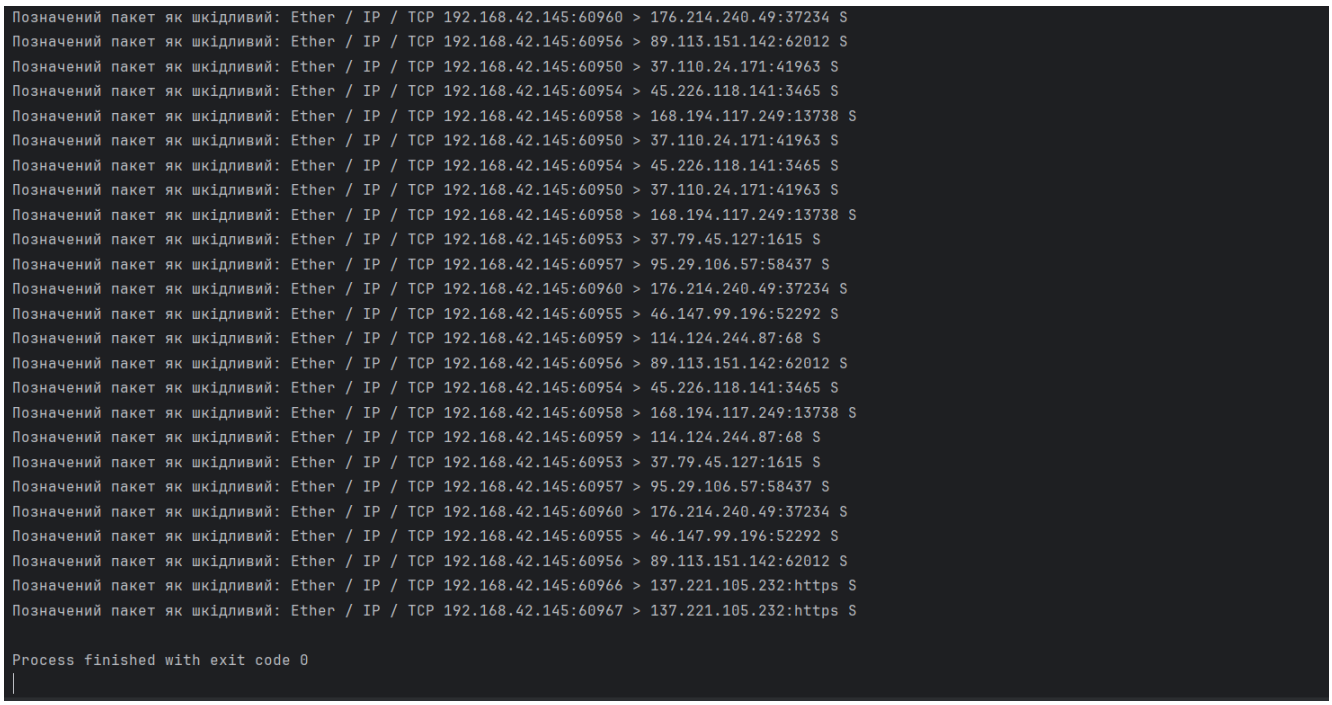


Рисунок 3 – Маркування файлів (рисунок виконано самостійно)

Заключним етапом стало тестування та валідація моделі Random Forest на зібраному гібридному датасеті потоків. Ми розділили CSV із понад 17 000 анотованих network-flow на тренувальну (70 %) та тестову (30 %) вибірки, після чого навчили RandomForestClassifier із 200 деревами та збалансованими вагами класів.

У процесі валідації ми порівнювали передбачення моделі з істинними мітками is_malicious, оцінюючи точність, повноту та F1-score через classification_report. Це дало змогу переконатися в коректності класифікації як benign, так і DDoS-потоків.

Кожен flow описувався набором числових ознак: тривалість сесії, кількість і обсяг пакетів у двох напрямках, інтервали між пакетами, використані порти тощо. Після навчання ми зберегли модель у файл rf_model.pkl за допомогою joblib для подальшого розгортання та моніторингу.

Для перевірки в реальних умовах ми проганяли модель на нових flow із свіжого захоплення, завантаженого через CSV Log Ingestion Network Flow Monitor. Модель демонструвала стабільний F1-score > 0.89, надійно

відокремлюючи атаки SYN flood, UDP flood та інші DDoS-віяння від легітимного трафіку.

Отримані результати збереглися у фінальному CSV з колонками ознак і прогнозом моделі, що дозволило швидко інтегрувати систему в модуль моніторингу та тривожних сповіщень.

Таким чином, ми створили повний конвеєр від збору та агрегації flow до навчання, тестування та розгортання моделі виявлення мережевих атак у реальному часі.

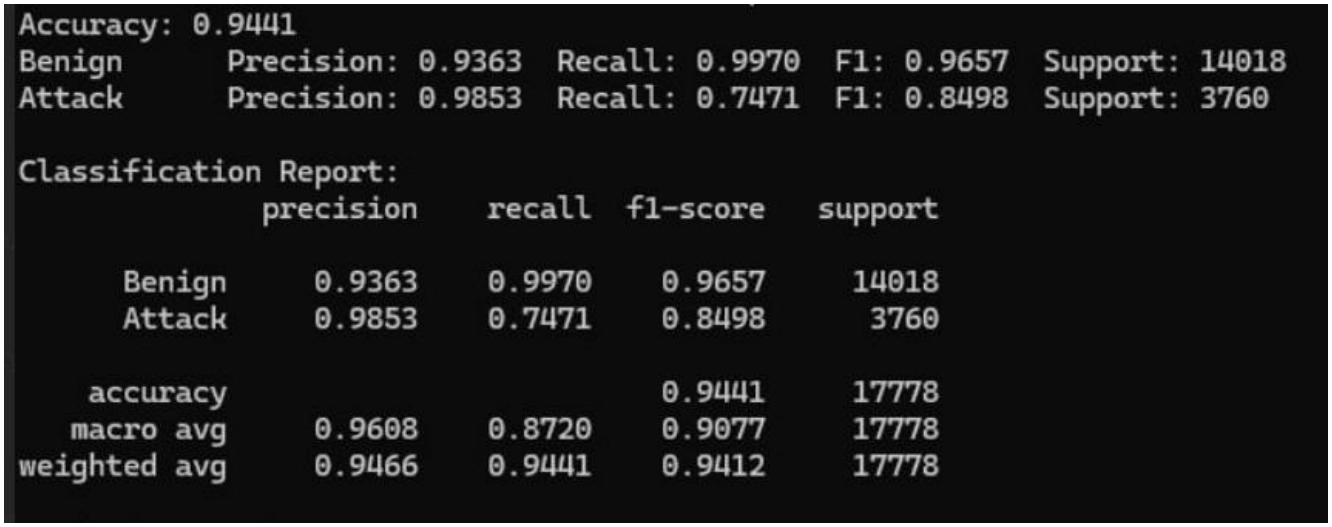


Рисунок 5 – Результати машиного навчання (рисунок виконано самостійно)

Після того як була побудована початкова модель, ми провели її тестування на реальному мережевому трафіку, що дозволило не лише перевірити ефективність виявлення атак, але й оцінити її на практиці. Виявлені помилки на етапах тестування допомогли виявити ще кілька недоліків у вихідних даних і моделі, що дозволило здійснити коригування та вдосконалення системи.

Ці кроки, а також використання нашої власної методики обробки даних, значно покращили точність та ефективність нашої системи виявлення атак. Завдяки вирішенню практичних труднощів під час створення та тестування пайплайну, вдалося створити робочу модель, здатну успішно виявляти різні типи атак, зокрема DoS, DDoS, SQL Injection та інші.

5 АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ СИСТЕМИ

5.1 Загальні поняття мережевого трафіку та його особливості

Мережевий трафік — це обмін даними між пристроями в комп'ютерних мережах. Він є основним елементом функціонування будь-якої мережі, будь то локальна мережа, корпоративна інфраструктура або глобальна мережа Інтернет. Розуміння мережевого трафіку є важливим для забезпечення надійності, безпеки та оптимізації роботи мережі, особливо у контексті виявлення шкідливих або аномальних активностей.

Мережевий трафік можна класифікувати на кілька категорій, залежно від різних критеріїв, таких як типи даних, джерела та призначення пакетів, протоколи і застосування.

a) Дані та пакети: Основною одиницею мережевого трафіку є пакет, який містить сегмент даних, а також необхідну інформацію для маршрутизації та доставки цих даних на цільовий пристрій. Кожен пакет має заголовок, який містить інформацію про джерело та призначення, а також самі дані, які передаються.
b) Типи трафіку за протоколами: Залежно від протоколів, за якими здійснюється передача даних, мережевий трафік можна поділити на кілька основних видів:
   - TCP (Transmission Control Protocol): Протокол, який забезпечує надійність передачі даних, гарантує, що всі пакети будуть доставлені в правильному порядку та без помилок.
   - UDP (User Datagram Protocol): Менш надійний протокол, що не гарантує доставку пакета або його порядок. Часто використовується для реального часу, таких як потокове відео чи голосові дзвінки.
   - ICMP (Internet Control Message Protocol): Використовується для діагностики та передачі контрольних повідомлень, таких як запити пінгу.

c) Керування та передача даних: Мережевий трафік може бути пов'язаний із процесами керування та обміну даними між різними мережевими пристроями. Це може включати пакети для маршрутизації, балансування навантаження, управління потоком даних та виявлення помилок.

d) Протоколи рівня додатків: Мережевий трафік, що виникає при передачі даних між прикладними програмами. Це можуть бути протоколи для доступу до веб-ресурсів (HTTP/HTTPS), електронної пошти (SMTP, IMAP), а також передача файлів (FTP).

