МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки

ХМЕЛИК Вадим Іванович

Операційний центр безпеки на основі інструментів з відкритим кодом / Security Operations Center Based on Open Source Tools

спеціальність: 125 – Кібербезпека та захист інформації

освітньо-професійна програма –Кібербезпека

Кваліфікаційна робота

Виконав студент групи КБм -21
В.І. Хмелик

_____

Науковий керівник
д.т.н., професор В.В. Яцків

_____

Кваліфікаційну роботу допущено
до захисту:

«____» _____ ___2025 р.

Завідувач кафедри

_____ В.В.Яцків

ТЕРНОПІЛЬ - 2025

Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки
Освітній ступінь «магістр»
спеціальність: 125 - Кібербезпека та захист інформації
освітньо-професійна програма –Кібербезпека

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ В.В.Яцків
____" _____ 2024 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

ХМЕЛИКУ Вадиму Івановичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи:
Операційний центр безпеки на основі інструментів з відкритим кодом / Security Operations Center Based on Open Source Tools
керівник роботи д.т.н., професор В.В. Яцків

затверджені наказом по університету від 20 грудня 2024 року № 938

2. Строк подання студентом закінченої кваліфікаційної роботи 5 грудня 2025 року.
3. Вихідні дані до кваліфікаційної роботи: завдання на випускну кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:
– дослідити принципи побудови та функціонування операційного центру безпеки;
– провести аналіз інструментів з відкритим кодом, що використовуються для побудови операційного центру безпеки;
– розробити архітектуру операційного центру безпеки на основі інструментів з відкритим кодом;
– реалізувати практичний сценарій роботи операційного центру безпеки;
– дослідити можливості запропонованого рішення.

5. Перелік графічного матеріалу у роботі.
– архітектура запропонованого рішення;
– схема побудови SOC;
– схема обробки подій під час тестування SOC.

6. Консультанти розділів кваліфікаційної роботи

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

7. Дата видачі завдання 20 грудня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строки виконання етапів кваліфікаційної роботи	Примітка
1	Дослідження принципів побудови операційного центру безпеки	12.2024 р. – 03.2025 р.	
2	Аналіз інструментів з відкритим кодом для побудови SOC	03.2025 р. – 05.2025 р.	
3	Проєктування та реалізація SOC на основі відкритого програмного забезпечення	05.2025 р. – 11.2025 р.	

Студент _____ В.І. Хмелик
(підпис)

Керівник роботи _____ д.т.н., професор Яцків В.В.
(підпис)

АНОТАЦІЯ

Хмелик В.І. Операційний центр безпеки на основі інструментів з відкритим кодом. – Рукопис.

Дослідження на здобуття освітнього ступеня «магістр» за спеціальністю 125 «Кібербезпека та захист інформації», освітньо-професійна програма «Кібербезпека». – Західноукраїнський національний університет, Тернопіль, 2025.

У роботі досліджено принципи побудови та функціонування SOC на базі інструментів з відкритим кодом. Обґрунтовано вибір і взаємодію компонентів, що забезпечують повний цикл роботи SOC у межах трирівневої архітектури. Розроблено та протестовано модель внутрішнього SOC, яка забезпечує централізований моніторинг, підвищення рівня кіберстійкості та ефективності реагування при мінімальних витратах.

Ключові слова: ОПЕРАЦІЙНИЙ ЦЕНТР БЕЗПЕКИ, РОЗВІДКА КІБЕРЗАГРОЗ, ІНФОРМАЦІЙНА БЕЗПЕКА.

ABSTRACT

Khmelyk V.I. Security Operations Center Based on Open Source Tools. – Manuscript.

Doctoral studies for the education level «Master» with the title 125 «Cybersecurity and Information Protection». – West Ukrainian National University, Ternopil, 2025.

The thesis investigates the principles of building and operating a Security Operations Center (SOC) based on open-source tools. The selection and interaction of components ensuring the full SOC lifecycle within a three-layer architecture are substantiated. A model of an internal SOC has been developed and tested, providing centralized monitoring, enhanced cyber resilience, and effective incident response with minimal costs.

Keywords: SECURITY OPERATIONS CENTER, CYBER THREAT INTELLIGENCE, INFORMATION SECURITY.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

SOC (Security Operations Center) - операційний центр безпеки;

ISMS (Information Security Management System) - система управління інформаційною безпекою;

IDS (Intrusion Detection System) – система виявлення вторгнень;

IPS (Intrusion Prevention System) – система запобігання вторгненням;

SIEM (Security Information and Event Management) – система управління подіями безпеки;

SOAR (Security Orchestration, Automation and Response) – оркестрація, автоматизація та реагування на інциденти безпеки;

MTTD (Mean Time to Detect) – середній час виявлення;

MTTR (Mean Time to Respond) – середлній час реагуваня;

FIM (File Integrity Monitoring) – контроль цілісності файлів;

TIP (Threat Intelligence Platform) – платформи аналізу загроз;

IoC (Indicators of Compromise) – індикатори компрометації;

IR (Incident Response) – реагування на інциденти;

IM (Incident Management) – керування інцидентами;

OS (Operating System) – операційна система;

CTI (Cyber Threat Intelligence) – розвідка кіберзагроз;

ВСТУП

Актуальність теми. Сучасний розвиток цифрових технологій істотно змінює способи ведення бізнесу, управління інформаційними системами та взаємодії користувачів у мережі. Водночас зростає кількість кіберзагроз, що становлять серйозну небезпеку для конфіденційності, цілісності та доступності інформаційних ресурсів. Кібератаки стають не лише частішими, а й більш складними, що вимагає від організацій переходу від реактивного до проактивного підходу у сфері кіберзахисту.

Одним із ключових елементів сучасної системи управління інформаційної безпеки (ISMS) [1-3] є операційний центр безпеки (SOC), що призначений для безперервного моніторингу, виявлення, аналізу та реагування на інциденти (IR) інформаційної безпеки в реальному часі та та керування ними (IM) [4-7]. Наявність SOC дозволяє організації централізовано відстежувати події безпеки, зменшувати час реагування на інциденти, своєчасно усувати вразливості та підвищувати загальний рівень кіберстійкості.

З огляду на тенденції розвитку відкритих технологій, дедалі більшого поширення набувають рішення на основі інструментів з відкритим програмним кодом, що дозволяють забезпечити гнучкість налаштування, прозорість реалізації, можливість інтеграції різнорідних систем і суттєво знижують витрати на впровадження SOC.

Мета і завдання дослідження. Метою роботи є дослідження принципів побудови та проєктування SOC на основі інструментів з відкритим кодом.

Відповідно до мети, необхідно вирішити такі завдання:

- дослідити принципи побудови та функціонування SOC;
- проаналізувати інструменти з відкритим кодом, що використовуються для побудови SOC;
- реалізувати архітектуру та практичний сценарій роботи SOC;
- дослідити можливості запропонованого рішення.

Об’єкт дослідження - процес забезпечення безпеки інформаційних систем шляхом моніторингу, аналізу та реагування на інциденти.

Предмет дослідження - архітектура, принципи функціонування та взаємодія компонентів SOC, реалізованих із використанням відкритих програмних рішень.

Методи досліджень: аналітичні методи для вивчення сучасних підходів до побудови SOC та їх архітектурних моделей; мтоди порівняння для оцінки можливостей і ефективності інструментів з відкритим кодом; експериментальні методи для оцінки компонентів взаємодії, часу реагування та можливостей автоматизованого аналізу подій.

Наукова новизна отриманих результатів. Запропоноване рішення забезпечує інтеграцію процесів моніторингу, аналітики, обробки інцидентів і обміну розвідувальними даними в єдиній SOC на основі інструментів з відкритим кодом. Обґрунтовано доцільність поєднання систем SIEM, SOAR і CTI у межах єдиної архітектури з метою підвищення рівня кіберстійкості інформаційних систем.

Практичне значення отриманих результатів. Запропоноване рішення може бути використане для побудови або вдосконалення SOC у державних, корпоративних та освітніх установах із використанням відкритих технологій без необхідності придбання комерційного ПЗ.

Публікації та апробація кваліфікаційної роботи.

1. Хмелик В., Давлетов Р. Дослідження побудови операційного центру безпеки / Збірник матеріалів науково-практичної конференції молодих вчених, аспірантів та студентів «Кібербезпека та комп’ютерно-інтегровані технології» (КБКІТ - 2025), Тернопіль, 2025.

2. Хмелик В. Дослідження архітектури операційного центру безпеки / Збірник матеріалів науково-практичного симпозіуму «Захист інформації», Тернопіль, 2025.

1. ДОСЛІДЖЕННЯ ПРИНЦИПІВ ПОБУДОВИ ОПЕРАЦІЙНОГО ЦЕНТРУ БЕЗПЕКИ

1.1 Функціонування операційного центру безпеки

SOC - це організаційна та технологічна структура, призначена для централізованого моніторингу, виявлення, аналізу та реакції на події інформаційної безпеки в режимі реального часу.

SOC функціонує як ядро системи управління інформаційною безпекою (ISMS), забезпечуючи взаємодію між технічними засобами захисту, аналітичними інструментами, базами знань про загрози, а також персоналом, який здійснює моніторинг і реагування (рисунок 1.1) [6-8].

Рисунок 1.1 – Операційний центр безпеки

Завдяки інтеграції різнорідних джерел даних, наприклад, систем виявлення / запобігання вторгнень (IDS/IPS), антивірусів, файрволів, систем контролю доступу, серверів логів тощо, SOC дозволяє створити єдину картину безпеки інформаційного середовища.

Основною метою SOC є забезпечення безперервного контролю за станом інформаційної інфраструктури, оперативне IR та IM, а також підвищення рівня захищеності організації.

Основними завданнями SOC є [7-10]:

- моніторинг подій безпеки – збір, кореляція та аналіз журналів подій з усіх компонентів IT-інфраструктури за допомогою систем управління подіями безпеки (SIEM).
- виявлення та класифікація інцидентів – ідентифікація аномалій, відхилень від базової поведінки або збігів із відомими сигнатурами атак.
- оцінювання ризиків і пріоритезація – визначення критичності інцидентів відповідно до впливу на активи організації.
- реагування IR – ініціювання відповідних дій (ізоляція вузлів, блокування облікових записів, активація плейбуків реагування).
- аналіз та розслідування – дослідження цифрових слідів для встановлення джерела атаки, методів проникнення та наслідків.
- звітність і аудит – формування аналітичних звітів для оцінювання ефективності заходів безпеки та дотримання політик.

Оцінка ефективності SOC проводиться за часовими показниками, що характеризують здатність системи швидко виявляти та усувати загрози [11-14]:

- MTTD (Mean Time to Detect) - середній час який, потрібен для детекції інциденту від моменту його виникнення

$$MTTD = \frac{\sum_{i-1}^{n}(t_{di} - t_{oi})}{n},$$

де t_{oi} - час виникнення інциденту і;

t_{di} - час виявлення і-го інциденту;

n - кількість проаналізованих інцидентів;

- MTTR (Mean Time to Respond) - середній час, витрачений на IR після виявлення інциденту

$$MTTR = \frac{\sum_{i-1}^{n}(t_{ri} - t_{di})}{n},$$

де t_{ri}- час повного усунення або відновлення системи.

SOC функціонує на перетині технологічних, організаційних і аналітичних процесів, поєднуючи засоби моніторингу, збору даних, кореляції

подій, IR та управління ризикам ідозволяє забезпечити безперервний контроль безпеки інформаційних ресурсів.

Основні функції SOC охоплюють такі ключові напрями:

- виявлення (Detection) – безперервний моніторинг подій безпеки для ідентифікації підозрілої активності чи відхилень у роботі систем;
- аналіз (Analysis) – перевірка, кореляція та класифікація інцидентів з метою визначення їх природи, масштабів та потенційного впливу;
- реагування (Response) – оперативне усунення загроз, ізоляція уражених систем, відновлення безпечного стану інфраструктури;
- звітування (Reporting) – документування інцидентів, формування аналітичних звітів і рекомендацій для підвищення рівня кібернетичної безпеки.

На рисунку 1.2 наведено модель, яка відображає ієрархію процесів і процедур, що забезпечують функціонування SOC [5-9].

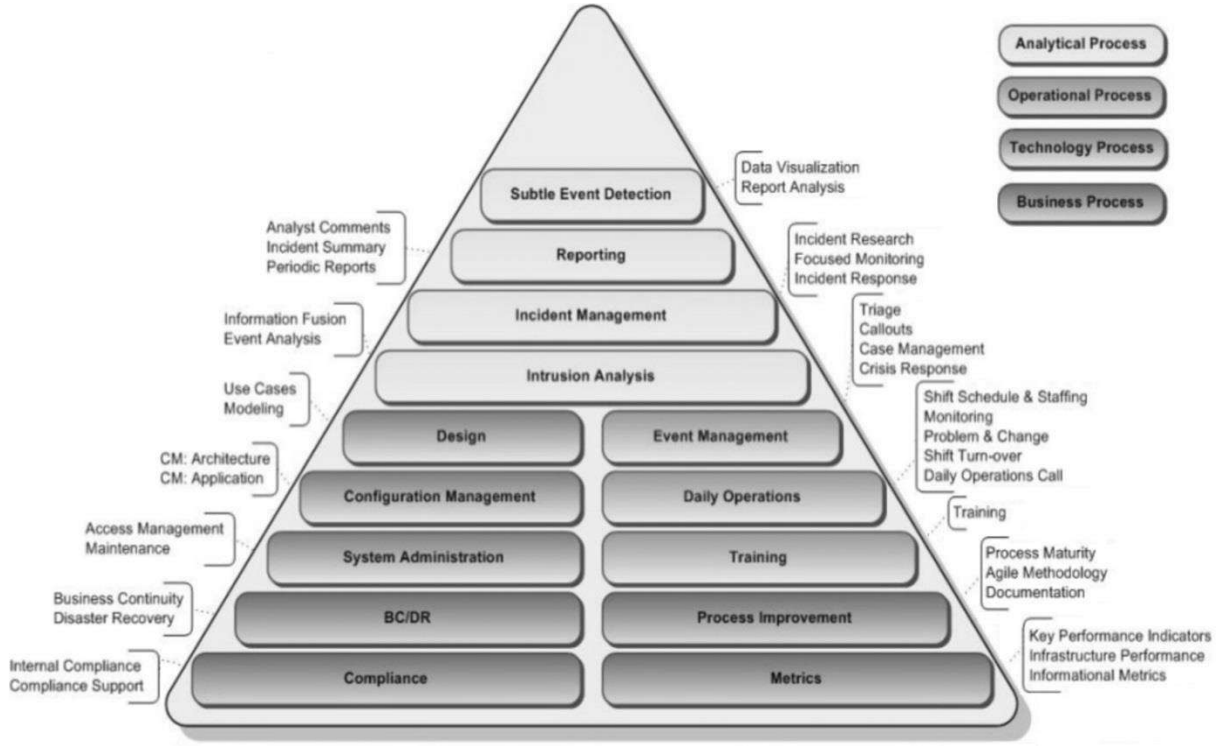

Рисунок 1.2 - Процеси та процедури SOC

Дана модель складається з 14 основних процесів та 36 процедур, розділених на чотири основні групи, що наведено в таблиці 1.1.

