НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Приладобудівний факультет
Кафедра автоматизації та систем неруйнівного контролю

«На правах рукопису»

УДК 004.056:004.8

До захисту допущено:
Завідувач кафедри
_____ Юрій КИРИЧУК
«___»_____2025 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою «Комп'ютерно-інтегровані системи та технології в приладобудуванні»

зі спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка»

на тему: «Автоматизація процесів моніторингу загроз у комп’ютерно-інтегрованих системах»

Виконав:
студент ІІ курсу, групи ПМ-з41мп
Трохимчук Джаміль Валентинович _____

Науковий керівник:
доцент каф. АСНК, к.т.н.
Богдан Галина Анатоліївна _____

Консультант з розробки стартап-проектів:
завідувач кафедри економічної кібернетики,
д.е.н., професор
Бояринова Катерина Олександрівна _____

Рецензент:
доцент каф. ІВТ, к.т.н.
Маркіна Ольга Миколаївна _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Приладобудівний факультет
Кафедра автоматизації та систем неруйнівного контролю

Рівень вищої освіти – другий (магістерський)

Спеціальність – 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка»

Освітньо-професійна програма «Комп'ютерно-інтегровані системи та технології в приладобудуванні»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Юрій КИРИЧУК
«___»_____20__ р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Трохимчуку Джамілю Валентиновичу

1. Тема дисертації: «Автоматизація процесів моніторингу загроз у комп’ютерно-інтегрованих системах», науковий керівник дисертації Богдан Галина Анатоліївна, доцент, затверджені наказом по університету від «04» листопада 2025 р. №4792-с.

2. Термін подання студентом проєкту: 10.12.2025.

3. Об’єкт дослідження: процеси забезпечення інформаційної безпеки та моніторингу інцидентів у комп’ютерно-інтегрованих системах.

4. Вихідні дані: система для моніторингу загроз у комп’ютерно-інтегрованих системах.

5. Перелік завдань, які потрібно розробити: проаналізувати засоби моніторингу та забезпечення безпеки КІС, архітектуру та принципи роботи SIEM-системи Wazuh для її впровадження; здійснити практичне практичне розгортання та конфігурацію компонентів системи — гіпервізора VMware ESXi, серверної частину Wazuh на базі Ubuntu та гетерогенні клієнтські системи (Windows, Linux) шляхом встановлення та реєстрації агентів; провести експериментальну перевірку ефективності виявлення загроз шляхом моделювання типових загроз безпеки; визначити перспективи інтеграції у промислові середовища та розробити концепцію стартап-проєкту.

6. Орієнтований перелік графічного ілюстративного матеріалу: презентація у вигляді ілюстрацій, що додатково розкривають суть магістерської дисертації

7. Орієнтований перелік публікацій:

Трохимчук Д.В. «АРХІТЕКТУРНІ ПРИНЦИПИ ПОБУДОВИ СИСТЕМИ МОНІТОРИНГУ ПОДІЙ БЕЗПЕКИ В КОМП'ЮТЕРНО-ІНТЕГРОВАНИХ СИСТЕМАХ НА БАЗІ ПЛАТФОРМИ WAZUH» — Збірник наукових праць Національного університету кораблебудування імені адмірала Макарова. Випуск 4, 2025.

8. Консультанти розділів дисертації

Розділ	**Прізвище, ініціали та посада консультанта**	**Підпис, дата**	
		Завдання видав	**Завдання прийняв**
Розробка стартап-проєктів	Завідувач кафедри економічної кібернетики, доктор економічних наук, професор Бояринова Катерина Олександрівна		

