ТЕМА: “ Методика виявлення кібератак/кіберінцидентів у системах управління інформацією та подіями безпеки із застосуванням генеративного штучного інтелекту*”*

ШИФР “Бетельгейзе”

ЗМІСТ

ВСТУП

Актуальність роботи. У сучасному цифровому світі обсяги даних, що генеруються інформаційними системами, зростають експоненціально. Разом із цим суттєво ускладнюється і характер кіберзагроз. Традиційні підходи до моніторингу безпеки виявляються недостатніми: вони потребують значного людського ресурсу, не завжди здатні оперативно реагувати на нові типи атак та часто генерують на надмірну кількість хибних спрацьовувань. Це зумовлює необхідність створення інтелектуальних, адаптивних та автоматизованих систем кіберзахисту.

Одним з ключових напрямів розвитку у цій сфері є поєднання рішень типу SIEM (Security Information and Event Management) з технологіями штучного інтелекту. Зокрема, використання генеративних моделей ШІ відкриває нові можливості для глибокого аналізу журналів роботи системних застосунків, виявлення аномалій, формування звітів та надання рекомендацій для реагування на кіберінциденти. Такі моделі здатні опрацьовувати велику кількість даних, інтерпретувати події у контексті, а також навчатися на основі історичних шаблонів загроз.

Мета роботи – підвищення ефективності виявлення кіберінцидентів шляхом інтелектуалізації процесу їхнього аналізу на основі застосування генеративного штучного інтелекту.

Відповідно до мети роботи взаємопов’язаними частковими завданнями є наступні:

1. Проаналізувати процеси збору та аналізу журналів подій інформаційної безпеки у системах кіберзахисту.
2. Розробити методику виявлення кібератак/кіберінцидентів у системах управління інформацією та подіями безпеки.
3. Розробити модуль автоматизованого аналізу журналів роботи системних застосунків у складі систему управління інформацією та подіями безпеки та оцінити ефективність запропонованих рішень.

Наукова новизна отриманих у ході дослідження результатів полягає у тому, що удосконалено методику виявлення кібератак/кіберінцидентів у системах управління інформацією та подіями безпеки, яка на відміну від існуючих, застосовує генеративний штучний інтелект для аналізу даних, що поступають в систему з журналів роботи системних застосунків.

Об’єкт дослідження – процес аналізу даних про події інформаційної безпеки в системах управління інформацією та подіями безпеки.

Предмет роботи – методи аналізу даних про події інформаційної безпеки в системах управління інформацією та подіями безпеку.

Практичне значення отриманих результатів полягає у застосуванні розробленого модуля для автоматизованого збору та аналізу журналів роботи системних застосунків, що дає змогу підвищити ефективність виявлення кіберінцидентів, скоротити час реагування на загрози та зменшити навантаження на офіцерів з кіберзахисту.

РОЗДІЛ І. АНАЛІЗ ПРОЦЕСІВ ЗБОРУ ТА АНАЛІЗУ ЖУРНАЛІВ ПОДІЙ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ У СИСТЕМАХ КІБЕРЗАХИСТУ

Зі зростанням складності та інтенсивності кібератак дедалі більше стає очевидною необхідність впровадження проактивних підходів до захисту інформаційних систем і технологій. Для реалізації таких підходів потрібне оперативне прийняття рішень, що дозволяє оцінити поточну ситуацію та вжити заходів ще до завершення кібератаки. Через жорсткі часові обмеження традиційні методи експертної оцінки стають недостатніми, що підсилює потребу у впровадженні автоматизованих систем. Автоматизація значно прискорює процес виявлення інцидентів безпеки, а крім того розробку і впровадження заходів протидії кіберзагрозам. Головною особливістю таких систем є високий рівень інтелектуальності, який відповідає професійному рівню фахівців з кібербезпеки.

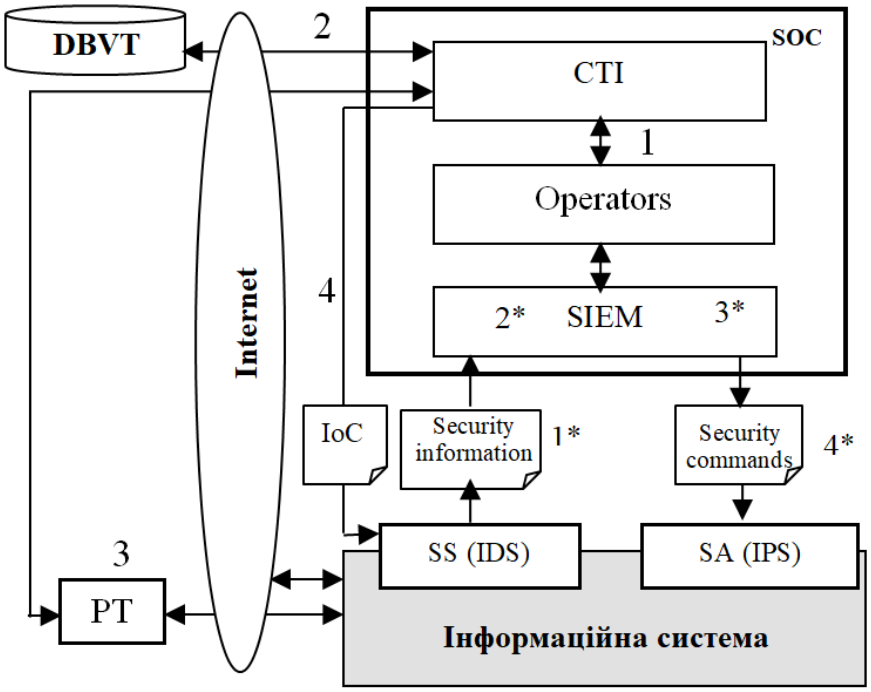
Кіберзахист – це постійно змінний процес, а не фіксований стан. Неможливо побудувати систему захисту один раз і розраховувати, що вона завжди буде ефективною проти будь-яких загроз [1]. Стандарти безпеки не можна встановити раз і назавжди, оскільки вони не гарантують повний захист у майбутньому.

Інфраструктура кіберзахисту (ІК) - це організаційно-технічна система, що поєднує апаратні та програмні засоби кіберзахисту, фахівців у цій галузі, а також організаційні заходи для забезпечення процесів кіберзахисту [2]. Кіберзахист складається з п’яти основних етапів, які виконуються послідовно, формуючи цикл кіберзахисту. Цей цикл повторюється безперервно, забезпечуючи постійний захист від кіберзагроз. Основою цього циклу є концепція управління ризиками загроз, що включає ідентифікацію та оцінку ризиків, виявлення реалізованих загроз і управління ризиками до прийнятного рівня [1].

Основою ІК є сукупність апаратних і програмних засобів, які виконують різні функції, зокрема: засоби системи виявлення вторгнень (IDS), системи запобігання вторгненням (IPS), центри операцій кібербезпеки (SOC), системи

управління подіями та інформацією безпеки (SIEM), а крім того засоби та заходи розвідки кіберзагроз (CTI) [1].

Всі процеси ІК можливо поділити на дві групи:



I. Процеси розвідки кіберзагроз (CTI):
1 – визначення характеристик ІС;
2 – визначення всіх актуальних загроз;
3 – визначення актуальних вразливостей за допомогою РТ
4 – формування набору індикаторів компрометації для сенсорів ІС
II. Процеси кіберзахисту (CD):
1* – спостереження за подіями безпеки ІС;
2* – збір і нормування інформації з сенсорів безпеки про стан ІС;
3* – аналіз подій і прийняття рішень про наявність атаки.
4* – прийняття рішення про протидію та його реалізація актуаторами безпеки.

Рисунок 1.1 – Модель інфраструктури кіберзахисту та процесів керування кібербезпекою [1]

Інфраструктура кіберзахисту складається з наступних компонентів [1]:

1. Інформаційна система (ІС) – об’єкт керування (Control object, CO).

2. Підсистема керування (Control center, CC) кібербезпекою ІС, яка включає:

– IDS – сенсори безпеки, що відстежують події в ІС та генерують відповідну інформацію;
– SOC – контролер, що збирає інформацію про події в ІС, приймає рішення щодо кіберінцидентів та здійснює заходи для відновлення кібербезпеки;
– IPS – актуатори безпеки, що реалізують впливи на ІС відповідно до рішень SOC.

3. До складу SOC входять засоби SIEM, CTI та фахівці з безпеки (офіцери, оператори).

Процеси керування кібербезпекою можуть бути представлені за допомогою графічної формалізації, що відображають їх структуру і взаємодію між компонентами (рисунок 1) [1].

Процеси CTI є попередніми відносно процесів CD. Вони забезпечують IDS ознаками можливих подій безпеки (IOCs), без яких неможливе виявлення кіберінцидентів та формування відповіді на атаку.

Основні етапи процесів CTI:

– визначення переліку ресурсів інформаційних систем (ІС), які потребують захисту, та відповідних інформаційних технологій;

– збір інформації про відповідні вразливості, загрози та ознаки подій безпеки з різних джерел на основі переліку ресурсів.

Джерела інформації поділяються на три групи:

– загальнодоступні безкоштовні бази даних про вразливості та загрози (наприклад, NVD, MITRE ATT&CK, CAPEC, OWASP, CERT/CC);

– бази даних спеціалізованих фірм, що надають інформацію за плату;

– власні підрозділи CTI, які визначають актуальні вразливості ІС шляхом тестування на проникнення (penetration testing, PT);

– формування набору індикаторів компрометації для сенсорів безпеки ІС.