Таблиця 1.1 – Групи процесів, що забезпечують функціонування SOC

Група процесів	Призначення
Аналітичні	Відповідають за виявлення, класифікацію та аналіз інцидентів, звітування про інциденти безпеки, тобто є ядром SOC-операцій
Операційні	Забезпечують постійну роботу SOC, щоденне управління подіями, моніторинг, контроль робочих процесів, планування змін, навчання і розвиток персоналу
Технологічні	Відповідають за інфраструктурну підтримку SOC, оновлення систем, налаштування програмного забезпечення, адміністрування, резервування тощо.
Бізнес-процеси	Формують стратегічну основу SOC, забезпечують відповідність нормативам, планування безперервності бізнесу, контроль ефективності, стратегічне управління.

1.2 Архітектурні моделі побудови SOC

Архітектура SOC визначає організаційно-технічну структуру, взаємозв'язок компонентів та інформаційні потоки, необхідні для забезпечення повного циклу моніторингу, виявлення, аналізу та IR інформаційної безпеки [7-10]. Ефективна архітектура SOC повинна забезпечувати централізовану обробку даних з різних джерел, інтеграцію з інфраструктурою організації та можливість масштабування відповідно до зростання обсягів інформації.

Архітектуру SOC можна умовно поділити на три рівні:

- рівень збору даних – охоплює джерела подій безпеки, які формують вхідний потік інформації;
- аналітичний рівень – здійснює обробку, нормалізацію, кореляцію та аналіз даних за допомогою SIEM-систем та інструментів аналітики загроз;
- рівень реагування – забезпечує оперативні дії з ліквідації або

мінімізації наслідків інцидентів, а також формування звітності та зворотного зв'язку для вдосконалення політик безпеки.

Взаємодія між рівнями забезпечує безперервний цикл виявлення, аналізу та IR.

З метою підвищення ефективності функціонування SOC доцільно використовувати різні технології в межах єдиної архітектури (рисунок 1.3), що формалізує модель роботи центру з точки зору компонентів і взаємозв'язків між ними [15].

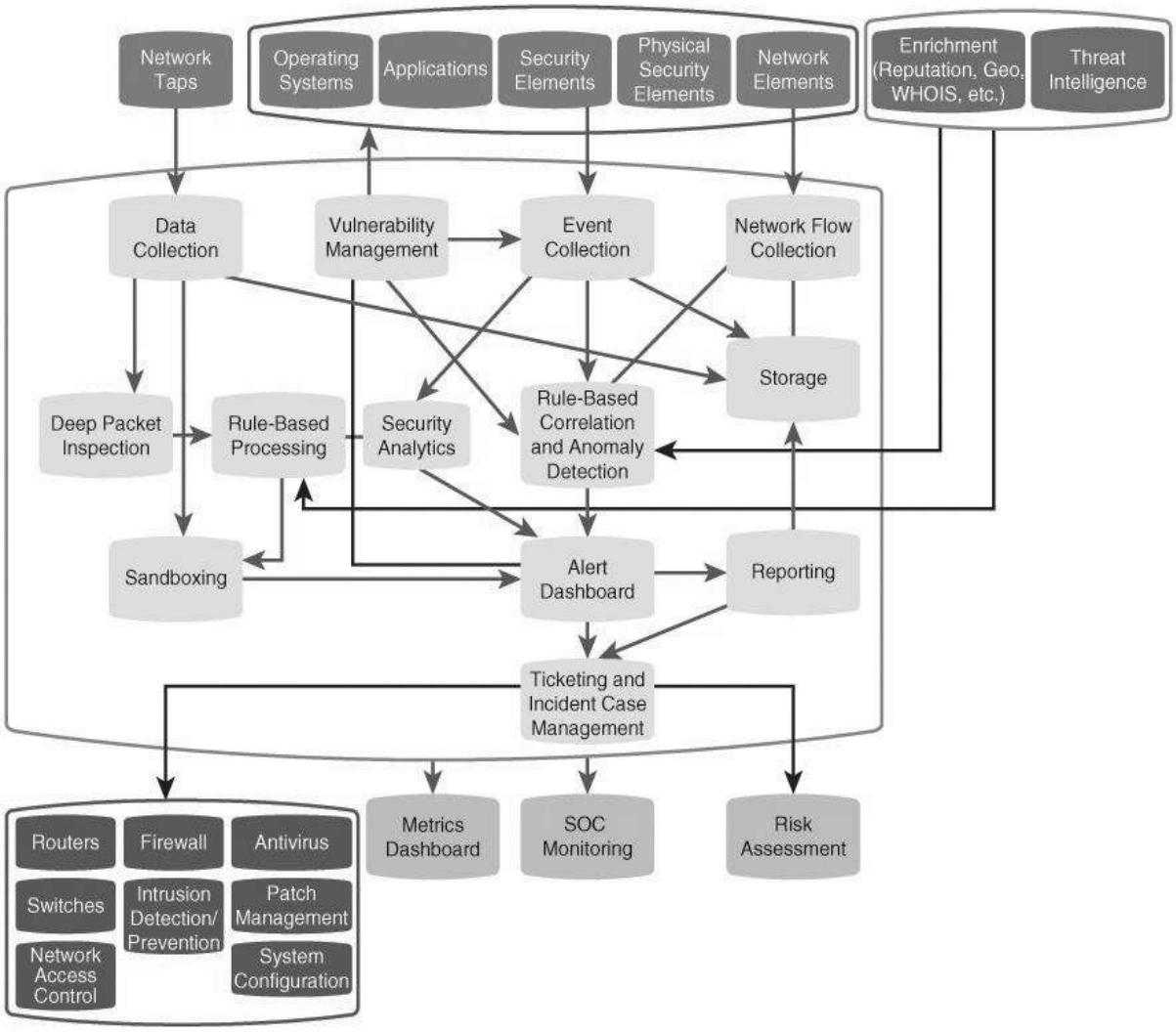

Рисунок 1.3 – Концептуальна архітектура SOC

Це базова модель побудови внутрішнього SOC, у якому організація володіє, адмініструє і контролює всі процеси безпеки. Така архітектура описує:

– вхідні дані SOC з позиції категоризованих джерел (агенти, мережеві сенсори, зовнішні TI-фіди);

– результати діяльності SOC – сповіщення, звіти, аналітичні висновки або автоматизовані дії;

– ключові технології (SIEM, IDS/IPS, SOAR, CTI, IR);

– взаємозв'язки між цими технологіями, що забезпечують цілісний моніторинг і IR;

– області збору метрик ефективності – частота сповіщень, MMTR, точність кореляції тощо.

На практиці також розглядають архітектури керованого захисту від загроз, які реалізуються великими постачальниками послуг безпеки, наприклад Cisco (рисунок 1.4) [15].

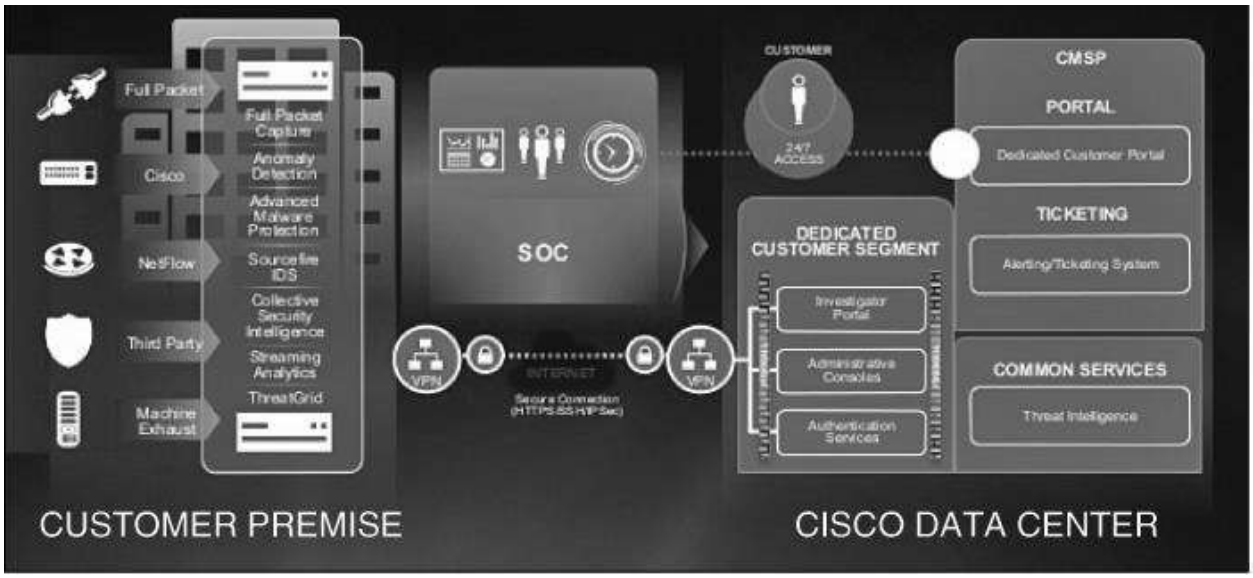

Рисунок 1.4 – Архітектура керованого захисту від загроз

Така модель забезпечує можливість передачі окремих або всіх функцій SOC, наприклад функції моніторингу IR та IM, зовнішньому постачальнику послуг безпеки, зберігаючи при цьому контроль за інцидентами на рівні організації.

Залежно від масштабу організації, процесів безпеки та доступних ресурсів, архітектура SOC може бути побудована з використанням різних моделей. Найбільш поширеними є наступні типи [15-17]:

– монолітна – усі функції (SIEM, IDS, FIM, SOAR) реалізовані в

одному комплексному рішенні, наприклад, Security Onion або Wazuh All-in-One. Такий підхід зручний для малих організацій та середовищ.

- модульна – окремі компоненти SOC (Wazuh, Suricata, Zeek, MISP, Shuffle) розміщені на різних вузлах і взаємодіють через API та черги повідомлень. Цей підхід забезпечує гнучкість, масштабованість та можливість розподіленої обробки даних.

Вибір архітектурної моделі SOC залежить від масштабу інфраструктури, рівня автоматизації процесів, кількості подій безпеки, що обробляються, та вимог до часу виявлення і IR. Такий підхід дозволяє організаціям адаптувати структуру SOC відповідно до власних потреб і ресурсних можливостей, забезпечуючи оптимальний баланс між ефективністю, вартістю та керованістю системи безпеки.

1.3 Компоненти архітектури SOC

Компоненти SOC [4-10] можна розподілити відповідно до рівнів функціональної архітектури, зокрема збору даних, аналітики та реагування. Такий підхід дозволяє системно представити всі елементи SOC за їхнім призначенням і роллю у забезпеченні повного циклу управління інформаційною безпекою.

1.3.1 Джерела даних і механізми збору інформації

Основою функціонування SOC є система збору даних про події інформаційної безпеки. Джерела даних формують основу аналітичної моделі безпеки. До них належать:

- системні журналиопераційних систем;
- мережеві події з маршрутизаторів, файрволів і IDS/IPS;
- події з антивірусних систем, проксі, VPN;
- моніторинг активності користувачів і FIM;
- телеметрія з хмарних платформ тощо.

Сукупність таких джерел формує інформаційну основу для виявлення

загроз і дозволяє здійснювати багаторівневий моніторинг інфраструктури.

Збір і агрегування подій здійснюється за допомогою агентів або протоколів передачі даних, які передають журнали подій до центрального сховища SIEM або Wazuh Manager. Передача даних здійснюється через безпечні канали (TLS, VPN, HTTPS) з використанням черг повідомлень або брокерів (Kafka, Redis). У сучасних реалізаціях SOC, таких як Wazuh або Security Onion, ці функції інтегровані безпосередньо в ядро системи та забезпечують централізований збір даних у реальному часі.

Ключову роль у SOC відіграють SIEM-системи, які забезпечують централізовані збір, зберігання та аналіз подій.

1.3.2 Системи моніторингу та кореляції подій

Наступним важливим компонентом SOC є система моніторингу, аналізу та кореляції подій, яка дозволяє виявляти підозрілі активності й порушення політик безпеки. Ключову роль тут займають SIEM, які виконують нормалізацію, збагачення, аналіз і кореляцію даних з усіх компонентів інфраструктури. Завдяки цим функціям SIEM є центральним аналітичним ядром SOC, формуючи сповіщення, звіти та правила виявлення аномалій.

Для аналізу мережевої активності використовуються системи мережевого моніторингу та аналізу трафіку (NIDS/NIPS, NTA). Вони здійснюють глибоку інспекцію пакетів, виявляють сканування портів, спроби експлуатації вразливостей або передавання шкідливого контенту.

Моніторинг кінцевих точок забезпечують рішення класу EDR/XDR, які відстежують процеси, файли, служби й активність користувачів на робочих станціях. Прикладом є Wazuh Agent [18], що виконує збір подій із системного журналу, перевірку цілісності файлів FIM та виявлення поведінкових аномалій.

Окреме місце займають платформи аналітзу загроз (TIP), такі як MISP (Malware Information Sharing Platform) [19], які дозволяють накопичувати та обмінюватися індикаторами компрометації (IoC) з внутрішніх і зовнішніх

джерел, збагачуючи контекст аналізу та підвищуючи точність виявлення нових типів атак.

На сучасному етапі популярними SIEM-рішеннями є Wazuh, ELK Stack, Splunk, IBM QRadar, Azure Sentinel [21, 22]. В контексті open-source екосистеми особливе значення має Wazuh, який об'єднує функції збору даних, IDS, FIM і кореляції подій в єдиній платформі. Аналітичні панелі забезпечують візуалізацію подій у реальному часі, що дозволяє скоротити час виявлення загроз і підвищити ефективність аналітиків.

1.3.3 Інцидент-менеджмент і реагування

Інцидент-менеджмент – це процес, який забезпечує структуроване IR та виявлені події безпеки. Його мета швидко ідентифікувати атаку, обмежити її наслідки, усунути причини та відновити нормальну роботу систем.

Типовий життєвий цикл інциденту включає наступні етапи:

- ідентифікація та реєстрація події;
- аналіз і підтвердження інциденту;
- локалізація та усунення загрози;
- відновлення та документування результатів;
- постінцидентний аналіз для вдосконалення процедур реагування.

