Хмельницький національний університет
Факультет інформаційних технологій
Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему

Метод аналізу трафіку з метою виявлення атак на комплексні системи захисту інформації

Галузь знань 12 – Інформаційні технології

Спеціальність 125 – Кібербезпека

КРМКБ.180129.22.01.15 ПЗ

Виконав: студент 2 курсу, група КБм-22-1 _____ Левандовський А.О
Підпис

Керівник доц.,к.т.н, доцент _____ Муляр І.В.
Підпис

Нормоконтролер старший викладач _____ Мостовий С.В.
Підпис

До захисту допускаю:
Зав. кафедри кібербезпеки, к.т.н., доц _____ Кльоц Ю.П.
Підпис

19 грудня 2023 р.

Хмельницький, 2023

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КІБЕРБЕЗПЕКИ

Освітній рівень МАГІСТР

Галузь знань 12 ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Спеціальність 125 КІБЕРБЕЗПЕКА

Освітня програма КІБЕРБЕЗПЕКА

ЗАТВЕРДЖУЮ

Зав. кафедри Ю.П. Кльоц

"30" 08 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Левандовському Андрію Олександровичу
Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Метод аналізу трафіку з метою виявлення атак на комплексні системи захисту інформації

Керівник роботи Муляр Ігор Володимирович
Прізвище, ім'я, по батькові, науковий ступінь, вчене звання
кандидат технічних наук, доцент

Затверджена наказом № 30 ректора університету, додаток №25 від 15.08.2023

2. Строк подання студентом проєкту (роботи) на кафедру 15.11.2023

3. Вихідні дані до проекту (роботи) Підвищення ефективності технології аналізу трафіку мережі з метою виявлення атак.

4 Зміст пояснювальної записки (перелік питань, які потрібно розробити) Вступ. Аналіз бази предметної області, й наукових досягнень. Опис існуючих методів аналізу трафіку. Метод аналізу трафіку з метою виявлення атак на КСЗІ. Реалізація розробленого методу. Висновки.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали і посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Мостовий С.В. Старший викладач кафедри кібербезпеки		

7. Дата видачі завдання «01» вересня 2023р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Вибір напряму дослідження і узгодження тематики КРМ з керівником	01.06.2023	
2	Ознайомлення з предметною областю; формулювання мети і задач дослідження; визначення об'єкта і предмета дослідження	04.09.2023	
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	18.09.2023	
4	Робота над розділом 2 – розробка моделей і методів для вирішення поставленої задачі	02.10.2023	
5	Робота над розділом 3 – розробка алгоритмів і технологій, їх аналіз	16.10.2023	
6	Робота над розділом 4 – апробація запропонованих рішень	06.11.2023	
7	Робота над науковою публікацією	10.11.2023	
8	Узгодження отриманих результатів, оформлення пояснювальної записки згідно вимог	15.11.2023	
9	Попередній захист роботи	17.11.2023	
10	Захист роботи на засіданні ЕК	06.12.2023	

Студент _____ А.О Левандовський
Підпис Ініціали, прізвище

Керівник проекту (роботи) _____ І.В.Муляр
Підпис Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Метод аналізу трафіку з метою виявлення атак на комплексні системи захисту інформації»

Автор роботи: Левандовський Андрій Олександрович

Керівник роботи: к.т.н., доц. : Муляр Ігор Володимирович

Загальний обсяг роботи: 73 сторінок, 1 таблиця, 31 рисунків, 57 посилань.

Ключові слова: захист інформації, аналіз трафіку , ксзі.

Метою кваліфікаційної роботи є розробка методу аналізу трафіку з метою виявлення атак на комплексні системи захисту інформації. Виявлення аномалій мережевого трафіку, Розкриття ефективності обраного методу. Створення надійної та ефективної системи захисту.

В результаті виконання кваліфікаційної роботи був підвищений рівень захисту інформації та ймовірність виявлення потенційних загроз для комп'ютерних систем і мереж.

Дата: 19.12.2023

ANNOTATION

Theme of qualification work: Method of traffic analysis for the purpose of detecting attacks on complex information protection systems

Author of the work: Andriy Oleksandrovich Levandovskyi

Mentor: Ph.D. Ihor Volodymyrovych Mulyar

Total volume of work:73 pages,1 tables, 31 figures, 1tables, , 57 links.

Keywords: information protection, traffic analysis, cips.

The purpose of the qualification work is to develop a method of traffic analysis in order to detect attacks on complex information protection systems. Detection of network traffic anomalies, disclosure of the effectiveness of the chosen method. Creating a reliable and effective protection system.

As a result of the qualification work, the level of information protection and the probability of detecting potential threats to computer systems and networks were increased.

19.12.2023

ЗМІСТ

ВСТУП

Зі зростанням кількості та витонченості кібератак захист інформації набуває все більшого значення. Для виявлення атак, які можуть бути невидимими для традиційних систем безпеки, важливим інструментом стає аналіз трафіку, тому розробка нових методів аналізу трафіку є актуальним завданням для забезпечення безпеки інформаційних ресурсів.

Сучасний стан інформаційних технологій підкреслює важливість захисту конфіденційних даних та забезпечення безпеки комплексних систем захисту інформації (КСЗІ).

В умовах постійно зростаючого обсягу цифрових даних та залежності сучасного суспільства від інформаційно-комунікаційних технологій, кібербезпека та забезпечення конфіденційності, цілісності та доступності інформації набувають все більшого значення.

Однак, оскільки технології стають все більш складними, кількість і витонченість атак на ці системи також зростає. Своєчасне виявлення та протидія таким атакам набуває все більшого значення для забезпечення безпеки критично важливих інформаційних ресурсів.

Метою даної дипломної роботи є розробка та реалізація методу аналізу трафіку для виявлення атак на КСЗІ. Дослідження фокусується на аналізі мережевого трафіку для виявлення аномалій та характеристик, які вказують на можливі загрози інформаційній безпеці. Метод спрямований на покращення реагування на атаки шляхом надання ефективного механізму виявлення та реагування на потенційні загрози.

Для досягнення цієї мети поставлені наступні завдання:

– провести аналіз існуючих методів виявлення атак на КСЗІ;

– розробити метод аналізу трафіку;

– порівняти ефективність розробленого методу з існуючими підходами.

Об'єктом дослідження є процедура впровадження методу аналізу трафіку з метою виявлення атак на ксзі.

Предметом дослідження є методи і алгоритми аналізу трафіку в мережі, а саме використовуючи нейроні мережі.

Наукова новизна полягає в:

– адаптувані й вдосконалені методів аналізу трафіку та алгоритму на основі штучного інтелекту;

– створені нейроної мережі що навчається на logs Web серверу nginx, та в подальшому може фільтрувати користувачів.

Сучасний динамічний інформаційний ландшафт вимагає високоефективних систем захисту інформації.

Дослідження зосереджене на методах аналізу трафіку, що використовуються для виявлення потенційних атак на складні системи інформаційної безпеки.

Збільшення кількості та складності кіберзагроз підкреслює необхідність розробки нових та вдосконалених методів виявлення аномалій в мережах

Магістерська робота буде складатися з розділів, що охоплюють теоретичні аспекти аналізу трафіку, огляд існуючих методів виявлення атак, опис розробленого методу, експериментальні результати та порівняльний аналіз з іншими підходами.

Практична цінність:

– дослідження має на меті підвищити рівень безпеки КСЗІ та зробити внесок у розробку загальних стратегій кіберзахисту в кіберсередовищі, що постійно розвивається.

– дослідження спрямоване на підвищення здатності виявляти атаки на КСЗІ шляхом ефективного аналізу мережевого трафіку. Це дозволить підвищити рівень безпеки інформаційних систем..

1 АНАЛІЗ БАЗИ ПРЕДМЕТНОЇ ОБЛАСТІ Й НАУКОВИХ ДСЯГНЕНЬ

1.1 Історія розвитку методів аналізу трафіку

Історія розвитку методів аналізу трафіку тісно пов'язана з появою комп'ютерних мереж.

На початку 1960-х років з'явилися перші комп'ютерні мережі, в тому числі ARPANET, яка була попередницею Інтернету. Основним завданням на той час було встановлення зв'язку між вузлами мережі, а аналіз трафіку був обмеженим, оскільки мережі використовувалися переважно для передачі невеликих обсягів даних.

Однак з поширенням мережевих протоколів, таких як TCP/IP у 1970-х і 1980-х роках, попит на інструменти для моніторингу та аналізу мережевого трафіку зріс. Щоб отримати повне розуміння мережевої передачі даних, стало необхідним розробити програмне забезпечення, здатне записувати та аналізувати дані на рівні пакетів [1,2].

Системи моніторингу мережі, такі як сніфери, з'явилися у 1980-х і 1990-х роках, щоб дозволити адміністраторам спостерігати за потоком трафіку в мережі. Ці інструменти надають інформацію про розподіл трафіку і допомагають виявити проблеми в мережі [3,4].

SNMP (Simple Network Management Protocol - простий протокол управління мережею) був запроваджений як засіб стандартизації моніторингу та управління мережею [5].

У 1990-х і 2000-х роках були розроблені системи виявлення вторгнень (IDS) для виявлення аномального мережевого трафіку і потенційних загроз мережевій безпеці. Було створено десять початкових етапів, які заклали основу для подальшого вдосконалення методів аналізу трафіку.

Ці етапи стали основою для появи більш досконалих інструментів і підходів, які в даний час використовуються для моніторингу та дослідження складних мережевих систем.

Метою розвитку методів аналізу трафіку комп'ютерних мереж завжди було виявлення аномальних подій і забезпечення безпеки інформаційних систем.

Аналіз мережевого трафіку є важливим напрямком досліджень для виявлення атак на комп'ютерні системи та забезпечення їх захисту.

Історія розвитку методів аналізу трафіку показує їх появу та еволюцію в часі, що робить цю сферу цікавою та важливою для вивчення.

Початковий етап розвитку передбачав ручний аналіз трафіку, під час якого фахівці з безпеки аналізували дані та вручну виявляли аномалії. Хоча це забирало багато часу та ресурсів, це стало початком більш ефективних методів [6-8].

Згодом були розроблені статистичні методи аналізу трафіку для виявлення стандартних патернів і аномалій на основі статистичних показників. Однак ці методи часто були обмеженими і недостатніми для виявлення складних атак.

Використання штучного інтелекту дозволило впровадити спеціалізовані системи для аналізу трафіку. Ці системи можуть автоматично виявляти аномалії, використовуючи правила і знання, розроблені експертами з кібербезпеки.

Існують різні методи виявлення аномалій, і найбільш підходящий метод буде залежати від конкретних обставин і наборів даних.

Інструменти виявлення перехоплюють дані мережевого трафіку під час його проходження дротовими та бездротовими мережами і копіюють їх у файл.

Цей процес називається перехопленням пакетів. Комп'ютери спроектовані так, щоб ігнорувати шум трафіку від інших комп'ютерів, але інструменти для перехоплення пакетів працюють інакше [9].

Програмне забезпечення для перевірки пакетів вимагає, щоб мережева карта, яка виступає інтерфейсом між комп'ютером і мережею, була налаштована в режимі проміскуїтету.

Це дозволяє комп'ютеру перехоплювати всі вхідні мережеві зразки і обробляти їх за допомогою інструменту перевірки пакетів.

Елементи, які можна перехопити, залежать від типу мережі. Видимість трафіку в дротових мережах залежить від конфігурації мережевих комутаторів, які централізують зв'язок між кількома підключеними пристроями. Комутатори можуть дозволити інструменту переглядати трафік з усієї мережі або обмежити його до певної частини мережі.

Інструменти перехоплення пакетів у бездротових мережах зазвичай мають можливість перехоплювати лише один канал за раз, якщо тільки хост-комп'ютер не має декількох бездротових інтерфейсів.

У випадку DDoS-атаки зловмисник генерує величезну кількість пакетів або запитів, які зрештою перевантажують цільову систему.

Зловмисник використовує кілька контрольованих або скомпрометованих джерел для підтримки атаки.

Використання сигнатурних методів призвело до розробки систем виявлення вторгнень (IDS), які ідентифікують потенційні вторгнення шляхом виявлення відомих шаблонів атак.

Тому дуже важливо визначити, що є нормальним, а що - аномальним пакетним трафіком в мережі.

Щоб забезпечити доступність хоста, хост повинен отримувати лише стільки трафіку, скільки він може обробити; це називається обмеженням швидкості.

Сучасні методи безпеки можуть розрізняти легальний і нелегальний трафік, аналізуючи окремі пакети. Для цього необхідно розуміти типові характеристики трафіку, який отримує ціль, і порівнювати кожен пакет з цим базовим рівнем. Такий підхід може ефективно запобігати зовнішнім атакам.

Для захисту від таких атак, як SQL-ін'єкції або підробка міжсайтових запитів, що використовують уразливості в додатку, рекомендується використовувати брандмауер веб-додатків (WAF). Крім того, важливо створити захист від нелегітимних запитів, які можуть надходити від поганих API та неочікуваних регіонів через їхні унікальні характеристики. Іноді зловмисники

можуть створювати індивідуальні засоби захисту, вивчаючи шаблони трафіку. Тому ця техніка може бути корисною для зменшення кількості атак [10].

Атака HTTP-флуд - це тип масової розподіленої атаки на відмову в обслуговуванні, яка бомбардує сервер HTTP-запитами, внаслідок чого він стає перевантаженим і не може відповідати на звичайний трафік. Як наслідок, це призводить до відмови в обслуговуванні додаткових запитів від існуючих користувачів.

Для запобігання HTTP-флуду важливо впровадити відповідні заходи безпеки. Ця атака відбувається на 7-му рівні моделі OSI, який охоплює інтернет-протоколи, в тому числі і HTTP.

Щоб максимізувати вплив своєї атаки, зловмисники зазвичай використовують або створюють ботнети. Ботнети створюються шляхом зараження декількох пристроїв шкідливим програмним забезпеченням, які потім використовуються для примноження зусиль зловмисників і запуску більш масштабної атаки [11,12].

HTTP GET-атаки та HTTP POST-атаки. У випадку HTTP GET-атаки кілька комп'ютерів або інших пристроїв координуються для надсилання численних запитів на отримання зображень, файлів або інших елементів з цільового сервера.

Це може призвести до перевантаження цілі вхідними запитами та відповідями, що призведе до відмови обслуговувати додаткові запити з легальних джерел трафіку. Коли користувач заповнює форму на веб-сайті, сервер повинен обробити вхідний запит і відправити дані на рівень зберігання, як правило, в базу даних.

Обробка даних форми і виконання команд бази даних може бути складним завданням, особливо в порівнянні з обчислювальною потужністю і пропускною здатністю, необхідними для відправки POST-запиту. Атака може використовувати цю невідповідність у споживанні ресурсів, надсилаючи безліч POST-запитів на цільовий сервер, поки він не буде перевантажений, що призведе до відмови в обслуговуванні.

Захист від атак HTTP-флуду може бути складним завданням, оскільки запити часто нагадують типовий трафік веб-сайтів. Ці атаки не містять шкідливого програмного забезпечення і не використовують жодних вразливостей. Замість цього сервер отримує потік автентифікованих запитів. Як наслідок, ці атаки можуть залишатися невиявленими на ранніх стадіях через низьке використання пропускної здатності [13,14].

Атаки рівня 7 є складними і часто вимагають багаторівневого захисту. Одним з можливих рішень є використання тесту CAPTCHA, подібного до того, що використовується при створенні облікового запису в Інтернеті, щоб переконатися, що машина, яка надсилає запит, не є роботом.

HTTP-флуд є значною проблемою, оскільки запити спочатку нагадують звичайний трафік веб-сайту. Сервер не піддається впливу шкідливого програмного забезпечення, і не робиться спроб використати вразливості в системі безпеки. Замість цього ці атаки перевантажують сервер великою кількістю авторизованих доступів.

Такий підхід використовує значно менше пропускної здатності, ніж втручання в код сайту, що може ускладнити виявлення атак на ранніх стадіях.

Запобігання атакам 7-го рівня є складним завданням, яке вимагає декількох підходів. Одним із можливих рішень є ідентифікація комп'ютера користувача, щоб визначити, чи не є він ботом, подібно до тесту CAPTCHA, який часто зустрічається при створенні облікового запису в Інтернеті [15,16].

Інші методи боротьби з HTTP-флудом включають розгортання брандмауера, веб-додатків, управління базою даних репутації IP-адрес для моніторингу та вибіркового блокування шкідливого трафіку, а також проведення аналізу.

Сучасні методи аналізу трафіку включають аналіз аномального трафіку, коли система ретельно вивчає стандартні шаблони і реагує на відхилення від цих норм. Останнім часом все більш популярним стає використання машинного навчання і штучних нейронних мереж для аналізу трафіку. Це дозволяє системам автоматично навчатися і адаптуватися до нових типів атак і аномалій [17-19].

Розвиток технологій обробки потокових даних призвів до вдосконалення методів аналізу мережевого трафіку, що дозволило оперативно та ефективно реагувати на зміни в мережі.

Ранні етапи аналізу трафіку були пов'язані з початковим впровадженням Інтернету. З розширенням мереж у 1970-х роках виникла потреба в інструментах для моніторингу та аналізу трафіку.

На початку розвитку комп'ютерних мереж аналіз трафіку мав обмежене застосування і використовувався переважно для моніторингу доступу до ресурсів.

Однак складність мережевих протоколів, таких як TCP/IP, збільшила потребу в більш глибокому аналізі мережевого трафіку. Як наслідок, з часом ця потреба зростала. Програми, здатні аналізувати дані на рівні пакетів, стали важливими інструментами [20-23].

З ускладненням мережевих протоколів, таких як TCP/IP, зросла потреба в більш глибокому аналізі мережевого трафіку. Інструменти, що дозволяють аналізувати дані на рівні пакетів, виявилися цінними.

Поява комп'ютерних мереж призвела до розвитку різних мережевих протоколів, у тому числі широко використовуваного TCP/IP для обміну даними. Як наслідок, для аналізу цих даних, а також для розуміння їх передачі та обробки потрібні були вдосконалені інструменти.

У 1990-х і 2000-х роках були розроблені системи моніторингу мережі, такі як Wireshark і tcpdump, які дозволили адміністраторам спостерігати і досліджувати трафік з більшою точністю.

Розвиток обчислювальних можливостей і мережевих технологій призвів до появи більш ефективних інструментів аналізу трафіку. Програмне забезпечення для збору та аналізу пакетів дозволяє інженерам та адміністраторам більш ефективно контролювати комп'ютерні мережі.

Системи виявлення вторгнень (IDS) є важливим елементом аналізу трафіку в наш час. IDS ідентифікують мережевий трафік, що може бути потенційно зловмисним, і допомогти у реагуванні на потенційні кібератаки [23-25].

Використання машинного навчання призвело до більш ефективного та автоматизованого аналізу трафіку. Ці алгоритми можуть автоматично виявляти будь-яку підозрілу активність або можливі загрози.

Інтеграція шифрування в мережевий трафік створила проблеми для аналітики. Тому фахівцям з безпеки необхідно терміново розробити методи аналізу зашифрованого трафіку і роботи з SSL/TLS.

Сучасні методи аналізу трафіку включають в себе виявлення загроз і дослідження ефективності заходів безпеки. Використання синтетичних даних і моделювання атак може підвищити безпеку мережі.

Важливо не тільки створювати нові інструменти безпеки, але й удосконалювати методи виявлення загроз.

Розвиток методів аналізу трафіку - це безперервний процес, оскільки кіберзлочинці постійно вдосконалюють свої методи. Тому методи кібербезпеки повинні постійно розвиватися, щоб залишатися ефективними..

1.2 Наявні методи аналізу трафіку

Розгортання машинного навчання (ML) та аналітики є важливою тенденцією. ML допомагає сучасним системам аналізу трафіку виявляти аномалії, точно класифікувати патерни та прогнозувати потенційні загрози.

З появою шифрування трафіку оцінка зашифрованого трафіку стала необхідною. Сучасні інструменти покладаються на різні підходи, такі як аналіз метаданих, для виявлення потенційних ризиків, вбудованих в зашифрований трафік.

Оскільки пропускна здатність мереж продовжує розширюватися, а обсяги даних збільшуються, сучасні методи аналізу трафіку повинні швидко і ефективно обробляти великі обсяги інформації в режимі реального часу [26].

Програми, що працюють у комп'ютерній мережі, зазвичай передають дані через розпізнані апаратні порти. Основною метою класифікації пакетів є

визначення місцезнаходження TCP SYN-пакету, який встановлює серверну частину клієнт-серверного з'єднання на основі TCP [27].

Після встановлення з'єднання необхідно визначити, чи доступний цільовий номер порту конкретній програмі, яка передала пакет. Класифікація пакетів на основі UDP не встановлює з'єднання, але працює подібно до класифікації пакетів на основі TCP.

Цей підхід має перевагу завдяки простоті реалізації та швидкості операцій класифікації.

Однак є й певні недоліки. Зокрема, деякі програмні продукти не мають портів, зареєстрованих в Адміністрації з присвоєння номерів в Інтернеті (IANA), що ускладнює управління простором IP-адрес.

Програмні додатки можуть використовувати для виконання певних завдань інші порти, ніж ті, що запрограмовані операційною системою. Наприклад, HTTP-сервер на Unix-подібних системах може використовувати порт, відмінний від "80".

Помилки в шифруванні на рівні IP можуть спричинити плутанину в заголовках TCP і UDP [28].

Альтернативний підхід полягає в тому, щоб зменшити залежність від порту і забезпечити надійну передачу даних шляхом впровадження механізмів відновлення стану сеансу і ретельної перевірки вмісту кожного пакета.

Ретроспективний аналіз трафіку передбачає запис даних про трафік на диск і подальший їх аналіз.

Перехоплюючи мережевий трафік, можна ідентифікувати події, дії та помилки, що відбуваються в додатках, активності користувачів та IT-інфраструктурі. Значення метрик записуються в базу даних, щоб визначити відповідний момент і кореляцію.

Більшість систем спостереження використовують метрики і бали, щоб надати огляд подій в будь-який момент часу.

Дослідження, проведені за останні 15 років щодо різних типів мережевого трафіку, вказують на те, що цей трафік є самоподібним або має фрактальні властивості.

Самоподібність - це характеристика процесу, при якій його поведінка і зовнішні ознаки залишаються незмінними при аналізі в різних масштабах.

Тому методи моделювання і розрахунку, що використовуються в мережевих системах, заснованих на пуассонівських потоках, можуть не давати повного і точного уявлення про мережеву активність.

Крім того, самоподібний трафік має чітку структуру, яка залишається незмінною в різних масштабах.

У ситуаціях, коли середній рівень трафіку є відносно низьким, можуть спостерігатися деякі відхилення в реалізації.

Самоподібний трафік, що проходить через вузли мережі, може спричинити погіршення продуктивності, що призводить до збільшення втрат пакетів і затримок.

На практиці пакети надходять на вузол не поодинці, а наборами, що може призвести до їх втрати через обмежений буфер, розрахований традиційними методами.

Щоб зменшити кількість первинних і вторинних помилок на 5%, класифікація на основі корисного навантаження однорангового трафіку передбачає перевірку його сигнатур на прикладному рівні.

Класифікація на основі корисного навантаження може бути структурована за допомогою наступних методів перевірки та обробки:

- PBNS (Packet Based No State);
- PBFS (Packet Based Per Flow State);
- MBFS (Message Based Per Flow State);
- та MBPS (Message Based Per Protocol State).

Поширення розподілених загроз і атак, в тому числі DDoS-атак, вимагає створення методів, здатних надійно виявляти і відбивати ці загрози на різних рівнях мережі.

Для забезпечення комплексної стратегії безпеки сучасні системи аналізу трафіку повинні доповнюватися додатковими заходами безпеки, такими як системи виявлення вторгнень (IDS/IPS), брандмауери, антивірусне програмне забезпечення та інші інструменти.

Ці сучасні методи аналізу трафіку дозволяють проводити високорівневий аналіз протоколів і додатків, полегшуючи ідентифікацію і класифікацію конкретних типів трафіку додатків.

Автоматизовані та інтелектуальні інструменти дозволяють швидше виявляти загрози та автоматично реагувати на заходи безпеки.

Вищезгадані тенденції свідчать про те, що сучасні методи аналізу трафіку виходять за рамки простого спостереження і переходять до більш складних і розумних систем, які можуть кваліфіковано виявляти і захищати від цілого ряду загроз в складних мережевих умовах.

На сьогоднішній день одним з найпоширеніших методів виявлення аномалій є інструменти виявлення атак (ADT) [29].

Ці інструменти можуть включати алгоритми, цілі системи або окремі програми. Вони призначені для виявлення та реагування на підозрілу активність, спрямовану на обчислювальні або мережеві ресурси. Однак жоден з існуючих інструментів виявлення атак не може повністю виявити аномальну активність в мережевому трафіку.

