ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «SIEM-СИСТЕМИ ДЛЯ ВИЯВЛЕННЯ МЕРЕЖЕВИХ ЗАГРОЗ»

на здобуття освітнього ступеня бакалавра
зі спеціальності 123 Комп'ютерна інженерія
(код, найменування спеціальності)
освітньо-професійної програми «Комп'ютерна інженерія»
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ *Даніл ЛИТВИНЕНКО*
(підпис) *Ім'я, ПРІЗВИЩЕ здобувача*

Виконав: здобувач(ка) вищої освіти гр. КІД-42
Даніл ЛИТВИНЕНКО
Ім'я, ПРІЗВИЩЕ

Керівник: Наталія ЛАЩЕВСЬКА
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

Рецензент: _____
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Комп'ютерної інженерії

Ступінь вищої освіти бакалавр

Спеціальність 123 Комп'ютерна інженерія

Освітньо-професійна програма «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ
Завідувач кафедри
Комп'ютерної інженерії
_____ Наталія ЛАЩЕВСЬКА
«_____» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Литвиненко Даніл Андрійович
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: «SIEM-системи для виявлення мережевих загроз»
керівник кваліфікаційної роботи: Наталія ЛАЩЕВСЬКА, кандидат технічних наук, доцент,
(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «24» лютого 2025р. №56
2. Строк подання кваліфікаційної роботи «05» червня 2025р.
3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, стандарти інформаційної безпеки, архітектура та компоненти SIEM-систем, формати журналів подій, методи виявлення загроз, налаштування та тестування Wazuh, порівняльний аналіз SIEM-рішень.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
    1. Теоретичні основи SIEM-систем
    2. Розгортання SIEM-системи з використанням існуючих інструментів
    3. Оцінка ефективності та перспективи розвитку
5. Перелік ілюстративного матеріалу: *презентація*
    1. Мета, об'єкт, предмет дослідження
    2. Актуальність роботи
    3. Принцип роботи SIEM-систем
    4. Архітектура SIEM-системи для виявлення мережевих загроз
    5. Етапи розгортання SIEM-системи через Docker
    6. Розгортання агентів Wazuh на вузлах
    7. Демонстрація збору даних агентами зі вузлів
    8. Формування звітів на основі зібраних даних
    9. Перевірка спрацювання алертів на події
    10. Покращення якості виявлення та автоматизація

11. Кошторис реалізації та порівняння з готовими рішеннями
12. Висновки
13. Публікації та апробація роботи

6. Дата видачі завдання «24» лютого 2025р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Збір даних	25.02-19.03.2025	
2	Обробка матеріалу	20.03-02.04.2025	
3	Розробка теоретичної частини	03.04-07.04.2025	
5	Розгортання SIEM-системи з використанням існуючих інструментів	08.04-27.04.2025	
6	Оцінка ефективності та перспективи розвитку	28.04-11.05.2025	
7	Висновки за результатами аналізу	12.05-15.05.2025	
8	Розробка презентації	16.05-26.05.2025	
9	Передзахист	27.05-29.05.2025	
10	Здача в деканат	30.05-05.06.2025	

Здобувач(ка) вищої освіти _____ Даніл ЛИТВИНЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Наталія ЛАЩЕВСЬКА
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 101 стор., 67 рис., 6 табл., 32 джерел.

Мета роботи – оптимізація процесу виявлення мережевих загроз за рахунок використання принципів кореляційного аналізу та автоматизації моніторингу в SIEM-системі Wazuh.

Об'єкт дослідження – процеси виявлення, аналізу та обробки мережевих подій безпеки.

Предмет дослідження – технології SIEM-моніторингу, засоби автоматизованого збору та обробки логів, методи кореляції подій та реакції на загрози.

Короткий зміст роботи: У роботі досліджено принципи побудови SIEM-систем для виявлення мережевих загроз на прикладі відкритого рішення Wazuh у поєднанні з ELK-стеком. Розгорнуто систему в середовищі Docker, налаштовано збір і обробку логів, реалізовано правила кореляції подій та систему сповіщень. Проведено аналіз типових атак (brute-force, port scanning) і перевірено ефективність детекції.

Оцінено продуктивність системи, визначено її обмеження та запропоновано шляхи вдосконалення: підключення threat intelligence-джерел, інтеграція з MITRE ATT&CK, автоматизація реакції на загрози та побудова дашбордів. Проведено порівняння з іншими SIEM-рішеннями (Splunk, OSSIM) та обґрунтовано доцільність використання Wazuh у невеликих і середніх організаціях.

КЛЮЧОВІ СЛОВА: ІНФОРМАЦІЙНА БЕЗПЕКА, SIEM, МЕРЕЖЕВА БЕЗПЕКА, WAZUH, DOCKER, ЖУРНАЛИ ПОДІЙ, ВИЯВЛЕННЯ ЗАГРОЗ, КІБЕРАТАКИ, KIBANA, ELK, КОРЕЛЯЦІЯ ПОДІЙ, АВТОМАТИЗАЦІЯ, THREAT INTELLIGENCE, MITRE ATT&CK.

ABSTRACT

Text part of the qualification work for obtaining a bachelor's degree: 101 pages, 67 figures, 6 tables, 32 sources.

The purpose of the work is to optimize the process of detecting network threats by using the principles of correlation analysis and monitoring automation in the Wazuh SIEM system.

The object of the study is the processes of detecting, analyzing and processing network security events.

The subject of the study is SIEM monitoring technologies, automated log collection and processing tools, event correlation methods and threat response.

Summary of the work: The work examines the principles of building SIEM systems for detecting network threats using the example of the open Wazuh solution in combination with the ELK stack. The system is deployed in a Docker environment, log collection and processing are configured, event correlation rules and a notification system are implemented. Typical attacks (brute-force, port scanning) are analyzed and the detection efficiency is checked.

The system performance is assessed, its limitations are identified and ways of improvement are proposed: connecting threat intelligence sources, integration with MITRE ATT&CK, automating threat response and building dashboards. Comparison with other SIEM solutions (Splunk, OSSIM) is made and the feasibility of using Wazuh in small and medium-sized organizations is justified.

KEYWORDS: INFORMATION SECURITY, SIEM, NETWORK SECURITY, WAZUH, DOCKER, EVENT LOGS, THREAT DETECTION, CYBERATTACKS, KIBANA, ELK, EVENT CORRELATION, AUTOMATION, THREAT INTELLIGENCE, MITRE ATT&CK.

ЗМІСТ

ВСТУП..... 9
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ SIEM-СИСТЕМ..... 10
1.1 Основи інформаційної безпеки в мережах 10
1.2 Принципи роботи SIEM-систем 16
1.3 Основні класи SIEM-рішень на ринку 22
1.4 Стандарти та вимоги до SIEM-систем 32
1.5 Інструменти виявлення загроз у SIEM..... 38
РОЗДІЛ 2. РОЗГОРТАННЯ SIEM-СИСТЕМИ З ВИКОРИСТАННЯМ ІСНУЮЧИХ ІНСТРУМЕНТІВ. 46
2.1 Архітектура обраної SIEM-системи..... 46
2.2 Побудова SIEM-системи на базі Wazuh та ELK 52
2.3 Алгоритми обробки подій та виявлення загроз 68
2.4 Тестування та аналіз роботи побудованої системи 75
РОЗДІЛ 3. ОЦІНКА ЕФЕКТИВНОСТІ ТА ПЕРСПЕКТИВИ РОЗВИТКУ 84
3.1 Методи підвищення ефективності роботи SIEM. 84
3.2 Порівняння Wazuh з іншими SIEM-системами..... 93
3.3 Напрями вдосконалення побудованої системи..... 101
ВИСНОВКИ..... 109
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ……...……110
Додаток А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)…….....113

ВСТУП

В умовах стрімкої цифровізації суспільства та зростання залежності бізнесу від IT-інфраструктури питання інформаційної безпеки виходить на перший план. Щоденне функціонування організацій супроводжується обробкою значних обсягів даних, які циркулюють у складних і розгалужених мережах. Така активність створює численні точки уразливості, які можуть бути використані зловмисниками, тому виникає потреба у системах, що забезпечують своєчасне виявлення та нейтралізацію загроз.

Одним із найефективніших підходів до забезпечення кібербезпеки є застосування SIEM-систем (Security Information and Event Management). Ці системи дозволяють централізовано збирати, обробляти та аналізувати події безпеки з різних джерел у режимі реального часу. Поєднання можливостей логування, кореляції подій, автоматичних оповіщень і зручної візуалізації дає змогу не лише оперативно виявляти загрози, а й виконувати аудит відповідності до міжнародних стандартів. Існуючі SIEM-рішення поділяються на комерційні та open-source, кожен тип має свої сильні сторони та обмеження.

Серед відкритих рішень особливу увагу привертає Wazuh — масштабована SIEM-платформа, що інтегрується зі стеком ELK (Elasticsearch, Logstash, Kibana). Завдяки підтримці rule-based і поведінкового аналізу, сумісності з MITRE ATT&CK, можливості автоматичної реакції на інциденти (Active Response) та високій гнучкості в налаштуванні, Wazuh є привабливим варіантом для впровадження у невеликих і середніх організаціях.

Водночас, щоб така система ефективно працювала в конкретному середовищі, необхідно грамотно побудувати її архітектуру, налаштувати джерела логів, адаптувати правила виявлення атак і перевірити все в умовах, наближених до реальних, що зумовлює актуальність розгортання власного SIEM-рішення на базі Wazuh, що дозволяє оцінити його ефективність, адаптивність і практичну придатність для виявлення мережевих загроз.

РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ SIEM-СИСТЕМ

1.1 Основи інформаційної безпеки в мережах

Зі стрімким розвитком інформаційних технологій і глобальним поширенням мереж значно зростає актуальність питань, що пов’язані із захистом інформаційних ресурсів, особливо коли йдеться про захист мережевої інфраструктури як основи комунікації між користувачами, пристроями та системами. Сучасна цифрова реальність вимагає глибокого розуміння основ інформаційної безпеки та ключових загроз, які постають перед інфраструктурою, що забезпечує безперервну обробку, зберігання і передавання даних. Саме в такому контексті особливої ваги набуває питання моніторингу, оскільки безперервне спостереження за поведінкою мережевих систем дозволяє своєчасно виявляти підозрілі дії та мінімізувати можливі ризики. Ключову роль у цьому процесі відіграє логування, яке створює основу для подальшого аналізу, побудови кореляцій та ідентифікації інцидентів безпеки з використанням сучасних аналітичних підходів [1-4].

Визначення основних загроз інформаційній інфраструктурі. Інформаційна інфраструктура охоплює широке коло компонентів — від апаратного забезпечення до каналів зв’язку, програмного забезпечення та процедур, що супроводжують обробку даних, тому будь-яка вразливість в одному із цих елементів може стати точкою входу для зловмисника, наслідком чого стає порушення цілісності, доступності або конфіденційності інформації. У контексті сучасної кіберзагрози до основних типів належать несанкціонований доступ, коли сторонні особи намагаються отримати доступ до ресурсів без відповідних дозволів, що може призвести до крадіжки або зміни даних, а також широке використання шкідливого програмного забезпечення, зокрема вірусів, троянів, програм-шифрувальників та інших засобів, які вражають системи зсередини. Подібні загрози добре знайомі

навіть звичайним користувачам, не кажучи вже про організації, які щоденно стикаються з атаками, спрямованими на порушення їхньої роботи.

Окрему групу становлять мережеві атаки, зокрема DDoS, що паралізують роботу сервісів шляхом перевантаження системи численними запитами, через що легітимні користувачі не можуть отримати доступ до ресурсів. До ще одного небезпечного типу загроз належать соціально орієнтовані атаки, наприклад фішинг, що ґрунтуються на психологічних маніпуляціях і експлуатації довіри користувачів, часто з використанням підроблених посилань або повідомлень, після чого зловмисники отримують контроль над обліковими записами або персональними даними, використовуючи їх для подальших шахрайських дій. Не менш небезпечними є й внутрішні загрози, коли порушення безпеки походить від співробітників або інших осіб із легітимним доступом, що можуть як свідомо зловживати своїми правами, так і ненавмисно спричинити критичні інциденти через необережність або незнання. Враховуючи численні потенційні вектори атак, виникає нагальна потреба у постійному автоматизованому моніторингу, який дозволяє в реальному часі аналізувати поведінку систем і своєчасно реагувати на загрози (див. рис. 1.1) [1].



Рисунок 1.1 – Класифікація основних кіберзагроз для інформаційної інфраструктури

На тлі цифрової трансформації та зростання складності мережевої взаємодії звичайних засобів захисту вже недостатньо, тому безперервний моніторинг у поєднанні з аналітичними інструментами, що враховують як технічні, так і поведінкові аспекти діяльності системи, є критично необхідним. Важливо не лише збирати дані про події, а й мати здатність оперативно їх:

— аналізувати;
— виявляти аномалії;
— корелювати інформацію з різних джерел;
— будувати профілі загроз;
— адаптувати політики безпеки відповідно до змін у середовищі.

Тільки за умови поєднання сучасних підходів до моніторингу, автоматизації реагування та безперервного вдосконалення засобів виявлення можна досягти ефективного функціонування інформаційної інфраструктури в умовах постійних загроз [1].

Роль моніторингу в забезпеченні мережевої безпеки. Моніторинг у сфері кібербезпеки — це безперервний процес спостереження за станом інформаційних систем, мереж, подій і всього, що в них відбувається з метою виявлення підозрілих або аномальних дій. У контексті комп'ютерної інженерії моніторинг охоплює критично важливі функції, пов'язані з безпекою інфраструктури, зокрема виявлення вторгнень у реальному часі за допомогою систем на кшталт IDS, IPS або SIEM. Такі рішення дозволяють оперативно реагувати на відхилення в поведінці мережі чи системи, зокрема:

— спроби несанкціонованого доступу;
— зміну конфігурацій;
— неочікувану активність користувачів чи додатків.

Важливою функцією є також аналіз інцидентів, що передбачає збирання та збереження логів подій для подальшого розслідування після виникнення загрози.

Моніторинг у цьому контексті охоплює:

— відстеження дій користувачів;
— контроль доступу;

— змін до критичних налаштувань систем;

— автоматичне формування звітів та надсилання оповіщень.

Усе це дозволяє адміністраторам швидко реагувати на інциденти. Особливе значення має централізований підхід, при якому інформація з усіх джерел, включно з мережевими пристроями, серверами та кінцевими точками, агрегується в єдину систему, де аналізується в реальному часі. Дане завдання виконує система управління інформацією та подіями безпеки (SIEM), яка наразі є стандартом для середовищ з високими вимогами до безпеки та критичними даними, які неможливо довірити стороннім постачальникам без контролю (див. рис. 1.2) [2].

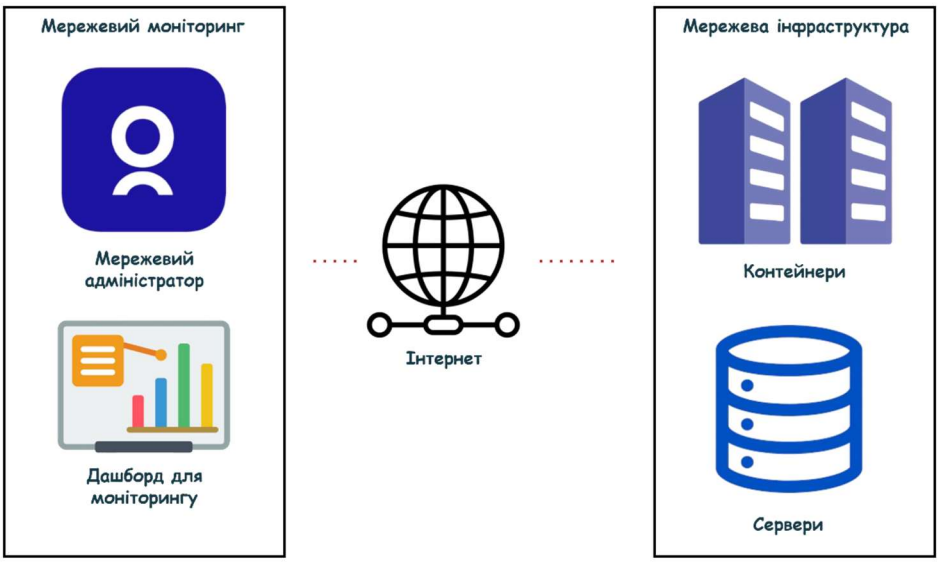
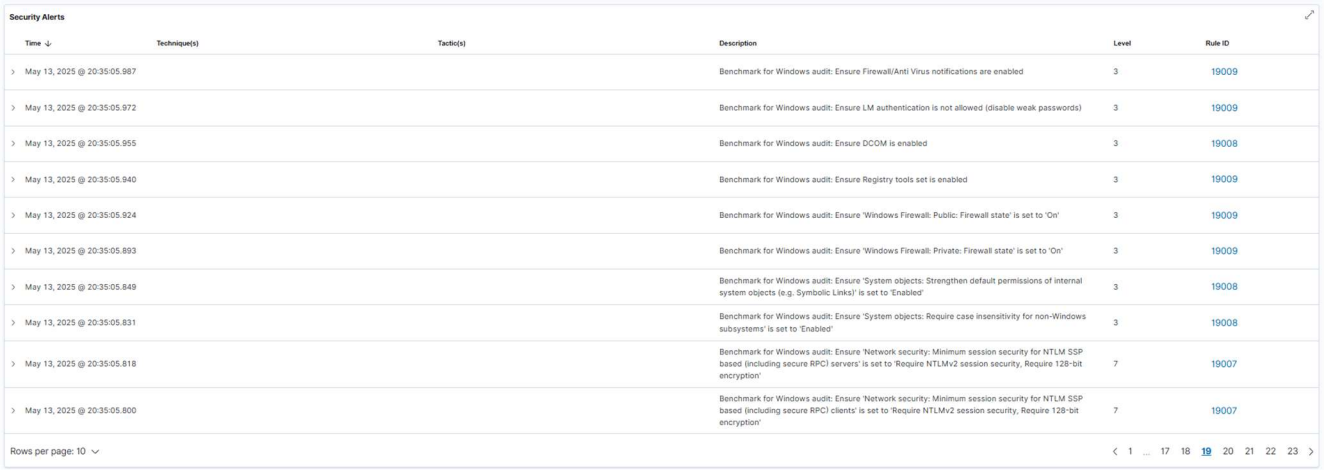


Рисунок 1.2 – Схема моніторингу мережевої активності

Поняття журналів подій, логів і їх значення в аналізі загроз. Логи або журнали подій є основним джерелом даних для процесів безпекового моніторингу, оскільки вони містять структуровану інформацію про всі події в межах системи — починаючи зі спроб входу й запитів до файлів, закінчуючи помилками, підключеннями до мережі та спробами зміни прав доступу (див. рис. 1.3).



Security Alerts

Time	Technique(s)	Tactic(s)	Description	Level	Rule ID
May 13, 2025 @ 20:35:05.987			Benchmark for Windows audit: Ensure Firewall/Anti Virus notifications are enabled	3	19009
May 13, 2025 @ 20:35:05.972			Benchmark for Windows audit: Ensure LM authentication is not allowed (disable weak passwords)	3	19009
May 13, 2025 @ 20:35:05.955			Benchmark for Windows audit: Ensure DCOM is enabled	3	19008
May 13, 2025 @ 20:35:05.940			Benchmark for Windows audit: Ensure Registry tools set is enabled	3	19009
May 13, 2025 @ 20:35:05.924			Benchmark for Windows audit: Ensure 'Windows Firewall: Public: Firewall state' is set to 'On'	3	19009
May 13, 2025 @ 20:35:05.893			Benchmark for Windows audit: Ensure 'Windows Firewall: Private: Firewall state' is set to 'On'	3	19009
May 13, 2025 @ 20:35:05.849			Benchmark for Windows audit: Ensure 'System objects: Strengthen default permissions of internal system objects (e.g. Symbolic Links)' is set to 'Enabled'	3	19008
May 13, 2025 @ 20:35:05.831			Benchmark for Windows audit: Ensure 'System objects: Require case insensitivity for non-Windows subsystems' is set to 'Enabled'	3	19008
May 13, 2025 @ 20:35:05.818			Benchmark for Windows audit: Ensure 'Network security: Minimum session security for NTLM SSP based (including secure RPC) servers' is set to 'Require NTLMv2 session security, Require 128-bit encryption'	7	19007
May 13, 2025 @ 20:35:05.800			Benchmark for Windows audit: Ensure 'Network security: Minimum session security for NTLM SSP based (including secure RPC) clients' is set to 'Require NTLMv2 session security, Require 128-bit encryption'	7	19007

Rows per page: 10 ‹ 1 … 17 18 19 20 21 22 23 ›

Рисунок 1.3 – Структура та приклад журналу подій

Вони поділяються на кілька типів, серед яких:

— системні журнали (відображають роботу операційної системи);

— мережеві логи (фіксуються дані про з’єднання, трафік і правила фаєрволів);

— журнали додатків (містять інформацію про дії користувачів та внутрішню логіку програмного забезпечення);

— журнали безпеки (фіксують спроби автентифікації, зміну ролей або виявлення шкідливих дій).

Значення логів полягає у можливості точної фіксації фактів, що дозволяє відтворити хронологію подій; виявлення аномалій на основі повторюваних або нетипових дій; підтвердження фактів інцидентів, які використовуються як докази; та побудові моделей поведінки користувачів на основі історичних даних. Саме логи є тим джерелом, на якому базуються автоматизовані системи моніторингу на кшталт SIEM, здатні в режимі реального часу обробляти великі обсяги інформації з метою виявлення загроз (див. рис. 1.4) [3].



Рисунок 1.4 – Потік даних у процесі аналізу логів

Основні підходи до виявлення атак і вторгнень. Сучасні методи виявлення атак та вторгнень базуються на кількох ключових підходах. Перший з них — сигнатурний аналіз, який передбачає пошук відомих шаблонів атак у потоках даних або логах. Його перевага — висока точність і швидкість виявлення загроз, що вже відомі, однак він повністю безсилий проти атак нульового дня або раніше невідомих методів. Іншим підходом є поведінковий аналіз, який вивчає нормальні моделі дій користувачів або систем і виявляє аномальні відхилення. Такий метод більш ефективний у випадку нетипових або нових загроз, однак генерує значну кількість хибнопозитивних спрацювань, потребує ретельного налаштування та адаптації до середовища. Також широко використовується кореляція подій, що дозволяє об’єднувати низку на перший погляд незначних подій у один цілісний інцидент для глибокого аналізу, наприклад, кілька невдалих спроб входу в систему, які завершуються успішною автентифікацією, можуть свідчити про атаку брутфорсом. І ще один перспективний напрям — використання машинного навчання, яке дозволяє самостійно знаходити нові шаблони загроз на основі великих обсягів логів. Цей метод застосовується в SIEM-системах із можливостями User and Entity Behavior Analytics (UEBA), але вимагає високої обчислювальної потужності та якісних навчальних даних. У сукупності всі ці підходи формують гнучку систему кіберзахисту, що здатна адаптуватися до змін ландшафту загроз (див. рис. 1.5) [4].

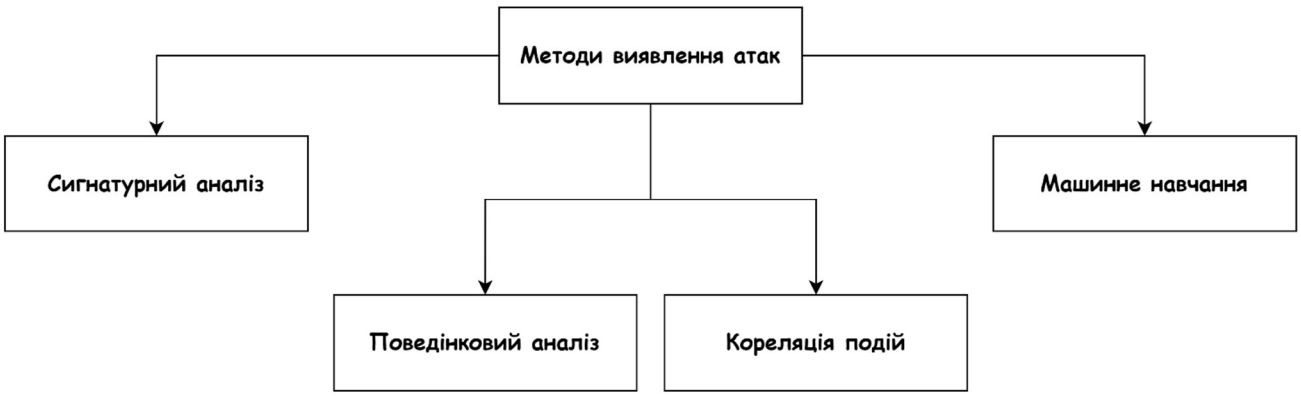


Рисунок 1.5 – Основні методи виявлення атак та їх взаємозв’язок

У підсумку, забезпечення інформаційної безпеки в мережах охоплює широкий спектр завдань — від виявлення типових загроз до побудови моделей їхньої поведінки. Моніторинг залишається незамінним елементом сучасної безпеки, а журнали подій — критично важливим джерелом даних для аналізу інцидентів. Підходи до виявлення атак постійно вдосконалюються — від традиційного сигнатурного аналізу до складних алгоритмів машинного навчання, що дозволяють працювати з непередбачуваними та складними загрозами. Усе це об’єднується в системах класу SIEM, які стали стандартом у сфері захисту корпоративних та державних інформаційних інфраструктур. Компанії, які обробляють критичні дані або мають власні дата-центри, зобов’язані впроваджувати такі системи на рівні інфраструктури, оскільки без цього неможливо забезпечити повноцінний контроль і відповідність сучасним вимогам безпеки. Натомість для звичайних користувачів або організацій, які використовують комерційні хмарні сервіси на зразок Google Drive або Microsoft 365, питання моніторингу й захисту значною мірою передані постачальнику сервісів. Однак у випадках, коли інфраструктура є приватною або критичною за своїм значенням, відповідальність за впровадження SIEM-рішень лягає на власника середовища, і саме такі системи забезпечують фундаментальний рівень безпеки [1-4].

1.2 Принципи роботи SIEM-систем

У цифровому середовищі, де щосекунди генерується мільйони подій у мережах та інформаційних системах, забезпечення своєчасного виявлення загроз вимагає всеосяжної автоматизації процесів, що поєднують функціонал збору, обробки, аналізу та зберігання подій безпеки з численних джерел, і саме в цьому контексті виникає потреба розуміння сутності SIEM-систем, їхньої структури, принципів роботи та переваг централізованого підходу до моніторингу, що дає

змогу реалізувати єдину точку контролю для подій, пов’язаних з інформаційною безпекою. SIEM — скорочення від Security Information and Event Management — становить клас систем, що забезпечує централізований збір, зберігання, аналіз і кореляцію подій безпеки з різноманітних джерел задля виявлення загроз і порушень політики в реальному часі або шляхом ретроспективного аналізу, і при цьому така система виконує комплексне перетворення необроблених даних на придатну до дій аналітичну інформацію [5-7].

Визначення SIEM – функції збору, нормалізації, аналізу логів. Основною складовою роботи такої системи є механізм збору логів (Log Collection), який охоплює отримання журналів подій з:

— операційних систем;

— мережевого обладнання;

— програмного забезпечення;

— баз даних;

— систем контролю доступу та інших джерел.

Цей процес може здійснюватися як за допомогою агентів, встановлених на вузлах, так і у безагентський спосіб через стандартні інтерфейси або протоколи доступу до логів, наприклад syslog. Після етапу збору відбувається нормалізація логів, що є перетворенням подій з різних джерел, кожне з яких має власну структуру повідомлень, у єдиний уніфікований формат, де використовуються стандартизовані поля на кшталт timestamp, event type, source IP, username та інші, що забезпечує уніфікований підхід до їх подальшого аналізу. Завдяки цьому дані з різних систем стають придатними до автоматизованої обробки, що дозволяє виявити складні зв’язки між подіями, які на перший погляд здаються незалежними [5].

Аналіз, як наступний етап обробки, передбачає виявлення закономірностей або відхилень у послідовностях подій: одна невдала спроба входу може не мати особливого значення, однак десятки або сотні таких спроб за короткий проміжок часу, поєднані з подальшим успішним входом або зміною прав доступу, вказують на можливу атаку типу brute-force або внутрішнє зловживання, і саме цей етап дозволяє не лише виявляти інциденти, а й створювати підставу для їх

підтвердження у розслідуваннях. Ще одним критичним компонентом є зберігання даних (Data Retention), де журнали подій архівуються на довготривалий період, з можливістю повнотекстового пошуку, що забезпечує можливість аудиту, комплаєнсу, історичного аналізу загроз і форензики навіть через роки після інциденту, при цьому дані можуть бути збережені як у локальних, так і у хмарних сховищах, в залежності від архітектури рішення [5].

Функції оповіщення та звітності реалізуються шляхом автоматичного надсилання повідомлень про події, які відповідають певним критеріям, що вказують на підозрілу або потенційно шкідливу активність, а також за допомогою регулярного формування звітів, які агрегують статистичні дані про стан інформаційної безпеки у системі, що дозволяє адміністраторам і аналітикам контролювати рівень захищеності IT-інфраструктури та оперативно реагувати на нові виклики (див. рис. 1.6).

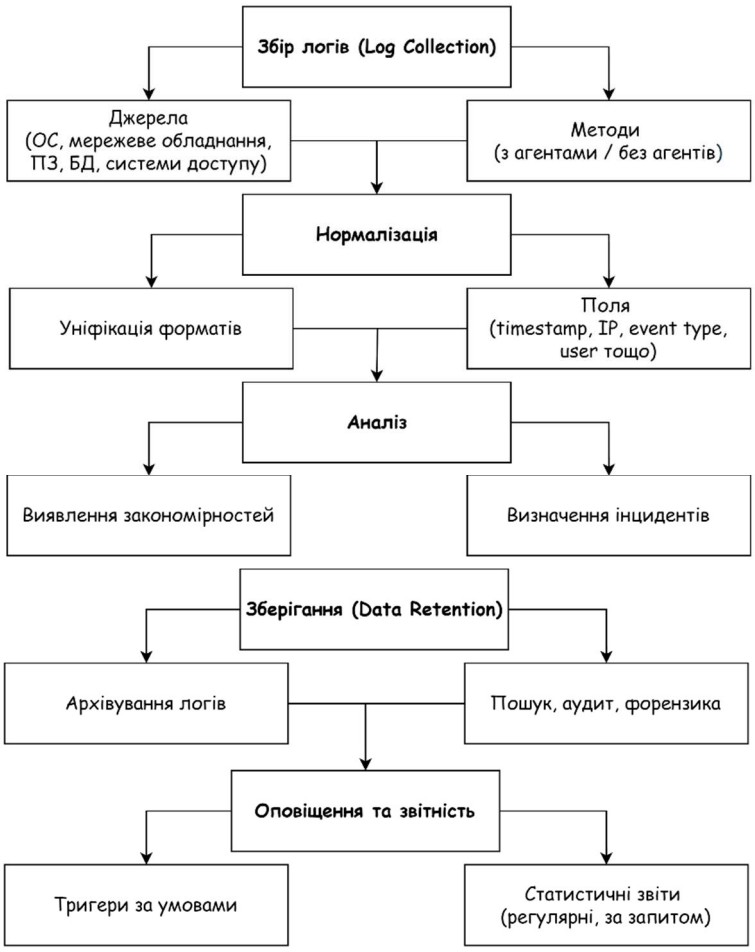


Рисунок 1.6 – Принцип роботи SIEM-системи

Усе це разом формує логіку роботи SIEM-системи як центрального елементу архітектури інформаційної безпеки, що дозволяє виявляти, розуміти і нейтралізовувати загрози на основі автоматизованої обробки великих обсягів подій, генерованих усіма компонентами інформаційної інфраструктури, і тим самим створює фундамент для побудови ефективного захисту у масштабних або критичних за даними середовищах.

Основні компоненти SIEM – агенти, колектори, рушії кореляції. SIEM-система являє собою комплексну архітектуру, побудовану на взаємодії ряду ключових компонентів, кожен з яких відповідає за специфічний етап обробки подій безпеки, і правильне налаштування кожного з них прямо впливає на ефективність роботи всієї системи, оскільки без такої структури вона не зможе виконувати свої основні функції.

Серед основних компонентів SIEM можна виокремити:

— програмні агенти;

— колектори;

— рушії кореляції.

Агенти — це програмні модулі, які встановлюються на джерелах подій, зокрема на серверах, робочих станціях, маршрутизаторах чи інших вузлах, де відбувається генерація логів, вони збирають події та передають їх далі, можуть виконувати попередню фільтрацію, шифрування або хешування даних, і, як приклад, у рішенні Wazuh використовується агент Ossec, що працює на клієнтських машинах і передає логи на сервер. Колектори, в свою чергу, є проміжною ланкою між агентами та аналітичним ядром системи, отримують події як від агентів, так і безпосередньо від джерел, наприклад через Syslog або API, відповідають за первинну маршрутизацію, обробку та нормалізацію подій, і в екосистемі ELK ці функції зазвичай виконує Filebeat або Logstash. Рушій кореляції або correlation engine — це центральний модуль аналітики, який використовує правила для виявлення підозрілих шаблонів поведінки на основі частоти подій, часових інтервалів, вмісту логів та інших характеристик, застосовуючи логіку умов, шаблони, регулярні вирази, що дозволяє поєднувати окремі події в складні інциденти. Також критично важливим

компонентом є сховище та база даних, де зберігаються як самі логи, так і метадані, історичні записи, і в сучасних реалізаціях, таких як Elasticsearch, зазвичай застосовуються NoSQL-рішення для забезпечення швидкого пошуку та фільтрації. Візуалізація реалізується через веб-інтерфейси, де за допомогою таких інструментів, як Kibana, адміністратори можуть будувати інтерактивні дашборди, здійснювати пошук, фільтрацію подій за часом, джерелом чи типом загрози, отже вся ця сукупність модулів формує цілісну SIEM-архітектуру, де кожен елемент доповнює інший і забезпечує узгоджене функціонування системи загалом (див. рис. 1.7) [6].

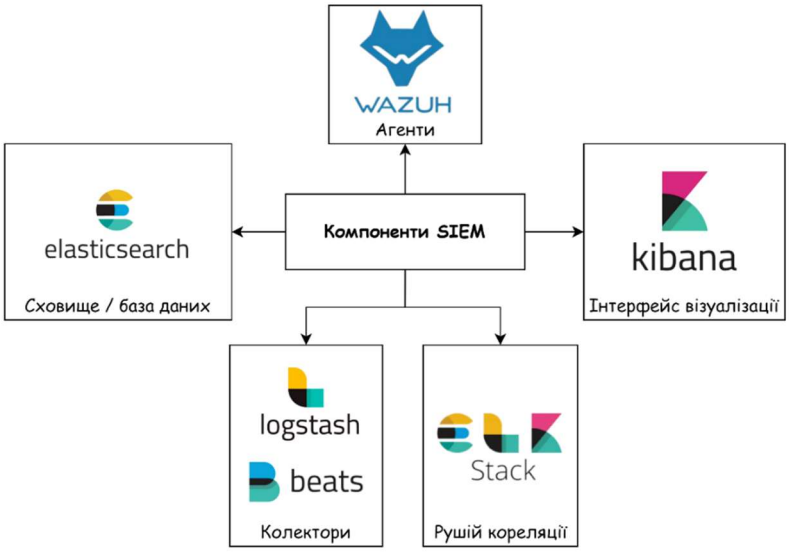


Рисунок 1.7 – Компоненти SIEM: агенти, рушії, колектори

Потоки даних у SIEM – від події до звіту. Увесь процес обробки даних у SIEM-системі являє собою послідовний потік, що починається з генерації подій у джерелі, наприклад, коли користувач вводить неправильний пароль і операційна система реєструє відповідну подію, яку одразу перехоплює агент або колектор. Далі подія передається в центральну систему, де проходить через етап нормалізації — тобто перетворення у стандартизований формат із уніфікованими полями, що дає можливість об’єднувати її з іншими логами з різних джерел. Після цього система виконує аналітичну обробку, де за допомогою рушія кореляції подія аналізується на предмет того, чи є вона окремим випадком чи елементом ширшого

шаблону — наприклад, якщо виявлено багаторазові невдалі спроби входу з різних IP-адрес протягом короткого проміжку часу, система класифікує цю поведінку як потенційну атаку. Після аналізу створюється інцидент безпеки, і система здійснює оповіщення адміністратора — наприклад, надсилає e-mail, повідомлення в консоль або записує інцидент у дашборд, після чого подія архівується в базі даних і може бути включена до регулярних звітів, які формуються щотижня, щомісяця або за запитом. Усе це забезпечує перетворення розрізнених, сирих подій у релевантну, структуровану інформацію, яка вже може бути використана для розслідувань, аудиту або оперативного реагування (див. рис. 1.8) [6-7].



