

Міністерство освіти і науки України  
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій  
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:

В.о. завідувача кафедри  
кібербезпеки  
та захисту інформації

Іван ПАРХОМЕНКО

«17» травня 2024 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА  
кваліфікаційної роботи

галузь знань *12 Інформаційні технології*

(шифр і назва галузі знань)

спеціальність *125 Кібербезпека*

(код і назва спеціальності)

освітній ступень *магістр*

освітньо-наукова програма *Кібербезпека*

(назва освітньої програми)

на тему: «Моделі виявлення аномалій кібербезпеки в інформаційних системах»

Виконавець: студент II курсу, групи КБм-21

Микола БЛИЗНЮК

(підпис)

(Ім'я, ПРІЗВИЩЕ)

	Ім'я, ПРІЗВИЩЕ	Підпис
Науковий керівник	Тетяна БАБЕНКО	
Нормоконтроль	Лариса МИРУТЕНКО	

Київ 2024

Міністерство освіти і науки України  
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій  
Кафедра кібербезпеки та захисту інформації

**ЗАТВЕРДЖЕНО:**

В.о. завідувача кафедри кібербезпеки  
та захисту інформації  
\_\_\_\_\_ Іван ПАРХОМЕНКО  
«17» листопада 2023 р.

**ЗАВДАННЯ**

на виконання кваліфікаційної роботи

спеціальності \_\_\_\_\_ 125 Кібербезпека  
(код і назва спеціальності)

освітній ступень \_\_\_\_\_ магістр

Здобувача(ки) \_\_\_\_\_ КБМ-21 \_\_\_\_\_ Близнюка Миколи Ігоровича  
(група) (прізвище ім'я по-батькові)

Тема кваліфікаційної роботи \_\_\_\_\_ Моделі виявлення аномалій кібербезпеки в інформаційних системах

**1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ**

Рішення засідання кафедри кібербезпеки та захисту інформації факультету інформаційних технологій протокол № 5 від 15.11.2023 р.

**2. МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ**

**Об'єкт досліджень** \_\_\_\_\_ Процес виявлення, ідентифікації та реагування на аномалії кібербезпеки в інформаційних системах

**Предмет досліджень** \_\_\_\_\_ Моделі виявлення аномалій кібербезпеки

**Мета** \_\_\_\_\_ Розробка моделі виявлення аномалій кібербезпеки в інформаційних системах

**Вихідні дані для проведення роботи** \_\_\_\_\_ Методи виявлення аномалій кібербезпеки в інформаційних системах

### 3. ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

**Наукова новизна** дослідження полягає в розробці та впровадженні нової моделі виявлення аномалій кібербезпеки, яка охоплює весь цикл виявлення аномалій штучним інтелектом

**Практична цінність** дослідження полягає в розробці методу виявлення аномалій для захисту інформаційних систем від кіберзагроз

### 4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Робота виконана у повному обсязі відповідно до теми.

### 5. ЕТАПИ ВИКОНАННЯ РОБОТИ

Найменування етапів робіт	Строки виконання робіт (початок-кінець)
Уточнення постановки задачі	17.11.2023 – 24.11.2023
Аналіз літературних джерел	25.11.2023 – 15.12.2023
Проведення аналізу предметної області виявлення аномалій	16.12.2023 – 29.12.2023
Дослідження існуючих методів для виявлення аномалій в інформаційних системах	30.12.2023 – 19.01.2024
Створення моделі для виявлення аномалій в інформаційних системах, заснованої на інтеграції SIEM та chatGPT	20.01.2024 – 16.02.2024
Розгортання архітектури для демонстрації запропонованого методу виявлення аномалій	17.02.2024 – 15.03.2024
Налаштування стеку ELK	16.03.2024 – 22.03.2024
Інтеграція SIEM з chatGPT та тестування працездатності моделі виявлення аномалій в ІС	23.03.2024 – 19.04.2024
Оформлення пояснювальної записки згідно методичних рекомендацій	20.04.2024 – 12.05.2024
Подача пакету документів на розгляд ЕК	13.05.2024 – 17.05.2024

## 6. РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

<b>Економічний ефект</b>	Мінімізація витрат через використання ресурсів з відкритим вихідним кодом
<b>Соціальний ефект</b>	Удосконалення процесу виявлення аномалій кібербезпеки, забезпечуючи більшу безпеку персональних та корпоративних даних від кіберзагроз, що сприяє зміцненню довіри до цифрових технологій у суспільстві

## 7. ДОДАТКОВІ ВИМОГИ

Завдання видав

\_\_\_\_\_ (підпис)

Тетяна БАБЕНКО

(Ім'я, ПРІЗВИЩЕ)

Завдання прийняв  
до виконання

\_\_\_\_\_ (підпис)

Микола БЛИЗНЮК

(Ім'я, ПРІЗВИЩЕ)

Дата видачі завдання: 17.11.2023 р.  
Термін подання кваліфікаційної роботи до ЕК 17.05.2024 р.

## РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Моделі виявлення аномалій кібербезпеки в інформаційних системах»: 91 сторінка, 48 рисунків та 4 таблиць. 61 літературне джерело.

Об'єкт дослідження – процес виявлення, ідентифікації та реагування на аномалії кібербезпеки в інформаційних системах.

Мета роботи – розробка моделі для ефективного виявлення аномалій кібербезпеки в інформаційних системах.

У роботі досліджено сучасні методи виявлення аномалій в інформаційних системах. Проведено аналіз відповідності типу аномалії до отримального методу її виявлення. Запропоновано модель виявлення аномалії, яка базується на інтеграції системи управління інформаційною безпекою та подіями інформаційної безпеки з штучним інтелектом.

Наукова новизна: дослідження полягає в розробці та впровадженні нової моделі виявлення аномалій кібербезпеки, що ґрунтується на передових методах МН та ШІ.

Актуальність теми: сучасні кіберзагрози характеризуються витонченістю та динамічною еволюцією, що зумовлює потребу в удосконалених моделях їх виявлення. Нещодавні дослідження свідчать про перспективність використання штучного інтелекту для розширення можливостей систем виявлення аномалій. Системи виявлення аномалій, засновані на машинному навчанні, можуть виявляти нові та модифіковані атаки, але вони часто вимагають великої кількості даних для навчання.

Модель виявлення аномалій, яка базується на інтеграції SIEM з chatGPT для аналізу логів та ідентифікації аномалій, може бути ефективним способом виявлення підозрілої поведінки, оскільки вона може бути адаптована до конкретних потреб організації. Ця модель забезпечує комплексне покриття різних аспектів кібербезпеки, від моніторингу та аналізу до сповіщення про інциденти, інтегруючи системи

управління подіями та інформаційною безпекою з можливостями AI для ідентифікації аномалій.

Ключові слова: аномалія, інформаційна система, SIEM, штучний інтелект, машинне навчання, модель виявлення, безпека.

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ**

ІБ	–	Інформаційна безпека
SIEM	–	Security information and event management
IDS	–	Intrusion Detection System
DLP	–	Data Leak Prevention
XDR	–	Extended detection and response
IPS	–	Intrusion Prevention System
SOC	–	Security Operations Center
IT	–	Information Technology
AI	–	Artificial Intelligence
ML	–	Machine learning
ІІ	–	Штучний інтелект
МН	–	Машинне навчання
GDPR	–	General Data Protection Regulation
SVM	–	Support Vector Machines
SOC	–	Security Operations Center

## ЗМІСТ

ВСТУП.....	10
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ АНОМАЛІЙ КІБЕРБЕЗПЕКИ.....	12
1.1 Поняття аномалії у контексті інформаційної безпеки.....	12
1.2 Класифікація аномалій .....	13
1.3 Передумови виявлення аномалій .....	17
1.4 Системи виявлення аномалій.....	18
1.5 Система управління інформаційною безпекою та подіями інформаційної безпеки .....	21
Висновки до розділу 1 .....	24
РОЗДІЛ 2 СУЧАСНІ ПІДХОДИ ДО ВИЯВЛЕННЯ АНОМАЛІЙ .....	26
2.1 Аналіз проблем виявлення аномалій кібербезпеки .....	26
2.2 Аналіз існуючих методів виявлення аномалій.....	28
2.2.1 Статичне виявлення аномалій.....	29
2.2.2 Виявлення аномалій на основі машинного навчання .....	30
2.2.3 Виявлення аномалій на основі аналізу даних.....	31
2.2.4 Нейромережевий метод.....	32
2.2.5 Генетичні алгоритми .....	33
2.2.6 Кластерний метод виявлення аномалій.....	33
2.2.7 Метод опорних векторів.....	35
2.2.8 Баєсовські мережі .....	36
2.2.9 Древа рішень .....	37
2.2.10 Модель на основі нелінійної динаміки.....	38
2.2.11 Гібридні системи .....	39
2.3 Відношення типів аномалій до методів їхнього виявлення.....	39
2.4 Опис запропонованої моделі виявлення аномалій .....	41
2.4.1 Методологія.....	42
2.4.2 Сфери застосування і переваги .....	43

	9
2.4.3 Виклики і обмеження .....	44
2.4.4 Концептуальна архітектура .....	45
Висновки до розділу 2 .....	47
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ЗАПРОПОНОВАНОЇ МОДЕЛІ .....	49
3.1 Топологія тестової інфраструктури для реалізації рішення .....	49
3.2 Побудова інфраструктури .....	52
3.3 Налаштування кластеру Elasticsearch .....	56
3.4 Налаштування балансиру навантаження Kemp .....	68
3.5 Налаштування віртуального фаєрволу pfsense.....	73
3.6 Створення програмного модуля для інтеграції Elasticsearch та ChatGPT .....	76
Висновки до розділу 3.....	82
ВИСНОВКИ.....	83
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	85
ДОДАТОК А.....	92
ДОДАТОК Б .....	93

## ВСТУП

В епоху цифрових технологій інформаційні системи (ІС) відіграють фундаментальну роль в інфраструктурі комерції, управління та комунікації на глобальному рівні. Ця критична залежність супроводжується зростанням кіберзагроз, роблячи аномалії кібербезпеки значною вразливістю. Аномалії кібербезпеки визначаються як відхилення від нормального функціонування ІС, що свідчать про потенційні інциденти або порушення безпеки [1]. Виявлення та обробка таких аномалій є першочерговим завданням для забезпечення захисту конфіденційних даних та підтримки цілісності систем.

Сучасні кіберзагрози характеризуються витонченістю та динамічною еволюцією, що зумовлює потребу в удосконалених моделях їх виявлення. Нещодавні дослідження свідчать про перспективність використання обчислювальних моделей та алгоритмів машинного навчання (МН) [58] для розширення можливостей систем виявлення аномалій [2]. Ці моделі, засновані на методах як контрольованого, так і неконтрольованого навчання, пропонують еволюційні підходи до виявлення тонких і складних патернів, що свідчать про загрози безпеці [3].

Метою цієї роботи є розробка моделі виявлення аномалій кібербезпеки в інформаційних системах. Ця модель буде ґрунтуватися на передових методах МН та ШІ для підвищення точності, ефективності та прогностичної здатності механізмів виявлення аномалій.

**Актуальність.** Необхідність у вдосконалених методах виявлення аномалій кібербезпеки зумовлена постійною еволюцією кіберзагроз. Невиявлена аномалія може мати значні наслідки, включаючи фінансові втрати, шкоду для репутації та порушення конфіденційності, цілісності або доступності даних [4].

З огляду на вищевказані мету та актуальність даної тематики, поставлено наступні задачі:

1. Аналіз наукових публікацій, досліджень та літератури, пов'язаних з методами виявлення аномалій в сфері кібербезпеки.

2. Розробка концепції моделі виявлення аномалій, що охоплює весь цикл даних: від генерації на хості до виявлення аномалій штучним інтелектом.
3. Розгортання запропонованої моделі в тестових умовах.
4. Перевірка працездатності розробленої моделі на реальних наборах даних.
5. Підтвердження ефективності запропонованої моделі шляхом узагальнення результатів дослідження.

Методи дослідження, що використовуються в роботі: аналіз предметної області з подальшим моделюванням та емпіричним дослідженням розробленої моделі.

**Наукова новизна** дослідження полягає в розробці та впровадженні нової моделі виявлення аномалій кібербезпеки, яка охоплює весь цикл виявлення аномалій штучним інтелектом.

**Практична цінність** дослідження полягає в розробці методу виявлення аномалій для захисту інформаційних систем від кіберзагроз.

Очікується, що розроблена модель виявлення аномалій кібербезпеки зможе більш точно ідентифікувати аномалії, мінімізуючи ймовірність помилкових спрацювань; буде оптимізована для швидкого та економного виявлення аномалій, що дозволить мінімізувати час простою та ресурси, необхідні для забезпечення кібербезпеки.

## РОЗДІЛ 1

### АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ АНОМАЛІЙ КІБЕРБЕЗПЕКИ

#### 1.1 Поняття аномалії у контексті інформаційної безпеки

Аномалії кібербезпеки - це відхилення від нормальної поведінки інформаційних систем, які свідчать про потенційні проблеми або загрози інформаційній безпеці. Ці аномалії можуть проявлятися у вигляді незвичної структури мережевого трафіку, неочікуваної поведінки системи, спроб несанкціонованого доступу або незрозумілих змін у даних. Основною метою виявлення аномалій в ІС є своєчасне ідентифікування цих порушень для запобігання, пом'якшення або реагування на потенційні інциденти безпеки [5]. На рисунку 1.1 зображено діаграму, яка описує аномалію.

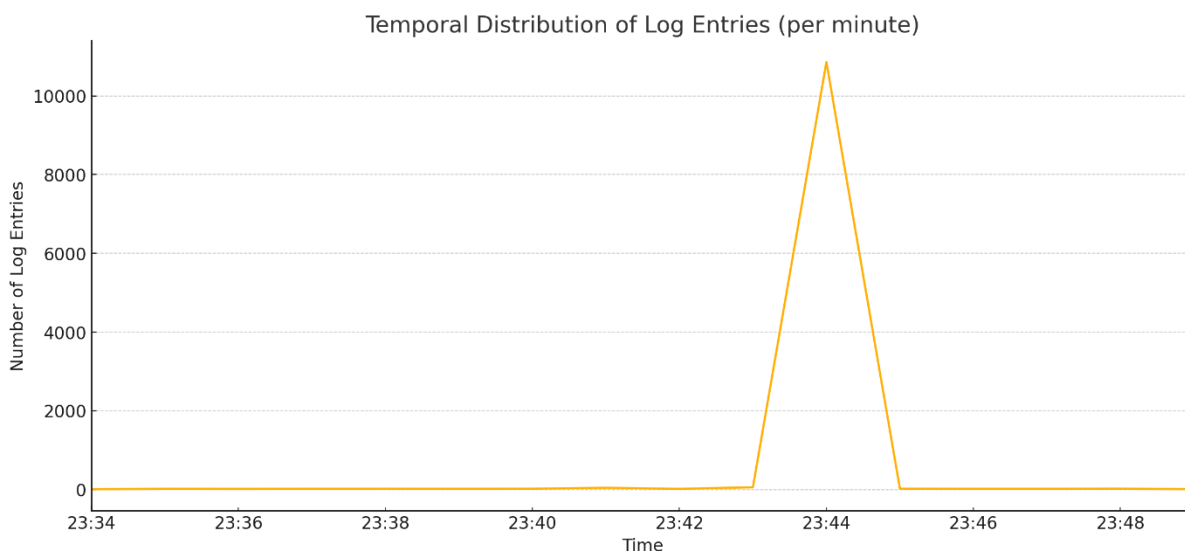


Рисунок 0.1 – Загальна діаграма, яка описує аномалії

Концепція аномалій кібербезпеки ґрунтується на припущенні, що інформаційні системи демонструють передбачувану, нормальну поведінку, засновану на встановлених моделях діяльності користувачів, мережевого трафіку та продуктивності системи. Аномалії виникають, коли спостережувана поведінка значно відхиляється від цих шаблонів, що потенційно вказує на загрозу безпеці, таку як

кібератака, зараження шкідливим програмним забезпеченням або інсайдерська загроза. Характер цих відхилень може бути найрізноманітнішим – від ледь помітних ознак складної спроби вторгнення до явних фактів компрометації системи [6].

Аномалії кібербезпеки становлять значну загрозу для конфіденційності, цілісності та доступності інформаційних систем. Вони можуть слугувати передвісниками витоку, крадіжки персональних даних, фінансового шахрайства та інших кіберзлочинів. Окрім безпосередніх ризиків для безпеки, невиявлені аномалії можуть призвести до довгострокових вразливостей в інформаційних системах, дозволяючи зловмисникам створювати постійні загрози або викрадати конфіденційну інформацію з плином часу. Наслідки таких інцидентів виходять за рамки втрати даних, включаючи юридичну відповідальність, фінансові штрафи та репутаційні збитки для постраждалих організацій [7].

## **1.2 Класифікація аномалій**

У сфері інформаційної безпеки можна ідентифікувати декілька типів аномалій:

1. Аномалії в поведінці користувачів: ці аномалії виникають, коли користувачі поведуться незвично для свого звичайного способу діяльності, або коли вони проводять дії, які не мають логічного пояснення [53]. Наприклад, звичайний користувач намагається підвищити привілеї.

2. Аномалії в мережевому трафіку: ці аномалії пов'язані з підозрілою активністю в мережі, такою як надмірний трафік або фігурування незвичних адрес. Наприклад, велика кількість запитів з одного джерела.

3. Аномалії в системних журналах: ці аномалії пов'язані зі змінами в системних журналах. Наприклад, видалення подій або модифікація журналів, що може свідчити про те, що зловмисник «замітає сліди».

4. Аномалії в даних: виникає, коли інформація, збережена у системі, включає помилкові або неочікувані вхідні дані. Це може бути, наприклад, наявність дубльованих записів, використання некоректних типів даних або форматів. Такі аномалії можуть викликати збої у програмному забезпеченні, яке обробляє ці дані, а

також можуть негативно вплинути на конфіденційність, цілісність та доступність інформації.

5. Аномалії в ресурсах: ці аномалії з'являються, коли система використовує надмірну кількість ресурсів (таких як CPU, RAM, диск тощо) або використовує їх неефективно. Наприклад, це може стосуватися процесів, які використовують багато ресурсів CPU, програм, що споживають більше місця на диску, ніж зазвичай, або запитів до баз даних, що потребують значної кількості пам'яті.

6. Аномалії в програмному забезпеченні: це нестандартні події або стани системи, що сповіщають про проблеми у програмному забезпеченні. Ці аномалії можуть включати помилки в коді або вразливості, що дозволяють зламати систему або призводять до втрати даних. Наприклад, така аномалія може проявлятися як помилка у коді програми, що спричиняє її нестабільну роботу або збій.

7. Аномалії в архітектурі системи: це відхилення представляють собою недотримання стандартних концепцій архітектури, які можуть виникати через помилки у фазах проектування, розробки або імплементації системи, а також внаслідок її змін. Виявлення таких аномалій часто є доволі складним процесом, оскільки вони можуть проявлятися лише у певних ситуаціях або за специфічних дій користувачів. Наприклад, дефекти в архітектурі можуть спричинити витік конфіденційної інформації через незахищені мережеві канали або викликати збої критично важливих систем у моменти підвищеного навантаження.

8. Аномалії в ідентифікації та автентифікації: ці аномалії стосуються забезпечення безпеки та контролю доступу до систем чи ресурсів. Випадки, коли користувач не може бути належно ідентифікований, наприклад через введення неправильного пароля або спроби несанкціонованої авторизація в обліковий запис, відносяться до аномалій в ідентифікації. З іншого боку, аномалії в автентифікації виникають, коли, незважаючи на коректну ідентифікацію і автентифікацію користувача, йому не вдається отримати доступ до певного ресурсу через недостатні права, як, наприклад, при спробі доступу до файлу або каталогу, до якого немає прав.

9. Аномалії в захисті від вразливостей: такі аномалії з'являються, коли існуюча система не в змозі адекватно захиститися від потенційних атак, які

використовують уже відомі вразливості. Як приклад, можна навести відхилення від встановлених кращих практик захисту у різноманітних системах, що може призвести до нападів через програмні недоліки; механізми захисту, які можна обійти; а також помилки користувачів під час використання систем.

10. Хмарні аномалії: ці аномалії виникають, коли в середовищах хмарних обчислень виникає несанкціонований доступ або коли з'являється незвична активність. Одним із прикладів може бути ситуація, де користувач, який зазвичай не має прав на доступ до певних хмарних ресурсів, раптово починає їх використовувати, або коли активуються сервіси, що раніше не застосовувались. Такі інциденти часто є ознаками проблем з безпекою.

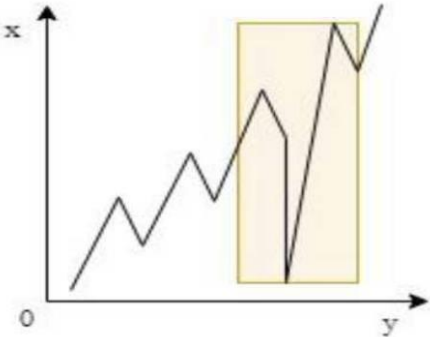
11. Аномалії, пов'язані з часом: такі аномалії виникають, коли активність відбувається у нехарактерний час. Наприклад, коли користувач логінується в систему в нетиповий для нього час або використовує системні ресурси під час пікового навантаження, це може бути ознакою нелегітимного доступу або атаки на систему.

Додатково, аномалії класифікують через призму мережевої безпеки.

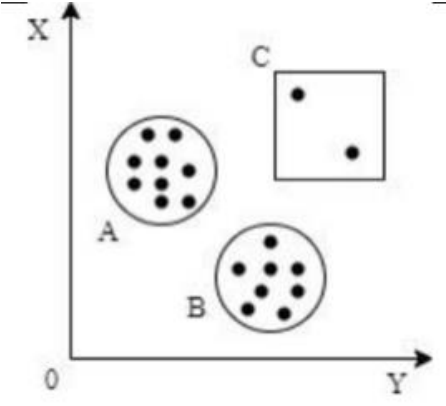
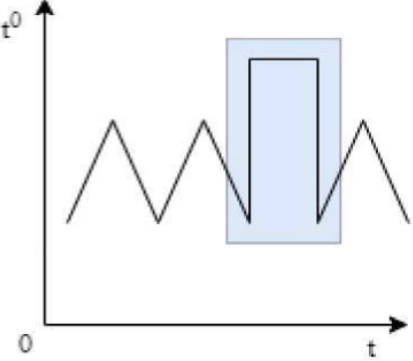
У таблиці 1.1 наведений опис даної класифікації.

Таблиця 1.1

Класифікація аномалій

Назва	Зображення	Пояснення
Контекстні аномалії		<p>Випадки, коли спостереження вважається аномальним у певному контексті, але не обов'язково у іншому. Наприклад, підвищене використання мережі може бути типовим під час робочого дня, але становитиме аномалію,</p>

## Класифікація аномалій

		якщо це відбудеться в нічний час.
Точкові аномалії		<p>Окремий випадок в даних значно відрізняється від більшості інших. Наприклад, спроба входу в систему з географічної локації, яка раніше не використовувалася цим користувачем. Ці аномалії часто легко ідентифікувати через очевидні відхилення від норми.</p>
Коллективні аномалії		<p>Ці аномалії стосуються групи подій, які разом відхиляються від очікуваної поведінки, хоча окремі події в групі можуть не виглядати підозріло. Наприклад, послідовність вхідних спроб з правильними паролями, але з різних несподіваних місць у короткий проміжок часу може бути індикатором атаки типу "credential stuffing".</p>

Аномалії в кібербезпеці можуть виникати в різних формах, кожна з яких вимагає специфічного підходу до виявлення та аналізу.

### 1.3 Передумови виявлення аномалій

Центральна ідея виявлення аномалій полягає в тому, що вторгнення є підмножиною аномальної діяльності [15].