Існує дві основні категорії моніторингу мережі для виявлення аномалій в них входять пасив нийантивий

Пасивний моніторинг мережі та Активний моніторинг мережі. Пасивний це

Зонди в комп'ютерній мережі отримують дані з мережі та оцінюють їх. Ці дані можуть бути спеціально призначені для зондів (наприклад, події, передані через SNMP), або це може бути дублікат мережевої активності, що відбувається незалежно від того, підключений зонд чи ні.

Активний моніторинг мережі на відміну від пасивного моніторингу, може включати датчики, які генерують додатковий трафік, що передається мережею. Цей трафік дозволяє регулярно оцінювати доступність і загальні параметри відстежуваних сервісів, мережевих каналів і пристроїв.

Активний та пасивний моніторинг мережі для виявлення аномалій:

Хоча може здатися, що активний моніторинг підвищує ефективність пасивного моніторингу, що робить його кращим вибором, активний моніторинг створює проблему, яка полягає в тому, що він генерує надлишкові мережеві дані.

Активний моніторинг поєднує пристрої моніторингу з виробничою мережею, що створює ризики для безпеки. Тому важливо враховувати потенційні ризики та недоліки активного моніторингу перед його впровадженням. Процес моніторингу не є повністю прозорим, а самі дані моніторингу можуть викликати проблеми з функціональністю мережі та аномалії, такі як збільшення навантаження на і без того завантажений сервер [30,31].

Методи виявлення аномалій сигнатурні або засновані на знаннях

Сигнатура точно описує тип даних, які шукає система. Наприклад, сигнатура може передбачати пошук пакетів з однаковими IP-адресами джерела та призначення або пошук певного вмісту в пакеті.

Статистичний підхід кількісно оцінює кількість переданих даних, які мають спільні риси. Наприклад, це може бути кількість TCP-з'єднань, ідентифікованих кожні п'ять хвилин. Аномалія виявляється, коли поточне значення (тобто кількість запитів за останні п'ять хвилин) значно відхиляється від встановленого базового рівня. Крім того, можна проаналізувати зміни в розподілі пакетів залежно від порту призначення.

Реальність виявлення аномалій не така проста, як може здатися. У певний момент виникає проблема, яка суттєво ускладнює можливість виявлення аномалій. Нижче ми розглянемо дві найважливіші проблеми.

Хибні спрацьовування відрізнити нормальний трафік від аномального може бути непросто. Те, що колись вважалося нормальним трафіком, може стати аномальним у майбутньому. Навіть за відсутності мережевих аномалій передача

даних може змінюватися. У цьому контексті виявлення зосереджується на оцінці ймовірності. Хоча методи і системи виявлення можуть відрізнятися, фундаментальна концепція залишається незмінною. Події присвоюється оцінка, і якщо вона перевищує заздалегідь визначений поріг, її ідентифікують як аномалію.

Поріг виявлення аномалій впливає на чутливість системи. Якщо чутливість занадто висока, система може швидко виявляти аномалії, але ціною збільшення помилкового маркування подій як аномальних, що призводить до хибних спрацьовувань [32,33].

І навпаки, зменшення чутливості зменшує кількість хибних спрацьовувань, але також зменшує виявлення реальних аномалій, оскільки деякі з них можуть бути недостатньо значними, щоб їх реєструвати як такі.

Наприклад, хибне спрацьовування може виникнути через непередбачене оновлення операційної системи, яке передає великий обсяг даних, або через несподівану ситуацію, що призводить до одночасного доступу до інтернет-магазину компанії незвично великої кількості клієнтів.

Загалом, неможливо виявити всі мережеві аномалії без помилкових спрацьовувань.

Хибні спрацьовування можуть помилково ідентифікувати легітимний трафік або сервіси як проблемні та обмежувати їхню роботу під час автоматичної обробки подій. Однак ручний аналіз таких аномалій вимагає значних витрат часу і зусиль.

Виявлення застарілих аномалій стає складним завданням при моніторингу зашифрованого трафіку.

Шифрування даних у комп'ютерних мережах постійно розширюється і вдосконалюється з міркувань конфіденційності та безпеки. Зашифрований зв'язок також впливає на виявлення аномалій, оскільки шифрування зменшує кількість даних, які можна відстежувати та аналізувати.

Наприклад, зашифровані електронні листи не дають доступу до електронних адрес. Дуже важливо знати, який рівень шифрування використовується. Більшість повідомлень шифруються лише на рівні додатків, що

дозволяє проводити статистичний аналіз IP-адрес, портів призначення тощо, що робить можливим виявлення аномалій [34].

Однак шифрування значно обмежує діапазон виявлених аномалій. Zeek представлений як одне з можливих рішень для аналізу аномального трафіку на підприємстві. Після первинного огляду Zeek надає велику колекцію журналів, які всебічно описують мережеву активність. Сюди входять HTTP-сесії із запитуваними URI, ключовими заголовками, типами MIME і відповідями сервера, а також DNS-запити і відповіді, SSL-сертифікати, основний вміст SMTP-сесій та інші деталі. Zeek також пропонує різноманітні інструменти аналізу та виявлення на додаток до ведення журналів [35].

Ці можливості включають вилучення файлів з HTTP-сесій, взаємодію із зовнішніми реєстрами для виявлення шкідливого програмного забезпечення, ідентифікацію популярних веб-сайтів і додатків, виявлення SSH-атак, повідомлення про вразливі версії програмного забезпечення, знайдені в мережі, і перевірку ланцюжків SSL-сертифікатів.

Якщо коротко, то Zeek оптимізований для інтерпретації мережевого трафіку і створення логів на його основі. Одним з основних недоліків цього інструменту є його крута крива навчання. Підприємствам, які шукають рішення для виявлення сигнатур, варто розглянути можливість використання системи виявлення вторгнень, розробленої в ході кваліфікаційної роботи.

1.3 Комплексні системи захисту інформації

Комплесні системи захисту інформації (КСЗІ) - це сукупність технічних, інженерних та організаційних стратегій, спрямованих на захист інформації від несанкціонованого доступу, витоку та розголошення.

Основною метою розробки системи інформаційної безпеки є забезпечення надійності даних. Система інформаційної безпеки складається зі структурованих

об'єктів і суб'єктів захисту даних, які включають методи, засоби захисту та заходи захисту.

Компоненти системи інформаційної безпеки є як інтегральними, так і організаційними. Вони відіграють життєво важливу роль у впровадженні заходів безпеки, а також забезпечують ефективне функціонування системи.

Система складається з взаємопов'язаних елементів. Мета системи запобігання вторгненням (СЗІ) - об'єднати всі компоненти захисту в єдине ціле. Кожен компонент повинен виконувати свою функцію і забезпечувати роботу інших компонентів. Компоненти повинні бути пов'язані між собою логічно і технологічно узгоджені.

Системність КСЗІ безпосередньо пов'язана з надійністю захисту інформації. Ризик порушень безпеки зростає, якщо окремі компоненти конфліктують між собою [36-38].

Системи виявлення вторгнень (IDS) - це програмне або апаратне забезпечення, яке виявляє несанкціонований доступ до комп'ютерної системи або мережі або контроль над ними, переважно через Інтернет. Система управління інформацією та подіями безпеки (SIEM) централізує інформацію про активність шкідливого програмного забезпечення або порушення стандартних операцій. Система SIEM отримує дані з різних джерел і використовує методи фільтрації загроз, щоб відрізнити несанкціоновану активність від помилкових спрацьовувань.

Про загрози безпеці повідомляється адміністратору або в оперативний центр безпеки. Системи виявлення вторгнень (IDS) можуть виявляти мережеві атаки, а системи запобігання вторгненням (IPS) можуть виявляти раніше невідомі атаки і вживати заходів для їх запобігання, наприклад, вихід з системи або запуск визначеного адміністратором сценарію.

На практиці програмні та апаратні рішення часто поєднують функціональність двох різних систем, що призводить до створення комбінацій IDPS (IDS та IPS) [39-41].

Існують різні типи IDS, від тих, що встановлюються на окремих комп'ютерах, до тих, що використовуються у великих мережах. Найпоширенішими класифікаціями є NIDS (мережеві системи виявлення вторгнень) і HIDS (системи виявлення вторгнень на основі хостів).

Прикладом системи виявлення вторгнень на основі хоста (HIDS) може бути система, яка контролює життєво важливі файли операційної системи, тоді як прикладом системи виявлення вторгнень на основі мережі (NIDS) може бути система, яка ретельно перевіряє вхідний мережевий трафік. IDS також можна згрупувати за методами виявлення загроз. Найвідомішими методами є виявлення на основі сигнатур, яке розпізнає аномальні шаблони як шкідливе програмне забезпечення, і виявлення аномалій, яке виявляє відхилення від "нормального" трафіку, часто з використанням машинного навчання.

Розглянемо традиційну класифікацію систем запобігання вторгненням (IPS): системи мережевого рівня, які перенаправляють трафік даних через маршрутизатор; системи рівня хоста, які виявляють зміни на окремому комп'ютері, наприклад, аналізуючи журнали або мережеву активність; і системи для оцінки вразливостей системи.

Зібрані дані повинні використовуватися системою виявлення вторгнень (IDS), щоб визначити, чи існує загроза мережевій безпеці..

Основними обов'язками податкової служби є:

– збір даних;

– інтерпретація даних та представлення результатів.

Системи можна класифікувати за кількома ознаками, включаючи тип зібраних даних, методи збору даних, метод інтерпретації даних та метод представлення результатів.

Крім того, несистемні характеристики можна розглядати як реакцію на результат, яка може бути як інформативною, так і активною.

У першому випадку зацікавлені сторони отримують інформацію.

Активні заходи передбачають блокування адрес зловмисника. За цією ознакою часто розрізняють IDS та IPS. IPS може включати в себе будь-які IDS. Моніторинг мережі може використовувати дані з вузла мережі або дані мережевого рівня [42].

Типи даних, пов'язані з окремим вузлом і його взаємодією, називаються типами даних вузла мережі.

Аналізуючи ці дані, можна визначити, чи відбулася атака на цей вузол. Збір даних безпосередньо з вузла часто є більш практичним, хоча і не обов'язковим.

Наприклад, мережеві сканери можуть отримати список відкритих портів на певному хості ззовні без виконання коду на ньому. До цієї категорії належать такі типи даних, кожен з яких має певні показники: активність мережевого хоста, налаштування мережевого хоста, файлові дані (списки, контрольні суми, метадані тощо) і дані процесів. До цієї категорії належать такі типи даних, кожен з яких має певні показники: активність мережевого хоста, налаштування мережевого хоста, файлові дані (списки, контрольні суми, метадані тощо) та дані процесів.

До цієї категорії належать такі типи даних, кожен з яких має певні показники: активність мережевого хоста, налаштування мережевого хоста, файлові дані (списки, контрольні суми, метадані тощо) та дані процесу. Важливо враховувати всі ці фактори.

Вузлами можуть бути як робочі станції, що не надають послуг як сервери, так і справжні сервери. Ці хости вразливі до різних методів атаки, включаючи додаткове навчання нейронної мережі та виявлення вузлів атаки. Тому будь-яку взаємодію з вузлом слід розглядати як потенційну спробу атаки. Мережеві дані дають повне уявлення про загальну мережеву взаємодію.

Повні мережеві дані зазвичай не збираються через їхню ресурсоємність. Передбачається, що зловмисник може бути відсутній в мережі або йому може знадобитися зв'язок із зовнішнім світом.

У таких випадках IDS досліджує трафік, що проходить через маршрутизатор. Для цього маршрутизатор має порт SPAN, який перенаправляє трафік на IDS. Дані також можна збирати з хоста, на якому працює IDS, що

забезпечує додатковий контроль. Трафік мережі також можна збирати на її вузлах, але це призводить до того, що мережевий адаптер хоста перехоплює весь трафік, який не очікується під час нормальної роботи.

Існує три методи збору даних від систем запобігання вторгненням (IPS): активний, пасивний і змішаний (комбінація обох методів). Пасивне виявлення передбачає спостереження за ситуацією і є найпоширенішим методом, який використовують IDS.

Хост-системи також зазвичай використовують цей метод.

Наприклад, автори утримуються від спроб видалити системний файл, увійшовши в систему як користувач і підтвердивши видалення. Замість цього вони аналізують права доступу до файлу за шаблоном бази даних і видають попередження, якщо збіг є невдалим.

Помилки в активному методі є результатом як відомих, так і невідомих дій у нечіткій системі. Реакції на ці дії потім аналізуються за допомогою інформаційної бази даних. Цей метод часто використовується в сканерах вразливостей і може застосовуватися як для аналізу баз даних, наприклад, вивчення типових реакцій на шаблони SQL-ін'єкцій, так і для поведінкового аналізу, який включає в себе таргетування і запити відповідей.

Наприклад, надсилання неправильного IP-пакету може призвести до аварійного завершення роботи вразливого сервера і припинення відповіді. Можна впровадити рішення для виявлення аномальної активності.

Останніми роками мережі стали найважливішим інформаційним ресурсом для захисту бізнесу.

Компанії розглядають несанкціонований доступ до інформації як пряму загрозу своїм інтересам.

Тому системи інформаційної безпеки (ІБ), такі як антивірусне програмне забезпечення та системи виявлення вторгнень, стали невід'ємними компонентами корпоративних мереж.

Система ІБ - це складна програмно-апаратна платформа, що базується на математичних моделях і методах [43,44].

Висока надійність систем забезпечується шляхом перевірки програмного коду та верифікації реалізованих методів і моделей. Оцінка відповідності проводиться відповідно до критеріїв продуктивності СЗІ, визначених на етапі специфікації, з використанням математичних методів.

Аналіз різних показників дозволяє оцінити як окремі елементи, так і загальну якість реалізації системи. Важливо зазначити, що атаки на КЗІ з кожним роком стають все більш витонченими, частими та масштабними. Тому вдосконалення та розвиток систем виявлення вторгнень в КСК є нагальним завданням. Основне призначення цих систем - виявлення мережевих атак і запобігання несанкціонованому доступу або використанню ресурсів.

Для захисту інформаційних активів необхідно оцінити існуючі системи виявлення атак та запобігання вторгнень з урахуванням безперервного та швидкого розвитку методів та способів зловмисного впливу на програмне забезпечення КМК.

Велика кількість інформаційно-технологічних систем, доступних на ринку, змушує користувачів обирати найкращу систему виявлення атак та запобігання вторгненням [45-47].

Однак цей вибір вимагає аналізу як поточних досягнень, так і потенційних майбутніх розробок.

В даний час численні системи класифікуються як IDS або IPS.

1.4 Постановка задачі

Метою даної кваліфікаційної роботи є вивчення існуючих методів аналізу мережевого трафіку та сучасних систем виявлення вторгнень.

Завдання полягає у виявленні характеристик трафіку, характерних для різних типів кібератак, та створенні методики аналізу трафіку в режимі реального часу для виявлення аномалій, які можуть свідчити про атаку. створенні методу аналізу трафіку, який може своєчасно виявляти порушення безпеки в IPS.

Запропонована методика буде реалізована у вигляді реально працюючого методу на основі штучного інтелекту.

Насамкінець розроблена методика буде оцінена шляхом її тестування на реальному трафіку та даних відомих кібератак.

Оцінка ефективності запропонованого підходу для виявлення кібератак на ксзі є одним з завдань на кваліфікаційну роботу.

Запропонований підхід має бути впроваджений для підвищення рівня кібербезпеки.

Для досягнення цієї мети використовуються різні методи аналізу трафіку.

До них відносяться сигнатурний аналіз, який базується на заздалегідь визначених сигнатурах атак, аналіз аномалій, який фокусується на виявленні відхилень від нормальної поведінки, кореляційний аналіз, спрямований на виявлення взаємозв'язків між подіями, і використання методів машинного навчання для навчання системи виявлення атак на основі великого обсягу даних.

Проблема пов'язана зі збільшенням кількості нелінійних атак, які можуть адаптуватися до захисних заходів. Для вирішення цієї проблеми потрібні нові підходи, які враховують динаміку кіберзагроз та їх еволюцію.

Наше дослідження спрямоване на розробку та оптимізацію методів аналізу трафіку для виявлення як відомих, так і раніше не виявлених атак.

2 МАТЕМАТИЧНА МОДЕЛЬ ВДОСКОНАЛЕНОГО АЛГОРИТМУ АНАЛІЗУ ТРАФІКУ

2.1 Дослідження й порівняння існуючих методів аналізу трафіку

Дослідження математичних моделей для існуючих методів аналізу трафіку є важливою сферою в галузі комп'ютерних наук і мереж. Ці дослідження спрямовані на створення ефективних і точних математичних моделей для ретельного вивчення різних аспектів мережевого трафіку.

Основні напрямки досліджень включають:

– моделювання трафіку а саме: розробка математичних моделей для опису характеристик і поведінки мережевого трафіку з використанням стохастичних процесів для моделювання невизначеності та мінливості трафіку;

– аналіз пропускної здатності також включає в себе оволодіння методами оцінки та аналізу пропускної здатності мережі;

– метою є розробка методів для виявлення аномалій в мережевому трафіку, використовуючи машинне навчання та статистичні методи;

– оптимізацію розподілу ресурсів та маршрутизації в мережах шляхом розробки комплексних моделей. Вивчення впливу різних параметрів на продуктивність мережі. Проводити дослідження математичних моделей для виявлення та запобігання мережевим загрозам та атакам у сфері мережевої безпеки. Розробка моделей для прогнозування пропускної здатності в залежності від різних факторів;

– проведення тестів на реальних мережах та збір емпіричних даних для перевірки отриманих результатів.

Сніфери - це програмне забезпечення, яке перехоплює весь мережевий трафік, в основному використовується для діагностики мережі. Цей інструмент є високоефективним для аналізу мережі.

Сніффер пакетів може бути програмним або апаратним пристроєм, призначеним для перехоплення, реєстрації та аналізу мережевого трафіку і даних.

Такі інструменти допомагають ідентифікувати, класифікувати та вирішувати проблеми мережевого трафіку, пов'язані з типом програми, джерелом або місцем призначення.

На ринку доступно багато інструментів, більшість з яких покладаються на цю технологію [48,49].

Для перехоплення мережевого трафіку використовуються інтерфейси прикладного програмування (API), широко відомі як pcap для Unix-подібних систем або libcap для систем Windows.

Ці дані потім можуть бути проаналізовані за допомогою високоякісних сніферів пакетів, що дозволяє вам без особливих зусиль виявити джерело проблеми і запобігти її повторенню в майбутньому.

Щоб зрозуміти важливість сніферів, дуже важливо розуміти механіку інтернет-маршрутизації.

Почнемо з основ: кожен електронний лист, веб-сторінка і файл розсіюються по всьому Інтернету у вигляді численних маленьких пакетів.

Керовані пакети даних передаються через стек протоколів, який називається протокол управління передачею/Інтернет-протокол (TCP/IP), який поділяється на чотири рівні.

Система складається з апаратного рівня, протоколу управління передачею (TCP) рівень, рівень Інтернет-протоколу (IP) та рівень прикладного протоколу.

Кожен пакет проходить через прикладний рівень мережі до TCP, який призначає йому номер порту. Потім пакет переходить на рівень IP і отримує IP-адресу призначення. Після того, як пакет має номер порту та IP-адресу, він стає готовим до передачі через Інтернет.

Дані пакета перетворюються на мережеві сигнали на апаратному рівні.

Після прибуття до місця призначення, дані про маршрутизацію пакета (номер порту, IP-адреса і т.д.) номер порту, IP-адреса тощо) видаляються, і пакет продовжує свій шлях через стек протоколів нового мережевого протоколу.

Коли пакет досягає вершини, він відновлюється до свого початкового стану.

Сніфери перехоплюють дані трафіку, коли він проходить через дротові або бездротові і копіюють їх у файл - процес, відомий як перехоплення пакетів.

Хоча комп'ютери ігнорують активність трафіку з інших пристроїв, сніфери пакетів можна використовувати для аналізу та моніторингу мережевого трафіку.

Інші комп'ютери або програми-перехоплювачі пакетів можуть повернути цей процес у зворотному напрямку.

Під час встановлення програми необхідно перевести мережеву інтерфейсну карту (NIC) в бітовий режим, що дозволить вашому комп'ютеру перехоплювати і обробляти всі дані, які проходять через сніфер пакетів і мережу.

Перехоплені дані залежать від типу мережі.

У дротових мережах розташування мережевих комутаторів відповідає за централізацію зв'язку з декількох підключених пристроїв, тим самим визначаючи, чи буде трафік видимим для мережевого сніффер по всій мережі або лише її частини.

У бездротових мережах перехоплення даних дещо складніше.

У мережевих середовищах інструменти перехоплення пакетів зазвичай здатні перехоплювати тільки один канал за за один раз, якщо тільки хост-комп'ютер не має декількох доступних бездротових інтерфейсів.

TCPDUMP це широко використовуваний інтерфейс командного рядка (CLI) і інструмент для дослідження пакетів з відкритим кодом, сумісний з платформами Unix і Linux.

Він був розроблений у 1987 році в Національній лабораторії Лоуренса Берклі і опублікований кількома роками пізніше.

Мова програмування С містить бібліотеку libpcap, яка перехоплює мережеві дані.

Libpcap надає інтерфейси для типових Unix-похідних систем, таких як FreeBSD та Linux. Для платформи Windows інтерфейсом libpcap є WinDump, широко відомий як Windump.

WinPcap використовується як адаптація бібліотеки libpcap для Windows.

Архітектура WinPCAP розширює стандартні можливості операційних систем сімейства Win32, дозволяючи передавати і приймати мережеві дані, оминаючи стек протоколів операційної системи і безпосередньо взаємодіючи з мережевим адаптером.

Крім того, він пропонує високорівневі API для низькорівневого управління процесами в додатках. WinPCAP складається з трьох основних компонентів:

– packet.vxd - драйвер пристрою перехоплення пакетів;

– packet.dll - низькорівнева динамічна бібліотека;

– libpcap - статична високорівнева бібліотека.

Ця архітектура використовується для розробки програм обробки пакетів, які функціонують як на Windows, так і на UNIX операційних системах [50].

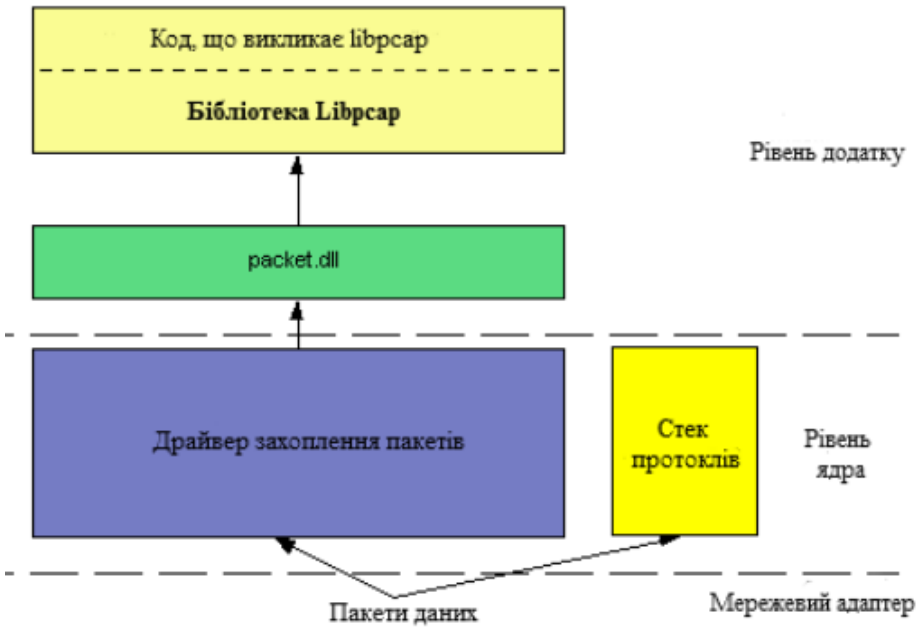


Рисунок 2.1 – Структура стека захоплення пакетів

Програмісти створили бібліотеку libpcap як незалежний від платформи API, який може працювати у різних програмах і усунути системні залежності для модулів збору даних модулів збору даних у кожній програмі. TCPDump розглядається як препаруючий пристрій.

За замовчуванням TCPDump перехоплює і друкує пакети з мережі.

Будь-які додаткові функції, наприклад, зберігання, виконуються спеціальними командами. Програма працює наступним чином:

За замовчуванням TCPDump перехоплює і друкує пакети з мережі:

– команди CLI читають і записують захоплений файл з мережі мережі у пакет перехоплення пакетів (PCAP) ;

– потім пакети фільтруються на основі певних заданих параметрів;

– захоплені дані виводяться на екран, дотримуючись встановлених параметрів.

Основний недолік TCPDump полягає в тому, що він не може запропонувати мережевому адміністратору візуального графічного інтерфейсу перехоплених даних для детального аналізу, оскільки він покладається виключно на CLI [51].

Однак, оскільки він представлений у текстовому форматі, він зручний для віддаленого використання через З'єднання Telnet.

Це більш легка і портативна утиліта для перехоплення пакетів завдяки інтерфейсу командного рядка.