Заключним етапом у роботі SOC є керування інцидентами (IM) та реагування на них (IR). Для цього використовуються системи SOAR [23], наприклад, Shuffle [24], TheHive [25], Cortex [26], які узгоджують взаємодію між різними системами безпеки та автоматизують повторювані операції з реагування. Вони дозволяють створювати типові сценарії реагування (playbooks). Використання SOAR дозволяє мінімізувати час між виявленням і нейтралізацією інциденту, а також зменшити навантаження на аналітиків.

Усі події, журнали та результати розслідувань зберігаються у центральному сховищі даних. Аналітична платформа формує дашборди та звіти про динаміку загроз, тенденції атак і ефективність контрзаходів.

Архітектура SOC доповнюється командою аналітиків, поділених за рівнями компетентності (рисунок 1.5):

– перша лінія підтримки – інженери первинної обробки, що виявляють інциденти й передають підтверджені події далі;

– друга лінія – аналітики SOC, які проводять кореляцію логів, оцінку впливу та класифікацію загроз;

– третя лінія – адміністратори безпеки й інженери, які здійснюють технічне реагування, відновлення систем і розробку політик безпеки.

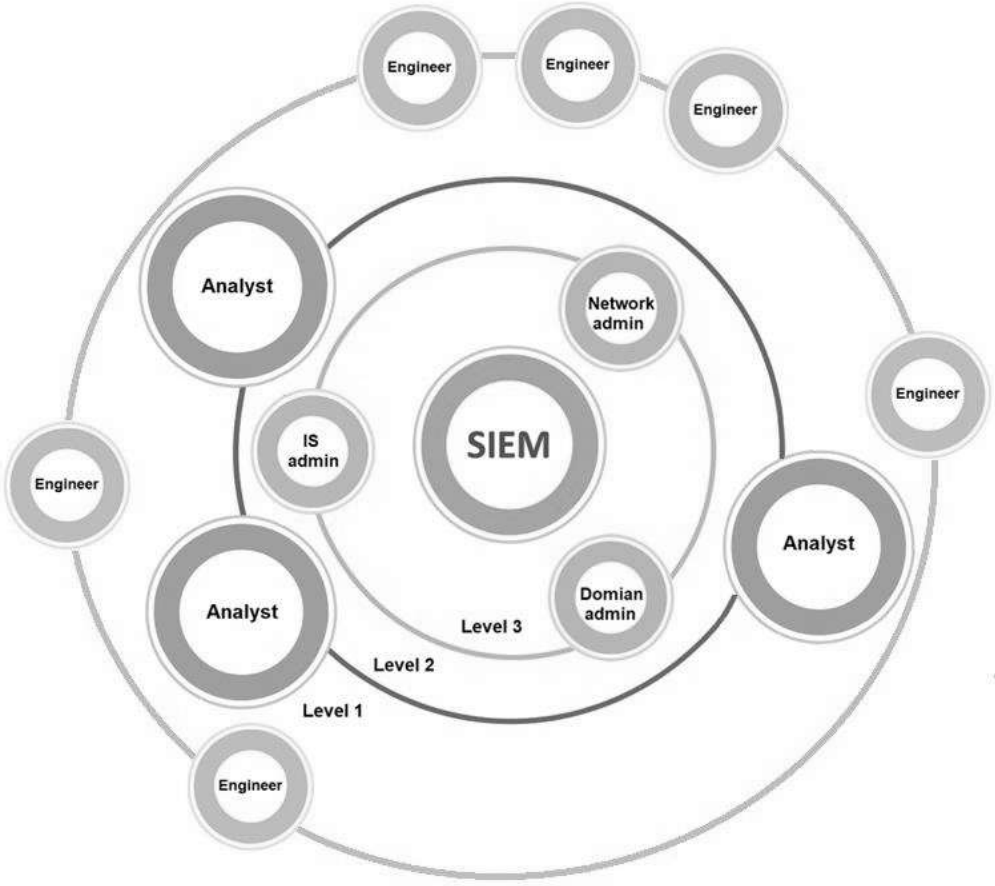

Рисунок 1.5 – Організаційна ієрархія SOC

Компоненти SOC утворюють взаємопов’язану систему, яка забезпечує повний цикл управління подіями інформаційної безпеки – від збору та кореляції даних до аналізу, реагування й документування інцидентів. Кожен компонент виконує власну роль: одні – забезпечують надходження достовірних даних, інші – їх обробку й аналітику, а системи реагування та аналітики загроз – реалізують оперативні дії та стратегічне вдосконалення безпеки.

В таблиці 1.2 наведено основні компоненти SOC

Таблиця 1.2 – Компоненти SOC

№	Компонент SOC	Приклади інструментів та технологій
1	Джерела даних	Windows Event Log, Syslog, NetFlow, CloudTrail, Suricata, Zeek, AWS, Azure, Google Cloud
2	Система збору та агрегування подій	Filebeat, Winlogbeat, Fluentd, Syslog-ng
3	Система керування інформацією та подіями безпеки	Wazuh, OpenSearch, ELK Stack, IBM QRadar, Splunk
4	Системи мережевого моніторингу та аналізу трафіку	Suricata, Zeek, Cisco Secure IDS, Palo Alto Networks Threat Prevention
5	Системи моніторингу кінцевих точок	Wazuh Agent, Osquery, CrowdStrike Falcon, Microsoft Defender XDR
6	Платформи аналізу загроз	MISP, OpenCTI, Anomali ThreatStream, Recorded Future
7	Системи оркестрації, автоматизації та реагування	Shuffle, TheHive, Cortex, Palo Alto Cortex XSOAR, Splunk Phantom
8	Сховище даних та система звітності	Elasticsearch, Kibana, Grafana, Tableau

У комплексі ці елементи формують основу ефективного SOC, який дозволяє своєчасно виявляти загрози, мінімізувати ризики та підтримувати стійкість інформаційної інфраструктури організації.

1.4 Сучасні тенденції у розвитку SOC

Розвиток SOC визначається зростанням складності кіберзагроз, збільшенням обсягів даних і необхідністю підвищення рівня автоматизації процесів моніторингу, IR та IM.

Залежно від масштабу та потреб організації SOC може бути реалізований у декількох формах:

- внутрішній SOC – функціонує в межах організації, забезпечуючи повний контроль над процесами безпеки;
- керований SOC – частково або повністю делегований зовнішньому провайдеру, що забезпечує моніторинг та реагування як послугу;
- гібридний SOC – поєднує внутрішні ресурси організації з можливостями зовнішніх аналітичних платформ.

Сучасні SOC еволюціонують від централізованих систем сповіщення до інтелектуальних і адаптивних платформ, здатних забезпечувати проактивний кіберзахист у режимі реального часу.

Проектування SOC супроводжується низкою викликів, серед яких великі обсяги подій безпеки, нестача кваліфікованих фахівців, складність інтеграції компонентів та висока вартість комерційних рішень. Одними з основних тенденціями розвитку SOC є впровадження хмарних та гібридних архітектур, що дозволяють поєднати переваги локальних та віддалених рішень, забезпечуючи масштабованість та безперервність моніторингу. З метою зменшення впливу людського фактору та зниження кількості хибнопозитивних сповіщень активно застосовуються технології штучного інтелекту та машинного навчання (AI/ML), зокрема поведінковий аналіз, кластеризація подій і предиктивна аналітикаи. Це дозволяє підвищити ефективність реагування та зменшити навантаження на аналітиків. Розвиток SOC також спрямований у бік автоматизації процесів за допомогою технологій SOAR, що дозволяють скоротити час між виявленням та усуненням загроз.

Використання інструментів з відкритим кодом, таких як Wazuh [27], Suricata [28], Zeek [29], OpenSearch [30], Shuffle [24], робить можливим створення повноцінного SOC навіть для організацій із обмеженим бюджетом. Для управління інцидентами використовуються платформи TheHive, Cortex, RTIR, ServiceNow, які дозволяють автоматизувати створення кейсів,

маршрутизацію завдань і співпрацю аналітиків. Інтеграція SOC із системами розвідки кіберзагроз (CTI), наприклад, MISP, підвищує якість аналізу, надаючи дані про актуальні загрози та індикатори компрометації.

Такі рішення забезпечують високу гнучкість, прозорість і можливість кастомізації під специфічні вимоги безпеки.

SOC є стратегічним компонентом інформаційної інфраструктури, який забезпечує проактивний підхід до виявлення, аналізу та усунення загроз, а також підтримує процес безперервного вдосконалення системи захисту інформації.

2. АНАЛІЗ ІНСТРУМЕНТІВ З ВІДКРИТИМ КОДОМ ДЛЯ ПОБУДОВИ SOC

2.1 Обгрунтування побудови SOC з використанням open-source рішень

Використання open-source інструментів в сфері кібербезпеки останніми роками стало ключовою тенденцією в організації SOC [21, 22, 31]. Такі рішення забезпечують високу гнучкість, прозорість та економічну ефективність, що особливо важливо для організацій із обмеженими ресурсами або тих, які прагнуть побудувати власну інфраструктуру безпеки без залежності від комерційних постачальників. Інструменти з відкритим кодом дозволяють адаптувати систему під специфічні потреби організації, інтегрувати різнорідні джерела даних та створювати масштабовані рішення на базі модульної архітектури. Крім того, активні спільноти розробників забезпечують постійне оновлення, підтримку та обмін кращими практиками, що підвищує рівень надійності та оперативності реагування на нові кіберзагрози.

Основними перевагами open-source платформ є:

- відсутність ліцензій – використання та розгортання не потребує комерційних витрат;
- прозорість коду – дозволяє здійснювати аудит, перевірку безпеки та адаптацію під власні потреби;
- гнучкість інтеграції – більшість систем підтримують API та стандарти STIX/TAXII для обміну даними;
- активна спільнота підтримки – швидке оновлення, усунення вразливостей, наявність документації;
- модульність архітектури – можливість об’єднання різних рішень у єдину SOC-екосистему.

Водночас використання інструментів з відкритим кодом має певні обмеження, які необхідно враховувати під час проєктування та впровадження SOC. Передусім такі рішення вимагають високого рівня технічної експертизи для встановлення, налаштування та подальшої підтримки компонентів

системи. На відміну від комерційних продуктів, опенсорс-рішення зазвичай мають обмежені гарантії рівня обслуговування (SLA) та відсутність офіційної технічної підтримки. Крім того, фрагментованість екосистеми зумовлює потребу у ручній інтеграції між компонентами, що може ускладнювати розгортання комплексного SOC. У базових конфігураціях багатьом таким платформам бракує повноцінних механізмів SOAR, а масштабування системи для великих корпоративних мереж часто потребує додаткової оптимізації продуктивності та ресурсів.

Таким чином, використання open-source підходу до побудови SOC є виправданим у випадках, коли важливими є гнучкість, контроль і відсутність ліцензійних витрат, а не швидке промислове впровадження.

2.2 Обгрунтування вибору інструментів для реалізації SOC

Для побудови повноцінного SOC доцільно поєднати кілька взаємодоповнюючих рішень. Для розроблення SOC у межах цієї роботи розглянуто наступні open-source платформи, що охоплюють усі етапи обробки інцидентів:

- Wazuh –ядро SIEM і моніторинговий центр [18];
- TheHive –система управління інцидентами [25];
- Cortex –платформа аналізу артефактів [26];
- MISP – база знань та засіб обмін інформацією [19].

Таке поєднання компонентів забезпечує:

- повний цикл моніторингу, аналізу, реагування та обміну інформацією;
- високу гнучкість і розширюваність за рахунок відкритого коду;
- низькі вимоги до ресурсів;
- відповідність сучасним моделям SOC (SIEM + SOAR + CTI).

Вибрані рішення мають активну спільноту підтримки, стабільні релізи та достатню кількість аналітичних модулів для навчальних і дослідницьких цілей. Така інтегрована open-source платформа дозволяє реалізувати

повноцінний SOC навіть у локальному середовищі без значних фінансових витрат.

2.2.1 Система моніторингу безпеки

Wazuh - це розподілена система моніторингу безпеки (рисунок 2.1), що поєднує функції IDS, HIDS, FIM, SIEM і системи управління відповідністю.

Рисунок 2.1 – Функції Wazuh

Архітектура Wazuh [18, 27] базується на клієнт-серверній моделі: агенти збирають події з робочих станцій, сервер здійснює кореляцію, а панель Kibana забезпечує візуалізацію (рисунок 2.2).

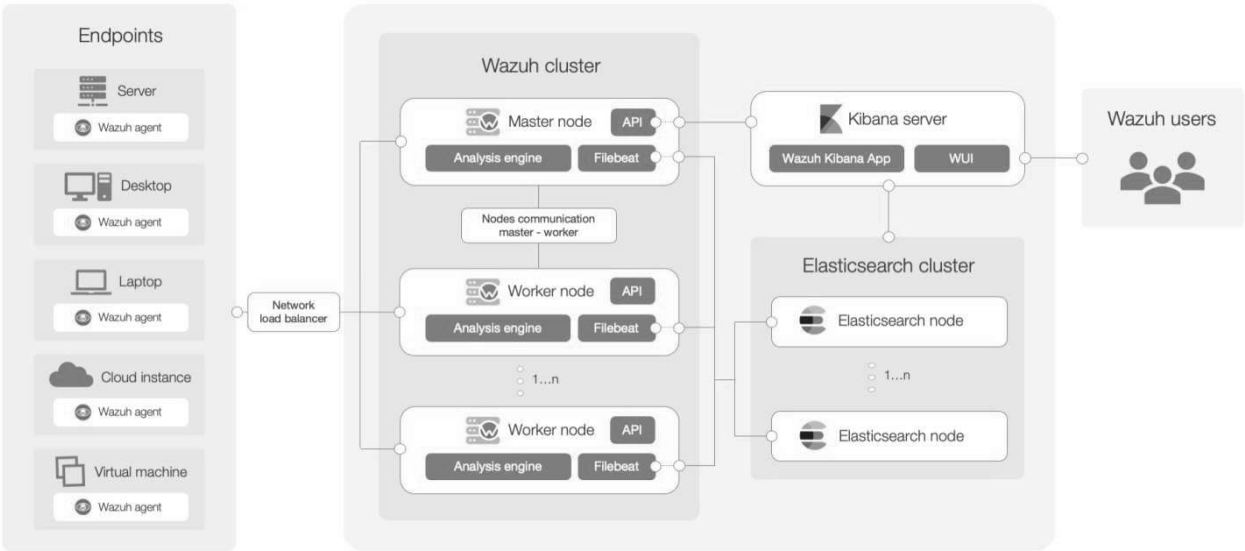

Рисунок 2.2 – Архітектура Wazuh

Основні можливості (рисунок 2.3) [18]:

- збір логів із серверів, хостів, контейнерів та хмарних сервісів;
- виявлення вторгнень і змін у системних файлах FIM;
- інтеграція з OSQuery, VirusTotal, Sysmon;
- підтримка моделі MITRE ATT&CK;
- активне IR для автоматичного блокування джерел атак.