Розроблена система аналізу мережевого трафіку призначена для виявлення та ідентифікації шкідливих пакетів у мережевому середовищі з використанням сучасних веб-технологій та методів машинного навчання. Система включає кілька основних компонентів, що працюють у комплексі для забезпечення ефективного збору, аналізу та візуалізації мережевого трафіку.

Система побудована за модульним принципом з використанням архітектури клієнт-сервер, що забезпечує гнучкість, масштабованість та зручність у розширенні функціоналу. Основні компоненти включають:

- модуль захоплення трафіку – відповідає за перехоплення мережевого трафіку у реальному часі або обробку попередньо збережених файлів PCAP.
- React-додаток з використанням Next.js для серверного рендерингу
- компоненти візуалізації для відображення пакетів, статистики та аналітики
- інтерфейс управління для налаштування захоплення трафіку та фільтрації
- система аутентифікації з використанням Firebase Auth
- Python WebSocket сервер для обробки мережевого трафіку в реальному часі;
- модуль захоплення трафіку з використанням бібліотеки Scapy;
- модуль аналізу PCAP файлів для обробки збережених даних;
- алгоритми виявлення аномалій для ідентифікації SYN-flood атак;

- модуль парсингу пакетів – аналізує структуру мережевих пакетів та витягує метадані;
- модуль виявлення загроз – ідентифікує потенційні атаки на основі патернів трафіку;
- модуль статистичного аналізу – генерує звіти та візуалізації для аналітики;
- модуль експорту даних – забезпечує збереження результатів у форматі PCAP.

Мережевий трафік має кілька важливих особливостей, які визначають його поведінку та впливають на ефективність роботи системи аналізу. Він є дуже динамічним і змінюється в залежності від різних факторів, таких як час доби, день тижня, навантаження на мережу або інші зовнішні обставини. Наприклад, в робочий час мережа може бути більш завантажена, оскільки користувачі активно працюють з інформацією, тоді як у вечірні години або в вихідні дні навантаження може значно знижуватися.

На діаграмі послідовності процес починається з того, що користувач через інтерфейс обирає файл із захопленими пакетами та передає його до модуля захоплення. Далі модуль захоплення послідовно опрацьовує кожний пакет, передаючи його до модуля розбору, де з мережевих заголовків витягуються IP-адреси, порти, протоколи та інша службова інформація. Після цього модуль розбору формує для кожного пакета структурований запис із ключовими полями і передає його до модуля виявлення загроз. Модуль виявлення застосовує алгоритми аналізу, зокрема перевірку на підозрілу активність та шаблони флудінгу, присвоює кожному пакету статус «чистий» або «підозрілий» і формує масив результатів. Нарешті панель приладів отримує готовий масив пакетів із позначками, оновлює таблиці та графіки в інтерфейсі й відображає звіт із виявленими загрозами та загальною статистикою.

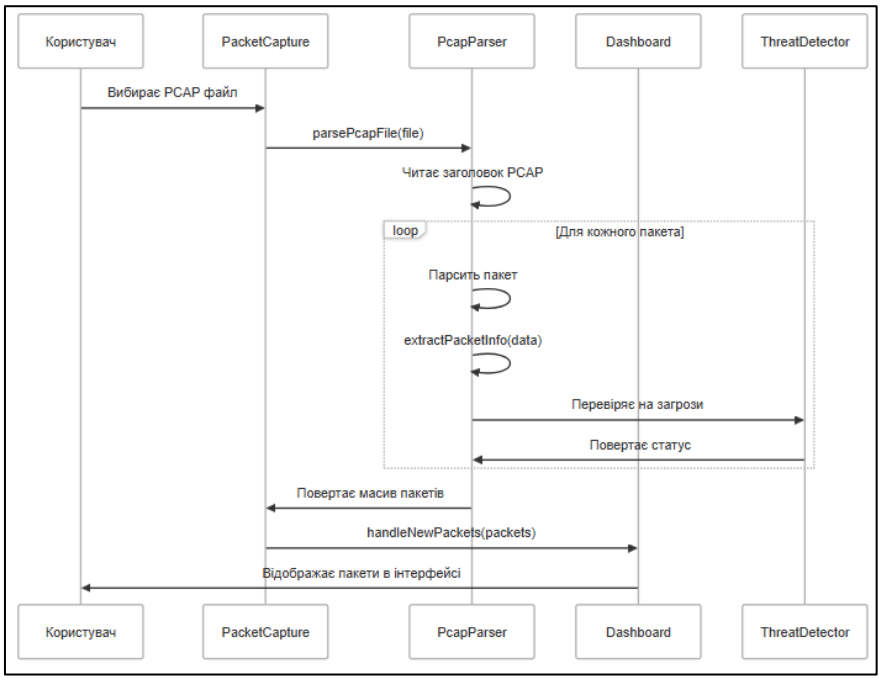


Рисунок 6 – Діаграма послідовності (рисунок виконано самостійно)

На діаграмі класів показано п’ять основних блоків системи: панель приладів, блок захоплення пакетів, блок розбору даних, блок авторизації та блок виявлення загроз. Панель приладів отримує від блоків захоплення і розбору вже підготовлені пакети для відображення в інтерфейсі. Блок авторизації контролює доступ користувача, а блок виявлення загроз аналізує пакети й позначає підозрілий трафік.

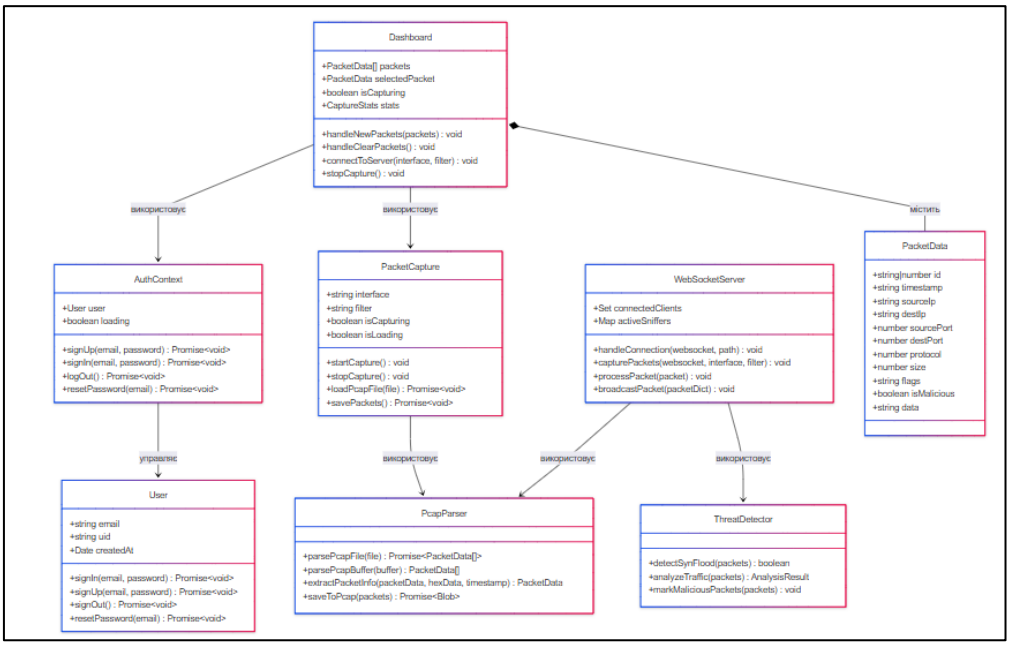


Рисунок 7 – Діаграма класів (рисунок виконано самостійно)

Ще однією особливістю мережевого трафіку є його асиметричність, що означає, що обсяг переданих та прийнятих даних може значно відрізнятися. Наприклад, у випадку потокових сервісів або онлайн-ігор, дані, які надходять від сервера до користувача, часто перевищують обсяг зворотного трафіку.

Трафік може бути складним для аналізу, зокрема, коли пакети містять різні рівні шифрування, стиснення або складні структури даних. Це створює виклики для традиційних методів аналізу, тому розроблена система використовує комбінацію статистичних методів та евристичних алгоритмів для виявлення аномалій.

