Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
В. о. завідувач кафедри
кібербезпеки та захисту
інформації
_____ Іван ПАРХОМЕНКО
«13» червня 2025 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи

галузь знань 12 Інформаційні технології
(шифр і назва галузі знань)
спеціальність 125 «Кібербезпека»
(код і назва спеціальності)
освітній ступень бакалавр
освітня програма Кібербезпека
(назва освітньо-професійної програми)

на тему: «Система автоматизованого реагування на інциденти для SIEM системи»

Виконавець: студент IV курсу, групи КБ-42

_____ **Владислав ЗОЛОТАРЬОВ**
(підпис) (ім'я прізвище)

	Підпис	**Ім'я, прізвище**
Керівник роботи		**Інна МИХАЛЬЧУК**
Нормоконтроль		**Юрій ЩЕБЛАНІН**

Київ 2025

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
В. о. завідувач кафедри кібербезпеки та захисту інформації
_____ Іван ПАРХОМЕНККО
«29» листопада 2024 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

спеціальності 125 Кібербезпека
(код і назва спеціальності)

освітньої програми Кібербезпека
(назва освітньої-професійної програми)

студенту **КБ-42** **Золотарьову Владиславу Вацлавовичу**
(група) (прізвище ім'я по батькові)

Тема кваліфікаційної роботи Система автоматизованого реагування на інциденти для SIEM-системи

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Тема кваліфікаційної роботи затверджена на засіданні кафедри кібербезпеки та захисту інформації протокол №6 від 28.11.2024 р.

2. ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Лог-файли з тестового середовища, структура подій у відповідному форматі, принципи роботи SIEM-систем та сценарії реагування, принципи взаємодії з Elasticsearch, бібліотеки Python для обробки логів і автоматизації дій.

3. ЗМІСТ РОЗРАХУНКОВО-ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Огляд методів автоматизації реагування в SIEM-системах, формалізація узагальненої моделі модуля реагування, опис рхітектури Elastic Stack, розробка сценаріїв автоматизованих дій, реалізація та тестування модулів на Python.

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Практична цінність Розроблена система автоматизованого реагування підвищує ефективність роботи аналітиків, забезпечує швидке виявлення та нейтралізацію кіберінцидентів, знижує навантаження на аналітиків безпеки і мінімізує час реагування на загрози. Рішення легко інтегрується у наявну інфраструктуру та сприяє покращенню загального рівня кіберзахисту організації.

5. ДАТА ВИДАЧІ ЗАВДАННЯ

Дата видачі завдання: 29 листопада 2024 року

Завдання видала	(підпис)	Інна МИХАЛЬЧУК (ім'я, прізвище)
Завдання прийняв до виконання	(підпис)	Владислав ЗОЛОТАРЬОВ (ім'я, прізвище)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів робіт	Строки виконання робіт (початок-кінець)	Відмітка про виконання
1	Уточнення постановки задачі	29.11.2024 –19.11.2024	*виконано*
2	Аналіз літератури	20.11.2024 –24.12.2024	*виконано*
3	Обґрунтування вибору рішення	24.12.2024 –28.12.2024	*виконано*
4	Огляд архітектур SIEM-систем та типових підходів до автоматизації	02.01.2025 –15.01.2025	*виконано*
5	Дослідження функціоналу популярних рішень SIEM систем	17.01.2025 –06.02.2025	*виконано*
6	Розробка сценаріїв автоматизованого реагування на інциденти	07.01.2025 –15.02.2025	*виконано*
7	Вибір та дослідження бібліотек для реалізації модулів	16.02.2025 –20.02.2025	*виконано*
8	Створення тестового середовища та підключення систем збору логів	26.02.2025 –11.03.2025	*виконано*
9	Реалізація модуля автоматизованого реагування з базовими сценаріями	14.03.2025 –10.04.2025	*виконано*
10	Проведення тестування, аналіз ефективності, доопрацювання модуля	11.04.2025 –15.05.2025	*виконано*
11	Оформлення пояснювальної записки	12.05.2025 –21.05.2025	*виконано*
12	Підготовка до захисту кваліфікаційної роботи	22.05.2025 – 13.06.2025	*виконано*

Завдання видала _____ Інна МИХАЛЬЧУК
(підпис) (ім’я, прізвище)

Завдання прийняв до виконання _____ Владислав ЗОЛОТАРЬОВ
(підпис) (ім’я, прізвище)

Терміни подання кваліфікаційної роботи до ЕК 13 червня 2025 року

РЕФЕРАТ

Кваліфікаційна робота складається зі вступу, трьох розділів, загальних висновків, списку використаних джерел, додатків, має 66 сторінок основного тексту, 32 рисунка та 2 таблиці. Робота містить 3 додатки із загальною кількістю сторінок - 6. Список використаних джерел містить 25 найменувань та займає 3 сторінки.

Темою кваліфікаційної роботи є розробка системи автоматизованого реагування на інциденти для SIEM-системи.

Метою роботи є розробка та впровадження програмних модулів для автоматизованого реагування на кіберінциденти в рамках SIEM-системи з використанням відкритих технологій, з метою підвищення ефективності реагування на кіберзагрози та зменшення навантаження на *SOC* аналітиків.

Об'єктом дослідження є системи виявлення та реагування на інциденти інформаційної безпеки в рамках SIEM-платформ.

Предмет дослідження є методи та засоби автоматизації реагування на кіберінциденти з використанням відкритих технологій та мови програмування Python у контексті SIEM-систем.

Методи дослідження кваліфікаційної роботи: експериментальне дослідження та програмна реалізація, застосовані для розробки та оцінки ефективності модулів автоматизованого реагування на кіберінциденти в межах SIEM-систем з використанням Python.

Практична цінність полягає у створенні модулів, які автоматизують реагування на типові кіберінциденти із використанням Python та стеку Elastic Stack. Запропоноване рішення може бути адаптоване до реальної інфраструктури.

Ключові слова: інформаційна безпека, SIEM, автоматизація, інцидент, реагування, Elastic Stack, лог-файли.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

API	- Application Programming Interface
GDPR	- General Data Protection Regulation
IoC	- Indicators of Compromise
MITRE ATT&CK	- Adversarial Tactics, Techniques, and Common Knowled
UFW	- Uncomplicated Firewall
SIEM	- Security Information and Event Management
SOAR	- Security Orchestration, Automation, and Response
ВМ	- Віртуальні машини
SOC	- Security Operation Center
EDR	- Endpoint Detection and Response
NTA	- Network Traffic Analysis

ЗМІСТ

ВСТУП

У сучасних реаліях, де більшість сфер життя напряму пов’язані з цифровими технологіями, інформаційна безпека стає одним із найважливіших пріоритетів будь-якої компанії, незалежно від її масштабу — особливо, якщо її прибуток безпосередньо залежить від репутації. Кібератаки стають усе винахідливішими та складнішими, їхня кількість зростає, а наслідки можуть бути катастрофічними — від втрати даних до порушення критично важливих процесів. Традиційні способи обробки даних вручну вже не є настільки ефективними, оскільки, навіть за своєчасного виявлення системою підозрілої активності, загрозу може бути недостатньо швидко ідентифіковано та нейтралізовано належними заходами безпеки.

Одним із головних засобів, що застосовуються в кібербезпеці, є SIEM-системи [1]. Технологія *SIEM* збирає дані журналів подій із різних джерел і виявляє нетипові дії за допомогою аналізу в реальному часі. Однак на практиці цього інструменту недостатньо для оперативного реагування, оскільки працівникам *SOC* щодня доводиться вручну перевіряти понад сотню сповіщень. Це призводить до помилок і пропуску реальних загроз, причиною чого можуть стати втома, людський фактор або навіть “сліпота уваги” — коли через велику кількість хибнопозитивних тривог працівник стає нечутливим до справжньої загрози. Ще однією проблемою, особливо для малих і середніх компаній, є висока собівартість впровадження SOAR-платформ. У цьому контексті особливої актуальності набуває впровадження рішень для автоматизації окремих процесів, що сприятиме підвищенню загальної ефективності системи кіберзахисту, зниженню навантаження на фахівців та забезпеченню прийнятного рівня результативності за умов обмежених ресурсів.

У цій роботі основну увагу зосереджено на вирішенні вищезазначених проблем. Метою є створення доступного рішення, яке, на відміну від комерційних

продуктів, не потребує значних інвестицій чи ліцензійних витрат і може бути адаптоване під потреби конкретної організації. Пропонується розглянути альтернативу, розроблену на основі безкоштовних інструментів. Основна ідея полягає в інтеграції спеціально створеного скрипту на мові Python із *SIEM*-системою. Це дозволяє автоматизувати окремі процеси, зокрема блокування IP-адрес, надсилання сповіщень про загрози тощо — без безпосереднього втручання людини, залишаючи лише подальшу обробку для фахівців. Запропоноване рішення буде протестовано за допомогою симульованих атак, що дозволить експериментальним шляхом перевірити його ефективність та функціональність в умовах обмежених ресурсів. У межах цієї роботи робиться спроба довести, що така розробка може слугувати основою для створення бюджетних, але водночас ефективних рішень у сфері кібербезпеки в майбутньому.

РОЗДІЛ 1
ТЕОРЕТИЧНІ ОСНОВИ АВТОМАТИЗОВАНОГО РЕАГУВАННЯ НА КІБЕРІНЦИДЕНТИ

1.1 Роль та структура SIEM-систем

Сучасна сфера кібербезпеки постійно еволюціонує, і, відповідно, кібератаки стають дедалі винахідливішими, використовуючи новітні технології. Це зумовлює потребу в адаптації систем захисту шляхом впровадження інноваційних інструментів, здатних не лише забезпечити своєчасне виявлення інцидентів, а й здійснювати аналіз потенційних ризиків.

SIEM-системи (Security Information and Event Management) є ключовим інструментом у протидії кібератакам. Вони поєднують функції збору інформації з різних джерел, обробки та аналізу даних, логування подій і формування звітів. Основна перевага таких систем полягає у здатності трансформувати неструктуровані події в узгоджену та зручну для аналізу інформацію, що дозволяє фахівцям своєчасно ідентифікувати загрози на ранніх етапах їх реалізації та вжити необхідних заходів реагування.

Головна цінність SIEM-систем полягає у здатності агрегувати журнали подій з різнорідних компонентів IT-інфраструктури, таких як мережеві пристрої, сервери, кінцеві точки та програмні додатки, а також у можливості нормалізувати ці дані до єдиного уніфікованого формату.

Функціонування SIEM-систем базується на методах кореляції та інтеграції даних з інших систем кібербезпеки, зокрема з використанням фреймворку *MITRE ATT&CK* [2]. Це дозволяє систематизувати та аналізувати тактики, техніки та процедури сучасних кібератак. Наприклад, підозріла активність під час автентифікації, зокрема значне зростання кількості невдалих спроб входу, може свідчити про атаку методом перебору паролів. У таких випадках можливо

автоматизувати запуск сценаріїв реагування — зокрема, блокування IP-адреси, надсилання сповіщення адміністратору або ініціювання додаткових перевірок.

Однак ефективність SIEM як інструменту виявлення безпосередньо залежить від якості конфігурації правил та інтеграції з іншими компонентами кібербезпеки, такими як система управління вразливостями або платформа розвідки загроз. Відповідно, якщо правила кореляції недостатньо адаптовані до особливостей конкретної організації, вони часто генерують велику кількість хибнопозитивних тривог, що, у свою чергу, призводить до перенавантаження аналітиків *SOC* і негативно впливає на загальну ефективність системи. Згідно з дослідженням SANS Institute (2022), середня кількість оповіщень *SIEM* на день для організації середнього розміру становить 10–15 тисяч, з яких лише 5–8 % класифікуються як критичні [3]. Це підкреслює необхідність ретельного відбору та оптимізації механізмів кореляції для зменшення навантаження та правильної пріоритизації подій.

На даний момент SIEM-рішення включають низку модулів, які збирають інформацію з логів і подій інфраструктурних компонентів, трансформуючи її у практичні інсайти щодо кібербезпеки. Для реалізації цієї мети *SIEM* використовує такі функціональні елементи:

1. *Накопичення даних* — збір та об’єднання інформації з хост-систем, мережевих пристроїв тощо.
2. *Інтеграція з джерелами розвідки загроз* — поєднання внутрішніх даних про вразливості з зовнішніми індикаторами загроз.
3. *Кореляція подій та моніторинг ризиків* — встановлення зв’язків між подіями, інцидентами та потенційними загрозами.
4. *Аналітична обробка* — застосування алгоритмів машинного навчання та статистичних методів для виявлення патернів у даних.
5. *Генерація сповіщень* — автоматичне оповіщення про аномальні події на основі аналізу.

6. *Інтерактивні панелі* — візуалізація даних у реальному часі, що спрощує виявлення тенденцій і відхилень для адміністраторів.

7. *Контроль відповідності* — агрегація логів відповідно до стандартів безпеки (наприклад, *GDPR* [4], ISO 27001) із подальшим формуванням звітів.

8. *Архівація* — довгострокове зберігання даних для аудиту.

9. *Форензика* — детальний аналіз інцидентів для відтворення їхньої хронології та виявлення причин.

10. *Проактивний пошук загроз* — можливість адміністраторів виконувати запити до логів для виявлення прихованих ризиків.

11. *Управління інцидентами* — механізми для оперативного виявлення та нейтралізації загроз.

12. Автоматизація *SOC* передбачає інтеграцію з системами кібербезпеки для забезпечення автоматизованого реагування на інциденти.

1.2 Визначення та функціональні можливості SIEM

SIEM (Security Information and Event Management) визначається як клас програмних рішень, які призначені для об'єднання, аналізу та інтерпретації даних про події безпеки з метою виявлення, дослідження та мінімізації кіберзагроз. В своїй архітектурі *SIEM* має дві компоненти, які доповнюють одне-одного: SEM (Security Event Management), що відповідає за моніторинг подій у режимі реального часу, та SIM (Security Information Management), що відповідає за довгострокове зберігання, нормалізацію та аналіз історичних логів. Тобто взаємодія цих компонентів дозволяє трансформувати сирі дані в інформацію, придатну для аналізу та прийняття оперативних рішень у разі потреби. Основна архітектура *SIEM* рішень графічно зображеена на рис. 1.1.

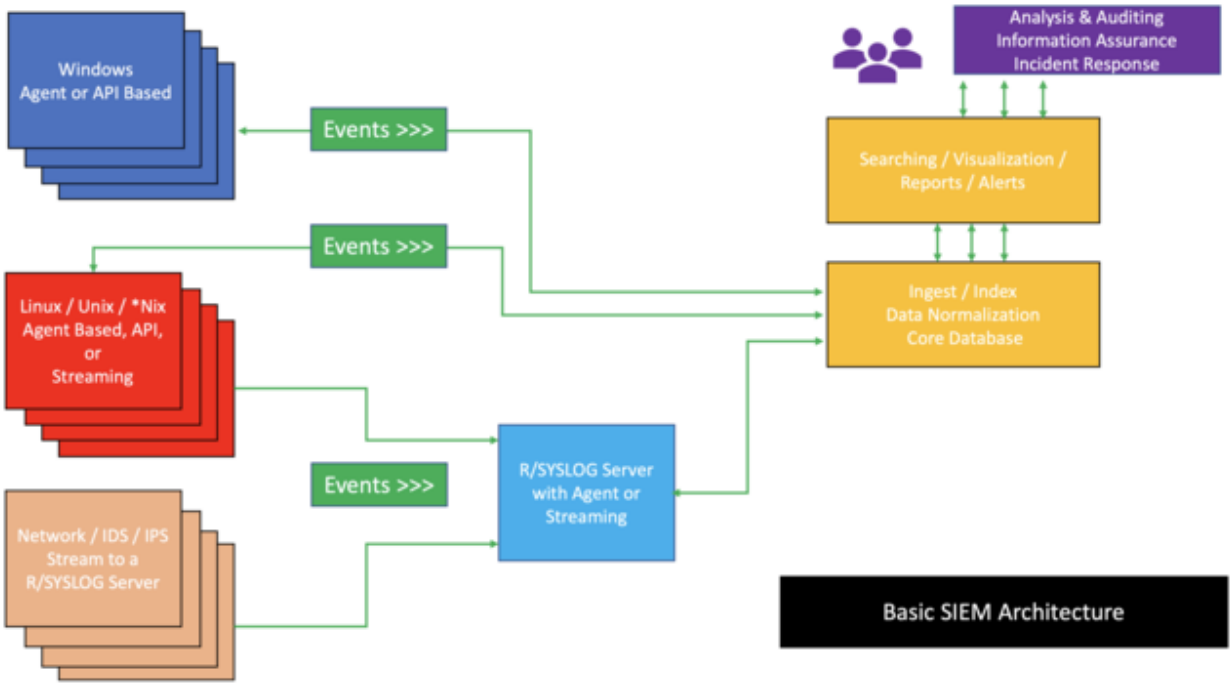

Рисунок 1.1 - Основна архітектура SIEM рішень

До ключових функціональних можливостей *SIEM* належать:

1. *Збір та нормалізація даних* — інтеграція з різноманітними джерелами (фаєрволами, IDS/IPS-системами, ERP-рішеннями) за допомогою підтримки відповідних протоколів (Syslog, SNMP, REST API [5]) з подальшою трансформацією логів до уніфікованого формату. Наприклад, журнали автентифікації з Active Directory та дані мережевого трафіку з фаєрволу конвертуються у структуровані поля, такі як «timestamp», «source IP», «event type», «destination IP» тощо.