Рисунок 1.8 – Схема передачі подій у SIEM

Переваги централізованого моніторингу подій безпеки. Централізований підхід до моніторингу подій безпеки за допомогою SIEM-систем забезпечує численні переваги, які складно досягти в умовах фрагментованого контролю над інформаційною інфраструктурою. Завдяки концепції «єдиного вікна спостереження» (Single Pane of Glass), адміністратор може бачити усі події, незалежно від їхнього джерела чи географічного розташування, в одному інтерфейсі, що спрощує пошук, аналіз, кореляцію та прийняття рішень. Швидкість реакції на загрози зростає завдяки автоматичним оповіщенням, які надходять у реальному часі, що дозволяє не лише виявляти загрозу, а й одразу починати дії з її усунення. Крім того, автоматизовані правила та шаблони дозволяють зменшити кількість людських помилок, оскільки рішення приймаються системою на основі аналізу великої кількості факторів, а не виключно на розсуд оператора. Повнота

розслідувань також підвищується завдяки централізованому зберіганню історії подій — можна легко відстежити, що сталося перед інцидентом, у якій послідовності, і відновити повну картину. Дотримання стандартів, таких як ISO/IEC 27001 або PCI DSS, теж забезпечується автоматично, оскільки журналювання, моніторинг, аудит та формування звітів — це невід'ємні складові функціоналу будь-якої повноцінної SIEM-системи. Ще однією важливою перевагою є масштабованість — архітектура таких систем дозволяє легко додавати нові джерела даних, створювати нові правила, інтегруватися з зовнішніми рішеннями чи платформами без суттєвої перебудови всієї інфраструктури, що особливо важливо в умовах швидкого розвитку компаній, зростання обсягів даних або змін у структурі мережі. Такий підхід дозволяє не лише керувати локальними системами, а й централізовано забезпечувати безпеку всієї організації в умовах зростаючої складності IT-ландшафту та збільшення числа загроз у цифровому просторі [7].

У підсумку SIEM-системи є невід'ємним стратегічним компонентом у забезпеченні кібербезпеки організації, оскільки вони не просто зберігають або аналізують журнали подій, а й активно виявляють складні шаблони атак, автоматично формують інциденти, забезпечують зворотний зв'язок та візуалізацію у зручному для адміністраторів вигляді, а завдяки своїй архітектурі можуть ефективно інтегруватися в будь-яку інформаційну екосистему, виконуючи функції моніторингу, реагування, звітності та довгострокового зберігання даних, що є критично важливим у реаліях сучасного кіберсередовища [5-7].

1.3 Основні класи SIEM-рішень на ринку

У міру розвитку цифрових інфраструктур, зростання складності мережевих середовищ та збільшення кількості кіберзагроз, компанії, організації та підприємства стикаються з необхідністю мати у своєму розпорядженні ефективні

інструменти для виявлення, обробки та реагування на інциденти інформаційної безпеки, що робить актуальним вибір серед широкого спектру SIEM-рішень, представлених на сучасному ринку. Ці рішення умовно поділяються на два основних класи:

— комерційні;
— відкриті (open source).

Кожен з них має свою специфіку впровадження, набір функцій, гнучкість, вартість, масштабованість, рівень підтримки, що в сукупності визначає доцільність їх застосування в тій чи іншій організаційній реальності залежно від наявних вимог до інформаційної безпеки, ресурсу інфраструктури, технічної команди та масштабу бізнесу [8-10].

Комерційні SIEM-рішення – ключові функції. Комерційні SIEM-рішення зазвичай пропонуються як комплексні платформи з розширеним функціоналом, що дозволяє швидко розгорнути інструмент моніторингу безпеки з мінімальними зусиллями щодо налаштування з нуля, тому вони часто орієнтовані на великі компанії з високими вимогами до безпеки та складною IT-інфраструктурою.

Їх перевага полягає в:

— готовності до масштабування;
— наявності вбудованих аналітичних рушіїв;
— повної інтеграції з різноманітними зовнішніми джерелами;
— підтримці машинного навчання;
— наданні технічної підтримки;
— автоматичних оновлень;
— розширених засобів звітності.

Наприклад, Splunk Enterprise Security є потужною платформою з підтримкою обробки великих обсягів даних у режимі реального часу, пропонуючи модулі поведінкового аналізу, кореляції подій, машинного навчання, а також розвинену візуалізацію даних через дашборди та KPI-інтерфейси, і попри свою високу вартість, це рішення виправдовує себе гнучкістю архітектури, масштабованістю та глибоким функціоналом [8].

Іншим прикладом комерційного рішення є IBM QRadar, яке забезпечує глибоку інтеграцію з аналітикою мережевого трафіку (Network Flow Analysis) та поведінковим аналізом користувачів (User Behavior Analytics), підтримує розширену кореляцію даних з використанням аналітичних можливостей Watson AI, що підвищує точність виявлення складних інцидентів та дозволяє адаптуватися до динамічних загроз, і хоча вартість впровадження залишається суттєвою, високий рівень інтелектуальності та автоматизації дозволяє підвищити ефективність служби безпеки в умовах великого навантаження. До списку відомих комерційних SIEM також входить Micro Focus ArcSight — одне з найбільш досвідчених рішень на ринку, яке орієнтоване на критичну інфраструктуру, включно з фінансовими установами, банками та енергетичним сектором, де важливе значення має надійність кореляції, масштабована архітектура та можливість адаптації під різні типи навантаження, хоча при цьому ArcSight потребує детального налаштування і може бути складнішим у впровадженні для малих команд без відповідної технічної підготовки [8].

Комерційні рішення забезпечують повний життєвий цикл моніторингу подій безпеки з гарантованим рівнем обслуговування, включаючи технічну підтримку 24/7, регулярні оновлення безпеки, аудит, сертифікацію відповідно до стандартів, а також інтеграцію з корпоративними засобами захисту, такими як DLP, IAM, SOAR, що робить їх доцільним вибором для середніх і великих підприємств, які потребують централізованого контролю, високої точності виявлення загроз і швидкого реагування без необхідності розробляти рішення з нуля або витрачати значні ресурси на їх підтримку.

Open-source рішення – можливості та обмеження. У контексті побудови систем моніторингу та виявлення загроз важливим фактором, який впливає на вибір програмного забезпечення, стає можливість:

— самостійного контролю;

— налаштування;

— адаптації інструментів без значних фінансових вкладень.

Тому саме на цьому етапі на перший план виходять open-source рішення, які приваблюють насамперед малі та середні організації своєю доступністю, відсутністю ліцензійних витрат і високим рівнем гнучкості. Вони дозволяють не лише знизити витрати на впровадження, а й отримати повноцінний функціонал для базового й навіть середнього рівня безпеки, тож одним із найпоширеніших варіантів, що заслуговує окремої уваги, є Wazuh — відгалуження від проєкту OSSEC, що інтегрується з E.L.K. стеком (ElasticSearch, Logstash, Kibana), надаючи інструменти для:

— гнучкого масштабування;

— аналізу журналів;

— контролю цілісності файлів;

— виявлення загроз через систему вбудованих правил.

Завдяки можливості розгортання в Docker або Kubernetes середовищі таку систему можна швидко запустити й налаштувати під конкретні потреби, що робить її надзвичайно привабливою для організацій із обмеженими ресурсами, хоча потрібно враховувати, що значна частина налаштувань виконується вручну, а офіційна технічна підтримка відсутня — є лише підтримка через спільноту [8-9].

Проте попри ці недоліки, активна спільнота користувачів компенсує відсутність офіційної підтримки, оскільки форуми й спеціалізовані майданчики, де обговорюють саме такі рішення, рясніють матеріалами від досвідчених фахівців та ентузіастів, що вже стикалися з подібними викликами. Аналогічно можна розглянути й OSSIM (Open Source SIEM), який є комплексним пакетом від AlienVault, що об’єднує Snort, OpenVAS, Nmap, OSSEC, Tcpdump та інші інструменти у єдину платформу, забезпечуючи багаторівневий захист з централізованою консоллю моніторингу. Дане рішення більше підходить для тестових середовищ або навчальних проєктів через обмежену функціональність порівняно з платною версією, однак воно дозволяє здобути практичний досвід, що й визначає його цінність у певних контекстах. Саме з цієї причини, попри наявність OSSIM, у роботі зосереджено увагу на Wazuh, адже останній пропонує оптимальне співвідношення між функціональністю, гнучкістю масштабування та простотою

інтеграції, що й стало визначальним чинником при його виборі для демонстрації можливостей open-source підходу до реалізації SIEM-систем [9].

Open-source рішення, на кшталт Wazuh, дозволяють будувати систему без жодних ліцензійних обмежень — єдина вартість полягає у витратах часу на встановлення, налаштування та обслуговування, які цілком компенсуються гнучкістю масштабування та інтеграції з будь-якими зовнішніми системами за допомогою лог-файлів або API (див. рис. 1.9).

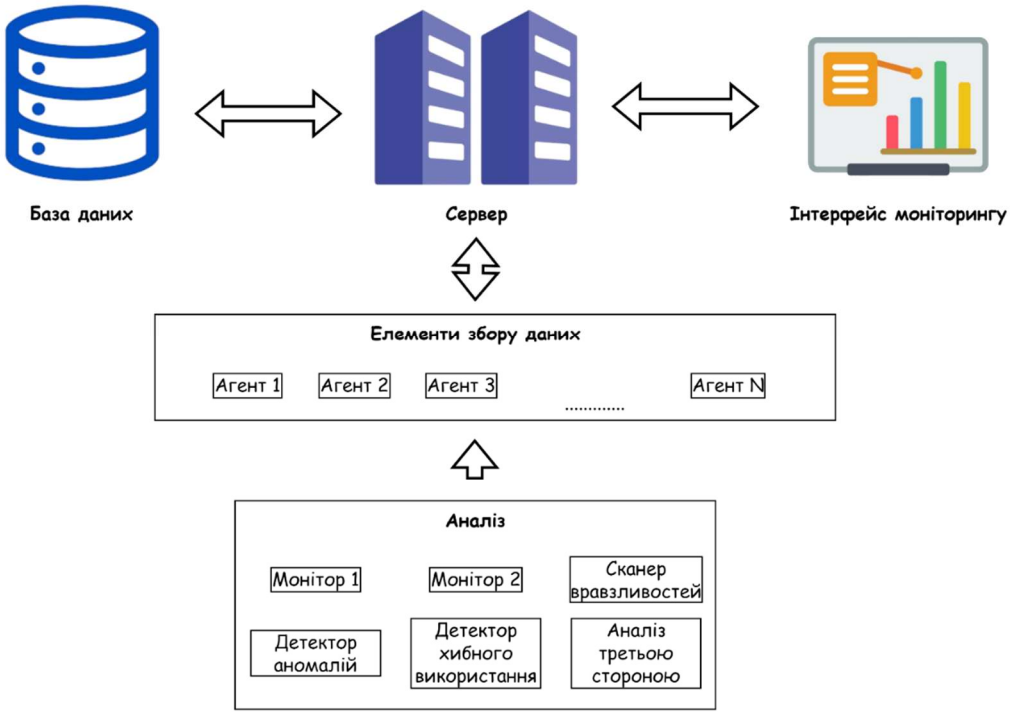


Рисунок 1.9 – Архітектура типової open-source SIEM-системи

Крім того, підтримка з боку спільнот, хай і неофіційна, досить потужна: велика кількість учасників готові ділитися знаннями, досвідом, шаблонами конфігурацій, що дає змогу швидко знаходити відповіді на проблеми, а отже — прискорює впровадження. Водночас необхідно враховувати і недоліки:

— відсутність офіційної технічної підтримки;
— обмежена продуктивність при обробці великих обсягів даних;
— потреба в глибоких технічних знаннях для конфігурації й обслуговування.

Усе це накладає додаткові вимоги до персоналу. Тим не менш, саме це й формує досвід, на базі якого згодом можна переходити до впровадження більш складних або комерційних рішень, уже розуміючи базові принципи їхньої роботи. У такому підході open-source інструменти стають не лише економічним, але й освітнім містком між експериментальним середовищем і промисловим впровадженням [8-9].

Порівняння за критеріями – вартість, масштабованість, підтримка. Якщо оцінювати різні типи SIEM-рішень за ключовими критеріями (див. табл. 1.1), такими як вартість, масштабованість, технічна підтримка, інтеграція з іншими системами, гнучкість налаштувань і можливості кореляції подій, то стає очевидно, що комерційні продукти типу Splunk, IBM QRadar або ArcSight суттєво випереджають за рівнем автоматизації, підтримки, інтеграції та точності аналітики, але водночас потребують значних фінансових вкладень на всіх етапах — від ліцензії до технічного супроводу. У той час як Wazuh та OSSIM, будучи представниками відкритих рішень, дозволяють з нульовими витратами побудувати базову або середню за складністю SIEM-інфраструктуру, в якій доведеться більше уваги приділяти ручному налаштуванню, моніторингу та усуненню несправностей. Якщо говорити конкретно про масштабованість, то всі комерційні рішення демонструють високий рівень підтримки великих інфраструктур із можливістю інтеграції в гібридні або мультихмарні середовища, натомість Wazuh теж цілком ефективно масштабується завдяки Docker і Kubernetes, тоді як OSSIM вважається менш придатним до таких сценаріїв через складність налаштування й обмежену підтримку великих обсягів даних [10].

Таблиця 1.1 – Порівняння основних SIEM-рішень за критеріям

Критерій	**Splunk**	**QRadar**	**ArcSight**	**Wazuh**	**OSSIM**
Вартість	Висока	Висока	Висока	Безкоштовно	Безкоштовно
Масштабованість	Висока	Висока	Висока	Висока (через Docker/ELK)	Середня
Гнучкість налаштувань	Висока	Середня	Висока	Висока	Середня
Технічна підтримка	Так	Так	Так	Community	Community
Інтеграція з іншими системами	Висока	Висока	Висока	Висока	Середня
Кореляція подій	Інтелектуальна	Гнучка	Потужна	Середня	Базова

Стосовно налаштувань, Splunk, ArcSight і Wazuh забезпечують високий ступінь гнучкості, тоді як QRadar та OSSIM — більш обмежені, хоча у випадку QRadar це компенсується глибоко інтегрованим інтерфейсом. Щодо технічної підтримки, то всі платні рішення передбачають офіційний супровід, що є критичним для підприємств, де простої системи неприпустимі, натомість у open-source середовищі ця функція перекладається на плечі користувачів, спільнот і документації. У питанні інтеграції з іншими системами ситуація подібна: Splunk, ArcSight та QRadar інтегруються з широким спектром рішень за допомогою плагінів або API, при цьому Wazuh, хоч і безкоштовний, теж має досить широкий потенціал інтеграції, що й вирізняє його серед open-source інструментів, тоді як OSSIM більш орієнтований на прості випадки або внутрішнє тестування [9-10].

Якщо перейти до оцінки кореляції подій, то Splunk вирізняється високим ступенем інтелектуальної обробки даних, QRadar пропонує гнучку архітектуру для створення правил, ArcSight — потужні можливості виявлення складних ланцюжків подій, тоді як Wazuh надає базові, але достатні можливості кореляції для середовищ середнього рівня складності, а OSSIM підтримує лише мінімальний рівень цієї функціональності. Таким чином, попри очевидну перевагу комерційних продуктів у масштабних впровадженнях, open-source рішення, за умов грамотного налаштування, здатні забезпечити функціональність, яка цілком конкурує з платними альтернативами — особливо коли йдеться про проєкти з обмеженим бюджетом, навчальні ініціативи або невеликі компанії, що прагнуть контролювати процес без капіталовкладень [9-10].

Тенденції розвитку SIEM – хмарні рішення, AI/ML, автоматизація. Технології систем централізованого моніторингу (СМ), до яких належать і SIEM-рішення, стрімко еволюціонують у відповідь на зростаючі виклики кібербезпеки, адже саме наявність цих викликів зумовила актуальність впровадження подібних систем. У разі відсутності проблем з безпекою компанії концентрувалися б на інших напрямках розвитку, проте постійне загострення кіберзагроз змушує адаптувати й розширювати можливості існуючих рішень (див. рис. 1.10).

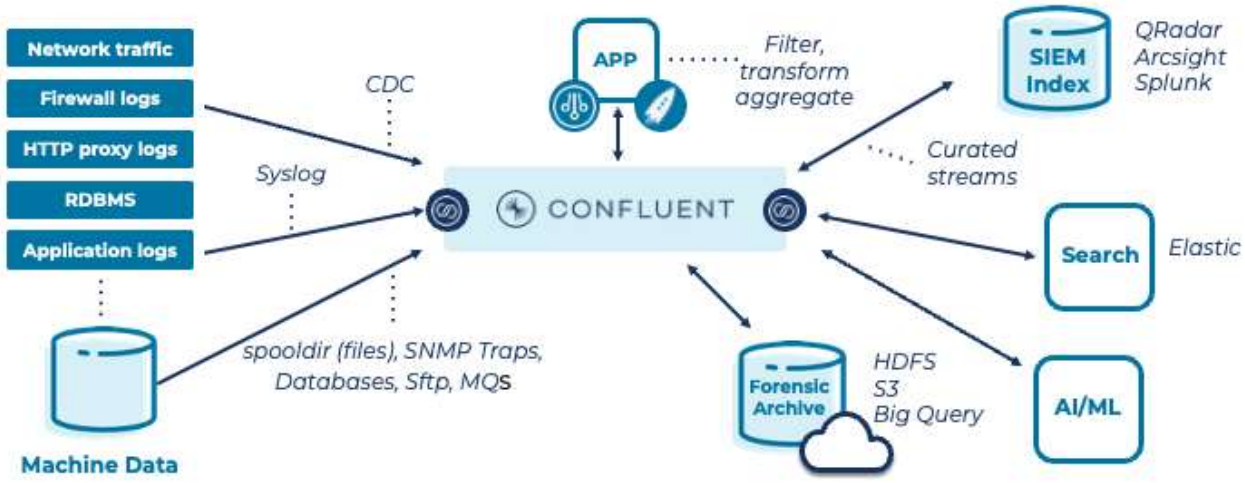


Рисунок 1.10 – Тенденції розвитку: хмарні SIEM, AI, SOAR [10]

У процесі модернізації або вибору нового інструменту безпеки важливо орієнтуватися на сучасні технологічні вектори розвитку, серед яких одним із головних є перехід до хмарної інфраструктури. Використання cloud-native SIEM-систем, таких як Microsoft Sentinel або Sumo Logic, забезпечує:

— гнучкість;
— масштабованість;
— зменшення навантаження на локальні ресурси;
— відсутність потреби у власному хостингу;
— мінімальну технічну підтримку з боку користувача.

Усе це дозволяє бізнесу сконцентруватися на стратегічних цілях. Разом з тим таке рішення неминуче підвищує залежність від стороннього постачальника, що може стати ризиком, зокрема для організацій із критичною інфраструктурою, які часто обирають локальні рішення, ізольовані від глобальної мережі, створюючи замкнені середовища з використанням VPN або навіть повністю автономних конфігурацій. У таких умовах хмарні рішення здебільшого залишаються пріоритетом для приватного сектору або проєктів, які потребують динамічного масштабування та швидкого впровадження [10].

Паралельно з розвитком хмарної інфраструктури спостерігається активна інтеграція SIEM з технологіями штучного інтелекту та машинного навчання, що дозволяє виявляти нетипові та складні загрози, які виходять за межі звичайних правил кореляції. У таких випадках застосування моделей поведінкового аналізу (UEBA – User and Entity Behavior Analytics) сприяє ідентифікації аномалій у діях користувачів і систем, що надає змогу розпізнати кібератаки ще до того, як вони наберуть загрозливих масштабів. Приклади комерційних рішень на кшталт QRadar або Splunk демонструють ефективність таких підходів, оскільки їх вартість обґрунтована наявністю вбудованих AI-модулів, які автоматизують значну частину аналітичних процесів, зменшуючи навантаження на фахівців з безпеки та підвищуючи точність виявлення атак [8, 10].

Ще одним важливим напрямком розвитку SIEM є автоматизація процесів реагування через впровадження SOAR-рішень (Security Orchestration, Automation

and Response), які доповнюють функціональність традиційних систем і дозволяють переходити від ручного адміністрування інцидентів до їх автоматизованої обробки, що суттєво скорочує час реагування на інциденти – від годин до лічених секунд – завдяки можливості автоматичного виконання дій на кшталт блокування IP-адрес, закриття портів, надсилання повідомлень або запуску скриптів усунення загроз. Така інтеграція дозволяє організаціям зменшити залежність від людського фактора та підвищити загальний рівень готовності до кіберінцидентів, що особливо критично у випадках, коли йдеться про захист важливих інфраструктур або чутливої інформації [8, 10].

Серед актуальних напрямів розвитку варто також виділити XDR (Extended Detection and Response) — концепцію, що передбачає розширену інтеграцію SIEM з іншими інструментами кібербезпеки, зокрема EDR (Endpoint Detection and Response) та NDR (Network Detection and Response). Така інтеграція дає змогу охопити більший обсяг даних, зібраних з різних джерел, що, у свою чергу, підвищує точність кореляції, пришвидшує виявлення загроз і забезпечує кращу координацію між різними системами безпеки в межах однієї організації, що також дозволяє створити єдину платформу для управління всіма аспектами захисту – від моніторингу до реагування – з мінімальним залученням операційного персоналу [9-10].

На сучасному ринку представлено широкий спектр SIEM-рішень — від потужних комерційних продуктів з повним спектром підтримки до гнучких open source платформ, які можна адаптувати під конкретні вимоги організації з мінімальними фінансовими витратами. Комерційні рішення, такі як ArcSight, IBM QRadar чи Splunk, пропонують високу надійність, масштабованість і підтримку, проте вимагають значних ресурсів на впровадження та обслуговування. Водночас open source альтернативи на зразок Wazuh або ELK дозволяють створити ефективну систему моніторингу навіть на базі існуючої інфраструктури, що є особливо привабливим варіантом для організацій із обмеженим бюджетом або проєктів, які прагнуть гнучкості та контролю над налаштуванням [10].

У контексті загальної динаміки кіберзагроз та потреби в оперативному реагуванні майбутнє SIEM-систем визначається інтеграцією інтелектуальних можливостей, хмарної масштабованості та автоматизованих рішень, що дозволяють будувати адаптивну і стійку до атак архітектуру. Поєднання хмарних технологій, штучного інтелекту, машинного навчання, автоматизації та розширеної взаємодії з іншими системами кіберзахисту формує наступний етап розвитку систем безпеки, здатних не лише виявляти, а й ефективно нейтралізовувати загрози в реальному часі. З огляду на постійне зростання складності атак і темпів їх поширення, саме така еволюція є логічною відповіддю на нові виклики цифрового середовища [8-10].

1.4 Стандарти та вимоги до SIEM-систем

В умовах цифрової трансформації підприємства, компанії та інші організаційні структури повинні не лише підтримувати існуючу IT-інфраструктуру, а й діяти згідно з міжнародними стандартами та нормативними вимогами, які регламентують обробку даних, зберігання логів і звітність щодо інцидентів безпеки, тому SIEM-рішення дедалі частіше стають інструментом, що інтегрує аудит, моніторинг та автоматизований контроль політик безпеки. Враховуючи це, виникає потреба глибше розуміти нормативні вимоги до таких систем, їхню роль у політиках інформаційної безпеки та критерії вибору конкретного рішення з урахуванням організаційної специфіки, що зумовлено тим, що кожна реалізація системи управління інформаційною безпекою є індивідуальною і формується на основі поточних ризиків, до яких вразлива конкретна організація. Оскільки компанії часто оперують чутливими цифровими даними, що становлять значну цінність, вони стають мішенню постійних кібератак. З огляду на майже повну цифровізацію бізнес-процесів, виникає необхідність у більш суворому дотриманні стандартів та вимог, що визначені міжнародними

нормами, які дозволяють не лише впровадити відповідні механізми безпеки, а й підтримувати їхню дієвість у мінливому кіберсередовищі, забезпечуючи відповідність актуальним загрозам і підвищуючи загальну стійкість IT-інфраструктури [11-13].

Міжнародні стандарти – ISO/IEC 27001, NIST, GDPR та інші. Серед ключових міжнародних стандартів, що стосуються SIEM, варто насамперед виділити:

— ISO/IEC 27001;
— NIST SP 800-92;
— SP 800-137;
— GDPR;
— PCI DSS;
— HIPAA;
— ISO/IEC 27035.

Так, ISO/IEC 27001 є основоположним стандартом управління інформаційною безпекою, що вимагає створення повноцінної системи управління інформаційною безпекою (СУІБ), де важливою складовою є функціональність SIEM для реєстрації інцидентів, виявлення аномалій та підтримки контрольованих заходів, зокрема таких як контроль A.12.4. Отримання сертифікації за цим стандартом передбачає наявність чітких процедур моніторингу та реагування на інциденти, що повинні бути задокументованими та реалізованими на практиці. У свою чергу, рекомендації Національного інституту стандартів і технологій США (NIST), зокрема документи SP 800-92 і SP 800-137, містять вимоги до стратегії моніторингу подій, форматів логування та механізмів зберігання. SP 800-92 виступає гідом щодо збору логів і визначає обов’язковість кореляції та аналітики даних, а SP 800-137 описує процес безперервного моніторингу як фундамент захисту критичної інфраструктури. Регламент ЄС GDPR регулює обробку персональних даних, і SIEM у цьому випадку використовується для виявлення несанкціонованого доступу до таких даних, відстеження змін і своєчасного оповіщення про витік даних — протягом 72 годин відповідно до статті 33. PCI DSS, що застосовується до компаній, які

працюють із платіжними картками, встановлює обов'язковість логування критичних подій, а HIPAA — до організацій, що мають справу з медичними даними, вимагає забезпечення звітності щодо доступу до захищеної медичної інформації (PHI). ISO/IEC 27035, у свою чергу, фокусує увагу на управлінні інцидентами інформаційної безпеки, описуючи процес реагування як ключовий компонент загальної структури безпеки (див. рис. 1.11).

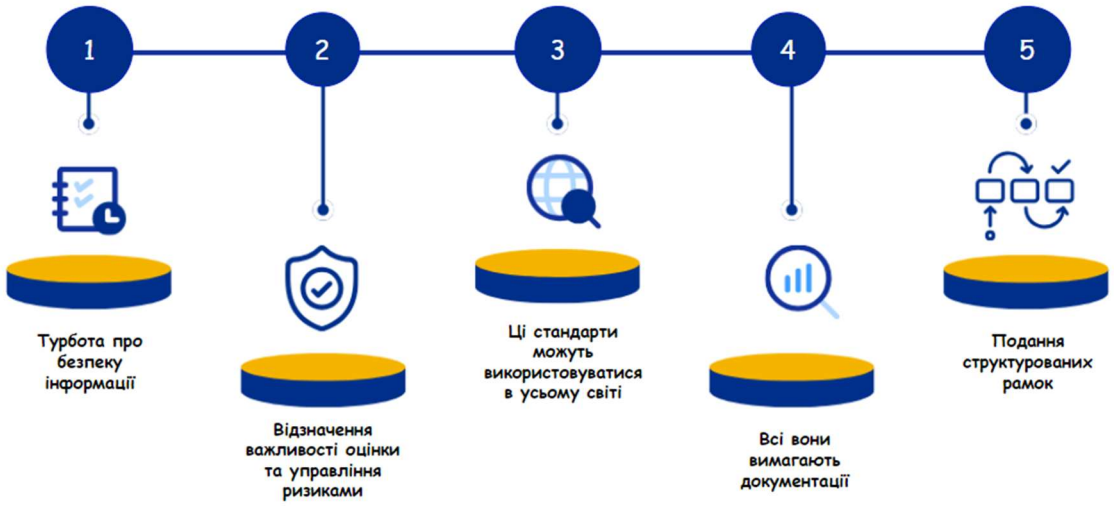


Рисунок 1.11 – Взаємозв'язок стандартів ISO/IEC 27001, NIST і SIEM

Всі ці стандарти створюють основу, на якій будується відповідність SIEM-систем нормативним вимогам, сприяючи:

— зниженню ризиків;

— забезпеченню прозорості обробки даних;

— підтримці довгострокової кіберстійкості [11].

Вимоги до зберігання логів, цілісності та звітності. Окрім відповідності стандартам, SIEM-системи повинні забезпечувати надійне логування, цілісність даних та звітність, що включає, передусім, обов'язкове логування з усіх критичних джерел у режимі реального часу, причому логи мають бути захищені від втрат завдяки чергам подій або резервним лог-серверам, що зберігають копії у разі атаки, спрямованої на знищення цифрових слідів. Важливим аспектом є забезпечення цілісності логів, що досягається шляхом захисту від змін, зокрема через

використання цифрових підписів, контрольних сум (SHA-256) або носіїв типу WORM, які не дозволяють редагування. У разі аудиту повинна бути можливість довести, що логи не були підроблені або змінені, і будь-яке розходження контрольної суми буде свідчити про потенційну компрометацію. Централізоване зберігання логів є ще одним критично важливим компонентом, адже воно дозволяє ефективніше управляти даними. Наприклад, у випадку з системою Wazuh, яка працює на базі Elasticsearch, логи зберігаються у централізованому сховищі з можливістю подальшої архівації. Регламентований термін зберігання логів зазвичай варіюється від трьох місяців (для оперативного доступу) до одного-п’яти років (для архівів), причому системи повинні підтримувати автоматичне видалення або архівацію відповідно до політики організації. Окрему увагу варто приділити звітності: SIEM-система має генерувати звіти для аудиторів, IT-служб та керівництва, які включають типи атак, джерела інцидентів, порушення політик доступу, а також ступінь відповідності стандартам. Такі звіти мають бути адаптованими до регуляторних вимог і формуватися за шаблонами, які відповідають ISO, GDPR та іншим стандартам (див. рис. 1.12).



Рисунок 1.12 – Цикл зберігання та перевірки логів

Сукупність зазначених вимог до логування, забезпечення цілісності та звітності створює підґрунтя для юридично значущої та технічно ефективної SIEM-системи, яка не лише фіксує події, а й дозволяє будувати на їх основі захисні механізми відповідно до актуального контексту загроз [12].

Політики безпеки і роль SIEM у відповідності до стандартів. Політика безпеки — це сукупність регламентованих правил, інструкцій та механізмів, що визначають порядок управління інформаційними активами в межах організації, при цьому SIEM-система стає важливим елементом реалізації цієї політики через фіксацію спроб несанкціонованого доступу, виявлення спроб підвищення привілеїв або порушень політик доступу, а кореляційні механізми забезпечують виявлення таких інцидентів у реальному часі, що дозволяє формувати автоматизовані сценарії реагування, зокрема миттєве блокування IP-адрес або інших джерел загроз, у поєднанні з SOAR-модулями це створює цілісну інфраструктуру реагування на інциденти. Водночас безперервний аудит дій користувачів, особливо тих, що мають розширені повноваження, реалізується через повне логування активності з фіксацією часових міток і деталей дій, що підвищує якість внутрішніх розслідувань та сприяє прозорості дій у межах корпоративного середовища, а ще одна важлива складова – підтримка безперервного вдосконалення через аналітику на основі логів, яка дає змогу аналізувати ефективність чинних політик безпеки, виявляти нові шаблони атак та відповідно оновлювати регламентовані правила безпеки, щоб забезпечити постійну актуальність засобів захисту [13].

Критерії вибору SIEM з урахуванням вимог організації. Вибір SIEM-системи для організації не обмежується технічними характеристиками, а базується на здатності системи відповідати чинним нормативним вимогам, цілям бізнесу та масштабу діяльності, тому одним із першочергових критеріїв є відповідність міжнародним стандартам, зокрема підтримка готових шаблонів звітності для GDPR, PCI DSS, ISO/IEC 27001, що дозволяє ефективно документувати дотримання політик безпеки та демонструвати регуляторам відповідність, крім цього, важливою характеристикою є масштабованість системи, адже організаціям малого та середнього бізнесу достатньо рішень на основі open-source, як-от Wazuh, тоді як для великих підприємств доцільно розглядати системи з:

— розподіленою архітектурою;

— високою доступністю;

— підтримкою горизонтального масштабування.

Критично важливим критерієм є підтримка широкого переліку джерел даних, зокрема Active Directory, хмарних сервісів, антивірусного ПЗ, мережевого обладнання, а також можливість інтеграції через API з зовнішніми рішеннями для повної уніфікації даних у межах інфраструктури організації, тому підтримка безперервного моніторингу, візуалізації у вигляді дешбордів, гнучких механізмів сповіщень, алгоритмів поведінкового аналізу та аналітики на основі AI доповнює загальну функціональність сучасного SIEM. Не менш важливо оцінити вартість впровадження — починаючи від ліцензійних платежів, витрат на розгортання, налаштування, навчання персоналу, підтримку, тоді як open-source рішення потребують більше часу на конфігурацію, однак забезпечують більшу гнучкість та глибше налаштування під специфіку потреб підприємства, при цьому залишаючись економічно ефективними [12-13].

SIEM-система виступає як технологічне ядро дотримання сучасних вимог кібербезпеки, забезпечуючи не лише моніторинг, збір та аналіз подій безпеки, а й формалізований підхід до відповідності нормативним документам на кшталт ISO/IEC 27001, NIST, GDPR, PCI DSS, де кожне впроваджене рішення базується на потребах конкретної організації з урахуванням галузі, ризиків та рівня оброблюваних даних, включно з можливістю інтеграції у вже наявну інфраструктуру з урахуванням бюджету, технічних обмежень та регуляторного середовища (див. рис. 1.13).

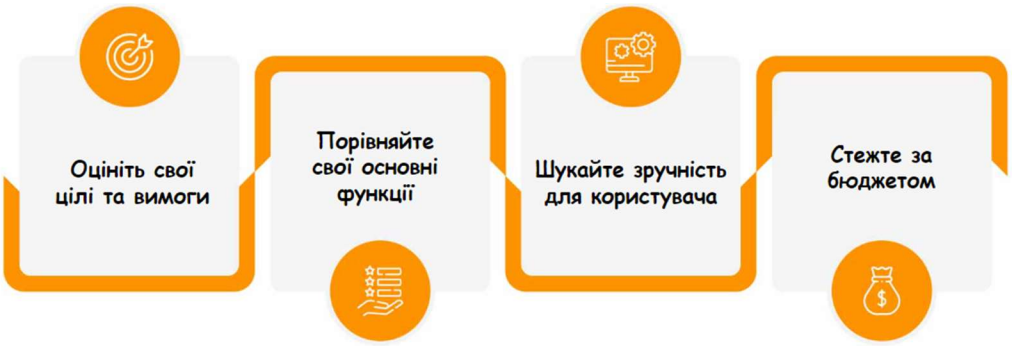


Рисунок 1.13 – Етапи вибору SIEM-системи

При правильному налаштуванні така система не лише дозволяє своєчасно виявляти загрози, а й виконує роль засобу формального підтвердження відповідності інформаційної безпеки вимогам, демонструючи системність підходу до кіберзахисту, професійність організації та її відповідальність у поводженні з цифровими активами в умовах зростаючої складності загроз [11-13].

