НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет Інформаційних технологій

ПОГОДЖЕНО	**ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ**
Декан факультету (Директор ННІ)	**Завідувач кафедри**
Інформаційних Технологій (назва факультету (ННІ))	Комп'ютерних Наук (назва кафедри)
_____ Ігор БОЛБОТ (підпис) (ім'я ПРІЗВИЩЕ)	_____ Белла ГОЛУБ (підпис) (ім'я ПРІЗВИЩЕ)
"___" _____ 2025 р.	"01" грудня 2025 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему «Методи забезпечення безпеки обміну інформацією у компанії»

Спеціальність 122 «Комп'ютерні Науки»
(код і найменування)

Освітня програма Інформаційні управляючі системи і технології
(назва)

Орієнтація освітньої програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Гарант освітньої програми

к.т.н., доцент _____ _____ Белла ГОЛУБ
(науковий ступінь та вчене звання) (підпис) (ім'я ПРІЗВИЩЕ)

Керівник магістерської кваліфікаційної роботи

ст. викладач _____ _____ Світлана ВАСИЛЮК-ЗАЙЦЕВА
(науковий ступінь та вчене звання) (підпис) (ім'я ПРІЗВИЩЕ)

Консультант магістерської кваліфікаційної роботи

к.т.н., доцент _____ _____ Яна КРИВОРУЧКО
(науковий ступінь та вчене звання) (підпис) (ім'я ПРІЗВИЩЕ)

Виконав _____ _____ Олександр КОЛЕСНІКОВ
(підпис) ((ім'я ПРІЗВИЩЕ здобувача)

КИЇВ – 2025

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет Інформаційних технологій

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук
доцент, к.т.н. _____ Голуб Б. Л.
(науковий ступінь, вчене звання) (підпис) (ПІБ)
"01" листопада 2024 року

З А В Д А Н Н Я

ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ

Колеснікову Олександру Олександровичу
(прізвище, ім'я, по батькові)

Спеціальність 122 «Комп'ютерні науки»
(код і назва)

Освітня програма Інформаційні управляючі системи та технології
(назва)

Орієнтація освітньої програми освітньо-професійна

Тема магістерської кваліфікаційної роботи Методи забезпечення безпеки обміну інформацією у компанії
затверджена наказом ректора НУБіП України від "01" листопада 2024р. №1964 «С»
Термін подання завершеної роботи на кафедру 01.12.2025
(рік, місяць, число)

Вихідні дані до магістерської кваліфікаційної роботи: журнали подій із систем безпеки (брандмауери, IDS/IPS, сервери); дані з мережевого обладнання та систем моніторингу трафіку; звіти про інциденти інформаційної безпеки у компанії; сучасні методи зберігання та аналізу даних (OLAP, сховища даних, SSAS/SSIS); нормативні документи та стандарти з інформаційної безпеки (ISO/IEC 27001, НД ТЗІ).

Перелік питань, що підлягають дослідженню:

1. Аналіз існуючих загроз інформаційного обміну у корпоративних мережах (класифікація, джерела, рівні небезпеки).
2. Дослідження методів збору та обробки даних із систем безпеки (логи, події, інциденти).
3. Проєктування сховища даних для моніторингу та аналізу кіберзагроз.
4. Використання OLAP-технологій (SSAS) для багатовимірного аналізу інцидентів безпеки.
5. Розробка аналітичних панелей для виявлення тенденцій атак і оцінки ефективності заходів реагування.
6. Оцінка ефективності системи забезпечення безпеки інформаційного обміну за показниками часу реагування, кількості повторних атак і рівня впливу на компанію.

Дата видачі завдання "01" листопада 2024 р.

Керівник магістерської кваліфікаційної роботи _____ Василюк-Зайцева С.В.
(підпис) (прізвище та ініціали)

Завдання прийняв до виконання _____ Колесніков О.О.
(підпис) (прізвище та ініціали студента)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Видача завдання	01.11.2024	
2	Аналіз предметної області	05.11.2024 – 26.11.2024	
3	Моделювання системи	26.11.2024 – 04.02.2025	
4	Розробка системи	04.02.2025 – 28.06.2025	
5	Аналіз результатів	30.06.2025 – 20.07.2025	
6	Оформлення записки	13.07.2025 – 03.10.2025	
7	Оформлення постеру	03.10.2025 – 20.10.2025	
8	Постерна сесія	28.10.2025 – 29.10.2025	
9	Перевірка на плагіат	14.11.2025	
10	Попередній захист	01.12.2025	
11	Захист	15.12.2025	

Керівник магістерської кваліфікаційної роботи

_____ Світлана ВАСИЛЮК-ЗАЙЦЕВА
(підпис) (ім'я ПРІЗВИЩЕ)

Студент

_____ Олександр КОЛЕСНІКОВ
(підпис) ((ім'я ПРІЗВИЩЕ здобувача)

РЕФЕРАТ

Робота присвячена створенню аналітичної системи для забезпечення безпеки обміну інформацією в компанії з метою мінімізації ризиків кіберзагроз та підтримки прийняття рішень щодо захисту даних. А також вирішенню актуальної проблеми переходу від реактивних до проактивних моделей захисту інформації в умовах зростання обсягів логів та складності кібератак.

Об'єкт дослідження: процеси обміну інформацією в компанії та відповідно логи систем безпеки (брандмауери, IDS/IPS, сервери), які фіксують ці процеси.

Предмет дослідження: методи забезпечення безпеки інформаційного обміну, які реалізовані через багатовимірний аналіз (OLAP) та технології Data Mining.

Використані методи: системний аналіз, багатовимірне моделювання (OLAP) у схемі «зірка», програмні ETL-процеси на SQL, та підходи Data Mining: K-Means (кластеризація), 1-Rule та Наївний Баєс (класифікація).

Мета роботи – розробка рекомендацій для впровадження ефективних проактивних методів забезпечення безпеки у роботі із інформацією задля мінімізації ризиків несанкціонованого доступу та забезпечення стабільної роботи IT-інфраструктури.

Наукова складова полягає у тому, що у ході дослідження було запропоновано та обґрунтовано інтеграцію багатовимірного аналізу OLAP та Data Mining для проактивного управління безпекою обміну інформацією, зокрема доведено значущість часового фактору для прогнозування рівня загроз.

Рекомендації щодо впровадження результатів: результати можуть бути застосовані для автоматизації виявлення атак типу Brute Force, оптимізації роботи центрів реагування (SOC) шляхом пріоритезації моніторингу та проведення аудиту ефективності згідно зі стандартом ISO 27001.

Прикладна значимість роботи: запропонована система забезпечує гнучкі інструменти для сегментації джерел загроз, моніторингу ключових показників ефективності (KPI), таких як середній час реагування, та ухвалення обґрунтованих рішень у сфері інформаційної безпеки.

Кількість сторінок – 55.

Кількість ілюстрацій – 19.

Кількість таблиць – 2.

Кількість додатків – 2.

Кількість джерел – 27.

ABSTRACT

The work is devoted to solving the urgent problem of transitioning from reactive to proactive information protection models in the context of increasing log volumes and the complexity of cyberattacks.

Object of research: information exchange processes in a company and corresponding security system logs (firewalls, IDS/IPS, servers) that record these processes.

Subject of research: methods of ensuring information exchange security implemented through multidimensional analysis (OLAP) and Data Mining technologies.

Methods used: system analysis, multidimensional modeling (OLAP) in a "star" schema, software-based ETL processes using SQL, and Data Mining approaches: K-Means (clustering), 1-Rule, and Naive Bayes (classification).

Purpose of the work is to develop recommendations for implementing effective proactive security methods in information handling to minimize unauthorized access risks and ensure stable IT infrastructure operation.

The scientific component consists in the fact that during the research, the integration of OLAP multidimensional analysis and Data Mining for proactive security management of information exchange was proposed and substantiated; specifically, the significance of the time factor for predicting threat levels was proven.

Recommendations for implementing the results: the results can be applied to automate the detection of Brute Force attacks, optimize Security Operations Centers (SOC) workflows by prioritizing monitoring, and conduct effectiveness audits in accordance with the ISO 27001 standard.

Applied significance of the work: the proposed system provides flexible tools for segmenting threat sources, monitoring Key Performance Indicators (KPI) such as average response time, and making informed decisions in the field of information security.

Number of pages – 55. Number of illustrations – 19. Number of tables – 2. Number of appendices – 2. Number of sources – 27.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

СД – сховище даних.

1R – One Rule алгоритм.

OLAP – On-Line Analytical Processing.

SQL – Structured Query Language.

SSAS – SQL Server Analysis Services.

SSIS – SQL Server Integration Services.

BI - Business intelligence

ВСТУП

Актуальність теми. Сучасні компанії, незалежно від їхньої сфери діяльності, критично залежать від ефективного та безпечного обміну інформацією. Однак, традиційні методи забезпечення безпеки, що ґрунтуються на реактивному підході, виявилися недостатніми в умовах постійного зростання кількості та складності кібератак, а також прихованих внутрішніх загроз. Обсяги логів, що генеруються системами безпеки у корпоративних мережах є надто великими для ручної обробки. Це вимагає переходу до проактивної моделі, що базується на інтелектуальному аналізі великих даних.

Об’єктом дослідження є процеси обміну інформацією в компанії та відповідно логи систем безпеки (брандмауери, IDS/IPS, сервери), які фіксують ці процеси.

Предметом дослідження є методи забезпечення безпеки інформаційного обміну, які реалізовані через багатовимірний аналіз (OLAP) та технології Data Mining.

Метою дослідження є розробка рекомендацій для впровадження ефективних проактивних методів забезпечення безпеки у роботі із інформацією задля мінімізації ризиків несанкціонованого доступу та забезпечення стабільної роботи IT-інфраструктури.

Методи та засоби використані у ході дослідження для досягнення мети та виконання поставлених завдань наступні: використання системного аналізу, багатовимірне моделювання (OLAP), мова SQL, засоби SSMS, SSAS, SSIS, та Data Mining.

Наукова новизна полягає у тому, що в ході виконання роботи було запропоновано та обґрунтовано інтеграцію багатовимірного аналізу OLAP та Data Mining для проактивного управління безпекою обміну інформацією.

Апробація результатів дослідження. Робота була представлена на XVI Міжнародній науково-практичній конференції молодих вчених «Інформаційні технології: економіка, техніка, освіта» (м. Київ, 2025р.).

Структура роботи. Робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків.

За струкутурою поділяється на чотири розділи. Перший розділ присвячено постановці завдання, огляду існуючих рішень. Другий розділ присвячено моделюванню: наведено його загальні положення, а також проведено моделювання системи за допомогою діаграм прецедентів та ER. Третій розділ надає опис основних використаних методів (OLAP, Data Mining), а також надає опис джерела даних, з якого дані надходять до сховища даних та як саме ці дані надходять до сховища. Четвертий розділ присвячено аналізу отриманих результатів та формуванню висновків по даним

Загальний обсяг роботи становить 55 сторінок, 19 рисунків, 2 таблиці. Список використаних джерел налічує 27 найменувань.

1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз існуючих загроз інформаційного обміну у корпоративних мережах

1.1.1 Сутність загроз та їх фіксація у логах систем безпеки

Забезпечення безпеки обміну інформацією ґрунтується на захисті інформаційних активів, що передаються, обробляються та зберігаються в корпоративному середовищі. Фундаментальні вимоги до захисту даних відображені у тріаді Конфіденційність, Цілісність та Доступність (CIA) (рис.1.1) [1].

Рис 1.1 CIA Triad

Порушення кожного з цих принципів фіксується у вигляді специфічних подій у логах систем безпеки [2]:

- Порушення Конфіденційності: Несанкціонований доступ, який може бути виявлений через аномально високу кількість невдалих спроб авторизації або Brute-Force атак.
- Порушення Цілісності: Зміна або знищення даних, фіксується як аномалії у роботі систем або сервісів.

- Порушення Доступності: Призводить до збоїв у роботі, часто є результатом DDoS-атак, які генерують великий обсяг мережевих логів.

1.1.2 Класифікація загроз за джерелами та необхідність інтелектуального аналізу

Сучасні загрози класифікують за різними критеріями. Для цілей інтелектуального аналізу найбільш важливим є поділ за джерелом:

- Зовнішні загрози: Атаки типу Advanced Persistent Threats (APT), які використовують багатокрокові схеми та маскуються під нормальний трафік [3].
- Внутрішні загрози: Дії співробітників, які можуть призвести до витоку даних. Виявлення цих загроз є критично важливим, оскільки вони часто не фіксуються традиційними системами [4].

Виявлення прихованої поведінки та аналіз кібератак на локальні мережі вимагає поєднання традиційних методів захисту із засобами глибокого аналізу. Для цього необхідно мати інструменти, здатні виділяти аномалії, як-от нетипові спроби автентифікації, з метою подальшого прогнозування рівня небезпеки.