2. *Кореляція подій* — застосування правил, що базуються на заданих логічних умовах, для виявлення шаблонів, потенційно пов'язаних із кіберзагрозами. Наприклад, послідовність подій на кшталт «невдала спроба входу → успішна автентифікація з нетипової геолокації → масове копіювання даних» може автоматично ініціювати сповіщення про потенційну компрометацію облікових даних (наприклад, Credential Stuffing).

3. Аналіз поведінки (UEBA): використання машинного навчання для побудови профілів кожного з користувачів та пристроїв з подальшим виявленням аномалій на основі минулих подій, таких як незвичний час активності або доступ до ресурсів із підвищеними привілеями.

4. Комплаєнс-менеджмент: автоматизація створення звітів для аудиту (наприклад, згідно з вимогами PCI DSS [6] або *GDPR*), що включає візуалізацію порушень, часові індикатори реакції та відстеження доступу до певних конфіденційних даних.

Одним з важливих аспектів, є можливість інтеграції *SIEM* із зовнішніми системами, платформами Threat Intelligence (на кшталт MISP або Alien Vault), що надає можливість оцінки за показниками компрометації (IoCs) - хешів шкідливого програмного забезпечення, IP-адрес ботнетів або інших якостей, які вони можуть набувати. Тобто система *SIEM* буде спиратися на метрики, що зберігаються в базі даних *IOC*, і в разі успішного збігу негайно зреагує відповідним чином. Наприклад, якщо *SIEM* виявить з'єднання з сервером, позначеним в Threat Intelligence як частина інфраструктури Threat Intelligenc, система автоматично підвищить серйозність події та запустить процес ізоляції інфікованого хоста.

Ефективність SIEM-системи є прямопропорційною до якості її налаштування. Згідно з даними дослідження Gartner (2023), приблизно 70% невдалих впроваджень SIEM-рішень пов’язані з неякісно налаштованими правилами кореляції або відсутністю адаптації системи до специфіки конкретного бізнесу. Унаслідок цього спостерігається надмірна уніфікація шаблонів, що, своєю чергою, призводить до пропуску атак, які експлуатують унікальні вразливості певної організації.

Крім того, одним із ключових обмежень традиційних SIEM-систем є їхня орієнтованість на реактивний підхід. Наприклад, виявлення атаки часто відбувається лише після її початку, на основі аналізу вже зафіксованих журналів подій. У той час як проактивні механізми, зокрема прогнозування потенційних

векторів атак на основі поточних вразливостей, залишаються недостатньо інтегрованими в основну логіку таких систем.

Це підкреслює необхідність розширення функціональності SIEM-рішень за рахунок інтеграції з додатковими інструментами кібербезпеки, такими як *EDR* чи *NTA*. Такі компоненти забезпечують глибший контекст для аналізу інцидентів і дозволяють системі не лише фіксувати події, але й будувати більш повну картину загроз у режимі реального часу.

З огляду на викладене, функціональність SIEM-систем не обмежується лише пасивним збором та агрегацією даних, а трансформується у потужний інструмент стратегічного управління ризиками, де кожен функціональний компонент відіграє критично важливу роль у формуванні цілісної картини кіберзагроз. Її ефективність залежить не лише від технічних характеристик, а й від здатності органічно інтегруватися в операційні процеси організації, що робить SIEM-рішення невід'ємною складовою побудови сучасної архітектури кібербезпеки.

1.3 Порівняльний аналіз сучасних SIEM-рішень

Вибір адаптованої SIEM-платформи для організації є критично важливим. Основні фунції *SIEM* зображенні на рис.1.2. Від вибору залежить не лише ефективність детекції загроз, а й операційна стійкість SOC. Різниця між сучасними рішеннями (Elastic, Splunk Enterprise Security, IBM Qradar та іншими) спостерігається за рахунок абсолютно різних архітектурних підходів, прийнятої моделі ліцензування, глибини інтеграції з екосистемою, можливості кастомізації.

Рисунок 1.2 - Основні процеси утворення SIEM архітектури

SIEM-рішення збирають дані з різних систем, пристроїв і додатків в єдиний загальний формат. Всі ці дані нормалізуються, а потім проходять через відповідний механізм політиків.

На рис. 1.3 основним компонентом SIEM-системи виступає модуль, що відповідає за фільтрацію та обробку даних, моніторинг журналів подій, застосування правил кореляції та генерацію сповіщень. Крім того, цей елемент виконує передачу логів до сховища даних і надсилає оброблену інформацію на рівень представлення (Presentation Layer), де вона візуалізується та аналізується аналітиками з кібербезпеки.

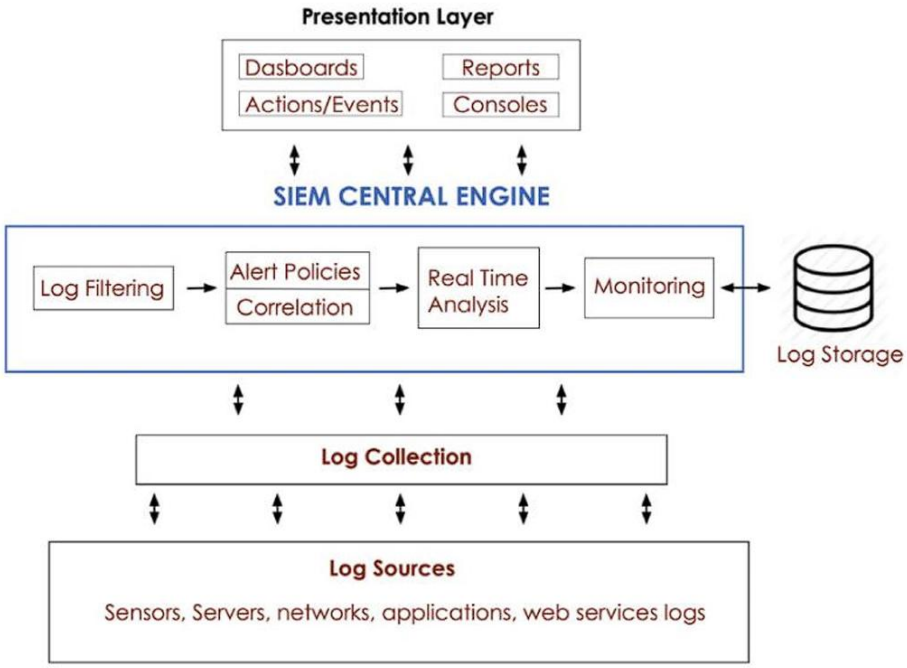

Рисунок 1.3 - Фізична архітектура SIEM

Структура обробки модулів SIEM-системи включає такі основні елементи:

- Джерело логів — компонент, що генерує журнали подій; ним може бути будь-який пристрій або система, здатна фіксувати дані (наприклад, сенсори, маршрутизатори, сервери, комутатори тощо).
- Агрегація журналів — процес трансформації отриманих логів у уніфікований формат для подальшого аналізу.
- Обробка даних — етап, на якому зібрана інформація передається до централізованого сховища SIEM-системи.
- Кореляція подій за заданими правилами — механізм формування та прив’язки сповіщень до відповідних подій згідно з визначеними умовами, що дозволяє суттєво знизити кількість хибнопозитивних спрацювань.

На сучасному ринку представлено широкий спектр SIEM-рішень, що відрізняються за функціональністю, рівнем автоматизації та вартістю впровадження. Один із методів оцінювання таких рішень — "Магічний квадрант

Гартнера" (Gartner Magic Quadrant) зображений на рис.1.4 [7]. Це аналітичний інструмент, що ґрунтується на комплексному дослідженні певного ринкового сегмента і дозволяє сформувати загальну картину конкурентоспроможності постачальників. Наприклад, у звіті за 2018 рік представлено ключових гравців ринку SIEM-рішень та їх позиціонування відповідно до візії розвитку та здатності до реалізації.

Рисунок 1.4 — Квадрант Гартнера для SIEM-систем

Splunk

Splunk — це рішення для збору, накопичення, обробки та аналізу машинно згенерованих даних (логів). На сьогодні платформа займає провідні позиції на ринках США та Європи й поступово розширює свою присутність в інших регіонах [8]. Її ключова перевага — універсальна сумісність: система може аналізувати інформацію від будь-яких джерел, що робить спектор її застосування практично необмеженим.

Особливості роботи:

1 Структурування даних: вхідні записи розділяються на поля та значення для подальшої роботи.

2 SPL-запити (власна мова запитів Splunk): дозволяє:

а. створювати вибірки та аналітичні звіти;

б. виконувати сортування, фільтрацію, агрегацію даних;

в. генерувати динамічні поля на основі обчислень;

г. інтегруватися з зовнішніми довідниками;

д. будувати інтерактивні візуалізації (графіки, діаграми, мапи);

е. налаштовувати автоматичні сповіщення про аномалії.

Результати аналізу можна інтегрувати в індивідуалізовані інтерфейси (див. рис. 1.5), що спрощує моніторинг для користувачів.

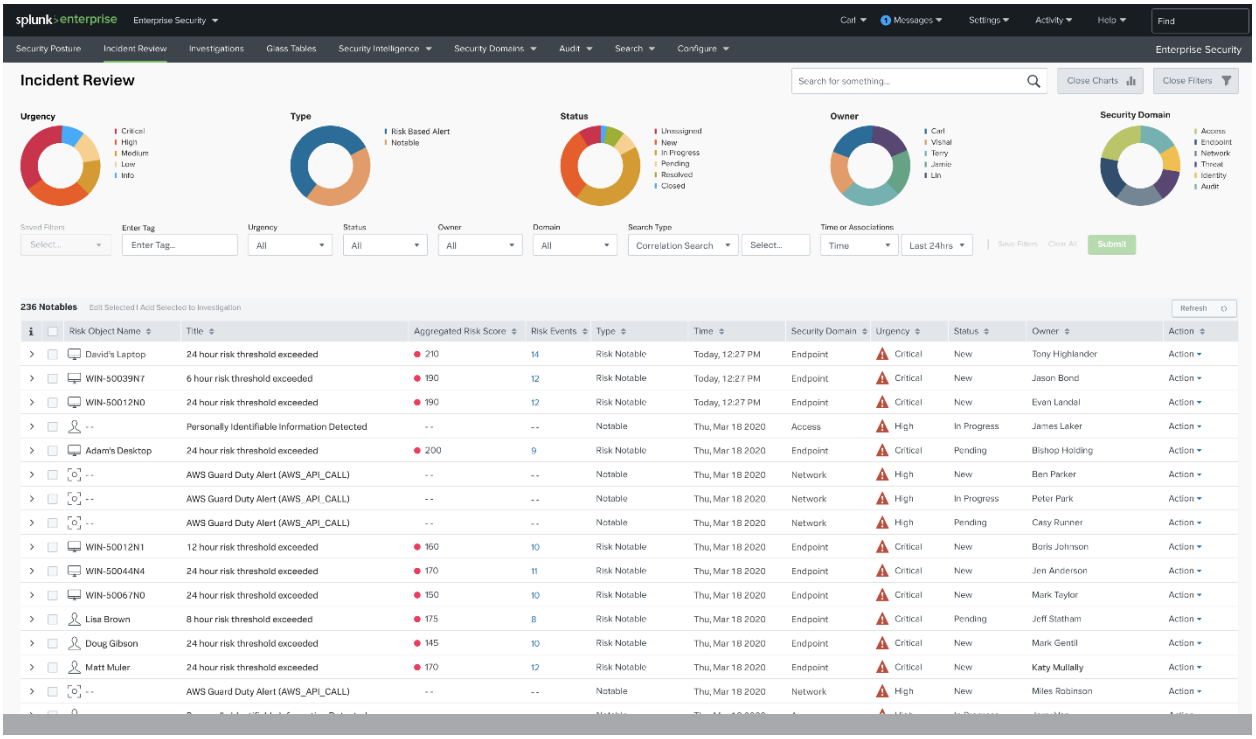

Рисунок 1.5 – Скрін вкладки аналітики по інцидентам інструменту Slpunk

Переваги:

1. Потужний інструмент для аналізу інформації: Забезпечує глибокий аналіз даних завдяки розширеним функціональним можливостям.

2. Інтуїтивно зрозуміла інформаційна панель із розширеною візуалізацією: Надає зручний інтерфейс для моніторингу ключових показників у реальному часі.

3. Простота використання аналізатора полів: Інтерфейс роботи з полями даних не потребує спеціальної технічної підготовки.

4. Висока продуктивність при обробці масштабних даних: Оптимізований для ефективної роботи з великими масивами інформації.

Недоліки:

1 Затримки при виконанні складних запитів: Деякі операції потребують значних ресурсів.

2 Складність інтеграції з нестандартними джерелами даних: Вимагає додаткових налаштувань для роботи з логами у спеціалізованих сховищах.

3 Преміальна цінова політика: Вартість ліцензій часто недоступна для невеликих компаній.

4 Орієнтація на корпоративний ринок: Рішення найкраще підходить для великих організацій зі складними потребами.

McAfee Enterprise Security Manager

McAfee Enterprise Security Manager все ще займає, одну з ключових позицій на ринку кібербезпеки, поступаючись лише IBM, Splunk та LogRhythm [9]. Його вважають досить гарним рішенням, для організацій, які хочуть отримати максимальний функціонал ‘з коробки’. Крім цього він ідеально підходить для користувачів інших продуктів McAfee завдяки власним інтеграціям та сумісності.

Особливості роботи:

McAfee ESM підтримує близько 430 різних джерел даних одразу після встановлення. Нові конектори досить часто додаються.

Дуже гарна пропускна здатність, тобто він може отримувати і обробляти одразу численну кількість подій, його архітектура розроблена для горизонтального масштабування через, що він може забезпечити практично безлімітну кількість прийнятих запитів.

Також користувачі, зазвичай зазначають дуже якісну підтрмку. Тобто, завжди є можливість, зв’язатися з предстаниками, щоб отримати технічну допомогу. Також доступний онлайн-портал обслуговування клієнтів кількома мовами.

На мою думку він займає таку високу позицію здебільшого через можливості масштабування - хости можна додати практично в будь-який момент, щоб покращити приймання даних, продуктивність запитів та резервування. Зображення інтерфейсу McAfee на рис. 1.6.

Рисунок 1.6 Вигляд McAfee Enterprise Security Manager

Також дуже важливим є здійснення безперервного спостереження за корпоративними інформаційними системами, накопичування даних про кіберзагрози та потенційні небезпеки. Цей інструмент надає можливість визначати пріоритетність загроз і прискорювати розслідування інцидентів, що є критичним для оперативного реагування. McAfee ESM ефективно взаємодіє з рішеннями інших виробників без залучення додаткових інтерфейсів, забезпечуючи сумісність із провідними системами у сфері кібербезпеки. Крім того, платформа інтегрується з McAfee Global Threat Intelligence, яка посилює стандартні можливості SIEM, завдяки чому ESM забезпечується доступом до актуальних відомостей про глобальні кіберзагрози. Практичним результатом є, наприклад, здатність ідентифікувати активність, пов'язану з компрометованими мережевими адресами.

Узагальнюючи викладене, даному рішенню властиві такі переваги як:

- Можливість співвідношення різних подій, які надійшли з різних платформ
- Є адаптивний режим навчання
- Дуже хороша технічна підтримка

До недолікв наведених систем можна віднести:

- Необхідність у досить великих потужностях, для комфортної роботи
- Велика кількість помилок при роботі
- Можливість застосування, лише для роботи з Windows або MAC.

IBM QRADAR

SIEM-платформа від IBM, технологічного лідера, посідає провідні позиції на ринку, що підтверджується її десятирічним безперервним перебуванням у квадранті лідерів Gartner, де вона неодноразово випереджала конкурентів [10]. Це рішення, інтегроване з низкою взаємопов'язаних систем, забезпечує максимально повне відстеження подій у мережі, оскільки його функціональність, доступна вже на етапі базової конфігурації, дозволяє охопити всі критичні аспекти кібербезпеки.

Платформа демонструє значну універсальність у зборі даних, аналізуючи інформацію від різноманітних джерел — від операційних систем і засобів захисту до баз даних, програмних рішень та інших компонентів інфраструктури. Інтерфейс зображено на рис. 1.7.