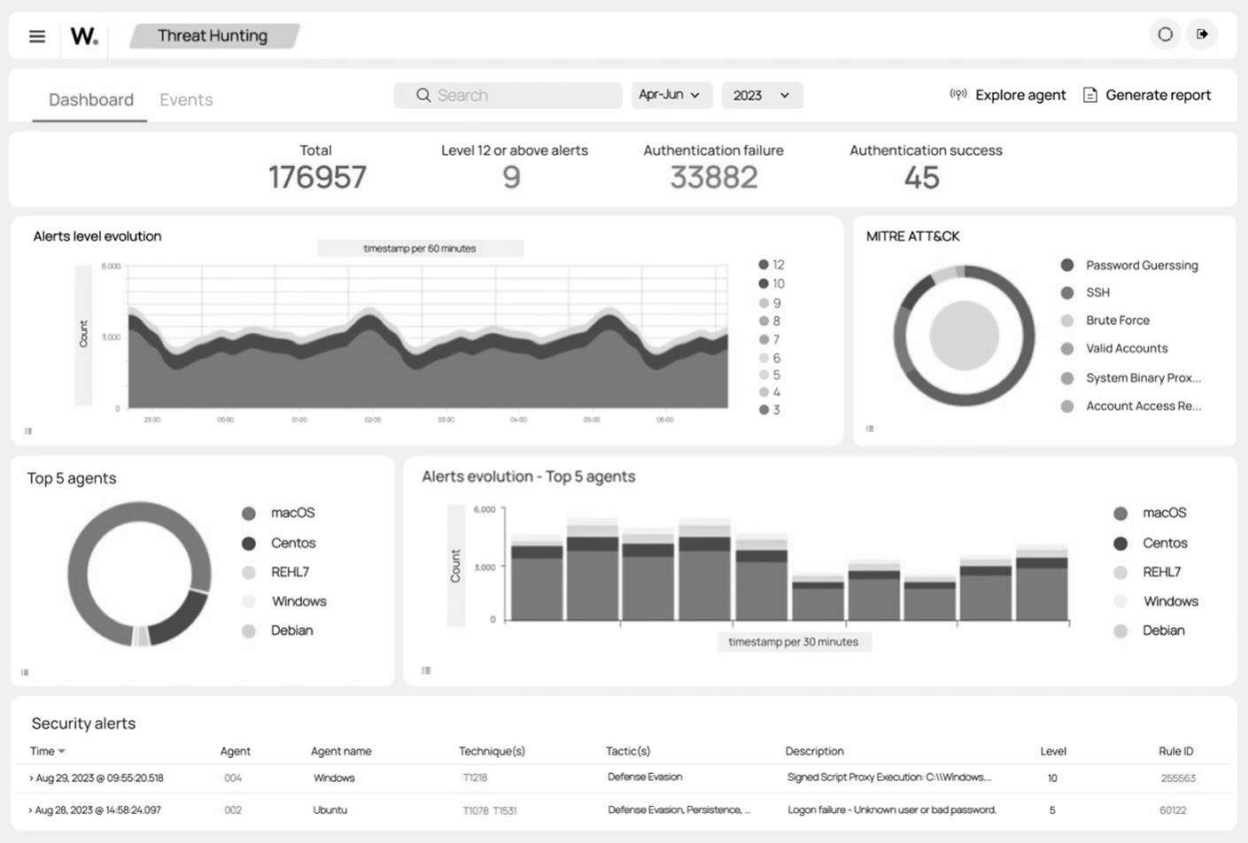

Рисунок 2.3 – Приклад візуалізації подій в інтерфейсі Wazuh

Таким чином, Wazuh є центральною складовою SOC і виконує роль ядра SIEM, де формується потік подій для подальшого аналізу аналітиками.

2.2.2 Платформа управління інцидентами

TheHive - це відкрита платформа реагування на інциденти (IR), яка забезпечує керування інцидентами, координацію команд SOC і створення кейсів для кожного виявленого інциденту [25]. Основна ідея полягає в автоматизації життєвого циклу інциденту: від отримання сповіщення до завершення розслідування (рисунок 2.4).

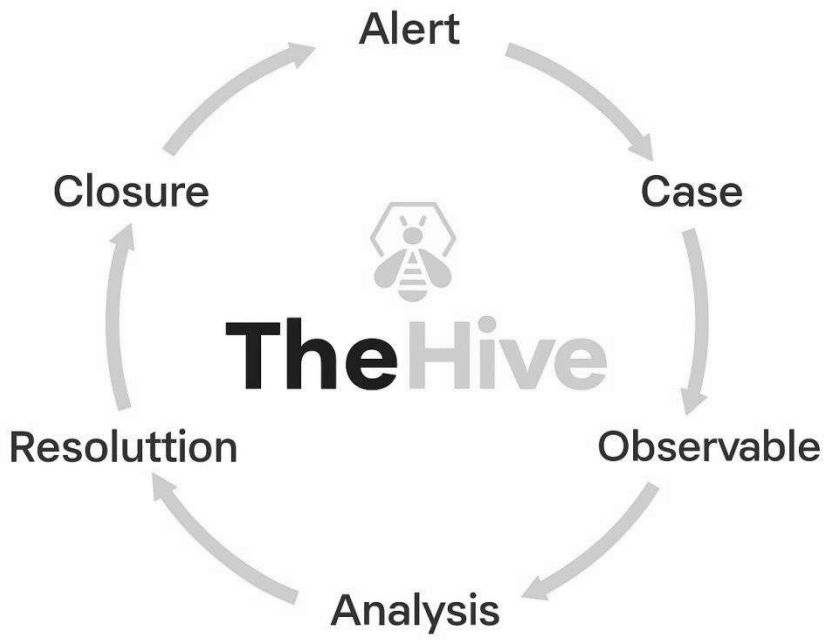

Рисунок 2.4 - Життєвий цикл інциденту в системі TheHive

Особливостями платформи є [25]:

- зручна панель керування для аналітиків (рисунок 2.5);
- шаблони кейсів, можливість розподілу ролей;
- інтеграція з Cortex, MISP, Wazuh;
- REST API для автоматизації створення кейсів із SIEM;
- підтримка багатокористувацького доступу з розмежуванням прав.

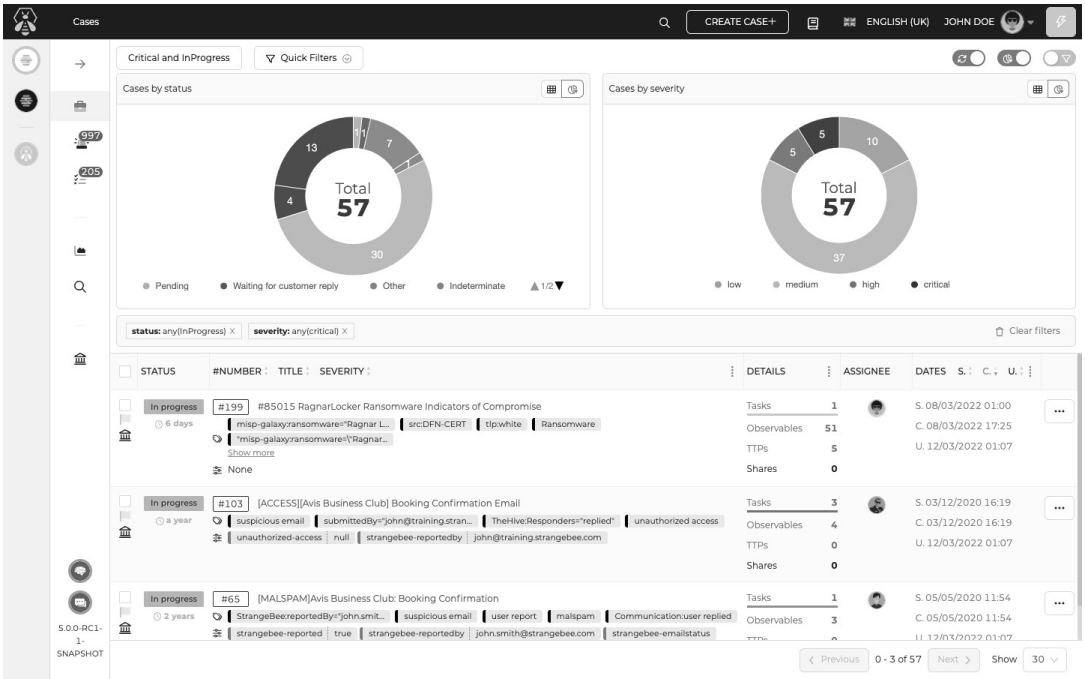

Рисунок 2.5 – Приклад графічного інтерфейсу

TheHive виконує функцію центрального вузла аналітики SOC, де зосереджено обробку та документування інцидентів.

2.2.3 Система аналізу артефактів

Cortex - це система виявлення та реагування на події безпеки, яка об’єднує інформацію з кінцевих пристроїв, мережевих джерел і хмарних сервісів, забезпечуючи автоматизований аналіз артефактів, отриманих у ході розслідувань, для підвищення точності й швидкості реагування на загрози [26]. Вона підтримує більше 100 модулів (“analyzers”) для перевірки IP-адрес, URL, файлів, доменів, хешів, тощо (рисунок 2.6).

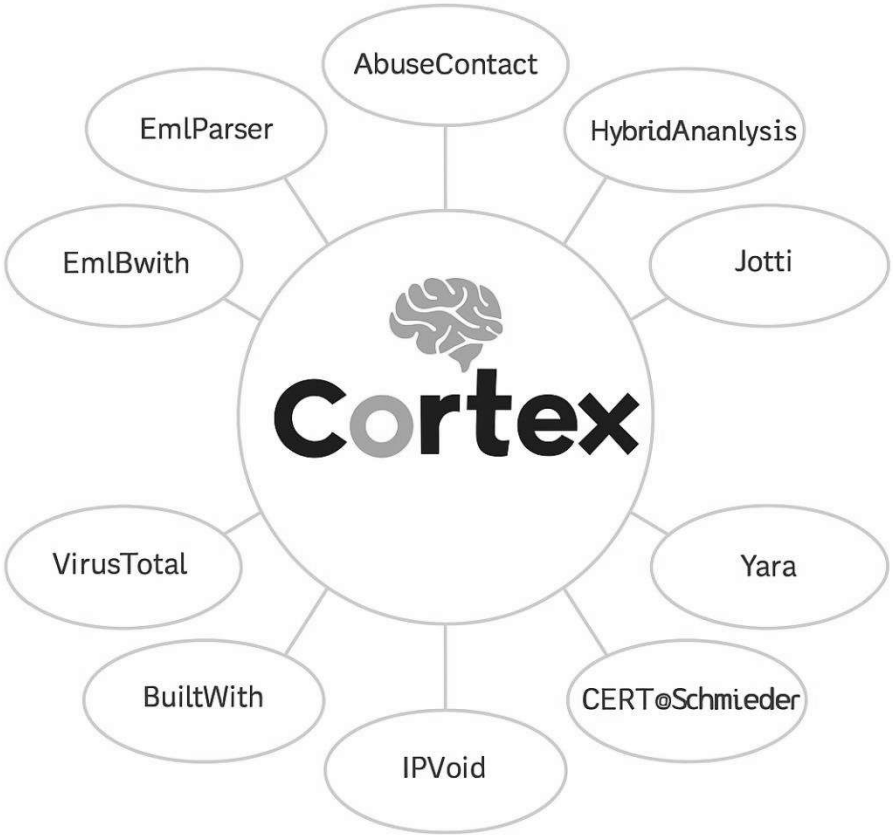

Рисунок 2.6 - Модулі автоматизованого аналізу артефактів Cortex

Основні функції [26]:

- інтеграція з TheHive для автоматичного запуску аналізу;
- виконання запитів до VirusTotal, AbuseIPDB, Shodan, Hybrid Analysis;
- підтримка обчислювальних workerів для масштабування;
- REST API для скриптової автоматизації.

На рисунку 2.7 наведено приклад інтерфейсу для розробки та зберігання аналізаторів та респондентів Cortex.

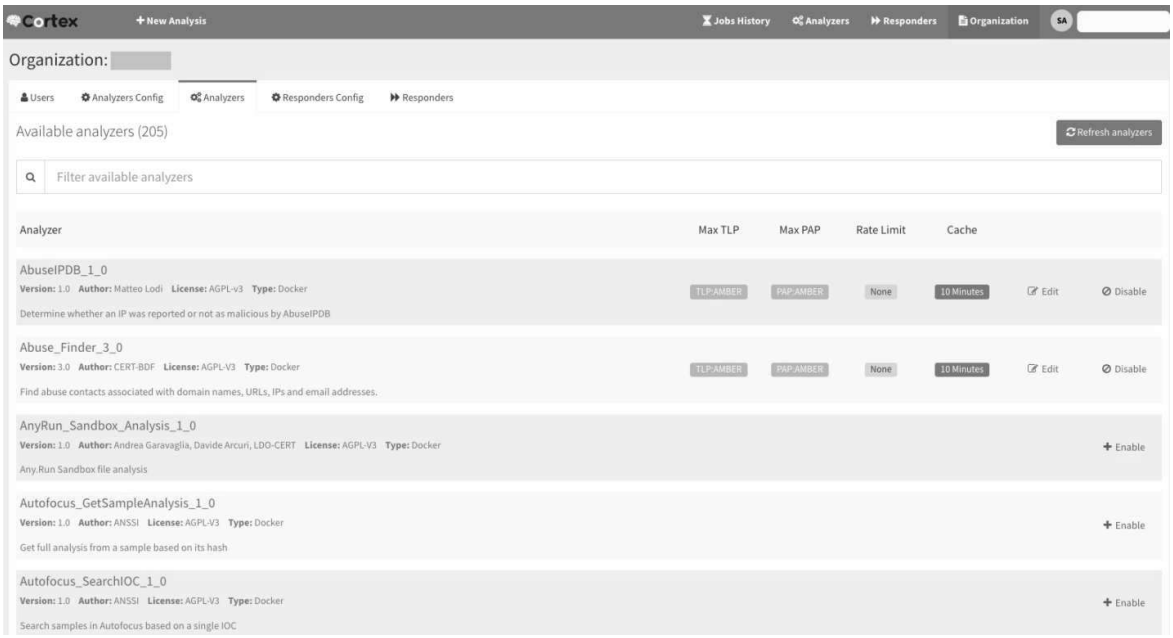

Рисунок 2.7 - Приклад графічного інтерфейсу Cortex

Cortex дозволяє значно скоротити час реагування, забезпечуючи оперативний аналіз загроз без необхідності ручного втручання аналітика.

