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 комплексного безпекового моніторингу гетерогенної мережі на основі Wazuh та Snort»

Виконав: студент 4-го курсу групи КБ-22-2
спеціальності 125 «Кіберберзпека»
_____ Владислав ЯРОШ
Керівник: професор кафедри комп'ютерної інженерії та кібербезпеки, доктор технічних наук, доцент
_____ Володимир ВОРОТНІКОВ
Рецензент: завідувач кафедри комп'ютерної інженерії та кібербезпеки, кандидат технічних наук, 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: Ярош Владислав Сергійович
Тема роботи: Проєкт підсистеми комплексного безпекового моніторингу гетерогенної мережі на основі Wazuh та Snort
затверджена Наказом університету від 2 лютого 2026 р. № 58/с
Термін здачі студентом закінченої роботи 9 червня 2026 р.

Вихідні дані роботи (зазначається предмет і об'єкт дослідження): Об'єктом дослідження є процес забезпечення моніторингу та виявлення кіберзагроз у гетерогенній комп'ютерній мережі шляхом використання систем Wazuh та Snort.

Предметом дослідження є технології проєктування та реалізації підсистеми комплексного безпекового моніторингу на основі IDS/IPS та SIEM-рішень.

Обладнання та програмне забезпечення:

1. Гіпервізор Oracle VM VirtualBox 7.2.6
2. ОС Ubuntu Server 24.04.4 LTS, Windows 10 Education 22H2 та Ubuntu Desktop 24.04.4 LTS
3. SIEM-система Wazuh 4.14
4. IDS/IPS-система Snort 3
5. Міжмережевий екран pfSense v. 2.8.1
6. Мови програмування Python та Bash
7. Програмний емулятор терміналу PuTTY v. 0.83

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

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Старший викладач кафедри КІ та КБ Ігор ФАЛЬКОВСЬКИЙ	20.04.2026	20.04.2026

Студент _____ Владислав ЯРОШ
Керівник _____ Володимир ВОРОТНІКОВ

Календарний план

№ з/п	Назви етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка завдання	11.02.2026	Виконано
2	Аналіз отриманого завдання	23.02.2026	Виконано
3	Аналіз літературних джерел	01.03.2026	Виконано
4	Оформлення 1 розділу пояснювальної записки	15.03.2026	Виконано
5	Проєктування архітектури моделі	25.03.2026	Виконано
6	Вибір технологій та інструментів для реалізації	29.03.2026	Виконано
7	Оформлення 2 розділу пояснювальної записки	01.04.2026	Виконано
8	Підготовка необхідних компонентів для мережі	04.04.2026	Виконано
9	Розгортання середовища та мережевої інфраструктури	07.04.2026	Виконано
10	Проведення базового налаштування мережі	15.04.2026	Виконано
11	Налагодження засобів моніторингу та механізмів реагування	27.04.2026	Виконано
12	Оформлення 3 розділу пояснювальної записки	19.05.2026	Виконано
13	Підготовка доповіді та презентації	01.06.2026	Виконано
14	Захист		Виконано

Студент _____ Владислав ЯРОШ

Керівник _____ Володимир ВОРОТНІКОВ

АНОТАЦІЯ

Мета кваліфікаційної роботи бакалавра полягає у дослідженні підходів до побудови підсистеми комплексного безпекового моніторингу гетерогенної мережі та аналізі сучасних програмно-апаратних засобів виявлення й реагування на кіберзагрози. Завданням роботи є застосування отриманих знань для проєктування та реалізації системи моніторингу подій безпеки на основі відкритих рішень Wazuh та Snort, яка може використовуватись для підвищення рівня захисту мережевої інфраструктури.

Кваліфікаційна робота бакалавра на тему «Проєкт підсистеми комплексного безпекового моніторингу гетерогенної мережі на основі Wazuh та Snort» складається з теоретичного, аналітичного, проєктного та практичного комплексів і пояснювальної записки.

Визначено основні завдання побудови системи моніторингу подій безпеки та централізованого аналізу мережевої активності. Проаналізовано сучасні SIEM та IDS/IPS-рішення, розглянуто особливості функціонування систем Wazuh і Snort, а також досліджено можливості їх інтеграції у єдину систему безпекового моніторингу. Показано процес проєктування мережевої інфраструктури, налаштування компонентів системи, реалізації механізмів виявлення атак та активного реагування на підозрілу активність. Наведено результати тестування системи із застосуванням мережевих атак та реагування на них.

Пояснювальна записка до випускної кваліфікаційної роботи містить 97 сторінок, 6 таблиць та 158 рисунків. Список використаних джерел містить 38 найменувань і займає 3 сторінки.

Ключові слова: WAZUH, SNORT, SIEM, IDS, IPS, МОНІТОРИНГ, КІБЕРБЕЗПЕКА, АТАКА, ЗАХИСТ, РЕАГУВАННЯ, ІНТЕГРАЦІЯ, АЛЕРТ, СПОВІЩЕННЯ, КОРЕЛЯЦІЯ.

					ККІКБ.КР.Б – 125 – 17 – ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Ярош В. С.		05.06.26	Проєкт підсистеми комплексного безпекового моніторингу гетерогенної мережі на основі Wazuh та Snort	Літ.	Арк. 4 / Аркушів 97
Керівник		Воротніков В.В.		05.06.26			
Реценз.		Єфіменко А.А.		05.06.26		ФІКТ, Гр. КБ-22-2	
Н. контр.		Воротніков В.В.		05.06.26			
Затверд.		Єфіменко А.А.		05.06.26			

ANNOTATION

The purpose of the bachelor's qualification work is to study approaches to building a subsystem of complex security monitoring of a heterogeneous network and analyze modern software and hardware tools for detecting and responding to cyber threats. The task of the work is to apply the knowledge gained to design and implement a security event monitoring system based on open solutions Wazuh and Snort, which can be used to increase the level of network infrastructure protection.

The bachelor's qualification work on the topic "Project of a subsystem of complex security monitoring of a heterogeneous network based on Wazuh and Snort" consists of theoretical, analytical, design and practical complexes and an explanatory note.

The main tasks of building a system of security event monitoring and centralized analysis of network activity are determined. Modern SIEM and IDS/IPS solutions are analyzed, the features of the functioning of the Wazuh and Snort systems are considered, and the possibilities of their integration into a single security monitoring system are investigated. The process of designing network infrastructure, configuring system components, implementing attack detection mechanisms and actively responding to suspicious activity is shown. The results of testing the system using network attacks and responding to them are presented.

The explanatory note to the final qualification work contains 97 pages, 6 tables and 158 figures. The list of used sources contains 38 names and takes up 3 pages.

Keywords: WAZUH, SNORT, SIEM, IDS, IPS, MONITORING, CYBER-SECURITY, ATTACK, PROTECTION, RESPONSE, INTEGRATION, ALERT, NOTIFICATION, CORRELATION.

ЗМІСТ

ВСТУП 8
РОЗДІЛ 1 АНАЛІЗ СУЧАСНИХ ЗАСОБІВ БЕЗПЕКОВОГО МОНІТОРИНГУ ТА ВИЯВЛЕННЯ КІБЕРЗАГРОЗ 10
1.1 Принципи функціонування систем моніторингу подій безпеки 10
1.2 Аналіз сучасних SIEM-систем та їх функціональних можливостей 13
1.3 Обґрунтування вибору SIEM-системи Wazuh 19
1.4 Аналіз сучасних IDS/IPS-рішень 23
1.5 Обґрунтування вибору IDS/IPS-системи Snort 27
1.6 Постановка технічого завдання 30
Висновки до розділу 31
РОЗДІЛ 2 ПРОЄКТУВАННЯ ПІДСИСТЕМИ КОМПЛЕКСНОГО БЕЗПЕКОВОГО МОНІТОРИНГУ 32
2.1 Проєктування гетерогенної мережевої інфраструктури 32
2.2 Розробка схеми адресації та мережевої взаємодії компонентів 34
2.3 Проєктування архітектури підсистеми моніторингу безпеки 36
2.4 Проєктування механізмів виявлення та реагування на атаки 38
2.5 Аналіз сценаріїв мережевих атак для тестування системи 41
Висновки до розділу 44
РОЗДІЛ 3 РОЗГОРТАННЯ, НАЛАШТУВАННЯ ТА ТЕСТУВАННЯ ПІДСИСТЕМИ БЕЗПЕКОВОГО МОНІТОРИНГУ 45
3.1 Розгортання віртуального середовища та мережевої інфраструктури 45
3.2 Налаштування SIEM-системи Wazuh 56
3.3 Налаштування IDS/IPS-системи Snort 60
3.4 Інтеграція Snort із Wazuh 64
3.5 Тестування виявлення та блокування мережевих атак 78
3.6 Аналіз результатів функціонування системи 91
Висновки до розділу 93
ВИСНОВКИ 94
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 95
ДОДАТКИ 98

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ОС – операційна система

ПЗ – програмне забезпечення

ІБ – інформаційна безпека

SIEM – Security Information and Event Management (інформація про безпеку та управління подіями)

Open-source – відкритий код

IDS – Intrusion Detection System (система виявлення вторгнень)

IPS – Intrusion Prevention System (система запобігання вторгненням)

MITRE ATT&CK – Adversarial Tactics, Techniques, and Common Knowledge (змагальні тактики, методи та загальновідомі знання)

CVE – Common Vulnerabilities and Exposures (загальновідомі вразливості та ризики)

BSD – Berkeley Software Distribution (дистрибутив програм Берклі)

IP – Internet protocol (інтернет протокол)

SOC – Security Operations Center (центр операцій безпеки)

ICMP – Internet Control Message Protocol (протокол міжмережевих керуючих повідомлень)

SSH – Secure Shell (протокол безпечного віддаленого керування доступом)

SYN – Synchronize (синхронізація)

DNS – Domain Name System (система доменних імен)

TCP – Transmission Control Protocol (протокол керування передачею)

FTP – File Transport Protocol (протокол передачі файлів)

ВМ – Віртуальна машина

LAN – Local Area Network (локальна мережа)

WAN – Wide Area Network (глобальна мережа)

NAT – Network Address Translation (перетворення мережевих адрес)

DAQ – Data Acquisition (збір даних)

ШІ – Штучний інтелект

ВСТУП

Актуальність теми. У сучасних умовах стрімкого розвитку інформаційних технологій та збільшення кількості кіберзагроз питання забезпечення інформаційної та мережевої безпеки набуває особливої актуальності. Комп'ютерні мережі підприємств, організацій та установ постійно піддаються різноманітним мережевим атакам. У зв'язку з цим виникає потреба у використанні сучасних засобів моніторингу, аналізу та оперативного реагування на події безпеки.

Особливої **актуальності** набуває питання побудови комплексних підсистем безпекового моніторингу для гетерогенних мереж, у яких використовуються різні операційні системи, мережеві сервіси та програмно-апаратні компоненти. Реалізація таких підсистем дозволяє забезпечити ефективне виявлення мережевих атак, централізований контроль подій безпеки та підвищення загального рівня захищеності мережевої інфраструктури. Одним із ефективних підходів до підвищення рівня захисту мережевої інфраструктури є використання SIEM-систем та IDS/IPS, які забезпечують централізований збір журналів подій, аналіз мережевої активності, виявлення потенційних загроз та автоматичне реагування на підозрілу активність. У сучасних мережах ця тема набуває особливого значення, адже використання таких систем дозволяє не тільки підвищити ефективність контролю за станом інформаційної безпеки, а й зменшити час реагування на інциденти та забезпечити централізований аналіз подій у мережевому середовищі.

Метою кваліфікаційної роботи є проєктування та реалізація підсистеми комплексного безпекового моніторингу гетерогенної мережі на основі SIEM та IDS/IPS-систем для виявлення, аналізу та реагування на мережеві атаки.

Визначена мета обумовлює наступні завдання:

- Проаналізувати сучасні підходи до організації безпекового моніторингу комп'ютерних мереж.
- Дослідити принципи функціонування SIEM та IDS/IPS-систем.
- Виконати аналіз сучасних систем моніторингу та виявлення кіберзагроз.
- Реалізувати модель гетерогенної мережі у віртуальному середовищі.
- Виконати налаштування засобів моніторингу та виявлення атак.

- Об’єднати усі компоненти в єдину підсистему моніторингу.
- Реалізувати механізми централізованого аналізу подій та активного реагування.
- Провести тестування виявлення та блокування мережевих атак.
- Проаналізувати результати функціонування розробленої системи.

Об’єктом дослідження є процес забезпечення моніторингу та виявлення кіберзагроз у гетерогенній комп’ютерній мережі.

Предметом дослідження є технології, методи та засоби реалізації підсистеми комплексного безпекового моніторингу на основі SIEM та IDS/IPS-рішень.

У процесі виконання кваліфікаційної роботи було розроблено проєкт підсистеми комплексного безпекового моніторингу гетерогенної мережі. У межах проєкту реалізовано централізований збір та аналіз подій безпеки, механізми виявлення мережевих атак, інтеграцію засобів моніторингу та автоматизоване реагування на підозрілу активність.

РОЗДІЛ 1 АНАЛІЗ СУЧАСНИХ ЗАСОБІВ БЕЗПЕКОВОГО МОНІТОРИНГУ ТА ВИЯВЛЕННЯ КІБЕРЗАГРОЗ

1.1 Принципи функціонування систем моніторингу подій безпеки

У сучасних комп'ютерних мережах постійно виконується велика кількість процесів, пов'язаних із передачею даних, взаємодією користувачів із серверами, роботою мережевих служб та функціонуванням прикладного програмного забезпечення. Під час роботи мережевої інфраструктури формуються журнали подій, у яких міститься інформація про мережеві підключення, запуск служб, помилки авторизації, мережеві запити та інші дії користувачів і системних компонентів. Аналіз таких подій дозволяє своєчасно виявляти підозрілу активність та потенційні порушення інформаційної безпеки [1].

Для автоматизації контролю за станом мережевої інфраструктури використовуються системи моніторингу подій безпеки. Основним призначенням таких систем є централізований збір, обробка та аналіз інформації, яка надходить із різних джерел мережі. До таких джерел можуть належати сервери, робочі станції, міжмережеві екрани, маршрутизатори, системи виявлення атак, мережеві служби та журнали операційних систем. Після отримання журналів подій система моніторингу виконує їх аналіз та перевірку на наявність ознак підозрілої активності. У більшості випадків події надходять у різних форматах, тому одним із перших етапів роботи є нормалізація даних. Під час цього процесу інформація приводиться до єдиного формату, що дозволяє спростити подальший аналіз і кореляцію подій.

Важливою особливістю сучасних систем моніторингу є можливість кореляції подій. Кореляція дозволяє об'єднувати інформацію з різних джерел та встановлювати взаємозв'язки між окремими подіями. Наприклад, велика кількість невдалих спроб авторизації разом із підозрілим мережевим трафіком може свідчити про спробу brute force-атаки. У такому випадку система формує повідомлення про можливий інцидент ІБ. Загальний принцип функціонування системи моніторингу подій безпеки наведено на рис. 1.1.

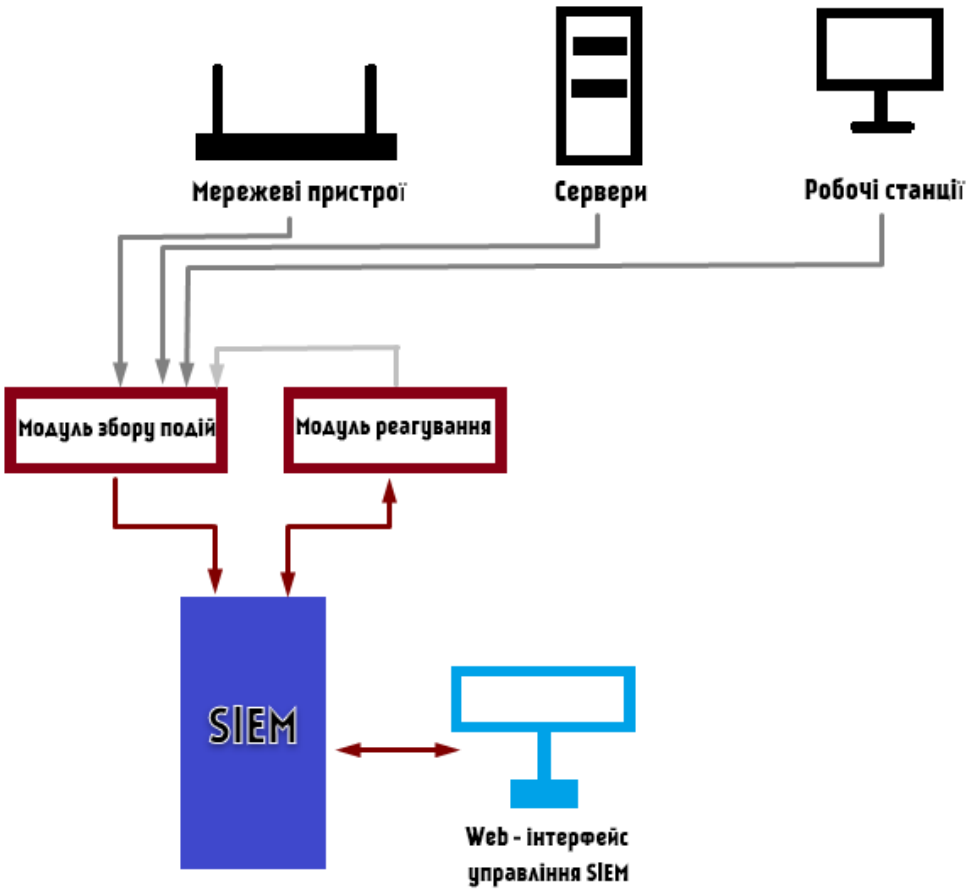


Рисунок 1.1 – Загальний принцип функціонування системи моніторингу подій безпеки

Системи моніторингу можуть використовувати різні підходи до виявлення загроз. Найбільш поширеними є сигнатурний та поведінковий методи аналізу. Сигнатурний метод базується на порівнянні отриманих подій із базою відомих шаблонів атак. Такий підхід дозволяє ефективно виявляти вже відомі типи загроз, однак має обмеження у випадку нових або модифікованих атак. Поведінковий аналіз ґрунтується на виявленні аномальної активності у роботі системи або мережі. У цьому випадку система визначає відхилення від нормальної поведінки користувачів чи мережевих служб. Такий метод дозволяє виявляти невідомі типи загроз, однак може створювати хибні спрацювання. Основні функції систем моніторингу подій безпеки наведено у табл. 1.1.

Таблиця 1.1

Основні функції систем моніторингу подій безпеки

Функція	Призначення
Збір подій	Отримання журналів подій та мережевої інформації
Аналіз активності	Виявлення підозрілих дій та аномалій
Кореляція подій	Встановлення взаємозв’язків між подіями
Сповіщення	Формування повідомлень про інциденти
Реагування	Виконання дій у відповідь на загрозу
Візуалізація	Відображення подій у вигляді дашбордів та графіків

Окрему роль у роботі систем моніторингу відіграють механізми реагування на інциденти. У разі виявлення небезпечної активності система може не лише повідомляти адміністратора безпеки, а й автоматично виконувати певні дії, наприклад блокування IP-адреси, завершення мережевого з’єднання або обмеження доступу до окремих ресурсів системи [2].

Крім цього, сучасні системи моніторингу дозволяють виконувати централізовану візуалізацію подій безпеки за допомогою dashboard та графічних панелей. Це значно спрощує аналіз мережевої активності та дозволяє швидше реагувати на потенційні інциденти інформаційної безпеки.

Таким чином, системи моніторингу подій безпеки є важливим компонентом сучасних засобів захисту інформації. Їх використання дозволяє забезпечити централізований контроль за станом мережевої інфраструктури, підвищити ефективність виявлення загроз та скоротити час реагування на інциденти інформаційної безпеки.

1.2 Аналіз сучасних SIEM-систем та їх функціональних можливостей

Зі збільшенням кількості мережевих пристроїв, серверів та користувачів значно зростає й обсяг подій безпеки, які необхідно контролювати. У сучасних мережах щосекунди формуються журнали авторизації, мережевих підключень, системних помилок, запуску служб та інших процесів. Ручний аналіз такої кількості інформації є практично неможливим, тому для централізованого контролю подій безпеки використовуються SIEM-системи [3].

SIEM-системи – це програмні комплекси, призначені для збору, обробки, аналізу та візуалізації подій інформаційної безпеки. Основним заданням SIEM-систем є централізація журналів подій та автоматизація процесу виявлення потенційних загроз у мережевому середовищі. Сучасні SIEM-системи дозволяють отримувати інформацію з великої кількості джерел:

- Серверів.
- Робочих станцій.
- Мережевого обладнання.
- Міжмережевих екранів.
- Систем виявлення атак.
- Баз даних.
- Хмарних сервісів.
- Операційних систем.

Після отримання інформації система виконує її нормалізацію, аналіз та кореляцію подій. Це дозволяє виявляти підозрілу активність, аналізувати інциденти безпеки та формувати централізовані повідомлення про загрози [4].

Однією з головних переваг SIEM-систем є можливість централізованого моніторингу всієї мережевої інфраструктури. Адміністратор безпеки отримує доступ до єдиного середовища аналізу подій, що значно спрощує контроль за станом інформаційної безпеки організації. Крім збору журналів подій, сучасні SIEM-рішення підтримують механізми автоматизації реагування на інциденти. У разі виявлення підозрілої активності система може автоматично створювати повідомлення, запускати сценарії реагування або передавати інформацію іншим засобам захисту

інформації. Загальна структура функціонування SIEM продемонстрована на рис. 1.2.

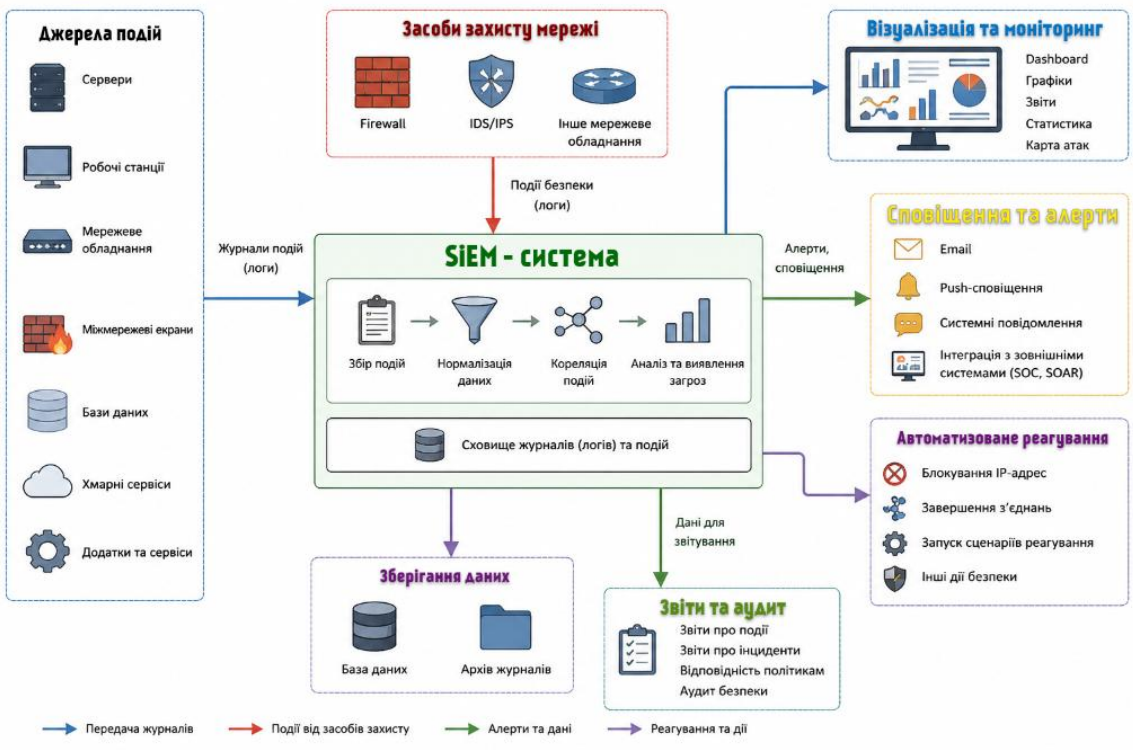


Рисунок 1.2 – Загальна структура функціонування SIEM-системи

На сьогодні існує велика кількість SIEM-рішень, які відрізняються функціональними можливостями, складністю впровадження, вимогами до ресурсів та вартістю використання. Найбільш відомими серед них є IBM QRadar, Splunk Enterprise Security, Microsoft Sentinel, Elastic Security та Wazuh.

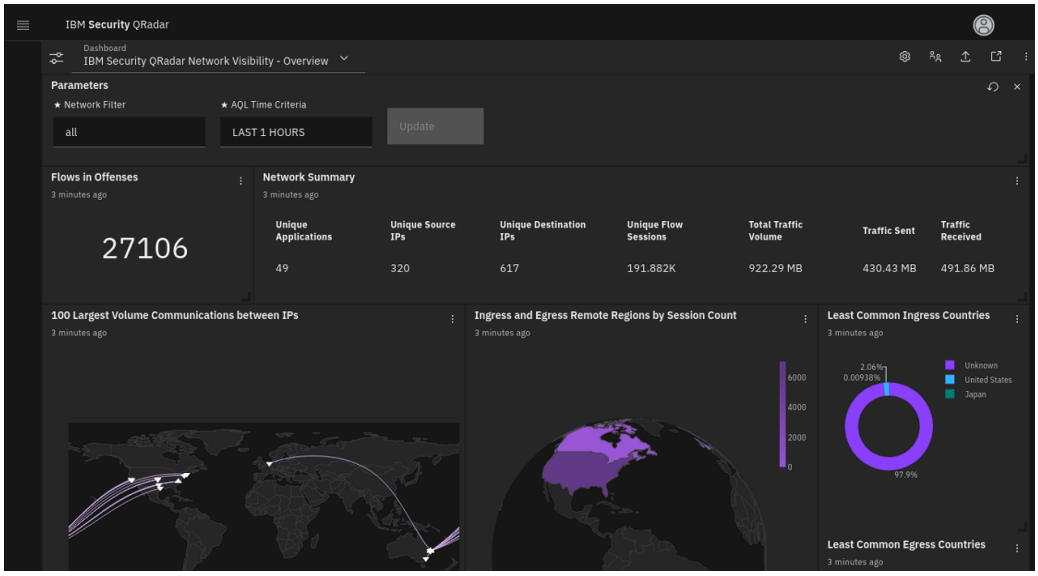


Рисунок 1.3 – SIEM IBM QRadar

IBM QRadar є корпоративною SIEM-системою, яка широко використовується у великих організаціях, а сама система забезпечує централізований аналіз журналів подій, підтримує кореляцію подій безпеки та дозволяє виявляти складні сценарії атак [5]. Перевагою є розвинуті механізми аналітики та підтримка великої кількості мережевих джерел, водночас, система потребує значних апаратних ресурсів та має високу вартість впровадження.

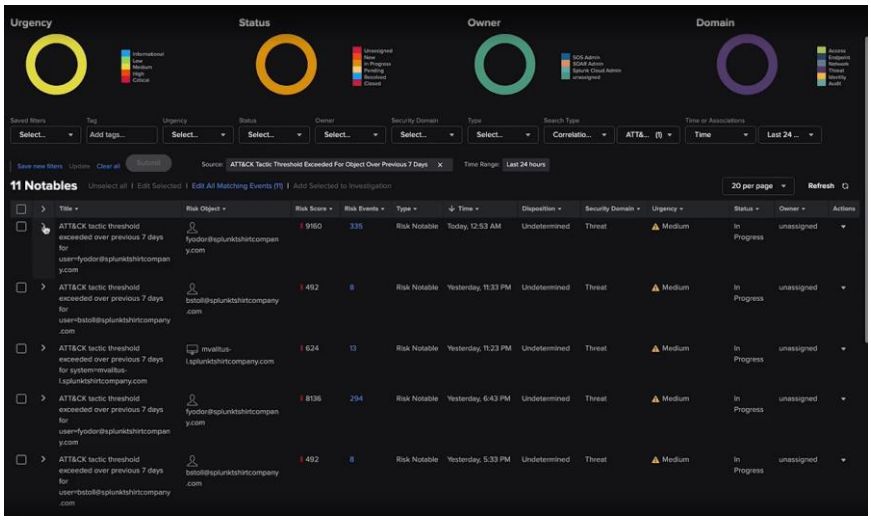


Рисунок 1.4 – Splunk Enterprise Security

Splunk Enterprise Security також належить до популярних SIEM-рішень. Основною перевагою Splunk є гнучкість налаштування та потужні засоби пошуку й аналізу журналів подій. Система підтримує створення dashboard, автоматизацію аналізу подій та інтеграцію з іншими засобами захисту інформації. Основним недоліком є складність адміністрування та висока вартість комерційної ліцензії [6].

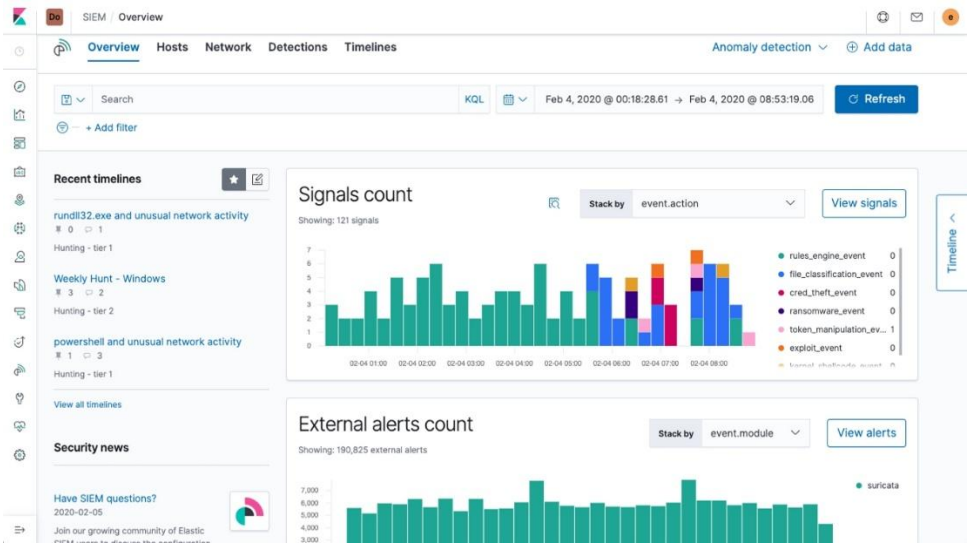


Рисунок 1.5 – Elastic Security

Elastic Security побудована на базі Elastic Stack та забезпечує можливість роботи з великими обсягами даних. Система підтримує збір журналів подій, пошук, фільтрацію та візуалізацію інформації. Перевагою Elastic Security є висока масштабованість та підтримка інтеграції з великою кількістю джерел даних [7].

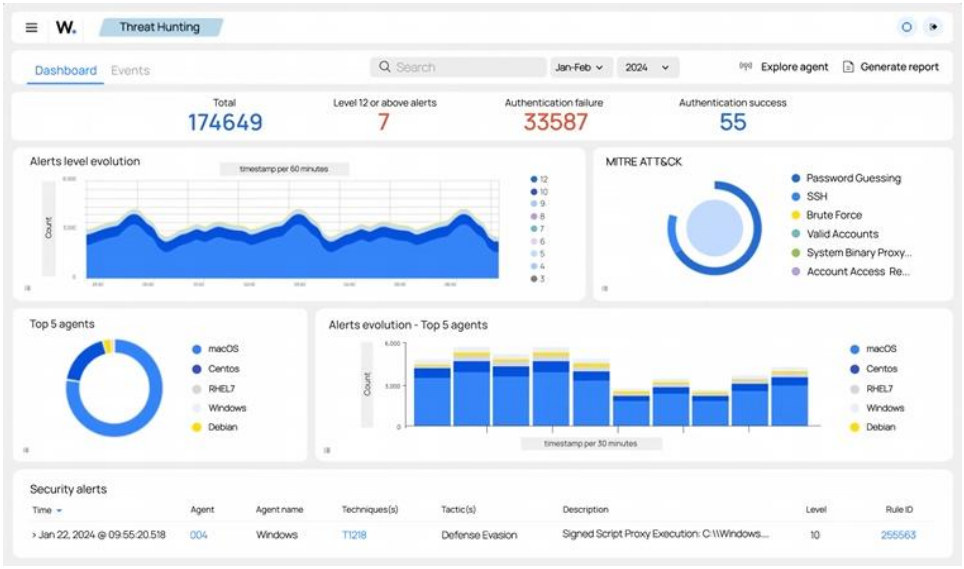


Рисунок 1.6 – Wazuh

Wazuh є однією з найбільш популярних відкритих SIEM-систем, яка активно використовується для моніторингу подій безпеки та аналізу мережевої активності. Система підтримує збір журналів подій із серверів, робочих станцій, мережевого обладнання та інших джерел даних. Крім цього, Wazuh забезпечує контроль цілісності файлів, виявлення вразливостей, моніторинг стану агентів та підтримує інтеграцію із системами виявлення атак [8].

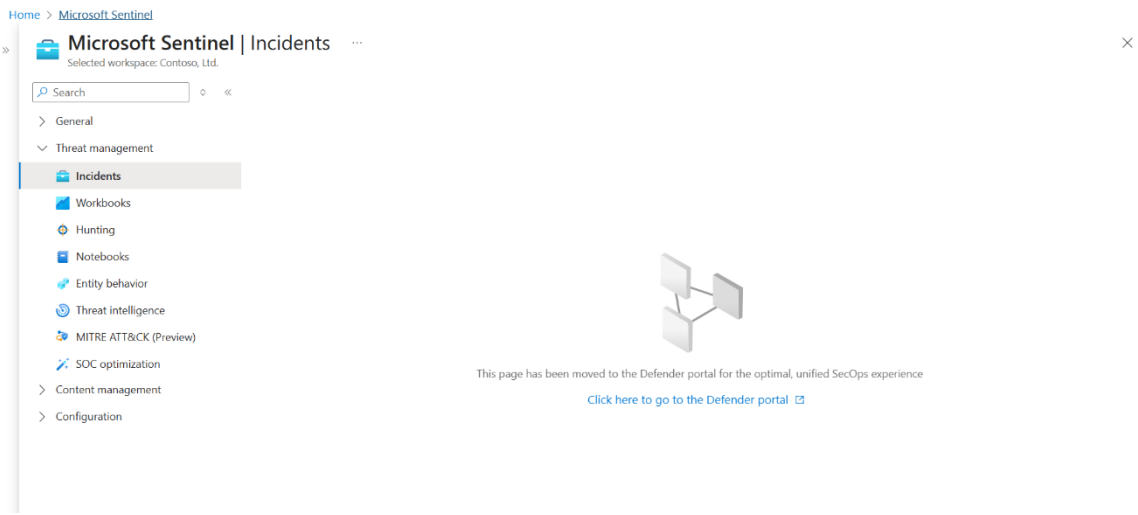


Рисунок 1.7 – Microsoft Sentinel

Microsoft Sentinel є хмарною SIEM-системою, яка забезпечує централізований моніторинг подій безпеки у хмарному середовищі. Система входить як компонент Microsoft Azure та підтримує автоматизацію реагування на інциденти, інтеграцію із сервісами Microsoft та використання механізмів аналітики для виявлення кіберзагроз. Перевагою є масштабованість та підтримка хмарної інфраструктури, однак використання системи залежить від сервісів Microsoft Azure та плану підписки [9].

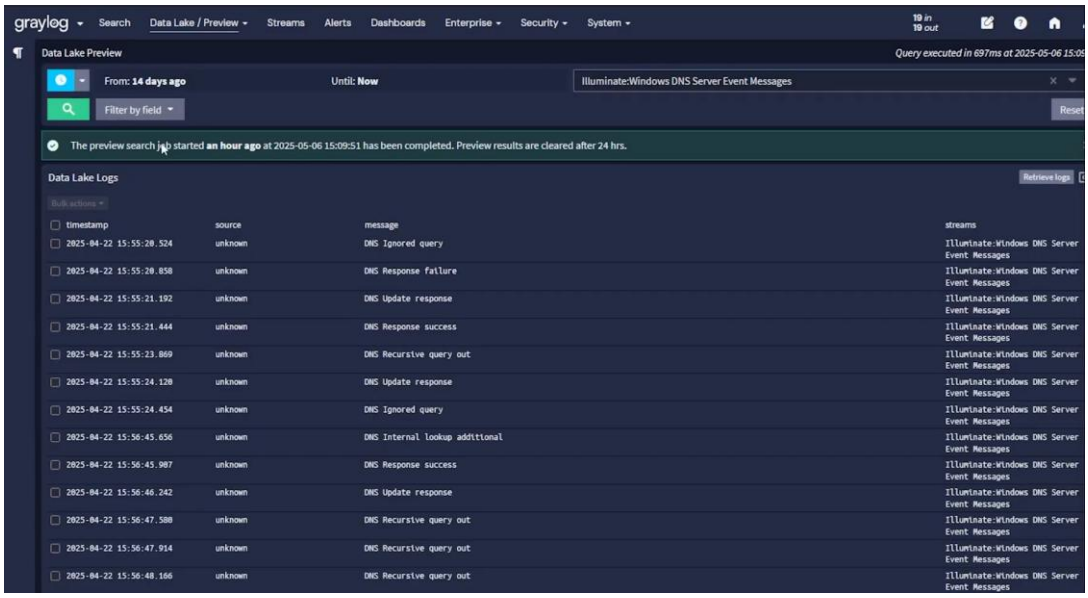


Рисунок 1.8 – Graylog Security

Graylog є системою централізованого збору та аналізу журналів подій, яка використовується для моніторингу мережевої інфраструктури та аналізу подій безпеки. Система підтримує пошук, фільтрацію та візуалізацію інформації, а також інтеграцію з різними джерелами даних. Однією з переваг є відносно просте розгортання та підтримка роботи з великими обсягами журналів подій [10].

Загалом, окремої уваги заслуговують відкриті SIEM-системи, які набувають все більшої популярності завдяки можливості безкоштовного використання та гнучкому налаштуванню. Такі рішення дозволяють реалізувати функції моніторингу без значних фінансових витрат та можуть використовуватись у невеликих організаціях, навчальних лабораторіях або тестових середовищах.

Порівняння сучасних SIEM-систем наведено у таблиці 1.2.

Таблиця 1.2

Порівняння сучасних SIEM-систем

SIEM-система	Тип системи	Основні переваги	Основні недоліки
IBM QRadar	Комерційна	Потужна аналітика, кореляція подій, підтримка великої інфраструктури	Висока вартість та значні вимоги до ресурсів
Splunk Enterprise Security	Комерційна	Гнучкість налаштування, потужний пошук, dashboard	Складність адміністрування та дорога ліцензія
Elastic Security	Комерційна/open-source	Масштабованість, інтеграція з Elastic Stack	Складніше початкове налаштування
Wazuh	Open-source	Безкоштовне використання, підтримка IDS/IPS, dashboard	Менша кількість корпоративних функцій
Microsoft Sentinel	Хмарна/комерційна	Масштабованість, інтеграція з Azure	Залежність від хмарної інфраструктури
Graylog	Open-source	Просте розгортання, централізований збір логів	Менша кількість функцій SIEM-аналітики

У сучасних SIEM-системах однією з найважливіших функцій є підтримка візуалізації подій безпеки. Використання графічних панелей дозволяє адміністраторам швидше аналізувати події, виявляти аномалії та оцінювати загальний стан мережевої інфраструктури. Крім цього, сучасні SIEM-рішення підтримують інтегра-

цію із системами виявлення атак, міжмережевими екранами, антивірусним програмним забезпеченням та іншими компонентами системи захисту інформації, що дозволяє створити централізовану систему моніторингу та підвищити ефективність реагування на інциденти безпеки.

Таким чином, сучасні SIEM-системи є важливим елементом забезпечення інформаційної безпеки комп'ютерних мереж. Їх використання дозволяє централізувати контроль подій безпеки, автоматизувати процес аналізу журналів подій та своєчасно виявляти потенційні кіберзагрози.

1.3 Обґрунтування вибору SIEM-системи Wazuh

Під час розробки підсистеми комплексного безпекового моніторингу одним із ключових етапів є вибір SIEM-системи, яка буде використовуватись як центральний компонент аналізу та обробки подій безпеки. Враховуючи значну кількість сучасних рішень у сфері моніторингу інформаційної безпеки, під час вибору системи необхідно враховувати не лише її функціональні можливості, а й складність розгортання, підтримку інтеграції, вимоги до апаратних ресурсів та доцільність використання в межах проєкту.

Для даного проєкту було обрано саме Wazuh, адже він є не лише SIEM-системою для централізованого моніторингу подій безпеки, а й комплексною платформою, яка поєднує можливості аналізу журналів подій, моніторингу вузлів мережі, виявлення вразливостей та автоматизованого реагування на інциденти безпеки. Завдяки підтримці великої кількості функцій система може використовуватись як один із центральних компонентів підсистеми захисту інформації [11].

Однією з причин вибору Wazuh є відкритий програмний код та можливість безкоштовного використання системи. На відміну від багатьох комерційних SIEM-рішень, система не потребує придбання ліцензії, що дозволяє використовувати її у навчальних, тестових та дослідницьких середовищах без додаткових фінансових витрат.

Він побудований на базі Elastic Stack, що забезпечує підтримку пошуку, обробки та візуалізації великих обсягів журналів подій, а сама система дозволяє

централізовано збирати інформацію з різних джерел мережі, виконувати аналіз отриманих подій та формувати повідомлення про потенційні загрози. Сучасний інтерфейс, підтримка дашбордів та графічних панелей значно спрощує аналіз мережевої активності та дозволяє швидше реагувати на інциденти інформаційної безпеки [12].

Приклад інтерфейсу Wazuh наведено на рис. 1.9.

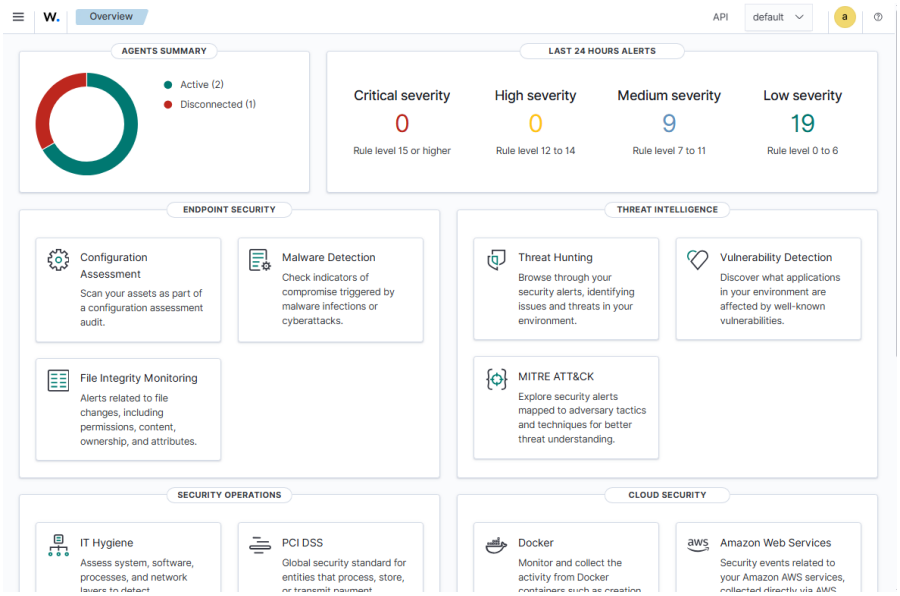


Рисунок 1.9 – Інтерфейс SIEM-системи Wazuh

Також система підтримує агентську архітектуру, що дозволяє виконувати моніторинг серверів, робочих станцій та інших вузлів мережі. Агенти можуть працювати з різними операційними системами, серед яких Linux, Windows та macOS [13]. Це дозволяє використовувати систему у гетерогенних мережах, де одночасно функціонують різні типи програмно-апаратних платформ.

Крім централізованого збору журналів подій, Wazuh підтримує:

- Контроль цілісності файлів.
- Моніторинг конфігурацій.
- Виявлення вразливостей.
- Аналіз системних журналів.
- Виявлення підозрілої поведінки.

Завдяки наявності вбудованих засобів аналізу система дозволяє виявляти аномальну активність у мережі та формувати повідомлення про можливі інциденти безпеки.

Варто зауважити, що ще однією важливою перевагою Wazuh є підтримка інтеграції з іншими засобами захисту інформації. Система може співпрацювати із системами IDS/IPS, міжмережевими екранами та антивірусним програмним забезпеченням, що дозволяє використовувати Wazuh не тільки як підсистему захисту інформації, а й централізовану платформу для аналізу подій безпеки та контролю стану мережевої інфраструктури. Підтримка миттєвого реагування також допомагає у разі виявлення підозрілої активності та може автоматично виконувати дії реагування, наприклад, запустити сценарії реагування та відповіді на загрозу.

Інтеграція з фреймворком MITRE ATT&CK використовується для класифікації тактик і технік кіберзагроз та дозволяє пов'язувати виявлені події безпеки з конкретними сценаріями атак та спрощує аналіз дій потенційного порушника [14]. Реалізовано спеціальний розділ та правила кореляції подій, які дозволяють відображати інформацію про виявлені техніки атак відповідно до матриці, що значно спрощує процес аналізу інцидентів безпеки та дозволяє адміністраторам швидше визначати характер загрози й можливі етапи реалізації атаки. Приклад розділу з тактиками та техніками наведено на рис. 1.10.

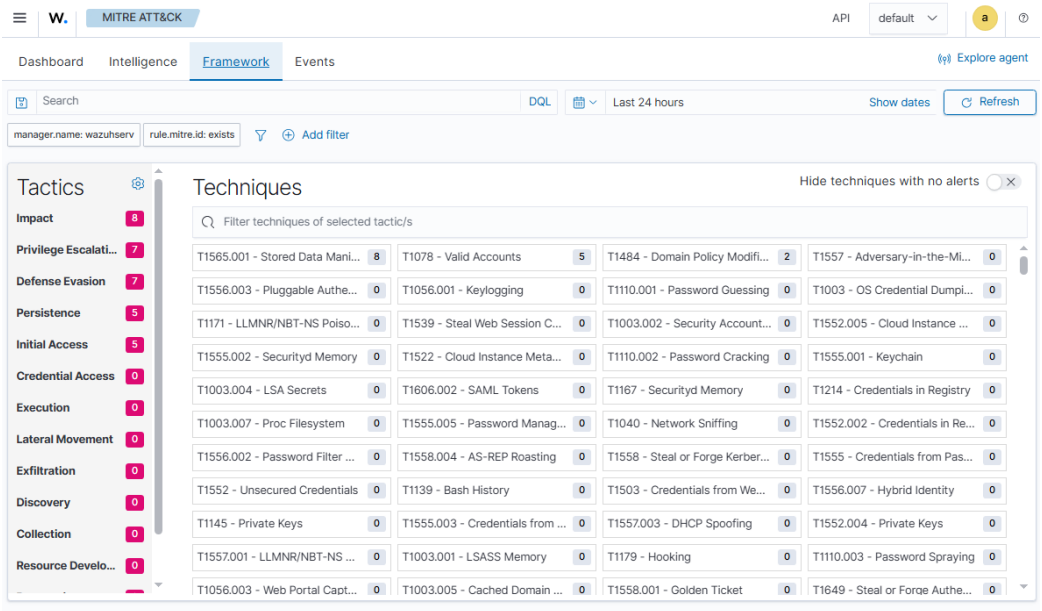


Рисунок 1.10 – Розділ MITRE ATT&CK у SIEM-системі Wazuh

Основні функціональні можливості розглянутої SIEM-системи Wazuh наведено у табл. 1.3.

Таблиця 1.3

Основні функціональні можливості Wazuh

Функція	Призначення
Збір журналів подій	Централізований збір та обробка подій безпеки
Контроль цілісності файлів	Виявлення змін у системних файлах
Виявлення вразливостей	Аналіз програмного забезпечення та пошук відомих CVE
Графічний інтерфейс	Візуалізація подій та статистики
Миттєве реагування	Автоматичне реагування на загрози
Інтеграція з системами IDS/IPS	Аналіз мережевих атак та подій безпеки

Таким чином, використання Wazuh дозволяє реалізувати сучасну централізовану систему моніторингу подій безпеки із підтримкою аналізу журналів подій, виявлення загроз та автоматизованого реагування на інциденти. Завдяки відкритому програмному коду, підтримці інтеграції та наявності розвинених засобів візуалізації система є доцільним вибором для реалізації підсистеми комплексного безпекового моніторингу в межах кваліфікаційної роботи.

Додатково під час вибору SIEM-системи було проведено порівняння сучасних рішень за критеріями функціональних можливостей, складності розгортання, підтримки інтеграції та доцільності використання у тестовому середовищі. Результати порівняльного аналізу наведено у додатку А.

1.4 Аналіз сучасних IDS/IPS-рішень

У сучасних комп'ютерних мережах важливу роль у забезпеченні інформаційної безпеки відіграють системи виявлення та запобігання мережевим атакам. Постійне зростання кількості кіберзагроз, поява нових методів атак та збільшення обсягів мережевого трафіку створюють необхідність використання засобів, здатних своєчасно виявляти підозрілу активність та реагувати на неї.

