Хмельницький національний університет
Факультет інформаційних технологій
Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр

Система виявлення аномального трафіку на основі сигнатур

КРКБ. 190114.19.01.11 ПЗ

Галузь знань 12 – Інформаційні технології

Спеціальність 125 – Кібербезпека

Виконав студент 4 курсу, група КБ-19-1 _____ Хмельовський В.Р
Підпис Ініціали, прізвище

Керівник д-р філософії _____ Стецюк М.В
Науковий ступінь, вчене звання Підпис, дата Ініціали, прізвище

Нормконтролер _____ 06.06.23 Мостовий С.В
Науковий ступінь, вчене звання Підпис, дата Ініціали, прізвище

До захисту допускаю:
Зав. кафедри кібербезпеки _____ Кльоц Ю.П
Підпис, дата Ініціали, прізвище

6 06 2023 р.

Хмельницький, 2023

Формат	Зона	Позиц	Позначення	Найменування	Кільк.	Прим.
А4		1	КРКБ.190114.19.01.11 ПЗ	Система виявлення	62	
				аномального трафіку на		
				основі сигнатур		
				Пояснювальна записка		
А2		2	КРКБ.190114.19.01.11 Е8	Модель процесу моніторингу	1	
				мережі задля виявлення		
				аномальної поведінки		
				Схема структурна		
А2		3	КРКБ.190114.19.01.11 Е8	Модель роботи аналізатора	1	
				трафіка та обробка recv()		
				Схема структурна		
А2		4	КРКБ.190114.19.01.11 Е8	Модель роботи алгоритма	1	
				одного пакета; захоплення		
				пакетів, які потрапляють на		
				інтерфейс; порівняння		
				IP-адрес		
				каналів витоку		
				Схема структурна		

Зм.	Арк.	№ Докум.	Підп.	Дата	КРКБ.190114.19.01.11 ВП		
Розробив		Хмельовський В.Р		06.06.23	Система виявлення аномального трафіку на основі сигнатур Відомість проєкту	Літера	Аркуш 1 / Аркушів 1
Перев.		Стецюк М.В		06.06.23		н	
Н. контр.		Мостовий С.В.		6.06.23		ХНУ, КБ-19-1	
Затв.		Кльоц Ю.П.		6.06.23			

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Кібербезпеки

Освітній рівень Бакалавр

Галузь знань 12 Інформаційні технології

Спеціальність 125 Кібербезпека

Освітня програма Освітньо-професійна програма підготовки бакалаврів

ЗАТВЕРДЖУЮ

Зав. кафедри Ю.П. Кльоц

"1" 03 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Хмельовський В.Р

Прізвище, ім'я, по батькові студента

1. Тема роботи Система виявлення аномального трафіку на основі сигнатур

Керівник роботи д-р філософії Стецюк М.В

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджено наказом ректора університету від 01 березня 2023р. №5

2. Строк подання студентом роботи на кафедру 01.06.2023 р.

3. Вихідні дані до проекту (роботи) проаналізувати роботу мережі; визначити що буде аномальним; розробити програмно-апаратний засіб для сканування мережі; створити словник сигнатур.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) Вступ. Аналіз роботи мережі. Визначення що буде нормальним, а що аномальним. Реалізація додатка для сканування мержевого трафіку. Висновки.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень): «Модель процесу моніторингу мережі задля виявлення аномальної поведінки», «Модель роботи аналізатора трафіка та обробка запитів recv()», «Моделі роботи алгоритма захоплення одного пакета; захоплення пакетів, які потрапляють на інтерфейс; порівняння IP-адрес»

6. Консультанти розділів курсового проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 1 березня 2023р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи	Термін виконання етапів проекту (роботи)	Примітка
1	Ознайомлення з предметною областю	12.03.2023	–
2	Пошук теоретичної інформації про системи виявлення аномального трафіку на основі сигнатур	20.03.2023	–
3	Дослідження існуючих рішень	04.04.2023	–
4	Постановка задачі	18.04.2023	–
5	Пошук теоретичної інформації про найкращі рішення для виявлення аномального трафіку на основі сигнатур	26.04.2023	–
6	Початок впровадження системи виявлення аномального трафіку на основі сигнатур	01.05.2023	–
7	Завершення реалізації виявлення аномального трафіку на основі сигнатур	26.05.2023	–
8	Оформлення пояснювальної записки згідно вимог	28.05.2023	–
9	Оформлення графічної частини	31.05.2023	–
10	Захист КР	08.06.2023	

Студент _____ Підпис Хмельовський В.Р Ініціали, прізвище

Керівник проекту (роботи) _____ Підпис Стецюк М.В Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Система виявлення аномального трафіку на основі сигнатур»

Автор роботи: Хмельовський Віктор Русланович.

Керівник роботи: Стецюк Микола Васильович.

Пояснювальна записка: 62 с., 3 додатки, 17 рис., 40 джерел.

Графічна частина: 10 презентаційних слайдів.

Метою роботи є розробка система моніторингу мережевих пакетів, та виявлення аномалій у їх поведінці.

Було проведе дослідження процесу виявлення аномалій у мережевому трафіку та було розглянуто існуючі та актуальні підходи до реалізації додатку. Визначено, що використання методу сигнатурного аналізу є досить ефективним і надійним способом виявлення небажаного трафіку.

Також у результаті був розроблений додаток, який виявляє підозрілий трафік, відповідно до створеного словника сигнатур (банк небажаних сигнатур).

06.06.2023

ANNOTATION

Course project: "System for detecting anomalous traffic based on signatures"

Author of the work: Khmelovskyi Viktor Ruslanovych

Supervisor: Stetsyuk Mykola Vasyliovych.

Explanatory note: 62 pp., 3 appendices, 17 figures, 40 sources.

Graphic part: 10 presentation slides.

The method of work is the development of a network packet monitoring system and the detection of anomalies in their behavior.

A study of process anomalies in network traffic was conducted and existing and relevant approaches to application implementation were identified. It was noted that the use of the signature analysis method is a quite effective and reliable way of detecting unwanted traffic.

Also, as a result, an application was developed that detects suspicious traffic, according to the created dictionary of signatures (bank of unwanted signatures).

06.06.2023

ЗМІСТ

ВСТУП

Стрімка популяризація і розвиток мереж значно ускладнили розрахункові можливості систем і зробили їх пов’язаними відносну один-одному, відповідно, менш захищені від шкідливих дій зовні. Ріст рівня автоматизації процесів обробки, зберігання, обробка інформації і також передача інформації також зазвичай відбивається на виникненні проблем по забезпеченню безпеки, а витрати на покриття збитків від діяльності зловмисників збільшується.

Варто згадати, що спостерігається стійка тенденція до збільшення кількості атак на розрахункові\обчислювальні системи і також мережі: способи, а також методики віддаленого керування постійно удосконалюються, а існуючі системи захисту інформації\мереж не дозволяють вчасно реагувати на змінення в цій сфері, тому спочатку варто віднайти, а після і вивчити мережеву атаку. Через це гемає можливості повністю виключити шкідливий трафік. Ці дії надають більше значення питанням реалізації ефективних методик виявлення несанкціонованого трафіку і реалізації актуальних засобів захисту інформації.

Дослідження та розвиток мережевих технологій доводять до збільшення складності систем і їх взаємодії, що відкривають шлях до недоброзичливих дій. Автоматизація процесів обробки, та зберігання, передача інформації збільшує ризики хакероподібних атак і несанкціонованого доступу.

В цьому контексті актуальною стає розробка систем виявлення підозрілої активності у мережі.

Актуальність даної теми обумовлена тим, що на даний момент досить активно розробляється і застосовуються різні методи по виявленню і передзахисту вторгнень, але вони не завжди є ефективними на практиці. Як висновок – усі технології захисту постійно вивчаються і вдосконалюються.

Існуючі на даний момент системи об’єднує загальна риса – захист локальної мережі від зовнішніх\зловмисних дій які спрямовані на мережу чи конкретний об’єкт пов’язаний з нею. В даній роботі розглядається проектування системи виявлення підозрілих дій з трафіком таким чином, щоб на основі цих дій людина

відповідальна за мережу (адміністратор) зміг прийняти адекватне своєчасне рішення вирішення проблеми безпеки і тим самим зменшити небезпеку, яка стосується локальної мережі.

Тому цілю даної роботи є розробка прототипу система візуального аналізу вхідного і вихідного трафіку для аналізу і виявлення підозрілої\несанкціонованої активності.

В рамках даної роботи будуть вирішені наступні задачі:

- аналіз роботи мережі;
- визначення аномального і неаномального трафіку;
- розробка програмного засобу для виявлення підозрілої активності;
- створення словника сигнатур.

1 АНАЛІЗ РОБОТИ МЕРЕЖІ

У процесі обробки та зберігання інформації неодмінно виникає необхідність обміну даними між користувачами цього процесу. З початку 80-х років почався бурхливий розвиток комп'ютерних мереж та супутнього мережного обладнання. Локальні та глобальні мережі досі продовжують розвиватися, виникають нові протоколи передачі даних, розширюються апаратні можливості мережного обладнання, зростає кількість підключених абонентів та сумарний обсяг трафіку.

Стрімкий розвиток області тягне за собою низку проблем. Одна з них полягає в тому, що при збільшенні кількості користувачів інформаційних послуг збільшуються й вимоги до мережевого та серверного обладнання, яке використовується для підтримки належного рівня якості обслуговування. Друга ґрунтується на необхідності захисту інформації, що циркулює у середині мережі.

Для вирішення цих проблем використовують аналіз трафіку та його моніторинг, що допомагає ефективно діагностувати та вирішувати проблеми при їх виявленні, не дозволяючи мережевому обладнанню простоювати довго. Так як інформація по мережі передається постійно, стає зрозуміло, що припинення роботи апаратури, або інші причини відмови в обслуговуванні, призводять до збитків організацій, або компаній, що надають послуги. У зв'язку з цим адміністраторам необхідно монітори рух мережевого трафіку та продуктивністю усієї мережі, а також перевіряти її на пробої у безпеці [1].

1.1 Класифікація засобів моніторингу та аналізу

Інструменти, які пропонуються для моніторингу та аналізу обчислювальних мереж, можна розділити на декілька груп:

– Системи управління мережею (Network Management Systems) - централізовані програмні системи, які збирають дані про стан мережевих пристроїв та інформацію про трафік у мережі. Функціонал даних програм не

обмежений моніторингом та аналізом мережі. Додатково, в напівавтоматичному або автоматичному (залежно від реалізації) режимі, здійснюються дії з управління мережею: налаштування та зміна адресних таблиць комутаторів та іншого обладнання, увімкнення та вимкнення портів пристроїв. До систем цієї категорії відносяться HPOpenView, SunNetManager, IBMNetView:

– вбудовані системи діагностики та управління (Embedded systems). Системи цього типу виконані як програмно-апаратні модулі, які встановлюються в мережеве обладнання, чи – в операційну систему як програмний модуль. Завдяки їм, є можливість керувати та діагностувати лише тим пристроєм, на якому він знаходиться. Прикладом таких систем є модуль керування концентратором ADAM-5630, який виконує функції автоматичної сегментації портів після виявлення неполадок, приписування портів внутрішнім сегментам концентратора та інші. Зазвичай, вбудовані модулі управління виконують роль SNMP-агентів, передаючи дані про стан пристрою в систему управління.

– засоби керування системою (System Management). Інструменти цієї групи виконують функції, аналогічні функціям систем керування, але по відношенню до інших об'єктів. У першому випадку об'єктом управління є програмне та апаратне забезпечення комп'ютерів у мережі, а в другому – мережеве обладнання. При цьому частина функцій цих видів систем можуть дублюватися.

– аналізатори протоколів (Protocol analyzers) – це програмні чи програмно-апаратні засоби, використовувані лише для моніторингу та аналізу трафіку у мережах. Чудовим аналізатором вважається той, який може захоплювати і декодувати пакети великої кількості протоколів, що застосовуються у мережах. Ця група систем може встановлювати деякі логічні передумови для захоплення окремих пакетів та виконувати повне декодування пакетів, тобто відображати у зручній для користувача формі вкладеність пакетів протоколів різних рівнів з розшифровкою змісту кожного поля пакета.

На моменті створення або вдосконалення мережі, часто виникає потреба в кількісному вимірі характеристик мережі, наприклад: затримки, що виникають на

різних етапах, частота виникнення вибіркових подій, інтенсивність потоків даних по лініях зв'язку, час реакції на запити.

– Обладнання для діагностики та сертифікації кабельних систем. Призначення цієї групи. Умовно можна виділити чотири підтипи такого обладнання: кабельні сканери, мережеві монітори, мультиметри та прилади для сертифікації кабельних систем.

– Експертні системи поєднують знання оточуючих, про виявлення причин аномальної роботи мережі та можливі способи повернення мережі у нормальний стан. Найчастіше представлено у вигляді окремих підсистем інших засобів моніторингу та аналізу мереж.

1.2 Системи виявлення та запобігання вторгненням

Впровадження подібних систем захисту інформації є необхідністю всім серйозним мережевим інфраструктурам, оскільки існують програми, які постійно вишукують вразливості у будь-якому обладнанні, підключеному до глобальної мережі. Наприклад, пошуковик Shodan [2] в автоматичному режимі збирає інформацію про підключені пристрої, які не мають будь-якої частини системи безпеки. Користувачі Shodan знаходять системи керування крематорієм, газовою станцією тощо, які не мають реквізитів доступу або вони налаштовані за умовчанням. Відповідно, до них можна легко пробитись та зменшити працездатність.

Проти такого впливу і спрямовані системи виявлення та запобігання вторгнення, тому вони є інструментом, що часто застосовуються в політиці безпеки:

– система виявлення вторгнень (англ. Intrusion Detection System) – апаратний чи програмний засіб, призначений для виявлення фактів

неавторизованого доступу (вторгнення чи мережевої атаки) в комп'ютерну систему чи мережу.

– система запобігання вторгнення (англ. Intrusion Prevention System) – апаратний чи програмний засіб, що здійснює моніторинг мережі або системи в реальному часі з метою виявлення, запобігання або блокування шкідливої активності.

Системи запобігання вторгнень можна вважати розширенням систем виявлення вторгнень, так як задача відстеження атак залишається тією ж самою. Але система запобігання вторгнення має відслідковувати вторгнення в реальному часі і одразу здійснювати дії щодо запобігання шкідливим діям. Для цього вони використовують: скидання з'єднання, блокування потоків трафіку в мережі, видачу сигналів оператору. Крім цього, такі системи можуть дефрагментувати пакети, змінювати порядок TCP пакетів для захисту від пакетів зі зміненими SEQ і ACK номерами тощо [3].