Основні етапи процесів CD [1]:

– IDS різних компонентів ІС на основі індикаторів компрометації визначають (детектують) події безпеки та формують відповідну інформацію у вигляді файлів журналів подій (logs);

– офіцери безпеки засобами SIEM збирають логи зі всіх компонентів ІС, після нормалізації та аналізу інформації приймають рішення про наявність кіберінциденту;

– після комплексного аналізу кіберінциденту приймається рішення про необхідні дії по реагуванню на подію безпеки;

– на основі прийнятого рішення формуються відповідні команди, що реалізуються засобами IPS.

Аналіз інфраструктури кіберзахисту дає змогу забезпечити ефективний захист інформаційних систем (ІС) від кіберзагроз. Її основні функції включають виявлення потенційних загроз і вразливостей в ІС, реагування на кіберінциденти, а також запобігання подібним атакам у майбутньому. Аналіз ІК допомагає оцінити рівень загроз, визначити вразливі місця і розробити стратегії захисту. Крім того, він сприяє вдосконаленню процесів кіберзахисту шляхом постійного моніторингу, аналізу та вдосконалення заходів безпеки [2].

Система управління інформацією та подіями безпеки (SIEM) - це система, яка збирає дані журналів подій з різних інструментів безпеки, щоб допомогти командам безпеки та компаніям отримати цілісне уявлення про загрози в мережі та об'єктах атак [12]. За допомогою інструментів SIEM аналітики з кібербезпеки виявляють, розслідують та усувають найсучасніші кіберзагрози, що дозволяє командам безпеки визначати пріоритети, інтерпретувати та аналізувати сукупні дані про інциденти кібербезпеки в єдиному місці. Завдяки SIEM організації мають унікальну можливість не лише протистояти існуючим кібератакам, але й краще розуміти дані про події, щоб запобігти майбутнім порушенням.

Процес управління журналами SIEM складається з п’яти ключових кроків [6]. Спочатку відбувається збір даних через захищені канали, що забезпечує цілісність інформації. Потім ці дані надійно зберігаються з використанням шифрування, архівування та резервного копіювання для подальшого доступу.

Наступним кроком є швидкий і ефективний пошук потрібної інформації за допомогою індексації і фільтрів. Кореляція дозволяє об’єднати розрізнені події в єдину картину атаки для кращого розуміння інцидентів і автоматизації реагування. Останній етап – виведення результатів для аналізу та прийняття рішень.

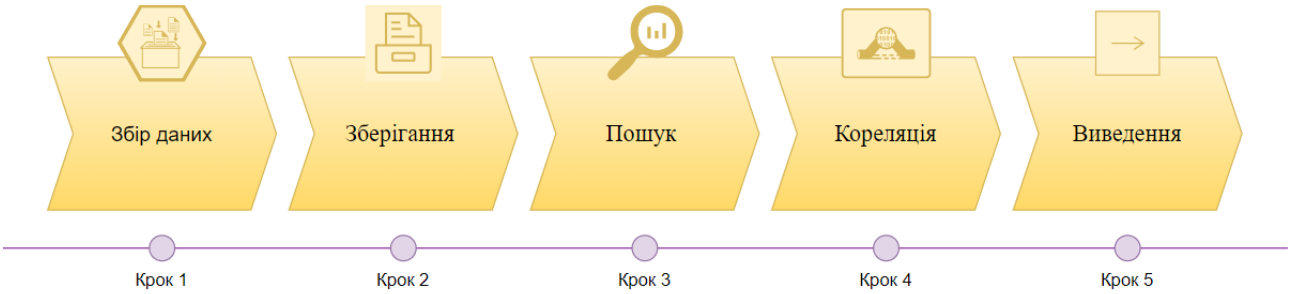


Рисунок 1.2 – Процес управління журналами SIEM [6]

Програмне забезпечення SIEM зіставляє події із заздалегідь визначеними правилами для оцінки серйозності та рівня загрози, щоб створити оповіщення SIEM. Виявлення на основі правил визначає базовий рівень підозрілої активності та зменшує витрати часу вашої команди безпеки на розслідування хибних спрацьовувань.

Спочатку зосереджені на управлінні журналами роботи та звітуванні про відповідність, роль SIEM значно еволюціонувала. SIEM-системи змінили сферу кібербезпеки ще з моменту появи штучного інтелекту. Зараз, завдяки інтеграції штучного інтелекту (ШІ) та машинного навчання (ML), ці рішення не лише можуть поглинати та нормалізувати величезні обсяги даних, а й аналізувати шаблони та аномалії, які можуть свідчити про кіберінцидент. [6]

Одним із ключових процесів у SIEM на основі ШІ є агрегація даних. Це означає збирання даних безпеки з численних джерел, зокрема мережевих пристроїв, серверів, баз даних, застосунків тощо [12]. Зібрана інформація включає журнали подій, дані про загрози та інші види безпекових даних. У сучасному цифровому середовищі агрегація є надзвичайно важливою, адже дозволяє отримати цілісне уявлення про стан безпеки організації. Проблема

полягає у різноманітності форматів і структур даних. Тут у гру вступає нормалізація – процес перетворення "сирих" даних з різних джерел у єдиний стандартний формат [12]. Це критично для точного аналізу та кореляції даних, незалежно від їх походження.

Після агрегації та нормалізації даних SIEM на основі ШІ застосовує алгоритми для покращеного виявлення загроз. Ці алгоритми навчаються на основі вже відомих шаблонів загроз і здатні виявляти нові, що розвиваються, через аналіз поведінкових патернів. Така здатність критично важлива в умовах постійної еволюції загроз. Аналітика на основі трендів дозволяє прогнозувати потенційні атаки та проактивно посилювати захист. [7]

Основні компоненти SIEM, що працюють на основі ШІ:

- обробка даних;
- джерела великих даних (Big Data);
- розпізнавання шаблонів;
- автоматизована реакція на інциденти;
- прогнозна аналітика.

SIEM із ШІ автоматизує збір даних, нормалізацію та збагачення за допомогою зовнішньої аналітики загроз. Це зменшує потребу в ручному створенні правил.

РОЗДІЛ ІІ. РОЗРОБКА МЕТОДИКИ ВИЯВЛЕННЯ КІБЕРАТАК/КІБЕРІНЦИДЕНТІВ У СИСТЕМАХ УПРАВЛІННЯ ІНФОРМАЦІЄЮ ТА ПОДІЯМИ БЕЗПЕКИ

Розроблена методика базується на ідеї інтеграції системи управління інформацією та подіями безпеки, оркестраторів, бази даних та генеративного штучного інтелекту.

Метою є підвищення точності виявлення аномалій, автоматизація реагування на кіберінциденти та зниження кількості хибних спрацьовувань у системах інформаційної безпеки.

Нижче наведено математичну основу, яка формалізує процес аналізу подій для оцінки ризиків і прийняття рішень.

2.1. Постановка задачі

Нехай $E = \{e_i\}_{i=1}^{N}$ – множина подій, де кожна подія e_i має набір атрибутів (timestamp, srcIP, dstIP, port, rule, msg, host, …).

Необхідно: побудувати функцію прийняття рішення (детектор $D(\cdot)$) $D: E \to \{0,1\}$, яка для кожного елементу з множини подій E, визначає чи є вона нормальною (штатною), чи аномальною або підозрілою (кіберінцидент):

$$D\{E_i\} = \begin{cases} 1, \textit{якщо подія } e_i \textit{ класифікується, як аномалія}; \\ 0, \textit{якщо подія } e_i \textit{ є нормальною}. \end{cases}$$

та функцію рекомендацій $R(\cdot)$ – $R: E \to A$, де A – множина дій/рекомендацій (alert, block IP, request TI, escalate, …).

За умови мінімізації функції втрат:

$$\xi = \min_{D} \left(\lambda_{FN} \cdot FN(D) + \lambda_{FP} \cdot FP(D) \right),$$

де $FN(D)$ – кількість пропущених кіберінцидентів, що залежить від параметрів детектора D;

$FP(D)$ – кількість хибних спрацювань детектора;

$\lambda_{FN,}$ λ_{FP} – вагові коефіцієнти (штрафи).

2.2 Методика вирішення поставленої задачі

2.2.1 Побудова ознак та відображення в простір ознак

Кожна подія e представляється у вигляді вектору ознак $x = [x_1, x_2, \ldots, x_d] \in R^d$

та далі використовується алгоритмами класифікації (детекції). Тут вектор x – це числова репрезентація події e, де кожен елемент вектору – це певна ознака: кількість спроб входу, рівень критичності, IP-адреса, ключові слова повідомлення тощо.

Тут категоріальні ознаки, наприклад, $rule = \{"\log in\_fail", \log in\_success", "malware\_detected"...\}$ можуть бути закодовані методами “one-hot” або “emdeddings”, а текстові, наприклад, msg = “Failed password for root from 10.0.0.5 port 4321” методами TF–IDF (Term Frequency – Inverse Document Frequency) або LLM-embedding (GPT).