Якщо розглядати зловмисника, що вторгається в систему, який не знає шаблонів діяльності легітимного користувача, існує висока ймовірність того, що діяльність зловмисника буде виявлена як аномальна. У ідеальному випадку, набір аномальних дій буде таким же, як набір вторгнення.

У такому разі, маркування всіх аномальних дій як вторгнення призводить до відсутності хибнопозитивних і хибнонегативних результатів.

Однак, діяльність вторгнення не завжди збігається з аномальною діяльністю.

Кумар і Стаффорд [10] вказали, що існує чотири можливості, кожна з яких має ненульову ймовірність:

- Вторгнення, але не аномальна. Система виявлення вторгнень не виявляє цей тип діяльності, оскільки вона не є аномальною. Ці випадки називаються хибнонегативними, тому що система виявлення вторгнень помилково повідомляє про відсутність вторгнень.

- Не вторгнення, але аномальна. Іншими словами, діяльність не є вторгненням, але через те, що вона є аномальною, система виявлення вторгнень повідомляє про неї як про вторгнення. Ці випадки називаються хибнопозитивними, тому що система виявлення вторгнень помилково повідомляє про вторгнення.

- Не вторгнення і не аномальна: це справжні негативні результати; діяльність не є вторгненням і не повідомляється як така.

- Вторгнення і аномальна: це справжні позитивні результати; діяльність є вторгненням і повідомляється як така.

## 1.4 Системи виявлення аномалій

Системи виявлення аномалій вказують на підозрілу активність, яка виходить за межі попередньо встановлених патернів поведінки.

Такого роду рішення захищають інформаційну систему в режимі реального часу від випадків, які можуть призвести до значних фінансових втрат, витоку даних та інших шкідливих подій [19].

Для того, щоб система виявлення аномалій виявляла відхилення від нормального стану, спочатку потрібно визначити, що така система повинна вважати за нормальний стан.

Це передбачає встановлення ключових параметрів, які можуть бути точно виміряні протягом тривалого часу, та встановлення порогових значень, за яких поведінка вважається нормальною або аномальною [20].

Для оцінки наявності аномалій у відповідних випадках використовуються дані про трафік, який може бути як мережевим, так і немережевим, в залежності від специфіки системи та мети моніторингу.

У мережах трафік складається з пакетів, як правило, фрагментованих на рівні IP.

Ці дані збираються протягом певного часу для подальшої нормалізації та визначення характеристик, на основі яких формується нормальна поведінка.

Ця поведінка порівнюється з попередньо визначеними параметрами нормальної діяльності системи [21]. У разі виявлення істотних відмінностей у параметрах, трафік вважається аномальним.

В іншому випадку відбувається корекція шаблону нормальної поведінки, і діяльність переходить в режим звичайної мережевої діяльності.

Алгоритм перевірки даних на відповідність шаблону часто є складним і вимагає точності, оскільки результати можуть впливати на майбутнє як інформаційної системи, так і організації в цілому [22].

Опис роботи алгоритму виявлення аномалій наведений на рисунку 1.2.

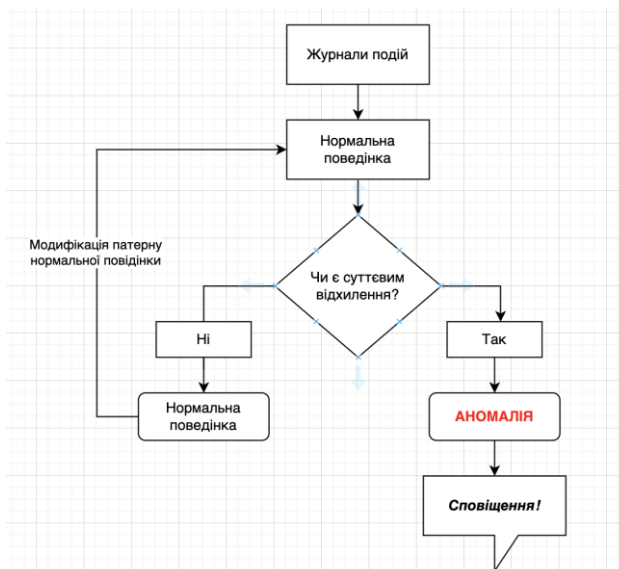


Рисунок 0.2 – Алгоритм виявлення аномалії

Отже, все, що відхиляється від норми, може бути небезпечним. У сфері інформаційної і кібербезпеки причиною аномалій можуть бути хакерські атаки.

За допомогою спеціальних рішень виявлення аномалій є можливість визначити слабкі місця у мережі або успішні кібератаки. Коли вони сповіщають про потенційну аномалію, фахівці з інформаційної безпеки повинні вживати контрзаходів [23].

Крім того, системи виявлення аномалій відіграють ключову роль у виконанні вимог різних стандартів і законодавчих актів, що регулюють захист інформаційної безпеки. У багатьох юрисдикціях, закон вимагає від організацій належного захисту своїх інформаційних систем від можливих загроз. Невиконання цих вимог може призвести до значних штрафів та судових позовів, особливо у випадках, коли недбалість у захисті даних призводить до серйозних порушень безпеки або витоків інформації [24].

Застосування систем виявлення аномалій допомагає забезпечити відповідність нормативним вимогам, наприклад, стандартам ISO/IEC 27001, які визначають вимоги до систем управління інформаційною безпекою, або GDPR (Загальний регламент захисту даних), який встановлює принципи захисту даних в ЄС. Такі системи дозволяють не тільки реагувати на потенційні загрози в реальному часі, але й демонструють дотримання зобов'язань щодо захисту даних і інформаційної безпеки,

що є важливим не тільки для уникнення штрафів, але й для збереження довіри клієнтів та репутації компанії.

За останні роки в сфері інформаційної безпеки утвердились різні концепції і системи виявлення аномалій:

- IDS: Системи виявлення вторгнень [50,52];
- SIEM: Система управління інформаційною безпекою та подіями інформаційної безпеки;
- XDR: Розширена система виявлення та реагування;
- DLP : Системи запобігання витоку даних;

Аудит даних систем також є важливою складовою вчасного та ефективного виявлення аномалій.

Безпечна конфігурація застосунків і операційних систем, так званий «харденінг», гарантує, що існує менше поверхонь для атаки [25]. Тому що, простіше кажучи, там, де немає вразливості, не може бути й зловживання. Відповідно, регулярний аудит системи також гарантує зменшення і помилкових спрацьовувань. Таким чином, команда SOC відчуває велике полегшення. Операційний центр безпеки (SOC) — це централізована функція в організації, яка використовує людей, процеси та технології для постійного моніторингу та покращення стану безпеки організації, одночасно запобігаючи, виявляючи, аналізуючи та реагуючи на інциденти кібербезпеки [26].

Алгоритми, засновані на машинному навчанні (ML) або інших формах штучного інтелекту (AI), можуть значно підвищити ефективність систем виявлення аномалій. Використання цих технологій дозволяє системам не тільки швидше ідентифікувати зміни в поведінці або даних, але й більш точно аналізувати та описувати ці зміни. Завдяки аналітиці, яка адаптується до нових інформаційних патернів, машинне навчання допомагає зменшити кількість помилкових спрацювань [27].

Це важливо, оскільки помилкові спрацювання можуть відволікати аналітичні ресурси від справжніх загроз, вимагаючи часу та зусиль на їх перевірку. Алгоритми ML здатні вчитися з історичних даних, визначати звичайні та незвичайні моделі

поведінки, що покращує точність виявлення істинно значущих аномалій [49]. Отже, використання машинного навчання та AI у системах виявлення аномалій не лише підвищує їхню швидкість роботи та точність, але й забезпечує кращий захист від потенційних кіберзагроз.

## **1.5 Система управління інформаційною безпекою та подіями інформаційної безпеки**

Система управління інформаційною безпекою та подіями інформаційної безпеки (SIEM) є критичним елементом в архітектурі кібербезпеки. В умовах непередбачуваного та постійно змінюваного кіберпростору, компанії зіштовхуються з безперервними викликами у захисті своїх інформаційних систем та мереж від зловмисних дій і складних кібератак. Хоча системи виявлення аномалій мають вирішальне значення для виявлення підозрілих дій і попередження про них, вони часто стикаються з обмеженнями у наданні достатнього контексту та загальної картини, необхідних для ефективного виявлення та реагування на загрози.

Тому, технології SIEM-систем у зміцненні інформаційної та кібербезпеки є надзвичайно важливими для підвищення прозорості операцій, прискоренні процесу ідентифікації загроз і вдосконаленні процедур реагування на інциденти. Сучасний кіберпростір ставить перед організаціями складні виклики, а кібератаки стають дедалі більш витонченими та важкими для виявлення [28]. Традиційні методи забезпечення безпеки часто обмежені за обсягом і не забезпечують повного огляду, що ускладнює адекватне реагування на еволюціонуючі загрози. Завдяки SIEM, що пропонує централізовану платформу для інтегрованого збору, аналізу і кореляції журналів подій, організації можуть ефективніше протистояти сучасним кіберзагрозам.

Технології SIEM надають широкий спектр можливостей:

- Централізований збір та аналіз даних з багатьох джерел: Системи SIEM агрегують інформацію з різних джерел(програм, кінцевих хостів, мережі) створюючи комплексний огляд стану захищеності.

- Виявлення загроз через кореляцію подій: З попередньо налаштованими правилами сповіщення на конкретні події в системі, SIEM можуть ідентифікувати складні зразки і аномалії, що дозволяє виявляти потенційні загрози на початкових етапах.

- Пріоритизація сповіщень та оптимізація реагування на інциденти: SIEM встановлюють пріоритети оповіщень, виходячи з їх важливості та контексту, дозволяючи командам з кібербезпеки зосередитись на найбільш критичних загрозах і ефективніше реагувати на інциденти.

- Забезпечення звітності та інформаційних дашбордів: Вбудовані інструменти для звітування та інформаційні панелі полегшують відповідність нормативним вимогам та надають важливу інформацію про актуальний стан безпеки і можливі вразливості.

Інтеграція систем виявлення аномалій з рішеннями SIEM забезпечує необхідний контекст для більш ефективного виявлення загроз, допомагаючи розрізнити справжні загрози від хибних спрацьовувань:

- Підвищення точності виявлення: За допомогою аналізу оповіщень від систем виявлення аномалій разом з іншою інформацією про подію і стан системи, SIEM дозволяє відсіювати хибні тривоги та концентруватися на реальних загрозах, тим самим зменшуючи навантаження на аналітичні команди.

- Забезпечення комплексного контексту: SIEM надають широке уявлення про безпекові події, що дозволяє командам безпеки глибше розуміти контекст атак та формувати більш ефективні стратегії реагування на інциденти.

Рішення SIEM займають все більше місця в сучасній кібербезпеці. Забезпечуючи покращену видимість, прискорюючи процеси виявлення загроз та оптимізуючи реакції на інциденти, SIEM системи допомагають організаціям ефективно захиститись від розвинених кіберзагроз [29]. Існує багато різних SIEM-рішень, кожне з яких пропонує унікальні функції та можливості. Серед них:

- Splunk: Це рішення відоме своєю масштабованістю та потужними аналітичними можливостями, яке хоча й може бути складним у впровадженні та

дорогим для великих розгортань, забезпечує комплексні можливості виявлення загроз та реагування на інциденти.

- Elastic SIEM: Це система з відкритим кодом, який має потужні можливості пошуку та аналітики, є економічно ефективним і легко інтегрується з іншими інструментами безпеки, дозволяючи налаштування та розширення, що робить його популярним вибором серед організацій.
- LogRhythm: Ця платформа пропонує централізоване управління, попередньо створені інформаційні дашборди для відповідності та автоматизовані робочі процеси, хоча й може мати менші гнучкі налаштування та вимагати більших витрат при масштабних розгортаннях.
- Sumo Logic: Ця хмарна SIEM пропонує можливості аналітики та візуалізації в режимі реального часу, що робить її ідеальною для організацій, що шукають масштабоване та гнучке рішення.

Хоча кожна система управління інформацією про події безпеки має свої унікальні переваги та обмеження, Elastic SIEM вирізняється своєю ефективністю, особливо в контексті систем виявлення вторгнень. Її відкритий код та масштабованість забезпечують значні переваги: доступність та економічна вигода відкритого вихідного коду, а також можливість адаптуватися до зростаючих обсягів даних і змін у вимогах кібербезпеки.

Elastic SIEM має потужні засоби для пошуку та аналітики, виходячи за рамки можливостей інших продуктів у цьому класі, надаючи фахівцям з безпеки інтуїтивно зрозумілі інтерфейси і можливість швидкого виявлення та розслідування підозрілої активності за допомогою складних запитів.

Легка інтеграція з різноманітними системами виявлення вторгнень та іншими інструментами безпеки сприяє консолідації даних і надає єдине вікно перегляду безпекових подій. Еластичність Elastic дозволяють організаціям кастомізувати його під свої специфічні потреби, включаючи розробку індивідуальних інформаційних дашбордів, звітів та правил виявлення загроз.

У межах цієї кваліфікаційної роботи запропоновано новий метод виявлення аномалій, заснований на інтеграції chatGPT та стеку ELK за допомогою скрипту,

написаного мовою програмування Python [56], які виявляють підозрілу поведінку в системних журналах Elasticsearch. Часто, системи виявлення аномалій засновані на правилах, які виявляють відомі атаки. Однак вони не можуть виявляти нові або модифіковані атаки. Системи виявлення аномалій, засновані на машинному навчанні, можуть виявляти нові та модифіковані атаки, але вони часто вимагають великої кількості даних для навчання. Інтеграція з chatGPT, яка виявляє аномалії в журналах подій, може бути ефективним способом виявлення аномальної поведінки, оскільки вона може бути адаптована до конкретних потреб організації. Даний метод має ряд переваг у порівнянні з традиційними підходами:

- Може виявляти як відомі, так і нові атаки.
- Може бути адаптований до конкретних потреб організації.
- Може бути реалізований з використанням широкодоступних інструментів.

Для дослідження ефективності даного методу планується провести експерименти на реальних даних з журналів безпеки.

## **Висновки до розділу 1**

У першому розділі детально аналізовано предметну область аномалій кібербезпеки, вводячи основні концепції та класифікації, що лежать в основі виявлення та управління цими аномаліями.

Аномалії у кібербезпеці розглядаються як значні відхилення від нормальної поведінки систем, що можуть вказувати на потенційні загрози або вразливості. Визначено декілька типів аномалій: аномалії в поведінці користувачів, аномалії в мережевому трафіку, аномалії в системних журналах, аномалії в даних, аномалії в ресурсах, аномалії в програмному забезпеченні, аномалії в архітектурі системи, аномалії в ідентифікації та автентифікації, аномалії в захисті від вразливостей, хмарні аномалії, аномалії, пов'язані з часом, кожна з яких має свої особливості та способи виявлення.

Також було досліджено різні системи моніторингу та виявлення аномалій такі як IDS, SIEM, XDR, та DLP з особливим фокусом на інтеграцію машинного навчання для підвищення точності та ефективності.

Безумовно, виявлення аномалій є критично важливим для попередження інцидентів кібербезпеки, дозволяючи оперативно реагувати на загрози, перш ніж вони спричинять значну шкоду інформаційним системам. Аналіз аномалій у кібербезпеці має стратегічне значення, оскільки забезпечує глибше розуміння потенційних загроз та способів їх виявлення та нейтралізації. Застосування новітніх технологій у сфері машинного навчання та штучного інтелекту в системах виявлення аномалій може значно покращити захист організацій від кібератак.

В розділі описується запропонований метод виявлення аномалій, який використовує інтеграцію chatGPT з стеком ELK через Python-скрипт для моніторингу системних журналів Elasticsearch. Традиційні системи виявлення аномалій часто обмежуються правилами, які ідентифікують лише відомі типи атак, проте не ефективні проти нових або модифікованих загроз. Натомість, запропонований метод, базуючись на штучному інтелекті, здатен виявляти і нові, і модифіковані атаки, підвищуючи адаптивність та ефективність систем безпеки. Це стає можливим завдяки гнучкості моделей штучного інтелекту, які можуть бути налаштовані під специфічні потреби організації, забезпечуючи більш точне та оперативне виявлення аномалій.

## РОЗДІЛ 2

### СУЧАСНІ ПІДХОДИ ДО ВИЯВЛЕННЯ АНОМАЛІЙ

#### 2.1 Аналіз проблем виявлення аномалій кібербезпеки

Своєчасне виявлення аномалій кібербезпеки має вирішальне значення для мінімізації впливу загроз безпеці. Рання ідентифікація дозволяє організаціям швидко ізолювати уражені системи, усунути вразливості та вжити контрзаходів для запобігання поточним атакам. Ефективне виявлення аномалій також відіграє вирішальну роль у дотриманні нормативних вимог, оскільки більшість стандартів і законів вимагають негайного виявлення та повідомлення про інциденти безпеки [30]. Передові методи виявлення, що включають машинне навчання і статистичний аналіз, все частіше застосовуються для підвищення точності і швидкості виявлення аномалій, зменшення кількості помилкових спрацьовувань і забезпечення більш цілеспрямованого реагування на реальні загрози [8].

Незважаючи на розвиток технологій та методологій, виявлення аномалій кібербезпеки в інформаційних системах може мати деякі виклики:

Складність сучасних мереж. Сучасні інформаційні системи характеризуються надзвичайною складністю, що включає в себе безліч пристроїв, протоколів і застосунків. Ця складність ускладнює виявлення аномалій, оскільки збільшує неоднорідність і обсяг даних, які необхідно аналізувати на предмет потенційних порушень безпеки [9]. Більше того, динамічна природа цих систем, з постійними оновленнями та змінами, додає ще один рівень складності до виявлення аномалій.

Обсяг і різноманітність даних. Великий обсяг і різноманітність даних в інформаційних системах створюють значні проблеми для виявлення аномалій. Традиційні методи виявлення аномалій намагаються ефективно обробляти та аналізувати великі масиви даних, що призводить до потенційних затримок або відхилень у виявленні порушень безпеки [10]. Крім того, різноманітна природа даних,

починаючи від структурованих і закінчуючи неструктурованими форматами, вимагає складних методів для ефективного аналізу.

Еволюція кіберзагроз. Кіберзагрози постійно розвиваються, стають все більш витонченими та складнішими для виявлення. Зловмисники постійно розробляють нові методи обходу заходів безпеки, що робить все більш складним для систем виявлення аномалій своєчасне виявлення та пом'якшення цих загроз [11]. Ця постійна еволюція вимагає безперервного оновлення та адаптації алгоритмів виявлення аномалій, щоб йти в ногу з новими загрозами.

Хибнопозитивні та хибнонегативні спрацювання. Однією з постійних проблем виявлення аномалій є баланс між False-positive та False-negative спрацюваннями. Висока чутливість систем виявлення може призвести до великої кількості хибних спрацювань, викликаючи непотрібні оповіщення і потенційно перевантажуючи команди, які відповідальні за безпеку [12]. І навпаки, зниження чутливості для зменшення кількості хибних спрацювань може призвести до збільшення кількості хибнонегативних спрацювань, коли реальні загрози не будуть виявлені. Досягнення правильного балансу є надзвичайно важливим, але водночас складним завданням.

Ресурсні обмеження. Ефективне виявлення аномалій вимагає значних обчислювальних ресурсів для обробки та аналізу даних. Однак не всі організації мають у своєму розпорядженні необхідні ресурси, що обмежує їхню здатність впроваджувати складні системи виявлення аномалій [13]. Це ресурсне обмеження стає ще більш вираженим в контексті малих і середніх підприємств.

Відсутність комплексних рішень. Незважаючи на те, що для виявлення аномалій існують численні інструменти та методи, бракує комплексних рішень, які могли б вирішити всі аспекти проблеми. Багато систем фокусуються на конкретних типах аномалій або загроз, залишаючи прогалини в загальній системі безпеки інформаційних систем [14]. Інтеграція різних інструментів для заповнення цих прогалин часто призводить до складності та проблем сумісності.

Виявлення аномалій кібербезпеки в інформаційних системах пов'язане з проблемами, починаючи від складності сучасних мереж і закінчуючи еволюцією

кіберзагроз. Обсяг і різноманітність даних, помилкові спрацьовування і негативні результати, обмеженість ресурсів і відсутність комплексних рішень - все це ускладнює ефективне виявлення аномалій. Вирішення цих проблем вимагає багатогранного підходу, який включає розробку більш досконалих алгоритмів виявлення, постійне оновлення систем безпеки для протидії новим загрозам, а також інтеграцію різних інструментів і методів в комплексну стратегію безпеки [31]. Лише завдяки такому підходу виявлення аномалій кібербезпеки може стати більш ефективним та результативним.

## **2.2 Аналіз існуючих методів виявлення аномалій**

Вчасне виявлення аномалій має вирішальне значення для забезпечення безпеки, оскільки, кожна інформаційна система є вразливою перед різного роду загрозами. Це можуть бути витoki даних, соціальний інжиніринг та інші форми кібератак. Навіть після впровадження комплексних заходів забезпечення безпеки, абсолютна гарантія захищеності системи неможлива, оскільки сфера інформаційної безпеки постійно змінюється і еволюціонує [32]. Тому необхідність постійного вдосконалення заходів забезпечення безпеки стає критичною для збереження надійності інформаційних систем.

Для ефективного виявлення аномалій в мережах інформаційних систем ключовим є логування і моніторинг кінцевих точок і трафіку між ними, оскільки це дозволяє виявляти незвичайні або аномальні патерни, що можуть свідчити про потенційні загрози безпеці інформації. Виявлення аномалій базується на припущенні, що поведінка хостів в системі відрізнятиметься від звичайної. Підхід до виявлення аномалій зазвичай складається з двох етапів: етапу навчання та етапу тестування [33]. У першому визначається нормальний профіль трафіку; в останньому вивчений профіль застосовується до нових даних. Існує низка архітектур і методів, запропонованих для виявлення аномалій.

### 2.2.1 Статичне виявлення аномалій

У статистичних методах виявлення аномалій система спостерігає за діяльністю кінцевих точок і створює профілі для представлення їх поведінки. Профіль зазвичай включає такі показники, як показник інтенсивності діяльності, показник розподілу записів аудиту, категоріальний показник (розподіл діяльності за категоріями) і порядковий показник (наприклад, використання CPU) [33]. Як правило, для кожного хоста підтримуються два профілі: поточний і збережений. У міру обробки системних/мережевих подій (наприклад, записів журналу аудиту, вхідних пакетів тощо) система виявлення вторгнень оновлює поточний профіль і періодично обчислює оцінку аномалії для конкретної події, порівнюючи поточний профіль із збереженим профілем за допомогою функції аномалії всіх показників у профілі. Якщо оцінка аномалії перевищує певний поріг, система виявлення вторгнень генерує сповіщення.

Статистичні підходи до виявлення аномалій мають ряд переваг. По-перше, ці системи, як і більшість систем виявлення аномалій, не вимагають попереднього знання самих атак. У результаті такі системи мають можливість виявляти «нульовий день» або найновіші атаки. Крім того, статистичні підходи можуть забезпечити точне сповіщення про зловмисну діяльність, яка зазвичай відбувається протягом тривалих періодів часу та є хорошими показниками атак типу «відмова в обслуговуванні» (DoS). Дуже поширеним прикладом такої діяльності є сканування портів. Як правило, розподіл сканувань портів є дуже аномальним порівняно зі звичайним розподілом трафіку [34]. Особливо, коли пакет має незвичайні функції (наприклад, створений пакет). Зважаючи на це, навіть сканування портів, які розповсюджуються протягом тривалого періоду часу, будуть записані, оскільки вони за своєю суттю будуть аномальними. Однак статистичні схеми виявлення аномалій мають і недоліки. Досвідчені зловмисники можуть навчити статистичне виявлення аномалій сприймати ненормальну поведінку як нормальну. Також може бути важко визначити порогові значення, які врівноважують ймовірність хибнопозитивних результатів з ймовірністю хибнонегативних результатів [35]. Крім того, статистичні методи потребують точних

статистичних розподілів, але не всю поведінку можна моделювати за допомогою суто статистичних методів.