2.2.4 Платформа обміну інформацією про загрози

MISP (Malware Information Sharing Platform) – це open-source платформа, створена для обміну даними CTI [19]. Вона забезпечує централізоване сховище індикаторів компрометації (IoC) та механізми взаємодії між організаціями (рисунок 2.8).

Рисунок 2.8 – Функції MISP

Основні можливості (рисунок 2.9) [19]:

- імпорт/експорт подій у форматах STIX, OpenIOC, CSV;
- синхронізація з іншими серверами MISP;
- візуалізація зв’язків між артефактами;
- інтеграція з TheHive, Cortex, Wazuh.

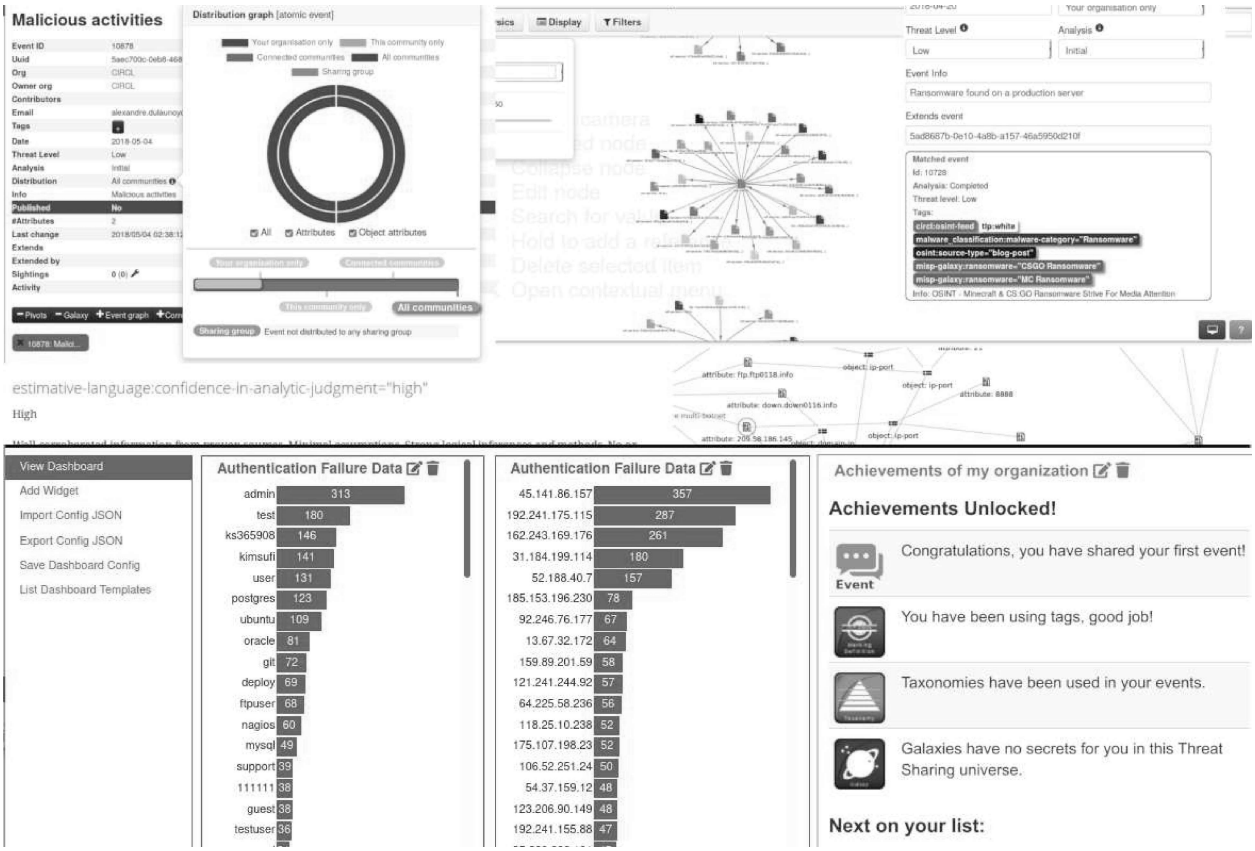

Рисунок 2.9 – Приклад відображення подій

MISP доповнює SOC, виконуючи роль платформи CTI та забезпечуючи проактивний аналіз загроз і обмін розвідувальними даними між різними організаціями, структурами або навіть державними установами. SOC фокусується на виявленні, аналізі та реагуванні на інциденти всередині організації тоді як MISP забезпечує зовнішній контекст – отримання та поширення інформації про відомі атаки, IoC, тактики, техніки, процедури. Разом вони утворюють замкнений цикл безпеки. Платформа MISP збагачує контекст виявлених подій, що дозволяє центру безпеки швидше та точніше реагувати на загрози

2.2.5 Порівняльна характеристика інструментів

Для об’єктивного порівняння розглянуті інструменти проаналізовано за ключовими критеріями.

Таблиця 2.1 – Порівняльні характеристики інструментів

Критерій	Wazuh	TheHive	Cortex	MISP
1	2	3	4	5
Основне призначення	Моніторинг подій, SIEM, виявлення вторгнень, SIEM	Обробка інцидентів, IR, IM	Автоматизований аналіз, SOAR	Обмін даними, CTI
Ефективність виявлення	Висока, завдяки кореляції подій, FIM, IDS	Середня – аналітична оцінка	Висока, завдяки автоматизації перевірок	Висока - широка база IoC і TI-фідів
Масштабованість	Висока (кластерна архітектура, розподілені агенти)	Середня (підтримка кластерів)	Висока (розподілені worker-и)	Висока (реплікація, синхронізація інстансів)
Простота налаштування	Середня (потрібна інтеграція з ELK)	Середня (налаштування кейсів і ролей)	Складна (багато інтеграцій)	Середня(налаштування фідів і синхронізації)
Рівень автоматизації	Active Response, правила кореляції	Автоматичне створення кейсів	Автоматичний аналіз	Автоматичний обмін IoC через API

Продовження таблиці 2.1

1	2	3	4	5
Інтеграція	ELK Stack, Osquery, Sysmon, VirusTotal	Cortex, MISP, Wazuh	TheHive, MISP	TheHive, Cortex, Wazuh
Тип взаємодії з SOC	Ядро SIEM і моніторингу	Оркестрація інцидентів	Аналіз і enrichment даних	Постачання CTI та IoC

Результати аналізу підтверджують, що найбільш збалансованою платформою для реалізації базового SOC є Wazuh, який виконує роль SIEM і інтегрується з іншими компонентами.

2.3 Інтеграція компонентів SOC на базі відкритого коду

Інтеграція між інструментами є ключовою умовою ефективного функціонування операційного центру безпеки. Вона забезпечує безперервний потік інформації про інциденти, автоматизовану взаємодію між підсистемами та зменшення часу реагування на загрози. У побудованому SOC взаємодія open-source компонентів реалізована за допомогою відкритих протоколів і REST API, що дозволяє досягти сумісності без використання комерційних рішень. У межах створеної системи реалізовано такі зв'язки між модулями:

- Wazuh → TheHive. Wazuh виконує роль системи моніторингу та виявлення подій. У разі спрацювання правила кореляції або детектування аномалії, Wazuh формує тривогу (alert) у форматі JSON, яка надсилається через Wazuh API або Logstash Output Plugin у TheHive для створення інциденту (case).
- TheHive → Cortex. Після створення кейсу у TheHive аналітик може ініціювати автоматичний аналіз артефактів (IP, URL, хеші файлів, e-mail тощо) через Cortex API. TheHive надсилає запит до відповідного

аналізатора, а Cortex повертає результат із класифікацією загрози або звітом про відсутність ризику.

- Cortex ↔ MISP. Cortex може використовувати дані з MISP як базу індикаторів компрометації (IoC) для перевірки отриманих артефактів. У свою чергу, результати нових аналізів можуть автоматично додаватися до бази MISP, поповнюючи сховище розвідувальних даних.
- MISP → TheHive. При надходженні нових подій у MISP, система може ініціювати створення нових кейсів у TheHive, якщо знайдені артефакти співпадають з уже виявленими у середовищі організації.

Взаємодія між компонентами реалізується через технології, що наведені у таблиці 2.2.

Таблиця 2.2 – Технології взаємодії між компонентами

Компоненти	Механізм інтеграції	Формат даних	Протокол
Wazuh → TheHive	API або Logstash Output	JSON	HTTPS (TLS)
TheHive → Cortex	REST API	JSON	HTTPS
Cortex ↔ MISP	Analyzer module / API	STIX, JSON	HTTPS
MISP → TheHive	MISP Feeds / Webhook	STIX, JSON	HTTPS

Для автентифікації між модулями використано токени доступу (Bearer Tokens), а для передачі даних - TLS 1.2+ шифрування, що гарантує цілісність та конфіденційність взаємодії.

Автоматизація процесів реагування в SOC є ключовим фактором для підвищення ефективності та швидкості усунення загроз. У межах розробленої SOC на основі відкритих інструментів реалізовано інтегрований сценарій взаємодії компонентів, який забезпечує повний цикл обробки подій безпеки. Послідовність дій у межах цього сценарію наступна:

1. На клієнтському вузлі агент Wazuh Agent фіксує підозрілу подію, наприклад спробу виконання шкідливого PowerShell-скрипта.
2. Подія надходить до Wazuh Manager, де спрацьовує кореляційне правило й формується alert.

3. Через Logstash Output Plugin ця подія автоматично надсилається у TheHive, де створюється новий інцидент (case).

4. Аналітик SOC запускає аналіз артефактів (наприклад, хешу файлу або IP-адреси) у Cortex, який взаємодіє з базами VirusTotal, AbuseIPDB та MISP.

5. Якщо результат підтверджує наявність загрози, TheHive ініціює Active Response у Wazuh, що може заблокувати IP або ізолювати хост.

6. Зібрані артефакти зберігаються у MISP як нові індикатори компрометації для подальшого аналізу

Інтеграція між Wazuh, TheHive, Cortex та MISP дозволяє створити єдине середовище моніторингу, реагування та обміну аналітикою, яке за функціональністю наближається до промислових SIEM/SOAR-рішень.

Завдяки відкритій архітектурі та підтримці REST API розглянуті інструменти формують повноцінну SOC-екосистему, у якій забезпечується:

- автоматизований обмін подіями між системами;
- централізоване управління інцидентами;
- швидкий аналіз артефактів;
- збагачення бази знань розвідувальними даними CTI.

3. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ SOC НА ОСНОВІ ВІДКРИТОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Архітектура розробленого SOC-рішення

Архітектура SOC розроблена з урахуванням принципів модульності, масштабованості та взаємної інтеграції компонентів на основі відкритого програмного забезпечення. Її метою є створення єдиного середовища, яке забезпечує повний цикл управління інцидентами інформаційної безпеки - від збору та кореляції подій до реагування й аналізу загроз.

Запропоноване рішення побудовано за трирівневою моделлю: рівень збору даних, рівень обробки та кореляції і рівень реагування та управління інцидентами (рисунок 3.1).

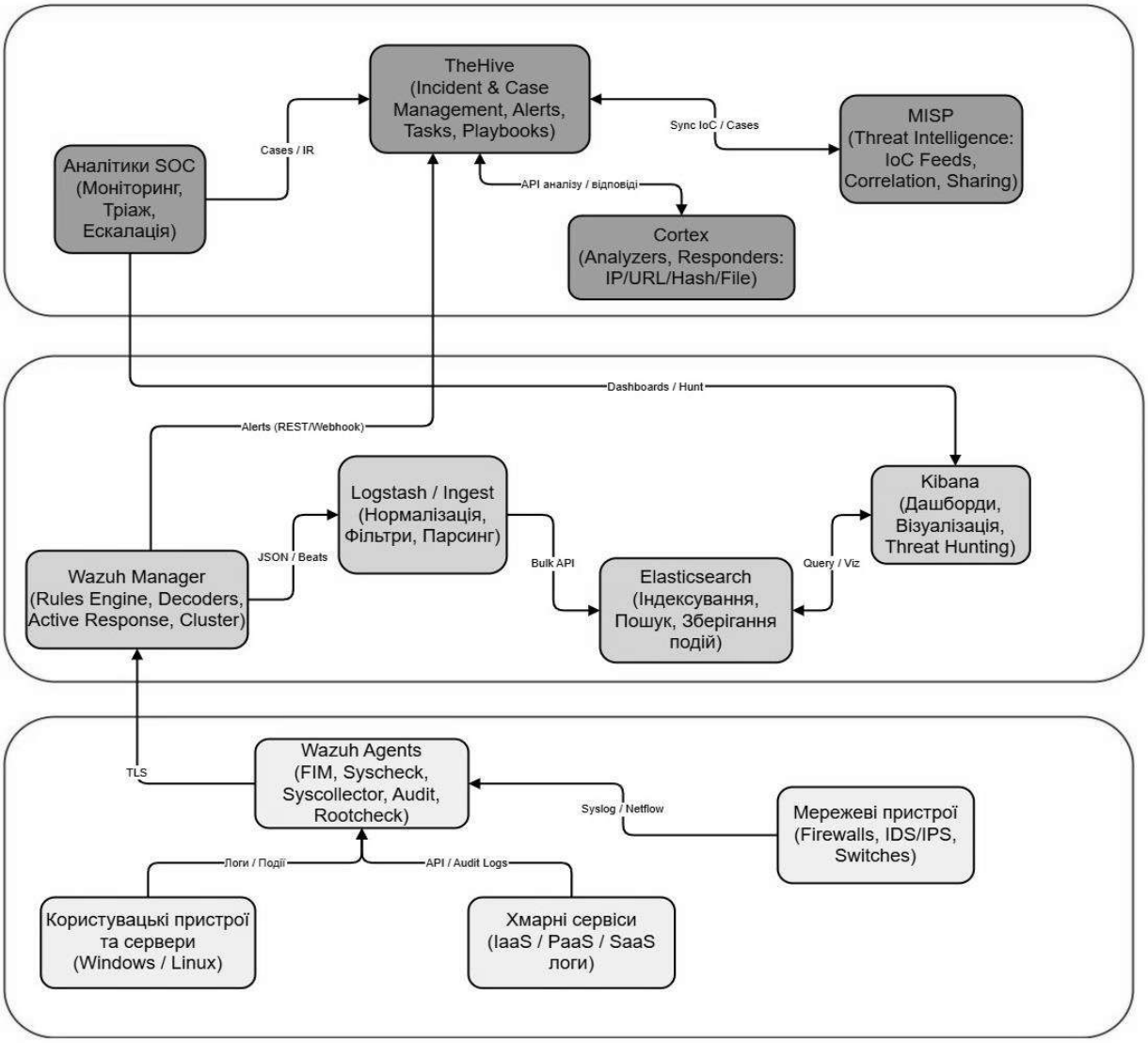

Рисунок 3.1 - Архітектура запропонованого рішення

Така структура відповідає архітектурним підходам сучасних SOC-платформ та дозволяє гнучко масштабувати систему залежно від кількості джерел даних і навантаження.