Дані системи використовуються для автоматизації процесу контролю над подіями, що протікають у комп'ютерній системі чи мережі, та аналізу цих подій з метою пошуку ознак проблем безпеки. Оскільки кількість різних способів та видів організації несанкціонованих вторгнень у мережі останнім часом значно збільшилась, то системи виявлення вторгнень стали обов'язковою частиною безпекової інфраструктури для більшості організацій. Цьому сприяють як велика кількість літератури з цього питання, яку потенційні зловмисники уважно вивчають, так і більш витончені підходи до виявлення спроб проникнення в інформаційні системи та мережи.

Сучасні системи виявлення вторгнень мають різну архітектуру та характер, основними з яких є: мережна та локальна. Мережеві системи встановлюють на виділених для цього комп'ютерах так, щоб вони могли у подальшому проаналізувати трафік, що протікає по локальній мережі. Локальні системи розміщуються на тих комп'ютерах, які потребують захисту, і вивчають певні події (програмні виклики або дії користувача).

Крім архітектури системи виявлення вторгнень також можуть розрізняти за методикою виявлення: частина систем шукає аномальну поведінку, інша – зловмисне.

1.2.1 Методики виявлення аномального та зловмисної поведінки користувачів

Системи виявлення аномальної поведінки (від англ. anomaly detection) засновані на тому, що системи виявлення вторгнень відомі ознаки, що характеризують правильну або припустиму поведінку об'єкта спостереження. Під «нормальним» чи «правильним» поведінкою розуміються дії, виконувані об'єктом і які суперечать безпековій політиці.

Системи виявлення зловмисної поведінки (misuse detection) засновані на тому, що наперед відомі ознаки, що характеризують поведінку зловмисника. Найбільш поширеною реалізацією технології виявлення зловмисної поведінки є експертні системи (наприклад, системи Snort, RealSecure IDS, Enterasys Advanced Dragon IDS). Розглянемо більш докладніше технології, які використовуються у даних системах (рис. 1.1).

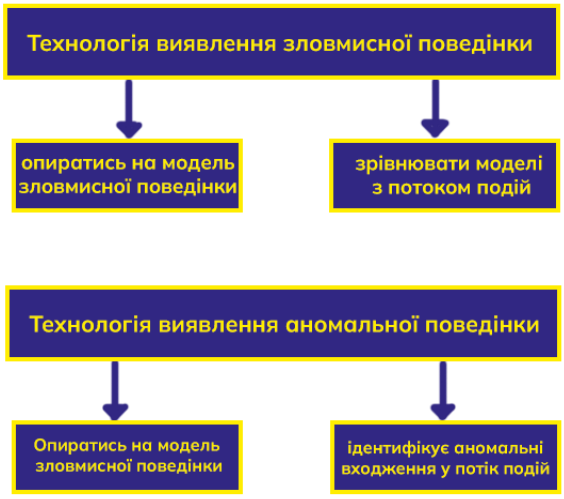


Рисунок 1.1 - Існуючі технології систем виявлення вторгнень

1.2.2 Технології виявлення аномальної діяльності

Датчики-сенсори аномалій ідентифікують незвичайну поведінку, так звані аномалії, у функціонуванні окремого об'єкта. Тому головна трудність у застосуванні їх на практиці пов'язана з нестабільністю самих об'єктів, що захищаються, а також й взаємодіючих з ними зовнішніх об'єктів. Як об'єкт спостереження може виступати мережа загалом, окремий комп'ютер, мережева служба (наприклад, файловий сервер FTP), користувач тощо. Датчики спрацьовують за умови, що напади відрізняються від «звичайної» (законної) діяльності. Тут варто відзначити, що в різних реалізаціях своє визначення припустимого відхилення для поведінки від дозволеного і своє визначення для «порога спрацьовування» сенсора спостереження [4].

Заходи та методи, що зазвичай використовуються у виявленні аномалії, включають наступні пункти:

– порогові значення: спостереження об'єктом виражаються як у вигляді числових інтервалів. Вихід за межі цих інтервалів вважається аномальною поведінкою. В якості об’єкта спостереження, можуть бути, параметри, наприклад: кількість файлів, до яких звертається користувач у даний період часу; число невдалих спроб входу в систему; завантаження центрального процесора тощо. Пороги можуть бути статистичними та динамічними (тобто змінюватися, підлаштовуючись під конкретну систему);

– параметричні: для виявлення атак будується спеціальний «профіль нормальної системи» на базі шаблонів (тобто є якась політика, як правило повинний притримуватись даний об'єкт);

– непараметричні: профіль будується на основі спостереження за об'єктом у період навчання;

– статистичні заходи: рішення про наявність атаки приймається за великою кількістю зібраних даних шляхом їх статистично попередньої обробки;

– міри на основі правил (сигнатур): вони дуже схожі на непараметричні статистичні заходи. У період навчання складається представлення про нормальну поведінку об'єкта, що записується у вигляді спеціальних правил [5]. Таким чином випливають сигнатури «нормальної» поведінки об'єкта;

– інші заходи*:* нейронні мережі, генетичні алгоритми, що дозволяють класифікувати деякий набір видимих сенсору-датчику ознак.

У сучасних системах виявлення аномалій переважно використовують перші два методи. Слід зазначити, що є дві крайності під час використання даних технологій:

– виявлення аномальної поведінки, яка не являє собою атаку, та віднесення її до класу атак (помилка другого роду);

– пропуск атаки, яка підпадає під визначення аномального поведінки (помилка першого роду). Цей випадок набагато небезпечніший, ніж помилкове віднесення до аномальної поведінки, до класу атак.

Тому при інсталюванні та експлуатації систем такої категорії звичайні користувачі та фахівці стикаються з двома досить неординарними задачами:

– визначення граничних значень характеристик поведінки суб'єкта для зниження ймовірності появи однієї з двох вищеописаних випадків;

– побудова профілю об'єкта – це завдання, що складно формалізується і вимагає багато часу і уваги, що також вимагає від спеціаліста безпеки великої попередньої роботи, високої кваліфікації та досвіду.

Як правило, системи виявлення аномальної активності використовують журнали реєстрації та поточну діяльність користувача як джерело даних для аналізу. До переваг систем виявлення атак на основі технології виявлення аномальної поведінки можна віднести те, що вони:

– не вимагають оновлення сигнатур та правила виявлення атак;

– здатні виявити нові типи атак, сигнатури для яких ще не розроблені;

– генерують інформацію, яка можна бути використати у системах виявлення зловмисної\аномальної поведінки [6].

Недоліками цих систем можна вважати наступне:

– генерують багато помилок другого роду;

– вимагають довгого та якісного навчання;

– зазвичай надто повільні у дії і вимагають значної кількості обчислювальних ресурсів [7].

1.2.3 Статистичний аналіз атак

Використання методів статистичного аналізу є найпоширенішим видом реалізації технології виявлення аномальної поведінки. Статистичні датчики збирають різну інформацію про типову поведінку об'єкта та формують її у вигляді профілю. У даному випадку профіль – це набір параметрів характеризуючих типову поведінку об'єкту. Він формується на базі статистики об'єкта, за яким введеться спостереження, із застосуванням методів математичної статистики (наприклад, методу сковзаючих вікон і методу зважування сум) [8].

Спочатку проходить період начального формування профілю, після цього дії об'єкта порівнюються з відповідними параметрами та при виявленні суттєвих відхилень подається сигнал про початок атаки. Параметри профілю можна систематизувати за поширеними групами:

– категоріальні параметри (назви файлів, команди юзера, відкриті порти тощо);

– числові параметри (кількість переданих даних за різними типами протоколів, навантаження центрального процесора, кількість файлів, до яких здійснювався доступ тощо);

– профіль, що не вписується в класифікацію нарівні з попередніми типами параметрів.

Також профілі мають механізми динамічних змін, для того щоб більш точно описувати поведінку об'єкта, у якому відбуваються зміни. Системи, що використовують статистичні методи, мають цілу низку переваг:

– не вимагають постійного оновлення банку сигнатур атак (це полегшує завдання супроводу даних систем);

– можуть адаптуватися до зміни поведінки користувача і тому більш чутливими до спроб вторгнення;

– можуть виявляти невідомі атаки, сигнатури для яких ще не написані і, отже, бути своєрідним стримуючим буфером доти, поки не буде розроблений відповідний шаблон для експертних систем;

– дозволяють виявляти складніші атаки, ніж інші методи, наприклад, розподілені по часу чи з об'єктів нападу [9].

Серед недоліків систем виявлення вторгнень можна назвати таке:

– у статистичних методах ймовірність отримання хибних повідомлень про атаку є набагато вищою, ніж при інших методах;

– статистичні методи не дуже вірно обробляють зміни у діяльності користувача (наприклад, коли менеджер виконує обов'язки підлеглого у критичній ситуації). Цей недолік може становити проблему в організаціях, де ці зміни є досить частими. В результаті можуть з'явитися як помилкові повідомлення про небезпеку, так і негативні помилкові повідомлення (пропущені атаки);

– система може сприймати діяльність, яка рівно відповідає атаці, як нормальну через свою адаптацію до нової поведінки, якщо зміни режиму роботи відбувались поступово;

– статистичні методи не в змозі виявити атаки з боку суб'єктів, яким неможливо описати шаблон типової поведінки;

– статистичні методи повинні бути попередньо налаштовані (мають бути задані граничні значення для кожного параметра, для кожного юзера);

– системи, побудовані виключно на статистичних методах, не можуть впоратись з виявленням атак з боку суб'єктів, котрі від самого початку виконують несанкціоновані дії, оскільки шаблон звичайної поведінки для них включатиме виключно атаки;

– статистичні методи на основі профілю нечутливі до порядку наслідковування подій [10].

Тим не менш, вже існують шляхи вирішення цих проблем, і їхня практична реалізація це лише питання часу. Очевидно, що статистичний метод є виключно реалізація технології аномальної поведінки.

1.2.4 Сигнатурний аналіз атак

Сигнатурний аналіз є досить поширеним методом виявлення підозрілої активності трафіку і ідентифікації раніше відомих атак. Цей підхід грунтується на зборі, та аналізу сигнатур атак, у яких загалом є унікальні властивості відомих шаблонів атак.

Сигнатури атаки являють собою певну послідовність байтів, які визначають конкретну атаку. Дана послідовність може мати в собі шаблони пакетів, ключів слова, значення заголовків, чи ідентифікуючі ознаки атаки. Сигнатури атаки зберігаються у базі даних (далі буде застосовано банк небажаних сигнатур), яка буде використовуватись під час аналізу мережевого трафіку.

Процес сигнатурного аналізу складає наступні кроки:

1) Збір сигнатур. Перед початком аналізу атаки чи атак варто створити банк небажаних сигнатур, який буде включати ключові дані, які будуть ідентифікуватись як загроза. Це може бути ручне виконання пошуку загроз, коли відповідальні особи досліджують відомі атаки і створюють словники сигнатур для їх виявлення. Також можуть бути використані автоматизовані системи, які самостійно аналізують інформацію і генерують відповідні сигнатури.

2) Виявлення сигнатур. В моменті аналізу трафіку система виявлення аномалій порівнює дані пакетів зі раніше збереженими сигнатурами атак. У випадку знаходження відповідної сигнатури, це бути означати, що у мережі може відбуватись раніше відома атака.

3) Інформування про можливу загрозу. У випадку виявлення підозрілого трафіку, система\програма може сповістити відповідальну особу про можливу атаку. Особа може вжити відповідні заходи для запобігання атаки, чи зменшення ризику впливу на систему [11].

Сигнатурний аналіз, як і має свої переваги, так і свої мінуси та обмеження. До переваг можна вказати про високу ефективність виявлення відомих атак та низьку кількість помилкових спрацювань. Але цей метод не здатний раціонально

виявляти нові атаки, які не були раніше відомі і не записані у словник сигнатур. Також існує ризик хибного пропуску атаки, якщо сигнатура не була записана до словника [12].

У практичній реалізації системи виявлення аномалій можуть використовуватись інші методи виявлення, механізми машинного навчання, додаткові алгоритми аналізу мережевого трафіку і.т.д.

Задля стабільної і безпечної роботи варто постійно оновлювати словники сигнатур.

1.3 Аналіз недоліків у сучасних системах виявлення вторгнень

З урахуванням всього сказаного вище, всі системи виявлення вторгнень можна визначити як системи, орієнтовані на пошук:

– сигнатур всіх відомих атак;

– аномалій взаємодії контрольованих об'єктів;

– спотворення оригінальної профільної інформації.

У ході роботи переважній частині сучасних систем використовуються лише сигнатурні методи розпізнавання атакуючих впливів або лиш пошук аномалій у поведінці відомої контрольованої мережі.

Також у немалій кількості відомих систем відсутній імітатор атак або любий інший засіб для перевірки вірність\правильність розгорнутої та експлуатованої системи виявлення аномалій (СВА), який би забезпечував простий і надійний засіб тестування конфігураційних параметрів, використаних у кожній конкретній комп'ютерній мережі [13].

Цей засіб, з логічних міркувань, повинен дозволяти імітувати діяльність вірусного ПЗ (наприклад, Bagle, CodeRed, MSBlast, NetSky), атак на відмову в обслуговуванні (наприклад, SYN-шторм), атак з метою підвищення привілеїв облікового запису (наприклад, вразливості в мережевих службах MS Internet Information Service 5.0, MS SQL Server 2000), атаку з метою перенаправлення

трафіку та нав'язування підробних даних (підміна ARP та нав'язування DNS служб). Бажано, щоб при цьому програмний засіб міг генерувати атаки розподіленого характеру.

Наприклад, архітектура деяких типів імітаторів СВА складається з набору агентів різних типів, спеціалізованих на вирішення підзавдань виявлення вторгнень. Агенти розміщуються на окремих комп'ютерах системи. У цій архітектурі відсутній «центр управління» сімейством агентів, тому що залежно від ситуації провідним може ставати будь-який із агентів, який ініціює функції кооперації та управління. У разі потреби вони можуть дублюватися у мережевому та локальному середовищі, або припинити своє функціонування. Залежно від ситуації (виду та кількості атак на комп'ютерну мережу, наявність обчислювальних ресурсів для реалізації функцій захисту), може знадобитися генерація кількох екземплярів агентів кожного класу. Передбачається, що архітектура цієї системи може адаптуватися до реконфігурації мережі, зміни трафіку та нових видів атак, використовуючи раніше накопичений досвід [14].

Загалом відсутність імітаторів атак для оцінки ефективності СВА не є основною проблемою. Реальними недоліками наявних систем виявлення є примітивність простого сигнатурного пошуку, низька ефективність при виявленні розподілених за часом і місцем складних атак, недостатня інтеграція інформації на рівні хосту та мережі для виявлення комбінованих атак та несанкціонованих проникнень.