### **2.2.2 Виявлення аномалій на основі машинного навчання**

Машинним навчанням можна вважати здатність програми та/або системи вивчати та покращувати свою продуктивність у певному завданні чи групі завдань з часом. Машинне навчання має на меті відповісти на багато тих самих питань, що й статистика чи аналіз даних. Однак, на відміну від статистичних підходів, які, як правило, зосереджуються на розумінні процесу які створили дані, методи машинного навчання зосереджені на створенні системи, яка покращує свою продуктивність на основі попередніх результатів. Іншими словами, системи, які базуються на парадигмі машинного навчання, мають можливість змінювати свою стратегію виконання на основі нової отриманої інформації.

Аналіз послідовності системних викликів. Один із широко розповсюджених методів машинного навчання для виявлення аномалій передбачає вивчення поведінки програми та розпізнавання значних відхилень від норми. У статті Forrest et al. [16] встановили аналогію між імунною системою людини та виявленням вторгнень. Вони зробили це шляхом пропозиції методології, яка передбачає аналіз послідовностей системних викликів програми для побудови нормального профілю. У статті було проаналізовано кілька програм на базі UNIX, таких як sendmail, lpr тощо, і показали, що кореляції в послідовностях системних викликів фіксованої довжини можна використовувати для створення нормального профілю програми [36]. Таким чином, програми, які показують послідовності, які відхиляються від звичайного профілю послідовності, можуть вважатися жертвами атаки. Розроблена ними система використовувалася лише в автономному режимі з використанням попередньо зібраних даних і використовувала досить простий алгоритм пошуку таблиці для вивчення профілів програм. Їхня робота була продовжена Hofmeug та ін. [17], де вони зібрали базу даних нормальної поведінки для кожної цікавої програми. Після створення стабільної бази даних для певної програми в певному середовищі база даних використовувалася для моніторингу поведінки програми. Послідовності

системних викликів формували набір нормальних шаблонів для бази даних, а послідовності, яких немає в базі даних, вказували на аномалії.

### **2.2.3 Виявлення аномалій на основі аналізу даних**

Щоб усунути ручні та спеціальні елементи з процесу побудови системи виявлення вторгнень, дослідники все частіше звертаються до використання методів інтелектуального аналізу даних для виявлення аномалій. Гроссман визначає інтелектуальний аналіз даних як «розкриття закономірностей, асоціацій, змін, аномалій і статистично значущих структур і подій у даних». Простіше кажучи, інтелектуальний аналіз даних — це здатність приймати дані як вхідні дані та витягувати з них закономірності чи відхилення, які не видно неозброєним оком. Інший термін, який іноді використовують, це відкриття знань. Інтелектуальний аналіз даних може допомогти покращити процес виявлення вторгнень, додавши рівень фокусу до виявлення аномалій. Визначаючи межі дійсної мережевої активності, інтелектуальний аналіз даних допоможе аналітику відрізнити аномальну активність від звичайного щоденного трафіку на мережі [37]. Система виявлення вторгнень, яка класифікує дані аудиту як нормальні або аномальні на основі набору правил, шаблонів або інших методів, може бути широко визначена як система виявлення вторгнень на основі класифікації. Процес класифікації зазвичай включає такі кроки:

Визначення атрибутів класу та класи з навчальних даних.

Визначення атрибутів для класифікації.

Навчання моделі, використовуючи навчальні дані.

Використання навченої моделі для класифікації невідомих зразків даних.

У літературі запропоновано різноманітні методи класифікації. До них належать методи генерації індуктивних правил, нечітка логіка, генетичні алгоритми та методи на основі нейронних мереж [38].

## 2.2.4 Нейромережевий метод

Застосування нейронних мереж є результатом неформально визначеної задачі, яка полягає у виявленні аномальних поведінкових патернів. Центральна ідея цього підходу включає надання нейронній мережі тренувального набору даних [57], які описують типову поведінку системи, дозволяючи мережі адаптуватися до цих даних. Таке тренування допомагає мережі розробити базове розуміння того, що вважається "нормальним". Вихідні дані мережі можуть бути представлені як коефіцієнт нормальності або як один із параметрів системи. При наявності сталості в вхідних даних припускається, що мережа зможе ефективно "навчатися" цим закономірностям [54]. Якщо під час роботи мережі вихідні дані виходять за рамки звичайної поведінки або значно відрізняються від вихідних параметрів системи, приймається висновок про можливу аномалію.

Для створення бази даних поведінки користувача збираються такі параметри, як часи активності користувача, місця початку робочих сесій та характеристики використання системних ресурсів. Ці дані потім перетворюються в числовий формат для подачі на вхід нейронної мережі з алгоритмом зворотнього поширення помилок (backpropagation neural network, BPNN). На виході мережі формується коефіцієнт, який приймає значення нуль для користувачів з нормальною поведінкою та одиницю для тих, чия поведінка вважається аномальною [39]. Мережа тренується на парах даних типу ("нормальні" параметри, 0) та ("аномальні" параметри, 1). Оскільки зібрання аномальних даних могло б вимагати від користувача ненормальної поведінки, такі дані часто генеруються штучно, що може ускладнити аналіз результатів на реальних даних.

В дослідженнях обмежена кількість команд дозволяє ідентифікувати користувачів, причому аналізується лише частота виконання цих команд, а не їх послідовність. Команди кодуються способом, який мінімізує кількість входів, необхідних для тренування мережі. У дослідженні представлено систему, яка ідентифікує користувачів, використовуючи обмежену кількість команд (100 команд), і не враховує їхню послідовність. Результати показують потенціал цього підходу, хоча

також відзначено велику кількість помилок типу "false positives" через нерегулярність поведінки користувачів, що вказує на потребу подальших досліджень та оптимізації методів аналізу.

### **2.2.5 Генетичні алгоритми**

Була створена генетична система виявлення вторгнень GBID, яка аналізує індивідуальну поведінку користувачів. Поведінка користувача аналізується як набір з трьох ключових елементів: індексу збігу, індексу ентропії та індексу новизни, які досліджуються з використанням генетичного алгоритму [40]. Профіль цієї поведінки потім застосовується для ідентифікації вторгнень, базуючись на аналізі попередньої поведінки.

У сфері кібербезпеки генетичні алгоритми ефективно ідентифікують аномальну поведінку, використовуючи їх здатність адаптуватися до динамічних умов та великих обсягів даних. Наприклад, вони можуть аналізувати логи серверів для виявлення аномалій у поведінці користувачів, вивчаючи такі параметри як інтервали між діями користувачів або їхню частоту дій. Використання генетичного програмування дозволяє створювати програми, які виявляють аномальні патерни поведінки, що значно підвищує здатність систем кібербезпеки реагувати на потенційні загрози швидко і точно. В цілому, генетичні алгоритми пропонують гнучке та масштабоване рішення для виявлення аномалій, що є критично важливим для забезпечення безпеки в сучасних інформаційних системах.

### **2.2.6 Кластерний метод виявлення аномалій**

Кластерні методи виявлення аномалій в сфері кібербезпеки об'єднують схожі елементи даних у кластери і розпізнають аномалії як такі, що не вписуються до жодного з кластерів або належать до дуже маленьких кластерів. Один з відомих кластерних методів — це k-середніх, який розподіляє даний набір даних на кілька груп, заснованих на схожості між елементами даних. В кожному з цих методів, точка

даних приєднується до кластеру, який має центроїд, найближчий до неї. Для виявлення аномалій з використанням методу k-середніх, алгоритм спочатку тренується на даних, які не містять аномалій, створюючи таким чином кластери, які відображають звичайні поведінкові патерни [41]. Після тренування, модель k-середніх може використовуватися для призначення кластерів новим точкам даних. Якщо точка даних розташована далеко від центроїдів усіх кластерів, то вона ймовірно представляє аномалію. Серед інших методів, що базуються на кластеризації, можна відзначити EM-алгоритм. Цей метод припускає, що більшість точок даних у наборі даних належать до основного розподілу, тоді як аномалії або викиди належать до іншого розподілу. EM-алгоритм спочатку визначає параметри розподілу, якому належать точки даних, а потім працює через дві фази: крок очікування, де оцінюється ймовірність того, що кожна точка даних належить до кожного кластера на основі поточних оцінок параметрів, і крок максимізації, де параметри оцінок коригуються залежно від ймовірностей, визначених на етапі очікування. EM-алгоритм продовжує чергувати ці дві фази, поки оцінки параметрів не стабілізуються. Після того, як кластери ідентифіковані, цей метод визначає аномалії, вимірюючи відстань між кожною точкою даних і центроїдом кластера, до якого вона належить. Точки, що розташовані значно далеко від центроїда, вважаються аномаліями.

DBSCAN також є методом кластеризації, який фокусується на виявленні областей високої щільності точок даних, які відокремлені менш щільними областями. Він працює шляхом створення області навколо кожної точки даних, визначеної заданою мірою відстані [42]. Точка вважається основною, якщо має достатню кількість сусідів у своїй близькості. Неосновні точки — це ті, що розташовані поблизу основної точки, але не мають достатньої кількості сусідів, щоб вважатися основними. DBSCAN створює кластери, об'єднуючи основні точки та будь-які неосновні точки неподалік. Точки, що не входять до жодного кластера або є частиною кластера з малою кількістю точок, визначаються як аномалії.

## 2.2.7 Метод опорних векторів

Метод опорних векторів (SVM — Support Vector Machines) являє собою витончений математичний інструмент для розробки функцій класифікації, що вирішують широкий спектр завдань в галузі машинного навчання. Його концепція базується на геометричному розумінні принципів класифікації. Припускається, є дві групи точок, які можна чітко розділити за допомогою площини в двовимірному просторі, що представляється лінією. Хоча існує безліч таких площин, вибір падає на ту, яка має однакову відстань до найближчих точок кожного з класів, відомих як опорні вектори. Завдання вибору найкращої площини зводиться до квадратичного програмування із застосуванням лінійних обмежень-нерівностей.

Переваги використання SVM зокрема включають:

- Здатність мінімізувати помилки класифікації, забезпечуючи високу точність визначення класів.
- Можливість використання лінійного класифікатора для аналізу даних, які на перший погляд не підлягають лінійному розділенню, тим самим зберігаючи простоту при забезпеченні ефективності.
- Ефективна робота з даними складної структури завдяки можливості використання різних ядерних функцій.
- Гнучкість алгоритму, де зміна в структурі даних потребує лише зміни ядерної функції, не торкаючись самого алгоритму.

Рішення в SVM зводиться до задачі квадратичного програмування, яке має однозначне рішення та для якого розроблено багато ефективних методів оптимізації, забезпечуючи роботу в реальному часі.

Однак, метод має також наступні недоліки:

- Чутливість до шуму: SVM дуже чутливий до шуму в тренувальному наборі даних. Наявність шуму може суттєво вплинути на положення оптимальної роздільної площини, що призводить до погіршення якості класифікації.
- Масштабування великих датасетів: SVM може бути неефективним для великих наборів даних через високу обчислювальну складність та вимоги до пам'яті,

особливо коли використовуються ядерні функції, оскільки потрібно обчислювати та зберігати велику матрицю подібності.

- Обмежена можливість інтерпретації: Моделі SVM, особливо з нелінійними функціями, можуть бути складними для інтерпретації, що ускладнює розуміння того, як рішення приймаються в межах моделі.

Для подолання викликів можна застосувати методи, засновані на теорії нечітких множин, хоча це значно збільшує обчислювальні витрати.

### 2.2.8 Байєсовські мережі

Байєсовські мережі відіграють важливу роль у сфері виявлення аномалій в системах кібербезпеки. Байєсовські мережі — це графічні моделі для статистичного моделювання, що базуються на теоремі Байєса. Ці мережі дозволяють обчислювати ймовірності подій, виходячи з залежностей між різними змінними в мережі. Кожен вузол у такій мережі представляє змінну, а стрілки між вузлами вказують на прямі ймовірнісні залежності [43].

Байєсовські мережі можуть використовуватися для моделювання складних залежностей між подіями в системі, що дозволяє прогнозувати аномальну поведінку. Наприклад, в роботі J. Han і J. Huang описується застосування цих мереж для аналізу системних журналів [17]. Автори розробили метод, який порівнює активність в системі з встановленими нормальними шаблонами поведінки. Цей підхід дозволяє не тільки виявляти аномалії, але й робити це з високою точністю завдяки попередньому навчанню на даних про нормальну активність.

Байєсовські мережі мають кілька значущих переваг, коли їх застосовують для виявлення аномалій у системах кібербезпеки. Однією з головних переваг є їхня адаптивність: ці мережі здатні ефективно реагувати на зміни у вхідних даних, оскільки базуються на ймовірностях, які можна динамічно оновлювати з появою нової інформації. Це робить їх ідеальними для систем, де умови або поведінка можуть швидко змінюватись [44]. Також, вони враховують невизначеність у вхідних даних, що є критично важливим у сферах, де повна впевненість у даних неможлива. Проте,

існують і певні недоліки. Налаштування баєсовських мереж може бути складним, оскільки потребує глибокого розуміння як структури даних, так і залежностей між ними. Цей процес може бути часомістким і вимагати високої кваліфікації в області статистики та ймовірностей. Крім того, для ефективного навчання мережі необхідні великі обсяги даних, які добре представляють всі можливі сценарії поведінки системи. У динамічних системах, де кількість змінних та їх взаємозв'язки можуть бути великими, збір та обробка такої кількості даних можуть стати серйозним викликом.

### 2.2.9 Дерева рішень

Дерева рішень (decision trees) використовуються не тільки для класифікації та прогнозування в загальних задачах аналізу даних, але й зокрема для виявлення аномалій у сфері кібербезпеки. Ці інструменти допомагають аналітично та візуально оцінювати потенційні загрози, використовуючи моделі, які можна легко інтерпретувати через їхню ієрархічну структуру. Дерево рішень для виявлення аномалій у кібербезпеці конструюється шляхом визначення атрибутів, які можуть вказувати на нестандартну поведінку системи або спроби несанкціонованого доступу. Кожна гілка дерева представляє рішення або вибір, що базується на конкретних атрибутах, а листя — це можливі кінцеві стани, які вказують на нормальну або аномальну діяльність.

Процес побудови дерева включає збір даних про зразки діяльності, які вже класифіковані як нормальні чи аномальні, утворюючи навчальну вибірку. Використовуючи ці дані, алгоритм навчання розвиває класифікатор, який в подальшому може бути застосований для ідентифікації нових, потенційно аномальних випадків.

Перед застосуванням на реальних системах, дерево рішень зазвичай піддається перехресній перевірці для оцінки його ефективності у виявленні аномалій [45]. Ця техніка включає розподіл навчальної вибірки на кілька частин, навчання на основі більшості цих частин, і тестування на решті, щоб забезпечити об'єктивність і надійність моделі. На заключних етапах, дерево рішень може бути скорочене шляхом

відсікання гілок, що не приносять значущого вкладу в класифікацію, або ж, навпаки, може бути розширене для забезпечення більшої точності та деталізації в аналізі. Цей підхід забезпечує можливість виявлення складних і рідкісних аномалій, що особливо важливо в умовах постійно зростаючої кількості кіберзагроз.

### **2.2.10 Модель на основі нелінійної динаміки**

Моделі на основі нелінійної динаміки використовуються для аналізу даних, особливо з складними системами, що проявляють непередбачувані і часто хаотичні зміни. Цей метод дозволяє розглядати змінні параметри систем, які змінюються у часі, що є критично важливим для розуміння динаміки аномалій. Аномальні дані часто мають відмінну від нормальних динаміку, що може виявитися у формі раптових сплесків активності, незвичайних патернів поведінки або інших непередбачуваних відхилень. Нелінійна динаміка дозволяє виявляти ці відхилення, оскільки вона спеціалізується на аналізі систем, що поведуться непередбачувано і мають складні взаємозв'язки.

Використання нелінійної динаміки стало ще ефективнішим з інтеграцією рекурентних нейронних мереж, таких як LSTM (Long Short-Term Memory). Ці мережі здатні зберігати інформацію на тривалий час і ідеально підходять для аналізу часових послідовностей даних. Дослідження А. Mosleh та його колег використовувало поєднання нелінійної динаміки з LSTM для виявлення аномалій у програмному забезпеченні [18]. Таке поєднання дозволяє моделям вчитися на історичних даних і виявляти незвичайні або аномальні патерни в поведінці нових даних. Ці моделі особливо корисні в сферах, де потрібно швидко виявляти потенційні збої або вторгнення. Наприклад, у моніторингу мережевого трафіку або управлінні фінансовими системами, де аномалії можуть вказувати на фінансові махінації або кібератаки [46]. Моделі на основі нелінійної динаміки забезпечують здатність виявляти складні аномалії, що можуть бути непомітними для традиційних методів аналізу, завдяки їх здатності обробляти велику кількість даних та враховувати нюанси у поведінці систем.

### **2.2.11 Гібридні системи**

Здатність моніторингу поточних систем виявлення вторгнень можна покращити за допомогою гібридного підходу, який складається як із стратегій виявлення аномалій, так і з сигнатурами [55]. У такій гібридній системі метод виявлення аномалій допомагає виявити нові або невідомі атаки, тоді як метод виявлення сигнатур виявляє відомі атаки. Техніка виявлення сигнатур також зможе виявляти атаки, ініційовані зловмисником, який намагається змінити патерни поведінки з метою перенавчати модуль виявлення аномалій, щоб він сприймав поведінку атаки як звичайну. Томбіні та ін. використали підхід, у якому техніка виявлення аномалій використовується для створення списку підозрілих предметів. Модуль класифікатора, який використовує метод виявлення сигнатур, потім класифікував підозрілі елементи на помилкові тривоги, атаки та невідомі атаки. Цей підхід працює на передумові, що компонент виявлення аномалій матиме високий рівень виявлення, оскільки пропущені вторгнення не можуть бути виявлені компонентом виявлення подальших сигнатур. Крім того, також передбачається, що компонент виявлення сигнатур зможе визначати помилкові тривоги [48]. Хоча гібридна система все ще може пропускати певні типи атак, зменшена частота помилкових спрацювань збільшує ймовірність перевірки більшості сповіщень.

### **2.3 Відношення типів аномалій до методів їхнього виявлення**

Попередньо було розглянуто і досліджено перелік типів та методів виявлення аномалій. В таблицях 2.1 та 2.2 відображено співвідношення між різними типами аномалій та найкращими методами їх виявлення.

Відношення типів забезпечує цінний ресурс для визначення найефективніших методів виявлення, враховуючи специфіку кожного типу аномалії. Це не тільки спрощує процес вибору відповідних методів для конкретних сценаріїв, але й сприяє кращому розумінню потенційних взаємодій та обмежень цих методів [59].

Таблиця 2.1.

## Відношення виду аномалії до методу виявлення

		Вид аномалії					
		Аномалії в поведінці користувачі в	Аномалії в мережевому трафіку	Аномалії в системних журналах	Аномалії в даних	Аномалії в ресурсах	Аномалії в програмному забезпеченні
Метод виявлення	Статичне виявлення аномалій						
	Виявлення аномалій на основі машинного навчання						
	Виявлення аномалій на основі аналізу даних						
	Нейромережовий метод						
	Генетичні алгоритми						
	Кластерний метод виявлення аномалій						
	Метод опорних векторів						
	Байесовські мережі						
	Дерева рішень						
	Модель на основі нелінійної динаміки						
	Гібридні системи						

Відношення виду аномалії до методу виявлення

		Вид аномалії				
		Аномалії в архітектурі системи	Аномалії в ідентифікації та автентифікації	Аномалії в захисті від вразливостей	Хмарні аномалії	Аномалії, пов'язані з часом
Метод виявлення	Статичне виявлення аномалій					
	Виявлення аномалій на основі машинного навчання					
	Виявлення аномалій на основі аналізу даних					
	Нейромережевий метод					
	Генетичні алгоритми					
	Кластерний метод виявлення аномалій					
	Метод опорних векторів					
	Байесовські мережі					
	Дерева рішень					
	Модель на основі нелінійної динаміки					
Гібридні системи						

Ці таблиці надають теоретичний фундамент для подальших досліджень у галузі виявлення аномалій кібербезпеки.

## 2.4 Опис запропонованої моделі виявлення аномалій

У цій кваліфікаційній роботі досліджується інтеграції штучного інтелекту (AI) [60] із системами SIEM. Використовуючи можливості штучного інтелекту, включаючи машинне навчання, це дослідження має на меті значно підвищити ефективність SIEM у виявленні аномалій кібербезпеки на ранніх етапах.

Під "моделлю виявлення аномалій" розуміється комплексний підхід до ідентифікації та аналізу потенційних загроз у інформаційних системах, який включає в себе не лише застосування машинного навчання та штучного інтелекту, а й весь життєвий цикл даних, а саме: генерація подій на кінцевих точках, доставка їх в систему управління інформаційною безпекою та подіями інформаційної безпеки,

експорт логів для аналізу chatGPT, аналіз їх, враховуючи попередньо зазначані параметри, отримання результатів про наявність аномалії в логах оброблених за допомогою chatGPT.

### **2.4.1 Методологія**

Дослідження розробляє методику інтеграції штучного інтелекту, зокрема chatGPT, із системою управління інформаційною безпекою та подіями, стек ELK, з метою виявлення інцидентів кібербезпеки. Методологія включає наступні етапи:

Розробка інтеграційного рішення:

Створення скрипта, який буде регулярно передавати логи з ElasticSearch до моделі штучного інтелекту chatGPT.

Перевірка здатності моделі обробляти та аналізувати великі обсяги логів в реальному часі.

Тренування та навчання моделі:

Наповнення чату в ШІ інформацією, на основі якої вона повинна класифікувати підозрілу поведінку і нормальну активність.

Оцінка здатності моделі класифікувати події як нормальні операції, підозрілі дії або інциденти кібербезпеки.

Валідація моделі:

Проведення тестування на реальних логах для перевірки ефективності виявлення інцидентів.

Співставлення результатів моделі з ручним аналізом логів для оцінки дієвості штучного інтелекту.

Реалізація та моніторинг:

Інтеграція штучного інтелекту в SIEM.

Моніторинг роботи моделі для виявлення потреби у коригуванні та вдосконаленні.

Аналіз результатів та оцінка ефективності:

Статистичний аналіз результатів виявлення, включаючи частоту вірних і помилкових позитивних спрацювань.

Оцінка впливу штучного інтелекту на процеси кібербезпеки та прийняття рішень.

Ця методологія надає основу для вивчення інтеграції сучасних технологій штучного інтелекту на системи кібербезпеки, зокрема SIEM, і може слугувати основою для подальших інновацій у цій області.

### **2.4.2 Сфери застосування і переваги**

Дана модель інтеграції штучного інтелекту з системою управління інформаційною безпекою та подіями (SIEM) може бути застосована в будь-якій сучасній ІТ-галузі, де існує потреба автоматизації процесів виявлення інцидентів кібербезпеки. Це особливо актуально для сфер, де високий об'єм даних та постійна потреба в їхньому моніторингу ставлять високі вимоги до оперативності та точності реагування на потенційні загрози.

Модель здатна адаптуватися до різних умов експлуатації, що робить її універсальним рішенням для підвищення ефективності реагування на інциденти кібербезпеки в широкому спектрі ІТ-організацій.