Переваги система відрізняється розширеною підтримкою понад 400 типів джерел журналів, що забезпечує гнучкість у зборі даних, а також наявністю DSM-редактора, який дозволяє користувачам аналізувати події відповідно до індивідуальних потреб. Окрім цього, вона інтегрується з системами управління вразливостями та оцінки ризиків, автоматизуючи процеси виявлення загроз, і пропонує вбудовані правила та шаблони звітів для ОС Windows, що спрощує адміністрування.

Недоліки незважаючи на широкий функціонал, платформа має низку обмежень. Наприклад, відсутня сумісність із TSM, що ускладнює її використання в певних інфраструктурах. Крім того, окремі пошукові запити потребують додаткового налаштування через недостатньо інтуїтивний синтаксис, а неможливість експорту документації з менеджера вразливостей обмежує оперативність роботи з результатами аналізу.

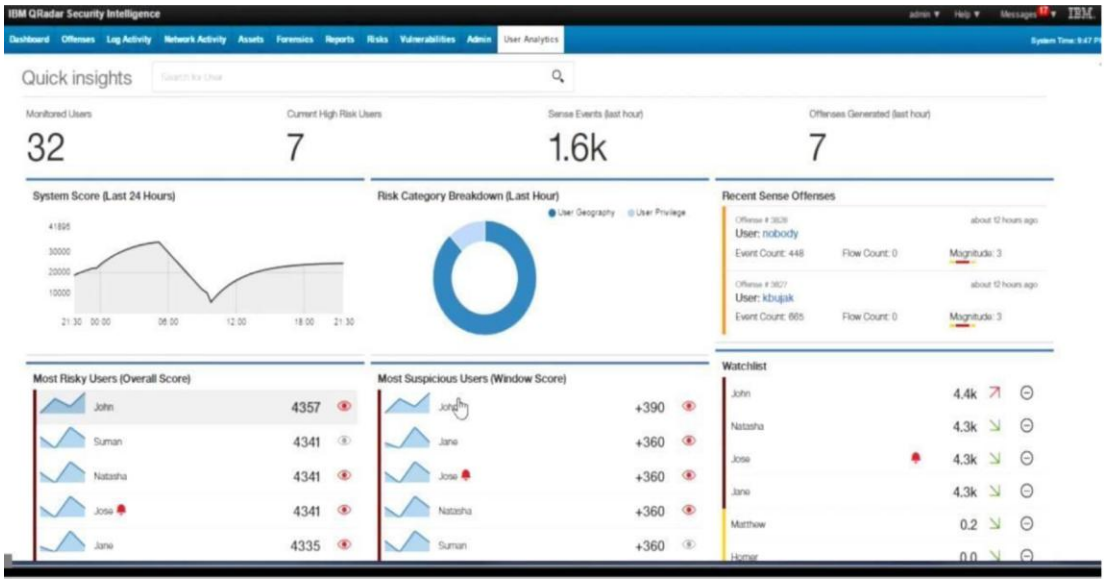

Рисунок 1.7 Сторінка графічні аналітики IBM QRadar

Ключовою характеристикою IBM QRadar Security Intelligence є здатність ідентифікувати та ранжувати загрози на основі оцінки ризиків, що реалізується завдяки поєднанню поглибленої аналітики й автоматизованої кореляції між критичними елементами інфраструктури — активами, користувачами, мережевою активністю, наявними вразливостями та зовнішніми індикаторами загроз. Оскільки система аналізує взаємозв'язки між цими компонентами, вона формує цілісну картину кіберзагроз, трансформуючи розрізнені події в логічні ланцюги, кожен з яких відповідає конкретному інциденту. Такий підхід дозволяє не лише визначати пріоритетність реагування, але й створювати індивідуальні процеси обробки для кожного випадку, що суттєво підвищує ефективність кібербезпеки.

Elastic Stack

Elastic Stack представляє собою набір інструментів з відкритим кодом, розроблених для спрощення пошуку, аналізу та візуалізації даних, які надходять із різноманітних джерел у реальному часі [11]. Ця платформа, що поєднує пошукову систему, генератор журналів, веб-інтерфейс та інструменти для передачі даних, відрізняється відкритістю архітектури, оскільки більшість її компонентів є безкоштовними та доступними для модифікації. Винятком є лише Elasticsearch, який вимагає ліцензії для повноцінного використання.

У 2019 році до сімейства додали Elastic SIEM, що являє собою офіційне SIEM-рішення від компанії Elastic, призначене для аналізу подій безпеки, пов'язаних із хостами та мережею. Воно дозволяє проводити розслідування через автоматичні сповіщення або інтерактивний пошук, проте, через пізній вихід на ринок, не було включено до квадранту Гартнера, що обмежує його видимість серед аналогічних пропозицій.

Важливою перевагою Elastic Stack є оптимізована продуктивність, економія апаратних ресурсів та інтуїтивність, що робить його доступним навіть для користувачів без глибоких технічних знань. Наприклад, Logstash виконує функцію

сервера для агрегації та обробки логів, тоді як Elasticsearch, як потужний пошуковий інструмент, спеціалізується на аналізі великих обсягів журналів подій.

Beats, як легкий агент, забезпечує передачу інформації до системи, а Kibana, виступаючи інтерфейсом, не лише візуалізує дані, але й інтегрується зі сторонніми інструментами збору, що пояснює її популярність серед фахівців. Попри те, що Kibana, Logstash та Beats є відкритими та безоплатними, їх можна комбінувати як між собою, так і з рішеннями інших розробників, що розширює сферу застосування платформи.

Elastic Stack (інтерфейс одного з компонентів зображено на рис. 1.8) надається на умовах вільного використання у формі локально встановленого програмного забезпечення, хоча для організацій, які потребують професійної підтримки, передбачені комерційні пакети з розширеним функціоналом. Elastic SIEM, для свого повноцінного застосування потребує підключення до інших елементів стеку. Водночас усі продукти Elastic, пропоновані як хмарні сервіси за моделлю SaaS, вимагають обов’язкової оплати, оскільки їхня архітектура не передбачає безкоштовного доступу, що обмежує їхнє використання для користувачів з обмеженим бюджетом.

Рисунок 1.8 – використання Kibana для візуалізації подій

Система відрізняється гнучкістю архітектури, оскільки її компоненти можуть бути впроваджені окремо, що дозволяє адаптувати рішення під специфічні потреби користувача. Проте існують певні обмеження, оскільки частина модулів закрита, що ускладнює їхнє модифікацію розробниками.

Потужна пошукова система Elastic Stack, здатна ефективно обробляти великі обсяги даних, проте безкоштовна версія не включає технічної підтримки, що змушує користувачів покладатися на власні ресурси у вирішенні складних завдань. Додатковою силою платформи є її кросплатформність: вона може функціонувати як на серверах, так і на персональних комп’ютерах, підтримуючи операційні системи Windows, Linux та macOS. Однак ця універсальність не компенсує відсутність професійного супроводу у базовому пакеті, що обмежує її застосування в корпоративних середовищах з високими вимогами до надійності.

1.4 Проблеми перенавантаження аналітиків

Системна проблема сучасної кібербезпеки є “втома від сповіщень”- феномен, що виникає в результаті надмірної кількості тригерів, які не відрізняються контекстною відповідністю, або мають високий рівень хибнопозитивних спрацювань. Цей феномен є супутником аналітиків центру оперативного реагування (SOC) причиною цього стає перевтома через надмірну кількість сповіщень, що генерується SIEM-системою. Однією з основних причин, що викликають перевтому є некоректна кореляція правил SIEM, заснованих на сигнатурному підході. Статичні шаблони, спрямовані на відомі загрози, часто генерують сповіщення про події, які не пов'язані з реальними загрозами. Наприклад, спам-атаки з динамічно змінюваними IP-адресами можуть неодноразово активувати один і той самий тригер, що збільшує ймовірність їх ігнорування аналітиками.

Проблема ще більше посилюється через те, що сповіщення не пріоритетизовані відповідно до потреб бізнесу. Наприклад, доступ до тестового середовища та корпоративної бази даних може вважатися однаково важливим, навіть якщо їхні потенційні ризики дуже різні. Також важливу роль відіграє потреба в використані ручної обробки, навіть при умовах інтеграції з SOAR-платформами. Третина всіх задач (розслідування складних атак, або розслідування *EDR* [12] і тд.) вимагають безпосереднього втручання аналітиків.

Постійний потік даних може призвести до втрати фокуса, що може призвести до пропуску ключових подій. Зокрема, у звіті Ponemon Institute (2023) зазначається, що аналітики у 60% організацій не мають часу реагувати на атаки, такі як SQL-ін'єкції, просто гублячись у потоці хибнопозитивних результатів.

1.5 Концепція SOAR як інструмент автоматизації кібербезпеки

Концепція SOAR (Security Orchestration, Automation and Response) , виникла як відповідь на сучасний етап розвитку кіберзагроз, яка характеризується використанням ШІ та цілеспрямованістю, що потребує переходу від фрагментованих методів реагування до комплексних автоматизованих рішень, де ручна обробка стає неефективною в умовах стрімкого зростання обсягу даних, через неможливість швидко реагувати та перевіряти всі підозрілі події.

У той час як SIEM-системи зосереджені на зборі, нормалізації та кореляції подій безпеки, платформи *SOAR* інтегрують три взаємопов'язані компоненти: координацію робочих процесів, автоматизацію повторюваних операцій і контекстно-орієнтовані механізми реагування.

Координація в контексті *SOAR* передбачає структурування багатьох процесів, що містять в собі взаємодію між різними інструментами кібербезпеки (наприклад, фаєрволами, системами ендпоінт-захисту, базами даних Threat Intelligence, зображено на рис. 1.9). Це робиться шляхом створення сценаріїв (плейбуків), які визначають послідовність дій у відповідь на певні тригери, від блокування IP-адрес до ізоляції скомпрометованих пристроїв від мережі. Такий спосіб дозволяє уникнути розрізненості в роботі SOC, в особливості коли над інцидентом працює кільки фахівців, або ж надходження даних з зовнішніх джерел. Автоматизовний сценарій реагування на витік даних передбачає виявлення IoC, автоматичне блокування шкідливих IP-адрес у мережевому обладнані, перевірка *SIEM* шляхом аналізу логів EDR-систем, формування звітів для відділу комплаєнсу.

Третій компонент переадбачає опрацювання змісту для відповідної реакції, це припускає використання машинного навчання та аналітики, для коректного виставлення пріоритетності інцидентів базуючись на їх критичності. *SOAR* - системи передбачають не тільки аналіз технічних загроз (наприклад, CVSS-бали вразливостей), а й бізнес контекст (фінансові ризики, репутаційні наслідки і тд.)

Такий підхід дозволяє уникнути феномену "сліпоти уваги". Наприклад, атака типу DDoS на інтернет-магазин може отримати вищій пріоритет порівняно зі спам-розсилкою у внутрішній мережі, завдяки аналізу потенційних фінансових втрат. Функціональність таких механізмів в першу чергу залежить від якості вхідних даних, оскільки автоматизовані рішення, засновані на помилкових тригерах *SIEM* можуть ініціювати помилки.

Наприклад, некоректний сигнал зі скомпрометованої IP-адреси може спричинити необґрунтоване блокування легітимних сервісів, що, у свою чергу, призведе до порушення угоди про рівень обслуговування або навіть простоїв у роботі систем. Складність сценаріїв підтримки зростає через характер кіберзагроз, які вимагають постійного оновлення правил та додаткових ресурсів, особливо для організацій з невеликим бюджетом або кваліфікацією персоналу. Навіть високий ступінь автоматизації не виключає мінімальну необхідність в людському контролі. Неправильні рішення в системі важко виправити, а їхні наслідки часто мають довгострокові наслідки.

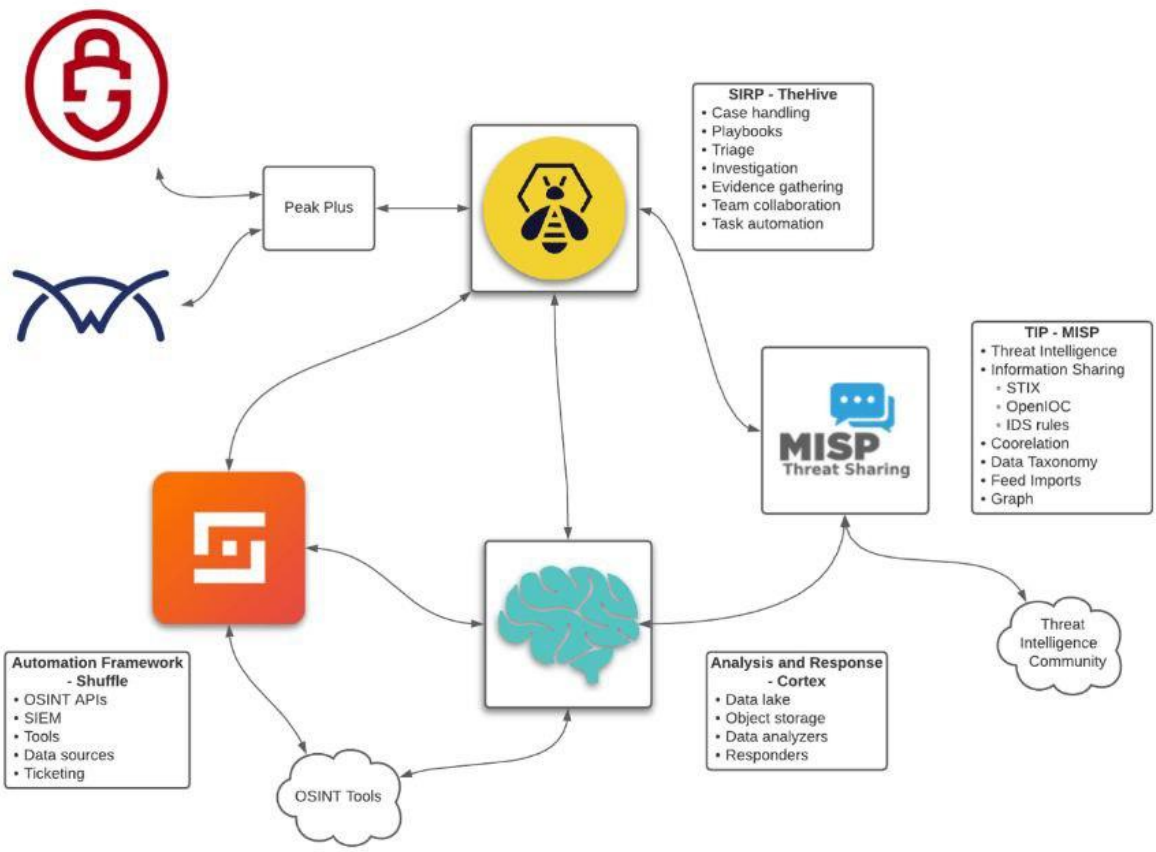

Рисунок 1.9 - Взаємодія ключових компонентів платформи кібербезпеки

1.6 Необхідність розробки доступних рішень для автоматизації

На сьогодні кіберпростір стрімко розвивається, паралельно з чим розвиваються і витонченість та хитрість загроз. Тому на даний момент автоматизація процесів є не забаганкою, а критичною необхідністю для будь-яких організацій. На цю мить в сфері кібербезпеки панує дисбаланс, оскільки тільки великі компанії можуть дозволити собі ефективний захист за допомогою комерційних рішень, як *SOAR* та *SIEM*, через високу вартість, важкість інтеграцій та вимог до технічних ресурсів, в той часу, як малі та середні організації не мають на це ресурсів.

Малі та середні підприємства становлять понад 90% бізнес-структур в усьому світі, але саме вони не мають можливості та ресурсів на боротьбу з атаками, що і робить їх бажаною та легкою жертвою, через що вони найчастіше потерпають. Надання доступних рішень дозволить таким організаціям підвищити рівень кіберзахисту.

Першим етапом у підвищенні ефективності захисту може стати скорочення так званих «вікон вразливості» шляхом автоматизації типових сценаріїв реагування, зокрема блокування IP-адрес, генерації звітів тощо. Наступним логічним кроком є адаптація систем до локальних потреб, оскільки глобальні платформи не завжди враховують специфіку регіональних загроз. У цьому контексті доступні рішення мають перевагу в гнучкості та можливості кастомізації під конкретні виклики без додаткових витрат.

Не менш важливим етапом є впровадження елементів штучного інтелекту, зокрема використання власних скриптів для інтелектуального аналізу логів та виявлення аномальної активності [13].

В умовах, коли кібератаки набувають масового характеру, створення доступних засобів захисту перестає бути лише технічним викликом і трансформується у критично важливу потребу. Кожна організація, незалежно від масштабу, повинна мати можливість користуватися ефективними інструментами для виявлення, реагування й профілактики загроз. Інвестиції у відкриті технології, доступний інтерфейс та спрощені моделі ліцензування є необхідною умовою формування сталого та безпечного кіберпростору на глобальному рівні.

