«Системні технології» 3 (164) 2026 «System technologies»
DOI 10.34185/1562-9945-3-164-2026-02
УДК 004.492.3

В.С. Горбатов, А.О. Журба

СТВОРЕННЯ ДАТАСЕТУ НА ОСНОВІ ПОДІЙ КЛАСИФІКОВАНИХ СИСТЕМОЮ ВИЯВЛЕННЯ МЕРЕЖЕВИХ ВТОРГНЕНЬ

Анотація. У статті представлено підхід до формування датасету для навчання моделей машинного навчання в контексті мережевої системи виявлення вторгнень Snort 3. На відміну від класичних датасетів, запропонований набір даних будується на основі нормалізованих байтових буферів інспектора та легкої телеметрії пакета, доступних під час онлайн-обробки трафіку. Ground truth задається контрольованим походженням трафіку (attack/benign PCAP), тоді як спрацювання правил Snort розглядаються як "teacher"-сигнал для подальшої побудови ризик-скорингу. Датасет сформовано для Fast Pattern-групи SIP/2.0 і містить десятки тисяч подій зі стандартизованим поділом на train/validation/test. Додатково виконано аналіз інформативності байтових позицій (на основі дивергенції Jensen–Shannon та ентропії) і кореляційний аналіз телеметрії, що підтверджує наявність локалізованого дискримінативного сигналу та відсутність тривіальних витоків через padding. Отриманий датасет може слугувати основою для нейромережевих моделей реального часу, які доповнюють сигнатурну детекцію оцінкою ризику.

Ключові слова: Snort 3, NIDS,, датасет, SIP/2.0, Fast Pattern, байтові дані, телеметрія пакета, ентропія, risk scoring, нейронні мережі реального часу, precision/recall.

Постановка задачі. Виявлення вторгнень є ключовим елементом захисту мереж, допомагаючи адміністраторам вчасно реагувати на шкідливі дії, такі як атаки та зловмисне ПЗ. Системи виявлення вторгнень (IDS) – обов'язковий рівень захисту критичних мереж від зростаючих загроз. З часом, атаки еволюціонують, в унісон з ними повинні еволюціонувати і методи їх виявлення. Проте дослідники часто стикаються з браком якісних і достовірних даних для тестування своїх напрацювань. Існуючі відкриті набори даних, містять різні типи атак і нормальний трафік, але мають обмеження, наприклад, неповне маркування або відсутність підтвердження правдивості. Тому створення датасетів на основі подій, маркованих як експертом так і системами виявлення вторгнень, може підвищити точність навчання та оцінки IDS, забезпечуючи більш реалістичні та різноманітні дані для розробок у сфері безпеки.

У роботі пропонується підхід до формування датасету, який використовує Snort 3 як класифікатор та джерело даних. **Мета** – отримати максимально наближений до реального процесу виявлення IDS набір даних, де джерелом teacher-міток є події Snort 3 (зокрема, спрацювання правил і Fast Pattern [1]), а ground truth визначається контрольо-

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

ваним походженням трафіку (attack/benign PCAP). Вхід моделі при цьому становлять сирі нормалізовані буфери, сформовані самим Snort (тобто те, що реально доступне системі під час аналізу даних на предмет атаки).

Поточний реліз зосереджений на прикладному трафіку SIP/2.0 як показовому прикладі Fast Pattern-групи, однак обраний формат (корисні дані + телеметрія) та джерело міток (події Snort) спроєктовано для перенесення на інші протоколи без зміни підходу. Варто зазначити, що робота не торкається шифрованих протоколів і постексплуатаційних стадій, а також, що специфіка ревізії правил Talos LightSPD задає «зріз» загроз у межах експерименту.

Варто зазначити що підхід є симуляційним, стартує з однієї Fast Pattern-групи і залежить від поточної ревізії правил; ці аспекти обговорюються в подальших розділах разом зі стратегіями пом'якшення (варіативність трафіку, розширення протокольного охоплення, оновлення правил).

У ході статті:

- формується IDS-центричний датасет, де події та мітки отримані безпосередньо від Snort 3, а вихідними даними є нормалізовані буфери, що використовує і формує IDS.

- пропонується протокол-агностичний байтовий формат для використання у сфері машинного навчання без ручного виділення характеристик.

Ця робота закладає основу для онлайн-інтеграції нейромережі у систему виявлення мережевих вторгнень. Використовуючи навчання за мітками «учителя» в реальному часі і узагальнення за межі правил, планується використання моделі як додаткового шару перевірки/ранжування спрацювань у реальних сценаріях.

Аналіз останніх досліджень і публікацій.

1. Підходи до створення IDS-датасетів

У галузі мережевої безпеки давно відзначається брак відкритих, якісних та актуальних датасетів для навчання і тестування систем виявлення вторгнень (IDS) [2]. Ефективність моделей машинного навчання для IDS значною мірою залежить від наявних наборів даних, однак отримати репрезентативний датасет непросто через відсутність стандартизованих методик його побудови, проблеми з документуванням процесу та неповні або неточні мітки класів [3]. Багато загальновживаних бенчмарк-датасетів, таких як KDD'99, нині вважаються застарілими: вони не відображають сучасного стану мережевого трафіку (наприклад, масового шифрування) та актуальних загроз [4]. В літературі описано кілька підходів до генерування даних для IDS, які можна класифікувати за джерелом трафіку на реальні, симульовані та гібридні [5]. Кожен із цих підходів має свої переваги і недоліки, які розглянемо нижче.

Реальний трафік. Перший підхід передбачає збір реальних мережевих трас з існуючих середовищ – наприклад, шляхом моніторингу виробничої мережі або використанням honeypot-систем для заманювання зловмисників. Перевагою є висока автентичність даних: реальний трафік містить правдоподібні фонові патерни та непередбачувані аномалії. Проте існують суттєві перешкоди: питання приватності користувачів і конфіденційності даних зазвичай унеможливлюють відкриту публікацію необроблених запи-

сів, вимагаючи анонімізації адрес і видалення корисного навантаження, що знижує цінність таких даних [6]. Крім того, пасивно зібраний трафік може не містити всіх необхідних видів атак: якщо під час захоплення не сталося певної атаки (наприклад, DDoS чи вторгнення), її не буде в датасет. Через ці труднощі чисто реальні датасети трапляються рідко: за оцінками, лише близько 17–20% публічних наборів містять виключно реальний нормальний трафік, і ця частка зменшується останніми роками [5]. Прикладом реального датасету є MAWILab, що містить довготривалі добові дампи трафіку з реальної магістральної мережі (15-хвилинні вибірки щодня) із мітками аномалій, отриманими шляхом кореляції кількох детекторів [7]. Інший підхід – розгортання honeypot-інфраструктур: так, датасет Kyoto 2006+ накопичувався протягом 2006–2015 рр. з мережі пасток і містить статистичні ознаки атак на ці пастки [8].

Симульований трафік. Другий підхід – емуляція мережевого трафіку у контрольованому середовищі (лабораторії). Більшість сучасних IDS-датасетів саме так і генеруються, враховуючи складність отримання реальних даних [5]. Дослідники розгортають тестову мережу (фізичну або віртуальну), в якій імітується легітимний фоновий трафік (робота користувачів, сервісів) та інсценуються атаки з відомими параметрами. Класичний приклад – набір DARPA’98/KDD’99, зібраний в рамках оцінювання IDS у 1998–1999 рр.: семитижневий трафік було згенеровано на випробувальному полігоні, де поряд із нормальним навантаженням було навмисно виконано 24 види атак, віднесених до чотирьох категорій (Denial of Service, Remote-to-Local, User-to-Root, Probe) [9]. Хоча KDD’99 став найбільш відомим бенчмарком, пізніші дослідження відзначили його недоліки – присутність багатьох дублікатів, нерівномірність класів та застарілість набору атак [10]. У відповідь було створено його похідні, зокрема NSL-KDD, де усунуто дублікатні записи та відфільтровано частину некоректних сесій, однак загалом ці набори все ще базуються на трафіку 90-х років.