Очікувані результати:

Ефективність виявлення шкідливих програм;

Точність у виявленні вторгнень;

Автоматизація рутинних процесів;

Зменшення помилкових спрацювань;

Неперервний моніторинг;

Вдосконалення процесів прийняття рішень;

### 2.4.3 Виклики і обмеження

Штучний інтелект (AI) з функціоналом NPL, особливо при інтеграції з системами SIEM, такими як ElasticSearch, для виявлення кіберінцидентів, може ефективно захищати від атак. Однак, існує ризик, що зловмисники можуть використовувати цю технологію для своїх цілей.

Серед викликів, які виникають з розвитком ШІ, є хакерські атаки на самі системи штучного інтелекту, а також потреба у вдосконаленні технологій для запобігання такому використанню. Далі представлено перелік цих викликів:

**Надійність.** Продуктивність штучного інтелекту значною мірою залежить від тривалого навчання на основі даних, що не завжди надійно. Дві різні речі для людей, можуть виглядати ідентичними ШІ. У таких ситуаціях штучний інтелект не завжди може розрізнити їх, що потенційно може призвести до неточних суджень. Крім того, їх здатність приймати рішення може бути ще більше послаблена через атаку на витік даних [4]. Навчальний процес штучного інтелекту часто працює як чорний ящик із обмеженою прозорістю його внутрішньої роботи. Щоб бути надійним, штучний інтелект повинен моти можливість пояснити свою логіку та принцип за яким він надає рішення[5].

**Невідомі загрози.** Моделі штучного інтелекту навчаються на існуючих даних, і в результаті ці моделі штучного інтелекту можуть зіткнутися з труднощами в розпізнаванні нових атак, яких немає в їхній базі даних.

**Конфіденційність даних.** AI вимагає доступу до особистих даних користувачів, щоб ефективно передбачати потенційні атаки або фільтрувати спам. Однак, таким чином даний аспект зачіпає конфіденційність, що відомо як парадокс конфіденційності [2]. Цієї проблеми неможливо уникнути, оскільки ці технології базуються на аналізі даних, наданих користувачем.

## 2.4.4 Концептуальна архітектура

На рисунку 2.1 зображена загальна архітектура запропонованого рішення. Джерелами логів можуть бути:

- сервери (Linux, Windows), мережеві пристрої (маршрутизатори, комутатори), пристрої безпеки (брандмауери, IDS) та програми;

- операційні системи: будь-які, включаючи дистрибутиви Linux (Ubuntu, CentOS) і Windows Server;

- множинності: кілька екземплярів у різних сегментах мережі та географічних точках.

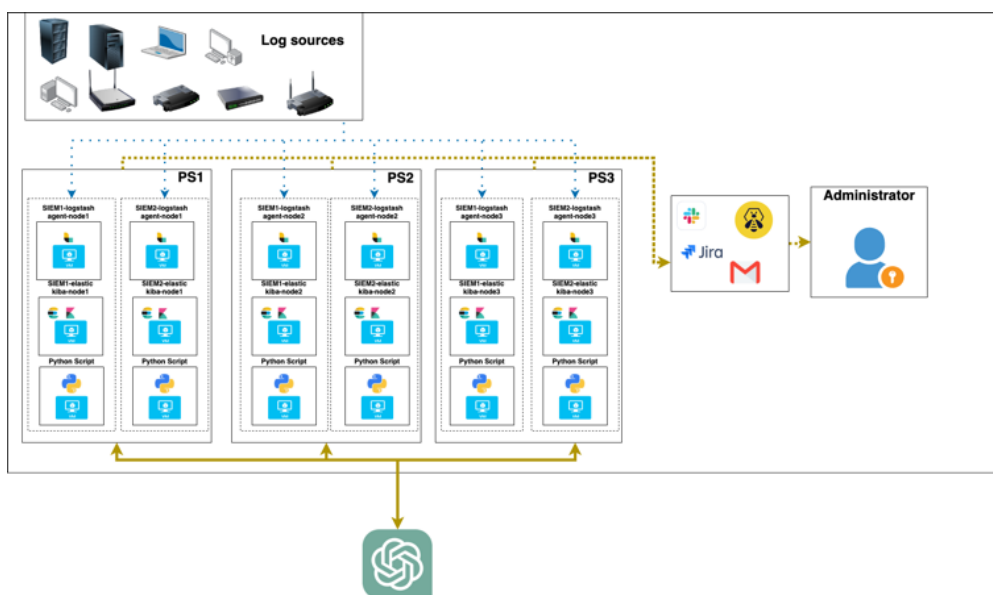


Рисунок 0.1 – Архітектура запропонованого рішення

Далі відбувається агрегація та нормалізація. Компоненти:

Filebeat (відправник логів). Розгортається на кожному джерелі журналів подій.

Logstash (процесор логів). Може починатися з одного екземпляра, але за потреби може масштабуватися по горизонталі.

Elasticsearch (сховище та пошукова система). Кластер з щонайменше трьома вузлами для забезпечення високої доступності та надмірності.

Щодо операційних систем, то перевага надається дистрибутивам Linux через їх стабільність та продуктивність (наприклад, Ubuntu, CentOS). Отже, логи

відправляються з джерел за допомогою Filebeat, обробляються і нормалізуються Logstash, а потім зберігаються в кластері Elasticsearch.

Доставка логів в chatGPT відбувається за допомогою скрипту Python. Скрипт періодично запитує Elasticsearch про нові журнали, аналізуючи партії подій. Отримує результати аналізу і генерує сповіщення на основі попередньо визначених критеріїв. Скрипт Python отримує журнали з Elasticsearch, надсилає їх до модуля аналізу журналів на основі GPT, після чого генератор оповіщень обробляє результати аналізу. Далі модуль аналізу логів на основі chatGPT. Це хмарний екземпляр, отже, підтримка відбувається на стороні постачальника. На початковому етапі може бути достатньо одного екземпляра. Однак варто розглянути можливість масштабування на основі обчислювального попиту та вимог до затримок.

Скрипт отримує журнали з Elasticsearch, надсилає їх до модуля аналізу журналів на основі NLP GPT, після чого генератор оповіщень обробляє результати аналізу. Генератор оповіщень отримує результати аналізу і генерує сповіщення на основі попередньо визначених критеріїв. Загальна діаграма послідовностей має вигляд зображений на рисунку .

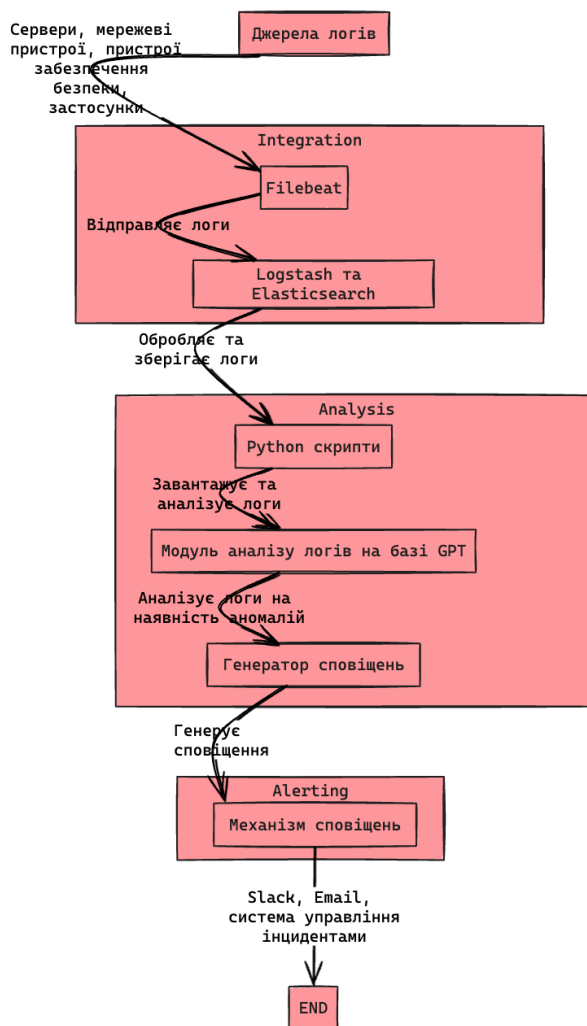


Рисунок 0.2 – Загальна діаграма послідовностей

Здатність штучного інтелекту аналізувати дані можна використовувати для ідентифікації та класифікації шкідливої активності. Штучний інтелект має значний потенціал, проте надзвичайно важливо забезпечити його технічну надійність і здатність виявляти нові загрози.

## Висновки до розділу 2

Складність і різноманітність сучасних інформаційних систем ускладнюють виявлення аномалій, що вимагає розробки новітніх технологічних рішень. Чим швидше аномалії будуть виявлені, тим швидше можна ізолювати уражені системи та мінімізувати збитки. У другому розділі кваліфікаційної роботи розглядається критичне значення своєчасного виявлення аномалій у сфері кібербезпеки, важливість

якого обумовлена потребою в мінімізації впливу кіберзагроз на інформаційні системи. Особливу увагу приділено викликам, з якими стикаються інформаційні системи, включаючи складність сучасних мереж, великий обсяг та різноманітність даних, еволюцію кіберзагроз, а також хибнопозитивні та хибнонегативні спрацьовування.

Розглянуті методи виявлення аномалій включають статистичний аналіз, машинне навчання, нейромережеві методи, генетичні алгоритми та інші, кожен з яких має свої специфічні застосування та обмеження. Додатково було проведено детальний аналіз кожного з методів виявлення аномалій. Як результат даного дослідження, було створено таблицю, що відображає типи аномалій та найкращі методи їх виявлення.

Незважаючи на різноманітність доступних методів, існує потреба у розробці комплексного рішення, яке могло б цілісно покрити всі аспекти кібербезпеки. Рішення для виявлення аномалії в інформаційній мережі можуть бути досить дорогими. Недостатність фінансових ресурсів може обмежувати можливості організацій у впровадженні ефективних систем виявлення аномалій. Тому, було описано запропоновану модель виявлення аномалій, яка базується на інтеграції стеку ELK, який буде в якості SIEM, та chatGPT, який буде на попередньо заданих параметрах ідентифікувати наявність аномалій у надісланих логах. В розділі зображено архітектуру запропонованого рішення та описано її компоненти. Також наведено діаграму алгоритму роботи моделі, починаючи з джерела логів, закінчуючи сповіщенням фахівця з безпеки про наявну аномалію в системі.

## РОЗДІЛ 3

### РЕАЛІЗАЦІЯ ЗАПРОПОНОВАНОЇ МОДЕЛІ

#### 3.1 Топологія тестової інфраструктури для реалізації рішення

Сегментація мережі має вирішальне значення для ефективного та безпечного керування складним мережевим середовищем та забезпечує ключові аспекти безперервної роботи бізнесу:

- **Безпека:** Сегментація покращує безпеку мережі шляхом поділу мережі на менші, керовані сегменти. Кожен сегмент може мати власну політику безпеки та засоби контролю, що ефективно обмежує розповсюдження порушень безпеки всередині сегментів. Наприклад, при компрометації одного сегменту злоумисником, порушення не обов'язково поширяться на інші частини мережі [51].

- **Продуктивність:** сегментація може спричинити зменшення завантаженості мережі, локалізуючи трафік в межах сегментів і відповідно зменшуючи обсяг трафіку, який проходить через всю мережу. Це покращує продуктивність мережі, оскільки між сегментами перетинається лише необхідний трафік.

Для побудови інфраструктури в тестових умовах було реалізовано сегментування мережі, котре наближене до найкращих практик проєктування корпоративних мереж на реальних підприємствах та компаніях, Таким чином була відображена можливість розгортання комплексу для виявлення аномалій в реальних системах.

Еталонним зразком для побудови топології рішення було взято найкращі практики сегментуванні мережі. На рисунку 3.1 [61] зображено умовні позначення при створенні топології інфраструктури.

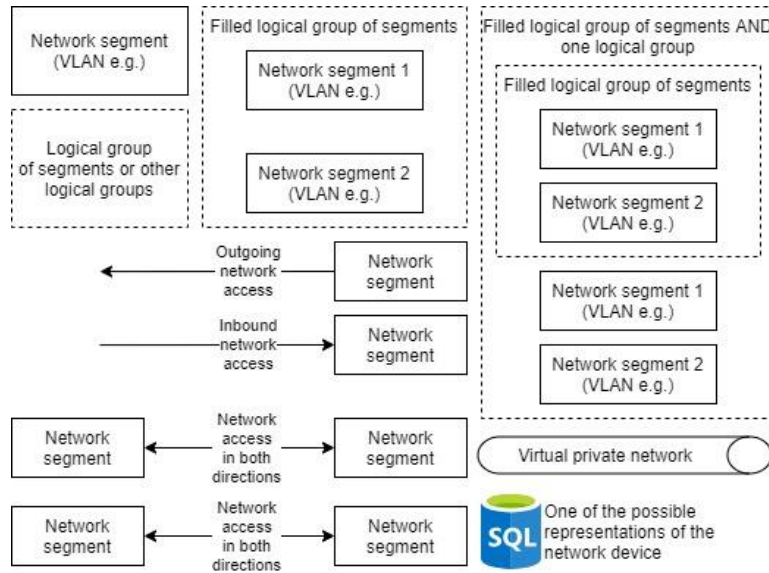


Рисунок 0.1 – Умовні позначення при створенні топології інфраструктури

На рисунку 3.2 показані найкращі практики сегментування мережі, які відображають логічне розділення ресурсів компанії для забезпечення безпеки в межах периметру організації [61].

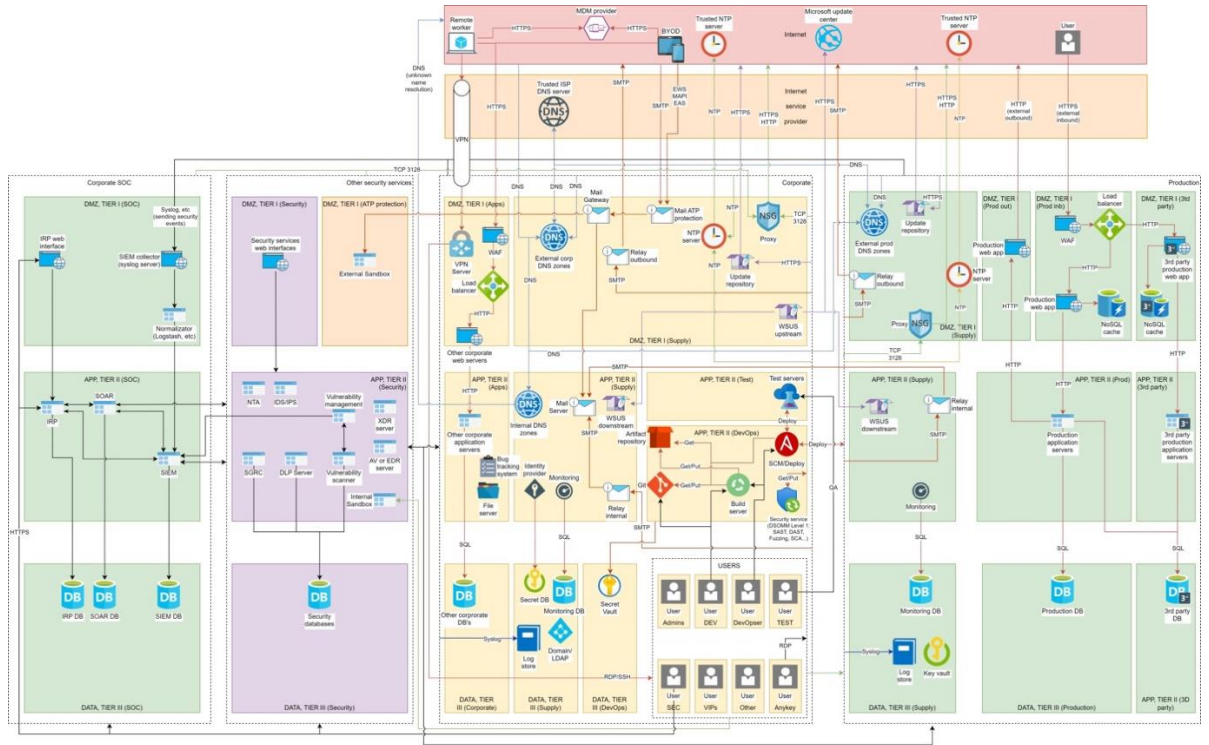


Рисунок 0.2 – Приклад побудованої топології за найкращими практиками

Для реалізації запропонованої моделі виявлення аномалій в межах домашньої лабораторії було реалізовано топологію, зображену на рисунку 3.3.

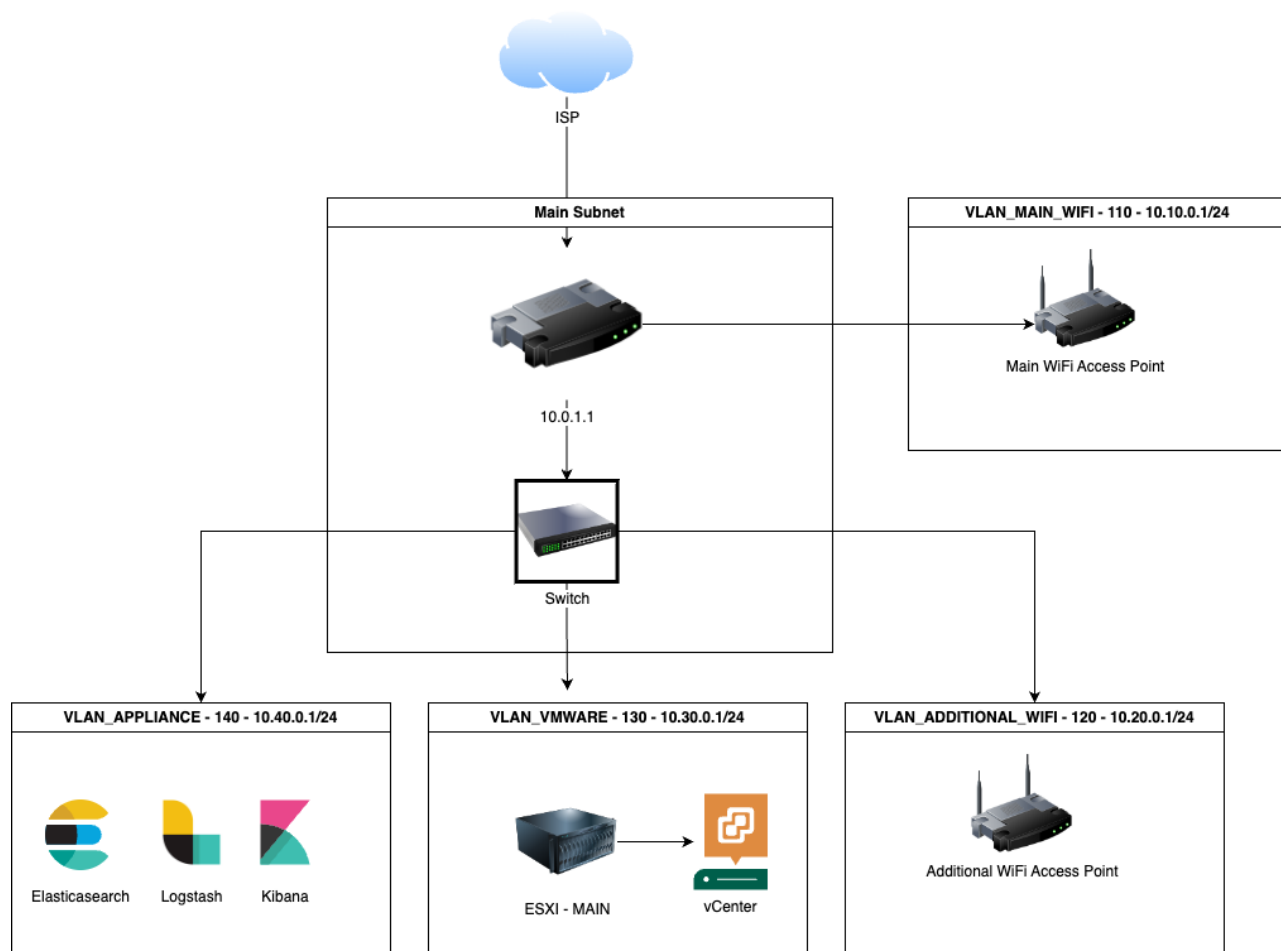


Рисунок 0.3 – Верхньорівнева топологія тестової інфраструктури

Логічне розділення окремих складових проекту реалізовано за допомогою віртуальних локальних мереж (VLAN), в окремих підмережах за маскою /24. В цілому, використання віртуальних локальних мереж для сегментації мережі є особливо ефективним з кількох причин:

- Віртуальні локальні мережі можуть забезпечити гнучкість логічного сегментування мережі без необхідності фізичного розділення пристроїв. Тобто, пристрої в одній VLAN можуть взаємодіяти так, ніби вони знаходяться в одній фізичній мережі, навіть якщо вони підключені до різних комутаторів.

- Віртуальні локальні мережі працюють за допомогою конфігурації існуючого мережевого обладнання, тому вони зменшують потребу в додатковому фізичному обладнанні.
- Віртуальні локальні мережі спрощують управління мережею. Зміни в структурі мережі можуть бути виконані за допомогою програмних конфігурацій без необхідності фізичного переміщення пристроїв або зміни кабелів.
- Віртуальні локальні мережі підвищують безпеку, оскільки при коректному налаштуванні відокремлюють конфіденційні дані і сервіси від інших компонентів мережі. Трафік у VLAN ізольований від решти мережі, що допомагає запобігти несанкціонованому доступу.
- Віртуальні локальні мережі дозволяють легко масштабувати сегменти мережі в міру зростання або зміни потреб. Додавання або модифікація VLAN не вимагає значних змін у фізичній інфраструктурі, що полегшує масштабування операцій.

Для виконання поставлених задач і цілей даний метод сегментації є найкращим, оскільки відображає реальну корпоративну мережу.

### **3.2 Побудова інфраструктури**

В рамках окремого VLAN\_APPLIANCE, як зображено на рисунку 3.4, було створено наступну низку віртуальних машин для побудови моделі та проведення дослідження.

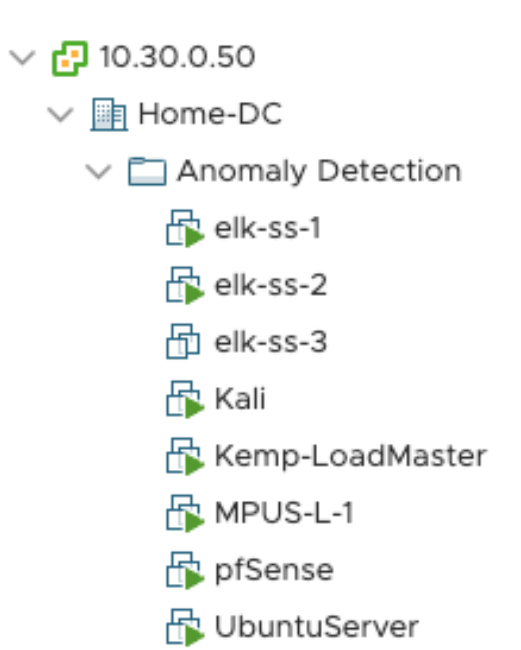


Рисунок 0.4 – віртуальні машини для побудови модулі

В таблиці 3.1 наведено опис віртуальних машин, які були підняті для тестового середовища.