1.5 Інструменти виявлення загроз у SIEM

SIEM-системи за своєю суттю не обмежуються лише збором і зберіганням логів, оскільки вони функціонують як активні елементи у виявленні та запобіганні кіберзагрозам, забезпечуючи цілісну картину подій у мережевому середовищі завдяки здатності до аналізу інформації в режимі реального часу, що дозволяє ідентифікувати аномальні дії, шкідливі прояви активності або відхилення від нормальної поведінки. На прикладі відкритого рішення Wazuh, яке є сучасною open-source платформою, можна простежити реалізацію ключових інструментів для виявлення загроз, серед яких як сигнатурні, так і поведінкові методи, а також можливості інтеграції з зовнішніми базами загроз, системами кореляції подій і засобами візуалізації на основі Kibana, що створює умови для побудови гнучкого й економічного рішення, придатного до масштабування в межах невеликих або середніх організацій, де часто існує потреба оптимізувати витрати без втрати контролю над безпекою [14-15].

Механізми виявлення загроз – сигнатурні та поведінкові методи. Платформа Wazuh підтримує два ключові підходи до виявлення загроз, один з яких базується на сигнатурному методі, тобто на правил-орієнтованому виявленні, де визначальний механізм полягає у використанні заздалегідь сформульованих правил (як-от правила у форматі "rules.xml"), які описують відомі патерни атак, і під час надходження логів агент платформи порівнює ці записи з уже існуючими шаблонами, викликаючи спрацювання у разі збігу. Наприклад, якщо фіксується

декілька поспіль невдалих спроб входу до системи, формується підозра на brute-force атаку, або якщо у логах Shell виявляється команда з потенційно шкідливою поведінкою, це може вказувати на експлуатацію вразливості. Серед основних переваг сигнатурного підходу — точність у виявленні вже відомих атак і оперативність реагування, що дозволяє суттєво скоротити час від виявлення до інформування відповідальних осіб або ініціювання автоматичних дій. Водночас недоліком цього підходу є обмеженість у виявленні нових, ще не задокументованих загроз і постійна залежність від актуальності бази сигнатур, яка потребує систематичного оновлення, інакше механізм втрачає ефективність [14].

Альтернативою, що дозволяє компенсувати ці обмеження, є поведінковий метод виявлення загроз, або так зване anomaly-based detection, який базується на аналізі поведінки користувачів, систем і процесів у часі для виявлення відхилень від усталеної моделі активності, при цьому Wazuh має змогу фіксувати параметри на зразок часу входу, геолокації, частоти команд, профілю дій користувачів, і при виявленні нетипового сценарію дій, наприклад авторизації з іншої країни, може генерувати тривогу як можливий індикатор компрометації облікового запису. До переваг цього підходу належить здатність виявляти атаки нульового дня (zero-day), які не мають відомої сигнатури, а також адаптація алгоритмів до специфіки конкретного середовища, що робить поведінкову аналітику ефективною в умовах швидко змінюваних загроз. Проте існують і складнощі, зокрема ризик хибнопозитивних спрацювань, які потребують додаткового аналізу та доопрацювання правил, і потреба у початковому налаштуванні з навчанням системи, щоб вона могла коректно ідентифікувати нормальну поведінку для конкретної організації (див. рис. 1.14) [14].

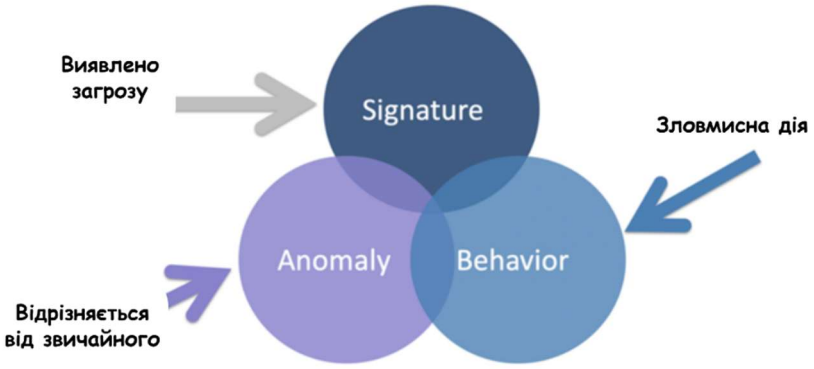


Рисунок 1.14 – Методи виявлення загроз: сигнатурний, поведінковий та аномальний

Інтеграція Wazuh з базами правил – VirusTotal, MITRE ATT&CK та інші. Для підвищення ефективності виявлення загроз SIEM-система Wazuh може бути інтегрована з зовнішніми джерелами інформації, які надають додатковий контекст для аналізу подій та прийняття рішень щодо інцидентів. Одним із таких ресурсів є VirusTotal — популярна платформа для перевірки файлів, IP-адрес і URL-адрес на наявність шкідливого програмного забезпечення. Wazuh підтримує інтеграцію з цим сервісом, надсилаючи хеші файлів до бази VirusTotal і отримуючи інформацію про їхню потенційну шкідливість, що дозволяє автоматизувати оцінку ризиків без участі адміністратора, спираючись при цьому на результати роботи понад 70 антивірусних рушіїв, що забезпечує більшу точність і гнучкість в порівнянні з використанням одного джерела [14-15].

Додатково можлива інтеграція з MITRE ATT&CK — структурованою базою знань про тактики, техніки й процедури, які використовують зловмисники. У рамках такої інтеграції події, зафіксовані в системі, можуть бути автоматично класифіковані відповідно до певних технік ATT&CK, наприклад, використання PowerShell для завантаження скриптів із мережі буде віднесене до техніки T1059.001 (див. рис. 1.15).

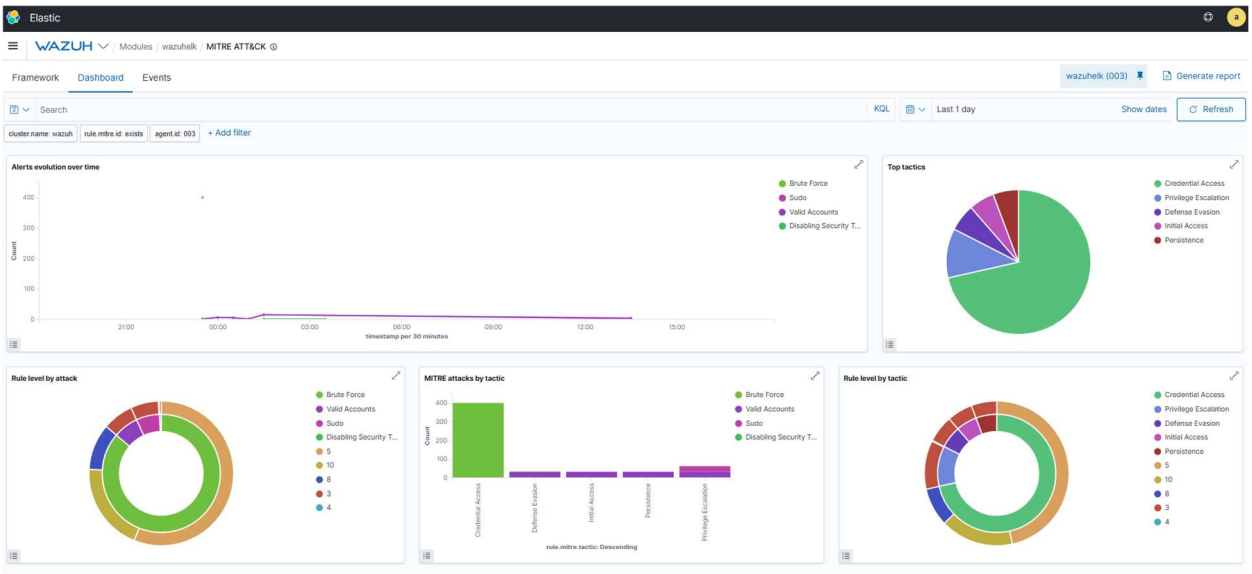


Рисунок 1.15 – Інтеграція з MITRE ATT&CK у системі Wazuh

Такий підхід дозволяє проводити картографування активності загроз, створювати моделі атак і вдосконалювати систему захисту в організації з урахуванням актуальних сценаріїв зловмисної поведінки. Окрім цього, Wazuh також підтримує інші інтеграції, такі як:

— GeoIP (виявлення геолокації підозрілих IP-адрес);

— ClamAV (антивірус для сканування файлів на агентах);

— OpenSCAP (модуль перевірки відповідності політикам безпеки).

Завдяки цим розширенням розкривається потенціал системи в цілому, що дозволяє охопити як технічні, так і організаційні аспекти моніторингу безпеки [14-15].

Кореляція подій та формування алертів. Усі зібрані події не існують у вакуумі й мають бути співвіднесені одна з одною для виявлення складних багатоступеневих атак, саме тому важливим компонентом Wazuh є кореляція подій, яка складається з двох основних етапів:

— початковий аналіз здійснюється на рівні агента;

— глибше опрацювання на сервері.

Цей процес дозволяє системі об’єднувати події за такими атрибутами як IP-адреса, ім’я користувача, часові мітки або шаблон активності. Наприклад, серія невдалих спроб входу в систему, що завершується успішним входом, після чого йде зміна системних файлів та підключення до невідомого сервера — усе це може бути

автоматично пов’язане в один ланцюг, який свідчитиме про потенційну компрометацію (див. рис. 1.16).

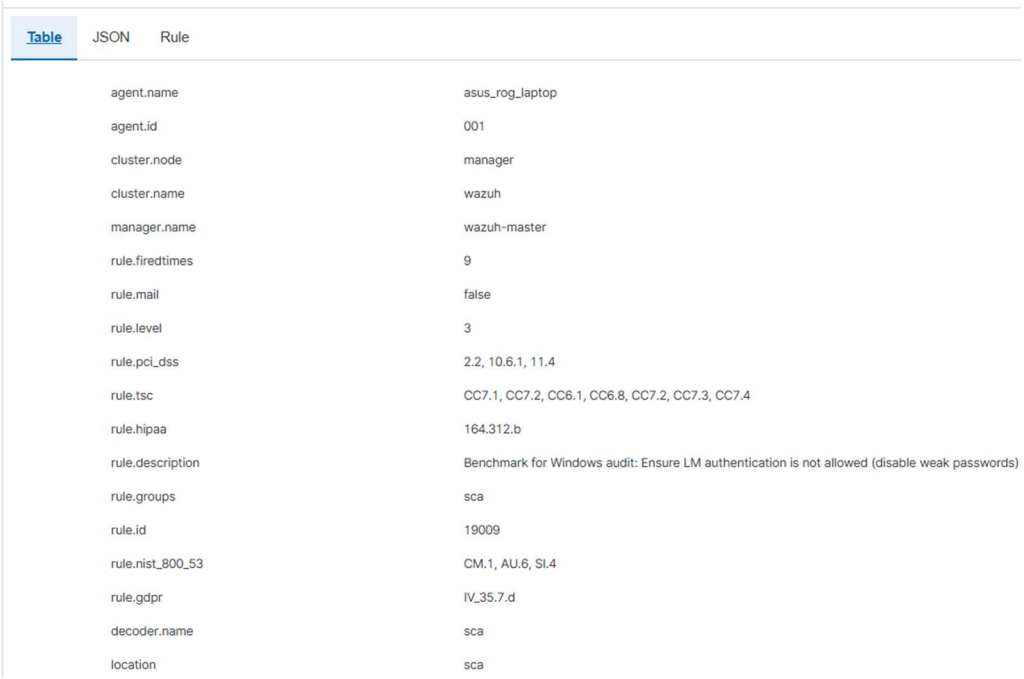


Рисунок 1.16 – Приклад кореляції подій безпеки

На основі таких сценаріїв генеруються алерти у форматі JSON, кожен із яких має рівень критичності від 0 до 15. Ці алерти можуть зберігатися локально, передаватися до Elasticsearch, а також бути налаштованими на надсилання повідомлень у поширені канали комунікації, такі як електронна пошта, Slack чи Telegram [15].

Візуалізація даних у Kibana через стек ELK. Для більш ефективного контролю над ситуацією та оперативного прийняття рішень Wazuh використовує Kibana — інтерфейс для візуалізації даних, який є частиною стеку ELK. У Wazuh є власний плагін для Kibana, що дозволяє отримати доступ до спеціалізованих панелей, таких як:

— список агентів;

— стан онлайн/офлайн;

— поточні алерти;

— фільтрація за джерелом критичністю або типом подій;

— модулі виявлення змін у файлах (FIM);

— виявлення вразливостей;

— інтеграція з базою вразливостей NVD.

Завдяки цим інструментам можливо будувати кастомні дашборди для моніторингу, з графіками за часом, тепловими картами активності IP-адрес, статистикою інцидентів за агентами. Ці панелі дозволяють зберігати шаблони для подальшого аудиту або аналітики, а також виконувати глибокий пошук в Elasticsearch за будь-яким полем, наприклад "user, hash, IP", з використанням логічних операторів та регулярних виразів (див. рис. 1.17) [15].

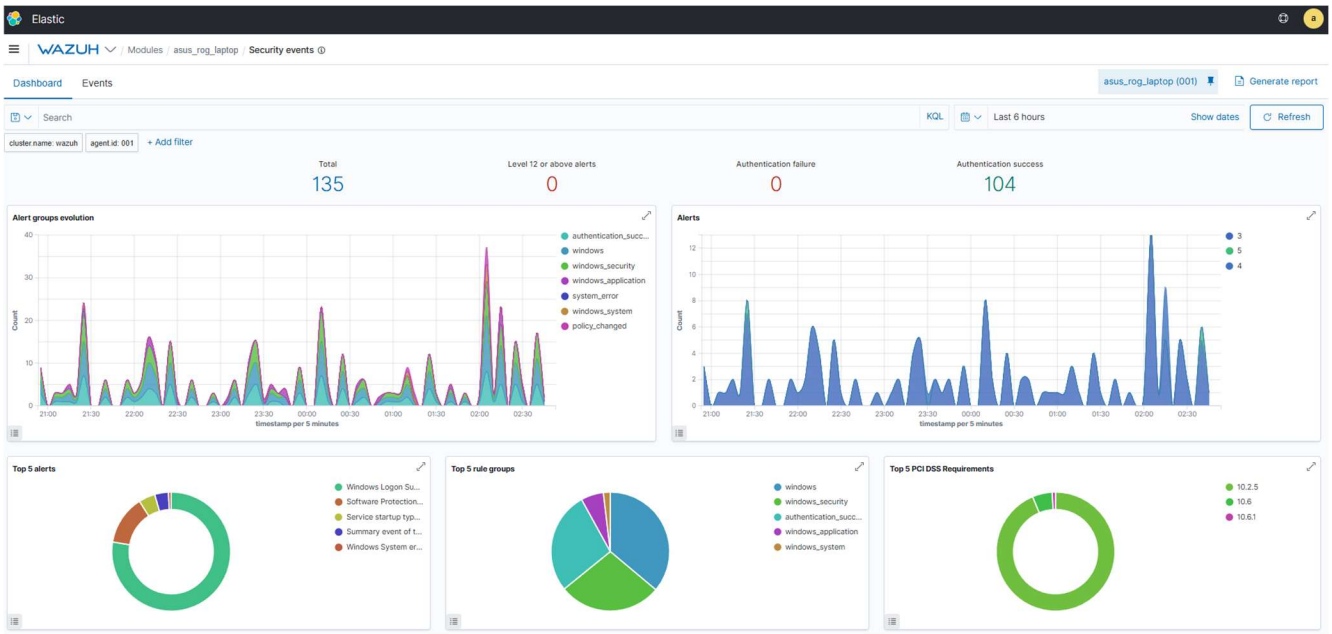


Рисунок 1.17 – Інтерфейс Kibana для візуалізації загроз

Такі можливості значно скорочують час реагування на інциденти за рахунок наочної візуалізації та виявлення прихованих закономірностей у поведінці системи або користувачів, що в комплексі дає змогу не лише швидко виявляти потенційні загрози, а й формувати чітку звітність для внутрішнього або зовнішнього аудиту без необхідності ручного формування аналітичних звітів.

Таблиця 1.2 – Основні інструменти виявлення загроз у SIEM-системі Wazuh

Напрямок	**Опис**	**Переваги**	**Обмеження / Особливості**
Сигнатурне виявлення	Аналіз логів на основі відомих шаблонів атак за наперед визначеними правилами	Висока точність при відомих загрозах, швидке реагування	Не виявляє невідомі атаки, потребує постійного оновлення правил
Поведінковий аналіз	Виявлення аномалій через відхилення від "нормальної" поведінки користувачів та систем	Можливість виявлення zero-day атак, адаптація до конкретної інфраструктури	Можливі хибні спрацювання, потребує налаштування та навчання
Інтеграція з зовнішніми джерелами	Підключення до VirusTotal, MITRE ATT&CK, GeoIP, ClamAV, OpenSCAP для підвищення точності виявлення	Підвищена інформативність, автоматизована перевірка, розширений контекст	Залежність від доступності зовнішніх сервісів, потреба в API-ключах
Кореляція подій	Поєднання множинних подій для формування цілісної картини атаки	Виявлення складних сценаріїв вторгнень, зменшення шуму	Вимагає правильної побудови логіки, можлива складність конфігурації

Продовження таблиці 1.2 – Основні інструменти виявлення загроз у SIEM-системі Wazuh

Напрямок	**Опис**	**Переваги**	**Обмеження / Особливості**
Автоматизація реагування (Active Response)	Автоматичне виконання скриптів у відповідь на події (блокування IP, вимкнення користувача тощо)	Миттєва реакція, мінімізація людського втручання	Потребує тестування скриптів для уникнення помилкових дій
Візуалізація в Kibana (через ELK Stack)	Інтерфейс для моніторингу, створення дашбордів, пошуку та фільтрації подій	Швидке виявлення інцидентів, зручність аудиту, графічне представлення даних	Потребує певних навичок роботи з Kibana та Elasticsearch

Усе це підсумовується в цілісну концепцію (див. табл. 1.2), в якій Wazuh демонструє приклад ефективного рішення у сфері SIEM, що поєднує сигнатурний і поведінковий аналіз, інтеграції з провідними джерелами інформації про загрози, гнучку систему кореляції подій і потужну візуалізацію, завдяки чому така система є придатною не лише для підприємств з високим рівнем інформаційної безпеки, а й для малих організацій, які прагнуть мати якісний захист без значних інвестицій у комерційні рішення [14-15].

РОЗДІЛ 2 РОЗГОРТАННЯ SIEM-СИСТЕМИ З ВИКОРИСТАННЯМ ІСНУЮЧИХ ІНСТРУМЕНТІВ

2.1 Архітектура обраної SIEM-системи

Для реалізації архітектури обраної SIEM-системи було обрано рішення на базі Wazuh з інтеграцією у стек ELK, який включає Elasticsearch, Logstash і Kibana, при цьому враховано особливості контейнеризованого розгортання за допомогою Docker, що дозволяє досягти високої гнучкості, масштабованості та значно спрощує розгортання в реальному середовищі. Саме Docker забезпечує можливість швидкої інсталяції, ізоляції компонентів і безпечної взаємодії між ними без потреби відкриття зовнішніх портів, що суттєво знижує ризики зовнішнього втручання та спрощує мережеву інфраструктуру [16-17].

Загальна структура Wazuh та стеку ELK – компоненти та їх зв'язки. Загальна структура архітектури Wazuh у зв'язці з ELK має модульну побудову та включає декілька основних компонентів, кожен з яких виконує чітко визначені функції в процесах збору, зберігання, обробки, аналітики та візуалізації даних. Центральним компонентом виступає Wazuh Manager — головний сервер, що приймає логи від агентів, виконує первинний аналіз, здійснює кореляцію подій і формує відповідні алерти. На клієнтських системах встановлюються Wazuh Agents, які відповідають за збір інформації про події, стан системи, змінені файли, активність процесів та інші критично важливі показники. Ці агенти надсилають зібрані дані безпосередньо до менеджера, де вони проходять перший етап обробки [16].

Далі оброблені події передаються до Elasticsearch — масштабованої системи зберігання та пошуку даних, яка приймає алерти й метадані, надаючи можливість швидкого доступу до них завдяки потужним механізмам фільтрації та індексації. Kibana використовується як інтерфейс візуалізації, що дозволяє адміністратору

працювати з графіками, панелями моніторингу, таблицями та іншими елементами аналітики, при цьому вона інтегрована із плагіном Wazuh, що додає спеціалізовані вкладки для відображення стану агентів, алертів, вразливостей і змін у системі [16].

Ключовим проміжним компонентом між Wazuh Manager і Elasticsearch виступає Filebeat — легкий агент для транспортування логів, що приймає вихідні події з менеджера та доставляє їх у сховище даних. Завдяки йому досягається розвантаження основного менеджера і гнучкість в обробці трафіку журналів. Вся взаємодія між компонентами відбувається через внутрішню Docker-мережу, що дозволяє уникнути потреби у зовнішньому відкритті портів, підвищуючи безпеку, стабільність і контрольованість системи, а також спрощуючи управління доступом і забезпеченням ізоляції служб (див. рис. 2.1) [16].

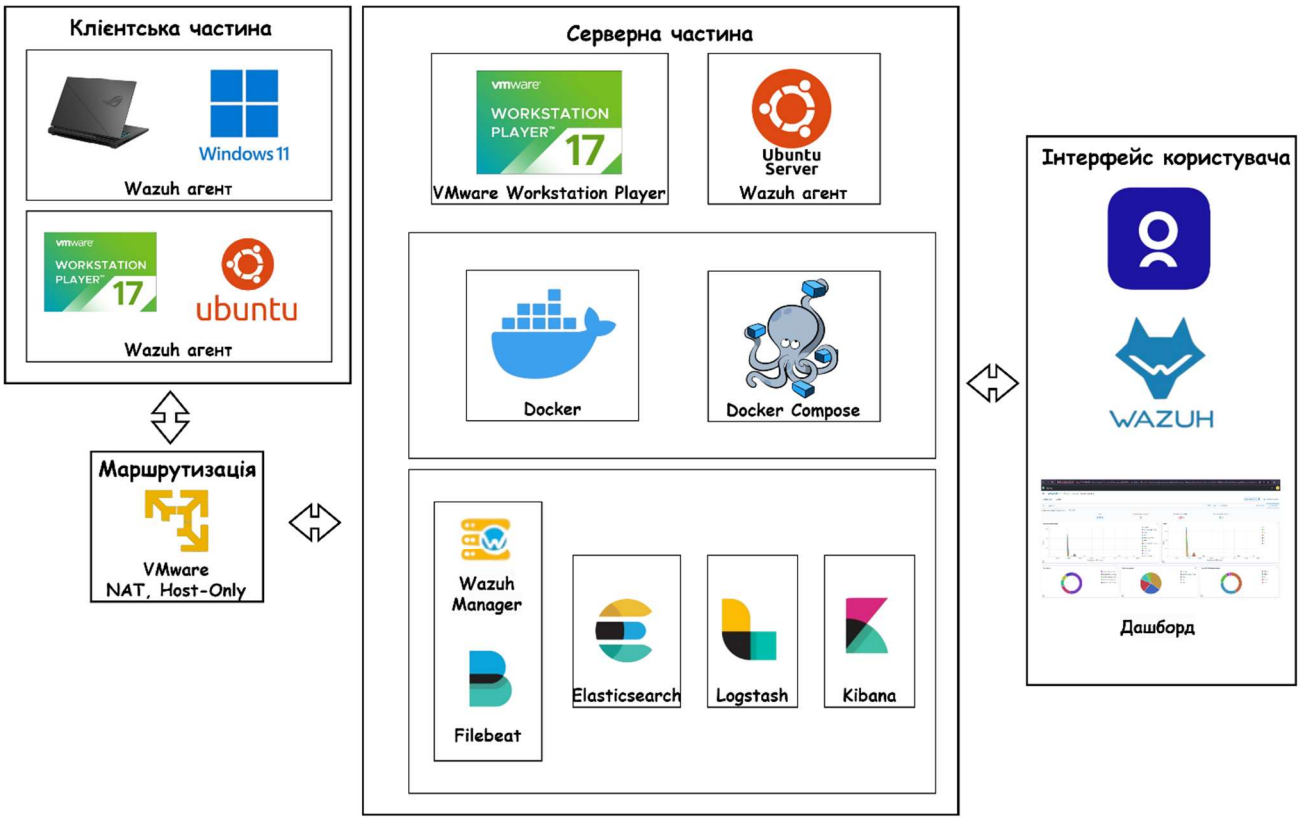


Рисунок 2.1 – Архітектура SIEM-системи на базі Wazuh та ELK

Docker-контейнери – Wazuh Manager, Elasticsearch, Logstash, Kibana, Filebeat. Оскільки для розгортання системи використовувалась контейнеризація за допомогою Docker, це забезпечило швидке, контрольоване та стандартизоване

середовище для запуску всіх необхідних компонентів без потреби у складній ручній інсталяції кожного окремого модуля. Такий підхід дозволив оперативно підготувати всю архітектуру SIEM-системи для подальшого аналізу її можливостей, що особливо важливо у випадках, коли час на налаштування обмежений або розгортання відбувається в динамічних середовищах із частими змінами. За рахунок контейнеризації стало можливим ефективно протестувати та масштабувати ключові сервіси, що становлять основу роботи платформи моніторингу безпеки [16].

Першим контейнером, що є критичним для роботи системи, виступає Wazuh Manager — ядро всієї інфраструктури, яке відповідає за:

— отримання подій з агентів;
— виконання первинної обробки;
— застосування заданих політик безпеки;
— виконання кореляції подій;
— ініціювання відповідей на загрози.

Він використовує внутрішні конфігураційні файли, які дозволяють налаштовувати правила, активні відповіді та логіку обробки даних згідно з вимогами конкретного середовища. Усі події, оброблені цим контейнером, далі передаються до Elasticsearch — другого ключового контейнера, який виконує функції сховища для всіх логів, подій і алертів. Його архітектура оптимізована для обробки великих обсягів даних у реальному часі, а також для забезпечення високої доступності та масштабованості всієї платформи моніторингу [16-17].

Наступним контейнером є Kibana, яка є інтерфейсом для візуалізації та взаємодії з даними, що зберігаються в Elasticsearch. Kibana має попередньо інтегрований Wazuh Dashboard, завдяки якому користувач отримує змогу:

— переглядати події;
— слідкувати за станом агентів;
— моніторити результати контролю цілісності файлів (FIM);
— моніторити події аудиту;
— налаштовувати інтерфейс відповідно до потреб;

— отримувати доступ до детальної інформації про всі зафіксовані інциденти.

Окремим елементом цієї структури є контейнер Filebeat, який відповідає за транспортування даних між Wazuh Manager та Elasticsearch. Він налаштовується для зчитування логів, згенерованих Wazuh, із певних директорій та передає їх у Elasticsearch у структурованому вигляді, часто у форматі JSON. Крім того, Filebeat підтримує фільтрацію повідомлень, агрегацію подій, а також може змінювати форматування вхідних логів відповідно до вимог подальшої обробки, що значно полегшує інтеграцію з іншими аналітичними або моніторинговими системами [16-17].

Для централізованого керування всією системою використовується Docker Compose — окрема утиліта, що дозволяє запускати, зупиняти й оновлювати всю сукупність контейнерів за допомогою одного конфігураційного файлу "docker-compose.yml", у якому детально прописано:

— сервіси;

— мережі;

— змінні середовища;

— залежності.

Команда запуску системи виконується у фоновому режимі через "docker-compose up -d" і забезпечує одночасне розгортання всіх необхідних елементів без потреби в індивідуальній ініціалізації кожного з них (див. рис. 2.2).



Рисунок 2.2 – Принцип взаємодії Docker-контейнерів у SIEM

Такий підхід дозволяє забезпечити узгоджене середовище виконання, пришвидшити процес розгортання та легко масштабувати систему у разі потреби [17].

Переваги контейнеризації для розгортання SIEM. Використання контейнеризації через Docker при розгортанні SIEM-систем забезпечує низку суттєвих переваг, що є особливо важливими як для експериментальних, так і для продуктивних середовищ. Такий підхід значно спрощує процес встановлення, адже Docker-контейнери дозволяють уникнути тривалих ручних налаштувань, завдяки чому повноцінне розгортання системи відбувається буквально за один крок за допомогою Docker Compose, що дозволяє істотно скоротити час, необхідний для запуску, та перейти безпосередньо до тестування або впровадження в експлуатацію. Усі компоненти працюють у ізольованих середовищах, що гарантує незалежність від операційної системи хост-машини, дозволяючи запускати контейнери на будь-якому пристрої, де встановлений Docker, і при цьому зберігати повну повторюваність і однакову поведінку системи незалежно від апаратного чи програмного середовища.

Крім цього, контейнеризований підхід забезпечує гнучкість у керуванні архітектурою, зокрема можливість масштабування окремих елементів, таких як Elasticsearch, без потреби зупиняти всю систему. Наприклад, можна додати ще один інстанс Elasticsearch або оновити лише один окремий контейнер, не зачіпаючи інші компоненти, що у традиційному підході було б пов’язано з ризиком впливу на всю інфраструктуру. Подібна структура дозволяє вмикати або вимикати частини системи окремо, оперативно змінювати конфігурації та адаптувати інфраструктуру до поточних потреб без повного демонтажу всієї системи.

Окрім технічної гнучкості, контейнеризація також дає можливість реалізувати додатковий рівень безпеки завдяки ізоляції, яку забезпечує архітектура контейнерів. Кожен з них функціонує у своєму власному ізольованому середовищі, що можна конфігурувати згідно з конкретними політиками доступу або мережевими обмеженнями. Така модель нагадує роботу у режимі пісочності (sandbox), коли дії одного компонента не мають прямого впливу на інші. Крім цього, легко налаштовуються правила міжмережевої взаємодії між контейнерами, що дає змогу створювати строго контрольовані канали зв’язку та мінімізувати ризики несанкціонованого доступу чи витоку даних.

Ще однією перевагою такого підходу є можливість легко реплікувати середовище, створюючи повноцінні копії як для розробки, так і для тестування, що особливо цінно у великих проєктах, де важливо відтворити середовище, максимально наближене до продуктивного, не витрачаючи час і ресурси на повторне встановлення всіх компонентів вручну. Традиційні способи розгортання часто стикаються з обмеженнями, пов’язаними з апаратним забезпеченням або несумісністю конфігурацій, у той час як Docker дозволяє швидко та надійно переносити систему між різними платформами без необхідності глибокої адаптації. У результаті контейнеризація не лише підвищує ефективність процесу впровадження SIEM-системи, а й робить управління нею більш передбачуваним, стандартизованим і масштабованим.

Варіанти масштабування – додавання агентів, розподіл навантаження. Система повинна бути здатною масштабуватися відповідно до зростання кількості джерел логів або рівня навантаження, що є критично важливим у контексті побудови SIEM-рішення та його подальшого впровадження на рівні конкретного підприємства чи організації. Така масштабованість забезпечує можливість адаптації до динаміки інфраструктури, зміни обсягів трафіку, кількості серверів чи робочих станцій, що потребують моніторингу. При розгортанні системи на базі Wazuh та стеку ELK передбачено кілька варіантів реалізації масштабування залежно від потреб і ресурсів. Одним із основних підходів є горизонтальне масштабування шляхом додавання нових агентів Wazuh, що дозволяє розширити охоплення інфраструктури без зміни вже налаштованого центру управління. Усі нові агенти можуть централізовано контролюватися з одного менеджера, що зберігає зручність в адмініструванні навіть за умов суттєвого зростання кількості клієнтських машин.

Додатково до цього реалізується вертикальне масштабування, яке передбачає збільшення ресурсів окремих контейнерів, зокрема для Wazuh Manager або Elasticsearch, шляхом виділення більшої кількості оперативної пам’яті чи процесорного часу, що є доцільним при зростанні навантаження або обсягів логів. Elasticsearch дозволяє створення повноцінного кластерного середовища з кількох

вузлів, що не лише підвищує продуктивність, а й забезпечує розподіл навантаження між інстансами, знижуючи ризик перевантаження та підвищуючи відмовостійкість у разі збоїв. Розгортання такого кластеру дозволяє автоматичну реплікацію індексів і швидке відновлення працездатності системи без втрати даних у випадку виходу з ладу окремих елементів.

У такому підході архітектура SIEM-системи, побудована на базі Wazuh у поєднанні зі стеком ELK та реалізована засобами контейнеризації, виявляється ефективною, гнучкою та придатною до динамічного масштабування. Завдяки модульному принципу побудови легко адаптується до змін у середовищі, а ізоляція компонентів у контейнерах значно спрощує оновлення, обслуговування та розширення системи. Вбудована підтримка масштабування дозволяє системі залишатися стабільною й ефективною навіть у разі суттєвого розширення мережі або зростання обсягів вхідних даних, що робить таке рішення придатним як для малих організацій з обмеженими ресурсами, так і для великих підприємств з розгалуженою IT-інфраструктурою та високими вимогами до безпеки.

2.2 Побудова SIEM-системи на базі Wazuh та ELK

Побудова SIEM-системи на основі Wazuh у поєднанні з ELK-стеком реалізується завдяки контейнеризації, яка дозволяє значно спростити:

— процес інсталяції;

— налаштування;

— подальшого управління всіма необхідними компонентами.

Сам процес створення такого середовища починається з підготовки інфраструктури, що включає встановлення контейнерного середовища для розміщення основних елементів архітектури SIEM-системи, таких як Wazuh Manager, Elasticsearch, Kibana та Filebeat. Після завершення етапів розгортання та підключення всіх сервісів, з’являється можливість здійснювати візуалізацію подій

і логів у Kibana, що свідчить про працездатність та коректну взаємодію між компонентами системи [18-19].

Підготовка середовища – встановлення Docker і Docker Compose. Контейнеризація, в основі якої лежить технологія Docker, дає змогу створити ізольоване середовище для кожного з компонентів SIEM-системи, уникнувши при цьому складних залежностей, пов’язаних із конфліктами бібліотек або версій програмного забезпечення. Для початку налаштування необхідно підготувати базове середовище, яке складається з:

— операційної системи;

— Docker Engine;

— утиліти Docker Compose.

Найзручнішим варіантом є використання Ubuntu Server, інсталяційний образ якого завантажується з офіційного сайту, після чого розгортається у віртуальній машині, наприклад у середовищі VMware Workstation. Така віртуальна машина має бути забезпечена достатньою кількістю ресурсів для коректного функціонування контейнерів, включаючи виділення оперативної пам’яті (8 ГБ), CPU-ядра (2 ядра) та дискового простору (20 ГБ) (див. рис. 2.3).