У першому випадку, TF–IDF (застосовується для токенів у полі повідомлення):

$$tf - idfF_{t,d} = tF_{t,d} \cdot \log \frac{N}{dF_t},$$

де $F_{t,d}$ – частота терма t у документі d, N– кількість документів, df_t– кількість документів з t.

У другому випадку, LLM-embedding: функція f_{emb} має виглчяд:

$$z = f_{emb}(msg) \in R^m,$$

де z – результат – вектор ознак (embedding-вектор), який “представляє зміст” тексту у вигляді чисел;

f_{emb} – алгоритм або модель, яка перетворює текст у числовий вектор;

msg – текстове поле події — наприклад, повідомлення із журналу системи безпеки (“Failed password for root from 10.0.0.5”);

R^m – простір дійсних чисел розмірності m — тобто вектор має m компонент.

Після представлення подій у вигляді векторів, застосовується формула косинусної схожисті (міра подібності векторів):

$$\cos\left(z_1, z_2\right) = \frac{z_1 z_2}{\|z_1\|\|z_2\|},$$

де z_1, z_2 – вектори подій (наприклад, текстові embeddings двох журналів);

$z_1 z_2$ – скалярний добуток векторів;

$\|z_1\|, \|z_2\|$ – довжини (норми) цих векторів.

Результатом є число у діапазоні від -1 до 1, яке показує схожість між подіями.

2.2.2 Модель детектування (класифікації)

Тут використовується генеративна модель для семантичного аналізу полів журналів і формування рекомендацій.

Формально:

Крок 1: промпт pформується на основі структури події $e: p = \phi(e)$.

Крок 2: LLM дає текстову відповідь $T = \text{LLM}(p)$ $T = LLM\left(p\right)$.

Крок 3: для структурування відповіді застосовують парсер/класифікатор g:

$$a = g\left(T\right) \in R^{|\text{A}|},$$

де T – текстова відповідь від генеративної моделі (наприклад, ChatGPT або LLM) після аналізу події;

$g\left(T\right)$ – функція, яка перетворює текст у набір числових оцінок для кожної можливої дії;

$a = \left[a_1, a_2, \ldots, a_{|\text{A}|}\right]$ – вектор цих оцінок (по одній на кожну дію);

A – множина можливих дій (наприклад: {"alert", "block_ip", "ignore", "escalate"}).

Отже: $g\left(T\right): \textit{текст} \rightarrow \textit{числові оцінки дій}, a \in R^{|A|}$.

Після цього застосовується функція softmax, яка переводить ці “сирі оцінки” a_i у ймовірності:

$$P(a_i|T) = \frac{\exp(\alpha a_i)}{\sum_j \exp(\alpha a_j)},$$

де a_i – оцінка “наскільки доречно виконати дію i” (вихід моделі);

α – коефіцієнт масштабування (чим більший, тим різкіше модель вибирає максимум);

$\sum_j \exp(\alpha a_j)$ – сума експонент усіх оцінок (для нормалізації).

Інтуїтивно, функція softmax “перетворює” набір оцінок у розподіл ймовірностей: найбільші *найбільші* $a_i \to$ *найбільші* $P(a_i)$, *найменші* $a_i \to$ *найменші* $P(a_i)$. Таким чином, система отримує ймовірність доцільності кожної можливої дії.

Крок 4: вибір дії (правило прийняття рішення).

Система обирає дію з найвищою ймовірністю, тобто дію, для якої ймовірність $P(a_i|T)$ є максимальною:

$$a^* = \arg\max_i P(a_i|T),$$

причому $P(a_i|T) > \tau$, де τ – заданий поріг (наприклад, 0.8).

Інакше кажучи, для уникнення “випадкових спрацьовувань”, дія виконується лише тоді, коли ймовірність перевищує заданий поріг.

2.2.3 Практична реалізація методики виявлення кібератак / кіберінцидентів

Для досягнення поставленої мети важливо вчасно фіксувати спроби атак на інформаційну інфраструктуру. Для цього необхідно здійснювати збирання журналів подій і моніторинг усіх можливих компонентів: серверів, хостів, міжмережевих екранів, антивірусних засобів, користувацьких пристроїв тощо.

У рамках проєкту Wazuh виконує роль SIEM-рішення: здійснює збір та агрегацію логів, їх аналіз і візуалізацію, що дозволяє своєчасно виявляти спроби вторгнень, шкідливе програмне забезпечення, вразливості у програмному забезпеченні, а також контролювати цілісність системних файлів. [12]

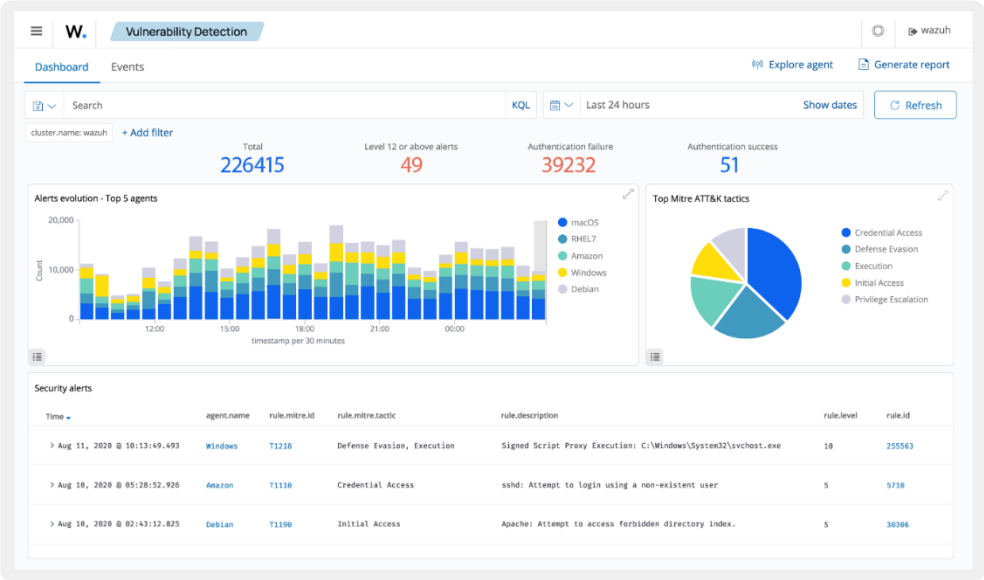


Рисунок 2.1 – Меню Wazuh

У рамках реалізації проєкту до системи Wazuh буде підключено кілька віртуальних машин з операційними системами Ubuntu та Windows, які виступають джерелами подій. Для підвищення ефективності виявлення мережевих атак до цієї інфраструктури кіберзахисту, також було інтегровано систему виявлення вторгнень (IDS) [3], що дозволяє в режимі реального часу фіксувати підозрілу активність на мережевому рівні та передавати відповідні журнали подій до Wazuh для подальшого аналізу.

Система Wazuh підтримує збір журналів подій як з хостових систем (серверів, робочих станцій, віртуальних машин), так і з мережевих пристроїв (маршрутизаторів, міжмережевих екранів, IDS/IPS-систем), що забезпечує комплексний підхід до моніторингу інформаційної безпеки та створює єдину точку для виявлення, кореляції й обробки подій безпеки.

Автоматизований збір, візуалізація та первинний аналіз логів черговими адміністраторами відповідно до затверджених регламентів забезпечили побудову безперервного процесу моніторингу та оперативного реагування на події інформаційної безпеки в режимі 24/7.

Після впровадження системи Wazuh для централізованого збору та аналізу логів виникла потреба в додатковій автоматизації процесів обробки та передачі

даних. Зокрема, необхідно було забезпечити автоматичну передачу релевантної інформації з Wazuh до зовнішньої бази даних для подальшого аналізу, візуалізації або збереження історичних подій. Такий підхід дозволяє уникнути ручних операцій, зменшити ризик втрати даних, прискорити реагування на інциденти та забезпечити гнучкість інтеграції з іншими системами.

Для реалізації цього завдання було обрано Shuffle – сучасну платформу типу SOAR (Security Orchestration, Automation and Response), яка надає можливості створення автоматизованих робочих процесів (workflow) без потреби у написанні складного коду [3]. Shuffle дозволяє інтегрувати Wazuh з базами даних, системами сповіщення, зовнішніми API та іншими компонентами інфраструктури.

Платформа Shuffle може виконувати низку додаткових завдань, які значно розширюють можливості системи кіберзахисту, а саме [3]:

– автоматичне надсилання сповіщень (наприклад, у Telegram, Slack або на електронну пошту) у разі виявлення критичних інцидентів;

– фільтрація та попередня обробка подій, що дозволяє зменшити обсяг даних, які надходять до систем зберігання;

– інтеграція з інструментами реагування – наприклад, можна автоматично заблокувати IP-адресу або закрити підозріле з’єднання на міжмережевому екрані;

– виконання запитів до зовнішніх сервісів Threat Intelligence для уточнення контексту інцидентів (наприклад, перевірка IP на наявність у базах шкідливих адрес);

– побудова логіки інцидент-менеджменту, коли платформа сама визначає подальші дії залежно від типу події та її критичності.

Wazuh при виявленні події (наприклад, спроби вторгнення, виявлення шкідливого ПЗ або підозрілої активності) генерує повідомлення про подію, який зберігається у відповідному лог-файлі (alerts.json). Цей файл може бути прочитаний або перехоплений спеціальним модулем, який служить тригером у системі Shuffle.