Мережеві адміністратори використовують її для віддаленого доступу до мережевих пристроїв.

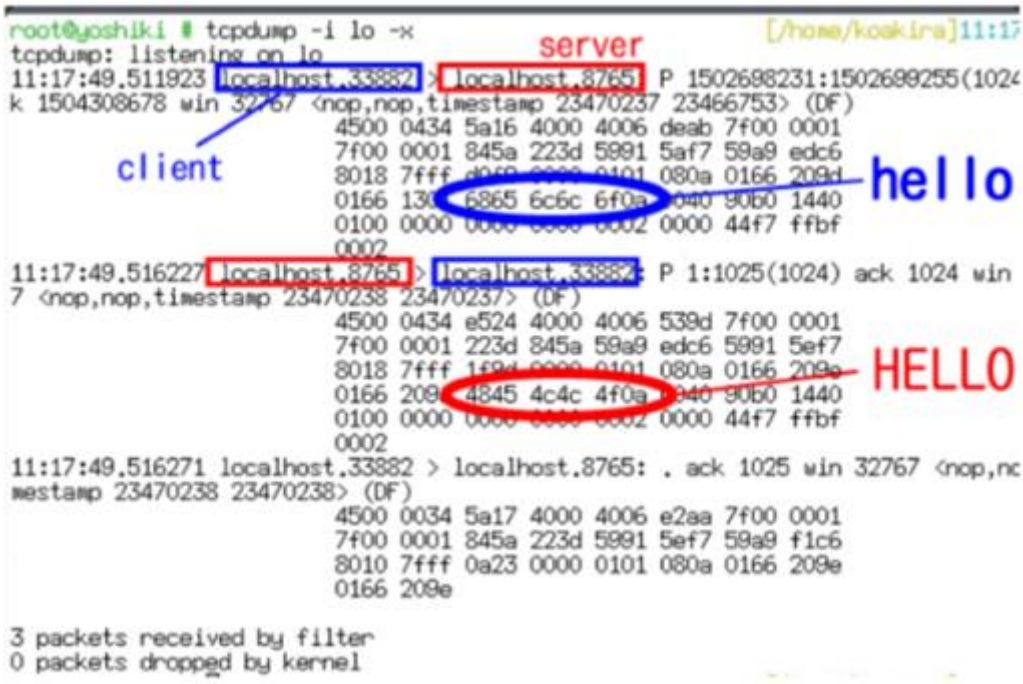


Рисунок 2.2 – Потік характеристок TCP/IP

У TCPDump є декілька інших недоліків. До них відносяться:

– обмеження в аналізі трафіку, можна використовувати тільки протоколи на основі TCP;

– звітність обмежується інформацією, що міститься в пакетах. Якщо в трафіку підміняється IP-адреса, система не може повідомити жодних додаткових деталей;

– пакети, заблоковані брандмауером, не відображаються у звіті.

Wireshark Наприкінці 1997 року вчений Джеральд Комбс винайшов інструмент До травня 2006 року він був відомий під назвою Ethereal, після чого його назву було змінено на Wireshark. Wireshark - це безкоштовне програмне забезпечення з відкритим вихідним кодом, яке пропонує графічний інтерфейс користувача (GUI) для аналізу і перехоплення мережевих пакетів, і написане за допомогою мови програмування C.

Мовою програмування C, випущеною під ліцензією GNU General Public License (GPL). Wireshark сумісний з різними Unix-подібними операційними системами, включаючи Mac OS X, Linux, платформу Solaris і Microsoft Windows.

Інтерфейс командного рядка, відомий як TShark, дозволяє користувачам взаємодіяти з програмою за допомогою команд і схожий на TCPDump, але з додатковими графічними можливостями. TShark підтримує ряд протоколів і надає можливості фільтрації та сортування. Зокрема, Wireshark використовується в інструментах мережевого криміналістичного аналізу (Network Forensic Analysis Tools, NFAT) в організаціях.

Wireshark перехоплює пакети з живих мереж і дозволяє переглядати раніше збережені файли даних. Для перехоплення пакетів підтримується формат "PCAP". Захоплені дані відображаються в байтовому і шістнадцятковому форматах, наочно демонструючи захоплену інформацію. Текст демонструє різноманітність типів пакетів і використовуваних протоколів. Крім того, він полегшує користувачеві компіляцію даних пакетів у TCP-потік.

Інтерфейс складається з трьох панелей; панель підсумків або список пакетів відображає проаналізовані пакети, включаючи номер кадру, дату, час, IP-адресу призначення та джерела, протоколи верхнього рівня, довжину пакета та інформацію про вміст трафіку, виділену кольором для кожного типу перехопленого пакета. Друга панель містить деталі перехопленого пакета. Щойно пакет буде ідентифіковано на панелі списку пакетів, його деталі з'являться на наступних двох панелях: деталі та байтова або шістнадцяткова панель.

Панель деталей представлена у вигляді ієрархічної структури протоколів, записаних для різних додатків, включаючи протокол керування передачею (TCP), протокол користувацьких дейтаграм (UDP), протокол керуючих повідомлень Інтернету (ICMP), протокол передачі гіпертексту (HTTP) та інші. Абревіатури для технічних термінів визначаються при першому використанні. Уникаючи особистих поглядів, мова, що використовується в тексті, є чіткою, об'єктивною і нейтральною, що відповідає принципам академічного письма. Також включено протокол передачі тексту (HTTP) та ін. Третя панель, відома як панель даних або байт-панель, показує необроблені перехоплені дані і представляє байти пакета в шістнадцятковому форматі, кодуванні ASCII і текстовому форматі.

Важливо пам'ятати, що при запуску Wireshark активує мережеву карту в бітовому режимі.

Режим дозволяє сніфферу відстежувати весь трафік, що проходить через цей інтерфейс, а не тільки трафік, що прямує до одного з налаштованих інтерфейсів Крім бітового режиму, налаштування порту відіграє важливу роль. Віддзеркалення портів може відображати дані з будь-якої точки мережі, незалежно від того від того, чи покриває порт всю мережу [52,53].

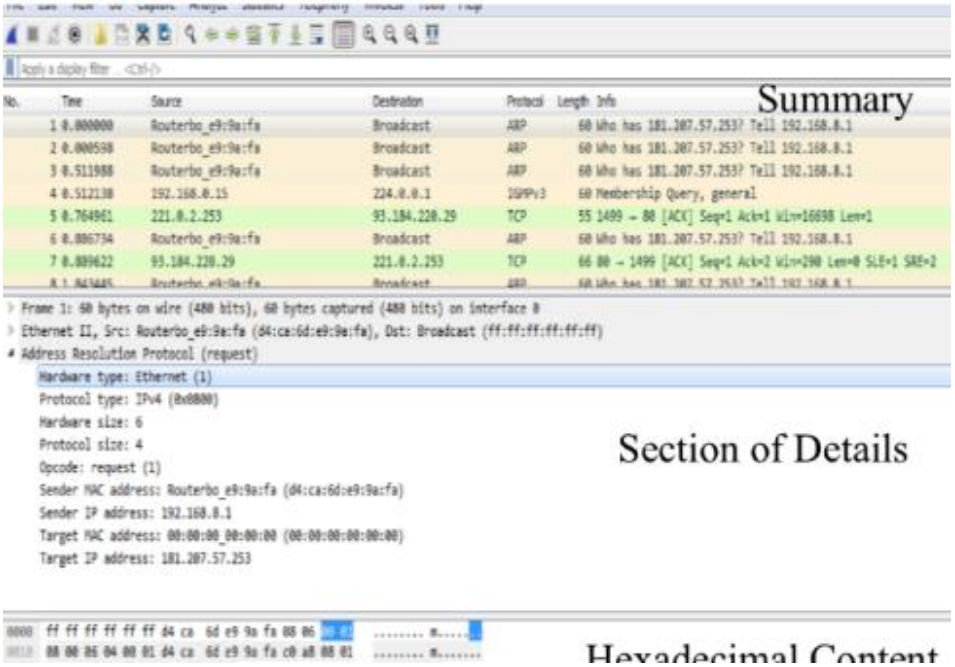


Рисунок 2.3 – Інтерфейс Wireshark

Colasof це інструмент аналізу мережевих протоколів, призначений для роботи виключно в операційній системі Windows для особистого використання мережевими адміністраторами. Інструмент також має можливості цілодобового моніторингу мережі. Capsa пропонує безкоштовні інструменти Colasoft, які забезпечують простоту використання для аналізу пакетів в режимі реального часу, а також надійний криміналістичний і поглиблений аналіз протоколів.

Програмне забезпечення надає ряд функцій, включаючи відкриття декількох інтерфейсів в одному екземплярі, пропонуючи користувачеві графічні інтерфейси і матричні представлення.

Крім того, воно дозволяє користувачеві створювати звіти, журнали та оповіщення за допомогою голосових та електронних сповіщень, якщо він володіє ліцензійною версією.

Аналіз пакетів забезпечує глибоке розуміння характеристик мережі. Крім того, графічний інтерфейс програми включає в себе ряд функцій, які графічно відображають перехоплену інформацію у вигляді графіків і матриць, показуючи протоколи, що використовуються для кожної характеристики мережевого трафіку.

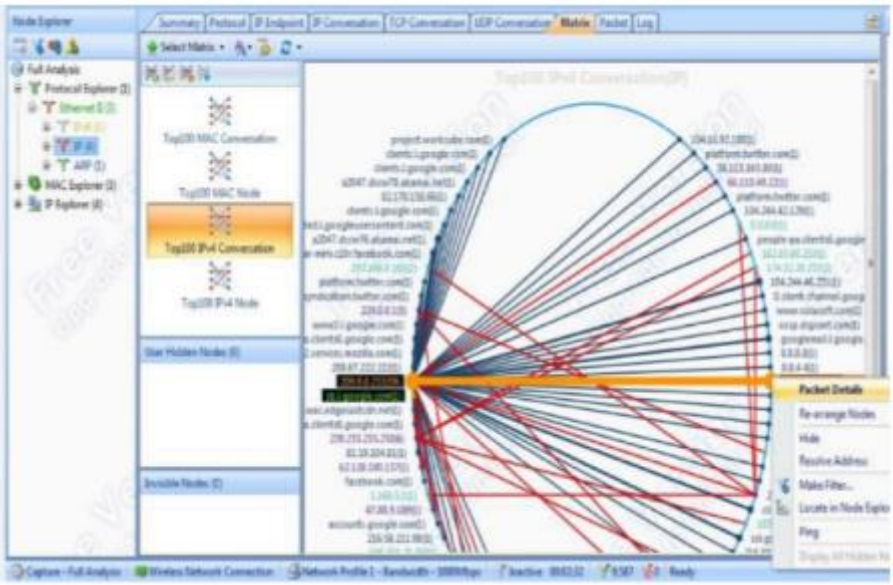


Рисунок 2.4 – Інтерфейс Colasoft

Використання Colasoft має кілька недоліків. По-перше, Colasoft є дорогим додатком, хоча існує безкоштовна версія, яка має обмежені функції [54,55].

Наприклад, безкоштовна версія не пропонує електронні та голосові сповіщення. Крім того, Colasoft працює лише в операційній системі Microsoft Windows і підтримує лише 300 протоколів. Це менше, ніж у інших пакетних інструментів, таких як Wireshark, що є недоліком.

Таблиця 2.1 Порівняльні характеристики TCPDump,WireShark,Colasoft

Параметри	Сніферні інструменти		
	TCPDump	WireShark	Colasoft
Відкритий код	+	+	-
Операційні системи	Linux	Linux,Windows	Windows
Число підтримки протоколів	TCP/IP	Більше 300	300
Користувацький інтерфейс	CLI	CLI і GUI	GUI
Вартість	Безкоштовно	Безкоштовно	999$
Libpcap основа	+	+	-
Визначення прихованих данних	-	+	+
Використання місця	484КБ	449МБ/89МБ	32МБ
Відображення шару протоколу	-	+	+
Декодування протоколу	Hex,ASCII	Hex,ASCII	Hex,ASCII, EBDIC
Відновлення TCP потоку	-	+	+
Кілька інтерфейсів	-	-	+
Сповіщення знаходження	-	-	+
Відновлення HTTP сторінки	-	-(показує контент трафіку індивідуально)	-(Показує лінки контенту трафіка)
Мережева комунікаційна матрична мапа	-	-	+
Оцінка трафіку критичного й некритичного для бізнесу	-	+	+(вбудована)
Можливість розробляти й налаштовувати розробникам	+	+	-(лише командою розробки)
UDP трафік	-	+	+

Colasoft пропонує покращену мережеву безпеку в порівнянні з Wireshark за рахунок надання звукових сповіщень та сповіщень електронною поштою.

Тим не менш, Colasoft підтримує лише 300 протоколів, що значно менше, ніж Wireshark, який підтримує 1100 протоколів. TCPDump - це економічно ефективний і дуже портативний пакетний інструмент для прослуховування, який займає всього 484 КБ пам'яті при установці. Хоча розмір інсталяційного файлу Wireshark починається з 18 МБ, після завершення інсталяції він займає 81 МБ в Windows і 449 МБ в операційних системах Linux. Для Colasoft, навпаки, потрібно лише 32 МБ. Таким чином, використання пам'яті Wireshark значно вище, ніж у Colasoft.

Оскільки Wireshark є інструментом з відкритим вихідним кодом, його легко може завантажити будь-хто, хто бажає переглянути його вихідний код і модифікувати його.

Така доступність дозволяє широкому колу розробників брати участь у вдосконаленні та налаштуванні цього інструменту, тоді як раніше цей процес обмежувався лише кількома розробниками, що входять до команди розробників Capsa компанії Colasoft.

Таким чином, Wireshark здобув репутацію ефективного рішення для перевірки пакетів для функцій програмування, що дозволяє задовольнити вимоги користувачів мережі до кастомізації без додаткових витрат.

В інструменті Colasoft користувачі повинні платити за конкретну конфігурацію для конкретного завдання моніторингу мережі.

З іншого боку, інструмент аналізу пакетів Wireshark корисний для отримання більшого досвіду в конфігуруванні TCP/IP шляхом розуміння структури мережі. Крім того, він працює на різних платформах, таких як Linux, Solaris, OS X і Windows.

Крім того, деякі дослідники в даний час працюють над підвищенням ефективності інструменту Wireshark. Зокрема, вони намагаються поліпшити

можливості інструменту перевірки пакетів Wireshark для виявлення типів атак типу "відмова в обслуговуванні" (DoS). Це особливо важливо для боротьби з атаками ping flooding, коли переважна кількість команд ping направляється на цільовий пристрій.

Всі три інструменти мають схожі мережеві властивості, проте кожен з них має свої конкурентні переваги.

Ми детально порівняли та оцінили якісні та кількісні параметри Wireshark, TCPDump та Colasoft.

Ці критерії включають кількість підтримуваних протоколів, відкритість вихідного коду, сумісність з платформами, використання бібліотеки libpcap, підтримку PCAP, дизайн інтерфейсу користувача, вартість, розшифровку шаблонів, виявлені аномальні пакети, підключення до мережі в матричній карті та реконструкцію TCP-потоку.

Кожен інструмент мережевого аналізатора не надає всіх параметрів мережі. Хоча інструмент Colasoft перевершує Wireshark у створенні матричних і графічних звітів, слід зазначити, що Wireshark має відкритий вихідний код, що дозволяє користувачам легко налаштовувати і розвивати його код відповідно до своїх конкретних вимог.

Крім того, Wireshark сумісний з різними платформами, включаючи Linux і операційні системи MS Windows, в той час як Colasoft працює тільки з операційними системами MS.

З іншого боку, TCPDump - це легкий інструмент, який займає мало місця в пам'яті, що робить його дуже конкурентоспроможним.

В результаті, це оптимальний вибір для віддаленого моніторингу мережі за допомогою інтерфейсу командного рядка.

Іншим важливим фактором, який слід враховувати в цьому параметрі, є кількість протоколів, які підтримує інструмент сніфферу пакетів.

Wireshark підтримує понад 1000 протоколів, що робить його цінним інструментом для моніторингу та контролю гетерогенних мереж з різними протоколами, включаючи ті, що використовуються для відео та аудіо додатків.

Для порівняння, інструмент Colasoft підтримує лише близько 300 протоколів, а TCPDump не підтримує протокол транспортного рівня User Datagram Protocol (UDP).

Беручи до уваги вартість, слід зазначити, що Wireshark і TCPDump є безкоштовними інструментами, тоді як Colasoft є дорогим варіантом, який перевищує вартість інших інструментів.

Незважаючи на це, Colasoft пропонує кращі можливості для виявлення аномальних протоколів, що дає йому конкурентну перевагу над такими інструментами, як Wireshark, які лише надають сповіщення.

Розроблений командою Capsa, Colasoft має чудовий графічний інтерфейс і численні функції безпеки.

Інтерфейс фільтрації вважається конкурентною особливістю інструменту Colasoft, який має зручний для користувача графічний інтерфейс, що полегшує просту фільтрацію та аналіз трафіку даних і протоколів.

Фундаментальною концепцією будь-якої мережі є семирівнева модель OSI.

Також відома як стек OSI, ця 7-рівнева архітектура побудови мережі була розроблена Міжнародною організацією зі стандартизації (ISO). Вона складається з двох різних моделей.

Горизонтальна модель, заснована на протоколах, полегшує комунікацію між додатками та процесами на різних машинах.

З іншого боку, вертикальна модель базується на послугах, що надаються сусідніми рівнями на одній машині.

У горизонтальній моделі обидва додатки потребують спільного протоколу

У вертикальних шарах, що прилягають один до одного, обмін даними здійснюється через інтерфейси.

Фізичний рівень - це найнижчий рівень, який безпосередньо передає дані.

Протоколи: Bluetooth, IRDA (інфрачервоний зв'язок), мідні дроти (вита пара, телефонна лінія), Wi-Fi тощо.

Канальний рівень забезпечує мережеву взаємодію на фізичному рівні.

Комутатори, концентратори та подібні пристрої є поширеними прикладами на цьому рівні. Представники протоколів включають Point-to-Point (PPP), стандарт прямого з'єднання комп'ютера з комп'ютером, Волоконно-оптичний розподілений інтерфейс даних , стандарт для передачі даних відстаню до 200 км.

Рисунок 2.5 – Модель OSI

2.2 Математична модель обраного методу аналізу трафіку

Для цього дослідження було обрано метод аналізу трафіку, який використовує навчання штучного інтелекту.

Рішення про застосування цього підходу зумовлене його ефективністю, точністю та можливістю розгортання в режимі реального часу.

Використання цієї методології призведе до швидкого виявлення вразливостей та покращення мережевого трафіку.

Основна перевага виявлення вторгнень у системах на основі AI - це швидкість. Це пов'язано з тим, що захист обчислювальних ресурсів вимагає

швидкого виявлення атак. Швидкість обробки даних має забезпечити швидку реакцію на вторгнення.

Ще однією перевагою є результат роботи у вигляді ймовірності, що полегшує прогнозування випадків виявлення вторгнень.

Ключова перевага використання методів штучного інтелекту полягає в його здатності "вивчати" ознаки атак і розпізнавати випадки, які не є типовими для попередніх спостережень.

Автоматизація виявлення аномалій дозволяє моделям ефективно виявляти аномальні патерни та поведінку, які можуть свідчити про кібератаки або внутрішні проблеми.

Крім того, системи зі штучним інтелектом можуть адаптуватися до нових загроз і змін у мережевому середовищі, навчаючись на льоту.

Це дозволяє ефективно використовувати ресурси.

Самооптимізація та можливості автоматичної корекції дозволяють ефективно використовувати системні ресурси.

Це призводить до спрощення прийняття рішень, надаючи операторам і мережевим адміністраторам чітку інформацію для швидкого та ефективного реагування на події.

Недоліки систем виявлення вторгнень на основі штучного інтелекту

По-перше, найважливішою проблемою, пов'язаною з використанням штучного інтелекту для систем виявлення вторгнень, є вибір навчальних даних.

Цей процес має вирішальне значення для успіху моделі.

По-друге, адаптивний аналіз даних нейронних мереж відрізняє їх від експертних систем, які використовують жорсткі, заздалегідь визначені правила для аналізу подій. Ваги зв'язків і передавальні функції вузлів мережі зазвичай заморожуються, як тільки мережа досягає прийнятного рівня успіху в ідентифікації подій.

Хоча мережевий аналіз дає достатню ймовірність успіху, підґрунтя для такого рівня точності є невизначеним.

Нейронна мережа, що використовується для аналізу мережевого трафіку, імітує структуру і функції людського мозку для виявлення закономірностей і вирішення таких завдань, як класифікація, виявлення аномалій і прогнозування.

Вхідні дані проходять через мережу, при цьому кожному нейрону присвоюються ваги, які визначають важливість інформації та створюють зв'язки з іншими нейронами.

Підсумовування ваг здійснюється для кожного нейрона, і функція активації визначає, чи виробляє нейрон вихідний сигнал.

Сигнал просувається через мережу до її останнього шару, вихідного, де нейрони приймають рішення про кінцевий результат аналізу трафіку, включаючи класифікацію або виявлення аномалій.

Результати роботи мережі оцінюються, вагові коефіцієнти коригуються під час навчання, а оновлення вносяться "на льоту", використовуючи свіжі дані для адаптації до змін у мережевому середовищі.

Функція нейронної мережі в аналізі мережевого трафіку полягає у виявленні складних закономірностей і зв'язків, які важко виявити звичайними методами.

Технічна термінологія повинна бути пояснена при першому використанні.

Кожен нейрон виконує роль обчислювального вузла, отримуючи і обробляючи дані від сусідніх нейронів, що в кінцевому підсумку дозволяє мережі виконувати такі завдання, як класифікація трафіку або виявлення аномалій.

Такий метод аналізу сприяє ефективному пристосуванню до змін в умовах роботи мережі та автоматичному виявленню і засвоєнню нових закономірностей, що суттєво розширює її можливості як інструменту для оцінки трафіку.

Нейронна мережа здатна адаптуватися до безлічі сценаріїв і точно виводити аномалії або класифікувати різні категорії мережевого трафіку, і все це без явного переліку.

Такий підхід до аналізу трафіку дозволяє виявити потенційні загрози та вразливості, а також пропонує значну гнучкість в обробці різних типів даних.

Нейронні мережі є цінним інструментом для захисту мереж і підвищення продуктивності за рахунок виявлення складних патернів і адаптації до різних умов.

Розглянемо сценарій, коли відбувається атака низької інтенсивності з накладанням типових мережевих подій та аномального трафіку.

Метод виявлення полягає у послідовному виділенні однорідних груп часового ряду (вхідних мережевих пакетів) за допомогою моделей розпізнавання образів.

Згодом для кожної виділеної групи будується модель прогнозування для виявлення сценарію атаки.

Як і всі методи машинного навчання, метод виявлення атак низької інтенсивності можна розділити на дві фази - фазу навчання і фазу класифікації.

Фаза навчання відповідає загальним принципам розробки моделей даних і продиктована виключно використовуваним методом навчання.

На етапі навчання будується класифікатор шляхом ітеративного налаштування його параметрів на навчальній вибірці.

Після цього отримана модель прогнозування часових рядів оцінюється на тестовому наборі, що складається з тестових прикладів.

Експерт повинен попередньо класифікувати як навчальні, так і тестові приклади хоча б частково.

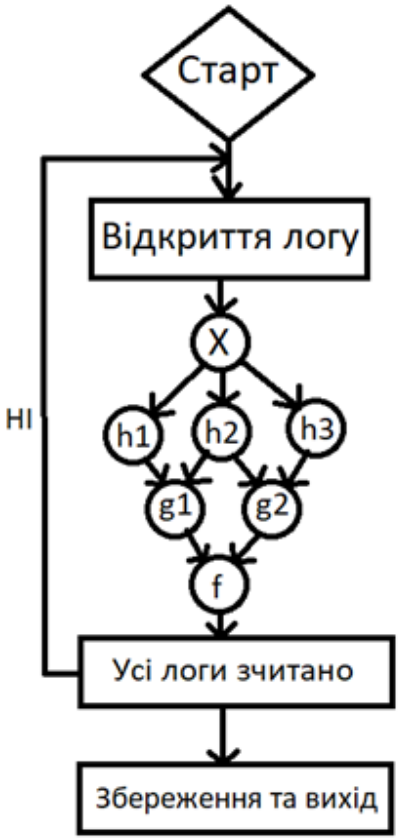


Рисунок 2.6 - Навчання в методі на основі штучного інтелекту.

Якщо навчений класифікатор дає очікуваний результат і відповідає критеріям класифікації, починається наступна фаза. Результатом етапу навчання є класифікатор з оптимізованими параметрами, здатний якісно класифікувати дані.

Метою етапу класифікації є обчислення міток класів для наборів даних, які раніше не були ідентифіковані за допомогою навченого класифікатора.

Результатом етапу класифікації є набір міток класів для раніше невідомих наборів даних.