1.1.3 Класифікація інфекційних загроз та їх вплив на обмін даними

Для забезпечення ефективності системи моніторингу, що базується на аналізі логів, необхідно провести детальний розгляд основних класів шкідливого програмного забезпечення (Malware), які є першоджерелами інцидентів безпеки. Ці програми становлять суттєву загрозу як для приватних користувачів, так і для корпоративного бізнесу, оскільки можуть призводити до блокування даних або крадіжки облікових відомостей.

Черв'як (Worm) є автономним зловмисним програмним забезпеченням, яке поширюється через вразливості програмного забезпечення або фішингові канали. Його ключовою особливістю є здатність до самовідтворення та швидкого поширення в мережі без потреби у файлі-носії [X]. Як тільки черв'як інсталюється в пам’ять комп'ютера, він починає заражати всю локальну мережу. Наслідки зараження черв’яками безпосередньо впливають на метрики DWH: вони можуть споживати пропускну здатність, перевантажувати веб-сервери та виснажувати системні ресурси, що відображається у зростанні метрики ThreatCount та зниженні AvgResponseTime.

На відміну від черв'яків, віруси (viruses) для своєї активації потребують інфікованої активної операційної системи або програми. Віруси зазвичай прикріплюються до виконуваного або текстового файлу. Вони залишаються

бездіяльними, доки не буде активований заражений файл хоста. Поширення вірусів відбувається через заражені веб-сайти, обмін файлами або завантаження вкладень електронної пошти. Як тільки вірус активується, він здатен розмножуватися і поширюватися через системи, викликаючи зміни у налаштуваннях та потенційні порушення цілісності даних.

Троянський кінь (Trojan) — це шкідлива програма, яка маскується під легітимний та надійний файл. На відміну від черв'яків, трояни не самовідтворюються, але вони створюють прихований "чорний хід" (бекдор) для хакерів, що дозволяє їм віддалено контролювати пристрій . Трояни можуть використовуватись для видалення, зміни та захоплення даних, а також для включення пристрою до складу ботнету.

Програма-вимагач (Ransomware) є однією з найсерйозніших загроз для бізнесу. Вона обмежує або повністю блокує доступ до файлів користувача або цілої системи, вимагаючи оплати (зазвичай у криптовалютах) за повернення доступу. Яскравим прикладом є атака WannaCry 2017 року. Щоб мінімізувати ризик, необхідно регулярно оновлювати операційну систему, підтримувати антивірусне програмне забезпечення в актуальному стані та створювати резервні копії найважливіших файлів.

1.1.4 Детальний опис загроз соціальної інженерії та мережевого контролю

Крім шкідливого програмного забезпечення, актуальні загрози безпеці обміну інформацією включають елементи соціальної інженерії та віддаленого управління, які також залишають чіткий слід у логах, що підлягає DM-аналізу.

А. Ботнети (Botnets)

Ботнет — це мережа заражених комп’ютерів (ботів), якими хакер може керувати віддалено. Ботнети використовуються для виконання **масштабних шкідливих дій**, включаючи DDoS-атаки, поширення інших типів шкідливих програм, Keylogging та розсилку фішингових повідомлень [X]. Їхня непомітна природа робить їх ідеальною зброєю для тривалого прихованого контролю.

Б. Фішинг (Phishing)

Фішинг є типом атаки соціальної інженерії, спрямованої на шахрайське отримання особистої та фінансової інформації. Атака успішна, оскільки створені електронні листи або веб-посилання виглядають так, ніби вони походять з надійних джерел (наприклад, від колеги або банку). Сучасні фішингові атаки є високо складними і можуть обдурити навіть досвідчених користувачів, особливо у випадках, коли зламано обліковий запис відомого контакту [X].

В. Шпигунське програмне забезпечення (Spyware)

Шпигунське програмне забезпечення таємно записує онлайн-активність користувача, збирає його дані, імена, паролі та звички серфінгу. Зазвичай воно поширюється як безкоштовне або умовно-безкоштовне програмне забезпечення, виконуючи свою приховану місію у фоновому режимі. Передаючи дані рекламодавцям або кіберзлочинцям, шпигунське програмне забезпечення робить пристрій м’якою мішенню для подальших, більш серйозних атак [X].

1.1.5 Основні запобіжні заходи та їх недостатність для проактивного захисту

Хоча абсолютного захисту від кіберзагроз не існує, зменшити ризик реалізації більшості атак можна, дотримуючись низки ключових запобіжних заходів:

- Оновлення ПЗ: Своєчасна інсталяція нових версій операційних систем та регулярне оновлення всіх програм для усунення відомих вразливостей.
- Антивірусний захист: Використання ліцензованих антивірусних рішень від надійних виробників.
- Резервне копіювання: Регулярне створення резервних копій найважливіших файлів (особливо актуально для захисту від програм-вимагачів).
- Політики безпеки: Недопущення до ПК сторонніх осіб, заборона відкривати підозрілі посилання, листи та файли [X].

Недостатність традиційних методів: Навіть при ідеальному дотриманні цих правил, системи залишаються вразливими до Zero-Day атак (атаки, для яких ще не існує сигнатур), а також до витончених APT-атак та інсайдерських загроз. Тому, для забезпечення повного захисту, необхідно, щоб аналітична система (DWH та Data Mining) постійно шукала аномалії у логах, які не були виявлені на рівні антивірусного або мережевого екрану. Це обґрунтовує необхідність проактивного аналізу.

1.2 Нормативні та технологічні основи забезпечення безпеки обміну інформацією

1.2.1 Вимоги стандартів ISO/IEC 27001 та НД ТЗІ

Надійний захист обміну інформацією повинен відповідати національним стандартам та міжнародним вимогам, зокрема стандартам серії ISO/IEC 27001 [5]. Особлива увага приділяється контролю A.16 "Управління інцидентами

інформаційної безпеки". ISO 27001 вимагає не лише реєстрації, але й оцінки ефективності заходів реагування.

Для кількісної оцінки ефективності управління інцидентами, необхідно ввести у Сховище Даних ключову метрику – AvgResponseTime (Середній час реагування). Можливість відстежувати цю метрику за різними вимірами (час, джерело, тип загрози) є прямим виконанням вимог ISO 27001 щодо аудиту та покращення процесів.

Крім міжнародних стандартів, вимоги до технічного захисту інформації в Україні регламентуються Нормативними документами технічного захисту інформації (НД ТЗІ) [6]. Ці документи визначають вимоги до технічних засобів, які генерують сировинні логи (наприклад, IDS/IPS), що є вихідним матеріалом для нашої аналітичної системи.

1.2.2 Порівняльний аналіз реактивних та проактивних методів захисту

Методи забезпечення безпеки обміну інформацією можна класифікувати за їхньою часовою орієнтацією: реактивні (традиційні) та проактивні (аналітичні), їхнє порівняння можемо побачити на таблиці 1.1.

Таблиця 1.1

Порівняння реактивного та проактивного підходів забезпечення безпеки

Проактивний підхід ґрунтується на постійному, глибокому аналізі логів та поведінки користувачів та систем [7]. Він вимагає інтеграції даних з різних джерел.

Критерій	**Реактивний Захист (Традиційний)**	**Проактивний Захист (Аналітичний)**
Принцип	Реагування на відомі сигнатури.	Прогнозування та виявлення аномалій.
Фокус	Захист периметра.	Аналіз вмісту та поведінки (UEBA, SIEM).
Недолік	Нездатність виявити APT-атаки; велика кількість хибних спрацювань.	Вимагає потужних аналітичних інструментів (DM, OLAP).

Перехід до проактивного методу є необхідною умовою для забезпечення надійного захисту в сучасних умовах.

1.3 Дослідження методів збору та обробки даних із систем безпеки

1.3.1 Архітектура збору даних та інтеграція SIEM-систем

Ключовою проблемою традиційних систем захисту є величезний обсяг сирих, неструктурованих даних (логів), що генеруються системами виявлення вторгнень (IDS) (рис. 1.2) , брандмауерами та серверами. У сучасній практиці, для агрегації та нормалізації цих даних використовуються SIEM-системи (Security Information and Event Management) (рис 1.3) [8], які збирають та корелюють події з різних джерел.

Рис 1.2 Приклад IDS Suricata

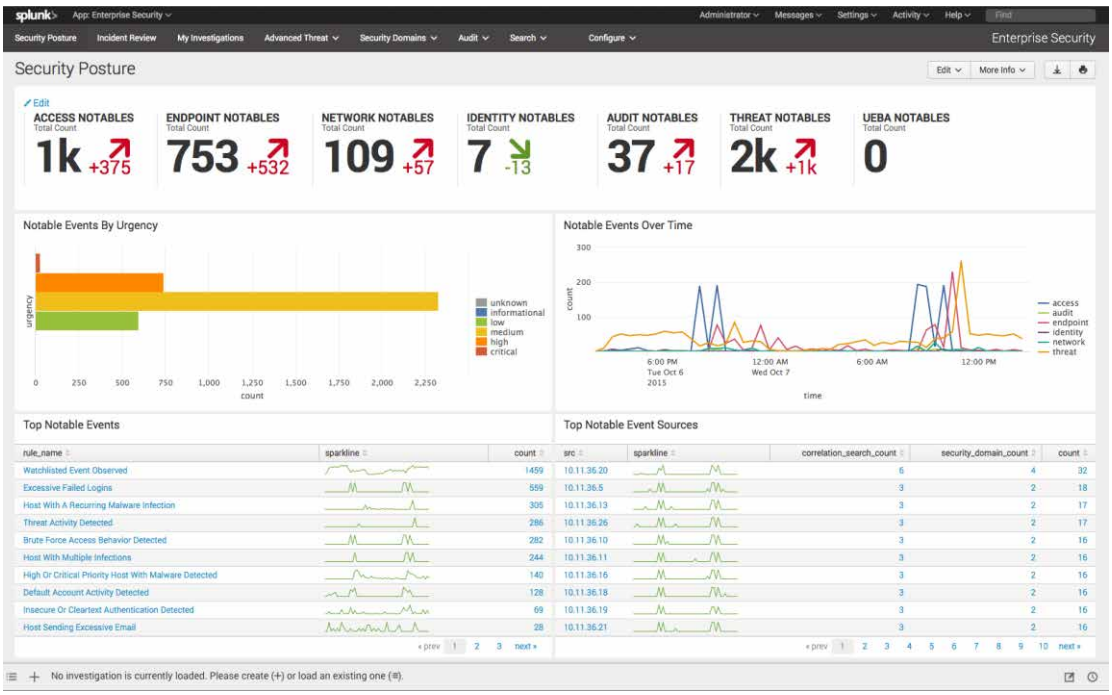


Рис 1.3 Приклад SIEM Splank

Як приклади інструментів, які генерують сирі дані та забезпечують агрегацію, можна навести систему виявлення вторгнень Suricata та аналітичну платформу Splunk [9]. Splunk , а зокрема Enteprise версія (в силу доступності) використовується для індексації та аналізу машиногенерованих даних [10], а його інтеграція з такими засобами, як Suricata, забезпечує централізований збір та первинну обробку мережевого трафіку [11].

В архітектурі, розробленій мною, логі збираються з IDS/IPS-систем, подібних до Suricata, і проходять етап агрегації, аналогічний роботі Splunk. Для уникнення складнощів графічних ETL-засобів та забезпечення максимального контролю за процесом трансформації, було обрано метод програмної реалізації ETL-логіки безпосередньо у середовищі SQL Server [12]. Це забезпечило нормалізацію та агрегацію даних, перетворивши їх на напівструктуровані поля, необхідні для подальшого аналізу.

Саме ці агреговані та нормалізовані дані слугують вихідним джерелом для завантаження в наше Сховище Даних (СД).

1.3.2 Обґрунтування програмної реалізації ETL

Процес ETL (Extract, Transform, Load) є критично важливим для СД. Він включає перетворення сирих мережевих даних у формат схеми «Зірки» [13]. З огляду на необхідність виконання специфічних для кібербезпеки трансформацій даних, які не підтримуються стандартними засобами, було обрано програмний підхід.

Такий метод дозволив вирішити основні інженерні проблеми, пов'язані з вихідною структурою логів:

- Виконати складні трансформації, необхідні для нормалізації, зокрема перетворення часових атрибутів у необхідні компоненти виміру Часу (Година, Дата).
- Забезпечити гнучкість при створенні похідних метрик, необхідних для аналізу KPI, таких як AvgResponseTime та ThreatCount.
- Гарантувати цілісність даних при завантаженні у СД, виконуючи послідовне наповнення вимірів та таблиці фактів.

1.4 Аналіз технологій OLAP та Data Mining як проактивних методів

1.4.1 Сховища Даних та OLAP-технології

Сховище Даних (СД) — це предметно-орієнтована, інтегрована, незмінна, залежна від часу колекція даних, організована для підтримки процесів прийняття управлінських рішень [14]. СД слугує основою для побудови OLAP-кубів.