Таблиця 3.1

Опис тестового середовища

IP	Віртуальна машина	Опис
10.40.0.51	elk-ss-1	Перша нода кластеру Elasticsearch, на котрій встановлено ОС Ubuntu Server 22.04, Kibana, Elastic Agent, застосовано політику Fleet Server та політики інтеграцій для збору логів
10.40.0.52	elk-ss-2	Друга нода кластеру Elasticsearch, на котрій встановлено ОС Ubuntu Server 22.04, Kibana, Elastic Agent, застосовано політику Fleet Server та політики інтеграцій для збору логів

IP	Віртуальна машина	Опис
10.40.0.53	elk-ss-3	Третя нода кластеру Elasticsearch, на котрій встановлено ОС Ubuntu Server 22.04, Kibana, Elastic Agent, застосовано політику Fleet Server та політики інтеграцій для збору логів
192.168.1.102	Kali	Хост зі встановленою ОС Kali linux, котрий виступає в ролі атакуючого
10.40.0.101	Kemp-LoadMaster	Хост зі встановленим Kemp Virtual LoadMaster для балансування навантаження та запитів між нодами кластеру elasticsearch
10.40.0.41	MPUS-L-1	Хост зі встановленою ОС Ubuntu Server 22.04 та виступає в ролі платформи на котрій запускається скрипт для експорту та обробки логів
10.40.0.101	pfSense	Хост зі встановленим віртуальним фаєрволом pfSense, котрий виступає віртуальним маршрутизатором між підмережами та є джерелом логів для подальшого дослідження
192.168.2.100	UbuntuServer	Хост зі встановленою ОС Ubuntu Server 22.04, котра виступає в ролі атакованого

Для побудови мережевої зв'язності тестового середовища додатково створено дві віртуальні портгруппи LocalDMZ\_1 та LocalDMZ\_2 в котрих розташовані хости Kali та Ubuntu Server. Даний процес зображено на рисунку 3.5.

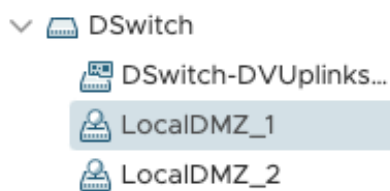


Рисунок 0.5 – Дві віртуальні портгрупи

Таким чином, топологія мережі має вигляд, зображений на рисунку 3.6.

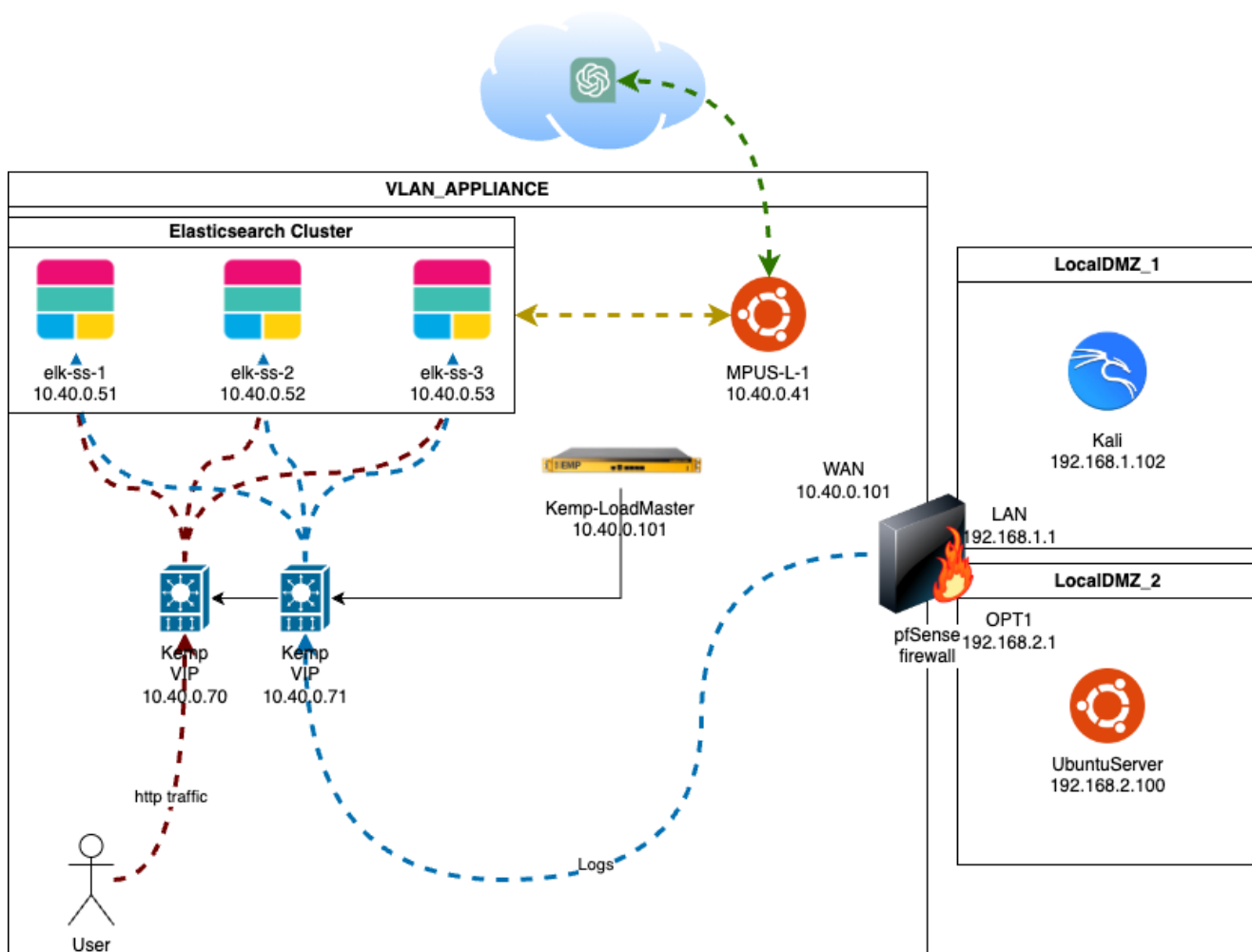


Рисунок 0.6 – Топологія інфраструктури для проведення дослідження щодо виявлення аномалій кібербезпеки в інформаційних системах

### 3.3 Налаштування кластеру Elasticsearch

Зазвичай інформаційні системи, особливо системи управління інформаційною безпекою та подіями інформаційної безпеки, повинні ефективно обробляти великі обсяги даних (подій, які відбуваються в межах компанії).

Elasticsearch, потужний пошуковий двигун та аналітична система, став досить поширеним завдяки своїй здатності швидко аналізувати великі набори даних.

Центральним елементом ефективності Elasticsearch є його кластерна архітектура, яка здатна забезпечити масштабування та відмовостійкість системи.

Застосування кластера з декількома нодами замість одиночної ноди відіграє ключову роль у забезпеченні відмовостійкості бізнесу, продуктивності обробки даних та масштабованості інформаційної корпоративної системи.

Серед переваг використання багатонодового кластера Elasticsearch у контексті кібербезпеки не тільки здатність забезпечити високий рівень доступності і відмовостійкості сервісу, але й можливість оптимізації швидкості обробки запитів та аналізу даних.

Ці аспекти є критично важливими для швидкого виявлення аномалій і адекватної реакції на інциденти безпеки, гарантуючи ефективність кіберзахисту великих інформаційних систем.

Кластер Elasticsearch з декількома вузлами є оптимальним рішенням для побудови розширених систем кібербезпеки. Ключові причини надані на рисунку 3.7:

Напря́м	Причина	Опис
<b>Масштабування та Розподіл Даних</b>	<b>Горизонтальне масштабування</b>	Кластери з декількома нодами дозволяють горизонтальне масштабування, що є критично важливим для обробки великих обсягів даних. Вона полягає у додаванні більше нод до кластера для підвищення його продуктивності та місткості.
	<b>Шардування</b>	Elasticsearch автоматично розподіляє дані (шарди) по всіх нодах кластера. Це дозволяє ефективно виконувати паралельну обробку запитів, що значно покращує швидкість пошуку та аналізу даних.
<b>Відмовостійкість та Доступність</b>	<b>Реплікація даних</b>	У кластері можна налаштувати реплікацію даних між нодами, що забезпечує високу доступність даних навіть у випадку відмови однієї чи кількох нод.
	<b>Відновлення після збоїв</b>	Elasticsearch може автоматично відновити репліки на інших нодах, забезпечуючи цілісність даних та неперервність бізнес-процесів.
<b>Балансування навантаження</b>	<b>Рівномірне розподілення навантаження</b>	Кластеризація дозволяє розподіляти навантаження запитів та індексації даних між декількома нодами, тим самим знижуючи ризик перевантаження одного сервера, що може стати критичним для систем, які залежать від швидкості обробки великих обсягів інформації.
<b>Підвищення продуктивності</b>	<b>Паралельна обробка</b>	Наявність кількох нод дозволяє одночасно обробляти багато запитів, значно зменшуючи час відповіді для кінцевих користувачів.
	<b>Ефективність аналітики</b>	Під час аналізу виявлення аномалій, можливість швидко обробляти та аналізувати дані є критично важливою, оскільки це дозволяє оперативно реагувати на потенційні загрози.

Рисунок 0.7 – Причини використання кластеру Elasticsearch з декількома вузлами

На рисунку 3.8 зображений процес реєстрації ноди elasticsearch в кластер за допомогою токєну. Аналогічний процес реалізовано з нодою 3.



```

#
network.host: 10.40.0.51
#
# By default Elasticsearch listens for HTTP traffic on the first free
# finds starting at 9200. Set a specific HTTP port here:
#
http.port: 9200
# For more information, consult the network module documentation.
#
# ----- Discovery -----
#
# Pass an initial list of hosts to perform discovery when this node is
# The default list of hosts is ["127.0.0.1", ":::1"]
#
discovery.seed_hosts:
- 10.40.0.51:9300
- 10.40.0.52:9300
#
# Bootstrap the cluster using an initial set of master-eligible nodes:
#
#cluster.initial_master_nodes: ["node-1", "node-2"]

# ----- Network -----
#
# By default Elasticsearch is only accessible on localhost
# address here to expose this node on the network:
#
network.host: 10.40.0.52
#
# By default Elasticsearch listens for HTTP traffic on the
# finds starting at 9200. Set a specific HTTP port here:
#
http.port: 9200
# For more information, consult the network module docum
#
# ----- Discovery -----
#
# Pass an initial list of hosts to perform discovery whe
# The default list of hosts is ["127.0.0.1", ":::1"]
#
#discovery.seed_hosts: ["host1", "host2"]
#
# Bootstrap the cluster using an initial set of master-e
#

```

Рисунок 0.10 – Налаштування конфігураційного файлу elasticsearch (2)

Kibana – це веб-інтерфейс, за допомогою якого відбувається візуалізація та управління даними, які зберігаються у Elasticsearch. Це програмне забезпечення з відкритим кодом, яке дозволяє користувачам легко створювати складні графіки та дашборди для глибшого аналізу великих обсягів даних, забезпечуючи цілком зрозумілі інструменти для прямого візуального представлення пошукових запитів, метрик, і логів.

Kibana використовується для візуального представлення даних у вигляді графіків, карт і табличних представлень. З її допомогою фахівці з безпеки можуть визначати тенденції, виявляти аномалії, і моніторити метрики в реальному часі.

Особливо це корисно в сфері кібербезпеки, оскільки швидке і точне інтерпретування даних може критично вплинути на реакцію на інциденти безпеки, що можуть бути індикаторами вторгнення.

Аналогічно, як і з Elasticsearch наявність декількох нод Kibana є важливим, враховуючи ряд причин:

- Відмовостійкість: наявність декількох нод Kibana забезпечує високий рівень безперервності бізнесу. У випадку недоступності однієї ноди, інші можуть продовжувати надавати доступ до візуалізацій і дашбордів.
- Балансування навантаження: запити користувачів можна розподіляти між декількома нодами Kibana, таким чином контролюючи навантаження на сервери. Це

допомагає уникнути перевантаження однієї ноди, особливо в моменти піку користувацької активності.

- Масштабування: З розширенням мережі відповідно збільшується обсяг даних і користувачів. Тому може знадобитися більше ресурсів для обслуговування всіх запитів. Додаткова нода Kibana дозволяє легко масштабувати сервіс, забезпечуючи ефективність обробки запитів та відображення візуалізацій.

Враховуючи ряд причин, описаних вище, було сконфігуровано три ноди Kibana. На рисунках 3.11 – 3.13 зображено налаштування конфігураційних файлів Kibana.

```

GNU nano 6.2 /etc/kibana/kibana.yml *
# Kibana is served by a back end server. This setting specifies the
server.port: 5601

# Specifies the address to which the Kibana server will bind. IP address
# The default is 'localhost', which usually means remote machines will
# To allow connections from remote users, set this parameter to a non-
server.host: "0.0.0.0"

# Enables you to specify a path to mount Kibana at if you are running
# Use the `server.rewriteBasePath` setting to tell Kibana if it should
# from requests it receives, and to prevent a deprecation warning at
# This setting cannot end in a slash.
#server.basePath: ""

# Specifies whether Kibana should rewrite requests that are prefixed
# `server.basePath` or require that they are rewritten by your reverse
# Defaults to `false`.
#server.rewriteBasePath: false

# Specifies the public URL at which Kibana is available for end users
# `server.basePath` is configured this URL should end with the same
#server.publicBaseUrl: ""

# The maximum payload size in bytes for incoming server requests.
#server.maxPayload: 1048576

# The Kibana server's name. This is used for display purposes.
#server.name: "your-hostname"

# ===== System: Kibana Server (Optional) =====
# Enables SSL and paths to the PEM-format SSL certificate and SSL key
# These settings enable SSL for outgoing requests from the Kibana server
#server.ssl.enabled: false
#server.ssl.certificate: /path/to/your/server.crt
#server.ssl.key: /path/to/your/server.key

# ===== System: Elasticsearch =====
# The URLs of the Elasticsearch instances to use for all your queries
elasticsearch.hosts:
- https://10.40.0.51:9200
- https://10.40.0.52:9200

GNU nano 6.2 /etc/kibana/kibana.yml *
# Kibana is served by a back end server. This setting specifies the
server.port: 5601

# Specifies the address to which the Kibana server will bind. IP address
# The default is 'localhost', which usually means remote machines will
# To allow connections from remote users, set this parameter to a non-
server.host: "0.0.0.0"

# Enables you to specify a path to mount Kibana at if you are running
# Use the `server.rewriteBasePath` setting to tell Kibana if it should
# from requests it receives, and to prevent a deprecation warning at
# This setting cannot end in a slash.
#server.basePath: ""

# Specifies whether Kibana should rewrite requests that are prefixed
# `server.basePath` or require that they are rewritten by your reverse
# Defaults to `false`.
#server.rewriteBasePath: false

# Specifies the public URL at which Kibana is available for end users
# `server.basePath` is configured this URL should end with the same
#server.publicBaseUrl: ""

# The maximum payload size in bytes for incoming server requests.
#server.maxPayload: 1048576

# The Kibana server's name. This is used for display purposes.
#server.name: "your-hostname"

# ===== System: Kibana Server (Optional) =====
# Enables SSL and paths to the PEM-format SSL certificate and SSL key
# These settings enable SSL for outgoing requests from the Kibana server
#server.ssl.enabled: false
#server.ssl.certificate: /path/to/your/server.crt
#server.ssl.key: /path/to/your/server.key

# ===== System: Elasticsearch =====
# The URLs of the Elasticsearch instances to use for all your queries
elasticsearch.hosts:
- https://10.40.0.51:9200
- https://10.40.0.52:9200

```

Рисунок 0.11 – Налаштування конфігураційного файлу kibana (1)

```

dential/token/kibana_token
{"created":true,"token":{"name":"kibana_token","value":"AAEAAWsYXN0a
WMva2liYW5hL2tpYmFuYV90b2t1bjpkdWRzVUhhUWFFOcUNWTFmo4NTl3SF93"}}root@e/
root@elk-ss-1:/usr/share/kibana/bin# ./kibana-keystore add elasticsea
rch.serviceAccountToken
Enter value for elasticsearch.serviceAccountToken: *****
*****
root@elk-ss-1:/usr/share/kibana/bin#

root@elk-ss-2:/usr/share/elasticsearch/bin# cd /usr/share/kibana/
root@elk-ss-2:/usr/share/kibana# cd bin/
root@elk-ss-2:/usr/share/kibana/bin# ./kibana-keystore add elasticsearch
serviceAccountToken
Enter value for elasticsearch.serviceAccountToken: *****
*****
root@elk-ss-2:/usr/share/kibana/bin#

```

Рисунок 0.12 – Налаштування конфігураційного файлу kibana (2)

```

#ops.interval: 5000
# Specifies locale to be used for all localizable strings, dates and
# Supported languages are the following: English (default) "en", Chi
#i18n.locale: "en"

# ===== Frequently used (Optional)=====
xpack.security.encryptedKey: "yxffsrgfwoaztwbkinceleqfxcxocveazuy>
xpack.reporting.encryptedKey: "fddhsrpzydreurfbyyylkipeqthuadnizvpb>

# ===== Saved Objects: Migrations =====
# Saved object migrations run at startup. If you run into migration>

#ops.interval: 5000
# Specifies locale to be used for all localizable strings, dates and
# Supported languages are the following: English (default) "en", Chi
#i18n.locale: "en"

# ===== Frequently used (Optional)=====
xpack.security.encryptedKey: "yxffsrgfwoaztwbkinceleqfxcxocveazuy>
xpack.reporting.encryptedKey: "fddhsrpzydreurfbyyylkipeqthuadnizvpb>

# ===== Saved Objects: Migrations =====
# Saved object migrations run at startup. If you run into migration>

```

Рисунок 0.13 – Налаштування конфігураційного файлу kibana (3)

На рисунку 3.14 зображено, що в інтерфейсі Kibana спостерігається успішно зібраний кластер з 3 нод Elasticsearch та 3 нод Kibana.

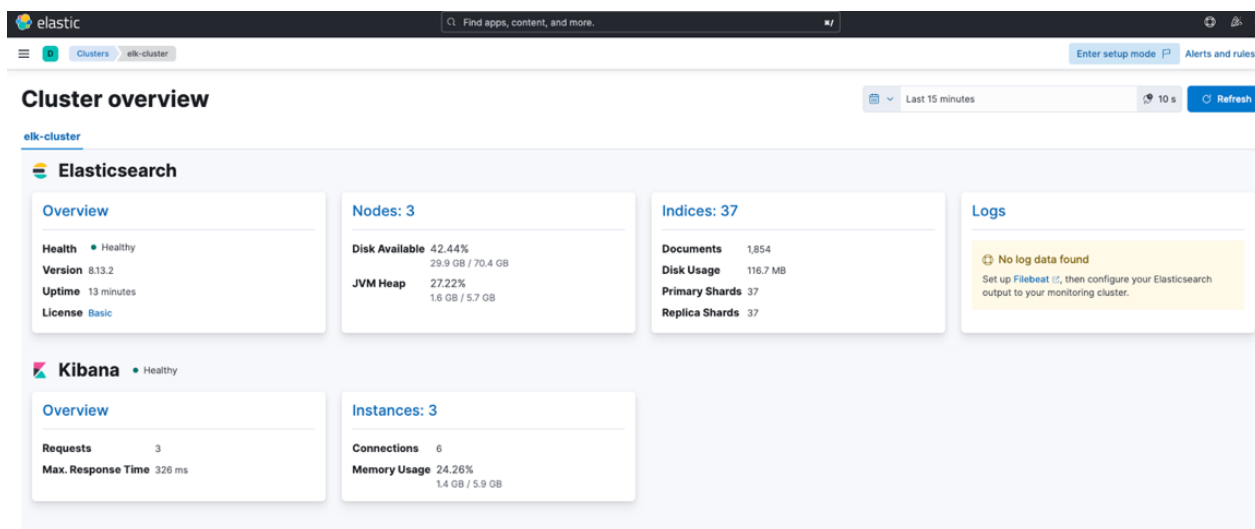


Рисунок 0.14 – Успішно зібраний кластер

Наступним кроком є Fleet Server, який є частиною Elastic Stack і функціонує як центральний сервер для управління і моніторингу агентів Elastic, які встановлені на віддалених машинах.

Він дозволяє централізовано управляти політиками і налаштуваннями, забезпечуючи збір даних з різних джерел в одному місці. Fleet Server діє як координаційний вузол, що приймає звіти стану від агентів і розсилає оновлення конфігурації.

Integrations у контексті Fleet Server означає підключення і налаштування зовнішніх джерел даних (наприклад, платформи pfsense, що є системою управління мережевим трафіком) для збору логів або інших метрик.

Ці інтеграції дозволяють налаштувати агенти для автоматичного збору і надсилання даних до Elasticsearch, де вони можуть бути аналізовані і візуалізовані за допомогою Kibana.

Fleet Server разом з Integrations використовуються для можливості оптимізації певних процесів, а саме:

- **Централізованого управління:** Керування всіма агентами Elastic з одного місця, включаючи такі процеси як оновлення політик, налаштувань і моніторинг стану.
- **Автоматизація збору даних:** Автоматичний збір логів, визначених характеристик і будь-яких інших даних з інтегрованих систем та пристроїв, мінімізуючи потребу в ручному зборі та аналізі подій.
- **Швидке розгортання:** Оперативне розгортання нових політик і налаштувань до агентів, що дозволяє оперативно реагувати на змінені умови безпеки або операційні потреби.

Наявність декількох нод Fleet Server також має свої переваги:

- **Масштабованість:** зі збільшенням кількості агентів і, відповідно, обсягу даних, однієї ноди може бути недостатньо для ефективного управління всіма агентами. За наявності декількох нод, є можливість розподілити навантаження та зберегти високу продуктивність системи.
- **Відмовостійкість:** у випадку відмови однієї ноди, інші можуть продовжувати виконувати функції управління та збору даних, забезпечуючи безперервність роботи системи.
- **Балансування навантаження:** розподіл навантаження між кількома нодами Fleet Server дозволяє оптимізувати швидкість обробки та зменшити затримку, особливо при високому обсязі збору даних.

Процес створення конфігурація для Fleet Server відбувається через інтерфейс Kibana. В цьому випадку, це зображено на рисунках 3.15 – 3.16 на прикладі першого Fleet Server.