В якості експлуатаційних недоліків можна відзначити велику кількість обчислювальних операцій для простого поділу належності подій на «свій-чужий» і неможливість обробки всієї інформації, яка надходить в реальному режимі часу на персональних комп'ютерах, так як швидкість обробки мережевого або іншого трафіку подій часто повільніше реального часу в 2 рази. Тому в деяких системах аналіз відбувається у відкладеному режимі. Що свідчить, що реалізація атаки на інформаційні та обчислювальні ресурси, що захищаються, не буде помічена вчасно і тим більше не буде відображена\оповіщена за допомогою наявних засобів захисту. У цьому режимі засоби виявлення атак можуть бути використані в

кращому випадку як засіб запису подій у хронологічному порядку всіх етапів атаки для подальшої експертизи [15].

2 ВИЗНАЧЕННЯ ЩО БУДЕ НОРМАЛЬНИМ, А ЩО АНОМАЛЬНИМ

Виявлення аномалій у поведінці мережі - рішення для виявлення аномалій поведінки мережі відстежують на наявність ненормальної поведінки, щоб виявити загрози та вжити заходів для їх усунення.

Однією з головних причин, що впливають на ефективність обчислювальної роботи мережі - є аномалії трафіку. Аномалії у трафіку мережі можуть бути викликані несправністю мережевого обладнання, випадковими чи навмисними діями з боку легітимних користувачів, невірною роботою додатків, діями зловмисників і т.д.

Виявлення аномалій поведінки мережі визначається як процес моніторингу мереж для виявлення аномальної поведінки [16]. Після виявлення аномалії функція виявлення аномалії поведінки мережі або ініціює автоматичну відповідь, або сповіщає групи безпеки (рис. 2.1).

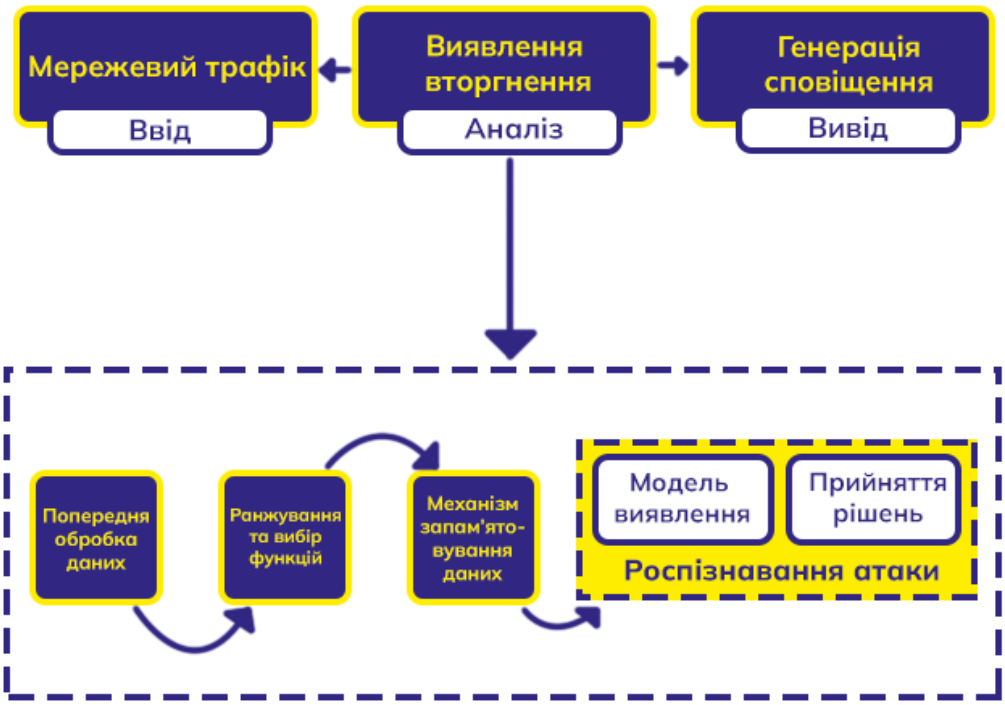


Рисунок 2.1 - Процес моніторингу мережі для виявлення аномальної поведінки

Виявлення аномалій поведінки мережі працює шляхом виявлення незвичайної поведінки мережі, наприклад, інтенсивний потік трафіку під час стану спокою [17].

Виявлення аномалій поведінки мережі виконується спільно із мережевими брандмауерами , програмним забезпеченням для моніторингу продуктивності мережі та іншими засобами. У той час як ці інші інструменти захищають мережу від відомих загроз, виявлення аномалій у поведінці мережі виявляє підозрілі дії, які можуть призвести до порушення роботи мережі через приховані зараження, крадіжку даних або інші шкідливі дії.

Поєднання можливостей сигнатури та виявлення аномалій дозволяє виявляти аномалії поведінки мережі для подальшого дослідження незвичайної мережевої активності. Характеристики мережі, які відстежуються програмами виявлення аномалій поведінки мережі, включають пакети, пропускну здатність, обсяг байтів, обсяг трафіку та використовуваний протокол [18]. Будь-яка підозріла подія реєструється у звіті та містить IP-адреси джерела та призначення, відповідні порти, протоколи, мітки часу тощо.

Ці критичні показники та багато інших відстежуються інструментами виявлення аномальної поведінки мережі в режимі реального часу. Сповіщення про аномалію спрацьовує, якщо виявляється дивна тенденція або викид, який може вказувати на присутність можливої загрози. Залежно від обраної конфігурації, програми виявлення аномалій у поведінці мережі. Також можуть контролювати поведінку окремих користувачів мережі [19].

Рішення для виявлення аномалій поведінки мережі працюють, «підмітаючи» всю мережу під час пошуку загрозливих чи підозрілих суб’єктів. Це помітна відмінність від систем безпеки периметра, брандмауера та кінцевої точки.

Ці рішення здатні виявляти лише загрози, що передаються через певну окрему ділянку мережі, де вони налаштовані. І навпаки, виявлення аномалій у поведінці мережі враховує три важливі властивості мережі — моделі потоку трафіку, аналіз пасивного трафіку та дані про продуктивність мережі — для виявлення кількох різних типів загроз, як-от:

– невідповідна поведінка мережі. Наприклад неавторизовані програми (кряк) або дивне використання портів відомою програмою. Виявляючи таку активність, рішення для виявлення аномалій у поведінці мережі та відповідні системи захисту можуть автоматично ідентифікувати та вимикати пов’язані мережеві процеси та повідомляти відповідний персонал безпеки;

– викрадання даних. Як-от передача підозріло великого обсягу даних. У разі виявлення такої активності засобів виявлення аномалій, поведінки мережі та відповідні рішення безпеки можуть автоматично контролювати вихідну передачу даних і повідомляти про це командам чи апарату безпеки в режимі реального часу. Деякі системи можуть визначати пункт призначення цих передач даних і визначати, чи це легітимна комунікація чи подія кібербезпеки;

– приховані загрози. Як-от вдосконалене зловмисне програмне забезпечення . Виявлення аномалій у поведінці мережі працюватиме з іншими рішеннями безпеки для розгортання відповідних заходів безпеки. Він сигналізує відповідальним особам чи засобам протидії аномаліям про виявлення загрози, яка могла ухилитися від безпеки периметра та проникнути в мережу [20].

Таким чином, для надійної передачі даних у мережі, можуть бути вжиті заходи щодо своєчасного виявлення аномалії, пошуку її джерела або джерел та вжиття заходів щодо її усунення (повідомлення про несправність, фільтрацію аномального трафіку тощо). Отож, для забезпечення надійної передачі даних у мережі, велике значення набуває розробка нових методів виявлення аномалій та заходи щодо її усунення.

На сьогоднішній день одними з найпоширеніших засобів, що використовуються для виявлення аномалій, є засоби виявлення атак. Дані засоби ідентифікують підозрілу (аномальну) активність, спрямовану на обчислювальні або мережеві ресурси та реагують на неї.

Однак жоден із існуючих засобів виявлення атак не здатний повністю виявляти аномальну активність у трафіку обчислювальній мережі. За статистикою близько 80% порушень відбуваються внутрішніми порушниками, тобто. працівниками організації. Засоби виявлення атак, що використовуються,

малоефективні при виявленні негативних впливів з боку внутрішніх зловмисників. Загалом можна виділити такі недоліки засобів виявлення атак на обчислювальну мережу:

– висока вартість комерційних систем виявлення атак;

– велика кількість помилкових спрацьовувань, а також високий відсоток пропуску реальних атак на обчислювальні мережі;

– слабкі можливості для виявлення нових та видозмінених атак;

– проблеми щодо джерела порушення і цілей атакуючого у разі антропогенної загрози;

– неможливість визначення деяких порушень на початкових етапах;

– великі вимоги до обчислювальних ресурсів систем, які працюють у режимі реального часу;

– висока кваліфікація експертів щодо виявлення атак, необхідна при впровадженні систем виявлення атак.

Мережевий трафік можна розглядати як сукупність всіх даних, які передаються через мережу. Цей трафік може бути аномальним або нормальним, залежно від того, наскільки він відповідає очікуваному шаблону який зберігається локально, або у БД і.т.д. під час роботи мережі [21].

Сигнатурний метод є одним зі способів виявлення аномального трафіку. Він базується на використанні підписів (сигнатур) для ідентифікації конкретних видів трафіку. Ці підписи можуть бути створені на основі попередньо відомих даних про трафік, які можуть включати типи пакетів, що передаються, їх розміри, шаблони взаємодії між комп'ютерами, тощо [22].

Нормальний мережевий трафік, який відповідає очікуваному шаблону, може бути визначений на основі попередньо зібраних даних про трафік. Наприклад, якщо звичайний трафік містить пакети певних розмірів, які передаються між певними комп'ютерами зі стандартними протоколами взаємодії, то такий трафік можна визначити як нормальний. Сигнатури, які використовуються для ідентифікації цього трафіку, можуть бути створені на основі цих даних [23].

Аномальний трафік, з іншого боку, не відповідає очікуваному шаблону і може включати пакети незвичайних розмірів, незвичайні типи протоколів взаємодії, або просто викликати підозру через свою велику кількість. Сигнатури для ідентифікації аномального трафіку можуть бути створені на основі відомих прикладів такого трафіку.

2.1 Виявлення аномального трафіку

Для виявлення аномального трафіку на базі сигнатурного метода, можуть використовуватися алгоритми, які порівнюють поточний трафік з попередньо відомими сигнатурами. Ці алгоритми можуть бути побудовані на основі різних методів, таких як регулярні вирази, маски, правила узгодження тощо.

Одним з таких методів є метод "хешування", який використовує швидку ідентифікацію трафіку. У цьому методі, попередньо збережені сигнатури зберігаються у вигляді хеш-таблиці, а потім порівнюються з хешем поточного трафіку. Якщо знайдено відповідний запис у таблиці, то цей трафік визначається як нормальний [24]. Інакше, якщо жодного відповідного запису не знайдено, трафік може бути визначений як аномальний.

Інший метод, який можна використати для виявлення аномального трафіку, це метод машинного (автоматизованого) навчання. У цьому методі, система використовується для навчання на відомих прикладах нормального трафіку, і використовує ці знання для виявлення аномальних випадків. Процес навчання може використовувати різні алгоритми, такі як нейронні мережі, дерева рішень, наївний Баєс, тощо.

Інший метод, який можна використати для виявлення аномального трафіку, це метод машинного (автоматизованого) навчання. У цьому методі, система використовується для навчання на відомих прикладах нормального трафіку, і використовує ці знання для виявлення аномальних випадків. Процес навчання

може використовувати різні алгоритми, такі як нейронні мережі, дерева рішень, наївний Баєс, тощо [25].

Виявлення на основі сигнатур зазвичай найкраще використовувати для визначення відомих загроз. Він працює за допомогою попередньо запрограмованого списку відомих загроз та їх індикаторів компрометації. Це може бути певною поведінкою, яка зазвичай передує зловмисній мережевій атаці, хешам файлів, шкідливим доменам, відомим послідовностям байтів або навіть вмісту заголовків тем електронних листів. Оскільки система виявлення вторгнень на основі сигнатур відстежує пакети, що проходять мережею, вона порівнює ці пакети з базою даних відомих або сигнатур атаки, щоб позначити будь-яку підозрілу поведінку.

Одним з вагомих аспектів сигнатурного методу є постійне оновлення банку небажаних сигнатур. Якщо банк сигнатур не оновлюється, або оновлюється лише по ситуації, то сигнатурний метод може не вбачити аномалії нового типу, які не були раніше відомі [26]. Тому важливо регулярно відслідковувати нові типи атак та додавати їх сигнатури до банку сигнатур.

Крім того, важливо розуміти, що сигнатурний метод можна обманути, в моменті коли зловмисники використовують нові або нестандартні методи атак, які не входять до банку сигнатур. Такі атаки можуть бути невиявленими, і потребують додаткового аналізу.

Система аналізу повинна забезпечувати захоплення 100% трафіку та надавати ефективні методи аналізу з навігацією за результатами [27].

Якщо говорити про комплексне вирішення задачі аналізу мережевого трафіку, то в першу чергу варто розділити її на три достатньо незалежні підзадачі (рис. 2.2): перехоплення трафіку, його зберігання та аналіз для подальших рішень роботи з ним.

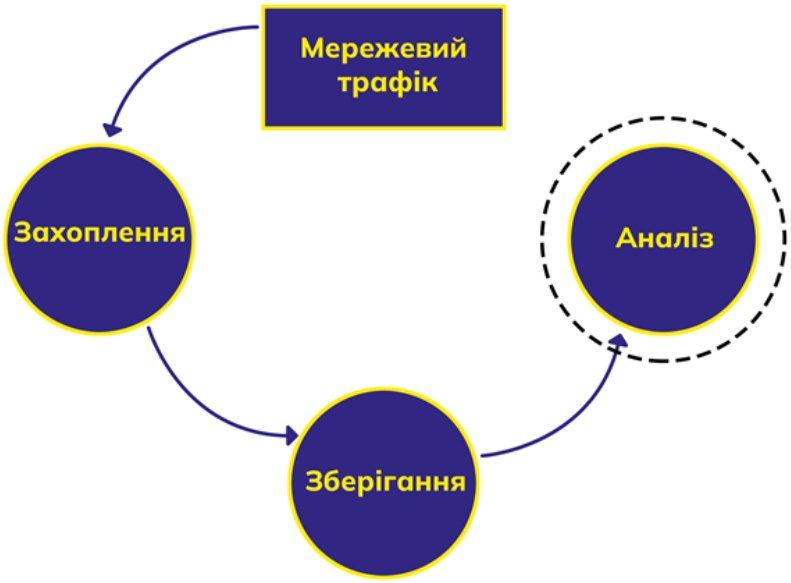


Рисунок 2.2 – Підзавдання системи аналізу мережевого трафіку

2.1.1Перехоплення трафіку

Захоплення трафіку здійснюється за допомогою сніфферів. У загальному випадку, сніффер (англ. sniffer - дослівно перекладається як нюхач або винюхувач) - це програма або програмно-апаратний пристрій, призначений для перехоплення трафіку.