Етапи методу можна сформулювати наступним чином:

– побудувати індивідуальну штучну нейронну мережу для кожної послуги (порту), що моніториться. Всі мережі працюють ідентично одна одній. Надалі розглядатитеметься виявлення атак на конкретний сервіс;

– захоплення певного набору мережевих пакетів з джерела даних для обраного сервісу, кількість яких залежить від значення розміру вікна;

– на етапі зменшення розмірності створюються вектори для карти, що самоорганізується;

– зменшується розмірність вхідних даних шляхом кластеризації векторів карти, що самоорганізується;

– створюємо вектори для багатошарового персептрона (MLP), кожна компонента вектора відповідає номеру кластера, до якого був розподілений пакет. Таким чином, вхідний вектор складається з набору мережевих пакетів, які були кластеризовані і зберігають інформацію про послідовність їх надходження в межах заданого вікна. Для цих пакетів заздалегідь визначена їхня класифікація до певного типу;

– вектори проходять аналіз за допомогою MLP, і набори, визначені на кроці 4 трафіку, класифікуються за категоріями. Таким чином, вони поділяються на два класи - або атака, або нормальне явище.

Запропонована модель являє собою хронологічно впорядковану послідовність подій, а саме - часовий ряд.

Властивості, що спостерігаються в атаці, охоплюють:

– порядок отримання пакетів;

– поля IP-заголовка;

– поля заголовка TCP HTTP;

– проміжки часу між надходженням пакетів.

– корисне навантаження протоколу HTTP;

– порядок надходження пакетів на вузол мережі;

– кількість пакетів, отриманих вузлом за певний проміжок часу

– кількість інформації, переданої в бітах за одиницю часу на вищезгаданий вузол

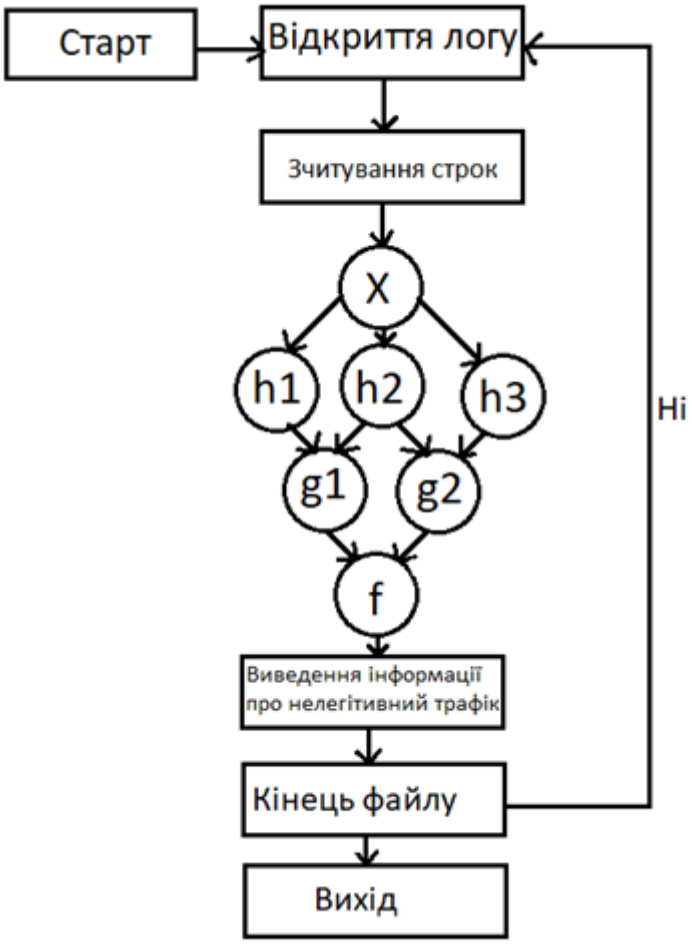


Рисунок 2.7 – Приклад роботи обраного методу.

2.3 Висновок

Вивчення та порівняння різних методів аналізу трафіку дозволило зробити кілька ключових висновків.

Було виявлено, що кожен з розглянутих методів має свої унікальні переваги та обмеження, залежно від конкретних умов і цілей аналізу.

Одним з важливих висновків є те, що метод з використанням штучного інтелекту є ефективним у виявленні нелегального трафіку. Це робить його привабливим для використання в сценаріях.

Також було виявлено, що метод на основі ШІ має ряд переваг, які роблять його ідеальним для цілей кваліфікаційної роботи. Однак його недоліки, які були

або будуть розглянуті в майбутній кваліфікаційній роботі, можуть обмежити його використання в [певних сценаріях].

З огляду на отримані результати, дослідження робить важливий внесок у розуміння сучасних методів аналізу трафіку. Крім того, враховуючи стрімкий розвиток технологій, слід звернути увагу на адаптацію методів до нових тенденцій в аналізі трафіку для забезпечення їх актуальності та ефективності в майбутньому.

3 МЕТОД АНАЛІЗУ ТРАФІКУ З МЕТОЮ ВИЯВЛЕННЯ АТАК НА КСЗІ

3.1 Опис запропонованого методу аналізу трафіку

3.1.1 PyBrain3 та алгоритми навчання

Для виконання завдання було використано мову програмування Python 3.7, бібліотеку PyBrain3 для побудови нейронної мережі, бібліотеку pickle для зберігання стану мережі, бібліотеку numpy для зручного маніпулювання даними та бібліотеку user_agents для отримання інформації про агента користувача.

PyBrain - сильний претендент на звання найкращої бібліотеки Python для отримання знань та реалізації широкого спектру алгоритмів, пов'язаних з нейронними мережами. Це чудовий приклад поєднання лаконічного синтаксису Python з майстерною реалізацією широкого спектру алгоритмів зі сфери машинного інтелекту.

PyBrain - це модульна бібліотека, створена для застосування різноманітних алгоритмів машинного навчання на мові Python.

Її основна мета - запропонувати дослідникам гнучкі та прості у використанні, але потужні інструменти для виконання завдань машинного навчання, а також для оцінки та порівняння продуктивності різних алгоритмів.

PyBrain розшифровується як Python-Based Reinforcement Learning, Artificial Intelligence and Neural Network Library - бібліотека для навчання з підкріпленням, штучного інтелекту та нейронних мереж.

Згідно з інформацією з веб-сайту, PyBrain - швейцарський армійський ніж у галузі нейро-мережевих обчислень

Сама бібліотека, будучи продуктом з відкритим вихідним кодом, доступна для використання в будь-якому проекті безкоштовно.

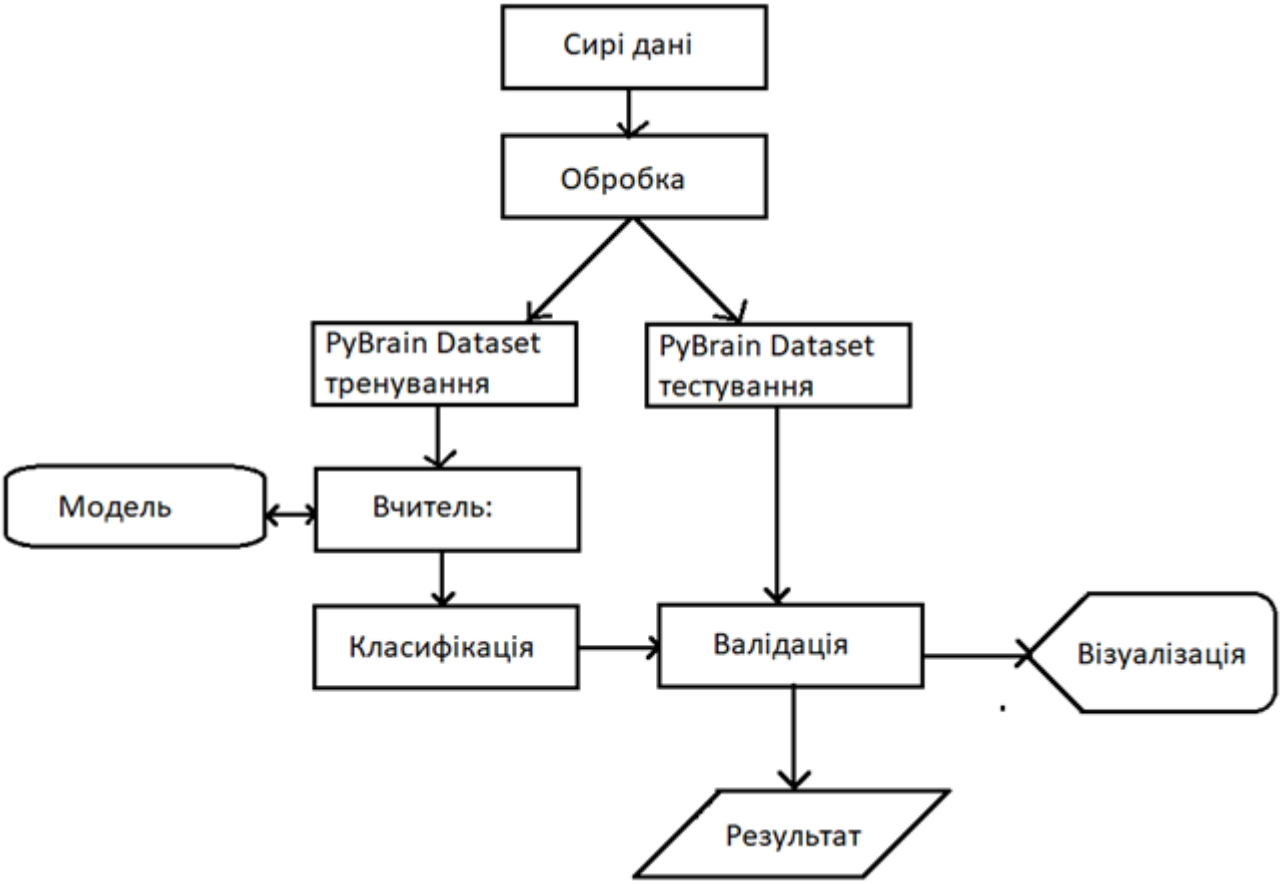


Рисунок 3.1 – Структура процедури використання

Алгоритмів навчання нейроної мережі є декілька :

– з вчителем;

– без вчителя;

– навчання з підкріпленням.

В кваліфікаційній роботі буде обрано алгоритм навчання з вчителем.

Їх також є декілька:

– метод зворотнього поширення (Back-Propagation) ;

– R-Prop (Resilient propagation) ;

– Support-Vector-Machines (інтерфейс до сторонньої бібліотеки LIBSVM) ;

– Evolino

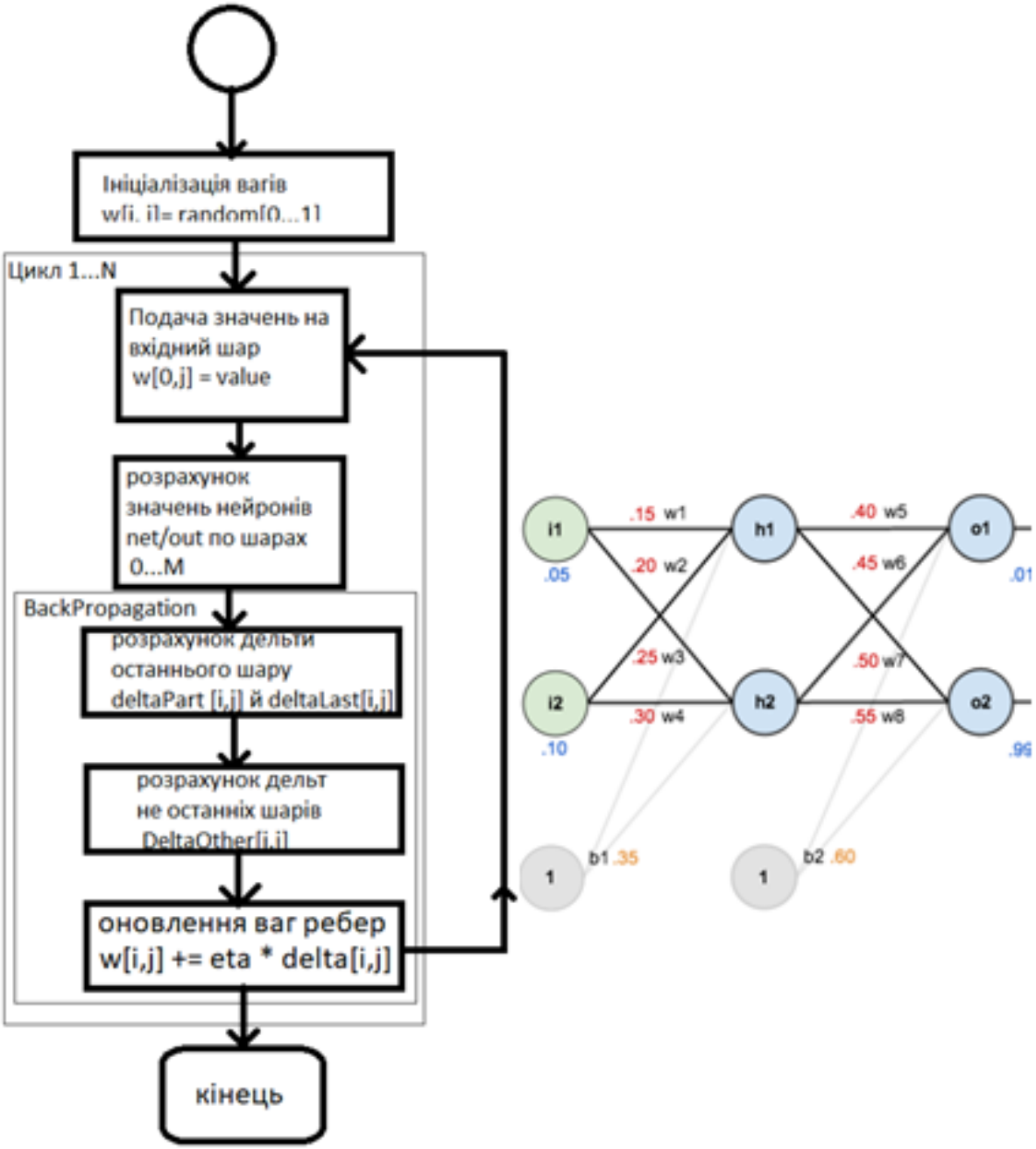


Рисунок 3.2 – Алгоритм навчання нейромережі

Алгоритм навчання з вчителем. Метод зворотнього поширення.

Це один з методів машинного навчання, в якому тестова система примусово навчається на прикладах "стимул-реакція".

Цей метод є формою кібернетичного експерименту. Хоча може існувати певна залежність між входом і виходом, вона до кінця не вивчена. Можна

використовувати лише кінцевий набір пар стимул-відповідь, відомий як навчальна вибірка.

На основі цієї інформації необхідно встановити кореляцію (створити прогнозну модель "стимул-реакція"). Це передбачає розробку алгоритму, здатного генерувати точну реакцію для будь-якого об'єкта. Для оцінки точності відповідей може бути реалізований функціонал якості, який також може бути застосований у навчанні на прикладах [56-57]. Ми зупинемось на методі зворотнього поширення (Back-Propagation). Методика обчислення градієнта, що використовується в процесі оновлення ваги багатошарового персептрона.

Фундаментальний принцип цього підходу полягає в поширенні сигналів помилок від виходів мережі до її входів, діючи в напрямку, протилежному звичайному поширенню сигналу.

Алгоритм розповсюдження помилки використовується на багатошаровому персептроні, який має певний набір входів. X1,....,Xn. набір виходів та внутрішніх вузлів.

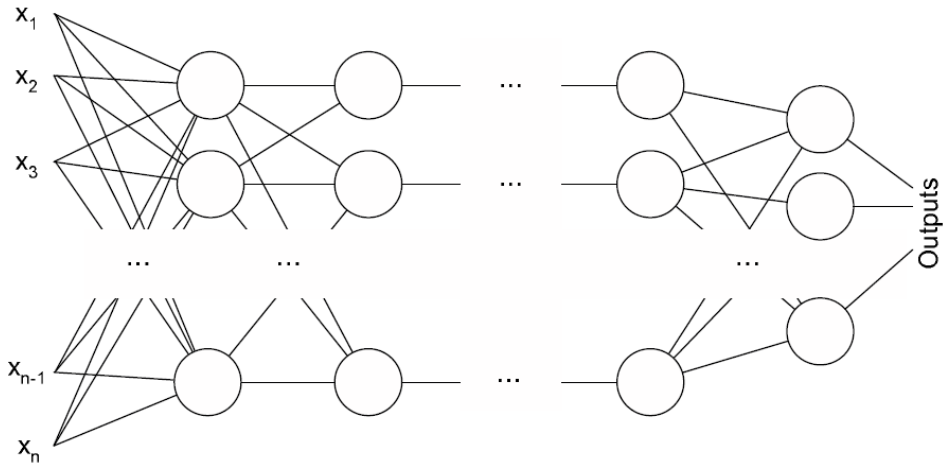


Рисунок 3.3 Багатошаровий персептрон

Всі вузли, включаючи входи та виходи, перенумеровано з використанням послідовної наскрізної системи нумерації від 1 до N, незалежно від топології шарів. Понзначимо через W_{ij} вагу на ребрі, що з'єднує і-ту та j-ту вершини, а вихід і-ї вершини позначимо через O_i. вихід і-го вузла.

Якщо ми маємо навчальний приклад (правильні відповіді мережі t_k k ∈ Outputs), то функція похибки, отримана за допомогою методу найменших квадратів, має такий вигляд.

$$E(\{w_{i,j}\}) = \frac{1}{2} \sum_{k \in \text{Outputs}} (t_k - o_k)^2 \tag{3.1}$$

На практиці мережі з простою структурою з двох рівнів нейронів, а саме прихованих та вихідних, демонструють відмінні результати.

Кожен вхід мережі з'єднується з усіма прихованими нейронами, а вихід кожного прихованого нейрона подається на вхід кожного вихідного нейрона. Достатньо лише подати кількість прихованих нейронів у цьому сценарії.

3.1.2 Алгоритм: BackPropagation:

$$(\eta, \alpha, \{x_i^d, t^d\}_{i=1,d=1}^{n,m}, \text{steps}) \tag{3.2}$$

1. Ініціалізувати $\{w_{ij}\}_{ij}$ маленькими випадковими значеннями
2. Повторити NUMBER_OF_STEPS раз: для всіх d від 1 до m:

2.1 Подати $\{X^d_i]$На вхід мережі і порахувати вихід o_i кожного вузла.

(3.3)

$$\{\Delta w_{ij}\}_{i,j} = 0$$

2.2 Для всіх k ∈ Outputs

$$\delta_k = -o_k(1 - o_k)(t_k - o_k) \tag{3.4}$$

2.3 Для кожного рівня l, починаючи з передостаннього, для кожної вершини j рівня l обчислити

$$\delta_j = o_j(1 - o_j) \sum_{k \in Children(j)} \delta_k w_{j,k}. \quad (3.5)$$

2.4 Для кожного ребра мережі {i, j}

3 Видати значення W_{ij}

$$\Delta w_{i,j}(n) = \alpha \Delta w_{i,j}(n-1) + (1-\alpha)\eta \delta_j o_i.$$
$$w_{i,j}(n) = w_{i,j}(n-1) - \Delta w_{i,j}(n). \quad (3.6)$$

Де а - коефіцієнт інерції, який використовується для згладжування різких стрибків при русі вздовж поверхні цільової функції.

Незважаючи на багато успішних застосувань, рекурентне поширення не є універсальним рішенням через його сумнозвісний тривалий процес навчання.

У складних випадках навчання мережі може зайняти кілька днів або навіть тижнів, і воно може бути зовсім не успішним.

Існує кілька потенційних причин для цього.

– Параліч мережі;

Під час навчання мережі значення ваг можуть значно збільшуватися в результаті корекцій. Це може призвести до того, що більшість або всі нейрони будуть працювати з дуже високими вихідними значеннями в області, де похідна функції стиснення мінімальна.

– Локальні мінімуми.

Зворотне поширення використовує форму градієнтного спуску, коли він спускається вздовж поверхні помилки, постійно підлаштовуючи ваги в напрямку мінімуму.

Поверхня похибки складної мережі має значні поглиблення, пагорби, долини, складки та балки у високовимірному просторі, що може призвести до того, що мережа натрапить на локальний мінімум, неглибоку долину, коли поблизу існує набагато глибший мінімум.

У точці локального мінімуму всі напрямки ведуть вгору, і мережа не може його уникнути.

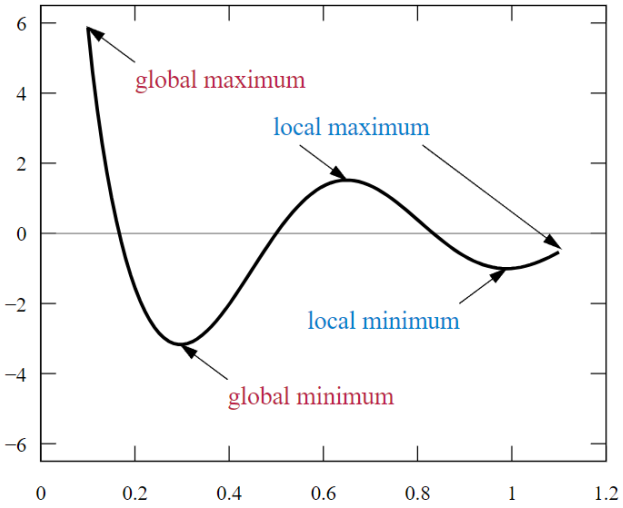


Рисунок 3.4 – Демонстрація локального мінімума

Основною проблемою при навчанні нейронних мереж є пошук шляхів уникнення локальних мінімумів: кожного разу, коли залишається локальний мінімум, наступний знаходиться знову, використовуючи той самий метод розповсюдження помилки, доки не стане неможливим уникнути його.

Якщо розмір фіксованого кроку дуже малий, збіжність буде відбуватися занадто повільно.

З іншого боку, якщо фіксований розмір кроку занадто великий, це може призвести до паралічу або стійкої нестабільності.

Крім того, важливо відзначити можливість надмірної адаптації, яка в основному пов'язана з помилковим дизайном топології та/або неправильним вибором критерію зупинки навчання.

Важливо зазначити, що при перенавчанні мережа втрачає здатність до узагальнення інформації.

Мережа вивчить весь набір навчальних зображень, але може неправильно ідентифікувати будь-які інші зображення, навіть якщо вони дуже схожі.

3.2 Обробка даних в мережі

PyBrain працює з використанням мережевих структур, які можуть побудувати практично всі складні алгоритми, що підтримуються бібліотекою. Наведемо кілька ілюстративних прикладів:

- мережі прямого поширення, які складаються з мереж Deep Belief Networks і Restricted Boltzmann Machines (RBM) ;
- рекурентні нейронні мережі (RNN), які використовують архітектуру довготривалої короткочасної пам'яті (LSTM) ;
- багатовимірні рекурентні мережі (MDRNN) ;
- мережамі Кохонена / Self-Organizing Maps;
- Reservoirs;
- нейронна мережа Коско / Двонаправлені мережі;
- створення топологій власної структури.

Оперування данимим (Building a DataSet)

Створена мережа необхідна для обробки даних, що є основною темою цього розділу.

Звичайний набір даних складається з вхідних та вихідних значень. Для цього PyBrain використовує модуль pybrain.dataset з додаванням класу SupervisedDataSet.

Для розробки класифікатора нам потрібні два набори даних, що містять "хороші" та "погані" запити.

Це може здатися кумедним, але для того, щоб виявити ботів, ми повинні спочатку ідентифікувати їх. Для цього ми скористаємося командою grep, щоб витягти з журналу запити, зроблені з IP-адрес з численними 503 помилками (обмеження швидкості nginx).

Таким чином, у нас повинно бути в цілому 3 access.log'a:

– набір даних, що містить "хороші" запити з access.log до початку DDoS-атаки;

– набір даних також включає "погані" запити, зібрані на початковому етапі;

– набір даних необхідно класифікувати, розрізняючи "хороші" і "погані" запити. Як правило, це передбачає моніторинг access.log з сервера, що зазнав DDoS-атаки, за допомогою команди tail -f.

Далі слід ознайомитися з розбором так званого комбінованого логу nginx, зображено на рисунку 3.5.

```
(?P<ip>[0-9.:a-f]+)   [^   ]+   [^   ]+   \[.*\]   "(?P<url>.*)"
(?P<code>[0-9]+)        (?P<size>[0-9]+)        "(?P<refer>.*)"
"(?P<useragent>.*)"$
```

Рисунок 3.5 – Nginx

Після того, як ви навчилися аналізувати лог, необхідно вибрати релевантні ознаки (маркери/характеристики), створити їх словник і використовувати його для генерації вектора ознак для кожного запису набору даних. Більш детальна інформація буде надана пізніше.