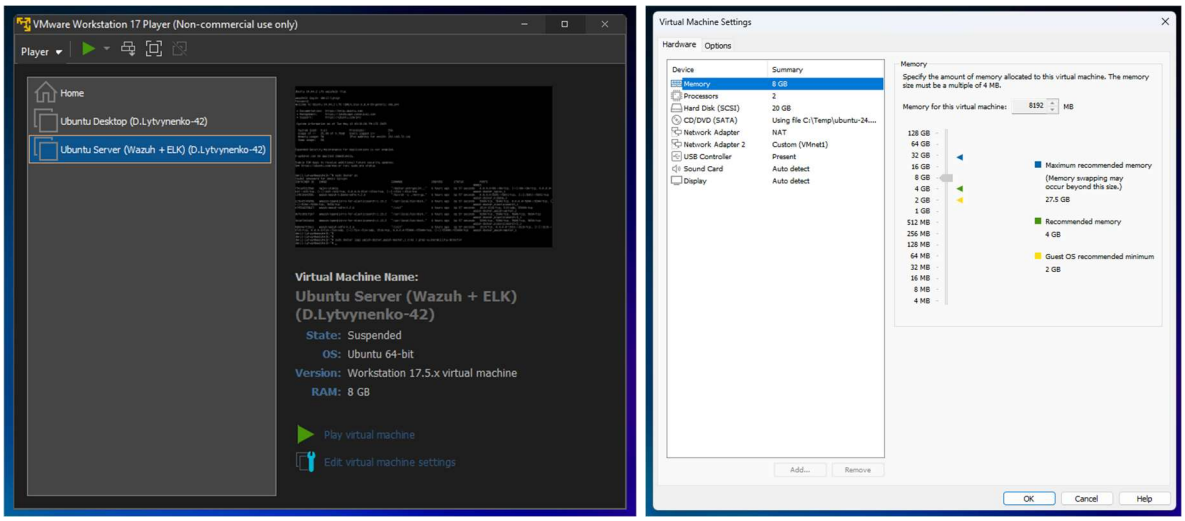


Рисунок 2.3 – Конфігурація Ubuntu Server у якості віртуальної машини

Після розгортання віртуального середовища першим етапом налаштування стає оновлення системи, що забезпечується виконанням команд "sudo apt update &&

sudo apt upgrade -y", які оновлюють пакети до останніх доступних версій, забезпечуючи стабільність і безпеку основи для подальшої інсталяції (див. рис. 2.4).

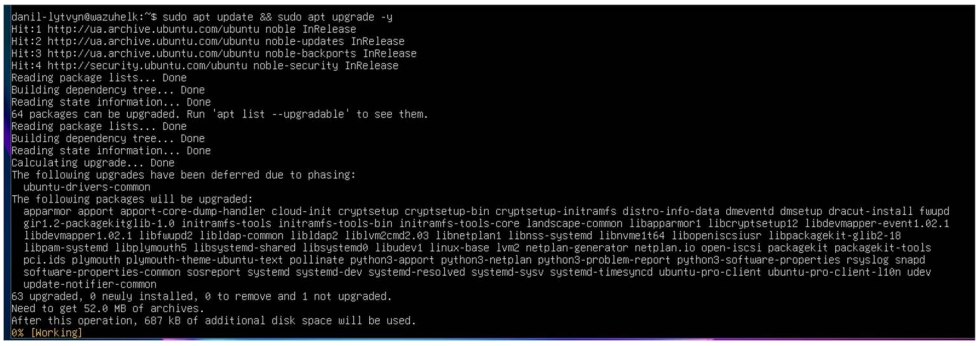


Рисунок 2.4 – Оновлення пакетів до актуальної версії

Наступним кроком є встановлення самого Docker, який забезпечує створення, управління та запуск контейнерів; це здійснюється за допомогою команди "sudo apt install docker.io -y", після чого інсталюється утиліта Docker Compose – "sudo apt install docker-compose -y", що дозволяє одночасно управляти декількома контейнерами через єдиний YAML-файл конфігурації (див. рис. 2.5).

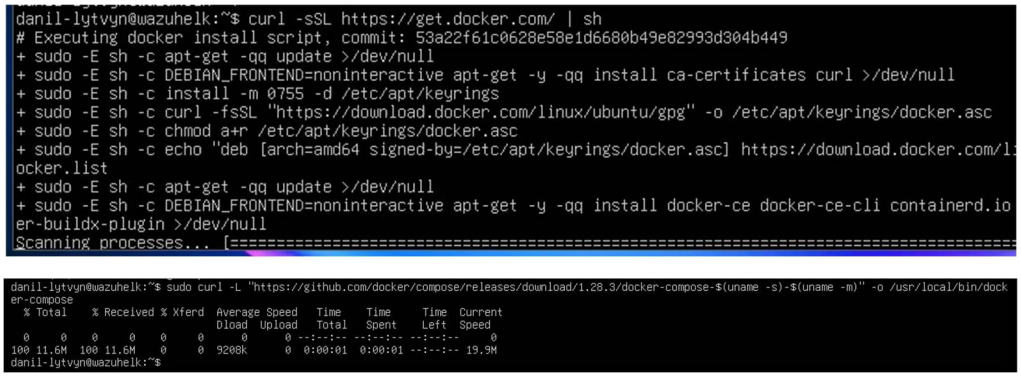


Рисунок 2.5 – Процес розгортання Docker та Docker Compose

Перевірити коректність інсталяції обох інструментів можна командами "docker --version і docker-compose --version' (див. рис. 2.6).



Рисунок 2.6 – Перевірка версії Docker Compose

У результаті цього етапу формується готова до роботи система, в межах якої можливо безперешкодно розгортати необхідні контейнери Wazuh, Elasticsearch, Kibana, Filebeat та інші допоміжні сервіси, що формують ядро SIEM-інфраструктури (див. рис. 2.7).

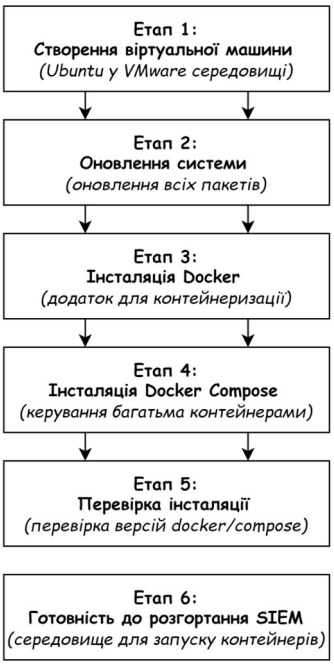


Рисунок 2.7 – Етапи підготовки SIEM системи як середовища на базі Docker

Такий підхід дозволяє забезпечити контрольовану, ізольовану і масштабовану архітектуру безпеки, придатну як для навчальних цілей, так і для розгортання в робочому середовищі з реальними даними.

Клонування офіційного репозиторію Wazuh Docker. Саме рішення Wazuh надає офіційний репозиторій, у якому вже підготовлено всі необхідні Docker-файли для повноцінного запуску системи без потреби у самостійному створенні базової структури. Початковим кроком для розгортання є клонування цього репозиторію за допомогою команди "git clone https://github.com/wazuh/wazuh-docker.git", після чого необхідно перейти до відповідного каталогу "cd wazuh-docker" (див. рис. 2.8).

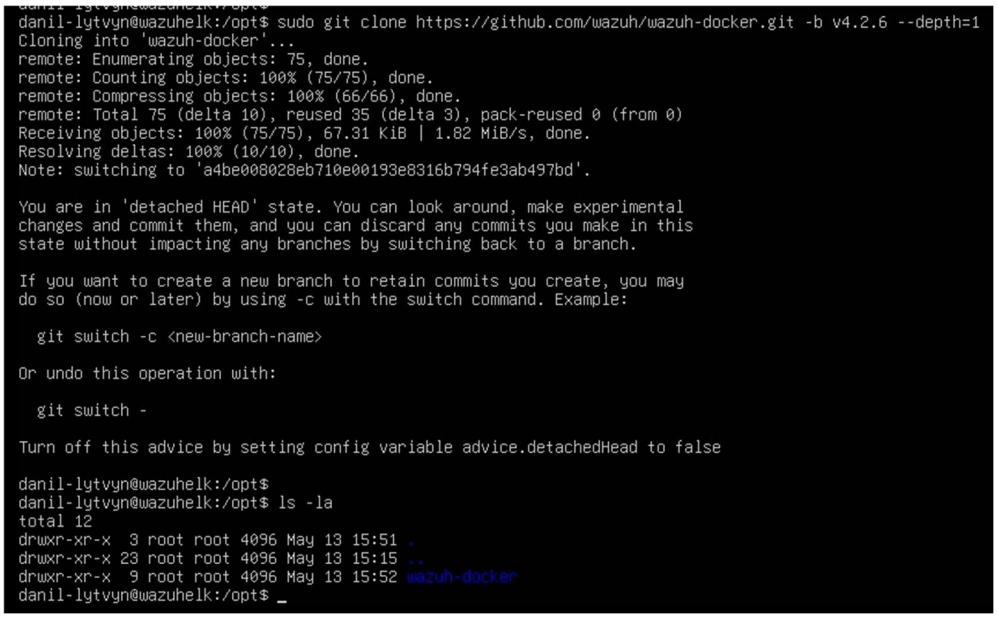


Рисунок 2.8 – Клонування репозиторію Wazuh зі ELK стеком

У цьому сховищі зібрано попередньо налаштоване середовище для запуску всіх компонентів SIEM-рішення, що включає Wazuh та ELK стек, а також пов’язані з ними конфігураційні шаблони та допоміжні скрипти. Структура репозиторію продумана таким чином, щоб забезпечити як легкий старт для стандартного розгортання, так і можливість точкового налаштування для конкретних сценаріїв [19].

Усередині репозиторію розміщено кілька ключових директорій, кожна з яких виконує певну функцію в архітектурі запуску. Каталог "config" містить конфігураційні файли для окремих сервісів, що дозволяє централізовано змінювати параметри без необхідності втручання у внутрішню структуру контейнерів. Каталог "certs" використовується для зберігання TLS-сертифікатів, необхідних для забезпечення шифрованої взаємодії між компонентами системи. Директорія "extensions" включає розширення для Kibana, зокрема офіційний плагін Wazuh, який забезпечує візуалізацію даних безпеки та інтеграцію функціональності SIEM безпосередньо в інтерфейс користувача. Основу запуску становить файл "docker-compose.yml", у якому описано, які саме контейнери необхідно запускати, у якій послідовності, з якими параметрами й змінними середовища (див. рис. 2.9).

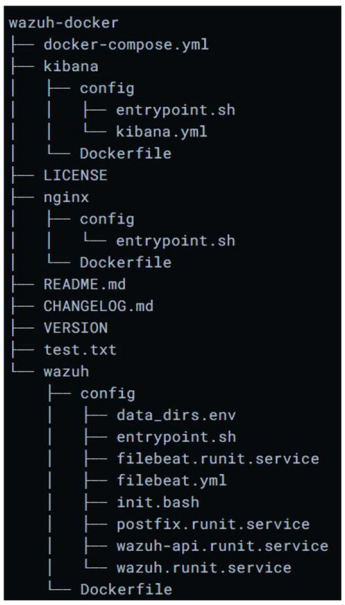


Рисунок 2.9 – Приклад структури офіційного репозиторію Wazuh Docker

Використання офіційного репозиторію дозволяє значно зменшити час початкового налаштування, оскільки структура вже протестована й підтримується командою Wazuh. Оновлення конфігурацій, залежностей та версій сервісів централізовано відслідковуються й синхронізуються, що усуває ризики сумісності при самостійному збиранні. При цьому зберігається можливість вносити зміни до наявних шаблонів, адаптуючи систему під власні вимоги, змінюючи порти, політики безпеки, ресурси чи зовнішні томи. Такий підхід поєднує у собі переваги готового рішення з відкритістю до кастомізації, дозволяючи будувати надійне й гнучке SIEM-середовище без потреби створення усього вручну [19].

Налаштування сертифікатів та запуск сервісів. Wazuh разом із стеком ELK використовує TLS/SSL для забезпечення захищеної комунікації між усіма компонентами, що входять до архітектури системи, і генерація сертифікатів виконується ще до запуску контейнерів, що дозволяє правильно налаштувати взаємну автентифікацію між сервісами. Для цього використовується спеціальна команда "docker-compose -f generate-opendistro-certs.yml run --rm generator", яка виконує запуск контейнера з підготовленим скриптом генерації сертифікатів, що у свою чергу створює набір самопідписаних сертифікатів для окремих компонентів:

— Elasticsearch як сервера;

— Filebeat як клієнта;

— Wazuh Manager як центральна частина архітектури обробки пакетів безпеки.

Генерація виконується один раз, і після цього сертифікати зберігаються у відповідному каталозі згідно з файловою структурою репозиторію, після чого можна переходити до безпосереднього запуску контейнерів (див. рис. 2.10).

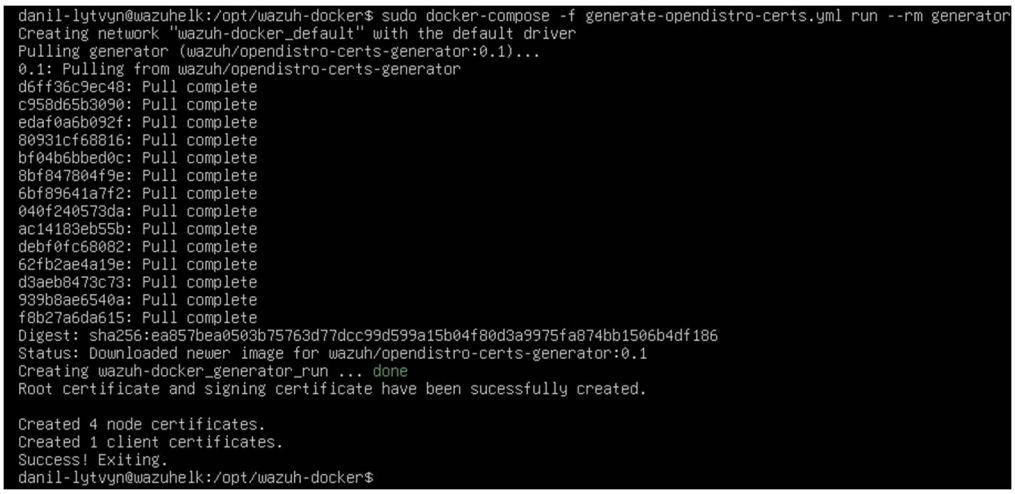


Рисунок 2.10 – Приклад генерації сертифікатів

Після генерації сертифікатів основний запуск системи відбувається за допомогою команди "docker-compose -f production-cluster.yml up -d", яка автоматично піднімає всі необхідні контейнери, пов’язані між собою через визначені у "docker-compose.yml" залежності (див. рис. 2.11).

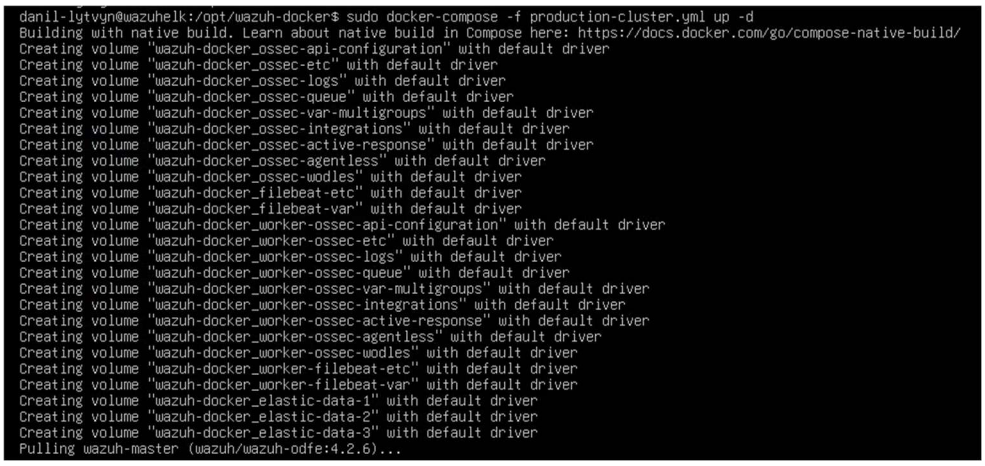


Рисунок 2.11 – Процес розгортання ELK стеку зі Wazuh

У типовому складі рішення до системи входять Wazuh Manager, Elasticsearch, Kibana та Filebeat, які разом формують повноцінну SIEM-архітектуру, придатну для збору, зберігання, обробки та візуалізації подій безпеки. Після запуску можна перевірити статус контейнерів за допомогою команди "docker ps", яка відобразить список усіх активних контейнерів із зазначенням їх поточного стану. Якщо вони позначені статусом "Up…", це свідчить про їх коректну роботу та відсутність помилок під час старту (див. рис. 2.12).



Рисунок 2.12 – Інтерфейс запуску контейнерів через Docker Compose

Далі варто перевірити, чи відкриті стандартні порти, через які здійснюється доступ до інтерфейсів та API-сервісів. Зокрема, Kibana зазвичай працює на порту ":5601" і використовується для взаємодії з користувачем у графічному середовищі, Wazuh API відкритий на порту ":55000", а Elasticsearch очікує підключення на порту ":9200". Доступність цих портів гарантує, що служби взаємодіють між собою згідно з архітектурною схемою, і дозволяє перейти до наступного етапу — налаштування системи вже безпосередньо через веб-інтерфейс (див. рис. 2.13).

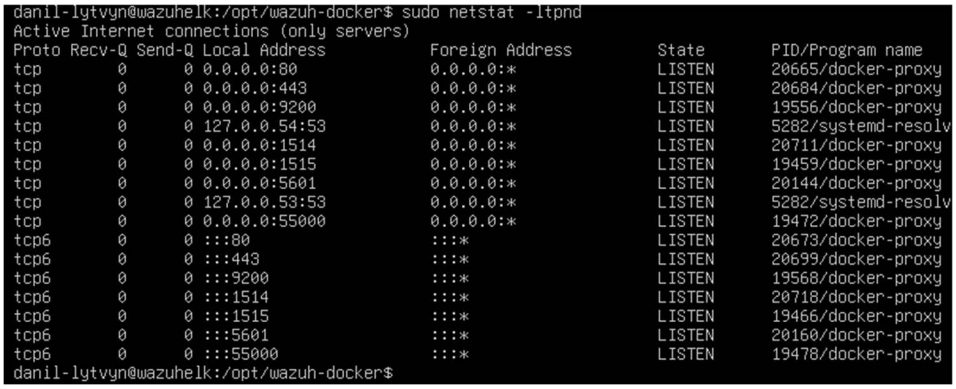


Рисунок 2.13 – Перевірка портів на конфліктність

Такий підхід дозволяє уникнути багатьох рутинних дій, пов’язаних із класичним розгортанням, і демонструє ефективність контейнеризації як способу швидкого та контрольованого запуску комплексної інфраструктури [19].

Доступ до інтерфейсу Kibana для перегляду подій. Після запуску SIEM-системи користувач отримує можливість підключатися до веб-інтерфейсу Kibana, який виконує функції основної платформи для перегляду, фільтрації та аналізу подій безпеки. Доступ до інтерфейсу здійснюється через браузер за IP-адресою сервера або віртуальної машини, на якій було розгорнуто систему, із зазначенням порту ":5601", що використовується сервісом Kibana. У налаштуваннях віртуальної машини або контейнерів можна перевірити IP-адресу, після чого для підключення достатньо відкрити адресу у форматі "https://<IP_сервера>:5601" або у локальному варіанті "https://localhost:5601" (див. рис. 2.14).



Рисунок 2.14 – Перевірка IP-адресації для роботи зі веб-інтерфейсом

Після коректного запуску контейнерів інтерфейс Kibana відображається у браузері та вимагає автентифікації користувача. За замовчуванням для доступу використовується обліковий запис із даними "admin" та "SecretPassword". Змінити ці параметри можна безпосередньо через конфігураційні файли або у налаштуваннях під час розгортання за допомогою Docker Compose, що забезпечує гнучке керування доступом без необхідності ручної модифікації налаштувань інтерфейсу (див. рис. 2.15) [18].

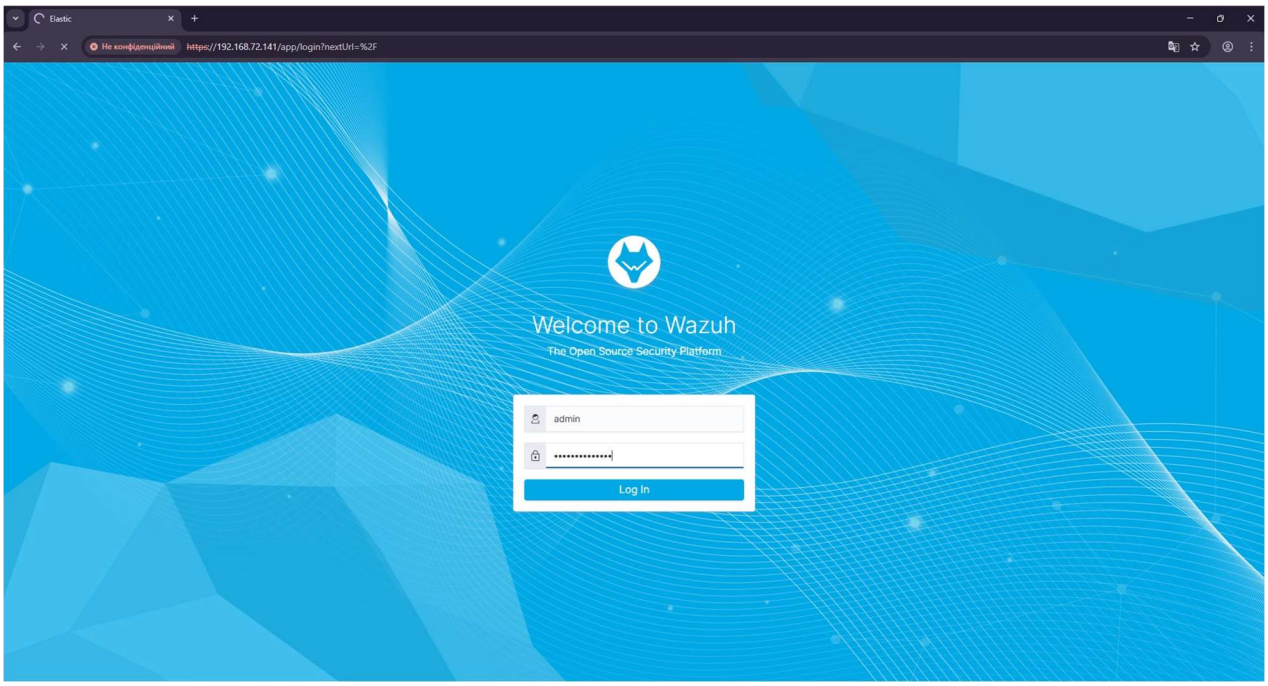


Рисунок 2.15 – Головна сторінка Kibana після розгортання

У Kibana автоматично активований плагін Wazuh, який надає спеціалізовані функції для моніторингу безпекових подій. Користувач отримує можливість переглядати списки підключених агентів та їх статуси, відстежувати спрацьовані алерти із розширеною деталізацією кожної події, контролювати зміни у файловій системі за допомогою File Integrity Monitoring, а також перевіряти відповідність політикам безпеки в рамках Compliance-модуля. Крім базових функцій Wazuh, інтерфейс Kibana забезпечує розширені можливості візуалізації, дозволяючи створювати графіки, таблиці для хронологічного аналізу подій та інтерактивні панелі для швидкого реагування на інциденти.

Таблиця 2.1 – Структура журналів подій, які обробляє Wazuh

№	Джерело журналу	Формат/Тип подій	Приклади подій	Обробка у Wazuh
1	/var/log/auth.log	Аутентифікація в Linux	Успішний/невдалий логін SSH	Модуль syslog, правило authentication
2	/var/log/apache2/access.log	HTTP-запити до вебсервера	GET /index.php?id=1' OR '1'='1	Web log analysis + правила web_attack
3	Windows Event Logs	Системні/безпекові події Windows	Події входу в систему, запуск процесів	Wazuh Windows Agent
4	/var/log/syslog	Системні повідомлення ядра та сервісів	Повідомлення про збої, перезапуски служб	Загальний лог аналіз
5	Журнали файлів FIM	Контроль змін у важливих файлах	Додано/видалено/змінено файл /etc/shadow	File Integrity Monitoring

Розгортання агентів Wazuh на різних вузлах. Для роботи з самою SIEM-системою, розгорнутою на базі Wazuh з інтеграцією через стек ELK, необхідно було також налаштувати та розгорнути агенти на декількох вузлах, зокрема агент був інстальований на Windows 11, Ubuntu Server, де безпосередньо функціонує сама SIEM-система, а також на Ubuntu Desktop, який планувалося використати як клієнт для генерації різних тестових подій і навантажень з метою перевірки функціональності системи виявлення загроз. Після початкового розгортання сама система не містить жодних подій чи журналів, оскільки не підключено жодного

джерела даних, тому першим етапом стало налаштування агента на вузлах, які будуть слугувати як джерела подій (див. рис. 2.16).

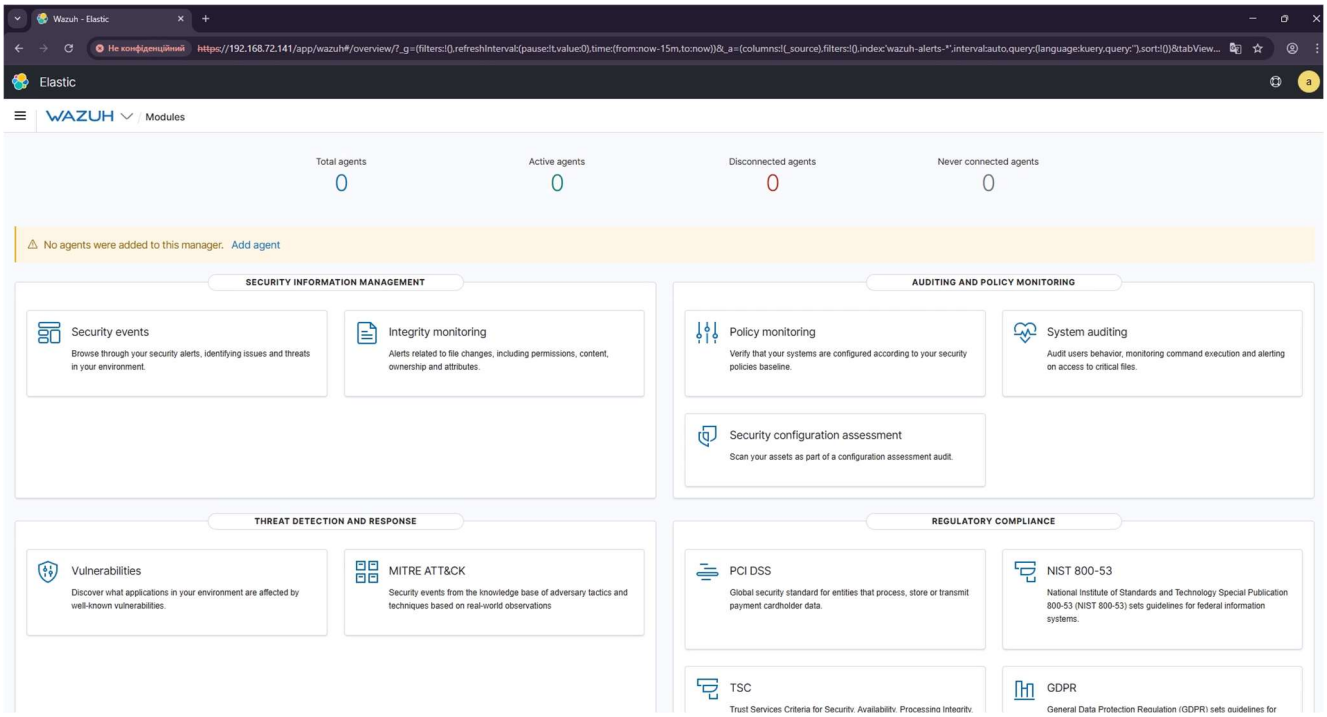


Рисунок 2.16 – Порожній веб-інтерфейс SIEM-системи після розгортання

Хоча сама система Wazuh з ELK уже містить велику базу вбудованих правил, сигнатур і алертів, які охоплюють численні категорії подій — від спроб автентифікації до мережевої активності — ці правила не потребують ручного створення, достатньо забезпечити надходження відповідних логів до центральної системи для того, щоб механізми аналізу почали працювати автоматично (див. рис. 2.17).

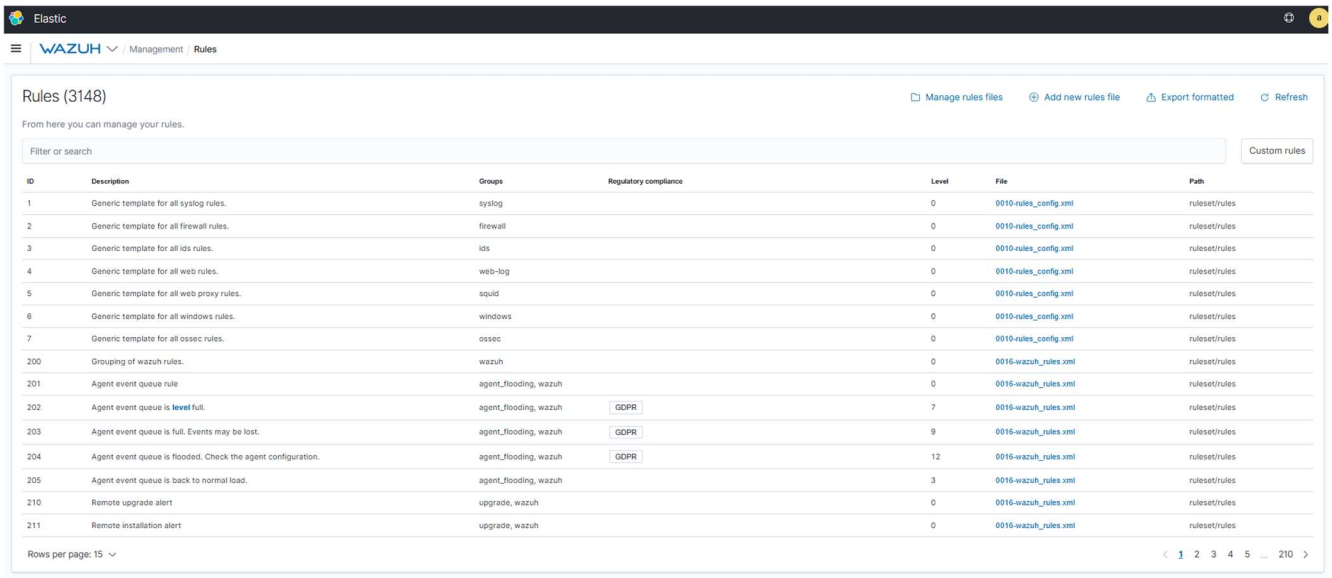


Рисунок 2.17 – Вбудована база правил Wazuh з ELK стеком

Саме з цією метою було підключено кілька агентів: першим було підключено агент на Windows 11, встановлену на ноутбуці, який запускав віртуальну машину із SIEM-системою через VMware Workstation, де за допомогою офіційного скрипта згенерованого Wazuh при додаванні нового агента було завантажено та встановлено агент на операційну систему Windows; цей агент передає події з локальної системи у SIEM, що дозволяє спостерігати за поведінкою ОС у реальному часі (див. рис. 2.18).

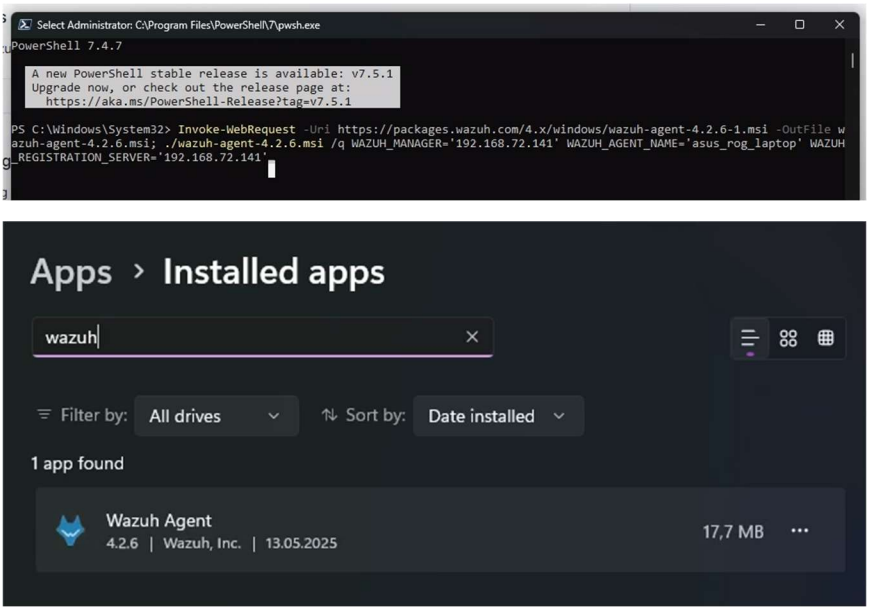


Рисунок 2.18 – Процес розгортання агента Wazuh на хост (Windows 11)

Паралельно було встановлено агент на Ubuntu Desktop, що є другою віртуальною машиною, котру використовувалося з метою ініціації різних тестових атак та навантажень, таких як brute-force на SSH-сервер або port scanning через nmap, для того щоб перевірити, чи Wazuh виявляє подібні дії та створює відповідні алерти з необхідним рівнем критичності (див. рис. 2.19).

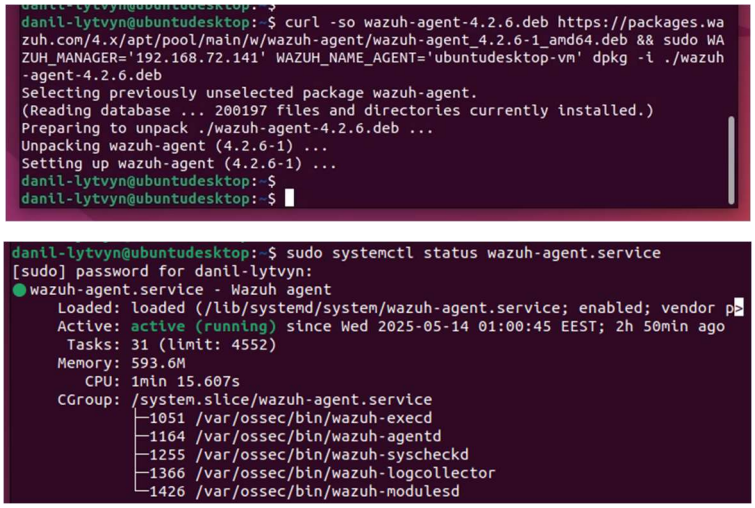


Рисунок 2.19 – Процес розгортання агента Wazuh на клієнт (Ubuntu Desktop)

Третім етапом стала інсталяція агента безпосередньо на сам сервер, де працює Ubuntu Server із встановленим Wazuh-менеджером і стеком ELK, щоб мати можливість отримувати внутрішні події самої SIEM-системи, моніторити її стан, контроль доступу, а також аналізувати трафік та активність з боку інших агентів (див. рис. 2.20).

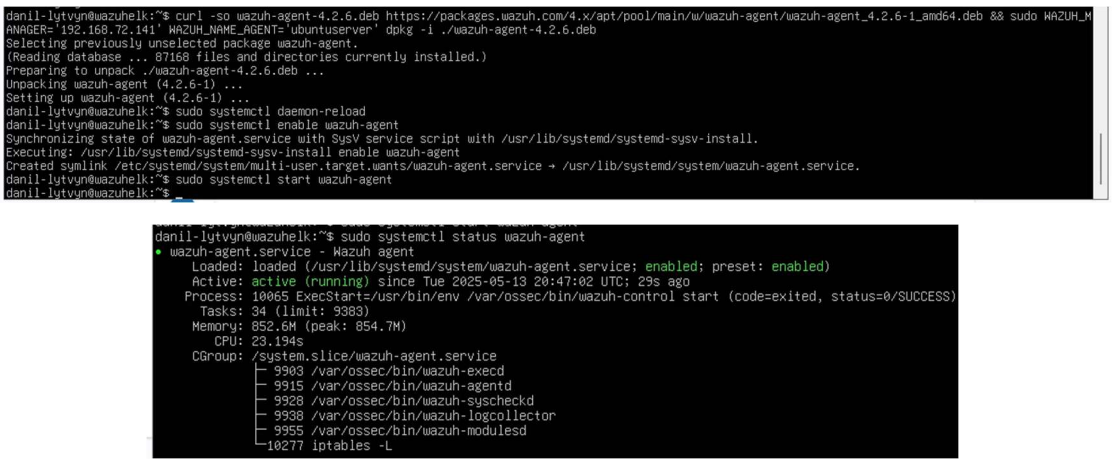


Рисунок 2.20 – Процес розгортання агента Wazuh на сервер (Ubuntu Server)

Після завершення налаштування агентів на всіх трьох вузлах, система почала приймати події, які проходили обробку згідно з наявними правилами, демонструючи повну працездатність усіх компонентів та коректне відображення інформації в інтерфейсі Kibana, що підтверджує правильну інтеграцію і можливість подальшого тестування детекції загроз (див. рис. 2.21-2.23).