Технологія OLAP (Online Analytical Processing) дозволяє здійснювати багатовимірний аналіз історичних даних [15]. У контексті ІБ, OLAP-технології використовуються для:

- Багатовимірного аудиту: Швидкий аналіз інцидентів за будь-якою комбінацією вимірів (Час vs Джерело vs Рівень Загрози).
- Оцінки метрик ефективності: Відстеження AvgResponseTime.

СД та OLAP-технології є інженерним фундаментом (Розділ 2), який перетворює логи на структуровану основу для Data Mining.

1.4.2 Технології Data Mining та їх застосування для проактивного захисту

Data Mining (DM) — це процес виявлення прихованих, невідомих, але потенційно корисних закономірностей у великих обсягах даних [16]. У сфері кібербезпеки DM застосовується для двох критичних аспектів проактивності.

Для реалізації проактивного підходу в системі було обрано два основні алгоритми, які найкраще відображають вимоги до аналізу логів безпеки.

1. Кластеризація (K-Means). Метою використання кластеризації є виявлення прихованих аномалій у поведінці джерел загроз. Алгоритм K-Means, що базується на поділі об'єктів на k кластерів із мінімізацією суми квадратів відстаней до центрів, є ідеальним для роботи з показниками, такими як FailedAuthCount. Його використання дозволяє автоматично сегментувати

IP-адреси на групи "Нормальної поведінки" та "Високого Ризику (Кластер 1)", що є недоступним для традиційного порогового моніторингу.

2. Класифікація (1-Rule). Алгоритм 1R (One Rule) є простим, але ефективним методом, який створює правила класифікації на основі одного найбільш інформативного атрибута. Його цінність полягає не лише у прогнозуванні, але й у високій інтерпретованості результатів, що є критично важливим для ухвалення рішень керівництвом. Вибір цього методу був обґрунтований необхідністю підтвердити, чи є часовий атрибут (Hour) достатньо значущим для прогнозування рівня загрози, не використовуючи складні моделі "чорної скриньки".

1.5 Постановка завдання

Необхідно розробити систему для аналізу результатів моніторингу безпеки обміну інформацією на основі логів, що надходять із систем безпеки. Для цього необхідно розробити сховище даних (DWH), здатне зберігати та структурувати ці логі, та реалізувати механізм наповнення цього сховища. Після цього, необхідно за допомогою засобів OLAP та інтелектуального аналізу даних (Data Mining) провести детальний аналіз даних, що були завантажені у сховище.

Система повинна надавати можливості для отримання відповідей на низку ключових питань:

1. Які типи загроз є найпоширенішими за певний період часу?
2. Який середній час реагування (AvgResponseTime) на загрози різних рівнів небезпеки?
3. Які джерела трафіку (IP_Address) генерують найбільшу кількість загроз?
4. Чи існує кореляція між часом доби (Hour) та рівнем загрози (ThreatLevelName)?

Розробка системи повинна проводиться у декілька етапів, що відповідають структурі магістерської роботи. Першим етапом є моделювання системи (Розділ 2). Метою цього етапу є визначення функціональних вимог (Діаграма прецедентентів) та концептуальної схеми Сховища Даних (Схема «Зірка»).

Другим етапом є розробка системи (Розділ 3). На цьому етапі необхідно реалізувати фізичну структуру СД у середовищі MS SQL Server.

Третім етапом є розробка механізму наповнення сховища даних (Розділ 3). Для цього було обрано SQL-метод, що передбачає генерацію симуляційних логів та їх трансформацію і завантаження (ETL) за допомогою SQL-скриптів. Під час

наповнення слід спочатку реалізувати наповнення таблиць вимірів, і на основі цих таблиць проводити наповнення таблиці фактів.

Четвертим етапом є проведення аналізу (Розділ 4). На основі даних, що містяться у сховищі, у середовищі Power BI (яке замінює SSAS та Reporting Services) необхідно побудувати звіти для відповіді на ключові питання. Також необхідно створити програмну реалізацію методів інтелектуального аналізу даних (KMeans та 1-Rule) та надати детальний аналіз їх результатів.

2 МОДЕЛЮВАННЯ СИСТЕМИ

2.1 Загальні положення моделювання

Моделювання є важливим етапом будь-якої розробки. Воно дозволяє зрозуміти фундаментальні особливості як певної системи, так і певного явища або технологічного процесу. Згідно класичного моделювання поняття моделі трактується як представлення об’єкта, системи чи поняття в деякій абстрактній формі, що є зручною для наукового дослідження [17].

Існує два основних підходи до проведення моделювання – структурний і функціональний. Структурний підхід використовується для виявлення елементів майбутньої системи та їх зв’язків. Функціональний підхід базується на оцінці та моделюванні саме конкретних функцій системи [18].

Для забезпечення стандартизації вигляду моделей використовується UML (Unified Modeling Language), яка надає стандартну нотацію для багатьох типів діаграм, що поділяються на три категорії: діаграми поведінки, діаграми взаємодії та структурні діаграми [19]. Для виконання етапу моделювання було використано наступні UML діаграми:

- Діаграма прецедентів
- Діаграма послідовності
- Діаграма активності

2.2 Діаграма прецедентів

2.2.1 Діаграма прецедентів

Моделювання системи є важливим етапом, що дозволяє визначити функціональні межі та структуру взаємодії. Розроблена система забезпечення безпеки обміну інформацією є прикладом аналітичної системи, що інтегрує технології оперування СД та Data Mining для проактивного виявлення загроз. Її

основне призначення полягає у перетворенні великого обсягу сирих логів, що генеруються різними вузлами корпоративної мережі, на структуровану інформацію, придатну для прийняття управлінських рішень.

Функціональні вимоги до системи були описані за допомогою діаграми прецедентів (Use-Case Diagram), яка відображає взаємодію користувачів та ключові функції системи [20]. Перелік основних прецедентів, що реалізуються системою:

- Моніторинг передачі даних: Виконується постійний збір логів мережевої активності та подій безпеки.
- Формування логів: Агрегація та первинна нормалізація зібраних даних для їх подальшого завантаження у СД.
- Виявлення загроз: Автоматизований процес аналізу даних з метою ідентифікації підозрілої активності та аномалій.
- Формування аналітичної звітності: Створення багатовимірних звітів на базі OLAP-куба, необхідних для аудиту ефективності заходів ІБ.
- Прийняття рішень щодо політики безпеки: Керівництво використовує аналітичну звітність для коригування та вдосконалення політик безпеки.

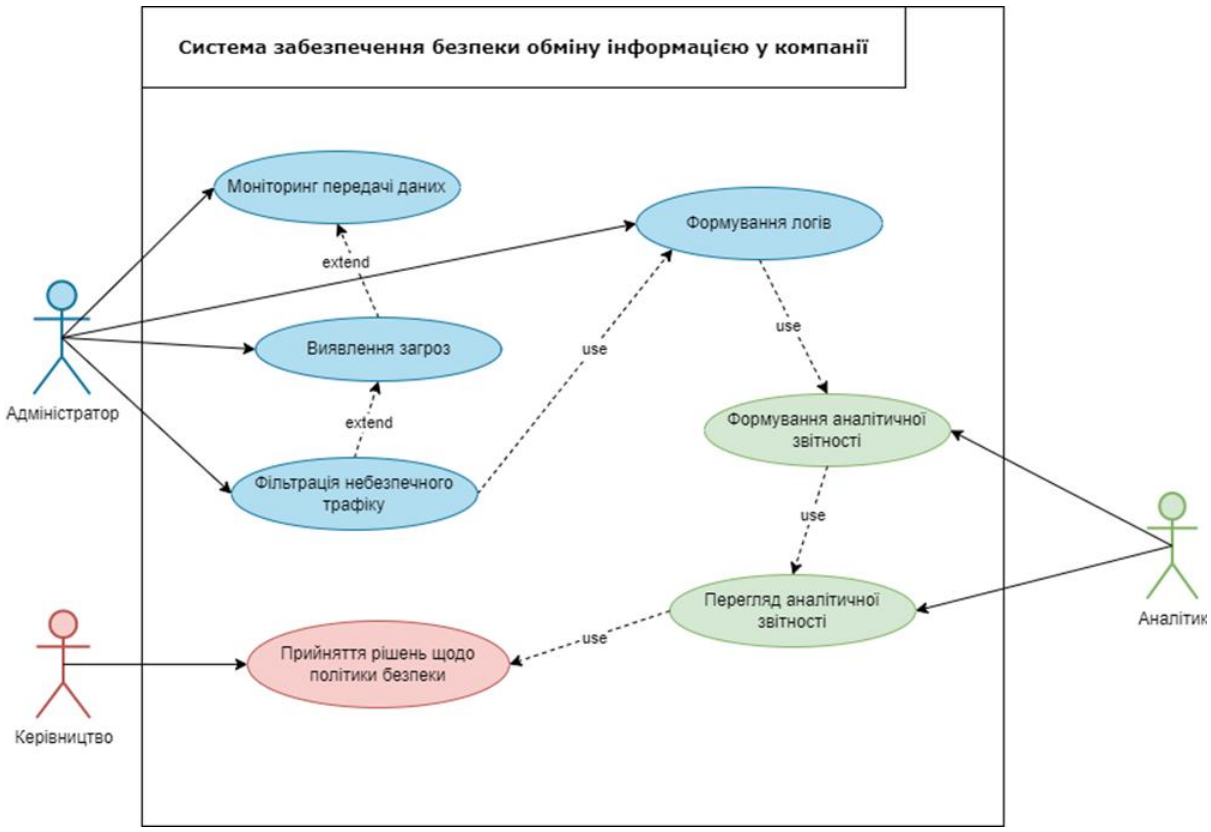


Рис 2.1 Use-Case Diagram

2.2.2 Опис акторів системи

У розробленій системі було ідентифіковано три основні ролі (актори): Адміністратор, Аналітик та Керівництво. У таблиці 2.1 наведений опис їхньої діяльності за ролями.

Таблиця 2.1

Розподіл діяльності акторів у системі

Актор	Прецедент (дія)	Опис дії
Адміністратор	Моніторинг передачі даних	Контролює мережевий трафік для виявлення підозрілих дій.
	Виявлення загроз	Аналізує трафік для ідентифікації потенційно небезпечної активності.
	Фільтрація небезпечного трафіку	Застосовує автоматичні заходи для усунення загроз шляхом блокування підозрілих дій.
	Формування логів	Здійснює запис усіх дій, пов'язаних із моніторингом, виявленням та фільтрацією трафіку.
Аналітик з інформаційної безпеки	Формування та перегляд аналітичної звітності	Створює звіти на основі зібраних логів для аналізу загроз.
		Аналізує звіти для виявлення тенденцій загроз і 'вузьких місць'.
Керівництво компанії	Прийняття рішень щодо політики безпеки	Оцінює результати аналітики та ухвалює рішення щодо вдосконалення стратегії безпеки.

2.3 Діаграма послідовності

Діаграма послідовності (Sequence Diagram) – тип діаграми взаємодії, що використовується для моделювання логіки сценаріїв використання [21]. Цей тип діаграм показує, яка інформація передається між об’єктами в системі під час виконання певного сценарію. Для моделювання системи було розроблено діаграму (рис. 2.2), що ілюструє ключовий процес – завантаження даних у сховище даних (ETL).

В якості об’єктів було виділено Програмний ETL-модуль, Базу даних Staging та СД. Розглянемо їх детальніше:

Програмний ETL-модуль (SQL-скрипти): У даній архітектурі це не окремий програмний компонент, а логічна сутність, що представляє набір SQL-скриптів, які виконуються Адміністратором у середовищі SSMS. Цей модуль виступає "рушієм", який ініціює всі процеси трансформації та завантаження даних.

База даних Staging (Raw_Log_Data): Проміжний вузол зберігання. Ця таблиця є джерелом (Extract) для ETL-процесу. Вона отримує сирі, згенеровані дані, які імітують логі систем безпеки.

СД (Виміри): Представляє набір таблиць вимірів у DWH (DimensionTime, DimensionSource тощо). Цей об'єкт отримує повідомлення "Lookup" (пошук ID) та "Insert" (вставка нових унікальних записів).

СД (Факти): Представляє центральну таблицю FactThreats. Цей об'єкт є кінцевим отримувачем даних, який фіксує метрики та всі отримані ID-ключі (Load).