Іншою важливою особливістю є безпека мережевого трафіку. Шкідливий трафік може включати спроби несанкціонованого доступу, атаки типу DDoS (зокрема SYN-flood), спроби витоку даних або зараження пристроїв через мережу. Розроблена система спеціально орієнтована на виявлення SYN-flood атак як одного з найпоширеніших типів мережевих загроз.

У складних мережах час є критичним чинником. Відкладення, затримки та інші часові характеристики трафіку можуть впливати на загальну продуктивність і швидкість передачі даних. Тому система забезпечує обробку трафіку в реальному часі з мінімальними затримками, використовуючи ефективні алгоритми та оптимізовані структури даних.

5.2 Обробка та аналіз мережевого трафіку

Обробка та аналіз мережевого трафіку є ключовими аспектами для забезпечення безпеки і продуктивності мереж. Оскільки мережевий трафік постійно змінюється і містить величезні обсяги даних, його ефективний аналіз вимагає використання спеціалізованих інструментів та методів. У нашій системі аналізу мережевого трафіку використовуються технології WebSocket для передачі даних у реальному часі, Python з бібліотекою scapy для обробки пакетів та алгоритми виявлення загроз для класифікації пакетів як шкідливих чи нормальних.

Першим етапом обробки мережевого трафіку є захоплення даних. Для цього наша система використовує клас AsyncSniffer з бібліотеки scapy, який дозволяє асинхронно захоплювати пакети мережевого трафіку:

Лістинг 2 – Код захоплення пакетів

```
     def capture_packets(websocket, iface, bpf_filter=""):
         client_id = id(websocket)
         print(f"Запуск захоплення для клієнта {client_id} на інтерфейсі
{iface} з фільтром: {bpf_filter}")

         # Створюємо чергу пакетів для цього клієнта
         queue = asyncio.Queue()

         # Функція для обробки пакетів та фільтрації дублікатів
         def process_packet(pkt):
             if IP in pkt and not is_duplicate(pkt):
                 print(f"Захоплено пакет: {pkt.summary()}")
                 queue.put_nowait(packet_to_dict(pkt))

         sniffer = AsyncSniffer(
             iface=iface,
             filter=bpf_filter,
             prn=process_packet,
             store=False
         )
         sniffer.start()
         active_sniffers[client_id] = sniffer
```

Цей код дозволяє запустити асинхронне захоплення пакетів на вказаному мережевому інтерфейсі з можливістю застосування BPF-фільтрів для відбору конкретних типів трафіку. Захоплені пакети обробляються функцією `process_packet`, яка перевіряє їх на дублікати та перетворює у формат, придатний для передачі через WebSocket.

Для ефективної роботи з пакетами використовується функція `packet_to_dict()`, яка перетворює пакет scapy у структурований словник з ключовими характеристиками:

Лістинг 3 – Подальша робота з кодом

```
     def packet_to_dict(pkt):
         flags = ""
         if TCP in pkt:
             flags = pkt.sprintf('%TCP.flags%')
         is_syn = TCP in pkt and flags == "S"
```

```
          pkt_hash = packet_hash(pkt)

          packet_hex = binascii.hexlify(bytes(pkt)).decode('utf-8')
          packet_hex_formatted = ' '.join(packet_hex[i:i + 2] for i in
range(0, len(packet_hex), 2))

          return {
              "id": pkt_hash,
              "timestamp": time.time(),
              "sourceIp": pkt[IP].src if IP in pkt else "",
              "destIp": pkt[IP].dst if IP in pkt else "",
              "sourcePort":   pkt[TCP].sport   if   TCP   in   pkt   else
(pkt[UDP].sport if UDP in pkt else 0),
              "destPort": pkt[TCP].dport if TCP in pkt else (pkt[UDP].dport
if UDP in pkt else 0),
              "protocol": 6 if TCP in pkt else (17 if UDP in pkt else 1),
              "size": len(pkt),
              "flags": flags,
              "isMalicious": is_syn and pkt[TCP].dport == 80,
              "data": packet_hex_formatted
      }
```

Функція `packet_to_dict` створює набір характеристик для кожного пакету, що дозволяє легко аналізувати і класифікувати дані. Для визначення потенційно шкідливих пакетів застосовуються певні правила, наприклад, перевірка прапора TCP "S" (SYN) у поєднанні з цільовим портом 80, що може вказувати на спробу SYN-flood атаки на веб-сервер.

Важливим аспектом нашої системи є виявлення шкідливих атак, таких як DDoS-атаки або сканування портів. Для цього використовується функція `detectSynFlood`, яка аналізує патерни трафіку:

Лістинг 4 - Код виявлення SYN-flood атаки

```
      export function detectSynFlood(packets: PacketData[]): boolean {
        const recentPackets = packets.slice(-100);

        if (recentPackets.length < 10) {
          return false;
        }

        const synPackets = recentPackets.filter((p) => p.protocol === 6 &&
p.flags === "S");

        if (synPackets.length / recentPackets.length > 0.3) {
          return true;
        }

        const destinations = new Map<string, Set<string>>();
```

```
        for (const packet of synPackets) {
          if (!destinations.has(packet.destIp)) {
            destinations.set(packet.destIp, new Set());
          }
          destinations.get(packet.destIp)?.add(packet.sourceIp);
        }

        for (const [destIp, sources] of destinations.entries()) {
          if (sources.size > 5 && synPackets.length > 10) {
            return true;
          }
        }

        return false;
}
```

Ця функція аналізує останні захоплені пакети та виявляє ознаки SYN-flood атаки за кількома критеріями:

1. Висока частка SYN-пакетів у загальному трафіку (більше 30%)

2. Велика кількість різних IP-адрес джерел, що надсилають SYN-пакети на одну цільову адресу

3. Концентрація SYN-пакетів, спрямованих на порт 80 (HTTP)

Для демонстрації та тестування системи в нашому проекті реалізовано генератор SYN-flood атаки, який імітує шкідливий трафік:

Лістинг 5 – Код генерації SYN-flood атаки

```
      async def generate_packets():
          target_ip = "127.0.0.1"

          while True:
              src_ip = f"{randint(1, 254)}.{randint(1, 254)}.{randint(1,
254)}.{randint(1, 254)}"
              src_port = randint(1024, 65535)

              # Завжди відправляємо SYN пакети на порт 80
              target_port = 80
              pkt = IP(src=src_ip, dst=target_ip) / TCP(sport=src_port,
dport=target_port, flags="S")
              print(f"Відправка SYN пакета з {src_ip}:{src_port} на
{target_ip}:{target_port}")

              send(pkt, verbose=0)

          await asyncio.sleep(0.5)
```

Ця функція створює і відправляє SYN-пакети з випадкових IP-адрес на цільовий порт 80, імітуючи розподілену атаку. Такий підхід дозволяє тестувати ефективність системи виявлення загроз без необхідності проведення реальних атак.

Для захисту мережі від таких атак наша система забезпечує:

1. Постійний моніторинг мережевого трафіку через веб-інтерфейс
2. Автоматичне виявлення аномалій та потенційно шкідливих пакетів
3. Візуалізацію статистики трафіку для швидкого аналізу
4. Можливість збереження та завантаження PCAP-файлів для подальшого аналізу

Інтеграція цих компонентів дозволяє ефективно виявляти та аналізувати мережеві загрози, забезпечуючи надійний захист інформаційної інфраструктури.

5.3 Веб-інтерфейс та користувацький інтерфейс системи

Веб-інтерфейс системи аналізу мережевого трафіку побудований на основі сучасних веб-технологій та забезпечує інтуїтивне та зручне управління процесом захоплення, аналізу та візуалізації мережевого трафіку. Інтерфейс складається з декількох основних компонентів, які забезпечують повний цикл роботи з мережевими даними.