Для вирішення даної задачі використовуються IDS та IPS-системи. IDS призначені для виявлення підозрілої активності та повідомлення адміністратора про можливі інциденти безпеки. IPS, у свою чергу, не лише виявляють атаки, а й можуть автоматично блокувати небезпечний трафік або виконувати інші дії реагування [2].

Принцип роботи цих систем базується на аналізі мережевого трафіку, системних журналів та активності вузлів мережі. Тобто, система перевіряє отримані дані на наявність ознак атак, аномальної поведінки або спроб порушення політик безпеки. У разі виявлення підозрілої активності формується повідомлення про інцидент або виконується блокування джерела загрози. Загальна структура принципу функціонування систем виявлення та запобігання вторгненням продемонстрована на рис. 1.11.

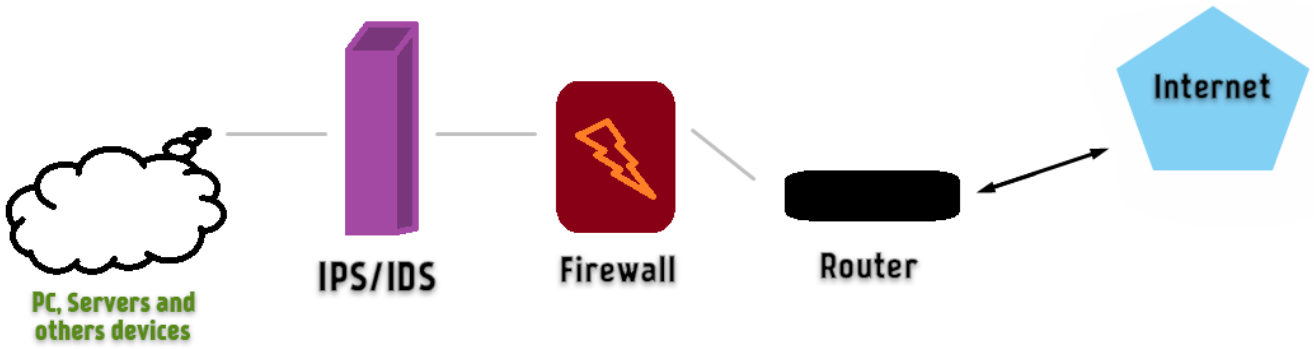


Рисунок 1.11 – Структура системи IPS/IDS

Основною відмінністю між IDS та IPS є спосіб реагування на загрозу. IDS-система працює у режимі моніторингу та аналізує мережевий трафік без прямого втручання у передачу даних. У разі виявлення атаки система формує повідомлення або записує інформацію про інцидент у журнал подій. IPS-система встановлюється

безпосередньо у потік мережевого трафіку та може автоматично блокувати підозрілі пакети або мережеві з'єднання [15].

Для виявлення атак можуть використовуватися декілька методів аналізу. Найбільш поширеними є сигнатурний та поведінковий методи. Сигнатурний метод базується на використанні бази правил або шаблонів відомих атак. Під час аналізу мережевого трафіку система порівнює отримані пакети із сигнатурами атак. У разі збігу формується повідомлення про виявлену загрозу. Перевагою даного підходу є висока точність виявлення відомих типів атак, однак система може не виявляти нові або модифіковані загрози. Поведінковий метод аналізу ґрунтується на виявленні аномальної активності у мережі або роботі вузлів системи. У цьому випадку система аналізує поведінку користувачів, мережевий трафік та роботу служб, визначаючи відхилення від нормального режиму функціонування. Такий підхід дозволяє виявляти нові типи загроз, однак може призводити до появи хибних спрацювань. У сучасних IDS/IPS-системах часто використовується комбінований підхід, який поєднує сигнатурний та поведінковий аналіз, що дозволяє підвищити ефективність виявлення атак та зменшити кількість помилкових спрацювань.

Залежно від способу розгортання IDS/IPS-системи поділяються на мережеві та хостові. Мережеві системи аналізують мережевий трафік та контролюють взаємодію між вузлами мережі. Хостові системи встановлюються безпосередньо на сервер або робочу станцію та контролюють події, які відбуваються на конкретному вузлі.

Важливою складовою роботи є підтримка правил виявлення атак, де для аналізу трафіку використовуються набори правил, які визначають ознаки підозрілої активності. Крім виявлення атак, сучасні IPS-системи підтримують автоматичне реагування на загрози. У разі виявлення небезпечної активності система може блокувати IP-адресу джерела атаки, завершувати мережеві з'єднання або обмежувати доступ до окремих ресурсів мережі.

У сучасних системах моніторингу безпеки IDS/IPS-рішення часто використовуються разом із SIEM-системами. У такому випадку інформація про виявлені

атаки передається до централізованої системи моніторингу, де виконується подальший аналіз подій безпеки. Це дозволяє не тільки створити комплексну систему захисту мережевої інфраструктури, а й підвищити ефективність реагування на кіберзагрози.

Одним із найбільш відомих IDS/IPS-рішень є Snort. Це відкрита система виявлення та запобігання вторгненням, яка використовується для аналізу мережевого трафіку, виявлення атак і ведення журналів подій. Snort підтримує роботу на Linux та Windows, використовує правила для виявлення підозрілої активності та може працювати як у режимі IDS, так і IPS [16]. Основною перевагою Snort є невелика вимогливість до ресурсів, велика кількість готових правил та поширеність у навчальних і практичних проєктах.

Suricata також є популярною відкритою IDS/IPS-системою. Вона призначена для аналізу мережевого трафіку, виявлення загроз та моніторингу мережевої активності. Suricata підтримує багатопотокову обробку трафіку, що дозволяє ефективніше використовувати ресурси процесора у навантажених мережах. Крім цього, система може працювати з правилами, подібними до правил Snort, що спрощує перехід між цими рішеннями [17].

Zeek більше орієнтований не на класичне блокування атак, а на глибокий аналіз мережевого трафіку. Він формує детальні журнали про мережеві з'єднання, DNS-запити, HTTP-трафік та інші протоколи. Його часто використовують для полювання на загрози, аналізу інцидентів та побудови систем мережевого моніторингу. Його перевагою є гнучкість і можливість детального аналізу трафіку, однак для початкового налаштування він є складнішим, ніж в інших системах [18].

Security Onion є не окремою IDS/IPS-системою, а готовою платформою для моніторингу мережевої безпеки. Вона поєднує декілька інструментів, зокрема Suricata, Zeek, засоби збору логів, графічних панелей та механізмів аналізу подій. Таке рішення зручно використовувати, коли потрібно швидко розгорнути комплексне середовище для виявлення атак і аналізу інцидентів. Але Security Onion потребує більше ресурсів, ніж окрема IDS/IPS-система [19].

Окремо також можна виділити комерційні рішення, наприклад Cisco Firepower, Fortinet FortiGate IPS та Palo Alto Threat Prevention. Такі системи зазвичай мають широку функціональність, підтримку виробника, оновлювані бази загроз та інтеграцію з корпоративною інфраструктурою. Проте їх використання пов’язане з вартістю ліцензій, потребою у спеціалізованому обладнанні та складнішим адмініструванням.

Детальне порівняння IDS/IPS-рішень наведено у табл. 1.4.

Таблиця 1.4

Порівняння IDS/IPS-рішень

SIEM-система	Тип системи	Основні переваги	Основні недоліки
Snort	Відкрита	Підтримка багатьох правил, просте розгортання, підтримка різних ОС	Для IPS-режиму потрібне додаткове налаштування
Suricata	Відкрита	Багатопотокова обробка, висока продуктивність, підтримка багатьох правил	У певних моніторингових деталях є складнішим для налаштування
Zeek	Відкрита	Детальний аналіз трафіку, зручний для пошуку вразливостей	Не є класичною IPS-системою
Security Onion	Відкрита	З самого початку готове середовище з IDS, логами та візуалізацією	Дуже високі вимоги до ресурсів
Cisco Firepower	Комерційна	Інтеграція з інфраструктурою на базі Cisco	Висока вартість
Fortinet FortiGate IPS	Комерційна	Інтеграція з firewall, оновлення сигнатур	Потребує ліцензії та спеціального обладнання

Вибір конкретного рішення та системи залежить від завдань проєкту, ресурсів середовища, потрібного рівня деталізації подій та можливостей інтеграції з SIEM-системою.

1.5 Обґрунтування вибору IDS/IPS-системи Snort

Для реалізації підсистеми виявлення атак у межах даного проєкту було обрано систему Snort, оскільки дане рішення є однією з найбільш відомих відкритих IDS/IPS-систем, а також являє собою систему аналізу мережевого трафіку, яка підтримує сигнатурний метод виявлення атак та дозволяє аналізувати мережеві пакети у реальному часі [20].

Загалом, система може працювати у декількох режимах:

- Режим IPS.
- Режим IDS.
- Режим звичайного логування.

Однією з основних причин вибору Snort є підтримка великої кількості правил виявлення атак, адже система використовує спеціальні файли, які містять сигнатури відомих загроз та дозволяють аналізувати мережевий трафік на наявність підозрілої активності. Крім готових правил, Snort підтримує створення власних сигнатур, що дозволяє адаптувати систему під особливості конкретної мережевої інфраструктури [21].

Загалом, він підтримує аналіз різних типів мережевих атак, серед яких:

- Сканування портів.
- Атаки Brute Force.
- DNS-флуд.
- ICMP-флуд та інші.

Крім цього, система дозволяє виконувати аналіз пакетів на рівні протоколів TCP, UDP, ICMP та HTTP, що дає можливість контролювати значну частину мережевої активності у тестовому середовищі.

Приклад роботи Snort продемонстровано на рис. 1.12.

```
04/27-21:25:09.836803 [**] [1:1000001:1] "ICMP connection detected" [**] [Priority: 0] {ICMP} 192.168.50.40 -> 192.168.50.10
04/27-21:25:10.841237 [**] [1:1000001:1] "ICMP connection detected" [**] [Priority: 0] {ICMP} 192.168.50.40 -> 192.168.50.10
04/27-21:25:11.844374 [**] [1:1000001:1] "ICMP connection detected" [**] [Priority: 0] {ICMP} 192.168.50.40 -> 192.168.50.10
04/27-21:25:12.874478 [**] [1:1000001:1] "ICMP connection detected" [**] [Priority: 0] {ICMP} 192.168.50.40 -> 192.168.50.10
```

Рисунок 1.12 – Виявлення мережевої активності за допомогою Snort

Snort підтримує інтеграції із SIEM-системами та засобами централізованого моніторингу. У межах даної кваліфікаційної роботи Snort використовується разом із SIEM-системою Wazuh, що дозволяє централізовано збирати та аналізувати повідомлення про виявлені атаки. Завдяки такій інтеграції події безпеки можуть відображатись у графічному інтерфейсі, використовуватись для кореляції подій та запуску механізмів активного реагування.

Важливою особливістю є підтримка роботи не лише у режимі пасивного виявлення атак, а і у режимі IPS із можливістю активного блокування підозрілих мережевих з'єднань. У такому режимі система може аналізувати мережевий трафік безпосередньо під час його проходження через вузол та виконувати автоматичне реагування на виявлену підозрілу активність.

Для реалізації такого підходу Snort може працювати у режимі «inline-mode», коли мережеві пакети проходять через механізм перевірки правил ще до їх подальшої передачі мережею. У межах Linux-систем для цього може використовуватись механізм «NFQUEUE», який дозволяє передавати мережеві пакети з «iptables» до Snort для подальшого аналізу, а у разі виявлення підозрілої активності система може не лише сформувати повідомлення про атаку, а і автоматично заблокувати відповідне мережеве з'єднання.

```
drop tcp any any -> $HOME_NET 22 (
msg:"Possible SSH brute force blocked";
detection_filter:track by_src,count 10,seconds 60;
sid:1000020;
rev:1;
)
```

Рисунок 1.13 – Правило для відкидання TCP-з'єднання через протокол SSH

На прикладі правила (рис. 1.13) видно, що воно дозволяє блокувати підозрілі TCP-з'єднання до SSH-служби у випадку виявлення характерної мережевої активності. На відміну від звичайного режиму IDS, у якому система лише формує повідомлення про подію, IPS-режим дозволить безпосередньо впливати на мережевий трафік та обмежити з'єднання ще під час виконання.

Ще однією перевагою Snort є відносно невеликі вимоги до апаратних ресурсів. У порівнянні з деякими комплексними платформами моніторингу безпеки він є простішим у розгортанні та налаштуванні, що також вплинуло на вибір даного рішення.

Основні функціональні можливості Snort наведено у табл. 1.5

Таблиця 1.5

Основні функціональні можливості Snort

Функція	Призначення
Аналіз мережевого трафіку	Контроль пакетів у мережі
Виявлення атак	Аналіз сигнатур та мережевих аномалій
Підтримка файлів з правилами	Використання та створення правил
IDS/IPS-режим	Виявлення та блокування атак
Журналювання подій	Формування відповідних лог-файлів
Інтеграція з SIEM	Передача подій до систем моніторингу

Тобто, використання Snort дозволяє реалізувати ефективну IDS/IPS-підсистему для аналізу мережевого трафіку та виявлення атак у тестовому середовищі, а підтримка сигнатурного аналізу, можливість інтеграції із обраною SIEM-системою, підтримка IPS-режиму та невеликі вимоги до ресурсів роблять його доцільним вибором для реалізації підсистеми безпекового моніторингу у межах кваліфікаційної роботи.

Під час вибору IDS/IPS-системи було виконано порівняльний аналіз сучасних рішень виявлення та запобігання мережевим атакам. Порівняння проводилось з урахуванням підтримки сигнатурного аналізу, можливостей інтеграції, складності налаштування та вимог до апаратних ресурсів. Результати аналізу наведено у додатку А.

1.6 Постановка технічного завдання

Метою кваліфікаційної роботи є проєктування та реалізація підсистеми комплексного безпекового моніторингу гетерогенної мережі із використанням SIEM-системи Wazuh та IDS/IPS-системи Snort.

Проєкт передбачає створення віртуального середовища моніторингу безпеки, яке забезпечуватиме централізований збір, аналіз подій безпеки, виявлення мережевих атак, візуалізацію подій, реалізацію механізмів реагування на інциденти.

Реалізація цієї підсистеми передбачає виконання таких завдань:

- Провести аналіз сучасних засобів безпекового моніторингу та виявлення кіберзагроз.
- Виконати аналіз сучасних SIEM-систем та IDS/IPS-рішень.
- Обґрунтувати вибір SIEM-системи Wazuh та IDS/IPS-системи Snort для реалізації проєкту.
- Розробити проєкт фізичної та логічної топології гетерогенної мережі.
- Реалізувати віртуальне середовище для моделювання мережевої інфраструктури.
- Виконати встановлення та налаштування SIEM-системи Wazuh.
- Реалізувати встановлення та налаштування IDS/IPS-системи Snort.
- Налаштувати централізований збір та аналіз журналів подій безпеки.
- Реалізувати інтеграцію Snort із SIEM-системою Wazuh.
- Налаштувати механізми візуалізації подій безпеки.
- Реалізувати механізми активного реагування для автоматизованого реагування на мережеві атаки.
- Провести моделювання мережевих атак та тестування підсистеми безпекового моніторингу.
- Виконати аналіз результатів функціонування реалізованої системи та оцінку ефективності виявлення атак.

Висновки до розділу

У першому розділі було проведено аналіз сучасних засобів безпекового моніторингу та виявлення кіберзагроз. Розглянуто принципи функціонування SIEM- та IDS/IPS-систем, а також їх роль у забезпеченні інформаційної безпеки комп'ютерних мереж. Обґрунтовано вибір Wazuh та Snort, які забезпечують централізований збір журналів подій, інтеграцію із засобами виявлення атак, а також виявлення й аналізу мережевих атак.

Результати проведеного аналізу підтвердили доцільність використання комплексного підходу до моніторингу мережевої безпеки, який поєднує централізований збір журналів подій, аналіз мережевого трафіку та механізми автоматизованого реагування на інциденти.

РОЗДІЛ 2 ПРОЄКТУВАННЯ ПІДСИСТЕМИ КОМПЛЕКСНОГО БЕЗПЕКОВОГО МОНІТОРИНГУ

2.1 Проєктування гетерогенної мережевої інфраструктури

Використання гетерогенного середовища дозволяє змоделювати роботу мережі, у якій одночасно використовуються різні операційні системи та програмні компоненти. Цей підхід наближує тестове середовище до умов реальної корпоративної мережі та дозволяє перевірити взаємодію між різними компонентами підсистеми безпеки.

Для реалізації кваліфікаційної роботи було спроєктовано гетерогенну мережеву інфраструктуру, яка включає сервери моніторингу, систему виявлення мережевих атак, клієнтські вузли та середовище генерації тестових атак.

У межах розробленої інфраструктури було передбачено використання окремого сервера моніторингу, який виконує функції SIEM-системи на базі Wazuh та забезпечує централізований збір і аналіз подій безпеки. Для виявлення мережевих атак використовується окремий вузол IDS/IPS з використанням Snort, через який проходить мережевий трафік тестового середовища. Також цей вузол виконує окрему роль у маршрутизації мережі. Крім цього, у мережі передбачено використання клієнтських вузлів та окремого вузла для моделювання атак і перевірки роботи системи моніторингу.

Для забезпечення маршрутизації трафіку між мережевими сегментами у межах тестового середовища було передбачено використання окремого BSD-вузла, який виконує функції маршрутизатора та міжмережевого екрана. Даний компонент забезпечує передачу пакетів між внутрішньою мережею та зовнішнім сегментом мережі.

Крім маршрутизації трафіку, BSD-вузол використовується для реалізації базових механізмів фільтрації мережевих пакетів. У межах проєктування підсистеми безпеки було передбачено обмеження окремих типів мережевих взаємодій між сегментами мережі. Зокрема, було реалізовано блокування ICMP-запитів із зовнішнього сегмента до внутрішньої мережі при збереженні можливості ініціювання

з’єднань із внутрішнього сегмента у зовнішній. Це дозволяє підвищити рівень ізоляції мережі та зменшити можливість зовнішнього сканування внутрішніх вузлів. Використання BSD-системи у ролі маршрутизатора також дозволяє реалізувати сегментацію мережі, контроль напрямків передачі трафіку та централізоване застосування правил мережевої безпеки. Це наближує створене тестове середовище до структури реальної корпоративної мережі, у якій міжмережеві екрани та маршрутизатори використовуються як окремі компоненти підсистеми захисту інформації.

Загальна структура гетерогенної мережевої інфраструктури наведена у додатку Б.

Під час побудови інфраструктури було використано декілька типів операційних систем. Серверна частина реалізована на базі Linux, оскільки дана ОС забезпечує стабільну роботу серверних служб, підтримує більшість сучасних відкритих рішень засобів моніторингу безпеки та дозволяє гнучко налаштовувати мережеві механізми захисту. Крім цього, Linux широко використовується у сфері кібербезпеки для реалізації систем аналізу трафіку, SIEM-рішень та IDS/IPS-комплексів.

Для моделювання клієнтського вузла у мережі використовується ОС Windows. Це дозволяє перевіряти роботу системи моніторингу у середовищі, яке наближене до реальної корпоративної мережі, оскільки значна кількість користувацьких робочих станцій у сучасних організаціях працює саме під керуванням Windows.

Для моделювання мережевих атак та перевірки ефективності роботи підсистеми безпекового моніторингу у тестовому середовищі було передбачено використання окремого вузла на базі Kali Linux. Дана ОС містить набір інструментів для аналізу мережі, генерації тестового трафіку та моделювання кіберзагроз, що дозволяє виконувати перевірку ефективності механізмів виявлення атак.

В межах розробленої інфраструктури було передбачено використання декількох мережевих сегментів. Це дозволяє розділити трафік систем моніторингу, клієнтських вузлів та зовнішніх підключень. Сегментація мережі також дозволяє більш ефективно контролювати маршрутизацію трафіку та моделювати різні сценарії атак.

Реалізація самого середовища тестування була виконана за допомогою технології віртуалізації через Oracle VirtualBox. VirtualBox – це кросплатформна програма віртуалізації, яка розширює можливості існуючого комп'ютера та його ОС, щоб він міг запускати кілька операційних систем у кількох віртуальних машинах одночасно [22]. Віртуальна інфраструктура дозволяє розгортати декілька вузлів на одному фізичному комп'ютері, спрощує процес моделювання атак та забезпечує ізоляцію тестового середовища від основної мережі. Використання віртуалізації дозволяє швидко змінювати конфігурацію середовища та відновлювати систему після тестування мережевих атак.

Спроєктована гетерогенна мережева інфраструктура дозволяє реалізувати середовище для моніторингу подій безпеки, аналізу мережевого трафіку та перевірки роботи механізмів виявлення атак. Використання різних операційних систем, сегментації мережі та окремих компонентів моніторингу дозволяє наблизити створене середовище до умов реальної корпоративної мережі та забезпечити ефективне тестування підсистеми безпеки.

2.2 Розробка схеми адресації та мережевої взаємодії компонентів

Для забезпечення коректної взаємодії між компонентами системи безпекового моніторингу було розроблено окрему схему IP-адресації та маршрутизації мережевого трафіку. У межах створеного середовища використовувалися дві логічно розділені підмережі:

- Внутрішній сегмент: 192.168.50.0/24.
- Транзитний сегмент: 192.168.40.0/24.

Внутрішня мережа застосовувалась для взаємодії серверів та клієнтських вузлів. У даному сегменті функціонували Wazuh-сервер, клієнтські системи Windows та Ubuntu, а також вузол Kali Linux для моделювання атак та перевірки роботи механізмів виявлення загроз. Транзитний сегмент використовувався для взаємодії між сервером Snort та BSD, який виступав в ролі маршрутизатора. Такий підхід дозволив реалізувати окремий канал маршрутизації між внутрішньою мережею та зовнішнім середовищем.

Адресацію вузлів у підсистемі наведено у табл. 2.1.

Таблиця 2.1

IP-адресація компонентів системи

Пристрій	Мережний адаптер/ Шлюз за замовчуванням	IP-адреса	Маска	/
Підмережа L	-	192.168.50.0	255.255.255.0	/24
Підмережа T	-	192.168.40.0	255.255.255.0	/24
Wazuh	Мережний адаптер	192.168.50.10	255.255.255.0	/24
	Шлюз за замовчуванням	192.168.50.6	-	
Snort	Мережний адаптер (enp0s3)	192.168.40.5	255.255.255.0	/24
	Мережний адаптер (enp0s8)	192.168.50.6	255.255.255.0	
BSD	Мережний адаптер	192.168.40.4	255.255.255.0	/24
PC – 1	Мережний адаптер	192.168.50.20	255.255.255.0	/24
	Шлюз за замовчуванням	192.168.50.6	-	
PC – 2	Мережний адаптер	192.168.50.30	255.255.255.0	/24
	Шлюз за замовчуванням	192.168.50.6	-	
Attacker Kali Linux	Мережний адаптер	DHCP	255.255.255.0	/24
	Шлюз за замовчуванням	192.168.50.6	-	

Основним шлюзом для вузлів локальної мережі виступав сервер Snort. Через який проходив увесь мережевий трафік внутрішнього сегмента, що дозволило реалізувати аналіз пакетів та контроль мережевої активності в режимі реального часу. Також для забезпечення маршрутизації трафіку між мережевими інтерфейсами на сервері Snort було активовано механізм “IP forwarding”, а також реалізовано NAT-маршрутизацію за допомогою вбудованих таблиць IP-адресації, що дозволило вузлам внутрішньої мережі отримати доступ до зовнішнього сегмента через BSD-маршрутизатор [23].

2.3 Проєктування архітектури підсистеми моніторингу безпеки

Архітектура системи була побудована таким чином, щоб забезпечити контроль мережевого трафіку, передачу журналів подій до SIEM-системи та можливість централізованого реагування на інциденти безпеки.

Загальна архітектура взаємодії компонентів продемонстрована на рис. 2.1.

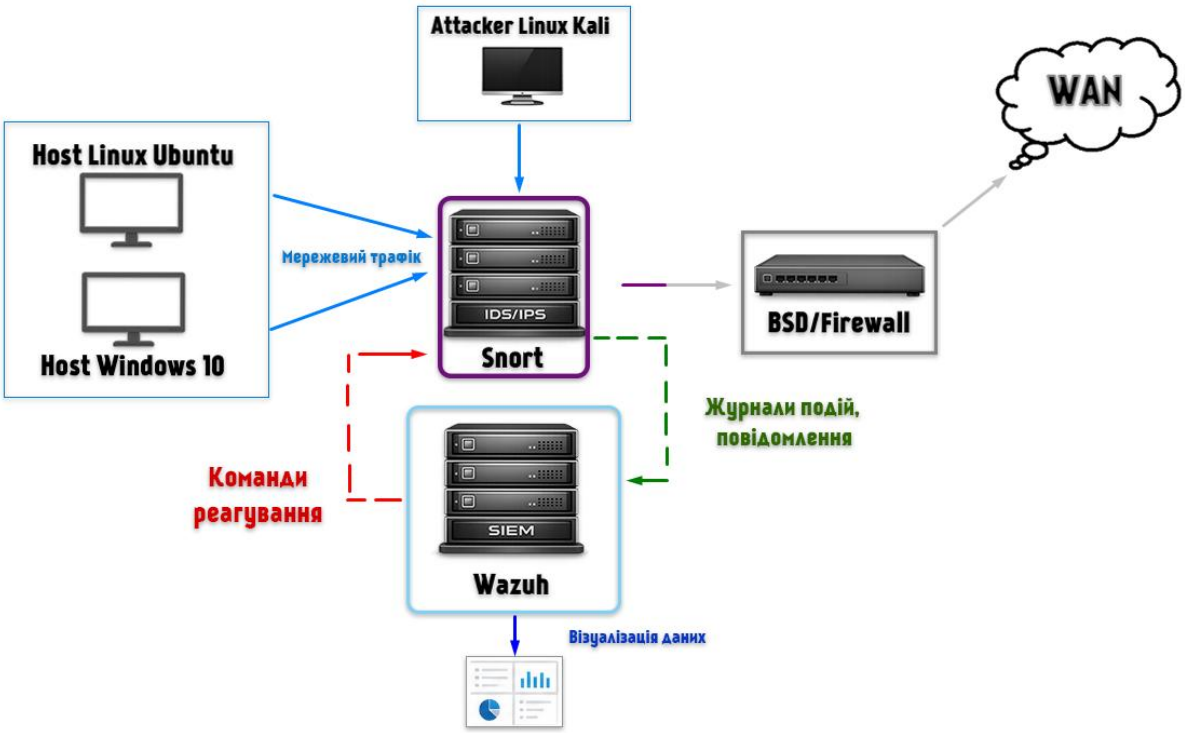


Рисунок 2.1 – Архітектура підсистеми комплексного безпекового моніторингу

Центральним компонентом архітектури виступає SIEM-система Wazuh, яка забезпечує централізований збір подій безпеки, аналіз журналів та кореляцію повідомлень про інциденти [24]. До Wazuh передаються повідомлення від IDS/IPS-системи Snort, системних служб Linux та клієнтських вузлів, що в результаті дозволяє реалізувати єдину точку моніторингу подій безпеки у межах тестового середовища.

Snort використовується як окремий вузол аналізу мережевого трафіку. Через даний сервер проходить мережевий трафік внутрішнього сегмента, що дозволяє виконувати сигнатурний аналіз пакетів та виявлення підозрілої активності. У межах архітектури Snort також виконує функції шлюзу локальної мережі та забезпечує передачу трафіку між внутрішнім і зовнішнім сегментами мережі. Після виявлення підозрілої активності Snort формує повідомлення про події безпеки та записує їх до

журналів подій. Надалі журнали передаються до Wazuh для централізованого аналізу та візуалізації у графічному середовищі. Такий підхід дозволяє поєднати функції IDS/IPS та SIEM у межах єдиної системи моніторингу.

На BSD-вузлі використовується дистрибутив pfSense, який використовується в ролі брандмауера та маршрутизатора. Він забезпечує маршрутизацію трафіку між сегментами мережі та дозволяє реалізувати базові механізми фільтрації пакетів.

Для моделювання атак та тестування роботи підсистеми моніторингу використовується окремий вузол Kali Linux як машина зловмисника. Дана система застосовується для генерації тестового мережевого трафіку, сканування мережі. Використання окремого тестового вузла дозволяє перевіряти ефективність IDS/IPS-механізмів у контрольованому середовищі.

У межах архітектури також було передбачено використання механізмів активного реагування. Тобто у випадку виявлення підозрілої активності система може автоматично виконувати блокування IP-адрес або обмеження мережевих з'єднань для конкретної адреси, що підвищує швидкість реагування на мережеві інциденти та зменшує вплив потенційних атак на компоненти інфраструктури. Сам алгоритм взаємодії компонентів системи з використанням механізмів активного реагування має таку структуру:

1. Генерація мережевого трафіку для проведення атаки.
2. Аналіз пакетів IDS/IPS-системою Snort.
3. Формування повідомлення про подію безпеки.
4. Передача журналів до Wazuh.
5. Кореляція та аналіз подій.
6. Відображення подій на панелі.
7. Реалізація механізмів активного реагування.

Для централізованого моніторингу подій у системі використовуються інформаційні панелі, або дашборд Wazuh. Інформаційні панелі дозволяють переглядати події безпеки, мережеві сповіщення, статистику атак та журнали системної активності. Крім цього, дашборд забезпечує можливість фільтрації подій, пошуку інцидентів та централізованого аналізу журналів безпеки [24].

2.4 Проєктування механізмів виявлення та реагування на атаки

У сучасних мережевих інфраструктурах значна кількість атак виконується автоматизовано, тому ефективність системи захисту значною мірою залежить від швидкості виявлення підозрілої активності та оперативності реагування на неї. У межах розробленого середовища було передбачено використання багаторівневого підходу до виявлення атак, а сам підхід поєднує:

- Аналіз мережевого трафіку.
- Перевірку сигнатур атак.
- Аналіз журналів подій.
- Кореляцію інцидентів.
- Автоматичне реагування на загрози.

Подібна архітектура відповідає базовим принципам функціонування SOC, де ключову роль відіграє централізований моніторинг подій безпеки та автоматизація реагування на інциденти [25].

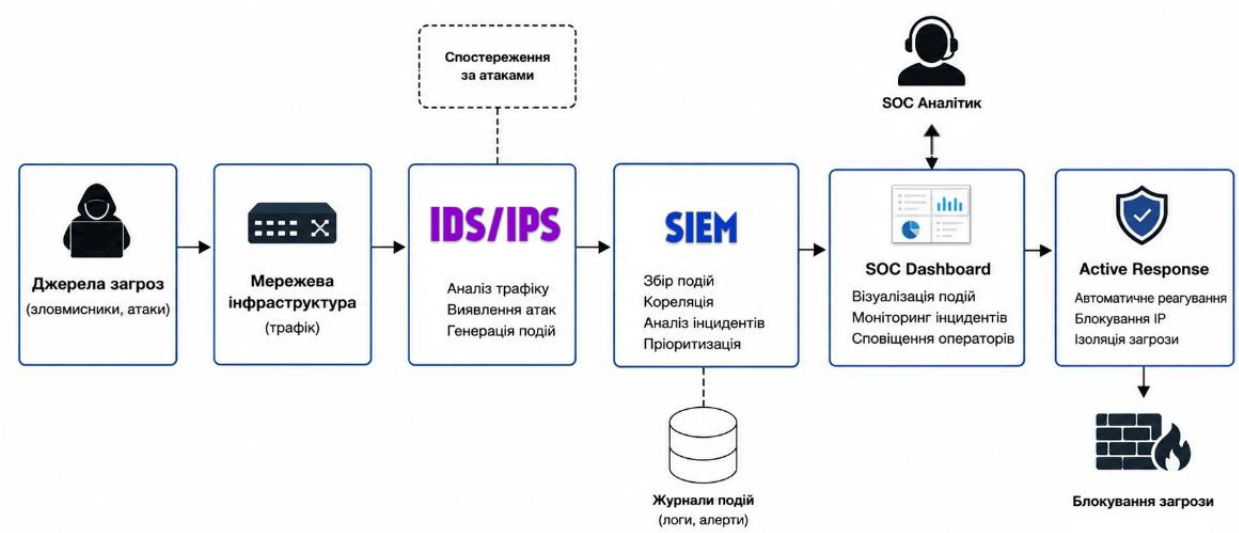


Рисунок 2.2 – Принципи функціонування SOC

Основною метою реалізації механізмів виявлення атак є своєчасне визначення підозрілої мережевої активності до моменту компрометації вузлів інфраструктури. Основна увага має приділятися не лише фіксації окремих подій, а й можливості кореляції декількох пов’язаних між собою дій для формування більш

точного уявлення про поведінку потенційного порушника. Тому у процесі проєктування було визначено перелік типів мережевої активності, які повинна виявляти підсистема моніторингу:

- Сканування портів.
- Моніторинг ICMP-запитів.
- Багаторазове ICMP-сканування.
- SSH Brute Force.
- SYN-сканування.
- Моніторинг DNS-трафіку.
- Виявлення DNS-тунелювання.
- Виявлення ICMP-тунелювання.

Одним із найпоширеніших етапів підготовки атаки є сканування мережі. Під час сканування портів виконується перевірка відкритих сервісів та активних вузлів мережі. Для виявлення подібної активності система повинна аналізувати велику кількість мережевих запитів від одного джерела протягом короткого проміжку часу. У випадку перевищення визначених порогових значень формується повідомлення про підозрілу активність.

ICMP-сканування використовується для визначення активних вузлів мережі шляхом надсилання великої кількості ехо-запитів. Було передбачено механізм аналізу інтенсивності ICMP-трафіку, що дозволяє виявляти ознаки мережевого сканування навіть у випадку відсутності TCP-з’єднань.

Окрему увагу було приділено атакам типу SSH brute force. Під час такої атаки виконується багаторазова спроба підбору облікових даних до служби SSH. У межах реалізованої системи здійснюється контроль кількості невдалих спроб авторизації та аналіз частоти підключень із однієї IP-адреси [26].

Для контролю мережевого трафіку також було реалізовано моніторинг DNS-активності. Аналіз DNS-запитів дозволяє виявляти ознаки DNS тунелювання – методу прихованої передачі даних через DNS-протокол. Контролюється частота DNS-запитів та повторюваність підозрілих доменів, що дозволяє визначати аномальну DNS-активність [27].

Окрім DNS-тунелювання було реалізовано виявлення ICMP-тунелювання. Даний механізм використовує ICMP-пакети для прихованої передачі інформації між вузлами мережі, а подібна активність може використовуватись для обходу мережевих обмежень або створення прихованого каналу зв'язку. Для виявлення ICMP tunneling система аналізує структуру ICMP-пакетів, частоту їх передачі та аномальну поведінку вузлів мережі.

Такоє у межах системи також було передбачено реалізацію механізмів кореляції подій безпеки. Кореляція дозволяє аналізувати ці взаємозв'язки між окремими подіями та визначати складні сценарії атак. Наприклад, поєднання ICMP-сканування, SYN-сканування та підозрілих SSH-підключень може свідчити про підготовку до компрометації вузла, що дозволяє зменшити кількість хибнопозитивних спрацювань та підвищити точність виявлення загроз [28].

Для централізованого аналізу подій безпеки використовується SIEM-платформа Wazuh, яка забезпечує збір повідомлень від IDS/IPS-системи Snort та формування централізованої бази подій безпеки [29].

Використання такої структури та єдиного центру моніторингу дозволяє вирішити одразу кілька проблем:

1. Спростити аналіз інцидентів.
2. Зменшити час реагування.
3. Централізувтаи обробку журналів подій.
4. Реалізувати автоматизовану кореляцію повідомлень безпеки.

Варто зауважити, що одним із важливих елементів архітектури також є механізм активного реагування. Даний механізм використовується для автоматичного реагування на події із високим рівнем критичності.

Тоюто, у випадку виявлення атаки система може виконувати:

- Обмеження доступу до сервісу.
- Завершення мережевого з'єднання.
- Блокування IP-адреси.
- Ізоляцію підозрілого вузла.

У межах проєктування було враховано, що автоматичне блокування повинно застосовуватись лише до подій, які мають високий рівень достовірності. Це дозволяє зменшити кількість хибних спрацювань та уникнути блокування легітимного трафіку.

Логіку взаємодії механізмів у разі виявлення та реагування на інциденти продемонстровано на рис. 2.3.

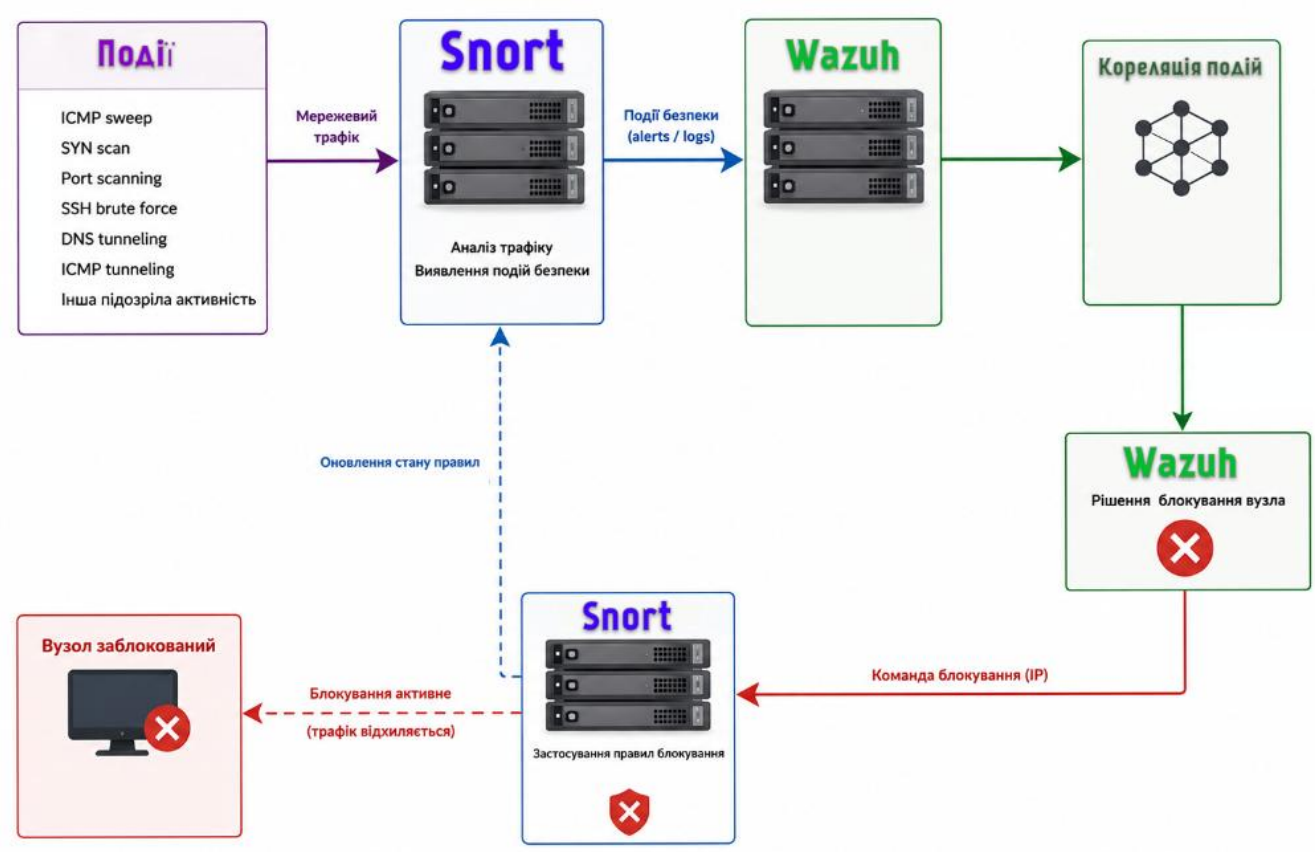


Рисунок 2.3 – Логіка взаємодії механізмів у разі виявлення атаки

В результаті, спроєктовані механізми виявлення та реагування дозволяють реалізувати комплексний контроль мережевої активності, автоматизувати процес аналізу подій безпеки та забезпечити своєчасне реагування на потенційні загрози. Використання централізованого моніторингу, кореляції подій та активне реагування дозволяє підвищити ефективність функціонування підсистеми безпекового моніторингу та наблизити її до принципів роботи сучасних SOC-центрів.

2.5 Аналіз сценаріїв мережевих атак для тестування системи

У сучасних умовах більшість кібератак складається не з однієї дії, а з декількох послідовних етапів. Зловмисники можуть виконувати розвідку мережі, аналіз

доступних сервісів, спроби підбору облікових даних або приховану передачу даних через дозволені мережеві протоколи. Саме тому тестування систем моніторингу повинно охоплювати декілька категорій мережевих загроз та сценаріїв атак [30].

Для оцінки ефективності спроєктованої підсистеми безпекового моніторингу необхідно сформувати набір тестових сценаріїв, які дозволяють перевірити здатність системи виявляти різні типи мережевих загроз та підозрілої активності. Основна увага приділялась атакам, які:

- Часто зустрічаються у реальних корпоративних мережах.
- Можуть бути виявлені за допомогою IDS/IPS-систем.
- Дозволяють перевірити механізми аналізу журналів подій.
- Створюють характерну мережеву активність.
- Дозволяють оцінити якість кореляції подій безпеки.

Одним із базових сценаріїв тестування є мережеве сканування. Подібний тип активності використовується зловмисниками на початковому етапі атаки для збору інформації про вузли мережі, відкриті порти, активні сервіси та дозволяє визначити потенційні точки входу до системи та оцінити доступність окремих вузлів.

Окремим різновидом мережевої розвідки є ICMP-сканування. Даний тип активності використовується для визначення активних вузлів локальної мережі шляхом надсилання великої кількості ICMP-пакетів. У корпоративних мережах подібна поведінка може свідчити про підготовку до подальших атак або про спробу побудови карти мережевої інфраструктури.

SYN scan є ще одним поширеним методом мережевого сканування. Особливістю такого типу активності є формування великої кількості TCP SYN-запитів до різних портів вузлів мережі. Аналіз подібної поведінки дозволяє перевірити здатність IDS/IPS-систем визначати ознаки мережевої розвідки та аномальної активності у мережевому трафіку [31].

Окрему увагу під час тестування доцільно приділити атакам типу brute force. Подібні атаки використовуються для багаторазового підбору логінів та паролів до мережевих сервісів. Найчастіше подібна активність спрямована на SSH, FTP або web-сервіси. У межах даної роботи розглядається SSH brute force, оскільки SSH є

одним із найпоширеніших протоколів віддаленого адміністрування серверів Linux. Компрометація облікових записів залишається однією з найпоширеніших причин несанкціонованого доступу до інформаційних систем. Навіть при наявності сучасних засобів захисту слабкі паролі або неправильне налаштування сервісів можуть стати причиною успішної атаки.

Для перевірки аналізу DNS-трафіку також було обрано сценарій DNS-тунелювання. Даний тип активності передбачає використання DNS-запитів для передачі прихованих даних між вузлами мережі. Через те, що DNS-запити зазвичай дозволені у більшості мереж, подібна активність може бути складною для виявлення традиційними мережевими засобами захисту. Саме тестування дозволяє оцінити можливість виявлення:

- Аномальної кількості DNS-запитів.
- Нетипових DNS-імен.
- Підозрілої структури DNS-трафіку.
- Повторюваних DNS-звернень до зовнішніх вузлів.

Для перевірки аналізу мережевого трафіку також можуть використовуватись сценарії ICMP-тунелювання. У цьому випадку ICMP-пакети використовуються для прихованої передачі даних між вузлами мережі. Подібний тип активності дозволяє перевірити здатність системи аналізувати нетипову поведінку ICMP-трафіку та виявляти ознаки потенційного прихованого каналу зв'язку [32].

Основні етеапи реалізації мережевої атаки наведено на рис. 2.4.



Рисунок 2.4 – Основні етапи реалізації мережевої атаки

Доцільно враховувати комбіновані сценарії атак, адже у реальних умовах зловмисники рідко використовують лише один тип активності. Наприклад, після етапу мережевого сканування може виконуватись підбір облікових даних або спроба експлуатації знайдених сервісів. Саме тому комплексне тестування дозволяє оцінити поведінку системи під час виникнення послідовних етапів атаки [33].

Також у межах проєкту потрібно врахувати можливість їх реалізації у віртуальному середовищі Oracle VirtualBox без необхідності використання додаткового фізичного обладнання, що дозволить створити контрольоване середовище для моделювання мережевих атак та подальшого аналізу поведінки системи моніторингу.

Висновки до розділу

У другому розділі виконано проєктування підсистеми комплексного безпекового моніторингу мережевої інфраструктури. Визначено архітектуру системи, склад її основних компонентів та принципи їх взаємодії у межах єдиного середовища безпекового моніторингу.

У процесі проєктування сформовано структуру віртуального тестового середовища із сегментацією мережі, використанням окремих вузлів маршрутизації, централізованого моніторингу та аналізу мережевого трафіку. Передбачено інтеграцію SIEM- та IDS/IPS-компонентів для забезпечення централізованого збору журналів подій, виявлення мережевих атак та підтримки механізмів реагування на інциденти інформаційної безпеки.

За результатами проєктування визначено основні технічні рішення щодо побудови підсистеми моніторингу, підготовлено структуру взаємодії між компонентами системи та сформовано основу для подальшої практичної реалізації й тестування розробленого рішення.

РОЗДІЛ 3 РОЗГОРТАННЯ, НАЛАШТУВАННЯ ТА ТЕСТУВАННЯ ПІДСИСТЕМИ БЕЗПЕКОВОГО МОНІТОРИНГУ

3.1 Розгортання віртуального середовища та мережевої інфраструктури

Для реалізації підсистеми комплексного безпекового моніторингу було розгорнуто тестове віртуальне середовище, яке дозволяє моделювати роботу корпоративної мережі та виконувати перевірку механізмів виявлення мережевих атак у контрольованих умовах.

У межах роботи для створення віртуальної інфраструктури використовувалась платформа Oracle VM VirtualBox, що дозволяє створювати ізольовані віртуальні машини, налаштовувати кілька мережевих адаптерів та моделювати взаємодію між окремими сегментами мережі. Використання VirtualBox також дозволило реалізувати тестове середовище без необхідності використання окремого фізичного обладнання.

Першим етапом практичної реалізації підсистеми безпекового моніторингу стало створення віртуального середовища, яке дозволяє моделювати роботу корпоративної мережі та проводити тестування мережевих атак у контрольованих умовах. Як було описано у попередньому розділі основна локальна мережа середовища була побудована на базі підмережі 192.168.50.0/24. У межах даної мережі розміщувались сервери та клієнтські вузли, між якими здійснювався обмін мережевим трафіком. Окремо використовувалась підмережа 192.168.40.0/24 для організації взаємодії між сервером з IDS/IPS та BSD.



Рисунок 3.1 – Створення віртуальної підмережі 192.168.50.0



Рисунок 3.2 – Створення віртуальної підмережі 192.168.40.0

Після створення мережі можемо створювати віртуальні машини. Після остаточного створення набору ВМ необхідних для підсистеми безпекового моніторингу починаємо додавати їх до Oracle VirtualBox.

Створюємо віртуальні сервери ОС Ubuntu Server 24.04 LTS на яких в подальшому буде розміщена SIEM-ситема Wazuh та Snort (рис. 3.3, 3.4).