9. Дата видачі завдання _____

Календарний план

№ з/п	**Назва етапів виконання дипломного проєкту**	**Термін виконання етапів проєкту**	**Примітка**
1	Узгодження технічного завдання, затвердження теми та написання вступу	01.09.2025 – 10.09.2025	Виконано
2	Аналіз засобів моніторингу та загроз у КІС	11.09.2025 – 25.09.2025	Виконано
3	Розгортання віртуалізованої інфраструктури	26.09.2025 – 17.10.2025	Виконано
4	Інсталяція та конфігурація SIEM Wazuh	18.10.2025 – 31.10.2025	Виконано
5	Проведення експерименту та моделювання атак	01.11.2025 – 07.11.2025	Виконано
6	Аналіз ефективності та перспективи інтеграції	08.11.2025 – 15.11.2025	Виконано
7	Розробка стартап-проєкту	16.11.2025 – 30.11.2025	Виконано
8	Оформлення дисертації та нормоконтроль	01.12.2025 – 09.12.2025	Виконано
9	Подання готової роботи керівнику	10.12.2025	Виконано
10	Перевірка на плагіат, подання готової роботи на кафедру, підготовка презентації та доповіді	11.12.2025 – 21.12.2025	Виконано

Студент Джаміль ТРОХИМЧУК

Керівник Галина БОГДАН

РЕФЕРАТ

Актуальність теми. В умовах стрімкої цифрової трансформації комп’ютерно-інтегровані системи стають фундаментом критичної інфраструктури, забезпечуючи глибоку взаємодію інформаційних та операційних технологій. Проте зростання складності таких систем суттєво розширює поверхню атак, де кіберінциденти можуть спричинити не лише втрату даних, а й фізичні аварії, зупинку виробництва та загрозу безпеці персоналу. Традиційні засоби захисту часто виявляються неефективними через розрізненість даних та проблему інформаційного перевантаження персоналу хибними спрацюваннями. Тому автоматизація моніторингу шляхом впровадження SIEM-систем є критично важливою задачею. Це дозволяє централізувати збір подій, забезпечити їх кореляцію в реальному часі та підвищити стійкість технологічних процесів до сучасних загроз.

Мета і задачі дослідження. Метою роботи є підвищення ефективності захисту комп’ютерно-інтегрованих систем шляхом розробки та впровадження автоматизованої системи моніторингу загроз на базі SIEM-технологій.

Для досягнення мети вирішено наступні **задачі**:

1. Проведено аналіз засобів моніторингу та забезпечення безпеки комп’ютерно-інтегрованих систем.

2. Проаналізовано архітектуру та принципи роботи SIEM-системи Wazuh для її впровадження.

3. Здійснено практичне розгортання та конфігурацію компонентів системи: гіпервізор VMware ESXi, серверну частину Wazuh на базі Ubuntu та гетерогенні клієнтські системи (Windows, Linux) шляхом встановлення та реєстрації агентів.

4. Розроблено сценарії та проведено моделювання типових загроз безпеки, зокрема невдалих спроб авторизації, для експериментальної перевірки працездатності системи.

5. Проаналізовано результати експерименту: дослідити спрацьовування правил кореляції в Wazuh, згенеровані журнали подій та надано оцінку ефективності системи щодо виявлення загроз.

6. Визначено перспективи розвитку запропонованого рішення та досліджено можливості його інтеграції у промислові комп'ютерно-інтегровані середовища.

7. Розроблено концепцію стартап-проєкту для комерціалізації рішення з автоматизації моніторингу, що включає аналіз ринку, розробку ринкової стратегії та маркетингової програми, бізнес-моделі реалізації стартап-проєкту та оцінювання його економічної ефективності.

Об'єкт дослідження — процеси забезпечення інформаційної безпеки та моніторингу інцидентів у комп'ютерно-інтегрованих системах.

Предмет дослідження — методи та засоби автоматизації виявлення кіберзагроз із використанням SIEM-системи Wazuh у віртуалізованому середовищі.

Методи дослідження. У роботі використано: методи системного аналізу (для дослідження архітектури SIEM та загроз КІС); методи віртуалізації (для розгортання дослідного стенду на VMware ESXi); методи імітаційного моделювання (для відтворення сценаріїв атак); експериментальні методи (для перевірки працездатності системи та аналізу журналів подій); методи економічного аналізу та маркетингового планування (для розробки стартап-проєкту).