Ми відібрали ознаки з журналів, вивчаючи як "хороші", так і "погані" журнали та визначаючи різні ознаки, або атрибути. Ми знайшли обмежену кількість ознак в об'єднаному журналі, і нижче наведено ті, які ми виділили:

– запит парситься на тип запиту (наприклад, HEAD, GET, POST), url і http_version. Потім URL-адреса розбирається на протокол, ім'я хоста, шлях і всі ключі з рядка query_string;

– Referer також парситься аналогічно до url;

– парсинг User-Agent може бути дещо складнішим, оскільки його формат відрізняється в різних браузерах. Відповідно до RFC2616, існує обмежена настанова з цього питання. Для досягнення кращих результатів необхідне подальше вдосконалення;

– :status Розглядатимуться лише випадки з кодами помилок 503, 404 або 403. Зазвичай під час DDoS-атаки сервер, як правило, відповідає з кодами помилок 500 або 502. Отже, ми досягли етапу, коли у нас є великий список усіх можливих ознак, які можуть з'явитися в запиті.

Ця компіляція слугує нашим словником, і вона необхідна для побудови вектора ознак для будь-якого можливого запиту. Вектор ознак, бінарний M-вимірний вектор (де M означає довжину словника), вказує на наявність або відсутність кожної ознаки словника в запиті.

Хеш-таблиця є вигідним вибором для структури даних словника через часті звернення до неї, наприклад, для перевірки наявності слова у словнику.

Приклад побудови словника та вектора ознак. Припустимо, що наша нейронна мережа навчена на двох прикладах, одному позитивному і одному негативному, ми можемо перейти до її тестування на прикладі журналу.

Тут ми записуємо дані з "негативного" логу, зображено на рисунку 3.6.

```
0.0.0.0 - - [10/Dec/2023:20:00:08 +0400] "POST /forum/index.php
HTTP/1.1" 503 107 "http://www.mozilla-europe.org/" "-"
```

Рисунок 3.6 – «Негативний» лог

```
0.0.0.0 - - [10/Dec/2023:15:00:03 +0400] "GET
/forum/rss.php?topic=347425 HTTP/1.0" 200 1685 "-" "Mozilla/5.0
(Windows; U; Windows NT 5.1; pl; rv:1.9) Gecko/2008052906
Firefox/3.0"
```

Рисунок 3.7 – «Позитивний» лог

```
['__UA___OS_U', '__UA_EMPTY', '__REQ___METHOD_POST',
'__REQ___HTTP_VER_HTTP/1.0', '__REQ___URL___NETLOC_',
'__REQ___URL___PATH_/forum/rss.php',
'__REQ___URL___PATH_/forum/index.php', '__REQ___URL___SCHEME_',
'__REQ___HTTP_VER_HTTP/1.1', '__UA___VER_Firefox/3.0',
'__REFER___NETLOC_www.mozilla-europe.org', '__UA___OS_Windows',
'__UA___BASE_Mozilla/5.0', '__CODE_503', '__UA___OS_pl',
'__REFER___PATH_/', '__REFER___SCHEME_http', '__NO_REFER__',
'__REQ___METHOD_GET', '__UA___OS_Windows NT 5.1',
'__UA___OS_rv:1.9', '__REQ___URL___QS_topic',
'__UA___VER_Gecko/2008052906']
```

Рисунок 3.8 – Отриманий словник

Запис тесту:

```
0.0.0.0 - - [10/Dec/2023:20:00:01 +0400] "GET
/forum/viewtopic.php?t=425550 HTTP/1.1" 502 107 "-"
"BTWebClient/3000(25824)"
```

Рисунок 3.9 – Тестовий запис

Його feature-vector зображено на рисунку 10.

```
[False, False, False, False, True, False, False, True,
True, False, False, False, False, False, False, False, False,
True, True, False, False, False, False]
```

Рисунок 3.10 – feature-vector

Зверніть увагу, наскільки розрідженим є вектор ознак; така поведінка спостерігатиметься для всіх запитів.

Розбиття набору даних. найкращою практикою є поділ набору даних на кілька частин. Я розділив його на дві частини у співвідношенні 70/30.

Training set Це місце, де навчається нейронна мережа.

Test set. використовується для оцінки роботи нейронної мережі.

Такий поділ пов'язаний з тим, що нейронна мережа з найменшою помилкою навчання буде давати більшу помилку на нових даних, тому що ми "перенавчили" мережу, заточивши її під навчальний набір.

Надалі, якщо нам потрібно буде підібрати оптимальні константи, набір даних слід розділити на 3 частини у співвідношенні 60/20/20: Навчальний набір, Тестовий набір і Перехресна перевірка.

Остання буде використана для підбору оптимальних параметрів нейронної мережі (наприклад, вагових коефіцієнтів).

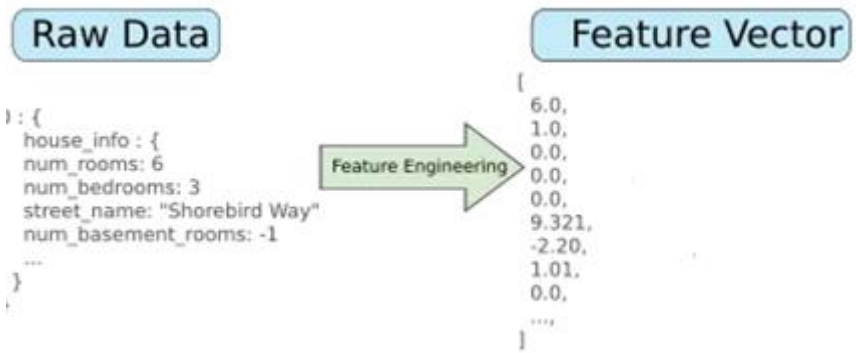


Рисунок 3.11 – Features

Features - це вибрані або оброблені дані, які використовуються як вхідні дані для алгоритмів, як правило, алгоритмів машинного навчання.

Важливо зазначити, що ознаки повинні бути об'єктивними і вільними від суб'єктивних оцінок, а технічні абревіатури слід пояснювати при пер шому використанні.

Методи, що використовуються для перетворення тексту в ознаки, можна описати як методи "векторизації тексту", оскільки вони служать єдиній меті - перетворити текст у вектори (або масиви, для простоти; або тензори, для складності), які потім можуть бути використані для живлення класичних моделей машинного навчання.

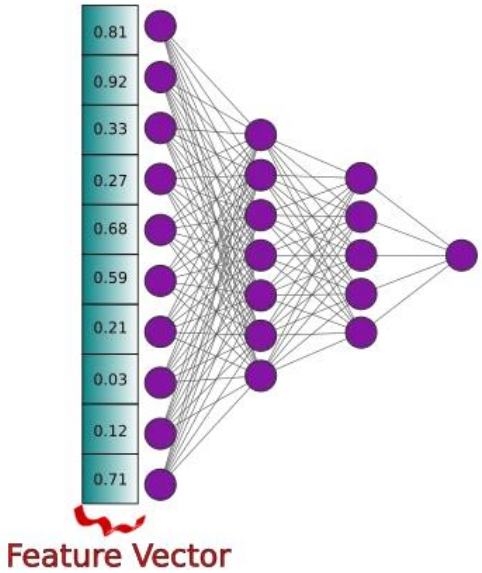


Рисунок 3.12 – Вектор ознак

Думаючи про наш бажаний результат як про вектор, корисно уявити, як текст може бути перетворений на функції.

feature-vector у контексті навчання нейронних мереж термін "вектор ознак" означає вектор числових значень, який відображає вхідні дані для нейронної мережі.

"Особливості" або "характеристики" позначають кожен елемент цього вектора, що відображає певну характеристику вхідного сигналу або даних.

Багато завдань машинного навчання, в яких використовуються нейронні мережі, охоплюють вхідні дані, що мають кілька різних ознак або характеристик.

Наприклад, при навчанні нейронної мережі розпізнаванню зображень обличчя кожен піксель можна розглядати як окрему особливість. Таким чином, повний набір пікселів формує вектор ознак для цієї задачі.

Вектор ознак допомагає нейромережі зрозуміти структуру та характеристики вхідних даних. Під час навчання моделі вектор ознак визначає вагу окремих ознак для досягнення оптимального відображення вхідних даних у вихідні.

Ілюстрація вектора ознак у задачі класифікації тексту виглядає наступним чином:

Feature vector =[$x1,x2,\ldots,xn$] де x1,x2,…, xn - числові значення ознак, які відповідають окремим характеристикам тексту.

Використання правильного та репрезентативного вектора ознак є вирішальним аспектом для успішного навчання нейронних мереж.

Навчання:

– спочатку дані проходять підготовку, і кожному екземпляру даних, наприклад, зображенню або тексту, присвоюється вектор ознак;

– потім нейронна мережа навчається на цих векторах ознак, і алгоритм оптимізує ваги нейронів для точної відповідності вхідним даним;

– у процесі навчання використовується функція втрат, щоб виміряти розбіжність між помітними відповідями нейронної мережі та очікуваними відповідями.

Застосування:

– нові дані перетворюються у вектор ознак за тими ж правилами, що і під час навчання;

– модель використовує вектор ознак для прогнозування або класифікації залежно від типу завдання;

– нейромережа видає результат, який може мати форму ймовірностей (у випадку класифікації) або числових значень (у випадку регресії).

У деяких випадках необхідно використовувати додаткові методи для обробки векторів ознак, зокрема нормалізацію, щоб гарантувати однакове масштабування всіх ознак, вибір ознак для вилучення найбільш релевантних ознак або створення нових ознак на основі переважаючих ознак.

Ці методи підвищують компетентність нейронної мережі, тим самим покращуючи її ефективність в обробці автентичних даних.

3.3 Висновок

Розглянуто та детально визначено інноваційний підхід до аналізу трафіку. Запропонований метод базується на нейронній мережі і має кілька ключових переваг. По-перше, він забезпечує високу продуктивність, що робить його ефективним інструментом для виявлення та аналізу основних закономірностей трафіку.

Слід зазначити, що використання цього методу може значно підвищити якість аналізу трафіку і допомогти захиститися від CSIS-атак. Його гнучкість і можливості роблять його цінним інструментом для цієї мети. Крім того, на основі проведених експериментів та порівняльного аналізу з існуючими методами можна стверджувати, що запропонований метод виявиться найкращим.

Таким чином, виходячи з вищесказаного, можна зробити висновок, що запропонований метод аналізу трафіку є перспективним і відкриває широкі можливості для виявлення атак на КЗЗ. Його впровадження може призвести до значного підвищення ефективності досліджень та зробити цінний внесок у розвиток сучасних технологій кібербезпеки.

4 РЕАЛІЗАЦІЯ РОЗРОБЛЕНОГО МЕТОДУ

4.1 Середовище розробки

Для виконання поставленої задачі було взято мову програмування Python 3.7, бібліотеку для побудови нейронної мережі PyBrain3

Бібліотеку для збереження стану мережі pickle, для легкої роботи з даними бібліотеку numpy, і для отримання інформації по User-Agent користувача бібліотеку user_agents.

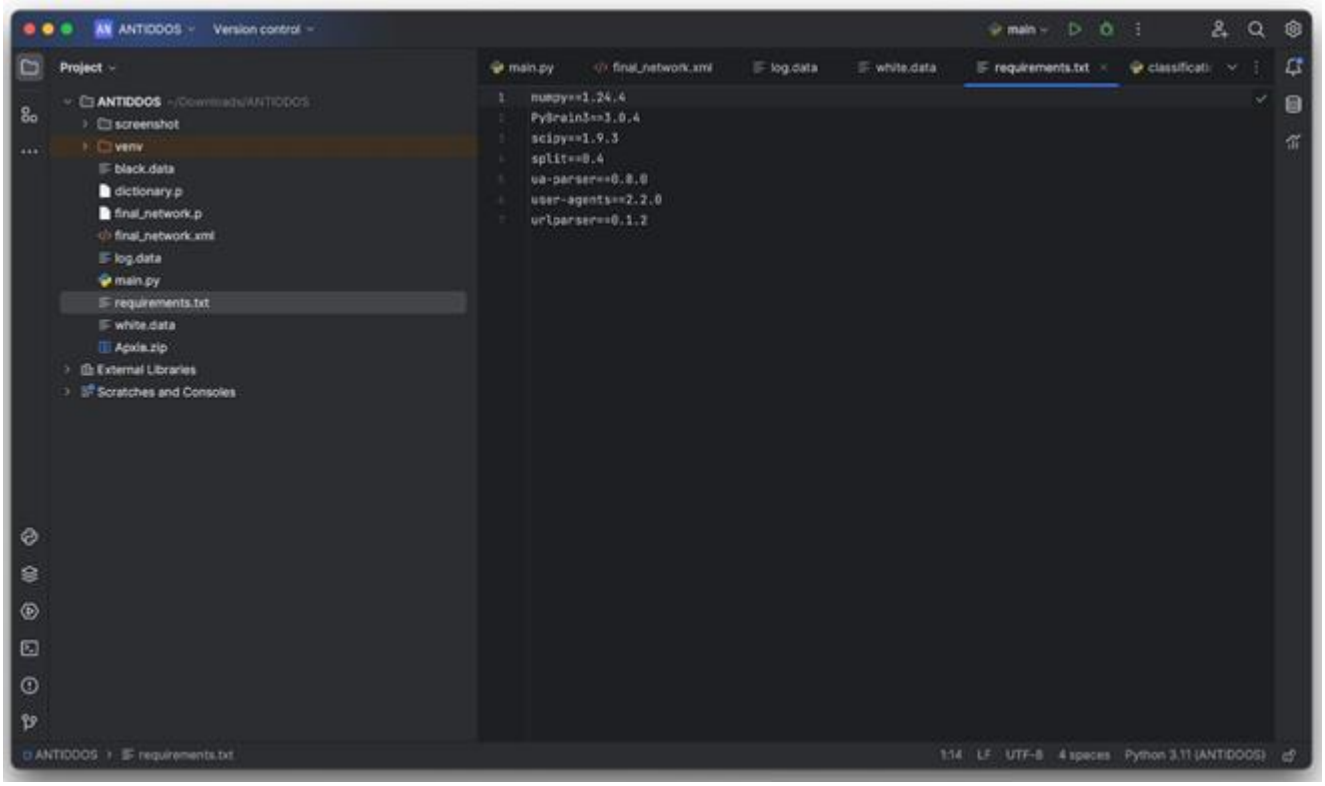


Рисунок 4.1 – Використані бібліотеки

Мова програмування Python - це загальноприйнята мова високого рівня, яку можна інтерпретувати. Найбільш помітною особливістю Python є його чіткий та зрозумілий синтаксис, що дозволяє ефективно розробляти програми та покращує їхню зрозумілість.

Python є інтерпретованою мовою, тобто програми можуть виконуватися без попередньої компіляції. Вона містить вбудовані високорівневі структури даних, включаючи списки, словники, кортежі та інші.

Вона полегшує об'єктно-орієнтоване програмування, використовуючи концепції об'єктів і класів.

Він пропонує широкий спектр попередньо встановлених функцій і модулів, які спрощують взаємодію з різними елементами програмування, включаючи операційну систему, мережу і бази даних. Вона також полегшує застосування різноманітних парадигм програмування, серед яких імперативне, функціональне та аспектно-орієнтоване програмування.

Python легко сумісний з іншими мовами програмування, такими як C та C++. Це дозволяє використовувати їхні бібліотеки та компоненти. Крім того, існує велика і зацікавлена спільнота розробників, яка активно сприяє розробці, підтримці та поширенню різних бібліотек і фреймворків.

Python є універсальною мовою, яка використовується для розробки широкого спектру додатків, включаючи веб-розробку, обробку даних, наукові обчислення, штучний інтелект тощо.

PyCharm - це IDE для розробки на мові Python. JetBrains розробила PyCharm, щоб зробити розробку та налагодження коду простішим та ефективнішим. Крім того, він пропонує інші банальні функції програмування.

PyCharm пропонує різноманітні функції, такі як редагування та налагодження коду, завершення коду, підтримка аналізу та інтеграція з фреймворками Python, системами контролю версій та віртуальними середовищами. Розробники можуть використовувати PyCharm для ефективного та швидкого написання, тестування та налагодження коду.

Це середовище розробки програмного забезпечення включає інтелектуальне автозавершення коду, вбудований відладчик для налагодження програм, а також підтримку систем контролю версій, рефакторингу коду, аналізу коду для виявлення помилок і різних фреймворків, таких як Django, Flask та інші.

PyCharm - популярний інструмент серед британських розробників Python для ефективного створення та управління програмними проектами. Він надає потужні функції, спеціально розроблені для мови програмування Python.

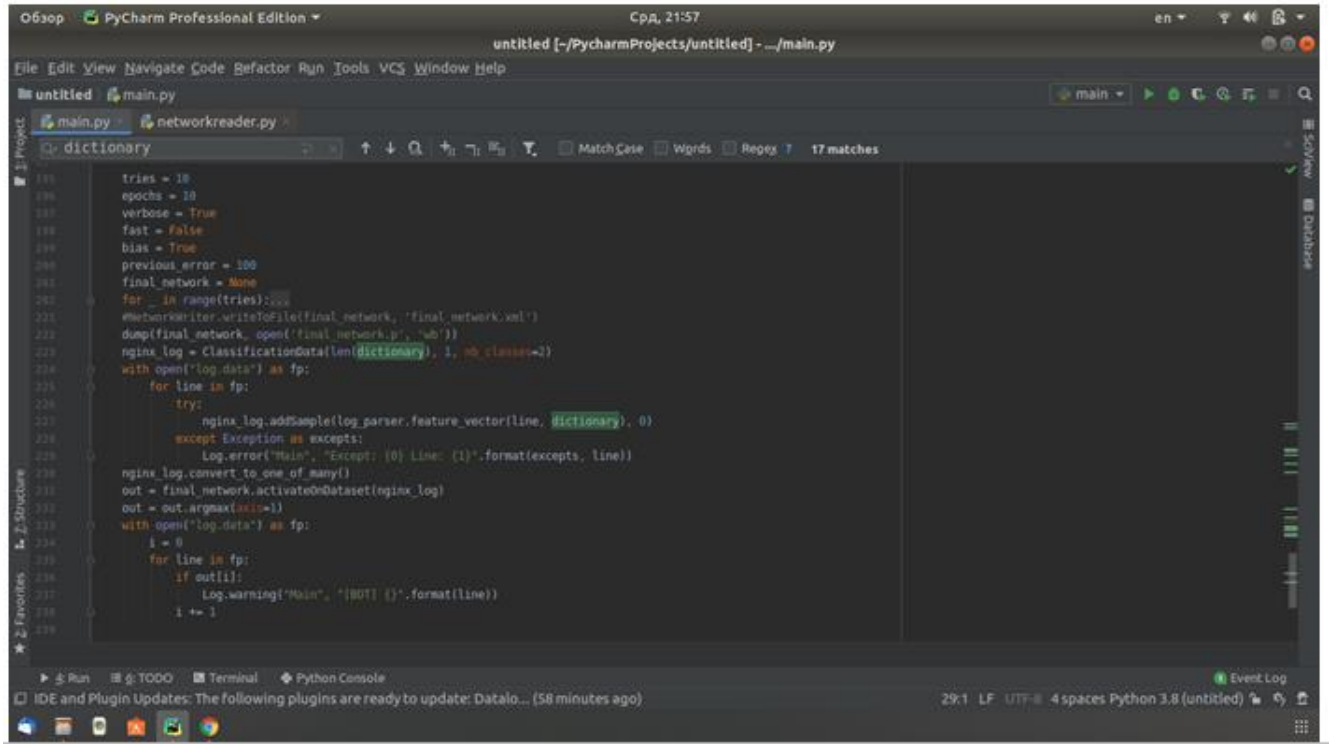


Рисунок 4.2 – Знімок екрану з додатку PyCharm з частиною коду на мові Python.

PyBrain - це зразкова бібліотека Python для отримання та реалізації численних нейромережевих алгоритмів. PyBrain - це модульна бібліотека, призначена для реалізації різноманітних алгоритмів машинного навчання на мові Python. Вона демонструє поєднання лаконічного синтаксису Python і відмінну реалізацію різноманітних алгоритмів з області машинного інтелекту.

PyBrain - це модульна бібліотека, призначена для реалізації різноманітних алгоритмів машинного навчання на мові Python. Її основна мета - надати досліднику гнучкі, прості у використанні, але в той же час потужні інструменти для реалізації завдань машинного навчання, тестування і порівняння ефективності різних алгоритмів.

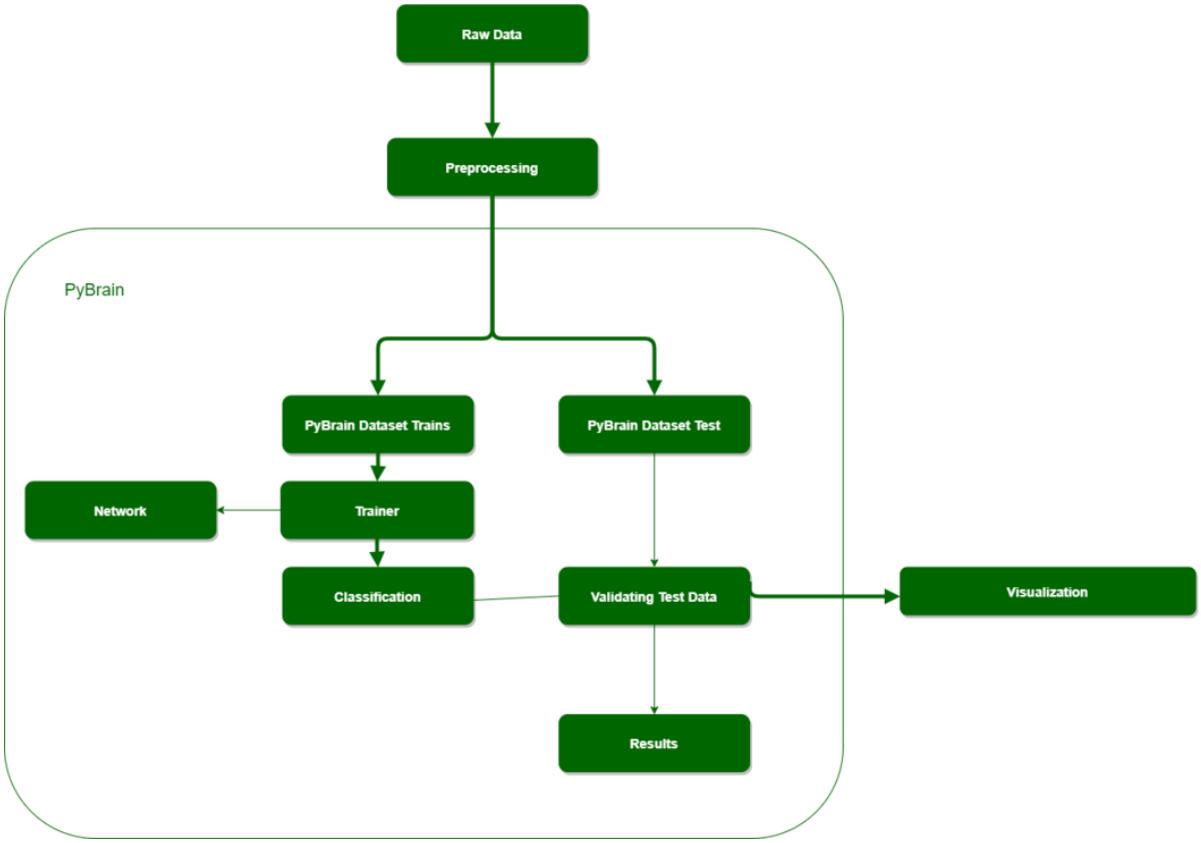


Рисунок 4.3 – Робочий процес PyBrain

Процес починається з сирих даних, за якими слідує попередня обробка. Потім дані групуються для навчання, і після підготовки набору даних тренером створюється мережа для тестування і навчання.

Дані спочатку навчаються в мережі тренером, після чого вихідні дані класифікуються на помилки навчання та помилки перевірки.

Потужний і зручний пакет для машинного навчання, наповнений безліччю функцій, був приємним інструментом для роботи.

Він особливо корисний для початківців у машинному навчанні. Крім того, цей пакет легко інтегрується з іншими бібліотеками Python, такими як Matplotlib або Pyplot, забезпечуючи ефективну візуалізацію даних.

Симулятори PyBrain також полегшують отримання даних для навчання та тестування моделей.

Це можна переглянути в Python за допомогою таких бібліотек, як matplotlib або pyplot. Останнім кроком є перевірка вихідних даних, щоб підтвердити, чи відповідають вони навченим даним.

Модуль pickle реалізує двійкові протоколи для серіалізації та десеріалізації структури об'єктів Python. "pickling" - це процес перетворення ієрархії об'єктів Python у потік байт, тоді як "unpickling" - це зворотна операція, коли потік байт (з двійкового файлу або байт-подібного об'єкта) перетворюється назад в ієрархію об'єктів.

Цей процес є важливим для ефективної передачі та зберігання даних. Перетравлення (і розбиття) також відоме як "серіалізація", "маршалінг "1 або "редукція"; однак, щоб уникнути непорозумінь, у цьому випадку використовуються терміни "розбиття" і "розбиття".