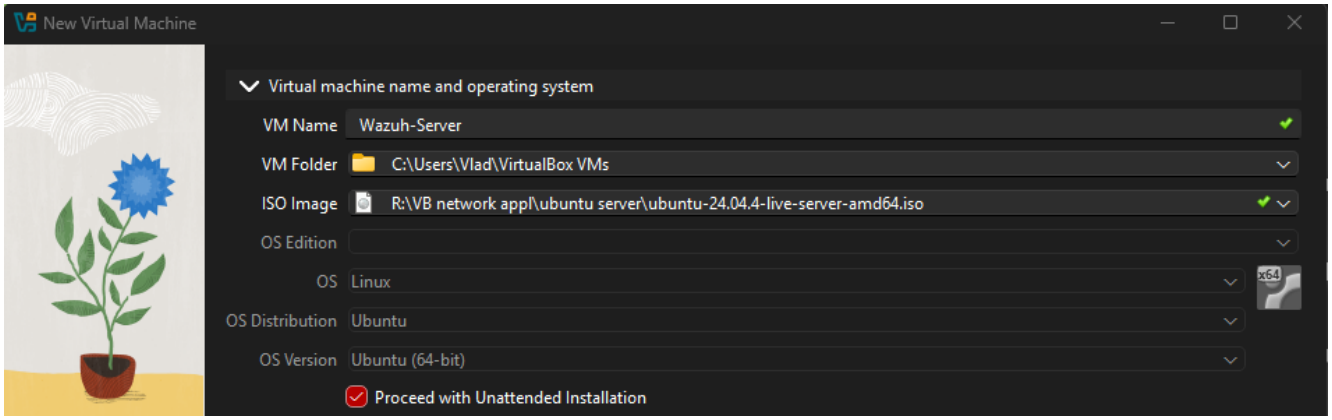


Рисунок 3.3 – Створення ВМ Wazuh-Server

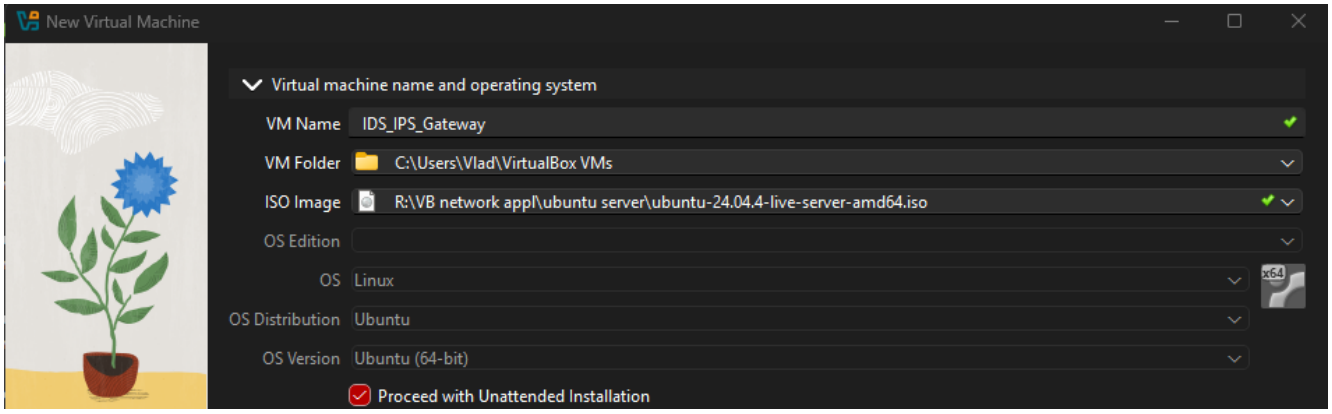


Рисунок 3.4 – Створення ВМ IDS_IPS_Gateway

Тепер створимо та імпортуємо образи клієнтських вузлів, а саме ОС Windows 10 Education 22H2 та ОС Ubuntu Desktop 24.04 LTS (рис. 3.5, 3.6).

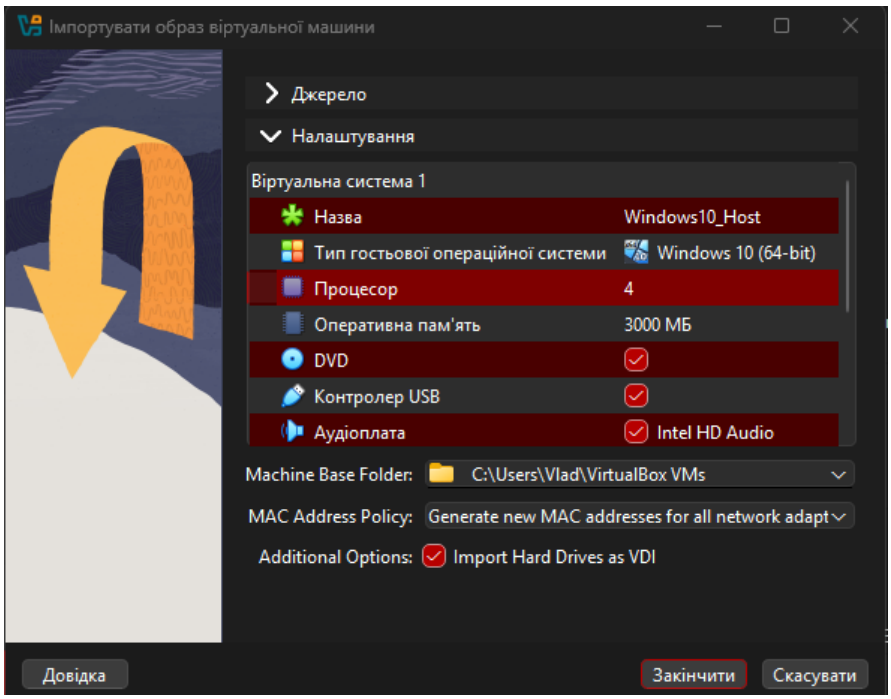


Рисунок 3.5 – Імпортування ВМ Windows 10

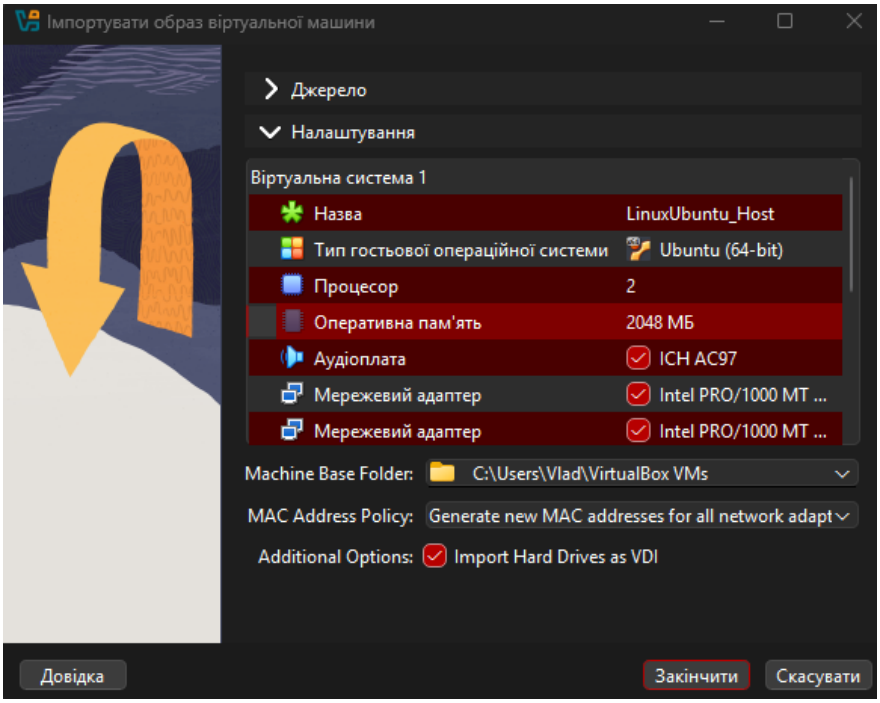


Рисунок 3.6 – Імпортування ВМ Ubuntu Desktop

Одразу додамо машину на базі Kali Linux для проведення тестових атак в мережі.

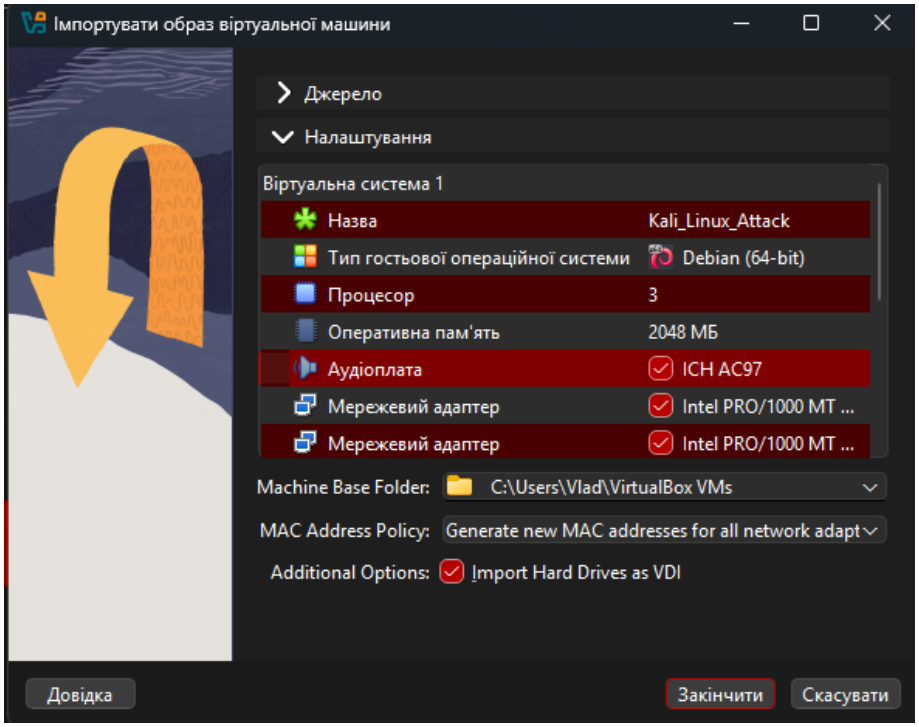


Рисунок 3.7 – Імпортування ВМ Kali Linux

Створення образу FreeBSD з дистрибутивом pfSense продемонстровано на рисунку 3.8.

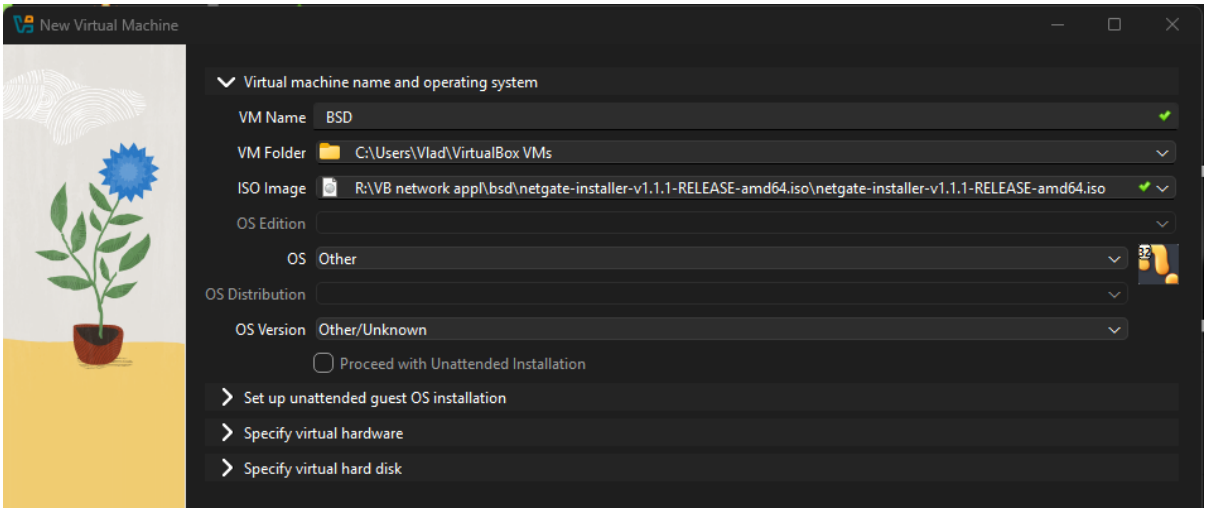


Рисунок 3.8 – Створення образу FreeBSD

Кожній машині виділяється свій ресурс. Важливо розподілити його так, щоб навантаження на основну робочу станцію було помірним, а схема могла працювати без проблем. Найбільше їх виділяється серверу з Wazuh, адже мінімальні вимоги складають 2 ядра процесору та 4 ГБ оперативної пам’яті. Проаналізувавши структуру та вимоги, було вирішено виділити йому 4 ядра та 7 ГБ оперативної пам’яті.

Після створення віртуальних машин виконується встановлення операційних систем. Під час встановлення ОС Ubuntu Server важливо звернути увагу на коректність всіх налаштувань, адже це є основою для подальшого встановлення всіх рішень. Під час першого запуску система пропонує вибрати варіант встановлення: звичайний або оптимізований. Оптимізоване встановлення дозволяє зменшити обсяг ОС на диску, але і прибирає деякий функціонал. Оскільки архітектура системи будується з розрахунком на подальше розширення системи, то найкращим варіантом буде звичайне встановлення ОС.

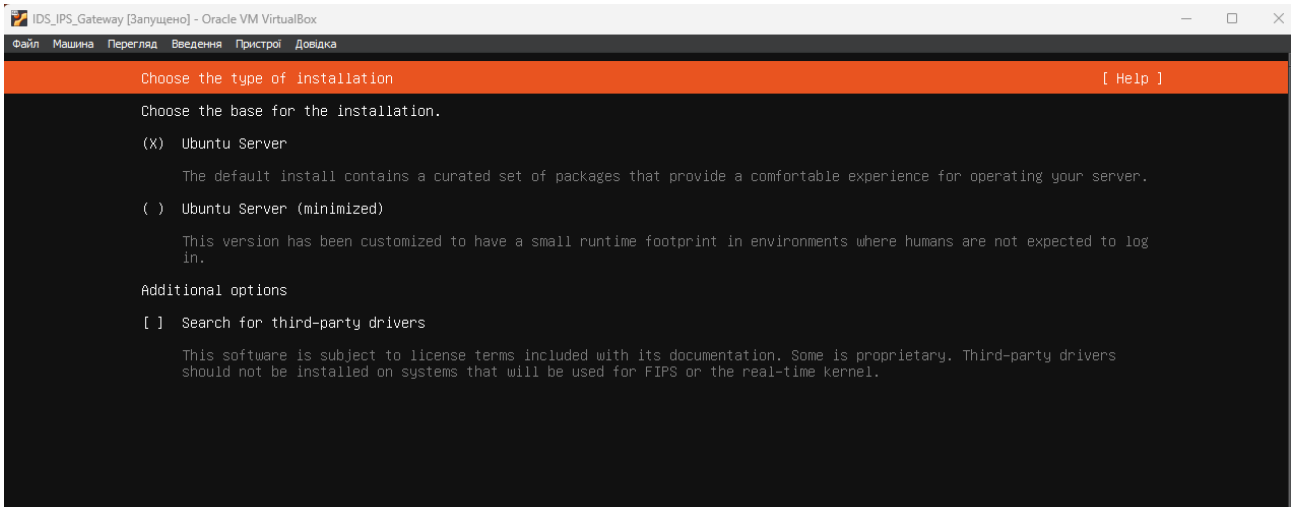


Рисунок 3.9 – Вибір режиму встановлення ОС Ubuntu Server

Також особливу увагу при встановлені потрібно звернути на розмежування дискового простору та додаткових можливостей системи. Система пропонує встановити додаткові пакунки для зручності користування з самого початку використання системи.

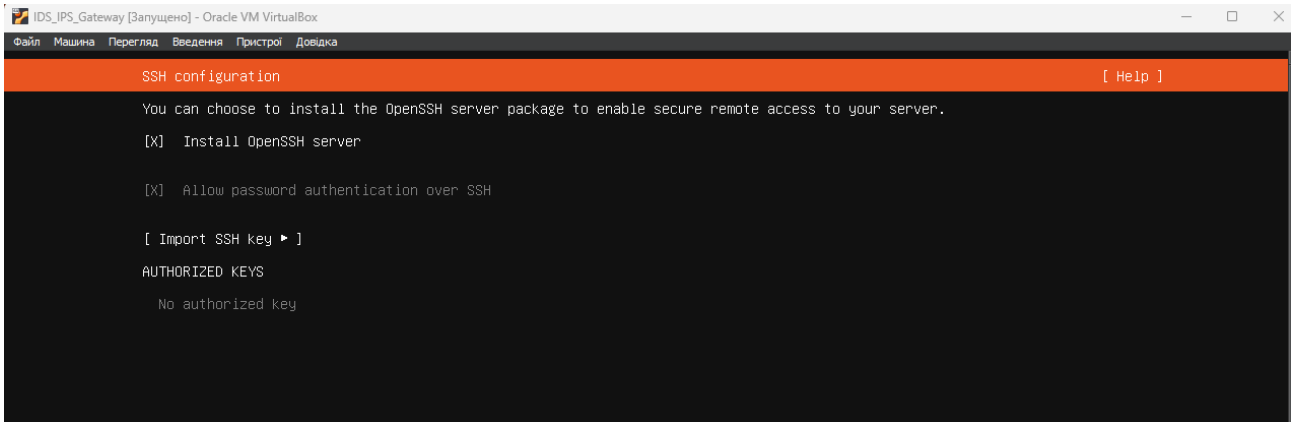


Рисунок 3.10 – Встановлення SSH на ОС Ubuntu Server

Однією з таких можливостей є встановлення SSH-сервера (рис. 3.10), що дозволяє віддалено керувати сервером за використання протоколу SSH. Обираємо дану функцію, адже керування віддаленим доступом за допомогою цього протоколу є популярним у сучасній інфраструктурі. Також цей протокол використовується і у інших задачах мережевої інфраструктури.

Перед встановленням FreeBSD pfSense в налаштуваннях віртуальної машини потрібно виставити відповідні мережеві адаптери, бо під час встановлення системи потрібно проводити налаштування мережевих адаптерів на виходи LAN та WAN.

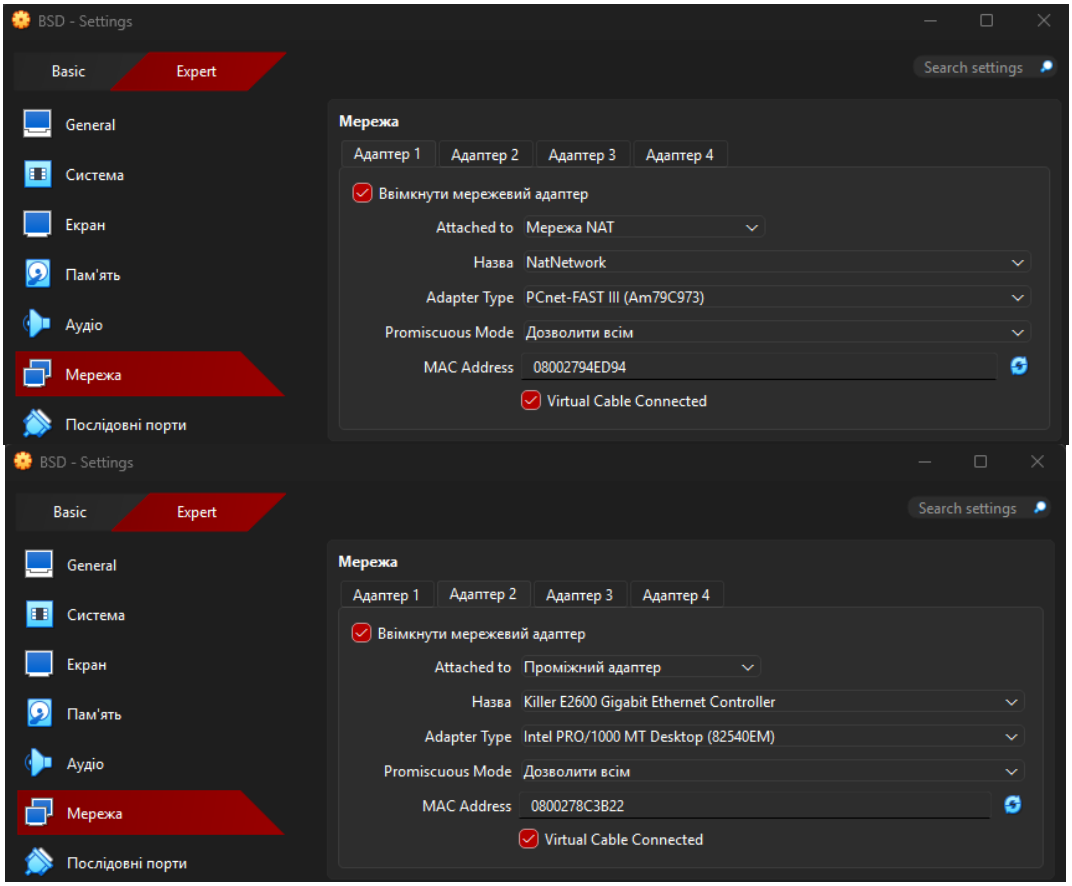


Рисунок 3.11 – Вибір мережевих адаптерів для ВМ FreeBSD pfSense

На рисунку 3.11 можемо побачити, що на перший адаптер призначена створена віртуальна мережа, а на другий призначений проміжний адаптер. Таке рішення дозволяє з'єднати внутрішню мережу із зовнішньою та забезпечити доступ до виходу в глобальну мережу Інтернет.

Для серверу з Snort також встановлюється два адаптери, які з'єднують внутрішню підмережу з FreeBSD. Різниця в тому, що на обидва адаптера виставляються створені віртуальні підмережі. Такий підхід дозволяє не тільки вихід в глобальну

мережу, а й забезпечує належну маршрутизацію всередині підмережі. Для Snort це означає повний контроль всього трафіку в мережі, оскільки він є шлюзом за замовчуванням для внутрішньої підмережі.

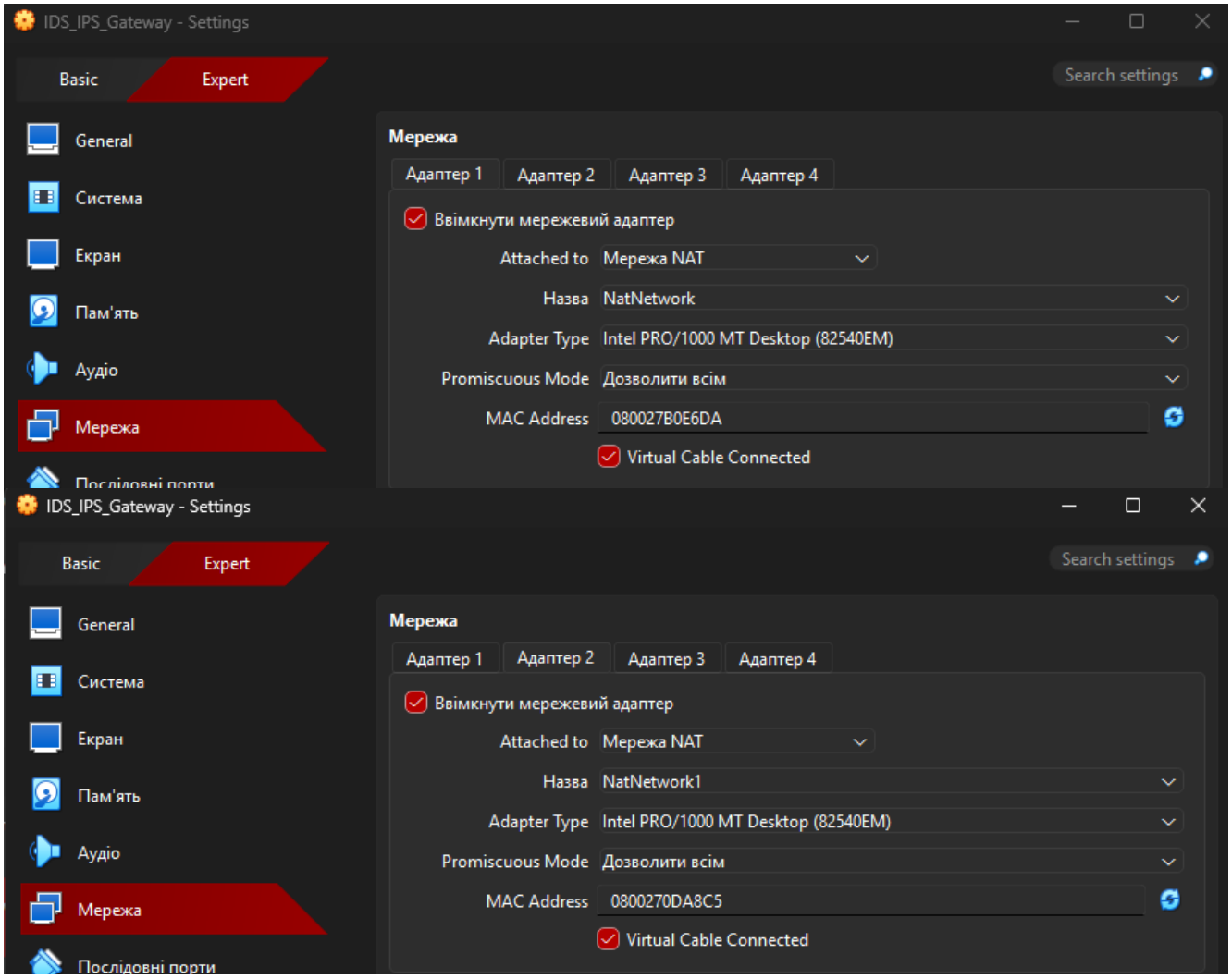


Рисунок 3.12 – Вибір мережевих адаптерів для ВМ IDS_IPS_Gateway

Для кінцевих вузлів та серверу з SIEM-системою Wazuh вибирається лише один адаптер з внутрішньою мережею.

Після налаштування мережевих адаптерів на віртуальних машинах, можна перейти до першого запуску ВМ FreeBSD pfSense. Система пропонує встановлення та налаштування мережевий адаптерів.

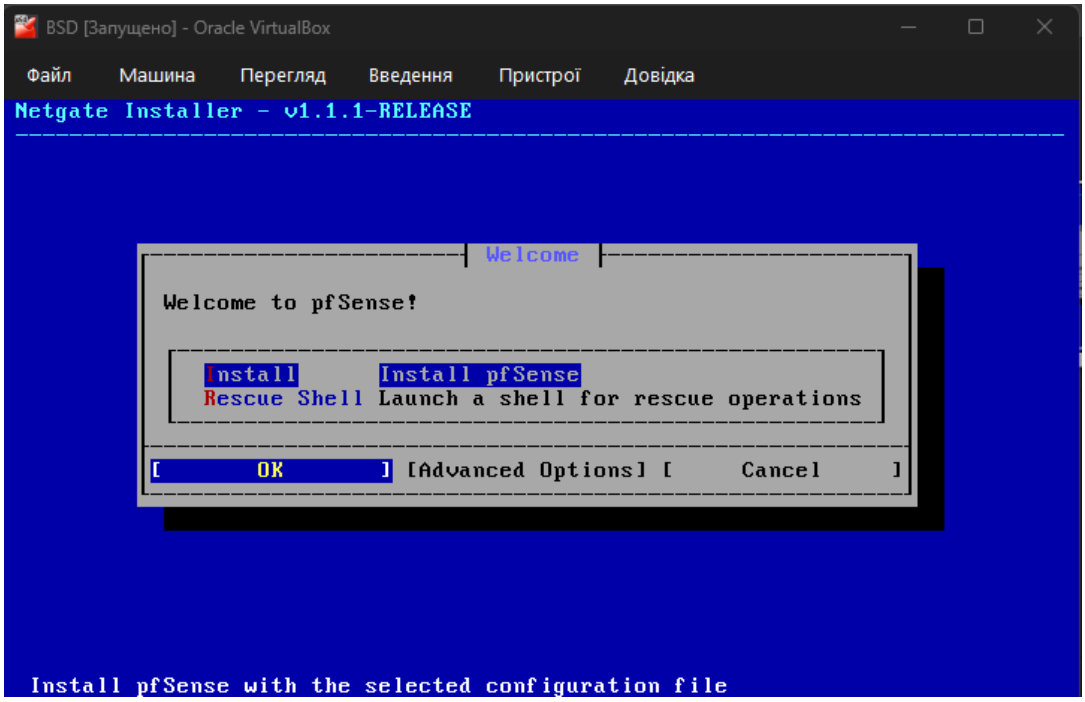


Рисунок 3.13 – Встановлення FreeBSD pfSense

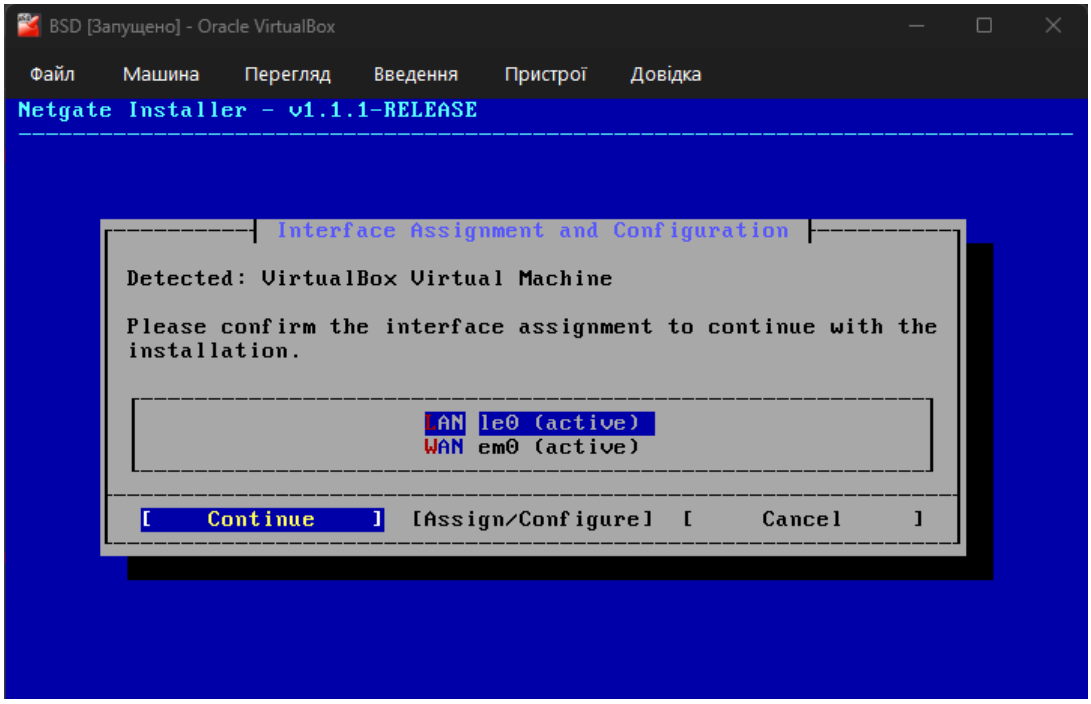


Рисунок 3.14 – Налаштуваня мережевих адаптерів

Після всіх проведених налаштувань система запропонує вибрати версію для інсталяції. Обираємо останню стабільну версію та встановлюємо (рис. 3.15).

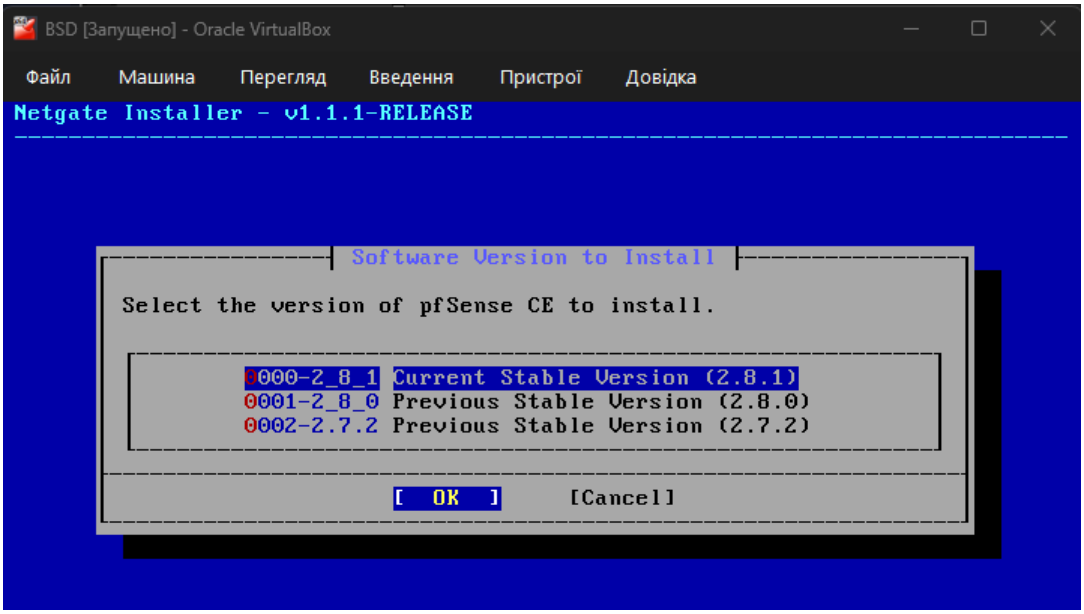


Рисунок 3.15 – Вибір версії для встановлення

Детальне встановлення та налагодження pfSense наведено у додатку В.

Після встановлення віртуальних машин необхідних для функціонування мережі, проведемо налаштування IP-адресації на них. Для початку налаштуємо адресацію на сервері, який буде шлюзом за замовчуванням. Після цього налаштовуємо адресацію ВМ Wazuh-Server (рис. 3.16), а в якості шлюзу за замовчуванням вказуємо попередній сервер.

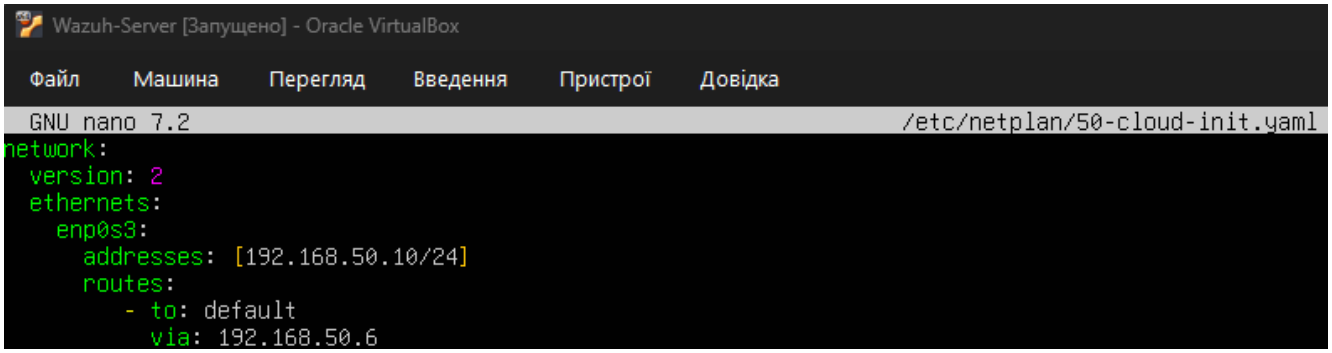


Рисунок 3.16 – Налаштування IP-адресації на ВМ Wazuh-Server

Налаштування адресації на кінцевих вузлах проводиться аналогічно, тільки в графічному інтерфейсі. Приклад налаштування IP-адресації на одному з кінцевих вузлів наведено на рис. 3.17.

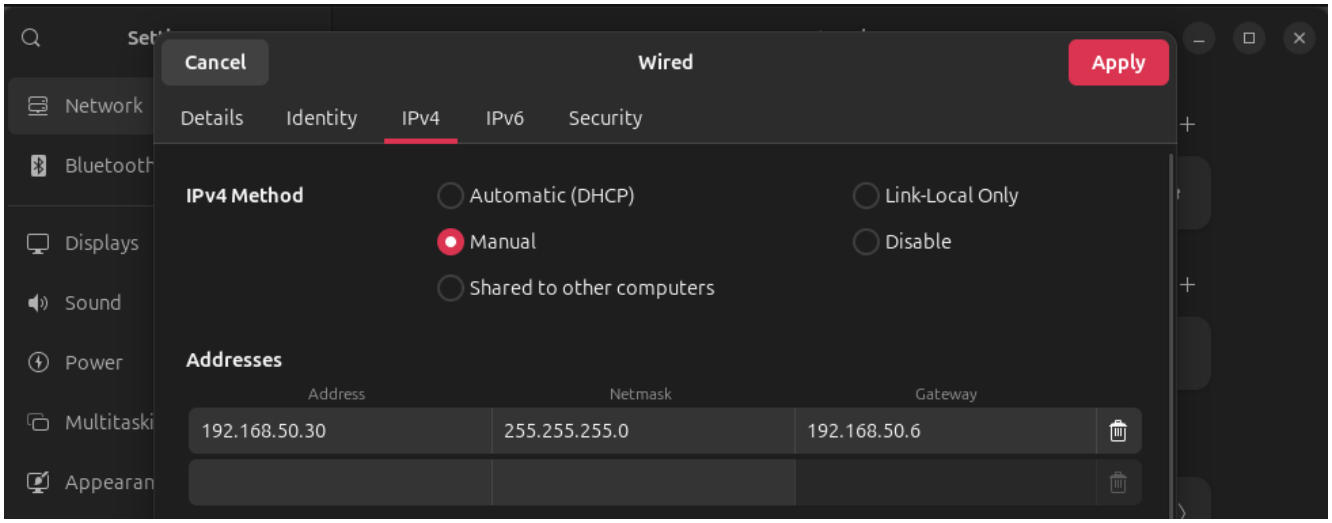


Рисунок 3.17 – Налаштування IP-адресації на ВМ Linux_Ubuntu_Host

Після проведених налаштувань залишається лише налагодження самої маршрутизації на сервері, який є шлюзом за замовчуванням для всіх інших пристроїв підмережі. Налаштування маршрутизації трафіку виконується за допомогою активації функції IP Forwarding та правил в таблиці маршрутизації.

```
snortgt@snortgt:~$ sudo sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
snortgt@snortgt:~$ sudo iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE
snortgt@snortgt:~$ sudo iptables -A FORWARD -i enp0s8 -o enp0s3 -j ACCEPT
snortgt@snortgt:~$ sudo iptables -A FORWARD -i enp0s3 -o enp0s8 -m state --state ESTABLISHED,RELATED -j ACCEPT
snortgt@snortgt:~$
```

Рисунок 3.18 – Налагодження маршрутизації трафіку

На рисунку 3.18 можемо побачити, що спочатку вмикається функція переадресації IP-адрес між інтерфейсами. Після цього створюється правило NAT, щоб пакети з внутрішньої мережі виходили через зовнішній інтерфейс з його IP-адресою. Далі налаштовується правило, яке дозволяє пересилання трафіку з внутрішнього інтерфейсу до зовнішнього та правило, яке дозволяє зворотний трафік для вже встановлених або пов'язаних з'єднань. В результаті таких налаштувань сервер діє як шлюз між двома мережами, забезпечуючи NAT і двосторонній обмін даними.

Для віддаленого підключення до серверів тестового середовища використовується PuTTY – програмне забезпечення та клієнт SSH та Telnet з відкритим вихідним кодом, яке використовується для віддаленого керування доступом [34].

Для підключення потрібно налаштувати правило переадресування порту між мережами всередині ПЗ VirtualBox. Оскільки SSH встановлено на обох серверах з ОС Ubuntu Server, то і переадресування порту буде стосуватися саме них.

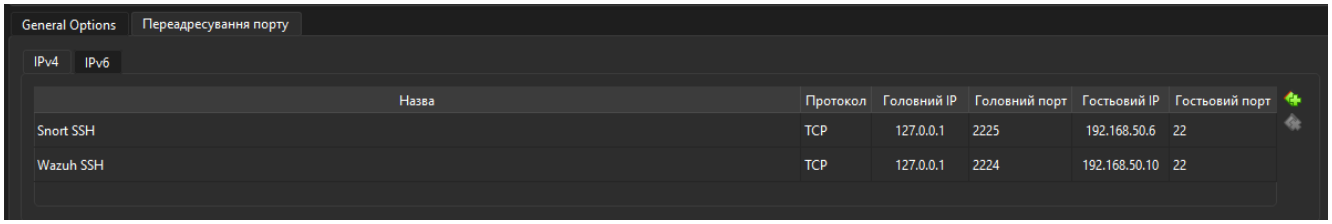


Рисунок 3.19 – Переадресування портів у Oracle VirtualBox

Тепер спробуємо під'єднатися за допомогою віддаленого доступу через PuTTY. Для підключення вводимо локальну IP-адреса та призначений порт, а також вибираємо тип підключення.

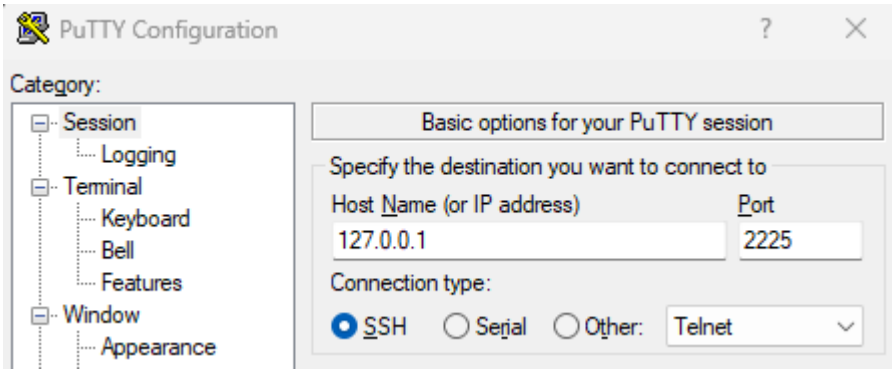


Рисунок 3.20 – Введення даних для підключення

Після встановлення з'єднання потрібно пройти автентифікацію за допомогою логіну та пароля користувача сервера.



Рисунок 3.21 – Автентифікація користувача

У разі успішної автентифікація відкривається термінал в якому можна віддалено керувати сервером (рис. 3.22).

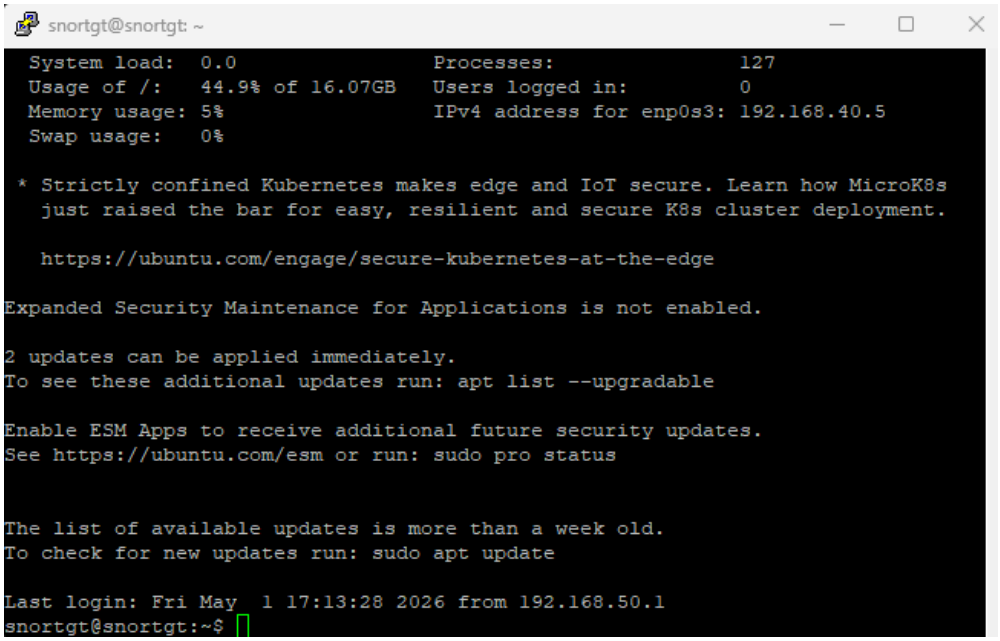


Рисунок 3.22 – Успішне підключення до сервера через протокол SSH

Таким чином, у межах даного етапу було розгорнуто віртуальне середовище та виконано базове налаштування мережевої інфраструктури, необхідної для подальшого розгортання та тестування компонентів підсистеми безпекового моніторингу.

3.2 Налаштування SIEM-системи Wazuh

Після розгортання віртуального середовища наступним етапом стало налаштування SIEM-системи Wazuh для централізованого моніторингу подій безпеки, збору журналів подій та аналізу мережевої активності. Для зручності встановлення було обрано варіант «Wazuh All-in-One», який включає всі необхідні компоненти: Wazuh Manager, Wazuh Indexer та Wazuh Dashboard. Подібний підхід дозволяє відразу розгорнути всі основні компоненти SIEM-системи на одному сервері та спрощує процес налаштування тестового середовища [35].

Перед встановленням системи виконувалося оновлення пакетів ОС та встановлення необхідних залежностей.

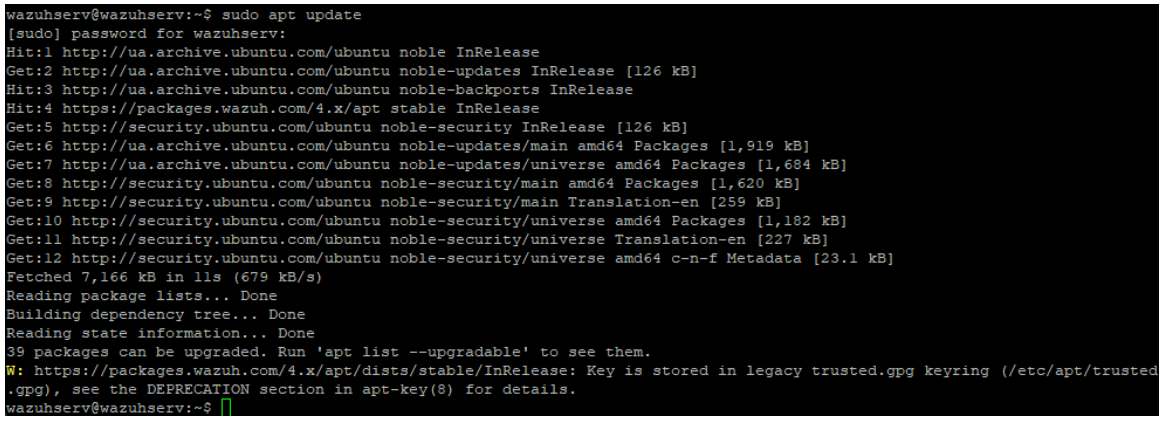


Рисунок 3.23 – Оновлення пакетів та необхідних залежностей

Після підготовки системи завантажуємо останнню версію Wazuh 4.14 та офіційний інсталяційний скрипт.



Рисунок 3.24 – Завантаження та виконання скрипта

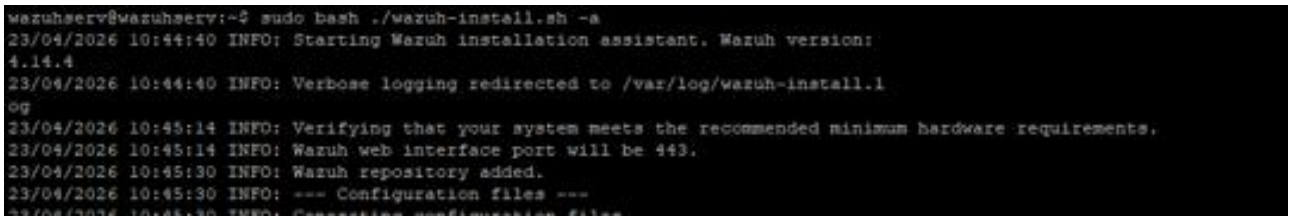


Рисунок 3.25 – Встановлення SIEM-системи Wazuh

Встановлення є довготривалим процесом та займає приблизно від 10 до 15 хвилин. Виконується покрокове додавання всіх компонентів та їх служб, а також налаштовуються зв’язки. Після завершення встановлення надаються дані створеного користувача для входу до веб-інтерфейсу.

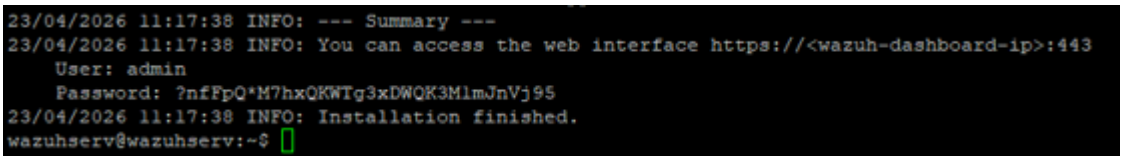


Рисунок 3.26 – Дані для входу

Тепер вмикаємо служби та створюємо залежності між компонетами.

```
wazuhserv@wazuhserv:~$ sudo systemctl enable --now filebeat
Synchronizing state of filebeat.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable filebeat
wazuhserv@wazuhserv:~$ sudo systemctl enable --now wazuh-dashboard
Created symlink /etc/systemd/system/multi-user.target.wants/wazuh-dashboard.service → /etc/systemd/system/wazuh-dashboard.s
ervice.
wazuhserv@wazuhserv:~$ sudo systemctl enable --now wazuh-manager
wazuhserv@wazuhserv:~$ sudo systemctl enable --now wazuh-indexer
Synchronizing state of wazuh-indexer.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable wazuh-indexer
```

Рисунок 3.27 – Запуск служб

Після встановлення системи було виконано налаштування доступу до Wazuh Dashboard через веб-інтерфейс. Щоб перейти через браузер потрібно ввести IP-адресу сервера та порт 443, а після завантаження ввести надані логін та пароль.