Сучасні емульовані датасети прагнуть бути ближчими до реальності. Для генерування нормального трафіку часто застосовують профілі поведінки користувачів: наприклад, Shiravi та ін. запропонували систему B-Profile, яка моделює типові дії користувачів (веб-перегляд, перегляд пошти, передача файлів тощо) за допомогою автоматів і скриптів [11]. Канадський набір CICIDS-2017 є показовим прикладом: у тестовій мережі Університету Нью-Брансвіка було реалізовано реалістичний фон із емуляцією 25 видів активності користувачів (HTTP, HTTPS, FTP, SSH, e-mail тощо), після чого дослідники програли в мережі кілька атакуючих сценаріїв. Зокрема, CICIDS-2017 включає такі типи атак, як brute force підбір паролів, експлойт Heartbleed, ботнет-діяльність, DoS/DDoS, web-атаки (SQL-ін’єкції, XSS) та приховане проникнення в мережу. Атаки були згенеровані публічними інструментами та скриптами, наближеними до реальних зловмисних дій [12]. Подібним чином був побудований австралійський датасет UNSW-NB15: у кібер-лабораторії використовували комерційний інструмент для генерації сучасного нормального трафіку і актуальних атак, зберігши їх у вигляді tcpdump-трейсів загальною тривалістю 31 година; з цих трас потім було витягнуто 49 ознак трафіку, розбитих на групи (рівні потоку, пакету, сесії тощо) [13]. Основна перевага симуляції –

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

повний контроль над процесом та точна розмітка: кожна атака відома наперед, тому кожному запису можна призначити правильну мітку (нормальний чи шкідливий).

Проте слабким місцем є реалістичність згенерованого трафіку. Якщо емульований фон занадто спрощений, моделі IDS можуть навчитися виявляти не стільки реальні атаки, скільки артефакти штучного трафіку. Наприклад, використання простих генераторів (типа iperf) створює монотонний, однорідний трафік, який різко відрізняється від хаотичного реального фону – класифікатори легко відділяють такий "штучний нормальний" трафік від аномалій, і це призводить до завищених оцінок якості IDS [5]. Дійсно, занадто детерміновані сценарії можуть зробити задачу виявлення тривіальною (модель підлаштовується під нехарактерні шаблони симуляції), тому при емулюванні важливо досягти різноманітності та складності, близької до бойових мереж.

Гібридні та синтетичні підходи. Щоб збалансувати реалізм і контрольованість, дослідники все частіше звертаються до гібридних методів побудови датасетів. Один популярний підхід – ін'єкція синтетичних атак у реальний трафік: береться захоплений трафік реальної мережі (без міток), після чого в нього «вставляються» додаткові пакети/сесії, що відповідають відомим атакам [5]. Результатом є комбінований датасет: фон у ньому реалістичний (бо взятий із живої мережі), а атаки – синтетично згенеровані, але з відомим місцем і часом для розмітки. Прикладом такого підходу є набір UGR'16, зібраний з потоку NetFlow-записів іспанського провайдера: його калібраційна частина містить чисто реальний трафік, а тестова – той самий трафік, доповнений вставленими атаками (низькоінтенсивні DoS, сканування портів, ботнет) [14].

Окремо варто згадати про повністю синтетичне генерування трафіку за допомогою статистичних моделей або генеративного штучного інтелекту. Історично мережевий трафік намагалися синтезувати на основі простих статистичних розподілів, але новітні підходи використовують генеративно-змагальні мережі(GAN) та інші моделі глибокого навчання [5]. Наприклад, у роботі Ring та ін. (2019) запропоновано застосувати GAN для генерування мережевих потоків (NetFlow) з використанням техніки IP2Vec для вбудовування адрес у векторний простір [15]. Показано, що такі моделі можуть навчитися створювати синтетичний трафік, статистично схожий на реальний. Втім, повністю синтетичний трафік сьогодні вважається найменш реалістичним підходом – попри відсутність проблем приватності та високу гнучкість, він може не враховувати багатьох тонкощів живих мережевих взаємодій. Тому цілком штучні набори зазвичай використовуються як допоміжні або для вузьких завдань, тоді як у загальному випадку рекомендується комбінувати методи і перевіряти синтетичні дані на відповідність реальності [5].

2. Формат даних, розмітка та типи атак

При створенні IDS-датасету важливо задокументувати його характеристики: формат подання трафіку, схему маркування даних та перелік типів атак, що містяться в наборі. За форматом можна виділити два підходи до публікації даних. Перший – це надання сирих мережевих трас, зазвичай у вигляді PCAP-файлів (повні пакети) або flow-записів (наприклад, NetFlow/IPFIX логи). Такий формат максимально гнучкий, але вимагає від дослідника самостійно опрацьовувати дані (виділяти ознаки, агрегувати пото-

ки тощо). Другий підхід – публікація вже оброблених вибірок: автори датасету самі витягують з сирих трас статистичні ознаки і надають підготовлений набір у табличній формі (CSV-файли або навіть готові векторизовані масиви) [3]. Наприклад, у згаданому вище CICIDS-2017 після захоплення, трафік було оброблені зв допомогою CICFlowMeter [16], для виділення числових ознак на кожен мережевий потік (тривалість, кількість байтів, пакетів, середні статистики і т.д.). Кожен запис при цьому містив ідентифікатори сеансу (IP-адреси джерела й призначення, порти, протокол) та був класифікований [12]. Інший інструмент, часто використовуваний для аналізу трас, – це Zeek [17]: він дозволяє зберегти метадані сесій і одночасно застосувати скрипти для автоматичного визначення відомих атак або аномалій. Зокрема, при створенні нового датасету HIKARI-2021 дослідники взяли набір ознак CICIDS-2017 і за допомогою скриптів Zeek згенерували аналогічні показники для свого трафіку [3]. Вибір формату залежить від цілей: пакетні дані дають більше гнучкості (можна обчислити будь-які нові ознаки, перевірити власні методи виявлення), тоді як готові фічі спрощують порівняння алгоритмів (усі моделі тренуються на однаковому наборі полів).

Наявність ground truth (правдивих міток) є обов'язковою вимогою для IDS-датасету, але забезпечити її непросто. В емульованих наборах, як зазначалося, розмітка робиться вручну під час генерування: дослідник точно знає, коли і яка атака виконувалась, тому може позначити відповідні пакети чи потоки як "атака", а решту як "нормальний" трафік. У реальних же мережевих журналах абсолютно точна розмітка практично недосяжна: часто невідомо напевне, чи була певна аномалія справжньою атакою, чи легітимною аномальною подією. Через це частина публічних датасетів з реальним трафіком надає лише непрямі мітки. Наприклад, у наборі SimpleWeb (University of Twente) тривалий трафік кампусної мережі було зібрано без явних міток, а згодом відфільтровано підозрілі підмножини за допомогою honeypot-сервера; отримані мітки "підозрілий трафік" опиралися на спрацьовування пастки, але не гарантували повного охоплення всіх атак [18]. Отже, документування процесу побудови і маркування є критично важливим фактором для наукової цінності датасету.

Склад атак у датасеті визначає, для яких задач IDS він придатний. Історично у наборі DARPA'99/KDD'99 всі атаки поділялися на 4 великі категорії: DoS (відмова в обслуговуванні), Probe/Scan (розвідка, сканування мережі), R2L (атаки, що дозволяють віддаленому користувачу отримати локальні привілеї) та U2R (підвищення привілеїв локального користувача) [9]. Пізніші набори розширювали цей перелік: з'явилися класи для шкідливого ПЗ та ботнетів, веб-атак на прикладному рівні, атаки на протоколи VoIP/Multimedia, цільові APT-атаки, тощо. Зрозуміло, неможливо одним набором охопити всі загрози, але репрезентативний датасет намагається містити різнопланові сценарії – від простих DoS до багатостадійних вторгнень. Важливо, щоб у датасеті були представлені як мережеві атаки, так і нормальний трафік різних типів, причому змішані таким чином, щоб відтворювати реальні умови – наприклад, атаки можуть маскуватися під легітимний трафік, відбуватися паралельно з фоновою активністю користувачів і т.д. Якщо набір даних не містить нормальної складової (є і такі випадки, коли публіку-

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

ють лише вибірку зловмисного трафіку для аналізу шкідливості), то його доводиться поєднувати з окремим набором звичайного трафіку для повноцінного тестування IDS [5].