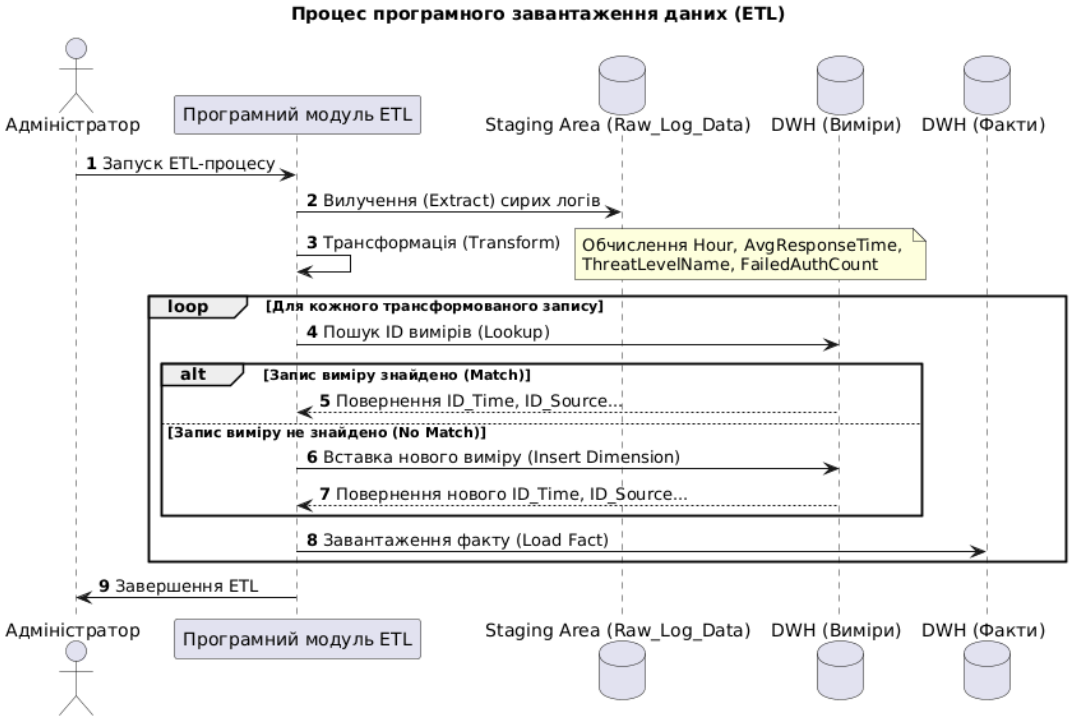


Рис. 2.2 Sequence Diagram

2.4 Діаграма активності

Діаграма активності (Activity Diagram) використовується для моделювання покрокової логіки виконання складного процесу. Для моделювання логіки проактивного аналізу та ідентифікації аномалій (ключова функція Data Mining) було розроблено діаграму активності (рис. 2.3).

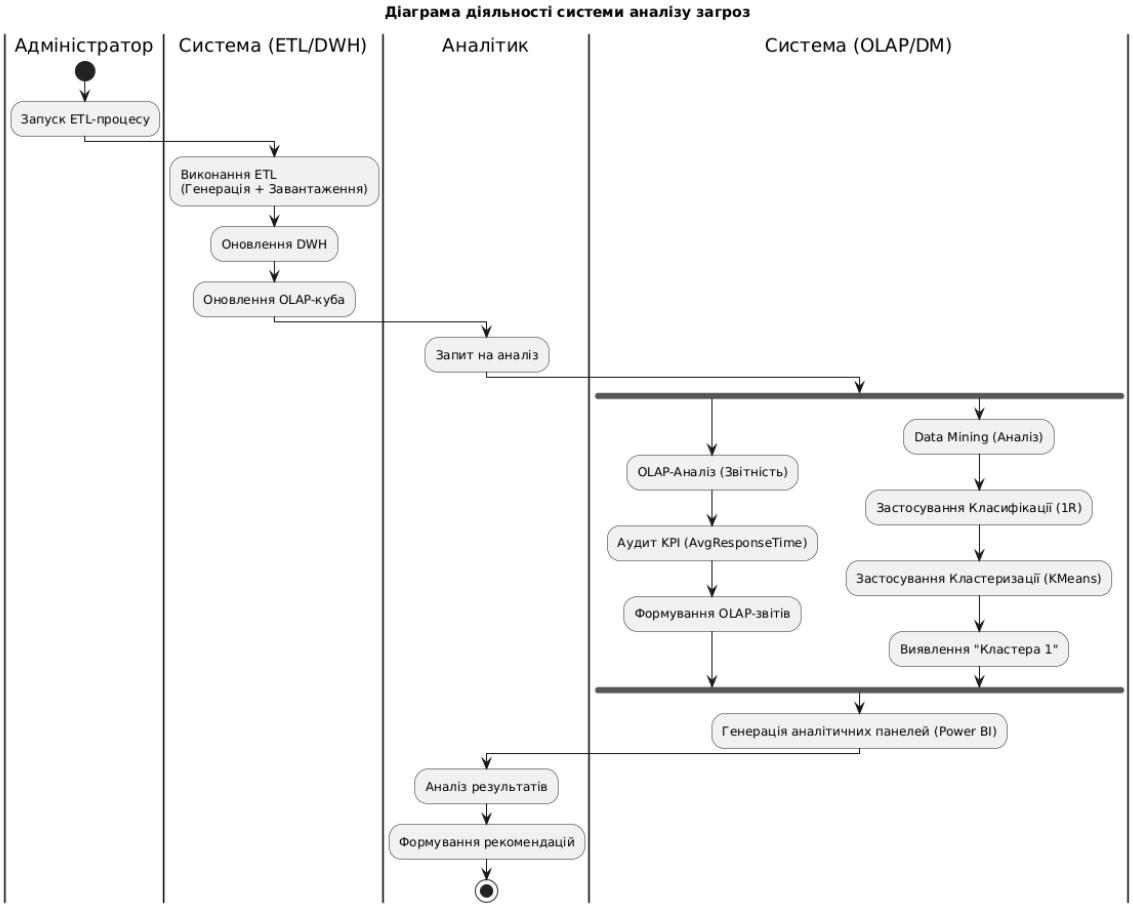


Рис. 2.3 Activity Diagram

2.5 Моделювання інформаційної предметної області

2.5.1 Аналіз вихідних даних

Вихідними даними для системи є журнали подій із систем безпеки (брандмауери, IDS/IPS, сервери). Ці сирі дані є неструктурованими і генеруються з високою швидкістю, що унеможливлює їх прямий аналіз. Для демонстрації ефективності проактивного методу було використано агреговані логи, які містять

необхідні для DM-аналізу атрибути. Зокрема, аналіз вимагає наявності часової мітки, ідентифікатора джерела, типу загрози, а також метрик для кластеризації (наприклад, FailedAuthCount).

2.5.2 Постановка задачі моделювання СД

Основним завданням етапу моделювання є перетворення реляційної (або напівструктурованої) моделі вихідних логів у багатовимірну модель СД. Це є необхідним кроком для підтримки OLAP-запитів, які є неефективними у традиційних реляційних базах даних.

Сховище даних є централізованим репозиторієм інформації, який використовується для аналізу кіберзагроз та їхнього впливу на систему безпеки компанії.

Дані надходять до сховища з різних джерел (в нашому випадку із симульованого). Аналітики безпеки, інженери даних та керівники інформаційної безпеки можуть використовувати ці дані для аналізу загроз, оцінки рівня безпеки та оптимізації заходів реагування.

3 РОЗРОБКА СИСТЕМИ

3.1 Архітектура та топологія системи

Архітектура аналітичної системи (рис 3.1) побудована за принципом трьох вузлів, що забезпечує гнучкість, масштабованість та чітке розділення функціональних обов'язків (розробка, зберігання, аналіз).

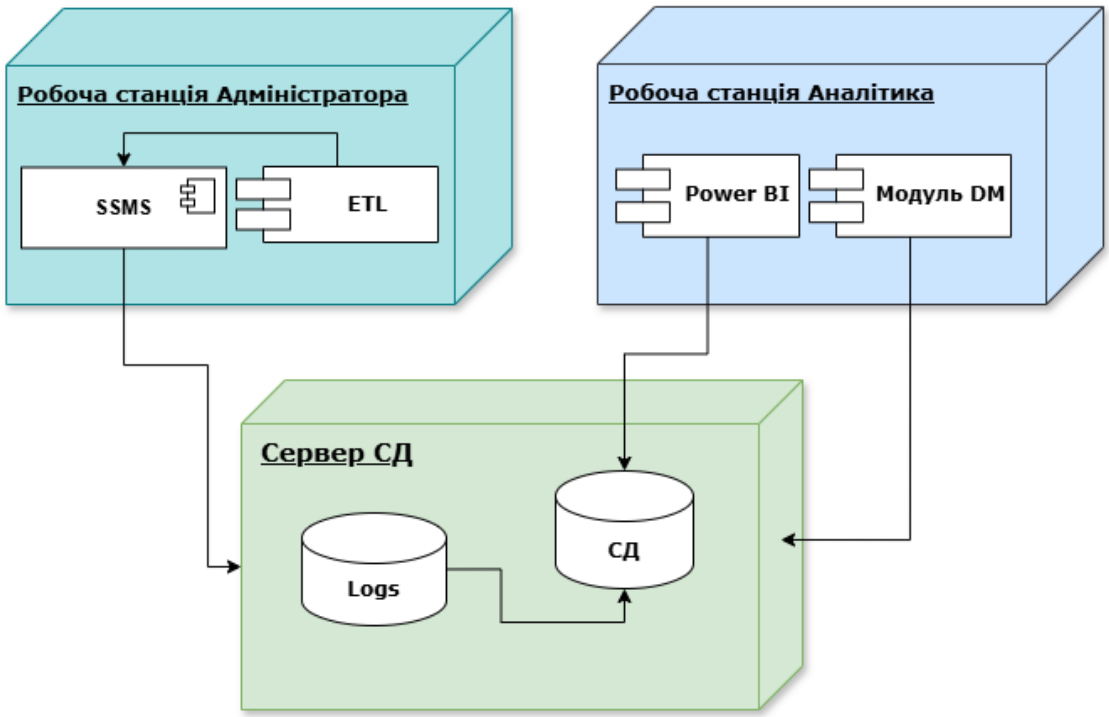


Рис. 3.1 Архітектура аналітичної системи

Створена топологія системи дозволяє ефективно інтегрувати логістику мережевих подій з аналітичними процесами. Ключові компоненти архітектури:

1. Робоча станція Адміністратора: Цей вузол містить середовище SSMS та модуль ETL. Адміністратор відповідає за запуск SQL-скриптів, які виконують генерацію симуляційних логів та їх подальше завантаження у СД.

2. Сервер СД : Це центральний вузол системи, який виконує функції зберігання та обробки. Він містить Staging Area (Raw_Log_Data), куди завантажуються згенеровані дані, та фінальне СД.
3. Робоча станція Аналітика: Цей вузол відповідає за аналіз та візуалізацію. Він містить Аналітичний пакет (Power BI), який підключається безпосередньо до СД для OLAP-аналізу та звітності, та Модуль DM (Python), який використовує дані з СД для виконання Кластеризації та Класифікації.

3.2 Проєктування СД

3.2.1 Обґрунтування схеми «Зірка»

Для забезпечення ефективного аналізу великих обсягів логів було обрано концептуальну модель Схема «Зірка» (Star Schema). Ця модель є найбільш оптимальною для багатовимірного аналізу (OLAP), оскільки вона мінімізує складність запитів та прискорює агрегацію даних. Схема сховища даних складається з центральної таблиці фактів (Fact Table) та пов’язаних з нею вимірних таблиць (Dimension Tables). Це забезпечує, що вся контекстна інформація (IP, час, рівень) зберігається в денормалізованому вигляді, а таблиця фактів містить лише числові метрики та зовнішні ключі.

На рис. 3.2 наведено схему сховища даних для аналізу кіберзагроз, що включає інформацію про джерела атак, їхні типи, рівень небезпеки, а також часові характеристики.