Перехоплення мережевого трафіку може здійснюватися:

– за допомогою "прослуховування" мережевого інтерфейсу;

– підключенням сніффера у розрив каналу;

– відгалуженням («дзеркалюванням») трафіку та направлення його копії на сніффер;

– за допомогою аналізу побічних електромагнітних випромінювань;

– через атаку на канальному чи мережевому рівні, що призводить до перенаправлення трафіку жертви на сніффер [28].

Розрізняють два види сніферів за їх місцезнаходженням:

– на маршрутизатори (шлюзі);

– на кінцевому вузлі мережі.

У першому випадку буде перехоплюватися весь трафік, що проходить через інтерфейси пристрою, у другому - або тільки трафік вузла мережі, якщо мережна карта працює в нормальному режимі, або пакети всіх пристроїв цього сегмента мережі) [29].

Створюються такі програми, ґрунтуючись на бібліотеці Pcap, що вільно розповсюджується (англ. «packet capture»). Вона призначена для використання спільно з мовами C/C++, а для роботи з бібліотекою іншими мовами, як-от Java, .NET, використовують обгортки. Для Unix-подібних систем це бібліотека libpcap, а Microsoft Windows – WinPcap [30].

Програмне забезпечення мережного моніторингу може використовувати libpcap або WinPcap, щоб захопити пакети у мережі, передачі пакетів у мережі. Крім того, підтримується збереження захоплених пакетів у файл і читання файлів, що містять збережені пакети.

Кінцева схема підключення зображена на рисунку 4

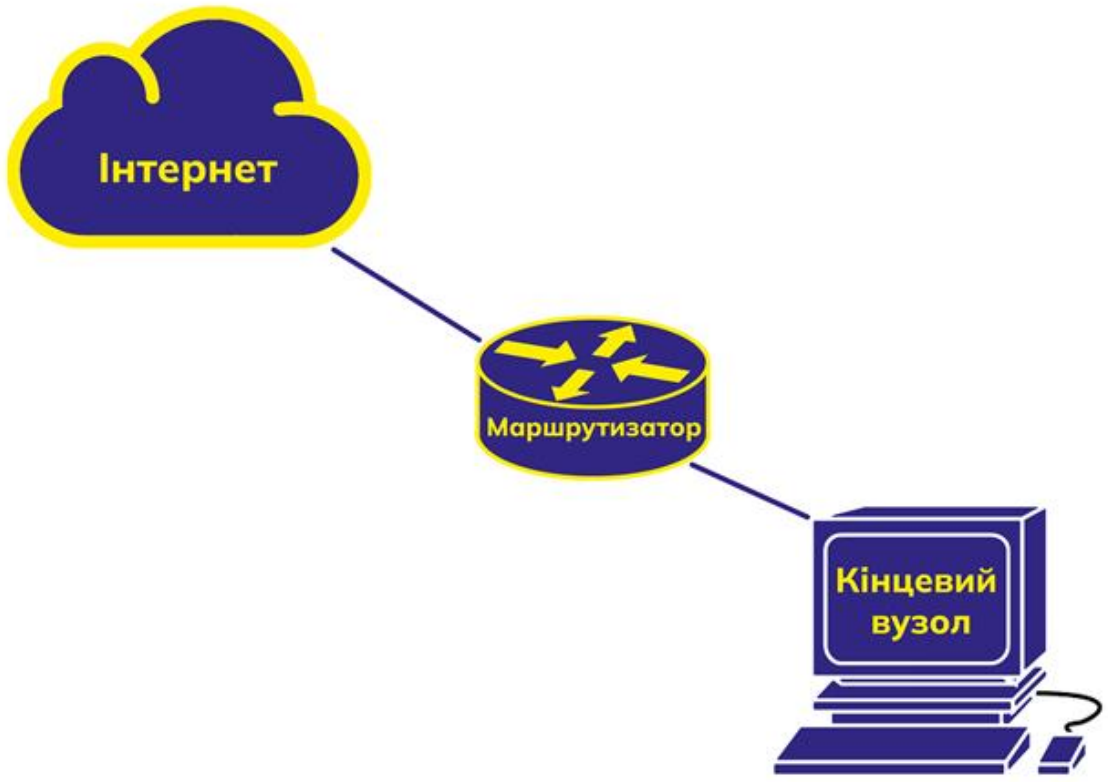


Рисунок 2.3 – Схема збору інформації

2.1.2 Аналіз трафіку

Більшість існуючих інструментів, зазвичай, проводить розбір заголовків мережевих протоколів, і відновлює сесії (базовий аналіз). У той же час існують досить специфічні завдання, для яких може не виявитися готовий інструмент, наприклад:

– аналіз тунельованих протоколів довільної глибини;

– аналіз сесій на рівні додатків (виділення зв'язків між потоками даних, що передаються по мережі);

– виконання певних сценаріїв (скриптів) у разі виявлення у трафіку попередньо заданих сигнатур.

Виділяють два режими роботи аналізаторів:

– у реальному часі;

– за попередньо збереженим трафіком.

Аналіз у реальному часі вимагає підтримки роботи інструментів в безперервному режимі з продуктивністю, достатньою для аналізу трафіку, що надходить на вхід. У цьому має бути забезпечена можливість обробки потенційно нескінченного вхідного потоку даних.

У разі відкладеного аналізу інструмент отримує вхідні дані з файлу, що дозволяє проводити більш детальний аналіз мережевого шляху на відміну з аналізом у режимі реального часу на аналогічному трафіку.

2.2 Структура програми для обліку мережевого обліку

2.2.1 Архітектра бібліотеки Winsock

Winsock – бібліотека, розроблена компанією Microsoft задля реалізації мережевих додатків які підтримуються ОС Windows. Вона побудована на основі моделі Windows Open System Architecture (WOSA) (див. таблиця 1) і дає змогу розробляти власні сервіс-провайдери (мережеві служби), які діютимуть на усі

програми, що використовують Winsock API. При цьому немає необхідності переписувати код цих програм або замінювати Winsock 2 DLL (Ws2_32.dll). Для розробки сервіс провайдерів Winsock включає Service Provider Interface (SPI). (табл. 2.1)

Таблиця 2.1 - Модель WOSA

Користувацький додаток	Користувацький додаток
Winsock 2 API	
Winsock 2 DLL WS2_32.DLL	
Транспортні функції Winsock 2 SPI для транспортних сервіс-провайдерів	Функції для роботи з простором імен (namespace) Winsock 2 SPI для роботи з простором імен
Транспортний сервіс-провайдер	Сервіс-провайдер простору імен

Winsock 2 SPI дає змогу розробляти два типи сервіс-провайдерів:

– транспортні сервіс-провайдери;

– сервіс-провайдери простору імен.

Транспортні послуги провайдери виконують функції, відповідальні за встановлення з'єднання, передачу даних, управління трафіком і контроль кількості помилок. Сервіс провайдери простору імен надають функції роздільної здатності імен [31].

Транспортні сервіс-провайдери в свою чергу діляться на:

– базові (base service providers);

– багаторівневі (layered service providers)

Базові реалізують низькорівневі функції транспортного протоколу, багаторівневі виконують функції вищого рівня і опираються на базовий сервіс-провайдер нижчележачий базовий сервіс. (див. рис.2.4)

Базові сервіс-провайдери і провайдери простору імен зазвичай поставляються виробниками операційних систем. Розширення базових функцій можливе з поміччю додаткових багаторівневих сервіс-провайдерів. Але варто звернути увагу, що скористатися цією функціональністю зможуть лише програми, які використовують Winsock API.

Кожен транспортний сервіс-провайдер підтримує один чи декілька протоколів. Наприклад, TCP/IP повинен підтримувати, як мінімум, протоколи TCP і UDP, IPX/SPX провайдер - IPX, SPX і SPX II. Кожен протокол, який підтримує конкретний провайдер, описується структурою WSAPROTOCOL_INFOW, а набір таких структур може бути представлений як каталог відомих протоколів.

Багаторівневі та базові послуги сервіс-провайдерів об'єднуються і утворюють ланцюг протоколу. Для опису ланцюгу загалом можна також використати структуру WSAPROTOCOL_INFOW.

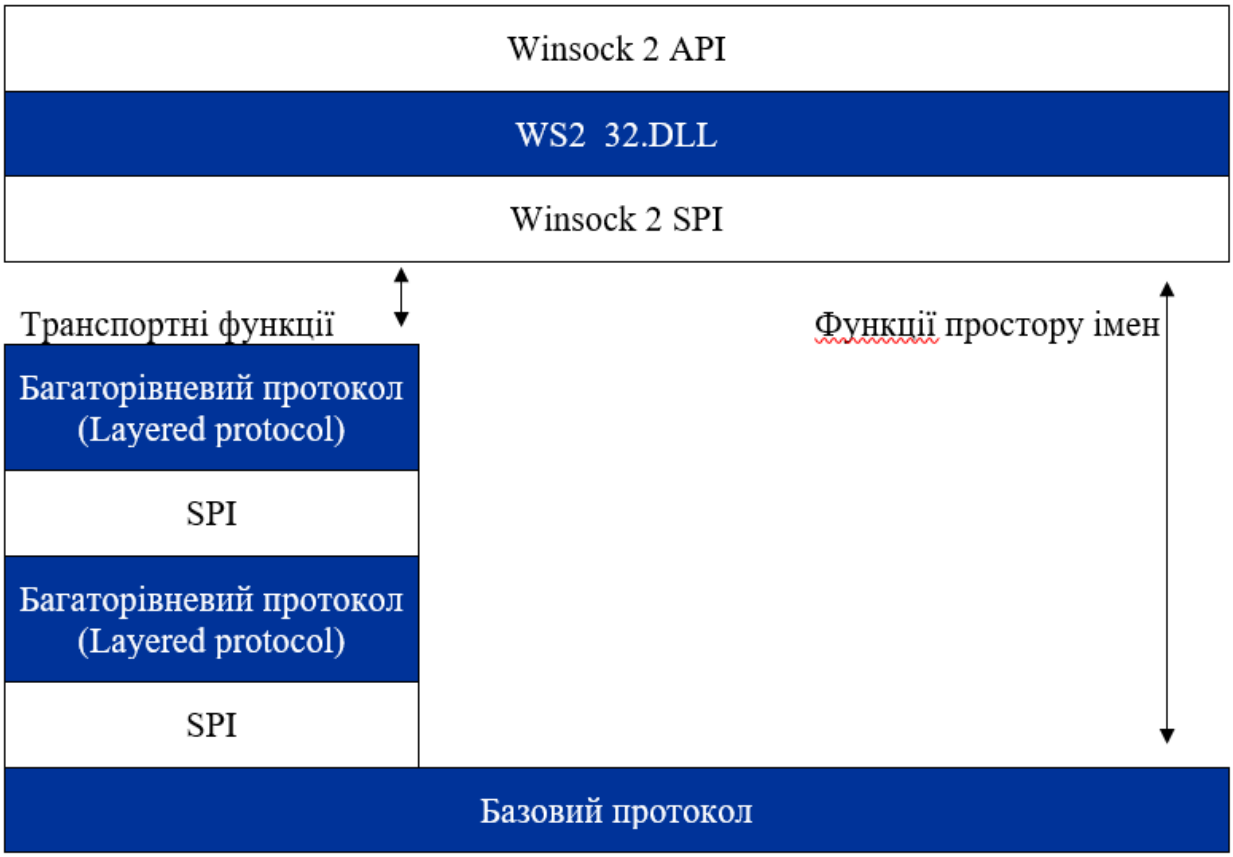


Рисунок 2.4 – Схема роботи базового сервіс-провайдера

2.2.2 Структура багаторівневого сервіс-провайдера

Будь-який багаторівневий сервіс-провайдер є динамічною бібліотекою з однією експортованою точкою входу – ініціалізуючою точкою входу, яке є функцією: WSPStartup(). Доступ до інших функцій сервіс-провайдера Ws2_32.dll отримує через таблицю адрес функцій (dispatch table), що надає провайдер. SPI також дозволяє транспортним сервісам-провайдерам здійснювати виклик функції Ws2_32.dll. Для цього Ws2_32.dll передає свою таблицю адрес функцій (upcall dispatch table), як параметр в ініціалізуючу функцію [33].

Структура WSPPROC_TABLE містить функції, які мають бути реалізовані в багаторівневому сервіс-провайдеру та адреси, які використовуються для заповнення вихідного параметра функції WSPStartup(). Кожен багаторівневий сервіс-провайдер повинен реалізовувати 30 функцій. Але здебільшого повна реалізація необхідна лише деяких із них. Усередині інших функцій буде просто передати виклик нижчележачому сервіс провайдеру.

Динамічна бібліотека сервіс-провайдера завантажується іншим провайдером, або бібліотекою Ws2_32.dll (залежно від положення провайдера у ланцюзі). Відразу після завантаження викликається функція WSPStartup(). Вона має зданість викликатися кілька разів. Для кожного успішного виклику WSPStartup() - буде викликано WSPCleanup(). Тому кожен сервіс-провайдер повинен мати лічильник посилань, який збільшуватиметься на одиницю при виклику WSPStartup() і зменшуватиметься при виклику WSPCleanup(). Якщо цей лічильник стане рівним нулю, сервіс-провайдер повинен підготуватися до розвантаження.

В моменті коли програма користувача викликає будь-яку Winsock API-функцію, Ws2_32.dll викликає відповідну Winsock SPI-функцію. Але не всі Winsock API функції мають відповідні Winsock SPI функції. Деякі з них реалізовані Ws2_32.dll і їх виклики не передаються сервіс-провайдеру. Також не передаються сервіс-провайдерам виклики для роботи з об'єктами синхронізації та функції очікування: вони безпосередньо передаються службам Windows.

2.2.3 Встановлення сервіс-провайдера

Для того, щоб функціональність, що надається сервіс-провайдером, стала доступною, він повинен бути коректно встановлений та зареєстрований бібліотекою Winsock. Відповідно використовується окрема інсталяційна програма, яка зазвичай надається розробником сервіс-провайдера. Інсталяційна програма додає необхідну інформацію до каталогу Ws2_32.dll. Для цього використовується функція WSCInstallProvider(). За допомогою WSCDeinstallProvider() викликавши цю інформацію можна видалити з каталогу Winsock.