Shuffle, у свою чергу, реагує на появу нового повідомлення про подію — автоматично активується відповідний workflow, який може включати:

– витяг ключової інформації з повідомлення про кіберінцидент (IP, правило, рівень загрози, час тощо);

– перевірку джерела події;

– умовну логіку, яка дозволяє відфільтровувати лише ті події, що відповідають визначеним критеріям (наприклад, лише критичні події рівня 10 або лише ті, що стосуються певних правил);

– надсилання сповіщень до відповідальних осіб;

– збереження відібраної інформації до бази даних;

– запуск автоматичних заходів реагування.

Завдяки такому підходу до бази даних потрапляє лише необхідна інформація, без зайвих технічних даних, що дозволяє зменшити навантаження, забезпечити швидшу обробку та зосередитися лише на справді важливих подіях.

Використання Shuffle дозволяє перетворити звичайну систему логування Wazuh на повноцінну автоматизовану екосистему для виявлення, реагування та документування інцидентів, що відповідає сучасним вимогам до безпеки інформаційних систем.

У рамках побудови системи моніторингу та автоматизації обробки інцидентів інформаційної безпеки виникла потреба в надійній системі управління базами даних (СУБД) для зберігання, обробки та подальшого аналізу зібраної інформації. З огляду на вимоги до стабільності, продуктивності, простоти інтеграції з іншими компонентами системи та відкритого коду, було обрано MySQL – одну з найпоширеніших та перевірених часом реляційних СУБД. [10]

Для забезпечення аналізу журналів роботи системних застосунків та формування висновків щодо характеру виявлених кіберзагроз було прийнято рішення інтегрувати розроблений модуль із системою генеративного штучного інтелекту.

Вимоги до генеративного ШІ:

– можливість обробки незалежних текстових запитів (логів) з контекстним аналізом;

– формування рекомендацій щодо реагування;

– простота інтеграції у середовищі Python;

– відкритість або відсутність необхідності в API-ключах для спрощення тестування.

Для реалізації поставлених задач було обрано g4f (GPT4Free) (рисунок 2.2) – Python-бібліотеку, яка забезпечує інтеграцію з великими мовними моделями (LLM), такими як ChatGPT, через відкриті або неофіційні інтерфейси доступу до моделей.

Основні переваги g4f:

– безкоштовність доступу до генеративних моделей;

– простота використання через зрозумілий синтаксис Python;

– відсутність необхідності реєстрації або отримання API-ключів.

```python
from g4f.client import Client

# Ініціалізація клієнта
client = Client()

# Виконання запиту до моделі GPT-4o-mini
response = client.chat.completions.create(
    model="gpt-4o-mini",  # Вказуємо модель GPT-4o-mini
    messages=[
        {"role": "user", "content": "Проаналізуй цей лог та сформулюй можливу загрозу: \n10.0.0.5 - -
[2025-06-10 13:55:36] \"GET /admin HTTP/1.1\" 404 -"}
    ],
)

# Виведення відповіді
print(response.choices[0].message.content)
```

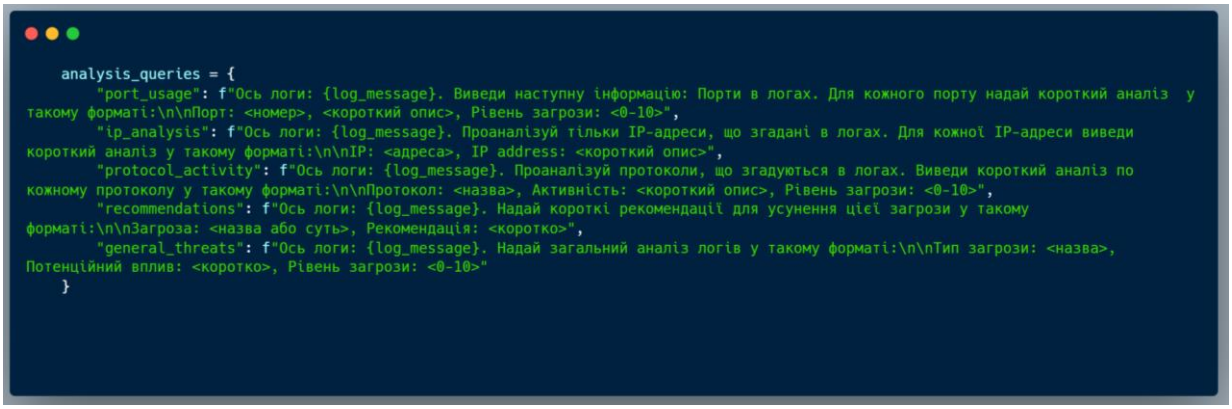
Рисунок 2.2 – Приклад інтеграції g4f з GPT-4o-mini через Python-бібліотеку g4f

Для ефективної взаємодії з генеративною моделлю штучного інтелекту була розроблена система промптів, що забезпечує чітке формулювання завдань для аналізу логів. Кожен промпт формулюється відповідно до конкретного сценарію використання, щоб отримувати релевантні, структуровані та зрозумілі відповіді [9].

Основні типи промптів, що використовуються:

– аналіз портів;
– аналіз IP-адрес;
– аналіз протоколів;
– рекомендації щодо усунення загроз;
– загальний аналіз загроз.

Реалізації промтів у коді (рисунок 2.3):



```
analysis_queries = {
    "port_usage": f"Ось логи: {log_message}. Виведи наступну інформацію: Порти в логах. Для кожного порту надай короткий аналіз у такому форматі:\n\nПорт: <номер>, <короткий опис>, Рівень загрози: <0-10>",
    "ip_analysis": f"Ось логи: {log_message}. Проаналізуй тільки IP-адреси, що згадані в логах. Для кожної IP-адреси виведи короткий аналіз у такому форматі:\n\nIP: <адреса>, IP address: <короткий опис>",
    "protocol_activity": f"Ось логи: {log_message}. Проаналізуй протоколи, що згадуються в логах. Виведи короткий аналіз по кожному протоколу у такому форматі:\n\nПротокол: <назва>, Активність: <короткий опис>, Рівень загрози: <0-10>",
    "recommendations": f"Ось логи: {log_message}. Надай короткі рекомендації для усунення цієї загрози у такому форматі:\n\nЗагроза: <назва або суть>, Рекомендація: <коротко>",
    "general_threats": f"Ось логи: {log_message}. Надай загальний аналіз логів у такому форматі:\n\nТип загрози: <назва>, Потенційний вплив: <коротко>, Рівень загрози: <0-10>"
}
```

Рисунок 2.3 – Розроблені промти

Кожен ключ словника відповідає певному сценарію аналізу:

– port_usage — аналіз портів;
– ip_analysis — аналіз IP-адрес;
– protocol_activity — аналіз протоколів;
– recommendations — рекомендації з усунення загроз;
– general_threats — загальний аналіз загроз.

Рисунок 2.4 – Алгоритм методики виявлення кібератак/кіберінцидентів у системі

На діаграмі показано послідовність етапів методики: збирання журналів подій із компонентів мережі у Wazuh, автоматизовану передачу релевантних подій до бази даних за допомогою платформи Shuffle, збереження даних у MySQL, формування промптів та їх аналіз генеративним штучним інтелектом, а крім того можливий запуск автоматизованих заходів реагування.

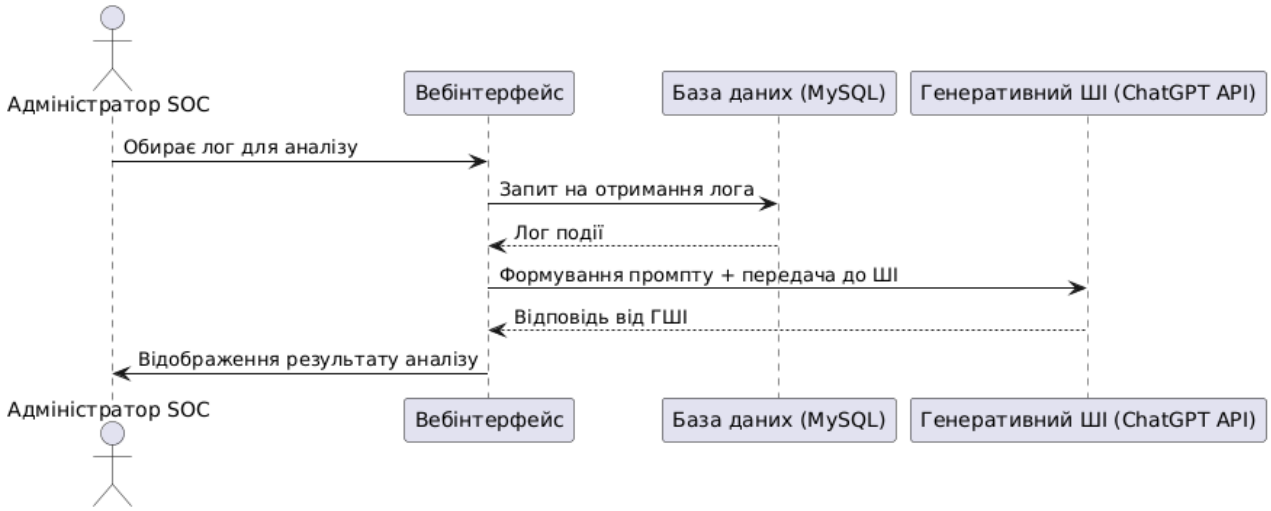


Рисунок 2.5 – Sequence діаграма взаємодії системи з ГШІ

Діаграма ілюструє послідовність дій між Адміністратором SOC, Веб-інтерфейсом, Базою даних (MySQL) та сервісом ГШІ. Ця схема показує типову взаємодію користувача з системою аналізу логів, де логи зберігаються у базі даних, а штучний інтелект використовується для аналізу цих логів за допомогою промптів, сформованих веб-інтерфейсом.

