**1 ВСТУП**

1.1 Огляд продукту

Розроблена система дозволяє створювати, поповнювати й анотувати датасети мережевого трафіку для подальшого використання у задачах виявлення атак. Платформа забезпечує обробку як реального, так і симульованого трафіку з підтримкою фільтрації, нормалізації та агрегування. Передбачено інтеграцію з інструментами захоплення даних, а також підтримку гібридного збору з мережевих потоків, логів і пкепів.

1.2 Мета

Метою системи є створення високоякісного, збалансованого та адаптивного датасету для навчання моделей виявлення зловмисного трафіку. Особливу увагу приділено автоматизації процесу збору, анотації та обробки трафіку для формування актуальних даних під сучасні кіберзагрози.

1.3 Межі

Архітектура обробки:

– система реалізована на Python з використанням Scapy, pandas, sklearn, joblib;

– передбачено обробку PCAP-файлів та NetFlow-даних для формування структурованих таблиць;

– реалізовані модулі попередньої обробки: фільтрація, агрегація, нормалізація, виділення ознак;

– підтримка форматів збереження: CSV, Parquet, Pickle;

– обробка здійснюється офлайн з подальшою валідацією моделей;

– результати зберігаються разом із метаданими: тип трафіку, джерело, мітки.

Алгоритмічна частина:

– реалізовано підтримку моделей Random Forest, SVM, нейронних мереж;

– можливість імпорту та оцінки публічних датасетів;

– реалізовані метрики: Precision, Recall, F1, затримка класифікації.

Функціональні обмеження:

– відсутність обробки зашифрованого трафіку;

– система не працює у реальному часі;

– класифікація тільки між «нормальним» і «зловмисним» трафіком без поділу на типи атак.

1.4 Посилання

В проєкті були використані такі посилання:

1. Четлан, Х. Н. (2017). Виявлення аномалії в мережевому наборі даних.
2. ОФ Лановий, ІВ Кобзев, ОС Удовенко «[Система підрахунку трафіку мережі з використанням засобів об’єктно-орієнтованого програмування](http://www.irbis-nbuv.gov.ua/cgi-bin/irbis_nbuv/cgiirbis_64.exe?C21COM=2&I21DBN=UJRN&P21DBN=UJRN&IMAGE_FILE_DOWNLOAD=1&Image_file_name=PDF/Pib_2008_7_2_46.pdf)». Право і Безпека 7,№ 2 с. 213-217 2008
3. Бельков Д. В., Жильцов С. А. Моделювання мережевого трафіку.
4. Головенко Б. В. Розробка програмного забезпечення для виявлення DDOS aтак : дис. – ТНТУ, 2022.
5. Ігнатенко А. І. Типи DDoS-атак на інтернет-ресурси //Шановні учасники ІІІ Всеукраїнської науково-практичної конференції «Кібербезпека в сучасному світі: актуальні виклики»!. – 2020. – С. 68.
6. Скомаровський В. В., Бондаренко І. О. Типи DDoS-атак на інтернет-ресурси : дис. – ВНТУ, 2023.
7. Рінг М. та ін. Огляд датасетів для виявлення вторгнень, заснованих на мережах // Комп'ютери та безпека. – 2019. – Т. 86. – С. 147-167. [1]
8. Фарах Т., Трайковіч Л. Anonym: Інструмент для анонімізації інтернет-трафіку // 2013 Міжнародна конференція IEEE з кібернетики (CYBCO). – IEEE, 2013. – С. 261-266.
9. Вуд М. С. та ін. Метод і пристрій для індексації метаданих мережевого трафіку: заявка на патент 12126656 США. – 2009. [3]
10. 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.
11. Лампінг У., Варнікке Є. Посібник користувача Wireshark // Інтерфейс. – 2004. – Т. 4. – №. 6. – С. 1.
12. 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.