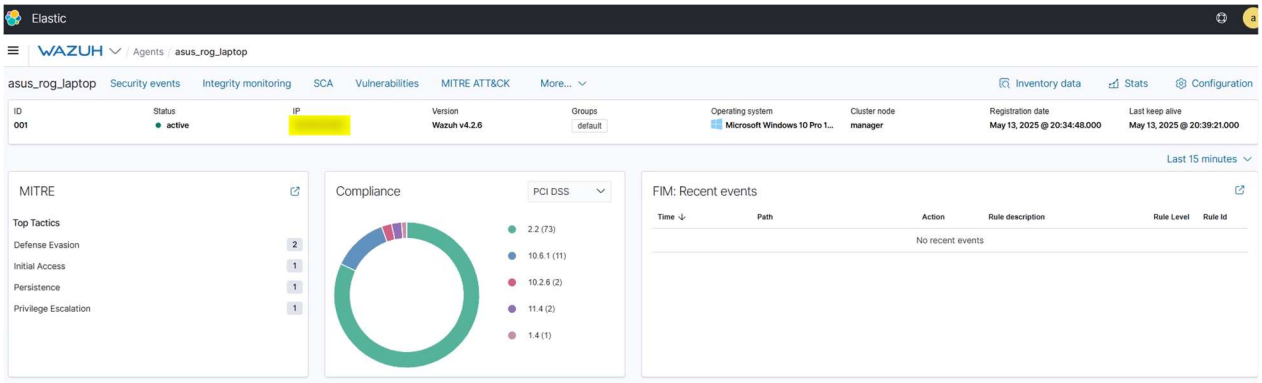


Рисунок 2.21 – Початок збору даних зі вузла Windows 11 (клієнт)

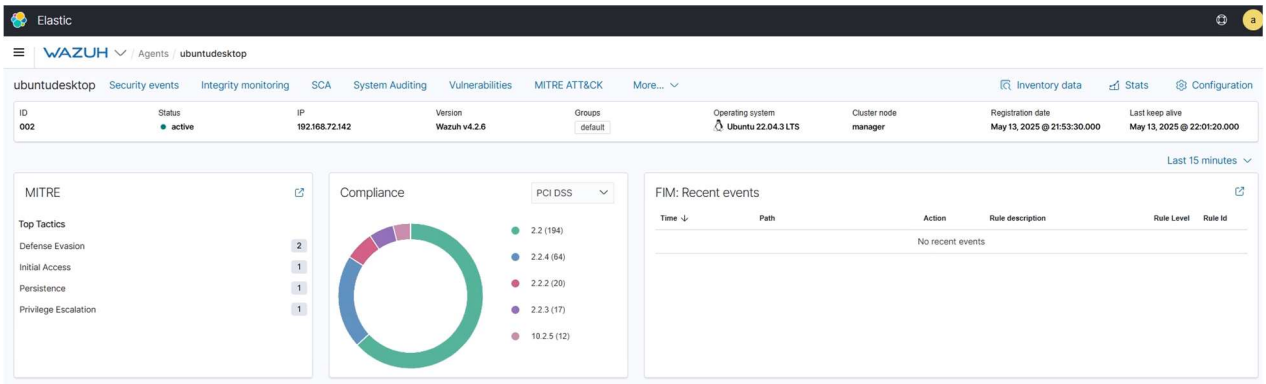


Рисунок 2.22 – Початок збору даних зі вузла Ubuntu Desktop (клієнт)

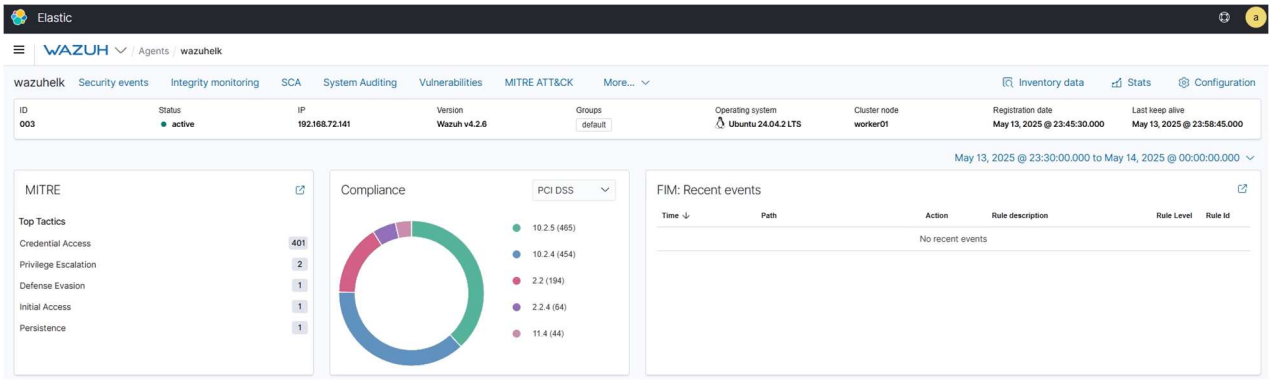


Рисунок 2.23 – Початок збору даних зі вузла Ubuntu Server (сервер)

У результаті побудова SIEM-системи на базі Wazuh та стека ELK із використанням контейнеризації Docker забезпечує швидке та ефективне отримання повнофункціонального середовища для моніторингу мережевого трафіку та подій безпеки. Завдяки наявності офіційних інструментів і репозиторіїв користувач може без значних труднощів розгорнути систему, виконати генерацію сертифікатів, запуск необхідних контейнерів та отримати доступ до Kibana для роботи з даними, а також можливість експорту даних про вузли (див. рис. 2.24).



Рисунок 2.24 – Фрагмент екпорту отриманих даних зі одного з вузлів (Ubuntu Desktop)

Такий підхід забезпечує не лише гнучкість у налаштуванні архітектури відповідно до потреб організації, а й можливість швидкого масштабування та адаптації системи у майбутньому, що важливо для підтримки високого рівня захисту інформаційної інфраструктури.

2.3 Алгоритми обробки подій та виявлення загроз

Wazuh як SIEM-рішення реалізує розгорнуту логіку обробки подій і виявлення загроз через багаторівневий аналіз логів, застосування правил детекції, кореляцію подій та ініціацію відповідних дій, що формує цілісну модель автоматизованого реагування на інциденти інформаційної безпеки. Така логіка ґрунтується на архітектурі, що підтримує масштабовану обробку великих обсягів вхідної інформації з різноманітних джерел, включно з агентами на хостах, мережевими пристроями, системами автентифікації тощо, і дозволяє в автоматичному режимі виокремлювати потенційно небезпечні активності у вигляді структурованих алертів із супровідними тегами, ідентифікаторами та описами. Завдяки цьому забезпечується не лише виявлення загроз у режимі реального часу, а й створюється основа для подальшого аналізу інцидентів, систематизації подій та формування стратегій захисту в межах єдиної системи моніторингу.

Принципи роботи правил детекції в Wazuh. Ключовим компонентом виявлення загроз є механізм правил детекції, який виконує роль логічного інтерпретатора вхідних логів, знаходячи відповідності з наперед визначеними шаблонами. Ці правила описуються у форматі XML і включають набір полів, зокрема унікальний ідентифікатор "rule id", рівень критичності "level", текстовий опис "description", умови зіставлення "match" або регулярні вирази "regex", а також категорії або групи, до яких це правило відноситься, наприклад "authentication_failures" або "ssh". Завдяки чітко структурованому формату опису правила легко модифікуються та масштабуються, що дозволяє адаптувати систему до нових загроз або змін у логіці атаки без необхідності повного перепроєктування. Ієрархічна структура дозволяє одному правилу наслідувати або уточнювати умови іншого, зберігаючи при цьому загальну логіку без дублювання коду, що підвищує ефективність налаштування складних сценаріїв виявлення.

Наприклад, правило з 'id="100001" та level="10", яке активується лише у разі, коли вже було зафіксовано повідомлення з id="18107", а текст лог-повідомлення

містить фразу "Failed password for root", дозволяє точно локалізувати спробу підбору пароля до root через SSH. У межах цього правила описується також його категорія як "brute_force", "sshd" та "authentication_failures", що спрощує фільтрацію та пошук відповідних подій в інтерфейсі аналітики (див. рис. 2.25).

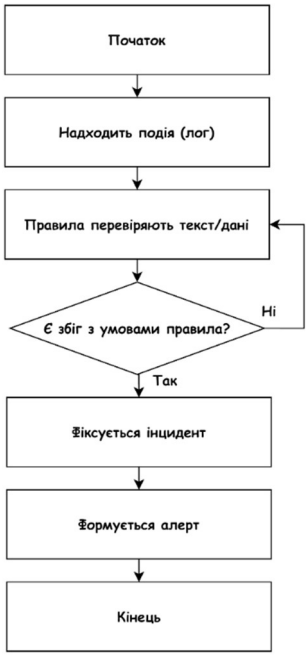
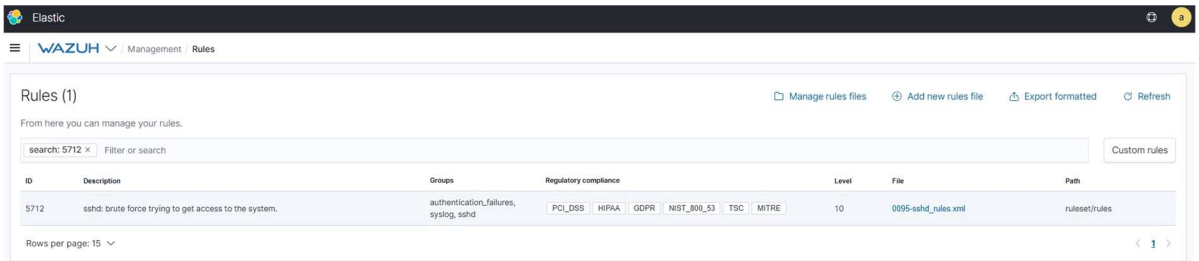


Рисунок 2.25 – Алгоритм принципу дії правила виявлення в Wazuh

Завдяки цьому система може швидко інтерпретувати низку подій як одну суттєву загрозу, визначити її критичність і активувати відповідний механізм реагування (див. рис. 2.26).




```
<rule id="100001" level="10">
  <if_sid>18107</if_sid>
  <match>Failed password for root</match>
  <description>Brute force SSH detected</description>
  <group>authentication_failures, sshd, brute_force</group>
</rule>
```

Рисунок 2.26 – Приклад правила детекції brute force атаки

Використання подібної структури дає змогу системі адаптивно реагувати на різні сценарії атак, автоматично зіставляючи події з відповідними шаблонами, і зводити інформаційне навантаження до компактного переліку актуальних інцидентів з контекстуалізованим описом, що значно підвищує зручність аналізу та дозволяє приймати обґрунтовані рішення щодо реагування або ескалації загроз.

Взаємодія з агентами – збір логів з клієнтських машин. Wazuh використовує агентську модель, у межах якої програмні агенти встановлюються безпосередньо на клієнтські машини для забезпечення безперервного збору інформації з операційної системи, її журналів та ключових процесів, що дає змогу централізовано отримувати події безпеки у режимі реального часу. Агенти виконують цілий спектр завдань, зокрема збір системних логів через стандартні шляхи, такі як "/var/log" на Linux або Windows Event Log для систем на базі Windows, а також можуть здійснювати моніторинг змін у критичних файлах завдяки механізму FIM (File Integrity Monitoring), фіксуючи кожне редагування, переміщення або видалення. Крім цього, агенти здатні виявляти потенційно небезпечні компоненти, зокрема руткіти, та аналізувати поведінку активних процесів з метою виявлення підозрілої активності, що може свідчити про експлуатацію вразливостей або несанкціоновані дії з боку користувачів чи стороннього програмного забезпечення (див. рис. 2.27).



Рисунок 2.27 – Алгоритм взаємодії агента Wazuh з клієнтом

Платформа підтримує встановлення агентів на широкий спектр операційних систем, включно з Linux-дистрибутивами (зокрема Ubuntu, CentOS, Debian та інші сумісні системи), всіма актуальними версіями Windows, а також macOS і різноманітними Unix-подібними системами, що розширює можливості інтеграції з різноманітними середовищами та дозволяє будувати єдину інфраструктуру моніторингу незалежно від гетерогенності середовища. Передача даних між агентами і сервером Wazuh здійснюється через зашифрований TCP-канал, зокрема через порти ":1514" і ":1515", що гарантує захищеність інформації у процесі обміну та мінімізує ризики перехоплення або модифікації трафіку під час транспортування даних.

Процес додавання нового агента до системи реалізується через консольне меню керування, яке доступне через утиліту "/var/ossec/bin/manage_agents", де спочатку потрібно створити запис для нового клієнта, після чого система генерує унікальний криптографічний ключ автентифікації. Цей ключ необхідно передати на клієнтську машину, де агент вже встановлений, і використати його для запуску агента та встановлення захищеного зв’язку із центральним менеджером. Така процедура забезпечує автентичність комунікації та унеможливлює підключення несанкціонованих клієнтів до SIEM-системи.

Побудова сценаріїв кореляції – rules, decoders. Для більш точного аналізу подій у будь-якій SIEM-системі необхідним є попередній етап обробки логів, що реалізується за допомогою декодерів, які виконують перетворення неструктурованих повідомлень у формат із фіксованими структурованими полями, такими як IP-адреса, дата, сервіс, ім’я користувача та інші ключові параметри, дозволяючи таким чином системі інтерпретувати вхідні дані уніфіковано та послідовно. Декодери виступають початковим фільтром, який розбиває сирі повідомлення на логічні складові, після чого відбувається співставлення отриманих структурованих значень із набором правил, визначених у системі. Ці правила, своєю чергою, перевіряють відповідність події заданим умовам, і у випадку їх виконання система ініціює генерацію алерту, який надалі може бути оброблений, візуалізований або переданий на відповідні інтегровані платформи.

Кореляція подій забезпечується комбінацією декількох механізмів, реалізованих у межах Wazuh, зокрема використанням ієрархічної побудови залежностей між подіями, що дозволяє визначати логічні зв’язки між на перший погляд непов’язаними повідомленнями. Додатково може використовуватися сценарна логіка, коли кілька окремих правил комбінуються у більш складну конструкцію, наприклад ситуація, коли декілька невдалих спроб входу з однієї IP-адреси за короткий проміжок часу розглядаються як потенційна атака типу brute force. Таймінг також виступає важливим елементом — система може враховувати частоту появи події або інтервали між пов’язаними інцидентами, формуючи таким чином динамічний аналіз поведінки об’єкта моніторингу у часі (див. рис. 2.28).



Рисунок 2.28 – Алгоритм аналізу та кореляції подій

На додачу до базових можливостей кореляції, у Wazuh реалізовано підтримку інтеграції з моделлю MITRE ATT&CK, що дозволяє додавати до кожного правила тактики й техніки з відповідної бази знань, класифікуючи події за структурованими категоріями атак. Така розширена семантична прив’язка підвищує релевантність і точність сценаріїв виявлення, дозволяючи ефективніше будувати логіку реагування на інциденти та краще інтерпретувати джерело і характер загрози. У результаті формується повноцінна система сценарного аналізу з гнучкою адаптацією до

конкретних умов, що охоплює як прості одиничні події, так і складні багатоступеневі атаки.

Реакція на загрози – сповіщення, логування, автоматизовані дії. Після ідентифікації потенційної загрози SIEM-система повинна не лише зафіксувати інцидент, але й ініціювати відповідні дії реагування в межах наявної інфраструктури безпеки, і у випадку Wazuh передбачено кілька типів таких реакцій, що можуть бути налаштовані залежно від сценарію використання. Найпоширенішим видом реагування є генерація алертів, які формуються на основі заданих правил і містять структуровану інформацію, включно з назвою правила, ідентифікатором агента, повідомленням про інцидент, категорією, рівнем загрози та іншими метаданими. Зібрані алерти зберігаються у сховищі Elasticsearch, що дозволяє їхню подальшу обробку, аналіз і візуалізацію за допомогою Kibana, формуючи зручну для аналізу панель безпеки з можливістю гнучкої фільтрації та пошуку.

Окрім статичних повідомлень, Wazuh підтримує активну взаємодію з користувачем або зовнішніми системами за допомогою механізмів сповіщення. Алерти можуть бути автоматично переадресовані через електронну пошту за протоколом SMTP, а також через Slack, Discord, HTTP вебхуки чи спеціалізовані інтеграції із зовнішніми платформами безпеки, як-от SIEM або SOAR-рішеннями, що дозволяє централізувати управління інцидентами. Такий механізм забезпечує оперативність інформування відповідальних осіб або сервісів, значно скорочуючи час до прийняття рішення щодо відповіді на інцидент.

Ще одним компонентом реагування, який підтримується у Wazuh, є механізм Active Response, що забезпечує автоматизоване виконання дій у відповідь на виявлену загрозу, прив’язану до конкретного правила. Наприклад, у разі спрацювання правила з ID="100001" може бути автоматично виконано команду на блокування IP-адреси через "iptables", завершення підозрілого процесу, деактивацію облікового запису або запуск скрипта, що реалізує більш складну логіку обробки (див. рис. 2.29).



Рисунок 2.29 – Алгоритм логіки реагування на загрози

Налаштування такого сценарію здійснюється через відповідну XML-конфігурацію, де визначається команда, область її застосування та ідентифікатор правила, на яке вона повинна реагувати (див. рис. 2.30).

```
<active-response>
  <command>disable-account</command>
  <location>all</location>
  <rules_id>100001</rules_id>
</active-response>
```

Рисунок 2.30 – Приклад сценарію активації на тригер створеного раніше правила

Механізм Active Response дозволяє реалізувати адаптивну та гнучку систему реагування, яка не лише фіксує інциденти, а й здатна автоматично протидіяти потенційним атакам у реальному часі, підвищуючи рівень захисту системи.

У підсумку архітектура обробки подій у Wazuh реалізована у вигляді багаторівневої структури, що включає:

— агентський збір даних;

— багатоступеневе декодування;

— аналіз на основі правил;

— механізми кореляції;

— ініціацію простих сповіщень;

— ініціацію складних автоматизованих дій.

Такий підхід дозволяє не лише гнучко налаштовувати поведінку системи під специфіку об’єкта моніторингу, а й забезпечувати її високу продуктивність і масштабованість у межах розгортання в різних типах інфраструктур.

2.4 Тестування та аналіз роботи побудованої системи

Після завершення розгортання системи моніторингу на основі Wazuh, Elasticsearch, Logstash та Kibana, інфраструктура була впроваджена в ізольованому середовищі з використанням контейнеризації через Docker, що дозволяє швидко масштабувати окремі її компоненти та забезпечує керованість усіх елементів. Наступним критично важливим етапом є проведення повноцінного тестування, спрямованого на перевірку працездатності механізмів виявлення загроз, точності спрацювання правил, швидкості обробки подій, стабільності функціонування при змінних навантаженнях і правильності взаємодії між модулями системи. Такий підхід дає змогу не лише підтвердити ефективність реалізованого рішення, але й виявити потенційні вузькі місця, що можуть проявитися лише під час фактичної роботи. Аналіз функціонування системи в умовах максимально наближених до реальних дозволяє глибше оцінити її відповідність практичним вимогам інформаційної безпеки [31-32].

Емуляція мережевих атак – brute-force, port scanning. У межах такого тестування здійснюється емуляція типових загроз, з якими регулярно стикаються корпоративні інформаційні системи, і першочерговою задачею є моделювання мережевих атак у контрольованому середовищі, що забезпечує безпечну перевірку реальних сценаріїв зловмисних дій. Серед таких атак насамперед перевіряється реалізація brute-force через SSH-протокол, що є надзвичайно поширеним вектором проникнення. Для створення навантаження та моделювання цієї атаки

використовується утиліта Hydra, яка дозволяє автоматизувати численні спроби входу за допомогою набору стандартних пар логін-пароль (див. рис. 2.31).

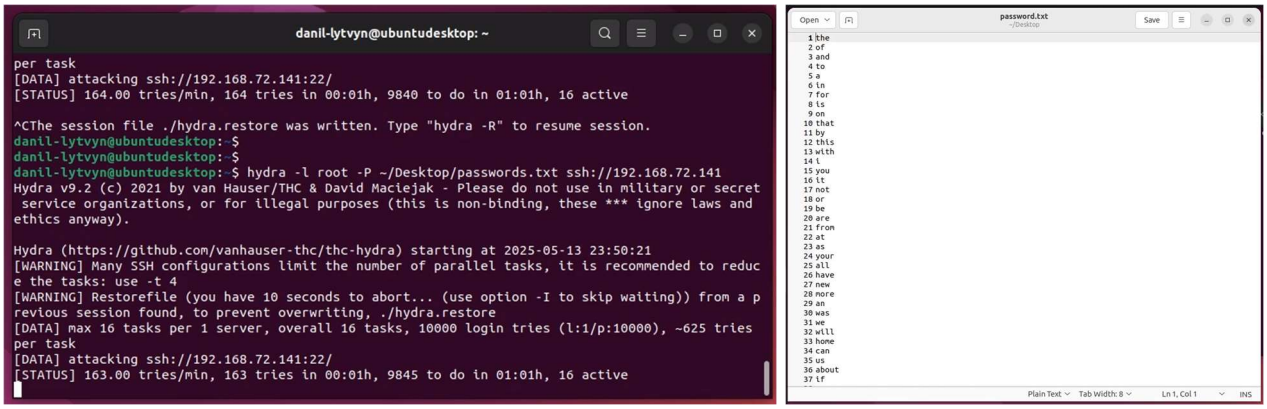


Рисунок 2.31 – Емуляція атаки типу brute-force через клієнт на сервер

Під час цього тесту очікується, що Wazuh згенерує низку алертів, які будуть згруповані, і при досягненні визначеного порогу кількості невдалих входів буде створено подію з підвищеним рівнем критичності, що свідчить про виявлення потенційного автоматизованого зламу системи (див. рис. 2.32).

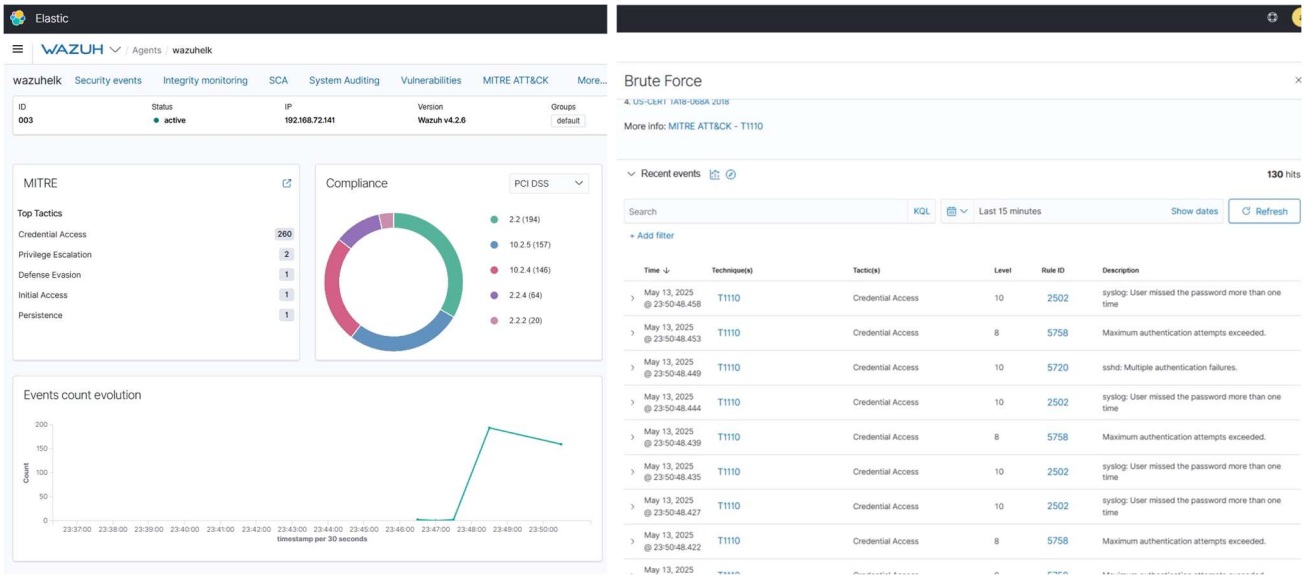


Рисунок 2.32 – Виявлення атаки типу brute-force у Wazuh

Ще одним поширеним методом перевірки є імітація порт-сканування, що використовується як етап попередньої розвідки перед цілеспрямованою атакою.

Для цього застосовується інструмент Nmap, за допомогою якого відбувається сканування IP-адреси з метою виявлення відкритих портів і активних служб. У таких випадках Wazuh здатний розпізнати аномалії в мережевому трафіку, класифікувати їх як спробу сканування, і відповідно створити алерт у групі "network scanning", що вказуватиме на потенційну загрозу, пов’язану з виявленням вразливих сервісів або відкритих точок входу до мережі (див. рис. 2.33).

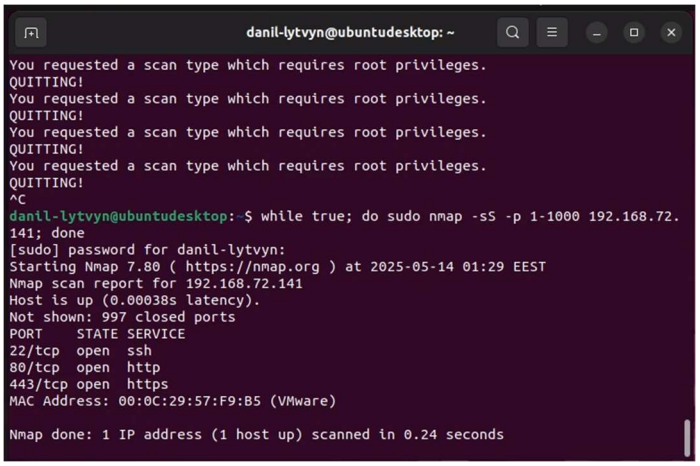


Рисунок 2.33 – Запуск короткого циклічного сценарію Nmap для імітації сканування портів через клієнт на сервер

Додатково можна проводити тестування менш очевидних, але не менш важливих типів загроз, зокрема відкриття незахищених HTTP-сервісів з публічним доступом, створення облікових записів поза встановленими політиками або запуск підозрілих скриптів, які можуть бути носіями шкідливого коду. Усі ці дії породжують події в системних логах, що автоматично перехоплюються агентами Wazuh, обробляються відповідно до правил, передаються в менеджер подій і виводяться для аналізу в Kibana. Таким чином, досягається автоматизоване реагування на широкий спектр інцидентів безпеки.

Загалом результати емуляцій свідчать про здатність побудованої системи своєчасно виявляти та класифікувати загрози, реагувати відповідно до закладених сценаріїв, забезпечувати оперативну візуалізацію даних і логів, що підтверджує її практичну придатність до використання у справжніх умовах інформаційного

середовища, а також демонструє рівень функціональності, що відповідає очікуванням від сучасної SIEM-системи.

Аналіз спрацьовування алертів у Kibana. Після проведення емуляції типових атак і генерації подій на стороні Wazuh, наступним кроком стає аналіз спрацьовування алертів у Kibana, яка виступає як основний інтерфейс аналітики, візуалізації та фільтрації зібраної інформації. Весь обсяг подій, згенерованих агентами та менеджером Wazuh, після обробки через Elasticsearch автоматично стає доступним для перегляду в Kibana, що дозволяє здійснювати гнучкий пошук та аналіз з урахуванням різноманітних параметрів. Серед основних критеріїв, за якими здійснюється фільтрація, використовуються рівень критичності подій (наприклад, "rule.level" >= "10"), назва правила, що спрацювало, група правил (наприклад, "brute_force" або "port_scanning"), а також IP-адреса джерела події. Така деталізація дозволяє швидко локалізувати інциденти, що мають підвищений ризик, а також оцінити масштаби й частотність певного типу загроз (див. рис. 2.34).

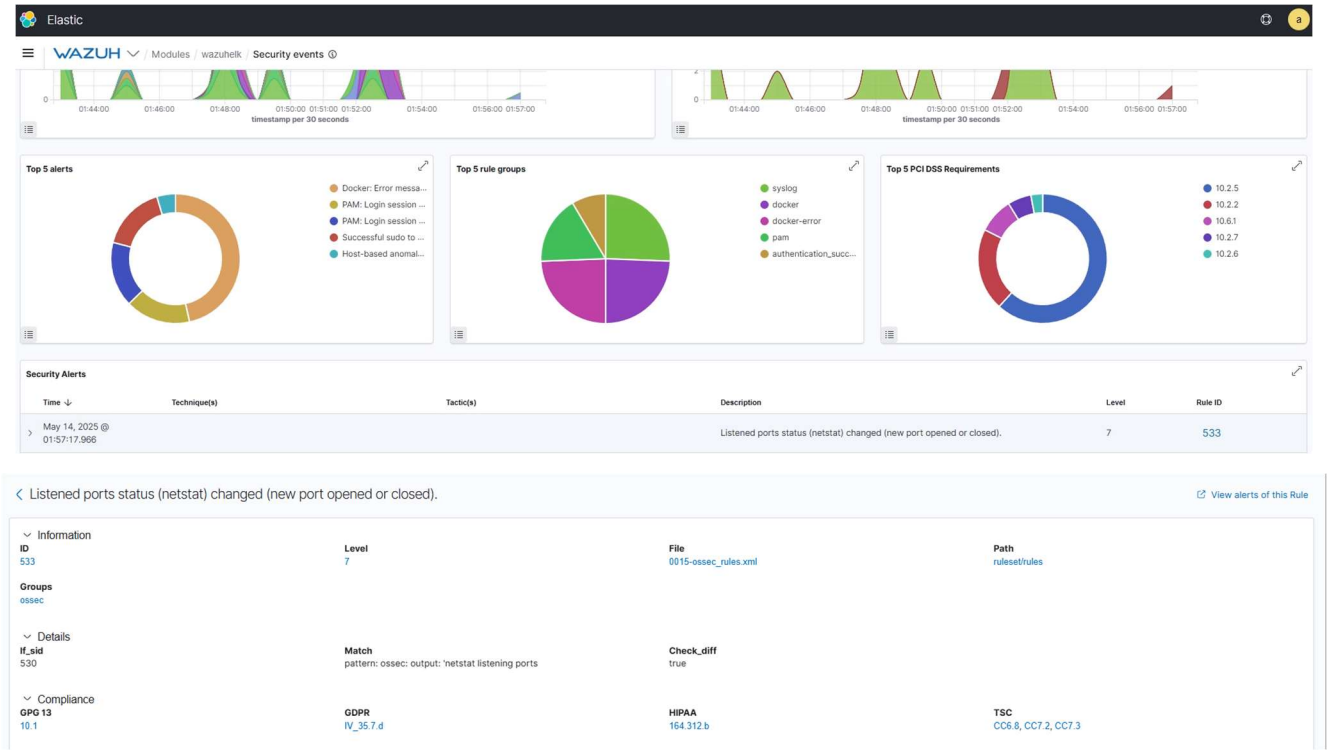


Рисунок 2.34 – Виявлення сканування портів у Wazuh

Kibana також забезпечує гнучкі можливості для побудови дашбордів, які надають інтерактивну візуалізацію даних:

— графіки частоти спрацювання алертів у часовому розрізі;

— геолокаційна карта джерел атак;

— діаграми поширених типів подій;

— найбільш активні IP-адреси, що взаємодіють із мережею.

Така візуалізація значно полегшує розуміння загальної ситуації в системі безпеки, надаючи змогу не лише відстежувати історичні тренди, але й миттєво реагувати на критичні події в реальному часі. Під час аналізу тестових атак було встановлено, що події, пов’язані з brute-force через SSH, детектуються вже після 3–5 невдалих спроб входу, що свідчить про оперативну роботу механізмів кореляції, тоді як спроби порт-сканування фіксуються протягом кількох секунд після завершення активної фази сканування за допомогою таких утиліт, як Nmap.

Кожна з подій супроводжується інформативними тегами, які додаються згідно з шаблоном правила, що спрацювало, і можуть включати категорії, такі як ID події, IP-адреса джерела, ім’я агента, а також зазначення служби, що була ціллю атаки, наприклад SSH або HTTP. За допомогою такого рівня деталізації можна створити власні правила сповіщення або здійснити перевірку вже згенерованих подій – один із таких алертів для виявлення brute-force (див. рис. 2.35).


```
{
  "rule": {
    "level": 12,
    "description": "SSH brute force attack detected",
    "id": "5712",
    "groups": ["authentication_failures", "ssh", "brute_force"]
  },
  "srcip": "192.168.1.100",
  "agent": {
    "name": "web-server"
  }
}
```


Рисунок 2.35 – Приклад створеного алерту для виявлення brute-force

Цей шаблон демонструє, яким чином задається ідентифікатор події, її опис, групи, до яких вона належить, IP-адреса джерела, а також назва агента, на якому було зафіксовано загрозу. Такий формат дозволяє легко відстежити усі релевантні

характеристики події, а Kibana, у свою чергу, дає змогу здійснити детальний перегляд усіх таких записів, порівняти їх із попередніми подіями, оцінити зміни в активності атак та зробити висновки щодо ефективності застосованих правил і фільтрів. У результаті формується цілісна картина взаємодії між компонентами SIEM-системи, в якій Kibana виступає не лише як інструмент відображення, але і як важливий елемент оцінки коректності та повноти реакції системи на інциденти інформаційної безпеки.

Продуктивність і навантаження системи у Docker. Оцінка продуктивності побудованої системи в середовищі контейнеризації Docker є важливою для розуміння її стабільності під навантаженням та здатності до масштабування в умовах зростаючої кількості подій, що надходять на обробку. Моніторинг ресурсів здійснювався за допомогою інструментів на кшталт "docker stats", що дозволило зафіксувати поточне споживання оперативної пам’яті та процесорного часу основними компонентами системи. Так, Wazuh Manager стабільно утримує обсяг споживання пам’яті в межах від 500 до 800 мегабайт, однак у моменти активного аналізу та пікового навантаження спостерігається зростання цього показника до приблизно 1,5 гігабайт. У той же час Elasticsearch виступає найбільш ресурсоємним елементом системи, із середнім використанням оперативної пам’яті в межах 1,5–2,5 гігабайт та навантаженням на CPU до 2 ядер у звичайному режимі функціонування. Kibana, як інтерфейс візуалізації та пошуку, має більш помірне споживання ресурсів, що становить близько 400 мегабайт оперативної пам’яті в типових умовах (див. рис. 2.36-2.37).