РОЗДІЛ ІІІ. РОЗРОБКА МОДУЛЮ АВТОМАТИЗОВАНОГО АНАЛІЗУ ЖУРНАЛІВ РОБОТИ СИСТЕМНИХ ЗАСТОСУНКІВ У СКЛАДІ СИСТЕМУ УПРАВЛІННЯ ІНФОРМАЦІЄЮ ТА ПОДІЯМИ БЕЗПЕКИ ТА ОЦІНКА ЕФЕКТИВНОСТІ ЗАПРОПОНОВАНИХ РІШЕНЬ

Розроблений програмний модуль призначений для інтерактивного моніторингу подій інформаційної безпеки, які автоматично надходять із системи збору повідомлень про інциденти Wazuh. Його основна функція – зручна візуалізація, фільтрація, групування та аналітика кіберінцидентів з подальшою обробкою за допомогою генеративного штучного інтелекту.

Користувач може обрати один або декілька інцидентів, згрупувати їх і передати до модуля ШІ, який проведе автоматичний контекстний аналіз. Серед доступних сценаріїв – аналіз за IP-адресами, портами, мережевими протоколами, загальний огляд загроз і формування рекомендацій щодо реагування. Результати аналізу повертаються у текстовій формі і виводяться безпосередньо в інтерфейсі.

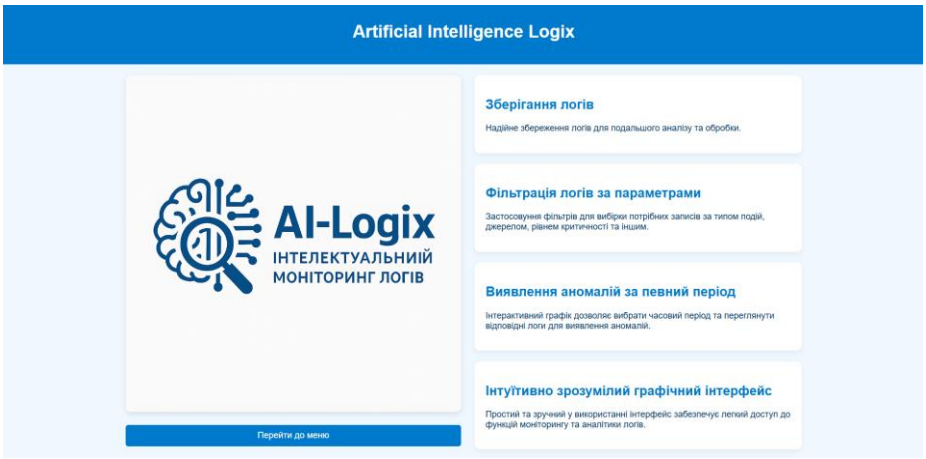


Рисунок 3.1 – Домашня сторінка сайту

Після запуску застосунку користувач може перейти до меню, яке відображається зліва на сторінці.

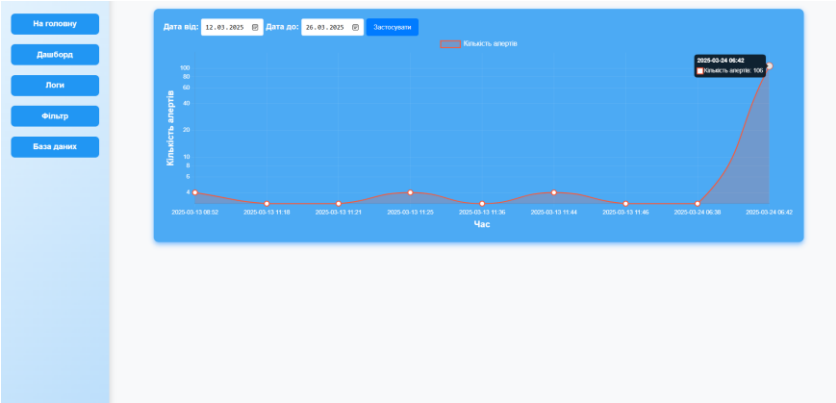


Рисунок 3.2 – Навігаційне меню та інформаційна панель

Дашборд слугує центральним місцем для оперативного моніторингу безпекової ситуації у вашій інфраструктурі. Завдяки фільтрації та пошуку подій за часом панель допомагає швидко виявляти тенденції, сплески логів і потенційні загрози. Це дозволяє адміністраторам та фахівцям з безпеки швидше реагувати на інциденти, приймати обґрунтовані рішення та адаптувати відображення даних під власні потреби через налаштовувані віджети та панелі.

На рисунку 3.3 представлено користувацький інтерфейс розділу журнали подій, доступ до якого здійснюється через відповідний пункт навігаційного меню.

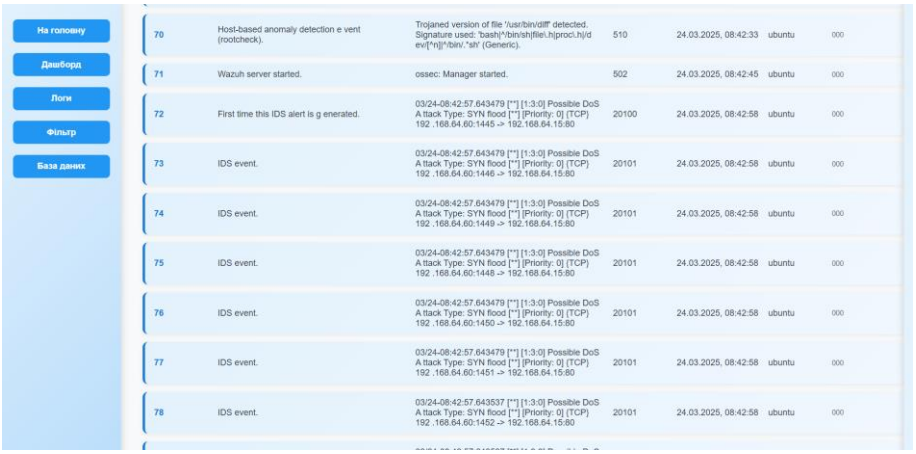


Рисунок 3.3 – Журнали подій

Даний розділ призначений для відображення журнальних записів (логів) системи безпеки, що агрегуються та зберігаються у базі даних.

Групування відібраних кіберінцидентів для подальшого аналізу засобами штучного інтелекту на рисунку 3.4.

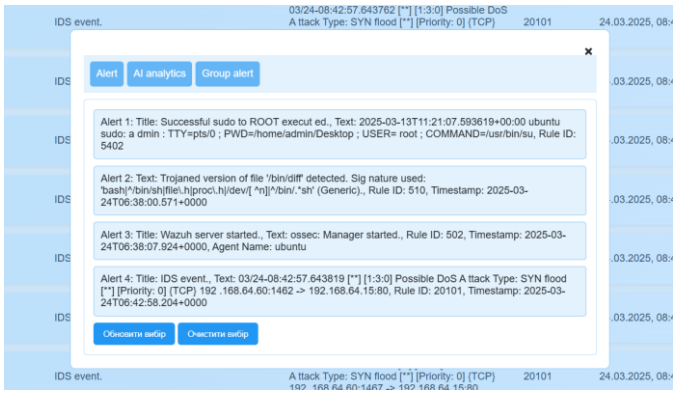


Рисунок 3.4 – Груповані повідомлення про інциденти

У вкладці аналізу відображається відповідь від штучного інтелекту, сформована на основі обраних сповіщень про події. Вивід є текстовим і містить оцінку загроз, виявлених у подіях безпеки, відповідно до обраного методу аналізу (порти, IP, протоколи тощо).

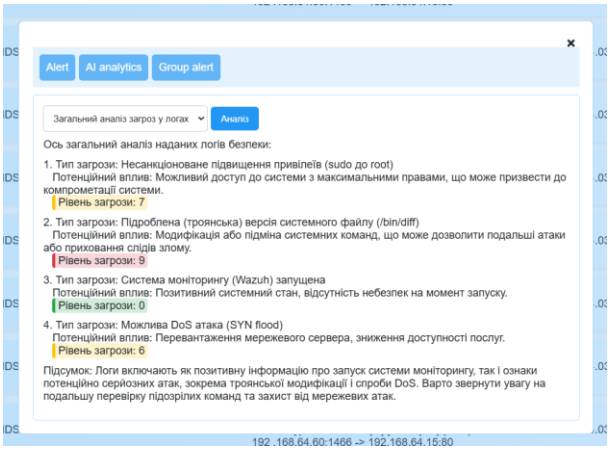


Рисунок 3.5 – Результат аналізу штучного інтелекту

З метою перевірки досягнення поставленої мети роботи – підвищення ефективності виявлення кіберінцидентів – було проведено оцінку ефективності розробленого модуля. Як основний показник ефективності обрано оперативність виявлення кіберінцидентів, тобто час, що витрачається на обробку журналу подій та отримання результатів аналізу.