3. Інструменти генерації трафіку та атак

Створення реалістичного трафіку для IDS-датасетів потребує використання різноманітних інструментів, що підтримують генерування як фонових навантажень, так і атакуючих впливів. Для симуляції легітимного трафіку застосовуються трафік-генератори або емуляційні платформи. Простими засобами є утиліти на кшталт Iperf), Ostinato, NetFlow Generator, або спеціалізовані інструменти під конкретні протоколи. Наприклад, для VoIP-трафіку існує утиліта SIPp, що програє сценарії дзвінків SIP за заданим шаблоном [19]. Варто зазначити що такі інструменти можуть створювати нереалістично одноманітний потік (з однаковими інтервалами, розмірами пакетів і т.д.), якщо їх використовувати без належної варіації [5]. Максимального реалізму можна досягти, залучивши до генерування людей: в деяких експериментах дослідники запрошували групу реальних користувачів працювати за комп'ютерами (переглядати веб, надсилати запити, керувати IoT-пристроями) для збору справжнього трафіку з контрольованим профілем дій [20]. Хоча це трудомістко, такий підхід забезпечує найбільш правдоподібні фонові дані.

Для генерації атак зазвичай використовуються наявні інструменти з арсеналу пентестингу та кібербезпеки. Щоб симульовані атаки були максимально близькі до реальних, їх виконують ті ж засоби, якими користуються зловмисники. Поширеним вибором є: сканери портів і вразливостей (nmap, Nessus), експлойт-фреймворки (Metasploit), інструменти для DoS/DDoS (наприклад, Low Orbit Ion Cannon, hping3), засоби для brute-force зламування паролів (Hydra, Medusa), генератори шкідливого трафіку (Scapy для кастомних пакетів, SOC traffic generator тощо). Такий підхід гарантує правдиву послідовність пакетів при атаці (відповідну протоколу і методиці вторгнення). Важливо також варіювати параметри атак: змінювати адреси, порти, час доби, інтенсивність, типи пакетів тощо, аби в межах одного набору даних не повторювались ідентичні шаблони зловмисних дій. Різноплановість атак підвищує генералізаційну здатність IDS, яка працює з ними.

Окрім інструментів власне генерування трафіку, у процесі створення датасету застосовуються засоби для захоплення та обробки мережевих даних. Стандартом дефакто є утиліта tcpdump [21] або її аналоги (Dumpcap, TShark), які використовуються для запису мережевого трафіку в файл pcap. Якщо потрібно отримати зведені flow-записи (агреговані сесії), часто використовують спеціальні монітори – наприклад, системи збору NetFlow/IPFIX або мережеві проби. Після отримання сирих даних великого обсягу їх обробляють флоу-екстракторами на зразок згаданого CICFlowMeter, що автоматично рахує статистики по кожному з'єднанню (кількість пакетів, байтів, тривалість, середні швидкості тощо)[16]. Інший підхід – використання IDS-систем для попереднього аналізу: можна пропустити трафік через Snort/Suricata або Zeek в режимі запису логів, отримавши на виході як детектовані атаки (для перевірки коректності міток), так і зведення по сесіях.

Комбінація різних інструментів – від трафік-генераторів до аналізаторів – є ознакою добре спроектованого процесу побудови датасету, оскільки дозволяє забезпечити і різноманітність даних, і перевірку їх якості перед публікацією.

4. Обмеження існуючих датасетів та напрями покращення

Незважаючи на значну кількість доступних IDS-датасетів, у літературі наголошується на їх обмеженнях та недоліках [2]. Однією з головних проблем, як згадувалося вище, є старіння. Атаки і мережеві поведінкові патерни швидко еволюціонують, тому набори, зібрані кілька років тому, втрачають актуальність. Моделі, натреновані на таких наборах, ризикують показувати погані результати в реальному середовищі з новими загрозами. Друга проблема – обмежений контекст і різноманітність. Багато датасетів охоплюють лише короткий проміжок часу або одну мережеву ситуацію, що не відбиває повної варіабельності фону. Для надійності бажано, щоб дані містили різні часові періоди (робочі години і вечір, будні і вихідні), різні режими навантаження мережі, різні типи легітимної активності [5].

Ще одним недоліком є неповнота або неточність розмітки в деяких наборах. Як згадувалося, якщо мітки отримані автоматично чи неповні (наприклад, позначені лише категорії атак замість точного переліку всіх атакуючих сесій), то оцінка IDS може бути ненадійною. Наявні роботи вказують на випадки виявлення помилок у відомих датасетах: так, аналіз CICIDS-2017 та CSE-CIC-2018 показав наявність дублікатів, невідповідностей міток часу та інших артефактів, що вимагали ручного чищення даних [22].

Через зазначені проблеми спільнота дедалі більше звертає увагу на покращення методів генерації та валідації датасетів. В нових дослідженнях рекомендується приділяти особливу увагу реалізму: наскільки можливо, включати актуальний реальний трафік (навіть якщо його треба анонімізувати) та сучасні сценарії атак, аби відображати поточний ландшафт загроз [31]. Якщо реальні дані недоступні, слід удосконалювати емуляцію легітимного трафіку – це залишається найскладнішим завданням, оскільки інструменти для генерування атак вже досить наближені до реальних, а от згенерувати різноманітний і реалістичний фон все ще важко [23].

В цілому, огляди літератури сходяться на тому, що проблема датасетів для IDS далека від вирішення, але поступово напрацьовуються принципи їх якісного створення: комбінування реальних і симульованих методів, прозоре документування процесу, забезпечення максимальної реалістичності та повноти, а також вільне розповсюдження з урахуванням етичних і правових норм. Такий підхід сприятиме появі більш надійних і стійких систем виявлення вторгнень у майбутньому.

Викладення основного матеріалу дослідження.

1. Вимоги до датасету

Перед генерацією датасету потрібно визначити ключові фактори що будуть задавати параметри даних та підходу до їх створення. У цьому розділі розібрані та аргументовані рішення, прийняті у ході дослідження предметної області.

1.1. Підхід до виділення трафіку

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

Як згадувалося у вступі, ця робота є підготовкою до прямої інтеграції нейронної мережі у IDS і ціллю є модель що навчається відтворювати та узагальнювати рішення Snort за його ж подіями. Тож варто пояснити чому події класифіковано саме за FastPattern групами.

На етапі проектування було виділено 3 варіанти прив'язки даних, базуючись саме на подальшій цілі роботи:

- Per-rule (окрема маленька модель на кожне правило).

Плюси: точна локалізація; можна агресивно оптимізувати під конкретні шаблони.

Мінуси: вибух кількості моделей, складність супроводу, ризик «зашумлення» від нестабільності окремих сигнатур; слабке узагальнення між близькими правилами.

- Per-instance/Per-service (одна велика модель на весь інстанс Snort або протокол).

Плюси: специфіку швидких шляхів зіставлення.

- Per Fast Pattern (одна модель на групу правил, що ділять спільний шаблон).

Плюси: добрий баланс між кількістю моделей і узагальненням; природний тригер – спрацювання Fast Pattern після якого можна викликати NN лише для відповідної підмножини трафіку; зберігається локальність семантики.

Мінуси: потребує коректної ідентифікації FP-груп і стабільності їх складу між ревізіями правил.

З огляду на вимоги ефективності, швидкодії і бажання мінімізувати зміни у Snort, ця робота зосереджена на рівні Fast Pattern-групи: NN викликається після FP-match, але до повного обчислення складних умов правила, і працює на вже нормалізованих байтах, які на цьому етапі доступні інспекторам.

1.2. Загальні вимоги до IDS-датасетів

Як вже згадувалося раніше, створюваний датасет призначено для тренування та валідації моделей, націлених на вузькі класи мережевих атак. У цьому конкретному випадку ціллю є один класс атак, що поділяють спільний базовий патерн у текстовому буфері.

Таким чином, одиницею спостереження є саме подія виявлення патерну. Так, для не виявлених атак і для benign-трафіку, один запис відповідає факту виявлення підозри (тобто спільного патерну). Для виявлених атак: один запис відповідає факту виявлення атаки. Число результуючих записів хоч і не є дуже важливою мірою, на даному етапі, але запланована кількість не менше 50 000 записів загалом. Множинні спрацювання на один приклад не передбачені; у разі винятків пріоритизується атака або перше виявлення.

Що стосується записів, кожен запис повинен включати наступні дані:

- is_attack – ground truth(експертне) твердження чи є трафік атакою;
- alerted – чи виявлена атака IDS;
- buffers – набір текстових даних які IDS виділив для пошуку атак;
- buffer_names – імена буферів даних;

Додаткові прикладні ознаки сесії/пакета (розмір, напрям, часові мітки), за наявності.