Висновки до розділу 1

Сучасна кібербезпека вимагає інтеграції *SIEM* з автоматизацією (*SOAR*) для ефективного управління загрозами. Однак ефективність залежить не лише від технічних можливостей, а й від адаптації рішень до потреб організації. Розробка

доступних інструментів, оптимізація правил кореляції та зменшення залежності від ручної обробки – критичні кроки для забезпечення кіберстійкості в умовах динамічного розвитку загроз.

У розділі було проаналізовано функціональність сучасних SIEM-систем як ключового компонента кіберзахисту організацій. Основні функції *SIEM* включають збір, нормалізацію та кореляцію логів з різних джерел, інтеграцію з Threat Intelligence для виявлення *IoC*, генерацію звітів.

Проведено порівняння провідних рішень: Splunk вирізняється аналітичними можливостями, IBM QRadar — стабільністю, McAfee ESM — масштабованістю, а Elastic Stack — доступністю, хоч і з обмеженим безкоштовним функціоналом.

Наголошено на проблемі перевантаження аналітиків через велику кількість хибнопозитивних сповіщень та необхідність оптимізації правил кореляції і пріоритезації подій.

SOAR-платформи доповнюють SIEM, автоматизуючи рутинні дії та пришвидшуючи реакцію на інциденти. Водночас, малий і середній бізнес часто не має доступу до дорогих рішень, що зумовлює актуальність open-source інструментів (Wazuh, Security Onion) та хмарних сервісів із гнучкими моделями оплати.

РОЗДІЛ 2
АНАЛІЗ СУЧАСНИХ ПІДХОДІВ ДО АВТОМАТИЗАЦІЇ РЕАГУВАННЯ

2.1 Огляд методів інтеграції SIEM з інструментами автоматизації

Автономний засіб захисту від шкідливих програм здатний нейтралізувати загрози, але він не дає комплексного підходу щодо послідовності дій при кібератаці. На противагу цьому, інтеграція *SIEM* із антивірусними рішеннями надає уніфікований моніторинг усіх подій безпеки, оскільки аналізує дані в контексті абсолютно всієї інфраструктури. Саме такий підхід дозволяє виявляти шаблони та аномальні дії, які за відсутності системного аналізу, могли б лишитися непомітними, що суттєво підвищує ефективність захисту. Саме в цьому й полягає головна перевага — можливість інтегрувати різноманітні рішення та застосовувати комплексний підхід до кожної атаки.

З власного досвіду можу підтвердити, що чим більше інструментів інтегровано в систему, тим більша кількість логів і подій надходить для аналізу. Це суттєво покращує прозорість процесів і дозволяє швидше виявляти загрози. Детальне відстеження кожного етапу — від шкідливих запитів до веб-сервера до отриманих відповідей — дозволяє суттєво скоротити час, потрібний для повного аналізу ситуації.

Якщо, ж ми детальніше розглянемо інтеграцію, то стане зрозуміло, що *SIEM* агрегує журнали з усіх кібербезпекових інструментів, включно з антивірусними рішеннями, та аналізує взаємозв’язки між цими даними для ідентифікації аномальних шаблонів. Наприклад, у разі виявлення інструментом захисту у кінцевих точок ознак інфікування шкідливим ПЗ, *SIEM* автоматично ініціює перевірку мережі на наявність додаткових індикаторів компрометації, таких як підозріла активність користувачів або незвичні запити до критичних ресурсів, що

дозволяє встановити повний ланцюг атаки. Переваги інтеграції SIEM з різними інструментами кібербезпеки розглянуто у табл. 2.1.

Таблиця 2.1

Переваги інтеграції *SIEM* з різними інструментами кібербезпеки

Інструмент	*Переваги інтеграції з SIEM*	*Приклади рішень*
1	*2*	*3*
Endpoint Protection	*– Автоматичне виявлення загроз на кінцевих пристроях.* *– Кореляція даних з мережевими подіями.*	*CrowdStrike, SentinelOne, Microsoft Defender*
Firewalls	*– Аналіз трафіку на предмет підозрілих шаблонів.* *– Виявлення порушень політик безпеки.*	*Palo Alto, Cisco Firepower, Fortinet*
IDS/IPS	*– Ідентифікація атак у реальному часі.* *– Зменшення хибно-позитивних сповіщень.*	*Snort, Suricata, Darktrace*
Vulnerability Management	*– Пріоритезація загроз на основі активних вразливостей.* *– Визначення ризиків для активів.*	*Tenable, Qualys, Rapid7*

1	*2*	*3*
Threat Intelligence	*– Автоматичне оновлення індикаторів загроз (IoCs).* *– Підвищення точності детекції.*	*MISP, ThreatConnect, IBM X-Force*
Network Traffic Analysis	*– Виявлення аномалій у мережевому трафіку.* *– Аналіз горизонтального пересування в мережі.*	*Darktrace, Vectra, Corelight*
Cloud Security	*– Моніторинг подій у хмарних середовищах.* *– Виявлення конфігураційних помилок.*	*AWS GuardDuty, Azure Sentinel, Prisma Cloud*
IAM (Identity Management)	*– Детекція несанкціонованого доступу.* *– Аналіз аномалій у поведінці користувачів.*	*Okta, Microsoft Active Directory, PingID*
Log Management	*– Централізоване зберігання журналів.* *– Покращений аналіз історичних даних.*	*Splunk, Elasticsearch, Graylog*

1	*2*	*3*
Email Security	*– Виявлення фішингових атак та шкідливих вкладень.*	*Proofpoint, Mimecast, Barracuda*
SOAR	*– Автоматизація реагування на інциденти.* *– Оптимізація робочих процесів SOC.*	*Palo Alto Cortex XSOAR, IBM Resilient*
Compliance Management	*– Автоматичне формування звітів для аудиту.* *– Відстеження відповідності стандартам.*	*RSA Archer, SolarWinds, Tripwire*

Отже у сучасному віртуальному середовищі жоден окремий інструмент не здатний самостійно забезпечити повноцінний захист вашого бізнесу. Проте об’єднання таких рішень, як SIEM, EDR, EPP [14] та антивірусні системи, формує інтелектуальну, масштабовану та адаптивну систему захисту. Це вже не лише захист, а стратегічний та прогнозований підхід до кібербезпеки, орієнтований на попередження загроз до їх реалізації. Якщо, ж збереження цілісності цифрових активів є пріоритетом, критично важливо перейти до комплексної стратегії. Загрози вдосконалюються, але й механізми захисту еволюціонують, забезпечуючи стійкість у світі, де кіберризики стають все складнішими та динамічнішими.

2.2 Використання API для взаємодії з зовнішніми системами

API (Application Programming Interface) розшифровується як «інтерфейс програмної взаємодії», що визначає механізми комунікації між програмними компонентами [15]. Його суть полягає в наданні функціоналу для обміну даними та синхронізації дій різних систем.

Інтеграція через API передбачає об’єднання вашого сервісу або застосунку зі сторонніми платформами. Наприклад:

- Платіжні шлюзи інтегруються з сайтом для автоматизації фінансових операцій.
- Віджети прогнозу погоди вбудовуються в блоги для надання актуальних даних.
- Системи електронного документообігу синхронізуються із зовнішніми сервісами, що дозволяє керувати всією документацією в одному інтерфейсі.
- CRM-системи автоматично отримують дані користувачів із сайту, усуваючи необхідність ручного введення.
- Інтеграція зі ШІ дає змогу автоматизувати створення контенту для інтернет-магазинів.

Принцип роботи API. API надає доступ до функцій інших систем, дозволяючи використовувати їхні можливості у підтримуваних розробниками продуктах. Тобто, ми можемо отримувати або передавати дані, взаємодіючи зі сторонніми сервісами. Це розширює функціональність рішення, та робить його більш зручним і потужним.

Переваги інтеграції через API:

- Ефективність. Використання готових API замість розробки функцій з нуля дозволяє заощадити ресурси. Наприклад, інтеграція *SIEM* із Threat Intelligence через API дає доступ до актуальних даних про загрози без власних розробок.

- Економія часу та коштів. IT-фахівці не витрачають час на створення базового коду — вони інтегрують готові рішення. Це дозволяє зосередитися на стратегічних завданнях, як-от аналіз загроз.
- Функціональна гнучкість. API додає нові можливості вашому продукту. Наприклад, інтеграція з *EDR* дозволяє автоматизувати відповідь на інциденти, покращуючи захист.
- Зниження навантаження. Постачальники API самостійно оновлюють свої рішення. Ви не витрачаєте ресурси на підтримку інтеграції, але маєте гарантію її стабільності.

API стає містком між інструментами захисту (*SIEM*, EDR, сканери вразливостей), забезпечуючи автоматичну кореляцію подій та швидке реагування. Наприклад, при виявленні підозрілої активності *SIEM* може через API ініціювати ізоляцію пристрою в EDR, запобігаючи розповсюдженню атаки.

2.3 Сценарії автоматизації: блокування IP-адрес, генерація звітів

Сучасні нормативно-правові акти все більше акцентуютъ увагу на оперативному реагуванні на інциденти та обов’язковому сповіщенні про них, часто вказуючи строго визначені часові періоди для таких дій. Наприклад, *GDPR* вимагає повідомляти органи нагляду про порушення захисту даних протягом 72 годин, тоді як HIPAA та CMMC встановлюють жорсткі вимоги щодо звітування, це стосуються як змісту, так і своєчасності подання інформації.

Платформи *SOAR* забезпечують дотримання цих вимог через інтеграцію спеціалізованих механізмів. По-перше, автоматизовані протоколи ескалації, які базуються на попередньо налаштованих робочих процесах, дозволяють ідентифікувати критичні інциденти, активувати сповіщення та інформувати відповідальні сторони, включно з регуляторами, у визначені терміни. По-друге, комплексні заходи реагування, такі як ізоляція скомпрометованих систем або

впровадження виправлень, централізуються та керуються через SOAR, що гарантує оперативність і системність у ліквідації наслідків. По-третє, постійне тестування, включаючи регулярні тренування з відпрацювання сценаріїв інцидентів, дає змогу організаціям оцінити ефективність своїх процедур в умовах, близьких до реальних, завдяки інтеграції з можливостями *SOAR* для моделювання різноманітних загроз.

Важливо розуміти різницю між автоматизацією та оркестрацією кібербезпеки: автоматизація виконує окремі рутинні завдання без участі людини (наприклад, сканування логів), тоді як оркестрація координує взаємодію різних інструментів (SIEM, EDR, сканери вразливостей) для комплексного реагування (наприклад, одночасне ізолювання зараженого пристрою й оновлення правил фаєрволу) [16].

З власного досвіду наведу приклад автоматизації на основі SOAR:

- SIEM-система (наприклад, Splunk) виявила аномальний трафік з IP-адрес, пов’язаних з ботнетом (підтверджено через AbuseIPDB).
- SOAR автоматично запустила сценарій блокування цих IP у фаєрволі (AWS WAF), ізолювала скомпрометовані сервери через *EDR* (CrowdStrike) та відправила сповіщення в Slack *SOC* і керівництву.
- Результат: час реагування скоротився з 2 годин до 2 хвилин, більшість атак було заблоковано на ранніх стадіях.

Сценарій автоматизації: Генерація звітів

Збір даних: щоденно *SIEM* агрегувала логи з усіх джерел (мережеві пристрої, сервери, додатки).

Обробка та аналіз:

- Python-скрипт автоматично фільтрував події за критеріями (кількість спроб входу, підозрілі IP). [17]
- Дані передавалися в шаблон Power BI через *API* для візуалізації.

Формування звіту:

SOAR генерував PDF-звіт з ключовими метриками:

- Кількість заблокованих атак.
- Статистика порушень відповідності *GDPR*.
- Найбільші 5 джерел з яких походили загрози.

Звіт автоматично відправлявся на пошту CISO та зберігався в SharePoint для аудиту.

Результат: Час на підготовку звітів скоротився з 4 годин до 15 хвилин на день, а помилки які виникали через ручну обробку зникли.

2.4 Проблеми та обмеження наявних рішень

У наш час, коли *SOAR* вже повинен бути достуний абсолютно кожній організації, цим часто нехтують. Більшість причини пов’язані з високою вартістю ліцензування, складністю інтеграції та розширення, відсутність закікавленості керівництва, через нерозуміння та некмпетентність [18]. Нажаль дуже багато компанії починають дбати, про свою безпеку лише після реальної зустрічі з інциндентом та його наслідками. Проте данна технологія, повинна бути детально вивчена перед впровадженням, щоб чітко розуміти її можливості, переваги та недоліки. Лише об’єктивне бачиння цього, зможе значто підсилити рівень кібербезпеки організації.

Всі основні недоліки основних та найпопулярніших рішень подані в табл. 2.2.

Таблиця 2.2

Проблеми та обмеження SOAR

Проблеми та обмеження	*Опис*	*Вплив на кібербезпеку*
1	*2*	*3*
Складність інтеграції	Необхідність підключення до різних інструментів (SIEM, EDR, фаєрволи).	Затримки у впровадженні, зниження ефективності через несумісність технологій.
Складність налаштування	Необхідність підключення до різних інструментів (SIEM, EDR, фаєрволи).	Ризик помилок (напр., хибне блокування IP) або пропуск реальних загроз.
Залежність від даних	Ефективність залежить від якості даних з SIEM/EDR.	Ризик помилок (напр., хибне блокування IP) або пропуск реальних загроз.
Високі витрати	Значні інвестиції у ліцензії, обладнання, навчання.	Недоступність для малих організацій.
Ризики кібербезпеки	SOAR може стати ціллю для атак.	Компрометація платформи → втрата контролю над автоматизацією.
Обмежена гнучкість	SOAR може стати ціллю для атак.	Неможливість реагування на специфічні атаки.
Необхідність оновлень	Необхідні регулярні оновлення для підтримки методів захисту від нових загроз.	Застарілі сценарії → зниження ефективності.
Людський фактор	Надія на автоматизацію знижує уважність операторів SOC.	Пропуск складних атак, які потребують креативного аналізу.

1	*2*	*3*
Хибні спрацьовування	Надія на автоматизацію знижує уважність операторів SOC.	Перевантаження команди, зниження довіри до системи.
Обмеження автоматизації	Деякі завдання (напр., юридичний аналіз) неможливо повністю автоматизувати.	Необхідність ручного втручання на критичних етапах.

Сценарії мінімізування ризиків:

- Тестування сценаріїв: Регулярно перевіряти їх на реальних імітаціях атак.
- Гнучкі *API*: Обирати платформи з підтримкою інтеграції зі сторонніми системами.
- Захист SOAR: Використовувати MFA, шифрування, сегментацію мережі.
- Комбінація з людиною: Зберігати експертний аналіз для складних випадків.

Висновки до розділу 2

Цей розділ детально досліджує критично важливі аспекти впровадження та експлуатації двох ключових технологій сучасної операційної безпеки: *SIEM* та *SOAR*. Їх синергетичне використання формує основу проактивного та ефективного захисту інформаційних активів. *API* відіграють важливу роль у побудові взаємодії між системами безпеки, забезпечуючи гнучкість і масштабованість. Завдяки API архітектура безпеки стає гнучкою та масштабованою. Нові інструменти можна додавати без повної перебудови системи, а обробка зростаючих обсягів даних та

подій спрощується через стандартизовані механізми обміну. В свою чергу *SOAR* платформи автоматизують рутинні, повторювані завдання, що виконуються аналітиками SOC під час розслідування та реагування на інциденти. Автоматизація процесів через *SOAR* значно знижує навантаження на аналітиків і скорочує час реагування на інциденти.

Водночас варто враховувати складність інтеграції, вартість рішень та ризики хибнопозитивних спрацювань. Загалом, ефективний захист потребує комплексного, адаптованого під організацію підходу з акцентом на інтеграцію, автоматизацію та контекстне реагування.

РОЗДІЛ 3
РОЗРОБКА ТА ТЕСТУВАННЯ СИСТЕМИ АВТОМАТИЗОВАНОГО РЕАГУВАННЯ

3.1 Створення та налаштування тестового середовища

Система автоматизованого реагування на інциденти буде реалізована на тестовому середовищі, яке буде розгорнуте з залученням двох віртуальних машин за допомогою програмного забезпечення VirtualBox [19]:

- Ubuntu Desktop 22.04 з встановленим Elastic Stack (Elasticsearch, Kibana, Filebeat), Python
- Kali Linux для проведення тестувань

На рис. 3.1 зображенно робочі столи успішно встановлених *ВМ*.