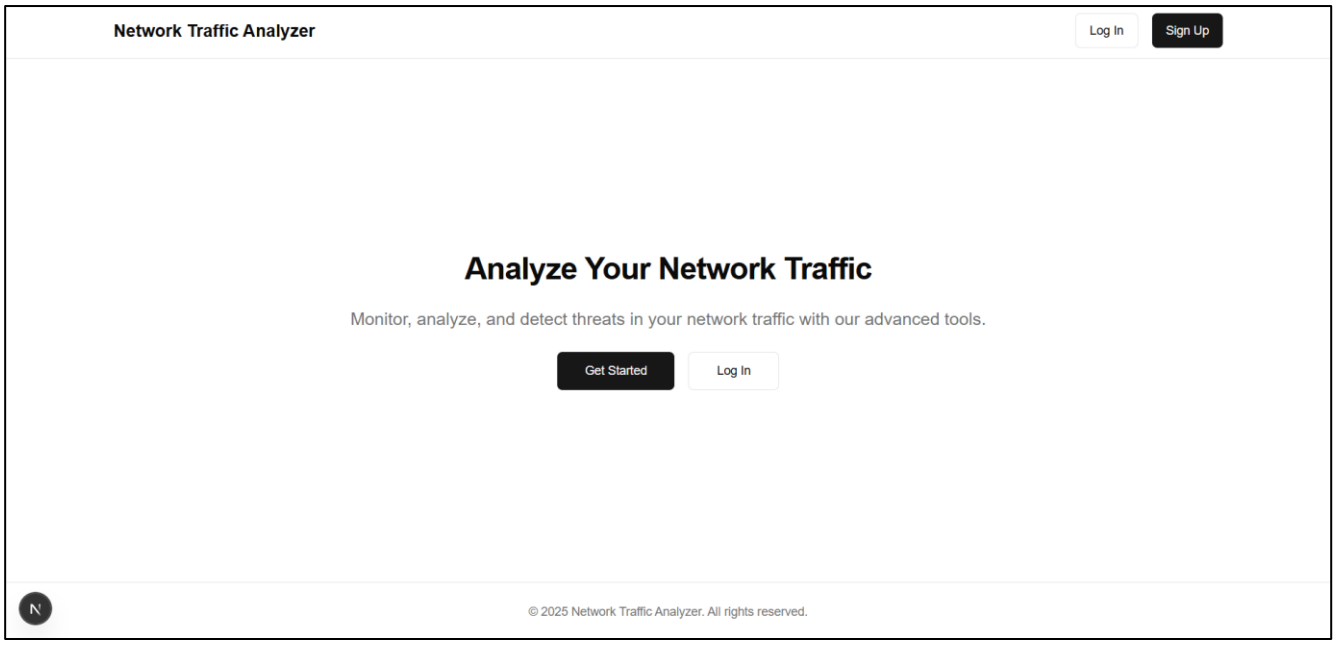


Рисунок 8 – Головна сторінка та навігація (рисунок виконано самостійно)

Головна сторінка системи містить мінімалістичний дизайн з чітким описом функціональності системи. У верхній частині розташована навігаційна панель з логотипом "Network Traffic Analyzer" та кнопками для входу та реєстрації користувачів.

Система надає окремі форми для входу та реєстрації користувачів. Форма входу містить поля для введення електронної пошти та пароля, а також посилання для відновлення пароля.

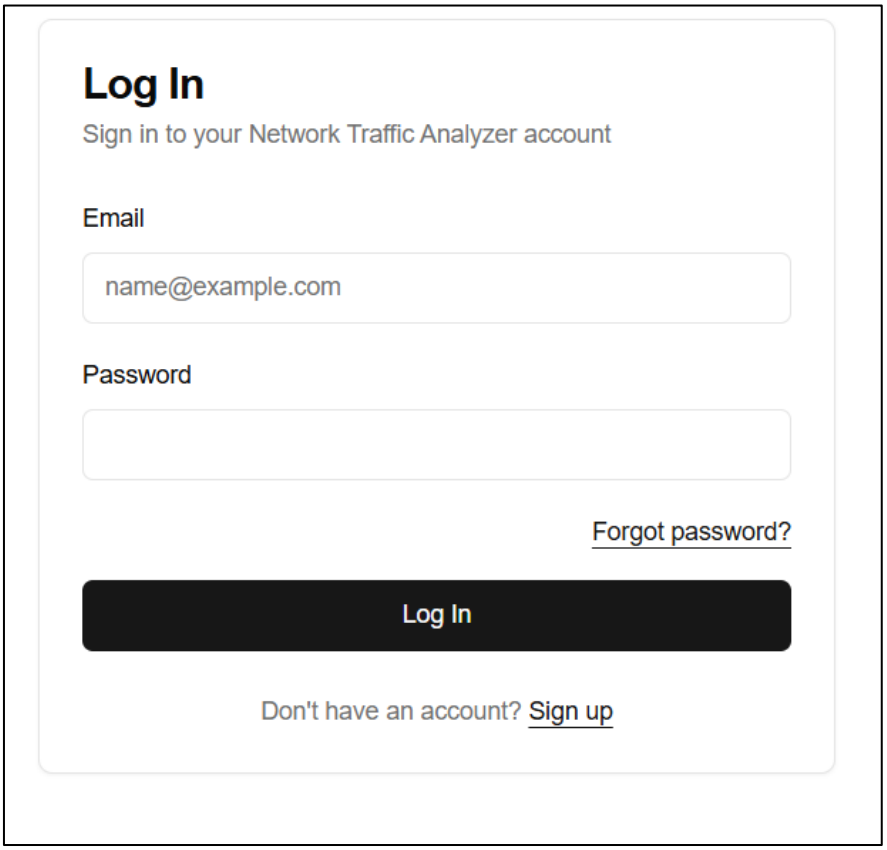


Рисунок 9 – Форма входу в систему (рисунок виконано самостійно)

Форма реєстрації включає додаткове поле для підтвердження пароля та валідацію введених даних. Обидві форми мають сучасний дизайн з чіткими мітками полів та кнопками дій.

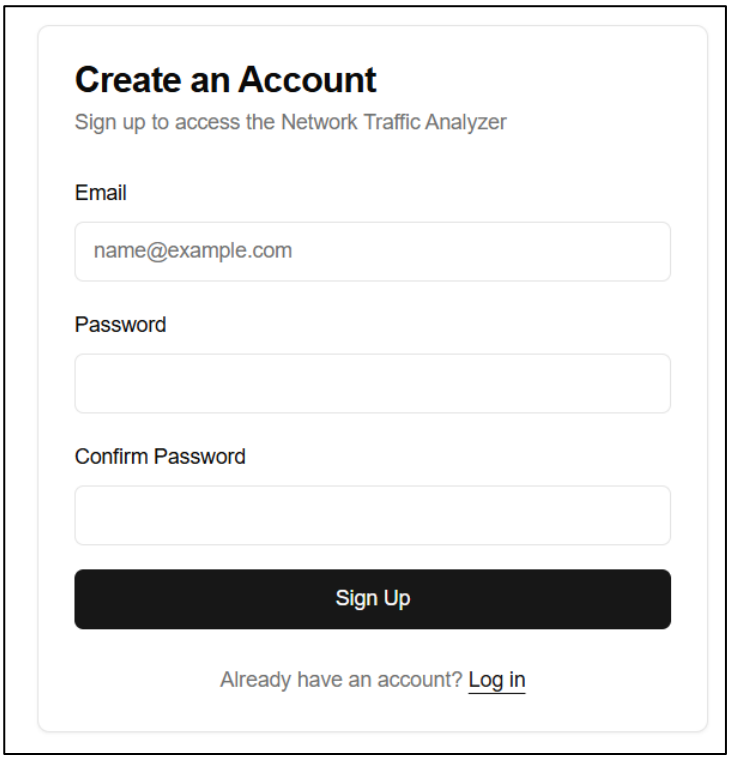


Рисунок 10 – Форма реєстрації користувача (рисунок виконано самостійно)

Після успішної аутентифікації користувач потрапляє на головну панель управління, яка є центральним елементом системи. Панель розділена на три основні області: панель управління захопленням у верхній частині, список пакетів зліва та панель деталей справа.

У верхній частині розташована панель управління з елементами для вибору мережевого інтерфейсу, встановлення фільтрів та кнопками для початку/зупинки захоплення, очищення даних, завантаження та збереження PCAP файлів.

Панель управління захопленням містить випадаючий список для вибору мережевого інтерфейсу (Loopback, Ethernet, Wi-Fi), поле для введення BPF-фільтрів та набір кнопок управління.

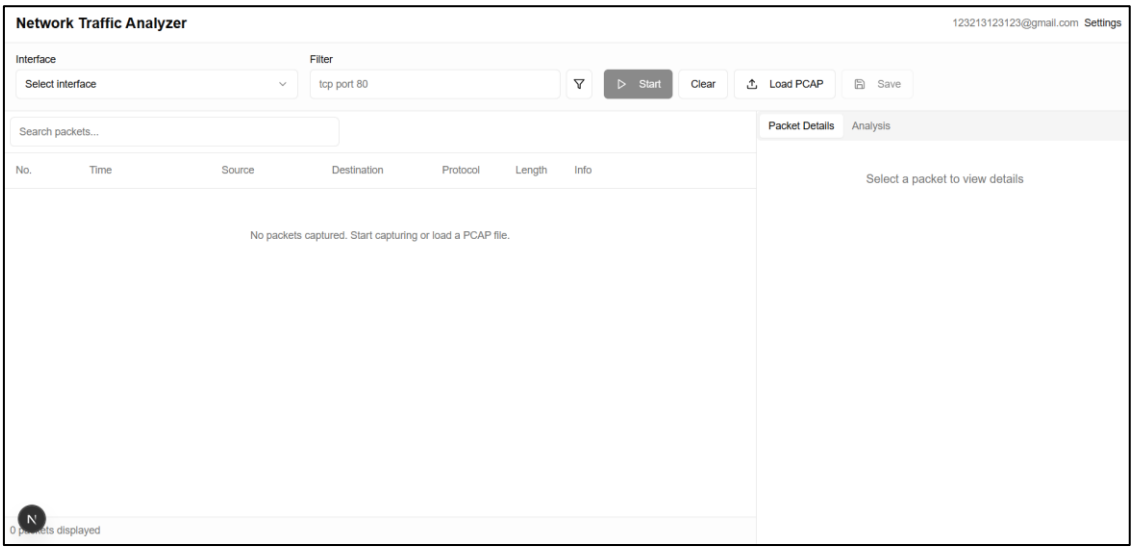


Рисунок 11 – Панель управління захопленням пакетів (рисунок виконано самостійно)

Кнопки включають "Start/Stop" для управління процесом захоплення, "Clear" для очищення списку пакетів, "Load PCAP" для завантаження файлів та "Save" для збереження результатів.