Структура WSAPROTOCOL_INFOW, яка надається кожним провайдером під час інсталювання, містить інформацію про те, якого типу цей провайдер - базовий, багаторівневий чи це ланцюг протоколів. Інсталяція ланцюга протоколів можливе виключно після вдалої інсталяції всіх його компонентів.

2.2.4 Обробка операцій вводу і виводу

Для обліку трафіку за допомогою сервіс-провайдера необхідно певним чином втілити функції, що відповідають за операції вводу/виводу. Це такі функції як WSPRecv(), WSPRecvFrom(), WSPSend(), WPSendTo() та WSPIoctl().

2.2.5 Блокуючі операції вводу і виводу

Бібліотека Winsock підтримує три види операцій вводу і виводу. Найпростіший із них – блокуючий, він використовується за замовчуванням. Блокуюча операція вводу і виводу повертає керування виключно після її завершення. Тому кожен потік може одночасно виконувати лише одну задану операцію. До прикладу, якщо потік в даний момент приймає дані, але зараз нічого

не було надіслано, потік блокується. Цей вид операцій вводу і виводу простий, але не завжди ефективний.

2.2.6 Неблокуючі операції вводу і виводу

Якщо сокет переведений в неблокуючий стан, то будь-яка операція відразу виконуватиметься або повертатиме код помилки WSAEWOULDBLOCK, що означає, що операція не може бути завершена негайно. В останньому випадку необхідна опція, яка б дозволяла б визначити момент, у який потрібно викликати операцію повторно, щоби вона коректно завершилася. Для цього було визначено набір мережевих подій. Для перемикання сокету в режим асинхронної доставки мережевих подій використовуються функції WSPAsyncSelect() та WSPEventSelect(). Після цього можна використати функцію WSPSelect() для визначення стану сокету.

2.2.7 Відкладенні операції вводу і виводу

Winsock 2 підтримує також відкладений ввід і вивід (overlapped) і потребує, щоб всі транспортні сервіс-провайдери його підтримували. Операції відкладеного 11 вводу і виводу можна реалізовувати над сокетами, які були створені за допомогою WSPSocket() зі вказаняям прапора WSA_FLAG_OVERLAPPED.

Для отримання даних клієнт використовує WSPRecv() або WSPRecvFrom(). Якщо виклик було зроблено до надходження даних, вони відразу поміщаються в буфер клієнта, що дає можливість уникнути зайвої операції копіювання. У разі надходження даних до буфера, сервіс провайдер реалізовує синхронний стиль операцій, при умові що вхідні дані зберігаються у внутрішньому буфері, поки клієнт не викличе операцію WSPRecv() або WSPRecvFrom().

Для надсилання даних застосовуються функції WSPSend() і WSPSendTo(), в якості параметрів які клієнт передає покажчик на буфер із даними. При цьому

мається на увазі, що вміст буфера не буде змінюватись, поки дані не будуть надіслані.

WSPIoctl() застосовується для виклику розширених операцій вводу і виводу.

Відкладені операції надсилання та отримання даних, повертають керування одразу. Якщо вони повертають 0, це означатиме, що операцію було завершено коектно. Тобто пов'язаний з цією операцією об'єкт типу «подія» переведений у сигнальний стан, або завершальна процедура поставлена в чергу. Якщо операція повертає SOCKET_ERROR, і при цьому код помилки дорівнює WSA_IO_PENDING, це означає, що операцію було успішно розпочато, далі буде передано сповіщення про відправку або отримання даних. Будь-який інший результат сигналізує про помилку.

2.2.8 Порти завершення операцій вводу і виводу

Порт завершення вводу і виводу – це механізм, за допомогою якого програма може застосовувати пул потоків (pool of threads) для обробки запитів на асинхронний ввід і вивід (asynchronous I/O requests). Цей спосіб є найбільш ефективним.

Один порт завершення вводу і виводу можна зв'язати з одним або більше дескриптором файлу (або сокету) за допомогою виклику CreateIoCompletionPort(). Коли відкладена операція вводу і виводу завершується, в чергу до порту надсилається пакет з інформацією про завершену операцію (I/O completion packet). Таким чином цей механізм може бути застосований для створення однієї точки синхронізації для декількох або більше дескрипторів об'єктів.

Керуючий потік може застосовувати функцію GetQueuedCompletionStatus() для очікування пакета з інформацією про успішно завершену операцію. Потоки блокуються в черзі на шляху до порту, на завершення і звільняються по порядку LIFO (last-in-first-out): коли надходить пакет завершення, система запускає останній заблокований потік.

2.2.9 Схема аналізатора мережевого трафіка

Схема роботи додатка для сканування мережевого трафіку представлена на рисунку 5. В якості прикладу наведено порядок передачі і обробки виклику recv().

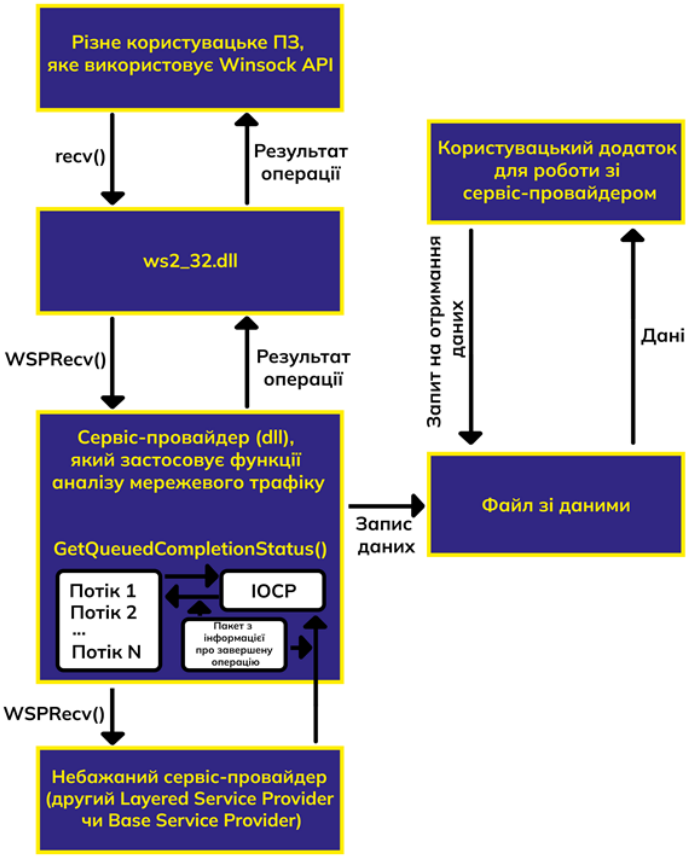


Рисунок 2.5 – Схема роботи аналізатора трафіка

2.2.10 Зберігання і подальша обробка динамічної інформації

При виборі методу зберігання динамічної інформації було розглянуто три варіанти:

– база даних;

– XML-файл;

– CSV-файл.

Отже, відповідно до поставленої задачі, був вибраний третій спосіб, у зв’язку з тим що структура, яку потрібно зберігати досить проста. Використання банку небажаних сигнатур CSV забезпечить більш чим достатній простір для роботи з наявними даними. Зберігання статистики у XML-файлі в даному випадку не раціональне, оскільки цей формат складніший і вимагає більше накладних витрат.

Запис у банк небажаних сигнатур забезпечує сервіс-провайдер при закриті сокета. При цьому зберігаються наступні дані:

– час;
– IP-адреса і порт джерела;
– IP-адреса і порт призначення;
– тип протокола, по якому була передана інформація;
– кількість переданих байт;
– кількість отриманих байт.

Для обробки файлу з отриманою інформацією від джерела, усередині користувацького додатку був реалізований кінцевий автомат-механізм, який змінює текстову інформацію в більш зручну для обробки структури. Відповідно, отримана структура висвітлюється в зрозумілій користувачу формі.

Також одна з переваг CSV формату є досить великий діапазон підтримуючих середовищ розробки\мов програмування, які в більшості вже мають в собі підтримку роботи з CSV файлами. Це значно спрощує процес зчитування та запису даних з банку сигнатур.

2.3 Висновок

Було проведено дослідження, та розроблено концепцію роботи програми виявлення аномалій на базі сигнатурного аналізу. Один з важливих аспектів цього дослідження є визначення того, що можна вважати «нормальним» трафіком, а що можна вважати «аномальним» трафіком.

У процесі аналізу було розглянуто властивості та характеристики трафіку, такі як обсяг передачі даних, часові інтервали, джерела та призначення пакетів

і.т.д. Було виявлено що нормальний трафік, загалом відповідає певним статистичним закономірностям, а також шаблонам, які можна використати для ідентифікації.

Висновок базується на тому, шо сигнатурний аналіз атак є ефективним інструментом для виявлення підозрілої активності мережі. Даний метод дозволяє відповідальним особам оперативно реагувати на потенційні загрози, та приймати відповідні заходи для захисту від аномалій чи шкідливих дій.

Але варто зазначити, що сигнатурний аналіз має свої обмеження. Він базується на раніше відомих сигнатурах та шаблонах, тому може некоректно виявляти нові атаки, чи виконувати хибні спрацювання. Також великий обсяг даних та швидкоплинний трафіку може виявлятися як аномальна поведінка.

3 РЕАЛІЗАЦІЯ ДОДАТКА ДЛЯ СКАНУВАННЯ МЕРЕЖЕВОГО ТРАФІКУ

Запропонований у даній роботі програмний продукт реалізований у вигляді 3-х компонентів:

– windows-додатка для інсталяції сервіс-провайдера;

– динамічної бібліотеки сервіс-провайдера;

– windows-додатка призначеного для керування сервіс-провайдером.

Для реалізації даного засобу сканування трафіку використовувалось середовище Microsoft Visual Studio 2022, де виконання програми було здійснено на мові C++. А також SQL Server Management Studio (SSMS), для реалізації банку небажаних сигнатур у якій мають зберігатись дані про вхідний, а також вихідний трафік [34].

3.1 Додаток для інсталяції сервіс-провайдера

Загальна схема працездатності додатка зображена на рисунку 3.1.

Рисунок 3.1 – Схема роботи додатка для інсталяції сервіс-провайдера

Вхідними даними для цього додатку, є ідентифікатори сервіс-провайдерів, поверху яких потрібно інсталювати динамічну бібліотеку. Вони залежать від конкретних версій операційних систем, а також від раніше інстальованих сервіс-провайдерів. Тому найбільш прийнятним варіантом в даній ситуації є представлення користувачу можливість вибирати самому сервіс-провайдери, поверх яких важливо інсталювати динамічну бібліотеку. Для цього інсталяційний додаток в загальному виводить список встановлених сервіс-провайдерів з коротким змістом про них, з якого користувач має можливість вибрати необхідне йому.

Після отримання інформації, додаток розпочинає інсталяцію сервіс-провайдера. Спочатку динамічна бібліотека інсталюється як закритий елемент каталога за допомогою функції WSCInstallProvider(). При цьому у структурі WSAPROTOCOL_INFLOW dwProviderFlags має тримати у собі біт PFL_HIDDEN. Це необхідно для отримання ідентифікатора елементу каталогу, якому буде відповідати сервіс-провайдер. Після цього ідентифікуються зв’язки провайдерів. Для цього також може використовуватись функція WSCInstallProvider. При цьому для кожного зв'язку потрібно згенерувати власний загальний (глобальний) ідентифікатор. Після вдалої інсталяції зв’язку, які містять провайдер, вміщуються у початок каталогу з допомогою функції WSCWriteProviderOrder() [35].

Знищення сервіс-провайдера також може виконувати інсталяційний додаток. Деінсталяція виконується у два етапи:

– спочатку видаляються усі зв’язки, у які входить провайдер;

– наступним кроком, видаляється сам сервіс-провайдер.

Це робиться з допомогою функції WSCDeinstallProvider(). При цьому якщо у системі існують провайдери, інстальовані поверх, то їх теж варто реінсталювати. Для цього вони спочатку мають видалитись, а вже після корекції зв’язків заново інсталюються з допомогою функції WSCInstallProvider() [36].

3.2 Динамічна бібліотека, яка реалізує функції керування трафіком

Основа функціональність здатність бібліотеки задана в реалізації основної функції WSPRecv(), WSPRecvFrom(), WSPSend(), WSPSendTo(), а також WSPloct(). Усі вони побудовані по одній схемі (рис. 3.2, 3.3).

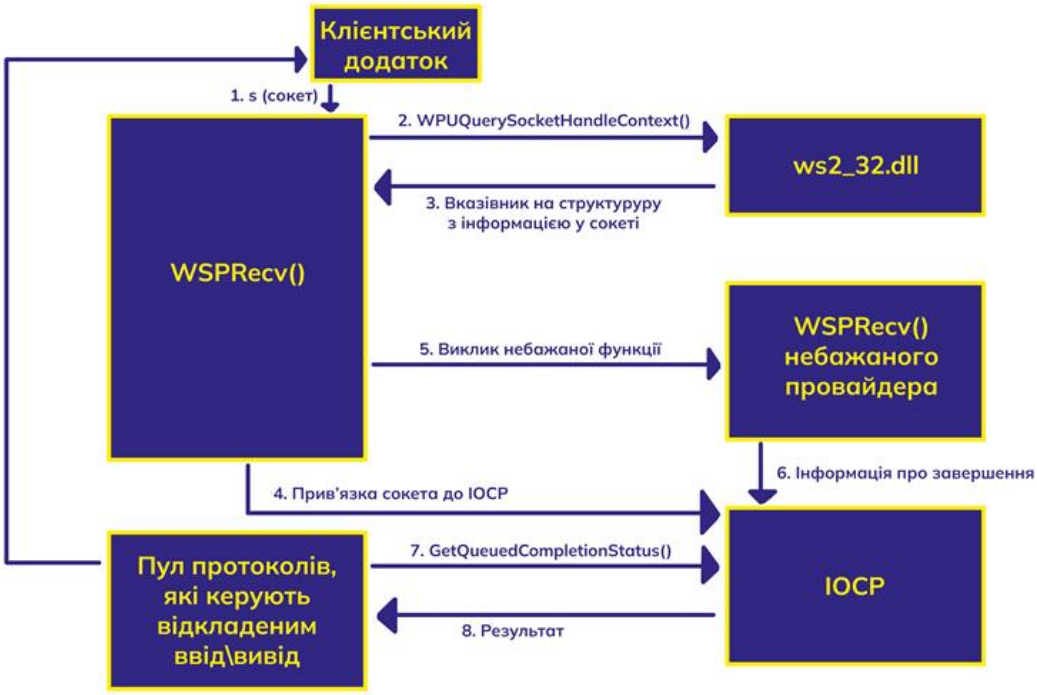


Рисунок 3.2 – Схема обробки виклику recv()

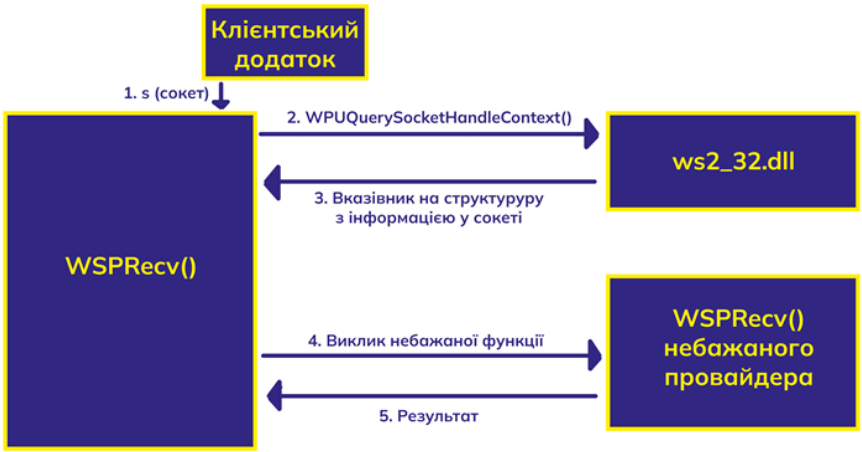


Рисунок 3.3 – Схема обробки запиту recv() у випадку блокуючого чи неблокуючого вводу\виводу

Усі необхідні дані про сокет зберігаються у спеціальній структурі. При закриті сокета додатком, тобто при виклику WSPSocket(), дані з структури зберігаються у банк небажаних сигнатур, розташовано на локальному серверу (на

комп’ютері). При цьому якщо файл ще не існує, він створюється в у нього записується заголовок. Якщо ж файл уже був створений раніше, дані просто доповнюються [37].