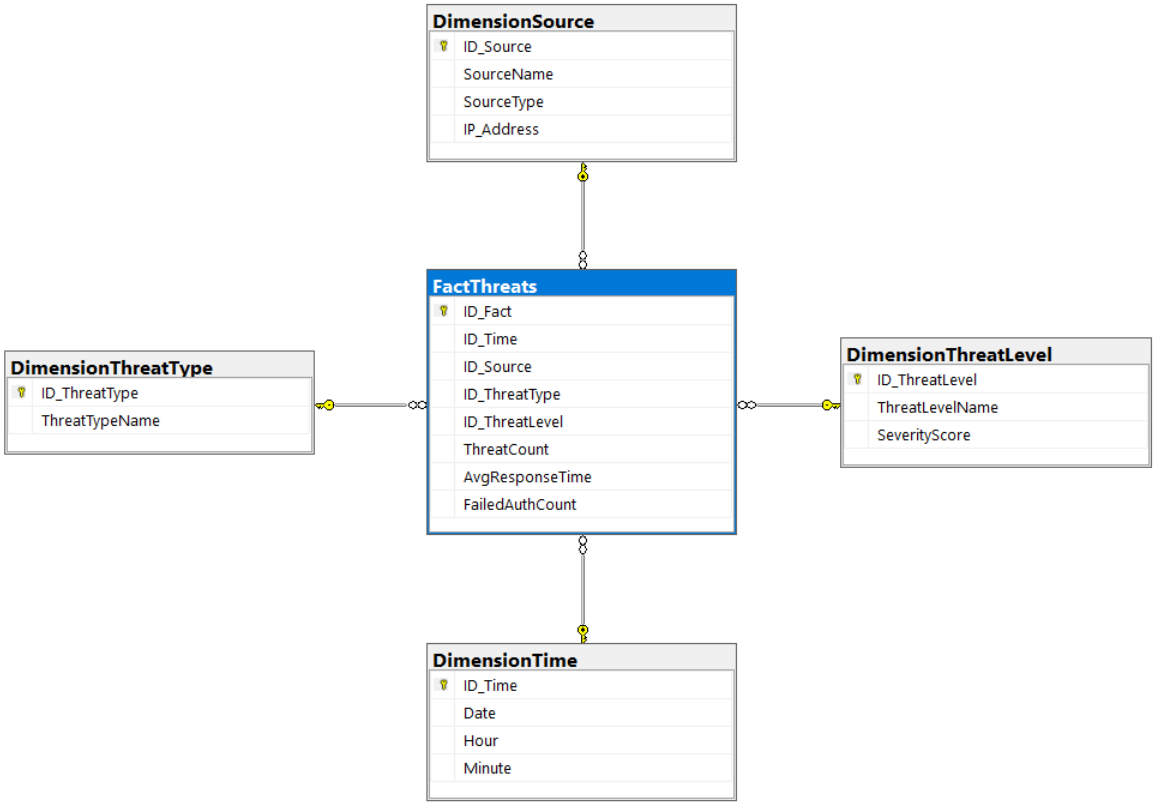


Рис. 3.2 Сховище даних

3.2.2 Опис схеми СД та атрибутів

Центральним елементом моделі є Таблиця Фактів (FactThreats), яка містить ключові метрики та зовнішні ключі до вимірних таблиць, їхній перелік та опис:

- ID_Time – Час фіксації загрози (зв'язок із DimensionTime).
- ID_Source – Джерело загрози (зв'язок із DimensionSource).
- ID_ThreatType – Тип загрози (зв'язок із DimensionThreatType).
- ID_ThreatLevel – Рівень небезпеки загрози (зв'язок із DimensionThreatLevel).
- ThreatCount – Кількість зафіксованих подібних загроз.
- AvgResponseTime– Середній час реагування на цю загрозу

Далі описано вимірні таблиці із посиланням на їх використання у DM-аналізі:

1. DimensionTime: Містить атрибути, необхідні для часового аналізу, зокрема Hour (Година доби), що є ключовим предиктором для алгоритму 1-Rule.

2. DimensionSource: Містить атрибути, які ідентифікують джерело (IP_Address, Source_Type, SourceName), необхідні для кластеризації KMeans.
3. DimensionThreatType: Містить назву типу загрози (ThreatTypeName).
4. DimensionThreatLevel: Містить назву рівня (ThreatLevelName, із градацією – High, Medium, Low) та оцінку серйозності (SeverityScore).

3.2.3 Створення об’єктів СД

На першому етапі розробки системи було створено фізичну структуру СД на платформі SSMS. Скрипт виконує створення всіх необхідних таблиць вимірів (DimensionTime, DimensionSource та ін.) (рис. 3.3) та таблиці фактів (FactThreats) (рис 3.4), а також проміжної таблиці Raw_Log_Data (Staging Area).

```
-- 1.1. DimensionTime (Час)
CREATE TABLE DimensionTime (
    ID_Time INT IDENTITY(1,1) PRIMARY KEY,
    Date INT NOT NULL,
    Hour INT NOT NULL,
    Minute INT NULL,
    CONSTRAINT UQ_DimTime UNIQUE (Date, Hour)
);

-- 1.2. DimensionSource (Джерело)
CREATE TABLE DimensionSource (
    ID_Source INT IDENTITY(1,1) PRIMARY KEY,
    SourceName NVARCHAR(100) NULL,
    SourceType NVARCHAR(100) NULL,
    IP_Address NVARCHAR(50) NOT NULL,
    CONSTRAINT UQ_DimSource UNIQUE (IP_Address)
);

-- 1.3. DimensionThreatLevel (Рівень Загрози)
CREATE TABLE DimensionThreatLevel (
    ID_ThreatLevel INT IDENTITY(1,1) PRIMARY KEY,
    ThreatLevelName NVARCHAR(50) NOT NULL,
    SeverityScore INT NOT NULL,
    CONSTRAINT UQ_DimLevel UNIQUE (ThreatLevelName)
);

-- 1.4. DimensionThreatType (Тип Загрози)
CREATE TABLE DimensionThreatType (
    ID_ThreatType INT IDENTITY(1,1) PRIMARY KEY,
    ThreatTypeName NVARCHAR(100) NOT NULL,
    CONSTRAINT UQ_DimType UNIQUE (ThreatTypeName)
);
```

Рис. 3.3 Фрагмент SQL-скрипту (створення таблиць вимірів)

```sql
-- 1.5. FactThreats (Факти загроз)
CREATE TABLE FactThreats (
    ID_Fact INT IDENTITY(1,1) PRIMARY KEY,
    ID_Time INT NOT NULL,
    ID_Source INT NOT NULL,
    ID_ThreatType INT NOT NULL,
    ID_ThreatLevel INT NOT NULL,
    ThreatCount INT NOT NULL,
    AvgResponseTime FLOAT NOT NULL,
    FailedAuthCount INT NULL,

    FOREIGN KEY (ID_Time) REFERENCES DimensionTime(ID_Time),
    FOREIGN KEY (ID_Source) REFERENCES DimensionSource(ID_Source),
    FOREIGN KEY (ID_ThreatType) REFERENCES DimensionThreatType(ID_ThreatType),
    FOREIGN KEY (ID_ThreatLevel) REFERENCES DimensionThreatLevel(ID_ThreatLevel)
);

-- 1.6. Таблиця Staging (Вхідні дані)
CREATE TABLE Raw_Log_Data (
    Start_time BIGINT,
    Source_IP NVARCHAR(50),
    Threat_Level NVARCHAR(50),
    Threat_Type_Name NVARCHAR(100),
    Failed_Auth_Count INT,
    Threat_Raw_Count INT,
    Response_Time_Sec FLOAT
);
```

Рис. 3.4 Фрагмент SQL-скрипту (створення таблиці фактів та вхідних даних)

3.3 Реалізація процесів ETL та завантаження агрегованих логів

3.3.1 Обґрунтування SQL-методу реалізації ETL

Процес ETL (Extract, Transform, Load) є критично важливим для СД. Він включає перетворення сирих мережевих даних у формат схеми «Зірки». Хоча для цих завдань часто використовуються графічні засоби, такі як SSIS (SQL Server Integration Services), їх налаштування для роботи зі специфічними типами даних або складними трансформаціями є невиправдано ускладненим [22].

З огляду на необхідність повного контролю над процесом трансформації, забезпечення цілісності даних та інтеграції логіки генерації симуляційних даних, було обрано програмний підхід. ETL-логіка була реалізована безпосередньо у середовищі SQL Server за допомогою набору оптимізованих SQL-скриптів.

Такий метод дозволив вирішити основні інженерні проблеми, пов'язані з вихідною структурою логів:

- Виконати складні трансформації, необхідні для нормалізації, зокрема перетворення часових атрибутів у необхідні компоненти виміру Time (Hour, Date).
- Забезпечити гнучкість при створенні похідних метрик, необхідних для аналізу KPI, таких як AvgResponseTime та ThreatCount.
- Гарантувати цілісність даних при завантаженні у СД, виконуючи послідовне наповнення вимірів та таблиці фактів.

3.3.2 Алгоритм генерації та наповнення СД

Наповнення СД здійснюється за допомогою SQL-скриптів, які виконують три послідовні кроки: генерацію симуляційних даних, завантаження вимірів та завантаження фактів.

Крок 1: Генерація вихідних даних (Staging Area). Перший скрипт, що наведений на рис. 3.5, використовує цикл WHILE для створення 500+ записів у проміжній таблиці Raw_Log_Data. Цей скрипт вбудовує необхідні для Розділу 4 закономірності Data Mining:

1. Кореляція 1R: Концентрація загроз рівня 'High' у нічні години (0, 1, 2).
2. Аномалії KMeans: Призначення аномально високих значень FailedAuthCount для специфічних IP-адрес ("Кластер 1").

Крок 2: ETL та Послідовне Завантаження Вимірів (Dimension Load). Другий скрипт, наведений в Додатку Б, використовує логіку INSERT INTO... SELECT DISTINCT для вилучення унікальних значень із Raw_Log_Data та їх завантаження у відповідні таблиці Dimensions. Це імітує логіку Lookup та забезпечує, що кожен вимір містить лише унікальні записи, а SQL Server автоматично генерує для них ID-ключі.

Крок 3: ETL та Фіксація Факту (Fact Load). Третій, фінальний крок, фрагмент коду якого наведений в Додатку Б використовує операцію JOIN для з'єднання таблиці Raw_Log_Data з усіма заповненими таблицями вимірів. Це дозволяє

отримати необхідні сурогатні ID-ключі (ID_Time, ID_Source, ID_ThreatType, ID_ThreatLevel) та зафіксувати їх у таблиці фактів FactThreats разом із розрахованими метриками.

4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

4.1 Засоби OLAP-аналізу

4.1.1 Побудова звітності в середовищі Power BI

Для побудови звітів було обрано середовище MS Power BI. Це аналітичне середовище, яке дає можливість легкого підключення до джерел інформації, об'єднання даних у модель та побудови візуальних графіків [23].

У даній роботі Power BI підключається безпосередньо до СД у SQL Server. Power BI автоматично розпізнає зв'язки між таблицею фактів та таблицями вимірів як можемо бачити на рис. 4.1, що дозволяє проводити багатовимірний аналіз.

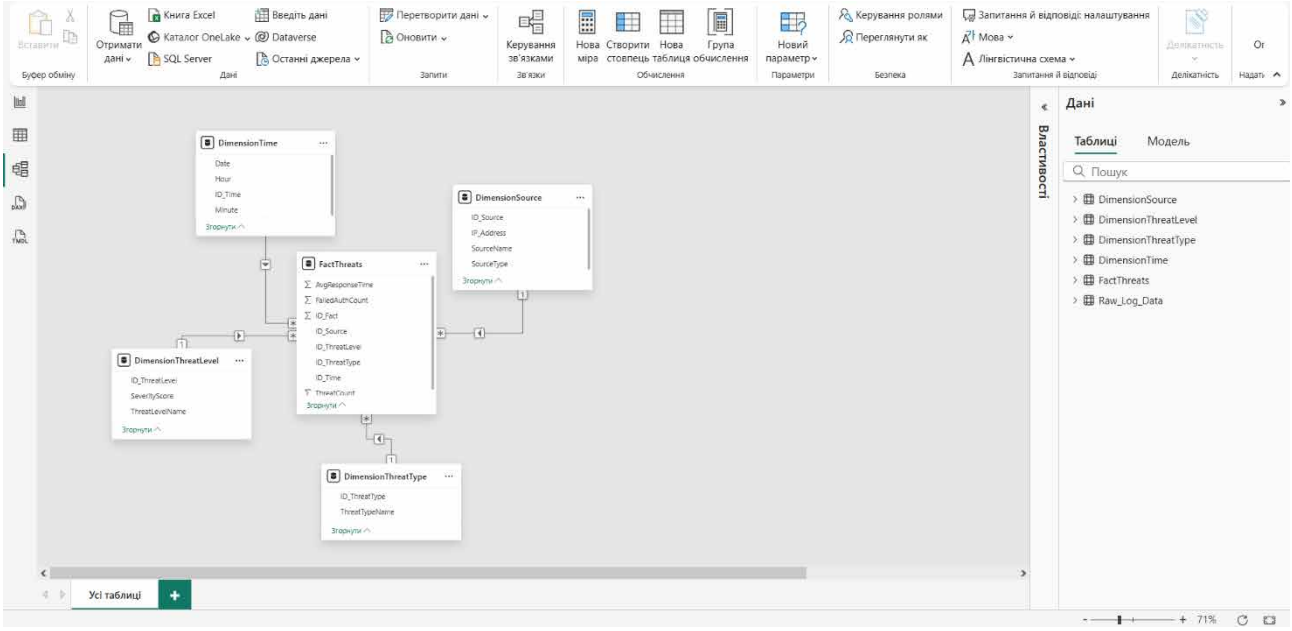


Рисунок 4.1 Модель СД у Power BI

Далі перейдемо безпосередньо до звітів. Було створено три ключові звіти у вигляді графіків, які відповідають на питання, поставлені у завданні дослідження.

Перший звіт (рис. 4.2) відповідає на питання "Які типи загроз є найпоширенішими?".

Рис. 4.2 Кругова діаграма розподілу загроз за типом

Проаналізувавши діаграму, видно, що найбільша частка загроз припадає на Reconnaissance, тобто розвідувальний тип (що відповідає нашим симуляційним даним), за ним ідуть Brute Force та Malware Scan. З огляду на це дослідження умовному аналітику це дає можливість проаналізувати та сфокусувати зусилля на захисті від конкретних типів атак.