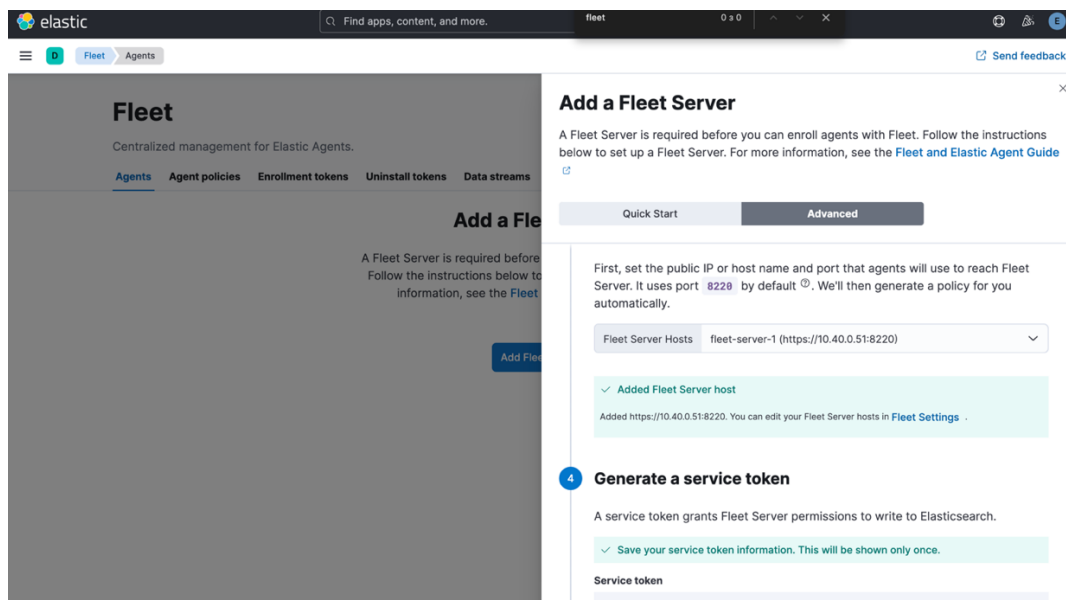


Рисунок 0.15 – Створення конфігурації для першого Fleet Server'a

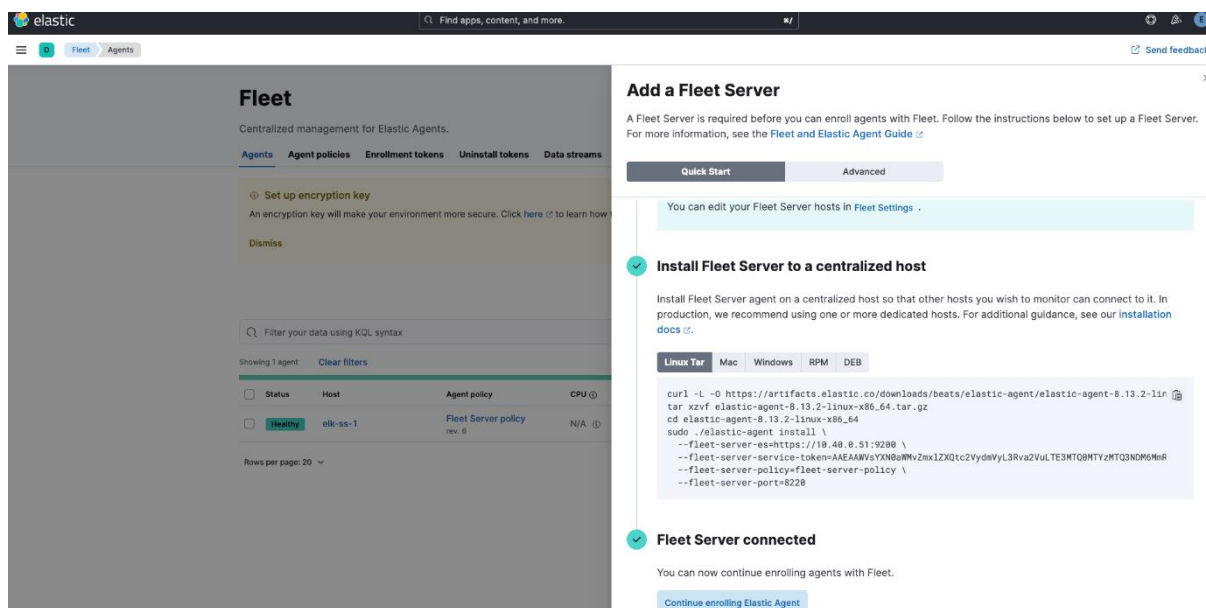


Рисунок 0.16 – Реєстрація першої ноди fleet server'a

В інтерфейсі Kibana можна відслідковувати стан кластеру Fleet server'a. Це зображено на рисунку 3.17.

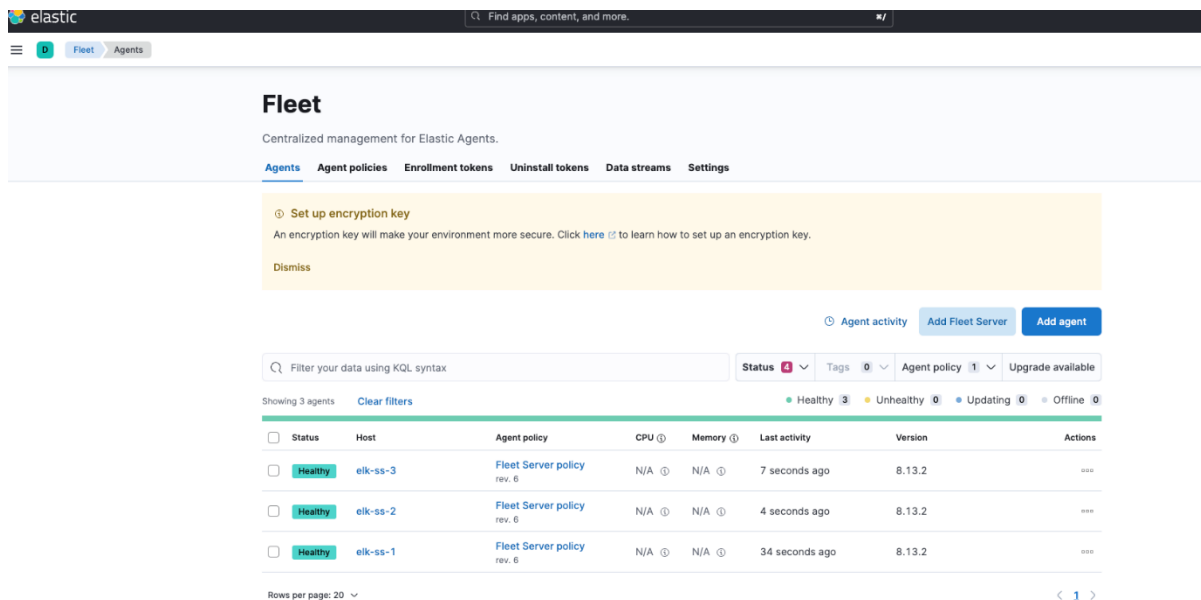


Рисунок 0.17 – Успішно зареєстровані три ноди fleet server’а

Elastic Agent є компонентом Elastic Stack, який використовується для збору подій з різних джерел, управління ними та доставка їх в Elasticsearch. Він надає можливість централізовано управляти збором даних на хостах і може бути віддалено керований через Fleet Server. Elastic Agent може виконувати різні ролі, включаючи моніторинг, логування, забезпечення безпеки, залежно від інтеграцій, які в нього встановлені.

Конфігурація використання Elastic Agent для збору логів auditd з Ubuntu Server.

1. Встановлення та Налаштування. Elastic Agent встановлюється на Ubuntu Server, де функціонує auditd. В рамках налаштування Elastic Agent, використовується інтеграція з auditd, яка спеціально розроблена для збору та передачі логів системного аудиту.

2. Конфігурація. За допомогою Fleet Server, адміністратор системи може дистанційно налаштувати політики інтеграції auditd, які визначають, які дані збиратимуться, як вони оброблятимуться та як часто відправлятимуться до Elasticsearch. Це дозволяє фільтрувати події та зменшити обсяг переданих даних, оптимізуючи використання ресурсів і забезпечуючи збір тільки необхідної інформації.

3. Збір та Відправлення Даних. Після налаштування, Elastic Agent автоматично збирає логи auditd з хоста Ubuntu Server. Це включає всі події, які

відслідковуються auditd, такі як зміни конфігурації системи, доступ до файлів, та інші безпекові події. Зібрані дані відправляються до Elasticsearch, де вони індексуються і стають доступними для аналізу через Kibana.

На рисунку 3.18 зображено, як відбувається створення окремої політики для збору подій з агентів.

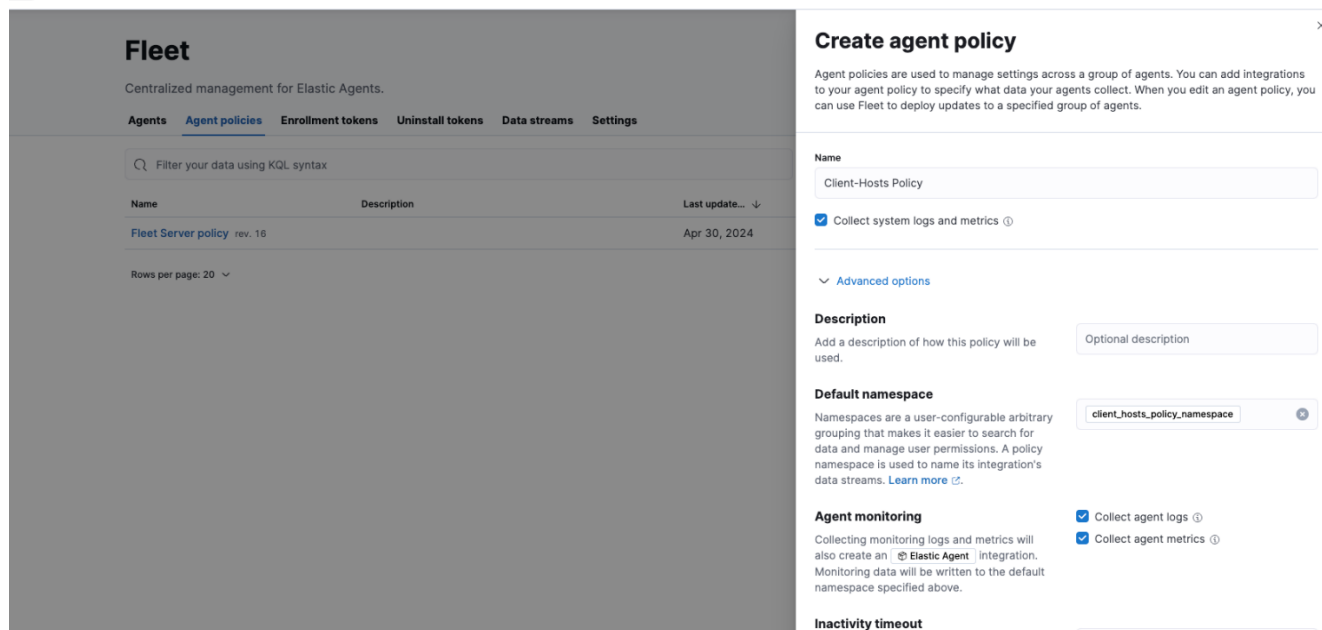


Рисунок 0.18 – Створення окремої політики для агентів, через які будуть збиратись логи auditd

Для того, щоб почати збирати логи необхідно першочергово встановити агент на хост, з якого будуть надсилатись події. Це зображено на рисунку 3.19. Агенти можуть бути використані різним чином: використовуючи політики, ми можемо додати доних будь-які інтеграції, котрі доступні в репозиторії, внаслідок чого, хост може виконувати або ж роль такого собі «проксі», через котрий будуть проходити логи з певного віддаленого джерела або збирати логи безпосередньо з самого хоста (як наприклад syslog або auditd).



Наступним кроком необхідно вказати агенту, які саме дані необхідно збирати, для цього потрібно додати інтеграцію auditd, як зображено на рисунку 3.21.

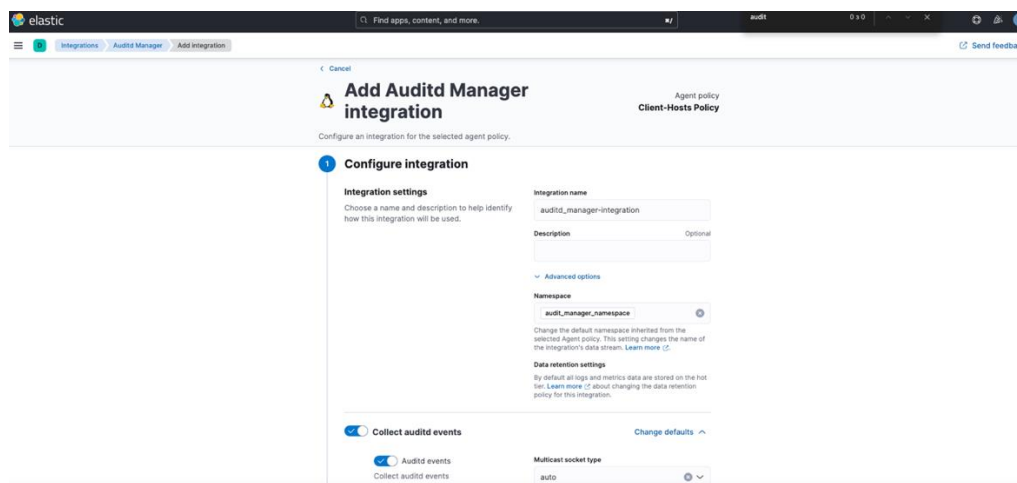


Рисунок 0.21 – Додання інтеграції в політику

Додавання інтеграції pfSense до політики Fleet Server. Це включає наступні кроки:

1. Створення та налаштування політики. Адміністратор системи створює нову політику управління в Fleet Server, призначену для хостів, що використовують pfSense. В рамках цієї політики інтегрується модуль pfSense, який включає налаштування для збору специфічних логів.

2. Додавання хостів до політики. Всі хости, на яких встановлено pfSense, визначаються та додаються під цю політику. Після додавання хостів, на кожному з них автоматично розгортається відповідна конфігурація Elastic Agent, налаштована згідно з інтеграцією pfSense.

3. Автоматизація збору даних. Elastic Agent на кожному хості починає збір логів з pfSense, виконуючи передбачені фільтри і перетворення даних відповідно до політики. Зібрані логи відсилаються до Elasticsearch, де вони агрегуються, аналізуються та стають доступними для моніторингу та візуалізації через Kibana.

В межах цієї роботи, інтеграція додається безпосередньо в політику Fleet Server, таким чином, інджесторами логів виступають самі ноди Fleet Server.

На рисунках 3.22 та 3.23 зображено додання інтеграції до певної політики агентів elasticsearch та можливості гнучкого налаштування прийому логів.

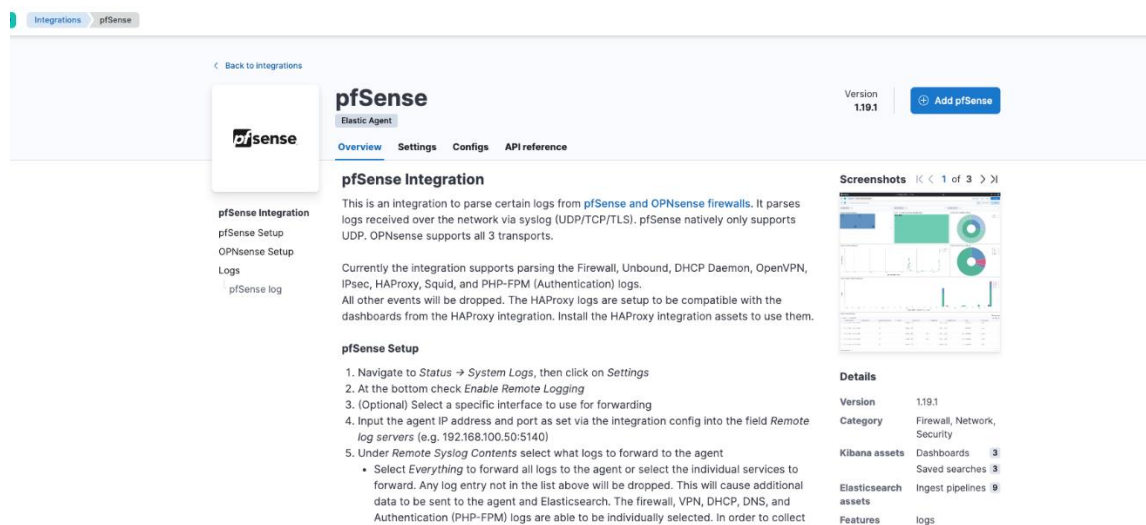


Рисунок 0.22 – Інтеграція pfSense для агенту Elasticsearch

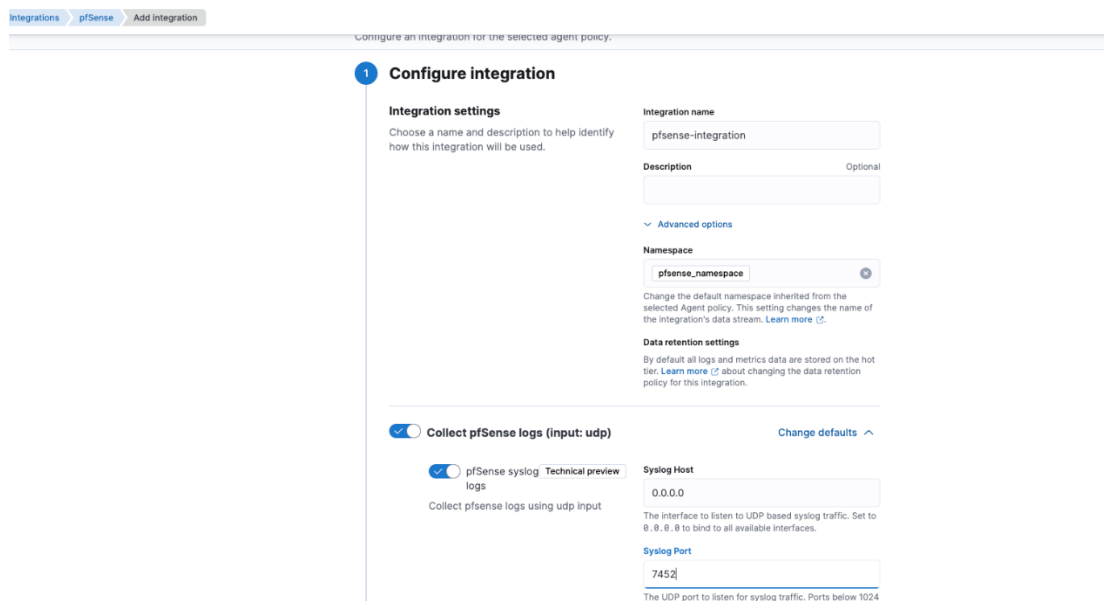


Рисунок 0.23 – Конфігурація інтеграції (індекс, порт, протокол)

### 3.4 Налаштування балансиру навантаження Keep

Балансири навантаження — це елементи мережі, що розподіляють вхідний трафік між кількома серверами або сервісами, розгорнутими у мережі. Їх основне завдання — забезпечити оптимальне розподілення ресурсів, підвищення доступності

та надійності сервісів, а також зменшення часу відгуку. Балансири навантаження можуть бути реалізовані як апаратні (фізичні) так і програмні (віртуальні) рішення.

Балансири навантаження використовують різні алгоритми для розподілу трафіку між серверами:

- **Круговий алгоритм (Round Robin):** Рівномірно розподіляє запити між усіма доступними серверами по черзі.
- **Least Connections:** Направляє нові запити до сервера з найменшою кількістю активних з'єднань.
- **IP Hash:** Вибір сервера базується на хеш-функції IP-адреси клієнта, забезпечуючи, що запити від одного користувача обслуговуються одним і тим же сервером.
- **Weighted Load Balancing:** Серверам присвоюються ваги залежно від їх потужності, і більш потужні сервери отримують більше запитів.

Є декілька причин використання балансирів навантаження:

- **Масштабованість.** Надають можливість легко додавати додаткові сервери без зупинки системи, підвищуючи обробну потужність мережі та можливість обслуговувати більше користувачів.
- **Висока доступність.** Мінімізують ризики відмови окремих серверів, оскільки вхідний трафік автоматично перерозподіляється на решту робочих серверів.
- **Збалансоване навантаження:** Забезпечують рівномірне розподілення навантаження, що запобігає перевантаженню окремих серверів і забезпечує оптимальне використання ресурсів.
- **Зменшення часу відповіді:** Оптимізують час обробки запитів.

На рисунках 3.24 – 3.32 зображено процес конфігурації балансира навантаження, а саме:

- створення віртуального IP з портом, котрий і буде приймати підключення;
- заведення реальних IP адрес, до котрих буде маршрутизуватись трафік;
- вибір налаштувань балансування.

Рисунок 0.24 – Створення віртуального IP для прийняття запитів до Kibana

Рисунок 0.25 – Додання першого серверу з портом 5601

Рисунок 0.26 – Додання другого серверу з портом 5601

Рисунок 0.27 – Додання третього серверу з портом 5601

[-Back](#) [Duplicate Sub](#)

**Basic Properties**

SubVS Name:  [Set Nickname](#)

SubVS Type:  [Set Type](#)

SubVS Weight:  [Set Weight](#)

SubVS Limit:  [Set Limit](#)

SubVS Rate Limit:  [Set Rate Limit](#)

Standard Options

QoS/Limiting

Advanced Properties

WAF

ESP Options

**Real Servers** [Add New](#)

Real Server Check Method:

Interval (sec):  [Use Global: 9](#)

Timeout (sec):  [Use Global: 4](#)

Retry Count:  [Use Global: 2](#)

Checked Port:  [Set Check Port](#)

URL:  [Set URL](#)

Status Codes:  [Set Status Codes](#)

Use HTTP/1.1:

HTTP Method:  [Set Method](#)

Custom Headers: [Show Headers](#)

Enhanced Options:

d	IP Address	Port	Forwarding method	Weight	Limit	Rate Limit	Status	Operation
	10.40.0.51	5601	nat	1000	0	0	Enabled	<a href="#">Disable</a> <a href="#">Modify</a> <a href="#">Delete</a>
	10.40.0.52	80	nat	1000	0	0	Enabled	<a href="#">Disable</a> <a href="#">Modify</a> <a href="#">Delete</a>
	10.40.0.52	5601	nat	1000	0	0	Enabled	<a href="#">Disable</a> <a href="#">Modify</a> <a href="#">Delete</a>

Рисунок 0.28 – Загальна конфігурація балансиру Kibana

**LoadMaster**

Add a new Virtual Service

Please Specify the Parameters for the Virtual Service.

Virtual Address:  [Refresh](#)

Port:

Service Name (Optional):

Protocol:

Рисунок 0.29 – Створення віртуального IP для прийому та балансування логів з pfSense

**LoadMaster**  
Add a Real Server to udp/10.40.0.71:7452 (Id:3)

---

Please Specify the Parameters for the Real Server

---

Real Server Address   
 Port   
 Forwarding method   
 Weight

---

The following Real Servers are already configured

The Real Server can be on one of the following networks  
10.40.0.0/24

Рисунок 0.30 – Додання першого серверу з портом 7452

**LoadMaster**  
Add a Real Server to udp/10.40.0.71:7452 (Id:3)

---

Please Specify the Parameters for the Real Server

---

Real Server Address   
 Port   
 Forwarding method   
 Weight

---

The following Real Servers are already configured  
10.40.0.51:7452

The Real Server can be on one of the following networks  
10.40.0.0/24

Рисунок 0.31 – Додання другого серверу з портом 7452

**LoadMaster**  
Add a Real Server to udp/10.40.0.71:7452 (Id:3)

---

Please Specify the Parameters for the Real Server

---

Real Server Address   
 Port   
 Forwarding method   
 Weight

---

The following Real Servers are already configured  
10.40.0.51:7452  
10.40.0.52:7452

The Real Server can be on one of the following networks  
10.40.0.0/24

Рисунок 0.32 – Додання третього серверу з портом 7452

### 3.5 Налаштування віртуального фаєрволу pfsense

pfSense є програмним забезпеченням з відкритим вихідним кодом, яке функціонує як мережевий фаєрвол і маршрутизатор. Він розроблений на основі потужної операційної системи FreeBSD.

Таким чином, pfSense забезпечує широкий спектр можливостей для забезпечення безпеки, включаючи функції маршрутизації, фільтрації вмісту та багато іншого. Він підходить для використання як в малих офісних мережах, так і в великих корпоративних середовищах.

Вбудовані інструменти для моніторингу і звітності надають можливість проводити аналіз мережевого трафіку і виявляти потенційні проблеми безпеки або збої в роботі мережі.

За замовчуванням логування на фаєрволі відімкнене, тож його необхідно увімкнути.

В свою чергу, як і більшість подібних програмних рішень, pfSense дозволяє використовувати віддалений сервер логування.

На рисунку 3.33 вказано IP та порт необхідного сервісу на балансірі навантаження (10.40.0.71:7452 UDP).