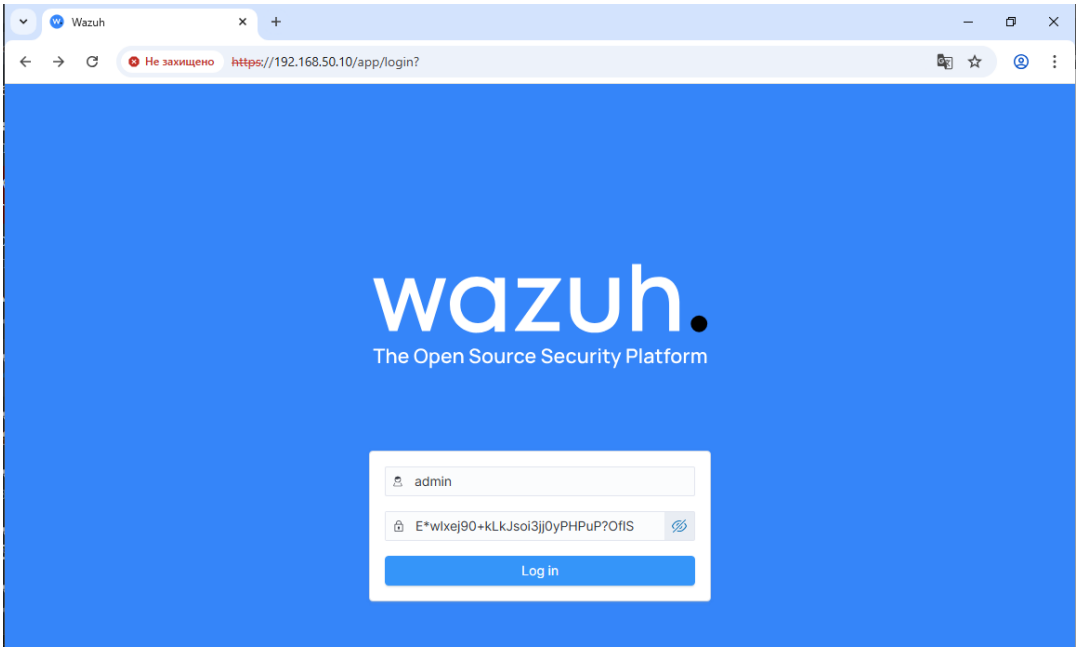


Рисунок 3.28 – Вхід до веб-інтерфейсу

Для централізованого моніторингу подій безпеки до Wazuh треба підключити агенти на окремих вузлах. Перед встановленням агентів потрібно створити свої групи для пристроїв, адже за стандартом всі вони додаються до групи default. Для цього потрібно перейти у вкладку «Agents management» та вибрати «Groups». Після цього натиснути «Add new group» та задати назву.

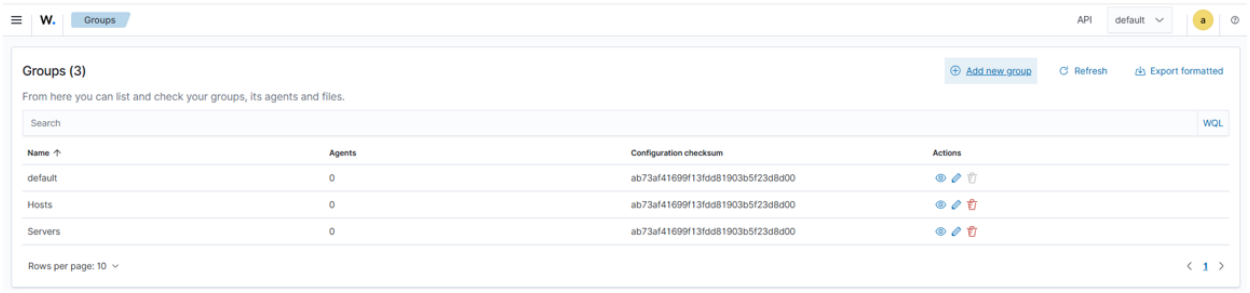


Рисунок 3.29 – Створення груп

Оскільки в підмережі присутні кінцеві вузли та сервери, то створимо дві групи: Hosts та Servers. В подальшому розширенні мережі до цих груп можна буде додавати нові пристрої, або створити нові групи для конкретних.

Для встановлення агентів потрібно перейти до «Deploy new agent». У відкритій вкладці потрібно вибрати тип ОС, адресу сервера Wazuh, назву агента та групу. Розглянемо на прикладі встановлення агента для кінцевого вузла з ОС Windows.

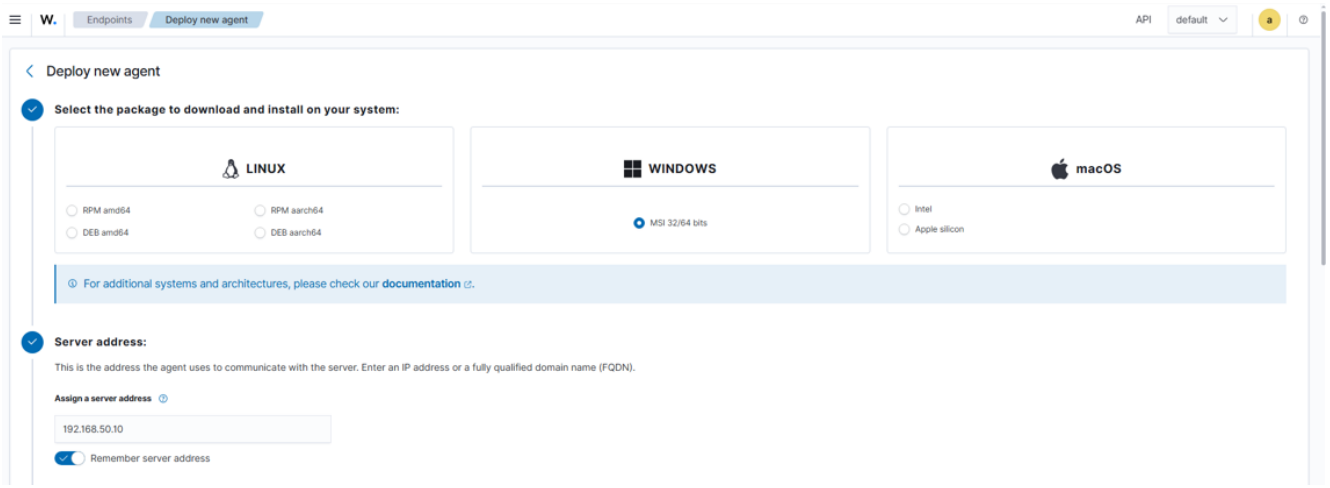


Рисунок 3.30 – Вибір ОС та IP-адреси сервера

Далі даємо агенту назву, з якою він буде відображатися у системі. Оскільки це кінцевий вузол, то обираємо групу «Hosts», після чого отримуємо посилання на завантаження агента через виконання скрипта в Power Shell.



Рисунок 3.31 – Вибір назви та групи

Тепер перевіряємо чи все правильно та копіюємо скрипт, після чого переходимо до ВМ «Windows10_Host», запускаємо Power Shell та виконуємо скрипт (рис 3.32).

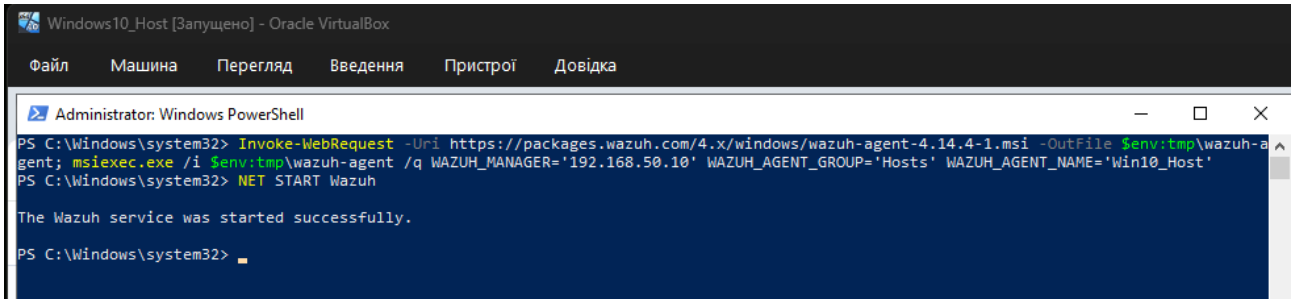


Рисунок 3.32 – Виконання інсталяційного скрипта

Встановлення агентів виконуються однаково на усіх типах ОС, єдина відмінність це вибір дистрибутиву для Linux – RPM amd64, DEB amd64, RMP aarch64, DEB aarch64 та для macOS – Intel або Apple silicon.

Після завершення інсталяції хост з'явиться у інформаційній панелі Wazuh.

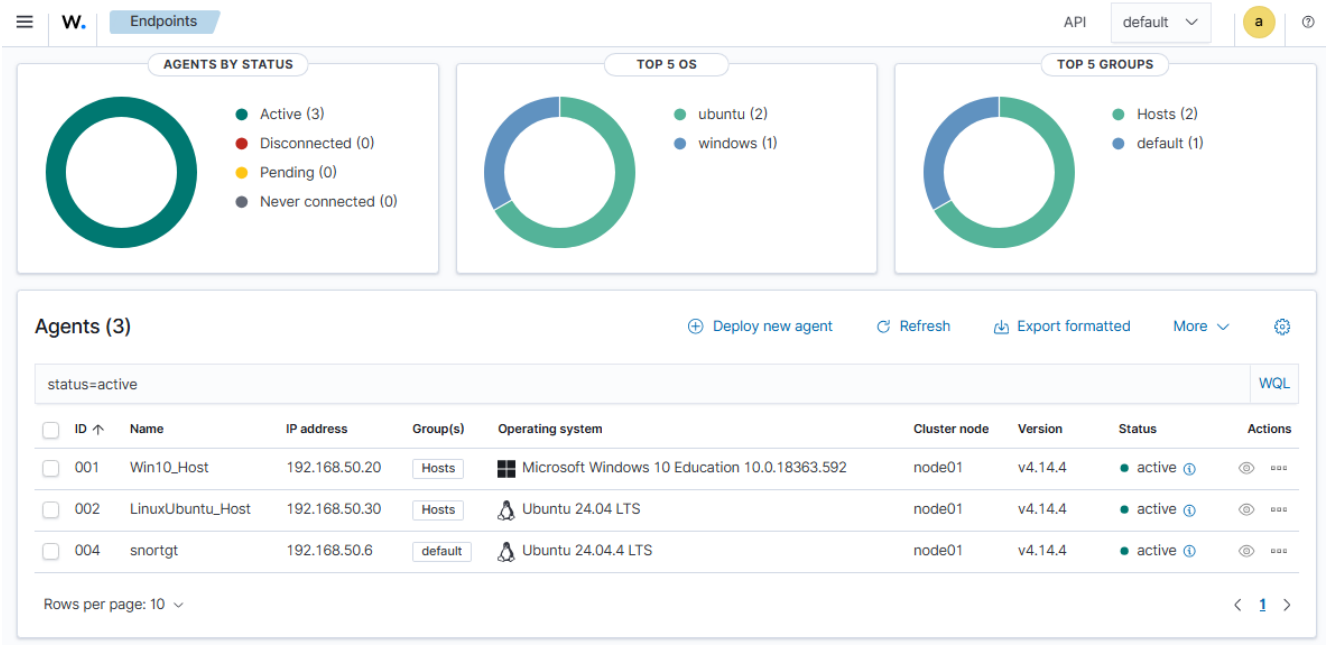


Рисунок 3.33 – Активні агенти

3.3 Налаштування IDS/IPS-системи Snort

Наступним етапом є розгортання та конфігурація IDS/IPS-системи Snort, яка використовується для аналізу мережевого трафіку та виявлення підозрілої активності.

Аналогічно перед встановленням необхідно виконати оновлення системних пакетів та залежностей. Після цього встановити компонент DAQ, який є технологією збору та перетворення даних для подальшого моніторингу та аналізу.

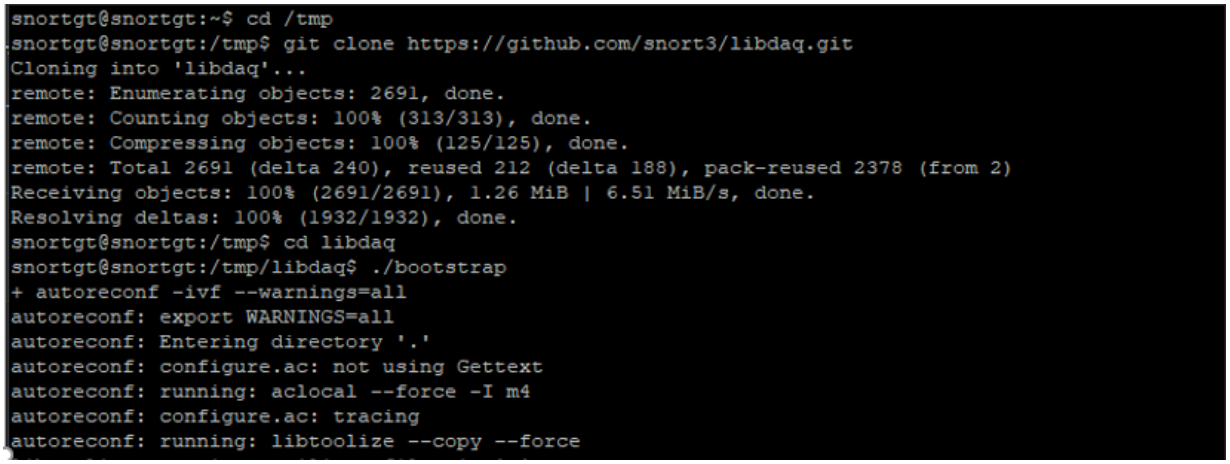


Рисунок 3.34 – Встановлення DAQ

Після підготовки середовища виконуємо встановлення Snort 3 із вихідних кодів.

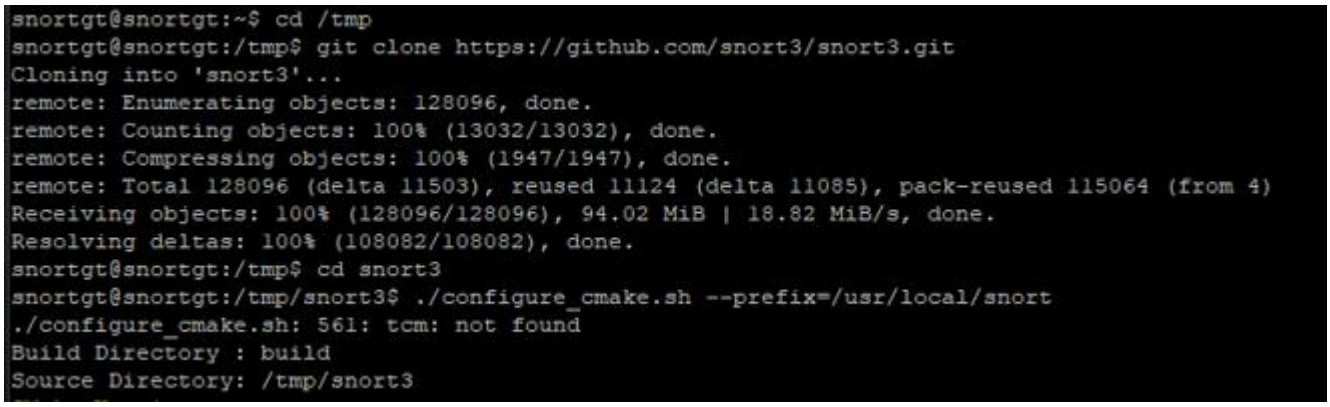


Рисунок 3.35 – Встановлення Snort 3

Встановлення завершено, тому перевіряємо працездатність системи за допомогою команди перевірки версії та тестового запуску Snort.



Рисунок 3.36 – Перевірка версії після встановлення

Після встановлення та перевірки версії можемо налаштувати та підготувати його до роботи. Починаючи з першої версії Snort 3 основні параметри IDS/IPS-системи знаходяться у конфігураційному файлі snort.lua. Для початку потрібно прописати мережу, яку має бачити та моніторити ПЗ.

```
  GNU nano 7.2                                /usr/local/snort/etc/snort.lua *
-- 7. configure outputs
-- 8. configure tweaks

--------------------------------------------------------------------------
-- 1. configure defaults
--------------------------------------------------------------------------

-- HOME_NET and EXTERNAL_NET must be set now
-- setup the network addresses you are protecting
HOME_NET = '192.168.50.0/24'

-- set up the external network addresses.
-- (leave as "any" in most situations)
```

Рисунок 3.37 – Налаштування підмережі у конфігураційному файлі snort.lua

Параметр «HOME_NET» використовується для визначення локального сегмента мережі, який в подальшому буде аналізуватися системою.

Одразу додамо в блок конфігурації параметр, який дозволить Snort використовувати локальні правила для моніторингу певних подій.

```
  GNU nano 7.2                                /usr/local/snort/etc/snort.lua *
--profiler = { }
--perf_monitor = { }

--------------------------------------------------------------------------
-- 5. configure detection
--------------------------------------------------------------------------

references = default_references
classifications = default_classifications

ips =
{

    rules = [[
        include /usr/local/snort/rules/local.rules
    ]],

    -- use this to enable decoder and inspector alerts
    --enable_builtin_rules = true,

    -- use include for rules files; be sure to set your path
    -- note that rules files can include other rules files
    -- (see also related path vars at the top of snort_defaults.lua)

    variables = default_variables
}

-- use these to configure additional rule actions
```

Рисунок 3.38 – Налаштування правил у конфігураційному файлі snort.lua

Для того щоб передавати отримані дані до Wazuh потрібно налаштувати вивід у спеціальний лог-файл. Тому переходимо до останнього блоку в конфігураційному файлі та налаштовуємо вивід подій у окремий файл.



Рисунок 3.39 – Налаштування виводу у конфігураційному файлі snort.lua

На даному етапі базове налаштування конфігураційного файлу – завершено. Тепер потрібно перевірити працездатність встановленої системи. Для цього переходимо до файлу з локальними правилами та створюємо одне з них. Наприклад, нехай це буде правило на визначення ICMP-запиту.

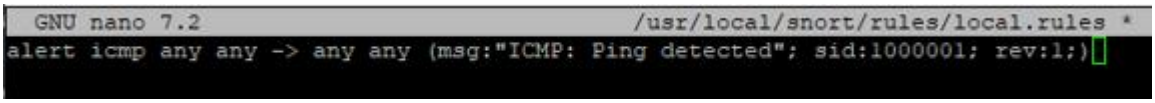


Рисунок 3.40 – Створення правила для перевірки

Після цього зберігаємо файл та запускаємо Snort командою з відповідними параметрами.

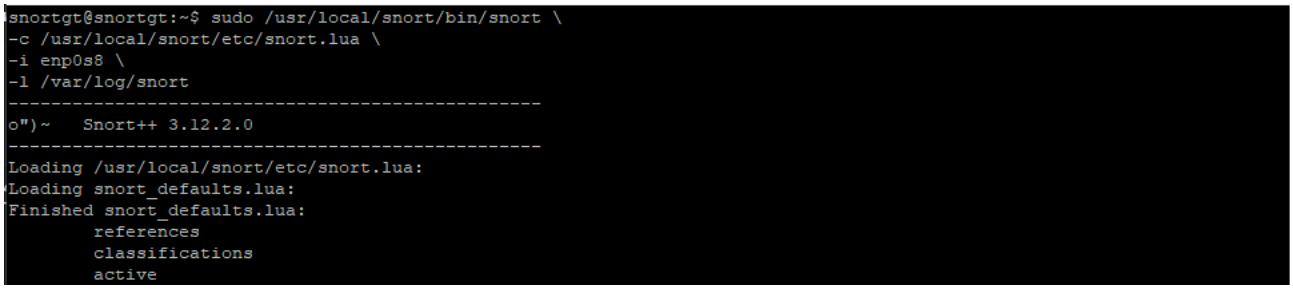


Рисунок 3.41 – Запуск Snort

На рисунку 3.41 можемо побачити, що запуск відбувається з вказанням шляху до основного конфігураційного файлу snort.lua, заданим мережевим інтерфейсом на якому він буде слухати трафік та визначеним каталогом для збереження логів.

Для перевірки створюємо ICMP-запит до одного з пристроїв мережі, після чого перевіряємо створення лог-файлу та його вміст. Якщо всі попередні налаштуваня проведено правильно, то у файлі має з’явитися детальеа інформація з цією подією.

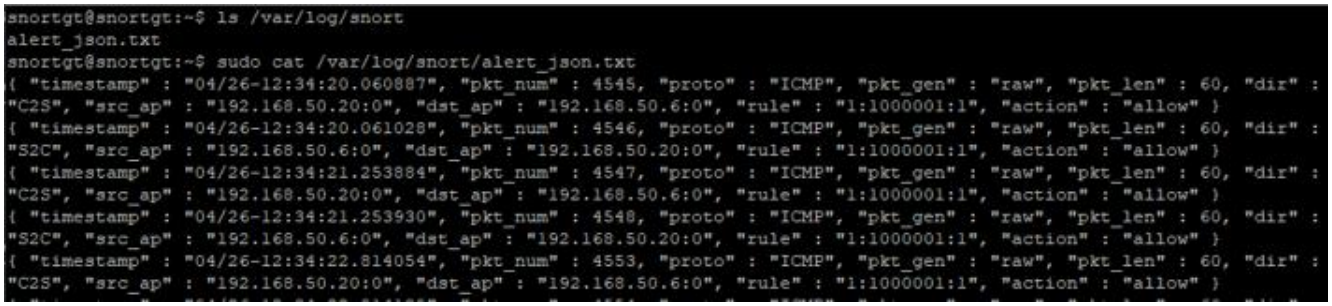


Рисунок 3.42 – Перевірка лог-файлу на вміст

На рисунку 3.42 можемо побачити, що у файлі присутні відомості ICMP-запит. Отже, система є працездатною, налаштована правильно та готова до подальшого використання.

3.4 Інтеграція Snort із Wazuh

Після завершення базових налаштувань обох систем можемо виконати їх інтеграцію. Основною метою інтеграції є централізоване отримання подій безпеки, реалізація механізмів кореляції мережевих подій та автоматизація реагування на потенційно небезпечну активність у мережі. Частина налаштувань виконувалась відповідно до офіційної документації Wazuh [36], проте значна кількість правил та механізмів реагування була адаптована під створене тестове середовище.

У межах реалізованої архітектури Snort виконує функцію первинного аналізу мережевого трафіку та формує повідомлення про події безпеки, тоді як Wazuh використовується для централізованого збору журналів, кореляції подій та автоматизованого реагування на атаки.

Агент на сервері з Snort був встановлений для читання та подальшої передачі подій з лог-файлу до сервера Wazuh. Для того щоб він міг читати цей файл у конфігураційному файлі ossec.conf потрібно прописати відповідний блок з захопленням цього файла агентом.

```
GNU nano 7.2                                    /var/ossec/etc/ossec.conf *
  <frequency>360</frequency>
</localfile>

<localfile>
  <log_format>full_command</log_format>
  <command>last -n 20</command>
  <frequency>360</frequency>
</localfile>

<localfile>
  <log_format>syslog</log_format>
  <location>/var/log/snort/alert_fast.txt</location>
</localfile>

<!-- Active response -->
<active-response>
```

Рисунок 3.43 – Налаштування блоку на читання лог-файлу

Після збереження налаштувань необхідно виконати перезапуск агента та надати йому права до лог-файлу.

```
snortgt@snortgt:~$ sudo nano /var/ossec/etc/ossec.conf
snortgt@snortgt:~$ sudo systemctl restart wazuh-agent
```

Рисунок 3.44 – Перезапуск агента на сервері Snort

```
snortgt@snortgt:~$ sudo chown root:wazuh /var/log/snort/alert_fast.txt
snortgt@snortgt:~$ sudo chmod 640 /var/log/snort/alert_fast.txt
snortgt@snortgt:~$
```

Рисунок 3.45 – Надання прав доступу до файлу

Тепер агент може читати файл та передавати дані до сервера Wazuh. Оскільки в минулому розділі було проаналізовано та визначено основні події для моніторингу, то відповідно до цього створюємо набір локальних правил з аналізом цих подій. Переходимо до файлу з локальними правила на сервері Snort та редагуємо його. На рисунках 3.46 та 3.47 продемонстровано налаштування правил.

```
 GNU nano 7.2                              /usr/local/snort/rules/local.rules *
#Звичайна активність
#ping (звичайний)
alert icmp any any -> $HOME_NET any (
msg:"ICMP connection detected";
itype:8;
sid:1000001; rev:1;
)

#Підключення через SSH
alert tcp any any -> $HOME_NET 22 (
msg:"SSH connection detected";
flags:S;
sid:1000002; rev:1;
)
```

Рисунок 3.46 – Налаштування правил звичайної активності

```
 GNU nano 7.2                                        /usr/local/snort/rules/local.rules *

#Аномальна активність
#ICMP-сканування
alert icmp any any -> $HOME_NET any (
msg:"ICMP sweep suspected";
itype:8;
detection_filter:track by_src, count 10, seconds 5;
sid:1000010; rev:1;
)

#SYN port scan
alert tcp any any -> $HOME_NET any (
msg:"Possible SYN scan";
flags:S;
detection_filter:track by_src, count 20, seconds 5;
sid:1000011; rev:1;
)

#SSH brute force
alert tcp any any -> $HOME_NET 22 (
msg:"SSH brute force suspected";
flags:S;
detection_filter:track by_src, count 10, seconds 10;
sid:1000012; rev:1;
)

#Часті запити DNS
alert udp any any -> any 53 \
(msg:"DNS high volume"; \
detection_filter:track by_src, count 40, seconds 60; \
sid:1000004; rev:1;)

#DBS-тунелювання
alert udp any any -> any 53 \
(msg:"DNS tunneling suspected"; \
detection_filter:track by_src, count 50, seconds 60; \
sid:1000013; rev:1;)
```

Рисунок 3.47 – Налаштування правил аномальної активності

У цих правилах використовується стандартна структура, де спочатку задається дія, далі протокол, напрямок трафіку, а потім набір параметрів у дужках [37]. Всередині дужок визначаються повідомлення та умови спрацювання, прапорці TCP та фільтри частоти, як наприклад, для виявлення надмірних запитів чи атак типу brute force. Також кожне правило має унікальний ідентифікатор та номер ревізії, що дозволяє підтримувати їх в актуальному стані. Основною метою даного етапу є розділення звичайної мережевої активності та потенційно небезпечної поведінки.

Після налаштування локальних правил для Snort налаштовуємо правила для Wazuh, щоб система могла розуміти що саме передає агент та як його візуалізувати у дашборді. На сервері Wazuh переходимо у директорію SIEM-системи, відкриваємо файл local_rules.xml та прописуємо блоки

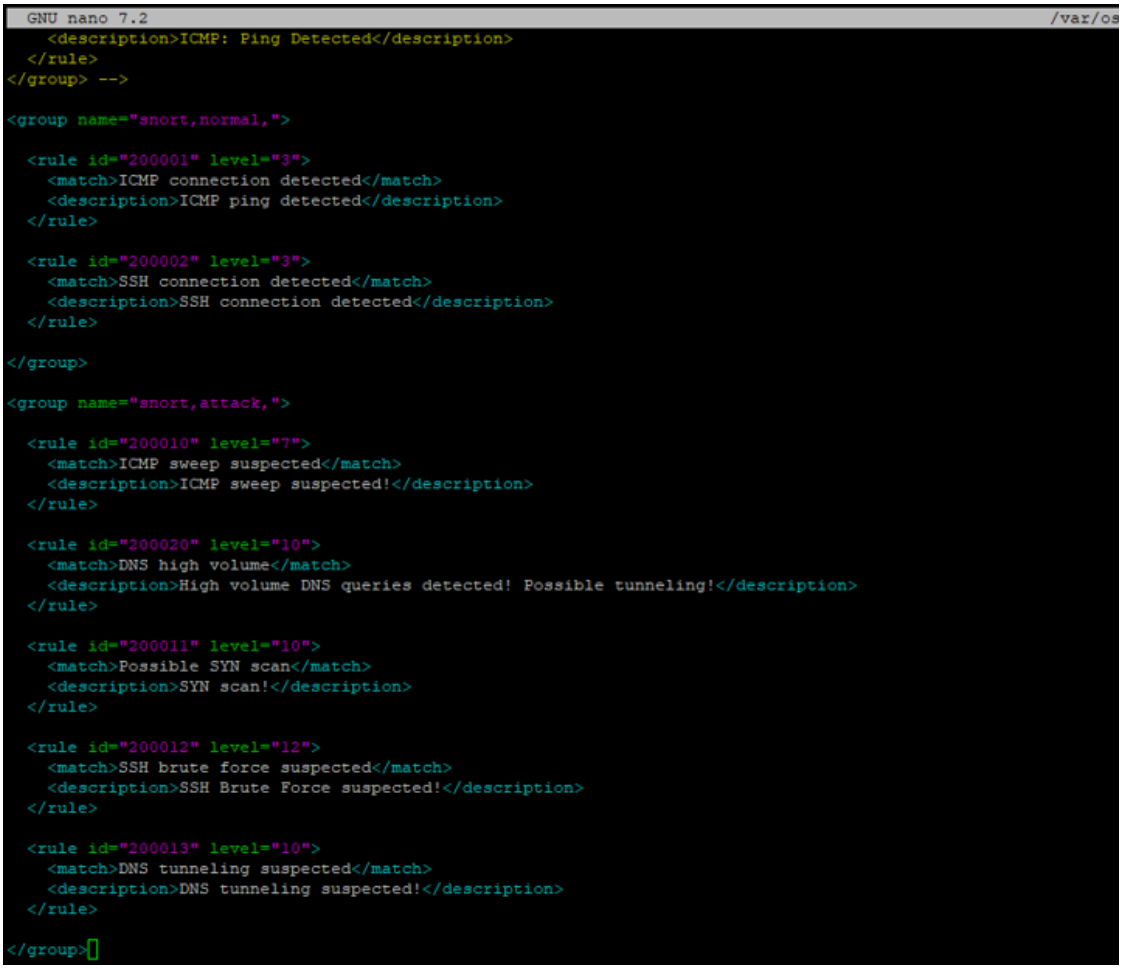


Рисунок 3.48 – Налаштування правил на сервері Wazuh

На рисунку 3.48 можемо побачити структуру, де кожне правило описує певну мережеву подію та рівень її важливості. У групі «snort, normal» визначено звичайну активність, тобто ICMP-запит та підключення за допогою протоколу SSH, які мають нищький рівень і фіксуються як звичайні з’єднання, а у групі «snort, attack» зібрано потенційно небезпечні події. Для кожного правила задано його унікальний ідентифікатор, рівень критичності та ключову фразу для збігу, а також короткий опис, який пояснює, що саме було виявлено. Саме цей опис буде відображатися у дашборді.

Однією з ключових можливостей SIEM-системи є кореляція подій безпеки. Потрібно реалізувати механізм об’єднання декількох пов’язаних подій у єдиний сценарій атаки. На відміну від Snort, який аналізує окремі пакети або короткі послідовності трафіку, Wazuh дозволяє оцінювати поведінку вузла протягом певного проміжку часу. У тому ж файлі на сервері створюємо групу та правила для кореляції подій.

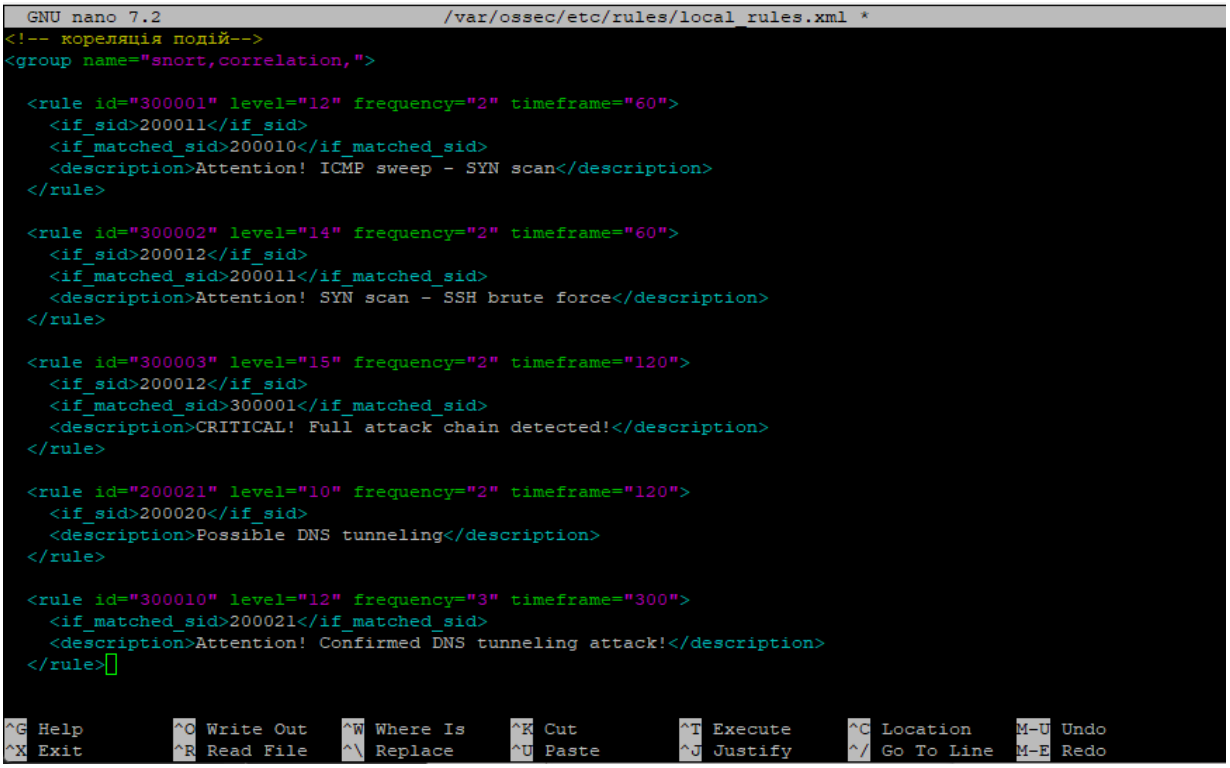


Рисунок 3.49 – Створення правил для кореляції подій

Логіка кореляції полягає в об’єднанні декількох окремих сповіщень у сценарії. Аналогічно до минулих правил кожне з них має свій ідентифікатор та рівень критичності, але тепер додається частота спрацювання та часовий інтервал. Ці параметри задають інтервали за який має статися атака. Тобто, якщо за вказаний проміжок стається декілька подій вказаних аномальних активностей, то це вважається атакою. Також задаються відповідні SID-номери подій, які мають співпасти, щоб правило активувалося. Окремо описано кореляцію DNS-тунелювання, де часті запити можуть перерости у підтверджену атаку.

Додатково у системі реалізуємо правило реагування на тестовий файл EICAR, який використовується для безпечної перевірки роботи антивірусних та SIEM-систем без застосування реального шкідливого ПЗ. У процесі налаштування до локальних правил Wazuh додається окреме правило з високим рівнем критичності, яке спрацьовує при виявленні сигнатури eicar у файлах або системних подіях. Додамо відповідний блок до правил у файлі.

```
<group name="local,syscheck,">

  <rule id="100500" level="12">
    <if_sid>550</if_sid>
    <match>eicar</match>
    <description>EICAR file detected!</description>
  </rule>

</group>
```

Рисунок 3.49 – Додавання правила на виявлення EICAR

Для прикладу розглянемо відображення цього файлу на вузлі Ubuntu Desktop. Для цього переходимо на нього та відкриваємо конфігураційний файл агента, змінивши там частоту оновлення FIM.

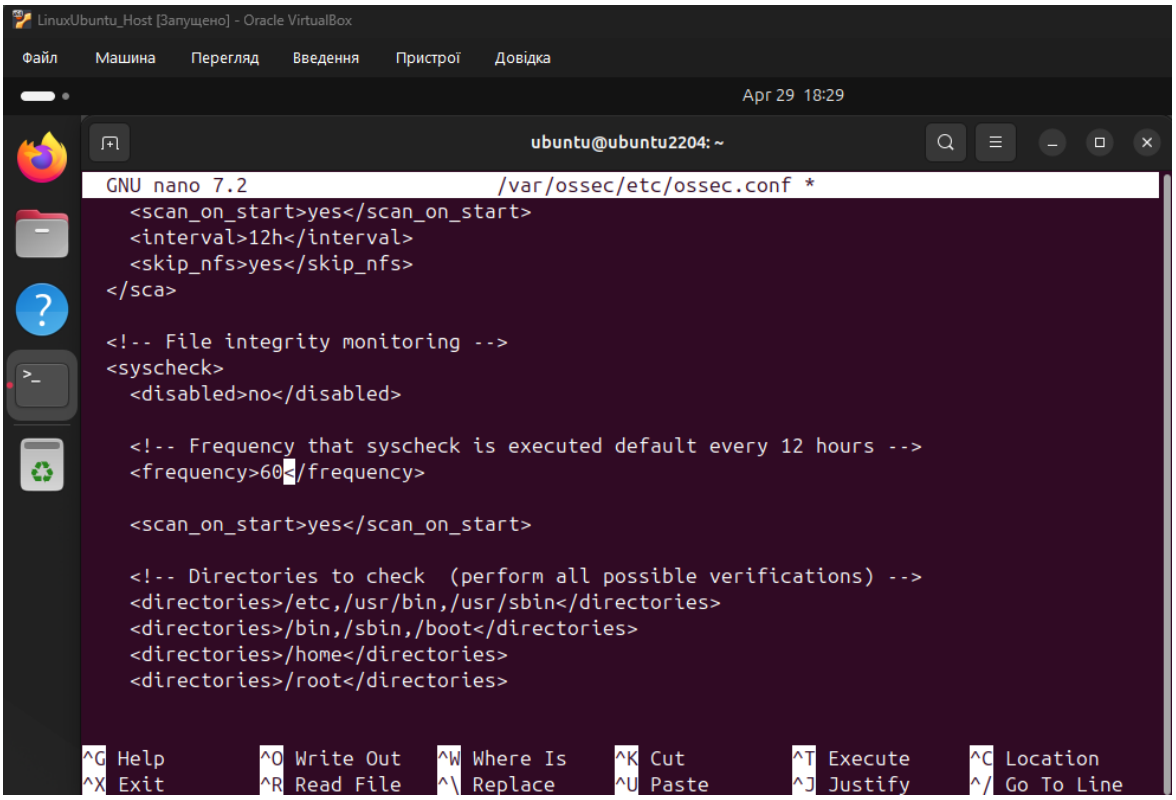


Рисунок 3.50 – Редагування конфігураційного файлу агента

За замовчування оновлення FIM відбувається кожні 12 годин, для перевірки це досить великий проміжок очікування, тому змінюємо його рівно на 1 хвилини.

Одразу створимо скрипт, який буде виконуватися у разі виявлення шкідливого файлу EICAR та автоматично видаляти його.

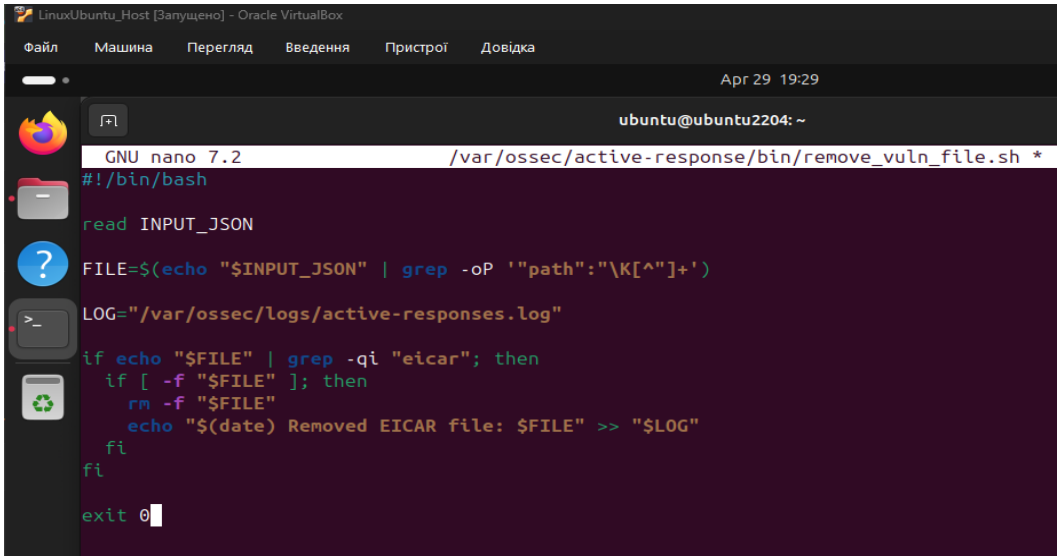


Рисунок 3.51 – Створення скрипта на видалення шкідливого файлу

Після цього переходимо на Wazuh-сервер та в конфігураційному файлі прописуємо команду на виконання скрипта та активне реагування у разі виявлення FIM конкретного файлу.

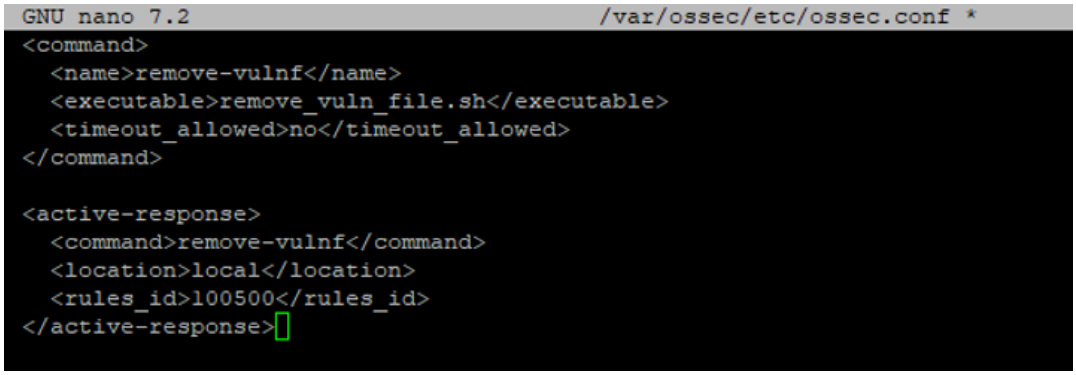


Рисунок 3.52 – Блок активного реагування та команда на видалення.

Таке рішення дозволяє перевірити коректність роботи механізмів моніторингу та журналювання у тестовому середовищі без ризику пошкодження системи. Після спрацювання правила подія автоматично передається до Wazuh Dashboard, де відображається як критичний інцидент безпеки.

Окремо можна виділи налаштування фіксацію DNS-звернення до доменів з загрозами, які є переліченими у Threat Intelligence Feeds. Це відомий список доменів, які можуть використовуватися для атак з подальшою компроментацією системи. Тому потрібно ізолювати систему від таких небажаних подій, оскільки ці звернення заслуговують окремої уваги від адміністратора системи.

Для початку на сервері створимо скрипт, який завантажить файл з цими доменами.



Рисунок 3.53 – Створення скрипта для завантаження файла

Для того щоб список залишався завжди актуальним та доповнювався необхідно додати автоматизацію до системної служби Cron, яка кожну годину буде виконувати скрипт і якщо додано нові домени, то вони будуть додаватися у спискок на сервері.

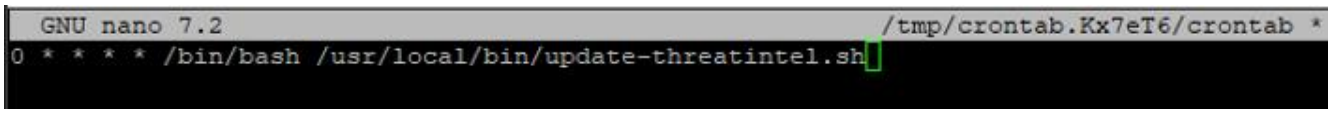


Рисунок 3.54 – Додавання автоматизації скрипта у Cron

Тепер перевіряємо працездатність автоматизації. Через деякий час скрипт мав спрацювати та додати файл з шкідливими доменами на сервер.



Рисунок 3.55 – Вивід шкідливих доменів

На рисунку 3.52 бачимо, що автоматизація дійсно працює, а на сервер доданий відповідний файл. Після цього перейдемо до правил Wazuh та додамо групу з відображенням події про шкідливі домени.

```
<group name="snort,dns,threatintel,">

  <rule id="300200" level="12">
    <match>DNS TI malicious domain</match>
    <description>DNS request to malicious domain from Threat Intelligence feed!</description>
  </rule>

</group>
```

Рисунок 3.56 – Додавання блоку з моніторингом шкідливих доменів

На сервері з Snort перейдемо до конфігураційного файлу snort.lua та додамо моніторинг відповідного каталогу і файлу.

```
 GNU nano 7.2                              /usr/local/snort/etc/snort.lua *

-- use these to capture perf data for analysis and tuning
--profiler = { }
--perf_monitor = { }

--------------------------------------------------------------------------
-- 5. configure detection
--------------------------------------------------------------------------

references = default_references
classifications = default_classifications

ips =
{

    rules = [[
        include /usr/local/snort/rules/local.rules
        include /usr/local/snort/rules/threatintel.rules
    ]],
```

Рисунок 3.57 – Додавання каталого до блоку

```
datasets =
{
  malicious_domains =
  {
    type = "string",
    load = "/usr/local/snort/etc/malicious_domains.txt"
  }
}
```

Рисунок 3.58 – Додавання файлу до блоку

Після цього можемо створити скрипт на генератор правила з переліком вказаних доменів у файлі.

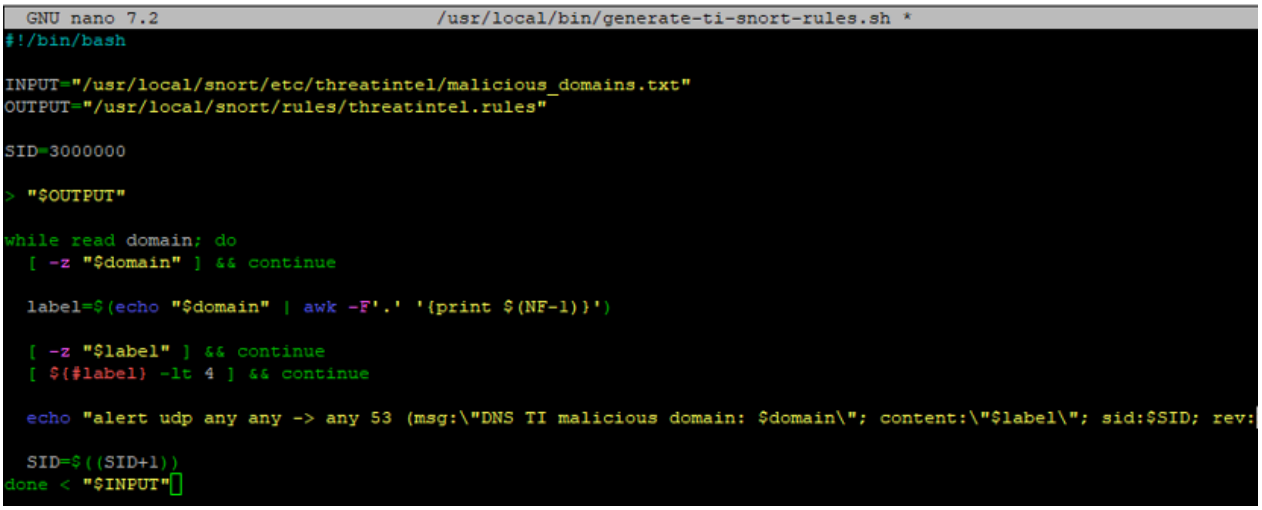


Рисунок 3.59 – Створення скрипта з генерацією правил

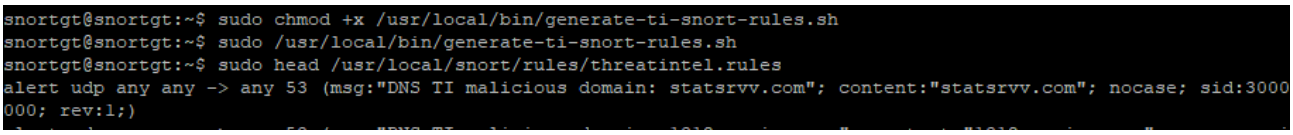


Рисунок 3.60 – Надання прав та перевірка працездатності генерації правил

Тепер Snort зможе детектити ці доменти, а агент передавати ці дані до SIEM-системи з подальшим відображенням алерта на дашборді.

Варто зауважити, що вся система також підтримує механізм активного реагування, який дозволяє автоматично виконувати дії у відповідь на події безпеки із визначеним рівнем критичності і є одним з рішень реалізації IPS в підмережі. На відміну від звичайного журналювання або формування повідомлень, активне реагування дає можливість не лише фіксувати атаку, а й одразу реагувати на неї без участі адміністратора. Реалізація такого механізму дозволяє зменшити час реагування на мережеві атаки та підвищує рівень автоматизації підсистеми безпекового моніторингу.

Тому на сервері з Snort створимо скрипт, який буде визначати IP-адресу з якої проводиться атака з події, які фіксує Snort. У разі виявлявлення такої адреси у скрипті буде виконуватися частина з реагуванням на атаку та виконанням функції відкидання пакетів та ізоляції атакуючої адреси на певний проміжок часу.