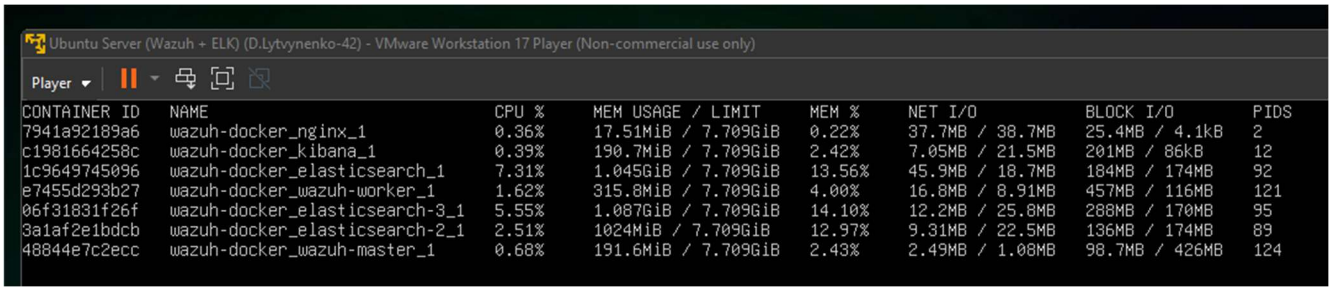


Рисунок 2.36 – Вивід інформації про навантаження системи в Docker середовищі

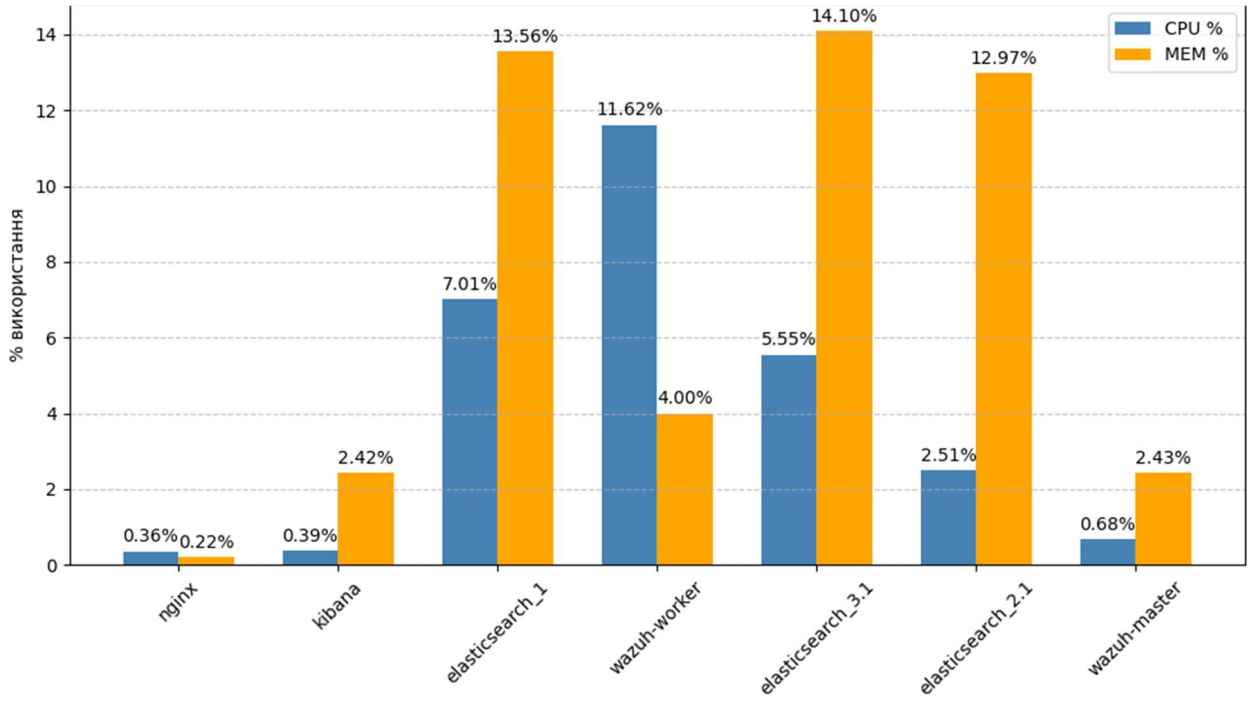


Рисунок 2.37 – Графік навантаження системи в Docker середовищі

У плані швидкодії можна зазначити, що середній час обробки однієї події системою складає приблизно одну-дві секунди, тоді як повна затримка між моментом генерації події і її появою в Kibana, включаючи всі етапи фільтрації, індексації та візуалізації, не перевищує п’яти секунд. При цьому система здатна без втрат опрацьовувати до десяти подій на секунду у фоновому режимі, що можна вважати показником стабільної роботи при типовому навантаженні. У разі навмисного створення пікового навантаження, наприклад, шляхом запуску емулятора на 500 подій протягом однієї хвилини, система продовжує функціонувати без збоїв, хоча фіксується зростання споживання процесорного часу, насамперед на стороні Elasticsearch, що підтверджує можливість масштабування архітектури завдяки гнучкості контейнерного середовища Docker, де за потреби можна додавати додаткові екземпляри агентів або навіть масштабувати окремі компоненти системи горизонтально, що забезпечить підтримку більшої кількості подій без втрати продуктивності чи затримки при генерації алертів. У межах поточної реалізації така масштабованість ще не була повністю задіяна, однак базові тести вже демонструють технічну готовність до подальшого розширення в умовах інфраструктури, що розвивається (див. табл. 2.2).

Таблиця 2.2 – Типові атаки та відповідні механізми детекції у Wazuh

№	Типова атака	Опис атаки	Приклад події (лог)	Як Wazuh виявляє?
1	Brute Force SSH	Багаторазові спроби входу через SSH	sshd: Failed password for invalid user	Правила authentication_failed, ssh
2	SQL Injection	Введення SQL-коду у форми сайту	Запис HTTP із підозрілими параметрами (' OR 1=1--)	Модуль web-log, правило sql_injection
3	Malware execution	Запуск шкідливого ПЗ	Запуск невідомого .exe у системі	Модуль FIM та VirusTotal інтеграція
4	Внесення змін до системних файлів	Спроба змінити критичні конфіги	Зміни в /etc/passwd або C:\Windows\System32	File Integrity Monitoring (FIM)
5	Аномальні підключення	Підключення до незвичних портів або адрес	Зовнішнє підключення до x.x.x.x:4444	Модуль Netstat та активні правила

Виявлені обмеження та рекомендації з налаштування. Результати тестування побудованої SIEM-системи, виконані на основі Wazuh, Elasticsearch та Kibana у контейнеризованому середовищі Docker, дозволили виявити низку технічних обмежень і практичних нюансів, які доцільно враховувати при подальшій експлуатації та масштабуванні. Одним із ключових аспектів стало те, що Elasticsearch є досить ресурсоємним компонентом, особливо під час активного зберігання великих обсягів логів навіть у контейнеризованому форматі, що вимагає ретельного підбору апаратної платформи. Окрему увагу слід приділити роботі

механізмів Active Response, які в середовищі Docker не завжди функціонують коректно без додаткових конфігурацій з боку хоста або мережевих політик контейнерів. Крім того, при зростанні обсягів проіндексованих даних спостерігається певне уповільнення завантаження інтерфейсу Kibana, що ускладнює оперативний аналіз подій через візуальні панелі.

В умовах реального розгортання системи рекомендується використовувати твердотільні накопичувачі (SSD) для зберігання індексів Elasticsearch, що дозволяє суттєво скоротити час доступу до даних, а також зменшити ризики затримок при пошуку та побудові аналітичних дашбордів. Доцільним буде обмеження обсягу логів, які надсилаються агентами, шляхом попередньої фільтрації або точного налаштування правил, що дозволить зменшити навантаження на систему зберігання та індексації. Обов’язковим є впровадження політик резервного копіювання та ротації логів, оскільки накопичення великих обсягів неактуальних записів призводить до погіршення продуктивності. У разі, якщо навантаження перевищує 1000 подій за хвилину, необхідно розгортати Wazuh у кластерному режимі для забезпечення стабільної обробки у багатокомпонентному середовищі.

Також для покращення загальної ефективності роботи доцільно винести Elasticsearch за межі Docker-інфраструктури й розмістити його на окремому сервері або у хмарному рішенні, що дозволить зменшити внутрішньоконтейнерні конфлікти та оптимізувати використання оперативної пам’яті.

Важливо також відключати непотрібні модулі, зокрема File Integrity Monitoring (FIM), на тих вузлах, де контроль змін у файловій системі не є критичним, що дозволить зменшити навантаження на агенти. Окрім цього, можливим варіантом оптимізації архітектури є створення окремого контейнера або сервера, відповідального за обробку алертів і їх інтеграцію з зовнішніми системами, наприклад, із месенджерами або платформами автоматизованого реагування, що дозволяє розділити обов’язки між компонентами та зробити систему більш гнучкою й адаптивною до зростання вимог.

РОЗДІЛ 3 ОЦІНКА ЕФЕКТИВНОСТІ ТА ПЕРСПЕКТИВИ РОЗВИТКУ

3.1 Методи підвищення ефективності роботи SIEM

Сучасні системи управління інформаційною безпекою та подіями, або SIEM-рішення, потребують не лише ретельної початкової конфігурації для адаптації до специфіки інфраструктури, але й постійного вдосконалення в процесі їхньої експлуатації. Зі зростанням обсягів зібраних журналів, ускладненням схем атак і збільшенням кількості підключених джерел даних виникає необхідність підвищення ефективності системи, зокрема через поліпшення точності виявлення загроз, зниження кількості хибнопозитивних спрацювань, автоматизацію реагування та розвиток механізмів оперативного моніторингу. Такий підхід дозволяє забезпечити збереження ефективності системи на високому рівні навіть після тривалої роботи, не допускаючи поступової втрати її здатності до якісного аналізу інцидентів. Серед основних напрямків покращення функціонування SIEM-систем виділяються декілька загальновизнаних і практично реалізованих методів, які застосовуються для адаптації до сучасного ландшафту кіберзагроз і підтримки належної продуктивності в умовах змінних навантажень [20-25].

Використання зовнішніх Threat Intelligence-джерел. Одним із ключових підходів до покращення ефективності систем є інтеграція з зовнішніми джерелами Threat Intelligence, які надають агреговані, заздалегідь зібрані відомості про загрози, що вже були ідентифіковані іншими організаціями або спільнотами. До таких даних відносяться:

- списки шкідливих IP-адрес;
- хеші шкідливих файлів;
- домени з поганою репутацією;
- індикатори компрометації;

— тактики і техніки зловмисників.

Завдяки інтеграції таких джерел SIEM-система може автоматично порівнювати вхідні події із базами відомих загроз і на основі збігів приймати рішення про створення алертів або підвищення рівня ризику, що забезпечує раннє виявлення атак, наприклад виявлення підозрілої активності, що вказує на зв’язок із C2-сервером, а також знижує навантаження на аналітиків, оскільки скорочується час на ручну обробку подій і підвищується точність класифікації інцидентів [20-21].

Серед популярних платформ Threat Intelligence, які використовуються в таких інтеграціях, можна назвати VirusTotal — сервіс для перевірки файлів і хешів на наявність шкідливих ознак, який дозволяє зіставляти локальні події з глобальними репутаційними списками. Додатково застосовуються такі сервіси, як AbuseIPDB і AlienVault OTX, що спеціалізуються на оцінці репутації IP-адрес, виявляючи ті, що пов’язані з DDoS-атаками, фішингом або іншими шкідливими активностями. У свою чергу, MITRE ATT&CK надає класифікацію тактик і технік зловмисників, що дозволяє не лише виявляти загрозу, але й зрозуміти її природу. Додаткові джерела, як Spamhaus або Emerging Threats, зосереджуються на виявленні фішингових та шкідливих доменів, які можуть використовуватися для атак через електронну пошту або інші канали.

У контексті використання платформи Wazuh для побудови SIEM-системи існує можливість інтеграції з Open Threat Exchange (OTX) від AlienVault, що дозволяє автоматично перевіряти IP-адреси, зафіксовані у подіях, на наявність у глобальних списках шкідливих об’єктів. Якщо джерело події співпадає з відомими загрозами з бази OTX, Wazuh автоматично підвищує пріоритет спрацювання або генерує окремий інцидент (див. рис. 3.1).

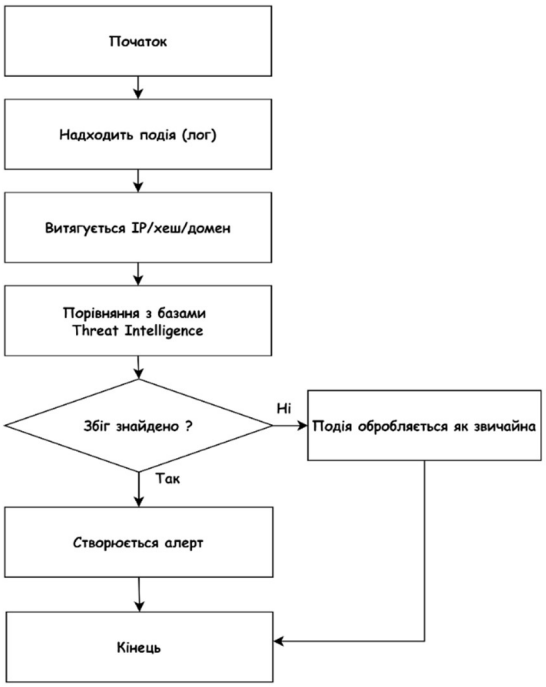


Рисунок 3.1 – Метод інтеграції зовнішніх Threat Intelligence-джерел

Така функціональність дозволяє покращити реакцію системи без потреби в ручному аналізі кожного підозрілого з'єднання, знижуючи навантаження на операторів безпеки та прискорюючи процес реагування, що критично важливо для зменшення часу, протягом якого загроза може залишатися активною в мережі [20-21].

Тонке налаштування правил виявлення – боротьба з false positives. Однією з найбільш розповсюджених і водночас критичних проблем при впровадженні SIEM-систем є велика кількість хибнопозитивних спрацювань, які створюють надлишкове інформаційне навантаження та знижують загальну ефективність реагування на реальні інциденти. Такий обсяг неактуальних або несуттєвих попереджень змушує аналітиків витрачати ресурси на перевірку подій, що не становлять загрози, ускладнюючи своєчасне виявлення дійсно небезпечної активності. Для вирішення цієї проблеми використовується підхід тонкого налаштування правил виявлення, що передбачає ретельну адаптацію логіки обробки подій відповідно до контексту інфраструктури, поведінкових моделей користувачів і особливостей конкретного середовища [22].

Зменшення кількості хибних спрацювань досягається шляхом коригування порогових значень для запуску тригерів, наприклад, визначення певної кількості невдалих спроб автентифікації як допустимої активності, тоді як перевищення цього порогу вже класифікується як підозріле. Такі пороги можуть враховувати кількість подій у певному часовому вікні, поведінку одного й того ж джерела або цілі, а також географічні чи часові фактори. Важливою практикою також є деталізація та уточнення умов спрацювання правил за допомогою виключень або обмежень, наприклад, фільтрація подій від внутрішніх IP-адрес або від перевірених облікових записів, для яких подібна активність є звичайною. Крім того, додається логіка контекстної обробки, яка враховує роль користувача, його права, тип доступу, час активності, а також геолокаційні показники, що дозволяє гнучко реагувати на зміну поведінки та зменшити ймовірність хибної тривоги (див. рис. 3.2) [21-22].

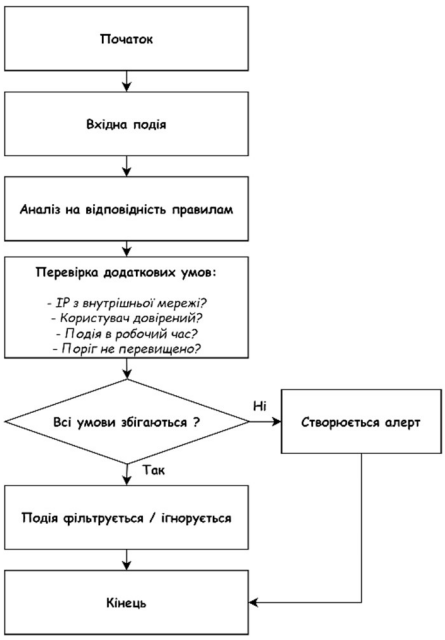


Рисунок 3.2 – Метод фільтрації помилкових спрацьовувань (false positives)

Ще одним аспектом є поділ сповіщень за критичністю, який дозволяє зосередити увагу аналітика на найбільш небезпечних подіях і водночас не ігнорувати менш суттєві сигнали, що накопичуються у фоновому режимі. У разі відсутності такої ієрархії сповіщень виникає алергія на попередження — стан, за

якого надмірна кількість тривог призводить до їх ігнорування, навіть коли серед них з’являються справді небезпечні [22].

У рамках системи Wazuh для реалізації цього підходу доступні кілька інструментів, зокрема декодери, що відповідають за обробку та інтерпретацію формату журналів із різних джерел, які можна адаптувати до специфіки конкретних сервісів або програм. Окрім цього, застосовується механізм "rules" — XML-файли, що містять логіку виявлення загроз, яку можна редагувати, створюючи власні сценарії в "local_rules.xml" на основі системних правил, що дозволяє розширювати чи звужувати рамки спрацювань без втрати сумісності з основною логікою системи. Також використовується практика створення списків довіри — "whitelisting", коли відомі надійні домени, служби або IP-адреси виключаються з аналізу або не викликають спрацювання (див. рис. 3.3).

```
<rule id="100010" level="10">
  <if_sid>5710</if_sid>
  <same_source_ip />
  <frequency>7</frequency>
  <timeframe>60</timeframe>
  <description>Brute force attempt detected (7 tries in 1 min)</description>
</rule>
```

Рисунок 3.3 – Приклад правила зі вказанням порогу для змешнення хибних спрацювань (false positives)

Одним із прикладів точного налаштування правила є конфігурація, яка визначає підозрілу активність на основі частоти спроб входу з одного джерела у певний часовий інтервал. Зокрема, правило з ідентифікатором 100010 у форматі XML описує ситуацію, коли сім спроб авторизації від одного джерела протягом однієї хвилини вважаються ознакою атаки типу brute force, після чого генерується подія з підвищеним рівнем критичності. Такий підхід дозволяє одночасно мінімізувати кількість фальшивих тривог і не втратити дійсно значущі індикатори потенційної компрометації, створюючи більш надійне, адаптивне і чутливе до контексту середовище для моніторингу безпеки.

Автоматизація реакції на загрози через Wazuh Active Response. У багатьох сценаріях забезпечення безпеки інформаційних систем своєчасна автоматична

відповідь на потенційну загрозу здатна зупинити атаку ще до того, як вона перейде в активну фазу або призведе до незворотних наслідків. У SIEM-системах, що підтримують механізм Active Response, реалізується можливість автономного запуску дій у відповідь на конкретні події без втручання людини, що значно підвищує швидкість реагування і дозволяє ефективно стримувати шкідливу активність. У цьому контексті Wazuh забезпечує досить гнучкий набір можливостей, які можуть бути використані в автоматизованому режимі, серед яких — блокування IP-адрес джерел атаки через "iptables" або інші системи фільтрації, переміщення чи повне видалення шкідливих файлів, вимкнення користувачів із системи та навіть запуск заздалегідь підготовлених сценаріїв реагування, які можуть містити складні логіки дій у випадку спрацювання конкретного правила [24].

Запуск реакції базується на поєднанні умов, де однією з основних є рівень небезпеки, який прив’язується до події. Наприклад, конфігурація може передбачати, що у разі фіксації інциденту з рівнем критичності 10 і вище, система автоматично виконає відповідну дію згідно з параметрами, заданими в конфігураційному файлі "ossec.conf" (див. рис. 3.4).

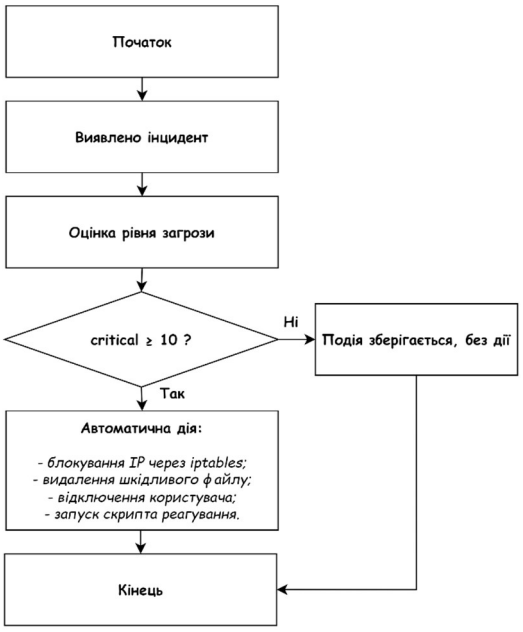


Рисунок 3.4 – Метод автоматизації відповіді на інциденти через Wazuh Active Response

Одним із прикладів такої конфігурації є наступний блок, у якому визначається, що при досягненні встановленого рівня спрацювання система повинна виконати команду "firewalldrop", що спрямована на блокування доступу з певного джерела на всіх вузлах (див. рис. 3.5).

```
<active-response>
  <command>firewalldrop</command>
  <location>all</location>
  <level>10</level>
</active-response>
```

Рисунок 3.5 – Приклад конфігурації для автоматизації реакції на загрози (через Wazuh Active Response)

Використання подібного механізму має низку беззаперечних переваг, насамперед це суттєве зменшення часу між виявленням інциденту та дією у відповідь, що мінімізує потенційні наслідки для системи або організації загалом, що дозволяє знизити навантаження на аналітиків безпеки, які можуть зосередитися на складніших випадках, що потребують ручного втручання або розслідування. Крім того, активна відповідь автоматизує рутинні процеси, зменшуючи ймовірність людських помилок у стресових ситуаціях або при великій кількості паралельних інцидентів.

Разом із цим, слід враховувати і потенційні ризики впровадження автоматичного реагування, зокрема можливість помилкових спрацювань, які можуть призвести до блокування легітимного користувача або процесу. У зв’язку з цим важливо здійснювати ретельне попереднє тестування усіх відповідей, налаштовувати детальну логіку умов активації дій і при необхідності вводити додаткові фільтри або підтвердження, особливо у критичних сегментах інфраструктури. Правильно налаштована система Active Response у Wazuh здатна стати ефективним засобом первинного стримування кіберзагроз, який функціонує в автономному режимі та підвищує загальний рівень захищеності середовища без залучення додаткових ресурсів.

Побудова дашбордів для швидкого моніторингу. Швидка реакція на інциденти можлива лише за умови, що аналітик має чітке й візуально доступне уявлення про поточний стан системи безпеки, для чого інтерфейс Kibana забезпечує можливість створення індивідуалізованих дашбордів, які відображають критичні події, інциденти та динаміку змін у загрозах. Серед найкорисніших елементів таких дашбордів слід відзначити, зокрема, топ-10 атак за типом, наприклад bruteforce чи scanning, перелік IP-адрес із найвищою активністю, кількість спрацювань за рівнями критичності, географічне розташування джерел загроз за допомогою geoIP, а також часові графіки активності, що дозволяють оперативно відслідковувати тренди [25].

Прикладами ефективних і поширених шаблонів дашбордів можна вважати Security Overview — оглядовий стан безпеки системи загалом, Authentication Failures — події, пов’язані зі спробами входу в систему, включно з підбором паролів, FIM (File Integrity Monitoring) — моніторинг змін у важливих файлах системи, а також стрічку реального часу з актуальними алертами, яка дає змогу стежити за подіями без затримки. Побудова таких дашбордів здійснюється за допомогою стандартних інструментів Kibana, таких як Visual Library (побудова графіків, кругових діаграм, heat maps), Save Searches (попередньо збережені фільтри для швидкого доступу до необхідної інформації) та Canvas (інструмент для створення брендованих інфографік, які можуть використовуватись у презентаційних цілях або для внутрішньої звітності) (див. рис. 3.6) [25].

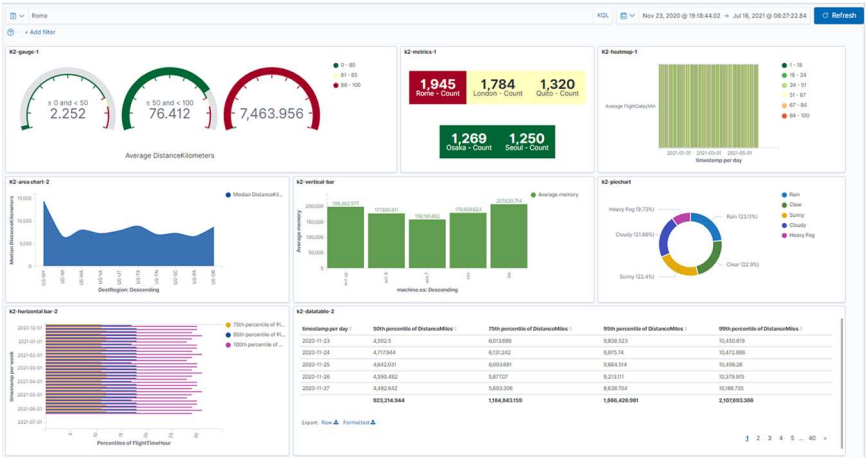


Рисунок 3.6 – Дашборд моніторингу безпеки у Kibana

Серед ключових переваг використання дашбордів варто відзначити зменшення часу, необхідного на первинний аналіз ситуації, що особливо важливо у випадку великого потоку подій, а також можливість представлення інформації у зручній формі для керівництва та швидке виявлення аномалій у поведінці користувачів або системи в цілому. Такий підхід сприяє не лише підвищенню ефективності реагування, а й створює основу для стратегічного планування заходів безпеки на основі реальних даних.

Таблиця 3.1 – Основні методи підвищення ефективності роботи SIEM-систем

№	Метод підвищення ефективності	Суть методу	Приклади / Інструменти в Wazuh	Очікуваний ефект
1	Інтеграція з Threat Intelligence	Використання зовнішніх баз IoC для перевірки логів	OTX, VirusTotal, AbuseIPDB	Виявлення загроз на ранній стадії
2	Тонке налаштування правил	Адаптація порогів, фільтрація trusted-даних, зменшення false positives	local_rules.xml, decoders, whitelist	Зменшення хибних спрацювань, фокус на реальні загрози
3	Автоматична відповідь на загрози	Виконання дій при спрацюванні алертів високого рівня	Active Response: iptables, disable-user, custom scripts	Швидке блокування атаки, мінімізація наслідків
4	Побудова дашбордів у Kibana	Візуалізація подій, трендів, розподілу атак для зручного моніторингу	Dashboards: Security Overview, GeoIP, Alerts Feed	Оперативне реагування, зручність аналізу

У цьому контексті вся система підвищення ефективності роботи SIEM має розглядатися як безперервний процес, що не обмежується лише початковою конфігурацією, а є результатом постійного вдосконалення, яке відбувається в процесі накопичення практичного досвіду, аналізу реальних загроз та поступового налаштування відповідно до специфіки середовища (див. табл. 3.1). Інтеграція зовнішніх джерел загроз значно покращує якість виявлення потенційних атак, зменшення кількості хибнопозитивних спрацювань дозволяє зосередити увагу аналітика на дійсно важливих інцидентах, автоматизовані відповіді забезпечують своєчасне стримування загроз, а візуалізація даних у вигляді дашбордів забезпечує швидке реагування і зручність контролю в реальному часі. Усі ці елементи у сукупності формують адаптивний і динамічний захисний контур, здатний реагувати на сучасні виклики у сфері інформаційної безпеки незалежно від їх складності чи природи.

3.2 Порівняння Wazuh з іншими SIEM-системами

Для вибору SIEM-системи як основи побудови ефективного кіберзахисту організації критично важливо враховувати не лише загальні можливості рішень, а й їхню відповідність наявним ресурсам, потребам масштабування, гнучкості налаштувань, підтримки та інших параметрів, особливо в умовах обмеженого бюджету. Сучасний ринок пропонує як комерційні, так і open-source рішення, що можуть бути обрані в залежності від технічних вимог, кадрової підготовки та фінансових можливостей організації, і порівняння таких систем, як Wazuh, Splunk Enterprise Security і OSSIM дозволяє краще орієнтуватися у виборі найоптимальнішого варіанту (див. табл. 3.2) [26-27].

Таблиця 3.2 – Порівняння Wazuh з іншими SIEM-системами

Критерій	**Wazuh**	**Splunk Enterprise Security**	**OSSIM**
Ліцензія	**Open-source (безкоштовно)**	Комерційна (від $2000/місяць)	Open-source
Ресурсоємність	**Середня**	Висока	Висока
Налаштування та гнучкість	**Висока, YAML/XML-конфігурації**	Дуже висока, через GUI/API	Обмежена гнучкість
Підтримка	**Спільнота + платна підтримка**	Професійна підтримка від вендора	Обмежена спільнотою
Масштабованість	**Висока (через Docker/Kubernetes)**	Відмінна (enterprise-рівень)	Середня
Інтерфейс користувача	**Через Kibana**	Власна аналітична консоль	Старий інтерфейс
Підтримка інтеграцій	**MITRE ATT&CK, VirusTotal, OTX**	Понад 1000 інтеграцій	Базові інтеграції
Автоматична реакція	**Active Response (bash/python)**	SOAR-модулі (автоматизована дія)	Часткова реалізація
Початкове розгортання	**Від 30 хв (Docker Compose)**	Тривалий, складний процес	Складне, потребує ручного налаштування
Актуальність	**Активна розробка**	Постійна розробка	Підтримка скорочується

Розглядаючи критерій ліцензування, Wazuh і OSSIM пропонуються безкоштовно як open-source платформи, тоді як Splunk Enterprise Security — це виключно комерційне рішення з вартістю, яка може перевищувати 2000 доларів на місяць, що автоматично ускладнює його впровадження в малих та середніх організаціях. З погляду ресурсоємності Wazuh демонструє середній рівень споживання системних ресурсів, тоді як Splunk і OSSIM вимагають значно потужнішої інфраструктури, що може стати бар'єром для впровадження у менш потужних середовищах [26].

Якщо звернутися до налаштувань і гнучкості, Wazuh забезпечує досить широкі можливості завдяки YAML/XML-конфігураціям, що дозволяє адаптувати систему до специфіки інфраструктури, Splunk забезпечує ще більшу гнучкість через наявність інтуїтивного GUI та API, тоді як OSSIM у цьому плані поступається, маючи лише базову гнучкість. У контексті технічної підтримки Wazuh має сильну спільноту користувачів, а також доступну комерційну підтримку, хоча більшість завдань можна вирішити саме завдяки активності ентузіастів. Splunk у цьому аспекті має перевагу за рахунок офіційної технічної підтримки від виробника, яка відповідає високим вимогам корпоративного рівня, у той час як OSSIM має лише обмежену підтримку, що базується на менш активній спільноті.

Щодо масштабованості, Wazuh легко інтегрується у середовище Docker та Kubernetes, що дає змогу гнучко нарощувати обчислювальні потужності, аналогічно до цього Splunk також добре масштабується, хоча через свій комерційний характер потребує більше ресурсів, а OSSIM має лише середній потенціал масштабування, через обмежену підтримку контейнеризації та складність конфігурації. Інтерфейс користувача у Wazuh реалізовано через Kibana, що забезпечує гнучку та інтуїтивну візуалізацію даних, Splunk має власну аналітичну консоль, зручну для поглибленого аналізу, тоді як інтерфейс OSSIM виглядає застарілим, обмеженим лише базовим функціоналом перегляду подій.

У контексті інтеграцій Wazuh підтримує такі джерела, як MITRE ATT&CK, VirusTotal, OTX та інші, що дозволяє розширювати аналітичні можливості системи,

тоді як Splunk має понад 1000 доступних інтеграцій завдяки своїй екосистемі, і це дає перевагу у складних інфраструктурах, у той час як OSSIM пропонує лише базову інтеграцію без можливості повноцінного розширення. Реакція на інциденти у Wazuh реалізується за допомогою Active Response (bash або python-скрипти), у Splunk це SOAR-модулі, що дозволяють будувати повністю автоматизовані сценарії реагування, тоді як у OSSIM автоматизація реалізована частково та вимагає додаткових зусиль для досягнення порівняної ефективності.

Щодо початкового розгортання, Wazuh можна встановити менш ніж за годину за допомогою Docker Compose, що робить його привабливим для швидкого пілотного впровадження, у той час як Splunk потребує значно більше часу та ресурсів через складну архітектуру, а OSSIM вимагає ручної конфігурації, що ускладнює початкове налаштування. У плані актуальності Wazuh перебуває в активній розробці, регулярно оновлюється та покращується відповідно до нових викликів у кібербезпеці, Splunk постійно підтримується та розвивається завдяки своїй комерційній природі, а OSSIM, навпаки, демонструє зменшення активності з боку розробників і спільноти, що ставить під сумнів його перспективність у довгостроковому плані.

Навіть без використання жорстких метрик або пріоритетності окремих критеріїв видно, що Wazuh є найбільш збалансованим варіантом для організацій, які прагнуть впровадити ефективне SIEM-рішення з мінімальними витратами, не жертвуючи при цьому ключовими функціональними можливостями (див. рис. 3.7).

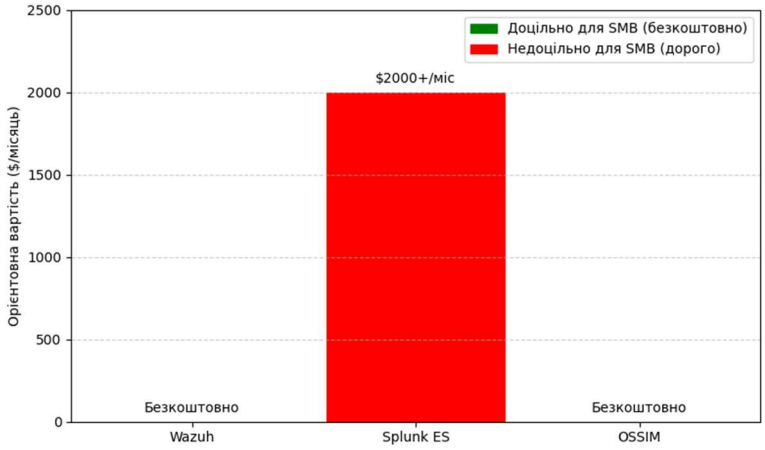


Рисунок 3.7 – Діаграма доцільності використання SIEM рішення Wazuh

Порівняння з Splunk – комерційне рішення. Щоб повноцінно оцінити переваги та обмеження Wazuh у контексті сучасних потреб організацій, доцільно здійснити безпосереднє порівняння з одним із найвідоміших комерційних рішень, зокрема зі Splunk Enterprise Security, який є прикладом потужної, високофункціональної SIEM-системи, орієнтованої на великі підприємства. Таке рішення забезпечує надзвичайно розвинену аналітику, можливості візуалізації, використання штучного інтелекту та машинного навчання, а також підтримку тисяч інтеграцій, що дозволяє охопити широкий спектр інфраструктурних сценаріїв та забезпечити глибоку автоматизацію аналізу та реагування. Завдяки цьому Splunk демонструє високу масштабованість, швидку обробку подій і високу точність детекції, однак усе це супроводжується суттєвими фінансовими витратами, які роблять його малодоступним для невеликих організацій або команд з обмеженим бюджетом [27].

На відміну від Splunk, Wazuh залишається доступним безкоштовно в рамках open-source моделі та може бути адаптованим до більшості інфраструктур завдяки підтримці YAML/XML конфігурацій і можливості швидкого розгортання з використанням Docker та інших інструментів контейнеризації. Якщо розглядати функціональність, Splunk дійсно перевершує Wazuh у галузі аналітики завдяки застосуванню вбудованих механізмів машинного навчання, однак Wazuh також дозволяє реалізовувати складні правила, кореляцію подій та гнучке створення політик безпеки, що забезпечує достатній рівень гнучкості в більшості сценаріїв використання. В аспекті інтеграцій Splunk пропонує ширший набір можливостей за рахунок підтримки тисяч плагінів та зовнішніх сервісів, однак Wazuh також має повноцінну підтримку базових та середнього рівня інтеграцій, зокрема через REST API, Log Forwarding і вбудовану взаємодію з такими сервісами як VirusTotal або MITRE ATT&CK, що робить його придатним для виконання ключових завдань з безпеки.