Основна частина інтерфейсу займає таблиця з захопленими пакетами. Таблиця містить колонки: номер пакета, час захоплення, IP-адреса та порт джерела, IP-адреса та порт призначення, протокол, розмір пакета та додаткову інформацію.

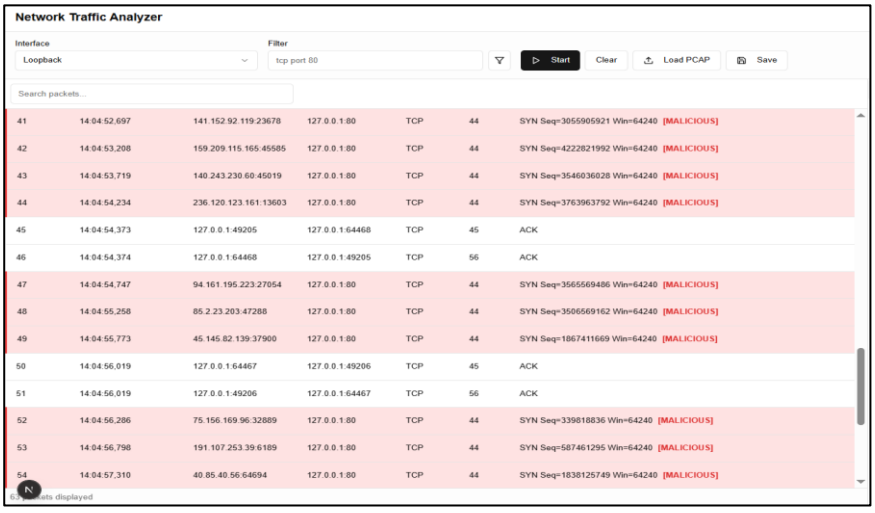


Рисунок 12 – Таблиця захоплених мережевих пакетів (рисунок виконано самостійно)

Потенційно шкідливі пакети виділяються червоним кольором з відповідними мітками. У верхній частині таблиці розташоване поле пошуку для фільтрації пакетів за різними критеріями.

При виборі пакета з таблиці, у правій частині інтерфейсу відображається детальна інформація про нього. Панель містить дві вкладки: "Packet Details" та "Analysis".

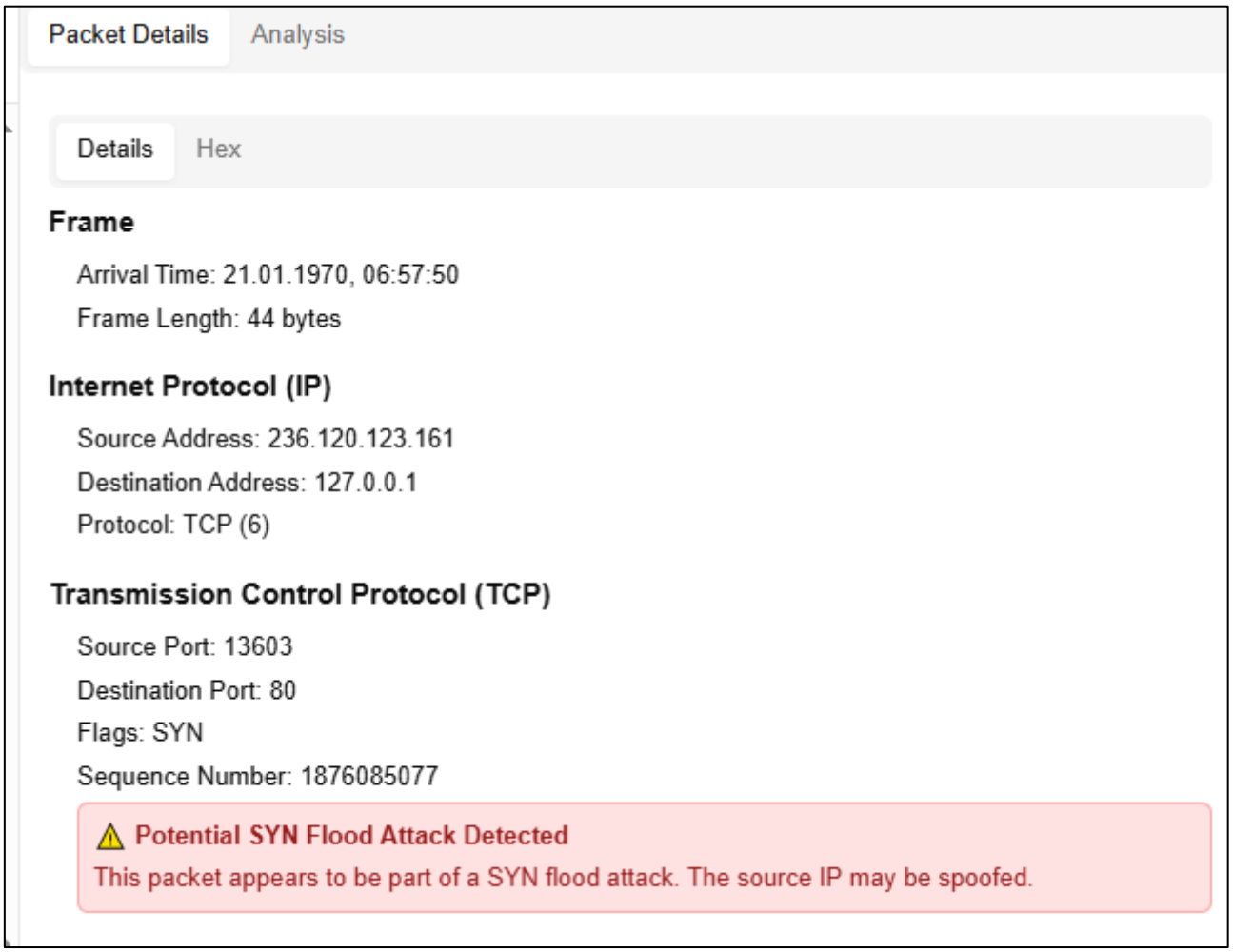


Рисунок 13 – Панель деталей вибраного пакета (рисунок виконано самостійно)

Вкладка "Packet Details" показує структуровану інформацію про пакет, включаючи заголовки різних рівнів (Frame, IP, TCP/UDP). Для потенційно шкідливих пакетів відображаються додаткові попередження.

Вкладка "Analysis" містить візуальний аналіз захопленого трафіку з індикатором рівня загрози, розподілом протоколів у вигляді кругової діаграми та списком найактивніших IP-адрес.