Другий звіт (рис. 4.3) демонструє розподіл загроз за часом доби, що є ключовим для виявлення тенденцій атак.

Рис. 4.3 Розподіл ThreatCount за DimensionTime.Hour

Проаналізувавши діаграму, видно, що найбільша частка загроз (особливо рівня 'High') припадає на неробочі години (00:00 – 02:00), що підтверджує необхідність автоматизованого проактивного моніторингу.

Третій звіт (рис. 4.4) демонструє аналіз ключового показника ефективності середнього часу реагування (AvgResponseTime) у розрізі рівня критичності загрози.

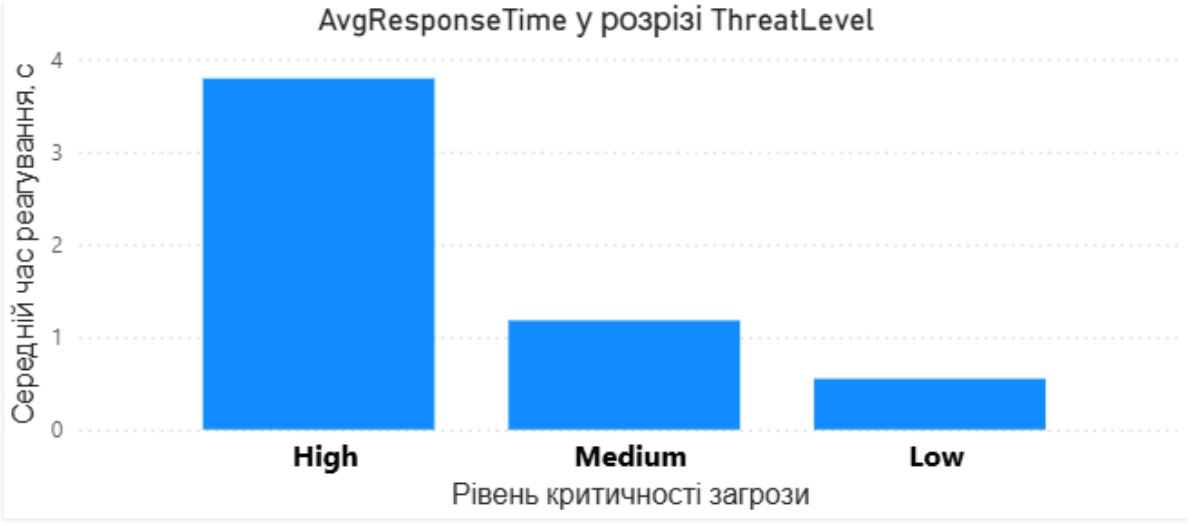


Рис. 4.4 AvgResponseTime(у секундах) за ThreatLevelName

Як бачимо, показники середнього часу реагування залежать від рівня критичності загрози. Протилежна логіці ситуація, де загрози рівня High мають найвищий показник AvgResponseTime, свідчить про системну проблему у процесі реагування. Ці інциденти є найбільш комплексними (наприклад, тривалі Brute Force або APT) і вимагають значно більше часу на аналіз та локалізацію порівняно з автоматично блокованими загрозами рівня Low.

4.2 Результати інтелектуального аналізу (Data Mining)

4.2.1 Кластеризація (Метод KMeans)

Кластеризація – це метод, що дозволяє проводити групування даних без попередньої інформації про їх належність. Основною ідеєю є об’єднати схожі об’єкти у кластери [24].

Для визначення оптимальної кількості кластерів було застосовано метод “лікоть” (Elbow method). На графіку залежності інерції від кількості кластерів видно, що найвиразніше “плече” спостерігається при k = 2. Це свідчить про те, що поділ на два кластери є оптимальним, нижче наведено приклад використання на рис. 4.4.

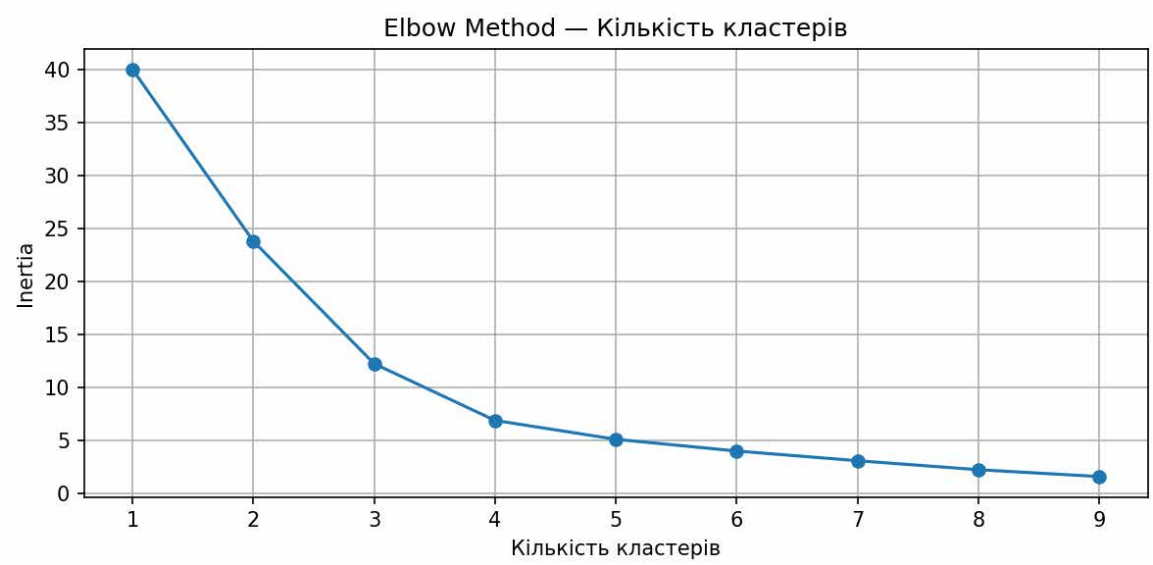


Рис. 4.5 Графік методу “лікоть” для визначення оптимальної кількості кластерів.

Рисунок показує, що при k = 2 інерція зменшується найзначніше, далі зниження вже не таке суттєве.

У межах роботи було використано алгоритм K-середніх (K-means). Цей метод базується на розділенні об'єктів на кластери з певним центром.

Для аналізу було взято дані з СД, зокрема ознаки IP_Address (як ідентифікатор) та FailedAuthCount (як ключова метрика аномалії).

Після того як кількість кластерів визначено, проводиться завантаження даних у компонент DataFrame та їх стандартизація.

Результати кластеризації (рис. 4.6) дозволили виявити два основні кластери:

- Кластер 0 (Нормальна поведінка): Включає більшість IP-адрес із низьким рівнем FailedAuthCount (0-5).
- Кластер 1 (Високий Ризик/Аномалія): Включає невелику групу IP-адрес (192.168.1.72 та ін.), які характеризуються аномально високим показником FailedAuthCount (30-80).

Після визначення оптимального значення k, було здійснено кластеризацію даних із використанням KMeans. Для візуалізації результатів було застосовано метод головних компонент (PCA), що дозволяє знизити розмірність до двох

головних компонент та наочно відобразити поділ загроз на кластери, нижче наведено приклад використання на рис. 4.6.

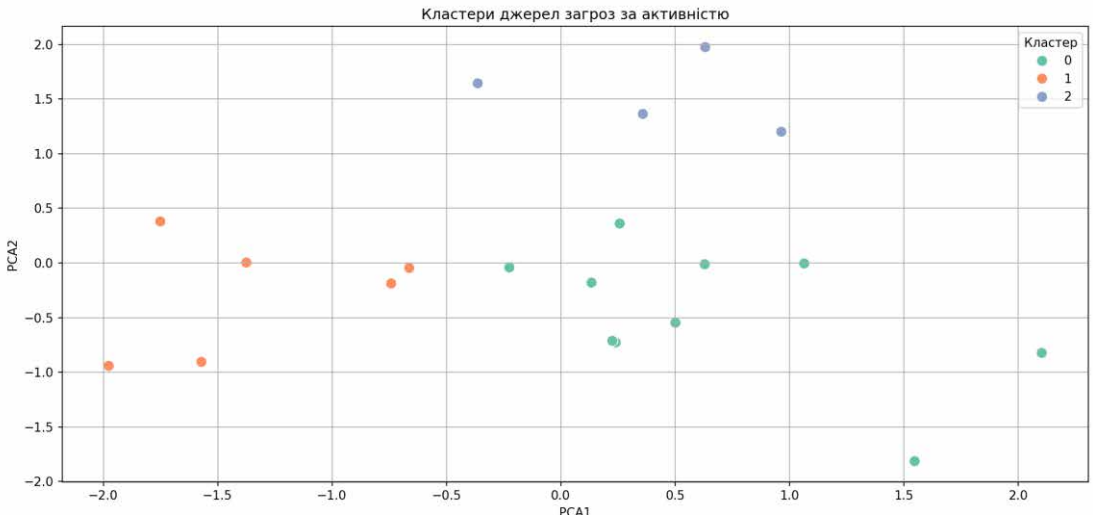


Рис. 4.6 Графік кластеризації FailedAuthCount по IP_Address

На графіку видно наступне:

Кластер 0 (зелений) – активні користувачі з великою кількістю дій і низьким рівнем помилок при вході. Найімовірніше, це штатні працівники або системні адміністратори, які регулярно взаємодіють із системою.

Кластер 1 (помаранчевий) – користувачі з високим рівнем невдалих спроб авторизації. Це може свідчити про спроби несанкціонованого доступу, атаки методом підбору пароля або використання вразливих облікових даних.

Кластер 2 (синій) – користувачі з помірною активністю та стабільною поведінкою. Вони не демонструють ознак аномальної активності, однак залишаються об’єктами стандартної політики безпеки.

4.2.2 Класифікація (Алгоритм 1Rule)

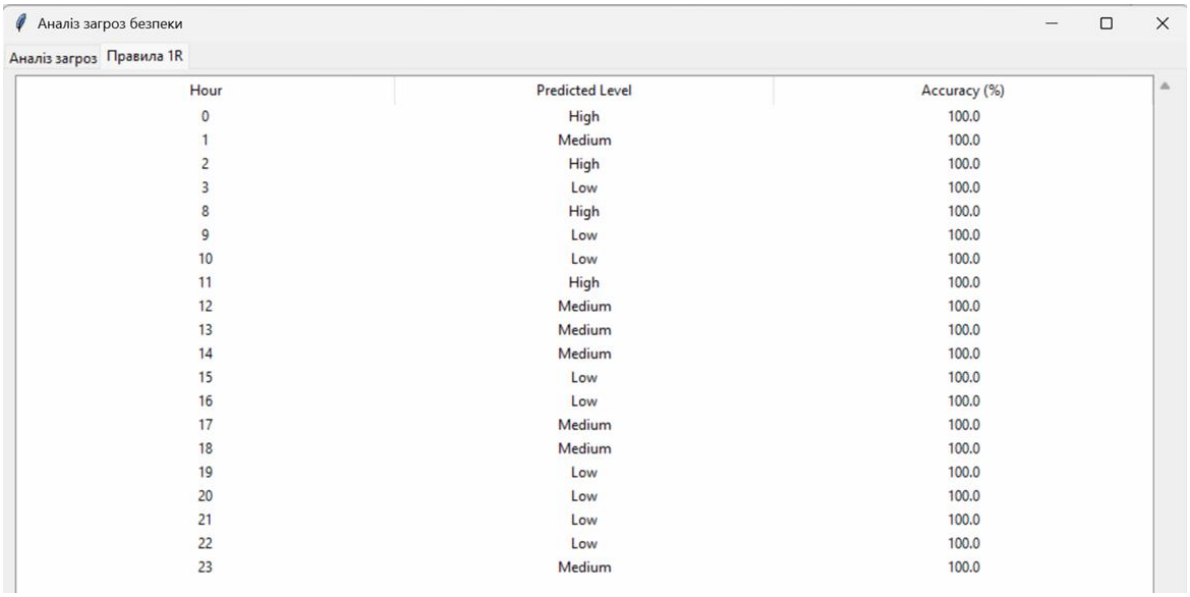
У межах дослідження було реалізовано класифікацію рівня небезпеки загроз (ThreatLevelName) за допомогою алгоритму 1R (One Rule). Для побудови класифікаційних правил було використано одну незалежну змінну – Hour (година виникнення загрози), яка виявилася найбільш інформативною серед інших доступних змінних.

Сутність методу полягає у тому, що алгоритм 1R є одним із найпростіших підходів до класифікації, який створює правила на основі лише одного атрибута [25]. Він аналізує всі можливі значення ознаки, підраховує кількість прикладів кожного класу для кожного значення та формує правило, що присвоює найбільш імовірний клас відповідному значенню.