Датасет повинен покривати більшість доступних атак у межах однієї Fast Pattern-групи (з потреби, не всіх правил, щоб зберегти простір для валідації), а також велику різноманітність benign-сценаріїв для оцінки хибнопозитивних спрацювань.

Результатом роботи є опубліковані PCAP файли, фінальні JSON-файли, Docker-оточення, а також зафіксовані версії Snort 3 і Talos LightSPD [24], для збереження відтворюваності.

1.3. Специфічні вимоги для використання зі Snort

Так як ключові дані будуть виділені напряму з рущія Snort 3, важливо виділити правила до формування датасету та оформлення ознак.

Ключові дані (payload байти) повині формуватися шляхом конкатенації нормалізованих буферів інспектора у фіксованому порядку (заданому самим інспектором). Довжина запису обмежена розміром 1024 байт із zero-padding для коротших випадків і crop для довших. Ідентифікуючі поля is_attack та alerted мають бути представлені у вигляді bool флагів.

Щодо нейтральної телеметрії, тут є обмеження зі сторони логерів Snort 3. З документації JSON логеру [25] був виділений наступний набір потенційно важливих ознак пакету/сесії: flowstart_time, seconds, proto, pkt_gen, pkt_len, eth_len, eth_type, ip_id, ip_len, tos, ttl, udp_len, dir, client_bytes, client_pkts, server_bytes, server_pkts. Зазначені поля повинні бути включені до кожного запису.

1.4. Приватність, безпека та ліцензування

Оскільки середовище повністю лабораторне, спеціальна анонімізація не вимагається. Використання PCAP слід супроводжувати застереженням про відповідальне застосування.

2. Методологія генерації

Постановка задачі робить датасет максимально близьким до умов онлайн-детекції, знімає залежність від конкретних парсерів і дає змогу перевіряти PU-сценарій (P = події, на які спрацював Snort; U = решта подій без спрацювань).

2.1. Вибір Fast Pattern-групи

Відповідно до методу структурної класифікації правил IDS[31], вибір Fast Pattern-групи здійснюється шляхом максимізації інтегрального критерію:

$$J(p) = w_1 \cdot \text{cov}erage(p) + w_2 \cdot selectivity(p) - w_3 \cdot overlap(p,\cdot) - w_4 \cdot churn(p) \quad (1)$$

де $\boldsymbol{w_i \geq 0, \sum w_i = 1}$.

Критерії відображають:

coverage – структурну репрезентативність групи;

selectivity – дискримінаційну здатність відносно benign-трафіку;

overlap – рівень конкуренції з іншими Fast Pattern;

churn – часову стабільність правил групи.

Покриття визначалося як:

$$coverage(p) = \frac{N_{rules}(p)}{N_{rules}^{total}} \quad (2)$$

де $N_{rules}(p)$ – кількість правил Talos LightSPD, що містять Fast Pattern p,

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

N_{rules}^{total} – максимальна кількість правил у вибраній підмножині (30).

Селективність оцінювалась на основі трьох незалежних PCAP-файлів з легітимним трафіком відповідних сервісів.

$$selectivity(p) = 1 - \frac{N_{matches}(p)}{N_{packets}} \quad (3)$$

де $N_{matches}(p)$ – кількість пакетів, що містять відповідний Fast Pattern,

$N_{packets}$ – кількість пакетів відповідного сервісу у PCAP.

Overlap визначався як середня кількість інших Fast Pattern, що спрацювали на той самий пакет:

$$overlap(p) = \frac{1}{M}\sum_{i=1}^{M} FP_{other}(packet_i) \quad (4)$$

де M – кількість спрацювань таргетного Fast Pattern,

FP_{other} – кількість спрацювань інших Fast Pattern з Talos LightSPD.

Churn оцінювався шляхом порівняння двох версій Talos LightSPD з інтервалом у 5 років:

$$churn(p) = \frac{N_{changed}(p)}{N_{rules}(p)} \quad (5)$$

де $N_{changed}(p)$ – кількість правил, що були змінені,

$N_{rules}(p)$ – загальна кількість правил групи.

Отримані значення були записані у таблицю 1. Важливо зазначити що результати оцінки сильно залежать від трафіку та набору правил, тож можуть відрізнятися.

Таблиця 1

Результати оцінки патернів

Pattern	Coverage	Selectivity	Overlap	Churn
SIP/2.0	1.00	0.74	0.44	0.667
NTLMSSP	0.43	0.86	0.77	0.667
X-MAILER	0.50	0.43	1.00	0.90

За відсутності пріоритетності критеріїв приймемо рівні ваги:

$$w_1 = w_2 = w_3 = w_4 = 1 \quad (6)$$

Тоді:

- SIP/2.0:

$$J = 1.00 + 0.74 - 0.44 - 0.667 = 0.633 \quad (7)$$

- NTLMSSP:

$$J = 0.43 + 0.86 - 0.77 - 0.667 = -0.147 \quad (8)$$

- X-MAILER:

$$J = 0.50 + 0.43 - 1.00 - 0.90 = -0.97 \quad (9)$$

Отримані значення демонструють що група з патерном SIP/2.0 має найбільше значення інтегральної функції якості, тож оберемо її для подальшого аналізу.

2.2. Топологія мережі для генерації датасету

Тож, мета дизайну – побудувати кероване, відтворюване середовище прикладного рівня (SIP/2.0) із достатньою варіативністю нормальної поведінки та систематично згенерованими «torture»-аномаліями, з єдиною точкою спостереження для подальшого журналювання IDS.

Компоненти та версії:

- PBX (SIP-сервер): Docker-образ andrius/asterisk[27], Asterisk v22.7.0.
- Клієнтські генератори SIP: Docker-образ sipp, SIPp v3.7.5 (для обох ролей – caller і callee).
- Генератор аномалій: SIPTorch[26] v0.1.0 з модифікаціями для підвищення варіабельності згенерованих атак (30 сценаріїв на основі SIP torture-кейсів[29], інші сценарії вимкнено[28]).
- Захоплення трафіку: tcpdump 4.99.1 (libpcap 1.10.1).

Логічна схема (рисунок 1):

1. Одна приватна Docker-мережа типу bridge; вузли синхронізують час із хостом.
2. PBX (центральний сервер);
3. Caller Farm (64 інстанси sipp);
4. Callee Farm (64 інстанси sipp);
5. Attackers (процеси SIPTorch, що запускаються з боку caller-вузлів).
6. Захоплення трафіку відбувалося на вхідному інтерфейсі сервера, між Caller Farm та сервером.

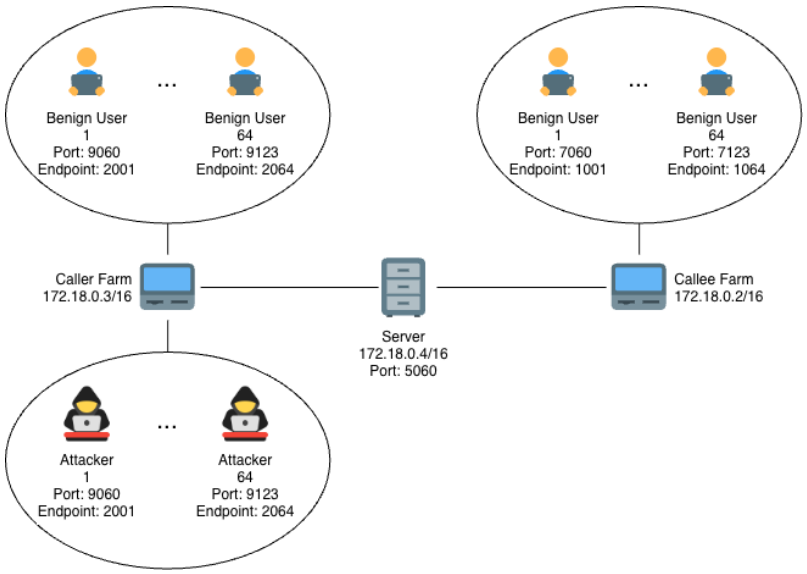

Рисунок 1 – Логічна схема мережі SIP

2.3. Безпечний профіль трафіку

За основу сценаріїв було взято існуючий набір sipp-scenarions [29], який надає велику кількість готових xml файлів з підготовленими SIP сесіями. Сесії було катекоризовано на клієнтські та серверні а також на позитивні та негативні. Також було значно модифіковано сценарії, для підвищення варіативності даних [30].