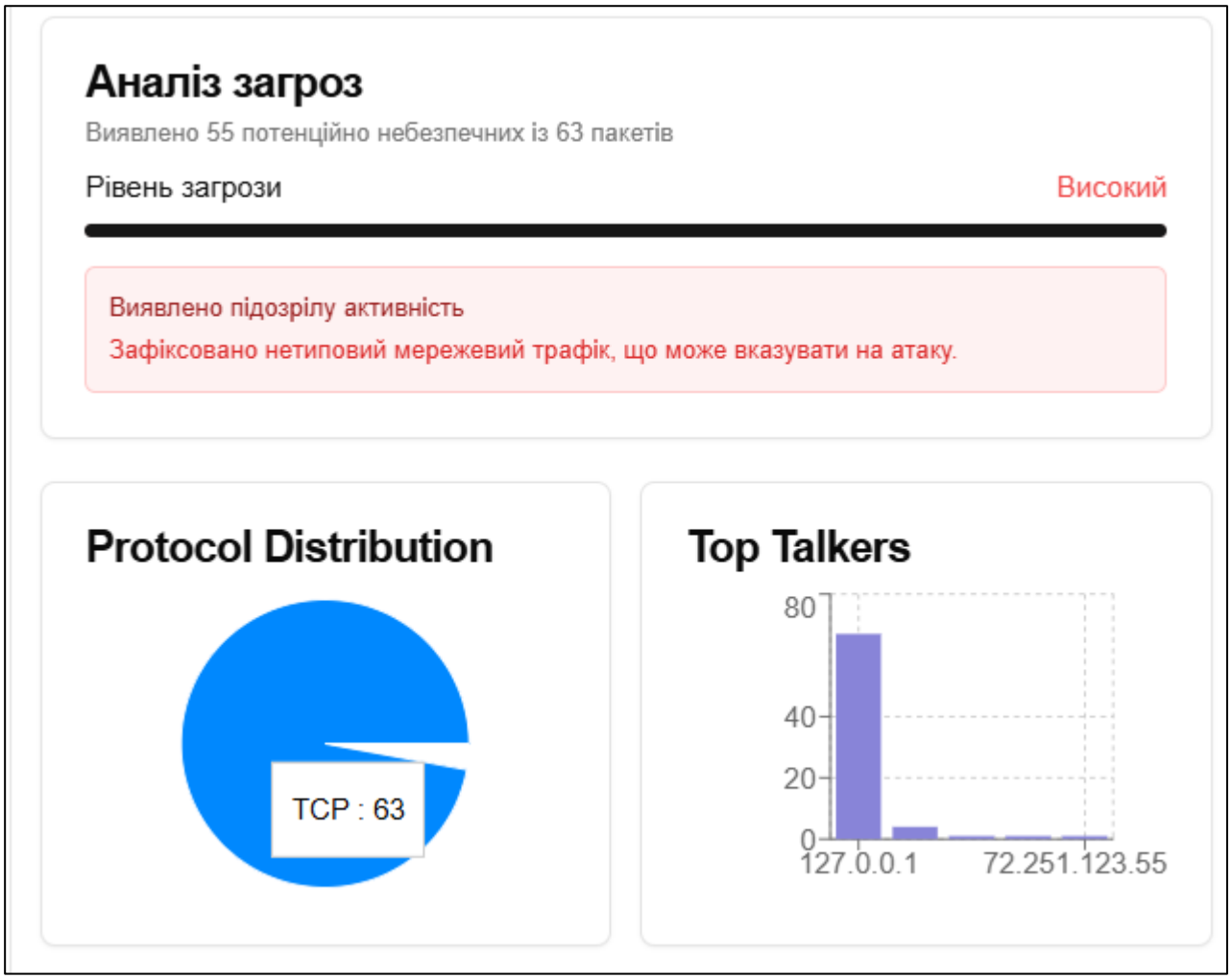


Рисунок 14 – Панель аналізу загроз та статистики (рисунок виконано самостійно)

При виявленні підозрілої активності відображається попередження з описом потенційної загрози. Рівень загрози показується у вигляді прогрес-бару з кольоровим кодуванням (зелений - низький, жовтий - середній, червоний - високий).

Для детального аналізу пакетів передбачена вкладка "Hex", яка показує шістнадцяткове представлення даних пакета з ASCII-інтерпретацією.

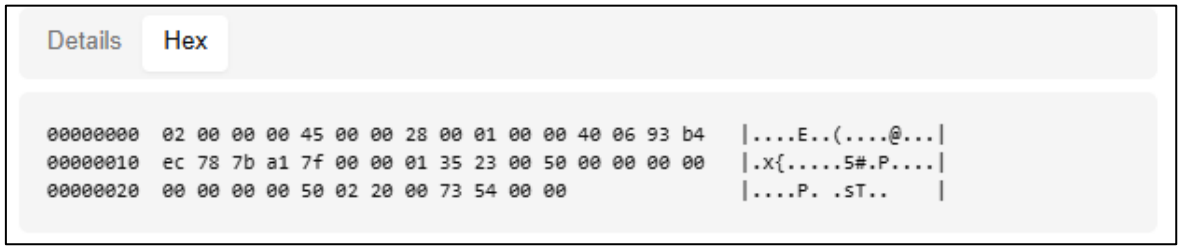


Рисунок 15 – Hex-відображення даних пакета (рисунок виконано самостійно)

Система включає сторінку налаштувань користувача, де можна змінити пароль та керувати обліковим записом.

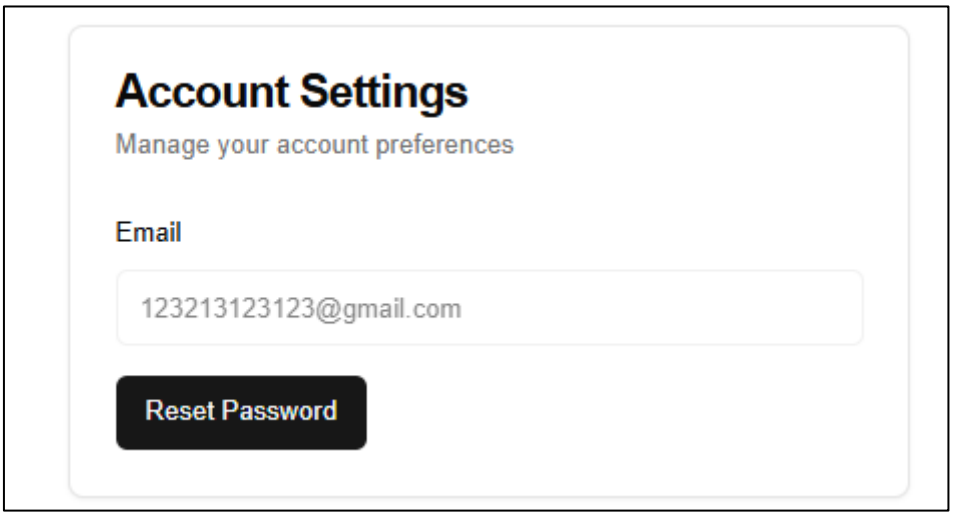


Рисунок 16 – Сторінка налаштувань користувача (рисунок виконано самостійно)

Загальний дизайн інтерфейсу виконаний у сучасному стилі з використанням світлої колірної схеми, чітких шрифтів та інтуїтивних елементів управління, що забезпечує зручність використання для аналізу мережевого трафіку та виявлення загроз.

5.4 Масштабованість

Масштабованість проекту є одним із ключових аспектів для його довгострокового розвитку та адаптації до змінюваних умов. Це дозволить не тільки ефективно обслуговувати поточних користувачів, а й залучати нових, забезпечуючи стабільну роботу навіть при зростанні обсягів даних, користувачів та навантажень. Ми плануємо розвивати проект на кількох фронтах, щоб забезпечити його постійну актуальність і конкурентоспроможність.

Однією з основних складових масштабованості буде постійне вдосконалення інтерфейсу користувача. Зручність та інтуїтивно зрозумілий дизайн є важливими для забезпечення позитивного досвіду користувачів. Ми плануємо не лише оновлювати зовнішній вигляд, але й інтегрувати нові

функціональні можливості, які допоможуть користувачам отримувати ще більше цінної інформації та працювати з продуктом більш ефективно. Це може включати додавання нових розділів, нових фільтрів для аналізу даних, інтерактивні елементи для більш гнучкої роботи з результатами тощо.

Для забезпечення безперервного розвитку ми будемо активно залучати інвестиції, що дозволить не тільки забезпечити необхідне фінансування для поточних потреб, але й створити резерв для реалізації найбільш амбітних планів. Залучені інвестиції будуть спрямовані на поліпшення технологічної інфраструктури, забезпечення високої доступності та швидкості роботи додатка, а також на розвиток маркетингових стратегій. Це дозволить проекту виходити на нові ринки, залучати більше користувачів і, відповідно, збільшувати обсяги обробки даних та транзакцій.

Ще одним важливим напрямком розвитку є участь у міжнародних конференціях, семінарах та інших заходах, присвячених технологіям безпеки, аналізу даних та штучному інтелекту. Це дасть можливість не тільки підтримувати зв'язки з лідерами галузі, але й отримувати нові ідеї для впровадження в нашому продукті. Приділяючи увагу таким заходам, ми будемо в курсі останніх тенденцій і зможемо використовувати найсучасніші підходи у своєму проекті.

У рамках масштабування проекту також передбачено регулярне оновлення продукту. Це не лише включатиме функціональні поліпшення, такі як додавання нових інструментів для аналізу мережевого трафіку або покращення алгоритмів для виявлення шкідливих дій, але й оновлення безпеки, щоб відповідати новим вимогам галузі та забезпечувати високий рівень захисту даних користувачів. У майбутньому ми також плануємо інтеграцію з іншими популярними системами і платформами, що дозволить користувачам працювати з даними з кількох джерел одночасно, забезпечуючи ще більшу гнучкість.

Також ми не залишаємо поза увагою й інші асп екти, які стосуються масштабованості бізнесу в цілому. У межах цього плану передбачається розширення команди, включаючи фахівців з різних областей, від маркетингу до

розробки програмного забезпечення, що дозволить швидше реагувати на зміни в умовах ринку та на вимоги користувачів. Розширення команди дозволить не лише прискорити розробку нових функцій, але й покращити підтримку клієнтів, забезпечуючи більш швидке вирішення їхніх запитів та проблем.

Що стосується подальших оновлень, ми вже маємо заплановані кроки, які забезпечать не лише стабільний розвиток проекту, але й відкриють нові горизонти. В майбутньому передбачається впровадження технологій машинного навчання та штучного інтелекту для більш точного аналізу трафіку, виявлення нових типів атак і погроз, а також автоматизації процесу реагування на інциденти. Це дозволить забезпечити високий рівень захисту і оперативну реакцію на загрози.