Фінансовий аспект залишається найсуттєвішою різницею, адже впровадження Splunk потребує щонайменше 2000 доларів на місяць, не враховуючи витрат на інфраструктуру, навчання персоналу та підтримку, що значною мірою обмежує його доступність. У цьому контексті Wazuh виглядає

значно вигідніше, адже дозволяє реалізувати комплексне рішення без необхідності закупівлі ліцензії, при цьому зберігаючи підтримку з боку активної спільноти та можливість придбання платної технічної допомоги при потребі. Таким чином, за співвідношенням ціни і функціональності Wazuh можна розглядати як раціональний вибір для організацій, де пріоритетом є практична користь і реальна функціональність, а не бренд або максимальна автоматизація на рівні корпоративних гігантів.

Порівняння з OSSIM – інше open-source рішення. Порівняння Wazuh з іншим відомим open-source рішенням, таким як OSSIM, створеним компанією AlienVault, дозволяє побачити відчутний контраст між актуальністю підтримуваних технологій і перспективністю подальшого розвитку. OSSIM був одним із перших відкритих SIEM-рішень, що набули популярності завдяки своїй доступності та набору базових можливостей аналізу безпеки, однак із часом основна увага розробників змістилася на комерційну версію — USM Anywhere, що призвело до поступового згортання активної підтримки самої OSSIM. Система почала втрачати актуальність, її архітектура залишилась монолітною, що унеможливлює ефективне масштабування, спостерігається недостатня підтримка сучасних інтеграцій, обмежена можливість автоматизації дій безпеки та низька гнучкість налаштувань, а також сповільнення оновлень і зменшення активності спільноти, що з кожним роком тільки посилює відчуття застарілості [26-27].

У порівнянні з цим Wazuh пропонує зовсім інший рівень життєздатності: система підтримує регулярні оновлення, має сучасну документацію, активну спільноту, а також тісно інтегрується з Elastic Stack, що забезпечує потужну візуалізацію та зручність для аналітиків. Завдяки підтримці Docker-контейнерів і хмарних середовищ Wazuh легко розгортається в сучасних інфраструктурах, а гнучкість у налаштуванні політик безпеки та правил детекції дозволяє адаптувати його під різні сценарії використання. Наявність Active Response, інтеграція з MITRE ATT&CK та можливість централізованого моніторингу надають цій системі відчутну перевагу в продуктивності та ефективності реагування. У результаті OSSIM практично витіснений Wazuh із більшості проєктів, де раніше

застосовувався, оскільки Wazuh не лише сучасніше, але й має більший потенціал розвитку, тому в ролі безкоштовного SIEM-рішення фактично не залишає простору для альтернатив, здатних забезпечити схожий баланс між потужністю, гнучкістю та доступністю. Більше того, Wazuh може успішно конкурувати не лише з іншими open-source продуктами, а й із деякими комерційними системами завдяки своєму широкому функціоналу та постійно оновлюваному середовищу.

Висновки щодо доцільності використання Wazuh у малих та середніх організаціях. Якщо оцінювати доцільність використання Wazuh у малих і середніх організаціях, освітніх установах або некомерційних структурах, де технічні та фінансові ресурси часто обмежені, переваги цього рішення стають особливо помітними. Безкоштовна ліцензія дозволяє уникнути витрат на програмне забезпечення, підтримка контейнеризованих розгортань значно спрощує інсталяцію навіть без глибоких знань DevOps, а можливість гнучкого налаштування дозволяє адаптувати політики безпеки до специфіки кожної організації. Вбудована система Active Response забезпечує базову автоматизацію реагування без додаткових модулів, а наявність великої кількості шаблонів, правил і документації у відкритому доступі значно прискорює навчання та впровадження. Хоча для початкового розгортання може знадобитися базовий технічний рівень підготовки, цей бар’єр легко долається завдяки доступним інструкціям, а обмеження у масштабуванні для великих інфраструктур можуть бути компенсовані використанням Kubernetes-кластерів або гібридних розгортань.

У підсумку Wazuh становить оптимальний вибір для організацій, які прагнуть отримати всі ключові можливості SIEM без значних фінансових інвестицій, пропонуючи поєднання ефективного виявлення загроз, широкої підтримки сучасних стандартів, гнучкості та стабільної підтримки спільноти. Завдяки цьому система може використовуватися як постійне рішення для моніторингу безпеки або як відправна точка для переходу на комерційні платформи в разі зростання потреб організації.

Аналіз гнучкості та інтеграційних можливостей. У сучасному кіберсередовищі важливою характеристикою SIEM-системи є не лише її

самостійна ефективність, а й здатність інтегруватися з іншими компонентами екосистеми безпеки, такими як XDR, EDR, SolarWinds, CMDB, IAM, а також з хмарними платформами та інструментами для розширеного аналізу. Гнучкість системи в цьому контексті визначається як її здатність адаптуватися до різних середовищ і джерел даних. Агенти Wazuh доступні для основних платформ, включно з Windows, Linux, macOS, а також Unix-подібними системами, що забезпечує широке охоплення середовищ корпоративної інфраструктури. Платформа підтримує інтеграцію з популярними хмарними провайдерами, такими як AWS, Azure та Google Cloud Platform, а також дозволяє використання модулів для взаємодії з такими сервісами як OTX (Open Threat Exchange), VirusTotal, MITRE ATT&CK, що розширює можливості детекції загроз на основі загальнодоступної та структурованої аналітики [26-27].

Можливість підключення користувацьких логів через Filebeat або API дозволяє збирати події з будь-яких джерел, що не мають стандартної підтримки, а використання REST API забезпечує інтеграцію з зовнішніми системами моніторингу або управління. Крім того, Wazuh може виконувати роль лог-форвардера до інших SIEM-рішень, якщо того потребує архітектура безпеки організації. Візуалізація даних реалізується через Kibana, яка підтримує інтеграцію з зовнішніми додатками і може бути кастомізована відповідно до конкретних вимог користувача, що дозволяє зберігати узгодженість з іншими інструментами візуалізації у великій інфраструктурі.

У порівнянні з іншими рішеннями видно чітку різницю: наприклад, інтеграція з AWS CloudTrail підтримується як у Wazuh, так і у Splunk, тоді як OSSIM такої підтримки не має. В аспекті MITRE ATT&CK інтеграція у Wazuh реалізована на високому рівні, у Splunk ще краще — завдяки глибокій побудові моделей на основі ATT&CK Framework, тоді як OSSIM має лише теоретичну можливість інтеграції, яка потребує ручного налаштування та великої кількості кастомізації. Щодо підтримки CMDB, у Wazuh вона офіційно не реалізована, однак можлива через нестандартні шляхи, наприклад використання проміжних API або написання скриптів, у OSSIM ситуація ще менш гнучка, оскільки підтримка

кастомної інтеграції вкрай обмежена, а у Splunk інтеграція з CMDB здійснюється напряму через плагіни, що дозволяє автоматично синхронізувати зміни та атрибути активів.

Коли мова заходить про інтеграцію з SOAR-системами, Wazuh не має прямої офіційної підтримки, проте її можна реалізувати за допомогою зовнішніх скриптів або використовуючи REST API. У OSSIM ця функціональність фактично відсутня, а у Splunk підтримка реалізована через власну платформу Phantom, яка дозволяє будувати повноцінні сценарії автоматизованого реагування. Таким чином, гнучкість Wazuh у плані інтеграцій охоплює більшість сучасних потреб і дозволяє адаптувати систему як до хмарних, так і до локальних сценаріїв із високим рівнем налаштувань, забезпечуючи її конкурентоздатність як у порівнянні з іншими open-source рішеннями, так і з частиною комерційних платформ.

3.3 Напрями вдосконалення побудованої системи

Розгортання SIEM-системи на базі Wazuh та стеку ELK є лише початковим кроком у створенні повноцінного центру моніторингу безпеки, оскільки у реальних умовах навіть базово функціональна система потребує поступового вдосконалення відповідно до динаміки розвитку кіберзагроз, змін у структурі інфраструктури, необхідності масштабування, появи нових регламентів або стандартів, а також потреб у глибшій інтеграції з іншими системами управління подіями. Зростання складності IT-інфраструктури неминуче тягне за собою потребу в удосконаленні SIEM-рішень, що проявляється як у технічних, так і в організаційних аспектах, серед яких одним із найбільш актуальних напрямів є інтеграція з хмарними платформами, яка в умовах сучасного гібридного середовища є не просто бажаною, а необхідною умовою для підтримки повноцінної безпеки [28-30].

Інтеграція з хмарними сервісами – AWS та Azure. Сучасна корпоративна інфраструктура дедалі частіше переносить свої сервіси до хмари або будується як

гібридна, тому SIEM-система має підтримувати ефективний моніторинг таких середовищ, що зумовлює необхідність інтеграції з провідними хмарними платформами, такими як AWS та Microsoft Azure. Wazuh у цьому контексті забезпечує підтримку через спеціалізовані модулі, які дозволяють збирати події з AWS CloudTrail — журналів дій користувачів і сервісів, AWS GuardDuty — механізму виявлення аномалій та вторгнень, або з Azure Activity Logs та Azure Security Center, які фіксують зміни й події в середовищі Microsoft та забезпечують аудит безпеки на рівні ресурсів. Обробка таких подій відбувається у форматі JSON через внутрішній механізм Cloud Integration Modules, який забезпечує запити до відповідних API хмарних сервісів, що дозволяє досягти автоматизованого збору даних із хмарних систем, при цьому підтримуючи централізований підхід до управління інцидентами (див. рис. 3.8) [28].

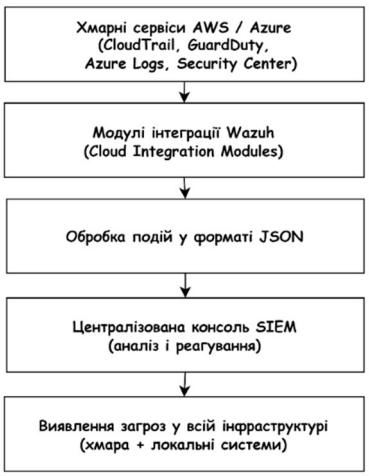


Рисунок 3.8 – Інтеграція SIEM з хмарними платформами

Результатом такого підходу є можливість отримання повної видимості над усією інфраструктурою організації незалежно від її розташування, що дозволяє виявляти підозрілі активності як у хмарних середовищах — таких як віртуальні машини, бази даних, мережеві шлюзи — так і в локальних системах, зводячи ці події до єдиного інформаційного простору. Такий підхід також забезпечує консистентність політик безпеки, зменшує фрагментацію даних і дозволяє гнучко реагувати на загрози, що виникають у різних середовищах одночасно.

Використання одного уніфікованого інтерфейсу для моніторингу, аналізу та реагування на інциденти забезпечує не лише оперативність у виявленні, а й аналітичну узгодженість, знижуючи ймовірність помилкових рішень або втрати критичної інформації внаслідок розрізненості джерел даних [28].

Підключення до CMDB та SOAR-платформ. Для досягнення більш глибокого рівня автоматизації реагування на інциденти та збагачення даних про події безпеки контекстом із внутрішньої IT-інфраструктури доцільним є підключення SIEM-рішень до систем типу CMDB (Configuration Management Database), а також до SOAR-платформ (Security Orchestration, Automation and Response), що дає змогу напряму пов’язувати події з реальними об’єктами середовища — такими як сервери, користувачі, додатки або інші компоненти, включені до конфігураційної бази. Перевага такої інтеграції полягає у можливості контекстуалізації інцидентів, що означає, наприклад, автоматичне розпізнавання критичності події на основі її зв’язку з конкретною бізнес-системою або активом, наприклад, якщо вторгнення або аномалія виявлені у зоні, пов’язаній із критичною базою даних або сервером облікових записів. Інтеграція з CMDB може бути реалізована як через безпосередні API-запити або через Elasticsearch, який забезпечує локальне поєднання івентів з відповідними конфігураційними записами та дозволяє формувати повний контекст події на основі структурованої інформації про актив (див. рис. 3.9) [28-29].

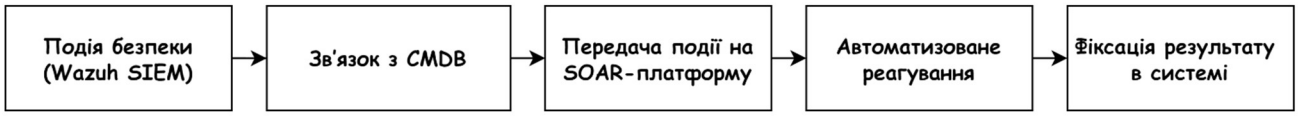


Рисунок 3.9 – Інтеграція SIEM з CMDB та SOAR

У разі інтеграції з SOAR-платформами, такими як The Hive, Shuffle, Cortex XSOAR або аналогічними інструментами, відкриваються можливості для повної автоматизації процесів реагування, де обробка інциденту може відбуватись за попередньо визначеним сценарієм — від класифікації до реалізації заходів впливу. Наприклад, при виявленні підозрілої активності можна автоматично створювати

інцидент у системі ServiceNow, надсилати сповіщення електронною поштою відповідальній особі, блокувати IP-адресу через інтеграцію з мережевим міжмережевим екраном або запускати додаткові скрипти для збору інформації. Така зв’язка реалізується шляхом передачі повідомлень з Wazuh до відповідних платформ через "webhook", "syslog" або "file output", що дозволяє обробляти події у реальному часі.

Ілюстрацією практичного сценарію є випадок, коли система фіксує вхід до облікового запису з нетипової IP-адреси, подія передається Wazuh, який через заздалегідь налаштований механізм інтеграції направляє інформацію на SOAR-платформу, де, відповідно до політик безпеки, виконується блокування IP-адреси на firewall, а також створюється запис про інцидент у системі керування сервісами. Така автоматизована реакція дозволяє не лише зекономити час, а й значно знизити ризики, пов’язані з людським фактором або затримками в обробці загроз, що особливо критично в умовах динамічного середовища сучасної кібербезпеки [28-29].

Автоматичне оновлення правил безпеки. Функціональна ефективність систем моніторингу інформаційної безпеки значною мірою залежить від своєчасного оновлення правил детекції загроз, оскільки нові типи атак та шкідливих активностей з’являються регулярно, а відповідно правила, які не оновлювались, швидко втрачають свою актуальність. Ручне оновлення наборів правил є не лише трудомістким, а й потенційно помилковим процесом, який знижує рівень захищеності та створює ризики для стабільної роботи всієї системи, тому з огляду на реалії постійної зміни кіберзагроз актуальним стає автоматизоване оновлення конфігурацій, сигнатур та відповідних файлів, зокрема правил типу YARA, Baselining, а також списків IOC (індикаторів компрометації), що надходять з джерел threat intelligence [29-30].

У цьому контексті інтеграція з такими платформами як OTX (Open Threat Exchange), MISP, VirusTotal та іншими сховищами індикаторів дозволяє системі автоматично отримувати та оновлювати інформацію про нові загрози, включаючи IP-адреси, хеші файлів, доменні імена та інші атрибути зловмисної активності.

Окрім цього, використовуються зовнішні скрипти, здебільшого написані мовою Python або Bash, які регулярно звертаються до репозиторіїв (наприклад, GitHub) для перевірки нових версій правил і при наявності змін — завантажують і впроваджують їх у середовище. Такий підхід дозволяє мінімізувати затримку між появою нової загрози і готовністю SIEM-системи до її виявлення, що критично при захисті активних середовищ.

Як приклад автоматизованого механізму можна навести базовий сценарій, що регулярно оновлює локальні правила Wazuh (див. рис. 3.10).

```
#!/bin/bash
# Сценарій для автоматичного оновлення правил
cd /var/ossec/etc/ruleset
git pull origin main
systemctl restart wazuh-manager
```

Рисунок 3.10 – Приклад сценарію автоматизації оновлення правил

Така автоматизація не тільки забезпечує актуальність даних, а й дозволяє уникати людських помилок під час ручного оновлення, однак слід враховувати, що кожне оновлення має супроводжуватись попереднім тестуванням на відповідність та працездатність, оскільки впровадження некоректного або конфліктного правила може призвести до некоректної роботи системи або повного порушення механізму обробки логів, що неприпустимо в умовах безперервного моніторингу. Для цього застосовується linking-модуль або система попередньої перевірки конфігурації, що дозволяє оцінити наслідки перед внесенням змін у продуктивне середовище (див. рис. 3.11).



Рисунок 3.11 – Схема автоматичного оновлення правил

Такий підхід дозволяє зберігати баланс між швидкістю оновлення та стабільністю функціонування інфраструктури безпеки.

Розширення функціоналу через власні модулі та скрипти. Система Wazuh дозволяє значно розширювати власну функціональність без необхідності внесення змін до ядра, що подібно до механізму розширень у Linux, де новий функціонал реалізується через додаткові модулі. Така гнучкість досягається завдяки підтримці динамічних декодерів, реактивних скриптів та можливості створення власних правил обробки, що адаптовані до специфіки журналів або бізнес-логіки організації. У практичному сенсі це дозволяє, наприклад, писати власні каскадні декодери для нестандартних форматів логів, що походять з внутрішніх систем, які не підтримуються типовими шаблонами, а також створювати правила кореляції, що враховують особливості архітектури та поведінки IT-інфраструктури конкретного підприємства [30].

Значна увага приділяється й реалізації реактивних дій на підставі виявлених інцидентів, де механізм Active Response дозволяє запускати локальні або віддалені скрипти у відповідь на певні типи подій. Наприклад, у випадку виявлення

аномальної активності з боку конкретного користувача можливо активувати скрипт, який блокує обліковий запис (див. рис. 3.12).

```
<active-response>
  <command>disable-user</command>
  <location>local</location>
  <rules_id>100001</rules_id>
</active-response>
```

Рисунок 3.12 – Приклад використання власного модуля Active Response

Разом із відповідним Bash-скриптом (див. рис. 3.13).

```
#!/bin/bash
# disable-user.sh
usermod -L $1
logger "User $1 has been disabled due to suspicious activity"
```

Рисунок 3.13 – Приклад використання власного сценарію для модуля Active Response

Такий підхід дозволяє миттєво реагувати на критичні події без залучення адміністратора. Водночас подальші сценарії розширення функціоналу передбачають, наприклад, створення скриптів, які взаємодіють із зовнішніми системами — відправляють IP-адресу порушника до блок-листу міжмережевого екрану, ініціюють оновлення записів у базі конфігураційного менеджменту або створюють інцидент у системі типу Jira Service Management із усією необхідною інформацією для ескалації та подальшого аналізу.

У підсумку, власні розширення, що реалізуються через модулі, декодери та реактивні сценарії, дозволяють суттєво підвищити адаптивність SIEM-рішення до нестандартних умов, розширити можливості взаємодії з зовнішніми платформами та забезпечити більш точну й контекстуалізовану відповідь на кіберзагрози.

Перспективи розвитку та гнучкість системи. Інтеграція зазначених функціональних можливостей перетворює Wazuh із базового засобу виявлення подій на централізований компонент повноцінної екосистеми кіберзахисту, що здатний забезпечити цілісну безпеку інфраструктури. Розвиток у напрямі глибшої

аналітики логів відкриває перспективи впровадження модулів машинного навчання для виявлення аномалій поведінки на основі історичних даних подій, що зберігаються у системі. Подібні підходи дозволяють виявляти не лише відомі загрози, але й нові, які ще не мають чітко визначених сигнатур. У паралельному напрямі розглядається побудова мультитенантної архітектури, орієнтованої на провайдерів безпеки як послуги (MSSP), що передбачає централізоване управління кількома клієнтськими середовищами без втрати роздільності та контролю [28-30].

Усі ці можливості реалізуються завдяки гнучкій архітектурі Wazuh, яка базується на відкритому коді, підтримці модульного розширення, сумісності з контейнеризованим розгортанням через Docker та постійній активності спільноти користувачів і розробників, зокрема на GitHub, форумах та технічних платформах. Відкритість коду та прозорість процесів дозволяють глибоко адаптувати систему під конкретні завдання без обмежень, притаманних закритим комерційним рішенням, що в поєднанні з широкою підтримкою розробників забезпечує постійне вдосконалення платформи та розширення її можливостей. У такій конфігурації Wazuh стає конкурентоспроможною альтернативою для багатьох дорогих SIEM-рішень завдяки гнучкості, масштабованості та адаптивності до різноманітних сценаріїв безпеки.

Підсумовуючи зазначене, можна стверджувати, що побудована SIEM-система на базі Wazuh та стеку ELK відкрита до поступового вдосконалення завдяки:

— модульному підходу;
— підтримці відкритих стандартів;
— широким можливостям інтеграції;
— спроможності еволюціонувати.

Усе це відповідно до зростаючих вимог у сфері кіберзахисту. Такий підхід дозволяє не лише реалізовувати збір і аналіз логів, але й формувати повноцінну платформу, орієнтовану на реагування, автоматизацію, впровадження threat intelligence та взаємодію з SOAR- і CMDB-системами, зберігаючи при цьому повний контроль над системою в межах локальної інфраструктури [28-30].

ВИСНОВКИ

У процесі реалізації SIEM-рішення було використано платформу Wazuh у поєднанні зі стеком ELK, що дозволило побудувати ефективну систему моніторингу подій безпеки. Завдяки використанню контейнеризації на базі Docker вдалося забезпечити гнучкість, масштабованість та швидке розгортання компонентів у тестовому середовищі. Було налаштовано збір логів з клієнтських вузлів, реалізовано базові правила кореляції подій та систему сповіщення.

Проведене тестування з використанням емуляцій типових атак, таких як brute-force та портове сканування, показало ефективність виявлення загроз. Система формувала коректні оповіщення та дозволяла візуалізувати події у зручному графічному інтерфейсі Kibana. Спостерігалося помірне навантаження на ресурси, що підтверджує доцільність використання рішення в середовищах з обмеженим технічним забезпеченням.

Результати порівняльного аналізу з іншими SIEM-системами, зокрема Splunk та OSSIM, засвідчили конкурентоспроможність Wazuh як open-source рішення. Платформа забезпечує необхідну функціональність для виявлення загроз, підтримує гнучке налаштування та інтеграцію з іншими системами, що робить її придатною для малих та середніх організацій без значних фінансових вкладень.

Також було визначено можливі напрями покращення системи, зокрема інтеграцію з хмарними сервісами, системами автоматизованого реагування, CMDB, а також впровадження автоматичного оновлення правил. Розширення можливостей за допомогою власних модулів та скриптів дозволить адаптувати систему до нових викликів і підвищити рівень безпеки інформаційного середовища.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Security Information and Event Management (SIEM). Real ... - Wazuh. URL: https://wazuh.com/platform/siem/ (дата звернення: 25.02.2025)
2. What Is SIEM? | Microsoft Security. URL: https://www.microsoft.com/en-us/security/business/security-101/what-is-siem (дата звернення: 28.02.2025)
3. SIEM: Security Information & Event Management Explained - Splunk. URL: https://www.splunk.com/en_us/blog/learn/siem-security-information-event-management.html (дата звернення: 03.03.2025)
4. Security information and event management (SIEM) systems - IRS. URL: https://www.irs.gov/privacy-disclosure/security-information-and-event-management-siem-systems (дата звернення: 07.03.2025)
5. What is SIEM? How Security Information & Event Management Works - Fortinet. URL: https://www.fortinet.com/resources/cyberglossary/what-is-siem (дата звернення: 11.03.2025)
6. SIEM Options: Open-Source vs. Enterprise vs. Managed | SageNet. URL: https://www.sagenet.com/insights/siem-options-open-source-vs-enterprise-vs-managed/ (дата звернення: 15.03.2025)
7. Open Source SIEM vs. Enterprise-Level SIEM: Which Is Right for You? - Core Security. URL: https://www.coresecurity.com/blog/open-source-siem-vs-enterprise-level-siem-which-right-you (дата звернення: 19.03.2025)
8. Top SIEM Compliance Use Cases: GDPR, PCI DDS, ISO, And - Stellar Cyber. URL: https://stellarcyber.ai/learn/siem-compliance-use-cases/ (дата звернення: 23.03.2025)
9. SIEM Open Source Overview - N-able. URL: https://www.n-able.com/es/blog/ossim-open-source-siem-overview (дата звернення: 27.03.2025)
10. Best SIEM Solutions: Top 10 SIEM systems and How to Choose - Exabeam. URL: https://www.exabeam.com/explainers/siem-tools/siem-solutions/ (дата звернення: 31.03.2025)
11. NIST, GDPR, PCI-DSS, ISO 27001, CSF & FCA on Cyber Incident ... - CM Alliance. URL: https://www.cm-alliance.com/cybersecurity-blog/nist-gdpr-pci-dss-iso-27001-csf-fca-on-cyber-incident-response (дата звернення: 04.04.2025)
12. NIST SIEM requirements and standards - definition & overview - Sumo Logic. URL: https://www.sumologic.com/glossary/nist-siem-requirements-and-standards/ (дата звернення: 08.04.2025)
13. ISO 2700x, NIST, GDPR, HIPAA, CIS Controls, PCI DSS, and COBIT - LinkedIn. URL: https://www.linkedin.com/pulse/navigating-essential-grc-standards-frameworks-iso-2700x-riya-pawar-crtzc (дата звернення: 12.04.2025)
14. SIEM and threat intelligence: Stay current on trending threats - IBM. URL: https://www.ibm.com/think/insights/siem-and-threat-intelligence-stay-current-on-trending-threats (дата звернення: 16.04.2025)

15. Threat intelligence | How SIEM works? | ManageEngine Log360. URL: https://www.manageengine.com/log-management/siem/threat-intelligence-solution.html (дата звернення: 20.04.2025)
16. Wazuh - Open Source XDR. Open Source SIEM. URL: https://wazuh.com/ (дата звернення: 24.04.2025)
17. Understanding Wazuh: The Free, Open Source Security Platform for ... - Medium. URL: https://osintph.medium.com/understanding-wazuh-the-free-open-source-security-platform-for-xdr-siem-48b3c3dfba9d (дата звернення: 28.04.2025)
18. Visualizing Dashboards and ELK SIEM | by Ibrahim Ayadhi - Medium. URL: https://medium.com/%40ibrahim.ayadhi/visualizing-dashboards-and-elk-siem-c4a669f3fae8 (дата звернення: 02.05.2025)
19. Kibana: Explore, Visualize, Discover Data - Elastic. URL: https://www.elastic.co/kibana (дата звернення: 05.05.2025)
20. What is SIEM Tuning and Why Does It Matter? - BitLyft. URL: https://www.bitlyft.com/resources/what-is-siem-tuning-and-why-does-it-matter (дата звернення: 25.02.2025)
21. SIEM Alerts Best Practices: Tuning for Fatigue Reduction - RedLegg. URL: https://www.redlegg.com/blog/siem-alert (дата звернення: 28.02.2025)
22. Tune detection rules | Elastic Docs. URL: https://www.elastic.co/guide/en/security/current/tuning-detection-signals.html (дата звернення: 03.03.2025)
23. Active Response - Capabilities - Wazuh documentation. URL: https://documentation.wazuh.com/current/user-manual/capabilities/active-response/index.html (дата звернення: 07.03.2025)
24. Incident response - Use cases - Wazuh documentation. URL: https://documentation.wazuh.com/current/getting-started/use-cases/incident-response.html (дата звернення: 11.03.2025)
25. Top 25 Kibana Dashboard Examples - Logit.io. URL: https://logit.io/blog/post/the-top-kibana-dashboards-and-visualisations/ (дата звернення: 15.03.2025)
26. Open Source SIEM vs. Enterprise-Level SIEM: Which Is Right for You? - Core Security. URL: https://www.coresecurity.com/blog/open-source-siem-vs-enterprise-level-siem-which-right-you (дата звернення: 19.03.2025)
27. SIEM Options: Open-Source vs. Enterprise vs. Managed | SageNet. URL: https://www.sagenet.com/insights/siem-options-open-source-vs-enterprise-vs-managed/ (дата звернення: 23.03.2025)
28. What is Cloud SIEM? - Palo Alto Networks. URL: https://www.paloaltonetworks.com/cyberpedia/what-is-cloud-siem (дата звернення: 27.03.2025)
29. Cloud SIEM - Datadog. URL: https://www.datadoghq.com/product/cloud-siem/ (дата звернення: 31.03.2025)
30. What You Need to Know About SIEM Integration with Your Existing ... - ClearNetwork. URL: https://www.clearnetwork.com/what-you-need-to-know-about-siem-integration-with-your-existing-infrastructure/ (дата звернення: 04.04.2025)

31. Литвиненко Д.А., Лащевська Н.О. Виявлення мережевих загроз за допомогою SIEM-систем. «Сучасний стан та перспективи розвитку IoT» : VI Науково-техн. конф., м.Київ, 15 квітня 2025 р. (подано до публікації).

32. Литвиненко Д.А., Лащевська Н.О. Виявлення мережевих загроз за допомогою SIEM-систем. «Застосування програмного забезпечення в ІКТ» : Всеукр. науково-техн. конф., м.Київ, 24 квітня 2025 р. С.421-423.

Додаток А

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра Комп'ютерної інженерії



КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня бакалавра

НА ТЕМУ: «SIEM-СИСТЕМИ ДЛЯ ВИЯВЛЕННЯ МЕРЕЖЕВИХ ЗАГРОЗ»

Виконав студент групи КІД-42
Литвиненко Даніл Андрійович
Керівник дипломної роботи:
Лащевська Наталія Олександрівна,
канд. техн. наук, доц.

Київ – 2025

Мета, об’єкт, предмет дослідження

Мета роботи – оптимізація процесу виявлення мережевих загроз за рахунок використання принципів кореляційного аналізу та автоматизації моніторингу в SIEM-системі Wazuh

Об’єкт дослідження – процеси виявлення, аналізу та обробки мережевих подій безпеки

Предмет дослідження – технології SIEM-моніторингу, засоби автоматизованого збору та обробки логів, методи кореляції подій та реакції на загрози

Актуальність роботи

У роботі досліджено принципи побудови SIEM-систем для виявлення мережевих загроз на прикладі відкритого рішення Wazuh у поєднанні з ELK-стеком. Розгорнуто систему в середовищі Docker, налаштовано збір і обробку логів, реалізовано правила кореляції подій та систему сповіщень. Проведено аналіз типових атак (brute-force, port scanning) і перевірено ефективність детекції.

Оцінено продуктивність системи, визначено її обмеження та запропоновано шляхи вдосконалення: підключення threat intelligence-джерел, інтеграція з MITRE ATT&CK, автоматизація реакції на загрози та побудова дашбордів. Проведено порівняння з іншими SIEM-рішеннями (Splunk, OSSIM) та обґрунтовано доцільність використання Wazuh у невеликих і середніх організаціях.