Ролі та сценарії.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

- Caller-farm (64 інстанси sipp). Кожен інстанс має унікальні облікові дані/endpoint/порт і випадково обирає один із 33 заздалегідь підготовлених сценаріїв: успішні/помилкові виклики, анонімні запити, із/без REGISTER/SUBSCRIBE, із/без медіа, з тайм-аутами.

- Callee-farm (64 інстанси sipp). Для відповіді використовується 23 варіанти профілів (нормальні та помилкові відповіді, з медіа та без, різні коди стану). Профіль обирається випадково незалежно від caller-сценарію.

На кожну спробу виклику caller випадково обирає callee; пари змінюються між ітераціями. Між сценаріями та між запусками одного сценарію варіюються ідентифікатори діалогів (Call-ID), теги, порядок заголовків, поля Contact/From/To, інтервали між повідомленнями, вказівники часу, тд. Це зроблено для зменшення штучної одноманітності даних.

2.4. Атакуючий профіль трафіку

Для генерації атак використано SIPTorch із 33 сценаріями (на основі RFC-кейсів). Сценарії включають: некоректні заголовки/типи вмісту, атипові схеми у Request-URI, відсутніх обов'язкових полів, надмірно великі скалярні значення, тощо.

Використано 64 атакувальні процеси, кожний з яких обирав випадкову ціль із 64 callee та випадковий «torture»-сценарій, генеруючи рандомні мутації полів. Для підвищення варіабельності трафіку було зроблено відповідні зміни у коді генераторів [28]. Це підвищує внутрішню різноманітність позитивних прикладів навіть у межах одного класу сценарію.

Така організація дає багату множину TP/FN залежно від конкретної реалізації сигнатур та їх ревізії, а також створює зони перетину з benign-поведінкою (коли повідомлення виглядають сигнатурно подібними, але є коректними), що критично для подальшого аналізу False Positive/False Negative.

2.5. Передобробка даних

Захоплений трафік було записано у форматі PCAP і оброблено використовуючи Snort 3 з DAQ PCAP. На виході з IDS сформовано подійні JSON-записи. Далі події було розподілено за джерелом PCAP на attack/benign і розбито на підвибірки (train/val/test) для подальшого аналізу. Нижче деталізовано ключові кроки.

2.5.1. Обробка Snort 3

Snort 3 запускався зі стандартним snort.lua, увімкненим alert_json та профілем правил Talos LightSPD rev. 2026-02-04-001. Для керованості подій застосовано:

- event_queue: не більш як 1 подія на пакет, і пріоритизація подій атак над суто інформаційними;
- модифікацію alert_json: додано вивід нормалізованих буферів SIP-інспектора (sip_header, sip_body) у поле buffers та їхніх назв у buffer_names. У якості роздільник було використано ASCII символ Record Separator (0x1E).

Для уніфікації байтового входу застосовано фіксовану довжину N = 1024 байти:

- якщо |buf| < 1024 – zero-padding (0×00) праворуч;
- якщо |buf| > 1024 – tail-crop до 1024 байтів.

Цілеспрямовано активовано підмножину PROTOCOL-VOIP SIP Torture, що входить до FP-групи SIP/2.0. У вивіреній конфігурації використовувались такі SID: 51494, 51499, 51501, 51502, 51503, 51504, 51506, 51507, 51508, 51509, 51510, 51511, 51512, 51515, 51744, 51745, 52087. Також було додано одне правило з ідентичним Fast Pattern, як інформаційне, для виявлення спрацювання FP групи.

2.5.2. Післяобробка івентів

Результуючі івенти були валідовані та оброблені за допомогою Python скрипта. За наявності дублю подій із тотожним буфером на одному й тому ж пакеті залишено першу (часово ранню) подію. Також було додано is_attack маркер до кожного івенту.

У телеметрії пакету, строкові дані були зіставлені з цілочисловими значеннями. Мапінги залишено у корені JSON документа, а саме:

- proto_mapping – для назви протоколів;
- pkt_gen_mapping – для назви модуля що подав пакет на обробку;
- eth_type_mapping – для Ethernet Type байту;
- dir_mapping – для напрямку пакету.

3. Результат

Сформовано подійний датасет, події розділено за джерелом PCAP на attack/benign і випадково розбито у співвідношенні 75/15/10 (train/val/test). Також, для атак

- Attack (усього): 26182 подій

Train: 19621 (75%) – Val: 3924 (15%) – Test: 2617 (10%)

- Benign (усього): 47212 подій

Train: 35633 (75%) – Val: 7163 (15%) – Test: 4416 (10%)

Підсумки детекції, загалом, до сплітів: TP = 4776, FP = 0, TN = 47212, FN = 21386. Дисбаланс у бік benign є очікуваним і відображає реальний контекст експлуатації IDS. Тож, можна бачити, що сигнатури дають високу точність, але пропускають значну частину варіативних torture-кейсів, тож саме сюди добре вписується ML-шар.

Власне набір даних у форматі JSON [32] та середовище для репродукції його запису [33] опубліковані на GitHub.

3.1. Описова статистика і кореляції

Основним вкладом цієї роботи є дані нормалізованих буферів мета цього аналізу - з’ясувати, у яких саме частинах вектора зосереджена інформація, що відрізняє атакуючий та легітимний трафік, а також перевірити, чи не виникає штучний сигнал через padding.

Так на рисунку 2 показано різницю Jensen–Shannon дивіргенції для байтових буферів, тобто наскільки розподіли байтів у кожній позиції відрізняються між класами. Найбільші відмінності зосереджені у верхній частині вектора, що відповідає початку SIP-повідомлення (заголовку та початку тіла). Далі інтенсивність різниці поступово зменшується, а в кінцевій частині буфера практично зникає. Це означає, що дискримінативний сигнал локалізований і пов’язаний зі структурованими елементами протоколу.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

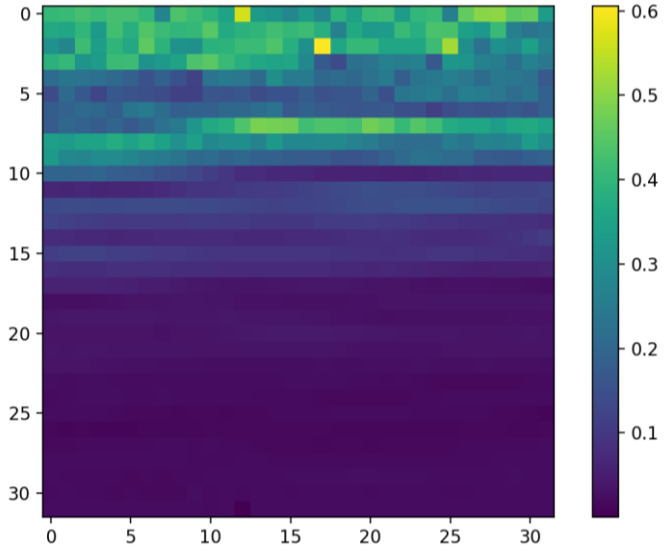

Рисунок 2 – Посимвольна різниця Jensen–Shannon дивіргенції

На рисунку 3 наведено різницю варіативності байтів між класами. Видно, що у верхній частині буфера атакуючі зразки демонструють більшу структурну різноманітність, тоді як нижня частина має низьку варіативність в обох класах. Це підтверджує, що інформативними є саме початкові позиції, а не весь 1024-байтний вектор рівномірно.

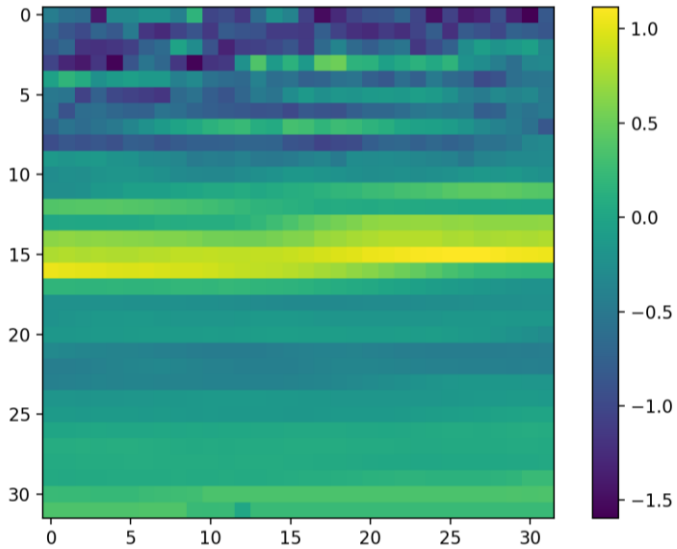

Рисунок 3 – Посимвольна різниця ентропії для буферів

Рисунок 4 показує аналіз частки padding для буферів. Можна побачити що кінцева частина вектора містить велику кількість доповнюючих байтів, однак різниця між класами у цій зоні не є суттєвою. Це важливо, оскільки свідчить про відсутність тривіального ефекту “витоку інформації” через довжину повідомлення: модель не може коректно розділити класи лише за рахунок різної кількості padding.