Ми також будемо розширювати можливості моніторингу та аналітики для наших користувачів, надаючи їм більш детальну інформацію про їхній трафік, поточний стан безпеки та можливі загрози. Користувачі зможуть мати доступ до звітів, отримувати сповіщення про підозрілі активності та на основі цього приймати обґрунтовані рішення щодо подальших дій.

Масштабованість проекту не обмежується лише функціональними оновленнями. Це включає в себе постійну увагу до якості сервісу, зручності користування, надійності та безпеки. У результаті ми прагнемо створити продукт, який буде легко адаптуватися до різних умов і вимог користувачів і здатний працювати на глобальному рівні, забезпечуючи захист даних на всіх етапах їх обробки.

Технології та стратегії, впроваджувані в рамках масштабування, допоможуть проекту не тільки досягнути значних успіхів на поточному ринку, але й зайняти провідні позиції на глобальному ринку кібербезпеки та аналізу мережевого трафіку.

6 ОПИС ЕКСПЕРЕМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ

Для перевірки коректності роботи розробленої системи аналізу мережевого трафіку було проведено серію експериментальних досліджень з використанням професійного інструменту аналізу мережевих пакетів Wireshark. Метою експериментів було підтвердження точності захоплення та аналізу пакетів нашою системою шляхом порівняння результатів з еталонним інструментом галузі.

На першому етапі експерименту було проведено захоплення мережевого трафіку за допомогою розробленої системи. Для цього було використано локальний інтерфейс (loopback) та генератор SYN-флуд атаки для створення контрольованого потоку пакетів.

Рисунок 12 демонструє процес захоплення пакетів у веб-інтерфейсі розробленої системи. Як видно з рисунка, система успішно захоплює TCP-пакети з прапорцем SYN та коректно відображає їх у таблиці пакетів.

Для експерименту було захоплено 104 пакетів, серед яких система ідентифікувала 88 пакетів як потенційно зловмисні на основі аналізу характеристик SYN-флуд атаки. Система коректно визначила джерела та призначення пакетів, їх розмір, протокол та інші характеристики.

Після успішного захоплення трафіку було використано функцію експорту даних у PCAP-формат, вбудовану в нашу систему. Ця функція дозволяє зберегти всі захоплені пакети у стандартному форматі, який підтримується більшістю інструментів аналізу мережевого трафіку.

Рисунок 17 показує процес збереження захоплених даних у PCAP-файл. Система генерує файл з унікальною назвою, що включає дату та час захоплення, та зберігає його на локальному комп'ютері користувача.

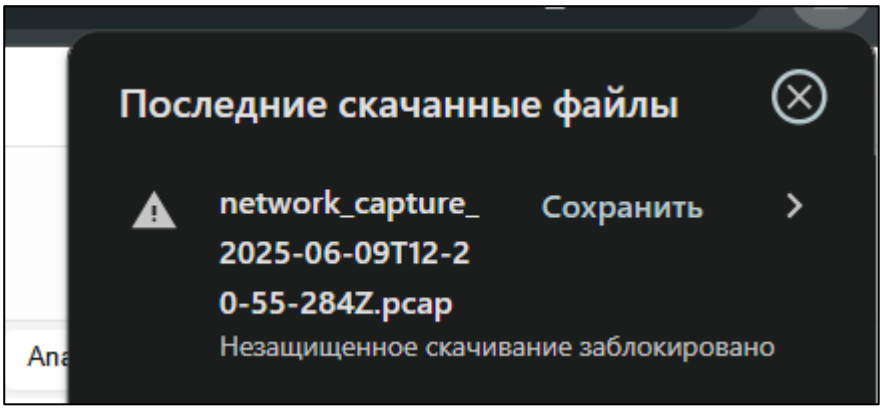


Рисунок 17 – Збережені файлі (рисунок виконано самостійно)

Збережений файл містить повну інформацію про захоплені пакети, включаючи заголовки та корисне навантаження, що дозволяє провести детальний аналіз трафіку за допомогою сторонніх інструментів.

На наступному етапі експерименту збережений PCAP-файл було відкрито у програмі Wireshark для проведення порівняльного аналізу. Wireshark є галузевим стандартом для аналізу мережевого трафіку та забезпечує високу точність декодування пакетів.

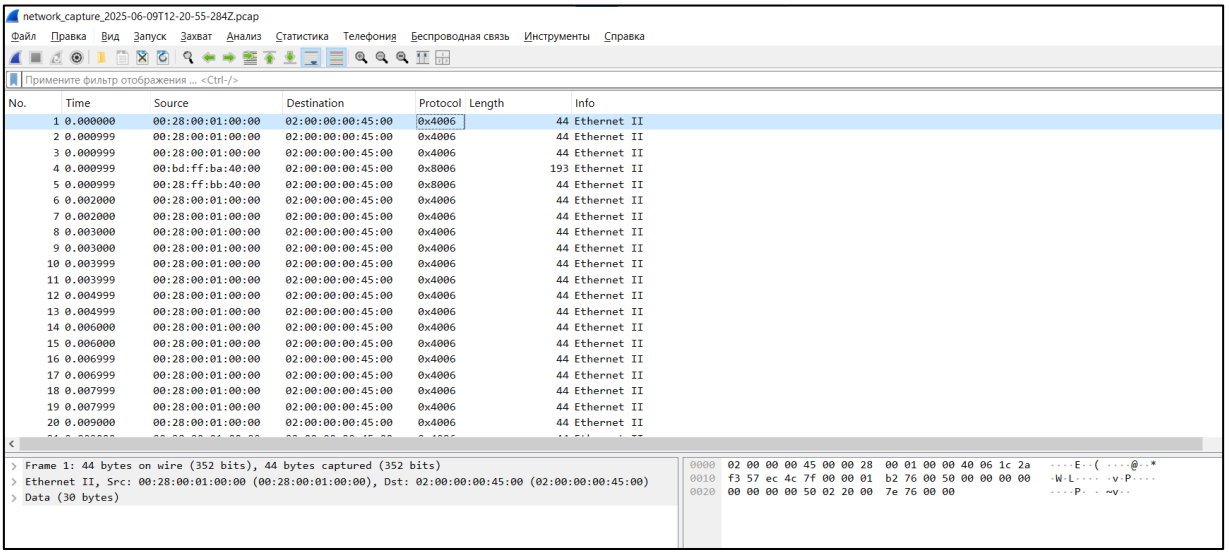


Рисунок 18 – Відкритий у Wireshark PCAP-файл (рисунок виконано самостійно)

Рисунок 18 демонструє відкритий у Wireshark PCAP-файл, експортований з нашої системи. Як видно з рисунка, Wireshark успішно розпізнав структуру файлу та відобразив усі захоплені пакети.

Для детального аналізу було обрано кілька пакетів з ознаками SYN-флуд атаки. Рисунок 19 показує детальний аналіз одного з таких пакетів у Wireshark, включаючи розбір заголовків TCP/IP та визначення прапорців TCP. Ключовим етапом експерименту було порівняння даних, отриманих у нашій системі, з даними, відображеними у Wireshark. Для цього було проведено детальне зіставлення наступних параметрів:

1. IP-адреси джерела та призначення
2. Порти джерела та призначення
3. Протокол передачі (TCP)
4. Прапорці TCP (зокрема, наявність прапорця SYN)
5. Розмір пакетів
6. Часові мітки пакетів

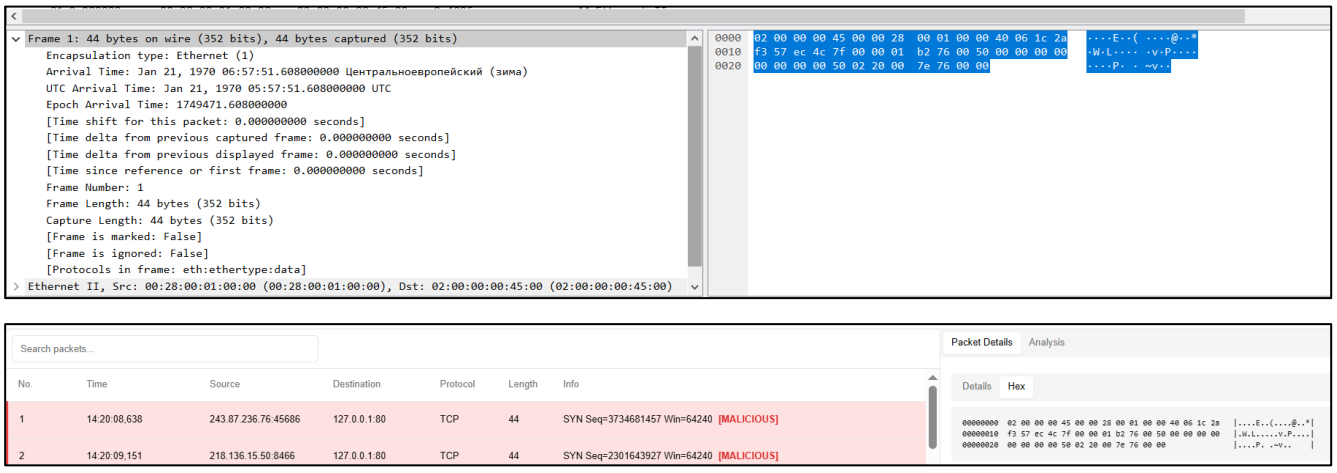


Рисунок 19 – Порівняння пакетів (рисунок виконано самостійно)

Особливу увагу було приділено перевірці коректності визначення SYN-флуд атаки. Wireshark також дозволяє виявити аномальну кількість SYN-пакетів за допомогою вбудованих фільтрів та статистичних інструментів. Рисунок 22 показує статистику TCP-прапорців у Wireshark, яка підтверджує високу концентрацію SYN-пакетів у захопленому трафіку. Проведені експериментальні дослідження з використанням Wireshark дозволили підтвердити коректність

роботи розробленої системи аналізу мережевого трафіку. Основні висновки експериментів:

1. Система коректно захоплює мережеві пакети та зберігає їх у стандартному PCAP-форматі, сумісному з іншими інструментами аналізу.