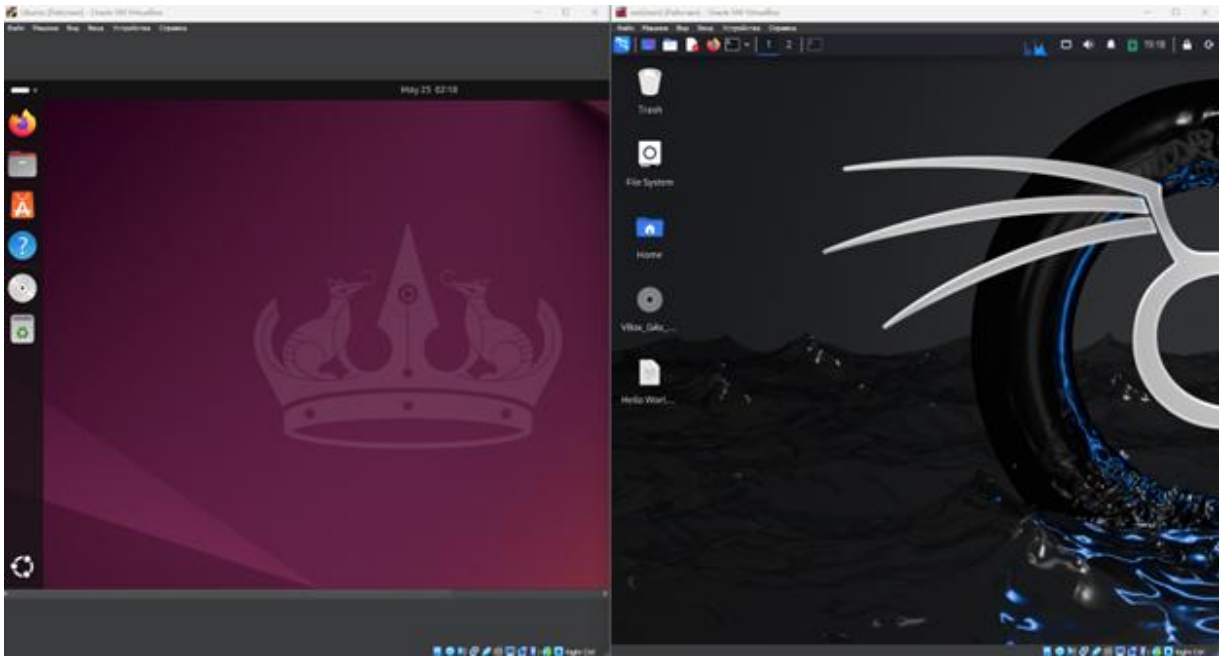

Рисунок 3.1 - Встановлені ВМ

3.1.1 Обґрунтування вибору відкритих технологій для розробки модуля

Вибір технологій для реалізації системи базувався на поєднанні масштабованості, гнучкості, ефективності та доступності. Зважаючи на ці критерії, було обрано Elastic Stack, який забезпечує унікальну комбінацію інструментів: Elasticsearch як потужне сховище для швидкого пошуку структурованих і неструктурованих даних, та Filebeat — легкий агент для збору логів із мінімальним навантаженням на систему, що критично важливо для середовищ з обмеженими ресурсами.

Важливість зберегти можливість інтеграцій призвела до вибору Python як мови програмування, адже його бібліотека elasticsearch не лише значно спростила взаємодію з *API* Elasticsearch, але й дозволила ефективно керувати системними процесами, зокрема через інтеграцію з *UFW* [20], яка незважаючи на свою простоту, забезпечила надійний механізм блокування.

Сумісність цих технологій виявилася вирішальною: Elasticsearch акумулював дані, зібрані Filebeat, Python-модуль аналізував їх, знаходячи аномалії, а *UFW* втілював відповідні заходи безпеки.

3.1.2 Архітектура Elastic Stack для збору та аналізу подій

Архітектура, була побудована на базі Elastic Stack (скріни запуску відповідних сервісів на рис. 3.3). *SIEM* формує єдиний механізм, який поєднує збір, обробку та інтерпретацію даних, що є критичним для ефективного виявлення аномалій у логах. Основу становить Elasticsearch – це розподілена пошукова та аналітична платформа, здатна індексувати величезні обсяги даних, що надходять із різних джерел. Її роль полягає не лише у зберіганні інформації, але й у забезпеченні швидкого пошуку та агрегації, що стає можливим завдяки оптимізованим алгоритмам розподілу даних

між вузлами, що буде дуже корисно при розширенні збору логів від однієї машини до всієї інфраструктури компанії.

В свою чергу данні надходять до Elasticsearch через Filebeat [21] (скрін встановлення інструменту Filebeat наведено на рис. 3.4). Filebeat приставляє з себе легкого агента, який, на відміну від доволі ресурсоємних рішень, зосереджений на безперервному моніторингу лог-файлів, таких як записи фаєрволу або системні події. Filebeat виступає проміжною ланкою, перетворюючи сирі логи у структуровані документи, готові до аналізу, що значно спрощує подальшу обробку. Наприклад, кожен запис про мережеве з’єднання перетворюється на JSON-об’єкт із полями source.ip, destination.port та message, які стають основою для пошукових запитів.

Також, ще одним ключовим елементом є Python скрипти, які, з’єднуючись з Elasticsearch, автоматично реагують на події, відповідно прописаних правил. Вони аналізують дані, застосовуючи наперед визначені правила, такі як виявлення IP-адрес із надмірною кількістю спроб підключення до різних портів, і ініціюють відповідні дії через інтеграцію з *UFW*. Таким чином, система не лише фіксує загрози, але й реагує на них у режимі, близькому до реального часу.

В свою чергу *UFW*, що розшифровується як Uncomplicated Firewall, являє собою інтерфейс командного рядка для керування мережевим екраном Netfilter (iptables) в Unix-подібних системах, зокрема в дистрибутивах на базі Debian та Ubuntu. Основна мета утиліти полягає у спрощенні складного процесу налаштування брандмауера. Незважаючи на свою простоту, *UFW* забезпечує потужний захист, оскільки він діє як фронтенд до високоефективної, але технічно складної системи iptables, надаючи логічний та інтуїтивно зрозумілий спосіб визначення правил дозволу або блокування мережевого трафіку.

Функціональність *UFW* орієнтована на зручність користувача, що проявляється у використанні простих команд, за допомогою яких можна керувати вхідними та вихідними з'єднаннями через порти, протоколи (TCP, UDP) або навіть

за іменами служб, зазначеними у файлі /etc/services, що суттєво спрощує процес конфігурації порівняно з прямим редагуванням правил iptables. Проте, незважаючи на спрямованість на простоту, *UFW* підтримує створення досить складних правил, включаючи вказівку конкретних IP-адрес, підмереж або навіть інтерфейсів, до яких вони повинні застосовуватись, а також дозволяє налаштовувати правила для логування подій, що є важливим для моніторингу та аналізу безпекових подій.

Нижче ми можемо бачити процес встановленя Elastic Stack на Ubuntu на рис. 3.2 та Filebeat на рис. 3.4, 3.9.

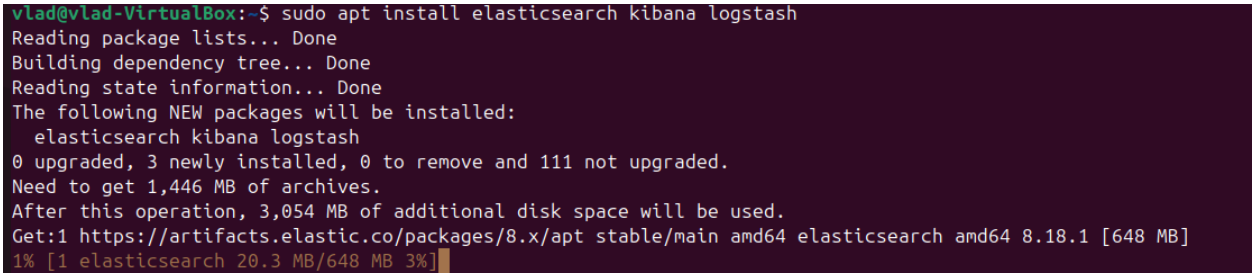

Рисунок 3.2 - Встановлення Elastic Stack

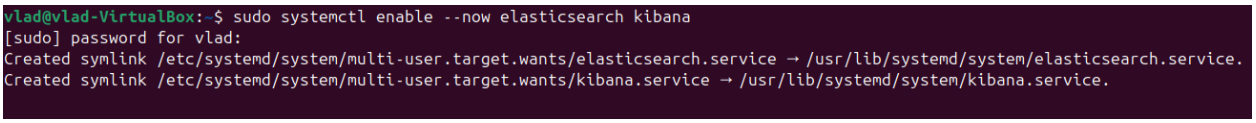

Рисунок 3.3 - Запускаємо сервіси elasticsearch та kibana

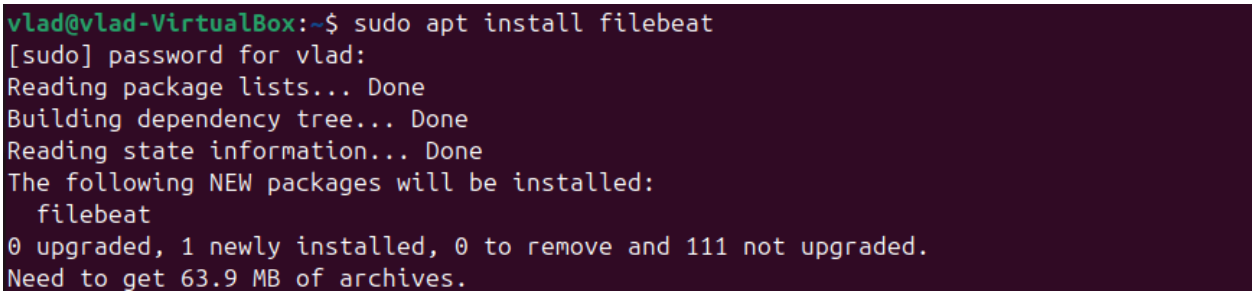

Рисунок 3.4 - Встановлення Filebeat для збору локальних логів

Підтвердження успішного встановлення та функціонування можемо побачити на рис. 3.5 та рис. 3.6

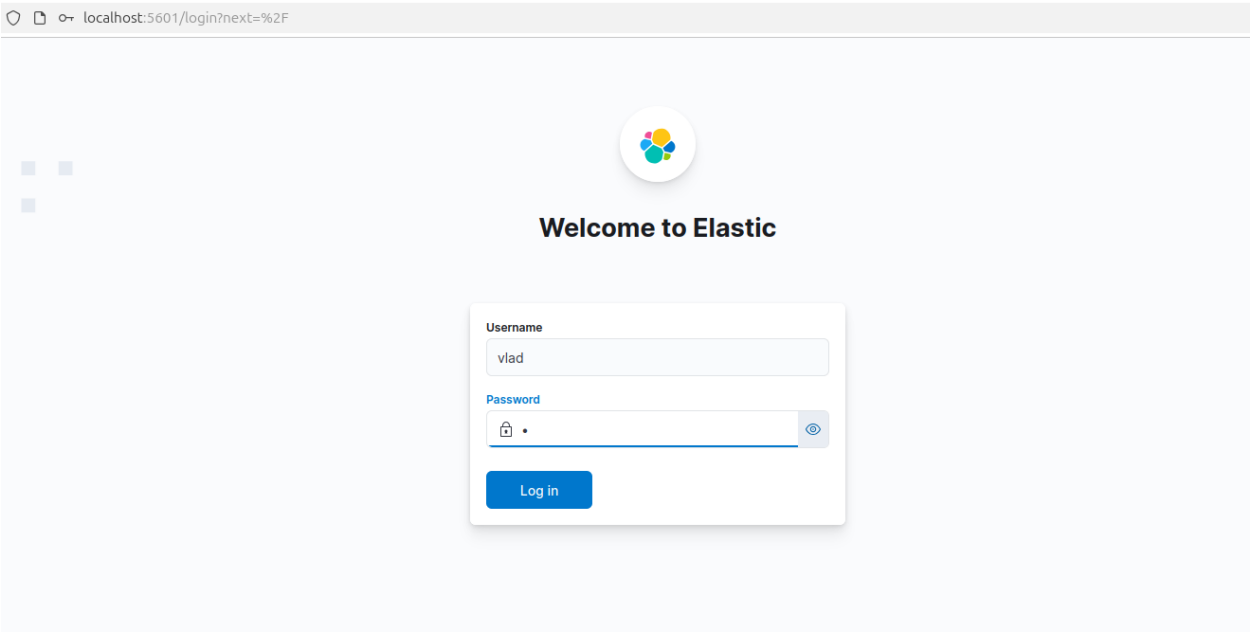

Рисунок 3.5 - Отриманий доступ через веб інтефейс до elastic

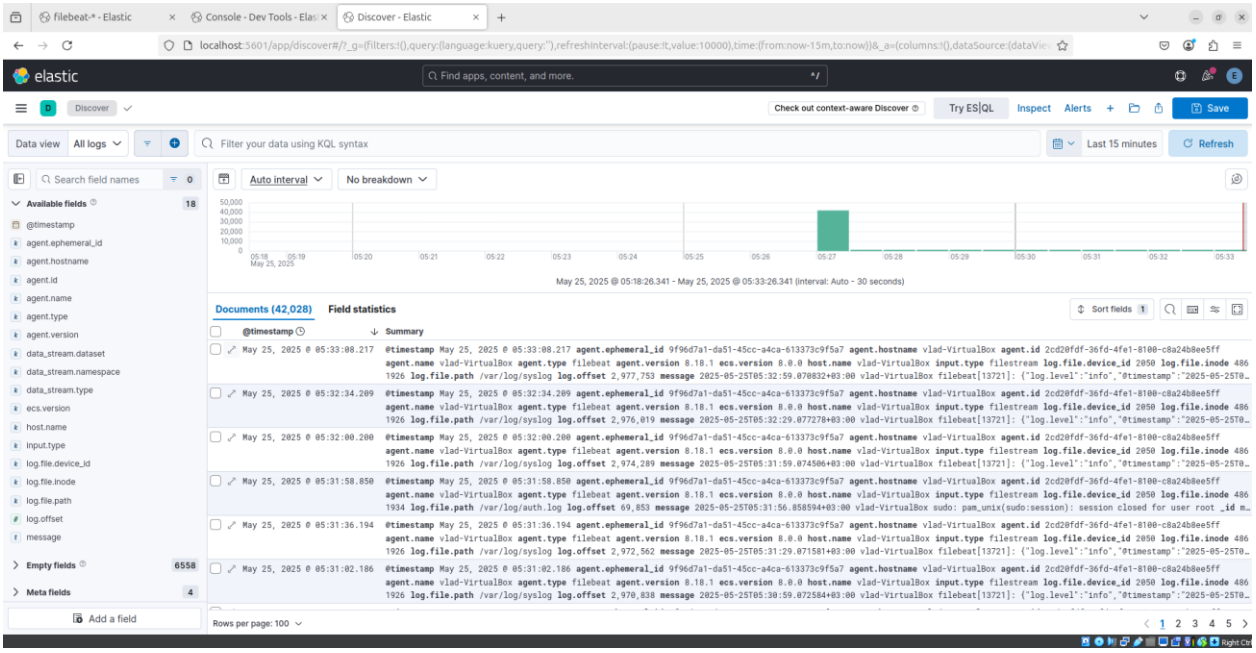

Рисунок 3.6 - Відображення корректного отримання логів

3.2 Розробка та реалізація модуля на базі Python

Вибір Python зумовлений широкою підтримкою бібліотек для обробки даних. Зокрема, elasticsearch-py забезпечує зручну взаємодію з Elasticsearch, а requests — ефективну роботу з HTTP-запитами, що є ключовим для реалізації аналізу логів і керування фаєрволом [22].

Основу модулів становить механізм періодичних запитів до Elasticsearch, реалізований за допомогою бібліотеки schedule. Запити спрямовані на виявлення мережевих аномалій — наприклад, сканування портів або надмірна кількість з'єднань з однієї IP-адреси. Для цього використовуються агрегації за полем source.ip з підрахунком унікальних destination.port, що дозволяє виявляти підозрілу активність.

Інтеграція з Filebeat забезпечується налаштуванням передачі логів у Elasticsearch із чіткими полями (source.ip, destination.port, event.description), що значно спрощує аналіз.

Керування фаєрволом *UFW* виконується через subprocess: при виявленні загрози формується команда типу *ufw* insert 1 deny from {ip}, яка попередньо перевіряє відсутність дублювання у правилах (*ufw* status).

Модуль BruteForce (рис. 3.7): Виявляє атаки грубої сили (brute-force attacks) на SSH (TCP/22) шляхом аналізу логів автентифікації в Elasticsearch. Алгоритм ідентифікує IP-адреси джерел з пороговою кількістю невдалих спроб автентифікації. При виявленні порушення модуль автоматично ініціює генерацію та впровадження правил блокування в мережевий екран (UFW).

Модуль ScanDetected (рис. 3.8): Сприймається до виявлення активності сканування портів. Він аналізує запити до цільових портів для визначення IP-адрес з аномально високою кількістю унікальних звернень до портів за часовий інтервал. Виявлення таких статистичних аномалій запускає автоматичне блокування джерела в UFW.