Принцип роботи SIEM-систем

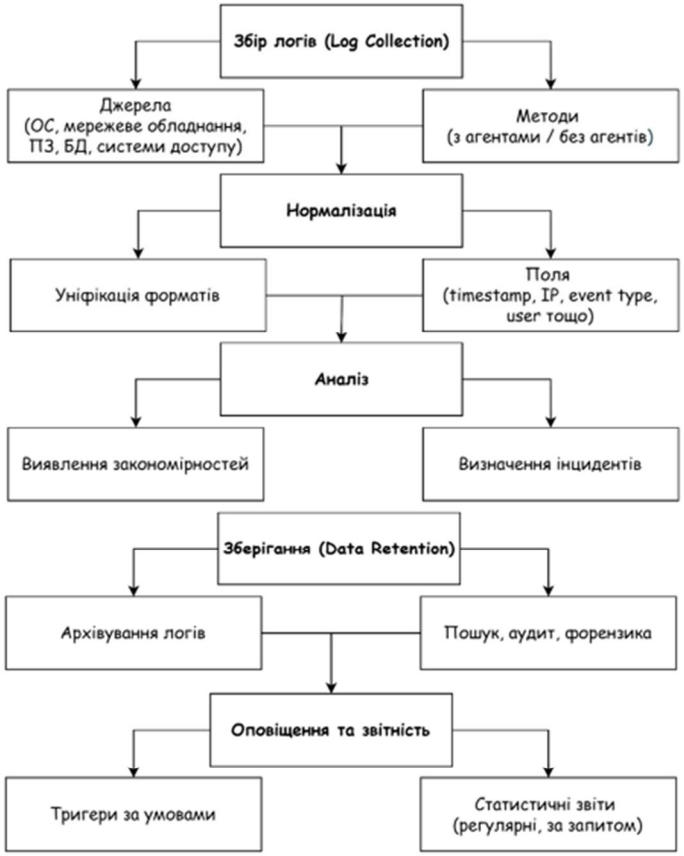


Рис. 1.6 – Принцип роботи SIEM-системи

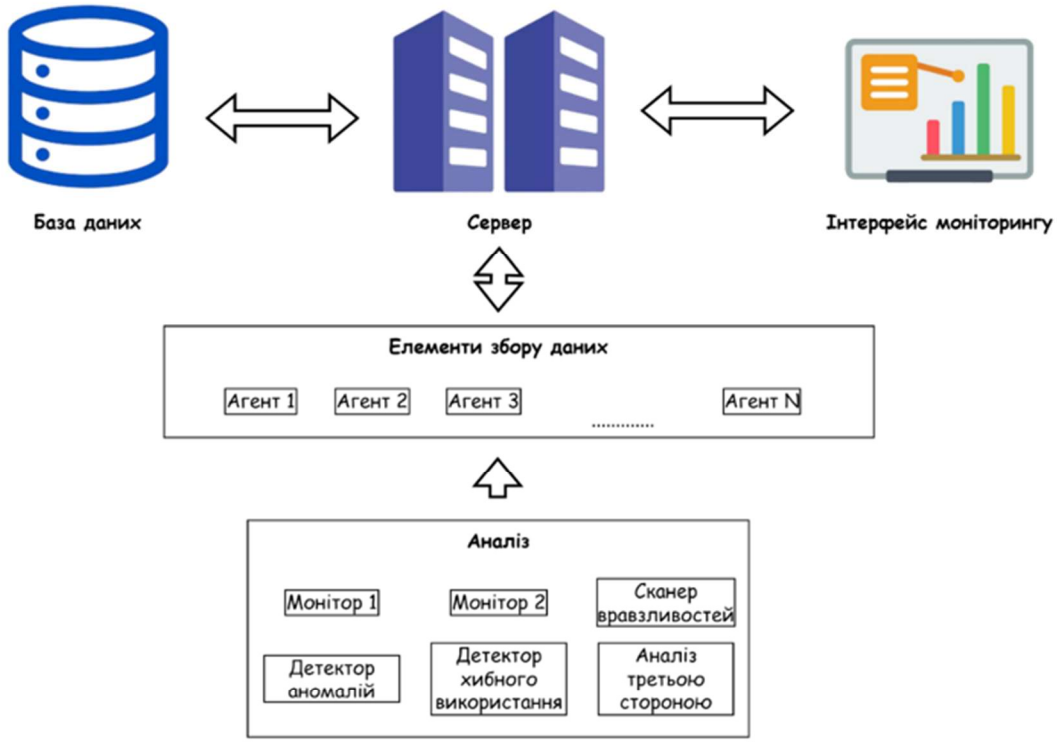


Рис. 1.9 – Архітектура типової open-source SIEM-системи

Архітектура SIEM-системи для виявлення мережевих загроз

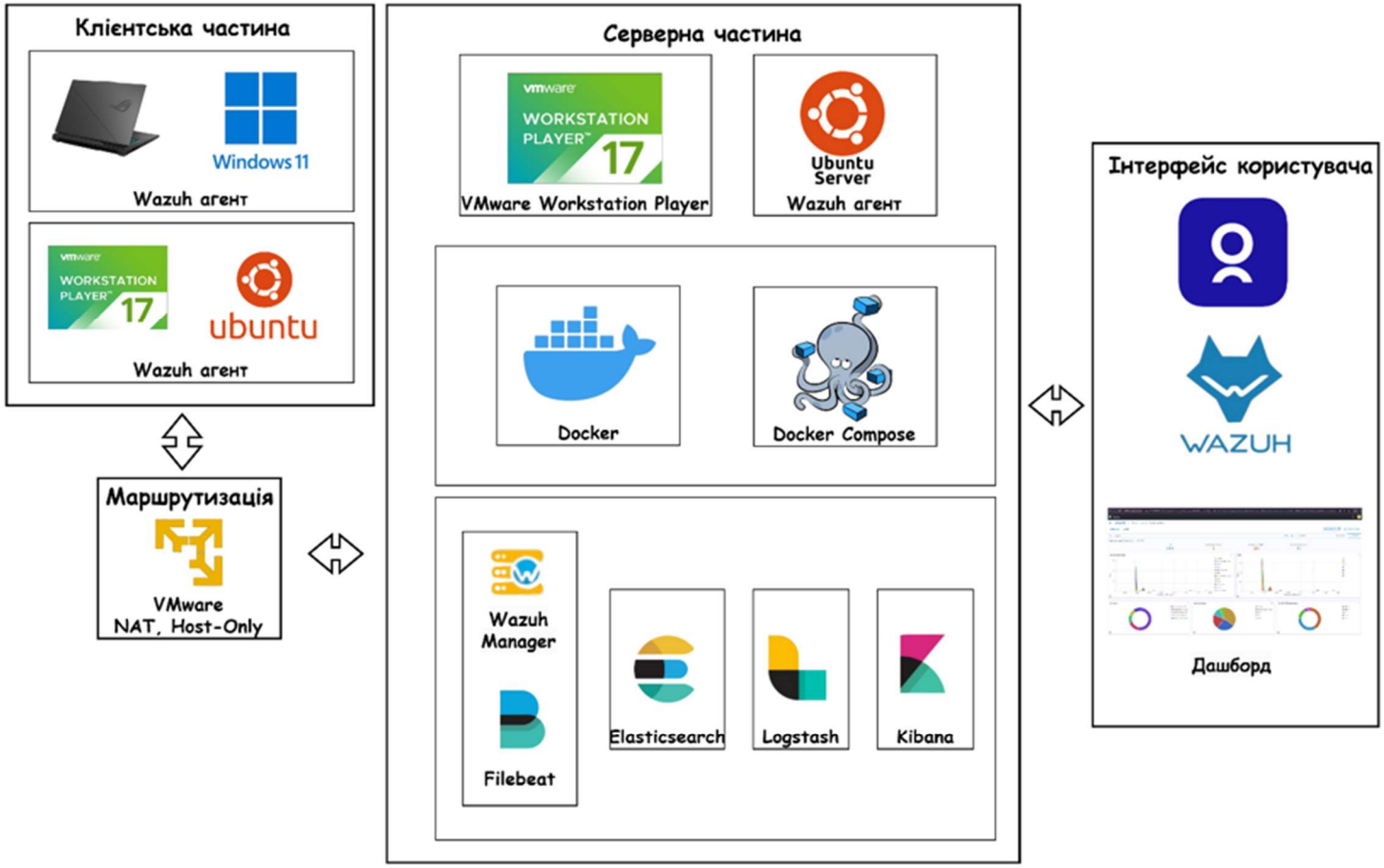


Рис. 2.1 – Архітектура SIEM-системи на базі Wazuh та ELK

Етапи розгортання SIEM-системи через Docker

```
danil-lytvyn@wazuhelk:~$ curl -sSL https://get.docker.com/ | sh
# Executing docker install script, commit: 53a22f61c0628e58e1d6680b49e82993d304b449
+ sudo -E sh -c apt-get -qq update >/dev/null
+ sudo -E sh -c DEBIAN_FRONTEND=noninteractive apt-get -y -qq install ca-certificates curl >/dev/null
+ sudo -E sh -c install -m 0755 -d /etc/apt/keyrings
+ sudo -E sh -c curl -fsSL "https://download.docker.com/linux/ubuntu/gpg" -o /etc/apt/keyrings/docker.asc
+ sudo -E sh -c chmod a+r /etc/apt/keyrings/docker.asc
+ sudo -E sh -c echo "deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/l
ocker.list
+ sudo -E sh -c apt-get -qq update >/dev/null
+ sudo -E sh -c DEBIAN_FRONTEND=noninteractive apt-get -y -qq install docker-ce docker-ce-cli containerd.io
er-buildx-plugin >/dev/null
Scanning processes... [=============================================================================================
```

```
danil-lytvyn@wazuhelk:~$ sudo curl -L "https://github.com/docker/compose/releases/download/1.28.3/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/dock
er-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 11.6M  100 11.6M    0     0  9200k      0  0:00:01  0:00:01 --:--:-- 19.9M
danil-lytvyn@wazuhelk:~$
```

Рис. 2.5 – Процес розгортання Docker та Docker Compose

```
danil-lytvyn@wazuhelk:/opt/wazuh-docker$ sudo docker-compose -f production-cluster.yml up -d
Building with native build. Learn about native build in Compose here: https://docs.docker.com/go/compose-native-build/
Creating volume "wazuh-docker_ossec-api-configuration" with default driver
Creating volume "wazuh-docker_ossec-etc" with default driver
Creating volume "wazuh-docker_ossec-logs" with default driver
Creating volume "wazuh-docker_ossec-queue" with default driver
Creating volume "wazuh-docker_ossec-var-multigroups" with default driver
Creating volume "wazuh-docker_ossec-integrations" with default driver
Creating volume "wazuh-docker_ossec-active-response" with default driver
Creating volume "wazuh-docker_ossec-agentless" with default driver
Creating volume "wazuh-docker_ossec-wodles" with default driver
Creating volume "wazuh-docker_filebeat-etc" with default driver
Creating volume "wazuh-docker_filebeat-var" with default driver
Creating volume "wazuh-docker_worker-ossec-api-configuration" with default driver
Creating volume "wazuh-docker_worker-ossec-etc" with default driver
Creating volume "wazuh-docker_worker-ossec-logs" with default driver
Creating volume "wazuh-docker_worker-ossec-queue" with default driver
Creating volume "wazuh-docker_worker-ossec-var-multigroups" with default driver
Creating volume "wazuh-docker_worker-ossec-integrations" with default driver
Creating volume "wazuh-docker_worker-ossec-active-response" with default driver
Creating volume "wazuh-docker_worker-ossec-agentless" with default driver
Creating volume "wazuh-docker_worker-ossec-wodles" with default driver
Creating volume "wazuh-docker_worker-filebeat-etc" with default driver
Creating volume "wazuh-docker_worker-filebeat-var" with default driver
Creating volume "wazuh-docker_elastic-data-1" with default driver
Creating volume "wazuh-docker_elastic-data-2" with default driver
Creating volume "wazuh-docker_elastic-data-3" with default driver
Pulling wazuh-master (wazuh/wazuh-odfe:4.2.6)...
```

Рис. 2.11 – Процес розгортання ELK стеку зі Wazuh

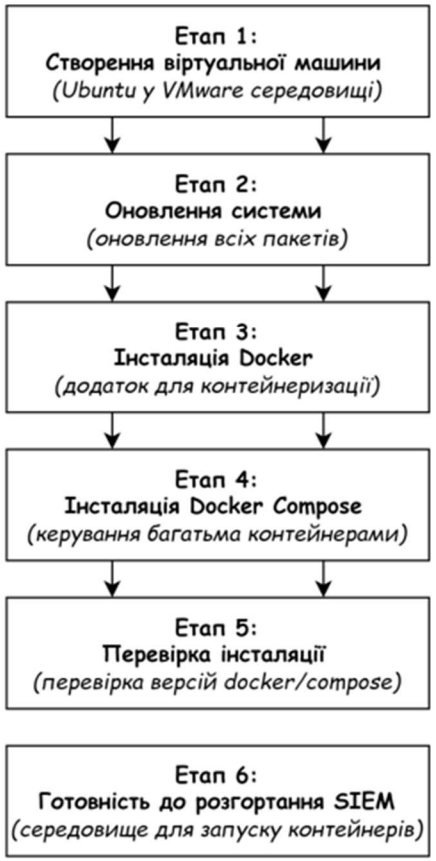


Рис. 2.7 – Етапи підготовки SIEM системи як середовища на базі Docker

Розгортання агентів Wazuh на вузлах

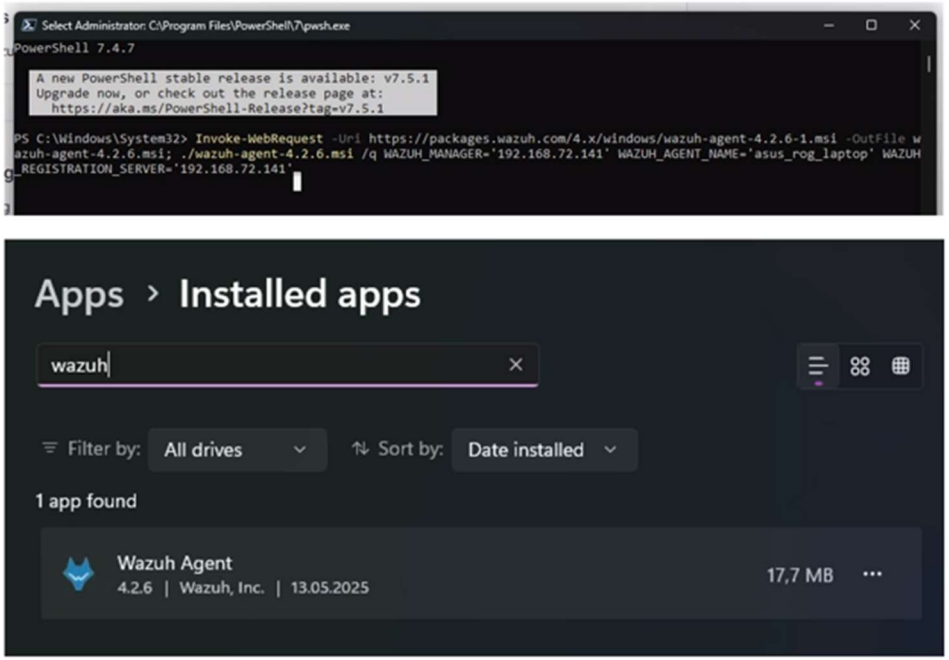


Рис. 2.18 – Процес розгортання агента Wazuh на хост (Windows 11)

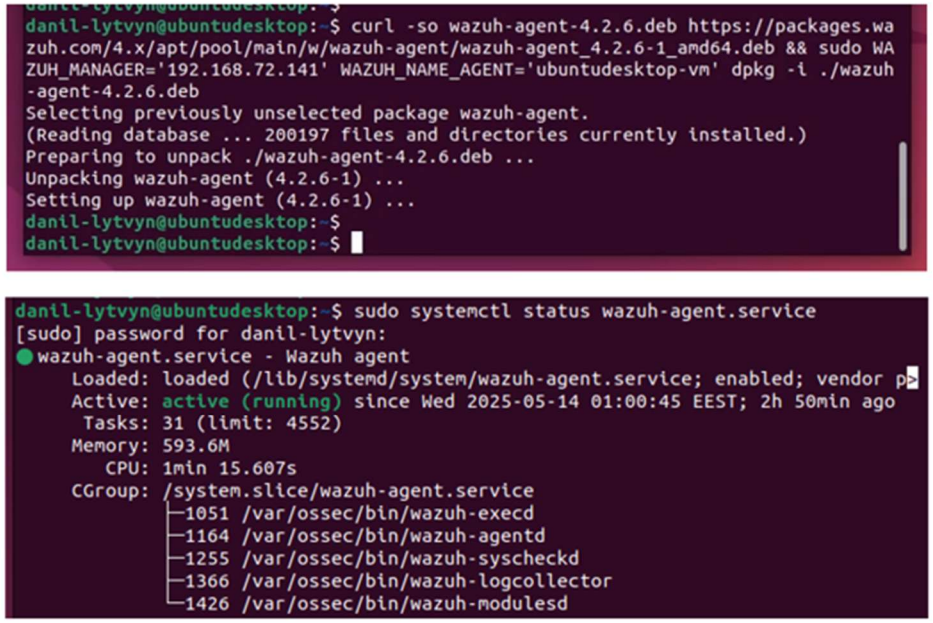


Рис. 2.19 – Процес розгортання агента Wazuh на клієнт (Ubuntu Desktop)

Демонстрація збору даних агентами зі вузлів

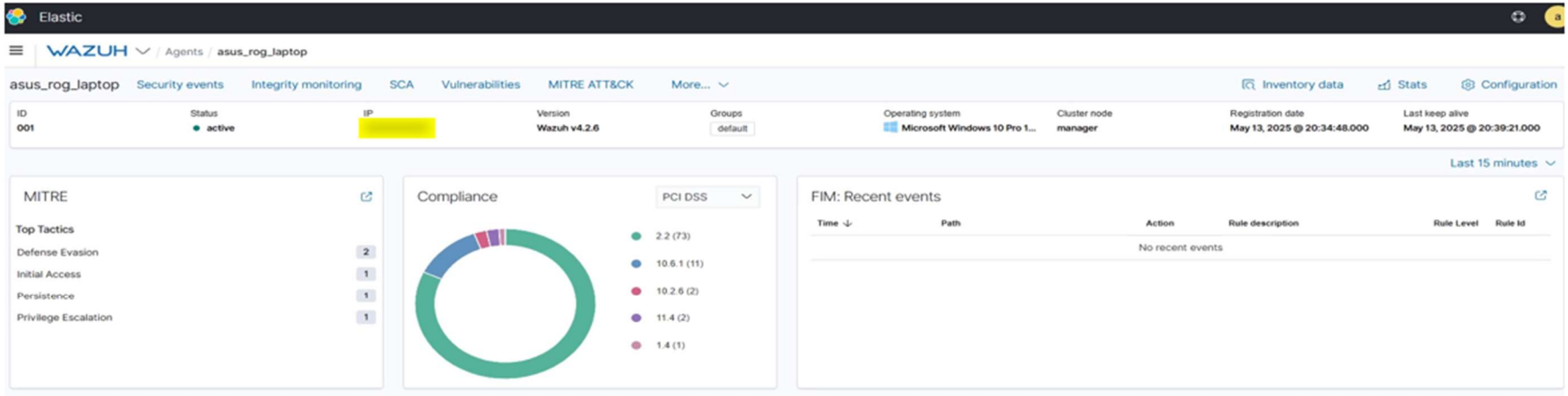


Рис. 2.21 – Початок збору даних зі вузла Windows 11 (клієнт)

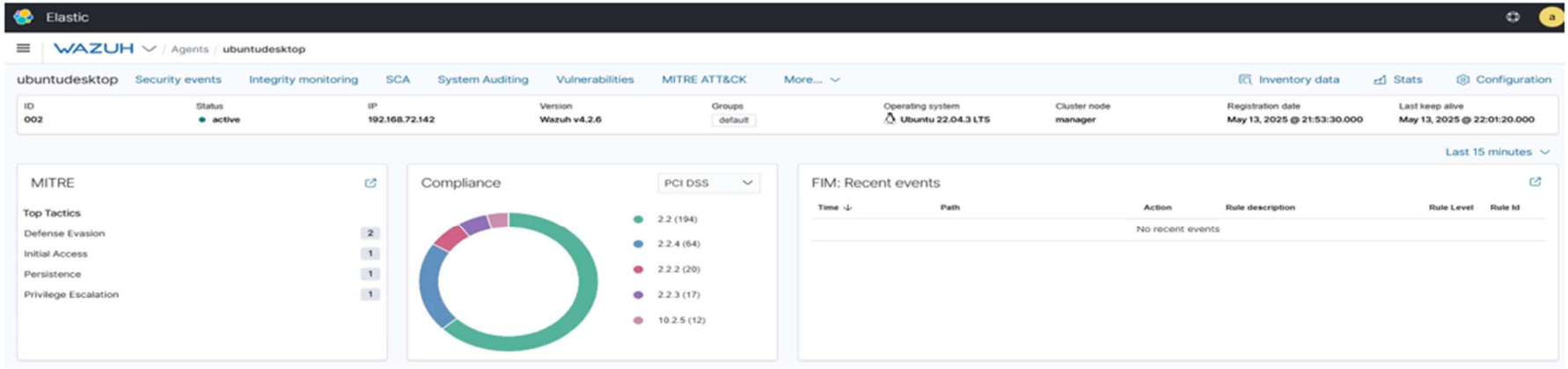


Рис. 2.22 – Початок збору даних зі вузла Ubuntu Desktop (клієнт)

Формування звітів на основі зібраних даних

WAZUH

info@wazuh.com
https://wazuh.com

Inventory data report

ID	Name	IP	Version	Manager	OS	Registration date	Last keep alive
002	ubuntudesktop	192.168.72.142	Wazuh v4.2.6	wazuh-master	Ubuntu 22.04.3 LTS	May 13, 2025 @ 18:53:30.000	May 13, 2025 @ 19:01:40.000

Hardware information

- 2 cores
- AMD Ryzen 7 4800H with Radeon Graphics
- 3.78GB RAM

OS information

- Linux
- #61~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Tue Apr 15 17:03:15 UTC 2
- x86_64
- 6.8.0-59-generic
- Ubuntu 22.04.3 LTS (Jammy Jellyfish)

Packages

Name	Architecture	Version	Vendor	Description
libfastjson4	amd64	0.99.9-1build2	Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>	fast json library for C
libsqlite3-0	amd64	3.37.2-2ubuntu0.1	Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>	SQLite 3 shared library

Processes

Name	Effective user	Priority	State
systemd	root	0	Interruptible sleep (waiting for an event to complete)
kworker/R-ipv6_	root	-20	Idle
kworker/R-kstrp	root	-20	Idle
kworker/u259:0-	root	-20	Idle
kworker/u260:0	root	-20	Idle
kworker/u261:0	root	-20	Idle
kworker/R-crypt	root	-20	Idle
kworker/R-mm_pe	root	-20	Idle
kworker/R-charg	root	-20	Idle
rcu_tasks_kthre	root	0	Idle
rcu_tasks_rude_	root	0	Idle

Network ports

Local IP	Local port	State	Protocol
127.0.0.53	53	-	udp
127.0.0.53	53	listening	tcp
::	9200	listening	tcp6

Network interfaces

Name	Mac	State	MTU	Type
br-fb2f34d3b4f3	6e:bf:20:a5:0a:51	up	1500	ethernet
veth6aed2ca	22:d9:3b:8b:06:12	up	1500	ethernet

Network settings

Interface	address	Netmask	Protocol	Broadcast
veth9de54eb	fe80::b04d:aff:fef8:9caa	ffff:ffff:ffff:ffff::	ipv6	-
vethc9fa086	fe80::2484:73ff:fed6:ded1	ffff:ffff:ffff:ffff::	ipv6	-
ens34	192.168.36.129	255.255.255.0	ipv4	192.168.36.255
vethc2e6487	fe80::b87a:80ff:fe34:281e	ffff:ffff:ffff:ffff::	ipv6	-

Рис. 2.24 – Фрагмент експорту отриманих даних зі одного з вузлів (Ubuntu Desktop)

Перевірка спрацювання алертів на події

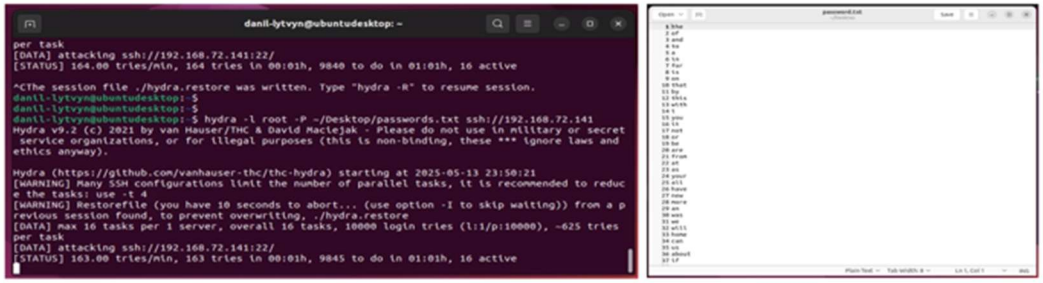


Рис. 2.31 – Емуляція атаки типу brute-force через клієнт на сервер

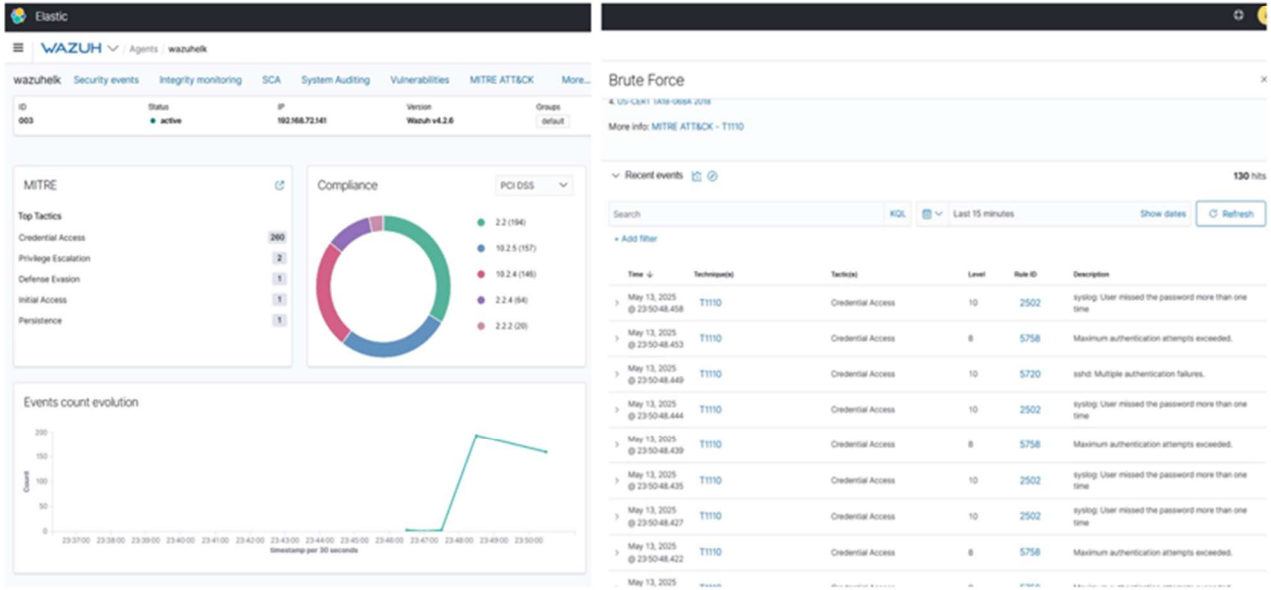


Рис. 2.32 – Виявлення атаки типу brute-force у Wazuh



Рис. 2.28 – Алгоритм аналізу та кореляції подій

Перевірка спрацювання алертів на події

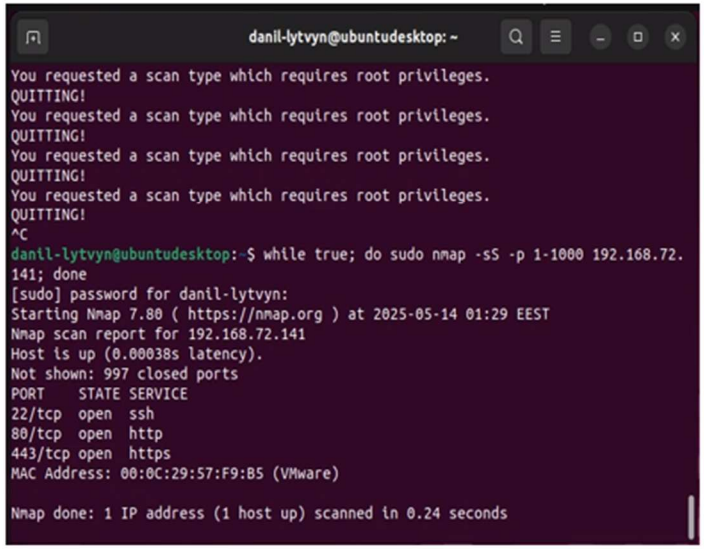


Рис. 2.33 – Запуск короткого циклічного сценарію Nmap для імітації сканування портів через клієнт на сервер

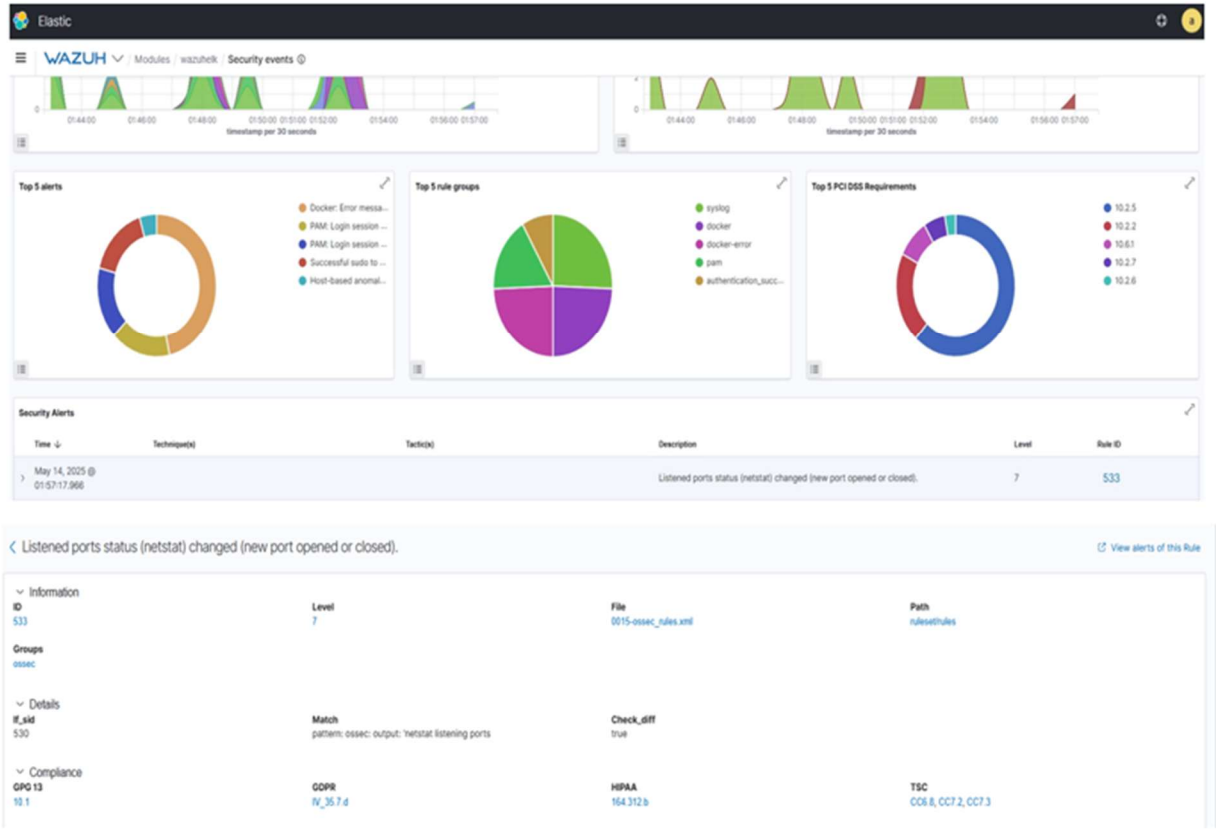


Рис. 2.34 – Виявлення сканування портів у Wazuh

Покращення якості виявлення та автоматизація

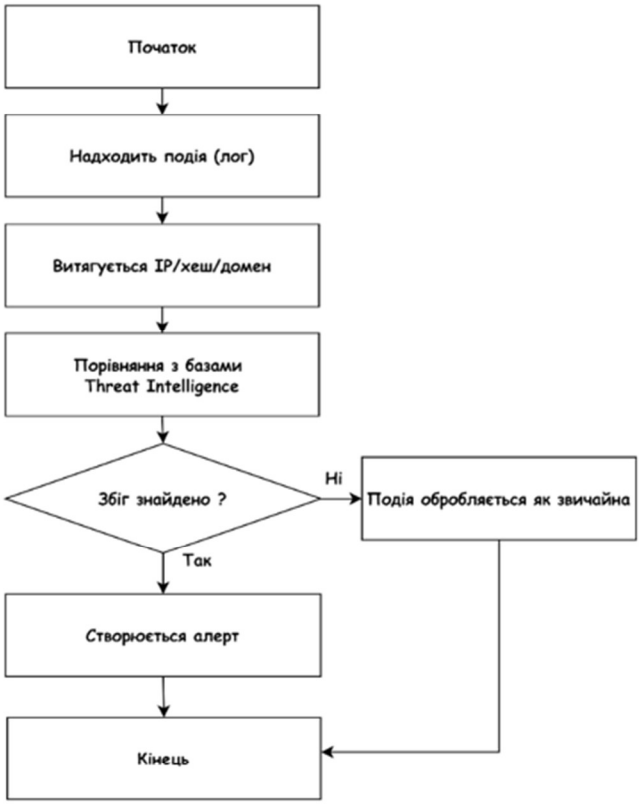


Рис. 3.1 – Метод інтеграції зовнішніх Threat Intelligence-джерел



Рис. 3.2 – Метод фільтрації помилкових спрацьовувань (false positives)

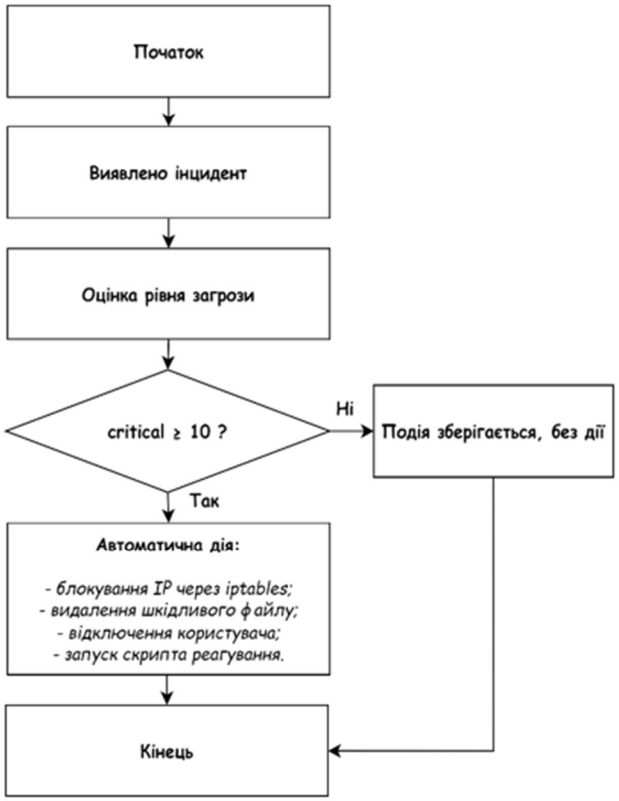


Рис. 3.4 – Метод автоматизації відповіді на інциденти через Wazuh Active Response

Кошторис реалізації та порівняння з готовими рішеннями

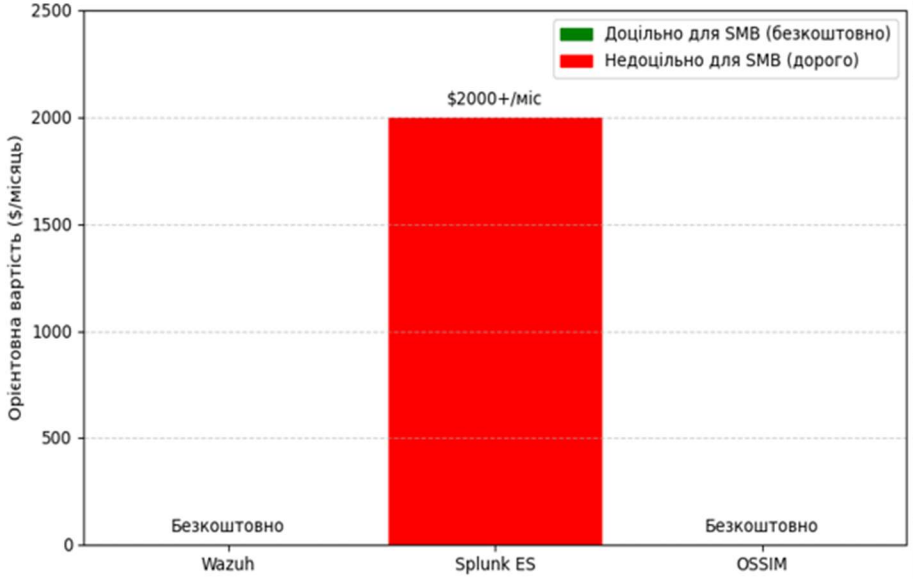


Рис. 3.7 – Діаграма доцільності використання SIEM рішення Wazuh

Табл. 3.2 – Порівняння Wazuh з іншими SIEM-системами

Критерій	Wazuh	Splunk Enterprise Security	OSSIM
Ліцензія	**Open-source (безкоштовно)**	Комерційна (від $2000/місяць)	Open-source
Ресурсоємність	**Середня**	Висока	Висока
Налаштування та гнучкість	**Висока, YAML/XML-конфігурації**	Дуже висока, через GUI/API	Обмежена гнучкість
Підтримка	**Спільнота + платна підтримка**	Професійна підтримка від вендора	Обмежена спільнотою
Масштабованість	**Висока (через Docker/Kubernetes)**	Відмінна (enterprise-рівень)	Середня
Інтерфейс користувача	**Через Kibana**	Власна аналітична консоль	Старий інтерфейс
Підтримка інтеграцій	**MITRE ATT&CK, VirusTotal, OTX**	Понад 1000 інтеграцій	Базові інтеграції
Автоматична реакція	**Active Response (bash/python)**	SOAR-модулі (автоматизована дія)	Часткова реалізація
Початкове розгортання	**Від 30 хв (Docker Compose)**	Тривалий, складний процес	Складне, потребує ручного налаштування
Актуальність	**Активна розробка**	Постійна розробка	Підтримка скорочується

Висновки

У процесі реалізації SIEM-системи на базі Wazuh та стеку ELK вдалося створити ефективне рішення для моніторингу подій безпеки. Завдяки контейнеризації через Docker забезпечено гнучкість та швидке розгортання, а налаштування збору логів, базових правил кореляції та сповіщень дозволило виявляти типові загрози, зокрема brute-force та портове сканування, із подальшою візуалізацією в Kibana.

Порівняння з іншими SIEM-рішеннями підтвердило конкурентоспроможність Wazuh як доступної open-source платформи для малих і середніх організацій. Перспективними напрямами розвитку є інтеграція з хмарними сервісами, CMDB, системами автоматизованого реагування та розширення функціоналу за рахунок власних модулів.

Публікації та апробація роботи

1. Литвиненко Д.А., Лащевська Н.О. Виявлення мережевих загроз за допомогою SIEM-систем. «Сучасний стан та перспективи розвитку IoT» : VI Науково-техн. конф., м.Київ, 15 квітня 2025 р. (подано до публікації).
2. Литвиненко Д.А., Лащевська Н.О. Виявлення мережевих загроз за допомогою SIEM-систем. «Застосування програмного забезпечення в ІКТ» : Всеукр. науково-техн. конф., м.Київ, 24 квітня 2025 р. С.421-423.