Для класифікації було використано агреговані дані з СД, зокрема:

- час виникнення загрози (Hour),
- оцінку важкості (SeverityScore),
- кількість виявлених загроз (ThreatCount),
- джерело надходження загрози (SourceName),
- тип загрози (ThreatTypeName),
- рівень загрози (ThreatLevelName).

При побудові правил, аналіз показав, що атрибут "Hour" найкраще підходить для класифікації рівня небезпеки загроз. Для кожної години було визначено клас загрози, який найчастіше зустрічається у відповідний період. У результаті було сформовано прості правила класифікації з високою точністю (рис. 4.7).



Hour	Predicted Level	Accuracy (%)
0	High	100.0
1	Medium	100.0
2	High	100.0
3	Low	100.0
8	High	100.0
9	Low	100.0
10	Low	100.0
11	High	100.0
12	Medium	100.0
13	Medium	100.0
14	Medium	100.0
15	Low	100.0
16	Low	100.0
17	Medium	100.0
18	Medium	100.0
19	Low	100.0
20	Low	100.0
21	Low	100.0
22	Low	100.0
23	Medium	100.0

Рис. 4.7 Таблиця правил класифікації за алгоритмом 1R

Отож, проаналізуємо результати. Після застосування алгоритму 1R до вибірки даних про загрози безпеки було встановлено, що найінформативнішим атрибутом для передбачення рівня небезпеки є час виникнення загрози. Зокрема:

- У період з 00:00 до 02:00 переважають загрози високого рівня (High) — точність передбачення 100%.
- У денний час (наприклад, о 12:00–14:00) часто зустрічаються загрози середнього рівня (Medium) — також із точністю 100%.
- Найнижчий рівень небезпеки (Low) фіксується вночі або надвечір (наприклад, о 3:00, 8:00, 15:00–21:00) — з такою ж високою точністю класифікації.

Цей аналіз підтвердив гіпотезу, що Година доби є найбільш значущим атрибутом для прогнозування рівня загрози, що дозволяє створювати адаптивні політики моніторингу

Цей аналіз підтвердив гіпотезу, що Година доби є найбільш значущим атрибутом для прогнозування рівня загрози, що дозволяє створювати адаптивні політики моніторингу.

4.2.3 Використання методу наївного Байеса

Метою є реалізація задачі класифікації рівня небезпеки загроз безпеки з використанням алгоритму Naive Bayes [26]. Аналіз проводився на основі даних із СД, обробка та візуалізація — у середовищі Power BI Desktop з інтеграцією Python-скриптів.

На етапі підготовки дані були завантажені SQL-запитом, який об'єднує кілька вимірів:

- ThreatCount — кількість зафіксованих загроз;
- AvgResponseTime — середній час реагування;
- SeverityScore — оціночний бал серйозності загрози;
- метадані джерела загрози: назва, IP-адреса, тип загрози, рівень загрози

На основі середнього значення SeverityScore було створено цільову змінну Risk_Level, яка класифікує загрози як:

- Dangerous — якщо значення вище середнього;
- Controlled — якщо значення нижче середнього.

Для класифікації було використано алгоритм Gaussian Naive Bayes з бібліотеки scikit-learn.

Основні ознаки:

- ThreatCount
- AvgResponseTime
- SeverityScore

Розподіл даних:

- 70% — тренувальна вибірка;
- 30% — тестова вибірка.

Це свідчить про високу якість класифікації в рамках вибраної вибірки, нижче наведено приклад використання на рис. 4.8

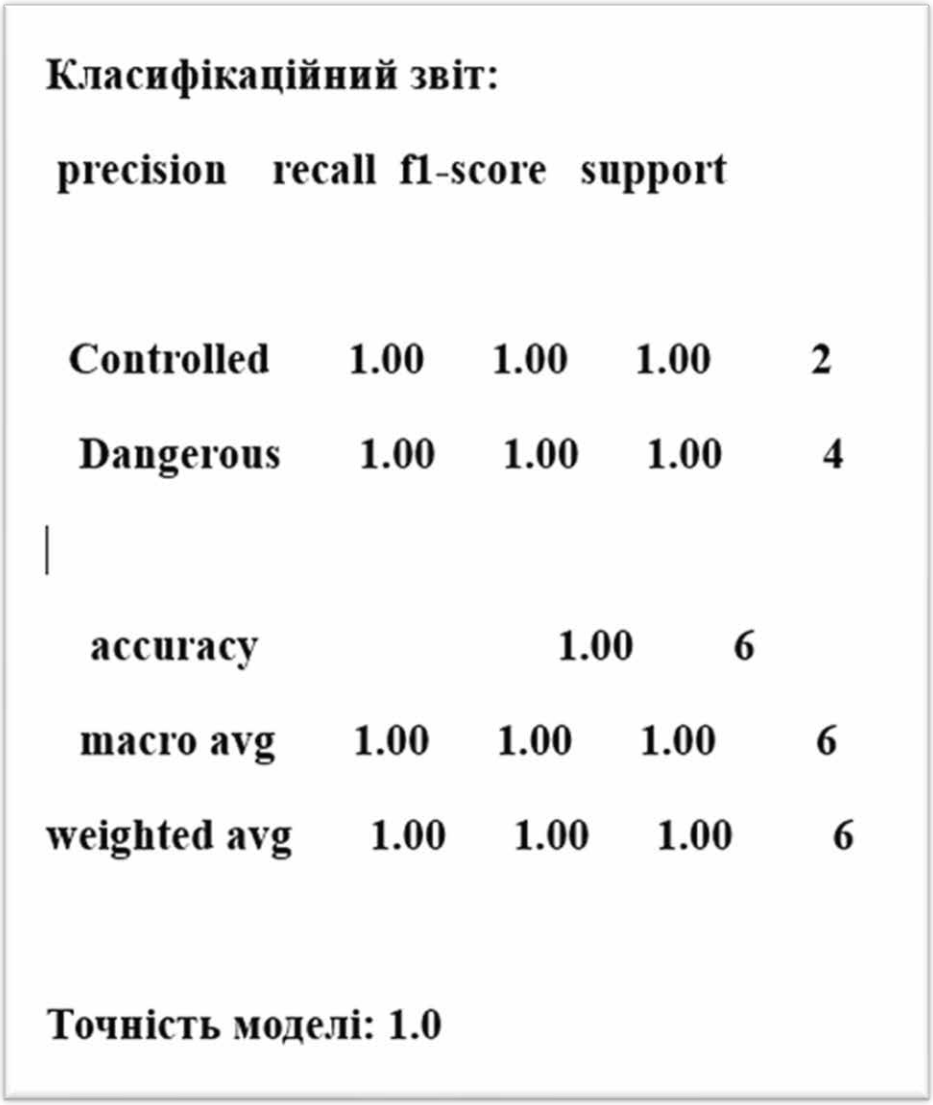
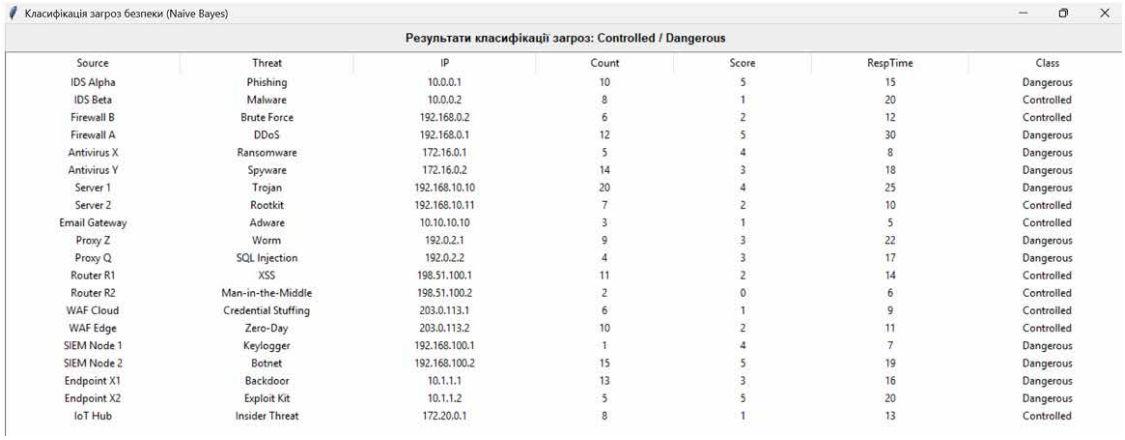


Рис. 4.8 Класифікаційний звіт після навчання моделі

Результати класифікації відображаються у графічному інтерфейсі, створеному за допомогою Tkinter [27]. Користувач бачить:

- назву джерела загроз;
- тип загрози;
- IP-адресу;
- кількість загроз, оцінку серйозності та час реагування;
- прогнозований клас (Controlled або Dangerous).

Нижче наведено приклад використання на рис. 4.9.



Source	Threat	IP	Count	Score	RespTime	Class
IDS Alpha	Phishing	10.0.0.1	10	5	15	Dangerous
IDS Beta	Malware	10.0.0.2	8	1	20	Controlled
Firewall B	Brute Force	192.168.0.2	6	2	12	Controlled
Firewall A	DDoS	192.168.0.1	12	5	30	Dangerous
Antivirus X	Ransomware	172.16.0.1	5	4	8	Dangerous
Antivirus Y	Spyware	172.16.0.2	14	3	18	Dangerous
Server 1	Trojan	192.168.10.10	20	4	25	Dangerous
Server 2	Rootkit	192.168.10.11	7	2	10	Controlled
Email Gateway	Adware	10.10.10.10	3	1	5	Controlled
Proxy Z	Worm	192.0.2.1	9	3	22	Dangerous
Proxy Q	SQL Injection	192.0.2.2	4	3	17	Dangerous
Router R1	XSS	198.51.100.1	11	2	14	Controlled
Router R2	Man-in-the-Middle	198.51.100.2	2	0	6	Controlled
WAF Cloud	Credential Stuffing	203.0.113.1	6	1	9	Controlled
WAF Edge	Zero-Day	203.0.113.2	10	2	11	Controlled
SIEM Node 1	Keylogger	192.168.100.1	1	4	7	Dangerous
SIEM Node 2	Botnet	192.168.100.2	15	5	19	Dangerous
Endpoint X1	Backdoor	10.1.1.1	13	3	16	Dangerous
Endpoint X2	Exploit Kit	10.1.1.2	5	5	20	Dangerous
IoT Hub	Insider Threat	172.20.0.1	8	1	13	Controlled

Рис. 4.9 Інтерфейс класифікації загроз з використанням Naive Bayes

ВИСНОВКИ

В рамках цієї магістерської кваліфікаційної роботи була успішно реалізована та досліджена Система забезпечення безпеки обміну інформацією у компанії на основі інтеграції технологій СД, OLAP-аналізу та Інтелектуального аналізу даних (Data Mining). Основна мета роботи, що полягала у створенні ефективного аналітичного інструменту для проактивного моніторингу та прогнозування загроз, була повністю досягнута.

1. Узагальнення теоретичних та інженерних результатів

На початковому етапі було проведено ґрунтовний системний аналіз сучасних кіберзагроз, який підтвердив критичну необхідність переходу від реактивного до проактивного управління інцидентами, що стало основою для подальшого архітектурного проектування. Центральним елементом системи є логічна модель СД, спроєктована за схемою «Зірка». Ця модель включає таблицю фактів FactThreats, що містить ключові метрики (AvgResponseTime, ThreatCount, FailedAuthCount), та чотири відповідні таблиці вимірів (Час, Джерело, Тип та Рівень Загрози). Розробка інженерної архітектури (Розділ 3) завершилася обґрунтуванням програмного ETL-методу, реалізованого за допомогою оптимізованих SQL-скриптів, що забезпечило надійне та послідовне завантаження структурованих даних у DWH.

2. Практична реалізація та результати досліджень

Практична цінність системи була доведена через аналіз даних, завантажених у СД. Функціонал OLAP-аналізу забезпечив ефективний моніторинг ключових показників ефективності (KPI), що є необхідним для аудиту відповідно до стандарту ISO 27001. Зокрема, було встановлено, як AvgResponseTime (Середній час реагування) корелює з різними рівнями критичності загроз, дозволяючи відстежувати оперативність відділу ІБ.

Вирішальним є те, що Data Mining підтвердив можливість проактивної оборони. Кластерний аналіз (KMeans) успішно ідентифікував "Кластер 1", невелику групу IP-адрес, яка демонструє аномально високий показник

FailedAuthCount. Це дозволяє системі проактивно виявляти джерела атак типу Brute Force. Крім того, класифікаційний аналіз (використання алгоритму 1R) виявив, що атрибут "Hour" (година доби) є найбільш інформативним предиктором рівня загрози. Ця кореляція, що досягла високої точності в тестовій вибірці, підтверджує, що служба безпеки може пріоритезувати ресурси для моніторингу в неробочий час.

3. Наукова новизна, значущість та напрямки подальших досліджень

Наукова новизна роботи полягає у розробці та дослідженні комплексної аналітичної моделі для кібербезпеки, що об'єднує багатовимірне моделювання СД та прогностичну силу алгоритмів Data Mining. Практична значущість підтверджується створенням функціональної основи для прийняття рішень, здатної скоротити час ідентифікації рутинних загроз та автоматично виявляти приховані патерни атак.