Для оцінки було проведено серію тестувань, в ході яких визначався час від моменту надходження журналу подій до моменту отримання результату аналізу для однієї події. Було здійснено порівняння:

– ручного аналізу логів оператором без використання генеративного ШІ;

– автоматизованого аналізу з використанням розробленого модуля генеративного ШІ.

Для оцінки були використані реальні журнали подій, отримані у межах функціонування CSIRT-ED. Загалом було проаналізовано понад 70 унікальних записів подій кіберінцидентів (рисунок 3.6), що стосуються 2 критичних інцидентів.

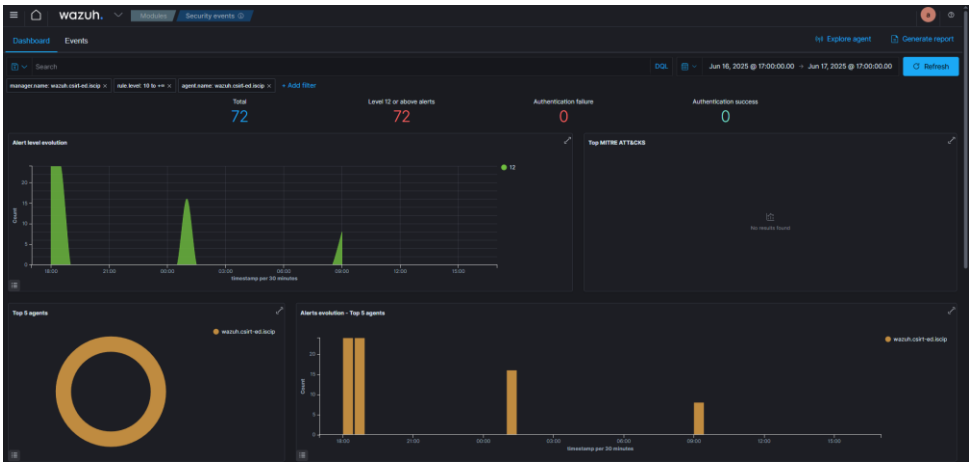


Рисунок 3.6 – Інформаційна панель Wazuh

Середній час аналізу одного інциденту вручну становив від 8 до 10 хвилин, залежно від складності інциденту та необхідності додаткового уточнення контексту загроз.

Результат аналітика вручну:

Зафіксовано 2 окремі спроби доставки шкідливого скрипту ssh.sh з IP-адреси 213.209.143.44 на honeypot-сервер T-Pot (172.16.40.5), розгорнутий у внутрішньому середовищі CSIRT-ED.

Дата та час спрацювання Fortigate:

• 16.06.2025 18:29:13

• 16.06.2025 18:36:49

Сигнатура атаки – Embedded.Linux.Malicious.Script (VID47441), яка ідентифікується як спроба доставки backdoor-компонента через HTTP-запит. В кожному з двох циклів Fortigate визначив загрозу, класифікувавши її як critical і action: dropped, що свідчить про успішне запобігання виконання зловмисного сценарію.

На стороні honeypot-системи Cowrie було зафіксовано введення команди зловмисником:

cd /tmp; wget http://213.209.143.44/ssh.sh -O- | sh; curl -o http://213.209.143.44/ssh.sh -O- | sh; tftp -r ssh.sh -g 213.209.143.44; chmod 777 ssh.sh; sh ssh.sh.

Застосування розробленого модуля з інтегрованим генеративним штучним інтелектом дозволило скоротити час формування первинного висновку до приблизно 30 секунд (пів хвилини), що забезпечило зменшення часу реагування в середньому у 16-20 разів.

Результат аналізу ШІ (рисунок 3.7) на основі наданих логів генеративна модель автоматично ідентифікувала ознаки спроби доставки шкідливого скрипту, визначила IP-адресу джерела атаки, тип загрози та її критичність, а також сформувала рекомендації щодо подальших дій для усунення та запобігання подібних атак у майбутньому.

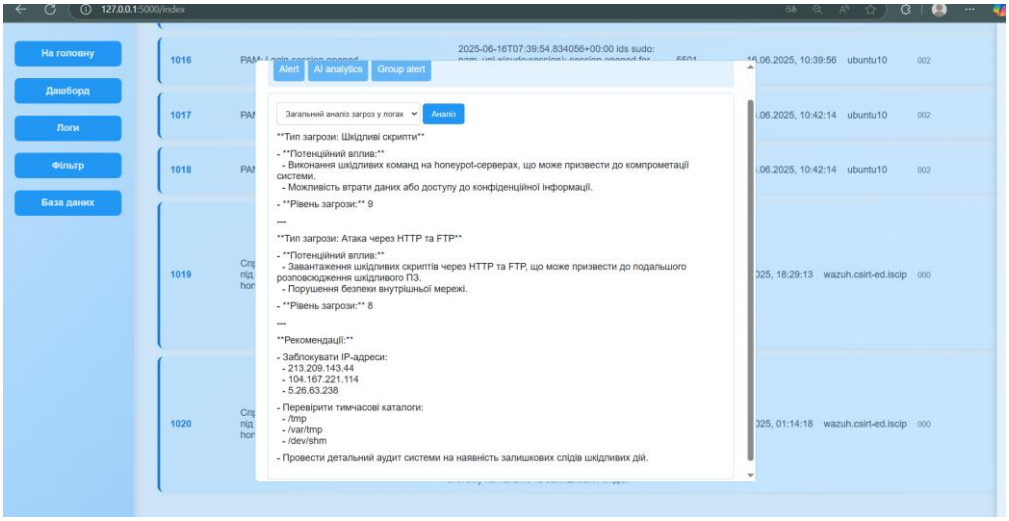


Рисунок 3.7 – Висновок ШІ

Результати тестування:

Метод аналізу	Середній час аналізу однієї події, сек.
Ручний аналіз (без ШІ)	1080 (18 хвилин)
Автоматизований аналіз (з ШІ)	30

Оцінка покращення:

Відсоток покращення оперативності обчислюється за формулою:

$$P = (T_{\text{ручн}} - T_{\text{авт}})/T_{\text{ручн}} \times 100\%$$

де: P – відсоток покращення;

$T_{\text{ручн}}$ – середній час аналізу однієї події вручну (без ШІ);

$T_{\text{авт}}$ – середній час автоматизованого аналізу однієї події (з використанням ШІ).

На основі отриманих результатів обчислено відсоток покращення оперативності:

$$P = (1080 - 30)/1080 \times 100\% = 97\%$$

Таким чином, використання розробленого модуля скорочує середній час аналізу подій на 97%.

Проведена оцінка ефективності підтверджує, що розроблене рішення забезпечує суттєве підвищення швидкості аналізу журналів у порівнянні з традиційними методами. Це дозволяє оперативно реагувати на потенційні кіберзагрози та значно знижує навантаження на персонал аналітичних підрозділів.

ВИСНОВКИ

1. У ході проведеного дослідження було проаналізовано та описано процеси збору та аналізу журналів подій інформаційної безпеки у системах кіберзахисту. Зокрема, визначено роль SIEM- та SOAR-систем у контурі кіберзахисту та запропоновано інтеграцію їх з генеративними мовними моделями. Обґрунтовано вибір системи управління інформацією та подіями безпеки Wazuh як платформа для збирання для збору та кореляції подій, а крім того системи Shuffle для автоматизації обробки інцидентів. Як середовища для зберігання даних про події безпеки було обрано реляційну базу даних MySQL, що забезпечує надійність та гнучкість у роботі з великою кількістю подій.

2. Удосконалено методику виявлення кібератак/кіберінцидентів у системах управління інформацією та подіями безпеки, яка на відміну від існуючих, застосовує генеративний штучний інтелект для аналізу даних, що поступають в систему з журналів роботи системних застосунків.

3. Розроблено модуль автоматизованого аналізу журналів роботи системних застосунків у складі систему управління інформацією та подіями безпеки та оцінити ефективність запропонованих рішень, який реалізує технології збору даних про події інформаційної безпеки, їх зберігання, фільтрації та аналізу, що дозволяє суттєво підвищити ефективність виявлення кіберінцидентів на 97% за показником оперативності

4. Розроблений модуль дозволяє в реальному часі отримувати інформацію про кіберінциденти, здійснювати пошук та фільтрацію подій за різними параметрами, групувати пов’язані кіберінциденти для комплексного аналізу. Окрему увагу приділено реалізації інтерактивного дашборду, який забезпечує візуалізацію активності в часовому розрізі, дає змогу виявляти пікові навантаження, аномальні сплески та тенденції в розвитку кіберінцидентів. Такий підхід дозволяє швидко орієнтуватися в ситуації та приймати обґрунтовані рішення.

5. Інтеграція з генеративним штучним інтелектом дала змогу автоматизувати розпізнавання аномалій, оцінку критичності загроз і формування

рекомендацій щодо подальших дій. Проведене тестування підтвердило працездатність рішення, його стабільність, зручність у користуванні та відповідність поставленим вимогам.