Інформація котра надходить, записується у наступному порядку:

– час;
– IP-адрес джерела;
– порт джерела;
– IP-адрес призначення;
– порт призначення;
– тип сокета (TCP, UDP і.т.д)
– кількість переданих байт;
– кількість отриманих байт.

3.3 Реалізація програмного коду додатку

3.3.1Увімкнення нерозбірливого режиму

Нерозбірливий режим – це спеціальний режим, завдяки якому мережевий адаптер перехоплює майже усі пакети, які повинні проходити через нього самого, незалежно від того, до кого ці пакети були адресовані. Це дає змогу реалізовувати детальний аналіз всього проходящого через нього трафіку (рис. 3.4) .

Рисунок 3.4 – Схема роботи процес налагоджування нерозбірливого режиму

У даній частині, використовується функція ioctlsocket(), для інсталяції параметрів сокета. Константа SIO_RCVALL, подана як другий аргумент, що вказує, що налаштування, які ми хочемо інсталювати, пов’язані з нерозбірливим режимом.

Третій аргумент ioctlsocket() містить у собі адресу змінної flag, яка вказує, чи варто увімкнути нерозбірливий режим (у прикладі змінна називається RS_Flag). Значення 1, що передається на змінну flag, вказує, що варто увімкнути нерозбірливий режим.

Після застосування нерозбірливого режиму, для виконання операцій вводу-виводу можна застосувати, функції мультиплексування, такі як select (), poll() або WSAAsyncSelect(). Ці функції дають змогу вибрати набір сокетів, які очікують ввід-вивід.

3.3.2 Створення сирого сокета

Створення сирого сокету – це тип сокета, який дає змогу взаємодіяти з мережевими рівнями протоколів безпосередньо, минуючи вищі рівні, такі як наприклад TCP або UDP (рис. 3.5).

Рисунок 3.5 – Схема створення сирого сокета

У прикладі наведеному вище, функція socket() викликається зі трьома аргументами:

– перший аргумент AF_INET – вказує, що використовується мережевий стек IPv4;

– другий аргумент SOCK_RAW – вказує на те що створюється саме сирий сокет;

– третій аргумент IPPROTO_IP – вказує на те що буде здійснена взаємодія з IP протоколом напряму.

Тут замість константи SOCK_STREAM (протокол TCP) чи SOCK_DGRAM (протокол UDP), ми беремо значення SOCK_RAW. Завдяки цьому, фактично отримується повний контроль за формування самого пакета.

3.3.3 Опис структури IP-пакета

```
typedef struct _IPHeader
{
  unsigned char  ver_len;
  unsigned char  tos;
  unsigned short length;
  unsigned short id;
     unsigned short flgs_offset;
     unsigned char  ttl;
     unsigned char  protocol;
     unsigned short xsum;
     unsigned long  src;
  unsigned long  dest;
  unsigned short *params;
  unsigned char  *data;
}IPHeader;
```

Рисунок 3.6 – Структура IP-пакета

Структура IP пакета визначає поля, які зберігаються у заголовку пакетів. Заголовок IP пакета включає в собі наступні поля:

– Ver_len – поле яке містить два значення: версію IP-протоколу, а також довжину заголовка IP-пакета. Зазвичай версія IP-протокола містить в собі від 4 до 6 значень, а довжину заголовку може бути у діапазоні від 20 до 60 байт;

– Tos – поле яке містить в собі прапора різного призначення, які можуть використовуватись для оцінки пріоритету трафіку;

– Length – поле яке вказує на довжину пакета, беручи до уваги заголовок та дані;

– Id – поле ідентифікатора яке використовується для ідентифікації IP-пакета, у випадку його фрагментації;

– Flgs_offset – поле яке в собі містить значення прапорів та заміщення, які можуть використовуватись для фрагментації IP-пакета;

– Ttl – поле час життя (самого TTL), максимальна кількість проміжних маршрутизаторів. Пакет може проходити до того, як буде відкладений;

– Protocol – поле яке вказує на протокол транспортного рівня, який може використовуватись для передачі даних;

– Xsum – поле контрольної суми, що використовується для перевірки цілісності IP-пакета;

– Src – поле яке містить IP-адресу відправника пакета;

– Dest – поле яке містить IP-адресу призначення пакета;

– Params – поле яке містить налаштування IP-пакета;

– Data – поле яке містить дані, які мають бути передані до транспортного рівня.

У цій частині використовується структура IPHeader, яка призначена для визначення полів заголовку IP-пакета. Кожне перелічене поле, май свій тип і

У полях params та data зберігаються налаштування, а ткаож частоково дані, які передаються через IP-пакет. Тип поля params-unsigned short, дозволяє зберігати масив налаштувань заголовку довжиною 320 біт. Тип поля data – unsigned char, дозволяє зберігати дані, що передаються до транспортного рівня (рис. 3.6).

3.3.4 Функція захоплення одного пакета

Функція RS-Sniff являє собою алгоритм для перехоплення мережевих пакетів і видавлювання заголовку IP (рис. 3.7).

Hdr – вказівник на IPHeader (заголовок IP)

Count – лічильник для відслідковування кількості байт, отриманих при прийомі пакета.

Здійснюється виклик функції recv для прийому пакета через сокет RS_SSocket.

Відповідно зберігається кількість прийнятих байт в змінну count і дані пакета в буфер RS_Buffer.

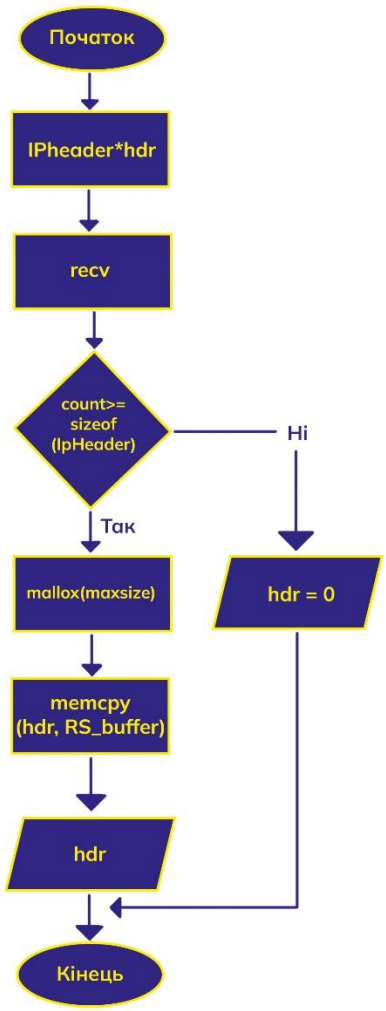


Рисунок 3.7 – Алгоритм захоплення одного пакета

Далі перевіряється, чи кількість прийнятих байт (count) більше чи рівно розміру структури IPHeader (sizeof(IPHeader)).

Якщо умова виконана:

– виділяється пам’ять для структури IPHeader, використовуючи функцію malloc і передається їй розмір MAX_PACKET_SIZE;

– копіюються дані з буферу RS_Buffer і виділену пам’ять для структури IPHeader, використовучи функцію memcpy;

– здійснюється виклик функції RS_UpdateNetStat для оновлення статистики мережі, передаючи кількість прийнятих байт count і вказівник на структуру IPHeader hdr;

– здійснюється повернення вказівника hdr, який зберігає в собі заголовок IP;

– якщо умова не виконана (кількість прийнятих байт менше розміру IPHeader) – повертається нульовий вказівник (0), щоб вказати на похибку чи відсутність заголовку IP [38].

3.3.5 Захоплення усіх пакетів, потрапляючих на мережевий інтерфейс

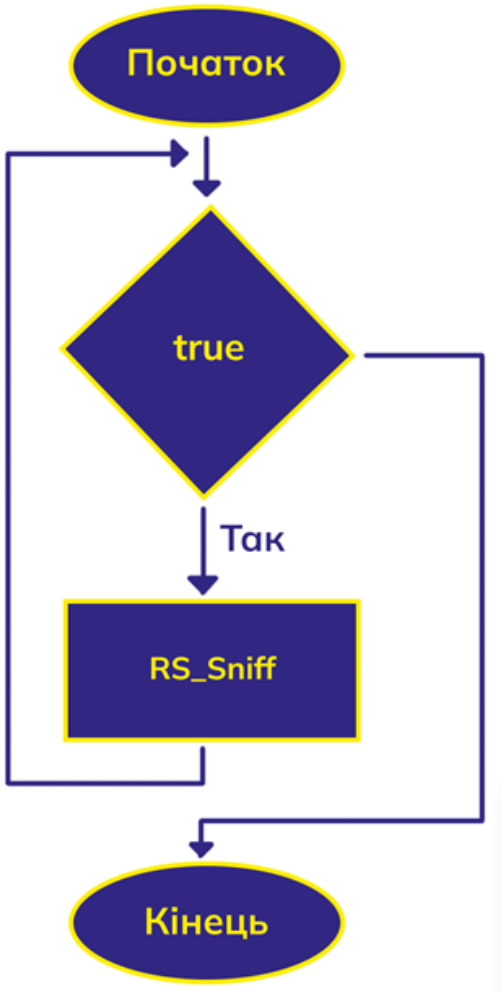


Рисунок 3.8 – Алгоритм захоплення пакетів, які потрапляють на мережевий інтерфейс

У даній частині коду представляється безкінечний цикл, який перехоплює IP-пакети за допомого функції RS_Sniff(). Процес обробки і виконання деяких задач виконується всередині циклу (рис. 3.8).

3.3.6 Реалізація графічної частини

Функція, яка визначає кольори для пакетів, які відображаються у графічному інтерфейсі. Функція приймає у якості налаштувань заголовок IP-пакета (структури IPHeader), IP-адресу мережевого інтерфейсу, якому належать пакети, та адреса віддаленого хоста.

Функція починає свою роботу з визначення кольору для приходящого пакета, який залежить від того, чи пустий він (якщо h->xsum ≠ 0, то пакет не є пустим). Якщо пакет не є просто пустим, то його колір світлим. Якщо пакет пустий – колір темний.

Наступні умови визначають, чи пакет для передачі, чи для отримання. Якщо пакет відправляється з мережі, то він відмічається червоним кольором, якшо приймається – зеленим.

Далі функція перевіряє тип протоколу у заголовку IP. Якщо протокол ICMP чи IGMP, то пакет відмічається синім кольором. Якщо протокол IP в IP (IPv4), то пакет є фіолетовим кольором. Якщо протокол TLS чи IP з шифруванням (являє собою протокол 115), то пакет відмічається жовтим кольором.

Умова “if (whost == h->dest || whost == h->src)” визначає, чи потрібно виділяти пакет зеленим кольором.

Також даний додаток має змогу перехоплювати ICMP пакети. Ці пакети застосовуються для передачі діагностичної інформації між пристроями, які знаходяться в одній мережі. Їх можна використати для відправки повідомлень про помилку, вимірювання життя пакета (ttl), перевірка хостів на доступність а також маршрутизаторів, і для перевірки зв’язку між пристроями.

Також, ICMP пакети можуть використовуватись для здійснення атак на мережі, такі як атаки DOS, і атаки з переповненням буферу.

Тому даний додаток буде мати змогу відмічати ICMP пакети конкретним кольором, а конкретно сірим.

Зазвичай в потрапляння даних пакетів у перелік перехоплених пакетів є безпечним і не несе загрози.

3.3.7 Перетворення IP-заголовку в строку

Рисунок 3.9 – Схема перетворення IP-заголовку в строку

Дана функція приймає вказівник IP-заголовку та перетворює його в строку зі текстовим представленням усіх полів заголовку. Використовуються функції

форматування строк «sprintf» та додаткові макроси, які розархівовують байти заголовку на окремі поля (рис. 3.9).

Основні активні дії даної функції:

– створення строки розмірністю «BUF_SIZE» та його обнулення за допомогою функції «memset»;

– застосування «sprintf» для форматування строки зі текстом представленням для кожного поля IP-заголовку, застосовуючи допоміжні макроси для розархівування байтів;

– повернення строки, як результат функції за допомогою «return».

Для перетворення IP-адрес з формату мережі в формат строки застосовується функція «iph2str», яка повертає рядок з текстом представленням IP-адреси [39].

3.3.8 Створення банку небажаних сигнатур

Процес підключення банку небажаних сигнатур у CSV форматі, можна розділити на декілька кроків. Основа частина є підключення функції до банку сигнатур, який буде викликатись при запуску додатка сканування мережевих пакетів.

1) Відкриття файлу CSV:

```
ifstream csvFile(“database.csv”)
```

Спочатку варто ініціалізувати файл банку небажаних сигнатур CSV. Це здійснюється за допомогою методу чи функції. У частинці коду використовується клас «ifstream».

Далі здійснюється зчитування даних з банку небажаних сигнатур: Після відкриття, здійснюється зчитування даних з CSV. Це досягається за допомогою відповідних методів чи функцій для робити з самим CSV-файлом. Зазвичай це включає в себе розбиття на рядки та подальше розбиття рядків на окремі поля. У даному випадку використовується функція «getline()» для зчитування рядків, та функцію «stringstream» для розбиття рядків на поля.