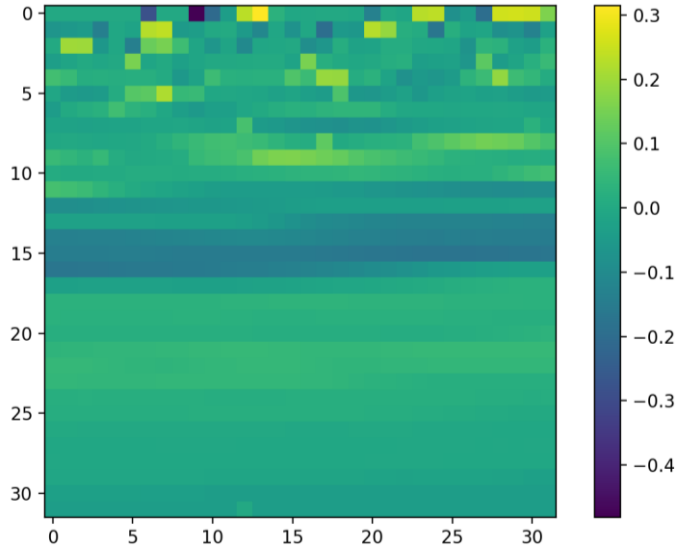

Рисунок 4 – Різниця частки padding байтів

Також було прораховано кореляції Пірсона між фічами з набору телеметрії. Як можна побачити на рисунку 5, найбільш виражені позитивні кореляції спостерігаються між розмірними характеристиками пакетів і байтовими лічильниками, що є очікуваним, оскільки ці параметри описують близькі аспекти однієї й тієї ж структури трафіку.

Водночас більшість службових або протокольних полів демонструють низький рівень кореляції з іншими змінними, що свідчить про їх відносну незалежність і потенційну інформативність у моделі. Відсутність великої кількості сильних кореляцій поза групами розмірних параметрів означає, що суттєвої мультиколінеарності в наборі ознак немає, а отже, фічі несуть комплементарну інформацію.

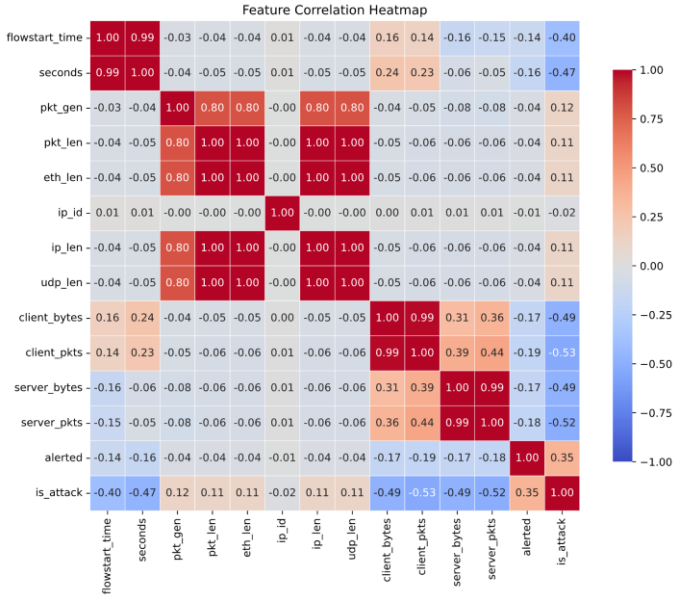

Рисунок 5 – Теплова карта кореляцій телеметрії

3.2. Порівняння з іншими датасетами

У таблиці 2 наведено загальне порівняння з іншими існуючими датасетами.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

Таблиця 2

Порівняння з існуючими датасетами

Параметри	KDD99	UNSW-NB15	CICIDS-2017	HIKARI-2021	Ця робота
Кількість унікальних IP адрес	11	45	16 960	7 991	≈129
Симулятивність	Так	Так	Частково	Частково	Так
Формат даних	dumpfile	pcap	pcap	pcap	pcap, JSON
Категорії атак	4	9	7	4	1
Методи виділення фічей	Bro-IDS	Argus, Bro	CICFlowMeter	Zeek, Python	Snort 3, Python
Кількість фічей	42	49	80	86	18 + 1024 байти

Висновки. У роботі запропоновано IDS-центричний підхід до формування датасету подій, у якому вхідними даними є нормалізовані байтові буфери інспектора Snort 3 та телеметрія пакета/сесії, доступні під час реальної онлайн-обробки. Ground truth задається контрольованим походженням трафіку (attack/benign PCAP), тоді як спрацювання правил Snort використовується як teacher-сигнал для сценарію “Snort-як-учитель”.

Сформовано датасет для Fast Pattern-групи SIP/2.0 обсягом 73 394 події (26 182 attack / 47 212 benign) зі сплітом 75/15/10 (train/val/test). Базова детекція Snort на цьому зрізі характеризується високою точністю (FP = 0) та обмеженою повнотою (Recall ≈ 0.183), що узгоджується з природою SIP-torture варіативності та обраною підмножиною сигнатур.

Аналіз байтових позицій (Jensen–Shannon дивіргенція, ентропія) показав, що дискримінативний сигнал локалізується переважно на початку повідомлення (заголовок/початок тіла), а padding-зона не створює тривіального “витоку” інформації між класами. Кореляційний аналіз телеметрії підтвердив очікувані кластери “довжин/обсягів” та загалом низьку мультиколінеарність інших полів.

Обмеження роботи: лабораторність середовища, фокус на одній FP-групі та конкретній ревізії правил, фіксована довжина байтового представлення (crop/pad), а також шумність teacher-міток, властива сигнатурному підходу. Подальші напрями включають розширення на інші FP-групи/протоколи, перевірку узагальнюваності на нових ревізіях правил і введення сценарних сплітів (scenario-holdout) для більш строгого оцінювання. Отриманий набір даних є основою для навчання моделей risk scoring/ранжування в режимі, сумісному з онлайн-детекцією IDS.

ЛІТЕРАТУРА

1. Горбатов В. С. Метод попередньої фільтрації сигнатур для прискорення пошуку атак системою виявлення мережевих вторгнень. Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті : Міжнар. науково-практ. конф., м. Дніпро, 13–14 груд. 2023 р. Дніпро, 2023. С. 136–137.