6. Отримані результати свідчать про доцільність використання генеративного штучного інтелекту в системах кіберзахисту та створюють основу для подальших наукових досліджень щодо розвитку інтелектуальних модулів реагування на кіберінциденти/кібератаки в умовах постійної еволюції кіберзагроз.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Яковів І.Б. Основи побудови комплексної системи захисту інформації для інформаційно-комунікаційної системи: навч. посіб. Київ: ІСЗЗІ НТУУ “КПІ ім. Ігоря Сікорського”, 2016. 89 с.
2. Кіберзахист критичної інфраструктури. Що необхідно змінити Україні? *Interfax-Ukraine*. URL: https://interfax.com.ua/news/blog/775090.html (дата звернення: 09.03.2025).
3. Intrusion Detection System (IDS). *Geeks for Geeks.* URL: https://www.geeksforgeeks.org/intrusion-detection-system-ids/ (date of access: 12.03.2025).
4. Miller D. Security information and event management (SIEM) implementation. New York: McGraw-Hill, 2011. 430 p.
5. What is SIEM? *ThreatDown by Malwarebytes.* URL: https://www.threatdown.com/glossary/what-is-siem (date of access: 14.03.2025).
6. SIEM Log Management: What It Is and Why It’s Vital for Cybersecurity. *BitLyft Cybersecurity.* URL: https://www.bitlyft.com/resources/siem-log-management-what-it-is-and-why-its-vital-for-cybersecurity (date of access: 22.03.2025).
7. Why Organizations Choose SOCFortress. *SocFortress.* URL: https://www.socfortress.co/ (date of access: 19.04.2025).
8. Shuffler Documentation. *Shuffler.* URL: https://shuffler.io/docs/about (date of access: 27.04.2025).
9. Introducing ChatGPT. *OpenAI.* URL: https://openai.com/blog/chatgpt (date of access: 27.04.2025).
10. SIEM vs Log Management. *CrowdStrike.* URL: https://www.crowdstrike.com/en-us/cybersecurity-101/next-gen-siem/siem-vs-log-management/ (date of access: 28.04.2025).
11. Snort – що це таке і як працює?. *VPN Unlimited.* URL: https://www.vpnunlimited.com/ua/help/cybersecurity/snort (дата звернення: 05.05.2025).

12. Wazuh SIEM log analysis and incident response tutorial. *Mark AI code.* URL: https://markaicode.com/wazuh-siem-log-analysis-incident-response-tutorial/ (date of access: 11.05.2025).
13. “БЕТЕЛЬГЕЙЗЕ” ...
... науково-практичної
конференції ...
...
... Київ...
2025. ...
14. “БЕТЕЛЬГЕЙЗЕ”, ...
програма ... свідоцтво про реєстрацію
авторського права № ... Київ: Державна організація «Український національний офіс інтелектуальної власності та інновацій», ... 2025. URL: ... (дата звернення: 18.09.2025).

ДОДАТОК 1

ФРАГМЕНТ ТЕКСТУ ПРОГРАМИ

project/main.py

```
from g4f.client import Client
import mysql.connector
from datetime import datetime, timedelta
from flask import Flask, render_template, request, jsonify

client = Client()

# Конфігурація для підключення до бази даних
DB_CONFIG = {
    "host": "ip-address",  # Замість "ip-address" вказуємо реальну IP-адресу сервера бази даних
    "user": "login DB",  # Замість "login DB" вказуємо ім'я користувача для підключення
    "password": "password DB",  # Замість "password DB" вказуємо пароль користувача
    "database": "name DB"  # Замість "name DB" вказуємо назву бази даних
}

# Функція обробки логів і аналізу в залежності від типу аналізу
def process_log():
    data = request.get_json()
    log_message = data.get('logMessage', '')
    analysis_type = data.get('analysisType', 'general')

    # Ключові запити для різних типів аналізу
    analysis_queries = {
        "port_usage": f"Ось логи: {log_message}. Виведи наступну інформацію: Порти в логах. Для кожного порту надай короткий аналіз  у такому форматі:\n\nПорт: <номер>, <короткий опис>, Рівень загрози: <0-10>",
        "ip_analysis": f"Ось логи: {log_message}. Проаналізуй тільки IP-адреси, що згадані в логах. Для кожної IP-адреси виведи короткий аналіз у такому форматі:\n\nIP: <адреса>, IP address: <короткий опис>",
        "protocol_activity": f"Ось логи: {log_message}. Проаналізуй протоколи, що згадуються в логах. Виведи короткий аналіз по кожному протоколу у такому форматі:\n\nПротокол: <назва>, Активність: <короткий опис>, Рівень загрози: <0-10>",
        "recommendations": f"Ось логи: {log_message}. Надай короткі рекомендації для усунення цієї загрози у такому форматі:\n\nЗагроза: <назва або суть>, Рекомендація: <коротко>",
        "general_threats": f"Ось логи: {log_message}. Надай загальний аналіз логів у такому форматі:\n\nТип загрози: <назва>, Потенційний вплив: <коротко>, Рівень загрози: <0-10>"
    }

    query = analysis_queries.get(analysis_type)

    if log_message and query:
        response = client.chat.completions.create(
            model="gpt-4o-mini",
            messages=[{"role": "user", "content": query}],    # Відправляємо запит на аналіз
            web_search=False
        )
```

```
        # Якщо відповідь є, повертаємо її користувачеві
        if hasattr(response, 'choices'):
            full_reply = "".join(chunk.message.content for chunk in response.choices if
chunk.message.content)
            full_reply = full_reply.replace("[[Login to OpenAI ChatGPT]]()", "").strip()
            return jsonify({"success": True, "reply": full_reply})

        return jsonify({"success": False, "reply": "Помилка обробки відповіді."})

    return jsonify({"success": False, "message": "Немає логу для аналізу."})


# Функція витягування всіх логів з бази даних
def fetch_logs():
    try:
        conn = mysql.connector.connect(**DB_CONFIG)  # Підключення до БД
        cursor = conn.cursor()
        cursor.execute("SELECT * FROM logs")  # Отримуємо всі логи
        return cursor.fetchall()
    except mysql.connector.Error as err:
        print(f"Error: {err}")
        return []
    finally:
        if 'conn' in locals():
            conn.close()

# Функція отримання кількість логів по хвилинах для графіка
def fetch_logs_for_chart():
    try:
        conn = mysql.connector.connect(**DB_CONFIG)
        cursor = conn.cursor()
        cursor.execute("SELECT Timestamp FROM logs")  # Отримуємо час всіх логів
        logs = cursor.fetchall()

        log_counts = {}
        for log in logs:
            raw_timestamp = str(log[0])  # Перетворюємо в рядок
            try:
                # Конвертуємо ISO-формат (з "T" і мілісекундами) у звичайний datetime
                datetime_obj = datetime.strptime(raw_timestamp[:19], "%Y-%m-%dT%H:%M:%S")
            except ValueError:
                print(f"Помилка обробки часу: {raw_timestamp}")
                continue  # Пропускаємо некоректні записи

            minute_str = datetime_obj.strftime("%Y-%m-%d %H:%M")  # Групуємо по хвилинах
            log_counts[minute_str] = log_counts.get(minute_str, 0) + 1

        sorted_logs = sorted(log_counts.items())  # Сортуємо за часом

        return jsonify({"logs": sorted_logs})

    except mysql.connector.Error as err:
```

```
        print(f"Database Error: {err}")
        return jsonify({"error": "Database error"}), 500
    except ValueError as err:
        print(f"Format Error: {err}")
        return jsonify({"error": "Timestamp format error"}), 500
    finally:
        if 'conn' in locals():
            conn.close()

# Функція фільтрації логів за датою та ключовим словом
def filter_logs():
    data = request.get_json()
    start_date = data.get('startDate')
    end_date = data.get('endDate')
    keyword = data.get('keyword')

    query = "SELECT * FROM logs WHERE 1=1"
    params = []

    if start_date:
        query += " AND Timestamp >= %s"
        params.append(start_date)
    if end_date:
        query += " AND Timestamp <= %s"
        params.append(end_date)
    if keyword:
        query += " AND (Title LIKE %s OR Text LIKE %s)"
        keyword_pattern = f"%{keyword}%"
        params.extend([keyword_pattern, keyword_pattern])

    try:
        conn = mysql.connector.connect(**DB_CONFIG)
        cursor = conn.cursor()
        cursor.execute(query, params)
        filtered_logs = cursor.fetchall()
        return jsonify({"logs": filtered_logs})
    except mysql.connector.Error as err:
        print(f"Error: {err}")
        return jsonify({"error": "Database error"}), 500
    finally:
        if 'conn' in locals():
            conn.close()
```