На нижньому рівні архітектури розташовані агенти системи Wazuh, встановлені на робочих станціях, серверах та мережевих пристроях організації. Їх основними завданнями є:

- збір системних журналів, безпекових подій та мережевого трафіку;
- реалізація FIM;
- виявлення аномалій поведінки користувачів або процесів;
- виконання базових дій реагування за допомогою механізму активного IR.

Зібрані дані передаються до центрального сервера Wazuh, де вони нормалізуються та зберігаються у сховищі Elasticsearch для подальшої обробки. Комунікація між агентами та сервером відбувається за захищеним каналом (TLS), що унеможливлює перехоплення або модифікацію даних під час передачі.

На рівні обробки та кореляції подій основну роль відіграє стек ELK (Elasticsearch, Logstash, Kibana), який інтегрований із Wazuh Manager.

- Logstash виконує попередню фільтрацію та нормалізацію даних, приводячи логи з різних джерел до уніфікованого формату.
- Elasticsearch виступає як високопродуктивне сховище даних та механізм пошуку, що дозволяє швидко здійснювати кореляцію подій.
- Kibana забезпечує візуалізацію подій безпеки, створення дашбордів для моніторингу стану системи, побудову статистичних графіків та звітів.

Вбудований рушій кореляції Wazuh Rules Engine дозволяє виявляти складні шаблони атак, комбінуючи події з різних джерел. Завдяки цьому можливо автоматично генерувати сповіщення про інциденти, які далі передаються до системи управління інцидентами.

На верхньому рівні функціонують компоненти, що реалізують логіку

оперативного реагування та аналітики.

– TheHive використовується як система керування інцидентами, де автоматично створюються кейси на основі тривог з Wazuh або аналітиками. Кожен кейс містить артефакти, етапи розслідування та журнали дій.

– Cortex інтегрований із TheHive для автоматизованого аналізу артефактів (IP-адрес, URL, хешів файлів, доменів тощо). Він може запускати аналіз за допомогою численних модулів (analyzers) і генерувати оцінку ризику.

– MISP (Malware Information Sharing Platform) служить платформою для обміну розвідувальними даними про загрози. Через API MISP надає інформацію про відомі IoC, що дозволяє SOC автоматично порівнювати внутрішні події з глобальними базами загроз.

Взаємодія між цими компонентами здійснюється через REST API, що забезпечує їх інтеграцію у єдину систему. У разі підтвердження інциденту аналітик SOC може ініціювати реакцію безпосередньо з інтерфейсу TheHive - наприклад, ізоляцію хоста, блокування IP-адреси або повідомлення адміністратора мережі.

Схему побудови SOC можна подати у вигляді логічної структури, шо наведена на рисунку 3.2.

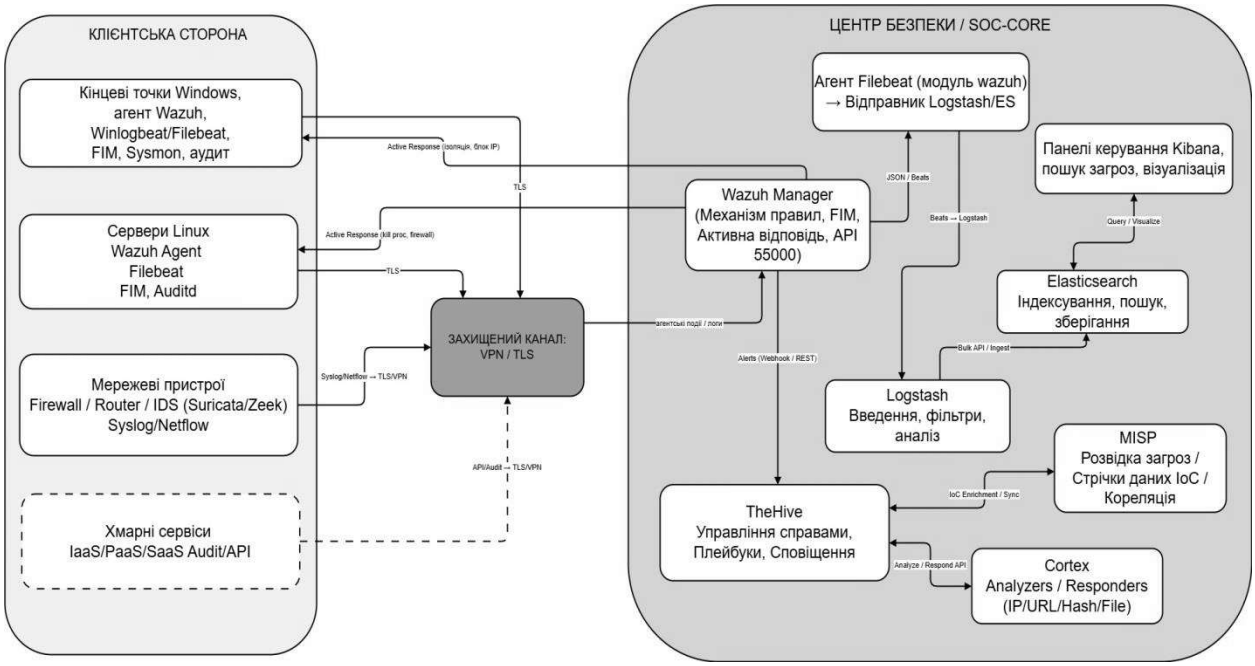

Рисунок 3.1 – Схема побудови SOC

У цій схемі Wazuh виконує функції збору та первинної обробки даних, ELK забезпечує зберігання та аналітику подій, а TheHive, Cortex і MISP - управління інцидентами та взаємодію з розвідувальними базами. Така архітектура дозволяє створити повноцінний SOC без використання комерційних SIEM або SOAR-платформ, зберігаючи при цьому високий рівень функціональності та гнучкість розгортання.

Архітектура розробленого SOC-рішення є комплексною системою, побудованою на відкритих технологіях, що дозволяє здійснювати моніторинг, кореляцію та реагування на інциденти в єдиному інформаційному середовищі. Обрані компоненти забезпечують взаємодію між рівнями збору даних, аналітики та управління інцидентами, формуючи централізовану платформу для забезпечення кібербезпеки організації.

3.2 Налаштування середовища для розгортання SOC

Налаштування середовища для розгортання SOC є важливим етапом практичної реалізації, оскільки від нього залежить стабільність, продуктивність і можливість інтеграції нових модулів у майбутньому. Для побудови SOC на основі відкритого програмного забезпечення було використано багаторівневу віртуалізовану інфраструктуру, яка поєднує серверну частину, агентське середовище та мережевий сегмент для тестування інцидентів.

Кінцеві пристрої, наведені на рисунку 3.2, не всі були реалізовані в даному проекті, мережеві пристрої не охоплені у зв’язку з обмеженими ресурсами. Середовище розгорнуто у вигляді ізольованого віртуального кластера, створеного за допомогою VirtualBox і VMware Workstation Pro, що забезпечує контроль над мережевими параметрами та зручність відновлення систем. У конфігурації використано чотири віртуальні машини:

- SOC-Server - головний сервер, для реалізації моніторингу та аналітики, на якому інстальовано Wazuh Manager, Elasticsearch Logstash і Kibana;

– IR-Server - окремий вузол для реагування та зборк розвідданих, з встановленими TheHive, Cortex та MISP;

– Client-1 - робоча станція з OS Windows, де розгорнуто агент Wazuh, Winlogbeat.

– Client-2 - робоча станція з OS Linux, для тестування Linux-журналів і перевірки FIM на якій розрогнутий Wazuh Agent та Filebeat.

В таблиці 3.1 наведено перелік компонентів для розгортання SOC.

Таблиця 3.1 – Апаратні та програмні компонентти SOC

Вузол	Вимоги	ОС	Відкриті порти
SOC-Server	2 vCPU 8 GB RAM	Ubuntu 20.04 LTS	TCP 1514, 1515, 55000 (Wazuh) TCP 5044 (Logstash) TCP 5601 (Kibana) TCP 9200 (Elasticsearch) TCP 22 (SSH)
IR-Server	2 vCPU 8 GB RAM	Ubuntu 20.04 LTS	TCP 9000 (TheHive) TCP 9001 (Cortex) TCP 443 (MISP, HTTPS) TCP 22 (SSH)
Client-1 Windows Endpoint	2 vCPU 4 GB RAM	Windows 10 Pro	TCP 3389 (RDP) TCP 1514–1515 (TLS до SOC)
Client-2 Linux Endpoint	2 vCPU 4 GB RAM	Ubuntu 20.04 LTS	TCP 22 (SSH) TCP 1514–1515 (TLS до SOC)

Як базову платформу використано Ubuntu Server 20.04 LTS, що поєднує стабільність і підтримку довготривалих оновлень без надмірного споживання ресурсів. Усі вузли об’єднано в єдину віртуальну мережу типу Host-Only Adapter із підмережею 192.168.56.0/24. Центральний сервер має статичну адресу 192.168.56.10, інші вузли отримують IP-адреси динамічно через DHCP. Така топологія дозволяє повністю контролювати маршрутизацію, не впливаючи на основну систему.

З огляду на обмежені ресурси робочої станції було обрано оптимізоване середовище з використанням перевірених стабільних версій програмного забезпечення (таблиця 3.2), сумісних між собою та придатних для локального розгортання.

Таблиця 3.2 – Версії та призначення компонентів ПЗ

Компонент	Версія	Призначення
Wazuh Manager	4.4.5	Система збору, кореляції та аналізу подій безпеки
Elasticsearch	8.8.2	Зберігання та індексація журналів подій
Logstash	8.8.2	Обробка, фільтрація й нормалізація логів
Kibana	8.8.2	Візуалізація даних і створення аналітичних панелей
Filebeat	8.8.2	Передавання логів із Wazuh до Elasticsearch
TheHive	5.0.18	Керування інцидентами та розслідуваннями
Cortex	3.1.4	Автоматизований аналіз артефактів
MISP	2.4.175	TIP та обміну IoC
Docker Engine	23.0.6	Контейнеризація сервісів TheHive, Cortex, MISP
OpenJDK	11	Виконує Java-додатки TheHive і Cortex

Вибір саме цих версій зумовлений їх сумісністю, невисокими вимогами до оперативної пам’яті (2-3 ГБ на сервер SOC і до 1 ГБ на IR-сервер), а також наявністю докладної офіційної документації для розгортання.

На сервері SOC було інстальовано Wazuh Manager (рисунок 3.3).

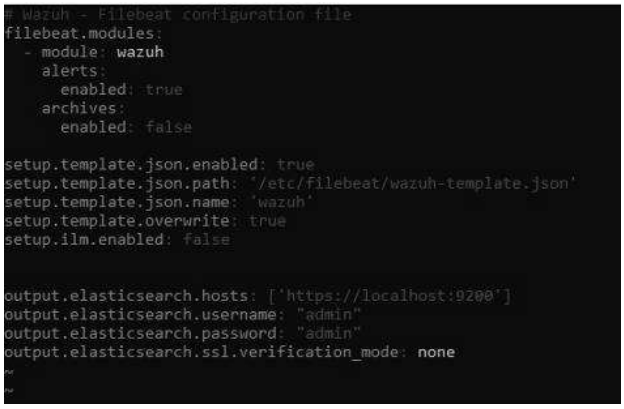

Рисунок 3.3 – Інсталяція Wazuh Manager

На клієнтських вузлах встановлено Wazuh Agent (рисунок 3.4), який

зареєстровано на сервері SOC:

```
/var/ossec/bin/manage_agents
```

Комунікація між агентами та сервером здійснюється через TLS, що гарантує цілісність і конфіденційність даних.

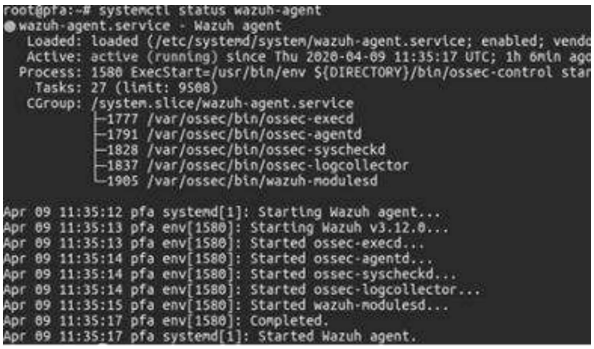

Рисунок 3.4 – Перевірка роботи агента Wazuh

Після інсталяції виконано налаштування таких параметрів:

– у файлі `/var/ossec/etc/ossec.conf` вказано локальну адресу менеджера та шляхи до журналів;

– у `/etc/filebeat/filebeat.yml` активовано модуль wazuh для передачі подій у Logstash;

– у `/etc/elasticsearch/jvm.options` обмежено пам’ять Java-машини до 2 ГБ (`-Xms2g`, `-Xmx2g`);

– портова конфігурація: 55000 (API), 1514 (агентські з’єднання), 5601 (Kibana).

Встановлені компоненти інтегровані із ELK (рисунок 3.5).

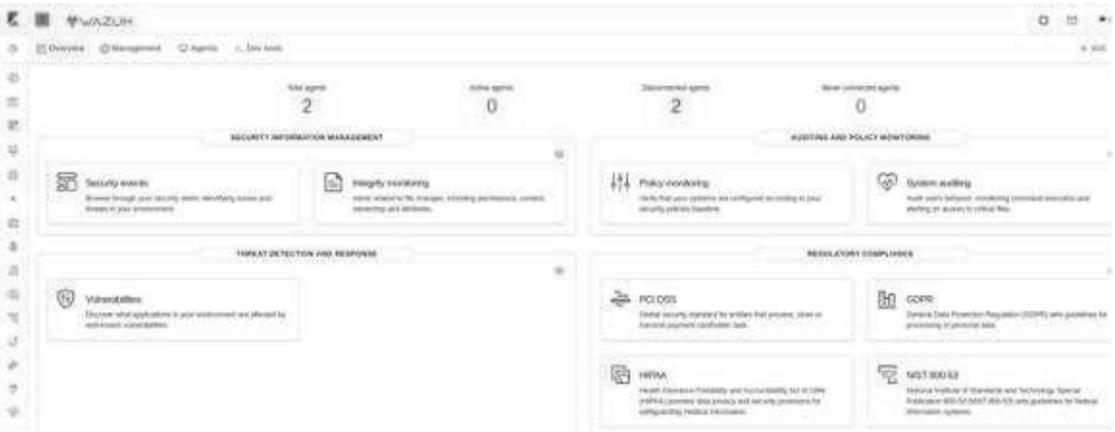

Рисунок 3.5 – Приклад інтерфейсу Kibana

Компоненти реагування та аналітики розгорнуто на окремому IR-сервері. Для спрощення конфігурації використано Docker Compose, який дозволив швидко підняти три контейнери:

```
thehive:5.0.18, cortex:3.1.4, misp:2.4.175
```

У конфігураційному файлі `/opt/thehive/application.conf` визначено інтеграцію з Cortex і MISP через REST API, а також створено облікові токени для аналітиків.У MISP підключено базу MariaDB 10.6 та ввімкнено HTTPS-доступ для захищеного обміну даними.