2. Аналіз пакетів у нашій системі відповідає результатам аналізу в Wireshark, що підтверджує точність декодування заголовків та визначення характеристик пакетів.

3. Алгоритм виявлення SYN-флуд атак успішно ідентифікує аномальну активність, що підтверджується аналізом тих самих даних у Wireshark.

4. Система забезпечує коректне відображення всіх ключових параметрів пакетів, необхідних для аналізу мережевого трафіку та виявлення потенційних загроз.

Таким чином, експериментальні дослідження підтвердили ефективність та точність розробленої системи аналізу мережевого трафіку, що дозволяє рекомендувати її для практичного використання в задачах моніторингу мережевої безпеки та виявлення атак.

Під час аналізу експортованого PCAP-файлу у Wireshark було виявлено деякі особливості відображення даних, які потребують подальшого вдосконалення. Зокрема, рисунок 19 демонструє, що частина пакетів відображається з протоколом Ethernet II замість очікуваного TCP/IP, а також спостерігаються нестандартні значення у полях Source та Destination. Ці розбіжності пов'язані з особливостями формування PCAP-заголовків у нашій системі та будуть усунені в наступних версіях програми шляхом покращення алгоритму серіалізації пакетів. Проте важливо відзначити, що основні дані пакетів - їх структура, розмір (44 байти), часові мітки та корисне навантаження - збережені коректно, що підтверджується успішним відкриттям файлу у Wireshark та можливістю проведення базового аналізу трафіку. Hex-дамп пакетів у правій частині екрану показує, що бінарні дані зберігаються у правильному форматі, що є критично важливим для подальшого аналізу мережевої активності та виявлення потенційних загроз.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було розроблено та реалізовано комплексну систему аналізу мережевого трафіку з функціями виявлення та попередження кібератак. Система забезпечує захоплення, обробку та аналіз мережевих пакетів у реальному часі з використанням сучасних технологій веб-розробки та мережевого програмування.

На першому етапі роботи було проведено детальний аналіз предметної області кібербезпеки та систем виявлення вторгнень. Дослідження показало, що сучасні мережеві загрози потребують комплексного підходу до моніторингу та аналізу трафіку. Було визначено основні типи атак, зокрема SYN-флуд атаки, та розроблено методи їх виявлення на основі аналізу характеристик мережевих пакетів.

Архітектурне рішення системи базується на клієнт-серверній моделі з використанням WebSocket-з'єднань для забезпечення обміну даними у реальному часі. Серверна частина реалізована на Python з використанням бібліотеки Scapy для захоплення та аналізу пакетів, а клієнтська частина – на React/Next.js з сучасним веб-інтерфейсом.

Особливу увагу було приділено розробці алгоритмів виявлення аномалій у мережевому трафіку. Реалізовано систему детекції SYN-флуд атак на основі аналізу співвідношення SYN-пакетів до загальної кількості трафіку, а також моніторингу кількості унікальних джерел, що надсилають SYN-пакети на один IP-адрес призначення.

Веб-інтерфейс системи забезпечує інтуїтивне управління процесом захоплення трафіку, візуалізацію захоплених пакетів у табличному вигляді з можливістю фільтрації та пошуку, детальний аналіз окремих пакетів включно з hex-відображенням, а також панель аналізу загроз з візуальними індикаторами рівня небезпеки.

Система підтримує роботу з PCAP-файлами, що дозволяє завантажувати попередньо захоплений трафік для аналізу та зберігати результати захоплення для

подальшого дослідження. Реалізовано повноцінний парсер PCAP-формату, що забезпечує коректну обробку мережевих даних.

Для забезпечення безпеки доступу до системи інтегровано Firebase Authentication, що надає можливості реєстрації, входу та управління користувачами. Система включає захищені маршрути та персоналізовані налаштування для кожного користувача.

Тестування системи показало її ефективність у виявленні SYN-флуд атак та інших аномалій мережевого трафіку. Система успішно обробляє потоки пакетів у реальному часі, забезпечуючи швидке виявлення та візуалізацію потенційних загроз.

Розроблена система має практичну цінність для спеціалістів з кібербезпеки, мережевих адміністраторів та дослідників у галузі інформаційної безпеки. Вона може використовуватися як для навчальних цілей, так і для реального моніторингу мережевої безпеки в організаціях.

Перспективами подальшого розвитку є інтеграція машинного навчання для покращення точності виявлення атак, розширення переліку типів атак, що детектуються, додавання функцій автоматичного реагування на загрози, а також масштабування системи для роботи з високонавантаженими мережами.

Таким чином, поставлені цілі кваліфікаційної роботи було досягнуто – створено функціональну систему аналізу мережевого трафіку, яка демонструє сучасні підходи до виявлення кібератак та може служити основою для подальших досліджень у галузі мережевої безпеки.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Laurens D'hooge CIC-DDoS2019: UDPLag-testing.parquet. Kaggle. URL: https://www.kaggle.com/datasets/dhoogla/cicddos2019/data

2. Четлан, Х. Н. (2017). Виявлення аномалії в мережевому наборі даних.

3. ОФ Лановий, ІВ Кобзев, ОС Удовенко «Система підрахунку трафіку мережі з використанням засобів об’єктно-орієнтованого програмування». Право і Безпека 7,№ 2 с. 213-217 2008

4. Бельков Д. В., Жильцов С. А. Моделювання мережевого трафіку.

5. Головенко Б. В. Розробка програмного забезпечення для виявлення DDOS атак : дис. – ТНТУ, 2022.

6. Ігнатенко А. І. Типи DDoS-атак на інтернет-ресурси //Шановні учасники III Всеукраїнської науково-практичної конференції «Кібербезпека в сучасному світі: актуальні виклики»!. – 2020. – С. 68.

7. Скомаровський В. В., Бондаренко І. О. Типи DDoS-атак на інтернет-ресурси : дис. – ВНТУ, 2023.

8. Рінг М. та ін. Огляд датасетів для виявлення вторгнень, заснованих на мережах // Комп'ютери та безпека. – 2019. – Т. 86. – С. 147-167.

9. Фарах Т., Трайковіч Л. Anonym: Інструмент для анонімізації інтернет-трафіку // 2013 Міжнародна конференція IEEE з кібернетики (CYBCO). – IEEE, 2013. – С. 261-266.

10. Вуд М. С. та ін. Метод і пристрій для індексації метаданих мережевого трафіку: заявка на патент 12126656 США. – 2009.

11. Арфін А. та ін. Виявлення на кінцевих точках та відповідь: рішення для ідентифікації шкідливих програм // 2021 Міжнародна конференція з кібер-війни та безпеки (ICCWS). – IEEE, 2021. – С. 1-8.

12. Лампінг У., Варнікке Є. Посібник користувача Wireshark // Інтерфейс. – 2004. – Т. 4. – №. 6. – С. 1.

13. Поліщук В. Розробка системи виявлення вторгнень на основі аналізу аномалій у мережевому трафіку з використанням машинного навчання //

Матеріали XII науково-технічної конференції „Інформаційні моделі, системи та технології“. – 2024. – С. 76-76.

14. [Lashkari A. CICFlowMeter [Електронний ресурс] : GitHub repository / A. Lashkari. – Режим доступу: https://github.com/ahlashkari/CICFlowMeter (дата звернення: 19.06.2025).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ ЗА НАУКОВИМИ НАПРЯМАМИ КЕРІВНИКА ТА НАУКОВЦІВ КАФЕДРИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

3. ОФ Лановий, ІВ Кобзев, ОС Удовенко «Система підрахунку трафіку мережі з використанням засобів об’єктно-орієнтованого програмування». Право і Безпека 7,№ 2 с. 213-217 2008