2. Kenyon A., Deka L., Elizondo D. Are public intrusion datasets fit for purpose

characterising the state of the art in intrusion event datasets. Computers & security. 2020. Vol. 99. P. 102022. URL: https://doi.org/10.1016/j.cose.2020.102022 (date of access: 21.02.2026).
3. Generating network intrusion detection dataset based on real and encrypted synthetic attack traffic / A. Ferriyan et al. Applied sciences. 2021. Vol. 11, no. 17. P. 7868. URL: https://doi.org/10.3390/app11177868 (date of access: 21.02.2026).
4. Survey of intrusion detection systems: techniques, datasets and challenges / A. Khraisat et al. Cybersecurity. 2019. Vol. 2, no. 1. URL: https://doi.org/10.1186/s42400-019-0038-7 (date of access: 21.02.2026).
5. Goldschmidt P., Chudá D. Network intrusion datasets: a survey, limitations, and recommendations. Computers & security. 2025. P. 104510.
URL: https://doi.org/10.1016/j.cose.2025.104510 (date of access: 21.02.2026).
6. Himabindu C. The challenges of effectively anonymizing network data. International journal of pharmacology and pharmaceutical technology. 2013. P. 15–22.
URL: https://doi.org/10.47893/ijppt.2013.1003 (date of access: 21.02.2026).
7. Kim J., Sim C., Choi J. Generating labeled flow data from MAWILab traces for network intrusion detection. Proceedings of the ACM workshop on systems and network telemetry and analytics. Phoenix, AZ, USA, 2019. P. 45–48.
8. Statistical analysis of honeypot data and building of Kyoto 2006+ dataset for NIDS evaluation / J. Song et al. The first workshop, Salzburg, Austria, 10 April 2011. New York, New York, USA, 2011. URL: https://doi.org/10.1145/1978672.1978676 (date of access: 21.02.2026).
9. Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation / R. P. Lippmann et al. DARPA information survivability conference and exposition. DISCEX'00, Hilton Head, SC, USA.
URL: https://doi.org/10.1109/discex.2000.821506 (date of access: 21.02.2026).
10. A detailed analysis of the KDD CUP 99 data set / M. Tavallaee et al. 2009 IEEE symposium on computational intelligence for security and defense applications (CISDA), Ottawa, ON, Canada, 8–10 July 2009. 2009.
URL: https://doi.org/10.1109/cisda.2009.5356528 (date of access: 21.02.2026).
11. Toward developing a systematic approach to generate benchmark datasets for intrusion detection / A. Shiravi et al. Computers & security. 2012. Vol. 31, no. 3. P. 357–374. URL: https://doi.org/10.1016/j.cose.2011.12.012 (date of access: 21.02.2026).
12. Sharafaldin I., Habibi Lashkari A., Ghorbani A. A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. 4th international conference on information systems security and privacy, Funchal, Madeira, Portugal, 22–24 January 2018. 2018. URL: https://doi.org/10.5220/0006639801080116 (date of access: 21.02.2026).
13. Moustafa N., Slay J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). 2015 military communications and information systems conference (milcis), Canberra, Australia, 10–12 November 2015. 2015. URL: https://doi.org/10.1109/milcis.2015.7348942 (date of access: 21.02.2026).
14. UGR‘16: A new dataset for the evaluation of cyclostationarity-based network IDSs / G. Maciá-Fernández et al. Computers & security. 2018. Vol. 73. P. 411–424. URL: https://doi.org/10.1016/j.cose.2017.11.004 (date of access: 21.02.2026).
15. Flow-based network traffic generation using Generative Adversarial Networks / M. Ring

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

et al. Computers & security. 2019. Vol. 82. P. 156–172.
URL: https://doi.org/10.1016/j.cose.2018.12.012 (date of access: 21.02.2026).
16. GitHub - ahlashkari/cicflowmeter: cicflowmeter-v4.0 (formerly known as iscxflowmeter) is an ethernet traffic bi-flow generator and analyzer for anomaly detection that has been used in many cybersecurity datsets such as android adware-general malware dataset (CIC-AAGM2017), IPS/IDS dataset (CICIDS2017), android malware dataset (cicandmal2017) and distributed denial of service (cicddos2019). GitHub.
URL: https://github.com/ahlashkari/CICFlowMeter (date of access: 21.02.2026).
17. The zeek network security monitor. Zeek. URL: https://zeek.org (date of access: 21.02.2026).
18. Simpleweb/University of twente traffic traces data repository / R.R.R. Barbosa et al. CTIT technical report series. 2010.
URL: https://api.semanticscholar.org/CorpusID:13251179.
19. Welcome to SIPp. Welcome to SIPp. URL: https://sipp.sourceforge.net/ (date of access: 21.02.2026).
20. CUPID: A labeled dataset with Pentesting for evaluation of network intrusion detection / H. Lawrence et al. Journal of systems architecture. 2022. P. 102621.
URL: https://doi.org/10.1016/j.sysarc.2022.102621 (date of access: 21.02.2026).
21. Home | TCPDUMP & LIBPCAP. Home | TCPDUMP & LIBPCAP.
URL: https://www.tcpdump.org/ (date of access: 21.02.2026).
22. Engelen G., Rimmer V., Joosen W. Troubleshooting an intrusion detection dataset: the CICIDS2017 case study. 2021 IEEE security and privacy workshops (SPW), San Francisco, CA, USA, 27 May 2021. 2021. URL: https://doi.org/10.1109/spw53761.2021.00009 (date of access: 21.02.2026).
23. Layeghy S., Gallagher M., Portmann M. Benchmarking the benchmark – Comparing synthetic and real-world Network IDS datasets. Journal of information security and applications. 2024. Vol. 80. P. 103689. URL: https://doi.org/10.1016/j.jisa.2023.103689 (date of access: 21.02.2026).
24. Soft Release: lightSPD, the new rules package for Snort 3. Snort Blog.
URL: https://blog.snort.org/2020/12/soft-release-lightspd-new-rules-package.html (date of access: 21.02.2026).
25. Snort 3 reference manual. GitHub.
URL: https://github.com/snort3/snort3/releases/download/3.10.2.0/snort_reference.html (date of access: 21.02.2026).
26. GitHub - 0xinfection/siptorch: A "SIP torture" (RFC 4475) testing framework. GitHub.
URL: https://github.com/0xInfection/SIPTorch (date of access: 21.02.2026).
27. GitHub - andrius/asterisk: asterisk PBX in docker – smallest asterisk ever!. GitHub.
URL: https://github.com/andrius/asterisk (date of access: 21.02.2026).
28. Comparing master...shuffle · VytalyGorbatov/SIPTorch. GitHub. URL: htt- htt-
ps://github.com/VytalyGorbatov/SIPTorch/compare/master...VytalyGorbatov:SIPTorch:shuffle (date of access: 23.02.2026).
29. saghul/sipp-scenarios: SIPp scenarios I use for testing SIP stuff. GitHub. URL: https://github.com/saghul/sipp-scenarios (date of access: 23.02.2026).
30. Comparing master...mutability · VytalyGorbatov/sipp-scenarios. GitHub. URL: https://github.com/VytalyGorbatov/sipp-scenarios/compare/master...VytalyGorbatov:sipp-

scenarios:mutability (date of access: 23.02.2026).
31. Горбатов В.С., Журба А.О. Метод класифікації мережевих вторгнень на основі структури правил систем виявлення вторгнень // Інформаційні технології і автоматизація – 2025 / Матеріали XVIII міжнародної науково-практичної конферен-ції. Одеса, 30-31 жовтня 2025 р. - Одеса, Видавництво ОНТУ, 2025 р. – С. 261-263.
32. GitHub - VytalyGorbatov/sip-dataset. GitHub.
URL: https://github.com/VytalyGorbatov/sip-dataset (date of access: 23.02.2026).
33. GitHub - VytalyGorbatov/sip-lab. GitHub. URL: https://github.com/VytalyGorbatov/sip-lab (date of access: 23.02.2026).

REFERENCES

1. Horbatov, V. S. (2023). Signature pre-filtering method for accelerating attack search by network intrusion detection systems. Modern Information and Communication Technologies in Transport, Industry, and Education: International Scientific and Practical Conference, Dnipro, December 13–14, 2023. Dnipro, pp. 136–137.
2. Kenyon A., Deka L., Elizondo D. Are public intrusion datasets fit for purpose characterising the state of the art in intrusion event datasets. Computers & security. 2020. Vol. 99. P. 102022. URL: https://doi.org/10.1016/j.cose.2020.102022 (date of access: 21.02.2026).
3. Generating network intrusion detection dataset based on real and encrypted synthetic attack traffic / A. Ferriyan et al. Applied sciences. 2021. Vol. 11, no. 17. P. 7868. URL: https://doi.org/10.3390/app11177868 (date of access: 21.02.2026).
4. Survey of intrusion detection systems: techniques, datasets and challenges / A. Khraisat et al. Cybersecurity. 2019. Vol. 2, no. 1. URL: https://doi.org/10.1186/s42400-019-0038-7 (date of access: 21.02.2026).
5. Goldschmidt P., Chudá D. Network intrusion datasets: a survey, limitations, and recommendations. Computers & security. 2025. P. 104510
URL: https://doi.org/10.1016/j.cose.2025.104510 (date of access: 21.02.2026).
6. Himabindu C. The challenges of effectively anonymizing network data. International journal of pharmacology and pharmaceutical technology. 2013. P. 15–22.
URL: https://doi.org/10.47893/ijppt.2013.1003 (date of access: 21.02.2026).
7. Kim J., Sim C., Choi J. Generating labeled flow data from MAWILab traces for network intrusion detection. Proceedings of the ACM workshop on systems and network telemetry and analytics. Phoenix, AZ, USA, 2019. P. 45–48.
8. Statistical analysis of honeypot data and building of Kyoto 2006+ dataset for NIDS evaluation / J. Song et al. The first workshop, Salzburg, Austria, 10 April 2011. New York, New York, USA, 2011. URL: https://doi.org/10.1145/1978672.1978676 (date of access: 21.02.2026).
9. Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation / R. P. Lippmann et al. DARPA information survivability conference and exposition. DISCEX'00, Hilton Head, SC, USA. URL: https://doi.org/10.1109/discex.2000.821506 (date of access: 21.02.2026).
10. A detailed analysis of the KDD CUP 99 data set / M. Tavallaee et al. 2009 IEEE symposium on computational intelligence for security and defense applications (CISDA), Ottawa, ON, Canada, 8–10 July 2009. 2009. URL: https://doi.org/10.1109/cisda.2009.5356528 (date of access: 21.02.2026).
11. Toward developing a systematic approach to generate benchmark datasets for intrusion