Для забезпечення базового рівня безпеки виконано такі дії:

- налаштовано `ufw`, дозволено тільки порти служб SOC;
- створено окремі ролі користувачів у TheHive: admin, analyst, responder;
- застосовано автентифікацію за токенами в Cortex і MISP;
- виконано налаштування резервного копіювання індексів Elasticsearch (`snapshot.repo`);
- реалізовано контроль цілісності файлів за допомогою Wazuh FIM.

На рисунку 3.6 наведено приклад основної панелі керування TheHive та Cortex.

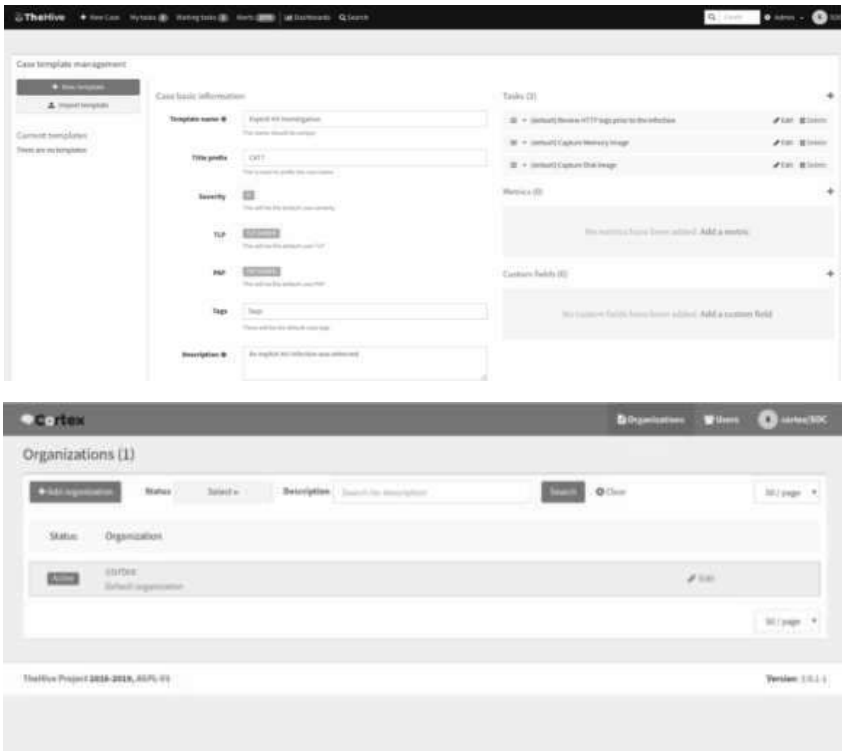

Рисунок 3.6 - Інформаційні панелі

Встановлено MISP та інтегровано з TheHive. Після завершення за адресою `IP` отримано доступ до панелі автентифікації (рисунок 3.7).

Рисунок 3.7 – Панель автентифікації MISP

Проведено відповідні налаштування, створено користувача, сервер MISP та інтегровано з іншими інструментами, зокрема Cortex та TheHive. Проведено налаштування API, журналювання та створено набір правил. На рисунку 3.8 наведено приклад перевірки попереджень TheHive.

Рисунок 3.8 – Відображення попереджень

На рисунку 3.9 наведено приклад вікна відображення хостів, який демонструє події, що генеруються в кінцевих точках.

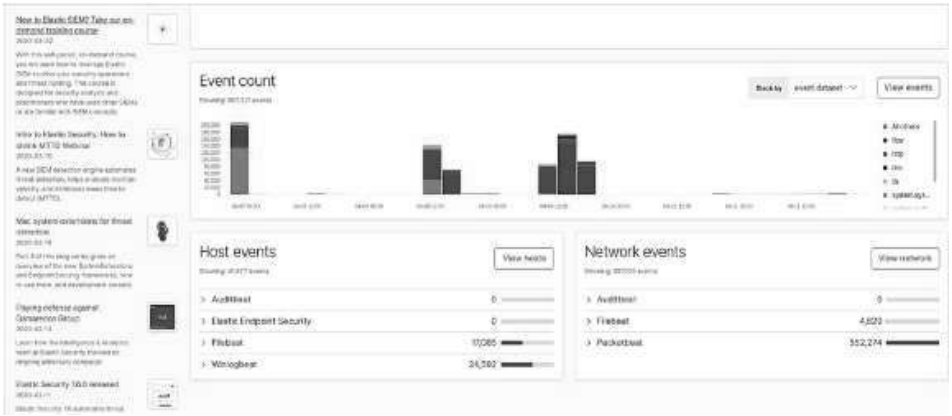

Рисунок 3.9 – Розділ Host events ELK

Після завершення розгортання всі модулі функціонують як SOC-комплекс. Події, зібрані агентами надходять до Wazuh Manager, обробляються через Logstash, зберігаються в Elasticsearch і відображаються у Kibana. Попередження автоматично передаються до TheHive, де створюються інциденти, що можуть бути передані до Cortex для аналізу або звірені з базою MISP.

На рисунку 3.10 наведено приклад відображення мережевих подій.

Рисунок 3.10 – Візуалізація мережевих подій

На рисунку 3.11 наведено приклад інтерфейсу панелі моніторингу.

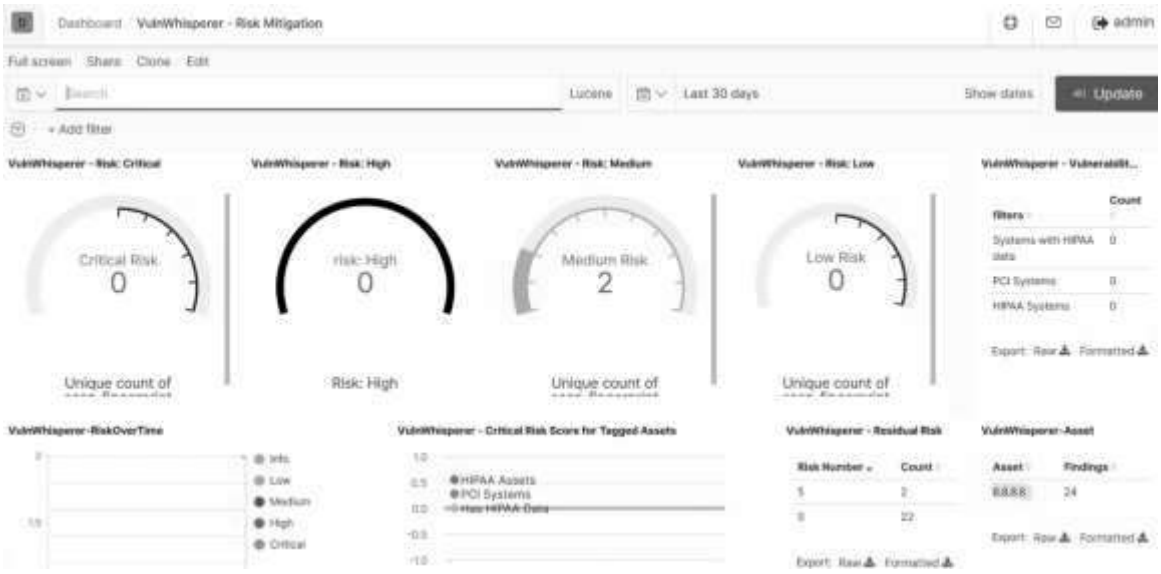

Рисунок 3.11 – Панель моніторингу Kibana

Таким чином, побудоване середовище забезпечує повний цикл моніторингу, кореляції, аналізу й реагування на інциденти безпеки у межах локальної інфраструктури.

3.3 Тестування роботи SOC

Мета тестування створеного SOC полягає у перевірці коректності роботи його компонентів, достовірності збору подій, ефективності виявлення інцидентів та автоматичного реагування на них. Експеримент виконувався у локальному середовищі з використанням агентів Wazuh, системи аналітики Kibana, платформ TheHive і Cortex та проведенням контрольованих інцидентів різних типів.

Для перевірки функціональності SOC застосовано сценарний підхід. На клієнтських вузлах ініціювалися події кіберінцидентів, що охоплюють основні типи атак та внутрішніх порушень безпеки, зокрема:

- зміни у системних файлах;
- спроба несанкціонованого доступу до SSH-сервісу;
- виконання потенційно небезпечних PowerShell-команд.

Кожна подія фіксувалася агентом Wazuh і відображалася у Kibana, після чого генерувалося попередження, яке передавалося до TheHive для подальшого аналізу.

Схему взаємодії компонентів SOC наведено на рисунку 3.12.

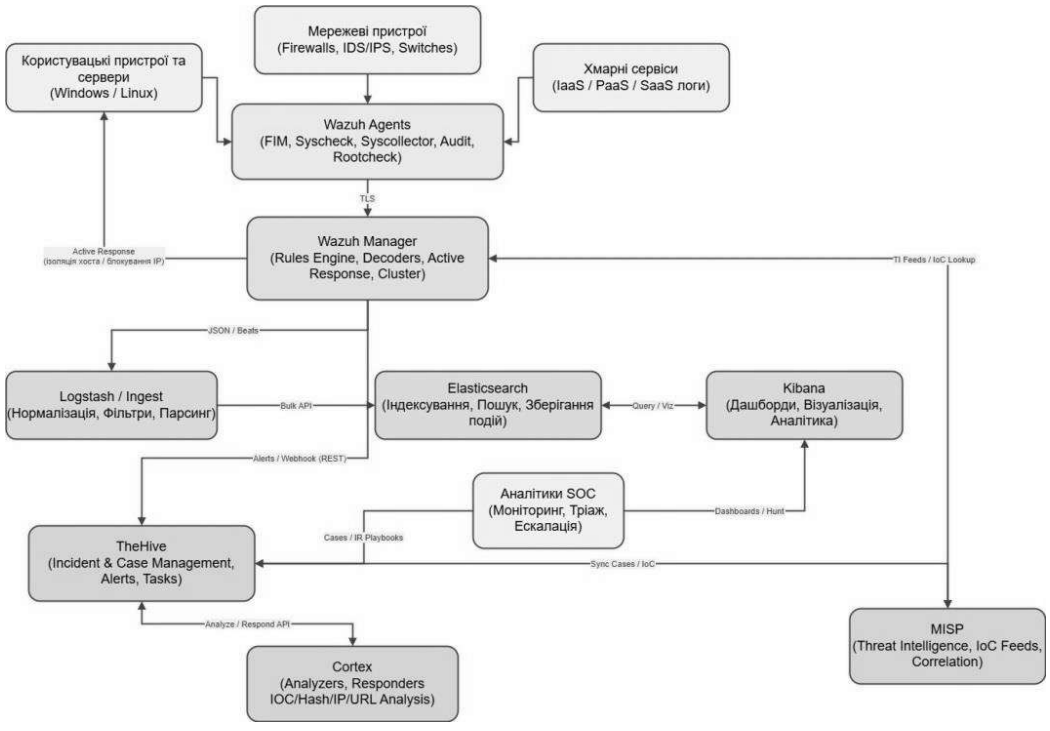

Рисунок 3.12 - Схема обробки подій під час тестування SOC

Процес тестування складався з таких етапів:

- генерація подій безпеки на агентських вузлах;
- збір та передача логів у Wazuh Manager через Filebeat до Logstash;
- кореляція та обробка подій у Elasticsearch з подальшою візуалізацією у Kibana;
- передача попереджень до TheHive через Wazuh Webhook;
- автоматизований аналіз артефактів у Cortex та звірення з базою IoC у MISP.

Перший тестовий сценарій включав перевірку роботу модуля File Integrity Monitoring (FIM) у Wazuh. На агенті Linux було створено та змінено системний файл `/etc/passwd` командою:

```
sudo echo "#test_change" >> /etc/passwd
```

У відповідь агент передав сповіщення менеджеру, яке відобразилося у Kibana.Фрагмент журналу:

```
{
    "rule": { "id": "550", "description": "Integrity checksum changed." },
    "agent": { "name": "linux-endpoint01" },
    "file": { "path": "/etc/passwd", "md5_after": "b2e1d...", "md5_before": "afc3e..." },
    "level": 7
}
```

На аналітичній панелі Kibana сформовано графік активності подій FIM за останню добу, де видно пікове значення під час тесту (рисунок 3.13).

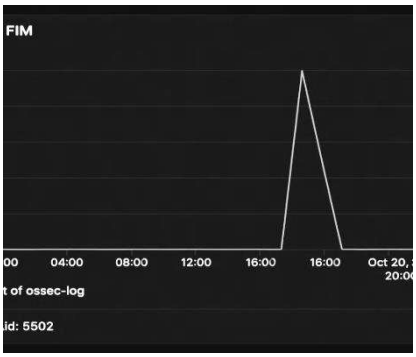

Рисунок 3.13 - Моніторинг змін системних файлів FIM у Kibana

Інцидент автоматично передано до TheHive, де створено кейс “ із тегом `config-modification`. Через Cortex було виконано аналіз артефакту (хешу файлу) за допомогою VirusTotalAnalyzer. В результаті відсутності збігу із

відомими IoC у MISP - невідомий файл було класифіковано як «Low Severity».

Другий сценарій забезпечував перевірку здатность SOC виявляти багаторазові спроби входу по SSH. Для цього на SOC-Server через інструмент hydra було виконано:

```
hydra -l root -P /usr/share/wordlists/rockyou.txt ssh://192.168.56.10
```

У Kibana з'явились події типу `sshd[xxx]: Failed password`, які були корельовані Wazuh-правилом 5710. Згенероване попередження мало рівень критичності 10 та опис:

Multiple failed SSH login attempts detected from 192.168.56.103 (possible brute-force attack).

Гістограма у Kibana показала кількість невдалих спроб, згенерованих атакою брутфорсу за часом (рисунок 3.14).