NumPy - це модуль Python з відкритим вихідним кодом, який пропонує скомпільовані, швидкі функції для поширених математичних та числових операцій. Вони згруповані у високорівневі пакети, які забезпечують функціональність, подібну до MatLab. NumPy (Numeric Python) надає основні методи для маніпулювання великими масивами та матрицями.

Модуль user_agents пропонує простий підхід до виявлення пристроїв, включаючи мобільні телефони та планшети, шляхом розбору рядків агентів користувача в браузері або HTTP. Його завдання полягає в тому, щоб надійно визначити назву та версію браузера користувача, версію операційної версію операційної системи, а також визначити, чи є користувацький агент мобільним пристроєм, планшетом або комп'ютером мобільний пристрій, планшет або ПК із сенсорним екраном.

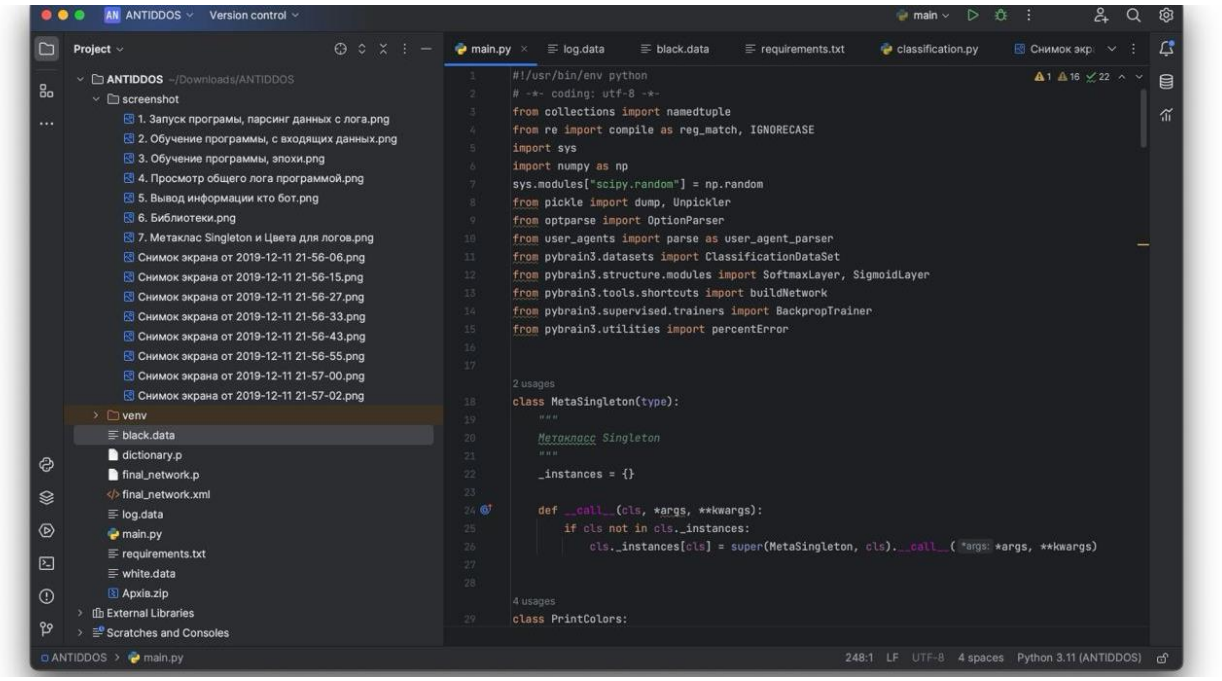


Рисунок 4.4 Бібліотеки які використовувались в проекті

4.2 Реалізація програмного забезпечення

Для створення моделі навчання, було створено метод, який переводить інформацію про log в програмне представлення, для подальшого переведення його в інформаційний потік для нейронної мережі.

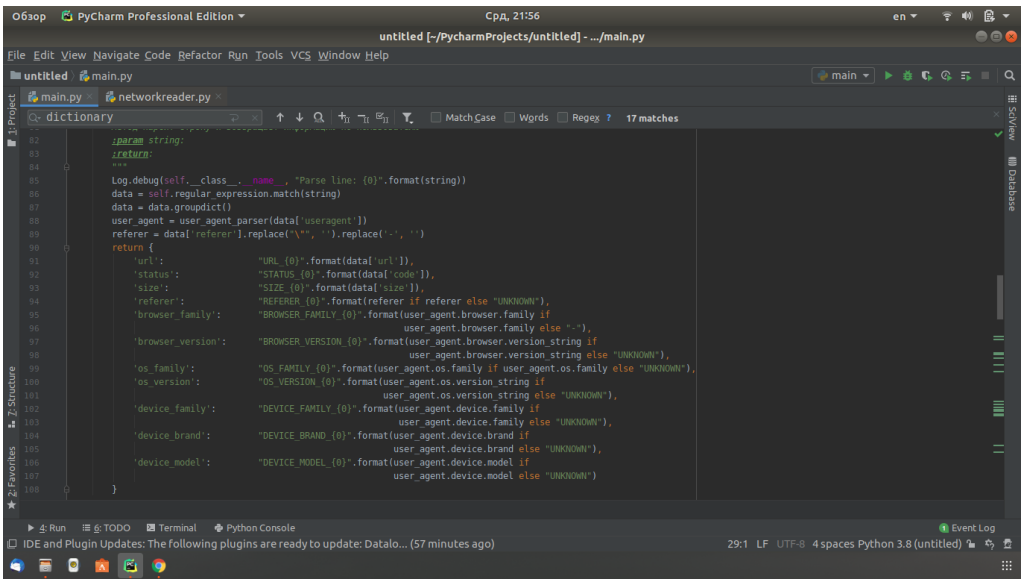


Рисунок 4.5 - Переведення логу в модель

Для запуску програмного забезпечення для парсингу логів необхідно в консоль ввести команду «python3.py main -m start»

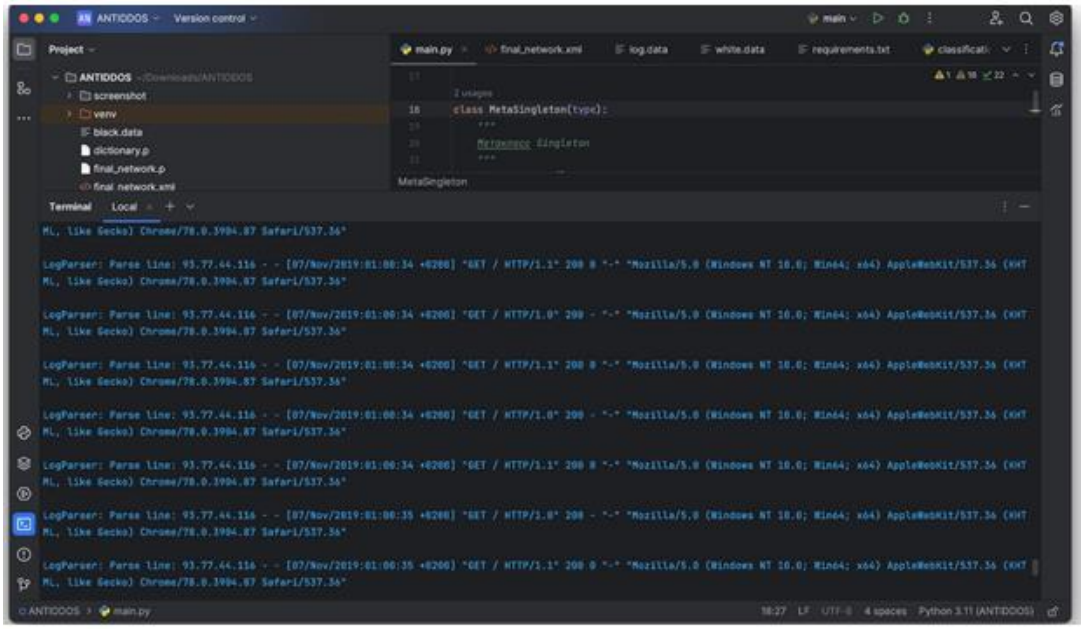


Рисунок 4.6 – Запуск програмного забезпечення і процес зчитування логів

Парсинг логів важливий для виявлення проблем, моніторингу систем, аналізу активності та усунення проблем безпеки; цей процес дозволяє витягувати зрозумілі дані з логів і використовувати їх для моніторингу, аналізу або вирішення проблем.

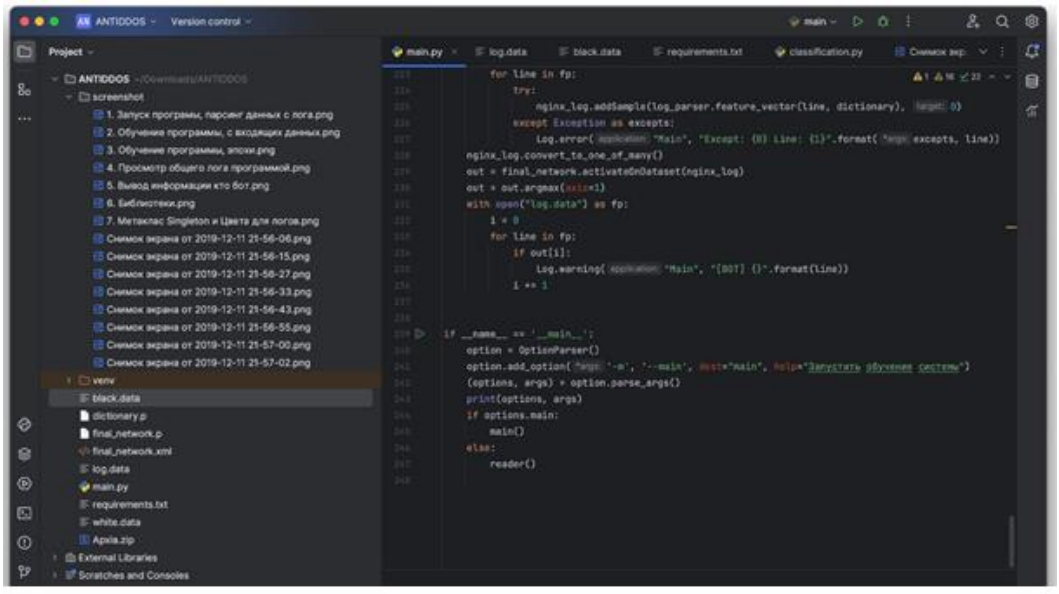


Рисунок 4.7 - Вхідна точка для запуску програми

Після зчитування усіх логів заданих для навчання, програмне забезпечення розпочинає процес навчання, який розділений на 10 епох.

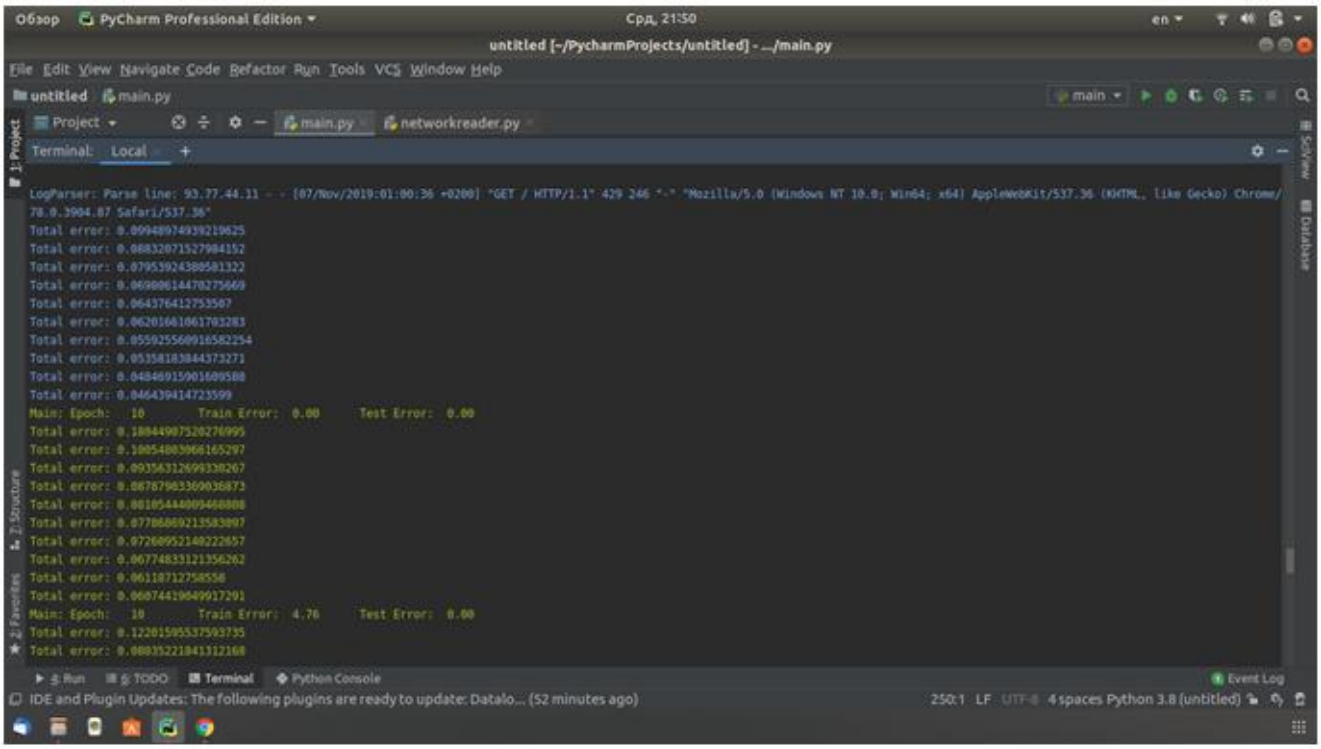


Рисунок 4.8 – Процес навчання з вхідних данних.

Навчання, кероване вхідною інформацією, - це процес, в якому система або модель вдосконалюється шляхом аналізу та обробки вхідної інформації.

Процес навчання, розділений на 10 епох, передбачає багаторазові ітерації навчання моделі на всьому наборі даних.

У машинному навчанні епоха - це одноразове проходження всіх прикладів з навчальної вибірки через модель.

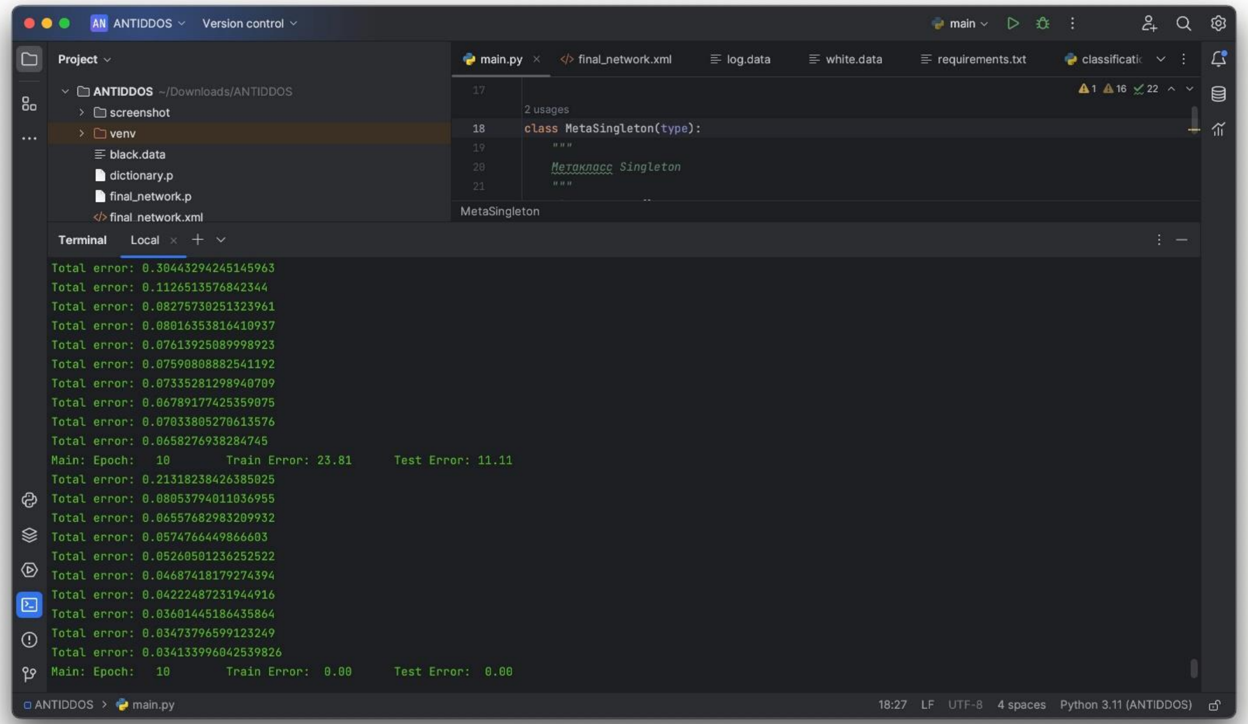


Рисунок 4.9 – Процес тренування системи

Після навчання, програмне забезпечення переходить у режим зчитування усіх логів, у яких є всі користувачі.

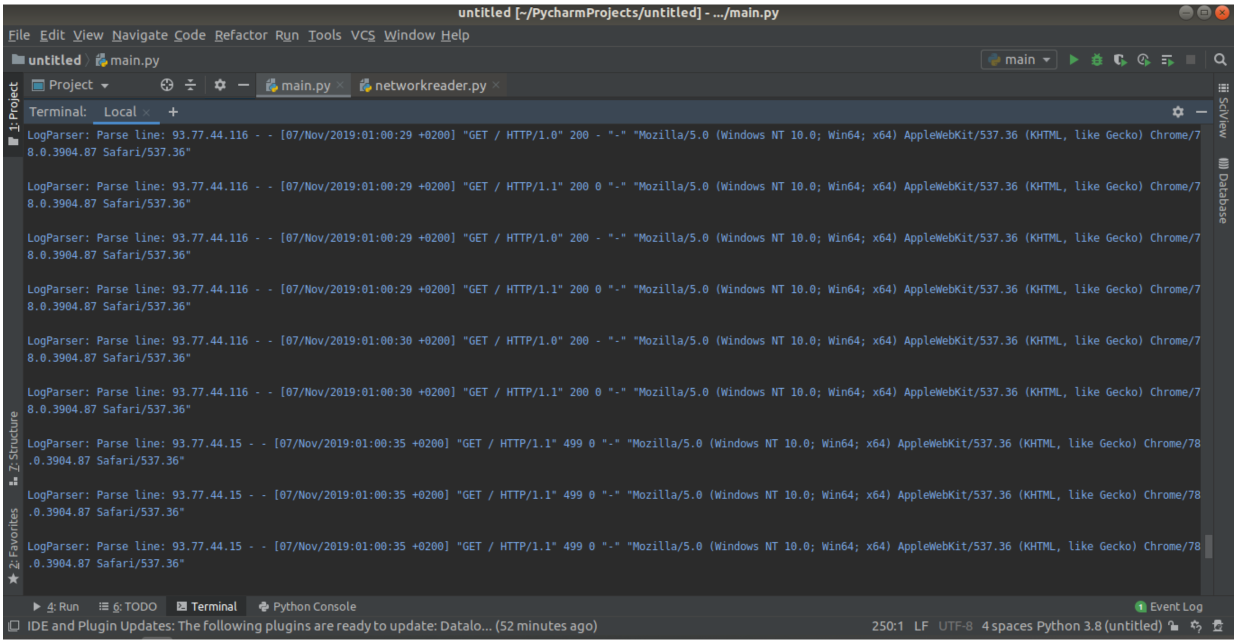


Рисунок 4.10 – Зчитування загального логу

Після того, як усі логи було просканировано, нейронна мережа виводить знайдений нелегітивний трафік.

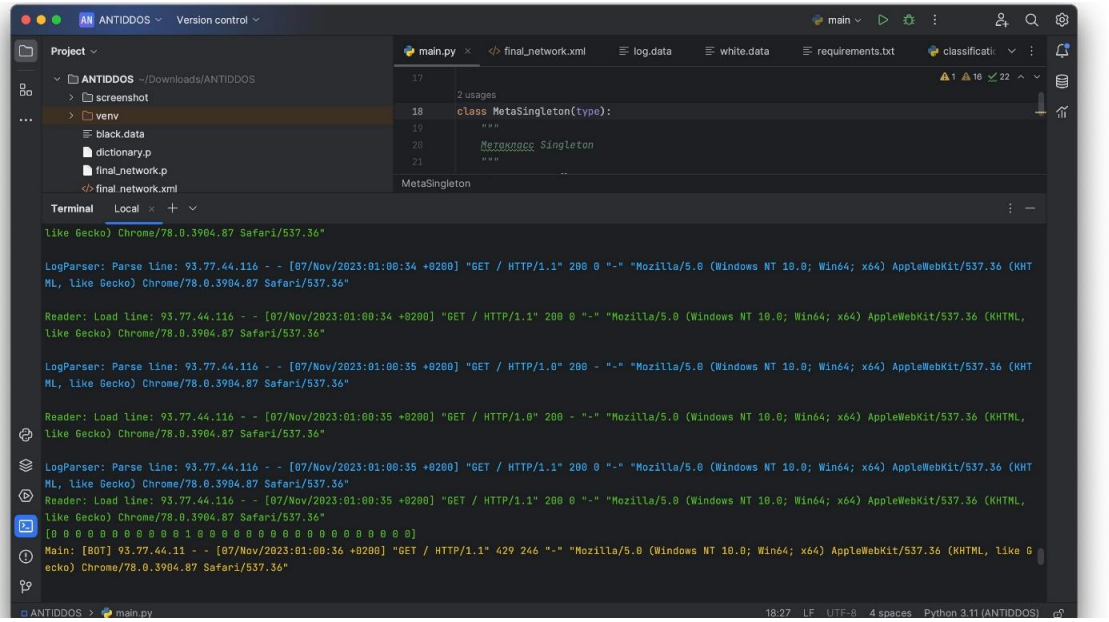


Рисунок 4.11 – Вивід нелегітивного трафіку

Після завершення завдання програма зберігає стан нейронної мережі та деталі навчальних даних у файлах, що дозволяє запустити попередньо навчену систему пізніше.

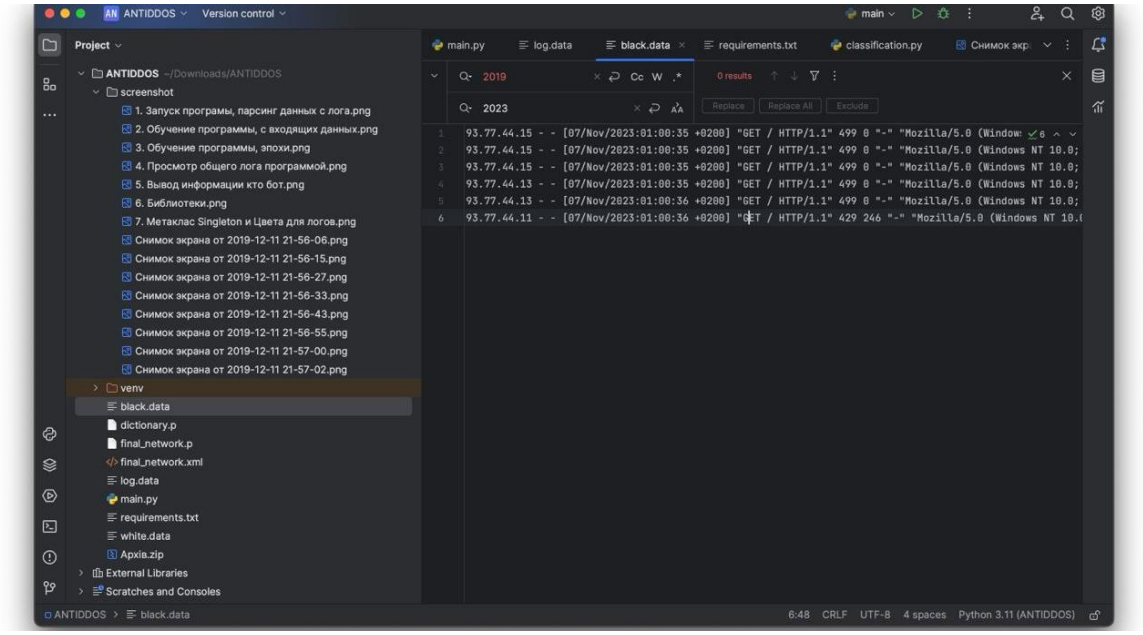


Рисунок 4.12 – Логи нелегітивного трафіку

Логи демонструють запис і відстеження небажаної або зловмисної активності в мережі або.

Ця функція має вирішальне значення для забезпечення кібербезпеки та виявлення можливих загроз.