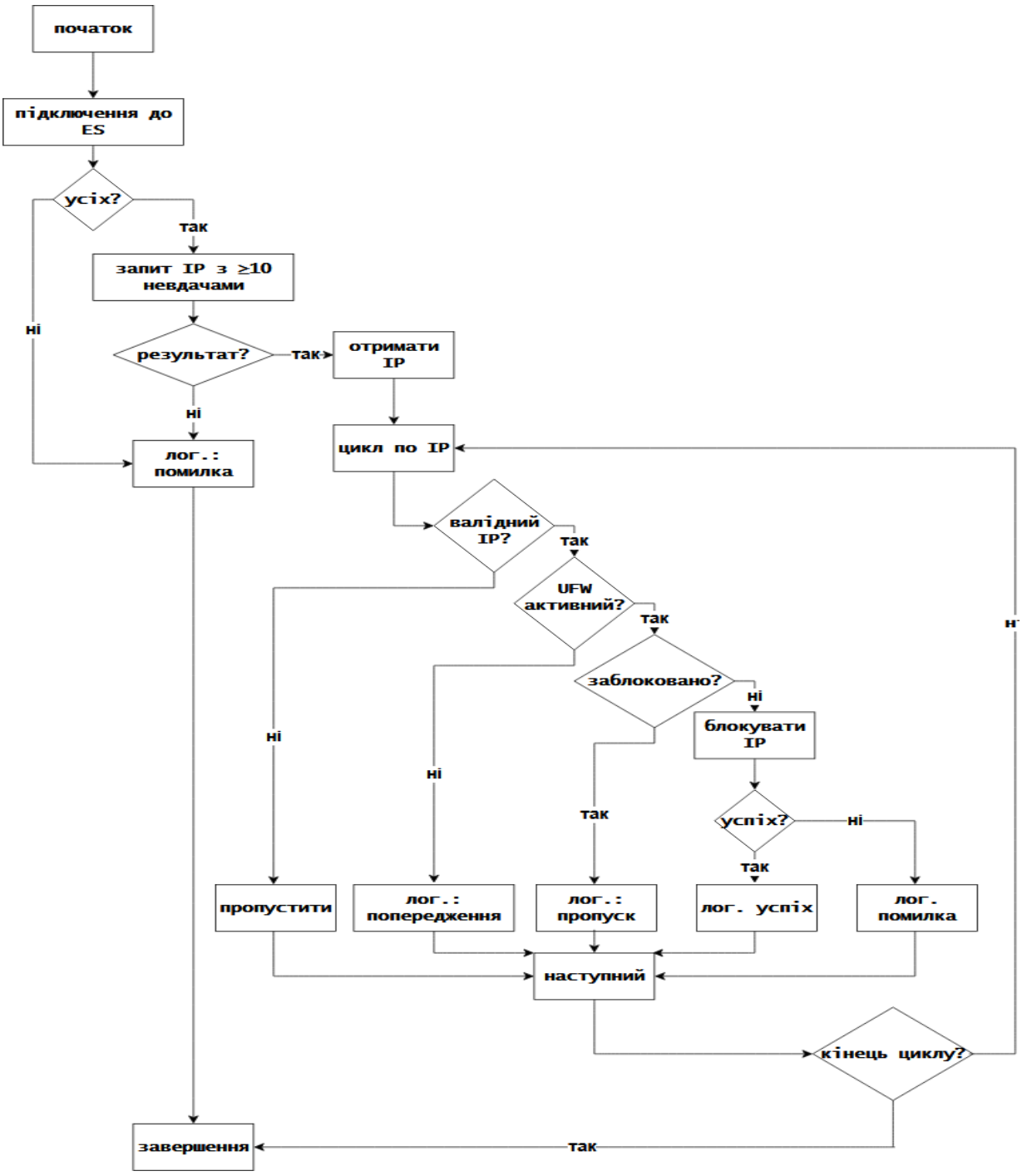

Рисунок 3.7 – Блок-схема для блокування атак типу BruteForce

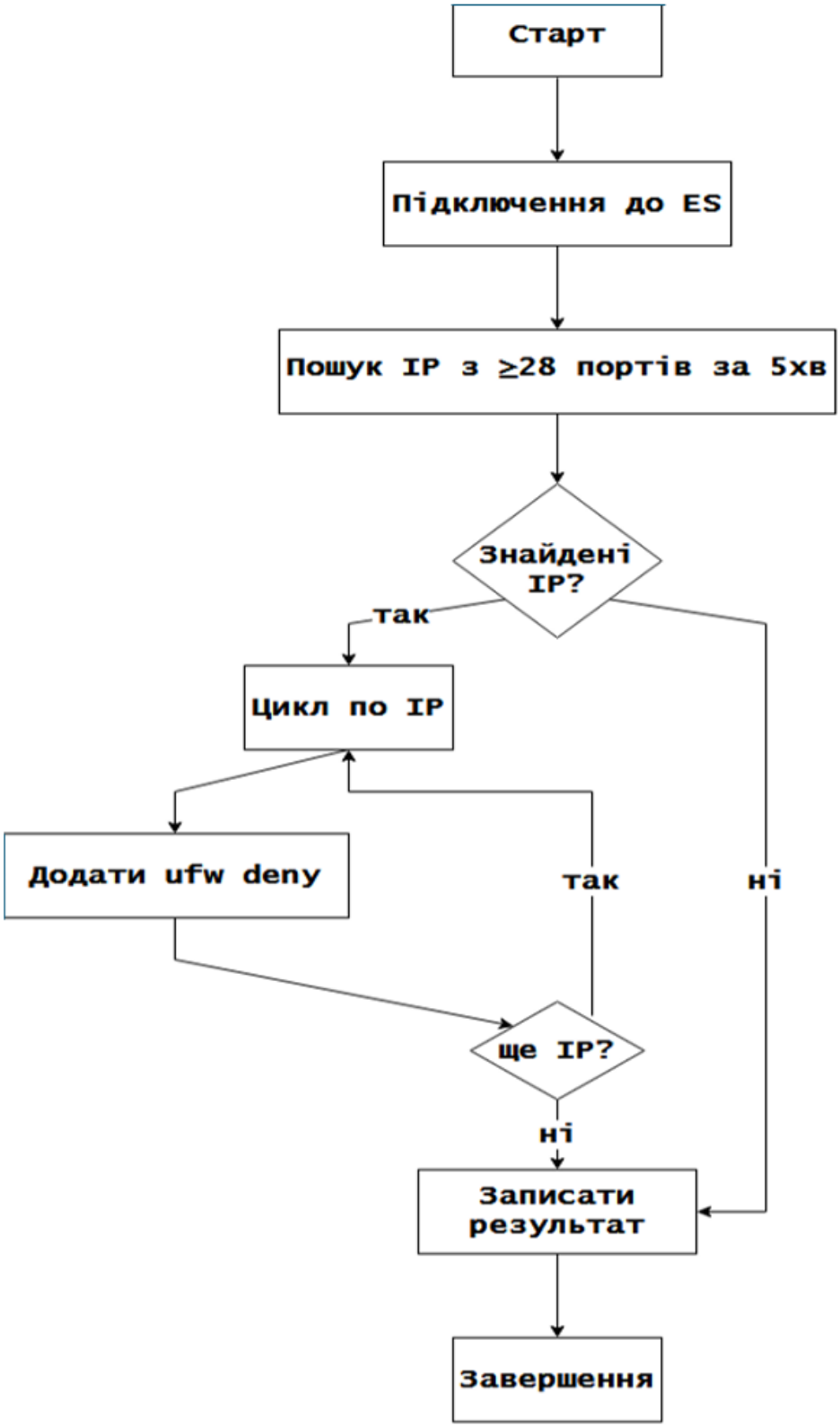

Рисунок 3.8 – Блок-схема коду для блокування IP-адрес, які здійснюють сканування портів

Повний програмний код побудований за блок-схемами на рис. 3.7 та 3.8 наведено в додатках.

Збереження всіх подій буде відбуватись у файл “siem_auto.log” за шляхом /home/vlad/. Особливу увагу приділено саме цьому файлу через його ключову роль у логіці роботи системи — у разі необхідності можна легко додати кастомні правила, які автоматично оновлюватимуться з періодичним очищенням, зберігаючи при цьому попередній стан. Це забезпечує гнучкість та високу адаптивність рішення під конкретні потреби.

3.2.1 Інтеграція Filebeat з Elasticsearch для отримання логів

Інтеграція Filebeat (встановлення інструменту зображено на рис. 3.9) з Elasticsearch, будучи фундаментальним етапом побудови системи, забезпечує безперервний потік даних із джерел логування до централізованого сховища, що є передумовою для подальшого аналізу. Процес починається з налаштування Filebeat (Запуск данного інструмента зображено на рис. 3.11), легкого агента, який, на відміну від важких лог-шейперів, зосереджений на мінімальному споживанні ресурсів, що робить його ідеальним для середовищ із обмеженими потужностями. Конфігурація Filebeat виконується через YAML-файл (рис. 3.10), де визначаються вхідні джерела даних, трансформації полів та параметри виведення, що дозволяє адаптувати його під конкретні потреби системи [23].

Основна увага приділяється визначенню шляхів до лог-файлів, таких як /var/log/ufw.log або /var/log/syslog, які містять записи про мережеву активність.

```
vlad@vlad-VirtualBox:~$ sudo apt install filebeat
[sudo] password for vlad:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  filebeat
0 upgraded, 1 newly installed, 0 to remove and 111 not upgraded.
Need to get 63.9 MB of archives.
```

Рисунок 3.9 - Встановлення Filebeat для збору локальних логів

```
filebeat.inputs:
# Each - is an input. Most options can be set at the input level, so
# you can use different inputs for various configurations.
# Below are the input-specific configurations.

# filestream is an input for collecting log messages from files.
- type: filestream

  # Unique ID among all inputs, an ID is required.
  id: my-filestream-id

  # Change to true to enable this input configuration.
  enabled: false

  # Paths that should be crawled and fetched. Glob based paths.
  paths:
    - /var/log/*.log
    - /var/log/syslog
    - /var/log/auth.log
    #- c:\programdata\elasticsearch\logs\*
```

Рисунок 3.10 Налаштування Filebeat, (редагуємо /etc/filebeat/filebeat.yml

```
vlad@vlad-VirtualBox:~$ sudo nano /etc/filebeat/filebeat.yml
vlad@vlad-VirtualBox:~$ sudo systemctl enable --now filebeat
Synchronizing state of filebeat.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable filebeat
Created symlink /etc/systemd/system/multi-user.target.wants/filebeat.service → /usr/lib/systemd/system/filebeat.service.
vlad@vlad-VirtualBox:~$
```

Рисунок 3.11 Запуск Filebeat

Перевірка надходження логів:

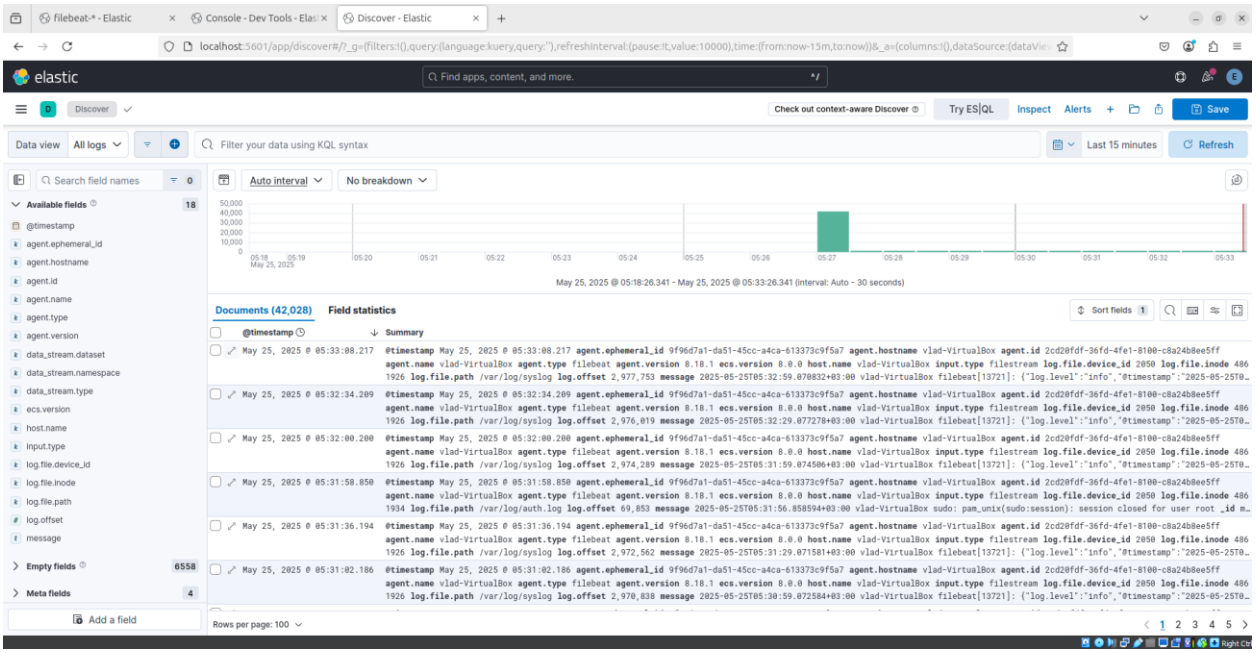

Рисунок 3.12 Інтерфейс Elastic

Відповідно, на рис 3.12 ми бачимо коректне надхлодження логів до Elastic, що підтверджує правильність наших налаштувань та конфігурацій.

3.3 Оцінка ефективності модуля в умовах імітації атак

Ефективність модуля оцінювалась через імітацію кібератак, спрямованих на перевірку здатності системи виявляти аномалії та оперативно реагувати на загрози. Тестування проводилось у контрольованому середовищі, де параметри атак чітко визначались, що дозволило виміряти як кількісні, так і якісні показники роботи системи. Імітація включала сканування портів і брутфорс-атаки.

Брутфорс-атаки на SSH імітувались за допомогою msfconsole та модуля ssh_login. (рис. 3.13 та рис. 3.14)

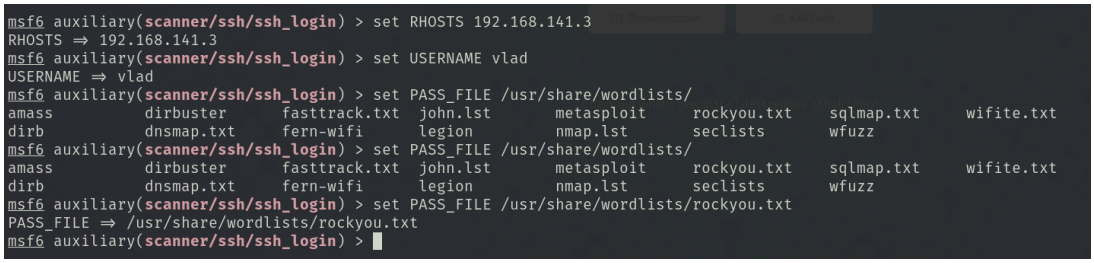

Рисунок 3.13 Налаштування модуля ssh_login

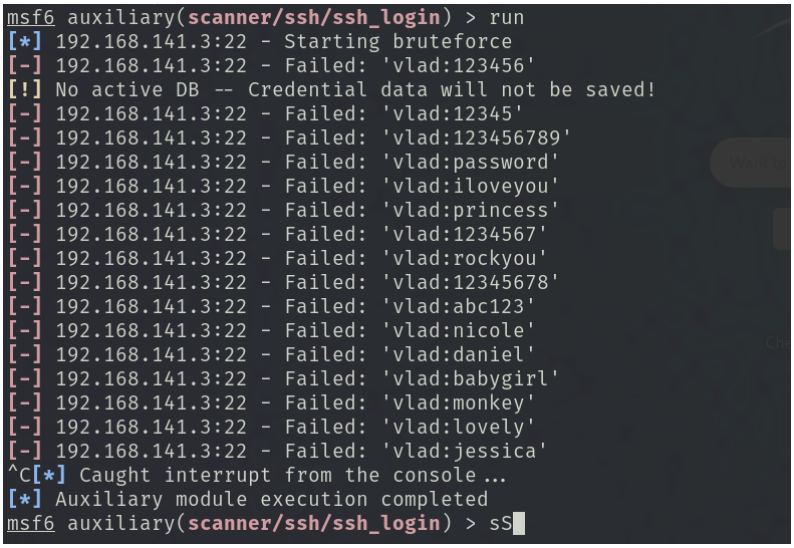

Рисунок 3.14 Процес запуску атаки типу Брутфорс

Далі, ми одразу бачимо в *SIEM* сплеск активіності пов’язаної з 22 портом, скрін якого подано на рис. 3.15.