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

detection / A. Shiravi et al. Computers & security. 2012. Vol. 31, no. 3. P. 357–374. URL: https://doi.org/10.1016/j.cose.2011.12.012 (date of access: 21.02.2026).
12. Sharafaldin I., Habibi Lashkari A., Ghorbani A. A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. 4th international conference on information systems security and privacy, Funchal, Madeira, Portugal, 22–24 January 2018. 2018. URL: https://doi.org/10.5220/0006639801080116 (date of access: 21.02.2026).
13. Moustafa N., Slay J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). 2015 military communications and information systems conference (milcis), Canberra, Australia, 10–12 November 2015. 2015. URL: https://doi.org/10.1109/milcis.2015.7348942 (date of access: 21.02.2026).
14. UGR‘16: A new dataset for the evaluation of cyclostationarity-based network IDSs / G. Maciá-Fernández et al. Computers & security. 2018. Vol. 73. P. 411–424. URL: https://doi.org/10.1016/j.cose.2017.11.004 (date of access: 21.02.2026).
15. Flow-based network traffic generation using Generative Adversarial Networks / M. Ring et al. Computers & security. 2019. Vol. 82. P. 156–172.
URL: https://doi.org/10.1016/j.cose.2018.12.012 (date of access: 21.02.2026).
16. GitHub - ahlashkari/cicflowmeter: cicflowmeter-v4.0 (formerly known as iscxflowmeter) is an ethernet traffic bi-flow generator and analyzer for anomaly detection that has been used in many cybersecurity datsets such as android adware-general malware dataset (CIC-AAGM2017), IPS/IDS dataset (CICIDS2017), android malware dataset (cicandmal2017) and distributed denial of service (cicddos2019). GitHub.
URL: https://github.com/ahlashkari/CICFlowMeter (date of access: 21.02.2026).
17. The zeek network security monitor. Zeek. URL: https://zeek.org (date of access: 21.02.2026).
18. Simpleweb/University of twente traffic traces data repository / R. R. R. Barbosa et al. CTIT technical report series. 2010.
URL: https://api.semanticscholar.org/CorpusID:13251179.
19. Welcome to SIPp. Welcome to SIPp. URL: https://sipp.sourceforge.net/ (date of access: 21.02.2026).
20. CUPID: A labeled dataset with Pentesting for evaluation of network intrusion detection / H. Lawrence et al. Journal of systems architecture. 2022. P. 102621.
URL: https://doi.org/10.1016/j.sysarc.2022.102621 (date of access: 21.02.2026).
21. Home | TCPDUMP & LIBPCAP. Home | TCPDUMP & LIBPCAP.
URL: https://www.tcpdump.org/ (date of access: 21.02.2026).
22. Engelen G., Rimmer V., Joosen W. Troubleshooting an intrusion detection dataset: the CICIDS2017 case study. 2021 IEEE security and privacy workshops (SPW), San Francisco, CA, USA, 27 May 2021. 2021. URL: https://doi.org/10.1109/spw53761.2021.00009 (date of access: 21.02.2026).
23. Layeghy S., Gallagher M., Portmann M. Benchmarking the benchmark – Comparing synthetic and real-world Network IDS datasets. Journal of information security and applications. 2024. Vol. 80. P. 103689. URL: https://doi.org/10.1016/j.jisa.2023.103689 (date of access: 21.02.2026).
24. Soft Release: lightSPD, the new rules package for Snort 3. Snort Blog. URL: https://blog.snort.org/2020/12/soft-release-lightspd-new-rules-package.html (date of access: 21.02.2026).
25. Snort 3 reference manual. GitHub.

URL: https://github.com/snort3/snort3/releases/download/3.10.2.0/snort_reference.html (date of access: 21.02.2026).
26. GitHub - 0xinfection/siptorch: A "SIP torture" (RFC 4475) testing framework. GitHub. URL: https://github.com/0xInfection/SIPTorch (date of access: 21.02.2026).
27. GitHub - andrius/asterisk: asterisk PBX in docker – smallest asterisk ever!. GitHub. URL: https://github.com/andrius/asterisk (date of access: 21.02.2026).
28. Comparing master...shuffle VytalyGorbatov/SIPTorch. GitHub. URL: https://github.com/VytalyGorbatov/SIPTorch/compare/master...VytalyGorbatov:SIPTorch:shuffle (date of access: 23.02.2026).
29. saghul/sipp-scenarios: SIPp scenarios I use for testing SIP stuff. GitHub. URL: https://github.com/saghul/sipp-scenarios (date of access: 23.02.2026).
30. Comparing master...mutability VytalyGorbatov/sipp-scenarios. GitHub. URL: https://github.com/VytalyGorbatov/sipp-scenarios/compare/master...VytalyGorbatov:sipp-scenarios:mutability (date of access: 23.02.2026).
31. Horbatov, V. S., Zhurba, A. O. (2025). Network intrusion classification method based on the rule structure of intrusion detection systems. Information Technologies and Automation – 2025: Proceedings of the XVIII International Scientific and Practical Conference, Odesa, October 30–31, 2025. Odesa: ONT University Publishing House, pp. 261–263.
32. GitHub - VytalyGorbatov/sip-dataset. GitHub.
URL: https://github.com/VytalyGorbatov/sip-dataset (date of access: 23.02.2026).
33. GitHub - VytalyGorbatov/sip-lab. GitHub. URL: https://github.com/VytalyGorbatov/sip-lab (date of access: 23.02.2026).

Received 30.03.2026.
Accepted 01.04.2026.
Published 30.04.2026

Towards creating a dataset based on events classified by a network intrusion detection system

This paper presents an approach to building a dataset for training machine-learning models in the context of the Snort 3 network intrusion detection system. Unlike conventional NIDS datasets, the proposed dataset is constructed from normalized inspector byte buffers and lightweight packet telemetry that are available during real-time traffic processing. Ground truth is defined by the controlled origin of traffic (attack/benign PCAP), while Snort rule triggers are treated as a "teacher" signal to support subsequent risk-scoring models. The dataset is generated for the SIP/2.0 Fast Pattern group and contains tens of thousands of events with a standardized train/validation/test split. In addition, we analyze byte-position informativeness using Jensen–Shannon divergence and entropy, and perform correlation analysis of telemetry features. The results indicate that the discriminative signal is largely localized in the early parts of the message (header and initial payload) and that padding does not introduce trivial information leakage between classes. The resulting dataset can serve as a foundation for real-time neural models that complement signature-based detection with probabilistic risk assessment.

Keywords: Snort 3, NIDS, dataset, SIP/2.0, Fast Pattern, byte-level features, packet telemetry, entropy, risk scoring, real-time neural networks, precision/recall.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

Горбатов Віталій Сергійович - аспірант кафедри Інформаційних технологій і систем ННІ ДМетІ УДУНТ.
ORCID: https://orcid.org/0009-0000-9061-8207
Журба Анна Олексіївна – к.т.н., доцент, доцент кафедри Інформаційних технологій і систем ННІ ДМетІ УДУНТ.
ORCID: https://orcid.org/0000-0002-4367-385X

Gorbatov Vitalii - post-graduate student, Department of information technology and systems, NNI DMetI, Ukrainian state university of science and technologies.
ORCID: https://orcid.org/0009-0000-9061-8207
Zhurba Anna - assistant professor, Department of information technology and systems NNI DMetI, Ukrainian state university of science and technologies.
ORCID: https://orcid.org/0000-0002-4367-385X