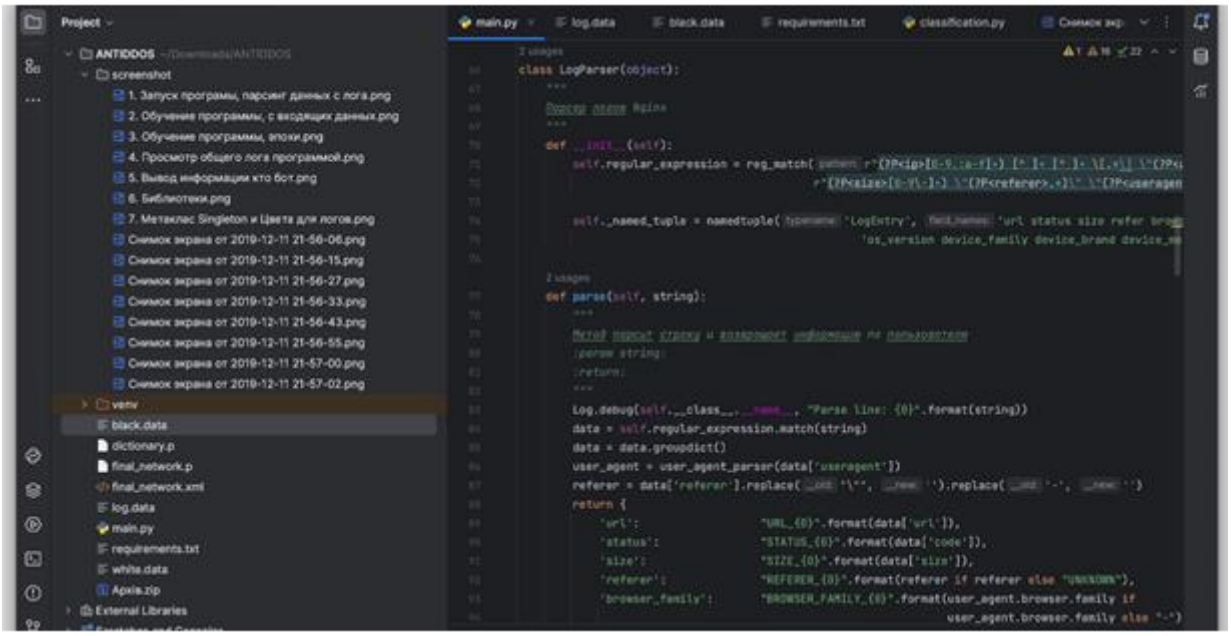


Рисунок 4.13 – Класс для парсингу логів

Парсинг файлів журналів передбачає аналіз і вилучення цінної інформації з журналів або файлів подій.

Цей процес виявляється корисним для моніторингу системи, виявлення помилок, аналізу журналів безпеки та виявлення аномалій.

Аналіз файлів журналів може бути складним завданням, особливо коли формат журналів є різноманітним або змінюється.

Тим не менш, при правильному застосуванні цей метод може сприяти ефективному моніторингу та оцінці активності вашої системи або мережі.

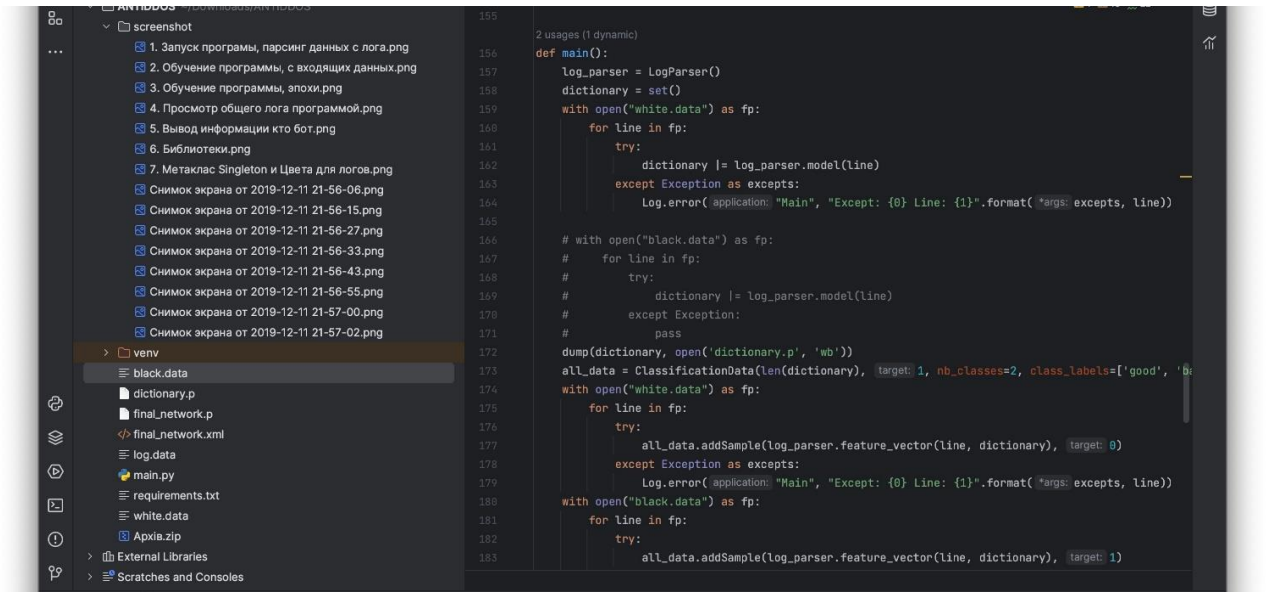


Рисунок 4.14 – Функція навчання системи для нелегітивного трафіку

Визначення методу розпізнавання та відокремлення негативного трафіку від легітимного передбачає збір та обробку різних даних, розробку та навчання моделі, визначення ключових показників ефективності та впровадження її в реальне середовище для постійного моніторингу та захисту від потенційних загроз.

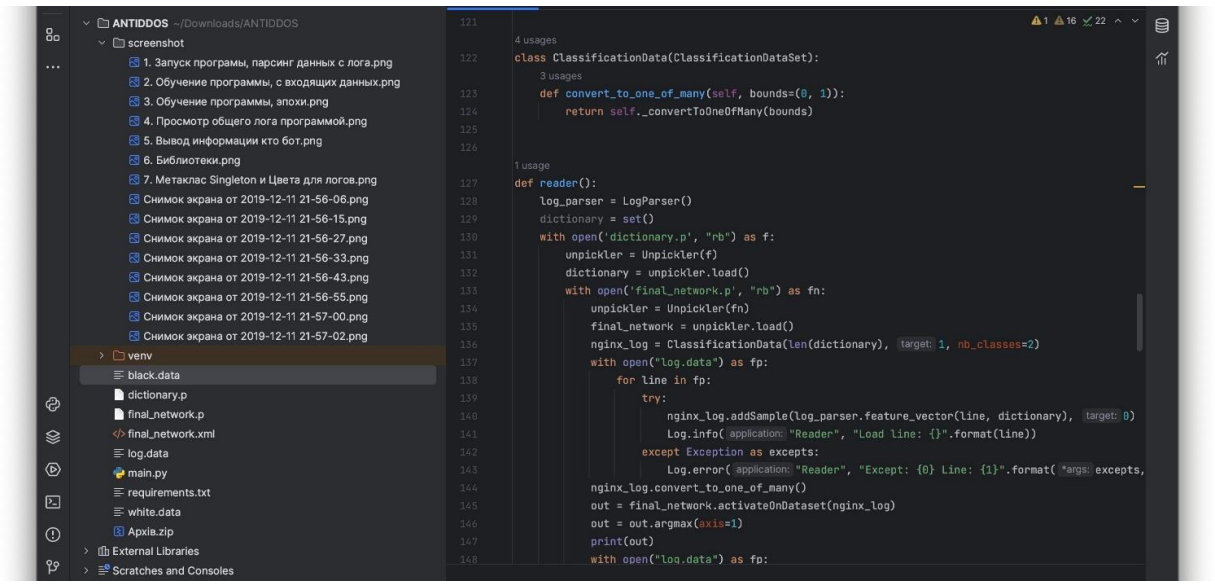


Рисунок 4.15 – Функція реалізації пошуку нелегітивного трафіку

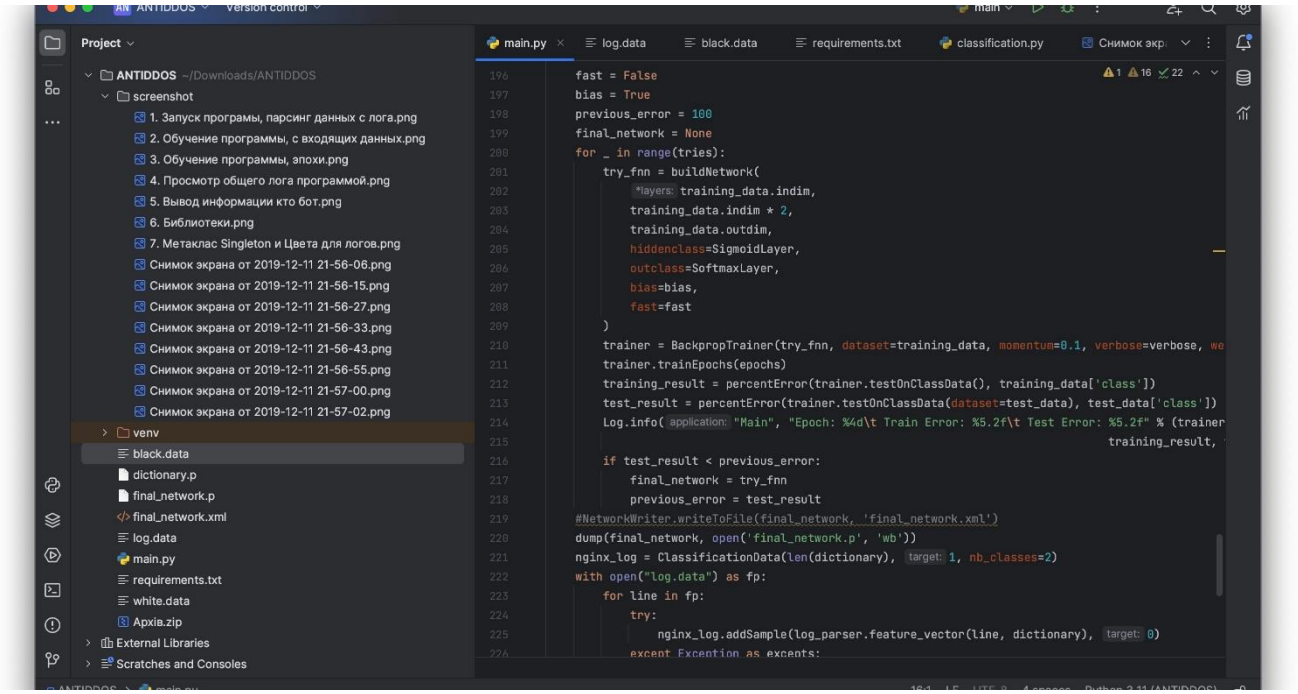


Рисунок 4.16 – Навчання епох

Процес навчання системи складається з певної кількості епох. Протягом кожної епохи модель поступово отримує інформацію через кілька навчальних циклів. Це включає в себе представлення всіх навчальних даних, навчання моделі на цих даних та оцінку результатів. Цикл повторюється протягом певної кількості епох для покращення реакції системи на вхідні дані.

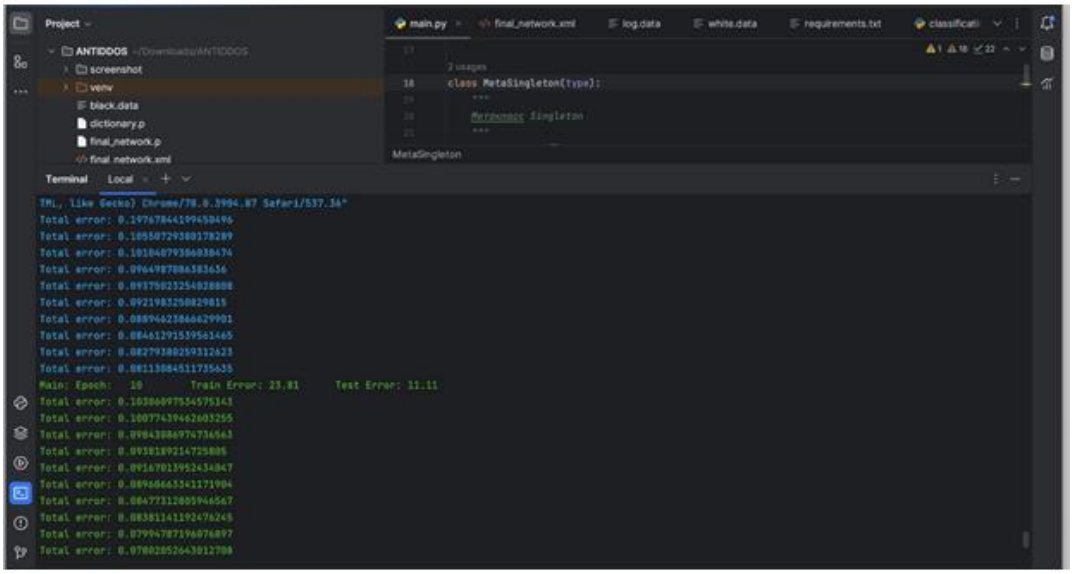


Рисунок 4.17 – Проходження епох

Кожна епоха передбачає повне проходження всіх даних у навчальному наборі. Цей цикл дозволяє моделі покращити свою здатність узагальнювати нові дані.

Процес є ітеративним, де кожна епоха складається з декількох ітерацій, кожна з яких використовує підмножину навчального набору для навчання моделі.

Після завершення кожної епохи модель стає більш вправною в адаптації до даних, які використовувалися для навчання, а тривалість процесу дозволяє моделі оптимізувати свої ваги для кращої реплікації шаблону в даних.

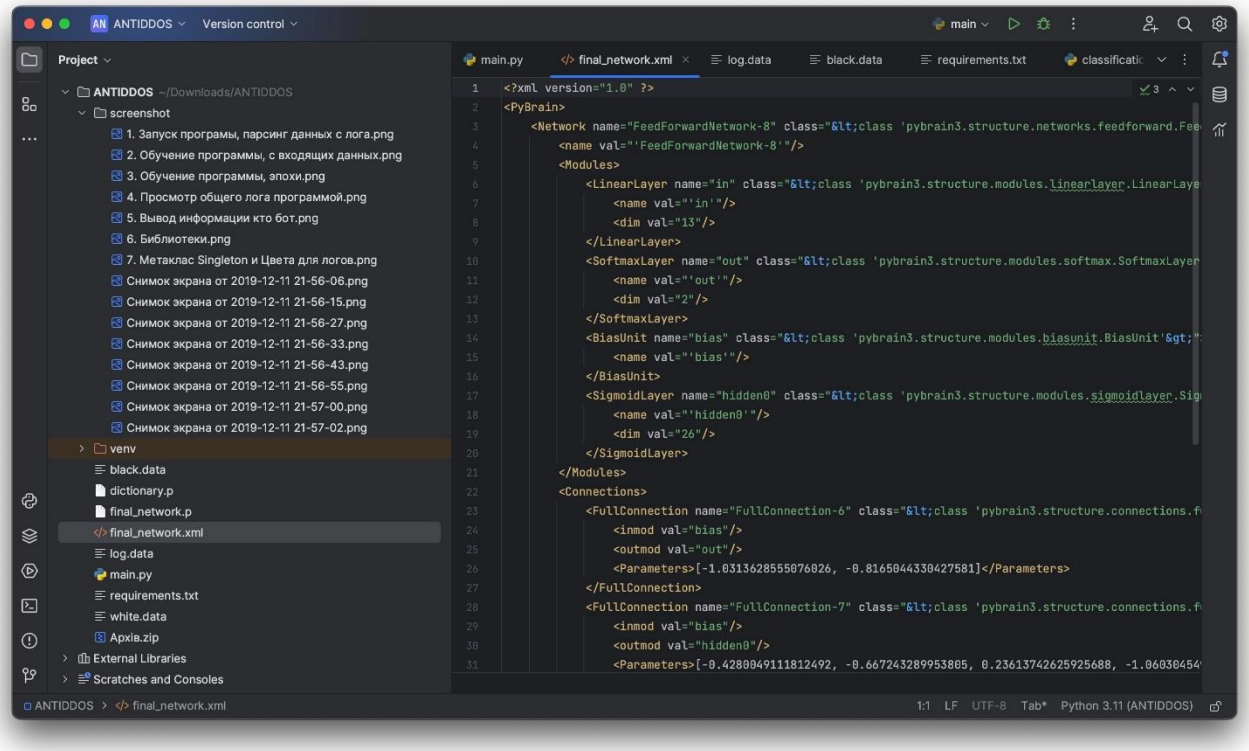


Рисунок 4.18 – XML навченої моделі

4.3 Висновок

Оцінено та застосовано метод аналізу трафіку за допомогою штучної нейронної мережі. Встановлено, що використання штучних нейронних мереж для аналізу мережевого трафіку є ефективним і перспективним напрямком у сфері кібербезпеки та моніторингу мережевої активності.

Основні переваги використання штучних нейронних мереж для аналізу трафіку полягають у можливості виявлення збоїв і вразливостей навіть у складних і нестабільних мережевих середовищах. Модель пройшла навчання на автентичних даних трафіку, що дозволило успішно адаптуватися до модифікацій мережі та виявити потенційні загрози.

Впровадження цього методу підтвердило його здатність ефективно фільтрувати, класифікувати та виявляти аномалії трафіку, що робить його цінним інструментом кібербезпеки. Подальші дослідження і вдосконалення можуть покращити результати і розширити його застосування.

Основною метою є створення надійного та ефективного механізму аналізу трафіку для швидкого виявлення потенційних загроз і захисту мережевих систем в режимі реального часу.

ВИСНОВКИ

У статті представлено метод аналізу трафіку для виявлення атак на складні системи інформаційної безпеки. Основною метою цього дослідження було створення ефективного підходу до виявлення та моніторингу незвичайного мережевого трафіку з метою порушення безпеки інформаційних систем.

Метод використовує штучну нейронну мережу для аналізу та класифікації мережевого трафіку. Модель пройшла навчання на основі різних даних трафіку, включаючи аномальний і типовий трафік. Модель показала високу ефективність у виявленні різних типів атак і аномалій, що робить її потужним інструментом для захисту складних систем від кіберзагроз.

Проведені експерименти підтвердили ефективність розробленої методики та її адаптивність до мінливих мережевих умов. Зібрані дані демонструють чудову точність і швидкість системи виявлення атак, що робить її придатною для розгортання в режимі реального часу.

Цей метод слугує інструментом для посилення кібербезпеки складних інформаційних систем, зменшення ризиків та оперативного реагування на потенційні загрози. Це надійний механізм захисту інформації в сучасному кіберпросторі.

У сучасній кібербезпеці аналіз трафіку є потужним інструментом для виявлення атак на складні інформаційні системи. Його впровадження підвищує обороноздатність та надійність інформаційних систем від різних типів кіберзагроз.

Важливою перевагою моделі є її адаптивність до змін у мережевому середовищі. Врахування динаміки та змін у структурі трафіку дозволяє підтримувати високу точність виявлення аномалій навіть у мінливих умовах.

Навчання моделі на автентичних даних трафіку гарантує, що вона зможе ідентифікувати певні особливості мережі, які можуть бути критично важливими для виявлення конкретних атак, що відбуваються в даному контексті.

Штучна нейронна мережа дозволяє автоматично виявляти і вивчати складні патерни трафіку, підвищуючи точність і здатність виявляти нові атаки.

Здатність оцінювати трафік у режимі реального часу робить цей метод життєво важливим інструментом для швидкого реагування на потенційні загрози та максимізації ефективності системи безпеки.

Розроблений метод може стати важливим заходом для посилення кібербезпеки, зменшення ризиків та гарантування надійного захисту складних інформаційних систем в умовах зростаючих кіберзагроз. Подальші дослідження в цій галузі можуть призвести до вдосконалення та розширення сфери застосування метод

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Алексеєнко А.А., Розломій. І.О. " службові мережеві протоколи для стеганографічної інскапсуляції." Інформаційні моделюючі технології, системи та комплекси (ІМТСК-2021). Третя міжнародна науково-практична конференція, 27-28 травня 2020 р., Черкаси, Україна.–Черкаси: Черкаський національний університет імені Богдана Хмельницького, 2021.–162 с.(збірка тез) В матеріалах конференції відображені результати теоретичних та (2021).

3. Штомпель М.А. "Мережева інженерія." (2022).

4. Гер В.М. Система моніторингу мережі в IoT інфраструктурі. MS thesis. КПІ ім. Ігоря Сікорського, 2022.

5. Колтаков О. А. "Розробка та дослідження системи моніторингу мережі." (2022).

6. Douglas M., and Schmidt K. Essential SNMP: Help for System and Network Administrators. " O'Reilly Media, Inc.", 2005.

7. Авксентьєва І.О. Аналіз мережевого трафіку. BS thesis. КПІ ім. Ігоря Сікорського, 2020.

8. Романчук В. І. "Аналіз структури мережевого трафіку та мережевих аномалій на прикладі сегмента локальної мережі кампусу Національного університету Львівська політехніка”." Вісник Національного університету Львівська політехніка. Радіоелектроніка та телекомунікації 796 (2014): 157-163.

9. Levchuk A. S. "Огляд програмних засобів для аналізу мережевого трафіку." Ukrainian Journal of Educational Studies and Information Technology 3.1 (2016): 34-37.

10. Колодчак О. "Сучасні методи виявлення аномалій в системах виявлення вторгнень." Вісник Національного ун-т «Львівська політехніка». Комп’ютерні системи та мережі 745 (2012): 98-104.

11. Попов Ю., Рузудженк С.,Погоріла К. "SQL-ін'єкції: огляд потенційних способів захисту." Комп’ютерні науки та кібербезпека 3 (2019): 22-26.

12. Бурса М.В. ,Остапенко Г.О. "Оцінка ризику реалізації розподілених атак типу “HTTP-флуд” на багатокомопонентні інформаційно-телекомунікаційні системи". Інформація та безпека 17.3 (2014): 424-427.

13. Мікрюков А.О., Малахін Ю.В. "Вибір варіанта рішення щодо захисту від DDoS атак типу HTTP-флуд на основі методу аналізу ієрархій." Молодь у науці: Нові аргументи. 2018.

14. Singh, Karanpreet, Paramvir Singh, and Krishan Kumar. "Application layer HTTP-GET flood DDoS attacks: Research landscape and challenges." Computers & security 65 (2017): 344-372.

15. Барабаш О.В. ,"Забезпечення функціональної стійкості інформаційних мереж на основі розробки методу протидії DDoS-атакам." (2018).

16. Gossweiler Rich, Maryam Kamvar, and Shumeet Baluja. "What's up CAPTCHA? A CAPTCHA based on image orientation." Proceedings of the 18th international conference on World wide web. 2009.

17. Von Ahn, Luis, "CAPTCHA: Using hard AI problems for security." Advances in Cryptology—EUROCRYPT 2003: International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4–8, 2003 Proceedings 22. Springer Berlin Heidelberg, 2003.

18. Оборін О. О. "Методи виявлення аномального трафіку в IoT." (2022).

19. Каганюк О. К., Бортник К. Я., Свиридюк В. В. "Аналіз аномальних станів трафіка комп’ютерної мережі на базі нейромереж." Комп'ютерно-інтегровані технології: освіта, наука, виробництво 18 (2015): 83-86.

20. Кльоц Ю.П., Петляк Н.С. "Виявлення аномального трафіку у загальнодоступних комп’ютерних мережах." measuring and computing devices in technological processes 3 (2022): 79-86.

21. Крилов С. О. "Розробка системи аналізу мережевого трафіку." (2019).

22. Parziale L. "TCP/IP tutorial and technical overview." (2006).

23. Pawar A. B., "Efficacy of TCP/IP Over ATM Architecture Using Network Slicing in 5G Environment." Smart Data Intelligence: Proceedings of ICSMDI 2022. Singapore: Springer Nature Singapore, 2022. 79-93.

24. Zeeshan A. "Network intrusion detection system: A systematic study of machine learning and deep learning approaches." Transactions on Emerging Telecommunications Technologies 32.1 (2021): e4150.

25. Бабкін А. А., and О. В. Кудін. "Огляд нейромережевих моделей систем виявлення вторгнень." Вчені записки Таврійського національного університету імені ВІ Вернадського Серія: Технічні науки 31.70 (2020): 77-82.

26. Довбешко С. В., Толюпа С. В., and Я. В. Шестак. "Застосування методів інтелектуального аналізу даних для побудови систем виявлення атак." Сучасний захист інформації 1.37 (2019): 6-15.

27. Козак, Є. Б. "Принципи впровадження моделей машинного навчання у сфері інтелектуального обслуговування промислового обладнання." Таврійський науковий вісник. Серія: Технічні науки 3 (2021): 19-28.