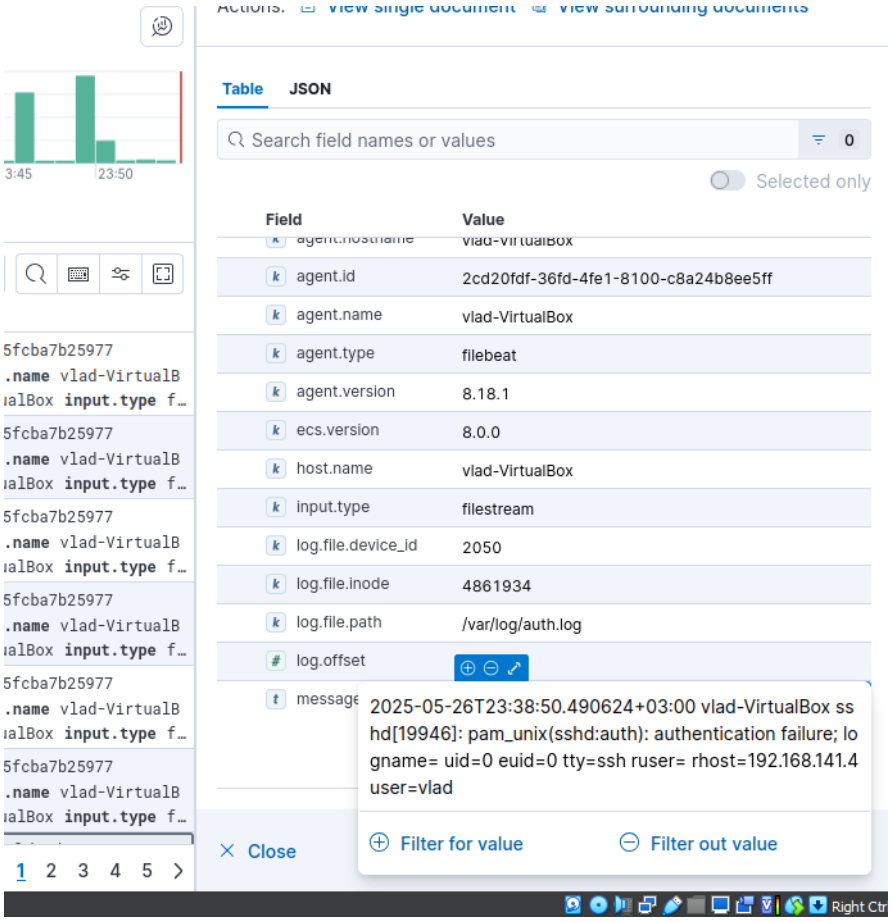

Рисунок 3.15 Вигляд атаки у SIEM

Зачекавши деякий час, перевіримо наш файл з логами, вміст якого наведено на рис. 3.16

Рисунок 3.16 - Вигляд запису логів у файл .txt

Відповідно, ми бачимо заблоковану айпі адресу з якої надходила атака та деяку коротку інформацію. Також ми можемо побачити підтвердження підтвердження, перевіривши правила на фаерволі (рис. 3.17):

```
vlad@vlad-VirtualBox:~/Desktop$ sudo ufw status numbered
[sudo] password for vlad:
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] Anywhere                   DENY IN     192.168.141.4
[ 2] 22/tcp                     ALLOW IN    Anywhere
[ 3] 22/tcp (v6)                ALLOW IN    Anywhere (v6)

vlad@vlad-VirtualBox:~/Desktop$
```

Рисунок 3.17 - Перевірка доданих правил у **UFW**

```
msf6 auxiliary(scanner/ssh/ssh_login) > run
[*] 192.168.141.3:22 - Starting bruteforce
[-] Could not connect: The connection with (192.168.141.3:22) timed out.
[!] No active DB -- Credential data will not be saved!
[-] Could not connect: The connection with (192.168.141.3:22) timed out.
[-] Could not connect: The connection with (192.168.141.3:22) timed out.
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) >
```

Рисунок 3.18 - Блокування при повторній атаці з тієїж IP

На рис. 3.18 ми ще раз, запускаємо атаку і бачимо, що підключення автоматично блокуються, через те, що IP вже заблокована в *UFW*, що підтвержує дієздатність нашої системи.

Для імітації сканування портів використовувався інструмент nmap, процес запуски зображенно на рис. 3.19 [24].

Модуль, аналізуючи логи через Elasticsearch, виявляв IP-адреси з аномальною кількістю унікальних портів, після чого автоматично додавав правила блокування в

UFW. Важливим аспектом стало вимірювання часу між початком атаки та моментом блокування, оскільки саме цей параметр визначає практичну придатність системи в реальних умовах. Середній час реакції становив 3 хвилини та 15 секунд, що обумовлено інтервалом запуску скрипта через Cron, проте це вже значно швидше ручної перевірки та обробки.

Відповідно була запущена атака. Яку система ефективно обробила.

```
msf6 auxiliary(scanner/ssh/ssh_login) > nmap 192.168.141.3
[*] exec: nmap 192.168.141.3

Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-27 13:27 EDT
```

Рисунок 3.19 - Сканування портів за допомогою утиліти nmap

```
Open  • siem_auto.log
runasroot.sh | elastic pass elastic.txt | filebeat.yml | elasticsearch.yml | syslog | siem_auto.log | Brute.py | filebeat.yml | siem_auto.log
2025-05-26 23:10:28,374 - INFO - POST https://localhost:9200/filebeat-*/_search [status:200 duration:0.046s]
2025-05-26 23:10:28,374 - INFO - Не знайдено IP для блокування.
2025-05-26 23:11:30,170 - INFO - POST https://localhost:9200/filebeat-*/_search [status:200 duration:0.027s]
2025-05-26 23:11:30,170 - INFO - Не знайдено IP для блокування.
2025-05-26 23:19:58,359 - INFO - POST https://localhost:9200/filebeat-*/_search [status:200 duration:0.026s]
2025-05-26 23:19:58,360 - INFO - Не знайдено IP для блокування.
2025-05-26 23:20:52,351 - INFO - POST https://localhost:9200/filebeat-*/_search [status:200 duration:0.029s]
2025-05-26 23:20:52,352 - INFO - Не знайдено IP для блокування.
2025-05-26 23:24:03,394 - INFO - POST https://localhost:9200/filebeat-*/_search [status:200 duration:0.031s]
2025-05-26 23:24:03,395 - INFO - Не знайдено IP для блокування.
2025-05-26 23:25:24,908 - INFO - POST https://localhost:9200/filebeat-*/_search [status:200 duration:0.027s]
2025-05-26 23:25:24,909 - INFO - Не знайдено IP для блокування.
2025-05-26 23:35:36,381 - INFO - POST https://localhost:9200/filebeat-*/_search [status:200 duration:0.030s]
2025-05-26 23:35:36,382 - INFO - Не знайдено IP для блокування.
2025-05-26 23:37:41,400 - INFO - POST https://localhost:9200/filebeat-*/_search [status:200 duration:0.028s]
2025-05-26 23:37:41,401 - INFO - Не знайдено IP для блокування.
2025-05-26 23:52:12,223 - INFO - POST https://localhost:9200/filebeat-*/_search [status:400 duration:0.300s]
2025-05-26 23:52:12,224 - ERROR - Помилка пошуку в Elasticsearch: BadRequestError(400, 'search_phase_execution_exception', 'compile error')
2025-05-26 23:52:44,517 - INFO - POST https://localhost:9200/filebeat-*/_search [status:400 duration:0.034s]
2025-05-26 23:52:44,518 - ERROR - Помилка пошуку в Elasticsearch: BadRequestError(400, 'search_phase_execution_exception', 'compile error')
2025-05-26 23:55:07,879 - INFO - POST https://localhost:9200/filebeat-*/_search [status:400 duration:0.092s]
2025-05-26 23:55:07,880 - ERROR - Помилка пошуку: BadRequestError(400, 'search_phase_execution_exception', 'compile error')
2025-05-26 23:55:07,920 - INFO - Знайдено IP 192.168.141.4 (59 спроб)
2025-05-26 23:55:07,921 - INFO - Команда: sudo ufw insert 1 deny from 192.168.141.4
2025-05-26 23:55:09,890 - INFO - INFO - Заблоковано IP: 192.168.141.4 (понад 10 спроб)
2025-05-27 00:30:01,000 - INFO - Заблоковано 192.168.141.4 за сканування 23 портів
```

Рисунок 3.20 - Опис успішного блокування атаки у лог-файлі

В свою чергу на рис. 3.20 наведено вигляд нашого лог-файлу після атак, де видно, що всі наступні сканування також були корректно заблоковані.

3.3.1 Аналіз часу реагування та точності спрацьовувань

Час реагування визначався як проміжок між моментом фіксації події в логах та додаванням правила в *UFW*. Середній час складав 185 секунд, що включало:

- 10–15 секунд на індексацію логів у Elasticsearch;
- До 3 хвилин на виконання скрипта через Cron;
- 2–5 секунд на обробку запиту та виконання команди *ufw*.

Точність вимірювалась через співвідношення істинно позитивних (TP) та хибно позитивних (FP) спрацьовувань. З 10 імітованих атак система коректно ідентифікувала 8, проте 1 випадок був пропущений через тимчасову несинхронізацію між Filebeat та Elasticsearch. Хибні спрацьовування (1 випадок) виник через помилкову класифікацію легального трафіку як шкідливого, при масових з’єднаннях з IP.

Висновки до розділу 3

Модуль продемонстрував достатню ефективність для використання в якості профілактичного засобу захисту. Поточний стан модуля демонструє високу ступінь ефективності при його використанні в якості проактивного профілактичного механізму. Його найбільша сила розкривається у сценаріях, де присутні чітко ідентифіковані сигнатури потенційно зловмисної активності, особливо у сценаріях із чіткими ознаками аномалій, такими як сканування портів або масові невдалі автентифікації.

Для подальшого вдосконалення запропоновано:

- Додаткові модулі з врахуванням більшої кількісті тривог
- Впровадження машинного навчання для виявлення складних аномалій, які не описуються статичними правилами.

- Оптимізація запитів до Elasticsearch через використання асинхронних методів та кешування.
- Інтеграція з іншими системами безпеки які використовуються в інфраструктурі
- Впровадження машинного навчання (ML) для проактивного виявлення складних загроз [25]

Реалізація запропонованих напрямків вдосконалення дозволить не лише компенсувати поточні обмеження швидкодії, а й суттєво підвищити глибину аналізу, проактивність та загальну ефективність системи, перетворивши її на потужний інструмент захисту від сучасних кіберзагроз. Модуль стане не просто детектором окремих аномалій, а комплексною платформою для аналізу безпеки, кореляції подій та автоматизації реагування.

ВИСНОВКИ

У моїй кваліфікаційній роботі було реалізовано систему автоматизованого реагування на кіберінциденти та її інтеграцію з SIEM-системою. Основною метою роботи була розробка модулів, здатних автоматично виявляти та обробляти події кібербезпеки в режимі реального часу, що було досягнуто шляхом вивчення взаємодії з SIEM-системою, створення тестового середовища та реалізацією функціонального прототипу.

У процесі дослідження:

- Проведено теоретичний огляд функціонування SIEM-систем та важливість автоматизації в процесі реагування на інциденти;
- Проаналізовано наявні підходи до інтеграції інструментів з *SIEM* із засобами автоматизації та досліджено типові проблеми;
- Обґрунтовано вибір технологій Elastic Stack як гнучкої та безкоштовної основи для побудови системи збору, аналізу та реагування на інцинденти;
- Розроблено сценарії автоматизованого реагування;
- Реалізовано рішення на мові Python з використанням відповідних бібліотек і взаємодію з Elasticsearch;
- Проведено тестування системи в умовах симуляції кібератак, які підтвердили здатність до швидкої реакції, стабільної роботи та точного виконання відповідно до їх логіки.

Практична цінність даної роботи полягає, перш за все, у створенні відкритого інструменту автоматизованого реагування, який, завдяки своїй гнучкості та адаптивності, може бути інтегрований у вже існуючу інфраструктуру інформаційної безпеки без необхідності впровадження дорогого комерційного програмного забезпечення або значних змін у поточних технологічних процесах. Такий підхід дозволяє організаціям з обмеженими ресурсами впроваджувати

ефективні засоби захисту, не витрачаючи надмірно великі кошти на ліцензії, оновлення або технічну підтримку.

Більше того, розроблене рішення значною мірою сприяє не лише скороченню часу реагування на потенційні кіберзагрози, але й забезпечує суттєве зниження навантаження на фахівців з кібербезпеки, оскільки частину рутинних завдань, пов’язаних із виявленням та первинною обробкою інцидентів, воно виконує автоматично. Це, у свою чергу, дозволяє фахівцям зосередитися на більш складних і стратегічно важливих аспектах забезпечення інформаційної безпеки.

У підсумку, результати проведеного дослідження однозначно підтверджують доцільність впровадження автоматизації в процеси реагування на інциденти, оскільки вони демонструють реальну можливість створення дієвої та надійної системи, здатної значно підвищити рівень захищеності інформаційної інфраструктури. Особливо важливим є той факт, що ефективність запропонованого підходу посилюється у разі подальшого вдосконалення системи шляхом глибшої інтеграції рішень типу SIEM із іншими компонентами екосистеми безпеки, що забезпечує комплексний і скоординований захист організації в умовах зростаючої кількості та складності кіберзагроз.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. SIEM Systems Overview. TechTarget Security. [Електронний ресурс]. URL: https://www.techtarget.com/searchsecurity/definition/SIEM (дата звернення: 03.05.2025).
2. MITRE ATT&CK Framework: Adversarial Tactics, Techniques, and Common Knowledge. MITRE Corporation. [Електронний ресурс]. URL: https://attack.mitre.org/ (дата звернення: 01.05.2025).
3. SANS Institute: Understanding SIEM Alert Volumes. SANS Institute. [Електронний ресурс]. URL: https://www.sans.org/blog/understanding-siem-alert-fatigue/ (дата звернення: 15.05.2025).
4. General Data Protection Regulation (GDPR). European Commission. [Електронний ресурс]. URL: https://gdpr-info.eu/ (дата звернення: 05.05.2025).
5. REST API: Principles and Best Practices. Red Hat Developer. [Електронний ресурс]. URL: https://developers.redhat.com/articles/rest-api-best-practices (дата звернення: 20.04.2025).
6. PCI DSS: Requirements and Security Assessment Procedures. PCI Security Standards Council. [Електронний ресурс]. URL: https://www.pcisecuritystandards.org/document_library (дата звернення: 10.06.2025).
7. Gartner Magic Quadrant for Security Information and Event Management. Gartner. [Електронний ресурс]. URL: https://www.gartner.com/reviews/market/security-information-and-event-management (дата звернення: 01.05.2025).
8. Splunk Enterprise Security: The Leading SIEM Platform. Splunk Inc. [Електронний ресурс]. URL: https://www.splunk.com/en_us/software/enterprise-security.html (дата звернення: 06.05.2025).

9. Trellix Enterprise Security Manager (ESM). Trellix (ex-McAfee). [Електронний ресурс]. URL: https://www.trellix.com/en-us/products/enterprise-security-manager.html (дата звернення: 06.05.2025).
10. IBM QRadar SIEM: Leader in Security Analytics. IBM Corporation. [Електронний ресурс]. URL: https://www.ibm.com/products/qradar-siem (дата звернення: 07.05.2025).
11. Elastic Stack: Open Source Search & Analytics. Elastic NV. [Електронний ресурс]. URL: https://www.elastic.co/elastic-stack/ (дата звернення: 03.05.2025).
12. Endpoint Detection and Response (EDR): Explained. CrowdStrike. [Електронний ресурс]. URL: https://www.crowdstrike.com/cybersecurity-101/endpoint-security/endpoint-detection-and-response-edr/ (дата звернення: 22.03.2025).
13. AI and Machine Learning in Cybersecurity. CSO Online. [Електронний ресурс]. URL: https://www.csoonline.com/article/ai-machine-learning-cybersecurity.html (дата звернення: 30.05.2025).
14. Endpoint Protection Platforms (EPP): A Complete Guide. Palo Alto Networks. [Електронний ресурс]. URL: https://www.paloaltonetworks.com/cyberpedia/what-is-an-endpoint-protection-platform-epp (дата звернення: 14.04.2025).
15. Application Programming Interface (API). Wikipedia. [Електронний ресурс]. URL: https://en.wikipedia.org/wiki/API (дата звернення: 11.05.2025).
16. Security Automation vs. Orchestration: Understanding the Difference. Swimlane. [Електронний ресурс]. URL: https://swimlane.com/blog/security-automation-vs-orchestration/ (дата звернення: 09.06.2025).
17. Python Documentation: Official Tutorial. Python Software Foundation. [Електронний ресурс]. URL: https://docs.python.org/3/tutorial/index.html (дата звернення: 01.04.2025).

