ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ

КАФЕДРА СИСТЕМ ТА ТЕХНОЛОГІЙ КІБЕРБЕЗПЕКИ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Технологія оптимізації правил у SIEM-системах для виявлення аномалій та зниження кількості хибнопозитивних виявлень»

зі спеціальності *125 Кібербезпека та захист інформації*
(код, найменування спеціальності)

освітньо-професійної програми *Інформаційна та кібернетична безпека*
(назва програми)

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

_____ Олександр ВОРОНА
(підпис)

Виконала: здобувач вищої освіти групи БСДМ-61
ЧЕЧИК Марина
(прізвище, ім'я)

Керівник завідувач кафедрою ГАЙДУР Галина
(науковий ступінь, вчене звання, прізвище, ім'я)

Рецензент _____
(науковий ступінь, вчене звання, прізвище, ім'я)

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ

Кафедра Систем та технологій кібербезпеки
Ступінь вищої освіти Магістр
Спеціальність 125 Кібербезпека та захист інформації
Освітньо-професійна програма Інформаційна та кібернетична безпека

ЗАТВЕРДЖУЮ
Завідувач кафедри
Систем та технологій
кібербезпеки
_____ Галина ГАЙДУР
“___” жовтня 2025 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

ЧЕЧИК Марині Олексіївні
(прізвище, ім’я)

1. Тема кваліфікаційної роботи: «Технологія оптимізації правил у SIEM-системах для виявлення аномалій та зниження кількості хибнопозитивних виявлень»

керівник кваліфікаційної роботи Гайдур Галина Іванівна, завідувач кафедрою
(прізвище, ім’я, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «___» жовтня 2025 року № ___.

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 25.12.2025 р.

3. Вихідні дані до кваліфікаційної роботи
інформаційні ресурси організації;
рішення SIEM Wazuh;
наукова та технічна література з питань оптимізації правил SIEM і виявлення аномалій;

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Дослідження проблеми оптимізації правил у SIEM-системах та впливу хибнопозитивних спрацювань на роботу SOC.

2. Аналіз методів та засобів оптимізації правил у Wazuh, включно з механізмами кореляції, декодерами, правилами та Local Rules.

3. Технологія оптимізації правил у Wazuh: розроблення та впровадження алгоритму fast-travel для виявлення аномальної активності користувачів.

5. Перелік графічного матеріалу

Презентація PowerPoint.

6. Дата видачі завдання 01.10.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ зп	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Визначення актуальності проблеми оптимізації правил у SIEM-системах та її впливу на ефективність роботи SOC.	01.10.2025 р.	
2.	Аналіз наукової, технічної та нормативної літератури з питань виявлення аномалій та зниження хибнопозитивних спрацювань у SIEM.	12.10.2025 р.	
3.	Аналіз існуючих підходів та рішень SIEM щодо оптимізації правил і покращення точності виявлення аномальної активності.	27.10.2025 р.	
4.	Дослідження функцій і механізмів обробки подій у Wazuh (декодери, local rules) як основи кореляції.	03.11.2025 р.	
5.	Розроблення алгоритму fast-travel для виявлення аномальної географічної активності користувачів.	07.11.2025 р.	
6.	Інтеграція алгоритму fast-travel у Wazuh: налаштування декодерів, pipeline та local rules.	15.11.2025 р.	
7.	Розроблення рекомендацій щодо оптимізації правил у Wazuh для зниження кількості хибнопозитивних виявлень.	25.12.2025 р.	
8.	Оформлення результатів дослідження та підготовка роботи до захисту.	26.12.2025 р.	

Здобувач вищої освіти _____ Марина ЧЕЧИК
(підпис) *(ім'я, прізвище)*

Керівник кваліфікаційної роботи _____ Галина ГАЙДУР
(підпис) *(ім'я, прізвище)*

ВІДГУК РЕЦЕНЗЕНТА
на кваліфікаційну роботу

здобувача ЧЕЧИК Марини

на тему: «Технологія оптимізації правил у SIEM-системах для виявлення аномалій та зниження кількості хибнопозитивних виявлень»

Актуальність: Зростання складності кіберзагроз, активне використання хмарних сервісів та збільшення обсягу телеметрії призводять до різкого зростання кількості подій безпеки, що опрацьовуються SIEM-системами. У таких умовах ефективність роботи SOC значною мірою залежить від якості та точності правил кореляції, які визначають здатність системи своєчасно виявляти аномалії та реальні атаки. Однією з найбільш критичних проблем є надмірна кількість хибнопозитивних спрацювань, що перевантажує аналітиків, знижує швидкість реагування та створює ризик пропуску справжніх інцидентів.

Оптимізація правил SIEM дозволяє зменшити кількість помилкових алертів, підвищити точність виявлення аномалій та забезпечити стабільну роботу SOC у сучасних архітектурах безпеки. Це робить дослідження технологій оптимізації правил SIEM актуальним і необхідним для організацій, які прагнуть підвищити рівень кіберстійкості та ефективність реагування на інциденти в умовах постійного зростання кіберзагроз.

Позитивні сторони:

1. На основі проведеного аналізу в роботі розкрито зміст проблеми оптимізації правил у SIEM-системах, визначено мету та завдання дослідження, а також охарактеризовано ключові аспекти впливу хибнопозитивних спрацювань на ефективність роботи SOC.
2. Досліджено методи та підходи до зниження кількості хибнопозитивних виявлень у сучасних SIEM-системах, зокрема детально проаналізовано можливості платформи Wazuh у частині кореляції подій, побудови правил та обробки аномалій.
3. Розглянуто зміст технології оптимізації правил у Wazuh, описано механізм роботи правил local rules, а також розроблено й впроваджено алгоритм fast-travel для виявлення аномальної активності користувачів та наведено рекомендації щодо його застосування.
4. Наведено приклад практичної реалізації оптимізованого правила SIEM, що підвищує практичну цінність роботи. Список використаної літератури свідчить про вміння здобувача працювати з науковими та технічними джерелами за тематикою дослідження.

Недоліки:

5. У кваліфікаційній роботі бажано було б провести детальніший аналіз застосування оптимізованих правил SIEM на прикладі конкретної організації або реального середовища SOC, що дозволило б підсилити практичну складову дослідження.
6. Розроблений алгоритм fast-travel міг би бути продемонстровані на реальних інцидентах, що підвищило б глибину оцінки ефективності методики.

Відзначені зауваження не впливають на загальну позитивну оцінку кваліфікаційної роботи.

Висновок: Враховуючи недоліки, кваліфікаційна робота заслуговує оцінку **“відмінно”**, а здобувач(ка) **ЧЕЧИК Марина** – присвоєння кваліфікації магістр з кібербезпеки за освітньо-професійною програмою інформаційна та кібернетична безпека.

Рецензент:

(науковий ступінь, вчене звання)	*(підпис)*	*(ім’я, прізвище)*

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ

ПОДАННЯ
ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ
ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Направляється здобувач ЧЕЧИК Марина до захисту кваліфікаційної роботи
(прізвище, ім'я)
спеціальності 125 Кібербезпека та захист інформації
освітньо-професійної програми Інформаційна та кібернетична безпека
(шифр і назва спеціальності)
на тему: «Технологія оптимізації правил у SIEM-системах для виявлення аномалій та зниження кількості хибнопозитивних виявлень».
Кваліфікаційна робота і рецензія додаються.

Директор інституту _____ Євгенія ІВАНЧЕНКО
(підпис) *(ім'я, прізвище)*

Висновок керівника кваліфікаційної роботи

Здобувач ***ЧЕЧИК Марина*** обрала тему роботи, метою якої було дослідити зміст технології забезпечення безпечної роботи гібридних працівників організації та розробка рекомендацій щодо її реалізації. Перелік використаних джерел свідчить про вміння здобувачем розбиратись в наукових питаннях та застосовувати їх при дослідженнях. Під час виконання кваліфікаційної роботи ***ЧЕЧИК Марина*** показала добру теоретичну та практичну підготовку, вміння самостійно вирішувати питання і робити висновки. Роботу виконував сумлінно, акуратно та вчасно за планом.

Все це дозволяє оцінити виконану кваліфікаційну роботу здобувача ***ЧЕЧИК Марини*** на оцінку **"відмінно"** та присвоїти йому кваліфікацію магістр з кібербезпеки за освітньо-професійною програмою інформаційна та кібернетична безпека.

Керівник кваліфікаційної роботи _____ Галина Гайдур
(підпис) *(ім'я, прізвище)*
"____" _____ 2025 року

Висновок кафедри про кваліфікаційну роботу

Кваліфікаційна робота розглянута. Здобувачка **ЧЕЧИК Марина** допускається до захисту даної кваліфікаційної роботи в Екзаменаційній комісії.

Завідувач кафедри Систем та технологій кібербезпеки
(назва)

_____ Галина ГАЙДУР
(підпис) *(ім'я, прізвище)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи: 54 сторінки, 14 рисунків, 1 таблиця, 10 джерел.

Об'єкт дослідження – процес виявлення аномальної активності та обробки подій у SIEM-системах.

Предмет дослідження – технологія оптимізації правил у SIEM-системі Wazuh для зниження кількості хибнопозитивних спрацювань та підвищення точності виявлення аномалій.

Мета роботи – розробити технологію оптимізації правил у Wazuh та сформувати рекомендації щодо зниження хибнопозитивних виявлень у SIEM-системах.

Методи дослідження – вивчення експлуатаційної документації Wazuh, аналіз подій безпеки, методи порівняння та моделювання аномальної активності, експериментальна розробка та тестування правил SIEM.

Платформа Wazuh забезпечує потужний набір інструментів для кореляції подій, аналізу поведінки та налаштування механізмів виявлення загроз. Вона включає систему декодерів, правилоорієнтовану модель кореляції, механізм Local Rules, підтримку Threat Intelligence та можливість гнучкого розширення функціоналу за допомогою користувацьких скриптів. Такі можливості дозволяють створювати точніші правила та знижувати рівень хибнопозитивних виявлень без втрати якості детекції.

У роботі проаналізовано проблему хибнопозитивних спрацювань у SIEM-системах, визначено її вплив на ефективність роботи SOC та обґрунтовано потребу в технологічній оптимізації правил. Проведено аналіз існуючих підходів та методів оптимізації SIEM, а також досліджено функції та механізми Wazuh у контексті виявлення аномальної активності.

На основі проведених досліджень запропоновано технологію оптимізації правил у Wazuh, що включає розробку та впровадження алгоритму fast-travel для виявлення географічної аномальної активності користувачів. Розроблено рекомендації щодо підвищення точності виявлення загроз та зниження кількості хибнопозитивних спрацювань у Wazuh шляхом удосконалення структури правил, використання контекстної інформації та оптимізації процесу кореляції.

Галузь використання – кібербезпека інформаційних систем та ресурсів організації.

ІНФОРМАЦІЙНІ РЕСУРСИ ОРГАНІЗАЦІЇ, ХИБНОПОЗИТИВНІ СПРАЦЮВАННЯ, ОПТИМІЗАЦІЯ ПРАВИЛ, ВИЯВЛЕННЯ АНОМАЛІЙ, FAST-TRAVEL, КОРЕЛЯЦІЯ ПОДІЙ

ABSTRACT

Text part of the qualification work: 54 pages, 14 figures, 1 table, 10 sources.

Object of research – the process of detecting anomalous activity and processing security events in SIEM systems.

Subject of research – the technology for optimizing rules in the Wazuh SIEM system to reduce the number of false positive alerts and improve anomaly detection accuracy.

The aim of the work – to develop a technology for optimizing rules in Wazuh and to provide recommendations for reducing false positive detections in SIEM systems.

Research methods – study of Wazuh operational documentation, analysis of security events, comparison methods, anomaly activity modeling, and experimental development and testing of SIEM rules.

The Wazuh platform provides a powerful set of tools for event correlation, behavioral analysis, and configuration of threat detection mechanisms. It includes a system of decoders, a rule-based correlation model, the Local Rules mechanism, Threat Intelligence support, and the ability to flexibly extend functionality through custom scripts. These capabilities allow the creation of more precise rules and significantly reduce the number of false positives without compromising detection quality.