project/static/js/scripts2.js

```
// Функція для відправки групованих алертів на сервер для обробки.
async function sendToG4F() {
    let message = groupedAttributes.map((attributes, index) => `Alert ${index + 1}:
${attributes.map(attr => `${attr.column}: ${attr.value}`).join(', ')}`).join('\n');
    if (!message) return alert('No alerts to send!');

    try {
```

```
        const response = await fetch('/process-log', { method: 'POST', headers: { 'Content-Type':
'application/json' }, body: JSON.stringify({ logMessage: message }) });
        const data = await response.json();
        document.getElementById('additionalInfoContent').innerText = data.success ? data.reply :
'Failed to process log.';
    } catch (error) {
        console.error('Error processing log:', error);
    }
}


// Функція для відправки вибраних логів на сервер для аналізу.
function analyzeLog() {
    const selectedLogs = getGroupedLogs();
    if (!selectedLogs.length) return alert("Ви не вибрали жодного алерту для аналізу.");

    fetch("/process-log", {
        method: "POST",
        headers: { "Content-Type": "application/json" },
        body: JSON.stringify({ logMessage: selectedLogs.join("\n---\n"), analysisType:
document.getElementById("analysisType").value })
    })
    .then(response => response.json())
    .then(data => {
    const container = document.getElementById("additionalInfoContent");
    container.innerHTML = "";

    if (!data.success) {
        container.innerText = "Помилка: " + data.reply;
        return;
    }

    const blocks = data.reply.split(/\n\s*\n/); // Розділити по блоках

    blocks.forEach(block => {
        const match = block.match(/Рівень загрози:\s*(\d+)/);
        let riskLevel = match ? parseInt(match[1]) : null;

        let highlightedBlock = block;
        if (riskLevel !== null) {
            let style = '';
            if (riskLevel <= 3) {
                style = 'background-color: #d4edda; border-left: 5px solid #28a745;';
            } else if (riskLevel <= 7) {
                style = 'background-color: #fff3cd; border-left: 5px solid #ffc107;';
            } else {
                style = 'background-color: #f8d7da; border-left: 5px solid #dc3545;';
            }

            highlightedBlock = block.replace(
                /Рівень загрози:\s*\d+/,
                `<span style="${style} padding: 2px 4px; border-radius: 4px;">Рівень загрози:
```

```
${riskLevel}</span>`
            );
        }

        const blockDiv = document.createElement("div");
        blockDiv.innerHTML = highlightedBlock;
        blockDiv.style.margin = "10px 0";
        blockDiv.style.whiteSpace = "pre-wrap";

        container.appendChild(blockDiv);
    });
})
```

ДОДАТОК 2

МАТЕРІАЛИ

VIII науково-практичної конференції курсантів (студентів), аспірантів, докторантів та молодих учених

“АКТУАЛЬНІ ПИТАННЯ ЗАСТОСУВАННЯ СПЕЦІАЛЬНИХ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СИСТЕМ”

18 червня 2025 року

Київ – 2025

УДК 621

Матеріали VIII науково-практичної конференції курсантів (студентів), аспірантів, докторантів та молодих учених “Актуальні питання застосування спеціальних інформаційно-комунікаційних систем”. Київ : ІСЗЗІ КПІ ім. Ігоря Сікорського, 2025. 516 с.

У матеріалах VIII науково-практичної конференції курсантів (студентів), аспірантів, докторантів та молодих учених “Актуальні питання застосування спеціальних інформаційно-комунікаційних систем” опубліковано тези доповідей, в яких висвітлюються питання дослідження, аналізу й узагальнення нових теоретичних і практичних результатів у сфері кібербезпеки, новітніх інформаційних технологій, штучного інтелекту та електронних комунікацій, підготовки офіцерів для сектору безпеки та оборони а також залучення здобувачів вищої освіти до активної наукової діяльності.

РЕЦЕНЗЕНТИ:

Олександр ПУЧКОВ	К.філос.н., професор
Сергій КОНЮШОК	К.т.н., доцент
Владислав ГОЛЬ	К.т.н., професор
Вадим РОМАНЕНКО	К.т.н., доцент
Дмитро МОГИЛЕВИЧ	Д.т.н., професор
Ірина КОНОВА	К.т.н., доцент

Рекомендовано до друку Вченою радою ІСЗЗІ КПІ ім. Ігоря Сікорського (протокол № 13 від 16.06.2025).

Секція № 4 “Новітні інформаційні технології та штучний інтелект в сфері кібербезпеки”

АВТОМАТИЗАЦІЯ ПРОЦЕСІВ КІБЕРЗАХИСТУ НА ОСНОВІ СУЧАСНИХ ІТ-РІШЕНЬ І ГЕНЕРАТИВНОГО ШТУЧНОГО ІНТЕЛЕКТУ

Анотація. Досліджено шляхи автоматизації процесів кіберзахисту із застосуванням сучасних ІТ-рішень і генеративного штучного інтелекту. Визначено переваги автоматизації та впровадження ШІ для підвищення ефективності й скорочення часу реагування на загрози.

Summary. Ways to automate cybersecurity processes using modern IT solutions and generative artificial intelligence are investigated. The advantages of automation and AI implementation for improving efficiency and reducing response time to threats are outlined.

Ключові слова: кібербезпека, кіберзахист, генеративний штучний інтелект, кіберзагрози, аналіз загроз.

У сучасному світі кіберзагрози стають дедалі складнішими, багаторівневими та чисельнішими. Вони характеризуються здатністю до швидкої еволюції тактик обходу та використанням розподілених методів атак, що ставить під сумнів ефективність традиційних механізмів захисту — сигнатурного аналізу, статичних індикаторів компрометації та фіксованих правил їхнього блокування. Такі підходи дедалі частіше демонструють низьку ефективність і схильність до хибнопозитивних спрацювань. У відповідь на ці виклики, рішення, засновані на новітніх ІТ-технологіях і генеративному штучному інтелекті, стають ключовими в побудові самонавчаючихся та проактивних систем кібербезпеки. Вони відкривають нові можливості для автоматизації процесів виявлення загроз, аналізу великих обсягів даних і швидкого реагування на кіберінциденти. З огляду на критичність часу реагування та точність виявлення, впровадження таких інновацій є важливим кроком до підвищення ефективності захисту інформаційних систем.

SIEM-системи, наприклад, такі як Wazuh, є одними з найбільш ефективних інструментів у контурі кіберзахисту. Wazuh дозволяє збирати, обробляти та аналізувати логи з різних джерел, таких як мережеві пристрої та сервери, що дозволяє своєчасно виявляти підозрілі активності та кіберінциденти.

Інтеграція з системами виявлення вторгнень (IDS) дозволяє значно розширити можливості моніторингу мережевого трафіку, що забезпечує підвищену точність і ефективність виявлення потенційних загроз. IDS-системи здійснюють постійний моніторинг мережі та аналізують трафік для виявлення аномальних патернів або підозрілих дій, що можуть свідчити про можливі кібератаки. Вони є важливим інструментом для своєчасного виявлення та реагування на загрози, що виникають у реальному часі.

Інтеграція таких систем з іншими платформами дозволяє ефективно з'єднувати різноманітні сервіси та інструменти, що значно спрощує автоматизацію процесів та забезпечує зручне управління робочими потоками.

Завдяки великій кількості доступних "ноудів" (nodes), кожен з яких є окремим інтеграційним модулем для популярних сервісів та API, ці платформи дають змогу налаштувати автоматичні дії, що виконуються при певних умовах або тригерах. Вони забезпечують безшовну інтеграцію між різними системами, дозволяючи без додаткових зусиль здійснювати обмін даними між ними, а також автоматизувати виконання рутинних завдань.

З іншого боку, одним із ключових аспектів у впровадженні генеративного ШІ є його здатність до обробки неструктурованих даних та створення інтелектуальних систем для виявлення та прогнозування кіберзагроз, пропонуючи стратегії їх нейтралізації. Це дозволяє ефективно планувати заходи щодо їх запобігання. Поєднання сучасних ІТ-рішень, генеративного штучного інтелекту та систем автоматизації створює новий рівень адаптивної та проактивної кібербезпеки, який здатен відповідати на виклики цифрової епохи.

Висновки: Впровадження генеративного штучного інтелекту в системи кіберзахисту має значний потенціал для підвищення їх ефективності, надійності та адаптивності. Поєднання ШІ з сучасними ІТ- рішеннями, такими як SIEM-системи (наприклад, Wazuh), IDS-системи та платформи автоматизації, дозволяє створювати системи кіберзахисту, які здатні швидко реагувати на загрози, аналізувати великі обсяги даних і прогнозувати сценарії атак. Це суттєво зменшує навантаження на фахівців з кібербезпеки, знижує ризик людської помилки та забезпечує високий рівень захисту інформаційної інфраструктури в умовах постійної еволюції кіберзагроз. Подальший розвиток у цьому напрямку відкриває перспективи створення автономних систем, здатних не лише виявляти, а й ефективно нейтралізовувати загрози в реальному часі.

ДОДАТОК 3

УКРАЇНА

СВІДОЦТВО

про реєстрацію авторського права на твір

№ 137641

Комп'ютерна програма «Artificial Intelligence Logix» («AIL»)

(вид, назва твору)

Автор (співавтори) **“БЕТЕЛЬГЕЙЗЕ”, Пучков Олександр Олександрович, Микитюк Артем В'ячеславович, Ланде Дмитро Володимирович**

(прізвище, ім’я, по батькові (за наявності), псевдонім (за наявності))

Авторські майнові права належать спільно **“БЕТЕЛЬГЕЙЗЕ”; Пучков Олександр Олександрович, просп. Валерія Лобановського, 4-Г, кв. 40, м. Київ, 03037; Микитюк Артем В'ячеславович, вул. Максима Залізняка, 3, м. Київ, 03142; Ланде Дмитро Володимирович, пров. Панаса Мирного, 4, кв. 19, м. Київ, 01011**

(прізвище, ім’я, по батькові (за наявності) фізичної особи / найменування юридичної особи, адреса)

Дата реєстрації 2 липня 2025 р.

Директор Державної організації «Український національний офіс інтелектуальної власності та інновацій»

Олена ОРЛЮК

УКРНОІВІ