18. Challenges in SOAR Implementation. Cyberbit. [Електронний ресурс]. URL: https://www.cyberbit.com/blog/soar/challenges-in-soar-implementation/ (дата звернення: 18.05.2025).
19. Oracle VM VirtualBox User Manual. Oracle Corporation. [Електронний ресурс]. URL: https://www.virtualbox.org/manual/UserManual.html (дата звернення: 25.03.2025).
20. Uncomplicated Firewall (UFW) Community Documentation. Ubuntu Community Wiki. [Електронний ресурс]. URL: https://help.ubuntu.com/community/UFW (дата звернення: 02.06.2025).
21. Filebeat: Lightweight Shipper for Logs. Elastic NV. [Електронний ресурс]. URL: https://www.elastic.co/beats/filebeat (дата звернення: 04.05.2025).
22. Elasticsearch Python Client (elasticsearch-py). Elastic NV. [Електронний ресурс]. URL: https://www.elastic.co/guide/en/elasticsearch/client/python-api/current/index.html (дата звернення: 07.05.2025).
23. Configuring Filebeat. Elastic NV. [Електронний ресурс]. URL: https://www.elastic.co/guide/en/beats/filebeat/current/configuring-howto-filebeat.html (дата звернення: 10.06.2025).
24. Nmap Network Scanning: The Official Nmap Project Guide. Nmap Project. [Електронний ресурс]. URL: https://nmap.org/book/ (дата звернення: 15.04.2025).
25. Machine Learning for Threat Detection in Cybersecurity. Towards Data Science. [Електронний ресурс]. URL: https://towardsdatascience.com/machine-learning-for-threat-detection-in-cybersecurity-9f5a4003b1a4 (дата звернення: 28.05.2025).

ДОДАТКИ

Додаток А

ПРИКЛАД НАЛАШТУВАННЯ FILEBEAT

Скріни налаштування в системі:

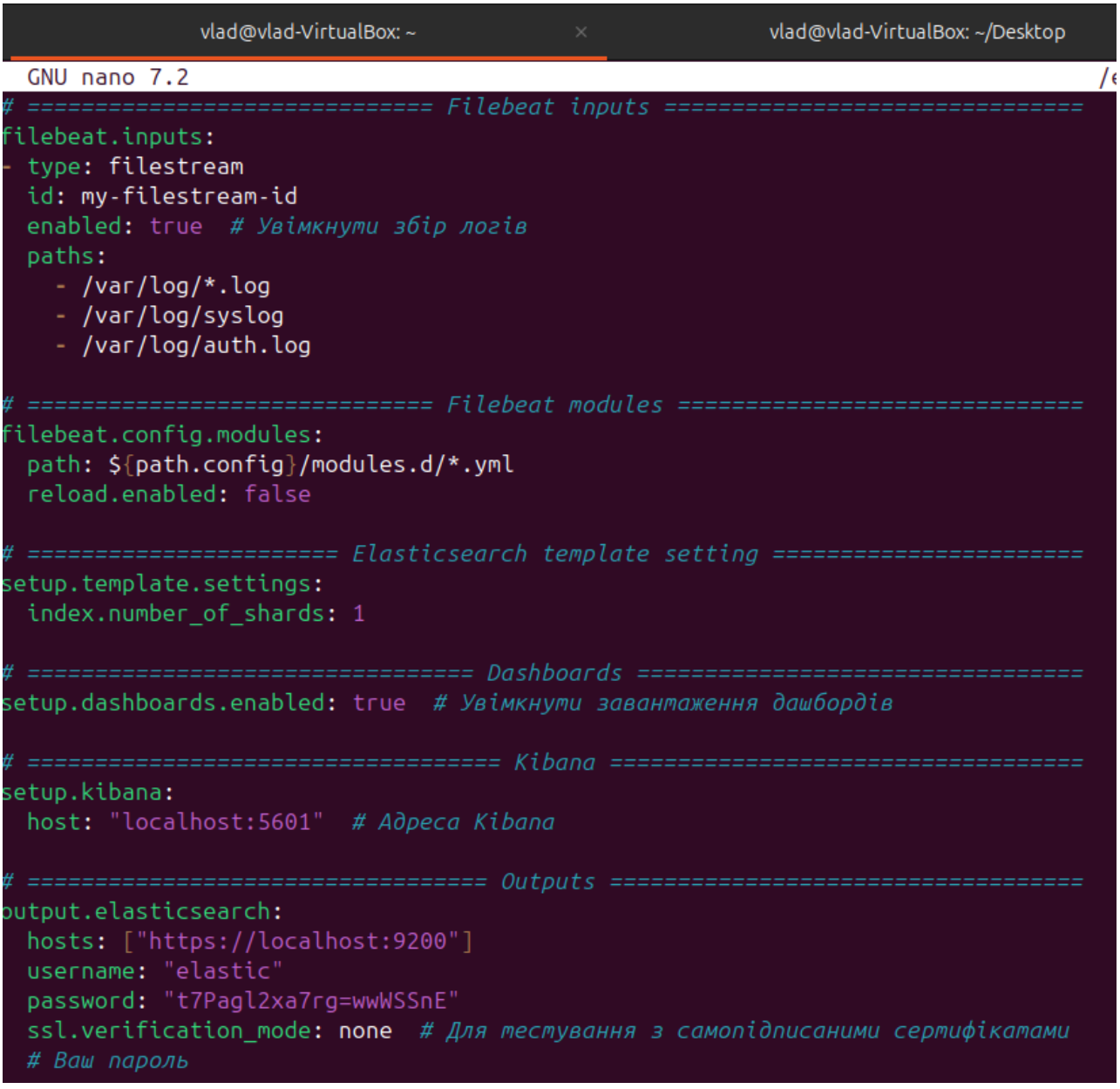

Рисунок А.1 Конфігурація файлу filebeat.yml

Продовження додатку А

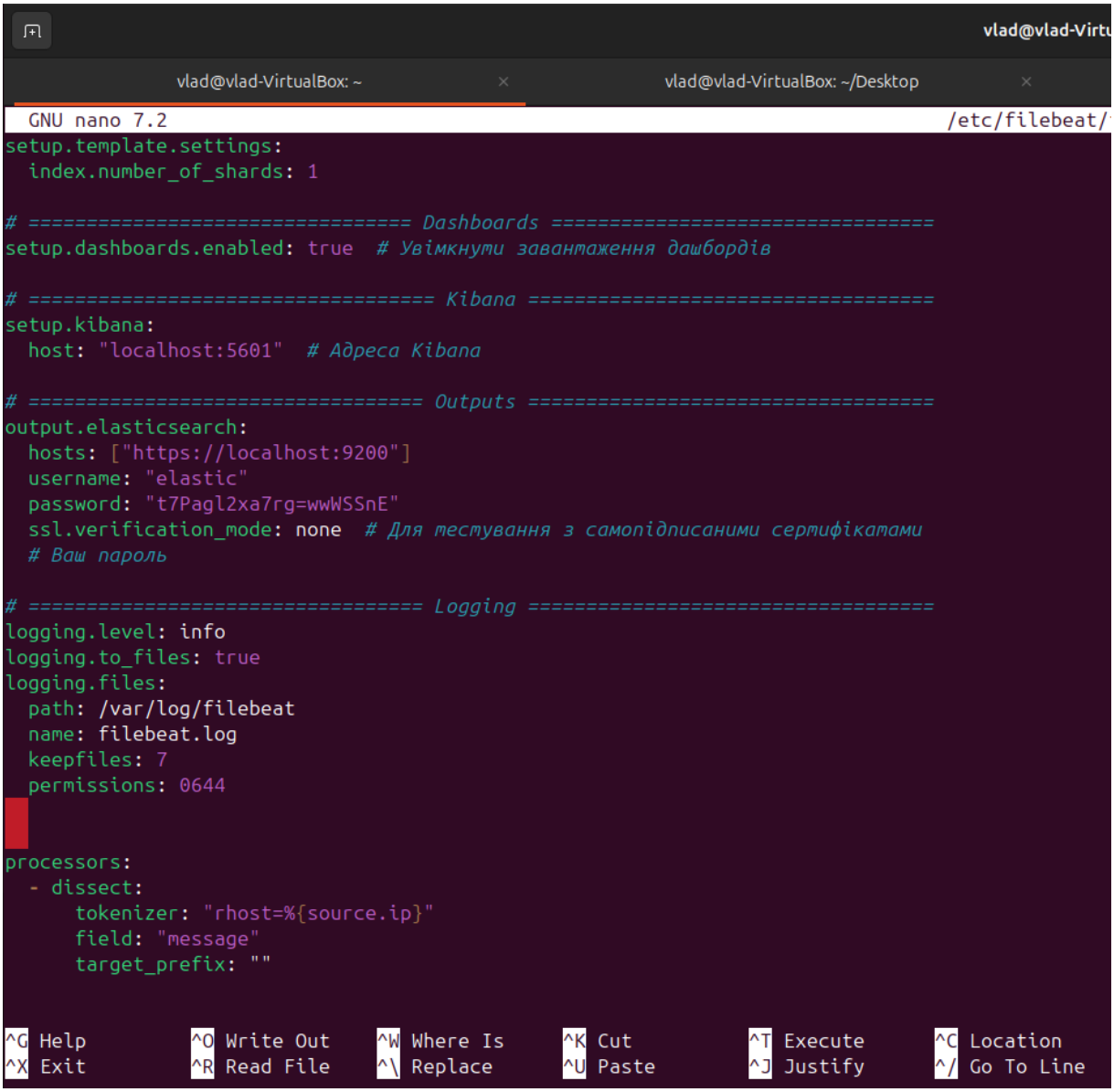

Рисунок А.2 - Конфігурація файлу filebeat.yml

Додаток Б

ПРОГРАМНИЙ КОД РОЗРОБЛЕНИХ МОДУЛІВ

Програмний код до ScanDetect:

```
#!/usr/bin/env python3
import os
from elasticsearch import Elasticsearch
import logging
from datetime import datetime
ELASTIC_HOST = "https://localhost:9200"
ELASTIC_USER = "elastic"
ELASTIC_PASSWORD = "t7Pagl2xa7rg=wwWSSnE"
MIN_PORTS = 20  # Блокувати при ≥20 унікальних портів
LOG_FILE = "/home/vlad/siem_auto.log"
# Налаштування логування
logging.basicConfig(
    filename=LOG_FILE,
    level=logging.INFO,
    format='%(asctime)s - %(levelname)s - %(message)s',
    datefmt='%Y-%m-%d %H:%M:%S'
)
# Підключення до Elasticsearch
es = Elasticsearch(
    ELASTIC_HOST,
    basic_auth=(ELASTIC_USER, ELASTIC_PASSWORD),
    verify_certs=False,
    headers={"Accept": "application/vnd.elasticsearch+json; compatible-with=8"}
)
def block_port_scanners():
    try:
        # Запит до Elasticsearch
        response = es.search(
            index="filebeat-*",
            body={
                "size": 0,
                "query": {"range": {"@timestamp": {"gte": "now-5m"}}},
                "aggs": {
```

Продовження додатку Б

```
                "ips": {
                    "terms": {"field": "source.ip", "size": 100},
                    "aggs": {"ports_count": {"cardinality": {"field": "destination.port"}}}
                }
            }
        }
    )
    # Блокування IP
    for bucket in response['aggregations']['ips']['buckets']:
        ip = bucket['key']
        count = bucket['ports_count']['value']

        if count >= MIN_PORTS:
            os.system(f"sudo ufw insert 1 deny from {ip}")
            logging.info(f"Заблоковано {ip} за сканування {count} портів")
  except Exception as e:
    logging.error(f"Помилка: {str(e)}")
if __name__ == "__main__":
  block_port_scanners()
```

Програмний код до BruteForce:

```
#!/usr/bin/env python3
import os
from elasticsearch import Elasticsearch
import logging
from datetime import datetime
# Налаштування (зміни за потребою)
ELASTIC_HOST = "https://localhost:9200"
ELASTIC_USER = "elastic"
ELASTIC_PASSWORD = "t7Pagl2xa7rg=wwWSSnE"
MIN_PORTS = 20  # Блокувати при ≥20 унікальних портів
LOG_FILE = "/home/vlad/siem_auto.log"
# Налаштування логування
logging.basicConfig(
  filename=LOG_FILE,
  level=logging.INFO,
```

Продовження додатку Б

```
    format='%(asctime)s - %(levelname)s - %(message)s',
    datefmt='%Y-%m-%d %H:%M:%S'
)
# Підключення до Elasticsearch
es = Elasticsearch(
    ELASTIC_HOST,
    basic_auth=(ELASTIC_USER, ELASTIC_PASSWORD),
    verify_certs=False,
    headers={"Accept": "application/vnd.elasticsearch+json; compatible-with=8"}
)
def block_port_scanners():
    try:
        # Запит до Elasticsearch
        response = es.search(
            index="filebeat-*",
            body={
                "size": 0,
                "query": {"range": {"@timestamp": {"gte": "now-5m"}}},
                "aggs": {
                    "ips": {
                        "terms": {"field": "source.ip", "size": 100},
                        "aggs": {"ports_count": {"cardinality": {"field": "destination.port"}}}
                    }
                }
            }
        )
        # Блокування IP
        for bucket in response['aggregations']['ips']['buckets']:
            ip = bucket['key']
            count = bucket['ports_count']['value']
            if count >= MIN_PORTS:
                os.system(f"sudo ufw insert 1 deny from {ip}")
                logging.info(f"Заблоковано {ip} за сканування {count} портів")
    except Exception as e:
        logging.error(f"Помилка: {str(e)}")
if __name__ == "__main__":
    block_port_scanners()
```

Додаток В

ГРАФІЧНЕ ЗОБРАЖЕННЯ СИСТЕМИ АВТОМАТИЗОВАНОГО РЕАГУВАННЯ НА ІНЦИДЕНТИ ДЛЯ SIEM-СИСТЕМИ

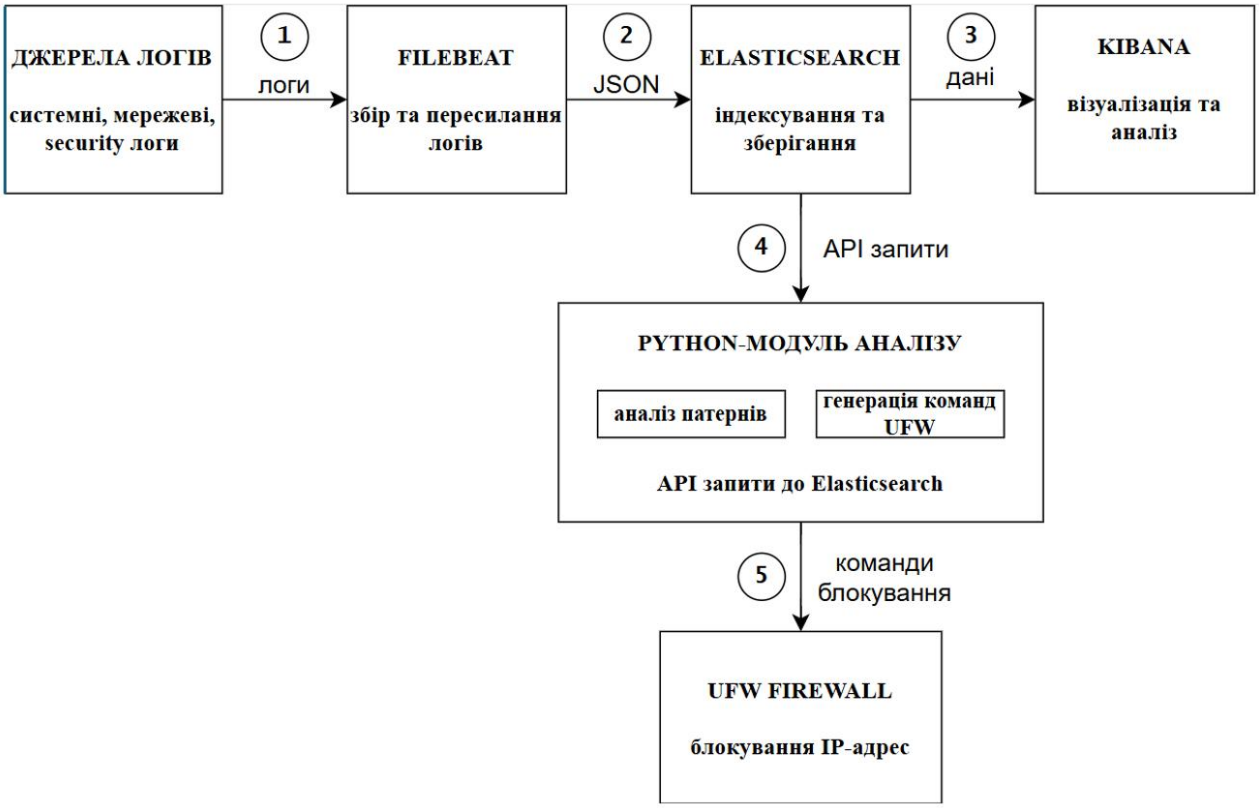

Рисунок В.1 - Графічне зображення системи