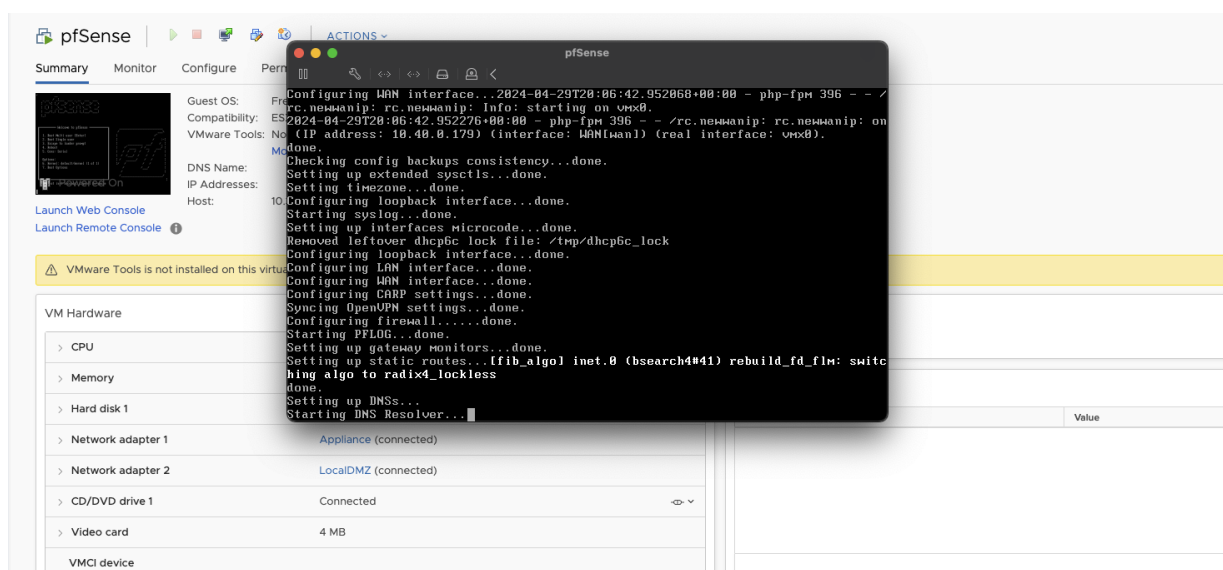
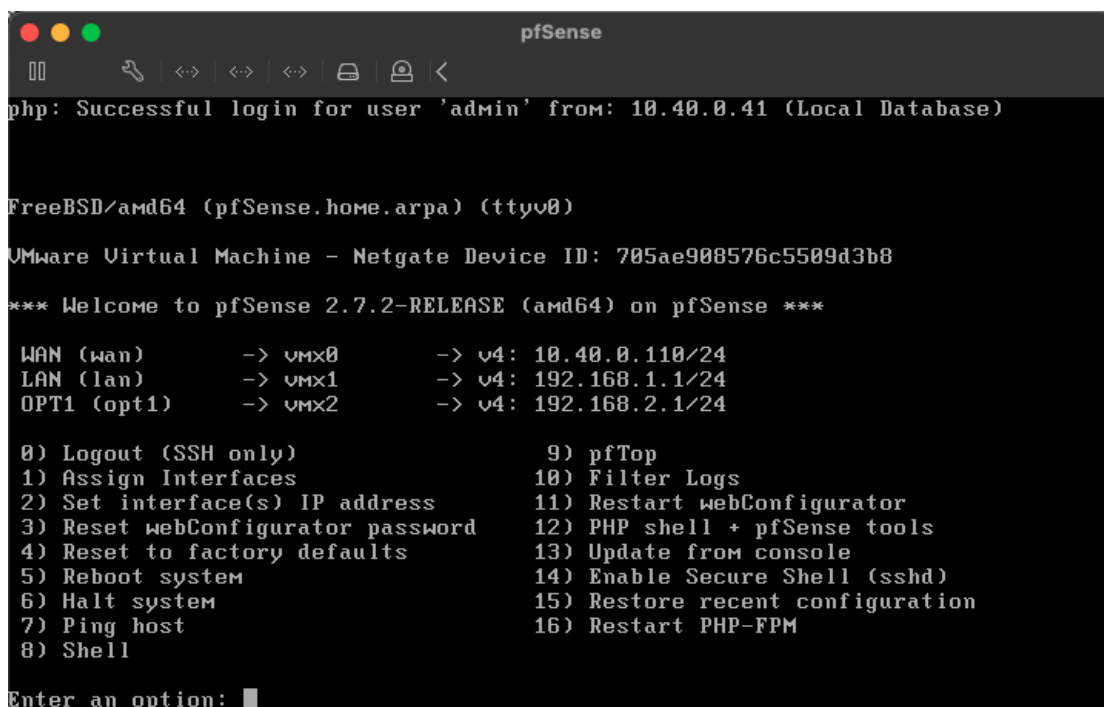


Рисунок 0.33 – Процес встановлення віртуального фаєрволу

На рисунках 3.34 – 3.38 зображений процес налаштування логуювання віртуального фаєрволу.



```

php: Successful login for user 'admin' from: 10.40.0.41 (Local Database)

FreeBSD/amd64 (pfSense.home.arpa) (ttyv0)
VMware Virtual Machine - Netgate Device ID: 705ae908576c5509d3b8

*** Welcome to pfSense 2.7.2-RELEASE (amd64) on pfSense ***

WAN (wan)      -> vmx0      -> v4: 10.40.0.110/24
LAN (lan)      -> vmx1      -> v4: 192.168.1.1/24
OPT1 (opt1)   -> vmx2      -> v4: 192.168.2.1/24

0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults    13) Update from console
5) Reboot system              14) Enable Secure Shell (sshd)
6) Halt system                15) Restore recent configuration
7) Ping host                  16) Restart PHP-FPM
8) Shell

Enter an option:

```

Рисунок 0.34 – Сконфігуровані інтерфейси у віртуального фаєрволу

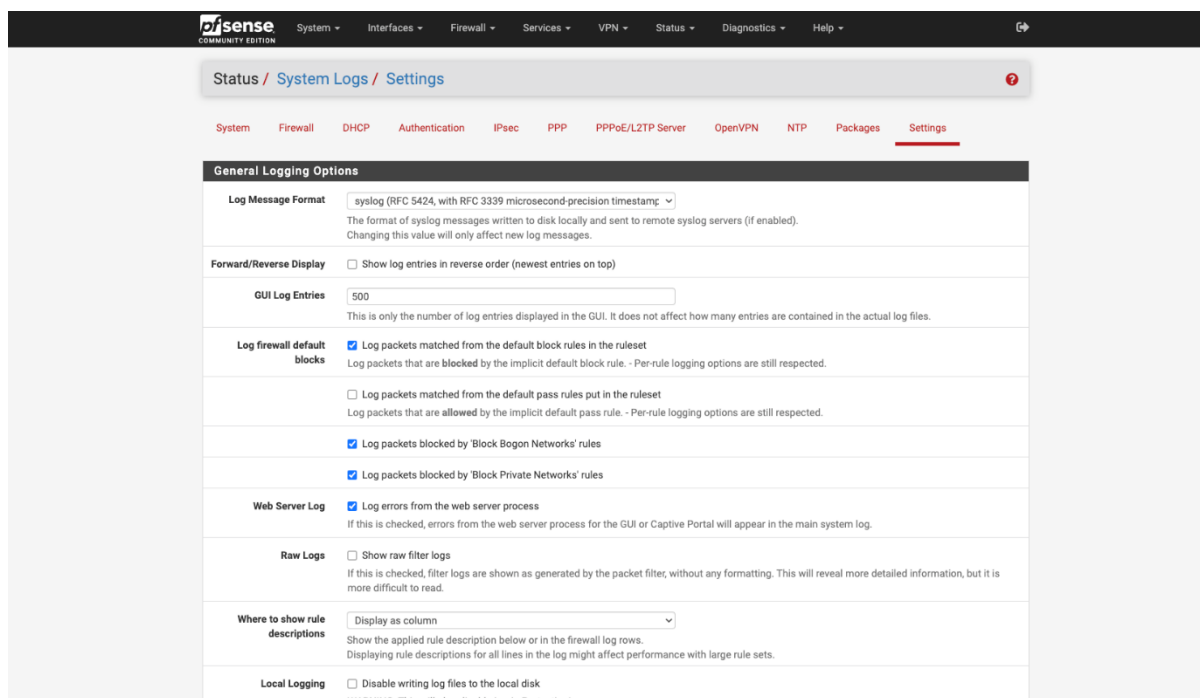


Рисунок 0.35 – Конфігурація логуювання на pfSense

**Log Retention Count**   
The number of log files to keep before the oldest copy is removed on rotation.

**Remote Logging Options**

**Enable Remote Logging**  Send log messages to remote syslog server

**Source Address**   
This option will allow the logging daemon to bind to a single IP address, rather than all IP addresses. If a single IP is picked, remote syslog servers must all be of that IP type. To mix IPv4 and IPv6 remote syslog servers, bind to all interfaces.  
NOTE: If an IP address cannot be located on the chosen interface, the daemon will bind to all addresses.

**IP Protocol**   
This option is only used when a non-default address is chosen as the source above. This option only expresses a preference; if an IP address of the selected type is not found on the chosen interface, the other type will be tried.

**Remote log servers**

**Remote Syslog Contents**  Everything

- System Events
- Firewall Events
- DNS Events (Resolver/unbound, Forwarder/dnsmasq, filterdns)
- DHCP Events (DHCP Daemon, DHCP Relay, DHCP Client)
- PPP Events (PPPoE WAN Client, L2TP WAN Client, PPTP WAN Client)
- General Authentication Events
- Captive Portal Events
- VPN Events (IPsec, OpenVPN, L2TP, PPPoE Server)
- Gateway Monitor Events
- Routing Daemon Events (RADVD, UPnP, RIP, OSPF, BGP)
- Network Time Protocol Events (NTP Daemon, NTP Client)
- Wireless Events (hostapd)

Syslog sends UDP datagrams to port 514 on the specified remote syslog server, unless another port is specified. Be sure to set syslogd on the remote server to accept syslog messages from pfSense.

Рисунок 0.36 – Задання віддаленого серверу логування (попередньо налаштованого балансиру навантаження)

**Action**   
Choose what to do with packets that match the criteria specified below.  
Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded.

**Disabled**  Disable this rule  
Set this option to disable this rule without removing it from the list.

**Interface**   
Choose the interface from which packets must come to match this rule.

**Address Family**   
Select the Internet Protocol version this rule applies to.

**Protocol**   
Choose which IP protocol this rule should match.

**Source**

**Source**  Invert match   /

**Destination**

**Destination**  Invert match   /

**Extra Options**

**Log**  Log packets that are handled by this rule  
Hint: the firewall has limited local log space. Don't turn on logging for everything. If doing a lot of logging, consider using a remote syslog server (see the Status: System Logs: Settings page).

**Description**   
A description may be entered here for administrative reference. A maximum of 52 characters will be used in the ruleset and displayed in the firewall log.

**Advanced Options**

Рисунок 0.37 – Увімкнення логування на інтерфейсах

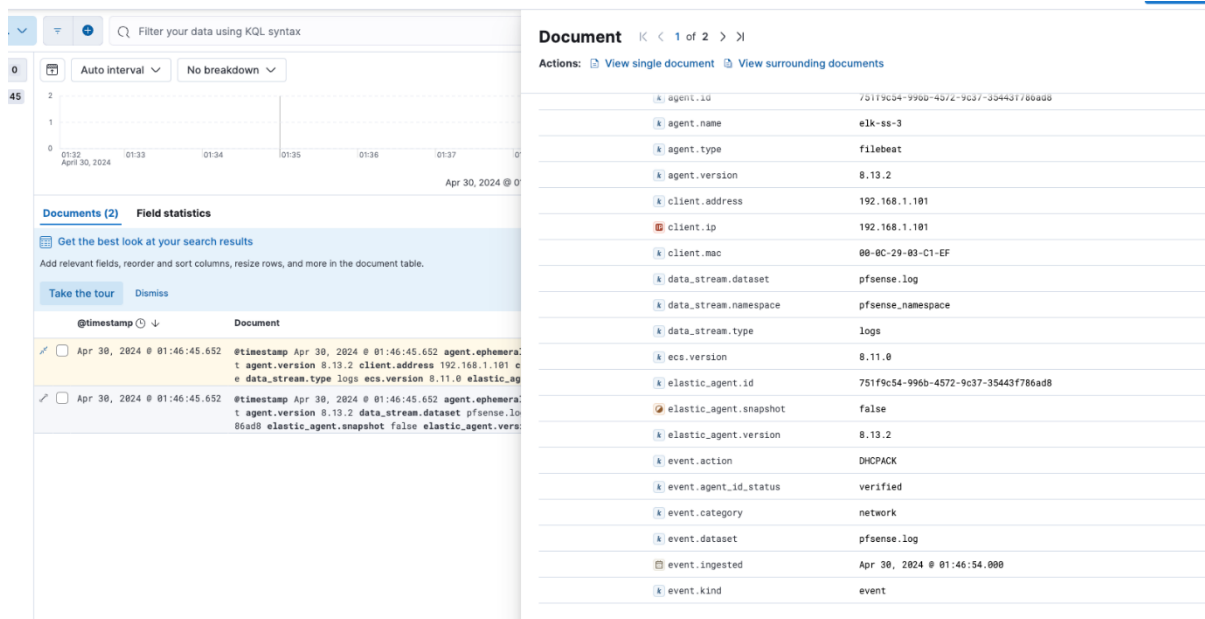


Рисунок 0.38 – Успішне отримання логів в Elasticsearch через балансир навантаження та відображення їх через Kibana

### 3.6 Створення програмного модуля для інтеграції Elasticsearch та ChatGPT

Для можливості проведення аналізу логів на предмет аномалій в них необхідно реалізувати експорт даних логів та доставку на обробку в модель. Оскільки Elasticsearch використовує REST API, було створено POST запит на генерацію репорту, що включатиме логи з індексу pfSense'a за останні 15 хвилин. Цей процес зображено на рисунках 3.39 – 3.40.

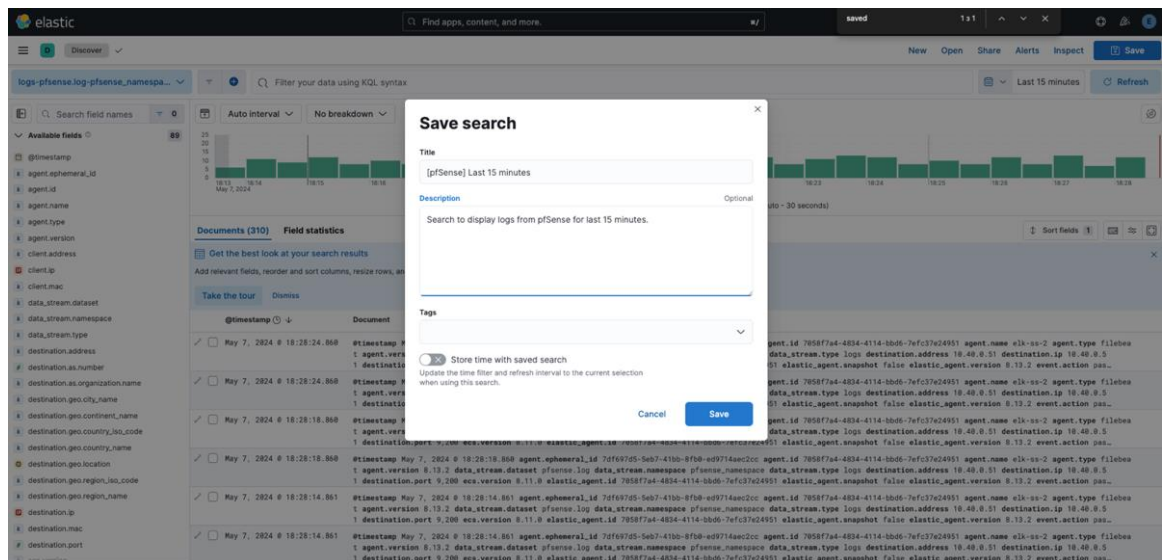


Рисунок 0.39 – Збереження пошукового запиту, що відображає логи з індексу pfSense'a за останні 15 хвилин

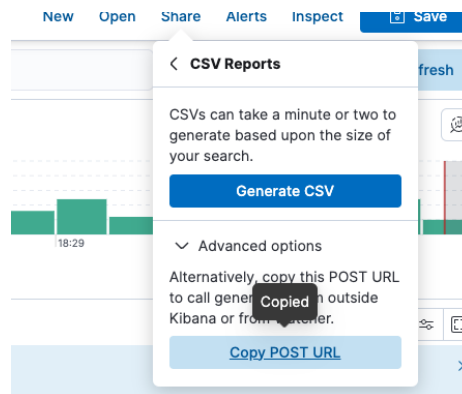


Рисунок 0.40 – Створення посилання для реалізації POST запиту для генерації звіту

Для автоматизації даного процесу було створено функцію:

```
def download_kibana_report(url, auth, headers, directory="logs", polling_interval=1):
    os.makedirs(directory, exist_ok=True)
    timestamp = datetime.now().strftime("%Y-%m-%d_%H-%M-%S")
    file_path = os.path.join(directory, f"{timestamp}.csv")
    initial_response = requests.post(url, headers=headers, auth=auth)
    if initial_response.status_code == 200:
        response_json = initial_response.json()
        download_path = response_json.get('path')
```

```
if download_path:
```

```
    download_url = f"http://10.40.0.70{download_path}"
```

Згенеровано логи з активністю мережевого сканування шляхом запуску nmap з хоста Kali на хост UbuntuServer. Це зображено на рисунку 3.41.

```
(kali@kali)-[~]
└─$ nmap -sV -sC 192.168.2.100 -p0-65000
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-04 10:42 EDT
Nmap scan report for 192.168.2.100
Host is up (0.00014s latency).
Not shown: 6499 closed tcp ports (conn-refused)
PORT      STATE      SERVICE VERSION
0/tcp    filtered  unknown
22/tcp   open      ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|_ 256 cc:7d:5f:a8:3f:8f:06:ac:38:25:92:fb:ee:e1:0a:0c (ECDSA)
|_ 256 80:27:62:17:be:75:a8:c6:16:66:77:c2:fb:e5:7d:e4 (ED25519)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.81 seconds

(kali@kali)-[~]
└─$ nmap -sV -sC 192.168.2.100 -p0-65000
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-04 10:42 EDT
Stats: 0:01:25 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 74.38% done; ETC: 10:44 (0:00:29 remaining)
Stats: 0:01:29 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 74.42% done; ETC: 10:44 (0:00:31 remaining)
Stats: 0:02:25 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 74.95% done; ETC: 10:45 (0:00:48 remaining)
Stats: 0:02:31 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 75.01% done; ETC: 10:45 (0:00:50 remaining)
Stats: 0:04:32 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 76.18% done; ETC: 10:48 (0:01:25 remaining)
```

Рисунок 0.41 – Запуск сканування UbuntuServer з хоста Kali

На рисунку 3.42 зображено, що події сканування відображаються в Kibana.

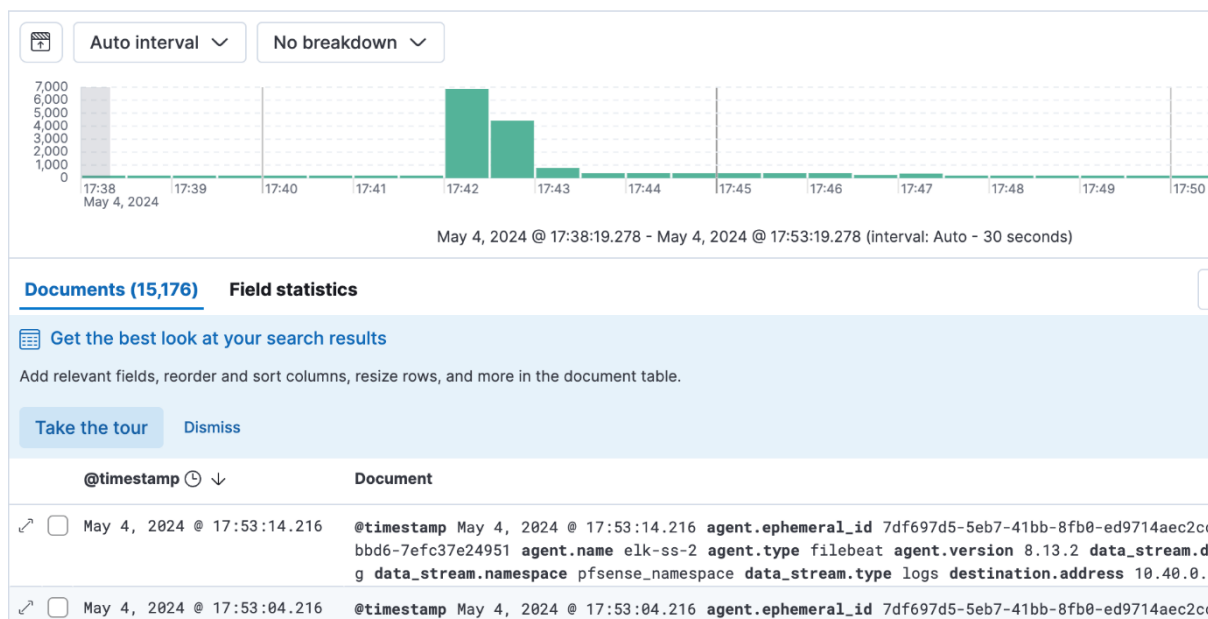


Рисунок 0.42 – Логи сканування, відображені в Kibana

Після експорту логів, проведено відпрацювання попередньо навченого чату в ChatGPT на предмет виявлення аномальної активності в логах (у експортованому лог-файлі). Цей процес зображено на рисунку 3.43.

### Anomaly Detection Suggestions

1. **Investigate Frequent IP Addresses:** Determine why `192.168.1.102` is the source of so many logs and why `192.168.2.100` is a common destination. Check if these behaviors align with expected network traffic and usage patterns.
2. **Review Rare Rule Triggers:** Investigate the context in which the rarely triggered rules were activated to understand if they represent security threats, system errors, or unusual user activities.
3. **Monitor Network Traffic:** Given the dominance of certain IPs and rules, continuous monitoring of these components may provide further insights into network health and security.

These findings should guide further detailed investigations to ascertain the nature of the activities and whether they pose any security concerns. If you have specific aspects of the log you want to explore further or need further analysis, please let me know! [↔]

Рисунок 0.43 – ChatGPT успішно виявив аномальну активність між двома хостами

Для автоматизації даного процесу було створено Assistant, як зображено на рисунку 3.44. Асистент “Linux Anomaly Detector” з попередньо заданим контекстом

для виявлення аномальної активності в рамках логів auditd. На рисунку 3.45 зображено знову згенерована активність користувача.

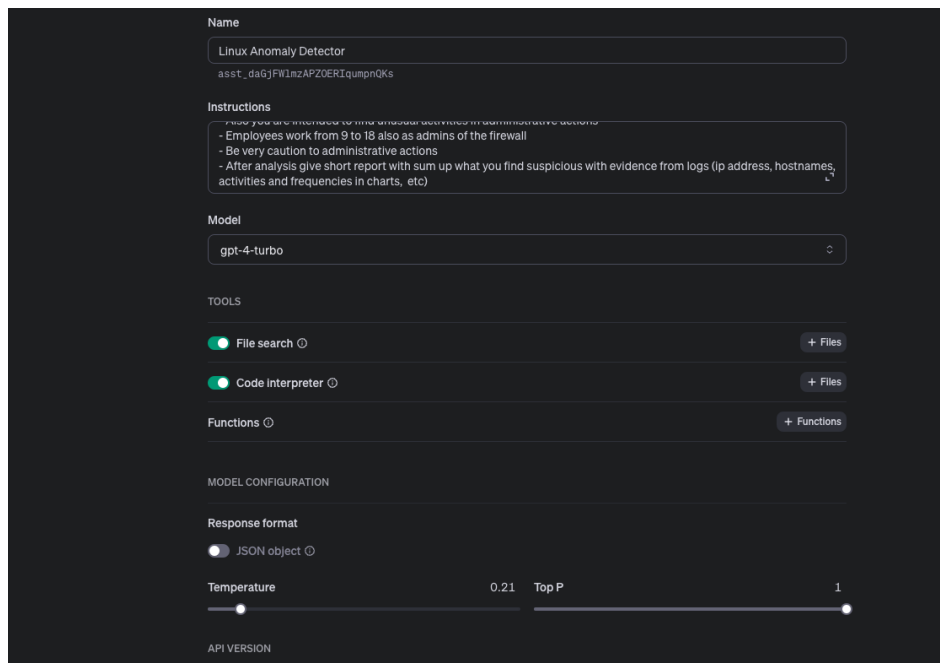


Рисунок 0.44 – Попереднє налаштування асистента



Рисунок 0.45 – Згенерована незвична активність від користувача

На рисунку 3.45 зображено логи в Kibana, на яких присутня активність користувача, серед якої присутні невдалі спроби ескалації привілеїв через sudo, підключення по ssh.