The work analyzes the problem of false positive alerts in SIEM systems, identifies their impact on SOC efficiency, and substantiates the need for technological optimization of detection rules. The study includes an analysis of existing approaches and methods for SIEM optimization and an examination of Wazuh's mechanisms in the context of anomaly detection.

Based on the conducted research, a technology for optimizing rules in Wazuh is proposed, including the development and implementation of the fast-travel algorithm for detecting geographically anomalous user activity. Recommendations have been developed to improve threat detection accuracy and reduce false positives in Wazuh by enhancing rule structure, using contextual information, and optimizing the correlation process.

Field of application – cybersecurity of organizational information systems and resources.

ORGANIZATIONAL INFORMATION RESOURCES, FALSE POSITIVES, RULE OPTIMIZATION, ANOMALY DETECTION, FAST-TRAVEL, EVENT CORRELATION

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

SOC - Security Operations Center

SIEM - Security Information and Event Management

MFA - Multi-Factor Authentication

UEBA - User and Entity Behavior Analytics

CDB - Constant DataBase

SID - Signature Identifier

IP - Internet Protocol

ID - Identifier

VPN - Virtual Private Network

CSV - Comma-Separated Values

ВСТУП

Актуальність дослідження. Підхід до забезпечення кібербезпеки, який ґрунтувався на простому накопиченні логів та застосуванні статичних правил детекції, вже не відповідає сучасним умовам. Сьогодні обсяги подій безпеки стрімко зростають, інфраструктури ускладнюються, а атакувальні техніки стають більш витонченими. У таких умовах центри моніторингу безпеки (SOC) змушені працювати з великими масивами телеметрії та тисячами щоденних спрацювань, значна частина яких є хибнопозитивними.

Надмірна кількість хибнопозитивних сповіщень призводить до перевантаження аналітиків, зниження швидкості реагування та формування alert fatigue, що збільшує ризик пропуску реальних інцидентів. Водночас класичні правила SIEM, побудовані без урахування контексту, поведінки користувачів або реальних сценаріїв атак, не забезпечують достатньої точності. Це формує потребу в нових підходах до оптимізації правил та застосуванні ефективних алгоритмів виявлення аномалій.

Сучасні системи, такі як Wazuh, надають значно ширші можливості для гнучкої побудови детекцій: використання правил, локальних кореляційних механізмів, а також інтеграції власних обчислювальних алгоритмів. Однак ефективність таких інструментів залежить від правильної оптимізації правил, адаптації їх під конкретні сценарії та мінімізації хибнопозитивних спрацювань. Оптимізація правил SIEM, включно з розробкою алгоритмів виявлення аномалій, є необхідною умовою підвищення ефективності SOC та забезпечення високого рівня кіберзахисту організації.

Вищесказане визначає актуальність теми даної кваліфікаційної роботи, основний зміст якої становлять дослідження та розроблення технології оптимізації правил у SIEM-системі Wazuh для зниження кількості хибнопозитивних виявлень та покращення точності детекції аномальної активності.

Об’єкт дослідження – процес виявлення аномальної активності та обробки

подій у SIEM-системах.

Предмет дослідження – технологія оптимізації правил у SIEM-системі Wazuh для зниження кількості хибнопозитивних спрацювань та підвищення точності виявлення аномалій.

Мета роботи – розробити технологію оптимізації правил у Wazuh та сформувати рекомендації щодо зниження хибнопозитивних виявлень у SIEM-системах.

Наукові завдання:

дослідити сутність проблеми хибнопозитивних спрацювань у SIEM-системах та їх вплив на ефективність роботи SOC;

проаналізувати сучасні підходи та методи оптимізації правил у SIEM для підвищення точності виявлення аномалій;

дослідити можливості та функціональні особливості платформи Wazuh у контексті кореляції подій та зниження хибнопозитивних виявлень;

проаналізувати методи та засоби оптимізації правил у Wazuh, включно з механізмами декодерів, правил і local rules;

розкрити порядок реалізації технології оптимізації правил у Wazuh та розробити рекомендації щодо зниження хибнопозитивних спрацювань.

Методи дослідження – вивчення експлуатаційної документації Wazuh, аналіз подій безпеки, методи порівняння та моделювання аномальної активності, експериментальна розробка та тестування правил SIEM.

Практичне значення одержаних результатів: запропоновано технологію оптимізації правил у SIEM-системі Wazuh, спрямовану на зниження кількості хибнопозитивних спрацювань та підвищення точності виявлення аномальної активності. Розроблено алгоритм fast-travel та рекомендації для фахівців з кібербезпеки щодо впровадження оптимізованих правил.

Результати кваліфікаційної роботи апробовані на Всеукраїнській науковій конференції «Актуальні проблеми кібербезпеки», яка відбулася 27 жовтня 2025 року в Державному університеті інформаційно-комунікаційних технологій, м. Київ.

1 ДОСЛІДЖЕННЯ ПРОБЛЕМИ ОПТИМІЗАЦІЇ ПРАВИЛ У SIEM-СИСТЕМАХ

1.1. Аналіз проблеми хибнопозитивних спрацювань у SIEM та вплив на роботу SOC

Системи управління інформаційною безпекою та подіями (SIEM) є ключовим елементом сучасних архітектур кіберзахисту, забезпечуючи централізоване збирання, кореляцію та аналіз подій безпеки. Проте ефективність роботи SIEM залежить не лише від можливостей самої платформи, але й від якості правил детекції, на основі яких формується більшість інцидентів. Однією з найсуттєвіших проблем у роботі SIEM є велика кількість хибнопозитивних спрацювань – помилкових інцидентів, які не становлять реальної загрози, але класифікуються системою як потенційні атаки.

Проблема хибнопозитивних спрацювань характерна для більшості організацій, де SIEM використовується для моніторингу джерел подій – систем автентифікації, мережевих пристроїв, серверів, робочих станцій, застосунків тощо. За даними одного з досліджень, майже дві третини всіх алертів, які генерує SOC, виявляються хибнопозитивними або дубльованими/шумовими [1]. Надмірна кількість таких подій перевантажує аналітиків SOC, сповільнює швидкість реагування і призводить до явища «alert fatigue» – коли аналітики втомлюються від великої кількості алертів і можуть пропускати реальні загрози.

Хибнопозитивні інциденти виникають з різних причин. Найпоширенішими є некоректно налаштовані або занадто загальні правила, відсутність контексту при формуванні події, зміни в інфраструктурі без відповідного оновлення правил, недостатня якість логів, а також поведінкові особливості користувачів, які не були враховані під час створення детекцій. Крім того, частина виявлених подій є нормальними технологічними процесами, але SIEM інтерпретує їх як підозрілі через недостатньо адаптовану логіку кореляції.

Високий рівень хибнопозитивних подій негативно впливає на роботу SOC у кількох напрямках. По-перше, суттєво зростає навантаження на аналітиків, які змушені витрачати значний час на обробку сповіщень, що не становлять реальної загрози. За даними звіту 2023 State of Threat Detection від Vectra AI, SOC-команди щодня отримують у середньому близько 4 500 сповіщень, і приблизно 67 % з них залишаються необробленими через перевантаження аналітиків та велику частку шумових або хибнопозитивних алертів (рисунок 1.1). По-друге, надмірна кількість незначущих подій знижує якість реагування, оскільки час, витрачений на перевірку шумових алертів, зменшує можливість своєчасно ідентифікувати та локалізувати реальні атаки. У результаті зростає середній час первинної обробки інцидентів, а SOC втрачає здатність підтримувати стабільний рівень оперативності. По-третє, велика кількість хибнопозитивних спрацювань підвищує ризик людської помилки, коли важливі події можуть бути проігноровані або віднесені до шуму через явище «alert fatigue» – професійної втоми від надмірної кількості сповіщень. Проблему підтверджують і інші аналітичні огляди, які вказують, що значна частина SOC стикається з тим, що понад 50 % алертів є некритичними або шумовими, що ускладнює виявлення справжніх атак.

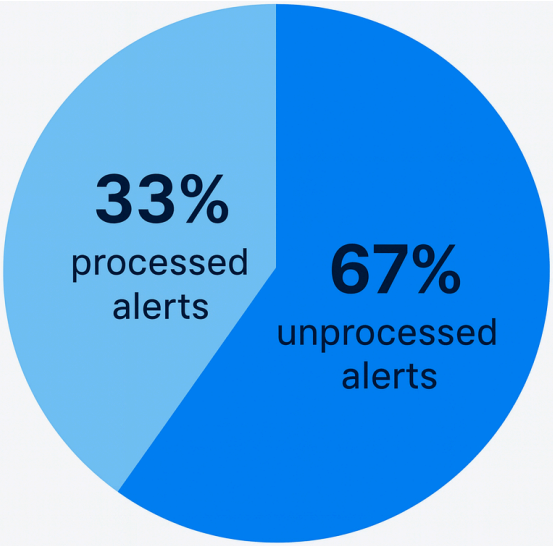

Рис. 1.1. Співвідношення шумових (неопрацьованих) алертів до опрацьованих

Таким чином, проблема хибнопозитивних спрацювань є критично важливим викликом для сучасних SOC. Її подолання потребує комплексного підходу до оптимізації правил SIEM, включно з уточненням умов спрацювання, впровадженням

контекстних перевірок, аналізом поведінкових моделей користувачів, адаптацією правил під реальне навантаження, та регулярним переглядом і коригуванням логіки детекцій відповідно до змін інфраструктури. Вирішення цього питання є необхідною умовою для підвищення ефективності SIEM та загальної кіберстійкості організації.

1.2 Підходи до оптимізації правил SIEM та сучасні методи виявлення аномалій

Ефективність SIEM значною мірою залежить від того, наскільки якісно налаштовані правила виявлення подій. Сучасні підходи до оптимізації правил можна умовно поділити на кілька груп:

1) Ручне налаштування та “тюнинг” правил.

Базовий рівень оптимізації полягає в регулярному перегляді наявних правил, уточненні умов спрацювання та видаленні або переробці застарілих детекцій. Практичні гіди від вендорів SIEM рекомендують:

– задавати більш точні умови спрацювання, поєднуючи кілька подій у кореляційні ланцюжки (наприклад - невдала автентифікація, за якою слідує успішний вхід з нетипової геолокації та підвищення привілеїв);

– коригувати порогові значення, періоди агрегації, використовувати списки довіри (whitelists) для легітимних, але шумних подій;

– вимикати або переписувати правила, які систематично генерують хибнопозитивні сповіщення.

Окремо виділяють адаптивні порогові значення, коли система автоматично підлаштовує поріг спрацювання залежно від історичних даних та сезонності активності, що дозволяє зменшити кількість помилкових алертів у пікові періоди та не пропускати аномалії при низькому фоні.

2) Контекстуальне збагачення та ризик-орієнтований підхід.

Ще одним напрямом оптимізації є додавання контексту до подій: важливості активу, ролі користувача, даних з Threat Intelligence, геолокації, відомостей про MFA тощо. У кейсах з практики показано, що врахування геолокації та логів

багатофакторної автентифікації дозволило скоротити кількість хибнопозитивних спрацювань на окремі сценарії (наприклад, «подорожуючі» користувачі) на десятки відсотків. Ризик-орієнтований підхід передбачає пріоритизацію алертів залежно від критичності активу та ймовірності компрометації, що також зменшує операційне навантаження на SOC.

3) Автоматизована оцінка якості правил та симуляція атак.

Зі зростанням кількості правил ручний аналіз їхньої ефективності стає непрактичним, тому з’являються рішення, які автоматизують аудит детекцій. Такі платформи моделюють відомі техніки атак (зокрема з матриці MITRE ATT&CK), запускають їх в середовища і перевіряють, які правила SIEM спрацювали, а які ні. Це дає можливість автоматично виявляти «німі» правила, оптимізувати надмірно шумні детекції та підтримувати актуальність бібліотеки правил.

4) Використання сучасних методів виявлення аномалій.

Класичні SIEM-правила зазвичай базуються на сигнатурному або евристичному підході: якщо подія відповідає певному шаблону, генерується сповіщення. Однак такий підхід погано масштабується та недостатньо ефективний проти нових або малопомітних атак. Тому дедалі більшого поширення набувають методи виявлення аномалій, що доповнюють або частково замінюють статичні правила.

На найпростішому рівні аномалії можуть визначатися статистичними методами – через відхилення від середнього значення, плаваючі середні, стандартні відхилення чи сезонні моделі для часових рядів. Такі підходи добре працюють для простих метрик (кількість логінів, обсяг трафіку) та можуть бути реалізовані без складних обчислювальних ресурсів.

Окремий клас методів пов’язаний з UEBA – аналітикою поведінки користувачів і сутностей. UEBA-системи, інтегровані з SIEM або побудовані поверх нього, формують поведінкові профілі користувачів, облікових записів, пристроїв, а потім за допомогою алгоритмів машинного навчання виявляють відхилення від звичного патерну (нестандартні години доступу, нетипові обсяги переданих даних, незвичні комбінації дій тощо). Таким чином, UEBA-будова дозволяє переходити від

простих сигнатурних правил до контекстного виявлення інсайдерських загроз, компрометації облікових записів та латерального руху в мережі.

Узагальнюючи, оптимізація правил SIEM і впровадження сучасних методів виявлення аномалій передбачають поєднання кількох рівнів: регулярний тюнінг правил, їхнє збагачення контекстом, автоматизовану перевірку якості детекцій і інтеграцію алгоритмів машинного навчання (зокрема UEBA). Такий комплексний підхід дозволяє одночасно зменшити кількість хибнопозитивних спрацювань і підвищити ймовірність виявлення складних атак, що є критично важливим для ефективної роботи SOC.

1.3 Аналіз існуючих SIEM-рішень щодо можливостей оптимізації правил та виявлення аномалій

У межах дослідження проблеми оптимізації правил SIEM важливо визначити, наскільки різні платформи підтримують можливості гнучкого тюнінгу, кореляції подій та виявлення аномальної активності. Це дозволяє оцінити, які інструменти є найбільш ефективними для зниження хибнопозитивних спрацювань та підвищення точності детекції. Оскільки на ринку домінують як комерційні enterprise-рішення, так і відкриті платформи, доцільно провести порівняльний аналіз трьох найбільш поширених SIEM-систем - Wazuh, Splunk та IBM QRadar - з точки зору їхніх механізмів кореляції, можливостей розширення правил та підтримки сучасних методів виявлення аномалій.

Wazuh - це відкритий SIEM/XDR-рішення, яке використовує агентську модель збору подій та аналіз на Wazuh Server [2]. Кореляція виконується в analysis engine, де на потоки подій послідовно застосовуються декодери та ruleset. Декодери розбирають сирі логи на поля, а правила, описані у XML, спрацьовують за комбінацією умов: значення полів, регулярні вирази, частота подій за певний інтервал часу, збіг за IP, користувачем, назвою процесу тощо.

Для тюнінгу використовуються:

1. окремі файли з rule groups та локальний файл local_rules.xml, який

дозволяє перевизначати або вимикати стандартні правила без змін базового ruleset;

2. CDB lists (white/black lists) для винятків та довірених об’єктів;

3. власні декодери для специфічних логів;

4. інтеграція з OpenSearch Anomaly Detection для побудови аномалій на основі часових рядів (наприклад, нетиповий обсяг подій від агента або сервісу).

Wazuh має готові правила для поведінкового виявлення (наприклад, підозріла активність процесів, нетипові зміни конфігурацій), а також підтримує розширення логіки скриптами та зовнішніми інструментами, що робить його гнучким для побудови власної технології оптимізації правил [3].

Wazuh використовує багаторівневу модель обробки подій, де кожен лог проходить чіткий pipeline:

1. Збір подій агентом

Агент отримує логи з таких джерел:

- системні журнали (syslog, auditd, journald);
- логи мережевих сервісів;
- інтегровані модулі (AWS CloudTrail, GCP, Azure, Docker).

2. Декодування (Decoders)

Декодер розбирає сирий лог на структуровані поля. Декодери обробляються у строгій послідовності: від першого match до кінця групи.

3. Застосування правил (Ruleset)

Кожне правило складається з таких блоків:

- match, regex, field matching;
- frequency (кількість подій за часовий інтервал);
- same_field (співпадіння певного поля між подіями);
- if_sid/if_level (кореляція зі спрацюваннями інших правил);
- timeframe;
- cdb_list (співставлення з довіреними/забороненими списками IP, користувачів, шляхів);
- classification та group для ледерів.

Важлива особливість: Wazuh використовує ланцюгову кореляцію через SID –

правило може спрацьовувати лише тоді, коли перед ним уже спрацювало інше правило (попередник).

4. Підвищення рівня інциденту

Правила мають рівень (0–15).

Наприклад, правило SID 5710 -> info, а SID 5715 -> high severity.

5. Локальний тюнінг

local_rules.xml дозволяє:

- вимкнути базове правило (<if_sid>xxx</if_sid><level>0</level>);
- створити власну розширену кореляцію;
- створити винятки через match, regex, ip_address або cdb_list.

Проведений аналіз показує, що Wazuh має гнучку та прозору модель обробки та кореляції подій серед розглянутих SIEM-рішень. Wazuh дозволяє повністю контролювати весь pipeline – від етапу декодування логів до кореляційних правил, включно зі створенням власних ланцюгів SID, CDB-списків, адаптивних винятків та кастомних аномальних детекцій [4]. Відкритий формат правил у XML, можливість будувати складні кореляційні сценарії на основі контексту, частоти подій та поведінкових характеристик, а також інтеграція з OpenSearch Anomaly Detection роблять Wazuh надзвичайно придатним для точного тюнінгу та зниження хибнопозитивних спрацювань.

В свою чергу у Splunk ES основою кореляції є correlation searches - заплановані або реального часу пошуки по індексованих даних, які за виконання заданих умов створюють notable events. Умови правил задаються пошуковою мовою SPL, що дозволяє будувати складні багатокрокові детекції (об’єднання різних sourcetype, join по полях, агрегування, статистика). Для зниження шуму Splunk ES впроваджує risk-based alerting (RBA): окремі correlation searches генерують risk events з певною вагою, які складаються в risk index; вже поверх суми ризиків створюються high-fidelity risk notables, що зменшує кількість одиничних сповіщень. Сучасні методи виявлення аномалій у Splunk реалізуються через Machine Learning Toolkit, моделі для часових рядів, кластеризації, а також окремий продукт UEBA, який доповнює SIEM поведінковим аналізом користувачів та сутностей.

Splunk Enterprise Security використовує індексаційну архітектуру, де всі події проходять етапи збору, нормалізації та кореляції на основі SPL-запитів та моделей даних:

1. Збір та індексація подій

Splunk приймає логи через forwarder або HEC із таких джерел:

- syslog, Windows EventLog, мережеві пристрої;
- хмарні сервіси (AWS, GCP, Azure).

2. Нормалізація (CIM — Common Information Model)

Splunk уніфікує події через CIM, що дозволяє правилам працювати однаково для різних джерел:

- Authentication,
- Network Traffic,
- Processes,
- Endpoint Activity тощо.

3. Кореляційні правила (Correlation Searches)

Основний механізм детекцій у Splunk — це SPL-запити, які виконуються за розкладом. Вони дозволяють:

- агрегувати події (stats),
- знаходити патерни в часових рядах,
- корелювати події між різними джерелами (join, transaction),
- застосовувати умови частоти (аналог frequency у Wazuh).

Кореляційний пошук, який повернув результат, створює notable event із рівнем важливості.

4. Механізми оптимізації й зниження шуму

Splunk підтримує кілька ключових інструментів тюнінгу:

- Suppression rules - блокування конкретних повторюваних/безпечних подій.
- Risk-Based Alerting (RBA) - замість шумних окремих подій генерація risk events, які акумулюються у risk score; notable створюється тільки після перевищення порогу ризику.

- Threat Intelligence Matching - автоматичне співставлення з TI-фідами.
- Machine Learning Toolkit та UEBA - виявлення поведінкових аномалій [5].

Отже, Splunk Enterprise Security є високопродуктивним SIEM-рішенням із широкими можливостями кореляції подій, аналітики та поведінкового виявлення аномалій. Його сильні сторони полягають у використанні SPL як універсального механізму побудови правил, підтримці CIM-моделей для нормалізації логів, гнучкості correlation searches, а також наявності інструментів зниження шуму, таких як Risk-Based Alerting та suppression-логіка. Крім того, Splunk пропонує інтегровані ML-засоби та UEBA-рішення, що розширюють можливості для виявлення складних аномалій та інсайдерської активності.

Водночас архітектура Splunk базується на інтенсивному використанні індексації та пошукових обчислень, що суттєво підвищує вимоги до інфраструктури. Кореляція в Splunk повністю прив’язана до SPL-запитів, тому ефективність тюнінгу правил значною мірою залежить від кваліфікації інженера. Крім того, кастомізація, машинне навчання та UEBA потребують окремих ліцензій, що робить Splunk дорогим рішенням для невеликих або середніх організацій.

У межах дослідження оптимізації правил SIEM Splunk демонструє високу аналітичну потужність, але значна складність і вартість розгортання знижують його практичну привабливість у порівнянні з більш відкритими та гнучкими платформами. І останній інструмент - QRadar. За кореляцію подій тут відповідає Custom Rules Engine (CRE), який в реальному часі проганяє усі events та flows через набір вбудованих і користувацьких правил. Правила складаються з rule tests (перевірок по полях подій, категоріях, геолокації, референсних множинах) та умов по частоті й часових вікнах. Для спрощення тюнінгу використовуються building blocks - багаторазові фрагменти логіки, які можна включати в різні правила, а також reference sets для зберігання списків IP, користувачів, хостів.

Для виявлення аномалій QRadar має окремий Anomaly Detection Engine (ADE), який будує базові лінії для мережевого трафіку або інших метрик і генерує offenses при відхиленні від історичної норми (наприклад, обсяг трафіку користувача на 75 %

вище середнього за два місяці) [6]. Рішення позиціонується як enterprise-класу з розвинутими можливостями інтеграції та SOAR, але вимагає окремих ліцензій, аплайнсів та навчання персоналу, що для середньої компанії значно підвищує поріг входу.

QRadar будує логіку детекцій на основі CRE (Custom Rules Engine), який у реальному часі аналізує потоки *events* та *flows*. Кореляція виконується через декларативні rule tests і reference sets:

1. **Збір і нормалізація подій**

QRadar приймає події з:

- syslog, Windows EventLog;
- мережевих пристроїв (firewalls, IPS, routers);
- хмарних сервісів;
- NetFlow/IPFIX (flows).

QRadar автоматично:

- нормалізує події через DSM (Device Support Modules);
- визначає категорії подій за QID (QRadar Identifier);
- уніфікує поля (source IP, username, magnitude, threat categorizations).

(DSM-аналітика є ключовою, бо саме вона робить події структурованими перед подальшим аналізом)

2. **Кореляційний механізм CRE (Custom Rules Engine)**

CRE застосовує правила в реальному часі. Кожне правило складається з tests:

- property tests (значення конкретного поля);
- threshold tests (frequency — кількість подій за проміжок часу);
- sequence tests (послідовність подій);
- geographical tests;
- building blocks tests (перевірки на основі багатократного reuse умов);
- reference set tests (співпадіння зі списками IP, user, hostname).

Це аналог match/regex/frequency/if_sid у Wazuh, але у вигляді окремих логічних блоків.

3. **Building Blocks (аналог rule groups у Wazuh)**

Building Blocks - це шаблонні умови, які можна використовувати у різних правилах, наприклад:

- "Privileged Users"
- "Remote Access Traffic"
- "Known Bad IPs"
- "High-Risk Ports"

Вони не створюють інцидентів, але спрощують створення складних правил.

4. **Reference Sets та Reference Tables**

   QRadar використовує:

   - Reference Sets - списки значень (IP, URL, user), які оновлюються автоматично;
   - Reference Tables - таблиці для зберігання структурованих даних.

   Це функціональний аналог CDB lists у Wazuh, але більш масштабований. Приклади використання:

   - whitelist/blacklist;
   - countries of interest;
   - suspicious usernames;
   - anomaly baselines.

   Reference Sets можуть оновлюватись автоматично через правила або зовнішні TI-фіди.

5. **Offenses — система інцидентів (аналог "підвищення рівня" у Wazuh)**

   QRadar не використовує рівні 0–15, але має:

   - magnitude (оцінка загрози);
   - credibility (довіра до джерела);
   - relevance (важливість активу).

   Комбінація цих параметрів впливає на те, чи подія стане offense.

6. **Тюнінг та оптимізація**

   QRadar має кілька важливих інструментів:

   - Rule Tuning — enable/disable, зміна тестів, винятки;

- Anomaly Detection Engine (ADE) — baseline-поведінка для мережевого трафіку та активності;
- Reference Set Auto-Update — динамічне навчання списків;
- Flow Analytics — аналіз нетипових патернів у NetFlow/IPFIX.

QRadar відомий тим, що частину аномалій може виявляти без ML, виключно через baseline аналіз.

IBM QRadar є багатоскладовим корпоративним SIEM-рішенням із розвиненою системою реального часу для кореляції подій (CRE), використанням building blocks та reference sets для масштабованого тюнінгу правил і вбудованим механізмом baseline-аналітики (Anomaly Detection Engine). Його сильними сторонами є автоматична нормалізація логів через DSM-модулі, підтримка послідовнісних кореляцій, можливість операцій із потоками NetFlow/IPFIX та формування offenses на основі кількох факторів (magnitude, credibility, relevance).

Разом із тим, QRadar має низку обмежень, важливих у контексті завдання оптимізації правил. Побудова складних детекцій вимагає використання власної системи rule tests, що робить процес тюнінгу менш гнучким порівняно з відкритими системами. Значна частина функціональних можливостей - включно з аналітикою поведінки, машинним навчанням та поглибленою кореляцією - доступна лише у розширених ліцензіях або окремих модулях. Крім цього, QRadar потребує значних ресурсів інфраструктури та належить до найбільш дорогих SIEM-рішень на ринку.

Порівняння трьох SIEM-рішень показує суттєві відмінності у підходах до кореляції, оптимізації правил та виявлення аномалій. Splunk забезпечує найбільш гнучкі пошукові можливості завдяки SPL та розвиненим моделям даних, але вимагає значних обчислювальних ресурсів і дорогих ліцензій, що ускладнює масштабування й тюнінг правил. IBM QRadar пропонує сильний CRE-движок із sequence-tests, reference sets та baseline-аналітикою, проте його екосистема закрита й менш гнучка в кастомізації, а впровадження є дорогим і складним. Wazuh, на відміну від них, надає повністю відкриту, прозору і гнучку модель кореляції, у якій можна змінювати, комбінувати та оптимізувати правила без ліцензійних обмежень. Завдяки локальним правилам, кастомним декодерам, CDB-lists та інтеграції з anomaly detection

рішеннями Wazuh забезпечує значно більше можливостей для глибокого тюнінгу та зниження хибнопозитивних спрацювань, що робить його оптимальним інструментом для дослідження та практичної реалізації оптимізації правил SIEM. Порівняння можливостей інструментів у таблиці 1.1:

Таблиця 1.1.

Порівняння інструментів Wazuh, Splunk, QRadar

Критерій	**Wazuh**	**Splunk**	**QRadar**
Тип рішення	Open-source	Пропрієтарний	Пропрієтарний
Модель обробки подій	Декодери + XML ruleset + SID-кореляція	SPL-пошук + CIM-моделі	CRE-tests + building blocks
Глибина кастомізації	Максимальна: декодери, правила, винятки	Висока, але через SPL та індексацію	Середня, залежить від CRE
Тюнінг правил	local_rules.xml, CDB-lists, кастомні кореляції	suppression, RBA, SPL tuning	rule tuning, reference sets
Виявлення аномалій	інтеграція з OpenSearch AD, behavior rules	ML Toolkit, UEBA	ADE baseline, limited ML
Придатність для оптимізації правил SIEM	Повна гнучкість і прозорість	Висока, але дорога та складна	Висока, обмежена логікою CRE
Хибнопозитиви – можливість зниження	Висока: CDB-lists, кастомні кореляції, enrichment	Висока: RBA, ML	Середня: baseline + reference sets

Висновки до 1 розділу

У першому розділі проведено комплексне дослідження проблеми оптимізації правил у SIEM-системах та проаналізовано ключові фактори, що впливають на

якість виявлення аномалій і рівень хибнопозитивних спрацювань у роботі SOC. Встановлено, що значна частина сповіщень, які генерують сучасні SIEM-рішення, не є реальними інцидентами безпеки та створюють суттєве операційне навантаження. За результатами профільних звітів, обсяг шумових і хибнопозитивних алертів у середньостатистичному SOC може перевищувати половину всього потоку сповіщень, а у деяких випадках - досягати двох третин [7]. Це обумовлює ризик пропуску критичних атак, формує явище alert fatigue та потребує системної оптимізації правил кореляції.

Аналіз підходів до оптимізації правил SIEM показав, що ефективне зменшення хибнопозитивних спрацювань досягається шляхом поєднання кількох методів: регулярного тюнінгу правил, їх збагачення контекстною інформацією, застосування ризик-орієнтованих моделей, автоматизованої перевірки якості детекцій та впровадження сучасних методів виявлення аномалій. Особливе місце займають UEBA-рішення та алгоритми машинного навчання, які дозволяють виявляти нетипову поведінку користувачів і систем навіть у відсутності сигнатур або чітко визначених шаблонів атак.

Порівняльний аналіз трьох популярних SIEM-рішень - Wazuh, Splunk Enterprise Security та IBM QRadar - продемонстрував суттєві відмінності у їхніх механізмах кореляції, можливостях кастомізації та ефективності оптимізації правил. Splunk забезпечує потужні аналітичні можливості, однак потребує значних ресурсів і глибоких знань SPL. QRadar має розвинуту модель кореляції CRE та baseline-аналітику, проте є дорогим, складним у розгортанні та менш гнучким у модифікації правил. На відміну від них, Wazuh пропонує відкриту, прозору і найголовніше - безкоштовну систему кореляції, що дозволяє глибоко модифікувати правила, створювати власні декодери, використовувати кастомні списки (CDB-lists), будувати складні ланцюги SID-кореляцій та інтегрувати зовнішні засоби для виявлення аномалій. Це робить Wazuh найбільш придатним рішенням для дослідження та практичної реалізації технології оптимізації правил SIEM у рамках цієї кваліфікаційної роботи.

Узагальнюючи результати аналізу, можна стверджувати, що оптимізація

правил SIEM є критичним напрямком для підвищення точності виявлення кіберзагроз і зниження операційного навантаження на SOC. Wazuh, завдяки своїй відкритості, гнучкості та широким можливостям кастомізації, є найоптимальнішою платформою для подальших досліджень і впровадження технології оптимізації правил, що і визначає вибір цього інструмента як основної платформи у наступних розділах роботи.

2 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ОПТИМІЗАЦІЇ ПРАВИЛ У WAZUH

2.1. Призначення та функції Wazuh у виявленні аномальної активності

Wazuh є відкритою платформою класу SIEM/XDR, призначеною для централізованого збору подій, кореляції, аналізу загроз та виявлення аномалій у інформаційних системах. Як було зазначено в попередніх розділах, архітектура Wazuh поєднує агентську модель збору даних, серверну систему обробки подій та механізми аналітики, що дозволяє будувати комплексні сценарії детекції і реагування. Основним призначенням Wazuh у контексті виявлення аномальної активності є формування поведінкових і контекстних сигналів, виявлення нетипових дій користувачів, процесів та мережевої активності, а також забезпечення можливості оптимізації правил для зменшення кількості хибнопозитивних спрацювань.

Ключовим механізмом виявлення аномалій у Wazuh є багаторівнева система кореляції правил. Аналіз подій виконується в analysis engine, де на кожний лог послідовно застосовуються декодери та правила. Правила дозволяють:

- визначати частотні аномалії (кількість подій за інтервал часу);
- виявляти неконсистентні або сумнівні послідовності подій;
- аналізувати зміни у привілеях, системних налаштуваннях та конфігураціях;
- зіставляти події зі списками довірених або заборонених значень (CDB-lists);
- проводити контекстну кореляцію на основі попередніх SID-спрацювань.

SID-кореляція є одним з найбільш гнучких механізмів Wazuh - завдяки їй можливо будувати сценарії на кшталт «успішний логін після декількох невдалих», «підозріла активність з IP, який раніше не зустрічався» або «послідовність системних змін, що не характерні для користувача».

Також Wazuh містить вбудовані поведінкові правила, що дозволяють визначати відхилення у діях користувачів і системних процесів, зокрема:

- нетипові дії адміністратора;

- аномальні зміни у системних файлах;
- підозрілу активність процесів і служб;
- нестандартні шаблони автентифікації та доступу;
- різкі скачки у кількості або структурі логів.

Ці правила можуть бути комбіновані з власними декодерами або CDB-списками для формування складних і точних поведінкових кореляцій.

За рахунок інтеграції з OpenSearch Anomaly Detection, Wazuh підтримує автоматичне:

- побудування базових поведінкових моделей (baseline);
- аналіз часових рядів для метрик (обсяг логів, частота системних операцій, активність користувачів);
- виявлення подій, що виходять за статистичні межі норми;
- виявлення довгострокових та малопомітних аномалій, що не підпадають під класичні сигнатури.

Такі механізми є особливо корисними у випадках, коли конкретний тип атаки або поведінки не може бути описаний статичним правилом.

Комбінація цих джерел створює багатовимірну картину активності, дозволяючи виявляти не тільки прямі порушення, але й непрямі аномалії (рисунок 2.1)

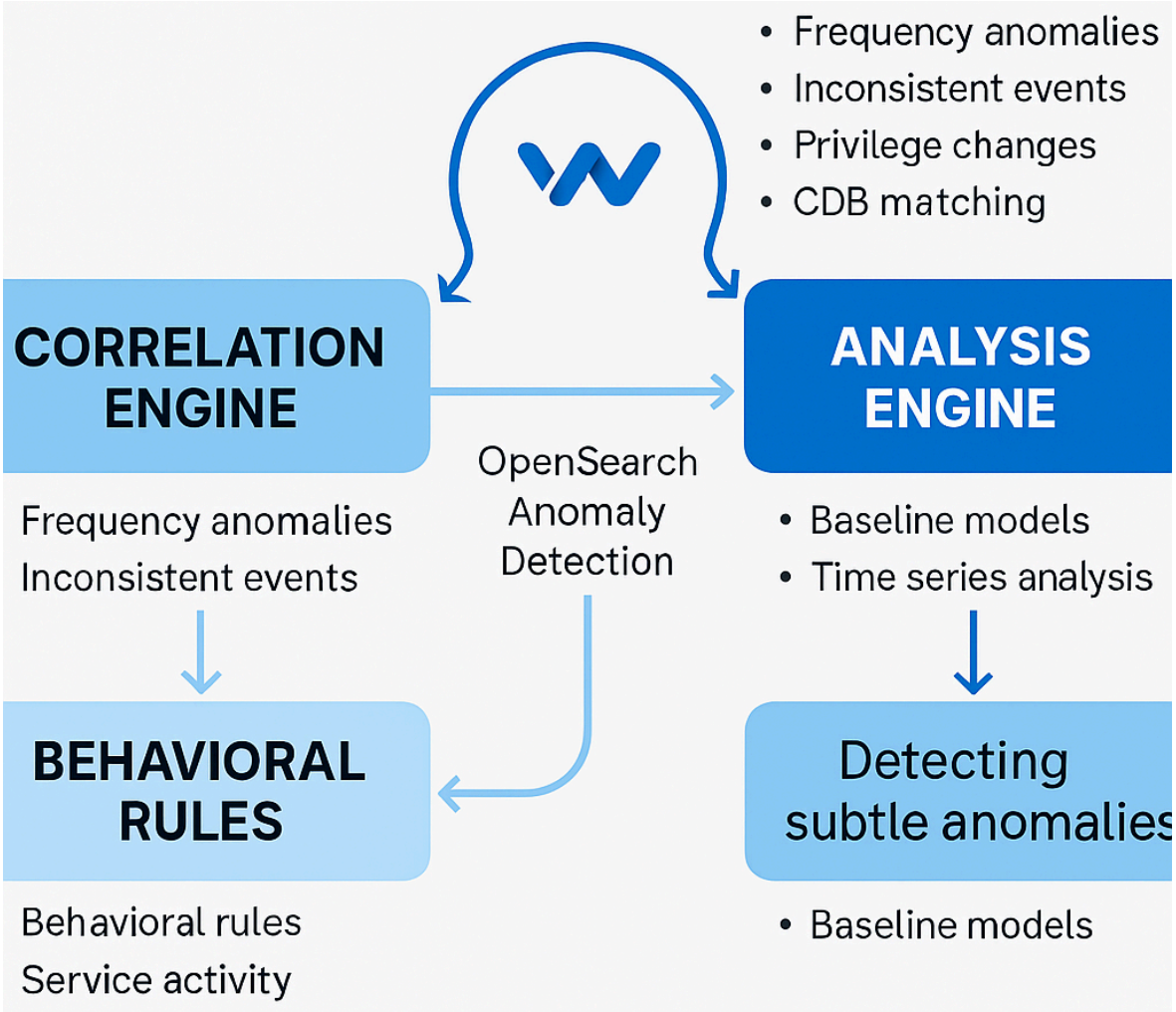

Рисунок 2.1 - Інструменти виявлення аномальної активності

2.2 Методи зниження хибнопозитивних виявлень у Wazuh

Зменшення кількості хибнопозитивних спрацювань є одним з ключових завдань при побудові ефективної системи моніторингу безпеки, і Wazuh надає широкий спектр механізмів, які дозволяють підвищити точність детекцій та адаптувати правила під особливості конкретного середовища. Основою оптимізації у Wazuh є можливість гнучкого керування правилом кореляції шляхом їх перевизначення у файлі local_rules.xml. Такий підхід дає змогу вимикати або модифікувати стандартні правила, змінювати рівні серйозності та уточнювати умови спрацювання, що дозволяє враховувати специфіку інфраструктури та зменшувати кількість шумових подій. Одним з найбільш дієвих засобів тюнінгу є використання CDB-lists, які дозволяють формувати довірені чи заборонені списки користувачів,

IP-адрес, шляхів або процесів і динамічно застосовувати їх у правилах для відсікання завідомо легітимної активності. Наприклад, типова ситуація коли PostgreSQL генерує лог, схожий на помилку автентифікації, хоча реально це не інцидент. Створимо правило, яке вимикає (level="0") спрацювання правила 2501, якщо в події зустрічається рядок "getFileContent". Або інший приклад - правило, що підвищує рівень події для конкретного сценарію: невдала спроба SSH-автентифікації (if_sid 5716) з IP-адреси 1.1.1.1 (бо всередині обмеженої мережі IP є небажаним/підозрілим).

local_rules.xml

```
<group name="local,syslog,sshd">

  <!--
  Dec 10 01:02:02 host sshd[1234]: Failed none for root from 1.1.1.1 port 1066 ssh2
  -->
  <rule id="100001" level="5">
    <if_sid>5716</if_sid>
    <srcip>1.1.1.1</srcip>
    <description>sshd: authentication failed from IP 1.1.1.1.</description>
    <group>authentication_failed,pci_dss_10.2.4,pci_dss_10.2.5,</group>
  </rule>

  <rule id="120001" level="0">
    <if_sid>2501</if_sid>
    <description>PostgreSQL false positive: 'getFileContent' detected as authentication failure</description>
    <match>getFileContent</match>
  </rule>

  <rule id="110001" level="0">
    <if_sid>81801</if_sid>
    <field name="data.srcip">93.170.0.252</field>
    <description>Ignore OpenVPN login from trusted IP 93.170.0.252</description>
  </rule>
```

Рис. 2.2 - Приклади правил у local_rules.xml

Важливу роль у зниженні хибнопозитивних сповіщень відіграє коректне налаштування частотних умов. Пороги спрацювання, часові інтервали та прив’язка до конкретних полів (наприклад, користувача або джерела) дозволяють адаптувати правила до поведінкових патернів організації і уникати реагування на природні піки активності. Оскільки Wazuh підтримує підхід SID-кореляції, система здатна враховувати історію подій і підвищувати точність детекцій шляхом аналізу послідовностей, тому одноразові або випадкові події не призводять до генерації інциденту, а значущі послідовності, наприклад серія невдалих логінів із подальшим успішним входом, можуть бути об’єднані у зважену кореляційну модель.

Не менш важливим аспектом оптимізації є коректне декодування вхідних логів. Неправильно спроєктований декодер може призводити до некоректного

зіставлення полів або помилкового визначення значень, що автоматично збільшує кількість хибнопозитивних спрацювань. Тому у Wazuh передбачена можливість створення власних декодерів, уточнення регулярних виразів та оптимізації структури полів для максимально точного аналізу подій. Чим якісніше побудований декодер, тим нижчий рівень шумових подій у ruleset. Наприклад, декодер agent-upgrade, показаний на рисунку 2.2, використовує складний регулярний вираз для виділення трьох окремих полів стану оновлення агента. У разі неправильної побудови regex або зміни формату службового повідомлення, значення полів можуть бути визначені некоректно, що призведе до спрацювання правил на основі хибних даних. Це демонструє, що якість декодерів безпосередньо впливає на рівень хибнопозитивних сповіщень у системі.

```
<decoder name="wazuh">
  <prematch>^wazuh: </prematch>
</decoder>

<decoder name="agent-buffer">
  <parent>wazuh</parent>
  <prematch offset="after_parent">^Agent buffer:</prematch>
  <regex offset="after_prematch">^ '(\S+)'.</regex>
  <order>level</order>
</decoder>

<decoder name="agent-upgrade">
  <parent>wazuh</parent>
  <prematch offset="after_parent">^Upgrade procedure |^Custom installation </prematch>
  <regex offset="after_prematch">on agent (\d\d\d)\s\((\S+)\):\s(\w+)</regex>
  <order>agent.id, agent.name, status</order>
</decoder>

<decoder name="agent-upgrade">
  <parent>wazuh</parent>
  <regex>aborted:\s(\.+)$|failed:\s(\.+)$|lost:\s(\.+)$</regex>
  <order>error</order>
</decoder>

<decoder name="agent-upgrade">
  <parent>wazuh</parent>
  <regex>started.\sCurrent\sversion:\sWazuh\s(\.+)$</regex>
  <order>agent.cur_version</order>
</decoder>

<decoder name="agent-upgrade">
  <parent>wazuh</parent>
  <regex>succeeded.\sNew\sversion:\sWazuh\s(\.+)$</regex>
  <order>agent.new_version</order>
</decoder>
```

Рис. 2.3 - Приклади декодерів Wazuh

Важливою перевагою Wazuh є підтримка поведінкових та контекстних правил, які дозволяють виявляти аномалії, що не вписуються у статичну сигнатурну модель. Такі правила можуть враховувати нетипові дії користувачів, аномальні зміни системних файлів, нестандартну активність процесів і служб або різкі відхилення у

структурі логів. Використання поведінкових правил дає змогу виявляти складні аномалії, що не можуть бути описані звичайними умовами match чи regex.

Додатково Wazuh інтегрується з OpenSearch Anomaly Detection, що дозволяє аналізувати часові ряди та формувати baseline нормальної активності. На основі статистичних моделей система здатна визначати відхилення, які не підпадають під жодне існуюче правило, забезпечуючи додатковий рівень точності та виявлення нетипових довготривалих аномалій. Це особливо актуально у випадках, коли поведінка користувачів чи систем змінюється поступово і не може бути описана окремою сигнатурою.

Узагальнюючи, зниження хибнопозитивних спрацювань у Wazuh досягається завдяки комбінації коректного декодування, гнучкого тюнінгу правил, застосування контекстних моделей, поведінкового аналізу та, за можливості, інтеграції зі статистичними методами виявлення аномалій. Завдяки такій архітектурі Wazuh забезпечує можливість точного й адаптивного контролю інцидентів безпеки та формує підгрунтя для побудови високоякісної системи моніторингу.

2.3. Механізм обробки подій у Wazuh: декодери, правила та Local Rules як основа кореляції

Механізм обробки подій у Wazuh базується на послідовності етапів декодування, нормалізації логів, застосування правил кореляції та додаткових локальних правил, які дозволяють адаптувати систему до специфіки організації. Декодери відповідають за структуризацію сирих логів, правила забезпечують детекцію аномальних та небажаних подій, а local_rules.xml використовується для тюнінгу, придушення шуму та уточнення логіки реагування. Така архітектура забезпечує гнучкість і точність, необхідну для зниження хибнопозитивних спрацювань і формування поведінкових моделей активності.

Декодери є першим елементом обробки подій. На цьому етапі Wazuh аналізує сирий текст логів та за допомогою регулярних виразів виділяє ключові поля, які надалі використовуються у правилах. Наприклад, у файлі 0005-wazuh_decoders.xml

визначені декодери для службових подій агента Wazuh (рисунок 2.3). Один із них розбирає повідомлення про помилки оновлення агента за допомогою складного регулярного виразу, який виділяє значення aborted, failed та lost. У разі некоректного визначення груп такого виразу система може неправильно витлумачити статус оновлення та згенерувати хибнопозитивний інцидент. Інший декодер для upgrade-процедур витягує ідентифікатор агента, назву хоста та статус операції. Якісна робота декодерів безпосередньо впливає на точність кореляції, оскільки правила базуються саме на тих полях, які виділені під час декодування.

0005-wazuh_decoders.xml

```
<!--
  -  Wazuh decoders
  -  Created by Wazuh, Inc.
  -  Copyright (C) 2015, Wazuh Inc.
  -  This program is a free software; you can redistribute it and/or modify it under the terms of GPLv2.
-->

<decoder name="wazuh">
  <prematch>^wazuh: </prematch>
</decoder>

<decoder name="agent-buffer">
  <parent>wazuh</parent>
  <prematch offset="after_parent">^Agent buffer:</prematch>
  <regex offset="after_prematch">^ '(\S+)'.</regex>
  <order>level</order>
</decoder>

<decoder name="agent-upgrade">
  <parent>wazuh</parent>
  <prematch offset="after_parent">^Upgrade procedure |^Custom installation </prematch>
  <regex offset="after_prematch">on agent (\d\d\d)\s\((\S+)\):\s(\w+)</regex>
  <order>agent.id, agent.name, status</order>
</decoder>

<decoder name="agent-upgrade">
  <parent>wazuh</parent>
  <regex>aborted:\s(\.+)$|failed:\s(\.+)$|lost:\s(\.+)$</regex>
  <order>error</order>
</decoder>

<decoder name="agent-upgrade">
  <parent>wazuh</parent>
  <regex>started.\sCurrent\sversion:\sWazuh\s(\.+)$</regex>
  <order>agent.cur_version</order>
</decoder>

<decoder name="agent-upgrade">
  <parent>wazuh</parent>
  <regex>succeeded.\sNew\sversion:\sWazuh\s(\.+)$</regex>
  <order>agent.new_version</order>
</decoder>

<decoder name="agent-restart">
  <parent>wazuh</parent>
  <prematch offset="after_parent">^Invalid remote configuration:</prematch>
  <regex offset="after_prematch">^ '(\S+)'.</regex>
  <order>module</order>
</decoder>

<decoder name="fim-state">
  <parent>wazuh</parent>
  <prematch offset="after_parent">^FIM DB: </prematch>
  <plugin_decoder offset="after_prematch">JSON_Decoder</plugin_decoder>
</decoder>
```

Рисунок 2.4 - Декодери 0005-wazuh_decoders.xml

Після етапу декодування на подію послідовно накладається ruleset. Стандартні правила Wazuh охоплюють широкий спектр системної та користувацької активності, однак їхня універсальність часто призводить до генерації небажаних або характерних для конкретного середовища подій. Для вирішення цієї проблеми використовується local_rules.xml, який дозволяє перевизначати, вимикати або уточнювати стандартні правила. Наприклад, у правилах було вимкнено (level="0")

спрацювання SID 2501, яке помилково інтерпретувало SQL-процедуру getFileContent як невдалу автентифікацію, що зменшило кількість шумових подій у логах PostgreSQL. Інше правило приглушувало регулярні OpenVPN-логіни з довіреної IP-адреси, що запобігало хибним сповіщенням у випадках легітимного доступу. Подібні винятки дозволяють уникнути постійних спрацювань на очікувану поведінку, тим самим підтримуючи чистоту аналітичної картини (рисунок 2.4).

Рис. 2.5 - Правило 2501

Особливе місце у local_rules.xml займають правила для роботи із службовими подіями Linux. Наприклад, події PAM session opened/closed для користувача root на тестових або dev-середовищах були винесені до окремих правил і переведені у рівень 0, оскільки ці події не мали безпекового характеру. Також локальні правила дозволили уникати спрацювань при модифікації файлів .gitconfig під час автоматизованих деплой-процесів, що для CI/CD-середовищ є типовою поведінкою (рисунок 2.5).

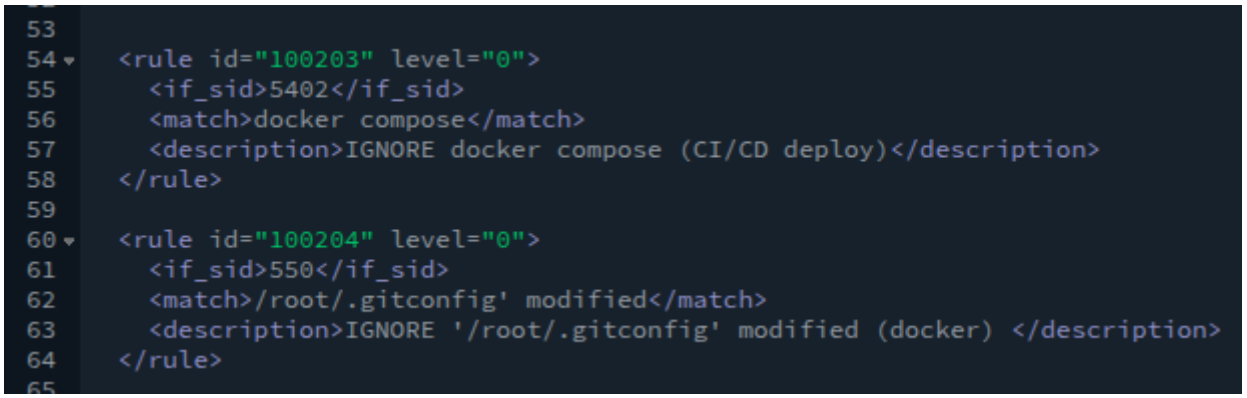

Рис. 2.6 - Правило для .gitconfig та CI/CD

Крім XML-правил, у Wazuh можлива розширена кореляція подій за допомогою зовнішніх скриптів та пошукових запитів до OpenSearch. Прикладом є механізм

детекції підключень поза робочим часом. Запит періодично аналізує події за допомогою скрипта, який обчислює годину за київським часом. Якщо час підключення припадає на проміжок до 8:00 або після 20:00, подія вважається аномальною. На рисунку 2.6 можна побачити інформацію про даний скрипт у веб-профілі сервера Wazuh, а також графічне відображення спрацювань. Такий підхід дозволяє реалізувати поведінкові моделі, які не можуть бути охоплені статичними XML-правилами, і є потужним інструментом адаптації Wazuh до організаційних політик доступу.

Рис. 2.7 - Дашборд скрипта аномальних підключень

Таким чином, обробка подій у Wazuh формує повноцінний конвеєр: сирі логи спочатку проходять через декодери, які структурують дані, після чого правила ruleset здійснюють первинну детекцію, а local_rules.xml забезпечує тюнінг, усунення хибнопозитивів і адаптацію системи до конкретного середовища. Розширені скрипти та зовнішні запити дозволяють будувати поведінкові моделі та складні сценарії аномального доступу. Усе це формує гнучку і точну кореляційну систему, яка є основою для подальшої оптимізації правил у наступних розділах роботи.

Висновки до 2 розділу

У другому розділі було проведено детальний аналіз механізмів виявлення аномальної активності та методів оптимізації правил у Wazuh, що дало змогу оцінити гнучкість та ефективність цієї платформи у практичних умовах.

Дослідження показало, що Wazuh не обмежується статичними сигнатурними підходами, а реалізує багаторівневу архітектуру кореляції подій, де ключову роль відіграють якісні декодери, поведінкові правила та адаптивні механізми обробки контексту. Саме коректно побудовані декодери забезпечують точне структурування інформації, на основі якої формуються подальші детекції, і є критичним чинником у зниженні кількості хибнопозитивних спрацювань.

Аналіз методів оптимізації правил засвідчив, що Wazuh підтримує широкий спектр технічних можливостей для адаптації системи до специфічних особливостей середовища. Використання локальних правил у файлі local_rules.xml дозволяє ефективно придушувати шумові події, коригувати рівні важливості інцидентів, формувати винятки для окремих сервісів і джерел, а також будувати більш точні кореляційні ланцюги. Додаткові можливості, такі як SID-кореляція та контекстне збагачення подій, забезпечують здатність системи враховувати історію попередніх спрацювань і виявляти складні поведінкові аномалії. Інтеграція з OpenSearch Anomaly Detection розширює аналітичний потенціал Wazuh, дозволяючи працювати з часовими рядами та baseline-моделями, що значно підвищує точність визначення нетипових відхилень у системній активності.

Окреме значення має можливість реалізації скриптових кореляцій, що дає змогу формувати правила, які виходять за межі XML-логіки. Зокрема, приклад з виявленням підключень поза робочим часом демонструє, що Wazuh може опрацьовувати події у взаємодії з OpenSearch, враховувати часові зони, поведінкові шаблони та організаційні політики доступу. Такий підхід поєднує класичну кореляцію з адаптивною аналітикою та суттєво зменшує ймовірність пропуску критичних інцидентів.

3 ТЕХНОЛОГІЯ ОПТИМІЗАЦІЇ ПРАВИЛ У WAZUH (ПРАКТИЧНА ЧАСТИНА)

3.1 Розробка скрипта/правила fast-travel у Wazuh (алгоритм, логіка та модель виявлення аномалії)

У межах даної роботи технологія оптимізації правил у Wazuh реалізується не шляхом прямого спрощення або видалення існуючих правил, а через зміну підходу до формування умов їх спрацювання. Оптимізація досягається за рахунок винесення складної контекстної та часової логіки за межі статичного ruleset у окремий аналітичний рівень, після чого правила Wazuh використовуються лише для фінальної зваженої кореляції та генерації алертів. Таким чином, правила перестають реагувати на одиничні події і починають працювати з попередньо збагаченими та агрегованими даними, що безпосередньо знижує кількість хибнопозитивних спрацювань. У практичній частині роботи для демонстрації можливостей оптимізації правил у Wazuh було розроблено окремий скрипт fast-travel, призначений для виявлення аномальних підключень користувачів до корпоративного VPN. Ідея полягає у виявленні ситуацій, коли той самий обліковий запис за короткий проміжок часу підключається з різних країн, що фізично неможливо або малоймовірно. Така модель відповідає класичній аномалії типу "impossible travel" і дозволяє зосередитися на потенційно компрометованих облікових записах, не генеруючи сповіщення при кожному звичайному вході з віддаленого розташування.

Скрипт реалізований мовою Python та інтегрований у Wazuh як зовнішня інтеграція, що запускається для подій з правилом OpenVPN-логіну у файлі local_rules. На вхід він отримує JSON алерт через stdin або шлях до файлу, що забезпечує сумісність з різними режимами запуску інтеграцій. На етапі парсингу

скрипт виділяє ключові параметри: IP-адресу джерела підключення, ім'я користувача, інформацію про агента та оригінальний текст лог-запису. У разі, якщо користувача визначити не вдається, для подальшої обробки використовується технічне значення "unknown". Після виділення IP-адреси виконується GeoIP-запит до зовнішнього сервісу ip-api.com з отриманням коду країни, назви країни та статусу запиту (рисунок 3.1). Для приватних адрес (локальні сегменти, внутрішні VPN-тунелі) скрипт не викликає зовнішній сервіс і примусово повертає фіксовані значення країни, що дозволяє уникати хибних визначень геолокації та зменшує кількість зайвих запитів. На основі отриманих даних формується перша подія типу enrichment з інтеграцією "geoip-openvpn", яка містить користувача, IP-адресу, країну, код країни, опис вихідного правила та повний текст оригінального лог-запису. Ця подія записується у окремий лог-файл у форматі JSON і надалі обробляється Wazuh за допомогою спеціального декодера та правила з локального ruleset, що забезпечує збагачення стандартних VPN-подій географічною інформацією.

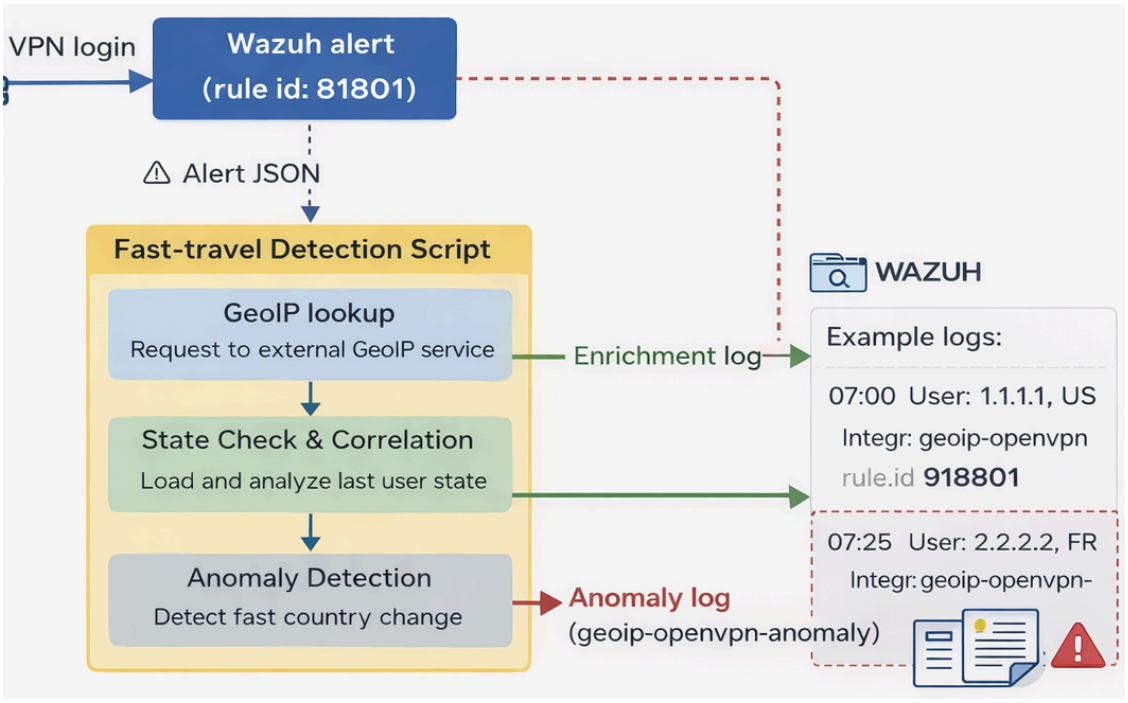

Рис. 3.1 - Схема скрипта

Ключовим елементом алгоритму fast-travel є підтримка стану для кожного користувача. Скрипт зберігає інформацію про останню відому країну, IP-адресу та час підключення у файлі geoip-openvpn-users.csv. При кожному новому логіні таблиця завантажується у вигляді словника, після чого для поточного користувача

визначається попередній код країни та мітка часу попередньої сесії. Якщо для користувача відсутні історичні дані або GeoIP-запит не завершився успішно, подальша кореляція не виконується, що дозволяє уникати спрацювань у разі неповних або некоректних вхідних даних.

Модель виявлення аномалії базується на порівнянні поточного коду країни з попереднім та обчисленні часової різниці між сесіями. Якщо попередня країна відрізняється від поточної, а інтервал між підключеннями менший за заданий поріг COUNTRY_CHANGE_MIN_INTERVAL_SEC (у реалізації використовується значення 7200 секунд, що відповідає двом годинам), скрипт вважає таку зміну геолокації підозрілою. У цьому випадку формується окрема подія з інтеграцією "geoip-openvpn-anomaly", яка містить користувача, попередній та поточний коди країн, часові мітки обох сесій, різницю в секундах та IP-адресу підключення. Додаткове поле reason отримує значення "country_changed_too_fast", що дозволяє однозначно ідентифікувати тип виявленої аномалії у Wazuh.

З технічної точки зору такий підхід реалізує станну, а не лише подієву кореляцію. Замість того щоб піднімати сповіщення при кожному логіні з нової країни, скрипт аналізує послідовність підключень одного користувача в часі. Це суттєво знижує кількість хибнопозитивних спрацювань у порівнянні з наївними правилами, які спрацьовують на будь-який "нетиповий" IP-діапазон. Легітимні сценарії, коли співробітник подорожує, але між підключеннями минає значний час, не потрапляють під критерій fast-travel, тоді як підозрілі випадки швидкої зміни країн протягом годин чи хвилин потрапляють у фокус SOC. Додатково відкидання приватних адрес та подій без коректно визначеного коду країни запобігає генерації алертів у випадках технічних збоїв або внутрішньої маршрутизації. Завдяки такій реалізації скрипт fast-travel виступає прикладом того, як за допомогою зовнішньої інтеграції та локальних правил Wazuh можна побудувати складну поведінкову модель поверх базового ruleset. Він не замінює стандартні механізми детекції, а доповнює їх кореляцією на основі історії підключень користувачів, що дозволяє одночасно зменшити кількість шумових подій і підвищити ймовірність вчасного виявлення компрометації облікового запису. У подальших підпунктах розділу буде

розглянуто інтеграцію цієї логіки з декодерами та правилами Wazuh, а також оцінено вплив впровадженого алгоритму на загальний рівень хибнопозитивних спрацювань у системі.

3.2. Інтеграція fast-travel алгоритму у Wazuh: декодери, pipeline, local_rules та генерація alert

Початковою точкою pipeline є стандартний алерт Wazuh для подій OpenVPN, який ідентифікується rule.id 81801 (рисунок 3.2). Саме ці події скрипт приймає на вхід у форматі JSON: у коді передбачено два сценарії отримання payload - через stdin або через шлях до файлу, який передається першим аргументом командного рядка. Це зроблено для сумісності з різними механізмами запуску інтеграцій у Wazuh, коли система може передавати дані або потоково, або як файл з alert-пакетом [8].

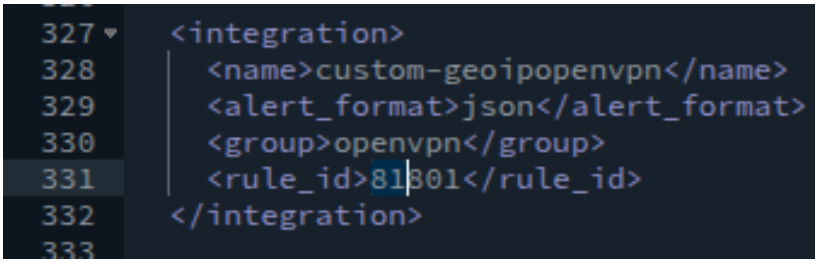

Рис. 3.2 - Правило 81801

Після прийому alert JSON скрипт нормалізує структуру даних, оскільки в Wazuh payload інтеграцій може відрізнятися залежно від способу виклику. У реалізації використовується логіка “alert = ar.get('alert') or ar”, тобто скрипт працює як з “плоским” об’єктом (rule/agent на верхньому рівні), так і з вкладеним форматом, який часто використовується Active Response. Далі відбувається витяг ключових параметрів для кореляції: з блоку data (або альтернативно з syscheck) читаються srcip та srcuser. Якщо цих значень немає, скрипт переходить до резервного парсингу з full_log, використовуючи регулярні вирази для виділення IPv4 та імені користувача у

форматі, характерному для VPN-логів. Таким чином pipeline не “ламається” через різні формати подій, а забезпечує стабільну нормалізацію вхідних даних. Якщо користувача визначити не вдається, задається технічне значення unknown, щоб не втрачати подію повністю і мати можливість діагностики, як було зазначено в попередньому розділі.

```
158  def main():
159      debug_log("=== script start ===")
160      ensure_paths()
161
162      ar = read_alert_json()
163      if not ar:
164          debug_log("No alert JSON, exiting early.")
165          return
166
167      alert = ar.get("alert") or ar
168      debug_log(f"alert keys: {list(alert.keys())}")
169
170      agent = alert.get("agent", {})
171      rule = alert.get("rule", {})
172      data_block = alert.get("data", {}) or alert.get("syscheck", {}) or {}
173      full_log = alert.get("full_log", "") or alert.get("decoder", {}).get("raw", "") or ""
174
175      srcip = data_block.get("srcip")
176      user = data_block.get("srcuser")
177      if not srcip:
178          srcip = extract_ipv4(full_log or "")
179      if not user:
180          user = extract_user(full_log or "")
181      if not user:
182          user = "unknown"
```

Рис. 3.3 - Витяг ключових параметрів

Наступний етап - збагачення GeoIP. Для публічних адрес виконується HTTP-запит до зовнішнього сервісу ip-api.com з отриманням полів status, query, country, countryCode (таймаут запиту 5 секунд). Для приватних IP застосовано окрему гілку: якщо IP визначено як приватний (через ipaddress.ip_address(ip).is_private), скрипт не викликає зовнішній GeoIP і примусово формує “успішну” відповідь із фіксованою країною та кодом. Такий механізм одночасно вирішує дві практичні задачі: прибирає випадкові помилки геолокації для внутрішніх сегментів і зменшує “шум” та залежність від зовнішнього сервісу, що прямо впливає на рівень хибнопозитивів, пов’язаних із некоректним визначенням країни. Після отримання GeoIP-результату скрипт формує та записує JSON-подію типу enrichment. У коді вона має поле integration зі значенням geoip-openvpn, містить

timestamp ts, rule_id первинного правила, інформацію про агента та блок data з нормалізованими полями користувача, IP, статусу геозапиту та країни. Додатково зберігаються original_description та original_full_log (рисунок 3.3), що дозволяє SOC мати контекст первинного спрацювання без необхідності вручну шукати сирі логи. Запис подій виконується у кастомний лог-файл у форматі “один JSON на рядок”, що є оптимальним для подальшого читання Wazuh як custom log source.

```
    event = {
        "integration": "geoip-openvpn",
        "ts": ts,
        "rule_id": rule.get("id"),
        "agent": agent,
        "data": {
            "user": user,
            "srcip": srcip,
            "geo_status": geo.get("status"),
            "country": geo.get("country"),
            "country_code": geo.get("countryCode")
        },
        "original_description": rule.get("description"),
        "original_full_log": full_log or alert.get("full_log")
    }
    write_json_event(event)

    curr_cc = geo.get("countryCode")
    if geo.get("status") != "success" or not curr_cc:
        debug_log("Geo status not success or no country code; stop.")
        return
```

Рис. 3.4 - Запис статусу користувача у логи

На цьому місці в pipeline підключається другий рівень Wazuh - обробка власних подій інтеграції. Схема така: Wazuh читає JSON-лог інтеграції як окреме джерело (custom log), декодер розпізнає ці записи та розкладає їх на поля, а правила у local_rules.xml (рисунок 3.5) виконують кореляцію вже по збагачених значеннях.

```
80
81  <!-- Anomaly (зміна країни надто швидко) -->
82  <rule id="919812" level="12">
83    <field name="integration">geoip-openvpn-anomaly</field>
84    <field name="reason">country_changed_too_fast</field>
85    <description>OpenVPN user changed country too fast</description>
86    <options>no_full_log</options>
87  </rule>
88
89  <!-- DEBUG: запасне правило на сирий матч (видали, коли 919812 стабільно світиться) -->
90  <rule id="919813" level="12">
91    <match>integration:geoip-openvpn-anomaly</match>
92    <match>reason:country_changed_too_fast</match>
93    <description>Anomaly matched by raw match</description>
94    <options>no_full_log</options>
95  </rule>
```

Рис. 3.5 - Запис статусу користувача у логи

У ruleset це відображено групою geoip,openvpn, де є правила для “heartbeat” інтеграції (фіксація факту, що запис містить integration=geoip-openvpn), а також правила для аномалії на основі integration=geoip-openvpn-anomaly і reason=country_changed_too_fast. Тобто генерація алерта на аномалію відбувається не в скрипті напряму, а через Wazuh ruleset - скрипт лише створює подію з потрібними полями, а SIEM застосовує до неї контрольовані правила, рівні важливості та маршрутизацію.

Алгоритм fast-travel реалізовано як станну кореляцію на рівні скрипта. Після enrichment-події скрипт перевіряє, чи GeoIP завершився успішно та чи отримано countryCode. Далі завантажується локальний стан з CSV-файлу geoip-openvpn-users.csv, який виступає простою “базою” останнього відомого місця користувача. Для поточного користувача зі стану зчитуються last_country_code та last_ts. Якщо попередня країна існує і відрізняється від поточної, обчислюється delta = ts - prev_ts. Якщо delta менший за поріг COUNTRY_CHANGE_MIN_INTERVAL_SEC (у реалізації 7200 секунд), формується окрема JSON-подія аномалії з integration=geoip-openvpn-anomaly. Вона містить prev_country_code, curr_country_code, prev_ts, curr_ts, delta_sec і reason=country_changed_too_fast (рисунок 3.6). Після цього стан користувача оновлюється: у CSV записуються поточна країна, IP і timestamp.

```
curr_cc = geo.get("countryCode")
if geo.get("status") != "success" or not curr_cc:
    debug_log("Geo status not success or no country code; stop.")
    return

state = load_state()
prev = state.get(user)
if prev:
    try:
        prev_cc = prev.get("last_country_code") or ""
        prev_ts = int(prev.get("last_ts") or "0")
    except Exception:
        prev_cc, prev_ts = "", 0
    if prev_cc and prev_cc != curr_cc and prev_ts > 0:
        delta = ts - prev_ts
        if delta < COUNTRY_CHANGE_MIN_INTERVAL_SEC:
            debug_log(f"ANOMALY user={user} prev={prev_cc}@{prev_ts} curr={curr_cc}@{ts} delta={delta}")
            anomaly = {
                "integration": "geoip-openvpn-anomaly",
                "ts": ts,
                "agent": agent,
                "data": {
                    "user": user,
                    "prev_country_code": prev_cc,
                    "curr_country_code": curr_cc,
                    "prev_ts": prev_ts,
                    "curr_ts": ts,
                    "delta_sec": delta,
                    "srcip": srcip
                },
                "reason": "country_changed_too_fast"
            }
            write_json_event(anomaly)
```

Рис. 3.6 - Розрахунок аномалії

Важливо, що навіть якщо аномалії немає, стан все одно оновлюється - це дозволяє моделі “переїхати” разом з користувачем і не генерувати зайві алерти у майбутньому, коли зміна країни була легітимною, але відбулася давно. Скрипт виводить алерт в чат через API, що зображено на рисунку 3.7.

Рис. 3.7 - Алерт

Таким чином, інтеграція fast-travel у Wazuh будує повний приклад “технології оптимізації правил” у практичному сенсі: первинна подія (rule.id 81801) перетворюється у збагачену структуру, над якою виконується станна кореляція, після чого результати повертаються у Wazuh у вигляді окремих JSON-подій. Декодери

забезпечують коректне розбиття цих подій на поля, локальні правила задають критерії й рівні серйозності, а SIEM-ланцюжок формує фінальний alert лише для справді аномальних випадків. Саме ця модель демонструє, як оптимізація детекції зменшує шум і підвищує точність реагування без втрати видимості легітимної активності.

Висновки до 3 розділу

У результаті реалізованого підходу стандартні правила Wazuh більше не спрацьовують безпосередньо на події VPN-логіну, а використовуються як елемент контрольованої генерації алертів на основі поведінкової моделі, що є практичним прикладом оптимізації правил SIEM.

Реалізований механізм інтеграції fast-travel алгоритму у Wazuh демонструє практичний підхід до оптимізації правил SIEM шляхом винесення складної логіки кореляції за межі статичного ruleset та перенесення її у керований скриптовий рівень. Побудований pipeline забезпечує стабільну обробку подій незалежно від формату вхідного alert, виконує нормалізацію ключових полів, збагачення геоконтекстом та формування структурованих подій, придатних для подальшої кореляції у Wazuh. Такий підхід дозволяє зберегти базові правила детекції, не порушуючи стандартну архітектуру платформи, та водночас розширити її аналітичні можливості. Використання станної моделі fast-travel з накопиченням історичних даних користувача та часовим порогом зміни країни забезпечує суттєве зниження хибнопозитивних спрацювань у порівнянні з одноразовими сигнатурними правилами. Аномалія формується лише у випадку, коли виконано сукупність умов - зміна геолокації, короткий часовий інтервал та підтверджена послідовність подій. Генерація фінального алерта здійснюється на рівні local_rules Wazuh, що дозволяє централізовано керувати рівнями критичності, політиками реагування та подальшою обробкою інцидентів у SOC. У результаті fast-travel алгоритм виступає прикладом ефективної технології оптимізації правил, яка зменшує шум, підвищує точність детекції та демонструє гнучкість Wazuh у реалізації контекстно-орієнтованих сценаріїв виявлення аномальної активності.

3.3. Рекомендації щодо оптимізації правил у Wazuh для зниження хибнопозитивних виявлень

Проведений аналіз та практична реалізація алгоритму fast-travel у середовищі Wazuh демонструють, що ефективне зниження кількості хибнопозитивних спрацювань можливе лише за умови комплексного підходу до оптимізації правил кореляції. Ключовою рекомендацією є відмова від генерації алертів на основі одиничних подій без урахування контексту та історії активності. Як показано у розділах 3.1 та 3.2, навіть формально підозрілі події, такі як зміна геолокації VPN-користувача, можуть бути легітимними за відсутності часових або поведінкових відхилень. Тому правила SIEM повинні будуватися навколо послідовностей і моделей поведінки, а не окремих лог-записів.

Важливим аспектом оптимізації є винесення складної логіки з правил у окремі обчислювальні етапи pipeline. Реалізація fast-travel алгоритму показує доцільність використання користувацьких скриптів для виконання станної кореляції, збагачення подій та обчислення часових інтервалів. Такий підхід дозволяє уникнути перевантаження ruleset складними умовами, зменшує кількість умовних перевірок у local_rules.xml та підвищує прозорість логіки детекції. Wazuh у цьому випадку використовується як механізм контрольованої генерації алертів на основі вже підготовлених і нормалізованих подій, що суттєво знижує ймовірність хибнопозитивних спрацювань.

Окрему увагу слід приділяти якості декодування та нормалізації даних. Практика показала, що нестабільні або неповні поля у подіях (відсутність srcuser, неоднозначні формати логів) є однією з причин шумових алертів. Рекомендовано використовувати багаторівневу стратегію вилучення даних: спочатку з структурованих полів, а у разі їх відсутності - з резервного парсингу full_log. Такий підхід дозволяє забезпечити стабільність pipeline і зменшити кількість помилкових детекцій, пов’язаних не з реальною аномалією, а з особливостями форматів логування.

Додатковою рекомендацією є використання enrichment-подій як проміжного рівня між сирими логами та алертами. Впровадження окремих JSON-подій із чітким

маркером integration дозволяє будувати локальні правила виключно на основі збагачених і перевірених даних. Це спрощує тюнінг правил, дозволяє чітко відокремити службові події від аномалій і знижує кількість помилкових спрацювань, які виникають через неоднозначність вихідних логів. Практична реалізація fast-travel показує, що генерація алертів повинна залишатися відповідальністю ruleset, а не самого скрипта, що забезпечує єдину модель керування рівнями важливості, маршрутизацією та подальшим реагуванням у SOC [9].

Нарешті, рекомендовано застосовувати адаптивні часові пороги та механізми збереження стану для поведінкових сценаріїв. Фіксовані правила без урахування історії користувача неминуче призводять до повторюваних хибнопозитивних алертів. Збереження попереднього контексту, як показано на прикладі CSV-стану у fast-travel алгоритмі, дозволяє системі “навчатися” легітимній поведінці користувачів і зменшувати кількість повторних спрацювань у стабільних сценаріях. Це є критично важливим для VPN-доступу, віддаленої роботи та гібридних середовищ, де поведінка користувачів має високу варіативність.

Таким чином, оптимізація правил у Wazuh повинна базуватися на поєднанні контекстної кореляції, збагачення подій, станної логіки та чіткого розмежування ролей між скриптами та ruleset. Запропоновані підходи дозволяють суттєво знизити рівень хибнопозитивних спрацювань без втрати чутливості до реальних аномалій та формують практичну основу для побудови ефективного й масштабованого SOC.

ВИСНОВКИ

У кваліфікаційній роботі проведено комплексне дослідження проблеми оптимізації правил у SIEM-системах у контексті виявлення аномальної активності та зниження кількості хибнопозитивних спрацювань. Обґрунтовано актуальність теми, яка зумовлена стрімким зростанням обсягів телеметрії безпеки, ускладненням IT-інфраструктур та підвищенням навантаження на центри моніторингу безпеки. Встановлено, що надмірна кількість хибнопозитивних алертів негативно впливає на ефективність роботи SOC, знижує швидкість реагування на інциденти та створює ризик пропуску реальних атак.

У першому розділі роботи досліджено природу хибнопозитивних спрацювань у SIEM-системах та їхній вплив на операційну діяльність SOC. Проаналізовано сучасні підходи до оптимізації правил, зокрема ручний тюнінг, контекстне збагачення подій, ризик-орієнтовані моделі та методи виявлення аномалій. Проведено порівняльний аналіз можливостей оптимізації правил у SIEM-рішеннях Wazuh, Splunk Enterprise Security та IBM QRadar. За результатами аналізу встановлено, що Wazuh вирізняється відкритою та прозорою моделлю кореляції, широкими можливостями кастомізації правил, використанням декодерів, local rules і CDB-lists, що робить його придатною платформою для практичної реалізації технології оптимізації правил.

У другому розділі проаналізовано методи та засоби оптимізації правил безпосередньо у Wazuh. Розглянуто призначення та функціональні можливості платформи у виявленні аномальної активності, зокрема механізми декодування логів, SID-кореляцію, частотні умови, контекстні та поведінкові правила. Показано, що зниження хибнопозитивних спрацювань у Wazuh досягається шляхом коректної побудови декодерів, точного налаштування правил у local_rules.xml, використання винятків і списків довіри, а також застосування поведінкових і статистичних методів виявлення аномалій. Окрему увагу приділено ролі кореляції подій та багаторівневого pipeline обробки логів як основи підвищення точності детекцій.

У третьому розділі реалізовано практичну частину роботи, у межах якої розроблено та впроваджено алгоритм fast-travel для виявлення аномальної географічної активності користувачів у Wazuh. Запропонований підхід базується на збагаченні подій автентифікації даними GeoIP, побудові станної моделі поведінки користувача та аналізі часових інтервалів між змінами країни доступу. Алгоритм інтегровано у pipeline Wazuh із використанням декодерів, custom log source та local rules, що дозволяє розмежувати етапи збагачення даних і генерації алертів. Показано, що відокремлення логіки обчислення аномалії від механізму сповіщення дозволяє зменшити кількість хибнопозитивних алертів і забезпечити адаптацію моделі до легітимної поведінки користувачів. Результати сучасних досліджень свідчать, що перспективним напрямом подальшого розвитку SIEM-систем, зокрема Wazuh, є поєднання класичних правил кореляції з методами машинного навчання для поведінкового аналізу подій безпеки, що дозволяє автоматично адаптувати моделі детекції до змін інфраструктури та додатково знижувати рівень хибнопозитивних спрацювань у SOC [10].

На основі проведених досліджень сформульовано рекомендації щодо оптимізації правил у Wazuh для зниження хибнопозитивних виявлень. Запропоновані рекомендації охоплюють питання побудови декодерів, використання контекстної та поведінкової кореляції, застосування stateful-алгоритмів, розділення enrichment та alerting-логіки, а також регулярного тюнінгу правил відповідно до змін інфраструктури та поведінкових патернів. Таким чином, у роботі досягнуто поставленої мети - розроблено технологію оптимізації правил у SIEM-системі Wazuh, яка дозволяє підвищити точність виявлення аномальної активності та зменшити кількість хибнопозитивних спрацювань. Отримані результати мають практичну цінність та можуть бути використані фахівцями з кібербезпеки під час побудови та вдосконалення систем моніторингу безпеки в реальних SOC-середовищах.

ПЕРЕЛІК ПОСИЛАНЬ

1. Cybersecurity and Cyberwar: What Everyone Needs to Know. Behl A., Behl K., Inc. 2017. URL: https://global.oup.com/academic/product/cybersecurity-and-cyberwar-9780190867473 (дата звернення: 26.11.2025).

2. Wazuh Documentation: Rules, Decoders and Correlation. URL: https://documentation.wazuh.com/current/user-manual/ruleset/index.html (дата звернення: 26.11.2025).

3. Guide to Intrusion Detection and Prevention Systems (IDPS). Behl A., Behl K., Inc. 2007. URL: https://doi.org/10.6028/NIST.SP.800-94 (дата звернення: 26.11.2025).

4. Zero Trust Architecture. NIST Special Publication 800-207, Inc. 2020. URL: https://doi.org/10.6028/NIST.SP.800-207 (дата звернення: 26.11.2025).

5. Splunk Enterprise Security Architecture and Correlation Searches. URL: https://docs.splunk.com/Documentation/ES (дата звернення: 26.11.2025).

6. IBM QRadar SIEM: Architecture and Offense Correlation. URL: https://www.ibm.com/docs/en/qsip (дата звернення: 26.11.2025).

7. Gartner. Market Guide for Security Information and Event Management. URL: https://www.gartner.com/en/documents/4017484 (дата звернення: 26.11.2025).

8. Wazuh Integrations and Custom Scripts.. URL: https://documentation.wazuh.com/current/user-manual/capabilities/integration.html (дата звернення: 26.11.2025)

9. SOC Survey: The Impact of Alert Fatigue on Incident Response. SANS Institute. URL: https://www.sans.org/white-papers/alert-fatigue (дата звернення: 26.11.2025).

10. S. A. Chamkar *et al. Improving Threat Detection in Wazuh Using Machine Learning. Sensors, 2025.* URL: https://www.mdpi.com/2624-800X/5/2/34?utm_source=chatgpt.com (дата звернення: 26.11.2025).