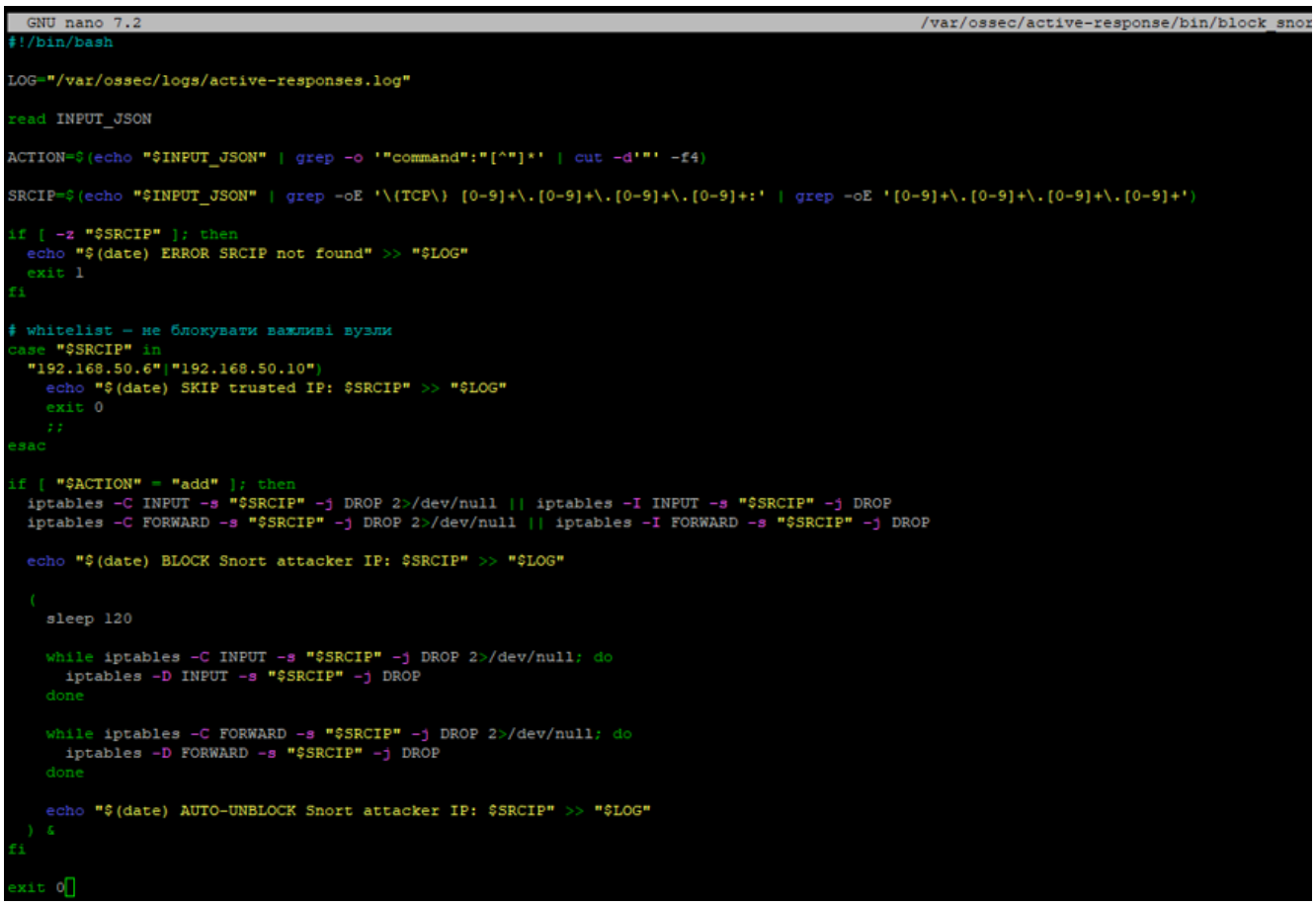


Рисунок 3.61 – Створення скрипта для блокування

В межах тестування проміжок ізоляції атакуючої машини буде в межах 2 хвилин. В реаліях таке блокування майже займати більше часу, оскільки аналітики повинні мати час на аналіз та подільші дії реагування на інцидент.

Після створення скрипта переходимо до сервера з Wazuh. Відкриваємо конфігурацій файл ossec.conf та до блоку «Active Response» додаємо власний параметри.

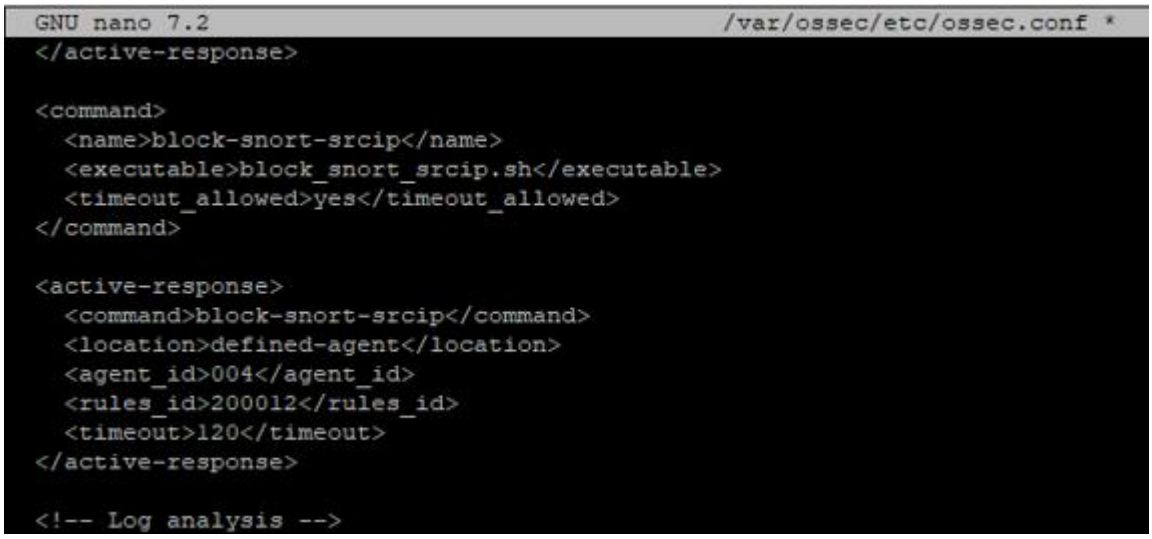


Рисунок 3.62 – Додавання блоку активного реагування на сервері Wazuh

Після всіх проведених налаштувань з інтеграції Wazuh та Snort, налаштування правил, подій, кореляції та механізмів активного реагування перезапускаємо служби агента та систем на обох серверах.

Обов'язково синхронізуємо час на сервері Snort, щоб події записувалися з правильним часом, адже у разі проведення атаки та діяльності після це є одним із найважливіших критерії аналізу.

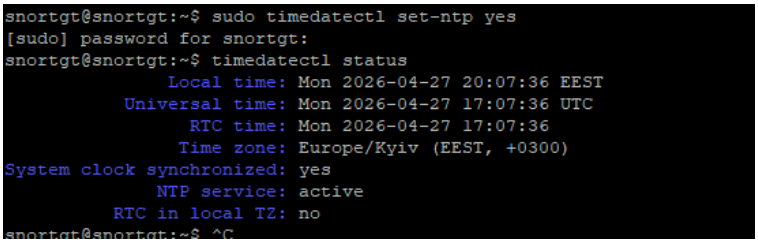


Рисунок 3.63 – Синхронізація часу на сервері

Додатково реалізуємо IPS на FreeBSD. Дистрибутив pfSense дозволяє проводити налаштування з трафіком, а оскільки він є маршрутизатором мережі, то налаштуємо правила брандмауера. Для прикладу зробимо логіку обміну ICMP-запитів між локальною та глобальною мережами. Щоб налаштувати правила через браузер переходимо до веб-інтерфейсу, далі відкриваємо вкладку «Firewall» та пункт «Rules».

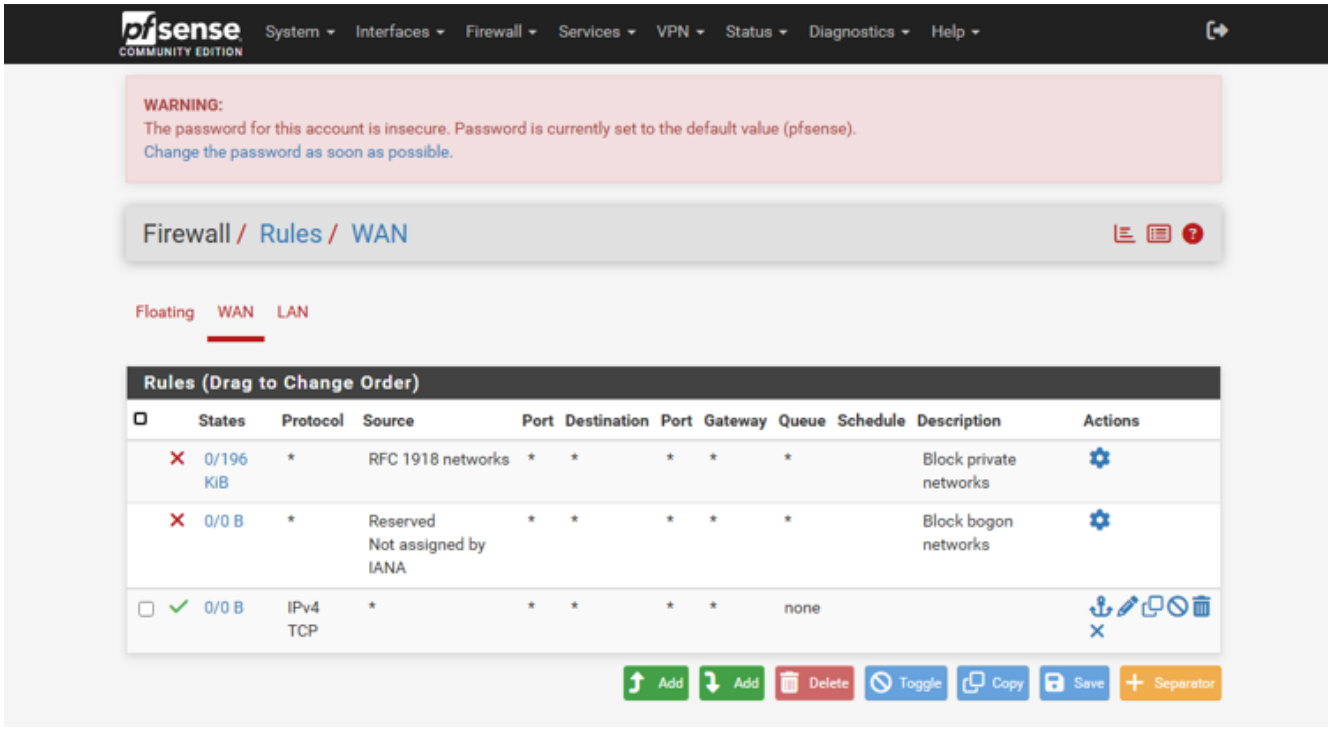


Рисунок 3.64 – Вкладка «Firewall» та пункт «Rules» у веб-інтефейсі pfSense

Натискаємо «Add» та створюємо правило для маршрутизації трафіку між мережами. У налаштуванні глобальної мережі вибираємо дію «block», відповідний інтерфейс, протокол IP-адресації та тип ICMP-запитів. Кінцевим призначенням вибираємо локальну мережу та надаємо опис правилу. Детальніше про налаштування правила блокування наведенео на рисунках 3.62 та 3.63.

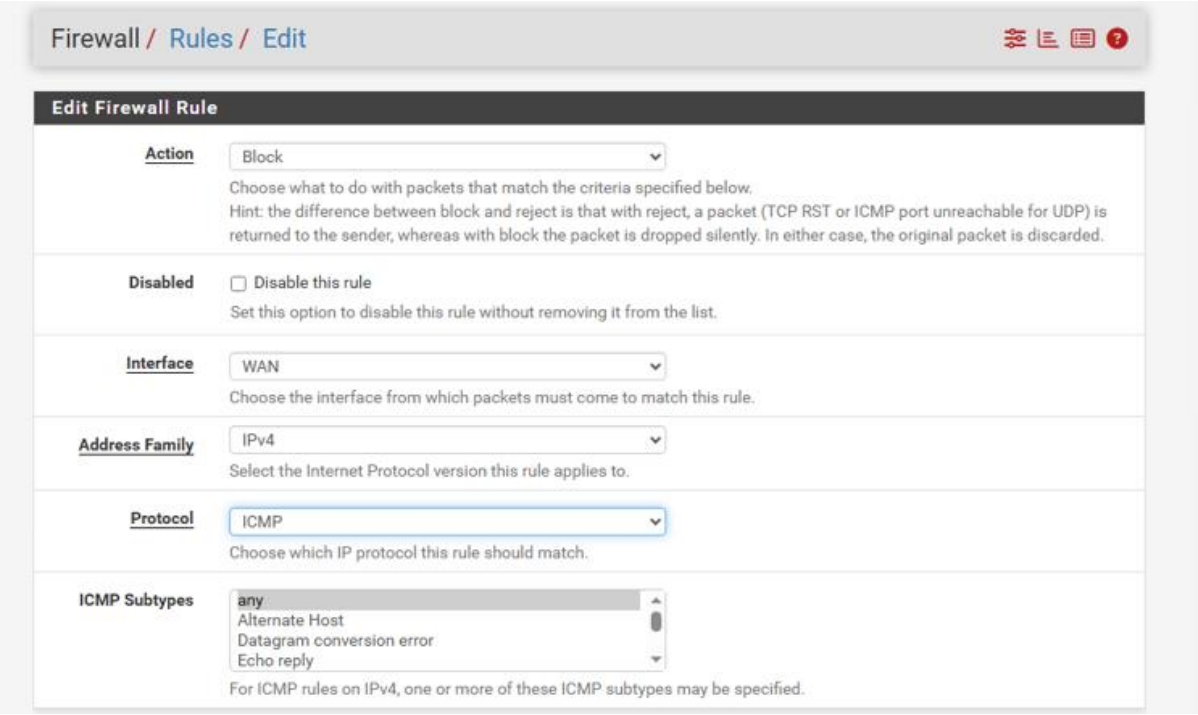


Рисунок 3.65 – Налаштування параметрів для правила брандмауера

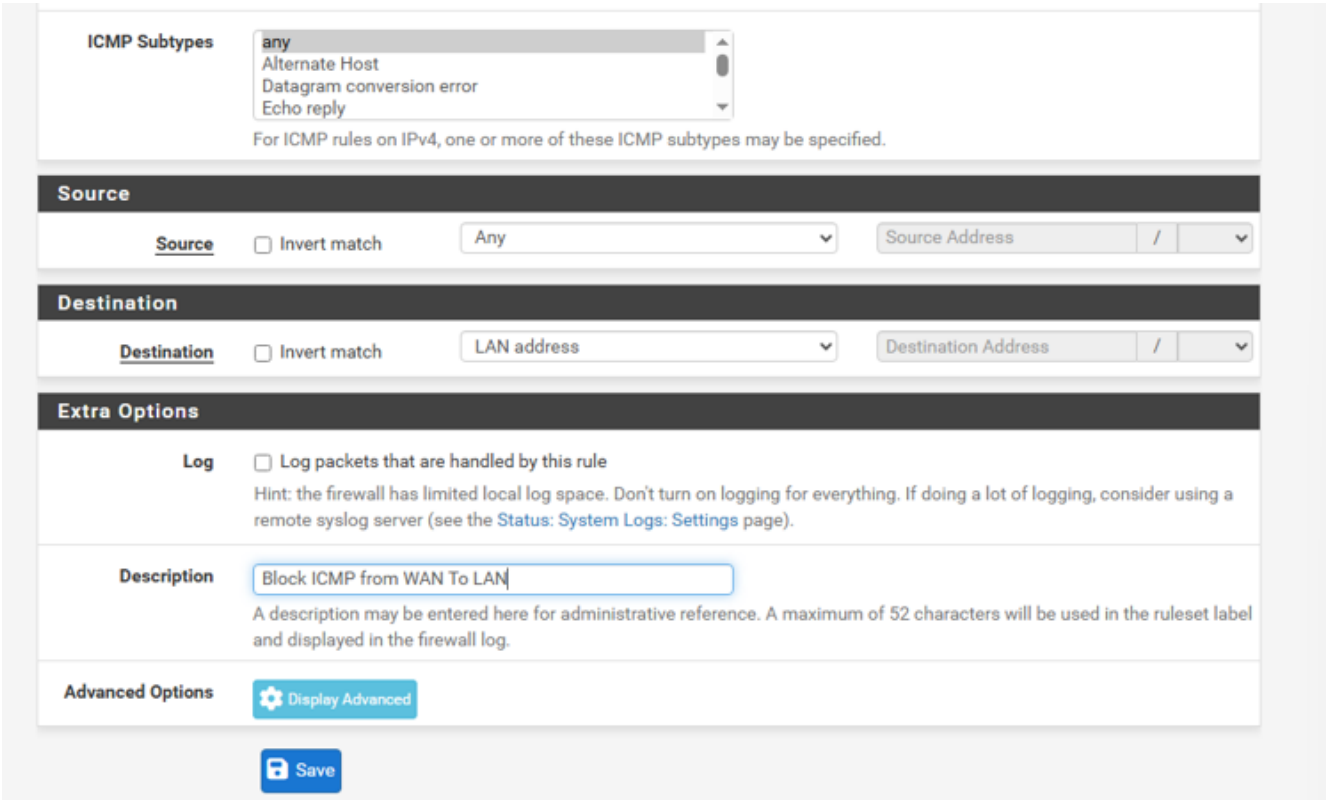


Рисунок 3.66 – Налаштування призначення та опису для правила

Після збереження правила застосовуємо створене правило та додаємо його до конфігурації

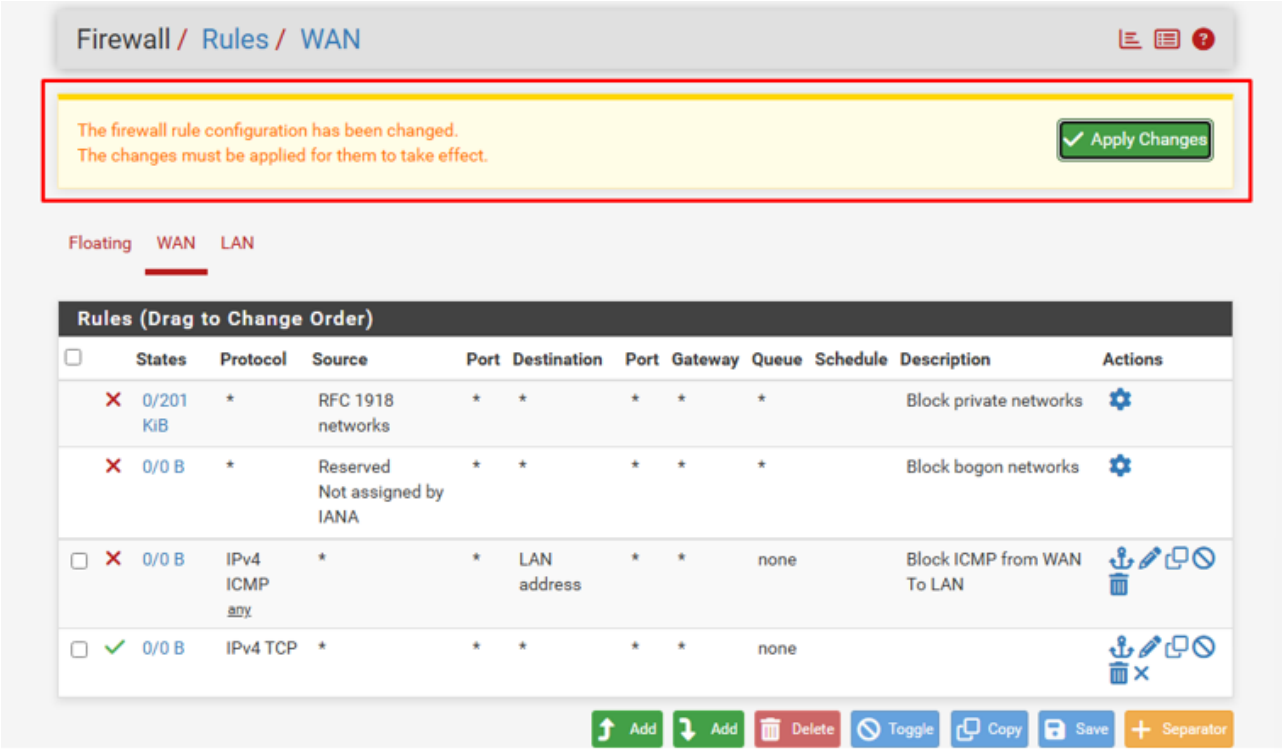


Рисунок 3.67 – Застосування правила

Після проведених дій всі запити протоколу ICMP з локальної мережі будуть проходити через BSD до глобальної мережі, а запити з глобальної мережі брандмауер буде відкидати. Таким чином запити будуть відкидатися на рівні маршрутизатора, навіть не дійшовши до локальної мережі, що підвищує безпеку внутрішньої мережі.

У межах даного етапу налаштування завершені. Інтеграцію обох систем проведено успішно, а налаштування журналів та відповідних механізмів безпеки мережі дають змогу перейти до наступного етапу тестування.

3.5 Тестування виявлення та блокування мережевих атак

Після завершення налаштування всіх компонентів підсистеми моніторингу можемо провести тестування усіх налагоджених механізмів. Для перевірки працездатності будемо використовувати Kali Linux як машину потенційного зловмисника.

Спочатку перевіримо звичайні активності в мережі. Проведемо ICMP-пінг з віртуальної машини до сервера з Wazuh.

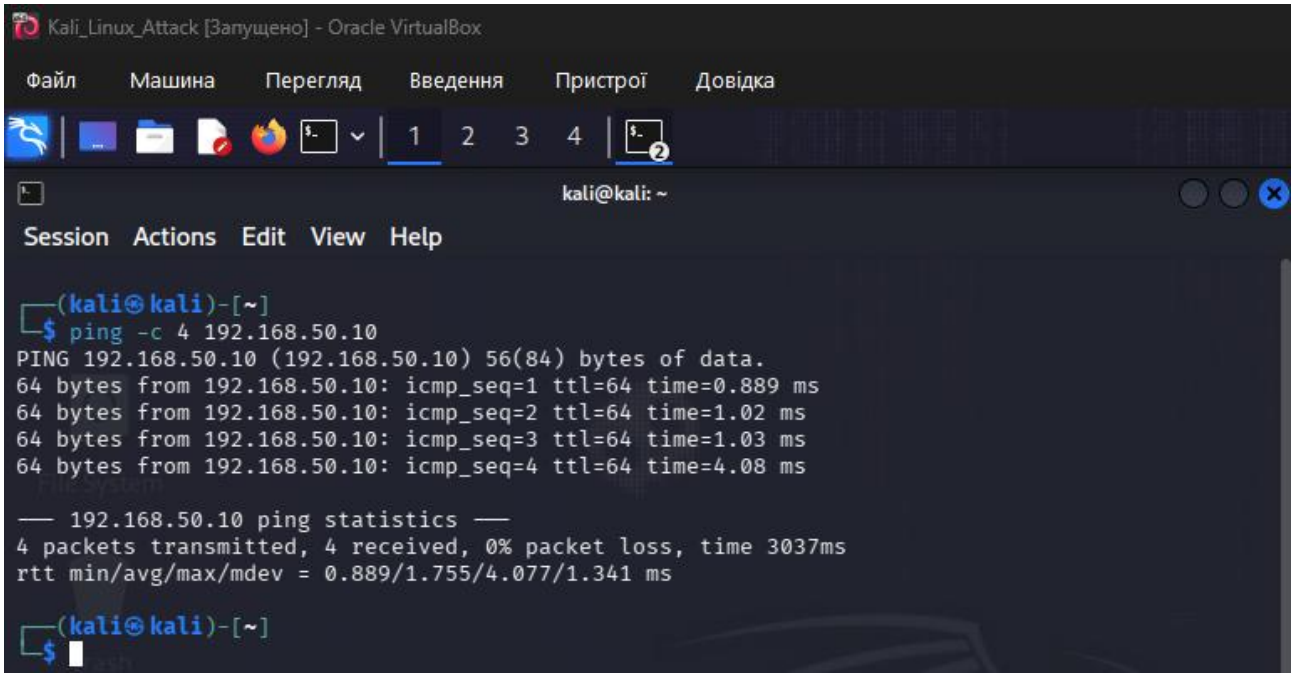


Рисунок 3.68 – ICMP-запит з Kali Linux до Wazuh-сервера

```
04/27-21:25:09.836803 [**] [1:1000001:1] "ICMP connection detected" [**] [Priority: 0] {ICMP} 192.168.50.40 -> 192.168.50.10
04/27-21:25:10.841237 [**] [1:1000001:1] "ICMP connection detected" [**] [Priority: 0] {ICMP} 192.168.50.40 -> 192.168.50.10
04/27-21:25:11.844374 [**] [1:1000001:1] "ICMP connection detected" [**] [Priority: 0] {ICMP} 192.168.50.40 -> 192.168.50.10
04/27-21:25:12.874478 [**] [1:1000001:1] "ICMP connection detected" [**] [Priority: 0] {ICMP} 192.168.50.40 -> 192.168.50.10
```

Рисунок 3.69 – Відображення події на Snort-сервері

Можемо побачити, що Snort зафіксував подію, тому переходимо до Wazuh та фіксуємо подію у дашборді.

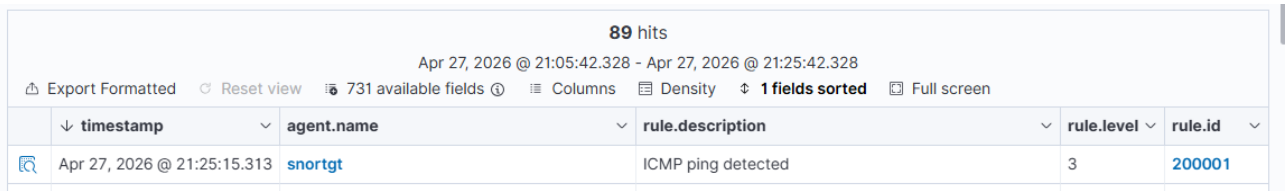


Рисунок 3.70 – Фіксація ICMP-запиту у Wazuh

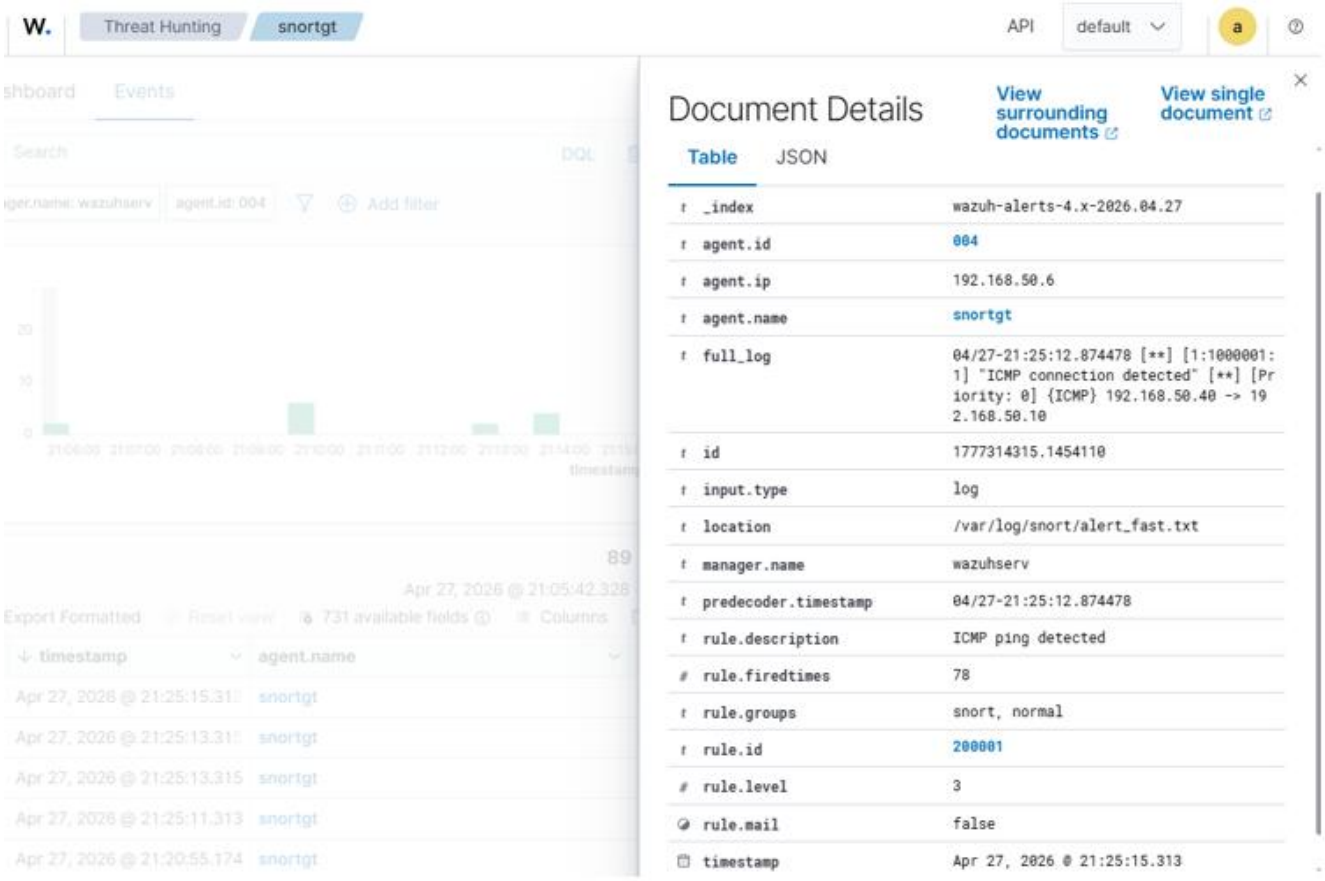


Рисунок 3.71 – Детальна інформація про подію ICMP-запиту

На рисунку 3.71 можемо побачити, що до Wazuh передалися всі відомості про подію. Є адреса агента, його ідентифікатор, повний опис події, її опис, номер правила, адреса призначення та час. Аналогічно відображається подія з підключенням за допомогою протоколу SSH.

Тепер перевіримо моніторинг аномальних активностей, адже це вже події вищого рівня. Перевіримо ICMP-сканування відправивши велику кількість ICMP-запитів до сервера.

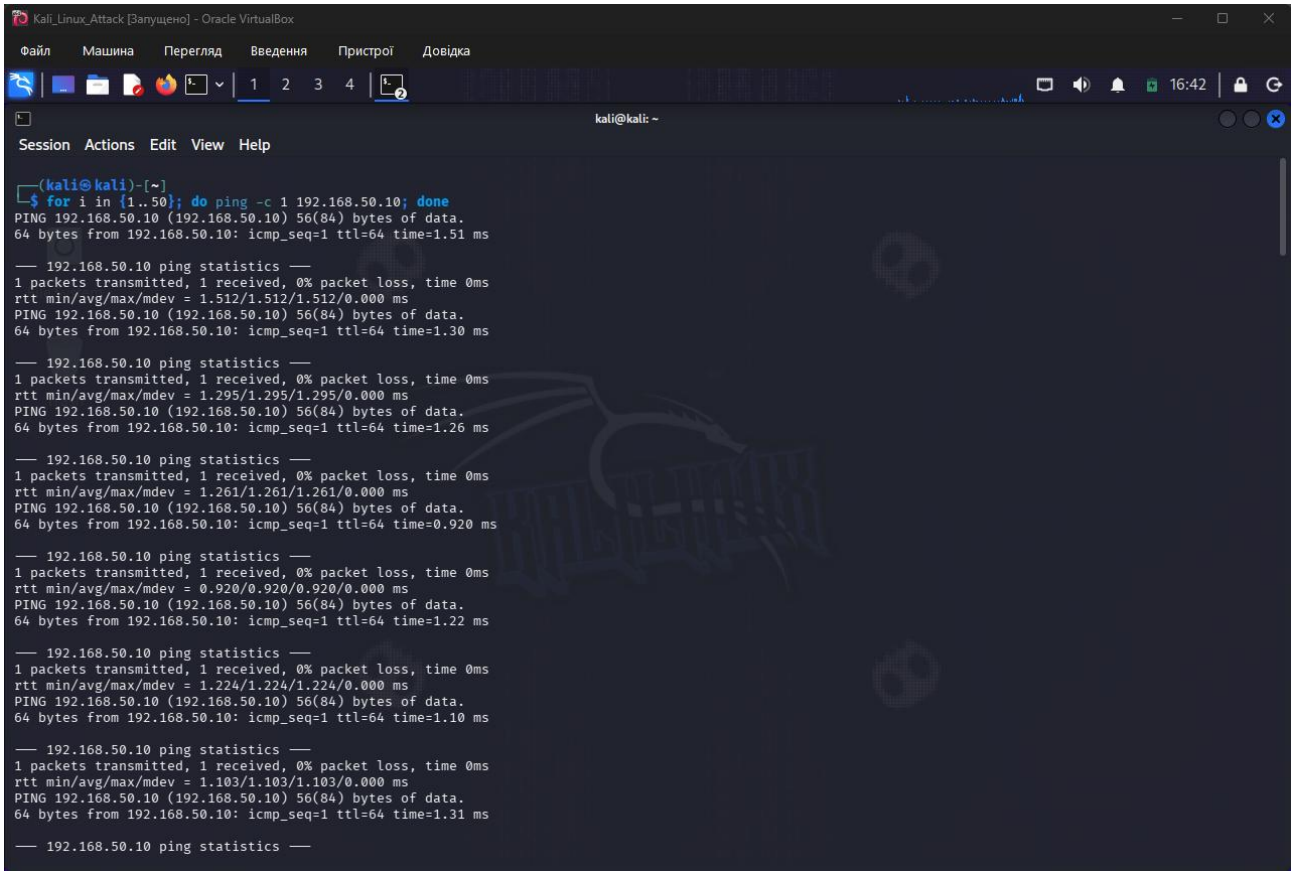


Рисунок 3.72 – Надсилання великої кількості запитів

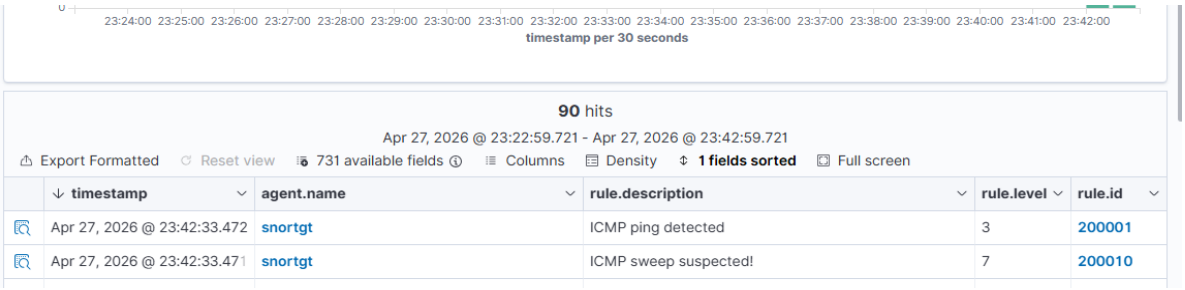


Рисунок 3.73 – Фіксація ICMP-сканування у Wazuh

Система зафіксувала велику кількість запитів та показала у дашборді відповідне правило, яке було налаштовано.

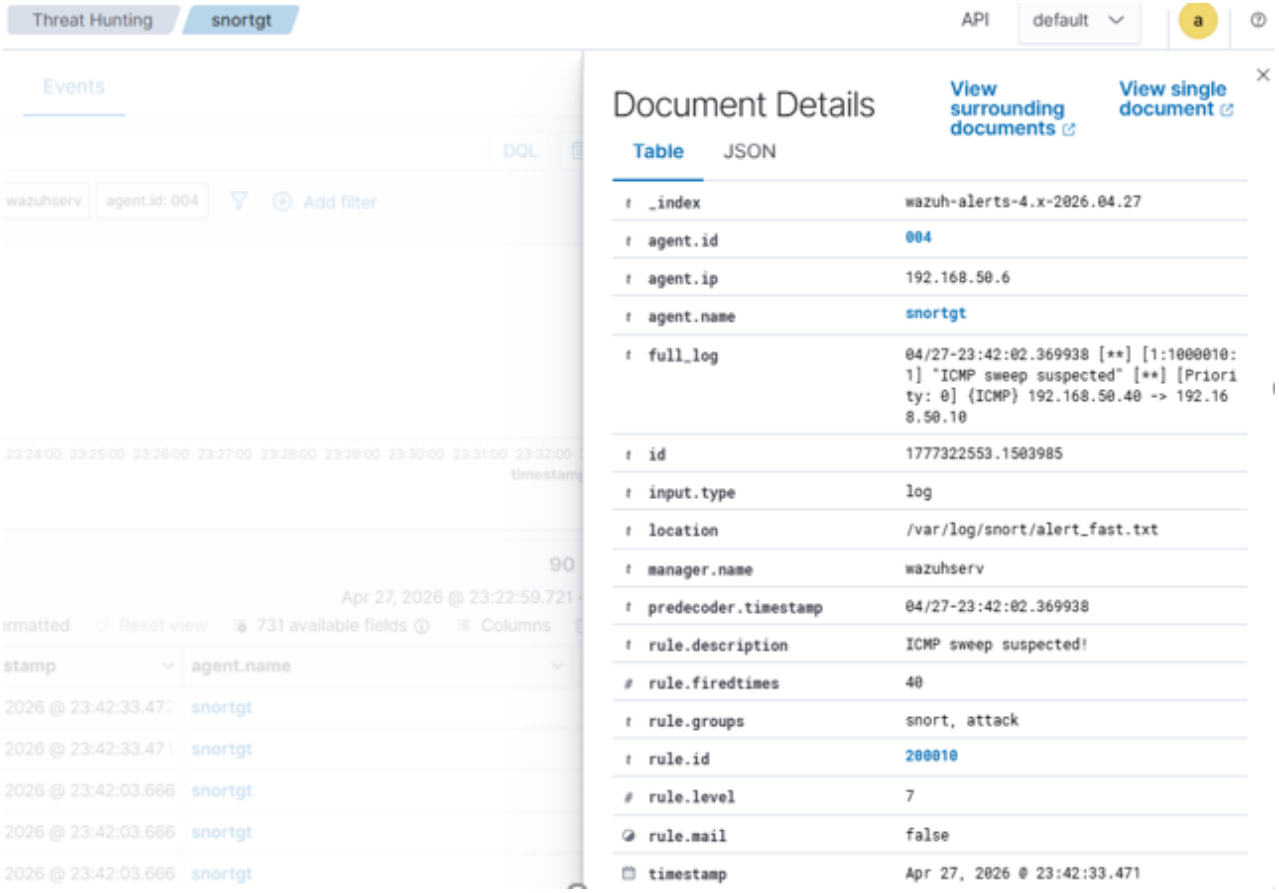


Рисунок 3.74 – Детальна інформація про ICMP-сканування

Детальна інформація показує аномалію та містить аналогічні відомості, як і з минулою подією.

Отже, базові правила моніторингу подій працюють. Тепер потрібно перевірити їх кореляцію. Розглянемо приклад кореляції подій із ICMP та SYN-сканування. Для цього треба виконати обидві події за певний проміжок часу, який вказаний у правилі кореляції. Якщо обидві події попадуть в один проміжок часу, то подія з’явиться у інтерфейсі. Для цього в терміналі виконаємо скрипт з генерації ICMP-запитів та використаємо інструмент Nmap, який виконає скаування портів

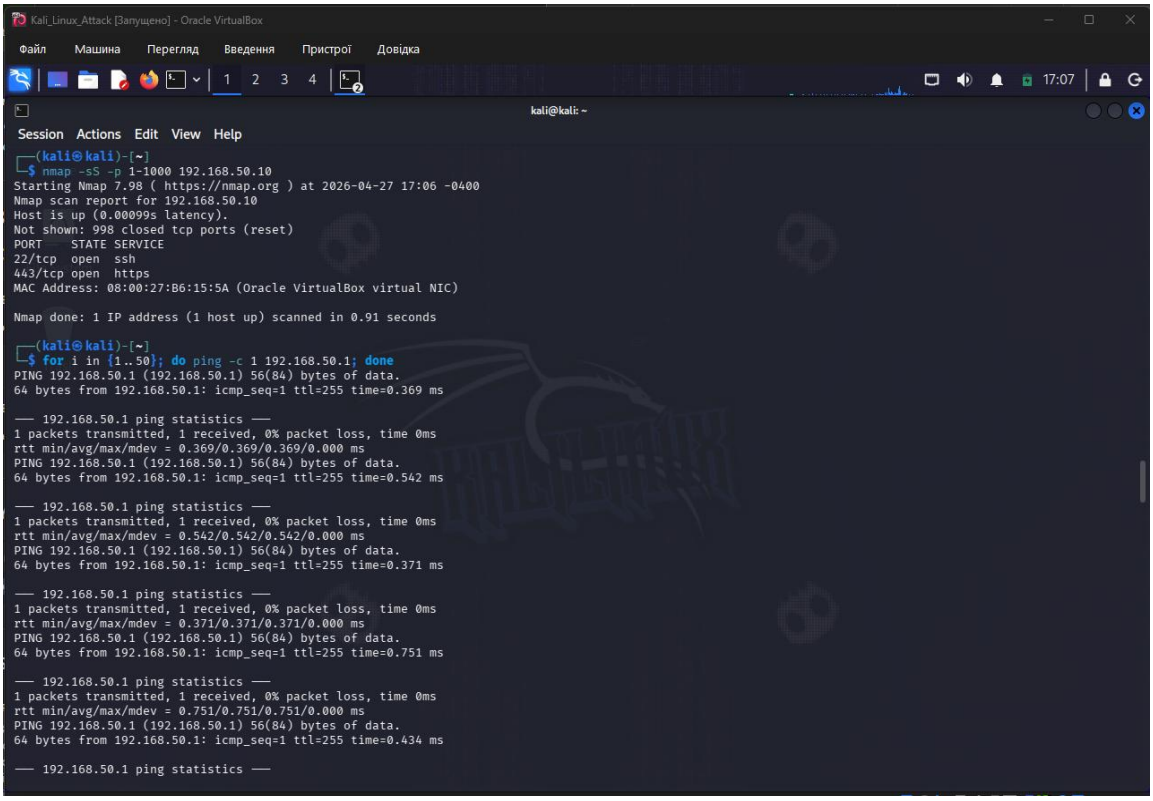


Рисунок 3.75 – Сканування портів та надсилання великої кількості запитів

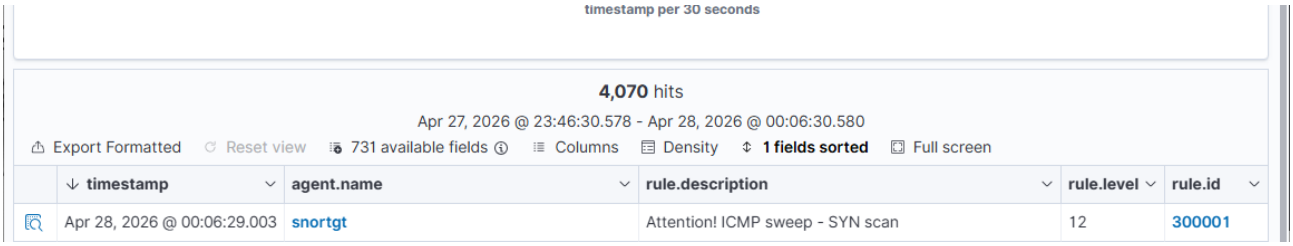


Рисунок 3.76 – Кореляція події у системі

На рисунку 3.76 можемо побачити, що кореляція події пройшла успішно і на дашборді з’явився відповідний алерт. Аналогічно працює кореляція SYN-сканування та атаки SSH brute force.

Також у кореляційні правила було додано «Full attack chain» – повний ланцюг атаки. Це правило застосовується якщо за певний проміжок стається 3 атаки підряд, що може повністю підтвердити кібератаку на певний вузол. В даному випадку цей проміжок складає 2 хвилини. Для перевірки запустимо ICMP-сканування, SSH brute force та сканування портів – разом.

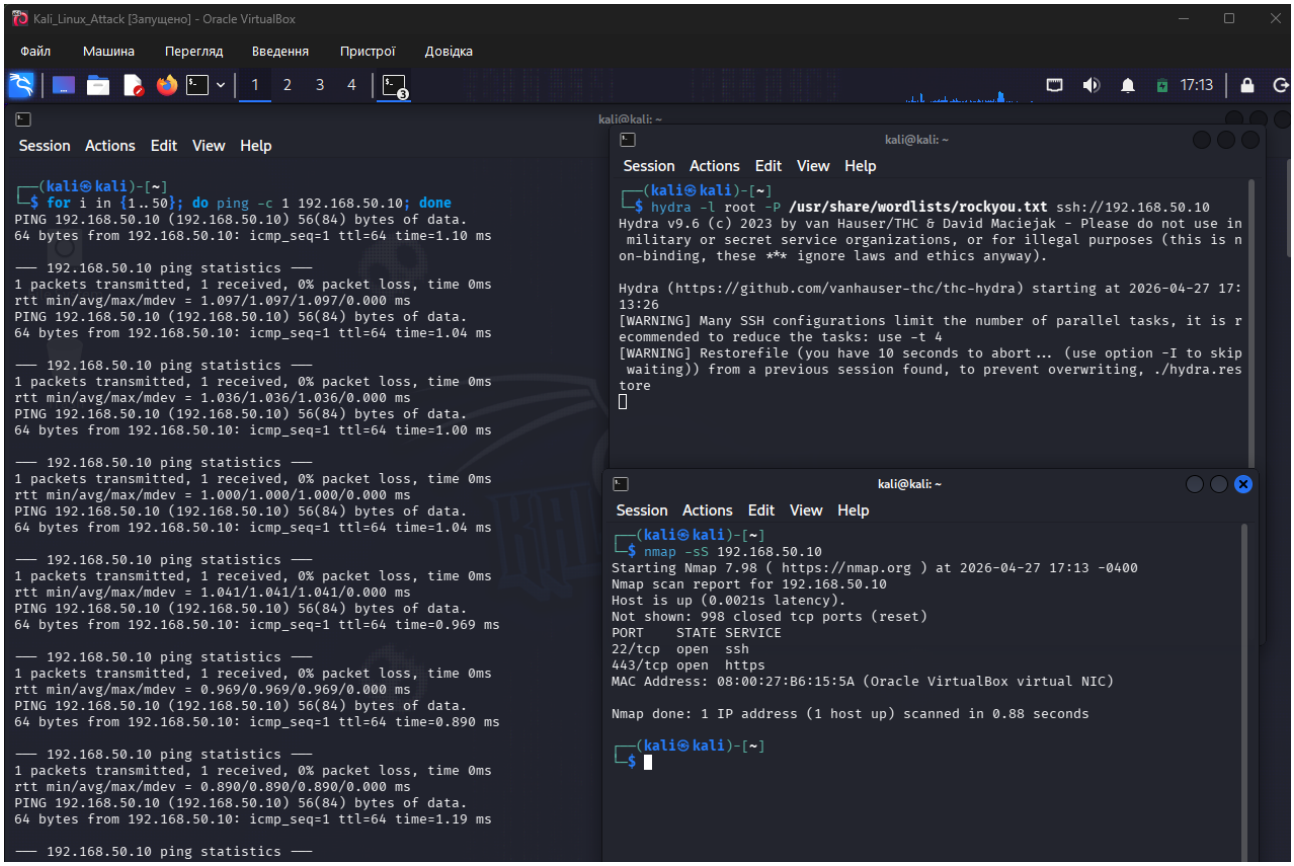


Рисунок 3.77 – Проведення атакуючих дій на Kali Linux

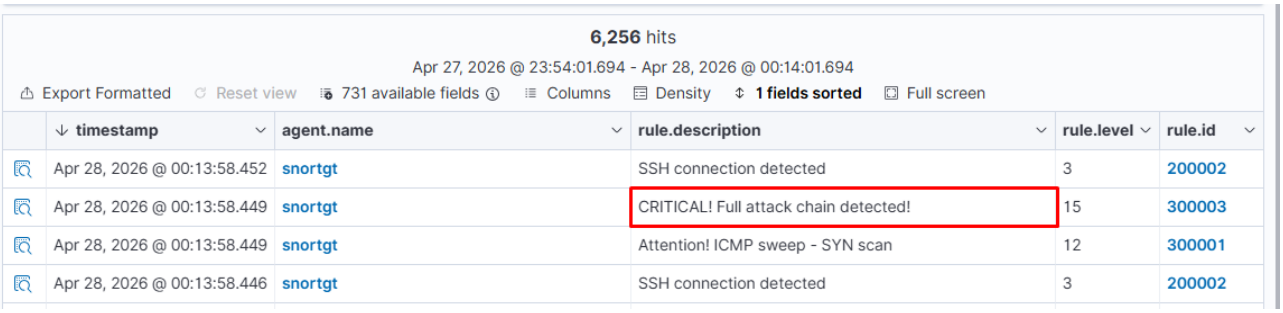


Рисунок 3.78 – Підтвердження атаки у Wazuh

Система починає отримувати безліч алертів про сканування, надсилання ICMP-запитів з подальшим сканування та атаку SSH brute force. На рисунку 3.78 можемо побачити, що все це корелюється і виводиться як повний ланцюг атаки. В результаті маємо повне підтвердження кібератаки на конкретний вузол з детальним описом та даними події.

Кореляція на підозру DNS-тунелювання відбувається, якщо за 2 хвилини прийде 2 повідомлення про велику кількість DNS-запитів. Система реагує на велику кількість DNS-запитів та проводить кореляцію надісланих подій, що в результаті створює алерт, який свідчить про можливе DNS-тунелювання.