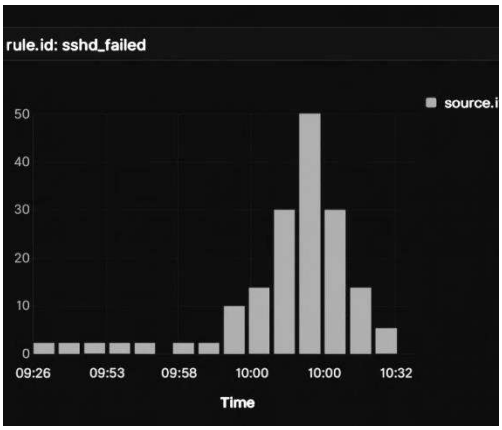

Рисунок 3.14 – Кількість невдалих SSH-входів

У TheHive автоматично створено інцидент типу «Intrusion attempt» із критичністю high, який автоматично передано до Cortex. Використано аналізатор GeoIP для визначення джерела підозрілої активності - локальна IP-адреса тестового вузла. Система виконала Active Response, додавши правило блокування у iptables, що зафіксовано в журналі:

```
ossec-agent: action: firewall-drop, srcip=192.168.56.103
```

Для перевірки моніторингу Windows-подій виконано тестовий сценарій, що включавнаступний PowerShell-скрипт:

```
Invoke-WebRequest -Uri http://malicious.test/file.ps1 -OutFile C:\Temp\file.ps1
```

Winlogbeat передав журнал події (ID 4104 - PowerShell ScriptBlock Logging) у Wazuh, де правило 61606 визначило її як потенційно шкідливу. У Kibana відображено сповіщення (рисунок 3.15).

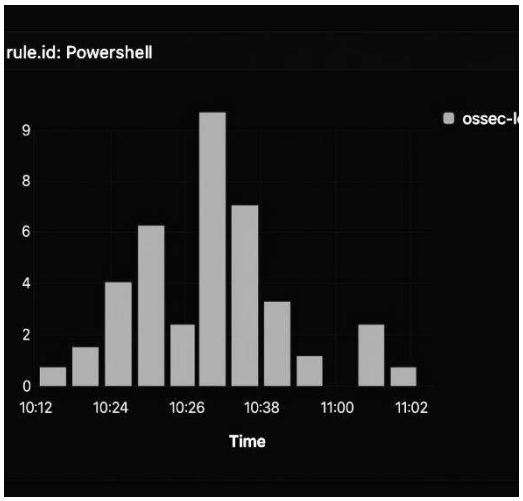

Рисунок 3.15 - Виявлення шкідливої PowerShell-активності у Kibana

Інцидент автоматично створено у TheHive та надіслано до Cortex для аналізу URL через URLHausAnalyzer. Отримано результат - домен у чорному списку, інцидент позначено як High Severity.

За підсумками проведених експерементів підтверджено коректну роботу всіх компонентів SOC:

- події з агентів успішно збираються, передаються до Wazuh Manager і індексуються в Elasticsearch;
- Kibana відображає дані у реальному часі у вигляді графіків, гістограм і сповіщень;
- TheHive автоматично створює інциденти на основі тригерів Wazuh;
- Cortex та MISP забезпечують подальший аналіз і збагачення даних.
- Active Response ефективно блокують джерела атак на рівні агента.

Результати тестування узагальнено в таблиці 3.3.

Таблиця 3.3 - Результати функціональних тестів SOC

Тип інциденту	Джерело подій	Рівень критичності	Виявлення	Реагування
Insider / MIS-configuration	Linux Agent	Medium	FIM → Wazuh → Kibana → TheHive	Аналіз хешу у Cortex (VirusTotal)
Network Attack	Linux Server	High	Wazuh rule 5710 → Kibana	Active Response (iptables block)
Malware / Suspicious Behavior	Windows Agent	High	Wazuh rule 61606 → Kibana	Cortex URLHausAnalyzer + TheHive case

В таблиці 3.4 наведено результати тестування та узагальнені оцінки ефективності SOC, що свідчать про ефективність виявлення та реагування навіть у локальному середовищі, зокрема середній час виявлення складає MTTD=3,13 секунди, середній час реагування MTTR=5,1 секунди.

Таблиця 3.4 - Результати тестування та оцінка ефективності SOC

Тестовий сценарій	Реакція	Час виявлення (с)	Час реагування (с)
FIM – System File Change	Створено кейс у TheHive	3,1	5,0
SSH Brute-force	Автоматичне блокування IP	2,5	4,2
PowerShell Malicious Script	Сповіщення та аналіз IoC	3,8	6,1

На рисунку 3.16 наведено порівняння часу виявлення та реагування для тестових сценаріїв.

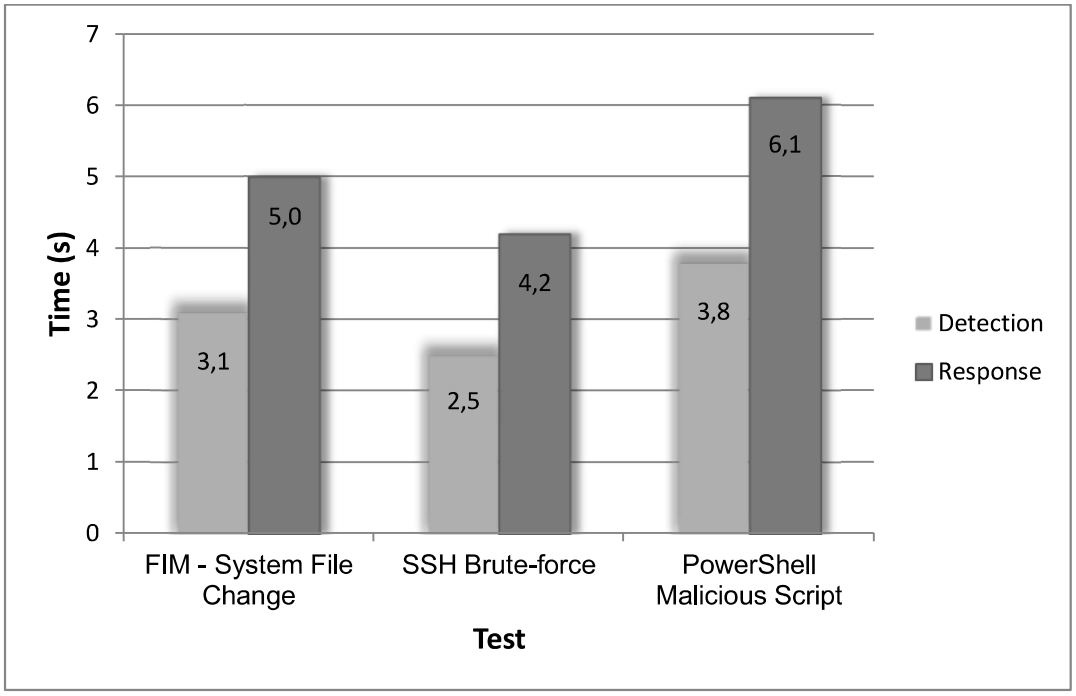

Рисунок 3.16 – Порівняння швидкості виявлення та реагування на загрози

Проведене тестування підтвердило стабільну роботу всіх компонентів побудованого SOC-рішення. Виявлені події відповідали реальним сценаріям загроз, що дозволяє оцінити SOC як працездатний прототип для моніторингу інфраструктури середнього рівня складності. Середнє навантаження на систему під час тестів не перевищувало 2,5 ГБ оперативної пам’яті, що свідчить про ефективність використання ресурсів і можливість масштабування без суттєвих апаратних витрат. Система забезпечує повний цикл управління інцидентами - від збору подій і кореляції до створення інцидентів, їх розслідування та автоматизованого реагування.

ВИСНОВКИ

Досліджено основні завдання, функції та принципи роботи SOC, що дозволило визначити його роль як ключового елементу SIEM, який забезпечує моніторинг, аналіз та реагування на інциденти у реальному часі.

Досліджено основні архітектурні моделі SOC - монолітну, модульну, внутрішню, керовану та гібридну, що дало змогу обґрунтувати вибір модульної архітектури на основі відкритого ПЗ, яка забезпечує гнучкість, масштабованість і можливість адаптації до різних організаційних потреб.

Проаналізовано основні компоненти SOC відповідно до трирівневої функціональної структури: збір даних, аналітика та реагування. Такий підхід дозволив систематизувати елементи SOC та узгодити їх ролі в повному циклі обробки інцидентів.

Обгрунтовано вибір компонентів проектованого SOC, які реалізують усі етапи життєвого циклу інциденту, від виявлення до реагування й обміну даними. Це дозволо створити узгоджену внутрішню екосистему безпеки.

Розроблено модель взаємодії компонентів SOC, яка забезпечує автоматизований обмін подіями, артефактами та результатами аналізу, що дозволяє створити єдину систему з узгодженим інформаційним потоком.

Спроєктовано архітектуру SOC із трирівневою структурою: збір даних, обробка подій і реагування, яка забезпечує логічну взаємодію між компонентами та підтримує масштабованість системи.

Реалізовано тестове середовище для розгортання SOC, проведено налаштування компонентів та перевірено працездатність проєктованої системи. Результати тестування підтвердили ефективність автоматизованого обміну даними між компонентами SOC і здатність системи до швидкого реагування на виявлені загрози.

Розроблена SOC об’єднує процеси моніторингу, аналізу, реагування й обміну даними, що дозволяє підвищити рівень кіберстійкості організаційної інфраструктури при мінімальних витратах.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. What is an Information Security Management System (ISMS)?. [Електронний ресурс].- Режим доступу: https://www.dataguard.com/blog/what-is-information-security-management-system/

2. Information Security Management System SaaS For ISO 27001. [Електронний ресурс].- Режим доступу: https://www.isms.online/information-security-management-system-isms/

3. ISMS: Information Security Management Systems Explained. [Електронний ресурс].- Режим доступу: https://www.splunk.com/en_us/blog/learn/isms-information-security-management-systems.html

4. Що таке SOC та навіщо він потрібен. [Електронний ресурс].- Режим доступу: https://gigatrans.ua/ua/news/chto-takoe-soc-i-zachem-on-nuzhen

5. Security Operations Center (SOC). [Електронний ресурс].- Режим доступу: https://www.wallarm.com/what/security-operations-center-soc

6. What Is A Security Operations Center? [Електронний ресурс].- Режим доступу: https://purplesec.us/learn/security-operations-center-soc/

7. Security Operation Center (SOC). [Електронний ресурс].- Режим доступу: https://blogs.halodoc.io/security-operation-center-soc/

8. Building an Intelligent Security Operations Center. [Електронний ресурс].- Режим доступу: https://www.balbix.com/insights/introduction-to-security-operations-center/

9. Overview of Security Operations Center Technologies. [Електронний ресурс].- Режим доступу: https://www.ciscopress.com/articles/article.asp?p=2455014&seqNum=7

10. Впровадження SIEM і SOC. [Електронний ресурс].- Режим доступу: https://www.h-x.technology/ua/siem-soc-implementation-ua

11. NIST SP 800-61r3. Incident Response Recommendations and Considerations for Cybersecurity Risk Management: A CSF 2.0 Community Profile / A. Nelson, S. Rekhi, M. Souppaya, K. Scarfone. – Gaithersburg, MD:

National Institute of Standards and Technology, 2025. – 63 p. – DOI: 10.6028/NIST.SP.800-61r3.

12. What Is MTTD? The Mean Time to Detect Metric, Explained. [Електронний ресурс].- Режим доступу: https://www.splunk.com/en_us/blog/learn/mean-time-to-detect-mttd.html

13. MTTD and MTTR in Cybersecurity Incident Response. [Електронний ресурс].- Режим доступу: https://www.wiz.io/academy/mttd-and-mttr

14. MTBF, MTTR, MTTA, and MTTF. [Електронний ресурс].- Режим доступу: https://www.atlassian.com/incident-management/kpis/common-metrics

15. Overview of Security Operations Center Technologies. [Електронний ресурс].- Режим доступу: https://www.ciscopress.com/articles/article.asp?p=2455014

16. Карташ І.В., Орєхов О.А. Модель архітектури операційного центру безпеки / Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Системи та технології кібернетичної безпеки». [Електронний ресурс].- Режим доступу: https://ela.kpi.ua/server/api/core/bitstreams/0fa36631-9745-44d2-83b3-46674fd86b56/content

17. The Role of the Security Operations Center (SOC) in Cloud Security. [Електронний ресурс].- Режим доступу: https://www.sysdig.com/learn-cloud-native/the-role-of-the-security-operations-center-soc-in-cloud-security

18. Wazuh. [Електронний ресурс].- Режим доступу: https://wazuh.com/

19. MISP. [Електронний ресурс].- Режим доступу: https://www.misp-project.org/

20. Alerting for Elasticsearch and Kibana. [Електронний ресурс].- Режим доступу: https://search-guard.com/alerting/?gad_source=1&gad_campaignid=21025831367&gbraid=0AAAADK5CRIy7Pq_wcZSQsAwoo4W2B4M0&gclid=Cj0KCQiA5abIBhCaARIsAM3-zFXBYroP_OgWo7aOD8daAVH0JlnwKEmPhjgFxBrsFQALpSbZ71ZxSsgaArbXEALw_wcB

21. Top 9 Open Source SIEM Tools for 2025. [Електронний ресурс].- Режим доступу: https://www.sentinelone.com/cybersecurity-101/data-and-

ai/open-source-siem-tools/

22. Top 12 Network Security Monitoring Tools for 2025. [Електронний ресурс].- Режим доступу: https://clouddle.com/blog/network-security-monitoring-tools/

23. Автоматизація реагування на інциденти (рішення SOAR) . [Електронний ресурс].- Режим доступу: https://www.seeton.pro/cybersecurity/soar/

24. Shuffle. [Електронний ресурс].- Режим доступу: https://shuffler.io/

25. TheHive Project - SOC Level 1 -Digital Forensics and Incident Response - TryHackMe Walkthrough & Insights. [Електронний ресурс].- Режим доступу: https://iritt.medium.com/thehive-project-soc-level-1-digital-forensics-and-incident-response-tryhackme-walkthrough-713c89aa5c6f

26. Cortex. [Електронний ресурс].- Режим доступу: https://strangebee.com/cortex/

27. Wazuh: Complete Guide to HIDS, SIEM, and Enterprise Threat Detection. [Електронний ресурс].- Режим доступу: https://medium.com/@sadashivpole/wazuh-complete-guide-to-hids-siem-and-enterprise-threat-detection-120d376bfbc0

28. Suricata. [Електронний ресурс].- Режим доступу: https://suricata.io/

29. Zeek. [Електронний ресурс].- Режим доступу: https://zeek.org/

30. OpenSearch. [Електронний ресурс].- Режим доступу: https://opensearch.org/

31. Open-Source SOC tools. [Електронний ресурс].- Режим доступу: https://www.wiz.io/academy/open-source-soc-tools