2) Алгоритм порівняння IP-адрес

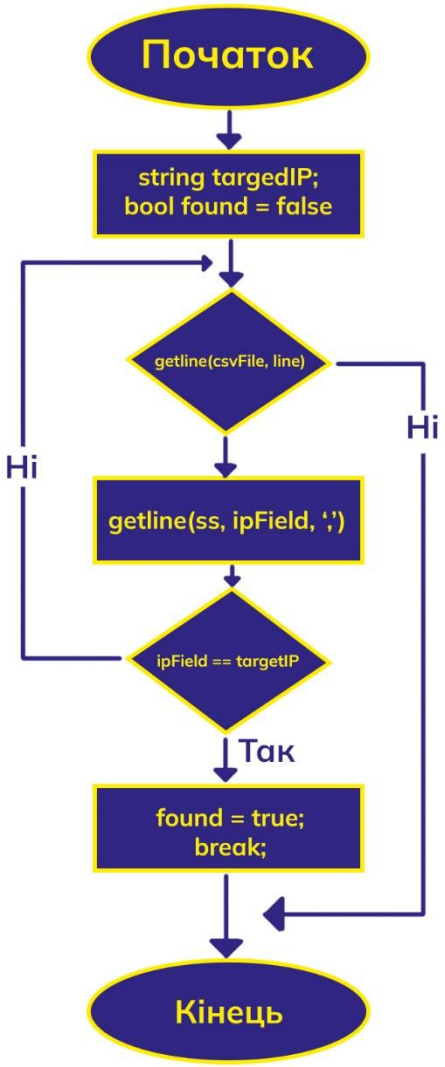


Рисунок 3.10 – Алгоритм порівняння IP-адрес

На цьому етапі визначається IP адреса з банку небажаних сигнатур.

Змінна «found» використовується для відслідковування того, чи кінцева IP-адреса була знайдена у банку небажаних сигнатур.

Далі виконується цикл, де кожен рядок банку сигнатур зчитується в змінну «line». Змінна «line» містить рядок банку сигнатур, який поділяється на окремі поля за допомогою об’єкту «stringstream». Значення IP-адреси виникає з поля IP за допомогою «getline» та роздільника «,».

Після успішного отримання значення IP-адреси, ми порівнюємо його з цільовою IP-адресою («targetIP»). Якщо дані співпадають, то встановлюється змінна «found» на «true» та цикл перевірки продовжується [40].

Відповідно до вищестореного коду, таблиця переліку IP-адрес зображена на рисунку 3.11.

database *

	A	B
1	216.58.215.57	google.com
2	50.7.210.2	uakino.com
3	142.250.186.206	youtube.com
4	204.79.197.239	
5	157.240.224.7	facebook.com
6	199.232.149.140	reddit.com
7	157.240.224.11	
8	35.186.241.51	Viber
9	199.232.149.140	
10	104.18.18.199	

Рисунок 3.11 – Таблиця переліку IP-адрес

3.3.9 Робота програми виявлення аномалій

Процес функціонування програми виявлення аномалій починається з перехоплення пакетів. При перехопленні пакета, програма здобуває доступ до IP-заголовку пакета, де зберігаються метадані про пакет, такі, як джерело, пункт призначення, довжина і.т.д. (рис. 3.12).

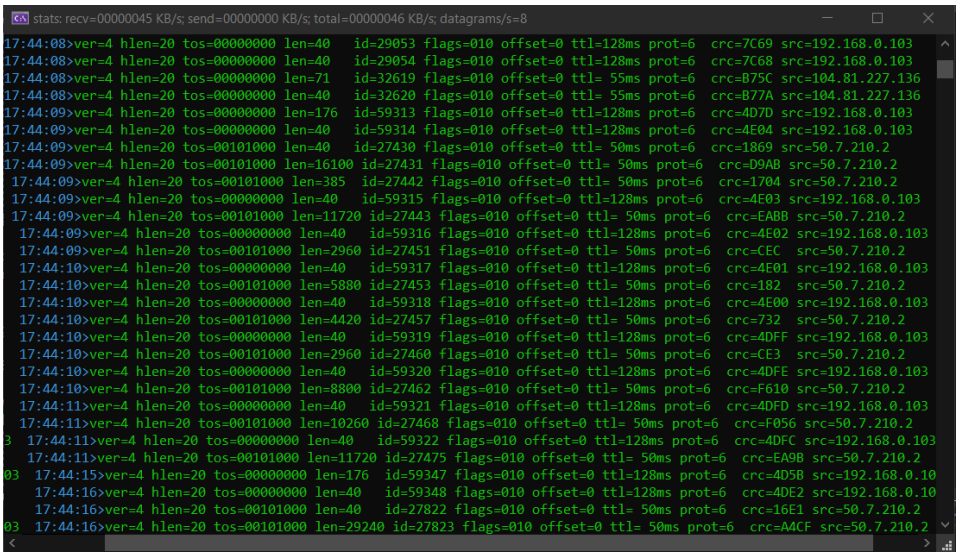


Рисунок 3.12 – Звичайний режим роботи програми виявлення аномалій

Далі є витягування IP-адреси з IP-заголовку пакета. Отриманий пакет порівнюється з банком небажаних сигнатур, у якому зберігаються введені раніше небажані IP-адреси. Якщо IP-адреса збігається з однією з адрес у банку небажаних сигнатур, програма відмічає пакет червоним кольором, тобто небажаним у трафіку.

У випадку виявлення небажаного пакета, програма відмічає червоним кольором пакет, що свідчить про знайдено небажано IP-адресу. Це дозволяє оператору\адміністратору\відповідальній людині вжити відповідних заходів для реагування на потенційну небезпеку (рис. 3.13).

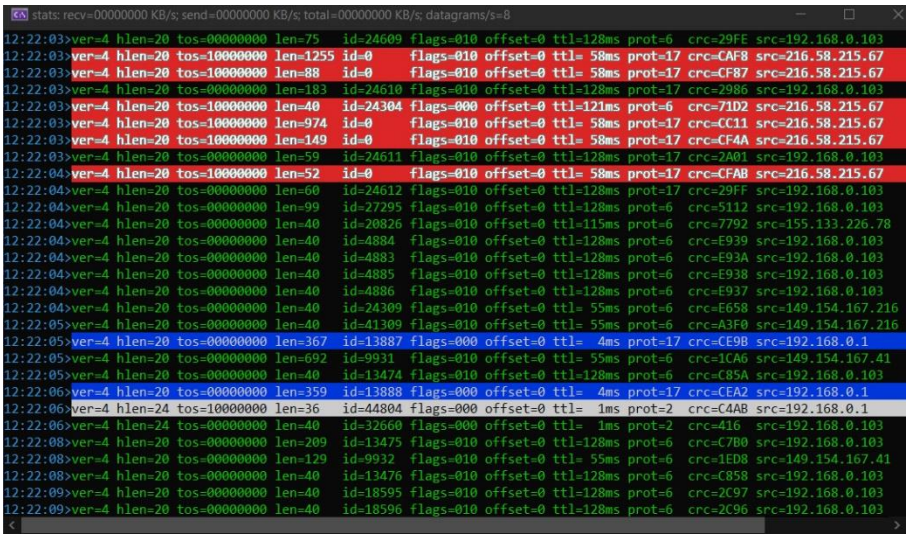


Рисунок 3.13 – Робота програми виявлення аномалій при співпаднні IP-адреси з банком небажаних сигнатур

Робота програми виявлення аномалій здійснюється у режимі реального часу, що дозволяє спостерігати за потоком пакетів і виявляти підозрілі одразу після їх виявлення. Це досить важливий аспект для ефективного моніторингу.

3.4 Висновок

У даному розділі було розроблено додаток для сканування трафіку з метою вияву аномалій та загроз безпеці. Реалізація додатка є основним етапом у забезпеченні безпеки мережевих систем, що дозволяє оперативно детектувати та реагувати на вразливості та атаки.

У першу чергу на шляху реалізації додатка було визначено критерії та властивості, за якими можна відрізняти аномальний трафіку від нормального. Як було зазначено раніше, сигнатурний аналіз став основою для цього визначення. Такі властивості, як обсяг передачі даних, часові інтервали та зміни мережевої активності, були використані для побудови моделі аномалій.

Далі була реалізація механізму перехоплення трафіку. Це мало в собі використання словників сигнатур (банк небажаних сигнатур), які дають змогу отримувати доступ до пакетів, які проходять через мережу.

Далі передбачається обробка та аналіз отриманих пакетів. Використовуючи раніше визначені критерії, додаток визначає, чи відповідають пакети нормальній поведінці, чи вони є аномалією.

Наступний етап полягав у виявлені аномалії. Додаток має механізм сповіщення, а саме відмічання підозрілого трафіку іншим кольором, що дозволяє оперативно приймати заходи для подальшого реагування та виправлення проблем.

ВИСНОВКИ

В даній роботі була розроблена програма виявлення аномалій, яка ідентифікує підозрілі пакети у мережі. Використання програми виявлення аномалій разом з банком небажаних сигнатур (IP-адрес) має досить велике значення для забезпечення безпеки та захисту від потенційних загроз.

В процесі розроблення програми виявлення аномалій було використано CSV банк сигнатур для зберігання небажаних IP-адрес. CSV банк небажаних сигнатур був обраний з метою спрощення та універсальності. Створення та редагування може здійснюватись звичайним текстовим редактором, а також імпортування та експортування даних з інших програм.

Спочатку було здійснено підключення банку небажаних сигнатур CSV до програми виявлення аномалій. Це виконується за допомогою ifstream, який відкриває потік читання з банку сигнатур.

Далі здійснюється обробка даних з CSV банку небажаних сигнатур. Кожен рядок у ній відповідає окремому запису про небажану IP-адресу. Задля отримання доступу до кожного запису, використовується цикл, який читає кожен рядок і розбиває його на поля за допомогою роздільника.

Зчитані дані зберігаються у відповідні змінні, чи структури даних для майбутнього використання. Можна створити вектор, чи список, де кожен елемент містить інформацію про небажаний IP-адрес.

Після завершення оброблення даних з CSV банку сигнатур, програма готова до перехоплення та перевірки пакетів та їх IP-адрес на достовірність про небажані адреси. Це дає змогу швидко реагувати на можливі загрози та вжити відповідні заходи для їх усунення.

Якщо ж пакет все таки має IP-адресу, яка збігається з однією з адрес у у банку небажаних сигнатур, програма виявлення аномалій буде вважати його підозрілим і відмічає його червоним кольором.

Отож, розроблена програма виявлення аномалій з використанням банку небажаних сигнатур має досить важливе значення для підвищення рівня безпеки

та захисту мережі від потенційних загроз. Використання CSV формату для зберігання небажаних сигнатур дозволяє гнучко створювати, редагувати та передавати дані між програмою та словником.

Після завантаження даних з банку сигнатур, програма готова до перехоплення і перевірки пакетів та їх IP-адреси. За допомого циклу, програма перевіряє кожен пакет на відповідність. Це дає можливість швидко реагувати на загрози і допомагає швидко прийняти рішення для їх усунення.

Таким чином програма виявлення аномалій є досить потужним інструментом для підвищення безпеки мереж, та виявлення підозрілих пакетів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. What Is Network Behavior Anomaly Detection? Definition, Importance, and Best Practices for 2022. Network behavior anomaly detection solutions monitor enterprise networks for abnormal behavior to detect threats and take remedial action. URL: https://www.spiceworks.com/tech/networking/articles/network-behavior-anomaly-detection/#lg=1&slide=0

2. Park Place Technologies. Network Traffic Analysis – Methods and How to Analyze. URL: https://www.parkplacetechnologies.com/blog/network-traffic-analysis-methods/

3. Shodan. Search Engine for the Internet of Everything. URL: https://www.shodan.io/

4. Comparitech. The Best Network Traffic Analysis (NTA) Tools for 2023. URL https://www.comparitech.com/net-admin/best-network-traffic-analysis-tools/

5. Internet usage statistics. Internet World Stats. URL: https://www.internetworldstats.com/

6. Cisco IOS IPsec Accounting with Cisco IOS NetFlow. Cisco Systems. URL: https://www.cisco.com/

7. Windows Network Data and Packet Filtering. NDIS Developer's Reference. URL: https://web.archive.org/web/20110207184825/http://ndis.com/

8. Repici D. J. The Comma Separtated Value (CSV) File Format. Creativyst Docs. URL: https://www.creativyst.com/

9. Alisha Cecil. A Summary of Network Traffic Monitoring and Analysis Techniques. URL: https://www.cse.wustl.edu/~jain/cse567-06/ftp/net_monitoring/index.html

10. WinPcap Documentation. WinPcap Documentation. URL: https://www.winpcap.org/docs/docs_412/html/main.html

11. Microsoft Docs – Windows Sockets. Windows Sockets 2. URL: https://learn.microsoft.com/en-us/windows/win32/winsock/windows-sockets-start-page-2

12. Microsoft Docs – Windows Sockets. WSAStartup function (winsock.h). URL: https://learn.microsoft.com/en-us/windows/win32/api/winsock/nf-winsock-wsastartup

13. Microsoft Docs – Windows Sockets. socket function. URL: https://learn.microsoft.com/en-us/windows/win32/api/winsock2/nf-winsock2-socket

14. Microsoft Docs – Windows Sockets. TCP/IP raw sockets. URL: https://learn.microsoft.com/en-us/windows/win32/winsock/tcp-ip-raw-sockets-2

15. Microsoft Docs - Windows Sockets. setsockopt function (winsock.h). URL: https://learn.microsoft.com/en-us/windows/win32/api/winsock/nf-winsock-setsockopt

16. GeeksforGeeks – Networking. Computer Network Tutorial. URL: https://www.geeksforgeeks.org/computer-network-tutorials/

17. CodeProject – Windows Socket. How to create customized InstallShield for a VC++ Application. URL: https://www.codeproject.com/Articles/5959/How-to-create-customized-InstallShield-for-a-VC-Ap

18. RFC 4180: RFC 4180. Common Format and MIME Type for CSV Files. URL: https://datatracker.ietf.org/doc/html/rfc4180

19. Usenix. Machine Learning Approaches to Network Anomaly Detection. URL: https://www.usenix.org/legacy/event/sysml07/tech/full_papers/ahmed/ahmed_html/sysml07CR_07.html

20. ISSN 1997-9266. Вісник Вінницького політехнічного інституту. 2021. № 2. АНАЛІЗ ВИКОРИСТАННЯ ТРАФІКУ ПРИ СКАНУВАННІ КОМП'ЮТЕРНИХ МЕРЕЖ РІЗНИМИ ВЕРСІЯМИ NMAP. URL: https://visnyk.vntu.edu.ua/index.php/visnyk/article/download/2612/2468/2939