1.5 Означення та абревіатури

|  |  |
| --- | --- |
| **Термін** | **Оновлене визначення** |
| Трафік | Потік пакетів даних, що циркулює в комп’ютерній мережі |
| PCAP | Формат збереження мережевого трафіку, що містить повні знімки пакетів |
| Метадані | Інформація про з’єднання: час, адреса, порт, протокол, тривалість |
| Зловмисний трафік | Трафік, що містить ознаки шкідливої активності: атаки, сканування, вторгнення |
| Анотація | Позначення зразків у датасеті як «нормальний» або «атака» |

2 ЗАГАЛЬНИЙ ОПИС

2.1 Перспективи продукту

* розширення спектру підтримуваних форматів трафіку та автоматизоване визначення типу протоколу;
* інтеграція інструментів реального часу для потокової обробки даних та online-аналізу;
* підтримка генерації синтетичного трафіку з параметрами, наближеними до реальних атак;
* реалізація веб-інтерфейсу для керування сценаріями збору, аналізу та валідації датасету;
* створення багатокористувацької платформи з розподілом ролей;
* впровадження механізмів автоматичної анотації трафіку за допомогою LLM або евристичних правил;
* розгортання в хмарному середовищі з масштабованою архітектурою для великих експериментів.

2.2 Функції продукту

Користувачі отримують такі можливості:

* завантаження PCAP-файлів або потокових логів для обробки;
* реєстрація та аутентифікація для персоналізованої роботи з проєктами;
* формування структурованих датасетів із фільтрацією, агрегацією та нормалізацією даних;
* анотація трафіку вручну або за допомогою автоматизованих правил;
* експорт готового датасету у форматах CSV, JSON, Parquet;
* порівняння ефективності різних методів створення датасету;
* запуск і перегляд результатів класифікації на базі тренованих моделей;
* ведення журналів обробки, зміни міток і результатів класифікації для забезпечення аудиту.

2.3 Характеристики користувачів

Адміністратори:

* керують користувачами, призначають ролі (аналітик, дослідник, оператор збору);
* налаштовують середовище обробки, правила фільтрації та політики анонімізації;
* відповідають за інтеграцію з зовнішніми джерелами трафіку та збереженням історії експериментів.

Дослідники:

* аналізують структуру мережевого трафіку, проводять анотацію та перевірку датасетів;
* тестують моделі класифікації на підготовлених наборах;
* експериментують з різними підходами до побудови датасету: PCAP, NetFlow, метадані.

Оператори збору:

* організовують процес захоплення мережевого трафіку з реальних або віртуальних середовищ;
* контролюють якість вхідних даних, джерела трафіку та коректність форматів;
* не обов’язково мають глибоку технічну підготовку, але працюють через зрозумілий інтерфейс збору.

2.4 Загальні обмеження

* наразі доступний лише веб-інтерфейс; CLI-модуль та мобільний клієнт — у розробленні;
* обробка даних здійснюється в пакетному режимі — streaming/real-time аналіз у майбутніх версіях;
* інтерфейс та документація підтримують лише українську та англійську мови.

2.5 Припущення та залежності

* потрібне стабільне інтернет-з’єднання для завантаження публічних датасетів і інтеграцій із зовнішніми сервісами;
* для збереження метаданих використовують PostgreSQL;
* для обробки даних потрібні бібліотеки Scapy, pandas, scikit-learn, joblib;

3 КОНКРЕТНІ ВИМОГИ

3.1 Вимоги до зовнішніх інтерфейсів

3.1.1 Інтерфейс користувача

* веб-інтерфейс (Next.js) для налаштування та візуалізації мережевого аналізу:
* панель запуску live-захоплення трафіку;
* перегляд списку збережених PCAP-файлів і їх попередній перегляд;
* налаштування фільтрів BPF (IP, порт, протокол) перед захопленням .
* інтерактивна консоль WebSocket у браузері для відправки команд (load\_pcap, save\_pcap) і отримання повідомлень зі sniffer-сервера.
* Fast API для конвертації pcap у flows.
* CLI-утиліта (main.py) для пакетної обробки локальних PCAP:
* генерація синтетичних SYN-пакетів через genSYN.py

3.1.2 Апаратний інтерфейс

– мережевий інтерфейс у проміск’юїтному режимі (NIC) або порт-мірування на комутаторі для захоплення трафіку;

– опціонально інструменти на кшталт Zeek (Bro) або tcpdump для передобробки вхідних PCAP.