28. Кравченко, С. М., Гришкун Є. О., Власенко О. В. "Методи класифікації машинного навчання з використанням бібліотеки scikit-learn." Вчені записки Таврійського національного університету імені ВІ Вернадського. Серія: Технічні науки (2020).

29. Naing, May Thae, Thiri Thitsar Khaing, Aung Htein Maw. "Evaluation of tcp udp traffic over software-defined networking." 2019 International Conference on Advanced Information Technologies (ICAIT). IEEE, 2019.

30. Коберникова Т., Федчук Т.. "Інформаційна технологія безпечного доступу до ресурсів DNS на базі тренованих моделей ідентифікаії тарфіку." SWorldJournal 21-01 (2023): 80-91.

31. Стеблик В. А. Мережевий моніторинг як засіб аналізу інформаційних процесів у локальній і глобальній мережах. MS thesis. 2020.

32. Олійник Є. О. " Забезпечення.безпеки мережі" Зміст: 537.

33. Гавриленко С. Ю., Зозуля В. Д. Дослідження методів виявлення аномалій в даних. Diss. Тарасенко В. П, 2022.

34. Gavrylenko, S., V. Zozulia. " Дослідження методів виявлення аномалій на етапі попередньої обробки данних." Системи управління, навігації та зв'язку. Збірник наукових праць 1.67 (2022): 52-56.

35. Попова В. Р., Бобрікова І. С. "Шифрування даних як один з методів захисту інформації." Стан досягнення та перспективи інформаційних технологій (2022): 70.

36. Gustavsson, Vilhelm. "Machine learning for a network-based intrusion detection system: an application using zeek and the cicids2017 dataset." (2019).

37. Константинова Л.В., Сосна О. С.. "Огляд щзасобів комплексних систем захисту інформації." Тези доповідей (2023): 7.

38. Хлапонін Ю.В. "Комплексні системи захисту інформації." (2022).

39. Гребенніков В.Д. Комплексні системи захисту інформації. Проектування, впровадження, супровід. Litres, 2022.

40. Бабкін А. А., Кудін О. В. "Огляд нейромережевих моделей систем виявлення вторгнень." Вчені записки Таврійського національного університету імені ВІ Вернадського Серія: Технічні науки 31.70 (2020): 77-82.

41. Корченко, А.О. "Методи ідентифікації аномальних станів для систем виявлення вторгнень." (2019).

42. Боднар О.Б. "Виявлення небезпечних входжень у комп'ютерну мережу за допомогою систем виявлення вторгнень та забезпечення захисту такої мережі." (2021).

43. Khadafi, Shah, Yuni Dian Pratiwi, and Enggar Alfianto. "Keamanan Ftp Server Berbasiskan Ids Dan Ips Menggunakan Sistem Operasi Linux Ubuntu." Network Engineering Research Operation 6.1 (2021): 11-24.

44. Голубничий Д. Ю. "Функціональна модель управління системою інформаційної безпеки." (2021).

45. Кухарська Н.П.,Полотай О.І. . "Аспекти інформаційної безпеки в управлінні безперервністю діяльності організації." (2020).

46. Лисенко, Н. О.,. "Огляд математичних методів у системах виявленян та попередженян." Актуальні проблеми автоматизації та інформаційних технологій 25 (2021).

47. Довбешко, С. В., С. В. Толюпа, Я. В. Шестак. "Застосування методів інтелектуального аналізу даних для побудови систем виявлення атак." Сучасний захист інформації 1.37 (2019): 6-15.

48. Толкаченко, Є. А.,. Огляд сучасних методів в системмах виявлення вторгнень (2021): 57-61.

49. Панченко І.О. "Розробка системи моніторингу та аналізу мережевого трафіку." (2020).

50. Харланов М. С. "Аналіз трафіку локальної мережі за допомогою сніферів." (2020).

51. Левандовський А.О. “Метод аналізу трафіку з метою виявлення атак на комплексні системи захисту інформаіїї”/ І.В. Муляр, А.О. Левандовський // Аналіз трафіку в комплексних системах захисту інформації (2023)

52. Jinsong, He, Zhang Yuehong, Hu Guosheng. "Design of ARINC664 bus Network Test System Based on WinPCap." 2020 5th International Conference on Electromechanical Control Technology Transportation (ICECTT). IEEE, 2020.

53. Boyanov, Petar. " investigating the network traffic using the command-line packets sniffer tcpdump in kali linux: investigating the network traffic using the command-line packets sniffer tcpdump in kali linux." Journal scientific and applied research 25.1 (2023): 31-44.

54. Jain, G. "Application of snort and wireshark in network traffic analysis." IOP Conference Series: Materials Science and Engineering. Vol. 1119. No. 1. IOP Publishing, 2021.

55. Iqbal, Haroon, Sameena Naaz. "Wireshark as a tool for detection of various LAN attacks." Int. J. Comput. Sci. Eng 7.5 (2019): 833-837.

56. Каланча, А. А., Клімушин П. С.. "Аналіз мережевого трафіку як спосіб протидії кіберзлочинності." Протидія кіберзлочинності та торгівлі людьми: зб. матеріалів Міжнар. наук.-практ. конф.(м. Харків, 27 трав. 2022 р.).–Харків: ХНУВС, 2022.–С. 42-43, 2022.

57. Авксентьєва І.О. Аналіз мережевого трафіку. BS thesis. КПІ ім. Ігоря Сікорського, 2020.

58. Мальцев А. Ю. "Огляд принципів глибокого навчання як динамічної теорії штучного інтелекту." Вчені записки (2021): 6202197.

59. Григоренко Д. С. " використання алгоритмів навчання без вчителя для сегментації клієнтів банку." інформатика, інформаційні системи та технології: 94.

УДК 004.4

Левандовський А.О., Муляр І.В.

Хмельницький національний університет

МЕТОД АНАЛІЗУ ТРАФІКУ З МЕТОЮ ВИЯВЛЕННЯ АТАК НА КОМПЛЕКСНІ СИСТЕМИ ЗАХИСТУ ІНФОРМАЦІЇ

У даній роботі було розглянуто метод аналізу трафіку як ефективний інструмент виявлення атак на інформаційні системи. Досліжено різні аспекти цього методу. Пропонований метод аналізу трафіку спрямований на виявлення аномалій та підозрілих патернів. Робота спрямована на висвітлення сучасних тенденцій у сфері аналізу трафіку для виявлення атак.

In this work, the method of traffic analysis was considered as an effective tool for detecting attacks on information systems. Various aspects of this method have been studied. The proposed method of traffic analysis is aimed at detecting anomalies and suspicious patterns. The work is aimed at highlighting modern trends in the field of traffic analysis to detect attacks..

У сучасному інформаційному суспільстві, де комп'ютеризація та мережеві технології стають неодмінною частиною ділового та особистого середовища, питання забезпечення безпеки інформації набувають все більш актуального характеру.

Інформаційні системи стають об'єктом спроб несанкціонованого доступу, втручання та руйнування з боку зловмисників, що може призвести до серйозних наслідків для організацій, користувачів та суспільства в цілому.

Комплексні системи захисту інформації (КСЗІ) - це сукупність технологій, процедур, методів та інструментів, які призначені для захисту конфіденційності, цілісності та доступності інформації в інформаційних системах. Ці системи використовуються для запобігання несанкціонованому доступу до даних, їхнього розголошення чи втрати, а також для забезпечення функціонування інформаційних ресурсів в нормальному режимі навіть в умовах атак або катастроф [1].

Ці системи можуть включати в себе різноманітні компоненти, такі як файрволи, антивіруси, системи виявлення вторгнень, шифрування даних, аутентифікаційні методи, фізичні заходи безпеки та багато іншого. Вони використовуються в різних сферах, які обробляють конфіденційну інформацію

Ці системи важливі для захисту інформації в сучасному світі, де інформація відіграє ключову роль у багатьох аспектах суспільства та бізнесу. Забезпечення безпеки інформації є завданням важливим як для державних структур, так і для підприємств і інших організацій.

Інтеграція технічних та організаційних заходів грає ключову роль у забезпеченні спільної роботи різних компонентів для досягнення оптимального рівня захисту. Комплексні системи захисту інформації продовжують розвиватися, адаптуючись до зростаючих технологічних викликів та різноманітних кіберзагроз, забезпечуючи високий рівень захисту для конфіденційної інформації та інфраструктури в інформаційному середовищі.

Організаційні та управлінські компоненти включають політики безпеки, тренування персоналу, процедури відновлення після інциденту та моніторинг безпеки.

Аналіз трафіку в комплексних системах захисту інформації є динамічним процесом, який вимагає поєднання технологій та людського експертного втручання для ефективного виявлення та відвернення кіберзагроз

Головна ідея - це стежити за тим, як дані пересуваються, шукати щось, що може бути нестандартним або небезпечним, і вчасно реагувати, щоб запобігти можливим проблемам у безпеці інформації.

Метою цієї роботи є вивчення, аналіз та розробка методу аналізу трафіку з метою виявлення атак на комплексні системи захисту інформації, а також розкриття ефективності таких методів у забезпеченні кібербезпеки та виявленні потенційно небезпечних аномалій у мережевому трафіку

Враховуючи постійний розвиток кіберзагроз та їхню вдосконаленість, аналіз трафіку виявляється ефективним інструментом у недопущенні серйозних наслідків для конфіденційності, цілісності та доступності інформації.

Метою використання методу аналізу трафіку є створення надійних та ефективних систем захисту, які здатні протистояти різноманітним загрозам у сучасному цифровому середовищі.

Подальший розвиток цього методу передбачає вдосконалення та адаптацію до змінюючогося характеру кіберзагроз, щоб забезпечити високий рівень безпеки інформації в умовах постійного технологічного розвитку.

Метод аналізу трафіку з метою виявлення атак на комплексні системи захисту інформації включає в себе кілька етапів та підходів.

- Моніторинг мережевого трафіку для виявлення несподіваного або підозрілого трафіку, який може вказувати на атаки або несанкціонований доступ;
- використання алгоритмів виявлення аномалій для ідентифікації незвичайних патернів у трафіку, що можуть вказувати на потенційні атаки або порушення;

- використання спеціалізованих систем виявлення вторгнень для виявлення атак, які відбуваються на основі аналізу мережевого трафіку та виявлення вразливостей у системах;
- використання бази даних з підписами відомих атак для порівняння з моніторинговим трафіком з метою виявлення відповідних атак;
- аналіз потоку даних з точки зору шаблонів атак та нормального трафіку для виявлення відхилень від очікуваного поведінки;
- аналіз журналів подій для виявлення незвичайної або підозрілої активності, яка може вказувати на потенційні атаки або вторгнення.

Отже цей метод використовуються для забезпечення максимального рівня захисту інформації та виявлення потенційних загроз для комп'ютерних систем і мереж. Цей метод допомагає підвищити рівень безпеки комплексних систем захисту інформації, виявлюючи атаки рано і запобігаючи їх негативним наслідкам. метод аналізу трафіку дозволяє ефективно виявляти аномалії у мережевому середовищі, вказуючи на можливі атаки або інші загрози інформаційній безпеці.

Важливим аспектом є не лише реакція на вже відомі атаки, але й передбачення та запобігання новим загрозам. Такий підхід дозволяє комплексним системам захисту інформації підтримувати високий рівень безпеки в непередбачуваному кіберпросторі.

Також рекомендується постійне вдосконалення методів аналізу трафіку, враховуючи змінюючийся характер кіберзагроз, та впровадження нових технологій для ефективного виявлення та запобігання атакам для забезпечення ефективного протидії невизначеним і постійно зростаючим кіберзагрозам у сучасному інформаційному середовищі..

Такий підхід допоможе забезпечити безпеку інформаційних систем в умовах постійно зростаючого рівня загроз та комплексності кібератак. метод аналізу трафіку є важливою ланкою у системах кібербезпеки, забезпечуючи необхідний рівень захисту для комплексних систем.

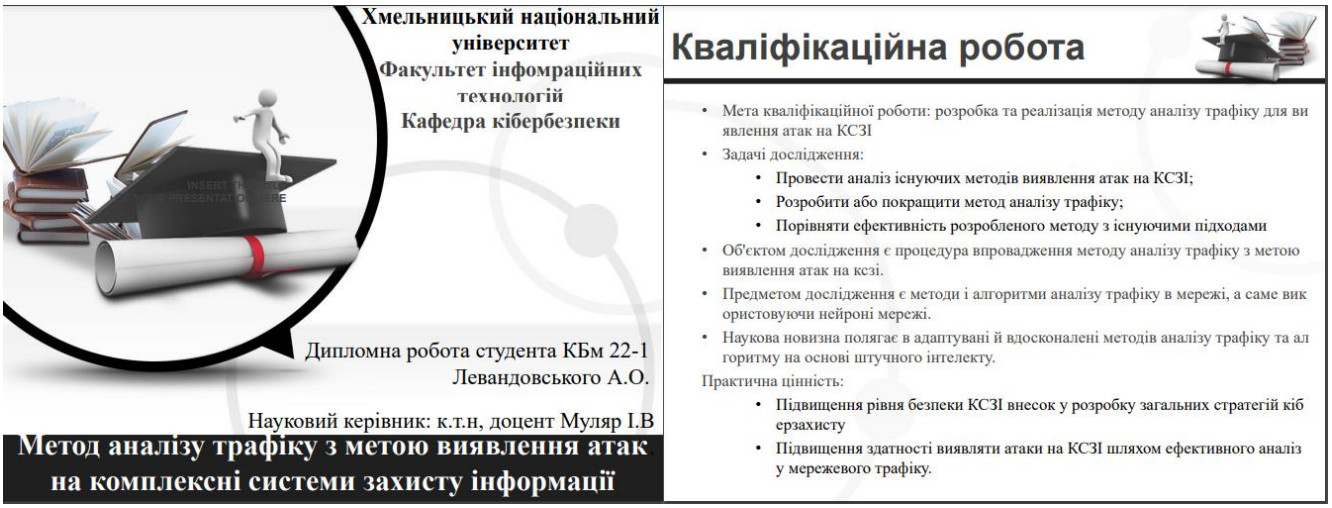
Перелік посилань

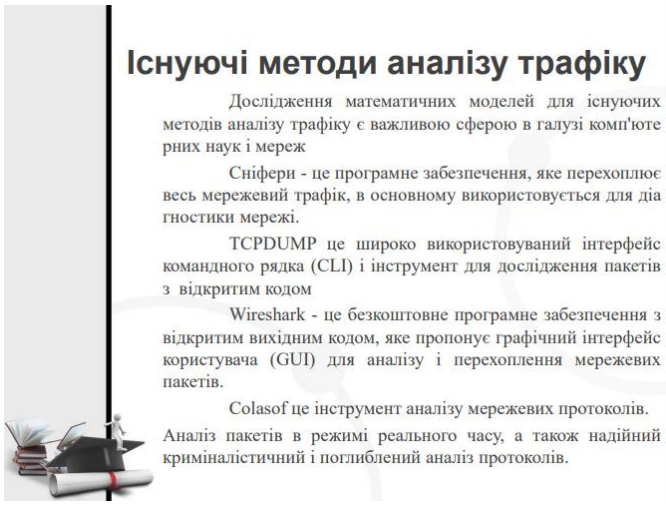
1. Комплексні системи захисту інформації Навчальний посібник. / Ю. Є. Яремчук, П. В. Павловський, В. С. Катаєв, В. В. Сінюгін // 2018
2. Технології захисту інформації Навчальний посібник. / А. В. Жилін, О. М. Шаповал, О. А. Успенський // 2020

Дані про авторів (не для друку)**:**

ПІБ автора	Телефон	Email
Левандовський Андрій Олександрович	+380682066801	Lewandovskiy.a77@gmail.com
Муляр Ігор Володимирович	+380679381544	muliariv@khmnu.edu.ua

ДОДАТОК Б Презентація кваліфікаційної роботи

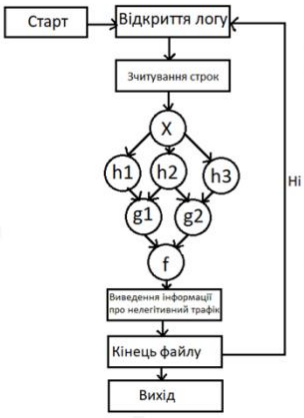




Якщо навчений класифікатор дає очікуваний результат і відповідає критеріям класифікації, починається наступна фаза. Результатом етапу навчання є класифікатор з оптимізованими параметрами, здатний якісно класифікувати дані.

Навчання в методі на основі штучного інтелекту

Приклад роботи методу



Для виконання завдання було використано мову про грамування Python 3.7, бібліотеку PyBrain3 для побудови ней ронної мережі, бібліотеку pickle для зберігання стану мережі, бібліотеку numpy для зручного маніпулювання даними та біб ліотеку user_agents для отримання інформації про агента кори стувача.

Алгоритм навчання з вчителем. Метод зворотнього поширення.

Це один з методів машинного навчання, в якому тест ова система примусово навчається на прикладах "стимул-реак ція".

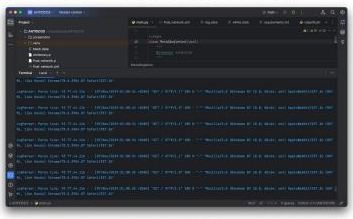
Фундаментальний принцип цього підходу полягає в поширенні сигналів помилок від виходів мережі до її входів, діючи в напрямку, протилежному звичайному поширенню сиг налу.

- Спочатку дані проходять підготовку, і кожному екзем пляру даних, наприклад, зображенню або тексту, при своюється вектор ознак;
- Потім нейронна мережа навчається на цих векторах ознак, і алгоритм оптимізує ваги нейронів для точної відповідності вхідним даним;
- У процесі навчання використовується функція втрат, щоб виміряти розбіжність між помітними відповідям и нейронної мережі та очікуваними відповідями.

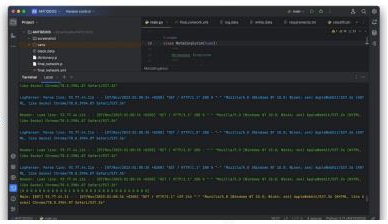
- Нові дані перетворюються у вектор ознак за тими ж прави лами, що і під час навчання;
- Модель використовує вектор ознак для прогнозування або класифікації залежно від типу завдання;
- Нейромережа видає результат, який може мати форму ймов ірностей (у випадку класифікації) або числових значень (у випадку регресії).

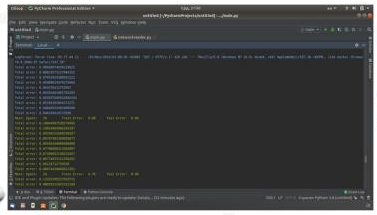
Зображення з роботи методу

Процес зчитування логів

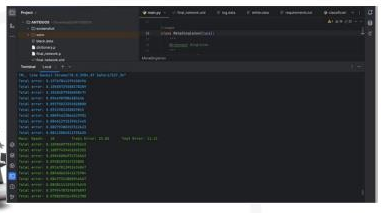


Вивід нелегітивного трафіку





Процес навчання з вхідних данних



Проходження епох

Дякую за увагу!!!

Завідувачу кафедри кібербезпеки
к.т.н., доц. Кльоцу Ю.П.
Левандовського А.О
ПІБ здобувача вищої освіти

Студента ФІТ, 2 курсу, групи КБм-22-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у хмельницькому національному університеті» від 31.08.2023, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

11.12.2023
дата


підпис

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 0.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 9%**

	Документ		Сумарний збіг по Базі Даних	
ID: 123329 Назва: Метод аналізу трафіку з метою виявлення атак на комплексні системи захисту інформації Додано в БД: 2023-12-15 Автора: Левандовський А.О. Керівники: Муляр І.В. Консультанти: Опоненти:	Символи	Лексеми	Символи	Лексеми
	82174	1265	779 (1%)	12 (1%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

UNICHECK
by Turnitin

Ім'я користувача:
Кафедра кібербезпеки

ID перевірки:
1016008671

Дата перевірки:
15.12.2023 12:17:54 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
15.12.2023 12:18:52 EET

ID користувача:
100008300

Назва документа: **Левандовський_на_плагіат**

Кількість сторінок: 77 **Кількість слів:** 12323 **Кількість символів:** 97323 **Розмір файлу:** 3.38 MB **ID файлу:** 1015694303

2.56%
Схожість

Найбільша схожість: 0.43% з джерелом з Бібліотеки (ID файлу: 1015661485)

2.13% Джерела з Інтернету	190	Сторінка 79
0.73% Джерела з Бібліотеки	46	Сторінка 80

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи	2

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КІБЕРБЕЗПЕКИ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності вподо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод аналізу трафіку з метою виявлення атак на комплексні системи захисту інформації

Автор: Левандовський Андрій Олександрович

Науковий керівник: Муляр Ігор Володимирович, к.т.н, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Оригінальність тексту роботи за результатами перевірки системою Unichek складає 97,44%, оригінальність тексту роботи за результатами перевірки системою Anti-Plagiarism v-15.257 складає 100%.

Згідно з правилами чинного Положення «Про систему забезпечення академічної доброчесності у хмельницькому національному університеті» від 31.08.2023, авторська робота, обсяг оригінального тексту у відсотках до загального обсягу матеріалу в якій складає 90-100 %, визнається роботою з високою унікальністю тексту і допускається до захисту.

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1. Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 2.56%, з яких 2.13% є збігами з одним джерелом, зумовленими наявністю типових фразеологічних виразів предметної області, а також формулюваннями, які утворюють загальновживані фрази.

2. Інші збіги є збігами в назвах використаних друкованих видань, розміщених в переліку джерел посилань

Виявлені системою Unichek модифікації стосуються математичних формул і не є порушенням академічної доброчесності.

Керівник роботи _____ І.В. Муляр

Гарант ОП _____ В.Ю. Тітова

Завідувач кафедри кібербезпеки _____ Ю.П. Кльоц

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА ДИПЛОМНУ РОБОТУ

ОПП «магістр»

Магістр Левандовський А.О.

Тема Метод аналізу трафіку з метою виявлення атак на комплексні системи захисту інформації

Галузь знань 12 – Інформаційні технології

Спеціальність 125 – Кібербезпека

Обсяг дипломної роботи ОПП «магістр»:

кількість листів креслень ; кількість сторінок записки 87

1. Короткий зміст ДР та прийнятих рішень В рамках магістерської роботи було проаналізовано предметну область, увагу зосереджено на методах аналізу трафіку, що використовуються для виявлення потенційних атак на складні системи інформаційної безпеки. Розроблено та покращено метод аналізу трафіку на основі алгоритму штучного інтелекту.
Запропоновано удосконалений метод виявлення атак на комплексні системи захисту інформації. Метод дозволяє з високою ймовірністю ідентифікувати аномалії та нелегітимний трафік в мережі на основі навчання штучного інтелекту .

2. Висновок про відповідність ДР дипломному завданню Дипломна робота ОП «магістр» у повній мірі відповідає поставленому завданню як в теоретичній, так і в практичній частині дипломної роботи.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі обґрунтовано актуальність дослідження проблеми забезпечення інформаційної безпеки соціальних мереж. Проведено аналіз існуючих підходів у цій сфері та окреслено напрямки удосконалення методів моніторингу й виявлення загроз. Сформульовано мету та конкретні завдання роботи. У першому розділі досліджено предметну область та наявні методи аналізу трафіку. Наступні розділи присвячені безпосередньо розробці математичної моделі ,методу аналізу трафіку та реалізації розробленого методу. Також розглянуті питання практичного застосування отриманих результатів.

4. Позитивні сторони проекту У роботі запропоновано низку інноваційних підходів у сфері кібербезпеки аналізу трафіку в мережі. Зокрема, розроблено метод аналізу трафіку з метою виявлення атак на ксзі. Це дозволяє здійснювати к оцінку потенційної небезпеки на мережу та прогнозувати ескалацію інформаційних загроз в трафіку.

5. Негативні сторони проект. Метод на основі алгоритму штучного інтелекту в деяких випадках потребую тривалого навчання, та схильний падати в локальні мінімуми.
Можливе допрацювання методу для кращих результатів.

6. Оцінка графічного оформлення та пояснювальної записки роботи Пояснювальна записка відповідає нормам для її оформлення.

7. Відгук про роботу в цілому В загальному дипломна робота заслуговує позитивної оцінки. Весь матеріал дипломної роботи структурований, чіткий та послідовний. Усі розділи роботи послідовні та логічні, що дозволяє чітко розуміти викладений матеріал в рамках тематики дипломної роботи.

8. Інші зауваження

9. Оцінка дипломної роботи Розглянувши позитивні та негативні сторони представленої дипломної роботи, можна зробити висновок, що вона заслуговує оцінку «добре». В

РЕЦЕНЗЕНТ (прізвище, ім'я, по батькові, посада, місце роботи) Завідувач кафедри автоматизації, комп'ютерно-інтегрованих технологій та робототехніки, доктор технічних наук, професор Мартинюк Валерій Володимирович

«19» грудня 2023. (підпис)

ХНУ. Зам. 54. тир. 6000. 2007