Окрім можливого тунелювання існує правило, яке остаточно може його підтвердити. Дана кореляція спрацьовує якщо за 2 хвилини 2 рази підряд спрацює правило кореляції про можливе DNS-тунелювання. Випадково таку подію створити неможливо, що унеможливлює хибні спрацювання.

Отже, в терміналі виконаємо скрипт на багаторазове звернення до пошукової системи Google.

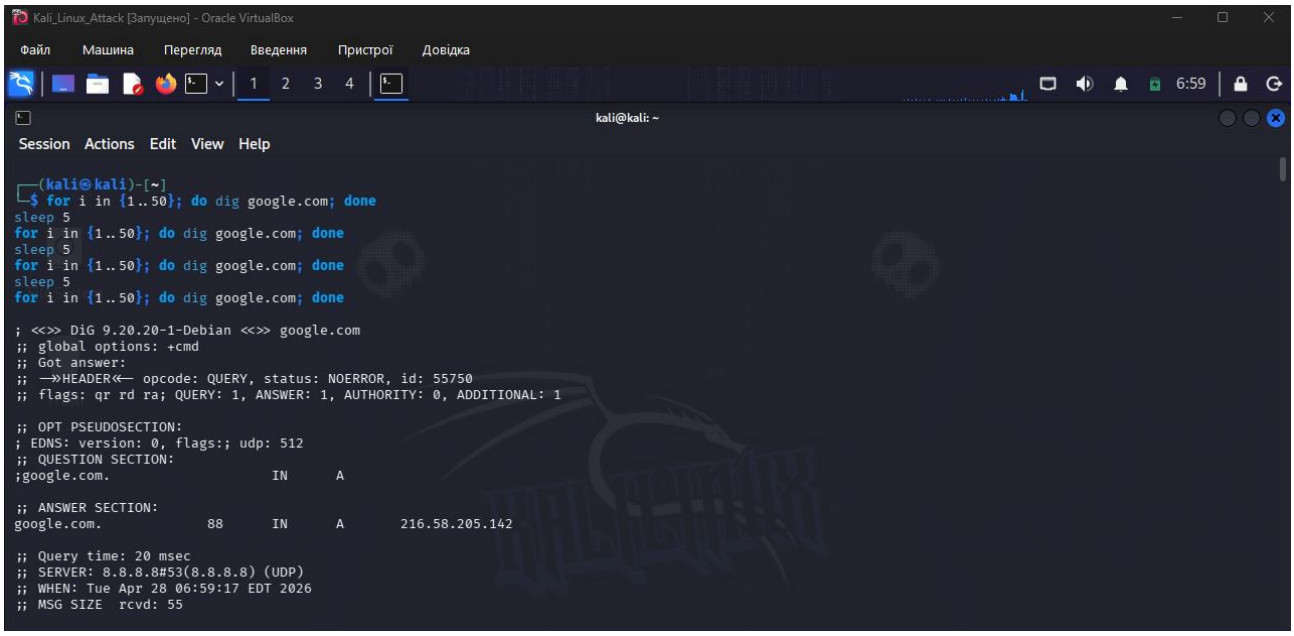


Рисунок 3.79 – Виконання скрипта на багаторазове звернення до Google

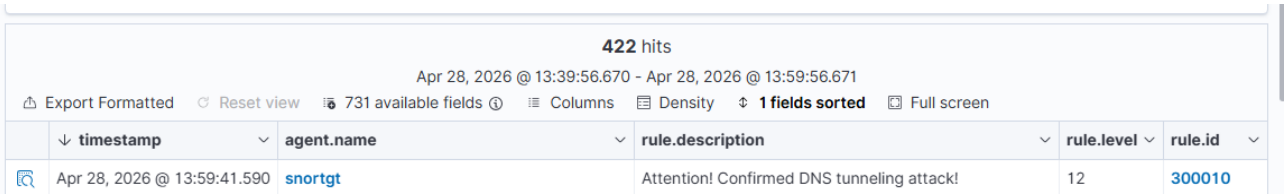


Рисунок 3.80 – Алерт про підтвердження DNS-тунелювання

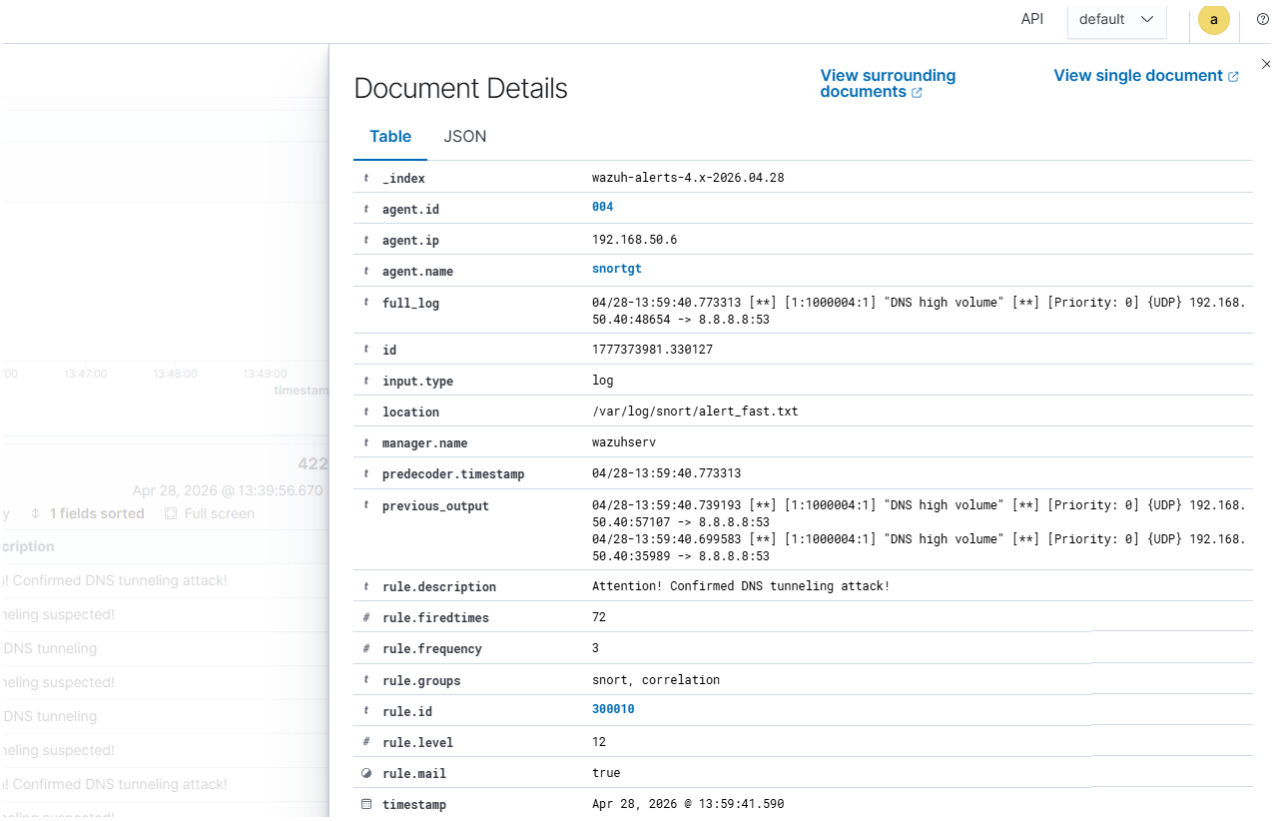


Рисунок 3.81 – Детальна інформація про подію DNS-тунелювання

Розглянувши детальну інформацію про атаку можна виділити пункт «previous_output», який детально показує причину спрацювання кореляції та відображення події на дашборді.

Перевіримо реалізацію активного реагування систем на проведення атаки. Для цього проведемо атаку типу SSH brute force на один із серверів. Але спочатку перевіримо таблицю маршрутизації на сервері, який є шлюзом для підмережі.

```
snortgt@snortgt:~$ sudo iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
snortgt@snortgt:~$
```

Рисунок 3.82 – Таблиця маршрутизації перед атакою

Можемо побачити чисту таблицю, яка означає повне функціонування маршрутизації в мережі. Тепер проведемо саму атаку використовуючи інструмент Hydra.

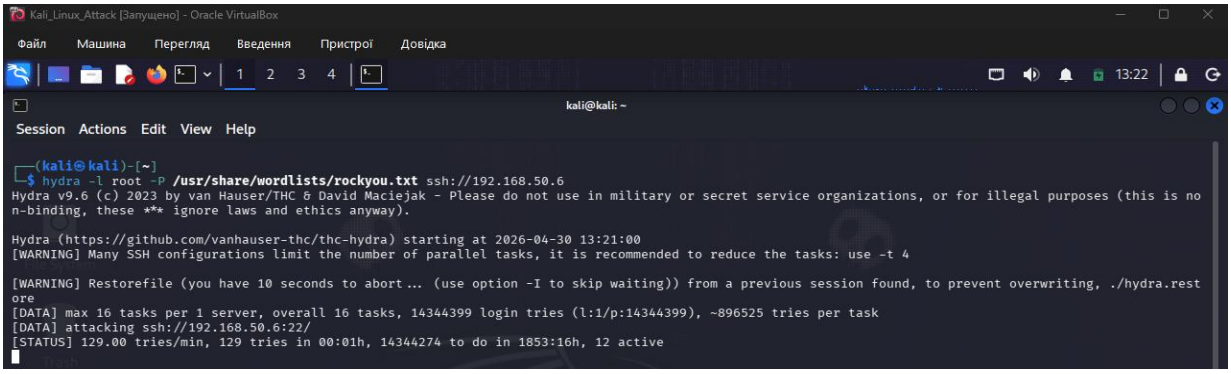


Рисунок 3.83 – Проведення атаки

```
sudo grep -E "BLOCK|UNBLOCK" /var/ossec/logs/active-responses.log | tail -n 20
 PM EEST 2026 BLOCK Snort attacker IP: 192.168.50.40
```

Рисунок 3.84 – Реагування IDS/IPS-системи на атаку

```
snortgt@snortgt:~$ sudo iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       0    --  192.168.50.40        0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
DROP       0    --  192.168.50.40        0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
snortgt@snortgt:~$
```

Рисунок 3.85 – Таблиця маршрутизаці після проведенян атаки

Система миттєво зреагувала на атаку та виконала скрипт, який виконав відкидання всіх пакетів з вказаної IP-адреси та ізолював вузол. В результаті пристрій заблокований в мережі на 2 хвилини. Проведемо повторну атаку та спробу ICMP-запиту для перевірки блокування.

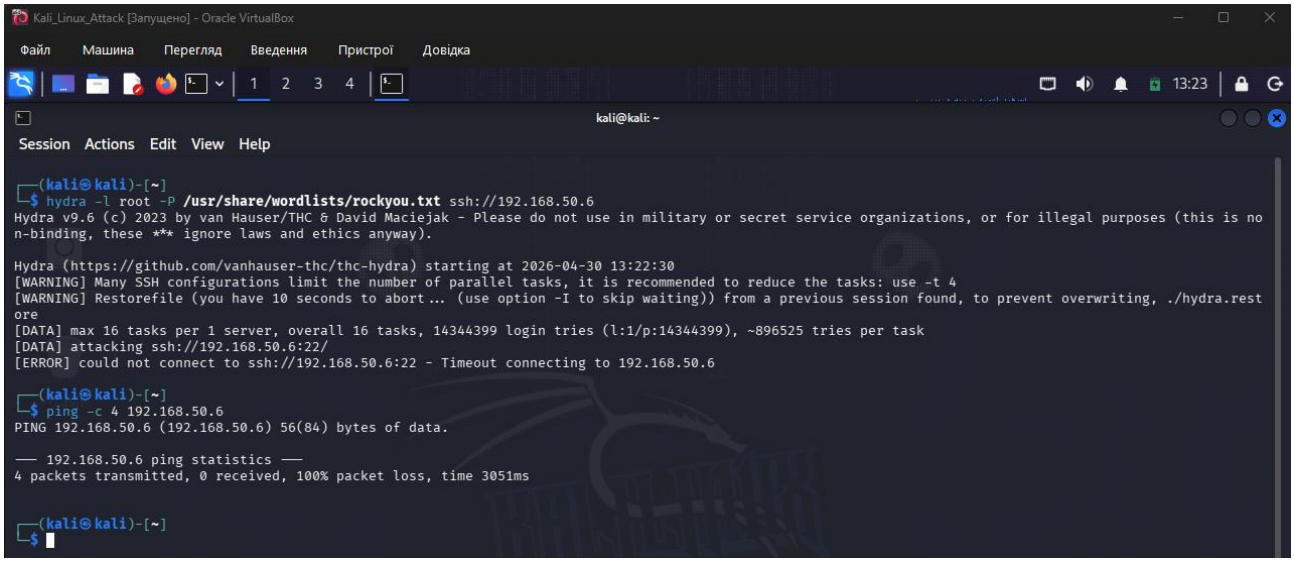


Рисунок 3.86 – Надсилання ICMP-запиту та повторна атака після блокування

На рисунку 3.93 можемо побачити, що атака виявилася невдалою, а пінг не проходить. Отже, блокування дійсно спрацювало і хост залишається ізольованим на 2 хвилини.

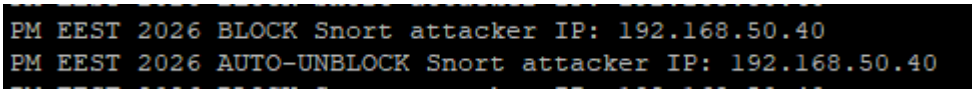


Рисунок 3.87 – Розблокування IP-адреси через вказаний проміжок часу

Через 2 хвилини система надіслала повідомлення про розблокування IP-адреси. Перевіримо доступність через ICMP-запит.



Рисунок 3.88 – Перевірка доступності вузла після розблокування

Розблокування виявилося успішним і вузол тепер має доступ до підмережі. Якщо буде проведена повторна атака, то він знову буде заблокований, що підтверджує працездатність активного реагування.

Перевірка реагування на загрозливі DNS-звернення відбувається зверненням до самих DNS-доменів з вузла. Для перевірки виконаємо команду nslookup на вузлі з Ubuntu Desktop, яка і перевірить доступність домену.

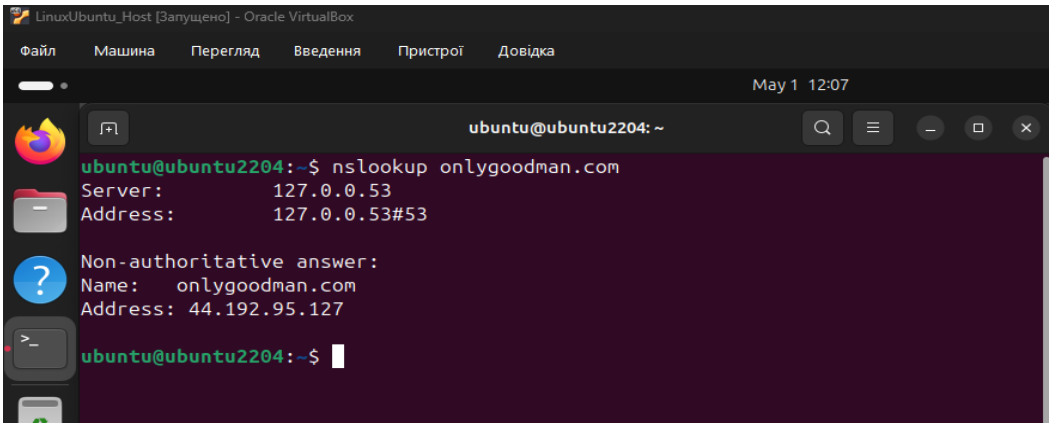


Рисунок 3.89 – Звернення до одного із загрозливих доменів

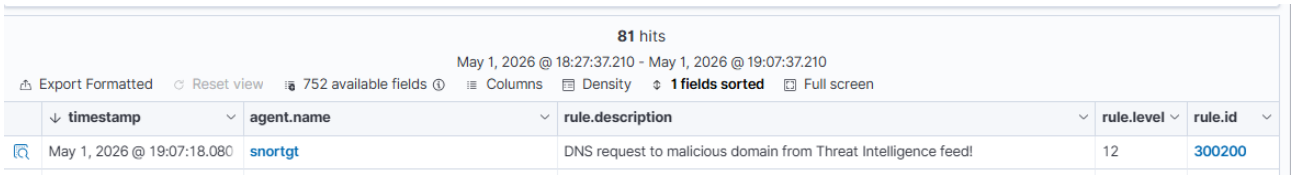


Рисунок 3.90 – Відображення події у Wazuh

Обидві системі відразу зафіксували та відреагували на подію показавши відповідний алерт на дашборді. Розгорнувши детальні відомості побачимо повний опис та домен до якого відбулося звернення.

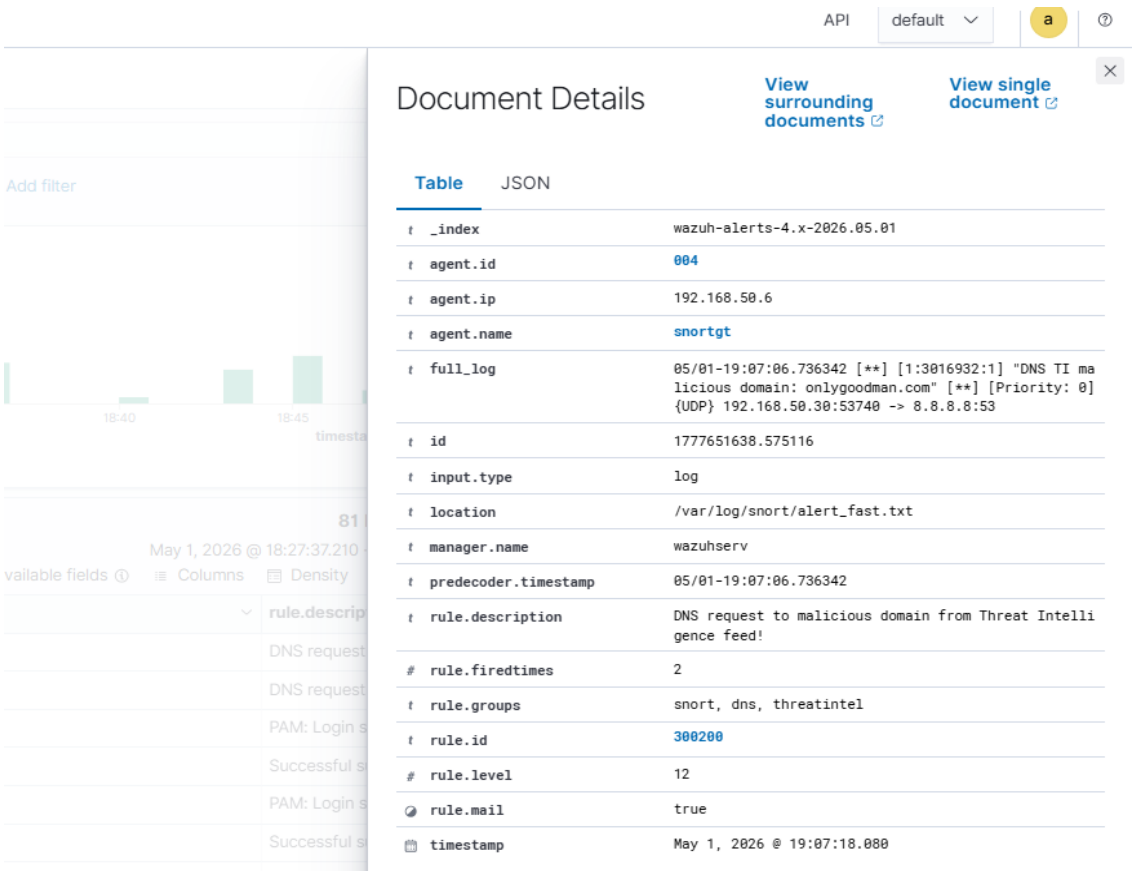


Рисунок 3.91 – Детальні відомості про звернення до загрозливого домену

Реагування системи на EICAR відбувається контролем файлів функцією FIM у Wazuh. Спочатку фіксується додавання файлу, а потім аналіз та висвітлення алерту про загрозу.

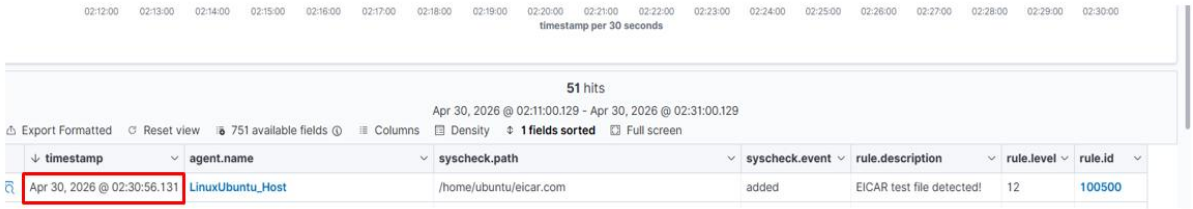


Рисунок 3.92 – Повідомлення про додавання шкідливого файлу EICAR

Видалення відбувається в автоматичному режимі у разі, якщо цей файл був дійсно доданий до вузла. Wazuh виконує механізм активного реагування та надсилає команду на видалення. Після успішного видалення файлу та перевірки агентом вмісту каталогів надходить алерт про видалення файлу.

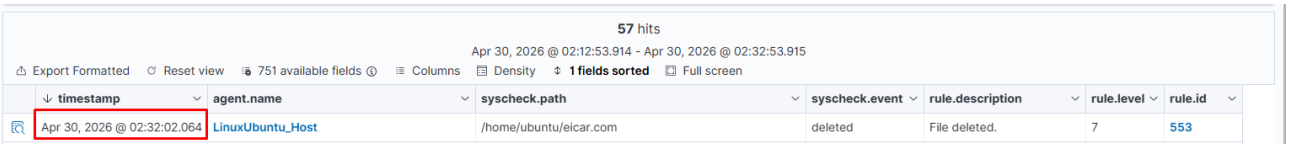


Рисунок 3.93 – Успішне видалення EICAR

Детальніше про перевірку реагування системи на загрози з повним тестуванням усіх механізмів та правил наведено у додатку Г.

Після проведення переревірки на реагування, доцільно створити відповідні графічні дашборди для зручності спостереження та монітрингу подій. Створимо перший дашборд, який буде показувати огляд всіх загроз підмережі на рівні SOC, який зафіксувала IDS/IPS-система. До нього додамо графік найпопулярніших атак, часового проміжку між атаками та таблицю розподілу ступеня серйозності сповіщень за весь час.

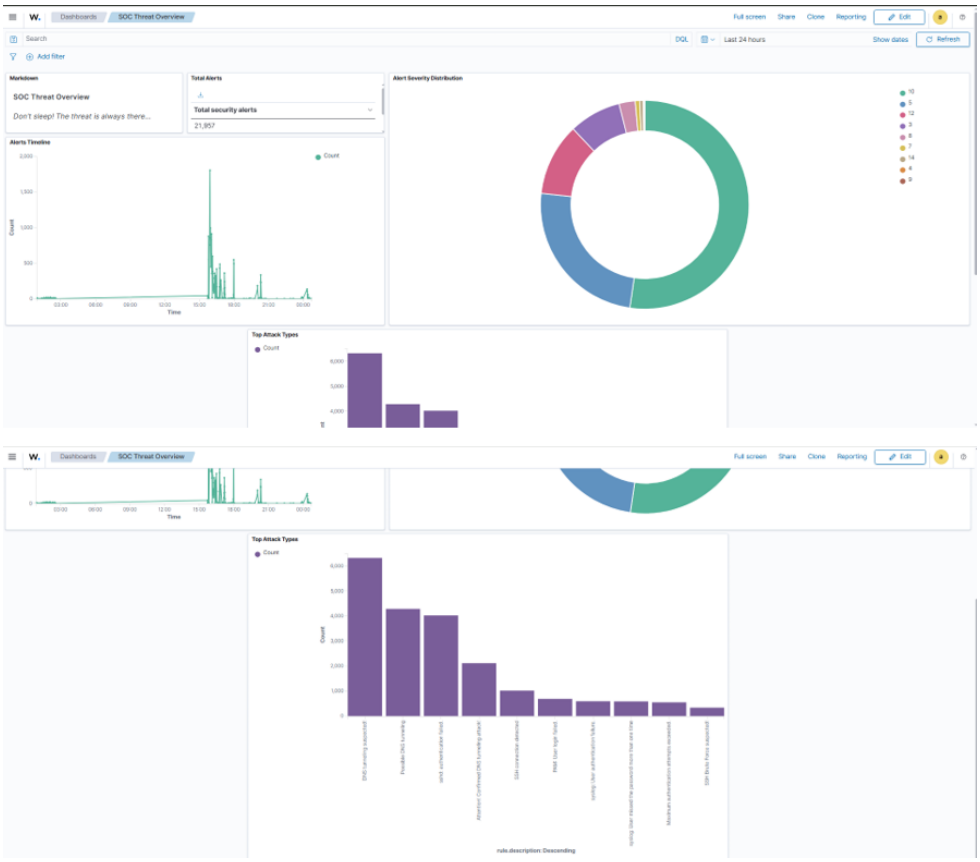


Рисунок 3.94 – Візуалізація дашборда «SOC Threat Overview»

Другий дашборд буде стосуватися аналізу атак. До нього додамо графік найчастіше спрацьованих правил, графік правил за описом, графік кількості подій за окремими агентами та таблицю про сповіщення високого рівня серйозності.

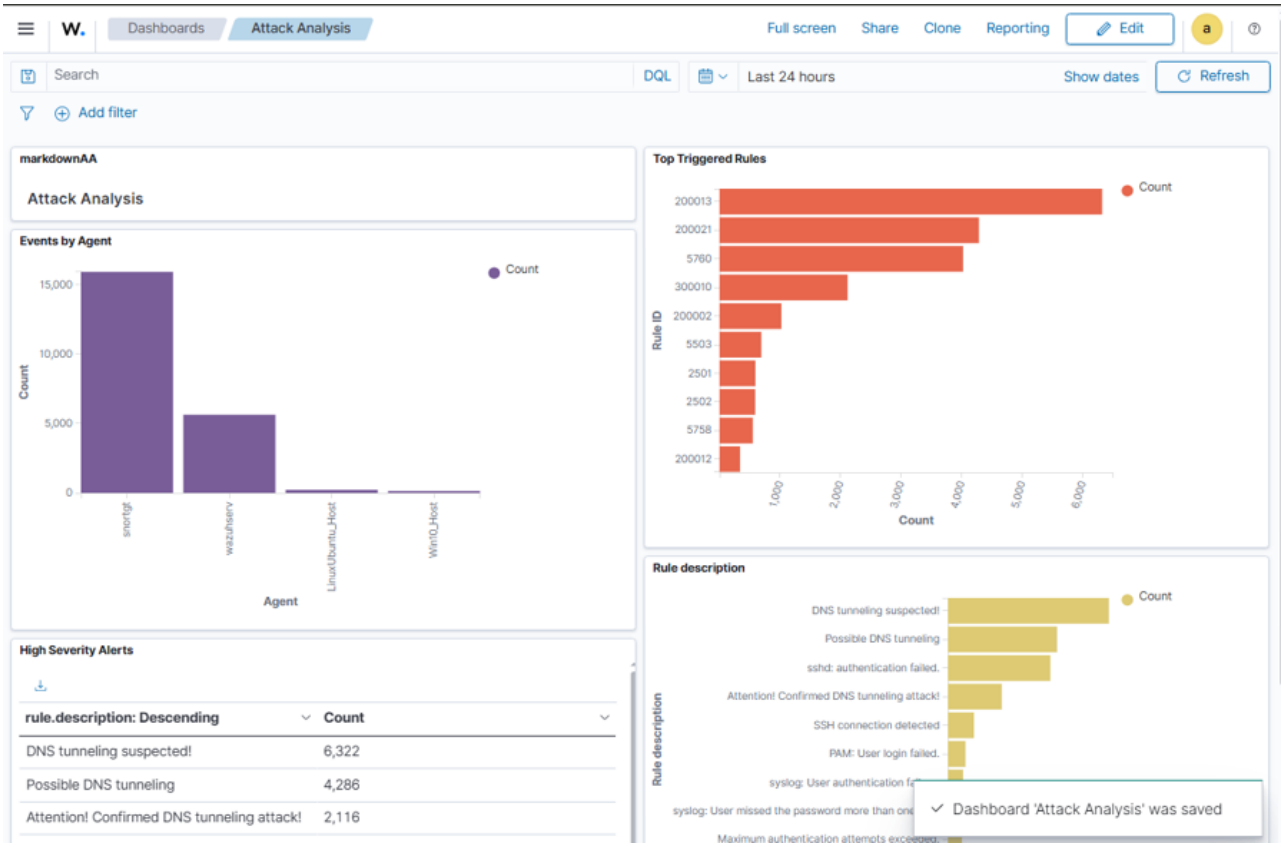


Рисунок 3.95 – Візуалізація дашборда «Attack Analysis»

Третій дашборд буде стосуватися ланцюга атак у підмережі. До нього додаємо графік ланцюга подій атак, графік хронології атак та кореляції правил.

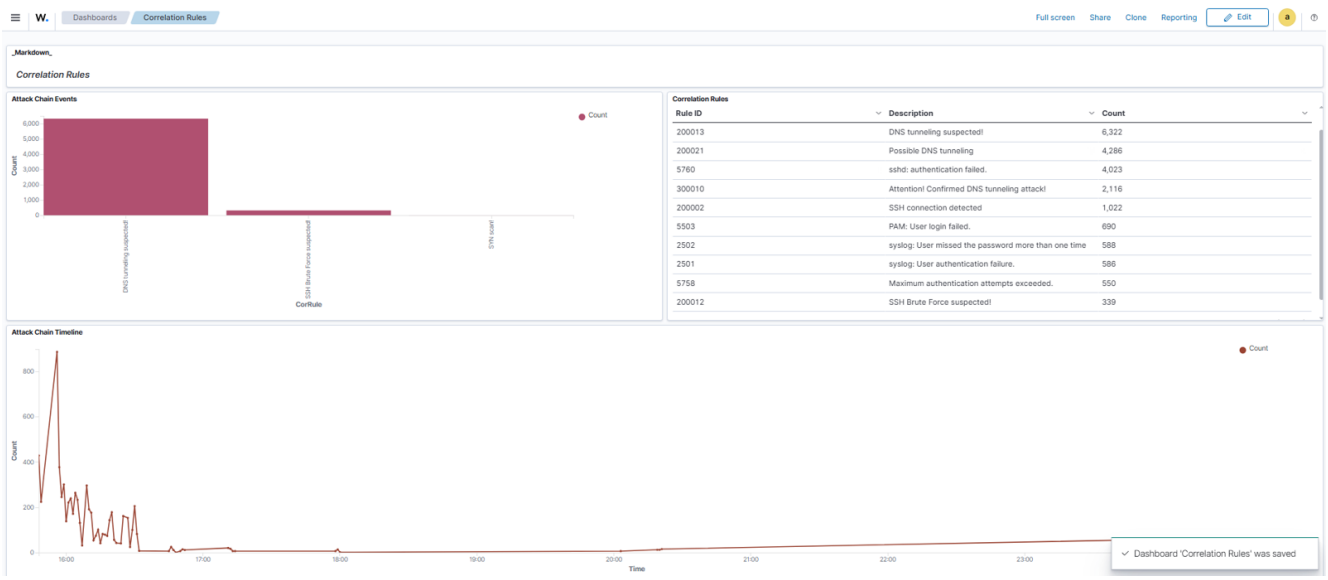


Рисунок 3.96 – Візуалізація дашборда «Attack Chain»

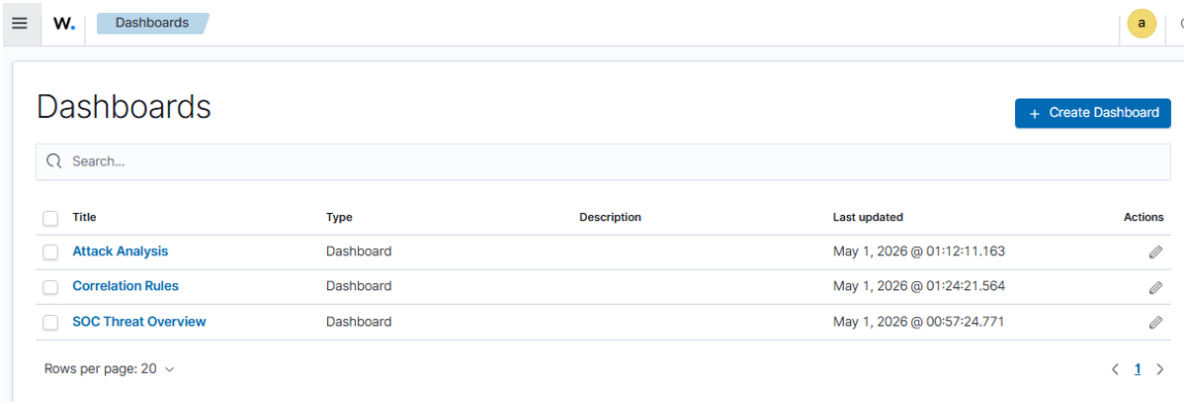


Рисунок 3.97 – Створені дашборди

В результаті створено 3 інформаційні дашборди, які мають важливу інформацію та допомагають полегшити моніторинг підемережі.

Створення дашбордів детальніше наведено у додатку Д.

3.6 Аналіз результатів функціонування системи

У ході практичної реалізації було перевірено роботу SIEM-системи Wazuh, IDS/IPS-системи Snort, механізмів кореляції подій та активного реагування. Основна увага приділялась не лише виявленню окремих підозрілих подій, а й аналізу повного ланцюга атаки – від генерації мережевого трафіку до автоматичного реагування системи.

У результаті тестування було підтверджено, що Snort коректно здійснює аналіз мережевого трафіку та формує події безпеки відповідно до створених локальних правил. Під час виконання ICMP-запитів, SYN-сканування, DNS-активності та SSH-підключень система успішно фіксувала підозрілі дії та передавала інформацію до Wazuh для подальшої обробки.

Проведені перевірки показали, що система здатна виявляти як прості аномалії мережевого трафіку так і більш складні сценарії, пов’язані з прихованими каналами передачі даних. Особливо це стосується DNS та ICMP-тунелювання, де використовуються нестандартні способи передачі інформації через дозволені мережеві протоколи. У звичайних умовах подібний трафік може залишатися непоміченим, однак використання правил Snort у поєднанні з кореляцією Wazuh дозволило фіксувати аномальну поведінку та підвищити рівень виявлення потенційних загроз.

Результати тестування ефективності функціонування системи та реагування на інциденти наведено у додатку Е.

Також під час дослідження було перевірено роботу активного реагування, де після спрацюваннння визначених правил Wazuh автоматично формував команди реагування, які передавались на сервер Snort для блокування джерела атаки через компонент «iptables». Було реалізовано базовий механізм IPS, де система не лише виявляє загрозу, а й виконує автоматичне реагування без втручання адміністратора.

Використання централізованої SIEM-системи дозволило значно спростити аналіз подій безпеки. Усі журнали від агентів та Snort збирались в одному місці, що забезпечило зручний моніторинг мережевої активності через веб-інтерфейс Wazuh. Крім того, система дозволила швидко визначати джерело атаки, тип загрози та час її виникнення.

Реалізована система підтримує IDS та IPS-функціональність шляхом використання Snort у режимі «inline inspection» із можливістю автоматичного блокування підозрілої мережевої активності.

Важливою частиною реалізованої системи стало використання інформаційних панелей моніторингу, які стали зручним інструментом відслідковування стану мережі, подій безпеки та результатів роботи IDS/IPS-системи. Завдяки дашбордам адміністратор може швидко виявляти підозрілу активність, аналізувати джерела атак, переглядати статистику подій та контролювати роботу механізмів без необхідності постійної роботи з командним рядком. Також використання візуалізації значно спрощує аналіз великої кількості подій та підвищує зручність адміністрування системи безпекового моніторингу.

Отримані результати демонструють ефективність інтеграції Wazuh та Snort для побудови базового SOC-рішення у віртуальному середовищі. У подальшому система може бути розширена за рахунок додавання систем аналізу кіберзагроз, кількості пристроїв, поведінкового аналізу трафіку, машинного навчання, штучного інтелекту, додаткових IDS-правил та інтеграції з іншими засобами кіберзахисту.

Висновки до розділу

У третьому розділі було виконано практичну реалізацію підсистеми комплексного безпекового моніторингу у віртуальному середовищі. Проведено розгортання мережевої інфраструктури, а також налаштування SIEM-системи Wazuh та IDS/IPS-системи Snort, а також реалізовано механізми інтеграції між компонентами системи. У процесі роботи було налаштовано централізований збір журналів подій, аналіз мережевого трафіку, локальні правила виявлення атак та механізми кореляції подій.

У межах тестування було перевірено виявлення різних типів мережевих атак. Результати тестування підтвердили працездатність підсистеми моніторингу, коректність передачі подій між компонентами системи та ефективність автоматизованого реагування на інциденти безпеки.

Отримані результати показали, що реалізована підсистема може використовуватися як базове рішення для моніторингу мережевої безпеки, виявлення підозрілої активності та автоматизації реагування на мережеві атаки у тестовому середовищі.

ВИСНОВКИ

У кваліфікаційній роботі було виконано проєктування та реалізовано підсистему комплексного безпекового моніторингу гетерогенної мережі на основі SIEM-системи Wazuh та IDS/IPS-системи Snort. Основною метою роботи стало дослідження сучасних підходів до моніторингу мережевої активності, виявлення кіберзагроз та автоматизації реагування на інциденти безпеки.

У процесі виконання роботи було проаналізовано принципи функціонування систем, розглянуто особливості роботи Wazuh і Snort, а також досліджено можливості їх інтеграції у єдину систему моніторингу подій безпеки.

У межах практичної частини було створено віртуальне середовище. Проведено налаштування мережевої інфраструктури, маршрутизації та взаємодії компонентів системи. Розгорнуто сервер Snort для аналізу мережевого трафіку та SIEM-систему Wazuh для централізованого збору, обробки й візуалізації подій безпеки. Було налаштовано передачу журналів подій між Snort і Wazuh, створено локальні правила виявлення атак та реалізовано механізм активного реагування для автоматичного блокування підозрілої активності.

Для перевірки працездатності системи проведено тестування різних типів мережевих атак. Результати тестування підтвердили коректність роботи системи моніторингу, успішне виявлення атак, передачу подій до Wazuh та працездатність механізмів автоматизованого реагування.

Реалізована система підтримує як IDS, так і IPS-функціональність шляхом використання Snort у режимі «inline inspection» із можливістю автоматичного блокування підозрілої мережевої активності.

У результаті виконання кваліфікаційної роботи було досягнуто поставленої мети – реалізовано базову підсистему комплексного безпекового моніторингу, здатну виконувати аналіз мережевого трафіку, централізований моніторинг подій безпеки, виявлення кіберзагроз та автоматичне реагування на підозрілу активність. Отримані результати можуть бути використані для подальшого розвитку систем моніторингу мережевої безпеки та дослідження сучасних засобів кіберзахисту.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Network Security Essentials: Applications and Standards // William Stallings. Pearson Education, 2017, 816 с.
2. Guide to Intrusion Detection and Prevention Systems (IDPS) // Karen Scarfone, Peter Mell. NIST Special Publication 800-94, 2007. URL: https://csrc.nist.gov/publications/detail/sp/800-94/final – с. 2-11.
3. Security Information and Event Management (SIEM) Implementation // David Miller. McGraw-Hill Education, 2011, 480 с.
4. Guide to Computer Security Log Management // Karen Kent, Murugiah Souppaya. NIST Special Publication 800-92, 2006. URL: https://csrc.nist.gov/publications/detail/sp/800-92/final – с. 3–15.
5. IBM QRadar SIEM // International Business Machines Corporation. 2026. URL: https://www.ibm.com/products/qradar-siem
6. Splunk Enterprise Security User Guide // Splunk Documentation. 2024. URL: https://docs.splunk.com/Documentation/ES/latest/User/Aboutoverviewdashboards
7. Elastic Security Platform // The Elascicsearch Platform. 2025. URL: https://www.elastic.co/platform
8. Wazuh - Open Source XDR. Open Source SIEM // Wazuh, Inc. 2026. URL: https://wazuh.com/
9. What is Microsoft Sentinel security information and event management (SIEM)? // Microsoft Learn. 2025. URL: https://learn.microsoft.com/en-us/azure/sentinel/overview?tabs=defender-portal
10. Simplified SIEM & Robust Cybersecurity for Enterprises // Graylog, Inc. 2026. URL: https://graylog.org/products/security/#
11. Wazuh Documentation // Wazuh Inc. 2025. URL: https://documentation.wazuh.com/
12. Elastic Stack and Security Analytics // Elastic Documentation. 2025. URL: https://www.elastic.co/security

13. Wazuh User Manual // Wazuh Inc. 2024. URL: https://documentation.wazuh.com/current/user-manual/index.html
14. MITRE ATT&CK Framework // MITRE. 2025. URL: https://attack.mitre.org/
15. Intrusion Detection Systems – CSRC // Rebecca Bace, Peter Mell. NIST, 2001. URL: https://csrc.nist.rip/library/NIST%20SP%20800-031%20Intrusion%20Detection%20Systems%20(IDS),%202001-11.pdf
16. Snort - Network Intrusion Detection & Prevention System // Snort. 2026. URL: https://www.snort.org/
17. Suricata User Guide // Open Information Security Foundation. 2026. URL: https://docs.suricata.io/
18. The Zeek Network Security Monitor // Zeek. 2026. URL: https://zeek.org/
19. Security Onion Documentation // Security Onion Solutions. 2025. URL: https://docs.securityonion.net/en/3/main/
20. Intrusion Detection with Snort // Rafeeq Ur Rehman. Prentice Hall PTR, 2003, 504 с.
21. Snort 3 Rule Writing Guide // Snort Documentation. 2025. URL: https://docs.snort.org/start/rules
22. VirtualBox User Manual // Oracle VM VirtualBox Documentation. 2025. URL: https://www.virtualbox.org/manual/topics/Introduction.html#ct_about-virtualbox
23. Linux IP Forwarding and NAT Configuration // Ubuntu Server Documentation. 2026. URL: https://ubuntu.com/server/docs
24. Wazuh Architecture Overview // Wazuh Documentation. 2026. URL: https://documentation.wazuh.com/current/getting-started/architecture.html
25. Security Operations Center (SOC) Concepts // IBM Security Learning Center. 2024. URL: https://www.ibm.com/think/topics/security-operations-center
26. Detecting SSH brute force attacks // Wazuh Documentation. 2025. URL: https://documentation.wazuh.com/current/proof-of-concept-guide/detect-brute-force-attack.html

27. DNS Tunneling Detection Guide // Palo Alto Networks Cyberpedia. 2024. URL: https://www.paloaltonetworks.com/cyberpedia/what-is-dns-tunneling
28. SIEM Correlation Rules: Fine-Tune Detection Logic at Scale // Cymulate. 2025. URL: https://cymulate.com/cybersecurity-glossary/siem-correlation-rules/
29. Wazuh Ruleset and Event Correlation // Wazuh Documentation. 2025. URL: https://documentation.wazuh.com/current/user-manual/ruleset/index.html
30. Network Security Testing Methodology // OWASP Foundation. 2024. URL: https://owasp.org/www-project-web-security-testing-guide/
31. TCP SYN Scan Techniques // Nmap Reference Guide. 2024. URL: https://nmap.org/book/synscan.html
32. DNS Tunneling Overview // Palo Alto Networks Cyberpedia. 2024. URL: https://www.paloaltonetworks.com/cyberpedia/what-is-dns-tunneling
33. Cyber Kill Chain Framework // Lockheed Martin. 2024. URL: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html
34. PuTTY – free SSH and telnet client for Windows // Putty. 2025. URL: https://putty.org/index.html
35. Wazuh Quickstart Guide // Wazuh Documentation. 2026. URL: https://documentation.wazuh.com/current/quickstart.html
36. Log data collection // Wazuh Documentation. 2026 . URL: https://documentation.wazuh.com/current/user-manual/capabilities/log-data-collection/index.html
37. Snort 3 Rule Writing Guide – Snort Rule Structure // Snort. 2025. URL: https://docs.snort.org/rules/
38. EICAR – Anti Malware Testfile // EICAR. 2026. URL: https://www.eicar.org/download-anti-malware-testfile/

ДОДАТКИ

Додаток А
Таблиця А.1.

Порівняння SIEM-рішень

Критерій	Wazuh	Splunk	IBM QRadar	Elastic Securitty	Microsoft Sentinel	Graylog
Тип ліцензії	Відкрита	Безкошто-вна/Комер-ційна	Комерційна	Безкоштовна /Комерційна	Комер-ційна	Відкрита /комерційна
Вартість впро-вадження	Безкоштовно	Висока	Висока	Середня	Висока	Низька
Централізо-ваний збір	Так	Так	Так	Так	Так	Так
Кореляція подій	Так	Так	Так	Так	Так	Частково
Графічний інтерфейс та візуалізація	Так	Так	Так	Так	Так	Так
Активне реагування	Так	Так	Частково	Обмежено	Так	Обмежено
Виявлення вразливос-тей	Так	Частково	Та	Ні	Частково	Ні
Інтеграція з системами IPS/IDS	Так	Так	Так	Так	Так	Так
Хмарна інте-грація	Частково	Так	Так	Частково	Так	Частково
Вимоги до ресурсів	Середні	Високі	Високі	Високі	Середні	Помірні
Складність розгортання	Середня	Висока	Висока	Середня	Висока	Середня
Функціонал	Повний при відкритій ліцензії	Обмежений при проб-ному періоді та розширений у ко-мерційній ліцензії	Повний у ко-мерційній ліцензії	Обмежений у безкоштовній та повний у комерційній	Загалом побудова-ний для роботи у Microsoft Azure	Відкрита ліце-нзія має обме-жену базу, ко-мерційна по-вну
Доцільність для середо-вища кваліфікацій ної роботи	Висока	Середня	Низька	Середня	Низька	Середня

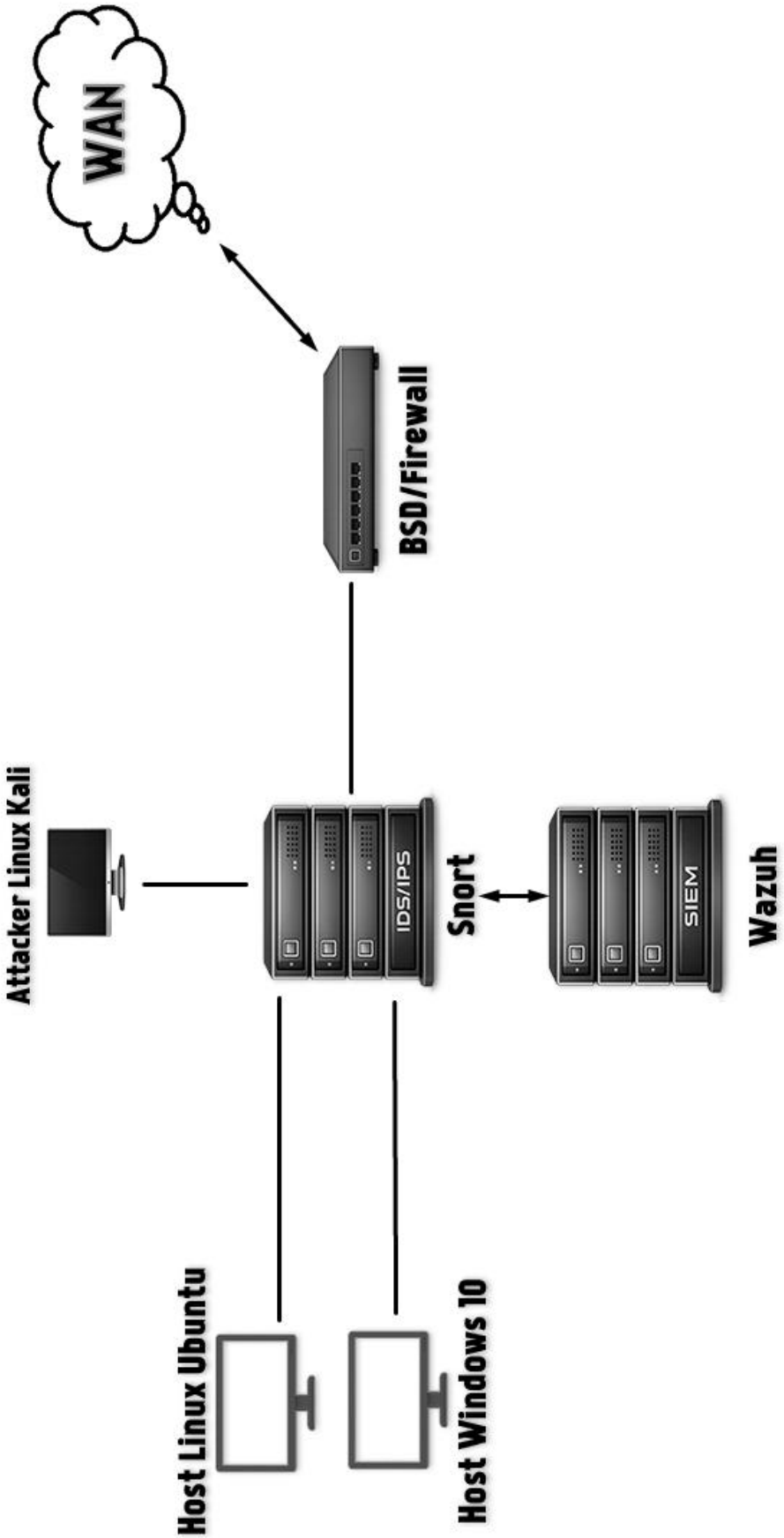
Таблиця А.2.

Порівняння IDS/IPS-рішень

Критерій	Snort	Suricata	Zeek	Security Onion	Cisco Firepower	Forti-Gate IPS	Palo Alto Threat Preven-tion
Тип системи	IDS/IPS	IDS/IPS	NSM/ IDS	Комплексна платформа	NGFW/ IPS	NGFW/ IPS	NGFW/ IPS
Відкрита	Так	Так	Так	Так	Ні	Ні	Ні
Сигнатур-ний аналіз	Так	Так	Частково	Так	Так	Так	Так
Підтримка IPS-режиму	Так	Так	Ні	Так	Так	Так	Так
Режим «inline in-spection»	Так	Так	Ні	Частково	Так	Так	Так
Підтримка власних пра-вил	Так	Так	Частково	Так	Обмежено	Обмежено	Обмежено
Аналіз мере-жевих про-токолів	Так	Так	Так	Так	Так	Так	Так
Інтеграція із SIEM-системами	Так	Так	Так	Так	Так	Так	Так
Вимоги до ресурсів	Помірні	Високі	Високі	Високі	Високі	Високі	Високі
Складність налашту-вання	Середня	Середня	Висока	Висока	Висока	Висока	Висока
Вартість ви-користання	Без-коштовно	Без-коштовно	Без-коштовно	Безкоштовно	Висока	Висока	Висока
Доцільність для середо-вища кваліфікацій ної роботи	Висока	Висока	Середня	Низька	Низька	Низька	Низька

Додаток Б

Загальна структура мережевої інфраструктури



Додаток В

Встановлення та налагодження FreeBSD pfSense

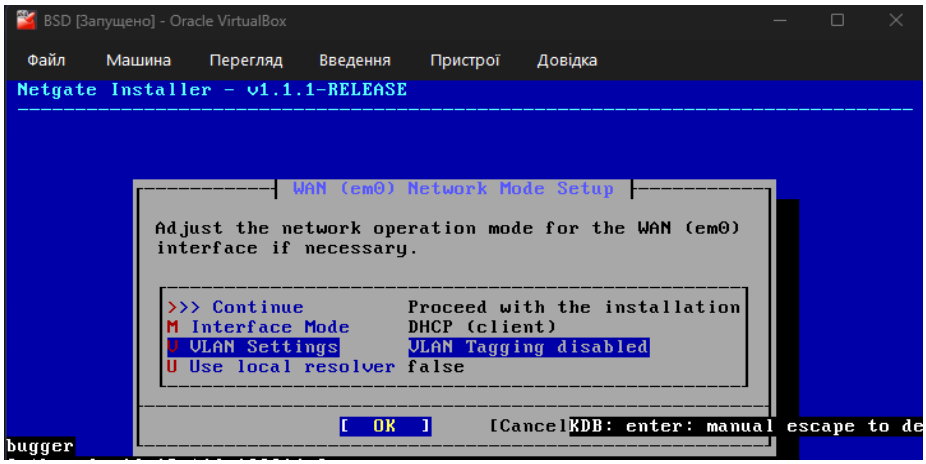


Рисунок В.1 – Налаштування WAN-інтерфейсу

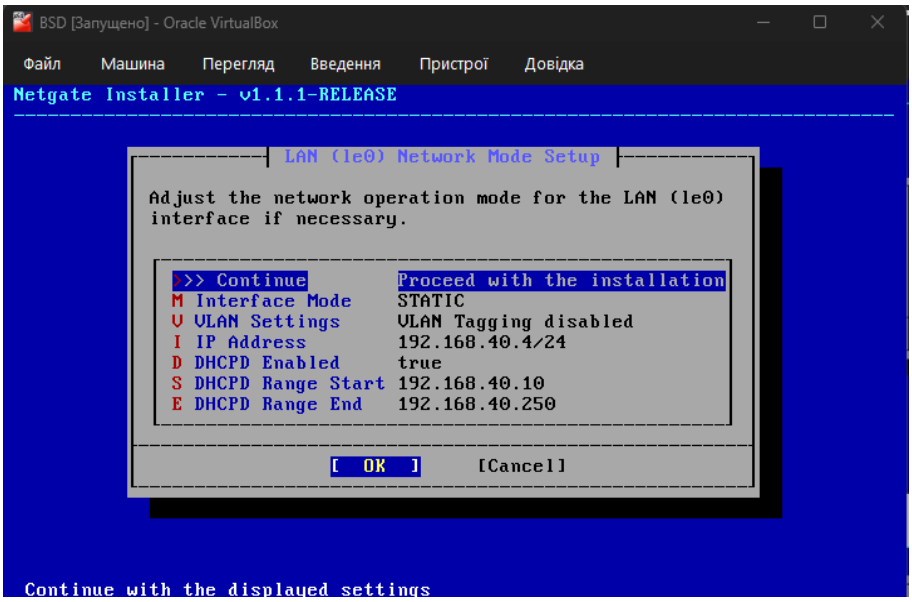


Рисунок В.2 – Налаштування LAN-інтерфейсу

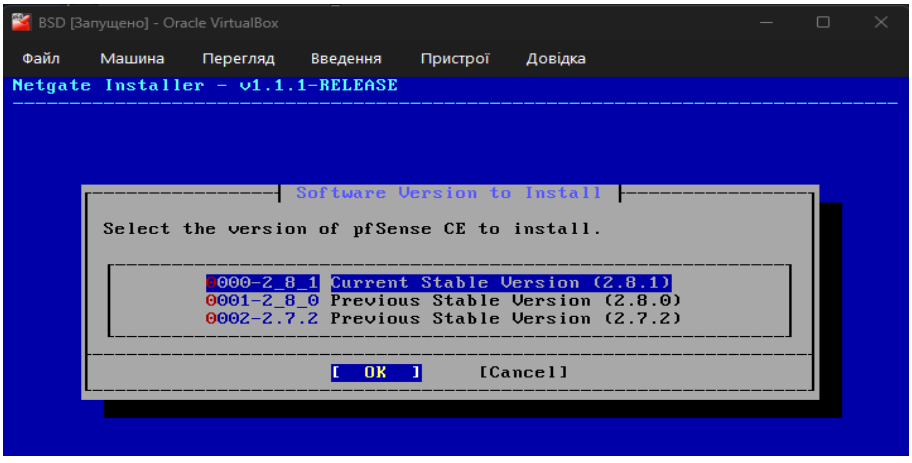


Рисунок В.3 – Вибір версії

Перевірка та аналіз роботи систем Wazuh і Snort

Підключення через SSH



Рисунок Г.1 – Спроба підключення

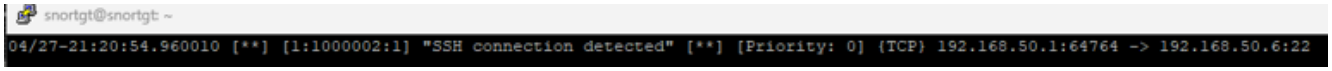


Рисунок Г.2 – Фіксація події

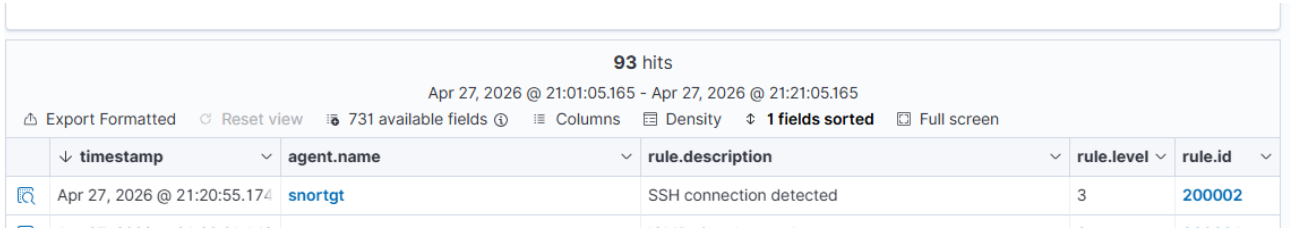


Рисунок Г.3 – Алерт у Wazuh

Перевірка DNS-запитів

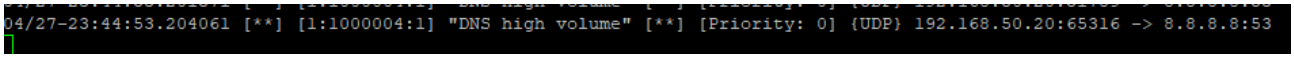


Рисунок Г.4 – Фіксація події

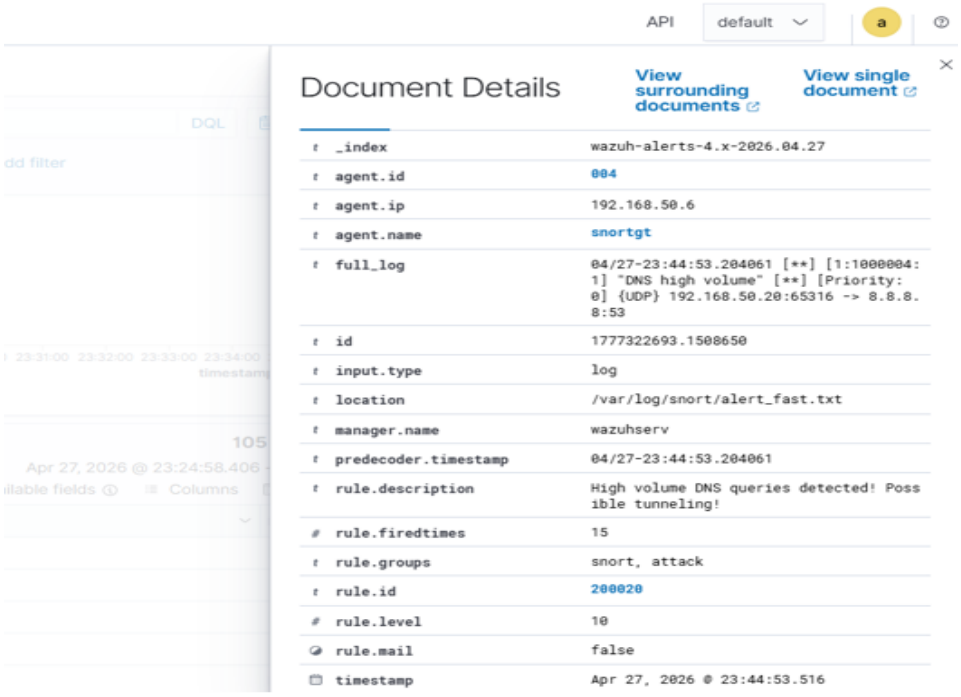


Рисунок Г.5 – Алерт у Wazuh

Перевірка сканування портів

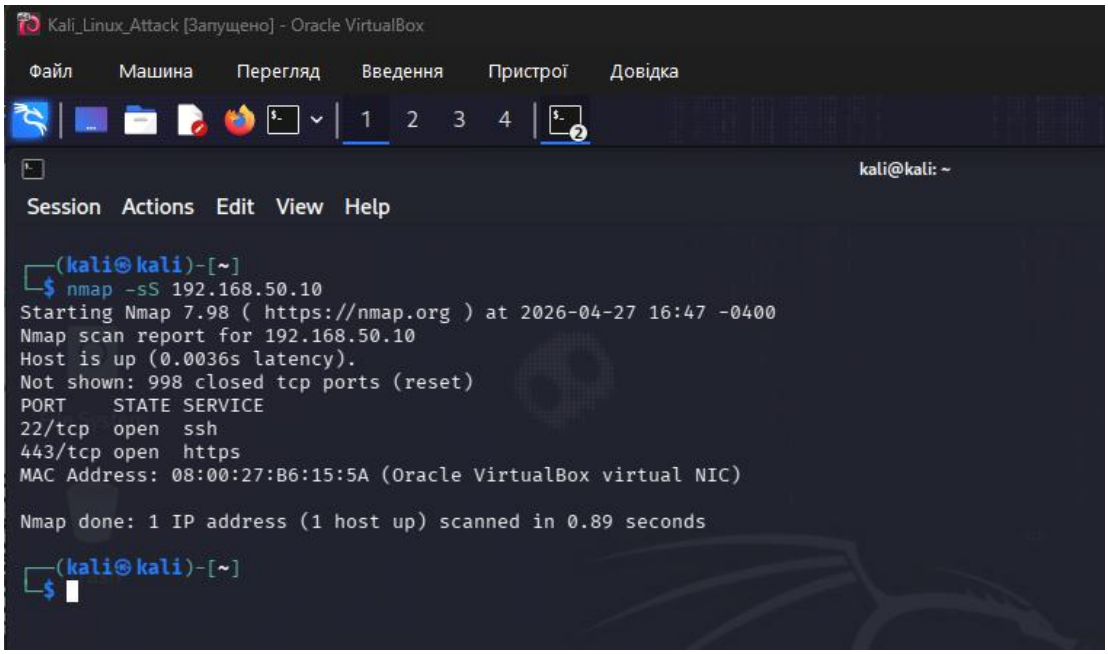


Рисунок Г.6 – Сканування через Nmap

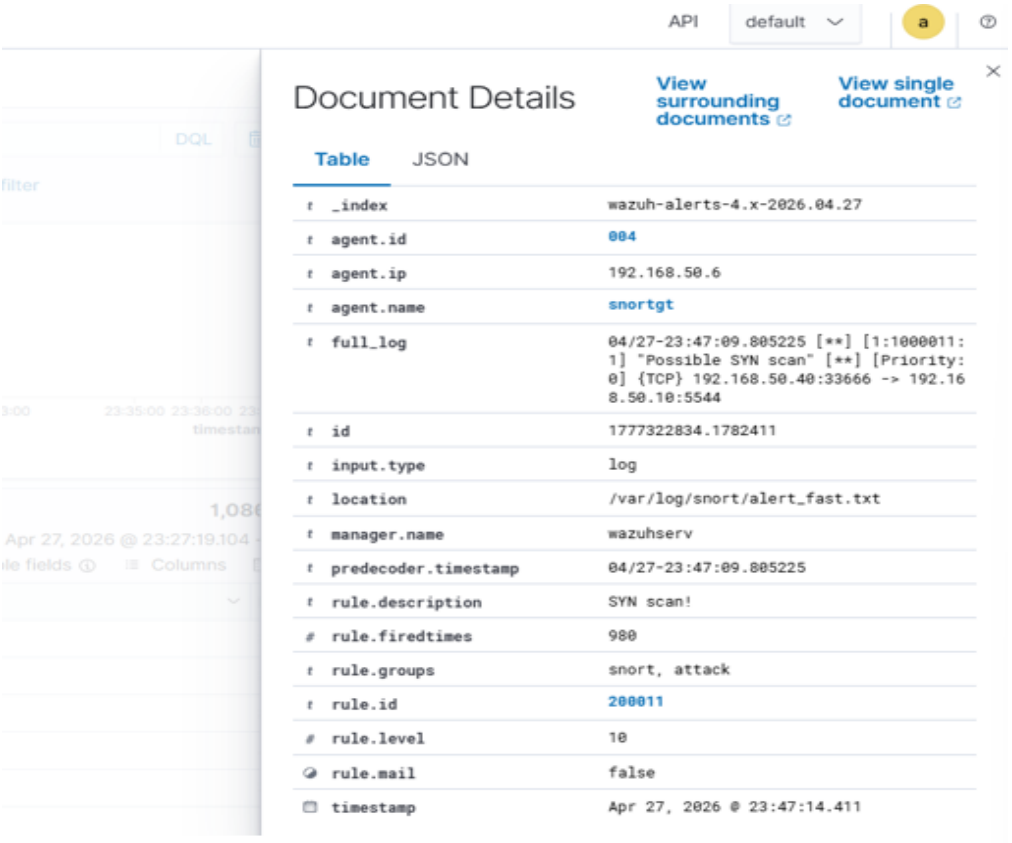


Рисунок Г.7 – Алерт у Wazuh

Перевірка атаки типу SSH brute force

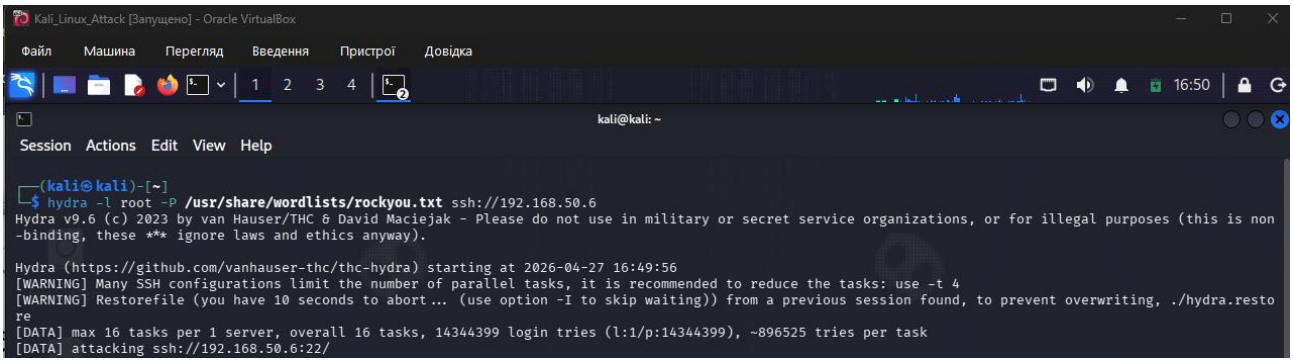


Рисунок Г.8 – Проведення атаки

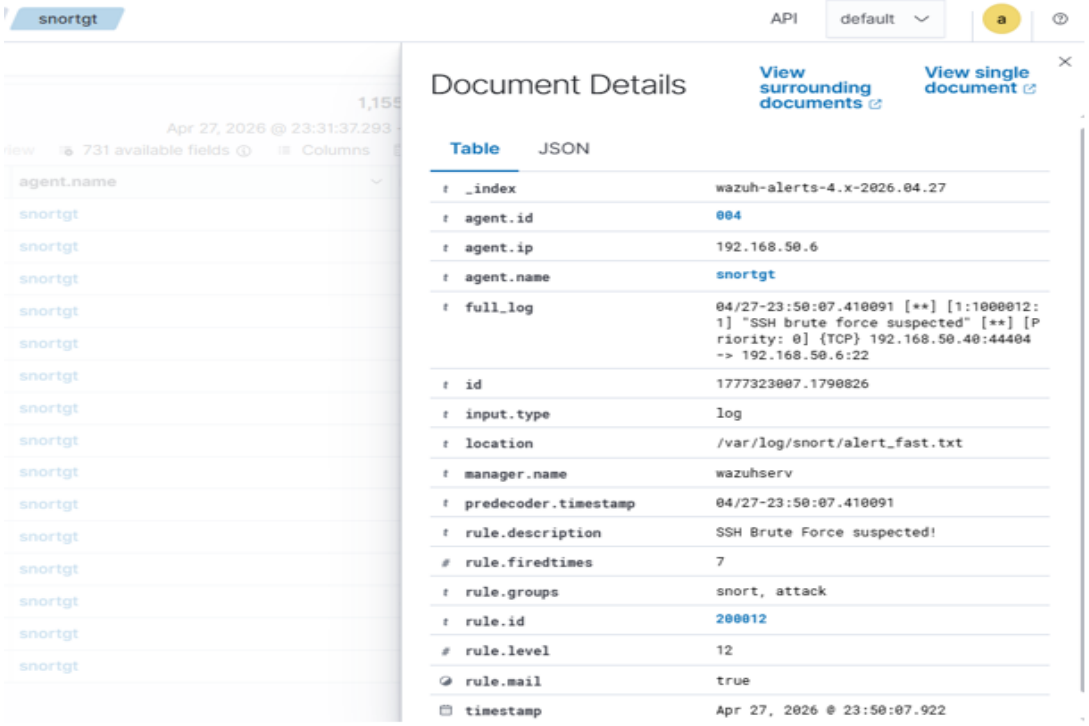


Рисунок Г.9 – Алерт у Wazuh

Перевірка SYN scan – SSH brute force

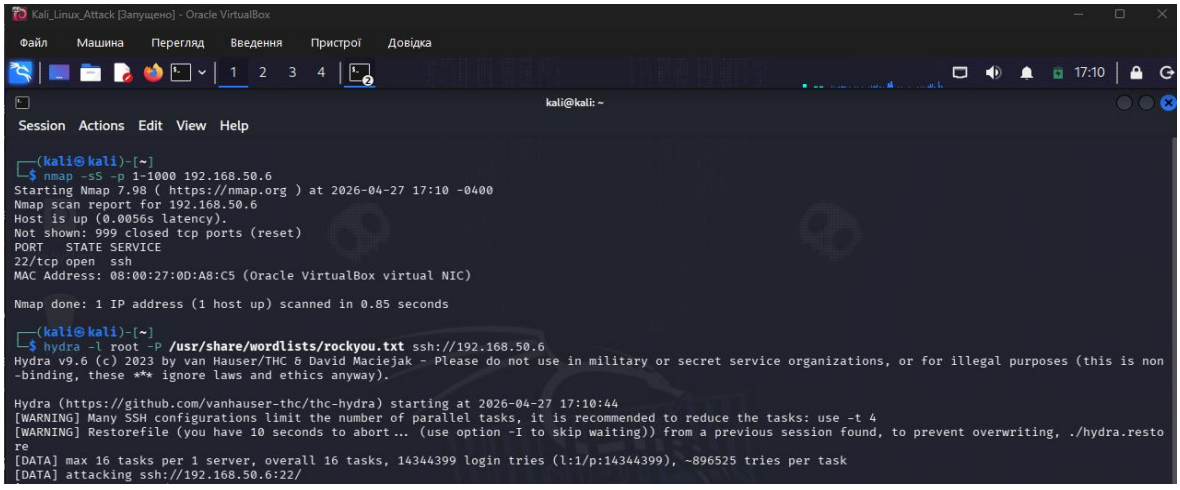


Рисунок Г.10 – Проведення атаки

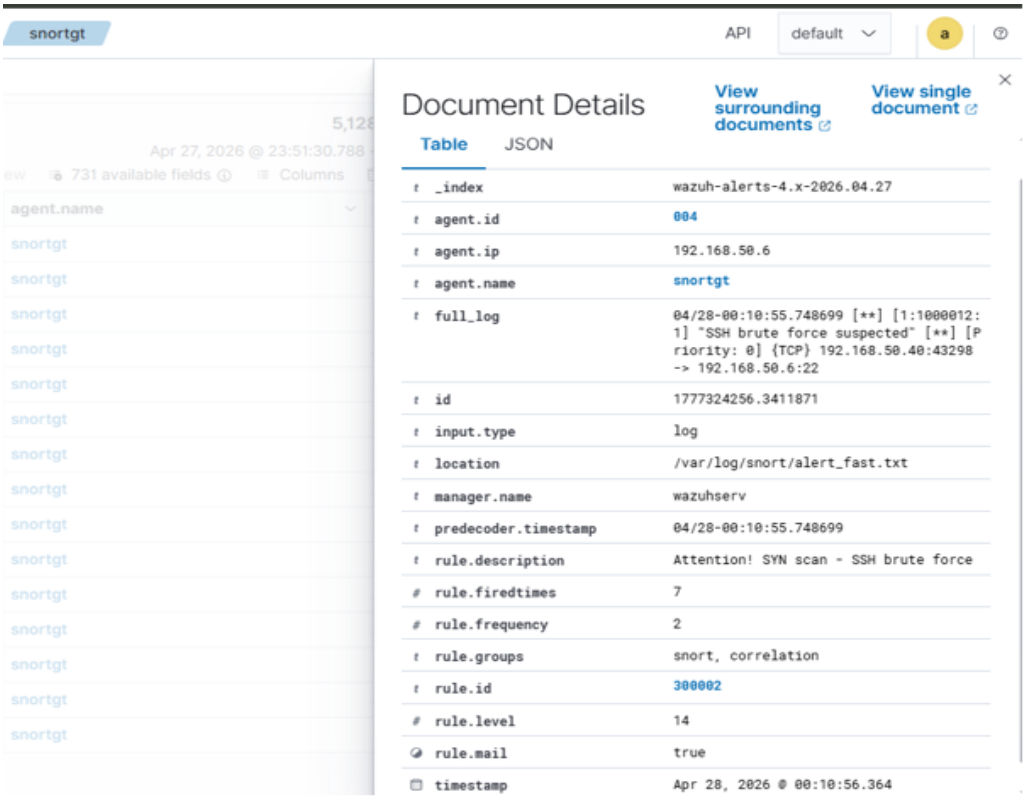


Рисунок Г.11 – Алерт у Wazuh

Перевірка підозри на DNS-тунелювання

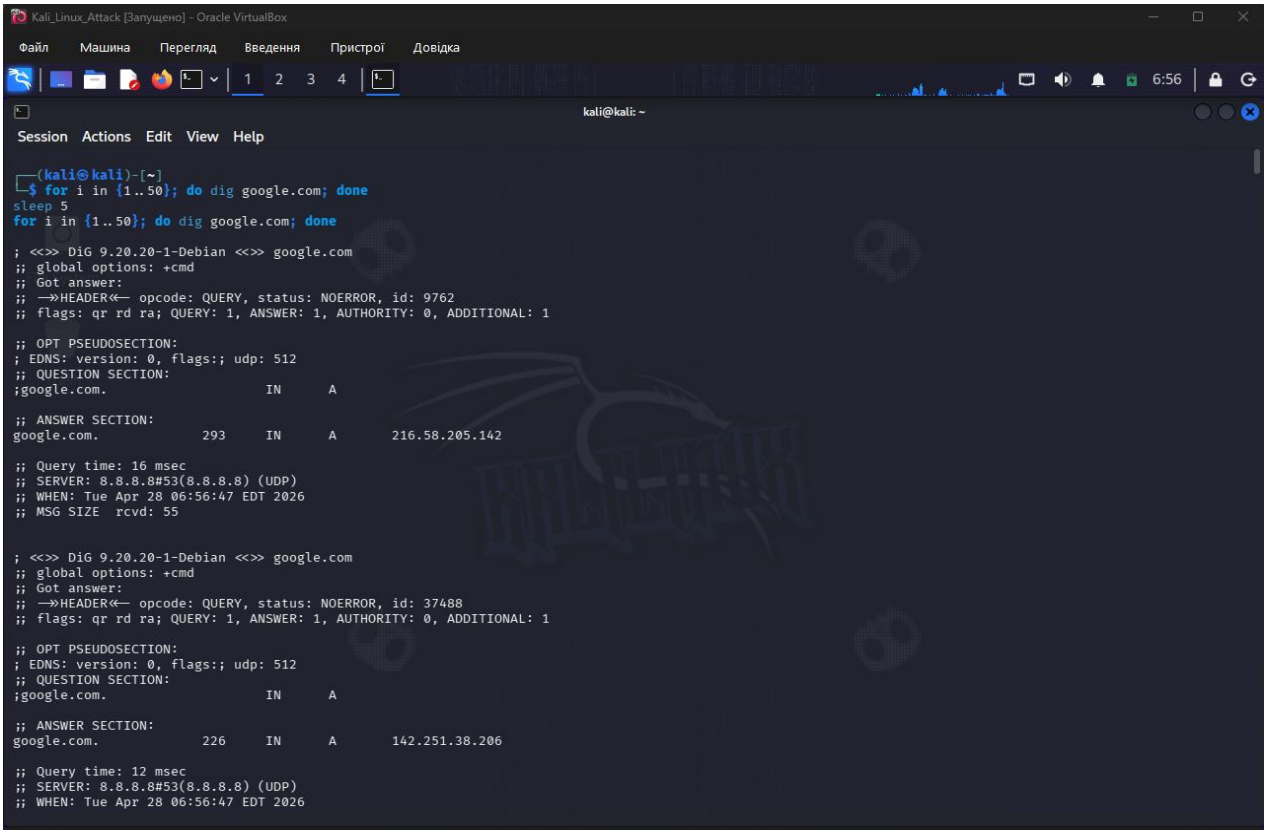


Рисунок Г.12 – Виконання скрипта



Рисунок Г.13 – Алерт у Wazuh

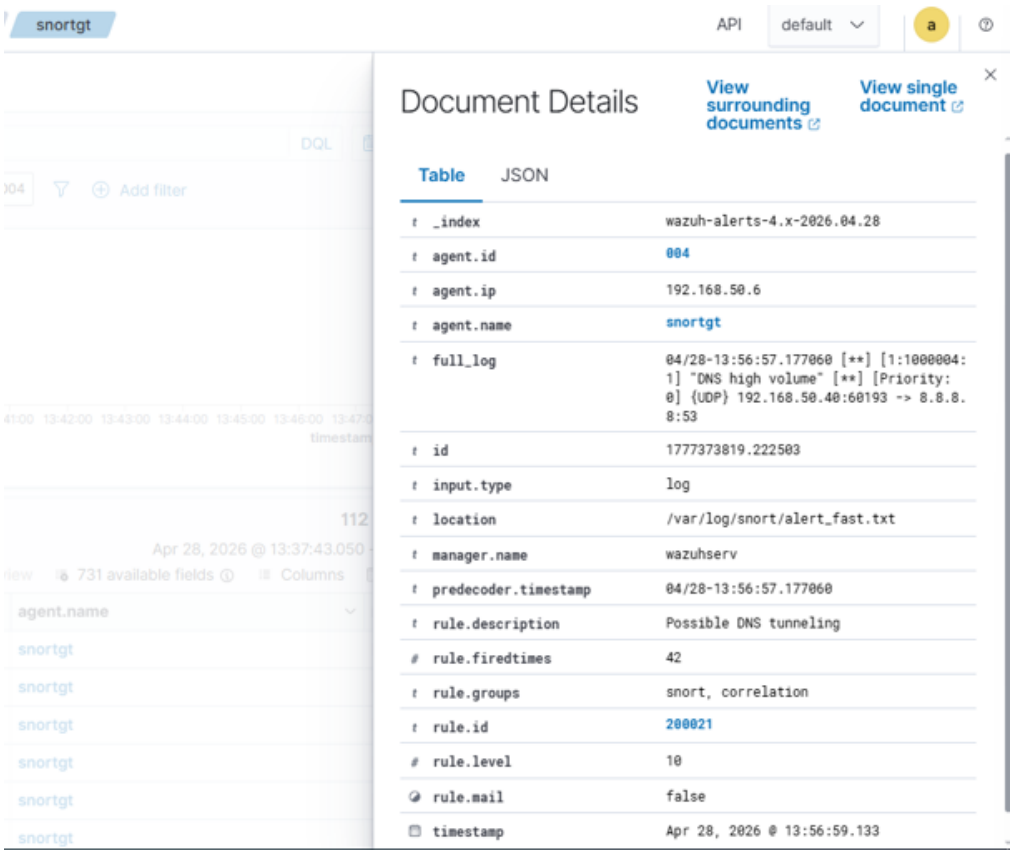


Рисунок Г.14 – Детальні відомості

Перевірка робити Firewall на FreeBSD pfSense

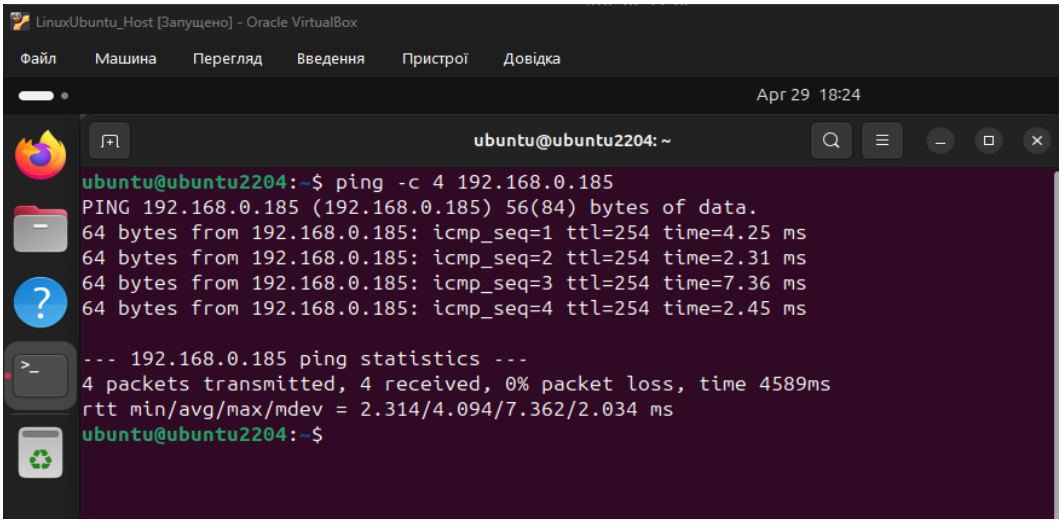


Рисунок Г.15 – Пінг з локальної мережі в глобальну

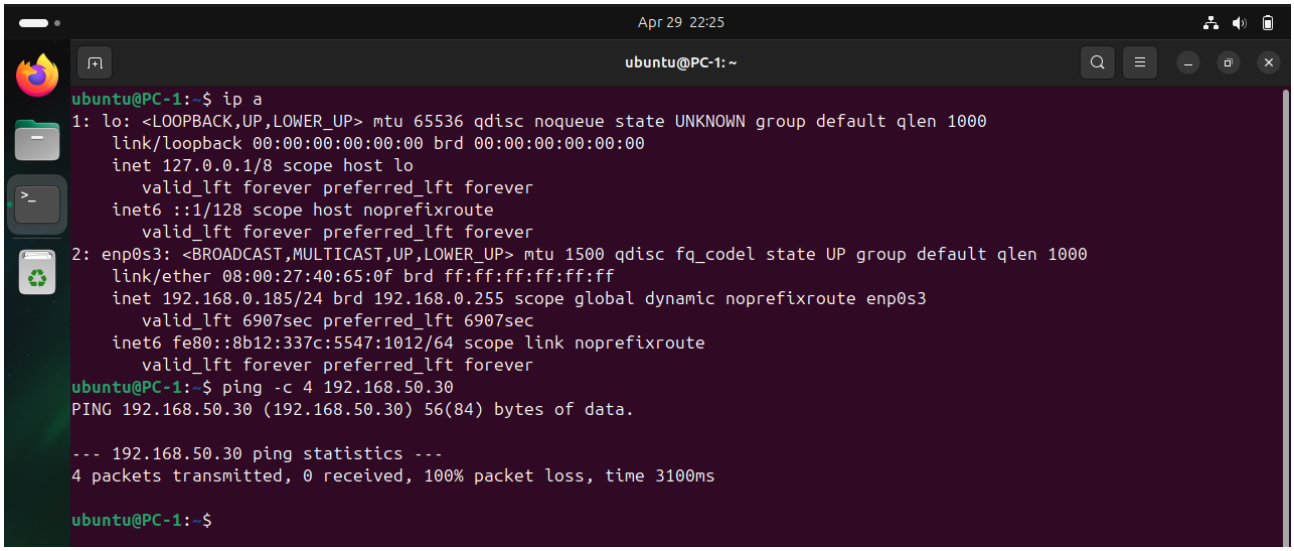


Рисунок Г.16 – Пінг з глобальої мережі в локальну

Перевірка EICAR

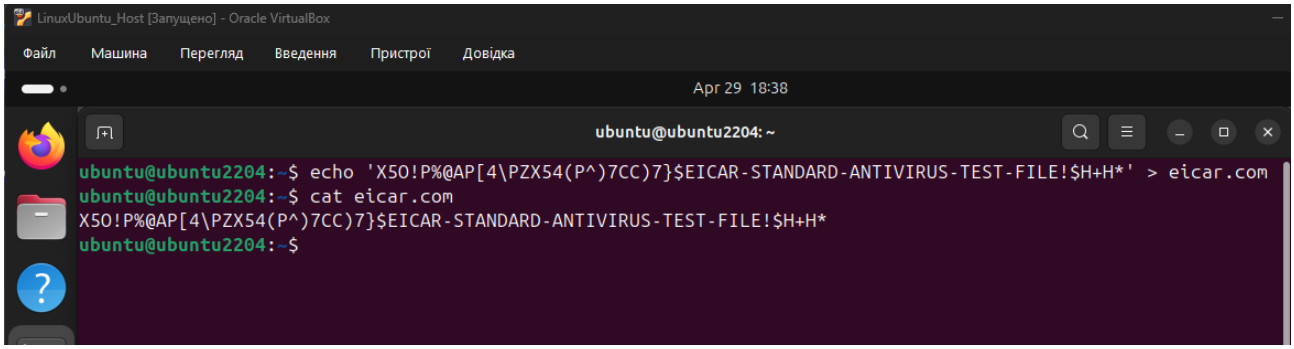
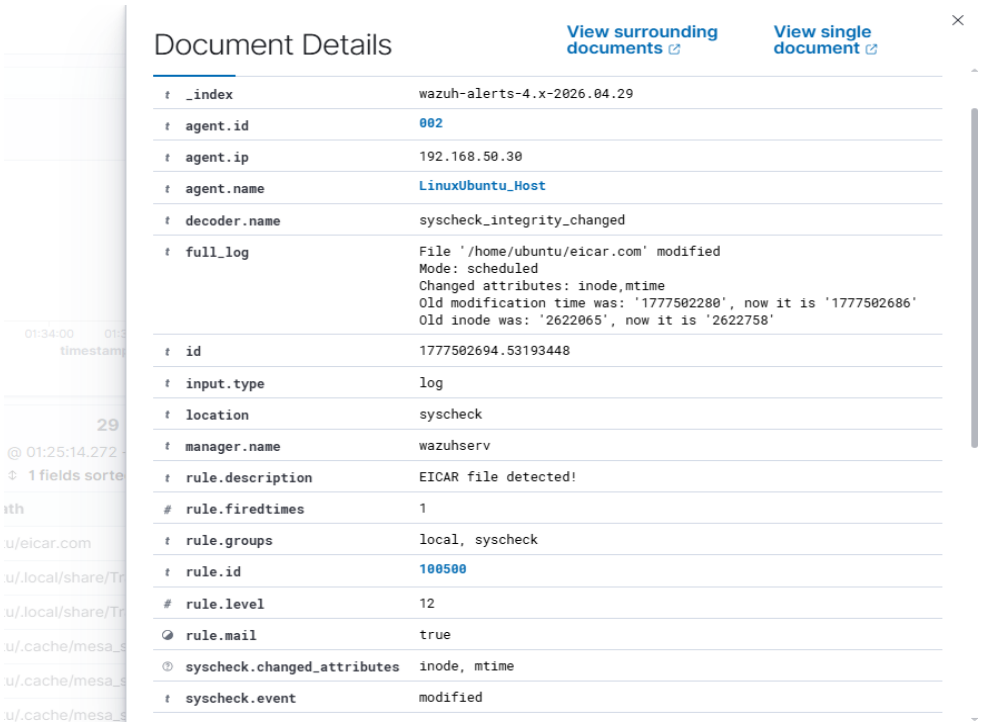


Рисунок Г.17 – Створення файлу



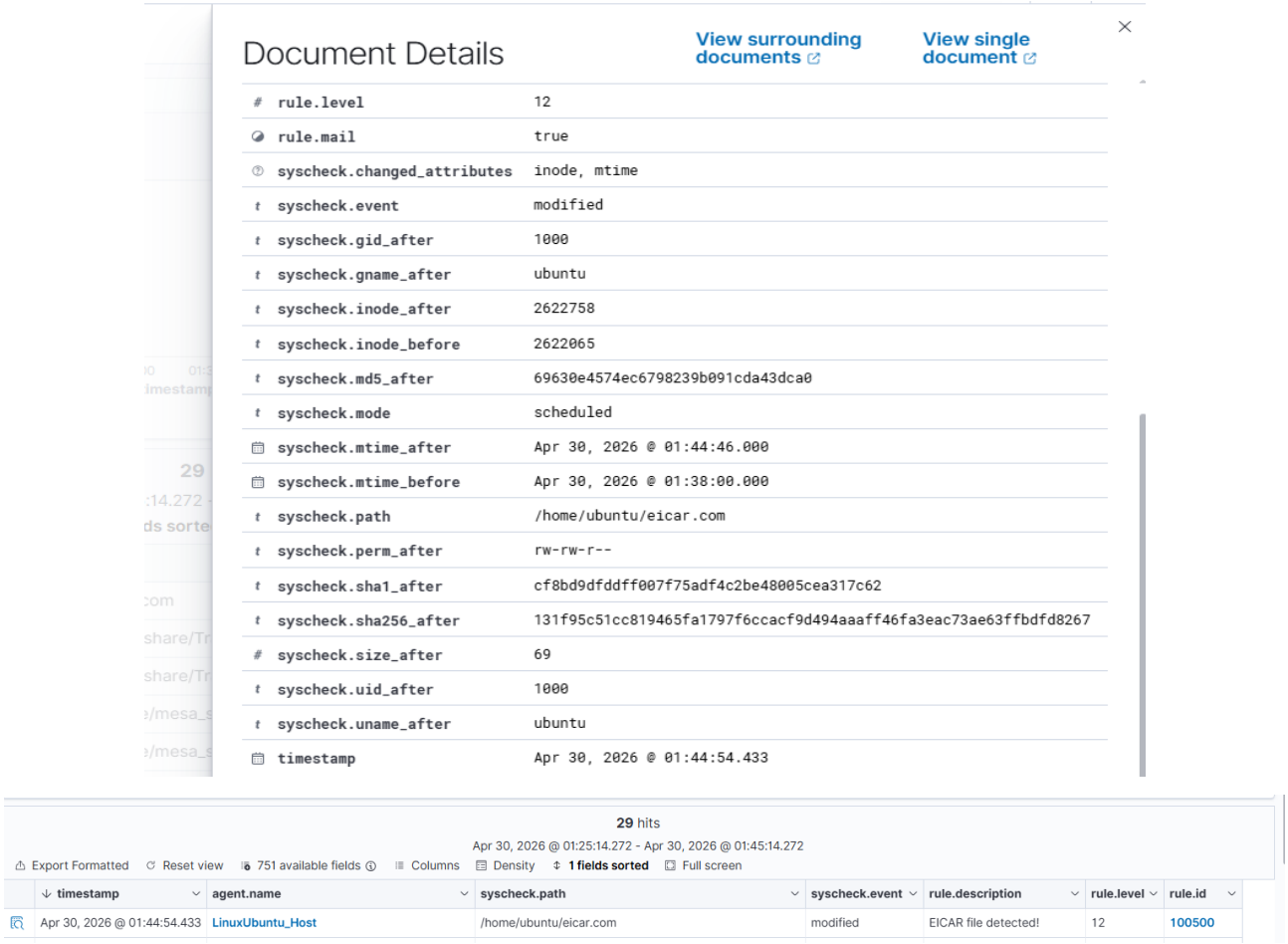


Рисунок Г.18 – Реагування системи на шкідливий файл

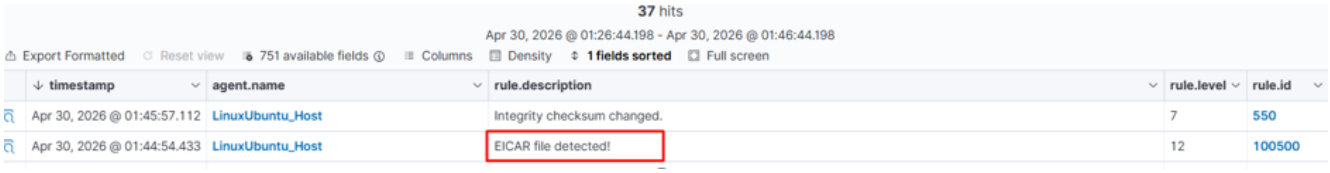


Рисунок Г.19 – Виведення алерта за власним правилом

Перевірка активного реагування на видалення файлу EICAR

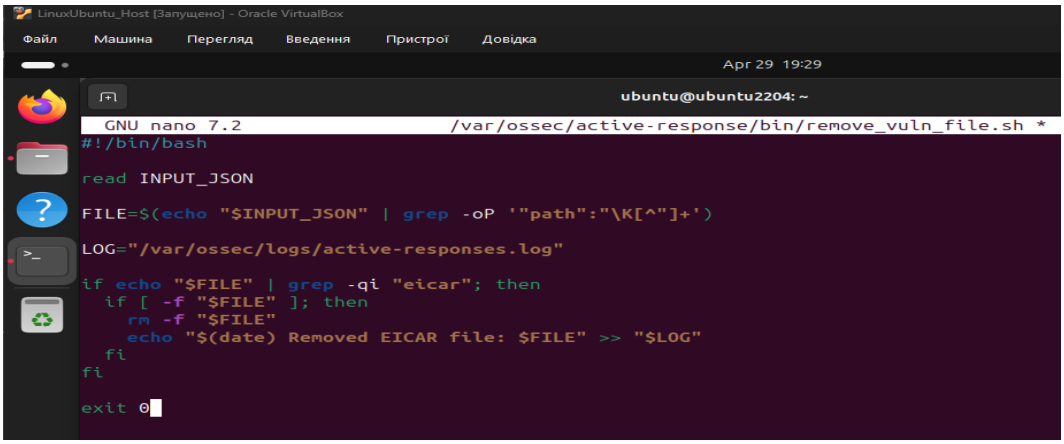


Рисунок Г.20 – Створення скрипта

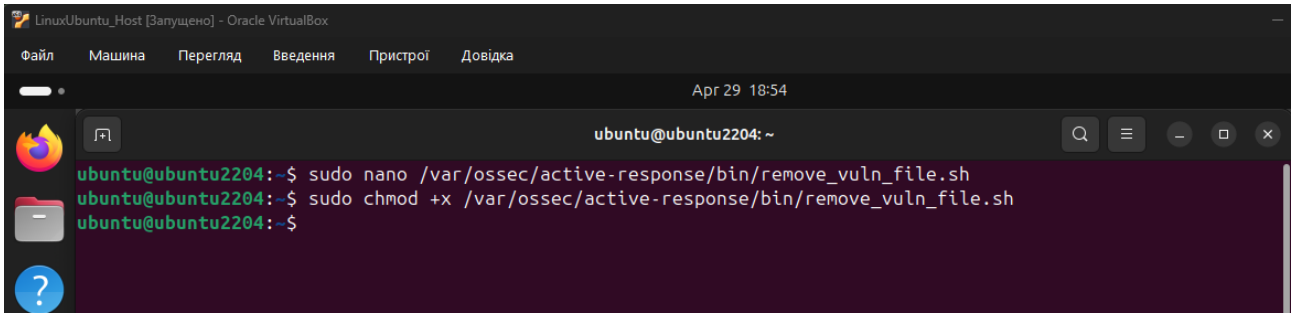


Рисунок Г.21 – Надання прав читання

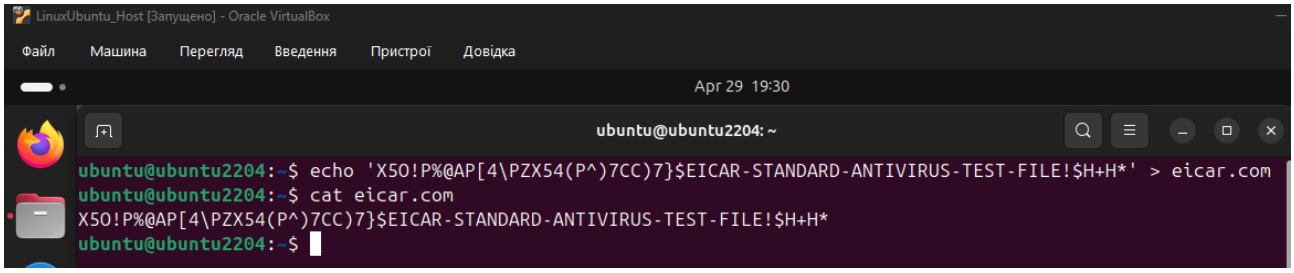


Рисунок Г.22 – Повторне створення

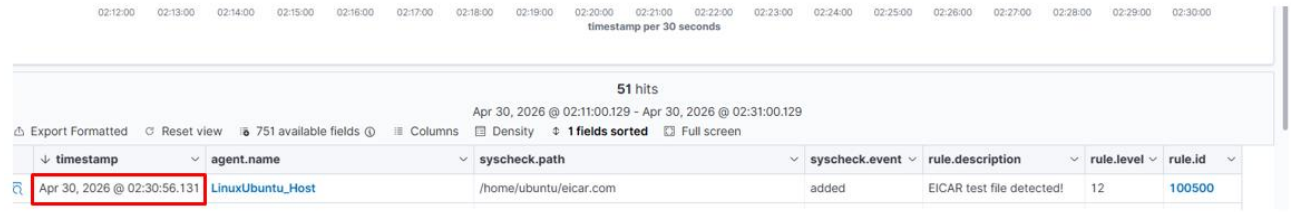


Рисунок Г.23 – Файл додано

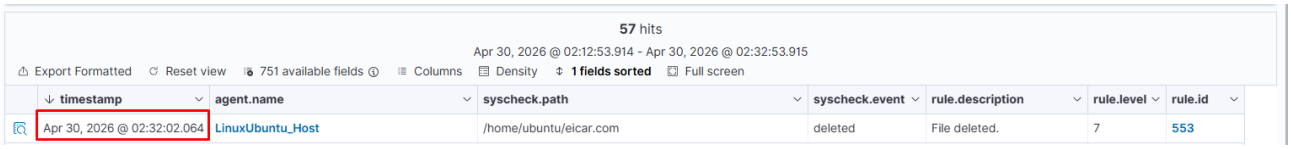


Рисунок Г.24 – Файл видалено

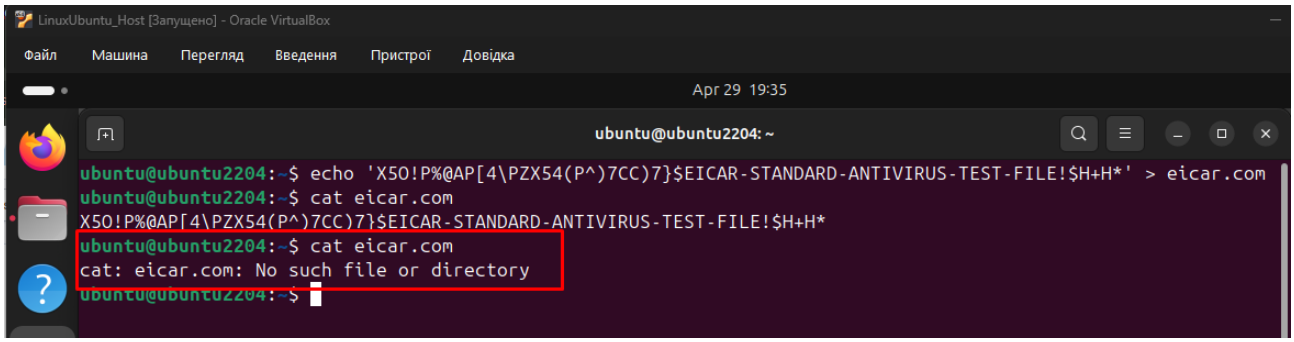


Рисунок Г.25 – Перевірка роботи активного реагування на хості

Додаток Д

Створення інформаційних дашбордів

Дашборд «SOC Threat Overview»

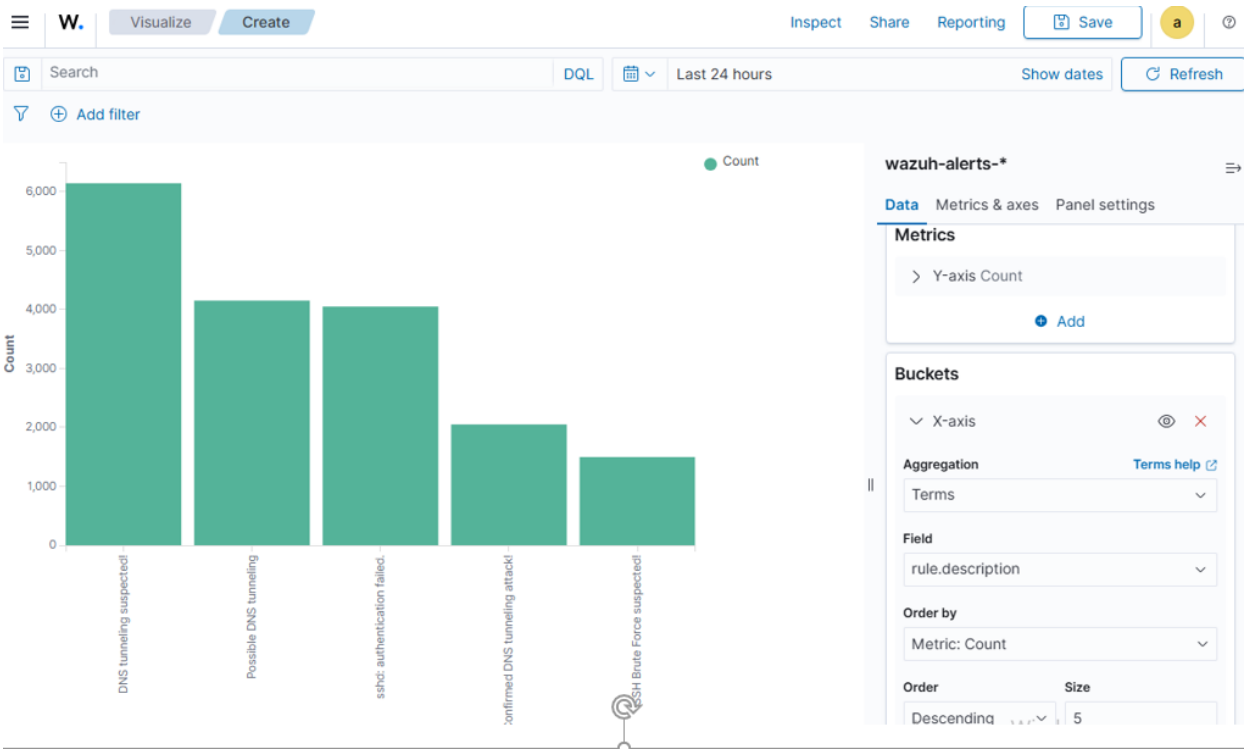


Рисунок Д.1 – Графік «Top Attack Types»

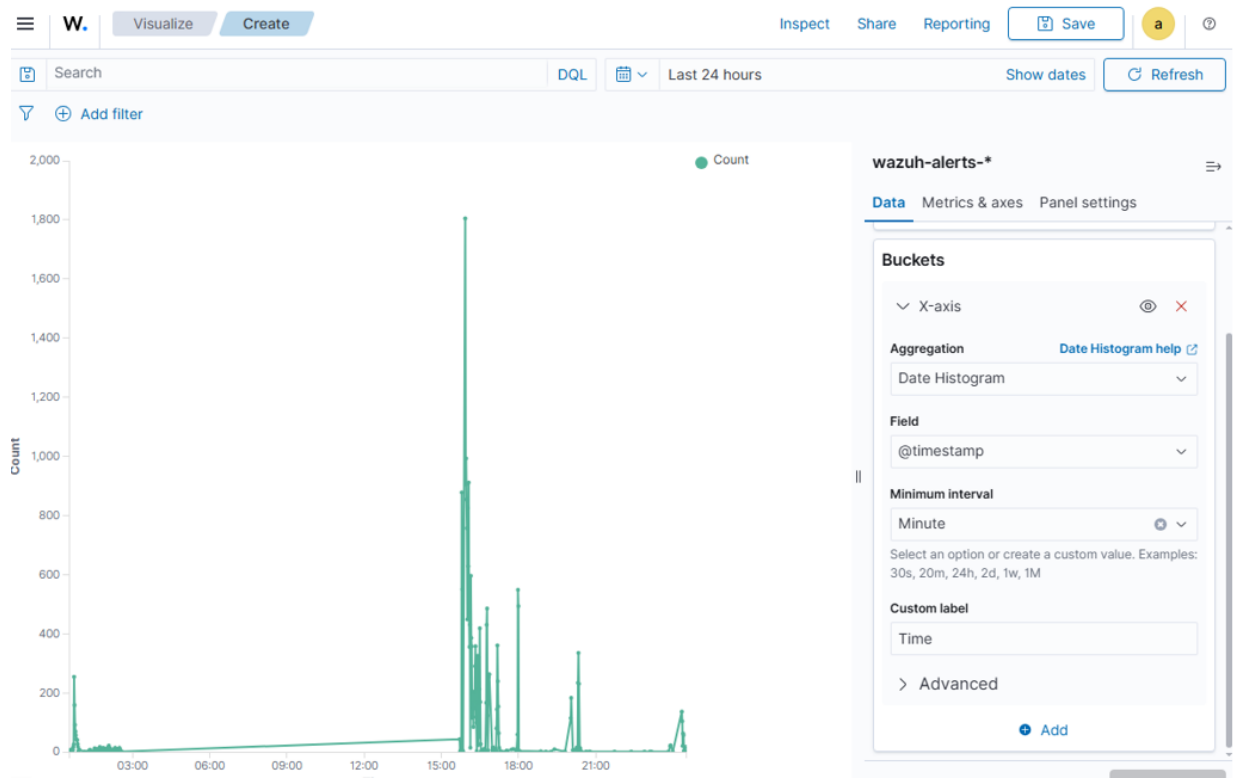


Рисунок Д.2 – Графік «Alerts Timeline»

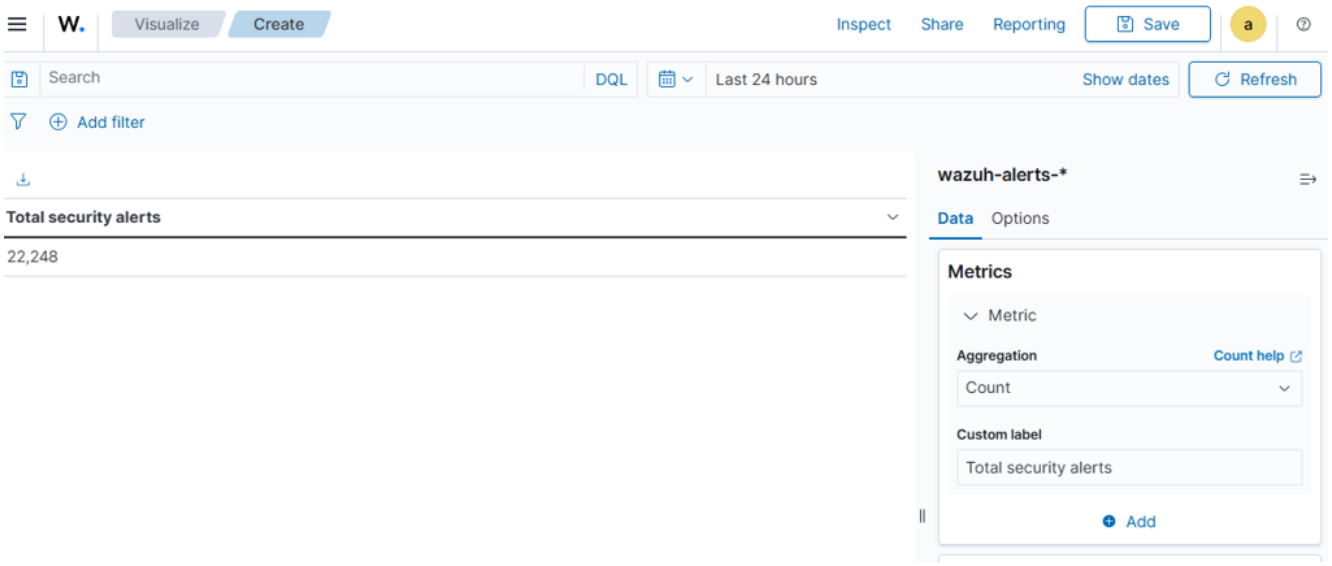


Рисунок Д.3 – Графік «Alert Severity Distribution»

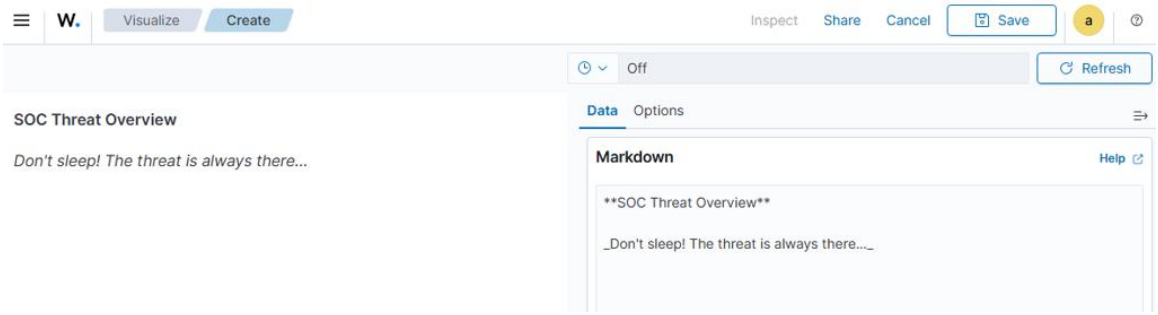


Рисунок Д.4 – Інформаційне поле

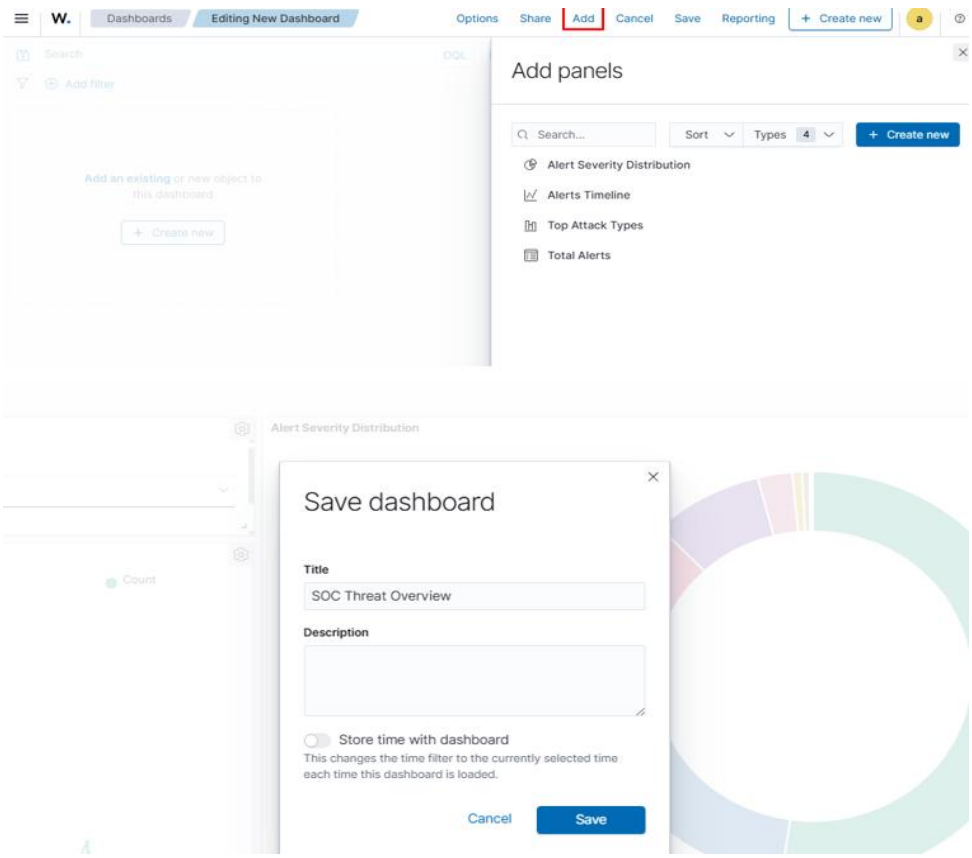


Рисунок Д.5 – Створення дашборду

Дашборд «Attack Analysis»

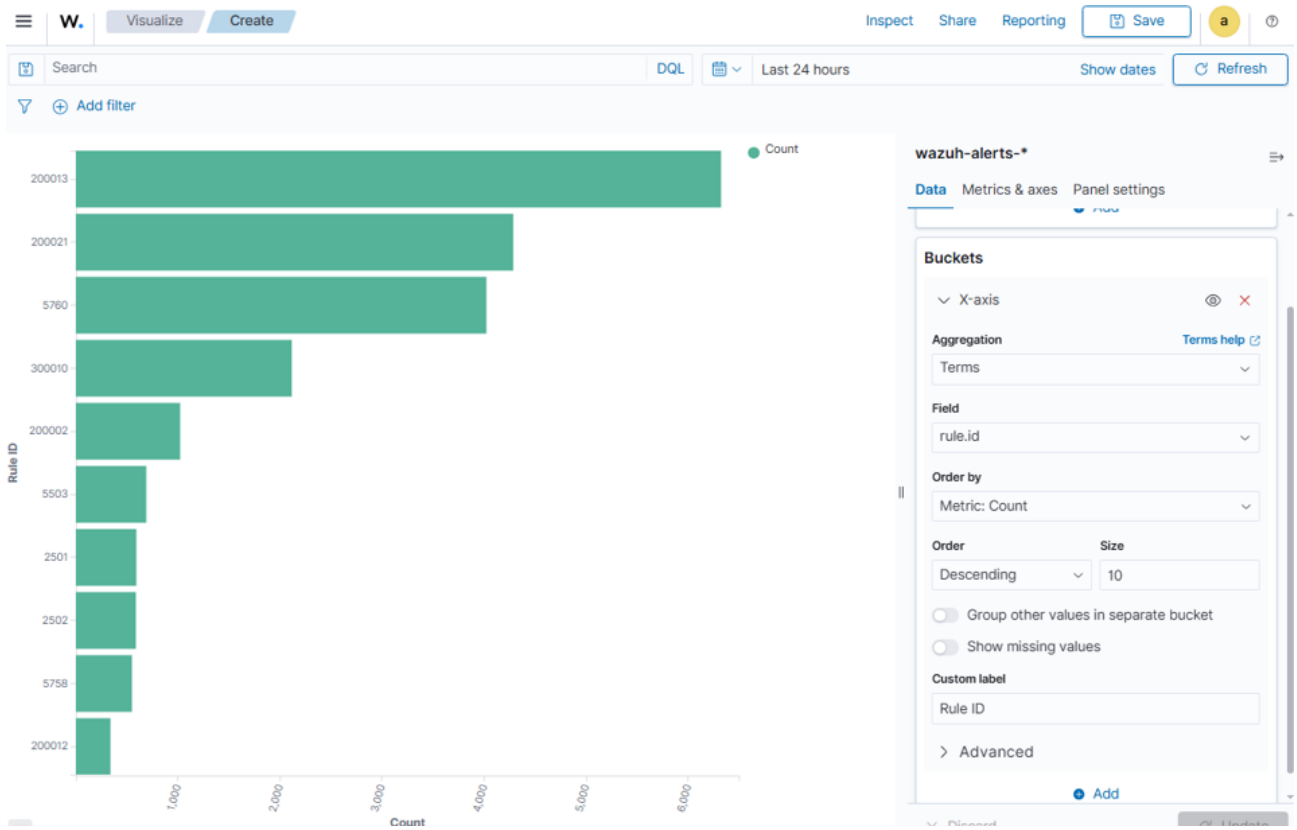


Рисунок Д.6 – Графік «Top Triggered Rules»

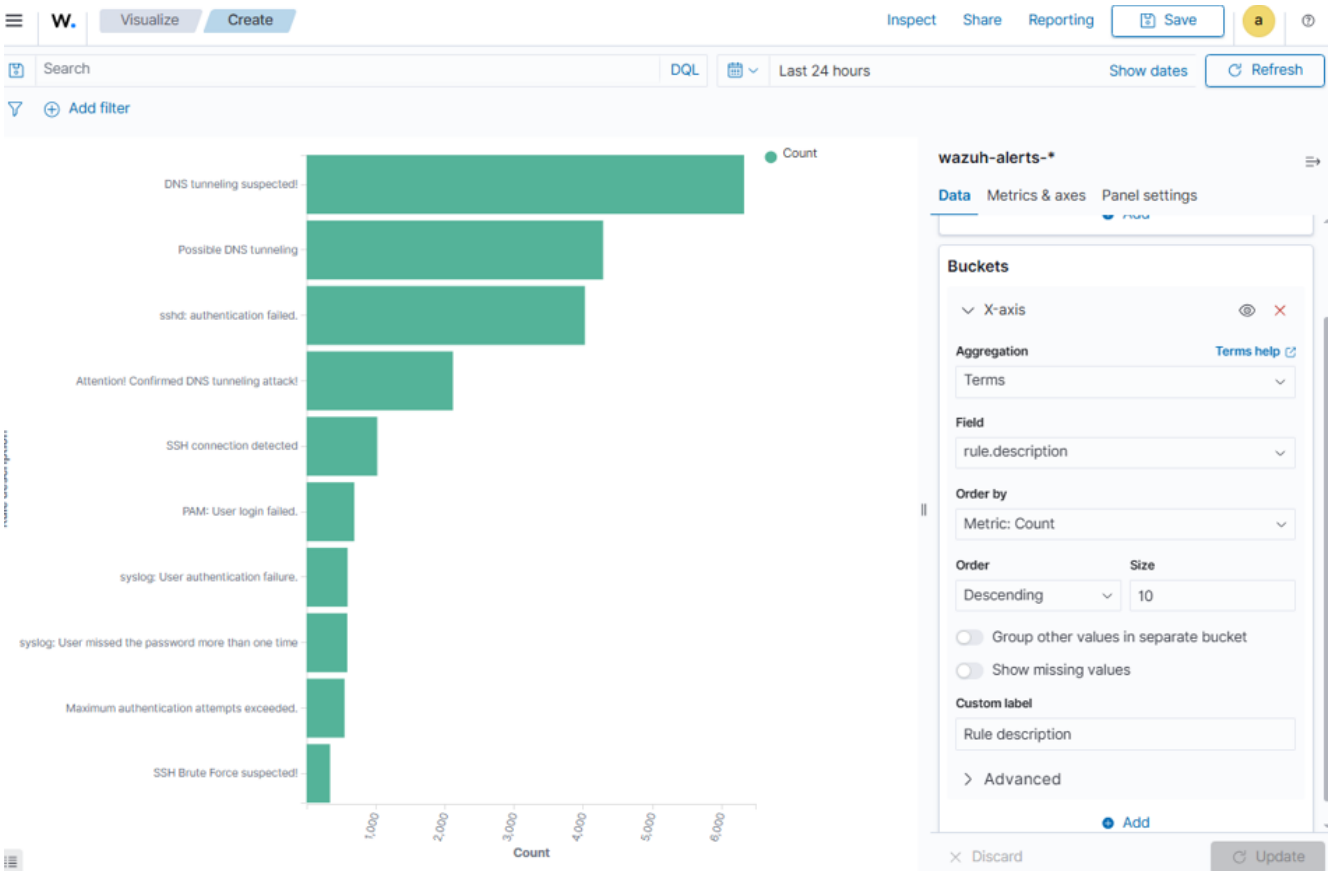


Рисунок Д.7 – Графік «Rules by Description»

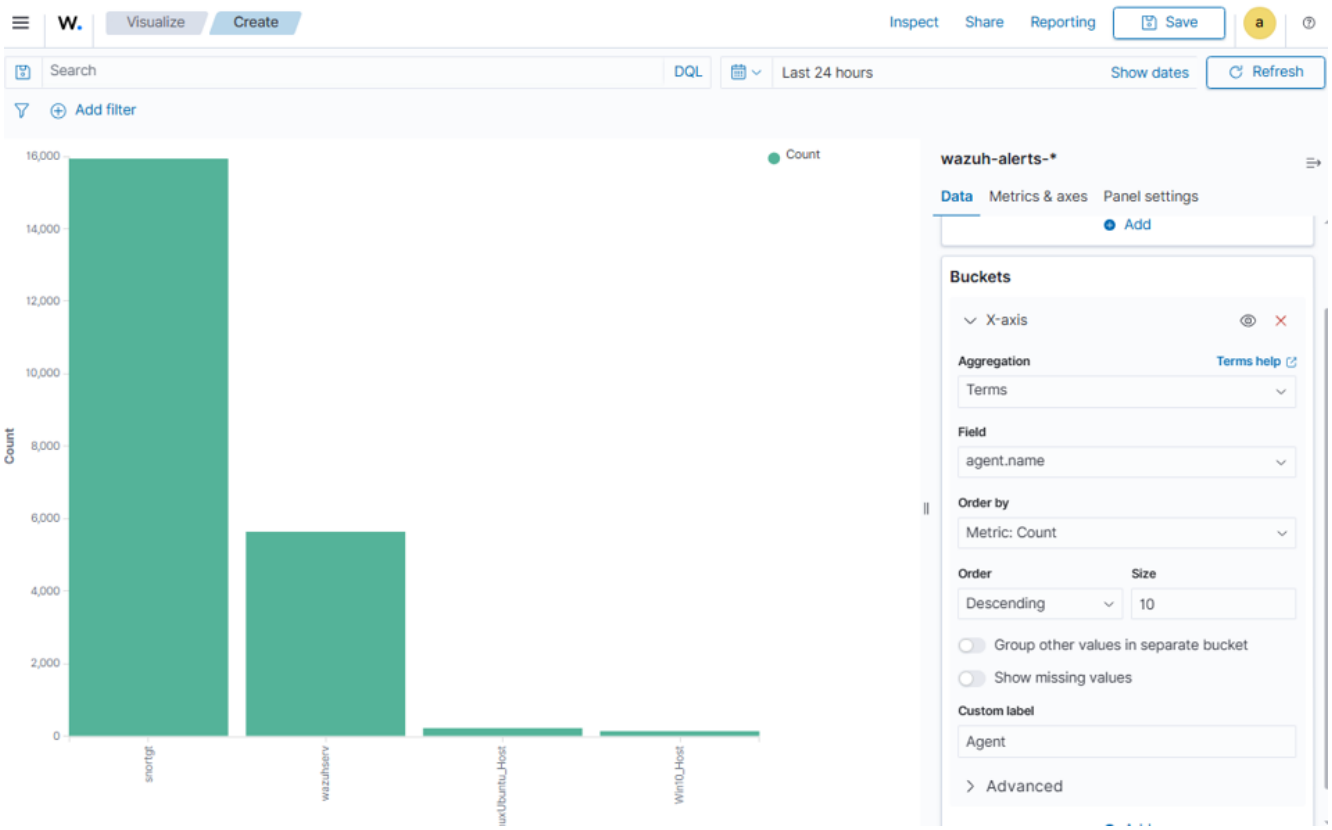


Рисунок Д.8 – Графік «Events by Agent»

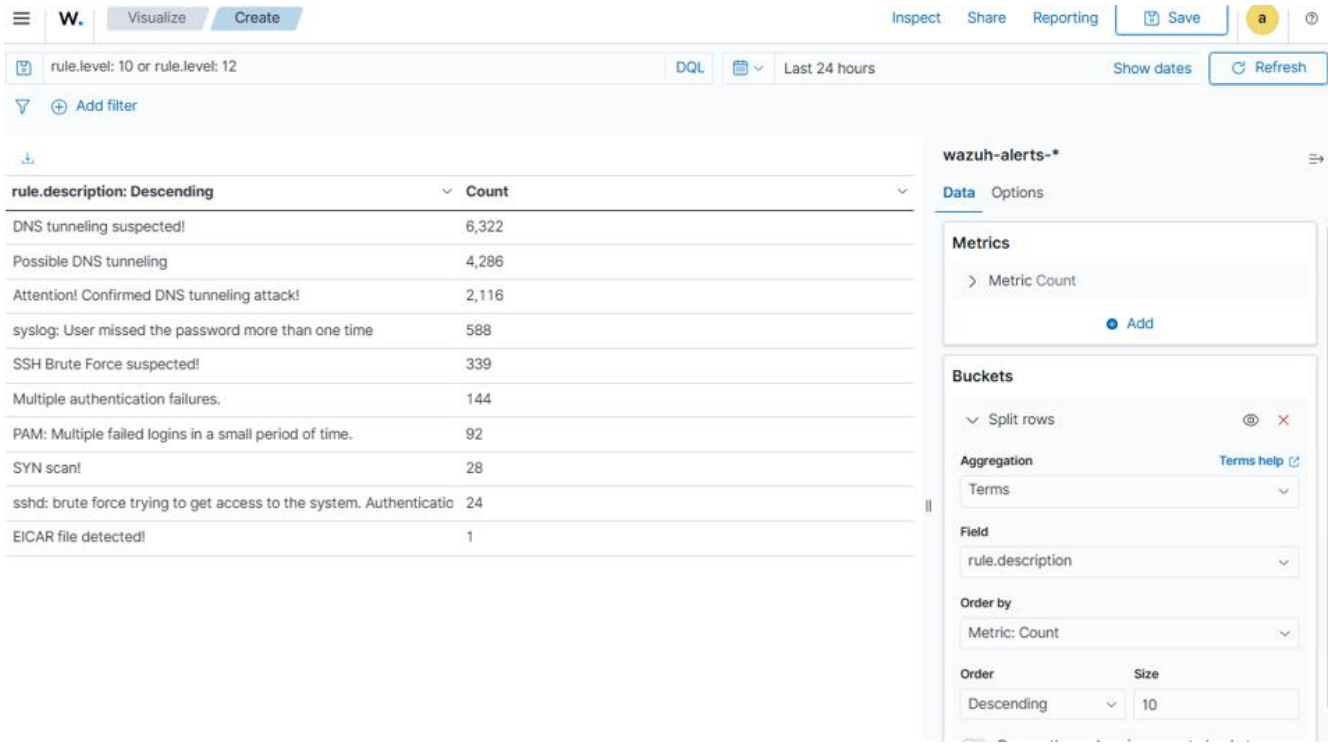


Рисунок Д.9 – Графік «High Severity Alerts»

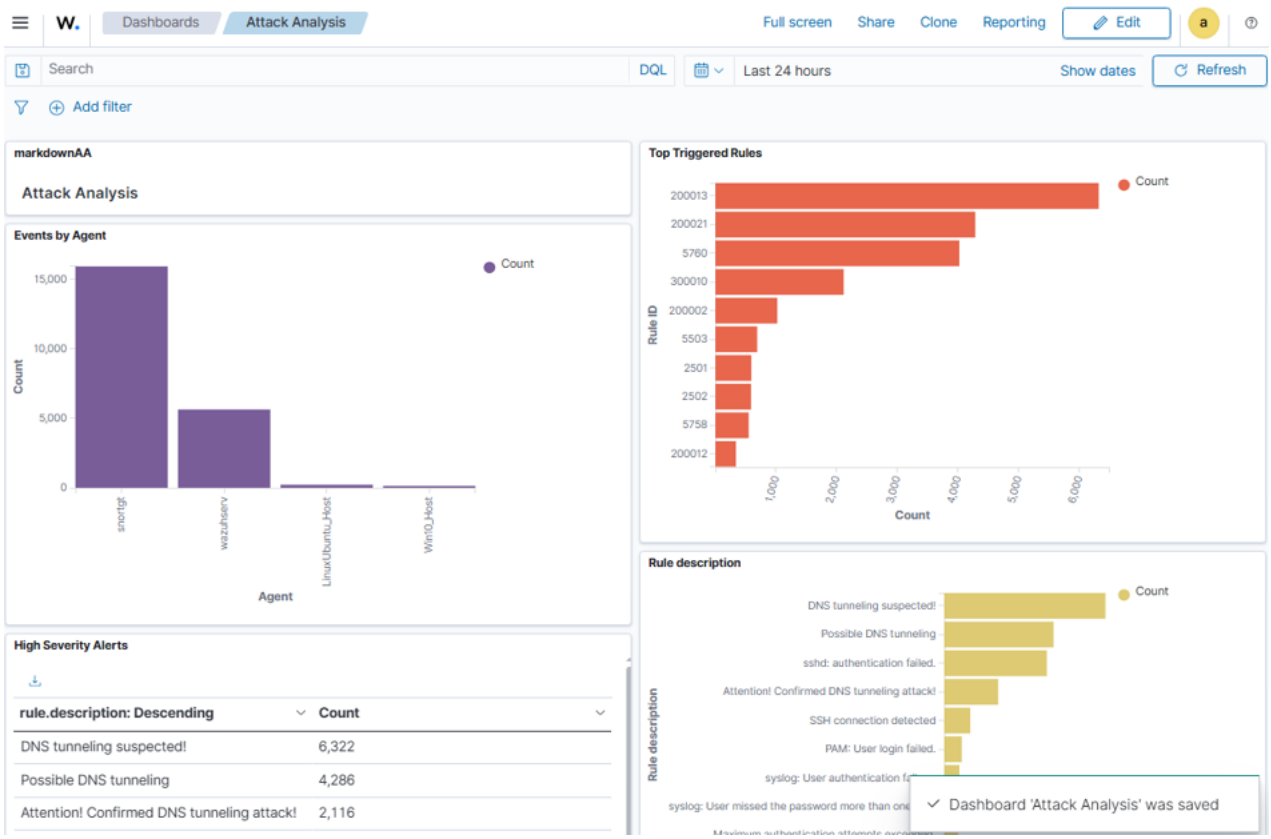


Рисунок Д.10 – Створення дашборду

Дашборд «Attack Chain»

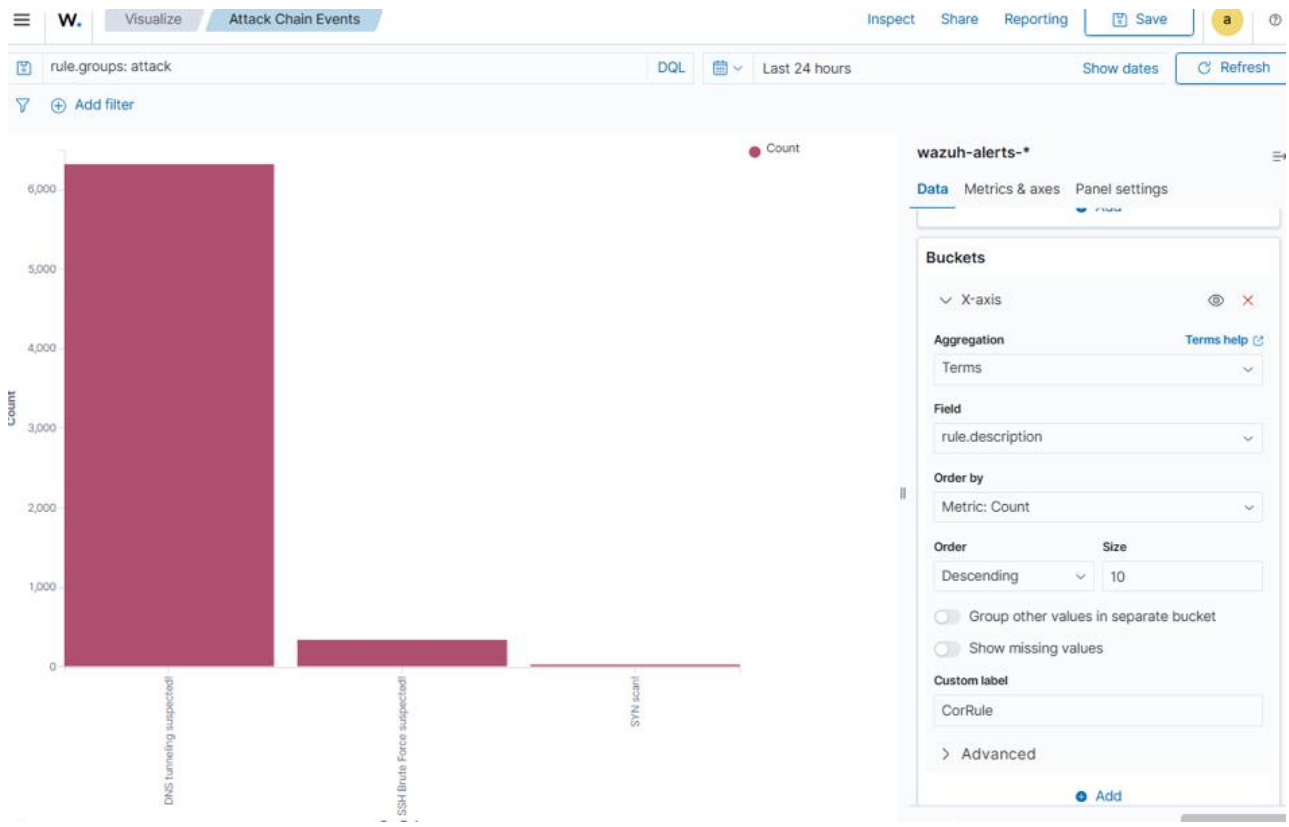


Рисунок Д.11 – Графік «Attack Chain Events»

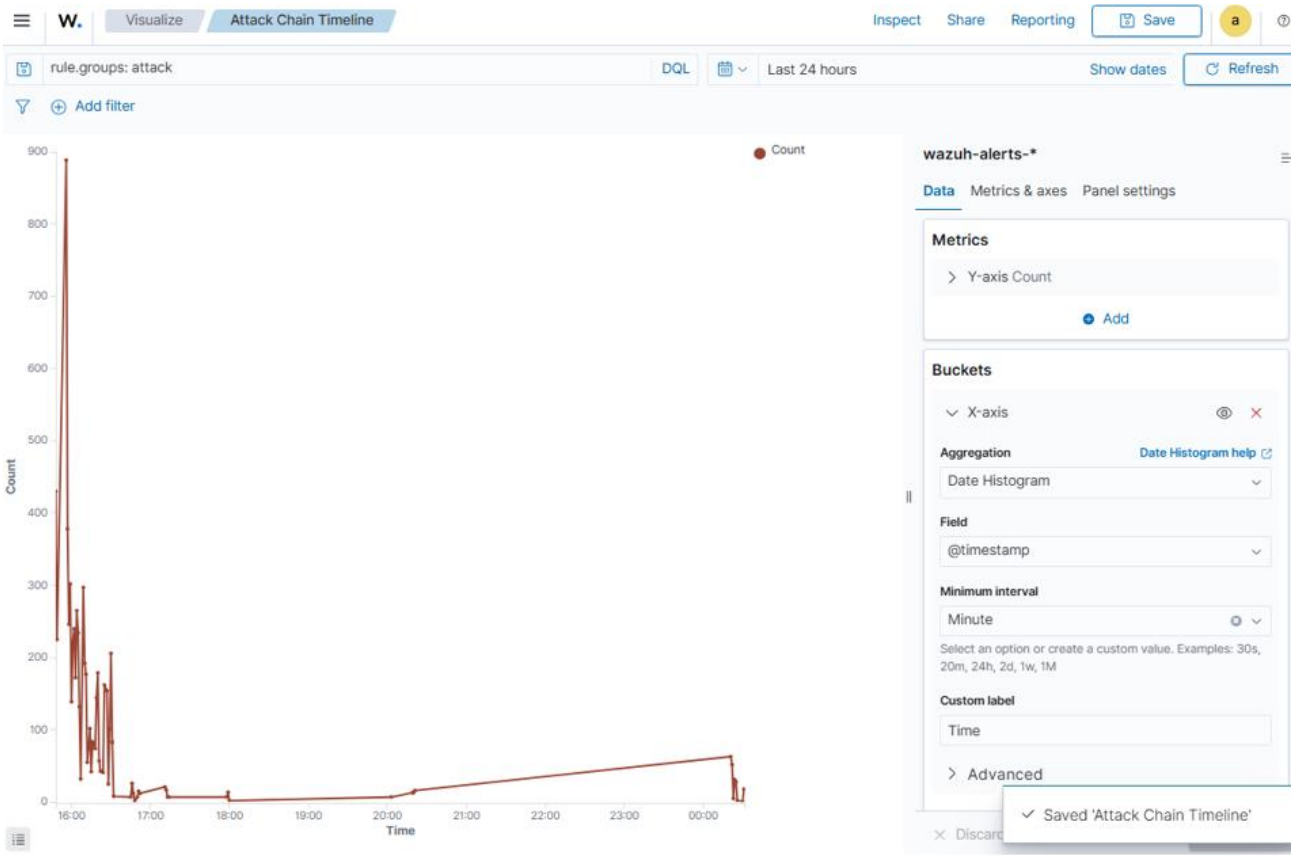


Рисунок Д.12 – Графік «Attack Chain Timeline»

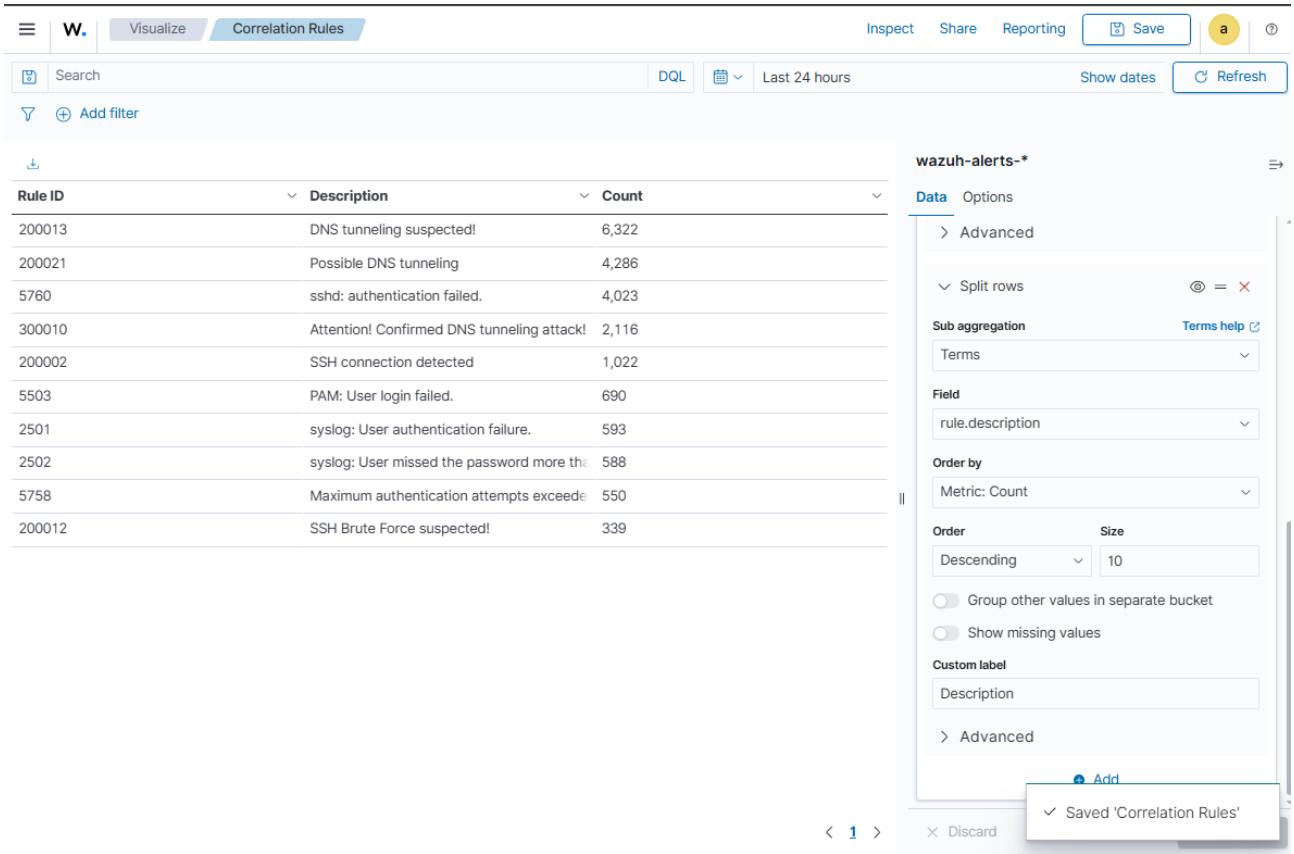


Рисунок Д.13 – Графік «Correlation Rules»

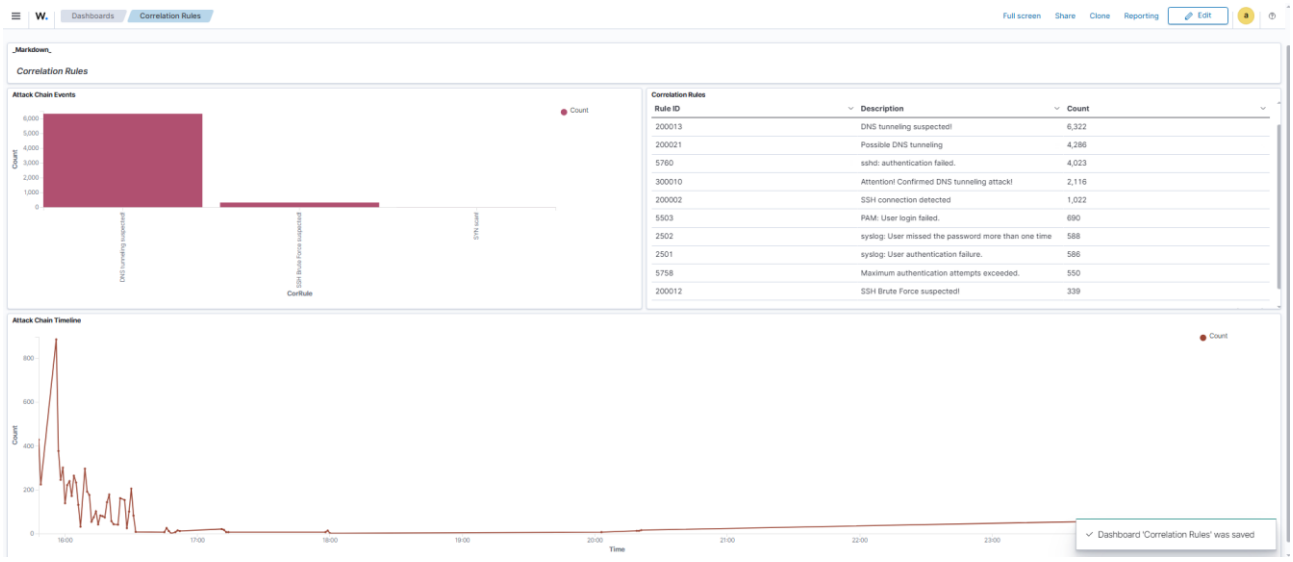


Рисунок Д.14 – Створення дашборда

Додаток Е

Таблиця Е.1.

Результат тестування підсистеми безпекового моніторингу

Тип атаки/ події	Інструмент реалізації	Очікуваний результат	Фактичний результат
ICMP-пінг	Термінал ОС та команда ping	Виявлення ICMP-активності у мережі	Snort зафіксував ехо-запит та передав подію до Wazuh
SSH-підключеннння	PuTTY	Виявлення підключення до SSH-служби	Система сформувала повідомлення про підключення до служби
ICMP-сканування	Термінал ОС та цикл команди ping	Виявлення масового ICMP-сканування	В системі вивело та спрацювало правило виявлення ICMP-сканування
SYN-сканування	Nmap	Виявлення сканування портів	Snort зафіксував велику кількість SYN-пакетів та передав повідомлення до системи
SSH brute force	Hydra	Виявлення багаторазових SSH-підключень	Система виявила атаку типу brute force та сформувала про це повідомлення
DNS-флуд	Вбудований браузер ОС	Виявлення великої кількості DNS-запитів	Система виявила та показала велику кількість DNS-запитів за короткий проміжок часу
Підозра на DNS-тунелювання	Термінал ОС та цикл звернення до веб-сайту	Виявлення аномально великої кількості DNS-запитів, що є можливою ознакою DNS-тунелювання	Система виявила та показала аномальну кількість DNS-запитів за короткий проміжок часу, спрацювало правило «DNS high volume»
Підключення до шкідливого DNS-домену	DNS-запит до домену з Threat Intelligence feed через команду nslookup	Виявлення звернення до небезпечного домену	Спрацювало правило «DNS TI malicious domain», в системі відобразився алерт з детальними відомостями.
Кореляція атак ICMP-сканування + SYN-сканування	Виконання команди ping та інструменту Nmap	Виявлення послідовної мережевої розвідки	Wazuh виконав кореляцію подій та сформував повідомлення про виявлення мережевої розвідки

Кореляція SYN-сканування + SSH brute force	Використання інструменту Nmap + Hydra	Виявлення комбінованої атаки на протокол та службу SSH	Wazuh виконав кореляцію та повідомив про комбіновану атаку
Повний ланцюг атаки (Full Attack Chain)	Використання команди ping та інструментів Nmap і Hydra	Виявлення комплексної атаки	Система миттєво зреагувала та сформувала повідомлення з найвищим рівнем пріорітету про проведення атаки в даний проміжок часу
Кореляція аномальної кількості DNS-запитів	Велика кількість DNS-запитів	Виявлення потенційного DNS-тунелювання	Wazuh провів кореляцію минулих правил про DNS-запити та сформував повідомлення про потенційну загрозу DNS-тунелювання
Підтвердження DNS-тунелювання	Багаторазове повторення події DNS-тунелювання	Підтвердження атаки через протокол DNS	Декілька разів спрацювала минула кореляція про потенційну загрозу DNS-тунелювання, що підтвердило проведення атаки
Виявлення файлу EICAR	Текстовий файл EICAR	Виявлення тестового шкідливого файлу	Wazuh зафіксував подію додання файлу у FIM та Threat Hunting
Активне реагування на EICAR	Функція Active Response у Wazuh	Автоматичне реагування на шкідливий файл та його видалення	Після виявлення файлу системою, надіслано команду на спрацювання скрипту, який автоматично видалив файл EICAR з системи, підтвердивши це повідомленням про видалення
Активне реагування на мережеву атаку	Функція Active Response у Wazuh + «iptables»	Автоматичне реагування на атаку та блокування джерела підозрілої активності	Після спрацювання правил та кореляції подій система автоматично виконала алгоритм блокування атакуючої IP-адреси у мережі використовуючи «iptables»