Отримана система є значним кроком у напрямку побудови ефективної інфраструктури управління інформаційною безпекою компанії. В якості перспективних напрямків для подальшої роботи можна виділити інтеграцію розробленого СД із реальними SIEM-платформами(типу Splank) для валідації моделі, а також розширення Data Mining компонента шляхом впровадження більш складних алгоритмів машинного навчання, таких як нейронні мережі, для виявлення тонших аномалій. Також доцільною є розробка модуля, який автоматично генерує рекомендації для зміни політик безпеки на основі виявлених аналітичних патернів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Основи інформаційної безпеки: Тріада CIA (Confidentiality, Integrity, Availability). URL: https://www.comptia.org/content/articles/what-is-the-cia-triad (дата звернення: 14.09.2025).
2. DDoS Quick Guide. US Cybersecurity and Infrastructure Security Agency (CISA). URL: https://www.cisa.gov/resources-tools/resources/ddos-quick-guide (дата звернення: 14.09.2025).
3. Advanced Persistent Threats (APT). MITRE ATT&CK Framework. URL: https://attack.mitre.org/groups/ (дата звернення: 14.09.2025).
4. Insider Threat. US Cybersecurity and Infrastructure Security Agency (CISA). URL: https://www.cisa.gov/topics/security-threats-and-risk/insider-threat (дата звернення: 14.09.2025).
5. ISO/IEC 27001:2022. Information security, cybersecurity and privacy protection — Information security management systems — Requirements. URL: https://www.iso.org/standard/82875.html (дата звернення: 15.09.2025).
6. Закон України "Про захист інформації в інформаційно-комунікаційних системах". URL: https://zakon.rada.gov.ua/laws/show/80/94-%D0%B2%D1%80 (дата звернення: 15.09.2025).
7. User and Entity Behavior Analytics (UEBA). Gartner Glossary. URL: https://www.gartner.com/en/information-technology/glossary/user-and-entity-behavior-analytics-ueba (дата звернення: 15.09.2025).
8. What is SIEM? (Security Information and Event Management). IBM. URL: https://www.ibm.com/topics/siem (дата звернення: 15.09.2025).
9. Suricata. Open Information Security Foundation. URL: https://suricata.io/ (дата звернення: 15.09.2025).
10. Splunk Enterprise. URL: https://roi4cio.com/catalog/product/splunk-enterprise (дата звернення: 15.09.2025).

11. Splunk TA for Suricata. URL: https://splunkbase.splunk.com/app/2760 (дата звернення: 15.09.2025).
12. What is ETL (Extract, Transform, Load)? Microsoft Azure. URL: https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-etl (дата звернення: 16.09.2025).
13. Городецька В.В., Левківський В.В., Вакалюк Т.А. Що таке ETL і для чого це потрібно. Оптимізація промальовування вебзастосунку..., м. Житомир. С. 1.
14. Kimball, R., Ross, M. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling. 3rd ed. Wiley, 2013. 552 p.
15. Що таке OLAP. URL: https://data-life-ua.com/db/shcho-take-olap/ (дата звернення: 16.09.2025).
16. Голуб Б. Л. Data Mining - основні положення. Київ, 2021. 12 с. URL: https://elearn.nubip.edu.ua/pluginfile.php/102629/mod_resource/content/3/DATA%20MINING%20–%20початок.pdf (дата звернення: 17.09.2025).
17. Стеценко, І.В. Моделювання систем: навч. посіб. [Електронний ресурс, текст] / І.В. Стеценко ; М-во освіти і науки України, Черкас. держ. технол. ун-т. – Черкаси : ЧДТУ, 2010. – 399 с.
18. Основні підходи до моделювання інформаційно-вимірювальних систем. Житомирська Політехніка. URL: https://learn.ztu.edu.ua/pluginfile.php/114218/mod_resource/content/0/Лекція%206%20OMIBC.pdf (дата звернення: 19.09.2025).
19. Уніфікована мова моделювання (Unified Modeling Language - UML). Maxym Zosym. URL: https://www.maxzosim.com/unifikovana-mova-modeluvannia/ (дата звернення: 20.09.2025).
20. In-depth Knowledge of UML Use Case Diagram: with Tutorial. MindOnMap. URL: https://www.mindonmap.com/uk/blog/what-is-a-uml-use-case-diagram/ (дата звернення: 20.09.2025).

21. Діаграма послідовності (Sequence Diagrams). URL: https://www.maxzosim.com/sequence-diagrams/ (дата звернення: 21.09.2025).
22. Діаграми UML для моделювання процесів і архітектури проекту. Evergreen. URL: https://evergreens.com.ua/ua/articles/uml-diagrams.html (дата звернення: 21.09.2025).
23. What Is Microsoft Power BI? Microsoft Power BI Documentation. URL: https://learn.microsoft.com/en-us/power-bi/fundamentals/power-bi-overview (дата звернення: 22.09.2025).
24. Стандартні методи кластеризації даних. Факультет комп'ютерних наук та кібернетики. URL: https://csc.knu.ua/media/study/asp/mod_probl_inf_tech_sys_analysis_ivohin/lecture/lec2.pdf (дата звернення: 23.09.2025).
25. Learn-One-Rule Algorithm - GeeksforGeeks. GeeksforGeeks. URL: https://www.geeksforgeeks.org/machine-learning/learn-one-rule-algorithm/ (дата звернення: 25.09.2025). What Is Microsoft Power BI? Microsoft Power BI Documentation. URL: https://learn.microsoft.com/en-us/power-bi/fundamentals/power-bi-overview (дата звернення: 17.09.2025).
26. Naive Bayes Classifiers - GeeksforGeeks. GeeksforGeeks. URL: https://www.geeksforgeeks.org/machine-learning/naive-bayes-classifiers/ (дата звернення: 26.09.2025).
27. Tkinter — Python interface to Tcl/Tk. Python Documentation. URL: https://docs.python.org/3/library/tkinter.html (дата звернення: 28.09.2025).

ДОДАТКИ

ДОДАТОК А

Генерація вихідних даних (Staging Area):

```
TRUNCATE TABLE Raw_Log_Data;

DECLARE @i INT = 1;
DECLARE @TotalRecords INT = 500;
DECLARE @CurrentTime DATETIME2 = '2025-01-15 00:00:00';
DECLARE @AnomalyIP1 NVARCHAR(50) = '192.168.1.72';

WHILE @i <= @TotalRecords
BEGIN
    SET @CurrentTime = DATEADD(minute, @i * 5, '2025-01-15 00:00:00');
    DECLARE @Start_time BIGINT = DATEDIFF(SECOND, '1970-01-01', @CurrentTime);

    DECLARE @CurrentIP NVARCHAR(50);
    DECLARE @IsAnomaly INT = ABS(CHECKSUM(NEWID()) % 20);

    IF @IsAnomaly = 1
        SET @CurrentIP = @AnomalyIP1;
    ELSE
        SET @CurrentIP = '192.168.1.' + CAST(ABS(CHECKSUM(NEWID()) % 190 + 10) AS
NVARCHAR(10));

    DECLARE @CurrentLevel NVARCHAR(50);
    DECLARE @CurrentType NVARCHAR(100);
    DECLARE @FailedAuth INT;
    DECLARE @ResponseTime FLOAT;
    DECLARE @CurrentHour INT = DATEPART(HOUR, @CurrentTime);

    IF @CurrentHour IN (0, 1, 2) OR @IsAnomaly = 1
    BEGIN
        SET @CurrentLevel = 'High';
        SET @CurrentType = 'Brute Force';
        SET @FailedAuth = CASE WHEN @IsAnomaly = 1 THEN ABS(CHECKSUM(NEWID()) % 50 +
30) ELSE 0 END;
        SET @ResponseTime = (ABS(CHECKSUM(NEWID()) % 40) + 20) / 10.0;
    END
    ELSE IF @CurrentHour >= 8 AND @CurrentHour <= 18
    BEGIN
        SET @CurrentLevel = 'Medium';
        SET @CurrentType = 'Reconnaissance';
        SET @FailedAuth = ABS(CHECKSUM(NEWID()) % 5);
        SET @ResponseTime = (ABS(CHECKSUM(NEWID()) % 15) + 5) / 10.0;
    END
    ELSE
    BEGIN
        SET @CurrentLevel = 'Low';
        SET @CurrentType = 'Malware Scan';
        SET @FailedAuth = ABS(CHECKSUM(NEWID()) % 3);
        SET @ResponseTime = (ABS(CHECKSUM(NEWID()) % 10) + 1) / 10.0;
    END

    INSERT INTO Raw_Log_Data (
        Start_time, Source_IP, Threat_Level, Threat_Type_Name, Failed_Auth_Count,
        Threat_Raw_Count, Response_Time_Sec
    )
    VALUES (
        @Start_time,
        @CurrentIP,
        @CurrentLevel,
```

```
    @CurrentType,
    @FailedAuth,
    ABS(CHECKSUM(NEWID()) % 40 + 10),
    @ResponseTime
);

SET @i = @i + 1;
    END
```

ДОДАТОК Б

ETL та Послідовне Завантаження Вимірів (Dimension Load):

```
-- Заповнення DimensionTime
;WITH TimeConversion AS (
    SELECT DISTINCT
        CAST(FORMAT(DATEADD(SECOND, T1.[Start_time], '1970-01-01'), 'yyyyMMdd') AS
INT) AS DateKey,
        DATEPART(HOUR, DATEADD(SECOND, T1.[Start_time], '1970-01-01')) AS Hour
    FROM Raw_Log_Data T1
)
INSERT INTO DimensionTime (Date, Hour)
SELECT
    TC.DateKey,
    TC.Hour
FROM
    TimeConversion TC
WHERE
    NOT EXISTS (SELECT 1 FROM DimensionTime dt WHERE dt.Date = TC.DateKey AND dt.Hour
= TC.Hour);


-- Заповнення DimensionSource
INSERT INTO DimensionSource (IP_Address, SourceName, SourceType)
SELECT DISTINCT
    T1.Source_IP,
    'Client' AS SourceName,
    CASE
        WHEN T1.Source_IP LIKE '192.168.%' THEN 'Внутрішнє'
        WHEN T1.Source_IP LIKE '10.%' THEN 'Внутрішнє'
        ELSE 'Зовнішнє'
    END AS CalculatedSourceType
FROM
    Raw_Log_Data T1
WHERE
    NOT EXISTS (SELECT 1 FROM DimensionSource DS WHERE DS.IP_Address = T1.Source_IP);


-- Заповнення DimensionThreatLevel
INSERT INTO DimensionThreatLevel (ThreatLevelName, SeverityScore)
SELECT DISTINCT
    T1.Threat_Level AS CalculatedThreatLevel,
    CASE
        WHEN T1.Threat_Level = 'High' THEN 5
        WHEN T1.Threat_Level = 'Medium' THEN 3
        ELSE 1
    END AS CalculatedSeverity
FROM
    Raw_Log_Data T1
WHERE
    NOT EXISTS (SELECT 1 FROM DimensionThreatLevel DTL WHERE DTL.ThreatLevelName =
T1.Threat_Level);

-- Заповнення DimensionThreatType
INSERT INTO DimensionThreatType (ThreatTypeName)
SELECT DISTINCT
    T1.Threat_Type_Name
FROM
    Raw_Log_Data T1
WHERE
```

```
    NOT EXISTS (SELECT 1 FROM DimensionThreatType DTT WHERE DTT.ThreatTypeName =
T1.Threat_Type_Name);
```

ETL та Фіксація Факту (Fact Load):

```
-- Фінальне Завантаження FactThreats
INSERT INTO FactThreats (ID_Time, ID_Source, ID_ThreatType, ID_ThreatLevel,
ThreatCount, AvgResponseTime, FailedAuthCount)
SELECT
    DT.ID_Time,
    DS.ID_Source,
    DTT.ID_ThreatType,
    DL.ID_ThreatLevel,
    T1.Threat_Raw_Count AS ThreatCount,
    CAST(T1.Response_Time_Sec AS FLOAT) AS AvgResponseTime,
    T1.Failed_Auth_Count AS FailedAuthCount
FROM
    Raw_Log_Data T1
JOIN DimensionTime DT ON
    DT.Date = CAST(FORMAT(DATEADD(SECOND, T1.[Start_time], '1970-01-01'), 'yyyyMMdd')
AS INT) AND
    DT.Hour = DATEPART(HOUR, DATEADD(SECOND, T1.[Start_time], '1970-01-01'))
JOIN DimensionSource DS ON
    DS.IP_Address = T1.Source_IP
JOIN DimensionThreatLevel DL ON
    DL.ThreatLevelName = T1.Threat_Level
JOIN DimensionThreatType DTT ON
            DTT.ThreatTypeName = T1.Threat_Type_Name;
```