21. Вікіпедія. Система виявлення вторгнень. URL: https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%B2%D0%B8%D1%8F%D0%B2%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B2%D1%82%D0%BE%D1%80%D0%B3%D0%BD%D0%B5%D0%BD%D1%8C

22. RFC 792 – Internet Control Message Protocol. DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION. URL: https://datatracker.ietf.org/doc/html/rfc792

23. Вікіпедія. CSV. URL: https://uk.wikipedia.org/wiki/CSV

24. MicrosoftDocs/win32. windows-sockets-start-page-2. URL: https://github.com/MicrosoftDocs/win32/blob/docs/desktop-src/WinSock/windows-sockets-start-page-2.md

25. GURU99. CSV vs Excel – Difference Between Them. URL: https://www.guru99.com/excel-vs-csv.html

26. Netscout. What is a Packet Sniffer?. URL: https://www.netscout.com/what-is/sniffer

27. Geeksforgeeks. Introduction to Sniffers. URL: https://www.geeksforgeeks.org/introduction-to-sniffers/

28. Georgia Weidman «Penetration testing». Attacks. Розділ – 3, с.339

29. Mumtaz M.Ali AL-Mukhtar. A Protocol Analyzer Using a Sockets Based Network Sniffer. Australian Journal of Basic and Applied Sciences. URL: http://www.ajbasweb.com/old/ajbas/2010/1632-1640.pdf. с.1634-1639

30. MDPI Open Access Journals. Anomaly Detection Module for Network Traffic Monitoring in Public Institutions. URL: https://www.mdpi.com/1424-8220/23/6/2974

31. ManageEngine. Network traffic anomaly detection: A fail-proof traffic monitoring technique. URL: https://www.manageengine.com/products/netflow/network-traffic-anomaly-detection.html

32. Matthew V. Mahoney. Network Traffic Anomaly Detection Based on Packet Bytes. Florida Institute of Technology, Melbourne, Florida. URL: https://cs.fit.edu/~mmahoney/paper6.pdf. с.2-5

33. ResearchGate. Anomaly Detection Module for Network Traffic Monitoring in Public Institutions. URL:

https://www.researchgate.net/publication/369146568_Anomaly_Detection_Module_for_Network_Traffic_Monitoring_in_Public_Institutions

34. Springer Open. Survey of intrusion detection systems: techniques, datasets and challenges. URL https://cybersecurity.springeropen.com/articles/10.1186/s42400-019-0038-7

35. Springer. Analysis of network traffic features for anomaly detection. URL: https://link.springer.com/article/10.1007/s10994-014-5473-9

36. І. Р. Арсенюк. ЗМЕНШЕННЯ КІЛЬКОСТІ ІНФОРМАТИВНИХ ОЗНАК ДЛЯ ЗАДАЧІ ДЕТЕКТУВАННЯ КОМП'ЮТЕРНИХ АТАК. Репозитарій ВНТУ. URL: http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/20635/5097.pdf?sequence=3. с.1-2

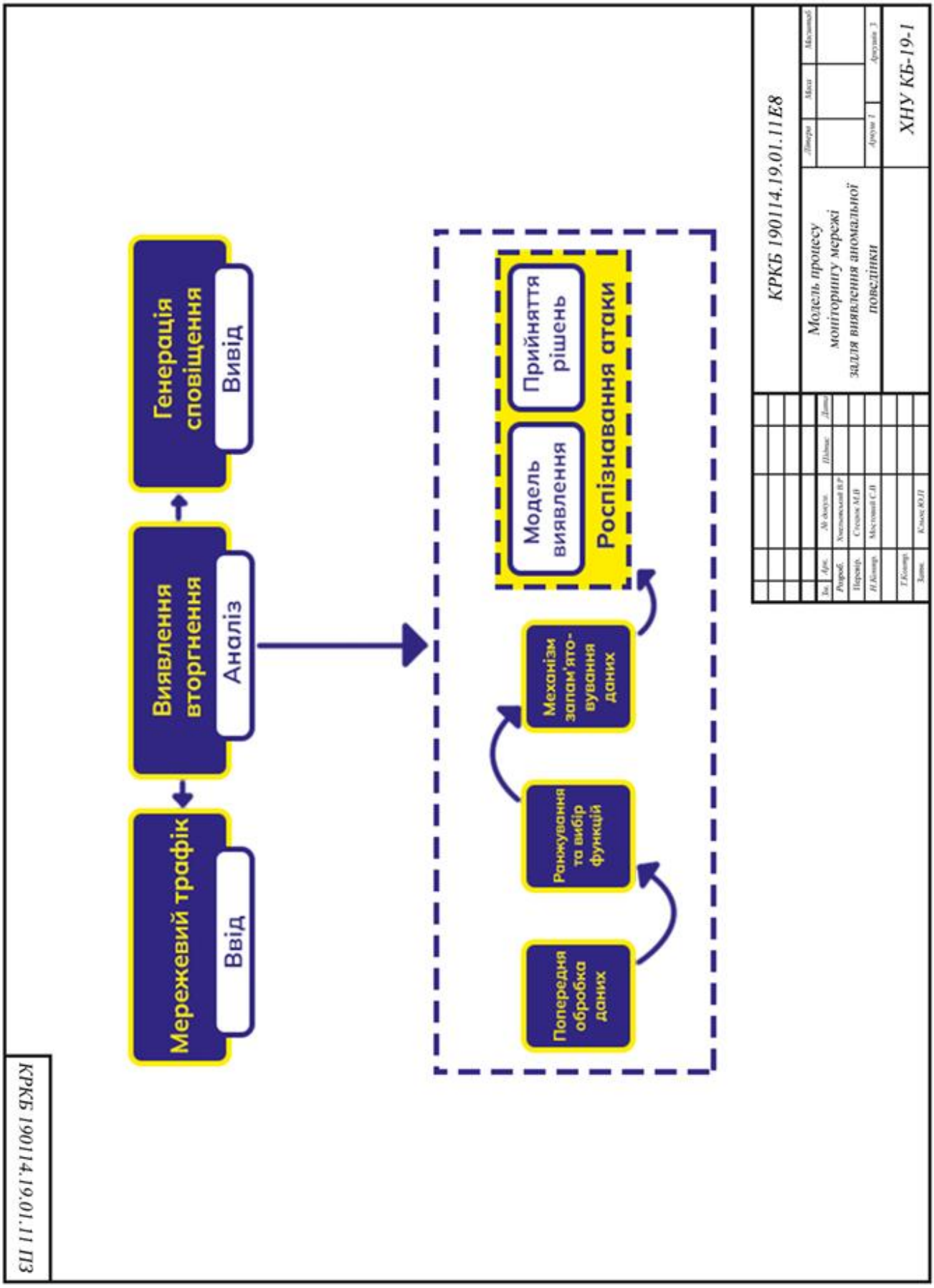
37. Rapid7. What is Network Traffic Analysis?. URL: https://www.rapid7.com/fundamentals/network-traffic-analysis/

38. IEEE Xplore. The Study and Design of Network Traffic Monitoring Based on Socket. URL: https://ieeexplore.ieee.org/document/6300756

39. ghacks.net. Monitor network traffic of Windows processes with Socket Sniff. URL: https://www.ghacks.net/2014/12/22/monitor-network-traffic-of-windows-processes-with-socket-sniff/

40. Washington University in St.Louis. A Survey of Network Traffic Monitoring and Analysis Tools. URL: https://www.cse.wustl.edu/~jain/cse567-06/ftp/net_traffic_monitors3/index.html

ДОДАТОК А


Мережевий трафік
Ввід
Виявлення вторгнення
Аналіз
Генерація сповіщення
Вивід
Попередня обробка даних
Ранжування та вибір функцій
Механізм запам'ятовування даних
Модель виявлення
Прийняття рішень
Роспізнавання атаки
КРКБ 190114.19.01.11 ПЗ
КРКБ 190114.19.01.11Е8
Модель процесу моніторингу мережі задля виявлення аномальної поведінки
ХНУ КБ-19-1

Зм.	Арк.	№докум.	Підпис	Дата	КРКБ.190114.19.01.11 ПЗ	Арк. 61

КРКБ.190114.19.01.11 ПЗ
Початок
IPheader*hdr
recv
count>= sizeof (IpHeader)
Ні
hdr = 0
Так
malloc(maxsize)
memcpy (hdr, RS_buffer)
hdr
Кінець
Початок
string targetIP; bool found = false
getline(ccvFile, line)
Ні
getline(ss, ipField, ',')
ipField == targetIP
Так
found = true; break;
Кінець
Ні
Початок
true
Так
RS_Sniff
Кінець
КРКБ 190114.19.01.11Е8
Модель роботи алгоритма захоплення одного пакета; захоплення пакетів, які потрапляють на інтерфейс; порівняння IP-адрес
ХНУ КБ-19-1

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

освітньо-кваліфікаційного рівня «бакалавр»

Студент Хмельовський Віктор Русланович
Тема: «Система виявлення аномального трафіку на основі сигнатур»
Галузь знань 12 «*Інформаційні технології*» Спеціальність 125 «*Кібербезпека*» Освітня програма «Кібербезпека»

Обсяг дипломної роботи освітньо-кваліфікаційного рівня «бакалавр»: кількість листів креслень 3; кількість сторінок записки 59;

1. Короткий зміст КР та прийнятих рішень У даній кваліфікаційній роботі було поставлено мету вивчити та дослідити питання, пов'язані з системою виявлення аномального трафіку на основі сигнатур. Робота спрямовувалась на розробку нової системи, яка б забезпечувала виявлення аномалій у мережевому трафіку та реагувала на потенційні загрози.

2. Висновок про відповідність КР завданню Кваліфікаційна робота повністю відповідає поставленому завданню. У роботі було досліджено систему виявлення аномального трафіку на основі сигнатур і розроблено новий підхід до цього завдання. Було проведено аналіз наявних методів і вибрано найбільш відповідний, оснований на сигнатурах

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі було розроблено характеристику до теми роботи, сформульовано мету та завдання дослідження. У другому розділі були представленні теоретичні основи виявлення аномального трафіку на основі сигнатур. Було досліджено принципи роботи сигнатурного аналізу, включаючи процес створення та використання сигнатур. Також у цьому розділі була описана конкретна розробка системи. Були використані методи роботи, включаючи використання WinSocket для перехоплення трафіку та обробки пакетів. У третьому розділі були представлені результати розроблень. Де був представлена система виявлення аномалій, і робота її разом зі словником сигнатур.

4. Позитивні сторони кваліфікаційної роботи полягають у тому що, реалізована система демонструє достатній рівень ефективності виявлення аномалій у мережевому трафіку. Використання даних сигнатурних методів дозволяє досить чітко ідентифікувати зловмисний або ненормальний трафік. Допомагає виявляти незареєстрвоанний трафік на підприємствах, що дозволить виявляти зловмисні дії або несанкціонований доступ.

5. Негативні сторони проекту: - відсутність графічного інтерфейсу, відсутність можливості зупинити сканування

6. Оцінка графічного оформлення та пояснювальної записки роботи. Графічне оформлення виконане відповідно до теми кваліфікаційної роботи із дотриманням усіх стандартів. У загальному графічне оформлення виконане на достатньому технічному рівні. Пояснювальна записка відповідає нормам для її оформлення та вимогам

7. Відгук про роботу в цілому В загальному кваліфікаційна робота заслуговує позитивної оцінки. Весь матеріал кваліфікаційної роботи структурований, чіткий та послідовний. Усі розділи роботи послідовні та логічні, що дозволяє чітко розуміти викладений матеріал в рамках тематики кваліфікаційної роботи. У пояснювальній записці багато наглядних пояснень. Графічний матеріал дозволяє наочно побачити доцільність та ефективність рішень, які були прийняті за основу для досягнення поставленої задачі проектування.

8. Інші зауваження -

9. Оцінка дипломної роботи Розглянувши позитивні та негативні сторони представленої кваліфікаційної роботи, можна зробити висновок, що робота заслуговує оцінки «відмінно/ В (4,50)».

РЕЦЕНЗЕНТ (прізвище, ім'я, по батькові, посада, місце роботи) д.т.н., професор Лисенко Сергій Миколайович

« 5 » червня 2023 . _____ (підпис)

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА СИСТЕМНОГО ПРОГРАМУВАННЯ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

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

Назва: Система виявлення аномального трафіку на основі сигнатур

Автор: Хмельовський Віктор Русланович

Галузь знань *12 «Інформаційні технології»*

Спеціальність: 123 – *«Комп'ютерна інженерія»*

Освітня програма: *«Кібербезпека»*

Науковий керівник: Стецюк Микола Васильович, д-р філос., ст. викл.

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних). 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 (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). 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 роботи та ідентичності 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 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.	
5	Інше:	

Підтвердження:

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

1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
3) окремі виявлені збіги є загальновживаними фразами або виразами, про що свідчить посилання системи на збіг з 10-30 джерелами на один фрагмент речення;
4) в якості запозичень в окремих місцях системою зафіксовано послідовності кодів, які є вхідними даними до великої кількості задач і не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;
5) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості, складає 9.3 % і адресується до першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру роботи, і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ М.В. Стецюк

Завідувач кафедри кібербезпеки _____ Ю.П. Кльоц

Дата: 07.06.2023

UNICHECK

Ім'я користувача:
Кафедра кібербезпеки

ID перевірки:
1015416712

Дата перевірки:
04.06.2023 21:13:23 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
04.06.2023 21:21:53 EEST

ID користувача:
100008300

Назва документа: **Хмельовський**

Кількість сторінок: **59** Кількість слів: **11348** Кількість символів: **87926** Розмір файлу: **5.24 MB** ID файлу: **1015079170**

9.32%
Схожість

Найбільша схожість: **2.08%** з Інтернет-джерелом (https://er.nau.edu.ua/bitstream/NAU/45803/1/%d0%a4%d0%9a%d0%...

9.32% Джерела з Інтернету	442	Сторінка 61
1.21% Джерела з Бібліотеки	117	Сторінка 65

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

6.75%
Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за **8 слів** та **0%**)

6.74% Вилучення з Інтернету	1	Сторінка 66
6.75% Вилученого тексту з Бібліотеки	33	Сторінка 66

Модифікації

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

Замінені символи	1

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 16%**

	Документ		Сумарний збіг по Базі Даних	
ID: 114649 Назва: Система виявлення аномального трафіку на основі сигнатур Додано в БД: 2023-06-04 Автора: Хмельовський В.Р. Керівники: Стецюк М.В. Консультанти: Опоненти:	Символи	Лексеми	Символи	Лексеми
	76512	682	959 (1%)	11 (2%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми