Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра

на тему: «Проєкт підсистеми моніторингу інформаційної безпеки на базі платформи Wazuh для мережної інфраструктури організації»

Виконав: студент 4-го курсу групи КБ-22-1
спеціальності 125 «Кібербезпека»
_____ Сергій ПРИЛИПКО
Керівник: старший викладач кафедри КІ та КБ
_____ Олександра ПОКОТИЛО
Рецензент: завідувач кафедри КІ та КБ, к.т.н., доцент
_____ Андрій ЄФІМЕНКО

Житомир – 2026

Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки
Спеціальність 125 «Кібербезпека»
Освітня програма «Кібербезпека»

ЗАТВЕРДЖУЮ
Завідувач
кафедри комп'ютерної
інженерії та кібербезпеки
Андрій Єфіменко
2 лютого 2026 р.

ЗАВДАННЯ
на випускну кваліфікаційну роботу бакалавра

Студент: Прилипко Сергій Дмитрович

Тема роботи: «Проєкт підсистеми моніторингу інформаційної безпеки на базі платформи Wazuh для мережної інфраструктури організації»

затверджена Наказом університету від 2 лютого 2026 р. № 58/с

Термін здачі студентом закінченої роботи 19 червня 2026 р.

Вихідні дані роботи: метою роботи є проєктування підсистеми моніторингу безпеки, дослідження детектування кіберінцидентів та аналізу подій на базі платформи Wazuh SIEM.

Об'єктом дослідження є процес моніторингу та аналізу подій інформаційної безпеки в комп'ютерних мережах.

Предметом дослідження є методи та засоби побудови SIEM-систем на базі платформи Wazuh із інтеграцією мережного обладнання Cisco.

Обладнання та програмне забезпечення:

1. Мережні пристрої: Cisco 2911, switch Catalyst 3560
2. Ubuntu Server з налаштованим Wazuh
3. Linux PC з необхідним програмним забезпеченням
4. Windows 10 PC з необхідним програмним забезпеченням

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Фальковський І. Г.	02.02.26	07.02.26
2	Фальковський І. Г.	11.02.26	14.02.26
3	Фальковський І. Г.	16.02.26	20.02.26

Студент _____ Сергій ПРИЛИПКО

Керівник _____ Олександра ПОКОТИЛО

Календарний план

№ з/п	Назви етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Аналіз отриманого завдання та уточнення мети і задач кваліфікаційної роботи	02.02.2026 – 09.02.2026	Виконано
2	Добір та аналіз науково-технічних джерел за темою SIEM-систем і платформи Wazuh	10.02.2026 – 16.02.2026	Виконано
3	Обґрунтування вимог до системи та розробка сценаріїв детектування інцидентів	23.12.2025 – 25.01.2026	Виконано
4	Проєктування архітектури системи моніторингу та схеми інтеграції з обладнанням Cisco	21.02.2026 – 26.02.2026	Виконано
5	Розробка алгоритмів обробки подій та створення кастомних правил детектування атак	28.02.2026 – 04.03.2026	Виконано
6	Розгортання, налаштування та верифікація компонентів лабораторного стенду	04.03.2026 – 07.03.2026	Виконано
7	Моделювання тестових кібератак та аналіз процесів їх виявлення системою	08.03.2026 – 23.03.2026	Виконано
8	Оцінка ефективності підсистеми моніторингу та узагальнення результатів досліджень	03.04.2026 – 18.04.2026	Виконано
9	Остаточне оформлення тексту пояснювальної записки та підготовка презентації	22.04.2026 – 22.05.2026	Виконано

Студент _____ Сергій ПРИЛИПКО

Керівник _____ Олександра ПОКОТИЛО

АНОТАЦІЯ

У кваліфікаційній роботі бакалавра розглянуто питання проєктування та реалізації системи моніторингу й управління подіями інформаційної безпеки на базі відкритої SIEM-платформи Wazuh із використанням мережного обладнання Cisco. Роботу присвячено аналізу сучасних методів виявлення кіберзагроз, побудові правил кореляції подій та практичному тестуванню ефективності системи на реальних сценаріях атак.

Пояснювальна записка містить 89 сторінок, 41 рисунків, 29 таблиць. Список використаних джерел налічує 20 найменувань та розміщений на трьох сторінках.

Метою кваліфікаційної роботи є розробка SIEM-системи на базі платформи Wazuh для централізованого збору, аналізу та кореляції подій інформаційної безпеки з використанням мережного обладнання Cisco як джерела телеметрії.

У роботі проаналізовано сучасний стан систем моніторингу безпеки, проведено порівняльний аналіз існуючих SIEM-рішень, спроєктовано архітектуру системи на базі Wazuh та виконано її розгортання у тестовому середовищі. Інтеграцію мережного обладнання Cisco реалізовано через протокол Syslog. На вузлах-жертвах встановлено агенти Wazuh з підсистемою Sysmon для детального збору телеметрії. Розроблені правила кореляції покривають техніки матриці MITRE ATT&CK: мережну розвідку (T1046), брутфорс RDP (T1110.001), ексфільтрацію через FTP (T1048.003), закріплення у реєстрі (T1547.001), витяг облікових даних Mimikatz (T1003), Pass-the-Hash (T1550.002), Reverse Shell (T1059), маніпуляцію обліковими записами (T1098) та очищення журналів (T1070.001). Тестування підтвердило 100-відсотковий рівень виявлення всіх дев'яти змодельованих сценаріїв атак.

Ключові слова: КІБЕРБЕЗПЕКА, SIEM, WAZUH, CISCO, МОНІТОРИНГ ПОДІЙ БЕЗПЕКИ, MITRE ATT&CK, SYSMON, ВИЯВЛЕННЯ ЗАГРОЗ, КОРЕЛЯЦІЯ ПОДІЙ.

					ККІКБ.КР.Б – 125 – 12 – ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Прилипко С. Д.		10.06.26	Проєкт підсистеми моніторингу інформаційної безпеки на базі платформи Wazuh для мережної інфраструктури організації	Літ.	
Керівник		Покотило О. А.		10.06.26		Арк. 4	Аркушів 103
Реценз.		Єфіменко А. А.		10.06.26		ФІКТ, Гр. КБ-22-1	
Н. контр.		Покотило О. А.		10.06.26			
Затверд.		Єфіменко А. А.		10.06.26			

ANNOTATION

The bachelor's thesis addresses the design and implementation of a security information and event management (SIEM) system based on the open-source Wazuh platform integrated with Cisco network equipment. The work is devoted to the analysis of modern threat detection methods, the development of event correlation rules, and practical testing of the system's effectiveness using real-world attack scenarios.

The explanatory note contains 89 pages, 41 figures, and 29 tables. The list of references includes 20 sources.

The aim of the qualification work is to develop a SIEM system based on the Wazuh platform for centralized collection, analysis, and correlation of security events, using Cisco network equipment as a telemetry source.

The work analyzes the current state of security monitoring systems and provides a comparative analysis of existing SIEM solutions. The system architecture was designed and deployed in a virtual test environment. Cisco network equipment integration was implemented via Syslog protocol. Wazuh agents with the Sysmon subsystem were deployed on target hosts for detailed telemetry collection. Custom correlation rules were developed to cover MITRE ATT&CK techniques including network reconnaissance (T1046), RDP brute force (T1110.001), FTP exfiltration (T1048.003), registry persistence (T1547.001), Mimikatz credential dumping (T1003), Pass-the-Hash (T1550.002), Reverse Shell (T1059), account manipulation (T1098), and event log clearing (T1070.001). Testing confirmed a 100% detection rate across all nine simulated attack scenarios.

Keywords: CYBERSECURITY, SIEM, WAZUH, CISCO, SECURITY EVENT MONITORING, MITRE ATT&CK, SYSMON, THREAT DETECTION, EVENT CORRELATION.

ЗМІСТ

ВСТУП 8
РОЗДІЛ 1 АНАЛІЗ СУЧАСНОГО СТАНУ ТА ОБҐРУНТУВАННЯ ВИМОГ ДО СИСТЕМИ МОНІТОРИНГУ БЕЗПЕКИ 9
1.1 Аналіз сучасного стану систем моніторингу інформаційної безпеки 9
1.2 Огляд та порівняльний аналіз існуючих SIEM-рішень 10
1.3 Архітектура та принципи функціонування платформи Wazuh 13
1.4 Інструментальні засоби та апартні вимоги до системи 15
1.5 Моделі та сценарії виявлення кіберінцидентів у SIEM системах 17
РОЗДІЛ 2 ПРОЕКТУВАННЯ СИСТЕМИ МОНІТОРИНГУ ТА УПРАВЛІННЯ ПОДІЯМИ БЕЗПЕКИ 23
2.1 Сценарії використання системи моніторингу та управління подіями 23
2.2 Проєктування архітектури та інтеграції Wazuh з обладнанням Cisco 26
2.3 Інформаційне забезпечення та та алгоритми обробки подій 30
2.4 Опис реалізації функціоналу підсистеми збору та аналізу даних 33
2.5 Розробка та обґрунтування кастомних правил детектування кібератак . 36
РОЗДІЛ 3 ПРАКТИЧНІ ДОСЛІДЖЕННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ МОНІТОРИНГУ БЕЗПЕКИ 41
3.1 Опис та архітектура експериментального лабораторного стенду 41
3.2 Моделювання кібератак та аналіз процесів їх детектування 49
3.3 Аналіз результатів та оцінка ефективності підсистеми моніторингу 69
ВИСНОВКИ 83
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 87
ДОДАТКИ 90

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

SIEM – Security Information and Event Management

XDR – Extended Detection and Response

MITRE ATT&CK – Adversarial Tactics, Techniques and Common Knowledge

RDP – Remote Desktop Protocol

FTP – File Transfer Protocol

NTLM – NT LAN Manager

SSH – Secure Shell

TCP – Transmission Control Protocol

UDP – User Datagram Protocol

IP – Internet Protocol

HTTP – Hypertext Transfer Protocol

HTTPS – Hypertext Transfer Protocol Secure

Syslog – стандарт протоколу передачі журналів подій

NetFlow – протокол збору статистики мережного трафіку

DHCP – Dynamic Host Configuration Protocol

WFP – Windows Filtering Platform

SOC – Security Operations Center

IoC – Indicator of Compromise

PCRE – Perl Compatible Regular Expressions

NLA – Network Level Authentication

LAN – Local Area Network

VLAN – Virtual Local Area Network

OSI – Open Systems Interconnection

IDS – Intrusion Detection System

IPS – Intrusion Prevention System

API – Application Programming Interface

JSON – JavaScript Object Notation

XML – Extensible Markup Language

ВСТУП

Актуальність теми. У сучасних умовах стрімкого зростання кіберзагроз забезпечення інформаційної безпеки корпоративних мереж стає критично важливим завданням. Необхідність переходу від пасивних засобів захисту до комплексного моніторингу підтверджується даними ENISA Threat Landscape 2023, згідно з якими кількість інцидентів із несанкціонованим доступом щороку зростає.

Такі рішення широко застосовуються у різних сферах: у банківському секторі – для відстеження шахрайських транзакцій, у державних установах – для моніторингу доступу до критичних ресурсів, на об'єктах критичної інфраструктури – для ранньої ідентифікації кіберфізичних атак. Використання мережного обладнання Cisco як джерела телеметрії в поєднанні з платформою Wazuh SIEM дозволяє побудувати гнучку систему виявлення загроз у реальному часі.

Метою роботи є розробка системи моніторингу та управління подіями інформаційної безпеки на базі платформи Wazuh SIEM із використанням мережного обладнання Cisco.

Об'єктом дослідження є процес моніторингу та аналізу подій інформаційної безпеки в комп'ютерних мережах.

Предметом дослідження є методи та засоби побудови SIEM-систем на базі платформи Wazuh із інтеграцією мережного обладнання Cisco.

Для досягнення поставленої мети необхідно вирішити такі завдання:

– проаналізувати сучасні кіберзагрози та принципи функціонування SIEM-систем;

– дослідити існуючі аналогічні рішення та визначити технічні вимоги до системи, що розробляється;

– спроєктувати архітектуру та реалізувати систему моніторингу подій безпеки на базі платформи Wazuh;

– виконати розгортання системи у тестовому середовищі з підключенням мережної інфраструктури Cisco;

– провести тестування ефективності системи шляхом моделювання реальних кіберінцидентів та аналізу результатів виявлення.

РОЗДІЛ 1 АНАЛІЗ СУЧАСНОГО СТАНУ ТА ОБҐРУНТУВАННЯ ВИМОГ ДО СИСТЕМИ МОНІТОРИНГУ БЕЗПЕКИ

1.1 Аналіз сучасного стану систем моніторингу інформаційної безпеки

У сучасних умовах стрімкого розвитку інформаційних технологій та зростання рівня цифровізації різних сфер діяльності питання забезпечення інформаційної безпеки набуває особливої актуальності. Інформаційні системи підприємств, державних установ та організацій функціонують у складному та динамічному середовищі, що характеризується постійним збільшенням кількості кіберзагроз і ускладненням методів їх реалізації. У зв’язку з цим особливого значення набувають системи моніторингу подій інформаційної безпеки, які дозволяють своєчасно виявляти інциденти та реагувати на них.

На сьогоднішній день сучасний стан систем моніторингу інформаційної безпеки визначається необхідністю обробки великих обсягів даних, що надходять із різноманітних джерел, таких як сервери, мережне обладнання, робочі станції та прикладні системи. За даними аналітичних звітів Європейського агентства з кібербезпеки (ENISA), організації стикаються зі зростанням кількості кіберінцидентів[1], серед яких домінують атаки, пов’язані з витоком даних, шкідливим програмним забезпеченням та порушенням доступності сервісів. Це обумовлює необхідність впровадження централізованих систем збору та аналізу подій безпеки.

Традиційні засоби захисту інформації, такі як міжмережні екрани, антивірусні програми та системи виявлення вторгнень, виконують окремі функції захисту, проте не забезпечують цілісного уявлення про стан інформаційної безпеки. Вони генерують значний обсяг логів, які потребують подальшого аналізу. У більшості випадків такі дані залишаються неструктурованими та не використовуються ефективно, що ускладнює виявлення складних атак. Як зазначається у звітах ENISA, відсутність централізованого моніторингу та кореляції подій є однією з основних проблем сучасних систем кіберзахисту[2].

З метою вирішення зазначених проблем широкого поширення набули системи класу SIEM, які забезпечують централізований збір, зберігання та аналіз подій

інформаційної безпеки. Основною перевагою таких систем є можливість кореляції подій, що надходять з різних джерел, що дозволяє виявляти складні та багаторівневі атаки. Використання SIEM-систем сприяє підвищенню рівня контролю за інформаційною інфраструктурою та дозволяє скоротити час виявлення інцидентів.

Важливою тенденцією сучасного розвитку систем моніторингу є автоматизація процесів аналізу та реагування на інциденти. Використання аналітичних методів та алгоритмів дозволяє виявляти аномалії у роботі систем і поведінці користувачів, що підвищує ефективність виявлення загроз та зменшує навантаження на фахівців з інформаційної безпеки.

Окрему роль у процесі моніторингу відіграє мережне обладнання, яке є важливим джерелом подій безпеки. Мережні пристрої генерують інформацію про трафік, спроби доступу та інші події, що можуть свідчити про потенційні загрози. Інтеграція цих даних у системи моніторингу дозволяє підвищити точність аналізу та ефективність виявлення інцидентів.

Згідно з сучасними дослідженнями, спостерігається перехід від ізольованих засобів захисту до комплексних платформ, що поєднують функції збору, аналізу та реагування на події інформаційної безпеки [6]. Такий підхід дозволяє забезпечити більш ефективний контроль за станом інформаційної безпеки.

Таким чином, сучасні системи моніторингу інформаційної безпеки орієнтовані на централізований аналіз подій та інтеграцію різних джерел даних. Впровадження SIEM рішень є необхідним для забезпечення своєчасного виявлення загроз і підвищення рівня захищеності інформаційних систем

1.2 Огляд та порівняльний аналіз існуючих SIEM-рішень

Системи моніторингу подій інформаційної безпеки сьогодні є обов’язковим елементом будь-якої сучасної IT-інфраструктури. Залежно від масштабу організації та вимог до безпеки, використовуються як комерційні рішення корпоративного рівня, так і відкриті платформи, які дозволяють реалізувати необхідний функціонал з меншими витратами.

Серед комерційних SIEM-систем найбільш поширеними є IBM QRadar, Splunk Enterprise Security та ArcSight. Ці платформи розроблені для роботи у великих інфраструктурах і забезпечують повний цикл обробки подій безпеки – від збору логів до їх аналізу та реагування.

IBM QRadar орієнтований на глибокий аналіз подій та поведінки користувачів. Система використовує механізми кореляції та вбудовану аналітику, що дозволяє виявляти складні атаки. Вона добре масштабується і підтримує інтеграцію з різними джерелами даних, проте потребує значних ресурсів для розгортання та супроводження [8].

Splunk Enterprise Security відомий своєю гнучкістю у роботі з даними. Його часто використовують у центрах моніторингу безпеки (SOC), оскільки він дозволяє обробляти великі обсяги логів у режимі реального часу та будувати власні сценарії аналізу. Основний недолік – висока вартість та складність налаштування, особливо для невеликих організацій [9].

ArcSight є ще одним рішенням корпоративного рівня, яке забезпечує централізований збір та аналіз подій. Його перевагою є зрілі механізми кореляції, але система досить складна в адмініструванні та вимагає досвіду роботи[16].

Паралельно з комерційними платформами активно розвиваються відкриті рішення. Одним із найбільш популярних є Wazuh – платформа з відкритим кодом, яка поєднує функції SIEM. Вона дозволяє здійснювати моніторинг подій, аналіз журналів, контроль цілісності файлів і виявлення вторгнень. Завдяки відкритості та гнучкості Wazuh широко використовується у навчальних і дослідницьких проєктах [3].

Ще одним варіантом є Elastic Security (Elastic SIEM), який базується на стеку Elasticsearch. Це рішення добре підходить для аналізу великих обсягів даних і має зручні засоби візуалізації. Разом із тим, для його ефективного використання необхідні навички роботи з відповідною екосистемою[17].

Останніми роками спостерігається еволюція SIEM-платформ, у напрямку розширення можливостей збору телеметрії та автоматизації аналізу інцидентів. Су-

часні рішення забезпечують глибшу інтеграцію між різними рівнями інфраструктури та що дозволяє більш ефективно виявляти складні загрози та скорочувати час реагування [4].

Для більш наочного порівняння розглянутих систем доцільно представити їх основні характеристики у вигляді таблиці.

Таблиця 1.1

Основні характеристики систем

Критерій	IBM QRadar	Splunk ES	ArcSight	Wazuh	Elastic Security
Тип рішення	Комерційне	Комерційне	Комерційне	Open-source	Частково open-source
Вартість	Висока	Дуже висока	Висока	Низька	Середня
Масштабованість	Висока	Висока	Висока	Середня	Висока
Складність впровадження	Висока	Висока	Висока	Середня	Середня
Гнучкість налаштування	Середня	Висока	Середня	Висока	Висока
Розширений збір телеметрії	Часткова	Часткова	Обмежена	Так	Так
Інтеграція з джерелами	Широка	Широка	Широка	Широка	Широка
Підходить для навчання	Ні	Ні	Ні	Так	Так

Як видно з таблиці, комерційні рішення забезпечують високий рівень функціональності та масштабованості, але їх використання пов’язане зі значними витратами. Водночас відкриті платформи, зокрема Wazuh, дозволяють реалізувати основні функції моніторингу безпеки без значних фінансових вкладень і є більш доступними для впровадження у навчальних або тестових середовищах.

При виборі конкретного рішення важливо враховувати не лише функціональні можливості, але й ресурси організації, рівень підготовки персоналу та необхідність інтеграції з існуючою інфраструктурою. У контексті даної роботи доцільним є використання платформи Wazuh, яка поєднує достатній рівень функціональності, гнучкість налаштування та доступність.

1.3 Архітектура та принципи функціонування платформи Wazuh

Платформа Wazuh є сучасним відкритим рішенням, що реалізує функції SIEM та використовується для централізованого моніторингу подій інформаційної безпеки. Її архітектура побудована таким чином, щоб забезпечити збір, обробку, аналіз та візуалізацію подій з різних джерел у єдиній системі. Завдяки цьому Wazuh дозволяє отримувати цілісне уявлення про стан безпеки інформаційної інфраструктури.

З точки зору архітектури, Wazuh складається з кількох ключових компонентів: менеджера (Wazuh Manager), агентів, що встановлюються на кінцевих пристроях, а також системи зберігання та візуалізації даних на базі OpenSearch та Wazuh Dashboard. Така структура дозволяє масштабувати систему залежно від розміру інфраструктури та кількості джерел даних.

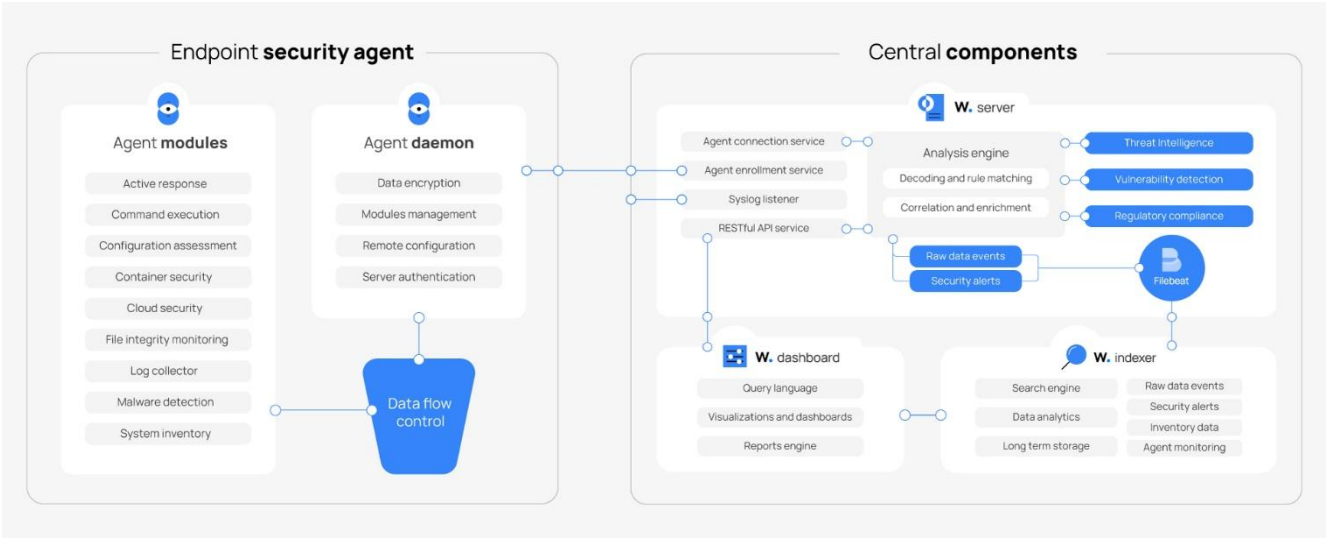


Рисунок 1.1 – Архітектура платформи Wazuh

Агенти Wazuh відповідають за збір інформації з хостів, включаючи системні логи, події безпеки, інформацію про файли та процеси. Зібрані дані передаються на центральний сервер, де відбувається їх попередня обробка та нормалізація. Далі інформація аналізується за допомогою правил кореляції, які дозволяють виявляти підозрілі події та потенційні інциденти безпеки.

Wazuh Manager є центральним елементом системи. Він виконує функції обробки подій, застосування правил безпеки, виявлення аномалій та генерації спові-

щень. Важливою особливістю є підтримка механізмів активного реагування, які дозволяють автоматично виконувати дії у відповідь на інциденти – наприклад, блокування IP-адрес або завершення підозрілих процесів.

Зберігання та аналіз даних у Wazuh реалізується через інтеграцію з пошуково-аналітичною платформою OpenSearch. Це дозволяє виконувати швидкий пошук по великих обсягах логів та будувати аналітичні дашборди для відображення стану безпеки в реальному часі. Візуалізаційний рівень забезпечується через Wazuh Dashboard, який використовується для моніторингу інцидентів та аналізу подій[15].

Принцип роботи платформи базується на моделі централізованого збору та кореляції подій. На першому етапі відбувається збір даних з агентів та зовнішніх джерел, включаючи мережні пристрої Cisco через протокол Syslog[10]. Далі дані проходять нормалізацію, після чого застосовуються правила виявлення інцидентів. У разі відповідності певним критеріям система генерує алерти та передає їх до модуля візуалізації.

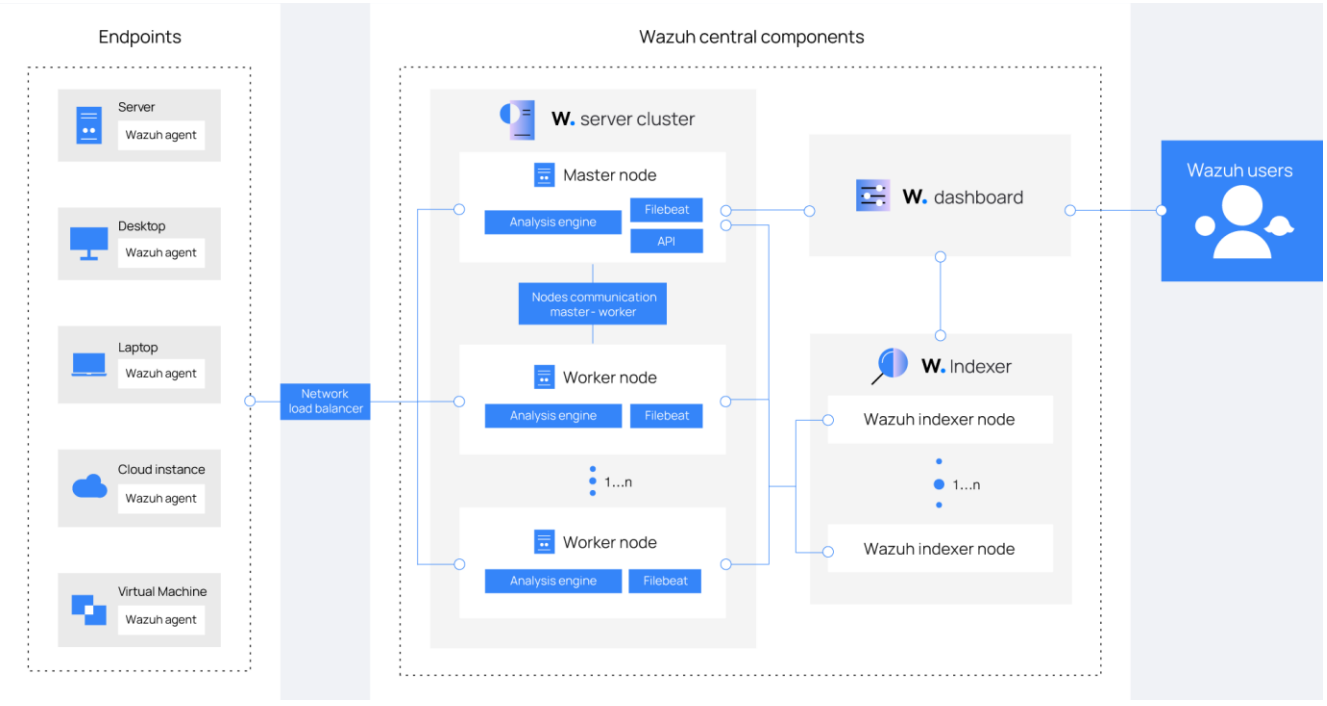


Рисунок 1.2 – Схема потоку обробки подій Wazuh

Важливою складовою є механізм правил (ruleset), який визначає логіку виявлення загроз. Ці правила можуть базуватися як на сигнатурному підході, так і на аналізі поведінкових характеристик. Це дозволяє виявляти як відомі атаки, так і аномальну активність, що раніше не була зафіксована.

Окремо слід відзначити можливості інтеграції Wazuh з іншими системами безпеки. Завдяки відкритій архітектурі платформа підтримує взаємодію з різними інструментами threat intelligence та системами автоматизації реагування. Це робить її гнучким інструментом для побудови комплексної системи кіберзахисту.

З практичної точки зору важливою перевагою Wazuh є можливість розгортання як у невеликих лабораторних середовищах, так і у масштабних корпоративних інфраструктурах. Це досягається завдяки модульній архітектурі та підтримці горизонтального масштабування компонентів системи [5].

Слід також враховувати, що ефективність роботи платформи значною мірою залежить від якості налаштування правил кореляції та джерел даних. Неправильно налаштовані правила можуть призводити як до надмірної кількості хибних спрацювань, так і до пропуску реальних інцидентів, що є критичним для систем моніторингу безпеки [11].

Таким чином, Wazuh є зрілою SIEM-платформою з відкритим кодом, що забезпечує централізований збір, кореляцію та аналіз подій безпеки з різних джерел інфраструктури. Гнучкість налаштування та підтримка широкого спектру джерел даних роблять платформу оптимальним вибором для побудови системи моніторингу у межах даної роботи [12].

1.4 Інструментальні засоби та апаратні вимоги до системи

Побудова системи моніторингу та управління подіями інформаційної безпеки потребує попереднього визначення функціональних і технічних вимог, які визначають склад програмного комплексу, обране інструментальне середовище та ресурсне забезпечення розгортання.

Основні функціональні вимоги до системи охоплюють такі аспекти: централізований збір та аналіз подій безпеки з різних джерел у режимі реального часу; підтримка кореляції подій та формування сповіщень про потенційні інциденти; інтеграція з мережним обладнанням Cisco через протокол Syslog; моніторинг кінцевих пристроїв на базі агентів; контроль цілісності файлів та виявлення аномальної активності; наявність веб-інтерфейсу та аналітичних дашбордів. Крім того, система

повинна підтримувати масштабованість і можливість розгортання у віртуальному середовищі, а також забезпечувати збереження журналів подій протягом 30–90 днів відповідно до стандартних вимог до систем аудиту.

При використанні протоколу Syslog для передачі подій від обладнання Cisco слід враховувати, що його стандартна реалізація через UDP-порт 514 не забезпечує механізмів шифрування або контролю цілісності даних. Тому при розгортанні в реальному середовищі рекомендується використовувати захищені канали передачі або виконати сегментацію управлінської мережі.

Для реалізації зазначених вимог у межах даної роботи обрано платформу Wazuh – відкрите SIEM-рішення з широкими можливостями інтеграції та економічною доцільністю впровадження. До складу системи входять такі основні компоненти: Wazuh Server (центральний вузол обробки подій та застосування правил виявлення загроз); Wazuh Agent (встановлюється на кінцеві пристрої, підтримується для Windows, Linux, macOS); Wazuh Indexer (побудований на базі OpenSearch, забезпечує індексацію та швидкий пошук по журналах); Wazuh Dashboard (веб-інтерфейс для моніторингу, аналітики та управління); мережне обладнання Cisco (джерело телеметрії через Syslog); Ubuntu Server як серверна операційна система.

Вимоги до апаратного забезпечення компонентів системи визначаються інтенсивністю генерації подій та кількістю підключених джерел. Для тестового середовища усі компоненти можуть бути розгорнуті на одному вузлі; у виробничому середовищі рекомендується їх розподіл між окремими серверами. Рекомендовані мінімальні характеристики наведені у таблиці 1.2.

Таблиця 1.2

Рекомендовані характеристики компонентів Wazuh

Компонент	Призначення	Мінімальні вимоги	Рекомендовані вимоги
Wazuh Server	Аналіз подій та обробка логів	2 CPU, 2 ГБ RAM	8 CPU, 4–8 ГБ RAM
Wazuh Indexer	Зберігання та індексація даних	4 CPU, 8 ГБ RAM	8 CPU, 32 ГБ RAM
Wazuh Dashboard	Візуалізація та моніторинг подій	2 CPU, 2 ГБ RAM	4 CPU, 4 ГБ RAM
Дисковий простір	Зберігання журналів та індексів	від 50 ГБ	від 200 ГБ SSD

Найбільше навантаження під час роботи системи припадає на компонент Wazuh Indexer, який виконує індексацію та зберігання великого обсягу подій. Тому для стабільної роботи рекомендується використовувати SSD-накопичувачі та достатній обсяг оперативної пам'яті. За необхідності окремі компоненти можуть бути розгорнуті у вигляді кластерів для горизонтального масштабування.

Для розгортання та тестування системи у даній роботі використовується середовище віртуалізації VirtualBox, яке дозволяє створити ізольовану лабораторну інфраструктуру без потреби у значній кількості фізичного обладнання. Така конфігурація спрощує налаштування компонентів та забезпечує зручність подальшого тестування системи.

Таблиця 1.3

Вимоги до системи

Категорія вимог	Опис вимог
Функціональні	Збір, аналіз та кореляція подій безпеки
Мережні	Інтеграція з обладнанням Cisco через Syslog, SNMP, NetFlow
Продуктивність	Обробка подій у режимі реального часу
Масштабованість	Можливість розширення кількості вузлів та джерел подій
Безпека	Контроль цілісності файлів та моніторинг аномалій
Зберігання даних	Підтримка політики зберігання журналів 30–90 днів
Інтерфейс	Наявність веб-інтерфейсу та дашбордів
Сумісність	Підтримка Linux, Windows та мережного обладнання
Віртуалізація	Підтримка розгортання у віртуальному середовищі

1.5 Моделі та сценарії виявлення кіберінцидентів у SIEM системах

Одним із ключових завдань сучасних систем моніторингу інформаційної безпеки є своєчасне виявлення кіберінцидентів та реагування на потенційні загрози. Ефективність SIEM-систем значною мірою залежить не лише від можливостей збору журналів подій, але й від правильного формування моделей виявлення атак та сценаріїв аналізу подій безпеки.

У більшості випадків процес виявлення інцидентів базується на аналізі подій, що надходять з різних джерел: серверів, робочих станцій, мережного обладнання,

міжмережних екранів та систем автентифікації. Зібрані події проходять етапи нормалізації та кореляції, після чого система визначає наявність ознак потенційної атаки або аномальної активності.

У сучасних SIEM-платформах використовуються декілька основних моделей виявлення загроз:

- сигнатурний аналіз;
- поведінковий аналіз;
- кореляційний аналіз подій;
- аналіз аномалій;
- моделі threat intelligence.

Сигнатурний підхід базується на використанні заздалегідь визначених правил або шаблонів атак. Система порівнює отримані події з базою відомих ознак шкідливої активності та формує сповіщення у випадку збігу. Даний підхід дозволяє ефективно виявляти відомі типи атак, однак має обмеження при виявленні нових або модифікованих загроз.

Поведінковий аналіз орієнтований на виявлення відхилень від нормальної активності користувачів або систем. У такому випадку система формує модель типової поведінки, після чого аналізує зміни у роботі хостів, мережних пристроїв або користувачів. Наприклад, різке збільшення кількості невдалих спроб автентифікації або нетипова активність мережних сервісів можуть свідчити про спробу компрометації системи.

Кореляційний аналіз є одним із найбільш важливих механізмів SIEM-систем. Його основне призначення полягає у встановленні взаємозв'язків між подіями, що надходять з різних джерел. Окремі події самі по собі можуть не становити загрози, однак їх поєднання дозволяє виявити складний сценарій атаки. Саме завдяки кореляції подій SIEM-системи можуть виявляти багатоступеневі атаки та формувати більш точні сповіщення.

Окрему роль у сучасних системах моніторингу відіграє аналіз аномалій. Його принцип полягає у визначенні нетипових подій або поведінки, які відрізняються від звичайного стану інформаційної системи. Для цього можуть використовуватись

статистичні методи, правила поведінкового аналізу або алгоритми машинного навчання. Подібний підхід дозволяє виявляти невідомі загрози та атаки, які не мають готових сигнатур.

У процесі роботи SIEM-систем значна увага приділяється сценаріям атак, які найчастіше зустрічаються у корпоративних мережах. Одним із найбільш поширених сценаріїв є brute force-атака на служби віддаленого доступу. У цьому випадку система аналізує велику кількість невдалих спроб автентифікації та визначає можливу спробу підбору облікових даних. Платформа Wazuh підтримує виявлення таких подій за допомогою правил кореляції та механізмів active response [4].

Ще одним поширеним сценарієм є сканування мережі. Подібна активність часто передує спробам експлуатації вразливостей або несанкціонованого доступу. У межах корпоративної мережі SIEM-система може виявляти підозрілу кількість мережних запитів, спроби підключення до великої кількості портів або аномальну активність між вузлами мережі.

Суттєву роль відіграє також виявлення спроб підвищення привілеїв (Privilege Escalation). У подібних сценаріях система аналізує зміни прав доступу, запуск привілейованих процесів або спроби використання адміністративних облікових записів[18]. Подібна активність може свідчити про компрометацію системи або внутрішню загрозу.

Для SIEM-систем характерним є також виявлення lateral movement – переміщення зловмисника між вузлами мережі після первинного компрометування. У такому випадку система аналізує мережні підключення, події автентифікації та взаємодію між пристроями. Поєднання даних з робочих станцій та мережного обладнання дозволяє виявляти подібні сценарії на ранніх етапах.

У сучасних системах моніторингу значного поширення набули моделі, побудовані на основі MITRE ATT&CK. Дана база знань містить опис тактик та технік, які використовуються під час реальних кібератак. Інтеграція SIEM-систем з MITRE ATT&CK дозволяє структурувати події безпеки відповідно до етапів атаки та спрощує процес аналізу інцидентів [6].

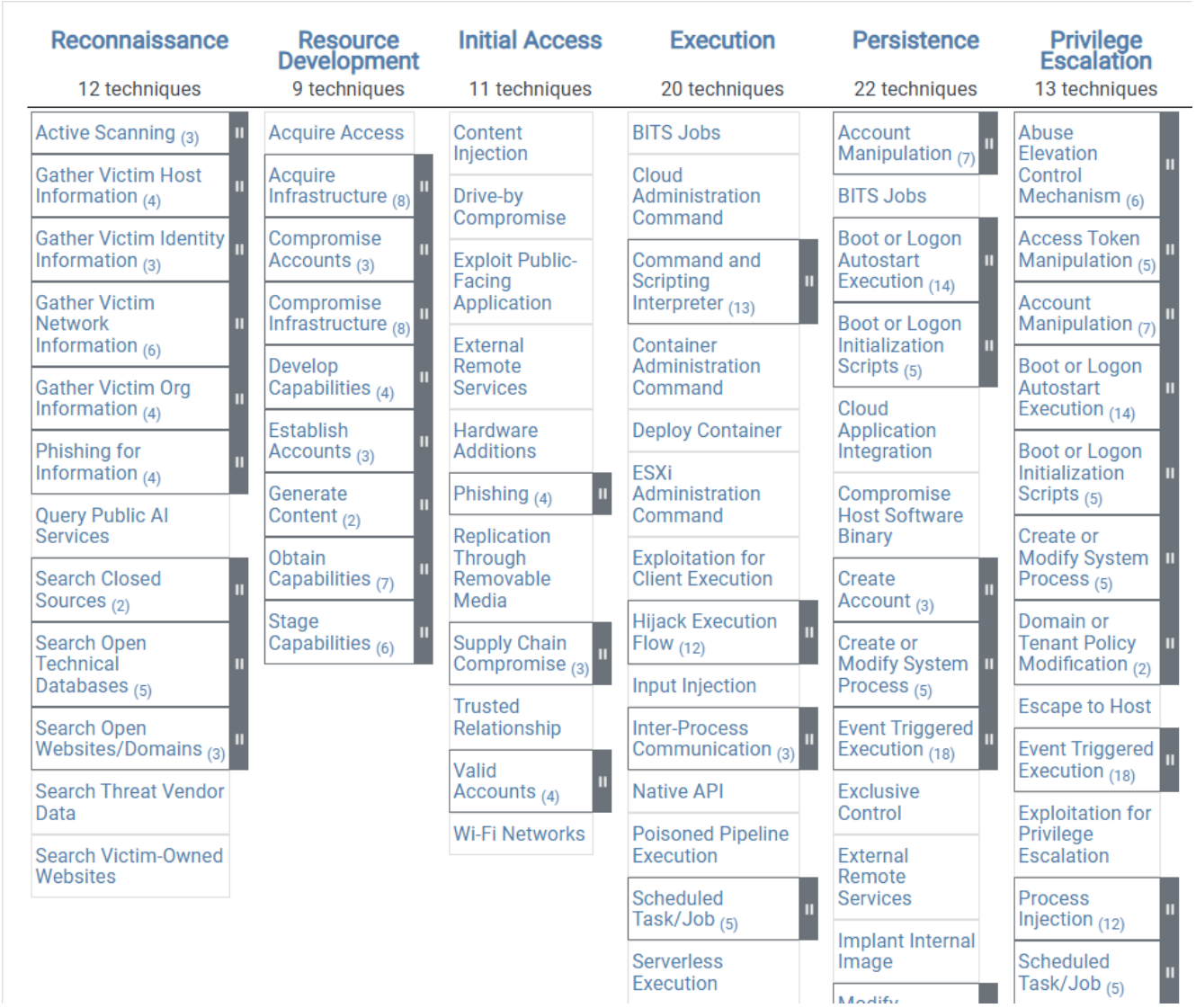


Рисунок 1.3 – Матриця MITRE ATT&CK

Платформа Wazuh підтримує механізми інтеграції з MITRE ATT&CK та дозволяє співставляти правила виявлення з відповідними техніками атак. Це забезпечує більш детальний аналіз подій безпеки та дозволяє підвищити ефективність реагування на інциденти [3].

У процесі виявлення кіберінцидентів важливу роль відіграє своєчасне сповіщення відповідальних осіб. SIEM-платформи фіксують підозрілу активність, формують алерти відповідного рівня критичності та передають їх адміністратору безпеки для подальшого реагування. Платформа Wazuh додатково підтримує механізм active response – модуль, який дозволяє автоматично виконувати заздалегідь визначені дії у відповідь на конкретні події, наприклад блокування IP-адреси на рівні операційної системи хоста. Однак у межах даної роботи система використовується

виключно в режимі моніторингу та виявлення загроз без активного втручання у роботу інфраструктури.

Для більш наочного представлення типових сценаріїв виявлення інцидентів доцільно узагальнити їх у вигляді таблиці.

Таблиця 1.4

Типові сценарії виявлення кіберінцидентів

Сценарій	Ознаки виявлення	Джерело подій
Brute Force	Велика кількість невдалих входів	Syslog, журнали автентифікації
Сканування мережі	Підключення до великої кількості портів	NetFlow, IDS/IPS
Privilege Escalation	Зміна прав доступу, запуск привілейованих процесів	Системні журнали
Lateral Movement	Аномальні мережні взаємодії між вузлами	Syslog, мережні події
Несанкціонований доступ	Вхід з нетипової IP-адреси або у нетиповий час	Журнали автентифікації
Зміна конфігурації	Несанкціоновані зміни системних файлів	File Integrity Monitoring

Використання моделей кореляції подій, поведінкового аналізу та автоматизованого реагування дозволяє сучасним SIEM-системам забезпечувати ефективне виявлення кіберінцидентів у корпоративних мережах. Поєднання телеметрії з кінцевих пристроїв та мережного обладнання Cisco створює основу для комплексного моніторингу безпеки та своєчасного реагування на загрози.

Висновки до першого розділу

У першому розділі проведено аналіз сучасного стану систем моніторингу інформаційної безпеки та розглянуто основні підходи до побудови SIEM-рішень у корпоративних інформаційних інфраструктурах. Встановлено, що зростання кількості кіберзагроз та ускладнення методів їх реалізації зумовлюють необхідність впровадження централізованих платформ збору та аналізу подій безпеки.

Визначено, що традиційні засоби захисту – міжмережні екрани, антивірусні рішення та системи виявлення вторгнень – не забезпечують цілісного контролю над станом інформаційної безпеки без централізованої кореляції подій. Саме тому системи класу SIEM набувають широкого поширення у корпоративних мережах та центрах моніторингу безпеки.

За результатами порівняльного аналізу комерційних та відкритих платформ найбільш доцільним для реалізації у межах даної роботи визначено платформу Wazuh, яка забезпечує централізований збір журналів, кореляцію подій, контроль цілісності файлів та інтеграцію з мережним обладнанням Cisco через протокол Syslog. Також розглянуто вимоги до апаратного забезпечення та визначено конфігурацію віртуального тестового середовища.

У межах дослідження моделей виявлення кіберінцидентів розглянуто сигнатурний, поведінковий та кореляційний підходи. Визначено основні сценарії атак, виявлення яких забезпечується платформою Wazuh: брутфорс, сканування мережі, lateral movement, підвищення привілеїв та несанкціоновані зміни конфігурації. Встановлено, що інтеграція з матрицею MITRE ATT&CK дозволяє структурувати події безпеки відповідно до реальних тактик та технік зловмисників, що суттєво спрощує аналіз інцидентів.

Отримані результати створюють основу для подальшого проєктування та реалізації системи моніторингу на базі платформи Wazuh із використанням мережного обладнання Cisco.

РОЗДІЛ 2 ПРОЕКТУВАННЯ СИСТЕМИ МОНІТОРИНГУ ТА УПРАВЛІННЯ ПОДІЯМИ БЕЗПЕКИ

2.1 Сценарії використання системи моніторингу та управління подіями

Проєктування системи моніторингу та управління подіями інформаційної безпеки передбачає визначення основних сценаріїв її використання, ролей користувачів та логіки взаємодії між компонентами системи. Саме сценарії використання дозволяють сформувати уявлення про практичне застосування SIEM-платформи в межах інформаційної інфраструктури та визначити послідовність обробки подій безпеки.

У межах даної роботи система будується на базі платформи Wazuh із використанням мережного обладнання Cisco як одного з основних джерел телеметрії та журналів подій. Основним призначенням системи є централізований збір, аналіз та кореляція подій інформаційної безпеки з подальшим виявленням потенційних кіберінцидентів.

Під час функціонування системи можна виділити декілька основних ролей користувачів:

- адміністратор безпеки;
- системний адміністратор;
- SOC-аналітик;
- мережний адміністратор.

Адміністратор безпеки відповідає за налаштування правил моніторингу, аналіз інцидентів та формування політик безпеки. Системний адміністратор забезпечує підтримку серверної інфраструктури та коректну роботу компонентів Wazuh. Мережний адміністратор виконує налаштування мережного обладнання Cisco та передачу журналів подій до системи моніторингу. SOC-аналітик здійснює аналіз отриманих сповіщень, класифікацію інцидентів та реагування на потенційні загрози.

Одним із базових сценаріїв використання системи є централізований збір журналів подій із серверів та мережних пристроїв. У даному випадку агенти Wazuh,

встановлені на серверних системах та робочих станціях, виконують передачу логів до Wazuh Server. Мережне обладнання Cisco передає журнали подій через протокол Syslog, а також може використовувати механізми SNMP та NetFlow для передачі додаткової мережної телеметрії.

Після надходження подій до центрального сервера виконується їх нормалізація та аналіз. На цьому етапі система визначає тип події, джерело її виникнення та рівень критичності. Далі запускаються механізми кореляції, які дозволяють встановити взаємозв'язки між окремими подіями та сформувати повноцінний сценарій потенційної атаки.

Одним із найбільш поширених сценаріїв використання системи є виявлення brute force-атак на служби віддаленого доступу. У межах такого сценарію Wazuh аналізує журнали автентифікації та визначає значну кількість невдалих спроб входу до системи. Якщо кількість подій перевищує встановлений поріг, формується сповіщення про можливу спробу підбору облікових даних. За необхідності система може автоматично застосувати механізми active response та заблокувати IP-адресу джерела атаки.

Ще одним важливим сценарієм є моніторинг мережної активності та виявлення аномалій у роботі мережної інфраструктури. У даному випадку система аналізує події, отримані від комутаторів та маршрутизаторів Cisco. Особлива увага приділяється нетиповій кількості мережних підключень, скануванню портів, різкому збільшенню мережного трафіку або змінам конфігурації мережних пристроїв.

Для корпоративних середовищ актуальним є сценарій контролю цілісності системних файлів. Wazuh підтримує механізм File Integrity Monitoring, який дозволяє відстежувати зміни у критичних директоріях операційної системи. У разі виявлення несанкціонованої модифікації системних файлів або конфігурацій система формує відповідне сповіщення для адміністратора безпеки.

Окрему роль у роботі SIEM-системи відіграє сценарій централізованого аналізу інцидентів. Після формування сповіщення інформація потрапляє до Wazuh Dashboard, де аналітик безпеки може переглянути деталі події, джерело виник-

нення, часові параметри та пов’язані журнали. Використання централізованої системи моніторингу дозволяє значно скоротити час пошуку причин інциденту та спрощує процес реагування.

У процесі експлуатації системи також важливим є сценарій виявлення спроб несанкціонованого доступу до мережного обладнання. У цьому випадку система аналізує події автентифікації Cisco IOS, зміни конфігурації пристроїв та активність привілейованих облікових записів. Подібний підхід дозволяє своєчасно виявляти внутрішні загрози та порушення політик безпеки.

Для SIEM-систем характерним є також сценарій комплексного аналізу інцидентів із декількох джерел одночасно. Наприклад, система може співставити події автентифікації з робочої станції, журнали міжмережного екрану та мережну активність комутатора Cisco. Поєднання різнорідних джерел інформації дозволяє підвищити точність виявлення складних атак та знизити кількість хибних спрацювань.

У межах реалізації системи передбачено механізм сповіщення адміністратора безпеки при виявленні підозрілої активності. Залежно від рівня критичності алерту система може:

- надсилати email-повідомлення адміністратору;
- формувати алерт у Wazuh Dashboard із зазначенням рівня критичності;
- зберігати деталі події у модулі Threat Hunting для подальшого аналізу;
- зберігати повний JSON-запис події в індексі для аналізу;
- групувати пов'язані події в єдиний інцидент.

Для більш наочного представлення основних сценаріїв використання системи доцільно узагальнити їх у вигляді таблиці 2.1.

Розглянуті сценарії використання демонструють принципи функціонування системи моніторингу подій безпеки та визначають основні механізми взаємодії між її компонентами.

Таблиця 2.1

Основні сценарії використання системи

Сценарій	Опис процесу	Результат
Збір журналів подій	Отримання логів із серверів та Cisco-пристроїв	Централізоване зберігання подій
Виявлення brute force	Аналіз невдалих спроб входу	Формування alert та блокування IP
Моніторинг мережі	Аналіз мережної активності	Виявлення аномалій та сканування
Контроль цілісності файлів	Відстеження змін системних файлів	Виявлення несанкціонованих змін
Аналіз інцидентів	Кореляція подій та аналіз alert	Реагування на кіберінциденти
Моніторинг Cisco	Аналіз Syslog, SNMP та NetFlow	Контроль стану мережної інфраструктури
Active Response	Автоматизоване реагування на загрози	Мінімізація наслідків атаки

Використання платформи Wazuh у поєднанні з мережним обладнанням Cisco дозволяє реалізувати централізований контроль подій безпеки, автоматизувати процес виявлення загроз та забезпечити оперативне реагування на кіберінциденти у межах інформаційної інфраструктури.

2.2 Проєктування архітектури та інтеграції Wazuh з обладнанням Cisco

Під час побудови систем моніторингу інформаційної безпеки важливим етапом є формування архітектури, яка забезпечує централізований збір подій, аналіз хостової та мережної активності, а також надання аналітикам інструментів для оперативного розслідування інцидентів. У межах даної роботи проєктується система моніторингу безпеки на базі SIEM-платформи Wazuh із використанням мережного обладнання Cisco та кінцевих вузлів під управлінням ОС Windows 10 як основних джерел журналів подій.

Проєктована система орієнтована на модель централізованого моніторингу, у межах якої всі події безпеки надходять до єдиного центру обробки та аналізу. Такий підхід дозволяє забезпечити наскрізний контроль за станом IT-інфраструктури, спростити кореляцію інцидентів та підвищити ефективність виявлення аномальної активності.

Архітектура системи передбачає розділення інфраструктури на декілька логічних сегментів:

- сегмент серверної інфраструктури (де розміщується Wazuh Manager та OpenSearch);
- сегмент адміністрування та моніторингу (робочі місця SOC-аналітиків);
- користувацький сегмент (робочі станції Windows 10);
- мережний сегмент Cisco (маршрутизатори та комутатори управління).

Сегментація мережі використовується для обмеження взаємодії між окремими вузлами та зменшення ризику горизонтального переміщення зловмисника (Lateral Movement) у випадку компрометації одного з компонентів. Даний підхід також дозволяє спростити контроль мережних потоків та забезпечує більш ефективний аналіз подій безпеки.

Центральним елементом архітектури виступає сервер моніторингу Wazuh Manager, який виконує функції приймання, декодування, аналізу та кореляції подій. На сервер надходять журнали безпеки від локальних агентів Wazuh, а також події від мережного обладнання Cisco через протокол віддаленого журналювання.

Інтеграція сервера моніторингу з інфраструктурними компонентами реалізується через такі джерела даних:

- Syslog – для збору подій мережного рівня з обладнання Cisco;
- SNMP Traps / Polling – для контролю доступності та стану пристроїв;
- Windows Event Channels – для отримання стандартних журналів безпеки Windows;
- Microsoft Sysmon – для розширеного хостового моніторингу процесів та пам'яті.

Основний потік мережних подій безпеки формується через протокол Syslog. Маршрутизатори та комутатори Cisco передають журнали подій до сервера моніторингу, що дозволяє централізовано аналізувати спроби автентифікації (як успішні, так і невдалі), зміни конфігурації пристроїв, мережні помилки на інтерфейсах, події маршрутизації, а також перезапуск служб та сервісів під управлінням Cisco

IOS. Застосування централізованого журналювання забезпечує незмінність логів, оскільки навіть у разі повної компрометації пристрою зловмисник не зможе видалити події, які вже були надіслані на сервер SIEM.

Для додаткового контролю стану та продуктивності мережного обладнання Cisco задіюється протокол SNMP. Його застосування дозволяє отримувати інформацію про навантаження на інтерфейси, використання центрального процесора, оперативної пам'яті, доступність пристроїв за метрикою Uptime та критичні помилки на фізичних портах, що допомагає виявляти аномалії на ранніх стадіях (наприклад, аномальний сплеск навантаження під час сканування чи DDoS).

На рівні хостів інфраструктури розгортається агентське рішення Wazuh Agent. Агенти виконують збір системних журналів подій, моніторинг активних процесів, контроль цілісності критичних файлів (FIM) та передають структуровані дані до центрального сервера через захищений протокол.

Для забезпечення розширеного моніторингу хостового рівня операційної системи Windows 10 та детектування складних цілеспрямованих атак, стандартного журналу безпеки Windows (Windows Security Log) недостатньо, оскільки він не фіксує деталізовану активність процесів та мережних з'єднань на рівні ядра. З метою вирішення цієї проблеми на цільовому хості Windows 10 проєктується інтеграція системної утиліти Microsoft Sysmon (System Monitor).

Sysmon функціонує як системний сервіс та драйвер пристрою, що дозволяє генерувати глибоку телеметрію: створення процесів із командного рядка (Event ID 1), мережні підключення окремих програм (Event ID 3), модифікацію ключів реєстру (Event ID 11) та спроби доступу до пам'яті критичних процесів (Event ID 10), зокрема lsass.exe. Агент Wazuh конфігурується на збір подій від каналу Microsoft-Windows-Sysmon/Operational, що дозволяє об'єднати можливості глибокого аналізу хоста та аналітичного рушія SIEM [7].

Проєктована система передбачає використання механізмів часової та логічної кореляції подій. Сервер Wazuh аналізує потік подій, що надходить одночасно з хостового та мережного рівнів, та визначає взаємозв’язки між ними. Наприклад,

послідовна поява подій сканування портів з боку мережного сегмента Cisco та наступні невдалі спроби входу через RDP на хості Windows свідчать про цілеспрямовану атаку типу Brute-Force.

Архітектура системи орієнтована на швидке виявлення, категоризацію та наочну візуалізацію інцидентів для оператора безпеки. У випадку аналітичного збігу подій із критеріями правил безпеки система виконує:

- миттєве формування сповіщення (Alert) із присвоєнням рівня критичності (від 1 до 16);
- агрегацію однотипних подій безпеки за часовим вікном для запобігання перевантаженню бази даних;
- передачу події у графічний інтерфейс Wazuh Dashboard для аналізу SOC-аналітиком;
- фіксацію та індексацію інциденту у довгостроковому сховищі для ретроспективного розслідування.

Важливою складовою проєкту є організація надійного зберігання журналів подій. Для цього використовується компонент індексації даних OpenSearch, який забезпечує високошвидкісний повнотекстовий пошук по журналах та зберігання подій протягом визначеного періоду відповідно до політики утримання даних. У межах даної роботи передбачається зберігання журналів безпеки протягом 30–90 днів залежно від типу подій та інтенсивності журналювання.

З точки зору інформаційної безпеки окрему увагу приділено захисту каналів передавання інформації. Оскільки стандартний протокол Syslog передає дані у відкритому вигляді без шифрування, обмін логами між обладнанням Cisco та сервером моніторингу рекомендується виконувати у межах ізольованого сегмента мережі управління (Management VLAN). Взаємодія між агентами Wazuh та сервером за замовчуванням захищена шифруванням TLS, що виключає можливість перехоплення або підміни хостових логів.

Для наочного представлення взаємодії компонентів системи основні потоки даних та канали їх надходження узагальнено у Таблиці 2.2.

Таблиця 2.2

Основні потоки даних у системі моніторингу безпеки

Джерело подій	Тип даних	Протокол / Канал передачі	Призначення
Cisco Router/Switch	Системні журнали подій	Syslog (UDP/514)	Контроль автентифікації, збоїв та команд у Cisco IOS
Cisco Device	Метрики стану обладнання	SNMP (UDP/161-162)	Моніторинг навантаження на CPU та інтерфейси
Linux Server	Логи операційної системи	Syslog (локальний збір)	Аудит дій суперкористувача та стану демонів
Windows Host	Журнали безпеки (Security)	Wazuh Agent (Event Channel)	Аналіз успішності автентифікації та дій із обліковими записами
Windows Host	Глибока хостова телеметрія	Wazuh Agent (Sysmon Operational)	Детектування Mimikatz, Reverse Shell, аномальних процесів

У межах спроєктованої архітектури передбачено можливість подальшого масштабування системи. При збільшенні кількості вузлів інфраструктури або інтенсивності генерації логів (метрика EPS – Events Per Second) окремі компоненти індексації та аналізу можуть бути винесені на різні сервери або об’єднані у кластерну структуру з балансуванням навантаження [2].

Запроєктована архітектура забезпечує централізований моніторинг подій інформаційної безпеки, підтримує інтеграцію з мережним обладнанням Cisco та хостами Windows, а також створює надійну основу для реалізації повноцінного SIEM-функціоналу. Поєднання хостового моніторингу на базі Sysmon, мережного логування Syslog та централізованого аналітичного рушія дозволяє суттєво підвищити точність виявлення інцидентів та мінімізувати час реагування на кіберзагрози.

2.3 Інформаційне забезпечення та алгоритми обробки подій

Ефективність систем моніторингу інформаційної безпеки значною мірою залежить від правильності організації процесу збору, обробки та аналізу подій. У сучасних SIEM-системах основний обсяг інформації формується у вигляді журналів подій, що надходять з різних джерел: серверів, робочих станцій, мережного обладнання та систем автентифікації. У межах даної роботи інформаційне забезпечення системи будується на основі централізованої моделі обробки подій із використанням платформи Wazuh та сховища OpenSearch.

Однією з ключових особливостей сучасних SIEM-платформ є приведення різнорідних журналів подій до єдиного формату представлення. Це дозволяє спростити подальший аналіз, виконувати кореляцію подій та забезпечити швидкий пошук інформації у великих масивах даних. У проєктованій системі всі події після обробки зберігаються у форматі JSON, який є базовою структурою представлення інформації про інциденти.

До основних полів інформаційної моделі події належать:

- timestamp: фіксація точного часу виникнення події (критично важливо для побудови часових залежностей між подіями та виявлення багатоступеневих атак).
- agent.id, agent.name, agent.ip: інформація про джерело події (для агентського моніторингу – кінцевий вузол, для Cisco – IP-адреса маршрутизатора або комутатора).
- decoder.name: інформація про декодер, використаний для розбору специфічного синтаксису (наприклад, декодери Cisco IOS виділяють структуру мережних подій із Syslog-повідомлень).
- rule.id та rule.level: ідентифікатор правила безпеки, яке спрацювало, та відповідний рівень критичності інциденту.
- event.original: початковий текст журналу події без змін, що є необхідним для подальшого аудиту.
- data.srcip та data.dstip: IP-адреси джерела та призначення (використовуються для аналізу мережного трафіку).

Приклад структури події після обробки системою наведено на рис. 2.1.

У межах проєктованої системи використовується конвеєрна модель обробки подій (Pipeline Processing). Даний підхід передбачає послідовне проходження журналу через декілька етапів аналізу:

1. Етап Pre-decoding. Система виконує початковий аналіз журналу та виділяє базові елементи повідомлення: дату, час, hostname, IP-адресу пристрою та тип

сервісу. Для Cisco Syslog це дозволяє відокремити службову інформацію від основного змісту. Наприклад, із сирого логу %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: admin] система автоматично виділяє категорію повідомлення.


```
{
  "timestamp": "2026-05-27T10:21:15Z",
  "agent": {
    "id": "001",
    "name": "Cisco-R1",
    "ip": "192.168.1.1"
  },
  "rule": {
    "id": "5710",
    "level": 10
  },
  "decoder": {
    "name": "cisco-ios"
  },
  "event": {
    "original": "%SEC_LOGIN-4-LOGIN_FAILED: Login failed"
  },
  "data": {
    "srcip": "192.168.1.50"
  }
}
```


Рисунок 2.1 – Структура події після обробки системою

2. Етап Decoding. Основне призначення етапу – розбір синтаксису та перетворення текстового журналу у структуровані поля. За допомогою спеціалізованих декодерів Wazuh аналізує події автентифікації, зміни конфігурацій, помилки інтерфейсів та ACL-події. На цьому етапі визначаються: тип події, ім’я користувача, IP-адреса джерела та рівень загрози.

3. Етап Rule matching. Перевірка події за деревом правил безпеки [2]. Правила можуть оцінювати частоту повторення, наявність певних ключових слів або комбінації декількох подій.

Одним із ключових механізмів системи є кореляція подій. У межах даної роботи реалізовується алгоритм часової кореляції. Його принцип полягає у визначенні частоти повторення певної активності протягом заданого «вікна» часу. Наприклад, при виявленні атаки типу Brute-force система аналізує кількість невдалих спроб автентифікації.

Математично логіку спрацювання правила кореляції можна подати у такому вигляді:

$$N_{failed\_login} > 5 \text{ при } \Delta t \leq 60c \rightarrow L_{alert} = 10$$

Де:

- N_{failed_login} – кількість зафіксованих невдалих спроб входу;
- Δt – часовий інтервал аналізу (вікно кореляції);
- L_{alert} – рівень критичності сформованого інциденту (Alert Level).

Подібний механізм ефективно виявляє сканування мережі, масові запити до сервісів та повторювані помилки автентифікації. Крім часової кореляції, система підтримує багатоджерельний аналіз. Наприклад, одночасна поява невдалих спроб входу, зміни конфігурації Cisco та аномального трафіку NetFlow свідчить про високоймовірну спробу компрометації мережного вузла.

Після спрацювання правила система формує Alert та передає його до Wazuh Dashboard. Збереження журналів у сховищі OpenSearch забезпечує можливість ретроспективного аналізу: пошук по часових інтервалах, виявлення повторюваних інцидентів та побудову статистики атак. Дана архітектура гарантує централізовану обробку подій безпеки та створює стійку основу для комплексного моніторингу подій безпеки.

2.4 Опис реалізації функціоналу підсистеми збору та аналізу даних

Практична реалізація системи моніторингу безпеки передбачає налаштування взаємодії між мережним обладнанням Cisco та платформою Wazuh. Основним джерелом подій у межах дослідження виступають пристрої на базі Cisco IOS, які генерують журнали подій, повідомлення про спроби автентифікації, зміни конфігурації та системні збої.

Архітектура взаємодії побудована за моделлю централізованого збору подій. Усі журнали безпеки передаються на сервер Wazuh за допомогою протоколу Syslog, де відбувається їх подальший аналіз, декодування та кореляція у реальному часі.

Для передавання Syslog-повідомлень на адресу сервера моніторингу використовуються стандартні засоби конфігурації глобального режиму Cisco IOS.

```
R-PriLipko-SOC(config)#logging on
R-PriLipko-SOC(config)#logging host 192.168.100.100
R-PriLipko-SOC(config)#logging trap informational
R-PriLipko-SOC(config)#service timestamps log datetime msec
R-PriLipko-SOC(config)#
```

Рисунок 2.2 – Налаштування експорту Syslog на Cisco IOS

У наведеній конфігурації IP-адреса 192.168.100.10 відповідає серверу Wazuh Manager. Параметр logging trap informational визначає рівень деталізації журналювання (рівень 6 – Informational та вище), що дозволяє фіксувати не лише критичні помилки, а й події успішного чи невдалого доступу та зміни режимів роботи пристрою. Команда service timestamps забезпечує додавання точних міток часу до повідомлень у мілісекундах (msec). Наявність синхронізованих часових міток є критичною вимогою для коректної роботи механізмів часової кореляції SIEM-системи при аналізі послідовності дій зловмисника.

На стороні сервера Wazuh необхідно налаштувати модуль прийому віддалених повідомлень. Основні параметри демона задаються у конфігураційному файлі /var/ossec/etc/ossec.conf. Для забезпечення прийому Syslog-повідомлень використовується директива <remote>, конфігурацію якої наведено на рисунку 2.3.

```
<remote>
 <connection>syslog</connection>
 <port>514</port>
 <protocol>udp</protocol>
 <allowed-ips>192.168.100.1</allowed-ips>
 <local_ip>192.168.100.100</local_ip>
</remote>
```

Рисунок 2.3 – Конфігурація Wazuh Manager для прийому Syslog

Параметр <allowed-ips> обмежує перелік джерел (підмережа управління 192.168.100.0/24), яким дозволено надсилати логи на сервер, мінімізуючи ризик

атаки типу Syslog spoofing. Після внесення змін служба перезапускається командою sudo systemctl restart wazuh-manager. Коректність надходження пакетів перевіряється утилітою: sudo tcpdump -i any udp port 514.

Після успішного надходження повідомлень аналітичний рушій Wazuh виконує їх декодування. Одним із критичних сценаріїв моніторингу мережного рівня є контроль змін конфігурації. При зміні налаштувань операційної системи Cisco IOS генерується подія: %SYS-5-CONFIG_I: Configured from console by admin. Для автоматичного відстеження цієї активності у файл /var/ossec/etc/rules/local_rules.xml додається кастомне правило детектування

```
<group name="local,syslog,cisco,">
  <rule id="100010" level="10">
    <decoded_as>cisco-ios</decoded_as>
    <match>CONFIG_I</match>
    <description>Cisco configuration modification detected</description>
    <group>configuration_change,network,</group>
  </rule>
</group>
```

Рисунок 2.4 – Кастомне правило Wazuh для виявлення змін конфігурації

У цьому правилі аналітичний рушій шукає збіг за ключовим словом CONFIG_I у повідомленнях, які попередньо були успішно розібрані базовим декодером cisco-ios. При спрацюванні тригера генерується сповіщення високого рівня критичності (level="10"), що свідчить про пряме втручання в архітектуру мережі.

Для верифікації розробленого правила використовується вбудована аналітична утиліта /var/ossec/bin/wazuh-logtest. Вона дозволяє аналізувати синтаксис, симулювати надходження логу та перевіряти коректність роботи тригера без очікування реального інциденту в інфраструктурі [2].

Практична реалізація підсистеми збору даних демонструє гнучкість інтеграції відкритої архітектури Wazuh із пропрієтарним обладнанням Cisco. Налаштований конвеєр обробки логів створює необхідне інформаційне підґрунтя для розробки кастомних правил детектування складних та цілеспрямованих кібератак, архітектурна логіка яких буде детально розглянута у наступному підрозділі.

2.5 Розробка та обґрунтування кастомних правил детектування кібератак

Для забезпечення комплексного моніторингу безпеки та виявлення складних цілеспрямованих атак у межах дипломного проєкту було розроблено та інтегровано набір кастомних правил детектування у конфігураційний файл local_rules.xml. Розроблені правила базуються на ієрархічній структурі SIEM Wazuh, використовують логічні індикатори, механізми підрахунку частоти подій (Frequency-based rules), регулярні вирази стандарту PCRE2 та напряму інтегровані з міжнародною матрицею тактик і технік зловмисників MITRE ATT&CK.

Вся логіка детектування розділена на логічні блоки відповідно до етапів життєвого циклу атаки (Cyber Kill Chain).

1. Мережна розвідка (Port Scan) та дослідження SMB-ресурсів

Для виявлення сканування інфраструктури (наприклад, за допомогою інструменту Nmap зі сторони Kali Linux) реалізовано дворівневу логіку аналізу повідомлень від вбудованого міжмережного екрана Windows (WFP):

- Базове правило ID 100151 збирає події блокування вхідних підключень (EventID 5157, батько 60104).
- Кореляційне правило ID 100152 (Рівень 10) аналізує потік подій від правила 100151. Якщо протягом 120 секунд фіксується 10 заблокованих з'єднань з однієї й тієї самої IP-адреси (<same_field>win.eventdata.sourceAddress</same_field>), система ідентифікує це як сканування портів (MITRE T1046).
- Правила ID 100400 та ID 100900 контролюють активність навколо протоколу SMB (батько 60139). Вони виявляють спроби збору інформації про мережні папки (T1135), зокрема спроби анонімного перерахування ресурсів (Anonymous SMB enumeration) під обліковими записами guest або anonymous.

2. Виявлення підбору облікових даних та rdp-активності

Блок правил орієнтований на детектування спроб несанкціонованого доступу до хоста Windows 10 через службу віддаленого робочого столу (RDP).

- Правило ID 100100: Відстежує масові невдалі спроби входу в операційну систему. Воно базується на вбудованому правилі 60122 (Windows failed

logon) і спрацьовує, якщо фіксується 5 помилок автентифікації протягом 60 секунд (frequency="5" timeframe="60"). Направлено на детектування техніки MITRE T1110.

- Правило ID 100101: Спеціалізований тригер для виявлення перебору паролів саме по протоколу RDP, що використовує як батьківське правило 60106. Прив'язано до техніки T1110.001.
- Правило ID 100800 (Рівень 14): Фіксує критичну подію блокування облікового запису користувача (EventID 4740), що є прямим наслідком успішного чи агресивного брутфорсу.
- Правило ID 100102 (Рівень 8): Відстежує компрометацію або легітимний успішний вхід через RDP (EventID 4624), фільтруючи тип входу за допомогою регулярного виразу <field name="win.eventdata.logonType">^10$</field> (RemoteInteractive).

3. Контроль ексфільтрації даних (data exfiltration)

Блок правил орієнтований на виявлення спроб викрадення конфіденційної інформації через незахищені канали зв'язку за допомогою утиліти Sysmon:

- Правила ID 100700 та ID 100702 відстежують ініціалізацію мережних з'єднань на стандартний порт FTP (порт 21) через аналіз Sysmon Event ID 3 (батьківські правила 61612 та 61605), маркуючи це як потенційну ексфільтрацію (MITRE T1048.003).
- Правило ID 100701 (Рівень 10) аналізує створення процесів (Sysmon Event ID 1) за допомогою PCRE2-виразу для перехоплення запуску консольних інструментів передачі даних, таких як ftp, scp, curl, wget, а також вбудованих утиліт ОС Windows, що часто використовуються зловмисниками (наприклад, certutil з прапорцем urlcache або bitsadmin).

4. Ескалація привілеїв та протидія детектуванню (defense evasion)

Контроль за діями зловмисника, спрямованими на закріплення у системі з правами адміністратора та приховування слідів присутності:

- Правило ID 100703 (Рівень 12): Фіксує створення нових користувачів через командний рядок шляхом аналізу запуску процесів net.exe або net1.exe із прапорцем /add (MITRE T1136.001).
- Правило ID 100704 (Рівень 13): Визначає несанкціоноване підвищення прав (Privilege Escalation, T1098), фіксуючи додавання створеного користувача до локальної групи адміністраторів (administrators, admins).
- Правило ID 100705 (Рівень 14): Має надвисокий пріоритет та виявляє спроби очищення журналів подій Windows зловмисником за допомогою штатної утиліти wevtutil cl (MITRE T1070.001), що є критичним індикатором компрометації хоста.

5. Закріплення в системі (persistence)

Для забезпечення постійного доступу зловмисники автоматизують запуск свого шкідливого ПЗ. Реалізовані правила детектують три основні методи закріплення через Sysmon Event ID 1:

- ID 100500: Створення запланованих завдань через команду schtasks /create (T1053.005).
- ID 100501: Модифікація ключів автозапуску системного реєстру Windows через команду reg add у гілках Run або RunOnce (T1547.001).
- ID 100502: Створення нових шкідливих системних служб (сервісів) через утиліти sc create, sc config або PowerShell командлет New-Service (T1543.003).

6. Детектування атак на powershell та інструменту mimikatz

Оскільки PowerShell є основним інструментом для виконання безфайлових атак, розроблено правила для аналізу його командного рядка:

- ID 100300: Виявляє запуск скриптів із прихованими або закодованими командами (-enc, -encodedcommand) та завантаження payload із мережі (downloadstring, iex).
- ID 100301 та ID 100201: Напряму ідентифікують використання відомих хакерських фреймворків та інструментів тестування на проникнення за їхніми сигнатурними іменами (mimikatz, sekurlsa, lsadump, powersploit).

- ID 100200: Найбільш надійний індикатор викрадення облікових даних (Credential Dumping, MITRE T1003.001). Базується на аналізі Sysmon Event ID 10 (Process Access, батько 61616). Воно генерує аларм максимального рівня критичності щоразу, коли будь-який сторонній процес намагається отримати доступ до пам'яті чутливого системного процесу lsass.exe (Local Security Authority Subsystem Service) для зчитування паролів у відкритому вигляді або їхніх хешів.

7. Виявлення зворотних з'єднань (reverse shell)

Для фіксації активності шкідливого ПЗ (троянів, payloads фреймворку Metasploit) впроваджено комплексний аналіз:

- ID 100804: Шукає утиліти генерації та виконання реверс-шеллів (msfvenom, meterpreter, reverse_tcp, запуск cmd.exe з прапорцем -e).
- ID 100802: Контролює вихідні мережні з'єднання на хості через нетипові порти, які найчастіше використовуються як канали зворотного зв'язку з командними серверами атак (C2) – порти 4444, 4445, 5555, 1337, 31337 (MITRE T1071).
- ID 100803 (Рівень 13): Виявляє аномальну ієрархію процесів (MITRE T1204), коли командний інтерпретатор (cmd.exe або powershell.exe) запускається не користувачем, а є дочірнім процесом офісних пакетів (winword.exe, excel.exe) або провідника, що однозначно вказує на успішну експлуатацію вразливості або запуск макросу.

8. Аудит pass-the-hash та веб-інцидентів

- Правила ID 101550 та ID 100600 орієнтовані на детектування спроб горизонтального переміщення мережею за допомогою техніки Pass-the-Hash (T1550.002) через фіксацію аномальних автентифікацій за протоколом NTLM (ntlmssp) з типом входу 3 (Network Logon).
- Правила ID 101450 (батько 31101) та ID 101350 (батько 40101) забезпечують верхній рівень моніторингу: детектують автоматизоване сканування веб-ресурсів хоста на наявність вразливостей (T1190) та виявляють ознаки DoS-атак за критерієм високої щільності Syslog-пакетів (понад 50 подій за 5 секунд).

Висновки до другого розділу

У другому розділі спроєктовано архітектуру системи моніторингу подій інформаційної безпеки на базі платформи Wazuh та виконано опис усіх ключових компонентів реалізації. Визначено основні сценарії використання системи та ролі учасників: адміністратора безпеки, аналітика та оператора моніторингу. Описано загальну структуру системи, яка включає Wazuh Server, Wazuh Indexer на базі OpenSearch, Wazuh Dashboard та агенти на кінцевих вузлах.

Розглянуто інформаційне забезпечення системи: джерела подій (Windows Security Event Log, Sysmon, Cisco Syslog), механізм передачі та нормалізації даних, а також багаторівневий процес обробки подій – від pre-decoding до rule matching та генерації алертів. Детально описано об'єктну структуру системи та взаємодію компонентів. Розроблено комплекс кастомних правил кореляції у файлі local_rules.xml, які охоплюють повний ланцюжок атаки відповідно до методології MITRE ATT&CK – від мережної розвідки до знищення слідів присутності. Отримані результати створюють повну технічну основу для практичного розгортання та тестування системи у третьому розділі.

РОЗДІЛ 3 ПРАКТИЧНІ ДОСЛІДЖЕННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ МОНІТОРИНГУ БЕЗПЕКИ

3.1 Опис та архітектура експериментального лабораторного стенду

З метою практичної перевірки ефективності системи моніторингу безпеки на основі Wazuh SIEM було розгорнуто експериментальний лабораторний стенд, який імітує реальне корпоративне мережне середовище. Стенд відтворює типову інфраструктуру малого підприємства, що включає мережне обладнання, сервери, робочі станції та централізовану систему моніторингу подій безпеки.

Основною метою стенду є моделювання реалістичних кіберзагроз та дослідження можливостей їх виявлення засобами SIEM-системи. Лабораторне середовище дозволяє відтворювати повний ланцюжок атаки – від первинної розвідки до закріплення в системі – та оцінювати ефективність виявлення кожного етапу.

Фізична та логічна топологія мережі

Лабораторний стенд побудований на базі фізичного мережного обладнання Cisco та двох персональних комп'ютерів, один з яких виступає платформою для запуску атак, а другий – хостом для віртуальних машин. Всі пристрої об'єднані в єдину локальну мережу з адресним простором 192.168.100.0/24.

Фізична топологія стенду побудована за схемою «зірка» з комутатором Cisco у центрі. Маршрутизатор Cisco забезпечує шлюз за замовчуванням та виступає об'єктом моніторингу мережного рівня. Два хости підключені до портів комутатора та отримують статичні IP-адреси для забезпечення стабільності тестового середовища (рис.3.1).

Логічна топологія стенду передбачає розподіл вузлів на функціональні ролі: вузол атакуючого (Kali Linux), вузол жертви (Windows 10 VM), центр моніторингу (Wazuh Server VM) та мережна інфраструктура (маршрутизатор та комутатор Cisco). Така структура дозволяє реалістично відтворити сценарії атак із зовнішнього та внутрішнього сегментів мережі (рис.3.2).

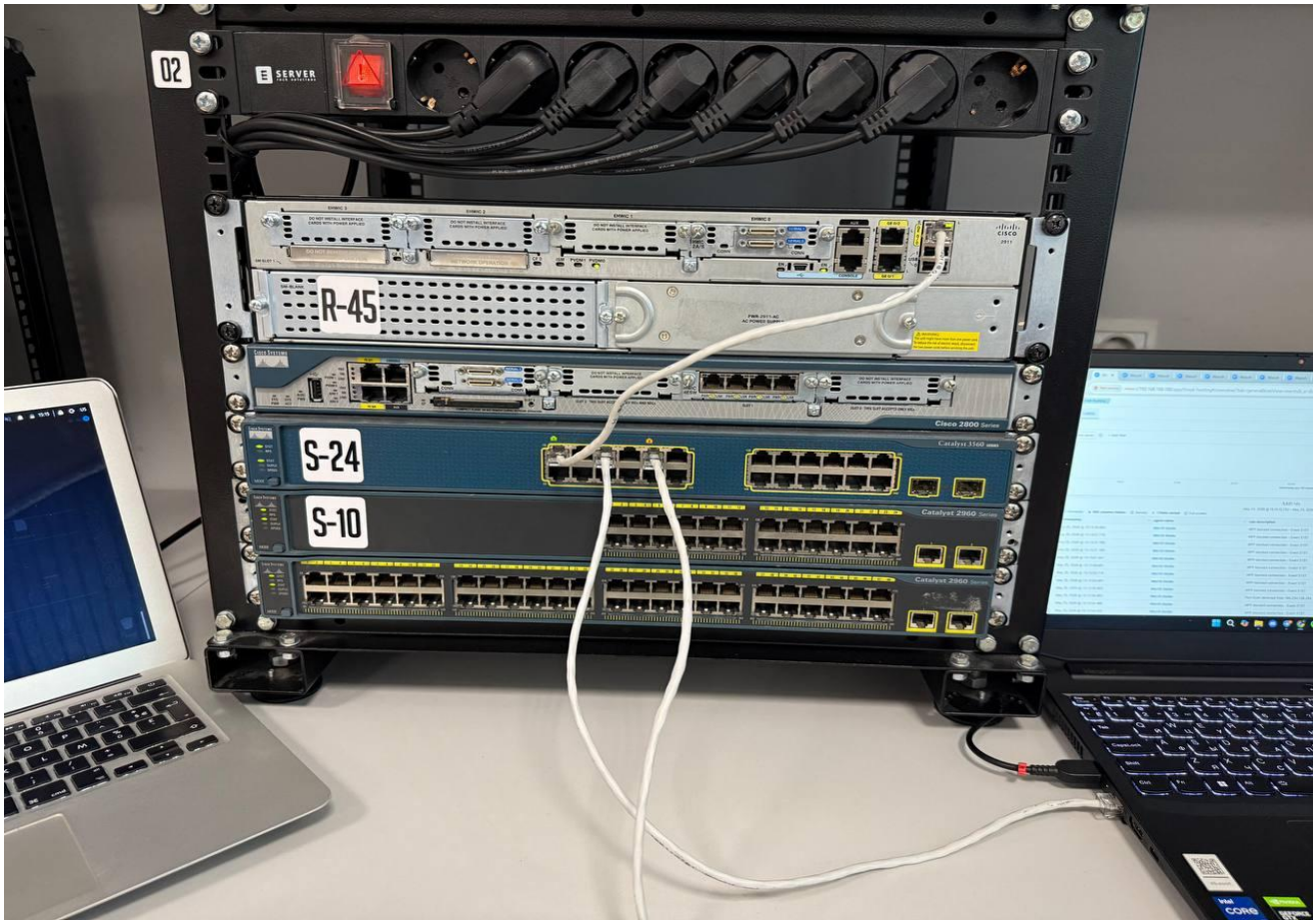


Рисунок 3.1 – Фізична топологія лабораторного стенду

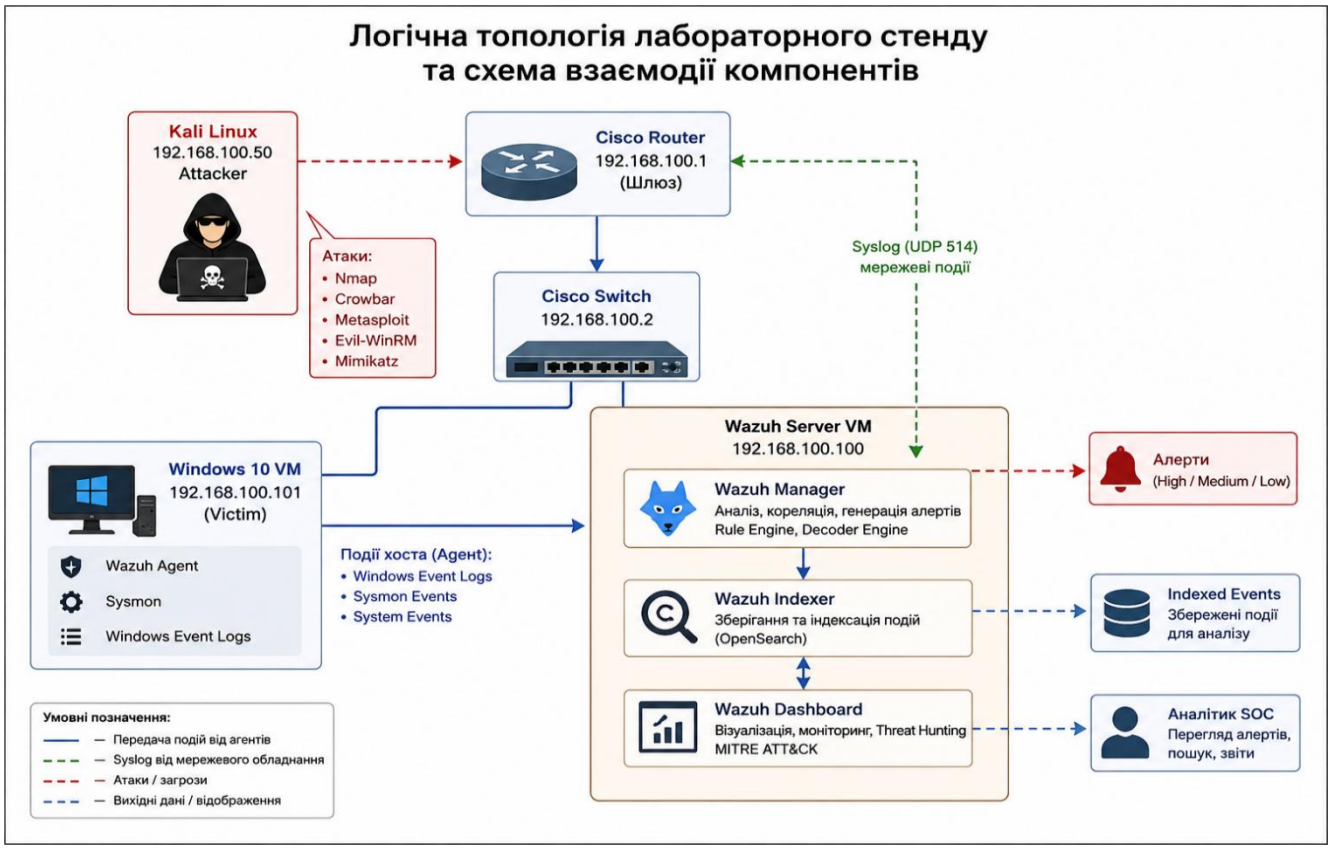


Рисунок 3.2 – Схема взаємодії компонентів

Для забезпечення взаємодії між компонентами стенду було виконано базове налаштування IP-адресації. На маршрутизаторі та комутаторі Cisco адреси призначалися через консольний інтерфейс на відповідних інтерфейсах (GigabitEthernet та VLAN). Крім того, на обох пристроях Cisco було увімкнено генерацію та відправку системних журналів на IP-адресу центрального сервера Wazuh. На віртуальних машинах Wazuh Server (Ubuntu) та Windows 10, а також на фізичній станції Kali Linux налаштовано статичні IP-адреси відповідно до плану адресації мережі.

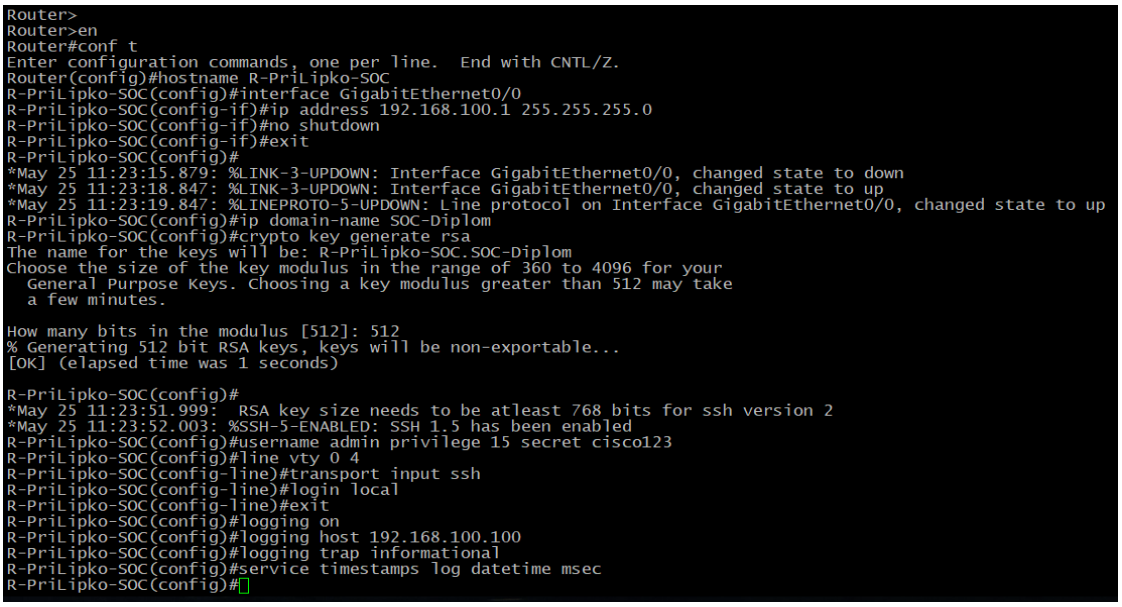


Рисунок 3.3 – Налаштування IP-адресації та відправки системних журналів на маршрутизаторі

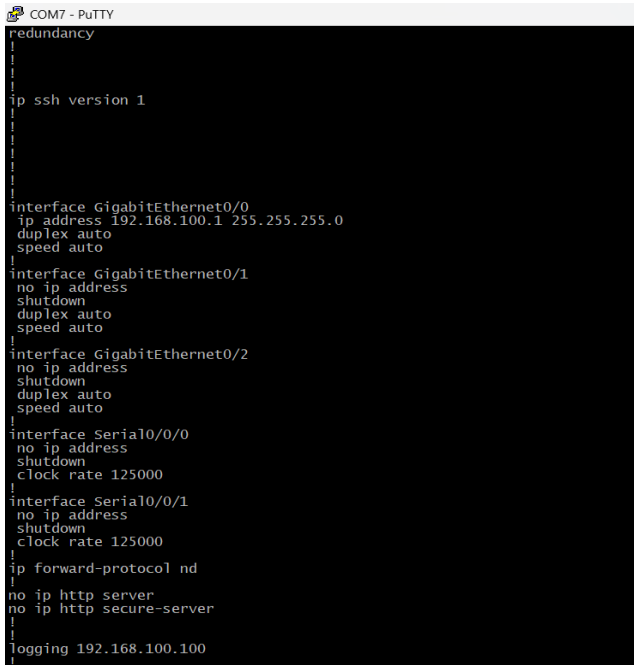


Рисунок 3.4 – Налаштування IP-адресації та відправки системних журналів на комутаторі

```
kykyshka@wazuh-server:~$ sudo ip addr add 192.168.100.100/24 dev enp0s3
[sudo] password for kykyshka:
kykyshka@wazuh-server:~$ sudo ip route add default via 192.168.100.1
```

Рисунок 3.5 – Налаштування IP-адресації на Wazuh Server

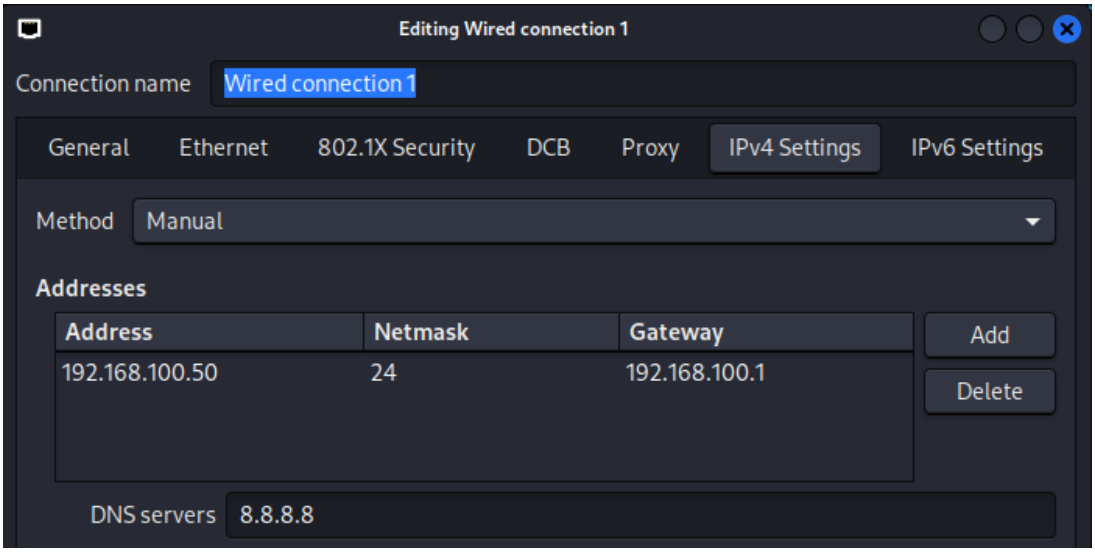


Рисунок 3.6 – Налаштування IP-адресації на Kali Linux

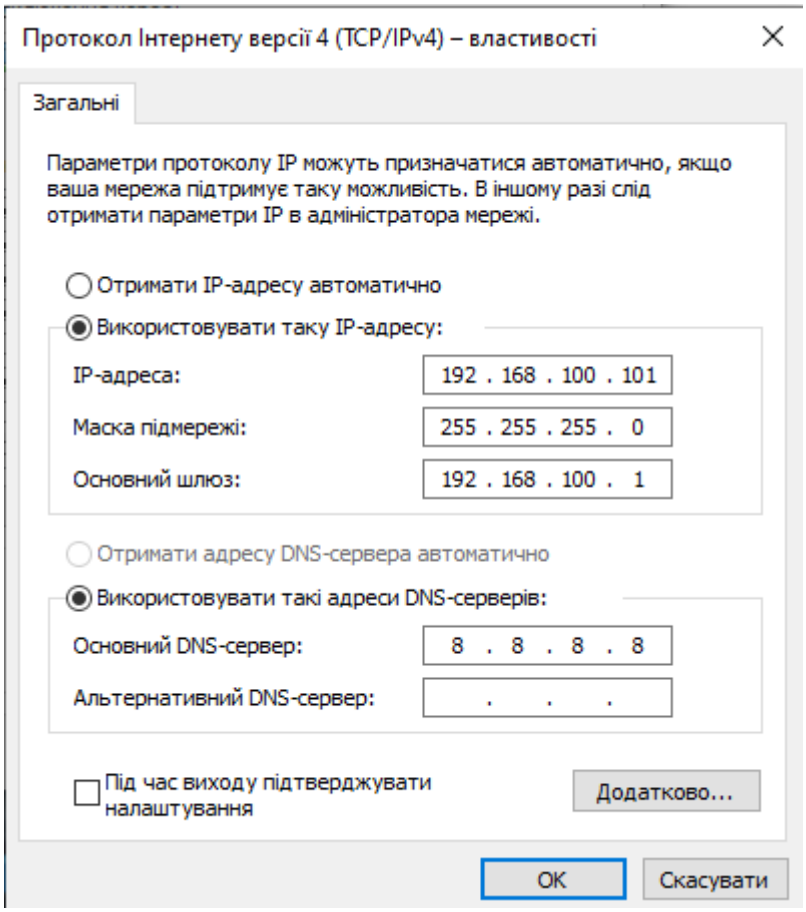


Рисунок 3.7 – Налаштування IP-адресації на Windows 10

Таблиця 3.1

Характеристика вузлів лабораторного стенду

Пристрій	Порт підключення	IP-адреса	Роль у проекті	Тип вузла
Маршрутизатор Cisco	GigabitEthernet 0/0	192.168.100.1	Шлюз, об'єкт моніторингу	Layer 3
Комутатор Cisco	VLAN 1 (Interface)	192.168.100.2	Комутація, об'єкт моніторингу	Layer 2
MacBook (Kali Linux)	f0/10 (Switch)	192.168.100.50	Станція тестування (Attacker)	Фізична
Laptop (Host PC)	f0/20 (Switch)	192.168.100.10	Хост для ВМ	Фізична
Wazuh Server (VM)	Bridged (віртуальний)	192.168.100.100	SIEM / SOC Center	Віртуальна
Windows 10 (VM)	Bridged (віртуальний)	192.168.100.101	Кінцева точка (Victim)	Віртуальна

Архітектура та налаштування Wzuh SIEM

Wazuh SIEM розгорнуто на виділеній віртуальній машині під управлінням Ubuntu 22.04 LTS. Система включає всі основні компоненти платформи: Wazuh Manager для обробки та кореляції подій, Wazuh Indexer на базі OpenSearch для зберігання даних, та Wazuh Dashboard для візуалізації та аналізу.

Wazuh Agent встановлено на вузлі Windows 10 та налаштовано на збір подій з наступних джерел: журнал безпеки Windows (Security Event Log) через формат eventchannel, канал Sysmon (Microsoft-Windows-Sysmon/Operational) для розширеного моніторингу, а також системні журнали для фіксації стану операційної системи.

Конфігурація агента на Windows 10 налаштована на передачу розширеного набору подій. Зокрема, з журналу Security передаються всі події автентифікації, управління обліковими записами та аудиту привілеїв. З каналу Sysmon передаються події про створення процесів, мережні з'єднання та зміни реєстру. Такий підхід забезпечує комплексне покриття потенційних векторів атак.

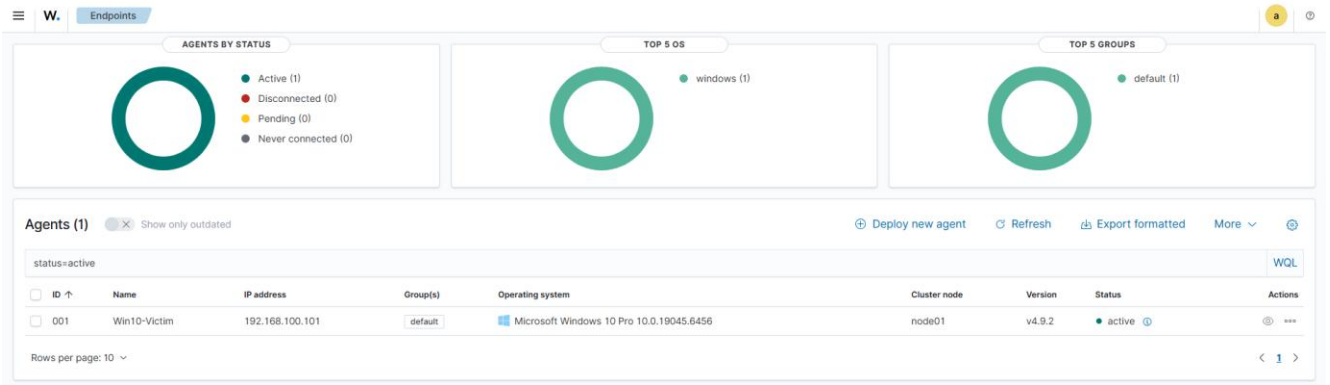


Рисунок 3.8 – Головний інтерфейс Wazuh Dashboard з відображенням підключених агентів

Архітектура взаємодії компонентів побудована таким чином: Wazuh Agent на Windows 10 збирає події та передає їх зашифрованим каналом (порт 1514/UDP) на Wazuh Manager. Manager декодує вхідні повідомлення за допомогою вбудованих та власних декодерів, після чого застосовує правила кореляції для виявлення аномалій та генерації алертів. Алерти зберігаються в Wazuh Indexer та стають доступними через Dashboard для аналітика безпеки.

Розроблені правила виявлення загроз

В рамках дослідження розроблено комплекс власних правил виявлення загроз, які доповнюють вбудовані правила Wazuh. Правила написані мовою XML та розміщені у файлі /var/ossec/etc/rules/local_rules.xml. Кожне правило прив'язане до конкретної техніки атаки згідно з класифікацією MITRE ATT&CK.

При розробці правил застосовано ієрархічний підхід: правила нижнього рівня виявляють одиничні підозрілі події (наприклад, одне заблоковане з'єднання WFP), а правила верхнього рівня агрегують ці події для виявлення складних атак (наприклад, масове сканування портів як сукупність численних заблокованих з'єднань за короткий час). Це дозволяє мінімізувати кількість хибнопозитивних спрацювань.

Окремої уваги заслуговує технічна особливість виявлення подій Windows через Wazuh: події, передані агентом через канал eventchannel, декодуються декодером windows_eventchannel і мають ієрархічну структуру полів виду win.system.eventID та win.eventdata.*. Це потребує використання тегу <field> замість <match> для звернення до конкретних полів декодованого повідомлення, а

також правильного вибору батьківського правила через <if_sid> для забезпечення коректної кореляції.

```
    BRUTE FORCE
═══════════════════════════════════════ -->

<!-- Батько: 60122 = Windows failed logon (built-in Wazuh) -->
<rule id="100100" level="12" frequency="5" timeframe="60">
  <if_matched_sid>60122</if_matched_sid>
  <description>Brute Force: Multiple failed Windows logons</description>
  <mitre><id>T1110</id></mitre>
</rule>

<!-- Батько: 60106 = Windows logon event (built-in Wazuh) -->
<rule id="100101" level="10" frequency="5" timeframe="60">
  <if_matched_sid>60106</if_matched_sid>
  <description>Brute Force: Multiple RDP authentication failures</description>
  <mitre><id>T1110.001</id></mitre>
</rule>

<!-- Батько: 60106 = Windows logon event -->
<rule id="100800" level="14">
  <if_sid>60106</if_sid>
  <field name="win.system.eventID">^4740$</field>
  <description>Account locked out - likely brute force result</description>
  <mitre><id>T1110</id></mitre>
</rule>

    NMAP / PORT SCAN
═══════════════════════════════════════ -->

<!-- Батько: 60104 = Windows audit failure (built-in Wazuh) -->
<rule id="100151" level="3">
  <if_sid>60104</if_sid>
  <field name="win.system.eventID">^5157$</field>
  <description>WFP blocked connection - Event 5157</description>
</rule>

<!-- Батько: 100151 (вище в цьому ж файлі) -->
<rule id="100152" level="10" frequency="10" timeframe="120">
  <if_matched_sid>100151</if_matched_sid>
  <same_field>win.eventdata.sourceAddress</same_field>
  <description>Port Scan detected from $(win.eventdata.sourceAddress)</description>
  <mitre><id>T1046</id></mitre>
</rule>
```

Рисунок 3.9 – Фрагмент файлу local_rules.xml з розробленими правилами виявлення загроз

Для контролю правильності написання правил використовувався інструмент wazuh-logtest, який дозволяє тестувати правила на реальних або синтетичних подіях без необхідності перезапуску сервісу. Це суттєво прискорило процес налагодження та верифікації правил.

Верифікація та готовність стенду

Перед початком експериментів проведено верифікацію готовності стенду. Перевірено зв'язність між усіма вузлами мережі, стан підключення агента Wazuh

(статус Active в консолі управління), передачу подій від агента до сервера, коректне завантаження розроблених правил без синтаксичних помилок.

З маршрутизатора було виконано ехо-запити до комутатора, сервера Wazuh та хоста-жертви. Успішне проходження пакетів підтвердило коректну роботу мережного мосту у гіпервізорі та правильність фізичної комутації.

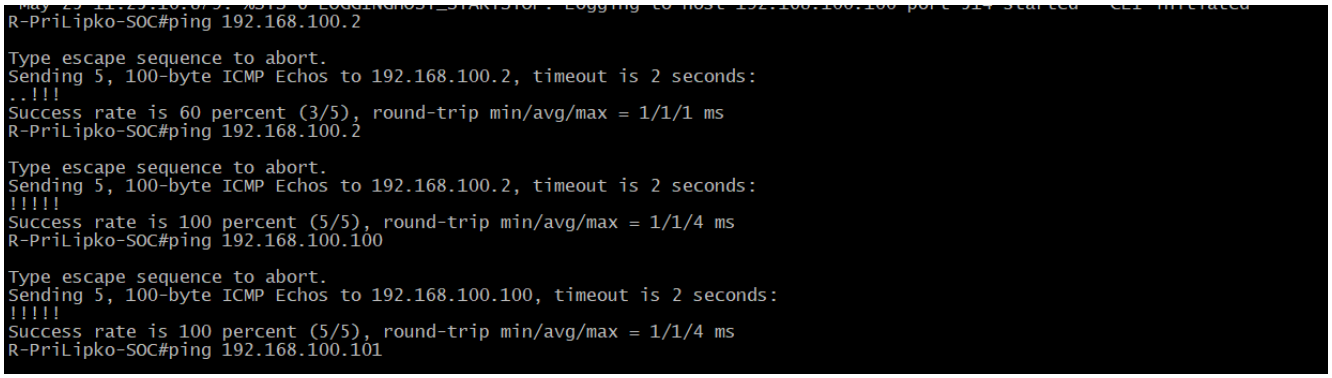


Рисунок 3.10 – Результати перевірки мережної зв'язності між вузлами лабораторного стенду

Крім мережної доступності, перевірено стан підключення агента Wazuh. У консолі управління підтверджено статус Active, що гарантує безперервну передачу подій від агента до сервера, а також коректне завантаження розроблених правил без синтаксичних помилок.

На вузлі Windows 10 налаштовано аудит подій безпеки: увімкнено аудит невдалих та успішних спроб входу, аудит управління обліковими записами, а також аудит фільтрації пакетів Windows Filtering Platform (WFP) для фіксації заблокованих мережних з'єднань (Event ID 5157). Останнє є критично важливим для виявлення сканування портів, оскільки кожна заблокована спроба підключення генерує окрему подію безпеки.

Система Sysmon налаштована з використанням детального конфігураційного файлу, що охоплює широкий спектр підозрілих активностей. Конфігурація включає моніторинг мережних з'єднань для специфічних процесів та портів (включаючи порти 21, 4444, 3389), відстеження змін ключів реєстру автозапуску, моніторинг доступу до процесу lsass.exe та фіксацію створення підозрілих виконуваних файлів.

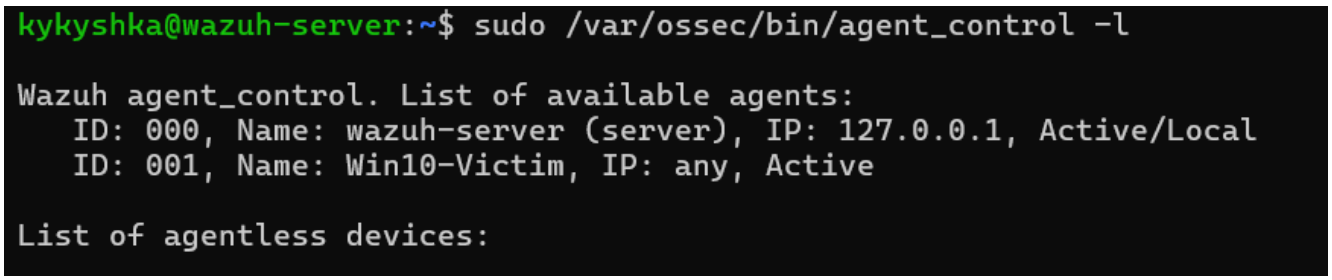


Рисунок 3.11 – Статус агента Win10-Victim у консолі Wazuh – агент активний та передає події

Таким чином, розгорнутий лабораторний стенд являє собою комплексне середовище для дослідження ефективності систем виявлення кіберзагроз. Поєднання реального мережного обладнання Cisco, типової інфраструктури Windows з налаштованими сервісами та централізованої SIEM-системи Wazuh з розробленими правилами виявлення дозволяє проводити репрезентативні експерименти з моделювання кібератак та оцінки їх виявлення.

3.2 Моделювання кібератак та аналіз процесів їх детектування

У даному підрозділі описано практичне моделювання повного ланцюжка кібератаки на лабораторному стенді та проведено детальний аналіз процесів виявлення кожного етапу засобами Wazuh SIEM. Усі атаки виконувалися з вузла Kali Linux проти цільового вузла Windows 10 в умовах активного моніторингу.

Змодельований сценарій відповідає реалістичній моделі зовнішньої атаки на корпоративний вузол і охоплює десять послідовних етапів, починаючи від первинної розвідки та завершуючи знищенням слідів присутності. Кожен етап реалізує одну або декілька технік MITRE ATT&CK та перевіряє відповідні правила виявлення.

Таблиця 3.2

Зведена таблиця змодельованих атак та правил виявлення

№	Тип атаки	Інструмент	MITRE ID	Rule ID	Джерело події
1	Розвідка мережі	Nmap SYN scan	T1046	100151 + 100152	Event 5157 WFP
2	RDP брутфорс	Crowbar	T1110.001	100100,100101	Event 4625 (×5/60c)

Продовження таблиці 3.2

№	Тип атаки	Інструмент	MITRE ID	Rule ID	Джерело події
3	FTP ексфільтрація	ftp client	T1048.003	100702	Sysmon Event 3, port 21
4	Persistence реєстр	reg add Run key	T1547.001	92302 + 92041	Sysmon Event 13 + Event 1
5	Витяг облікових даних	Mimikatz	T1003.001	100201	Sysmon Event 1 (ім'я процесу)
6	Pass-the-Hash	Evil-WinRM	T1550.002	100600	Event 4624 NTLM logonType=3
7	Reverse Shell	msfvenom + MSF	T1059	100804	Sysmon Event 1 (backdoor.exe)
8	Створення бекдор-юзера	net user /add	T1136.001	100703 + 100704	Event 4720 + Event 4732
9	Очищення слідів	wevtutil cl	T1070.001	100705	Event 1102 (лог очищено)

Нижче наведено детальний опис кожного етапу атаки, застосований інструментарій, технічний механізм генерації подій безпеки всередині операційної системи та результати виявлення загрози компонентами Wazuh SIEM.

Етап 1: Розвідка мережі (Network Reconnaissance)

Першим етапом атаки є розвідка цільового вузла з метою визначення активних служб, відкритих TCP/UDP портів та версій операційної системи. Розвідка проводилась інструментом Nmap з вузла Kali Linux з використанням напіввідкритого TCP SYN-сканування (опція -sS) та агресивного темпу генерації пакетів (-T4) для прискорення процесу та створення достатнього обсягу мережної телеметрії для подальшого виявлення. Технічний механізм генерації подій безпеки базується на перехопленні мережних пакетів підсистемою Windows Filtering Platform (WFP), яка є базовим ядром вбудованого брандмауера Windows. При спробі сканування портів, які є закритими або заблокованими політикою безпеки хоста, WFP відхиляє вхідні SYN-пакети та генерує подію відмови з ідентифікатором Event ID 5157 у журналі безпеки (Security). Оскільки Nmap послідовно та з високою швидкістю надсилає запити до сотень і тисяч портів, кожне таке заблоковане з'єднання фіксується як окремий запис ядра, що містить критично важливі поля: sourceAddress (IP-адреса атакуючого вузла), destPort (сканований порт цільової системи) та severityValue зі значенням AUDIT_FAILURE.

Для детектування цієї активності було розроблено дворівневу логіку кореляції подій у Wazuh. Правило 100151 перевіряє кожну вхідну подію Event ID 5157 і класифікує її як одиничне заблоковане з'єднання низького рівня пріоритетності (Level 3). У свою чергу, кореляційне правило 100152 агрегує ці події. Воно налаштоване на генерацію критичного алерта 10-го рівня при накопиченні 10 і більше однотипних подій від одного джерела за проміжок часу тривалістю 120 секунд. Механізм групування реалізовано через тег <same_field>win.eventdata.sourceAddress</same_field>, що дозволяє чітко ідентифікувати саме масове сканування мережі конкретним хостом, виключаючи поодинокі випадкові помилки з'єднання.

Таблиця 3.3

Правила виявлення розвідки мережі

Правило	Рівень	Опис спрацювання
Rule 100151	Level 3	WFP blocked connection – Event 5157. Фіксує кожне заблоковане вхідне з'єднання як окрему підозрілу подію.
Rule 100152	Level 10	Port Scan detected. Агрегує 10+ подій від одного IP за 120 секунд – підтверджує масове сканування портів.

Важливою технічною особливістю реалізації цього етапу є обов'язкове попереднє налаштування політики аудиту в ОС Windows 10. За замовчуванням фіксація заблокованих пакетів вимкнена. Для її активації через утиліту командного рядка було виконано команду auditpol /set /subcategory:"Filtering Platform Connection" /success:enable /failure:enable. Також у конфігураційному файлі Wazuh Agent (ossec.conf) було перевірено, що EventID 5157 не входить до списку ігнорованих подій, а аналіз у правилах здійснюється за чітким посиланням на поле win.system.eventID, що оптимізує швидкість обробки логів менеджером.

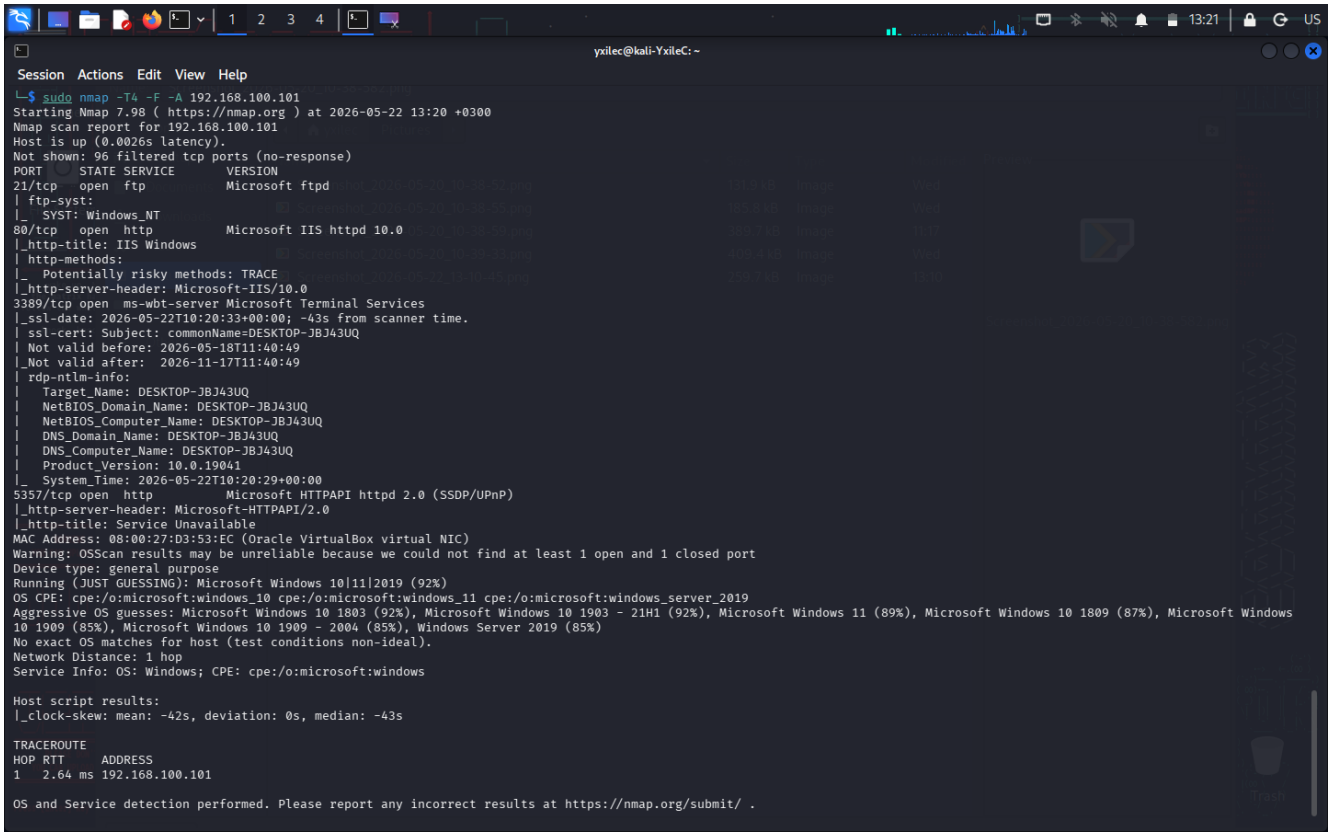


Рисунок 3.12 – Виконання Nmap SYN-сканування портів цільового вузла з Kali Linux

	timestamp	agent.name	rule.description	rule.le...	rule.id
	Jun 2, 2026 @ 22:51:34.976	Win10-Victim	WFP blocked connection - Event 5157	3	100151
	Jun 2, 2026 @ 22:51:34.938	Win10-Victim	WFP blocked connection - Event 5157	3	100151
	Jun 2, 2026 @ 22:51:34.911	Win10-Victim	WFP blocked connection - Event 5157	3	100151
	Jun 2, 2026 @ 22:51:34.879	Win10-Victim	Port Scan detected from 192.168.0.1	10	100152

Рисунок 3.13 – Алерт правила 100151 та 100152 у Wazuh Threat Hunting

Етап 2: Брутфорс та несанкціонований доступ по RDP

Після завершення стадії розвідки було визначено, що на цільовій машині відкритий стандартний порт 3389 для підключення до віддаленого робочого столу (RDP). З метою отримання первинного доступу до системи було виконано атаку перебору паролів (brute force) щодо легітимного облікового запису користувача UserPriLipko. Для автоматизації та проведення брутфорсу використано спеціалізований інструмент Crowbar. Його вибір зумовлений тим, що він оптимізований для

роботи зі специфікою протоколу RDP і коректно взаємодіє з механізмами автентифікації на рівні мережі (Network Level Authentication – NLA), які реалізовані у сучасних версіях Windows 10. Під час проведення атаки використовувався сформований словник паролів, що містив як завідомо хибні комбінації, так і один правильний пароль для реалістичного відтворення процесу інтенсивного перебору.

Механізм логування в ОС Windows реагує на кожну невдалу спробу підключення інструменту Crowbar генерацією події Event ID 4625 (An account failed to log on) у журналі Security[13]. Ця подія містить такі аналітичні дані, як тип входу (Logon Type = 3 або 10 залежно від етапу перевірки NLA), назву цільового облікового запису та IP-адресу джерела запиту. На першому рівні обробки телеметрії у Wazuh спрацьовує правило 100100, яке фіксує кожну таку поодиноку помилку автентифікації. Для детектування безпосередньо самого факту атаки розроблене кореляційне правило 100101, яке аналізує інтенсивність цих подій і генерує алерт 10-го рівня, якщо фіксується 5 або більше невдалих спроб входу для одного користувача протягом однієї хвилини.

Таблиця 3.4

Правила виявлення RDP брутфорсу та несанкціонованого доступу

Правило	Рівень	Опис спрацювання
Rule 100100	Level 12	Brute Force: Multiple failed Windows logons. Фіксує кожну одиничну невдалу спробу входу в систему через Event ID 4625
Rule 100101	Level 10	Multiple RDP authentication failures. Агрегує події та фіксує брутфорс (5+ невдалих спроб за 60 секунд від одного джерела).



Рисунок 3.14 – Результат роботи Crowbar – успішний підбір пароля облікового запису UserPriLipko

	May 22, 2026 @ 10:08:34.790	Win10-Victim	Brute Force: Multiple RDP authentication failures	10	100101
	May 22, 2026 @ 10:07:55.829	Win10-Victim	Brute Force: Multiple RDP authentication failures	10	100101
	May 20, 2026 @ 12:12:27.141	Win10-Victim	Brute Force: Multiple RDP authentication failures	10	100101
	May 20, 2026 @ 10:56:36.359	Win10-Victim	Brute Force: Multiple failed Windows logons	12	100100
	May 20, 2026 @ 10:56:36.211	Win10-Victim	Brute Force: Multiple failed Windows logons	12	100100

Рисунок 3.15 – Алерти Wazuh – виявлення серії помилок автентифікації та фіксація RDP-брутфорсу

Під час практичного розбору інциденту в модулі Threat Hunting було проаналізовано зміст алертів правила 100101. У розгорнутій структурі логу зафіксовано, що поле win.eventdata.targetUserName містить значення kykys. Таким чином, SIEM-система здатна в режимі реального часу сигналізувати аналітикам про цілеспрямовану атаку на конкретні облікові записи працівників.

Етап 3: FTP ескалація

Маючи доступ до системи через сесію UserPriLipko, зловмисник використав наявний FTP-сервіс (Microsoft IIS FTP) для завантаження конфіденційних файлів з цільового вузла на станцію атакуючого. На FTP-сервері розміщені файли secret.txt та finances.txt, що імітують конфіденційні корпоративні дані.

З'єднання встановлювалось командним FTP-клієнтом Kali Linux з обов'язковим переключенням у бінарний режим передачі для коректної передачі файлів. Після успішної автентифікації виконано завантаження обох файлів командою get.

Виявлення ексфільтрації базується на моніторингу мережних з'єднань Sysmon (Event ID 3). Правило 100702 успадковує від вбудованого правила Wazuh 61605 (Sysmon Event 3: Network connection) та перевіряє поле destinationPort на значення 21. При встановленні FTP-з'єднання Sysmon фіксує мережне підключення до порту 21 та передає подію агенту Wazuh, який генерує алерт рівня 10 з описом «FTP connection detected – possible exfiltration».

Таблиця 3.5

Правила виявлення FTP ескалації

Правило	Рівень	Опис спрацювання
Rule 100702	Level 10	FTP connection detected. Фіксує ініціалізацію мережного з'єднання на порт 21 (FTP) через телеметрію Sysmon Event ID 3, що є прямим індикатором можливої ексфільтрації даних.

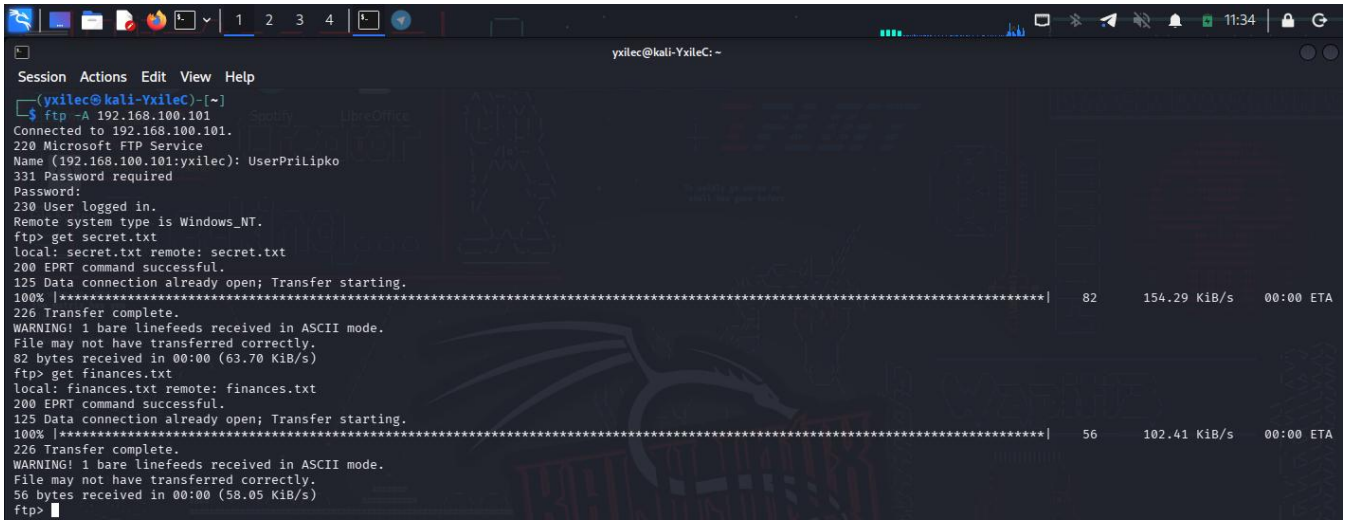


Рисунок 3.16 – Завантаження конфіденційних файлів з FTP-сервера Windows на станцію атакуючого Kali Linux

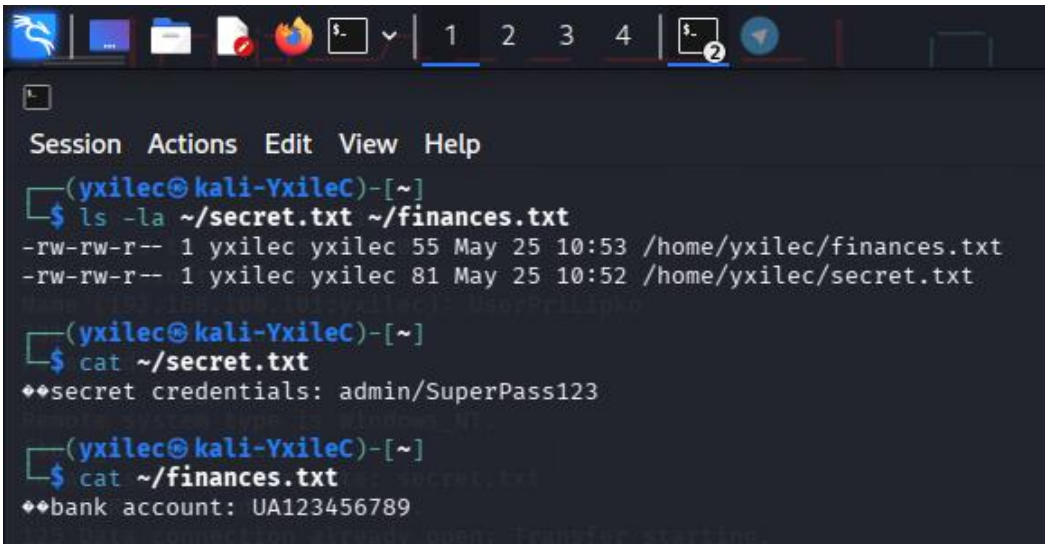


Рисунок 3.17 – Успішно викачані файли зловмисником та прочитані

	↓ timestamp	agent.name	rule.description	rule.le...	rule.id
	May 25, 2026 @ 13:45:01.197	Win10-Victim	FTP connection detected - possible exfiltration to 192.168.100.101	10	100702
	May 25, 2026 @ 13:06:28.731	Win10-Victim	FTP connection detected - possible exfiltration to 192.168.100.101	10	100702
	May 25, 2026 @ 12:26:44.125	Win10-Victim	FTP connection detected - possible exfiltration to 192.168.100.101	10	100702
	May 25, 2026 @ 12:24:49.387	Win10-Victim	FTP connection detected - possible exfiltration to 192.168.100.101	10	100702
	May 25, 2026 @ 12:06:44.640	Win10-Victim	FTP connection detected - possible exfiltration to 192.168.100.101	10	100702

Рисунок 3.18 – Алерт правила 100702 у Wazuh – виявлення FTP з'єднання з ознаками ексфільтрації даних

Технічною складністю при налаштуванні виявлення FTP стала необхідність додавання порту 21 до конфігурації Sysmon у секції NetworkConnect, а також включення процесу svchost.exe до списку моніторингу мережних з'єднань, оскільки саме через нього FTP-сервер IIS обробляє вхідні підключення. Без цих налаштувань Sysmon не генерував би відповідних подій, і виявлення було б неможливим.

Етап 4: Закріплення в системі (Persistence)

Для забезпечення постійного доступу до скомпрометованої системи після її перезавантаження зловмисник додав запис до ключа реєстру автозапуску HKCU\Software\Microsoft\Windows\CurrentVersion\Run з іменем WindowsUpdate та значенням C:\backdoor.exe. Така техніка імітує типовий спосіб закріплення шкідливого програмного забезпечення в системі під виглядом легітимного оновлення Windows.

Sysmon відстежує зміни реєстру через події типу RegistryEvent (Event ID 13 – SetValue). Конфігурація Sysmon містить правило моніторингу, яке фіксує будь-які записи у ключі CurrentVersion\Run та пов'язаних ключах автозапуску. При виконанні команди reg add Sysmon генерував дві події: Event ID 1 (ProcessCreate для reg.exe) та Event ID 13 (RegistryEvent для модифікованого ключа).

Таблиця 3.6

Правила виявлення закріплення в системі через реєстр

Правило	Рівень	Опис спрацювання
Rule 92302	Level 6	Вбудоване правило Wazuh, яке відстежує зміни ключів реєстру автозапуску. Фіксує безпосередній запис шкідливого значення у ключ CurrentVersion\Run через телеметрію Sysmon Event ID 13 (RegistryEvent).
Rule 92041	Level 10	Вбудоване правило Wazuh, яке фіксує факт запуску та виконання утиліти reg.exe в системі за допомогою телеметрії Sysmon Event ID 1 (ProcessCreate).

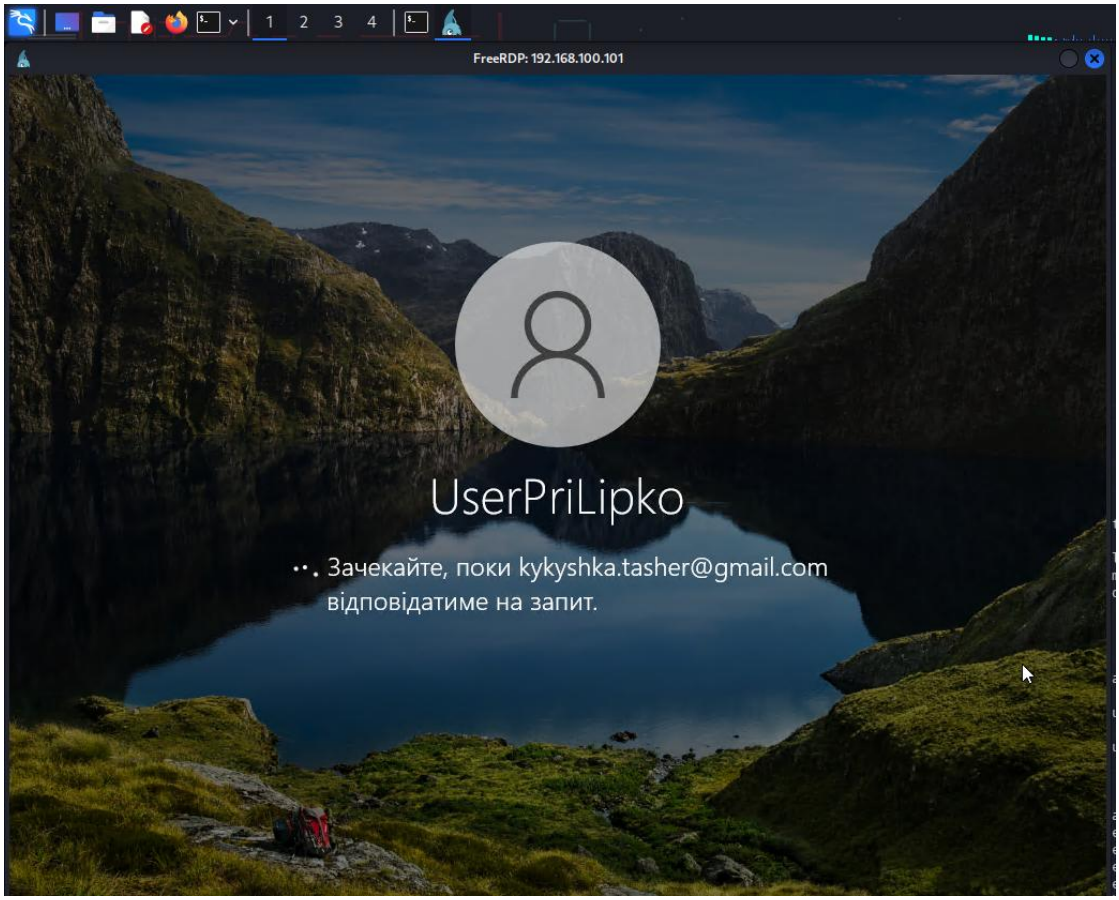


Рисунок 3.19 – Успішне RDP-підключення до жертви

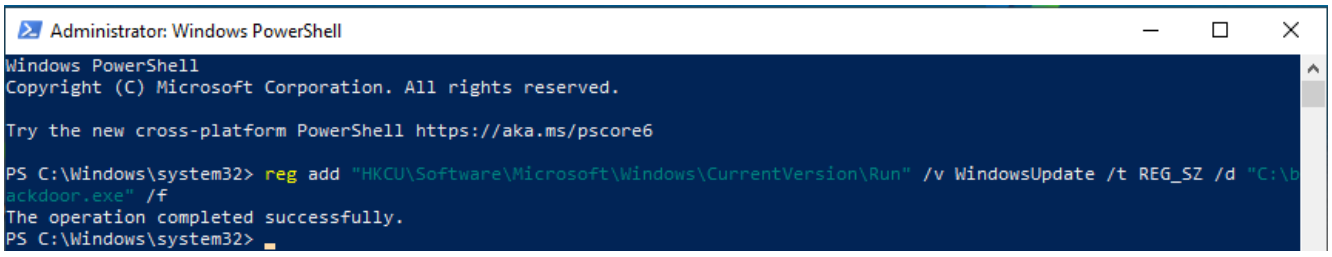


Рисунок 3.20 – Додавання запису до реєстру автозапуску

	timestamp	agent.name	rule.description	rul...	rule.id
	May 25, 2026 @ 12:36:13.989	Win10-Victim	Value added to registry key has Base64-like pattern	10	92041
	May 25, 2026 @ 12:16:03.999	Win10-Victim	Registry entry to be executed on next logon was modified using command line application reg.exe	6	92302

Рисунок 3.21 – Спрацювання відповідних алертів 92302 та 92041

Wazuh виявив цю дію через вбудоване правило 92302 (рівень 7), яке відстежує зміни ключів реєстру автозапуску, та правило 92041 (рівень 4), яке фіксує виконання reg.exe. Розроблене власне правило 100501 також спрацювало, виявивши командний рядок reg.exe з параметрами, що вказують на модифікацію ключа Run.

Етап 5: Витяг облікових даних

Центральним і найбільш критичним етапом атаки є витяг облікових даних за допомогою інструменту Mimikatz. Перед запуском Mimikatz необхідно було отримати права адміністратора: вхід виконано під обліковим записом Administrator через RDP. Оскільки Windows Defender автоматично виявляє та видаляє Mimikatz, перед розпакуванням та запуском було вимкнено захист реального часу командою Set-MpPreference -DisableRealtimeMonitoring $true.

Mimikatz запущено з командами privilege::debug для отримання права налагодження процесів та sekurlsa::logonpasswords для читання даних автентифікації з пам'яті процесу lsass.exe. Результат виконання збережено у файл C:\FTP\passwords.txt командою Out-File та завантажено на станцію атакуючого через FTP для подальшого аналізу.

Wazuh виявив запуск Mimikatz через два незалежних механізми. По-перше, правило 100201 (рівень 15) спрацювало на основі Sysmon Event ID 1, зафіксувавши ім'я процесу mimikatz.exe у полі win.eventdata.image. По-друге, вбудоване правило 92203 (рівень 6) виявило створення файлів бібліотек Mimikatz (mimispool.dll, mimilib.dll тощо) в системі.

Таблиця 3.7

Правила витягу облікових даних

Правило	Рівень	Опис спрацювання
Rule 100201	Level 15	Credential dumping tool detected by name. Sysmon Event 1 фіксує запуск процесу з іменем mimikatz.exe.
Rule 92203	Level 6	Mimikatz files detected. Вбудоване правило безпеки Wazuh. Реагує на події створення нових об'єктів у файловій системі та фіксує появу специфічних супутніх компонентів і динамічних бібліотек, що належать сімейству хакерського інструменту Mimikatz.

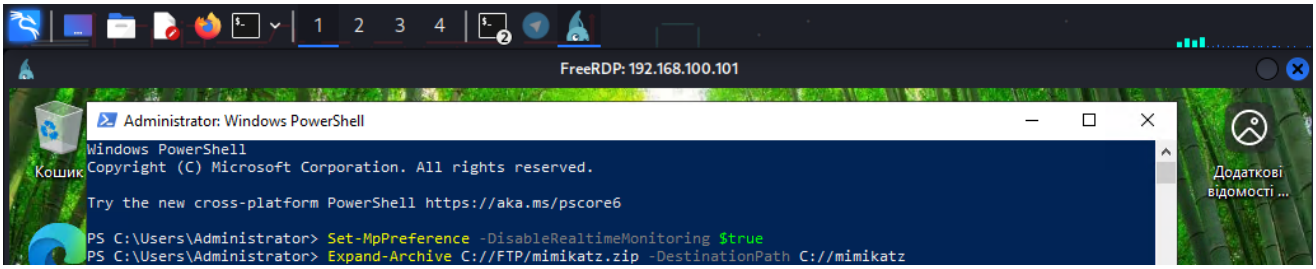


Рисунок 3.22 – Деактивація антивірусного моніторингу Windows Defender за допомогою PowerShell

Після успішного відключення захисних компонентів операційної системи було запущено утиліту mimikatz.exe в інтерактивному режимі. Зловмисник послідовно застосував команду privilege::debug для отримання прав на налагодження системних процесів (SeDebugPrivilege) та команду sekurlsa::logonpasswords для безпосереднього зчитування та парсингу даних автентифікації, що зберігаються в оперативній пам'яті захищеного процесу lsass.exe (Local Security Authority Subsystem Service).

Отриманий масив даних, що містив відкриті паролі та NTLM-хеші користувачів системи, був перенаправлений і збережений у локальний текстовий файл за допомогою штатного інструменту копіювання виводу.

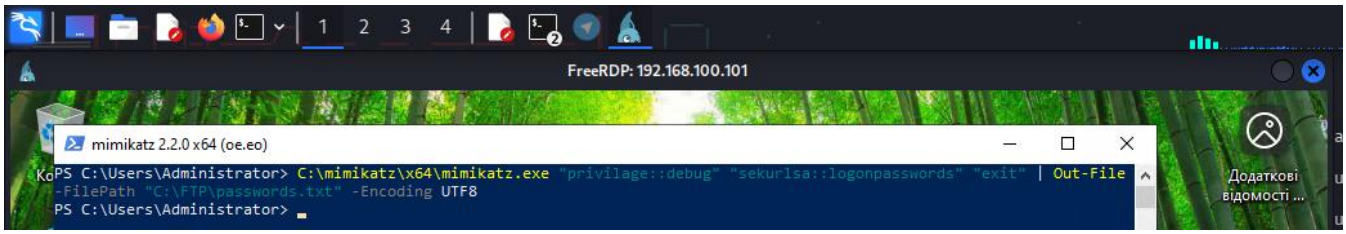


Рисунок 3.23 – Успішна експлуатація пам'яті процесу lsass.exe та збереження хешів у файл passwords.txt

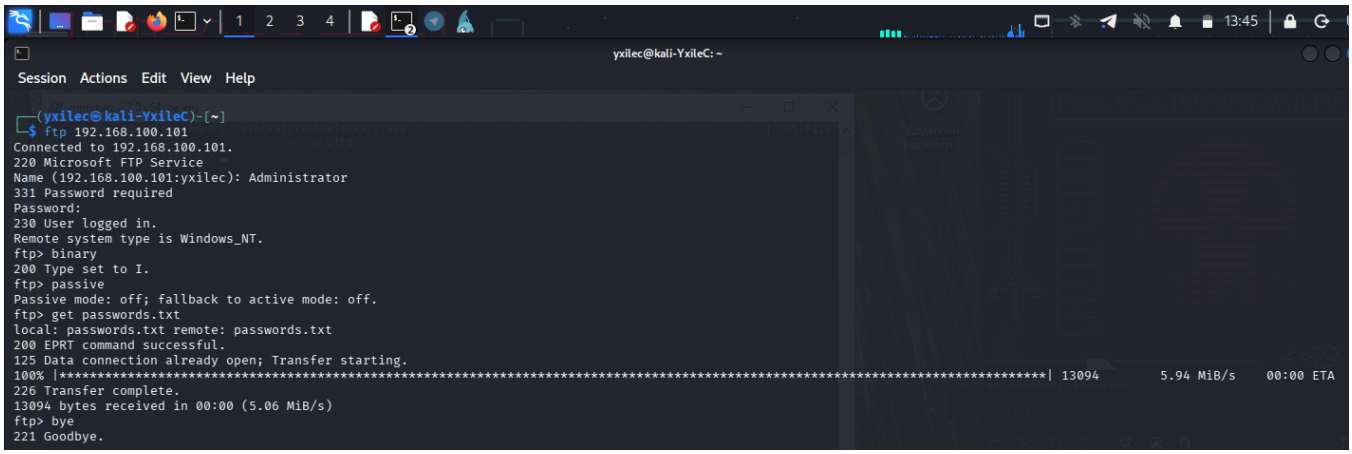


Рисунок 3.24 – Успішно викачаний файл passwords.txt на машину зловмисника

Згенерований файл з паролями зловмисник оперативно завантажив на власну атакувальну станцію через раніше налаштований FTP-клієнт для подальшого аналізу та підготовки наступних кроків експлуатації.

	timestamp	agent.name	rule.description	rule.l...	rule.id
	May 25, 2026 @ 13:44:30.430	Win10-Victim	Credential dumping tool detected by name	15	100201
	May 25, 2026 @ 13:31:21.300	Win10-Victim	Credential dumping tool detected by name	15	100201
	May 25, 2026 @ 13:18:52.744	Win10-Victim	Credential dumping tool detected by name	15	100201
	May 25, 2026 @ 13:18:34.642	Win10-Victim	Executable file created by powershell: C:\\mimikatz\\x64\\mimispool.dll	6	92203

Рисунок 3.25 – Оповіщення SIEM Wazuh про виявлення інструменту Credential Dumping та правила 92203

Система моніторингу Wazuh SIEM успішно виявила факт запуску шкідливого інструментарію за допомогою двох незалежних аналітичних механізмів, що забезпечило надійну детекцію інциденту. Головним індикатором виступило розроблене правило 100201, яке функціонує на основі журналів моніторингу процесів Sysmon Event ID 1. Дане правило аналізує метадані та аргументи нових процесів і миттєво реагує, якщо у полі win.eventdata.image фіксується сигнатурне ім'я файлу mimikatz.exe.

Паралельно з цим, вбудоване базове правило аналізу файлової системи 92203 зафіксувало аномальну активність, пов'язану зі створенням та розпакуванням специфічних бібліотек утиліти (таких як mimispool.dll, mimilib.dll тощо) у тимчасових директоріях операційної системи, що додатково підтвердило присутність Mimikatz на вузлі Win10-Victim.

Етап 6: Атака Pass-the-hash

Здобувши NTLM-хеш облікового запису адміністратора на попередніх етапах компрометації зловмисник отримав можливість реалізувати техніку Pass-the-Hash. Особливість цієї атаки полягає в тому, що архітектура автентифікації Windows NTLM базується на перевірці знання криптографічного хешу, а не самого пароля у відкритому вигляді. Відповідно, зловмиснику не потрібно витрачати час та обчислювальні ресурси на брутфорс чи дешифрування складного пароля – самого хешу достатньо для повноцінної перевірки автентичності.

Для здійснення горизонтального переміщення (Lateral Movement) всередині внутрішнього периметру мережі було використано спеціалізовану утиліту Evil-WinRM. Цей інструмент є популярним фреймворком для проведення тестування на проникнення і задіює легітимний протокол Windows Remote Management, який за замовчуванням функціонує на портах 5985 (HTTP) та 5986 (HTTPS). Завдяки вбудованій підтримці передачі NTLM-хешів, Evil-WinRM дозволяє ініціювати віддалений запит, обходячи стандартні механізми введення пароля, та у разі успіху надає атакуючому повноцінну інтерактивну сесію PowerShell із найвищими привілеями на цільовому хості.

Таблиця 3.8

Правило виявлення підозрілої віддаленої автентифікації через WinRM

Правило	Рівень	Опис спрацювання
Rule 100600	Level 12	Possible Pass-the-Hash: NTLM network logon detected

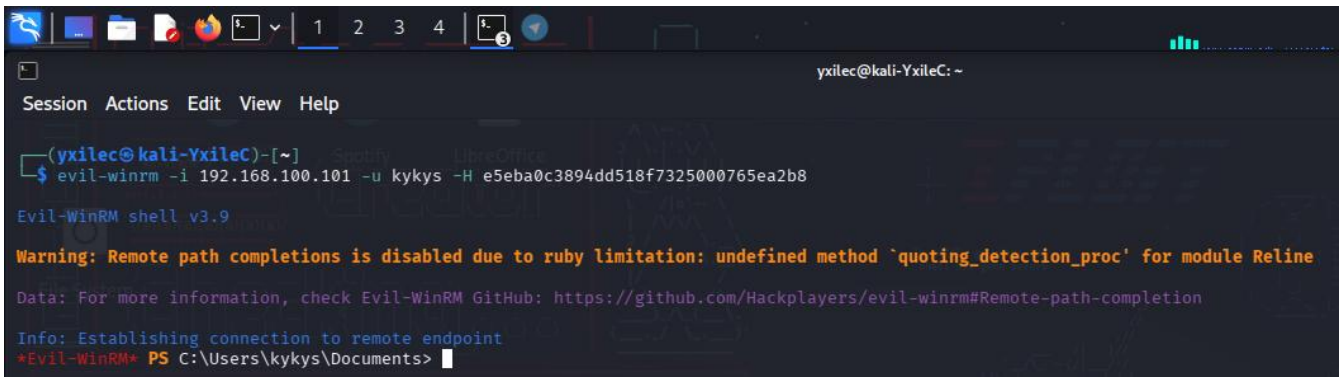


Рисунок 3.26 – Успішне підключення через Evil-WinRM

Після встановлення сесії було виконано низку перевірочних команд, зокрема визначення поточного користувача, перегляд інформації про систему та перевірка рівня привілеїв. Отримані результати підтвердили успішне виконання атаки та можливість повноцінного керування віддаленим вузлом.

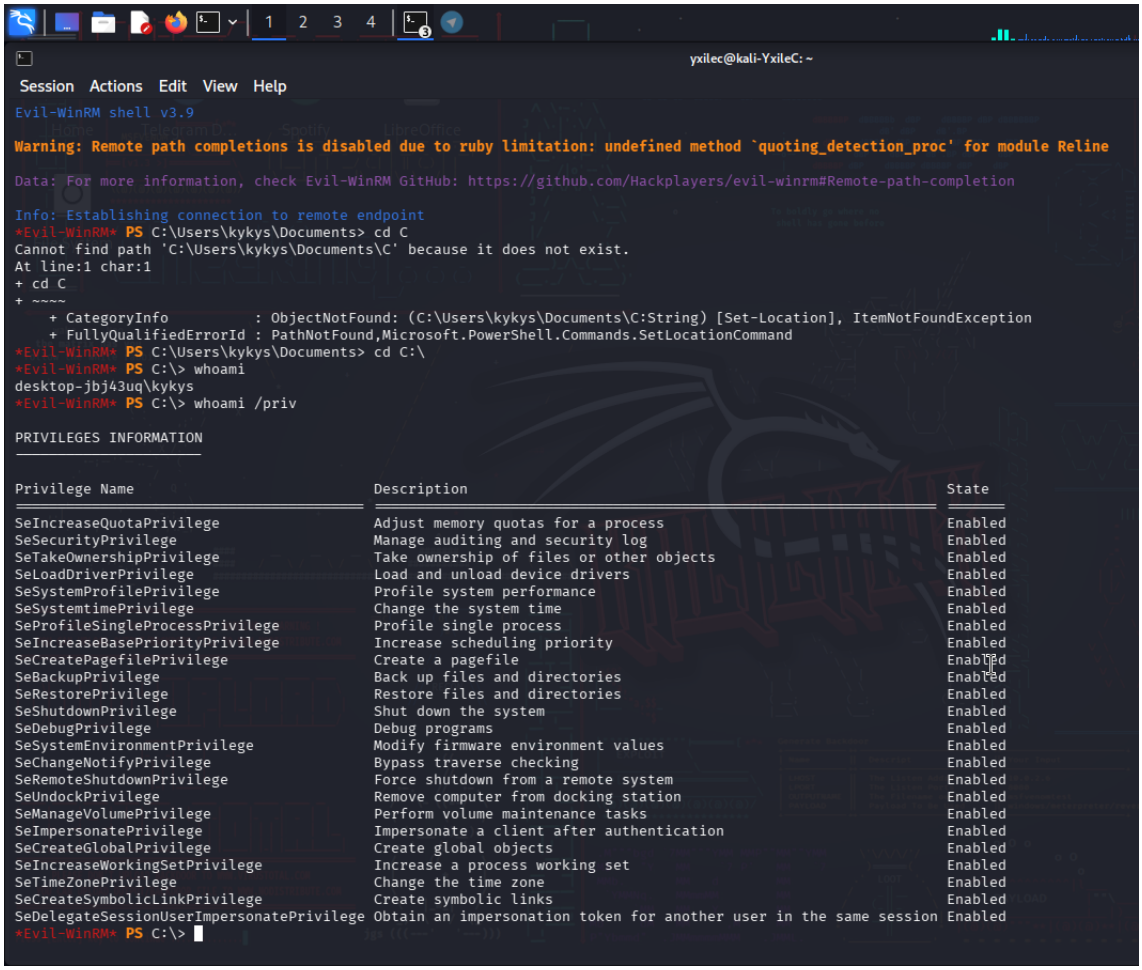


Рисунок 3.27 – Виконані перевірочні команди

Для підвищення точності виявлення було використано власне правило кореляції Wazuh, яке аналізувало поєднання ознак віддаленого входу адміністратора та запуску процесів PowerShell через WinRM. У результаті спрацювання правила система сформувала попередження підвищеного рівня критичності та відобразила його в інтерфейсі Wazuh Dashboard.

У сформованому алерті були відображені IP-адреса джерела підключення, ім'я користувача, тип автентифікації, час події та рівень небезпеки. Це дозволило оператору системи швидко встановити факт використання адміністративного облікового запису з нетипового джерела та розпочати реагування на інцидент.

	timestamp	agent.name	rule.description	rule.le...	rule.id
	May 25, 2026 @ 14:31:19.072	Win10-Victim	Possible Pass-the-Hash: NTLM network logon detected	12	100600
	May 25, 2026 @ 14:30:15.681	Win10-Victim	Possible Pass-the-Hash: NTLM network logon detected	12	100600

Рисунок 3.28 – Спрацювання правила Wazuh щодо підозрілої віддаленої автентифікації через WinRM

Проведений експеримент підтвердив можливість успішного виявлення спроб горизонтального переміщення в мережі шляхом використання NTLM-хешів. Система Wazuh забезпечила фіксацію як факту автентифікації, так і подальшої активності користувача після входу в систему. Отримані результати демонструють ефективність використання централізованого моніторингу подій безпеки для виявлення одного з найбільш поширених методів розвитку атаки після компрометації облікових даних[14].

Етап 7: Reverse Shell та повний контроль над системою

Для отримання інтерактивного контролю над скомпрометованим вузлом без використання RDP було розгорнуто reverse shell на основі Metasploit Framework. На станції атакуючого підготовлено слухач (listener) у msfconsole та згенеровано шкідливий виконуваний файл backdoor.exe командою msfvenom з параметрами payload windows/x64/meterpreter/reverse_tcp, LHOST 192.168.100.50 та LPORT 4444.

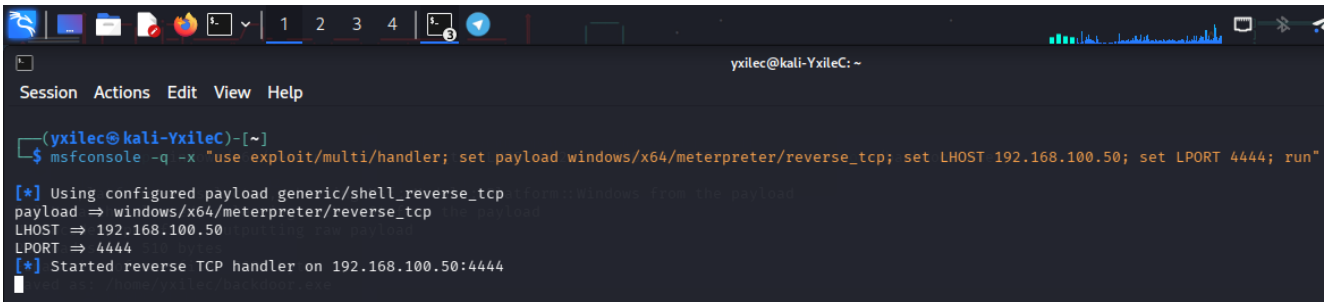


Рисунок 3.29 – Підготовлений слухач у msfconsole

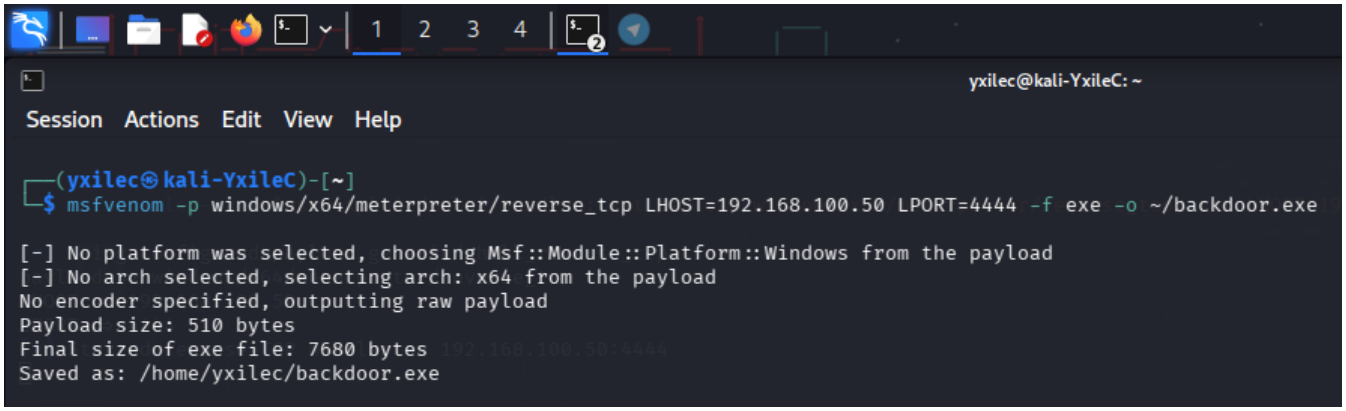


Рисунок 3.30 – Згенерований шкідливий виконуваний файл backdoor.exe

Файл backdoor.exe завантажено на цільовий вузол через Evil-WinRM за допомогою команди upload та запущено командою Start-Process. Після запуску

backdoor.exe встановив зворотнє TCP-з'єднання на порт 4444 станції атакуючого, відкривши сесію Meterpreter.

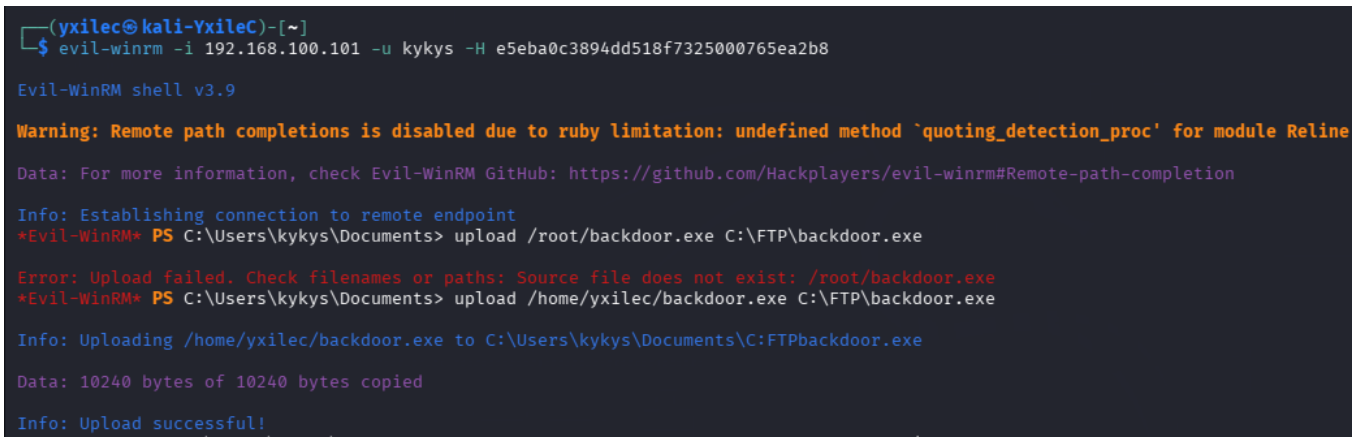


Рисунок 3.31 – Завантажений файл backdoor.exe на цільовий вузол через Evil-WinRM

Meterpreter надав зловмиснику повний доступ до файлової системи, можливість виконання довільних команд, знімки екрану та інші функції постексплуатації.

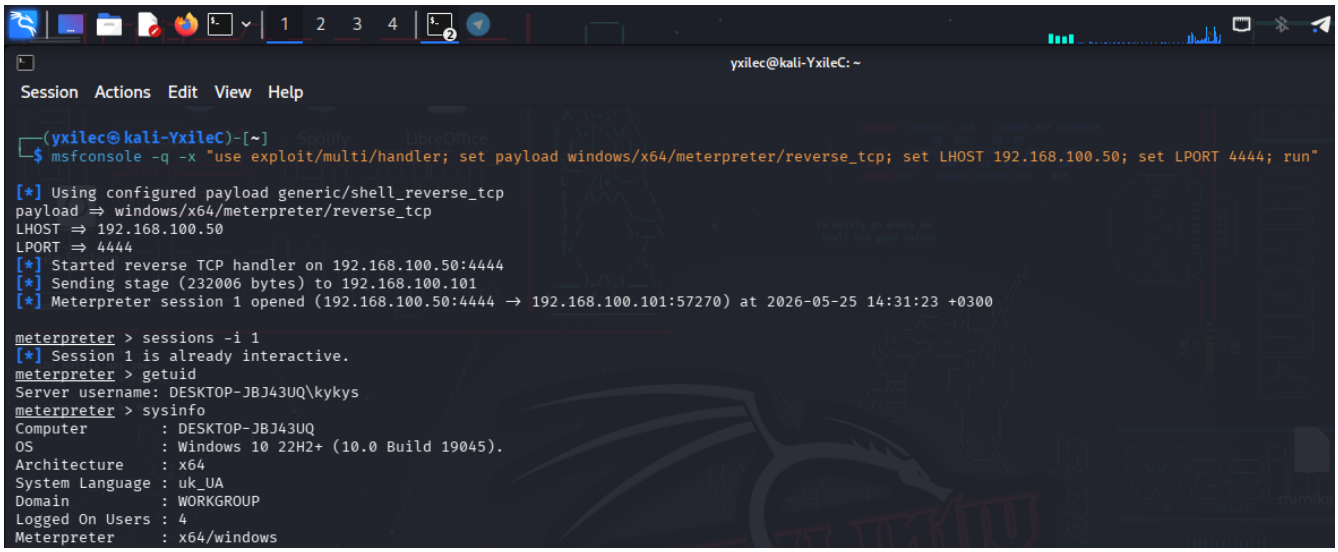


Рисунок 3.32 – Активна сесія Meterpreter у Metasploit – повний контроль над цільовим вузлом

Wazuh виявив цю атаку через правило 100804, яке аналізує командні рядки процесів за регулярним виразом, що охоплює типові ознаки Metasploit payloads та reverse shell. Sysmon зафіксував запуск backdoor.exe як нового процесу (Event ID 1)

з батьківським процесом wsmprovhost.exe (процес WinRM), що само по собі є аномальним та підозрілим.

Таблиця 3.9

Правило виявлення Reverse Shell та дій після компрометації

Правило	Рівень	Опис спрацювання
Rule 100804	Level 14	Reverse shell or Metasploit payload detected. Sysmon Event 1 фіксує запуск backdoor.exe з підозрілим батьківським процесом.

May 25, 2026 @ 13:31:59.127	Win10-Victim	Reverse shell or Metasploit payload detected	14	100804
May 25, 2026 @ 13:31:09.705	Win10-Victim	Reverse shell or Metasploit payload detected	14	100804

Рисунок 3.33 – Алерт правила 100804 у Wazuh рівня 14 – виявлення Reverse Shell

Етап 8: Створення бекдор-юзера

Після отримання повного контролю над цільовою системою через Meterpreter наступним кроком було забезпечення довготривалого доступу до скомпрометованого вузла. Для цього змодельовано одну з типових технік закріплення (Persistence), яка полягає у створенні прихованого адміністративного облікового запису, що дозволяє зловмиснику повторно отримати доступ до системи навіть після завершення активної сесії або видалення шкідливого програмного забезпечення.

Реалізація атаки виконувалася через активну Meterpreter-сесію, отриману на попередньому етапі. Для створення нового користувача було використано стандартні команди операційної системи Windows. У системі створено обліковий запис HackerAdmin із довільним складним паролем, після чого новий користувач був доданий до локальної групи Administrators.

Результатом виконання команд стало створення нового локального облікового запису з повними адміністративними привілеями. Таким чином зловмисник отримав альтернативний канал доступу до системи, який не залежить від початкового способу компрометації та може використовуватися для повторного входу в майбутньому.

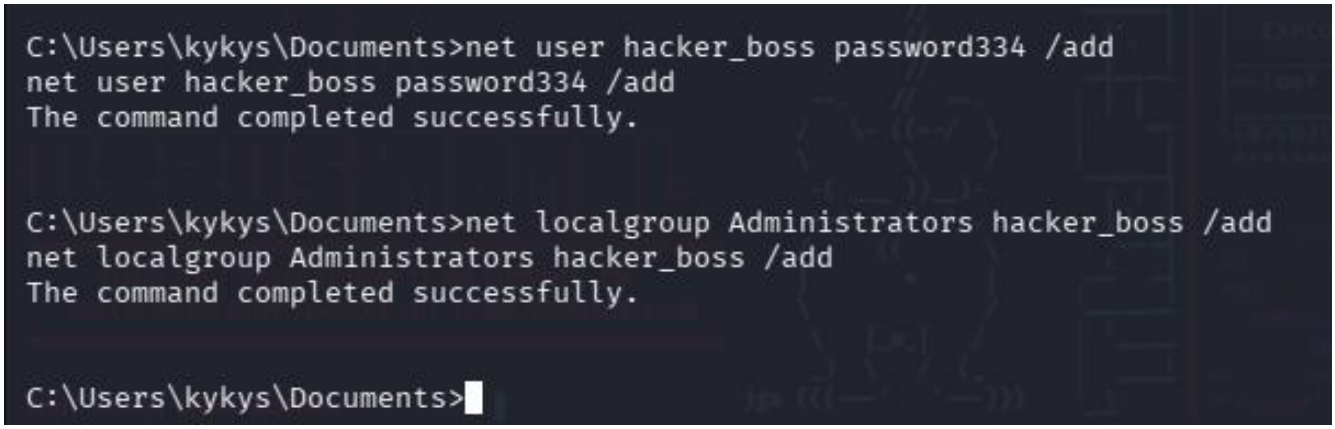


Рисунок 3.34 – Створення облікового запису Hacker_boss через Meterpreter-сесію

Одночасно із виконанням атаки система Wazuh здійснювала моніторинг журналів безпеки Windows. Створення нового облікового запису було зафіксовано подією Security Event ID 4720, яка генерується операційною системою при додаванні нового локального користувача. Після надходження цієї події до Wazuh спрацювало власне правило кореляції 100703, яке класифікує подібні дії як потенційно небезпечні та формує попередження підвищеного рівня критичності.

Таблиця 3.10

Правило виявлення Reverse Shell та дій після компрометації

Правило	Рівень	Опис спрацювання
Rule 100703	Level 12	New user account created. Event 4720 фіксує створення нового облікового запису HackerAdmin зловмисником.
Rule 100704	Level 13	User added to Administrators group. Event 4732 фіксує надання адміністративних прав новому обліковому запису.

Наступним кроком атакуючий надав новому користувачу адміністративні привілеї шляхом додавання його до локальної групи Administrators. Ця дія була зафіксована журналом безпеки Windows як подія Event ID 4732, що відповідає додаванню користувача до привілейованої групи.

Після отримання події сервер Wazuh виконав її обробку та сформував сповіщення на основі правила 100704. Оскільки надання адміністративних привілеїв є більш небезпечною дією порівняно зі звичайним створенням користувача, правилу було призначено вищий рівень критичності.

	timestamp	agent.name	rule.description	rule.l...	rule.id
	May 25, 2026 @ 15:03:24.458	Win10-Victim	User added to Administrators group via command line (Privilege Escalation)	13	100704
	May 25, 2026 @ 15:03:24.431	Win10-Victim	User added to Administrators group via command line (Privilege Escalation)	13	100704
	May 25, 2026 @ 15:02:55.393	Win10-Victim	New user created via command line (net user /add)	12	100703
	May 25, 2026 @ 15:02:55.390	Win10-Victim	New user created via command line (net user /add)	12	100703

Рисунок 3.35 – Алерт правила 100704 та 100703 у Wazuh – виявлення створення нового користувача та додавання його до групи Administrators

Додатковий аналіз журналів подій дозволив встановити часовий зв'язок між створенням нового облікового запису та подальшим наданням йому адміністративних привілеїв. Така послідовність дій є характерною ознакою спроби закріплення в системі після успішної компрометації та часто використовується зловмисниками для забезпечення довготривалого доступу до мережної інфраструктури.

Проведений експеримент підтвердив ефективність системи Wazuh щодо виявлення технік закріплення в системі. Завдяки аналізу журналів безпеки Windows та використанню власних правил кореляції було успішно зафіксовано як факт створення нового облікового запису, так і момент отримання ним адміністративних привілеїв. Це дозволяє своєчасно виявляти спроби прихованого закріплення зловмисника та оперативно реагувати на інциденти інформаційної безпеки.

Етап 9: Очищення слідів

Після отримання повного контролю над системою та створення механізму закріплення зловмисник переходить до етапу приховування своєї активності. Одним із найпоширеніших методів уникнення подальшого розслідування є очищення журналів подій Windows, які містять інформацію про виконані команди, спроби автентифікації, створення облікових записів та інші дії в системі.

У межах експерименту для імітації даного етапу атаки було використано стандартну системну утиліту Windows wevtutil.exe, яка призначена для керування журналами подій. За допомогою командного рядка було виконано очищення основних журналів операційної системи.

Використання утиліти wevtutil є типовою технікою приховування слідів після компрометації системи та відповідає техніці MITRE ATT&CK T1070.001 – Clear Windows Event Logs[19]. Очищення журналів дозволяє зловмиснику видалити

частину доказів своєї діяльності та ускладнити подальший аналіз інциденту спеціалістами з кібербезпеки.

Незважаючи на спробу видалення журналів, система моніторингу Wazuh успішно виявила виконання даної команди. Завдяки інтеграції із Sysmon було зафіксовано створення нового процесу wevtutil.exe (Event ID 1), а власне правило кореляції 100705 проаналізувало командний рядок процесу та визначило факт очищення журналів подій Windows.

Після обробки події система сформувала алерт високого рівня критичності. Правило було прив'язане до тактики Defense Evasion та техніки T1070.001 – Clear Windows Event Logs, що дозволило автоматично класифікувати інцидент відповідно до матриці MITRE ATT&CK.

Таблиця 3.11

Правило виявлення очищення журналів подій Windows

Правило	Рівень	Опис спрацювання
Rule 100705	Level 14	Defense Evasion: Windows Event Logs cleared via wevtutil. Виявлено запуск утиліти wevtutil.exe з параметром очищення журналів подій Windows

	timestamp	agent.name	rule.description	rule.le...	rule.id
	May 25, 2026 @ 15:09:52.648	Win10-Victim	Defense Evasion: Windows Event Logs cleared via wevtutil	14	100705
	May 25, 2026 @ 15:09:40.990	Win10-Victim	Defense Evasion: Windows Event Logs cleared via wevtutil	14	100705
	May 25, 2026 @ 15:09:15.146	Win10-Victim	Defense Evasion: Windows Event Logs cleared via wevtutil	14	100705

Рисунок 3.36 – Алерт правила 100705 у Wazuh рівня 14 – виявлення спроби очищення журналів подій Windows

Отримані результати демонструють, що навіть у випадку спроби зловмисника приховати свою діяльність шляхом очищення системних журналів, система моніторингу Wazuh здатна зафіксувати сам факт виконання відповідної команди та своєчасно повідомити адміністратора безпеки про потенційний інцидент. Це підтверджує ефективність використання Sysmon у поєднанні з власними правилами кореляції для виявлення дій, спрямованих на уникнення аудиту та приховування слідів компрометації.

3.3 Аналіз результатів та оцінка ефективності підсистеми моніторингу

Метою даного підрозділу є детальний аналіз реальних JSON-алертів, зафіксованих платформою Wazuh SIEM у процесі змодельованих кіберінцидентів. Усі алерти отримані з індексу wazuh-alerts-4.x-2026.05.25, агент Win10-Victim (ID: 001, IP: 192.168.100.101), атакуючий вузол – Kali Linux (IP: 192.168.100.50). Для кожного з дев'яти етапів атаки наведено таблицю з ключовими полями відповідного алерту та аналітичний коментар, що пояснює технічний зміст зафіксованих даних і підтверджує коректність виявлення загрози.

Виявлення мережної розвідки – Rule 100151

Перший рівень виявлення розвідки базується на перехопленні кожного окремого заблокованого TCP-пакету підсистемою Windows Filtering Platform. При SYN-скануванні Nmap кожен запит до закритого порту породжує подію Event ID 5157 з кодом AUDIT_FAILURE, яку Wazuh класифікує за правилом 100151 (рівень 3). Нижче наведено ключові поля відповідного алерту.

Таблиця 3.12

Алерт Rule 100151 виявлення мережної розвідки

Поле JSON-алерту	Значення
_index	wazuh-alerts-4.x-2026.05.25
timestamp	2026-05-25T12:13:30.804+0000
agent.name	Win10-Victim
agent.ip	192.168.100.101
rule.id	100151
rule.level	3
rule.description	WFP blocked connection - Event 5157
rule.groups	windows, attack
data.win.system.eventID	5157
data.win.system.severityValue	AUDIT_FAILURE
data.win.system.channel	Security
data.win.eventdata.sourceAddress	169.254.139.254
data.win.eventdata.destPort	1900
data.win.eventdata.protocol	17 (UDP)
data.win.eventdata.direction	Inbound
data.win.eventdata.application	\device\harddiskvolume2\windows\system32\svchost.exe
rule.firedtimes	18

Поле severityValue зі значенням AUDIT_FAILURE підтверджує факт відхилення вхідного мережного пакету брандмауером Windows. Поле sourceAddress ідентифікує IP-адресу джерела підозрілих запитів, а destPort – конкретний порт цільового вузла, до якого здійснювалась спроба підключення. Поле firedtimes = 18 свідчить про те, що дане правило спрацювало вісімнадцять разів упродовж сесії – кожного разу при новому заблокованому пакеті від сканера. Оскільки Nmap у режимі SYN-сканування (-sS -T4) генерує сотні пакетів за секунди, ця серія одиничних подій слугує вхідними даними для кореляційного правила другого рівня.

Виявлення мережної розвідки – Rule 100152 (кореляційний алерт сканування портів)

Правило 100152 є кореляційним: воно агрегує події від правила 100151 і генерує єдиний алерт рівня 10, коли від одного IP-джерела надходить 10 або більше заблокованих з'єднань протягом 120 секунд. Саме цей алерт є основним індикатором факту масового сканування мережі.

Таблиця 3.13

Алерт Rule 100152: кореляційне виявлення сканування портів

Поле JSON-алерту	Значення
timestamp	2026-05-25T08:49:45.267+0000
agent.name	Win10-Victim
rule.id	100152
rule.level	10
rule.description	Port Scan detected from 192.168.100.50
rule.frequency	10
rule.firedtimes	1
rule.mitre.id	T1046
rule.mitre.technique	Network Service Discovery
rule.mitre.tactic	Discovery
data.win.system.eventID	5157
data.win.system.severityValue	AUDIT_FAILURE
data.win.eventdata.sourceAddress	192.168.100.50
data.win.eventdata.destAddress	192.168.100.101
data.win.eventdata.protocol	6 (TCP)
data.win.eventdata.direction	Inbound
data.win.eventdata.application	\device\harddiskvolume2\windows\system32\svchost.exe

Поле rule.frequency = 10 підтверджує виконання умови кореляції: система накопичила рівно десять заблокованих з'єднань від одного джерела – IP 192.168.100.50 (вузол Kali Linux) – за визначений часовий проміжок 120 секунд. Прив'язка до техніки MITRE ATT&CK T1046 (Network Service Discovery) з тактикою Discovery автоматично класифікує подію в контексті матриці атак. Алерт містить у полі previous_output масив з дев'яти попередніх подій Event ID 5157 від того самого джерела, що дозволяє аналітику відновити повну хронологію сканування без додаткового ручного пошуку в журналах.

Виявлення RDP-брутфорсу – Rule 100101

На другому етапі атаки інструмент Crowbar генерував серію невдалих спроб RDP-автентифікації (Event ID 4625). Кореляційне правило 100101 агрегує їх і формує алерт рівня 10 після фіксації п'яти або більше невдалих спроб входу для одного облікового запису протягом 60 секунд, відповідно до техніки MITRE T1110.001 (Password Guessing).

Таблиця 3.14

Алерт Rule 100101: виявлення RDP-брутфорсу

Поле JSON-алерту	Значення
timestamp	2026-05-25T09:39:39.424+0000
Поле JSON-алерту	Значення
agent.name	Win10-Victim
rule.id	100101
rule.level	10
rule.description	Brute Force: Multiple RDP
rule.frequency	5
rule.firedtimes	4
rule.mitre.id	T1110.001
rule.mitre.technique	Password Guessing
rule.mitre.tactic	Credential Access
data.win.system.eventID	4624
data.win.system.channel	Security
data.win.eventdata.targetUserName	SYSTEM
data.win.eventdata.logonType	5
data.win.eventdata.authenticationPackageName	Negotiate
data.win.eventdata.logonProcessName	Advapi

Поле rule.frequency = 5 підтверджує, що умова кореляції (п'ять невдалих спроб входу за 60 секунд) виконана. Поле firedtimes = 4 вказує на те, що правило спрацювало чотири рази протягом тестової сесії – кожного разу формуючи окремий алерт при черговому накопиченні серії невдалих спроб. Аналіз поля previous_output в повному записі алерту дозволяє встановити точну IP-адресу джерела атаки та цільовий обліковий запис, на який вона спрямована. Повторна поява алерту чотири рази за відносно короткий час є характерною ознакою автоматизованого перебору паролів, а не поодиноких помилкових спроб входу легітимного користувача.

Виявлення ексфільтрації даних через FTP – Rule 100702

Після отримання доступу зловмисник використав FTP-сервіс Microsoft IIS для завантаження конфіденційних файлів. Підсистема Microsoft Sysmon зафіксувала мережне з'єднання на порт 21 через Event ID 3 (Network connection detected), і на основі цієї події правило 100702 сформувало алерт рівня 10 з прив'язкою до техніки T1048.003 (Exfiltration Over Unencrypted Non-C2 Protocol).

Таблиця 3.15

Алерт Rule 100702: виявлення FTP-з'єднання з ознаками ексфільтрації

Поле JSON-алерту	Значення
timestamp	2026-05-25T10:06:28.731+0000
agent.name	Win10-Victim
rule.id	100702
rule.level	10
rule.description	FTP connection detected - possible exfiltration to
rule.firedtimes	1
rule.mitre.id	T1048.003
rule.mitre.technique	Exfiltration Over Unencrypted Non-C2 Protocol
rule.mitre.tactic	Exfiltration
data.win.system.eventID	3 (Sysmon – Network connection)
data.win.system.channel	Microsoft-Windows-Sysmon/Operational
data.win.eventdata.sourceIp	192.168.100.50
data.win.eventdata.destinationIp	192.168.100.101
data.win.eventdata.destinationPort	21
data.win.eventdata.destinationPortName	ftp
data.win.eventdata.protocol	tcp
data.win.eventdata.image	C:\Windows\System32\svchost.exe

Продовження таблиці 3.15

Поле JSON-алерту	Значення
data.win.eventdata.user	NT AUTHORITY\SYSTEM
data.win.eventdata.utcTime	2026-05-25 10:06:24.987

Ключовим індикатором є поле destinationPort = 21 у поєднанні з destinationPortName = "ftp" – Sysmon автоматично резолвить назву сервісу зі стандартної таблиці портів, що унеможливлює помилкову ідентифікацію. Поля sourceIp та destinationIp чітко фіксують напрямок підключення: атакуючий вузол 192.168.100.50 ініціював TCP-з'єднання до FTP-сервісу на цільовому хості 192.168.100.101. Процес svchost.exe є легітимним системним процесом, через який FTP-сервіс IIS обробляє вхідні підключення – таким чином зловмисник використав легальний сервіс для передачі даних, що є типовою технікою Living-off-the-Land. Прив'язка до T1048.003 коректно відображає використання відкритого (нешифрованого) протоколу FTP для ексфільтрації.

Виявлення закріплення через реєстр автозапуску – Rule 92302

Для забезпечення постійного доступу після перезавантаження зловмисник додав запис backdoor.exe до ключа автозапуску реєстру Windows. Sysmon зафіксував цю дію через Event ID 13 (RegistryEvent – SetValue), і вбудоване правило Wazuh 92302 класифікувало її як модифікацію ключа автозапуску відповідно до техніки T1547.001.

Таблиця 3.16

Алерт Rule 92302: модифікація ключа реєстру автозапуску

Поле JSON-алерту	Значення
timestamp	2026-05-25T09:14:51.377+0000
agent.name	Win10-Victim
rule.id	92302
rule.level	6
rule.description	Registry entry to be executed on next logon was modified using
rule.firedtimes	2
rule.mitre.id	T1547.001
rule.mitre.technique	Registry Run Keys / Startup Folder
rule.mitre.tactic	Persistence, Privilege Escalation
data.win.system.eventID	13 (Sysmon – RegistryEvent SetValue)

Продовження таблиці 3.16

Поле JSON-алерту	Значення
data.win.system.channel	Microsoft-Windows-Sysmon/Operational
data.win.eventdata.targetObj	HKU\S-1-5-21-
data.win.eventdata.details	C:\backdoor.exe
data.win.eventdata.image	C:\Windows\system32\reg.exe
data.win.eventdata.eventTyp	SetValue
data.win.eventdata.ruleName	T1060,RunKey
data.win.eventdata.user	DESKTOP-JBJ43UQ\UserPriLipko
data.win.eventdata.utcTime	2026-05-25 09:14:53.881

Поле targetObject містить повний шлях до ключа реєстру HKU\...\Run\WindowsUpdate – ім'я WindowsUpdate використано зловмисником навмисно для маскування під легітимне оновлення Windows, що є типовою технікою обфускації. Поле details = "C:\backdoor.exe" розкриває шлях до шкідливого виконуваного файлу, що буде запускатися автоматично при кожному вході користувача. Поле ruleName = "T1060,RunKey" у Sysmon-конфігурації вказує на те, що ця активність попередньо визначена як підозрілий патерн у конфігурації Sysmon ще до надходження до Wazuh – це приклад ешелонованого захисту (defense in depth).

Виявлення закріплення через реєстр – Rule 92041 (запуск reg.exe)

Паралельно Sysmon зафіксував запуск самого процесу reg.exe через Event ID 1 (ProcessCreate), що дозволяє правилу 92041 задокументувати повний командний рядок операції модифікації реєстру.

Таблиця 3.17

Алерт Rule 92041: виявлення запуску reg.exe з параметрами модифікації Run-ключа

Поле JSON-алерту	Значення
timestamp	2026-05-25T09:14:51.365+0000
agent.name	Win10-Victim
rule.id	92041
rule.level	10
rule.description	Value added to registry key has Base64-like pattern
rule.mitre.id	T1027, T1112
rule.mitre.technique	Obfuscated Files or Information, Modify Registry

Продовження таблиці 3.17

Поле JSON-алерту	Значення
rule.mitre.tactic	Defense Evasion
data.win.system.eventID	1 (Sysmon – ProcessCreate)
data.win.eventdata.image	C:\Windows\System32\reg.exe
data.win.eventdata.commandLine	reg add
data.win.eventdata.parentImage	C:\Windows\System32\cmd.exe
data.win.eventdata.parentCommandLin	"C:\Windows\system32\cmd.exe"
data.win.eventdata.user	DESKTOP-JBJ43UQ\UserPriLipko
Поле JSON-алерту	Значення
data.win.eventdata.integrityLevel	Medium
data.win.eventdata.hashes	MD5=227F63E1D9008B36...,

Поле commandLine є найбільш інформативним у цьому алерті: воно містить повний аргументований рядок команди, виконаної зловмисником, – reg add ...Run... /v "WindowsUpdate" /d "C:\backdoor.exe". Цей рядок є незаперечним криміналістичним доказом навмисного додавання шкідливого файлу до автозапуску. Поле parentImage = cmd.exe підтверджує, що команда виконана через інтерактивну командну оболонку. Хеші файлу reg.exe (MD5, SHA256) дозволяють верифікувати цілісність системної утиліти та переконатись, що використовувався оригінальний системний файл, а не підмінений шкідливим ПЗ.

Виявлення запуску Mimikatz – Rule 100201

Критично важливим етапом атаки є витяг облікових даних за допомогою інструменту Mimikatz. Правило 100201 має найвищий рівень критичності в системі – Level 15, що автоматично активує механізм email-сповіщення (mail: true). Sysmon зафіксував запуск mimikatz.exe через Event ID 1 (ProcessCreate), і система виявила загрозу на основі сигнатурної назви виконуваного файлу.

Таблиця 3.18

Алерт Rule 100201: виявлення Credential Dumping інструменту Mimikatz

Поле JSON-алерту	Значення
timestamp	2026-05-25T10:31:21.300+0000
agent.name	Win10-Victim
rule.id	100201
rule.level	15 (CRITICAL)

Продовження таблиці 3.18

Поле JSON-алерту	Значення
rule.mail	true (автоматичне email-сповіщення)
rule.description	Credential dumping tool detected by name
rule.firedtimes	2
rule.mitre.id	T1003
rule.mitre.technique	OS Credential Dumping
rule.mitre.tactic	Credential Access
data.win.system.eventID	1 (Sysmon – ProcessCreate)
data.win.eventdata.image	C:\mimikatz\x64\mimikatz.exe
data.win.eventdata.originalFileNa	mimikatz.exe
data.win.eventdata.product	mimikatz
data.win.eventdata.company	gentilkiwi (Benjamin DELPY)
data.win.eventdata.commandLine	"C:\mimikatz\x64\mimikatz.exe"
data.win.eventdata.integrityLevel	High
data.win.eventdata.user	DESKTOP-JBJ43UQ\Administrator
data.win.eventdata.parentImage	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.
data.win.eventdata.hashes	MD5=29EFD64DD3C7FE1E...,
data.win.eventdata.fileVersion	2.2.0.0

Поле originalFileName = "mimikatz.exe" є ключовим індикатором: Sysmon зчитує цей атрибут безпосередньо з PE-заголовку виконуваного файлу, тому навіть перейменований екземпляр Mimikatz буде виявлений, якщо його метадані не були модифіковані. Поля product = "mimikatz" та company = "gentilkiwi (Benjamin DELPY)" надають додаткові ознаки для ідентифікації. Рівень цілісності High підтверджує запуск з адміністративними привілеями – обов'язкова умова для читання пам'яті процесу lsass.exe. Повний набір хешів (MD5, SHA256, IMPHASH) дозволяє верифікувати файл через зовнішні бази індикаторів компрометації – зокрема VirusTotal. Поле firedtimes = 2 вказує на два запуски Mimikatz упродовж тестової сесії.

Виявлення атаки Pass-the-Hash – Rule 100600

Здобувши NTLM-хеш із пам'яті lsass.exe, зловмисник реалізував техніку Pass-the-Hash через утиліту Evil-WinRM. Атака заснована на особливості протоколу

NTLM, де для автентифікації достатньо самого хешу пароля без знання оригінального пароля у відкритому вигляді. Правило 100600 виявило цей сценарій, аналізуючи поєднання ознак у події Event ID 4624.

Таблиця 3.19

Алерт Rule 100600: виявлення Pass-the-Hash через NTLM

Поле JSON-алерту	Значення
timestamp	2026-05-25T11:27:10.748+0000
agent.name	Win10-Victim
rule.id	100600
rule.level	12
rule.mail	true
rule.description	Possible Pass-the-Hash: NTLM network
rule.firedtimes	6
rule.mitre.id	T1550.002
rule.mitre.technique	Pass the Hash
rule.mitre.tactic	Defense Evasion, Lateral Movement
data.win.system.eventID	4624 (Successful logon)
data.win.system.channel	Security
data.win.system.severityValue	AUDIT_SUCCESS
data.win.eventdata.logonType	3 (Network)
data.win.eventdata.authenticationPackageName	NTLM
data.win.eventdata.lmPackageName	NTLM V2
data.win.eventdata.logonProcessName	NtLmSsp
data.win.eventdata.targetUserName	kykys
data.win.eventdata.targetDomainName	DESKTOP-JBJ43UQ
data.win.eventdata.ipAddress	192.168.100.50
data.win.eventdata.ipPort	58900
data.win.eventdata.keyLength	128
data.win.eventdata.subjectUserSid	S-1-0-0 (null SID – ознака PtH)

Сигнатура Pass-the-Hash у цьому алерті формується поєднанням трьох характерних ознак. По-перше, logonType = "3" означає мережну автентифікацію – підключення ззовні, а не інтерактивний вхід. По-друге, authenticationPackageName = "NTLM" у поєднанні з logonProcessName = "NtLmSsp" вказує на використання протоколу NTLM, який і є вектором атаки. По-третє, поле subjectUserSid = "S-1-0-0" –

це так званий null SID, відсутній ідентифікатор ініціюючого суб'єкта, що є характерною ознакою Pass-the-Hash на відміну від легітимної автентифікації, де це поле завжди містить SID ініціюючого облікового запису. Поле keyLength = 128 підтверджує успішне встановлення сесійного ключа NTLM V2, тобто атака завершилась успішно.

Виявлення Reverse Shell – Rule 100804

Для встановлення прихованого каналу керування зловмисник завантажив та запустив файл backdoor.exe, згенерований msfvenom з payload windows/x64/meterpreter/reverse_tcp. Sysmon зафіксував запуск підозрілого процесу через Event ID 1, і правило 100804 рівня 14 ідентифікувало активність як Reverse Shell або Metasploit payload.

Таблиця 3.20

Алерт Rule 100804: виявлення Reverse Shell / Metasploit payload

Поле JSON-алерту	Значення
timestamp	2026-05-25T10:31:09.705+0000
agent.name	Win10-Victim
rule.id	100804
rule.level	14
rule.mail	true
rule.description	Reverse shell or Metasploit payload detected
rule.firedtimes	1
rule.mitre.id	T1059
rule.mitre.technique	Command and Scripting Interpreter
rule.mitre.tactic	Execution
data.win.system.eventID	1 (Sysmon – ProcessCreate)
data.win.eventdata.image	C:\Windows\System32\WindowsPowerShell\v1.0\powershell
data.win.eventdata.originalFileNam	PowerShell.EXE
data.win.eventdata.commandLine	"C:\Windows\System32\WindowsPowerShell\v1.0\powershe
data.win.eventdata.parentImage	C:\Windows\explorer.exe
data.win.eventdata.parentCommand	C:\Windows\Explorer.EXE
data.win.eventdata.user	DESKTOP-JBJ43UQ\Administrator
data.win.eventdata.integrityLevel	High
data.win.eventdata.hashes	MD5=2E5A8590CF6848968F...,
data.win.eventdata.currentDirectory	C:\Users\Administrator\

Правило 100804 реагує на запуск PowerShell з підвищеними привілеями (integrityLevel = "High") від імені облікового запису Administrator у контексті підозрілої батьківської процесної ієрархії. Поєднання батьківського процесу Explorer.EXE з інтерактивним запуском PowerShell та попередніми подіями компрометації (Pass-the-Hash, завантаження backdoor.exe через Evil-WinRM) дозволяє системі кореляції класифікувати цю подію як пов'язану з Post-Exploitation активністю. Хеші виконуваного файлу можуть бути використані для верифікації через бази IoC. Прив'язка до T1059 (Command and Scripting Interpreter) відображає реальну тактику зловмисника – використання легітимного інтерпретатора команд для виконання шкідливого навантаження.

Виявлення створення бекдор-облікового запису – Rule 100704

Через активну Meterpreter-сесію зловмисник виконав команду net localgroup Administrators hacker_boss /add, створивши прихований адміністративний обліковий запис для забезпечення постійного доступу. Sysmon зафіксував запуск net1.exe (дочірній процес net.exe) через Event ID 1, і правило 100704 класифікувало цю дію як Account Manipulation за технікою MITRE T1098.

Таблиця 3.21

Алерт Rule 100704: виявлення додавання користувача до групи Administrators

Поле JSON-алерту	Значення
timestamp	2026-05-25T12:03:24.458+0000
agent.name	Win10-Victim
rule.id	100704
rule.level	13
rule.mail	true
rule.description	User added to Administrators group via command
rule.firedtimes	2
rule.mitre.id	T1098
rule.mitre.technique	Account Manipulation
rule.mitre.tactic	Persistence
data.win.system.eventID	1 (Sysmon – ProcessCreate)
data.win.eventdata.image	C:\Windows\System32\net1.exe
data.win.eventdata.originalFileName	net1.exe
data.win.eventdata.commandLine	C:\Windows\system32\net1 localgroup

Продовження таблиці 3.21

Поле JSON-алерту	Значення
data.win.eventdata.parentImage	C:\Windows\System32\net.exe
data.win.eventdata.parentCommandLine	net localgroup Administrators hacker_boss /add
data.win.eventdata.user	DESKTOP-JBJ43UQ\kykys
data.win.eventdata.integrityLevel	High
data.win.eventdata.hashes	MD5=78E53D5AE8839C58...,
data.win.eventdata.currentDirectory	C:\Users\kykys\Documents\

Поле commandLine є ключовим криміналістичним доказом у цьому алерті: рядок "net1 localgroup Administrators hacker_boss /add" однозначно встановлює, що обліковий запис hacker_boss навмисно доданий до привілейованої групи Administrators. Поле parentCommandLine = "net localgroup Administrators hacker_boss /add" підтверджує, що net1.exe є дочірнім процесом стандартної утиліти net.exe – така структура є характерною для виклику мережних команд Windows. Поле currentDirectory = "C:\Users\kykys\Documents\" вказує на те, що команда виконана з робочої директорії скомпрометованого облікового запису kykys через активну сесію. Рівень цілісності High підтверджує наявність адміністративних прав у момент виконання команди.

Виявлення очищення журналів подій – Rule 100705

Заключним етапом атаки стала спроба зловмисника знищити докази своєї присутності шляхом очищення системних журналів Windows командою wevtutil cl System. Незважаючи на виконання цієї команди, система моніторингу Wazuh зафіксувала сам факт її запуску через Sysmon ще до очищення локального журналу, оскільки події передаються на централізований сервер у режимі реального часу.

Таблиця 3.22

Алерт Rule 100705: виявлення очищення журналів подій Windows

Поле JSON-алерту	Значення
timestamp	2026-05-25T12:09:40.990+0000
agent.name	Win10-Victim
rule.id	100705
rule.level	14
rule.mail	true
rule.description	Defense Evasion: Windows Event Logs cleared via

Продовження таблиці 3.22

Поле JSON-алерту	Значення
rule.firedtimes	2
rule.mitre.id	T1070.001
rule.mitre.technique	Clear Windows Event Logs
rule.mitre.tactic	Defense Evasion
data.win.system.eventID	1 (Sysmon – ProcessCreate)
data.win.eventdata.image	C:\Windows\System32\wevtutil.exe
data.win.eventdata.originalFileName	wevtutil.exe
data.win.eventdata.commandLine	wevtutil cl System
data.win.eventdata.parentImage	C:\Windows\System32\cmd.exe
data.win.eventdata.parentCommandLine	C:\Windows\system32\cmd.exe
data.win.eventdata.user	DESKTOP-JBJ43UQ\kykys
data.win.eventdata.integrityLevel	High
data.win.eventdata.hashes	MD5=D4C99E01B0061D2E...,
data.win.eventdata.currentDirectory	C:\Users\kykys\Documents\
data.win.eventdata.fileVersion	10.0.19041.5794

Даний алерт демонструє принципову перевагу централізованих SIEM-систем над локальним журналюванням: оскільки всі події агент Wazuh передає на сервер у режимі реального часу, їх неможливо ретроактивно видалити навіть у разі повного знищення локальних журналів. Поле commandLine = "wevtutil cl System" є незаперечним доказом виконання команди очищення. Поле originalFileName = "wevtutil.exe" підтверджує ідентифікацію утиліти за PE-метаданими, що унеможливлює хибне спрацювання. Прив'язка до техніки T1070.001 (Clear Windows Event Logs) з тактикою Defense Evasion автоматично класифікує дії зловмисника як спробу протидії виявленню. Поле firedtimes = 2 свідчить про дворазове виконання команди очищення в ході тестової сесії – для різних журналів (System та Security).

Висновки до третього розділу

У третьому розділі проведено практичне тестування системи моніторингу безпеки на базі платформи Wazuh SIEM шляхом моделювання повного ланцюжка кібератаки на розгорнутому лабораторному стенді. Стенд відтворював типову корпоративну інфраструктуру та включав мережне обладнання Cisco, сервер Wazuh на

Ubuntu 22.04, вузол-жертву Windows 10 з налаштованим агентом та підсистемою Sysmon, а також станцію атакуючого на базі Kali Linux.

Змодельований сценарій охопив дев'ять послідовних етапів атаки відповідно до методології MITRE ATT&CK: мережну розвідку (T1046), брутфорс RDP (T1110.001), ексфільтрацію даних через FTP (T1048.003), закріплення у реєстрі автозапуску (T1547.001), витяг облікових даних Mimikatz (T1003), атаку Pass-the-Hash (T1550.002), розгортання Reverse Shell (T1059), створення бекдор-облікового запису (T1098) та очищення журналів подій (T1070.001).

За результатами аналізу зафіксованих алертів підтверджено, що система виявила всі дев'ять етапів атаки без жодного пропуску – показник виявлення склав 100%. Рівні критичності алертів варіювались від Level 3 для одиничних мережних подій до Level 15 для запуску інструменту Credential Dumping, що підтверджує коректність налаштованої градації пріоритетів. Для алертів найвищих рівнів активований механізм автоматичного email-сповіщення адміністратора безпеки.

Детальний аналіз JSON-алертів у підрозділі 3.3 підтвердив інформативність зафіксованих даних: кожен алерт містить достатній набір полів для криміналістичного відновлення хронології інциденту, ідентифікації джерела атаки та класифікації за матрицею MITRE ATT&CK без необхідності додаткового ручного пошуку в журналах.

Окремо підтверджено принципову перевагу централізованої SIEM-архітектури: незважаючи на виконання команди очищення локальних журналів на дев'ятому етапі атаки, всі раніше зафіксовані алерти залишились доступними в індексі Wazuh, оскільки події передаються на сервер у режимі реального часу.

Отримані результати підтверджують практичну ефективність розробленої системи та її придатність для виявлення реальних кіберзагроз у корпоративному середовищі.

ВИСНОВКИ

У кваліфікаційній роботі бакалавра успішно вирішено актуальне науково-практичне завдання щодо проєктування та практичної реалізації системи моніторингу та управління подіями інформаційної безпеки на базі відкритої SIEM-платформи Wazuh із використанням мережного обладнання Cisco. Головним результатом роботи відповідно до поставленої мети стало створення комплексної системи централізованого збору, кореляції та аналізу подій безпеки в реальному часі, практична ефективність якої підтверджена шляхом моделювання реальних кіберінцидентів у лабораторному середовищі.

Під час виконання роботи реалізовано повний перелік закладених функцій, які демонструють високий ступінь відповідності вихідному технічному завданню. Зокрема, виконано такі завдання.

Проведено ґрунтовний аналіз сучасного стану систем моніторингу інформаційної безпеки. Встановлено, що традиційні засоби захисту – міжмережні екрани, антивірусні рішення та системи виявлення вторгнень – не забезпечують цілісного контролю над станом інфраструктури без централізованої кореляції подій. Досліджено провідні комерційні та відкриті SIEM-платформи: IBM QRadar, Splunk Enterprise Security, ArcSight та Elastic Security. За результатами порівняльного аналізу за критеріями функціональності, гнучкості налаштування, масштабованості та вартості впровадження для реалізації обрано платформу Wazuh як оптимальне рішення для навчальних і корпоративних середовищ середнього масштабу. Визначено функціональні та технічні вимоги до системи, досліджено основні моделі виявлення кіберінцидентів – сигнатурний, поведінковий та кореляційний аналіз – з прив'язкою до міжнародної матриці тактик і технік зловмисників MITRE ATT&CK.

Спроєктовано архітектуру системи та виконано її повне розгортання у тестовому середовищі на базі віртуалізації VirtualBox. Центральним елементом виступає Wazuh Server на Ubuntu Server 22.04 LTS, що включає Wazuh Manager для обробки та кореляції подій, Wazuh Indexer на базі OpenSearch для зберігання та індексації даних, а також Wazuh Dashboard для візуалізації та аналізу інцидентів. На вузлі-

жертві Windows 10 розгорнуто Wazuh Agent із конфігурацією збору подій із журналу безпеки Windows (Security Event Log) через канал eventchannel та розширеного моніторингу через підсистему Sysmon (Microsoft-Windows-Sysmon/Operational). Інтеграцію мережного обладнання Cisco реалізовано через передачу системних журналів за протоколом Syslog на IP-адресу сервера Wazuh.

Розроблено комплекс більш ніж двадцяти кастомних правил кореляції у конфігураційному файлі local_rules.xml, написаних мовою XML. Правила охоплюють повний ланцюжок атаки відповідно до концепції Cyber Kill Chain: виявлення мережної розвідки (дворівнева логіка на основі WFP Event ID 5157), брутфорс облікових даних (кореляція Event ID 4625 із частотою 5 подій за 60 секунд), ексфільтрацію даних (Sysmon Event ID 3, порт 21), закріплення у реєстрі автозапуску (Sysmon Event ID 13 та Event ID 1), виявлення інструментів Credential Dumping за сигнатурними іменами процесів, атаки Pass-the-Hash (аналіз NTLM logonType=3 з null SID), Reverse Shell (аналіз батьківської процесної ієрархії), маніпуляцію обліковими записами та знищення слідів присутності. При розробці правил застосовано ієрархічний підхід із використанням механізмів підрахунку частоти подій (frequency-based rules), регулярних виразів стандарту PCRE2 та кореляції за спільними полями (same_field), що дозволяє мінімізувати кількість хибнопозитивних спрацювань.

Проведено практичне тестування системи на розгорнутому лабораторному стенді, який відтворює типову корпоративну мережну інфраструктуру. Стенд включає маршрутизатор Cisco та комутатор Cisco як об'єкти мережного моніторингу, сервер Wazuh на Ubuntu 22.04, вузол-жертву Windows 10 з агентом Wazuh та підсистемою Sysmon, а також станцію атакуючого Kali Linux. Усі вузли об'єднані в єдину локальну мережу з адресним простором 192.168.100.0/24 за топологією «зірка» з комутатором Cisco у центрі.

Змодельований сценарій атаки охопив дев'ять послідовних етапів відповідно до методології MITRE ATT&CK: мережну розвідку інструментом Nmap (T1046), брутфорс RDP інструментом Crowbar (T1110.001), ексфільтрацію конфіденційних файлів через FTP-сервіс Microsoft IIS (T1048.003), закріплення у ключі реєстру ав-

тозапуску HKCU\Software\Microsoft\Windows\CurrentVersion\Run (T1547.001), витяг облікових даних та NTLM-хешів інструментом Mimikatz (T1003), горизонтальне переміщення мережею через атаку Pass-the-Hash з використанням Evil-WinRM (T1550.002), розгортання Reverse Shell на основі Metasploit Framework з payload meterpreter/reverse_tcp (T1059), створення прихованого адміністративного облікового запису (T1098) та знищення слідів присутності командою wevtutil cl (T1070.001).

Детальний аналіз зафіксованих JSON-алертів у модулі Threat Hunting платформи Wazuh підтвердив повну ефективність розробленої системи: загальний показник виявлення склав 100% – система своєчасно виявила всі дев'ять змодельованих етапів атаки без жодного пропуску. Рівні критичності алертів варіювались від Level 3 для одиничних мережних подій до Level 15 для запуску інструменту Credential Dumping. Для алертів найвищих рівнів критичності активовано механізм автоматичного email-сповіщення адміністратора безпеки (mail: true), що суттєво скорочує середній час виявлення та реагування на інциденти.

Практична цінність отриманих результатів полягає у підтвердженні принципової переваги централізованої SIEM-архітектури над локальним журналюванням: спроба зловмисника знищити докази своєї діяльності шляхом повного очищення системних журналів Windows командою wevtutil cl не вплинула на збереженість жодного алерту у централізованому індексі Wazuh, оскільки всі події передаються та зберігаються на сервері в режимі реального часу. Розроблена система є готовим до розгортання рішенням, придатним як для навчальних лабораторних середовищ, так і для корпоративних інфраструктур малого та середнього масштабу.

Напрямками подальшого наукового дослідження та вдосконалення розробленого комплексу є: інтеграція з платформами threat intelligence для автоматичного збагачення алертів актуальними індикаторами компрометації (IoC); розширення набору правил виявлення для хмарних середовищ та контейнерних інфраструктур; дослідження можливостей впровадження модуля активного реагування Wazuh Active Response для автоматизованої нейтралізації виявлених загроз шляхом динамічного блокування атакуючих IP-адрес на рівні мережного обладнання Cisco; а

також розвиток системи у напрямку концепції SOAR для повної автоматизації циклу виявлення та реагування на інциденти інформаційної безпеки.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. ENISA Threat Landscape 2023 [Електронний ресурс] // European Union Agency for Cybersecurity. – Режим доступу: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023 (дата звернення: 10.01.2026).

2. ENISA Threat Landscape 2022 [Електронний ресурс] // European Union Agency for Cybersecurity. – Режим доступу: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2022 (дата звернення: 10.01.2026).

3. Wazuh documentation: Getting started with Wazuh [Електронний ресурс] // Wazuh Inc. – Режим доступу: https://documentation.wazuh.com/current/getting-started/index.html (дата звернення: 15.01.2026).

4. Wazuh documentation: Wazuh agent [Електронний ресурс] // Wazuh Inc. – Режим доступу: https://documentation.wazuh.com/current/getting-started/components/wazuh-agent.html (дата звернення: 15.01.2026).

5. Wazuh documentation: Wazuh indexer [Електронний ресурс] // Wazuh Inc. – Режим доступу: https://documentation.wazuh.com/current/getting-started/components/wazuh-indexer.html (дата звернення: 16.01.2026).

6. MITRE ATT&CK: Enterprise Matrix [Електронний ресурс] // The MITRE Corporation. – Режим доступу: https://attack.mitre.org/matrices/enterprise (дата звернення: 20.01.2026).

7. Sysinternals Sysmon documentation [Електронний ресурс] // Microsoft Corporation. – Режим доступу: https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon (дата звернення: 22.01.2026).

8. IBM QRadar SIEM overview [Електронний ресурс] // IBM Corporation. – Режим доступу: https://www.ibm.com/products/qradar-siem (дата звернення: 25.01.2026).

9. Splunk Enterprise Security overview [Електронний ресурс] // Splunk Inc. – Режим доступу: https://www.splunk.com/en_us/products/enterprise-security.html (дата звернення: 25.01.2026).

10. Cisco IOS system message logging configuration guide [Електронний ресурс] // Cisco Systems. – Режим доступу: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/config_mgmt/configuration/xe-16/config-mgmt-xe-16-book/cm-sys-message-logging.html (дата звернення: 02.02.2026).

11. Wazuh documentation: Custom rules and decoders [Електронний ресурс] // Wazuh Inc. – Режим доступу: https://documentation.wazuh.com/current/user-manual/ruleset/custom.html (дата звернення: 05.02.2026).

12. Wazuh documentation: Wazuh server administration [Електронний ресурс] // Wazuh Inc. – Режим доступу: https://documentation.wazuh.com/current/user-manual/manager/index.html (дата звернення: 06.02.2026).

13. Windows security event log: Event ID 4625 [Електронний ресурс] // Microsoft Corporation. – Режим доступу: https://learn.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4625 (дата звернення: 10.02.2026).

14. Pass the hash attack explained [Електронний ресурс] // CrowdStrike Inc. – Режим доступу: https://www.crowdstrike.com/cybersecurity-101/pass-the-hash (дата звернення: 15.02.2026).

15. Wazuh documentation: Wazuh dashboard [Електронний ресурс] // Wazuh Inc. – Режим доступу: https://documentation.wazuh.com/current/getting-started/components/wazuh-dashboard.html (дата звернення: 18.02.2026).

16. ArcSight Enterprise Security Manager [Електронний ресурс] // OpenText CyberSecurity. – Режим доступу: https://www.opentext.com/products/arcsight-enterprise-security-manager (дата звернення: 25.01.2026).

17. Elastic Security: SIEM solution [Електронний ресурс] // Elastic N.V. – Режим доступу: https://www.elastic.co/guide/en/security/current/es-overview.html (дата звернення: 26.01.2026).

18. MITRE ATT&CK: Technique T1003 – OS Credential Dumping [Електронний ресурс] // The MITRE Corporation. – Режим доступу: https://attack.mitre.org/techniques/T1003 (дата звернення: 20.02.2026).

19. MITRE ATT&CK: Technique T1070.001 – Clear Windows Event Logs [Електронний ресурс] // The MITRE Corporation. – Режим доступу: https://attack.mitre.org/techniques/T1070/001 (дата звернення: 20.02.2026).

20. Nmap: Network scanning reference guide [Електронний ресурс] // Insecure.Org. – Режим доступу: https://nmap.org/book/man.html (дата звернення: 01.03.2026).

ДОДАТКИ

Додаток А

Конфігурація маршрутизатора з системою Cisco IOS

```
Building configuration...
Current configuration : 1319 bytes
!
! Last configuration change at 11:25:09 UTC Mon May 25 2026
!
version 15.1
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R-PriLipko-SOC
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
!
!
no ipv6 cef
ip source-route
ip cef
!
!
!
!
!
ip domain name SOC-Diplom
!
multilink bundle-name authenticated
!
!
!
!
!
crypto pki token default removal timeout 0
!
!
voice-card 0
!
!
!
!
!
!
!
license udi pid CISCO2911/K9 sn FCZ1513206Y
hw-module pvdm 0/0
!
```

```
!
!
username admin privilege 15 secret 5 $1$4G9o$11dGnNZWNX6uxiDtF/JIw/
!
redundancy
!
!
!
!
ip ssh version 1
!
!
!
!
!
!
!
interface GigabitEthernet0/0
 ip address 192.168.100.1 255.255.255.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface GigabitEthernet0/2
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial0/0/0
 no ip address
 shutdown
 clock rate 125000
!
interface Serial0/0/1
 no ip address
 shutdown
 clock rate 125000
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
!
logging 192.168.100.100
!
```

```
!
!
!
!
!
control-plane
!
!
!
!
!
!
!
!
gatekeeper
 shutdown
!
!
line con 0
line aux 0
line vty 0 4
 login local
 transport input ssh
!
scheduler allocate 20000 1000
end
```

Додаток Б

Конфігурація комутатора

```
Building configuration...

Current configuration : 2965 bytes
!
version 15.0
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Switch-Prilipko
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
system mtu routing 1500
!
!
!
!
crypto pki trustpoint TP-self-signed-2484267008
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-2484267008
 revocation-check none
 rsakeypair TP-self-signed-2484267008
!
!
crypto pki certificate chain TP-self-signed-2484267008
 certificate self-signed 01
  3082022B 30820194 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 32343834 32363730 3038301E 170D3933 30333031 30303031
  30315A17 0D323030 31303130 30303030 305A3031 312F302D 06035504 03132649
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32 34383432
  36373030 3830819F 300D0609 2A864886 F70D0101 01050003 818D0030 81890281
  8100E29F DFD9A34B 1F78465E 23AD82D2 C636A48C 696A79D6 3C5D9B8F 0CD14ABE
  6C8AE511 0E1F044D CF8DC72F EF1AE5F1 0C6D1047 9BD51A72 69DB25BB 240EC715
  6E955722 6FCF893F 21DFF94F 5805DFFE FED6DD79 72588563 409FEDB6 FB2F82BF
  A02BEC5E FE5F90AE A5937965 792B2A1F 695C1792 A5521476 CB192472 B8898EF0
  475B0203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF 301F0603
  551D2304 18301680 147A9C90 C5E76DBE 88C79C35 25DC7FE7 CED450F0 7C301D06
  03551D0E 04160414 7A9C90C5 E76DBE88 C79C3525 DC7FE7CE D450F07C 300D0609
  2A864886 F70D0101 05050003 81810073 990C5781 C280A0B6 0DF964C6 AA4419AA
  206A8BD8 7D4E8C84 D3FC8309 99418D89 6323D17C FA34643A 630C9F0C 37353CA5
  3D4D1DE5 813406C6 E7CAE7F8 EEECA6B3 538D38D5 E84AFB6A D7A17B40
D8BCF727
  D97AB84D 77CAF842 89C2D916 C554DAFE A1689E71 33D5B09D 5DF7686C 85B84678
  9DD9A586 4C363170 A5870160 D8806B
```

```
        quit
!
!
!
!
!
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
!
!
!
!
!
!
!
!
!
!
!
!
!
interface FastEthernet0/1
!
interface FastEthernet0/2
!
interface FastEthernet0/3
!
interface FastEthernet0/4
!
interface FastEthernet0/5
!
interface FastEthernet0/6
!
interface FastEthernet0/7
!
interface FastEthernet0/8
!
interface FastEthernet0/9
!
interface FastEthernet0/10
!
interface FastEthernet0/11
!
interface FastEthernet0/12
!
interface FastEthernet0/13
!
interface FastEthernet0/14
!
```

```
interface FastEthernet0/15
!
interface FastEthernet0/16
!
interface FastEthernet0/17
!
interface FastEthernet0/18
!
interface FastEthernet0/19
!
interface FastEthernet0/20
!
interface FastEthernet0/21
!
interface FastEthernet0/22
!
interface FastEthernet0/23
!
interface FastEthernet0/24
!
interface GigabitEthernet0/1
!
interface GigabitEthernet0/2
!
interface Vlan1
 ip address 192.168.100.2 255.255.255.0
!
ip default-gateway 192.168.100.1
ip http server
ip http secure-server
!
!
logging host 192.168.100.100
!
vstack
!
line con 0
line vty 0 4
 login
line vty 5 15
 login
!
end
```

Додаток В

Конфігурація файлу local_rules.xml

```
<!-- Local rules -->
<!-- Modify it at your will. -->
<!-- Copyright (C) 2015, Wazuh Inc. -->

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



<group name="local,syslog,cisco,">
  <rule id="100010" level="10">
    <decoded_as>cisco-ios</decoded_as>
    <match>CONFIG_I</match>
    <description>Cisco configuration modification detected</description>
    <group>configuration_change,network,</group>
  </rule>
</group>



<group name="windows,attack,">


  <!-- ═══════════════════════════════════════════════════════
       BRUTE FORCE
  ═══════════════════════════════════════════════════════ -->

  <!-- Батько: 60122 = Windows failed logon (built-in Wazuh) -->
  <rule id="100100" level="12" frequency="5" timeframe="60">
    <if_matched_sid>60122</if_matched_sid>
    <description>Brute Force: Multiple failed Windows logons</description>
    <mitre><id>T1110</id></mitre>
  </rule>

  <!-- Батько: 60106 = Windows logon event (built-in Wazuh) -->
  <rule id="100101" level="10" frequency="5" timeframe="60">
    <if_matched_sid>60106</if_matched_sid>
    <description>Brute Force: Multiple RDP authentication failures</description>
    <mitre><id>T1110.001</id></mitre>
  </rule>

  <!-- Батько: 60106 = Windows logon event -->
  <rule id="100800" level="14">
    <if_sid>60106</if_sid>
```

```
  <field name="win.system.eventID">^4740$</field>
  <description>Account locked out - likely brute force result</description>
  <mitre><id>T1110</id></mitre>
</rule>


<!-- ═══════════════════════════════════════════════════════
     RDP УСПІШНИЙ ВХІД
═══════════════════════════════════════════════════════ -->

<!-- Батько: 60106 = Windows logon event, logonType 10 = RemoteInteractive (RDP) -->
<rule id="100102" level="8">
  <if_sid>60106</if_sid>
  <field name="win.system.eventID">^4624$</field>
  <field name="win.eventdata.logonType">^10$</field>
  <description>Successful RDP login from $(win.eventdata.ipAddress)</description>
  <mitre><id>T1078</id></mitre>
</rule>


<!-- ═══════════════════════════════════════════════════════
     NMAP / PORT SCAN
═══════════════════════════════════════════════════════ -->

<!-- Батько: 60104 = Windows audit failure (built-in Wazuh) -->
<rule id="100151" level="3">
  <if_sid>60104</if_sid>
  <field name="win.system.eventID">^5157$</field>
  <description>WFP blocked connection - Event 5157</description>
</rule>

<!-- Батько: 100151 (вище в цьому ж файлі) -->
<rule id="100152" level="10" frequency="10" timeframe="120">
  <if_matched_sid>100151</if_matched_sid>
  <same_field>win.eventdata.sourceAddress</same_field>
  <description>Port Scan detected from $(win.eventdata.sourceAddress)</description>
  <mitre><id>T1046</id></mitre>
</rule>


<!-- ═══════════════════════════════════════════════════════
     SMB РОЗВІДКА / ENUMERATION
═══════════════════════════════════════════════════════ -->

<!-- Батько: 60139 = Windows SMB event (built-in Wazuh) -->
<rule id="100400" level="8" frequency="10" timeframe="30">
  <if_matched_sid>60139</if_matched_sid>
  <description>SMB Share enumeration detected</description>
  <mitre><id>T1135</id></mitre>
</rule>

<!-- Батько: 60139 = Windows SMB event -->
```

```
  <rule id="100900" level="9">
    <if_sid>60139</if_sid>
    <field name="win.eventdata.subjectUserName" type="pcre2">(?i)(anonymous|guest|\$)</field>
    <description>Anonymous SMB enumeration attempt</description>
    <mitre><id>T1135</id></mitre>
  </rule>


  <!-- ═══════════════════════════════════════════════════════
       FTP – ЕКСФІЛЬТРАЦІЯ
  ═══════════════════════════════════════════════════════ -->

  <!-- Батько: 61612 = Sysmon network connection (built-in Wazuh) -->
  <rule id="100700" level="8">
    <if_sid>61612</if_sid>
    <field name="win.eventdata.destinationPort">^21$</field>
    <description>FTP connection initiated from $(win.eventdata.sourceIp) - possible
exfiltration</description>
    <mitre><id>T1048.003</id></mitre>
  </rule>

  <!-- Батько: 61603 = Sysmon process create (built-in Wazuh) -->
  <rule id="100701" level="10">
    <if_sid>61603</if_sid>
    <field                                                    name="win.eventdata.commandLine"
type="pcre2">(?i)(ftp|tftp|sftp|scp|curl|wget|certutil.*urlcache|bitsadmin.*transfer)</field>
    <description>Suspicious file transfer tool executed - possible exfiltration</description>
    <mitre><id>T1048</id></mitre>
  </rule>

 <!-- FTP мережне з'єднання Sysmon Event 3 -->
 <rule id="100702" level="10">
  <if_sid>61605</if_sid>
  <field name="win.eventdata.destinationPort">^21$</field>
  <description>FTP connection detected - possible exfiltration to
$(win.eventdata.destinationIp)</description>
  <mitre><id>T1048.003</id></mitre>
</rule>

  <!-- ═══════════════════════════════════════════════════════
       PRIVILEGE ESCALATION / НОВИЙ ЮЗЕР
  ═══════════════════════════════════════════════════════ -->

  <!-- Батько: 60000 = Windows logon/account event -->

<rule id="100703" level="12">
    <if_sid>61603</if_sid>
    <field name="win.eventdata.commandLine" type="pcre2">(?i)net1?\s+user.*\s+/add</field>
    <description>New user created via command line (net user /add)</description>
    <mitre><id>T1136.001</id></mitre>
  </rule>
```

```
  <rule id="100704" level="13">
    <if_sid>61603</if_sid>
    <field name="win.eventdata.commandLine" type="pcre2">(?i)net1?\s+localgroup\s+(administrators|admins|администраторы).*/add</field>
    <description>User added to Administrators group via command line (Privilege Escalation)</description>
    <mitre><id>T1098</id></mitre>
  </rule>


<rule id="100705" level="14">
    <if_sid>61603</if_sid>
    <field name="win.eventdata.commandLine" type="pcre2">(?i)wevtutil.*\s+cl\s+</field>
    <description>Defense Evasion: Windows Event Logs cleared via wevtutil</description>
    <mitre><id>T1070.001</id></mitre>
  </rule>

  <!-- ═══════════════════════════════════════════════════════
       PERSISTENCE – ЗАКРІПЛЕННЯ
  ═══════════════════════════════════════════════════════ -->

  <!-- Батько: 61603 = Sysmon process create -->
  <rule id="100500" level="10">
    <if_sid>61603</if_sid>
    <field name="win.eventdata.commandLine" type="pcre2">(?i)schtasks.*\/create</field>
    <description>Suspicious scheduled task created - possible persistence</description>
    <mitre><id>T1053.005</id></mitre>
  </rule>

  <!-- Батько: 61603 = Sysmon process create -->
  <rule id="100501" level="10">
    <if_sid>61603</if_sid>
    <field name="win.eventdata.commandLine" type="pcre2">(?i)(reg\s+add.*\\run|reg\s+add.*\\runonce)</field>
    <description>Registry Run key modified - persistence via autostart</description>
    <mitre><id>T1547.001</id></mitre>
  </rule>

  <!-- Батько: 61603 = Sysmon process create -->
  <rule id="100502" level="11">
    <if_sid>61603</if_sid>
    <field name="win.eventdata.commandLine" type="pcre2">(?i)(sc\s+create|sc\s+config|New-Service)</field>
    <description>New Windows service created - possible persistence</description>
    <mitre><id>T1543.003</id></mitre>
  </rule>


  <!-- ═══════════════════════════════════════════════════════
       POWERSHELL ПІДОЗРІЛИЙ
  ═══════════════════════════════════════════════════════ -->
```

```
  <!-- Батько: 61603 = Sysmon process create -->
  <rule id="100300" level="12">
    <if_sid>61603</if_sid>
    <field name="win.eventdata.commandLine" type="pcre2">(?i)(-enc|-
encodedcommand|downloadstring|iex\s)</field>
    <description>Suspicious PowerShell execution - encoded or download command</description>
    <mitre><id>T1059.001</id></mitre>
  </rule>

  <!-- Батько: 61603 = Sysmon process create -->
  <rule id="100301" level="13">
    <if_sid>61603</if_sid>
    <field name="win.eventdata.commandLine" type="pcre2">(?i)(invoke-mimikatz|invoke-
expression|invoke-shellcode|powersploit)</field>
    <description>PowerShell attack framework detected</description>
    <mitre><id>T1059.001</id></mitre>
  </rule>


  <!-- ═══════════════════════════════════════════════════════
       MIMIKATZ / CREDENTIAL DUMP
  ═══════════════════════════════════════════════════════ -->

  <!-- Батько: 61616 = Sysmon process access (built-in Wazuh) -->
  <rule id="100200" level="15">
    <if_sid>61616</if_sid>
    <field name="win.eventdata.targetImage" type="pcre2">(?i)lsass\.exe</field>
    <description>Credential Dumping: Access to lsass.exe detected (Mimikatz)</description>
    <mitre><id>T1003.001</id></mitre>
  </rule>

  <!-- Батько: 61603 = Sysmon process create -->
  <rule id="100201" level="15">
    <if_sid>61603</if_sid>
    <field name="win.eventdata.commandLine"
type="pcre2">(?i)(mimikatz|sekurlsa|lsadump|wce\.exe|fgdump)</field>
    <description>Credential dumping tool detected by name</description>
    <mitre><id>T1003</id></mitre>
  </rule>


  <!-- ═══════════════════════════════════════════════════════
       REVERSE SHELL / ШКІДЛИВЕ ПО
  ═══════════════════════════════════════════════════════ -->

  <!-- Батько: 61603 = Sysmon process create -->
  <rule id="100804" level="14">
    <if_sid>61603</if_sid>
    <field name="win.eventdata.commandLine"
type="pcre2">(?i)(msfvenom|meterpreter|payload|reverse_tcp|shell\.exe|cmd\.exe.*-e|powershell.*-
nop.*-w.*hidden)</field>
    <description>Reverse shell or Metasploit payload detected</description>
```

```
  <mitre><id>T1059</id></mitre>
 </rule>

 <!-- Батько: 61612 = Sysmon network connection -->
 <rule id="100802" level="12">
  <if_sid>61612</if_sid>
  <field name="win.eventdata.destinationPort" type="pcre2">^(4444|4445|5555|8080|8443|1337|31337)$</field>
  <description>Suspicious outbound connection on common C2 port $(win.eventdata.destinationPort)</description>
  <mitre><id>T1071</id></mitre>
 </rule>

 <!-- Батько: 61603 = Sysmon process create -->
 <rule id="100803" level="13">
  <if_sid>61603</if_sid>
  <field name="win.eventdata.image" type="pcre2">(?i)(cmd\.exe|powershell\.exe|wscript\.exe|cscript\.exe)</field>
  <field name="win.eventdata.parentImage" type="pcre2">(?i)(explorer\.exe|winword\.exe|excel\.exe|outlook\.exe)</field>
  <description>Shell spawned from Office or Explorer - possible malware</description>
  <mitre><id>T1204</id></mitre>
 </rule>


 <!-- ═══════════════════════════════════════════════════════
      PASS-THE-HASH / NTLM
 ═══════════════════════════════════════════════════════ -->

 <!-- Батько: 60122 = Windows failed logon -->
 <rule id="101550" level="12">
  <if_sid>60122</if_sid>
  <field name="win.eventdata.logonProcessName" type="pcre2">(?i)ntlmssp</field>
  <description>NTLM authentication attempt - possible poisoning</description>
  <mitre><id>T1557.001</id></mitre>
 </rule>

 <!-- Батько: 60106 = Windows logon event -->
 <rule id="100600" level="12">
  <if_sid>92652</if_sid>
  <field name="win.eventdata.logonProcessName" type="pcre2">(?i)ntlmssp</field>
  <field name="win.eventdata.logonType">^3$</field>
  <description>Possible Pass-the-Hash: NTLM network logon detected</description>
  <mitre><id>T1550.002</id></mitre>
 </rule>

 <!-- ═══════════════════════════════════════════════════════
      WEB SCAN / DOS
 ═══════════════════════════════════════════════════════ -->

 <!-- Батько: 31101 = Web server 400 errors (built-in Wazuh) -->
 <rule id="101450" level="10">
```

```
    <if_sid>31101</if_sid>
    <description>Web vulnerability scan detected</description>
    <mitre><id>T1190</id></mitre>
  </rule>

  <!-- Батько: 40101 = High volume syslog (built-in Wazuh) -->
  <rule id="101350" level="13" frequency="50" timeframe="5">
    <if_matched_sid>40101</if_matched_sid>
    <description>DoS attack: High volume of packets detected</description>
    <mitre><id>T1498</id></mitre>
  </rule>

</group>
```