Для експорту логів до створеного асистента розроблено функцію:

```
def send_log_to_chatgpt(api_key, assistant_id):
list_of_files = glob.glob('logs/*')
latest_file = max(list_of_files, key=os.path.getctime)
with open(latest_file, 'r') as file:
    log_content = file.read()
gen_client = openai.OpenAI(api_key=api_key)
gen_chat = gen_client.beta.threads.create(
    messages=[
        {
            "role": "user",
            "content": log_content
        }
    ]
)

query = gen_client.beta.threads.runs.create(thread_id=gen_chat.id, assistant_id=assistant_id)
```

На рисунку 3.46 відображено повністю автоматизований процес експорту логів з Elasticsearch, відправки запиту з даними логами до асистенту ChatGPT та отримання результату про наявність в логах аномалій.

```
Report saved as logs/2024-05-08_01-15-58.csv
Query completed
The data displays several activities at unusual times, specifically between 01:12 and 01:13 AM, which is outside regular work hours (9:00 AM to 6:00 PM). Here are detailed suspicious activities:

1. **Multiple Failed Authentication Attempts:**
- UID 1000 (John) made repeated failed attempts to execute 'sudo', associated with unsuccessful PAM authentication and using the 'su' command. These occurred between 01:12:43.963 and 01:13:08.443.
- Outcome consistently marks "fail".
- Occurred on terminal pts/0 or /dev/pts/0, via 'usr/bin/sudo'.
- Multiple similar failed authentication messages (e.g., 'USER_AUTH ... res=failed').

2. **Login Activity from Unknown Source Addresses:**
- SSH login at 01:12:44.011 with a successful outcome, but the hostname records an unusual IP '192.168.1.102', marked for multiple activities.
- 'op=login' and 'op=CRED_ACQ' indicate successful credential acquisitions and sessions starting, raising concerns about the legitimacy given the time and repeating IP address.

Given the timestamps and the patterns suggesting multiple failed attempts followed by a successful login, these should be reviewed further for potential security threats or policy violations. The involvement of admin level commands (sudo, su) in unintended hours, and from specific IP addresses, further elevates the concern.
```

Рисунок 0.46 – Успішне виявлення аномалії за допомогою асистента

### Висновки до розділу 3

У третьому розділі кваліфікаційної роботи було реалізовано процес побудови та налаштування комплексної моделі для виявлення аномалій у кібербезпеці, розгорнутого у домашній інфраструктурі. Ця модель передбачає комплексний підхід до ідентифікації та аналізу потенційних загроз у ІС, який включає не лише застосування МН та ШІ, а й весь життєвий цикл даних: від генерація подій на кінцевих точках до отримання результатів про наявність аномалії в логах оброблених за допомогою chatGPT.

Перш за все, була розроблена топологія мережі, яка ефективно інтегрує всі компоненти системи, забезпечуючи надійність та високу доступність сервісів. Основою системи є кластер Elasticsearch, який дозволяє ефективно обробляти та аналізувати великі обсяги даних. Налаштування балансиру навантаження K8s забезпечує рівномірне розподілення навантаження між вузлами кластера, підвищуючи тим самим ефективність обробки запитів.

Додатково було налаштовано віртуальний фаєрвол pfSense, що забезпечує захист всієї мережевої інфраструктури від зовнішніх та внутрішніх загроз. Також, було налаштовано доставку логів з віртуального фаєрволу pfSense в Elasticsearch, імітуючи таким чином реальну інфраструктуру.

Особливу увагу було приділено розробці програмного модуля, який інтегрує Elasticsearch зі штучним інтелектом для автоматичного виявлення аномалій у логах. Цей модуль автоматично вивантажує логи, аналізує їх та надає звіт про наявність потенційних аномалій.

Загалом, побудована система за запропонованою моделлю демонструє високу ефективність у виявленні та реагуванні на аномалії в сфері кібербезпеки, що робить її важливим інструментом для забезпечення безпеки інформаційних систем будь-яких компаній.

## ВИСНОВКИ

У даній кваліфікаційній дипломній роботі було проведено аналіз предметної області поняття аномалій, а також проблематики виявлення аномалій у сфері кібербезпеки, що є ключовим аспектом для забезпечення надійності та безпеки інформаційних систем.

В першому розділі дослідження акцентувалася увага на класифікації та основних методах виявлення аномалій, включаючи інтеграцію сучасних технологій машинного навчання та штучного інтелекту. Особлива увага приділена використанню концепцій, таких як IDS, SIEM, XDR, та DLP, для покращення точності та оперативності виявлення аномалій та потенційних загроз.

У другому розділі описано труднощі, пов'язані зі зростаючою складністю інформаційних систем та посиленням кіберзагроз, що вимагає вдосконалення методів виявлення аномалій. Було розглянуто різноманітність методів від статистичного аналізу до нейромережових підходів, підкреслюючи потребу у комплексному рішенні, яке інтегрує різні технології для ефективного виявлення та реагування на інциденти кібербезпеки. Також було розроблено таблицю відповідності типу аномалії до методу виявлення.

Третій розділ підтвердив практичну значущість теоретичних розробок шляхом реалізації комплексної моделі для виявлення аномалій. Продемонстровано, як така модель може бути успішно розгорнута в домашній інфраструктурі, забезпечуючи високий рівень захисту за допомогою автоматизації процесів аналізу та виявлення аномалій.

Високий потенціал інтеграції штучного інтелекту, машинного навчання та інших сучасних технологій у процеси виявлення та реагування на кіберзагрози робить цю кваліфікаційну роботу актуальним внеском у розвиток стратегій кібербезпеки. Інноваційний підхід до аналізу та моніторингу аномалій, представлений у даній роботі, демонструє, як технічні рішення можуть підвищити здатність організацій протистояти новим та еволюційним кіберзагрозам, забезпечуючи стійкість

інформаційних систем. Завдяки своєчасному виявленню та реагуванню на аномалії, можливо мінімізувати збитки та підвищити ефективність захисних механізмів, що є вирішальним для сучасного цифрового суспільства.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Claude Fachkha, "Security Monitoring of the Cyber Space", 2016 <https://arxiv.org/pdf/1608.01468v1.pdf>
2. Igor Linkov and Alexander Kott, "Fundamental Concepts of Cyber Resilience: Introduction and Overview", 2018 <https://arxiv.org/pdf/1806.02852v1.pdf>
3. Ugur Halden, Umit Cali, Ferhat Ozgur Catak, Salvatore D'Arco, Francisco Bilendo, "Anomaly Detection in Power Markets and Systems", 2022 <https://arxiv.org/pdf/2212.02182v1.pdf>
4. Chris van der Ploeg, Robin Smit, Alexis Siagkris-Lekkos, Frank Benders, Emilia Silvas, "Anomaly Detection from Cyber Threats via Infrastructure to Automated Vehicle", 2021 <https://arxiv.org/pdf/2104.11790v1.pdf>
5. H. Almazarqi, "Profiling IoT botnet activity," Ph.D. dissertation, University of Glasgow, 2024. <https://theses.gla.ac.uk/84102/>
6. Z.A. Ruttle et al., "Simulation and Analysis of Cyber Attacks on Power and Energy Systems," in Proceedings of the 2024 IEEE Power & Energy Society General Meeting.  
<https://ieeexplore.ieee.org/abstract/document/10454166/>
7. L.L. Scientific, "SECURE AND ISOLATED COMPUTING IN VIRTUALIZATION AND CLOUD ENVIRONMENTS: A SYSTEMATIC REVIEW OF EMERGING TRENDS AND CHALLENGES," Journal of Theoretical and Applied Information Technology, vol. 102, no. 3, 2024.  
<https://www.jatit.org/volumes/Vol102No3/4Vol102No3.pdf>
8. S. Benter et al., "A Review of MAC Spoofing Detection Methods in Wired Networks," SSRN Electronic Journal, 2024.  
[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=4730223](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4730223)
9. M. Alkathami, "Anomaly detection in IoT-based healthcare: machine learning for enhanced security," *Scientific Reports*, 2024. <https://www.nature.com/articles/s41598-024-56126-x>

10. Y. Peng et al., "Improving the robustness of steganalysis in the adversarial environment with Generative Adversarial Network," *Journal of Information Security and Applications*, 2024.  
<https://www.sciencedirect.com/science/article/abs/pii/S2214212624000462>
11. F. Louati et al., "Big-IDS: a decentralized multi-agent reinforcement learning approach for distributed intrusion detection in big data networks," *Cluster Computing*, 2024.  
<https://link.springer.com/article/10.1007/s10586-024-04306-9>
12. T. Yi et al., "An Anomaly Behavior Characterization Method of Network Traffic Based on Spatial Pyramid Pool (SPP)," *Computers & Security*, 2024.  
<https://www.sciencedirect.com/science/article/pii/S016740482400110X>
13. N.G. Camacho, "The Role of AI in Cybersecurity: Addressing Threats in the Digital Age," *Journal of Artificial Intelligence General science*, 2024.  
<https://ojs.boulibrary.com/index.php/JAIGS/article/view/75>
14. A. Coscia et al., "Automatic decision tree-based NIDPS ruleset generation for DoS/DDoS attacks," *Journal of Information Security and Applications*, 2024.  
<https://www.sciencedirect.com/science/article/pii/S2214212624000395>
15. S. Kumar, E.H. Spafford, An application of pattern matching in intrusion detection, The COAST Project, Department of Computer Sciences, Purdue University, West Lafayette, IN, USA, Technical Report CSD-TR-94- 013, June 17, 1994.
16. A Sense of Self for Unix Processes [Электронный ресурс]. – 2000. – Режим доступа до ресурсу:  
[https://www.researchgate.net/publication/2358308\\_A\\_Sense\\_of\\_Self\\_for\\_Unix\\_Processes](https://www.researchgate.net/publication/2358308_A_Sense_of_Self_for_Unix_Processes)
17. Anomaly Detection Using System Logs:A Deep Learning Approach [Электронный ресурс]. – 2012. – Режим доступа до ресурсу:  
[https://www.researchgate.net/publication/356252048\\_Anomaly\\_Detection\\_Using\\_System\\_Logs\\_A\\_Deep\\_Learning\\_Approach](https://www.researchgate.net/publication/356252048_Anomaly_Detection_Using_System_Logs_A_Deep_Learning_Approach)
18. LSTM-based Anomaly Detection for Non-linear Dynamical System [Электронный ресурс]. – 2020. – Режим доступа до ресурсу:  
[https://www.researchgate.net/publication/341998291\\_LSTM-based\\_Anomaly\\_Detection\\_for\\_Non-linear\\_Dynamical\\_System](https://www.researchgate.net/publication/341998291_LSTM-based_Anomaly_Detection_for_Non-linear_Dynamical_System)

19. Anomaly Detection in Automobile Control Network Data with Long Short-Term Memory Networks [Электронный ресурс]. – 2016. – Режим доступа до ресурсу: [https://cradpdf.drdc-rddc.gc.ca/PDFS/unc246/p804546\\_A1b.pdf](https://cradpdf.drdc-rddc.gc.ca/PDFS/unc246/p804546_A1b.pdf)
20. Anomaly Detection: A Survey [Электронный ресурс]. – 2009. – Режим доступа до ресурсу: [https://www.researchgate.net/publication/220565847\\_Anomaly\\_Detection\\_A\\_Survey](https://www.researchgate.net/publication/220565847_Anomaly_Detection_A_Survey)
21. Network Intrusion Detection and Prevention [Электронный ресурс]. – 2009. – Режим доступа до ресурсу: <https://download.e-bookshelf.de/download/0000/0028/43/L-G-0000002843-0002340348.pdf>
22. Anomaly detection in network traffic using machine learning techniques: A comprehensive review [Электронный ресурс]. – 2018. – Режим доступа до ресурсу: <https://www.turcomat.org/index.php/turkbilmat/article/download/13626/9773>
23. Anomaly-based network intrusion detection: Techniques, systems and challenges [Электронный ресурс]. – 2009. – Режим доступа до ресурсу: <https://www.sciencedirect.com/science/article/abs/pii/S0167404808000692>
24. Anomaly Detection with Variational Autoencoders (VAE): Unveiling Hidden Patterns [Электронный ресурс]. – 2022. – Режим доступа до ресурсу: <https://medium.com/@prajeeshprathap/anomaly-detection-with-variational-autoencoders-vae-unveiling-hidden-patterns-42631834ffbf>
25. Anomaly intrusion detection based on genetic algorithm and statistical analysis for web servers [Электронный ресурс]. – 2012. – Режим доступа до ресурсу: <https://www.scribd.com/document/281155096/Thesis-Network-Anomaly-Detection-With-Incomplete-Audit-Data-2006>
26. Support vector machine based anomaly detection for access control system [Электронный ресурс]. – 2017. – Режим доступа до ресурсу: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8963404/>
27. Interpreting Unsupervised Anomaly Detection in Security via Rule Extraction [Электронный ресурс]. – 2019. – Режим доступа до ресурсу: [https://www.researchgate.net/publication/377635863\\_Interpreting\\_Unsupervised\\_Anomaly\\_Detection\\_in\\_Security\\_via\\_Rule\\_Extraction](https://www.researchgate.net/publication/377635863_Interpreting_Unsupervised_Anomaly_Detection_in_Security_via_Rule_Extraction)

28. Anomaly Detection of System Architecture Based on Decision Tree [Электронный ресурс]. – 2011. – Режим доступа до ресурсу: <https://arxiv.org/pdf/2212.12092>
29. Anomaly detection in system logs using Bayesian networks [Электронный ресурс]. – 2012. – Режим доступа до ресурсу: [https://www.researchgate.net/publication/341713286\\_Anomaly\\_Detection\\_in\\_Smart\\_Homes\\_Using\\_Bayesian\\_Networks](https://www.researchgate.net/publication/341713286_Anomaly_Detection_in_Smart_Homes_Using_Bayesian_Networks)
30. Anomaly Detection in Authentication System Using Clustering Analysis [Электронный ресурс]. – 2018. – Режим доступа до ресурсу: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9439459>
31. Deep Semisupervised Learning-Based Network Anomaly Detection in Heterogeneous Information Systems [Electronic resource] - 2022 - Access mode: [https://cdn.techscience.cn/ueditor/files/cmc/TSP\\_CMC\\_70-1/TSP\\_CMC\\_18773/TSP\\_CMC\\_18773.pdf](https://cdn.techscience.cn/ueditor/files/cmc/TSP_CMC_70-1/TSP_CMC_18773/TSP_CMC_18773.pdf)
32. Anomaly detection using a clustering technique [Electronic resource] - 2012 - Access mode: [https://www.researchgate.net/profile/Jayant-Gadge/publication/258650166\\_Anomaly\\_Detection\\_using\\_a\\_Clustering\\_Technique/links/569cebcc08ae5c9fe6c0e91f/Anomaly-Detection-using-a-Clustering-Technique.pdf](https://www.researchgate.net/profile/Jayant-Gadge/publication/258650166_Anomaly_Detection_using_a_Clustering_Technique/links/569cebcc08ae5c9fe6c0e91f/Anomaly-Detection-using-a-Clustering-Technique.pdf)
33. A distributed approach to anomaly detection [Electronic resource] - 1997 - Access mode: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=2c8cc46892e0c9c6bc735afa0e0836a32d7f25fc>
34. Anomaly detection using data mining methods in it systems: a decision support application [Electronic resource] - 2018 - Access mode: <https://dergipark.org.tr/en/pub/saufenbilder/article/365931>
35. An Information System Supporting Insurance Use Cases by Automated Anomaly Detection [Electronic resource] - 2022 - Access mode: <https://www.mdpi.com/2504-2289/7/1/4>

36. A data analytics-based energy information system (eis) tool to perform meter-level anomaly detection and diagnosis in buildings [Electronic resource] - 2021 - Access mode: <https://www.mdpi.com/1996-1073/14/1/237>
37. Timing behavior anomaly detection in enterprise information systems [Electronic resource] - 2007 - Access mode: <https://oceanrep.geomar.de/id/eprint/14519/>
38. An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems [Electronic resource] - 2001 - Access mode: <https://onlinelibrary.wiley.com/doi/abs/10.1002/qre.392>
39. A survey of network anomaly detection techniques [Electronic resource] - 2016 - Access mode: <https://www.sciencedirect.com/science/article/pii/S1084804515002891>
40. Anomaly detection in complex real world application systems [Electronic resource] - 2017 - Access mode: <https://ieeexplore.ieee.org/abstract/document/8101009>
41. Improving one-class SVM for anomaly detection [Electronic resource] - 2003 - Access mode: <https://ieeexplore.ieee.org/abstract/document/1260106>
42. Anomaly detection using process mining [Electronic resource] - 2009 - Access mode: [https://link.springer.com/chapter/10.1007/978-3-642-01862-6\\_13](https://link.springer.com/chapter/10.1007/978-3-642-01862-6_13)
43. Algorithms for anomaly detection of traces in logs of process aware information systems [Electronic resource] - 2013 - Access mode: <https://www.sciencedirect.com/science/article/pii/S0306437912000567>
44. Anomaly detection: A survey [Electronic resource] - 2009 - Access mode: <https://dl.acm.org/doi/abs/10.1145/1541880.1541882>
45. Sequential anomaly detection techniques in business processes [Electronic resource] - 2017 - Access mode: [https://link.springer.com/chapter/10.1007/978-3-319-52464-1\\_18](https://link.springer.com/chapter/10.1007/978-3-319-52464-1_18)
46. Smart anomaly detection in sensor systems: A multi-perspective review [Electronic resource] - 2021 - Access mode: <https://www.sciencedirect.com/science/article/pii/S1566253520303717>
47. Information-theoretic measures for anomaly detection [Electronic resource] - 2000 - Access mode: <https://ieeexplore.ieee.org/abstract/document/924294>

48. Anomaly-based network intrusion detection: Techniques, systems and challenges [Електронний ресурс]. – 2009. – Режим доступу до ресурсу: <https://www.sciencedirect.com/science/article/abs/pii/S0167404808000692>
49. A machine learning approach to anomaly detection. [Electronic resource] - 2003 - Access mode: <https://cs.fit.edu/media/TechnicalReports/cs-2003-06.pdf>
50. Denning D. An Intrusion Detection Model / D. Denning // IEEE Transactions on Software Engineering. – 1987. [Електронний ресурс]. – 1987. – Режим доступу до ресурсу: <https://www.cs.colostate.edu/~cs656/reading/ieee-se-13-2.pdf>
51. М. В. Грайворонський, О. М. Новіков. Безпека інформаційнокомунікаційних систем — 2009
52. Guide to Intrusion Detection and Prevention Systems [Електронний ресурс]. – 2007. – Режим доступу до ресурсу: <https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-94.pdf>
53. Behavior analysis and anomaly detection for a digital substation on cyber-physical system [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://pure.korea.ac.kr/en/publications/behavior-analysis-and-anomaly-detection-for-a-digital-substation->
54. Deep Learning for Anomaly Detection: A Review [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://arxiv.org/pdf/2007.02500>
55. Hybrid Intrusion Detection Systems (HIDS) [Електронний ресурс]. – 2006. – Режим доступу до ресурсу: [https://cdn.intechopen.com/pdfs/14361/InTech-Hybrid\\_intrusion\\_detection\\_systems\\_hids\\_using\\_fuzzy\\_logic.pdf](https://cdn.intechopen.com/pdfs/14361/InTech-Hybrid_intrusion_detection_systems_hids_using_fuzzy_logic.pdf)
56. Python. Режим доступу: <https://www.python.org>
57. The Significant Features of the UNSW-NB15 and the KDD99 Data Sets for Network Intrusion Detection Systems. Режим доступу: <http://surl.li/aswuw>
58. Network Anomaly Detection: A Machine Learning Perspective [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: [https://www.researchgate.net/profile/Dhruba-K-Bhattacharyya/publication/307936101\\_Network\\_Anomaly\\_Detection\\_A\\_Machine\\_Learni](https://www.researchgate.net/profile/Dhruba-K-Bhattacharyya/publication/307936101_Network_Anomaly_Detection_A_Machine_Learni)

ng\_Perspective/links/5bfb74a292851ced67d7f669/Network-Anomaly-Detection-A-Machine-Learning-Perspective.pdf

59. ТАКСОНОМІЯ АНОМАЛІЙ В КОНТЕКСТІ КІБЕРЗАХИСТУ ТА ЗАХИСТУ ІНФОРМАЦІЇ [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://ela.kpi.ua/server/api/core/bitstreams/ecdf61dd-487e-46e2-817a-c80f2198eff7/content>

60. AI in Anomaly Detection: Uncovering Hidden Threats in Data in Real Time [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://www.linkedin.com/pulse/ai-anomaly-detection-uncovering-hidden-threats-data-real-kashyap/>

61. Best-practice-for-network-segmentation [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://github.com/sergiomarotco/Network-segmentation-cheat-sheet>

**ДОДАТОК А**  
**СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ**

**Тези наукових доповідей:**

1. Близнюк М. Integration of AI (ChatGPT) with SIEM (ElasticSearch) for detecting cybersecurity incidents / Mykola Blyzniuk / VII Міжнародна науково-практична конференція “Проблеми кібербезпеки інформаційно-телекомунікаційних систем” (PCSITS)” 26 КВІТНЯ 2024, КИЇВ, Україна, стр. 128-131.



```
file.write(download_response.content)
        return f"Report saved as {file_path}"
    elif download_response.status_code != 503:
        return f"Failed to download the report:
{download_response.status_code} - {download_response.text}"
    else:
        return "Download path not found in the initial
response."
else:
    return f"Failed to initiate report generation:
{initial_response.status_code} - {initial_response.text}"

def send_log_to_chatgpt(api_key, assistant_id):
    list_of_files = glob.glob('logs/*')
    latest_file = max(list_of_files, key=os.path.getctime)

    with open(latest_file, 'r') as file:
        log_content = file.read()
    gen_client = openai.OpenAI(api_key=api_key)
    gen_chat = gen_client.beta.threads.create(
        messages=[
            {
                "role": "user",
                "content": log_content
            }
        ]
    )
    query =
gen_client.beta.threads.runs.create(thread_id=gen_chat.id,
```

```
assistant_id=assistant_id)

    while query.status != "completed":
        query =
gen_client.beta.threads.runs.retrieve(thread_id=gen_chat.id,
run_id=query.id)
        print
        time.sleep(1)
    else:
        print("Query completed")

    response =
gen_client.beta.threads.messages.list(thread_id=gen_chat.id)
    messages=response.data

    latest_message = messages[0]
    print(latest_message.content[0].text.value)

url = "url"
auth = ('user', 'pass')
headers = {'kbn-xsrf': 'true'}
result = download_kibana_report(url, auth, headers)
print(result)

send_log_to_chatgpt('api_key', 'assistant_id')
```