3.1.3 Програмний інтерфейс

* WebSocket-сервер на ws://localhost:8000 (порт можна змінити в mainSERVER.py):
* команда "load\_pcap": завантаження PCAP у пам’ять сервера;
* команда "save\_pcap": запис збережених пакетів у файл;
* команди "start\_sniff" / "stop\_sniff" (якщо додати) — керування live-захопленням
* REST API (за потреби розширити):
* GET /api/pcaps — список файлів;
* POST /api/classify — передача даних у syn\_detector.py для класифікації поточного потоку.

3.1.4 Комунікаційний протокол

* + Live-захоплення та управління через WebSocket (JSON-повідомлення);
  + CLI і API на базі HTTP(S) з JSON для обміну метаданими та результатами аналізу.

3.1.5 Обмеження пам’яті

* + одноразове завантаження PCAP до 2 ГБ (далі — ризик OOM);
  + зберігання проміжних CSV/Parquet у каталозі scripts/results до 30 днів

3.1.6 Операції

* + завантаження PCAP із диска чи live-захоплення через WebSocket;
  + генерація синтетичного трафіку (SYN-флуд) за допомогою genSYN.py;
  + Передобробка: фільтрація, агрегація, екстракція ознак у syn\_detector.py;
  + Анотація «нормальний»/«зловмисний» (ручна або по моделі);
  + Тренування та валідація моделей у скрипті syn\_detector.py;
  + Експорт готового датасету (dataset.csv) і моделі (rf\_model.pkl).

3.1.7 Функції продукту

* Live-моніторинг мережевого інтерфейсу та запис PCAP;
* генерація контролюваного трафіку для тестів;
* пакетна обробка існуючих PCAP у CLI;
* автоматична класифікація за Random Forest (модель у scripts/rf\_model.pkl);
* візуалізація базових метрик (кількість SYN, відповіді, хости) у Next.js UI.

3.1.8 Припущення і залежності

* + платформа: Python 3.8+ з пакетами scapy, websockets, pandas, scikit-learn;
  + Node.js 16+ для фронтенду (Next.js, Tailwind CSS);
  + доступ до NIC з правами на проміск’юїтний режим;
  + Windows/Linux з підтримкою Raw sockets.

3.2 Властивості програмного продукту

* + від live-захоплення до збереження PCAP/CSV;
  + екстракція ознак, фільтрація за BPF, агрегація потоків;
  + RF-модель із звітом Precision/Recall/F1;
  + однакові вхідні PCAP за тих самих параметрів дають однаковий вихід.

3.3 Атрибути програмного продукту

3.3.1 Надійність

* + фільтрація дублікатів (з seen\_packets у mainSERVER.py) запобігає повторному аналізу.

3.3.2 Доступність

* + UI без встановлення додаткового ПЗ (крім браузера);
  + CLI для автоматизації в скриптах.

3.3.3 Безпека

* + обробка лише незашифрованого трафіку (TLS вимкнено за замовчуванням);
  + режим sandbox: моделі класифікації працюють локально, без зовнішніх викликів.

3.3.4 Супроводжуваність

* + розбиття на модулі: mainSERVER.py (sniffer), main.py (CLI), scripts/ (аналіз, генерація);
  + Документація в README і коментарі до коду.

3.3.5 Переносимість

* + Docker-контейнер для швидкого розгортання (Dockerfile можна додати);
  + скрипти працюють на будь-якій ОС із Python 3.x.

3.3.6 Продуктивність

* + Live-захоплення до 10 000 пакетів/с без втрат на сучасному NIC;
  + пакетна обробка PCAP до 1 ГБ за < 3 хв у CLI.

3.4 Вимоги до зберігання даних

* + файли PCAP у каталозі scripts/pcaps/;
  + CSV-датасети у корені проекту (dataset.csv);
  + моделі RF у scripts/rf\_model.pkl;
  + логи sniffer-сервера у logs/.

3.5 Інші вимоги

* + інтерфейс і звіти англійською (за замовчуванням) з можливістю додати українську;
  + дотримання академічної доброчесності при роботі з публічними датасетами;
  + оформлення результатів і скриптів згідно з вимогами університету.