Наукова новизна одержаних результатів:

1. Удосконалено архітектуру системи моніторингу безпеки для КІС шляхом інтеграції SIEM-системи Wazuh у віртуалізоване середовище VMware ESXi, що забезпечує гнучкість розгортання та масштабування без значних апаратних витрат.

2. Дістало подальшого розвитку застосування методів кореляції подій у гетерогенних системах, що дозволяє автоматизувати виявлення загроз у реальному часі на рівні розподілених агентів.

Практичне значення одержаних результатів полягає у створенні діючого прототипу автоматизованої системи моніторингу, готового до інтеграції в інфраструктуру підприємств. Розроблена концепція стартап-проєкту доводить економічну ефективність використання даного рішення для забезпечення кіберстійкості комп'ютерно-інтегрованих систем.

Ключові слова: моніторинг загроз, комп'ютерно-інтегровані системи, автоматизація, інформаційні технології, операційні технології, SIEM, Wazuh.

ABSTRACT

Relevance of the topic. In the context of rapid digital transformation, computer-integrated systems are becoming the foundation of critical infrastructure, ensuring deep interaction between informationand operational technologies. However, the increasing complexity of such systems significantly expands the attack surface, where cyber incidents can cause not only data loss but also physical accidents, production stoppages, and threats to personnel safety. Traditional protection tools often prove ineffective due to data fragmentation and the problem of information overload on personnel caused by false positives. Therefore, automating monitoring through the implementation of SIEM systems is a critical task. This allows for centralized event collection, real-time correlation, and increased resilience of technological processes against modern threats.

Aim and tasks of the research. The aim of the work is to increase the efficiency of protection for computer-integrated systems by developing and implementing an automated threat monitoring system based on SIEM technologies.

To achieve this aim, the following tasks were resolved:

1. An analysis of monitoring tools and security assurance for computer-integrated systems was conducted.

2. The architecture and operating principles of the Wazuh SIEM system were analyzed for its implementation.

3. Practical deployment and configuration of system components were carried out: the VMware ESXi hypervisor, the Wazuh server part based on Ubuntu, and heterogeneous client systems (Windows, Linux) were integrated by installing and registering agents.

4. Scenarios were developed, and simulation of typical security threats, particularly failed authorization attempts, was conducted for experimental verification of the system's performance.

5. The results of the experiment were analyzed: the triggering of correlation rules in Wazuh and generated event logs were investigated, and an assessment of the system's efficiency regarding threat detection was provided.

6. The prospects for the development of the proposed solution were determined, and possibilities for its integration into industrial computer-integrated environments were investigated.

7. A concept for a startup project was developed to commercialize the automated monitoring solution, which includes market analysis, development of a market strategy and marketing program, a business model for the startup implementation, and an assessment of its economic efficiency.

Object of research — processes of information security assurance and incident monitoring in computer-integrated systems.

Subject of research — methods and tools for automated cyber threat detection using the Wazuh SIEM system in a virtualized environment.

Research methods. The work uses: systems analysis methods (to investigate SIEM architecture and CIS threats); virtualization methods (to deploy the test bench on VMware ESXi); simulation modeling methods (to reproduce attack scenarios); experimental methods (to verify system performance and analyze event logs); methods of economic analysis and marketing planning (to develop the startup project).

Scientific novelty of the obtained results:

1. The architecture of the security monitoring system for CIS has been improved by integrating the Wazuh SIEM system into the VMware ESXi virtualized environment, which ensures deployment flexibility and scalability without significant hardware costs.

2. The application of event correlation methods in heterogeneous systems has been further developed, allowing for automated real-time threat detection at the level of distributed agents.

Practical value of the obtained results lies in the creation of a working prototype of an automated monitoring system, ready for integration into enterprise infrastructure. The developed concept of the startup project proves the economic efficiency of using this solution to ensure the cyber resilience of computer-integrated systems.

Key words: threat monitoring, computer-integrated systems, automation, information technology, operational technology, SIEM, Wazuh.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

КІС	—	Комп’ютерно-інтегровані системи
API	—	Application Programming Interface (Інтерфейс прикладного програмування)
APT	—	Advanced Persistent Threat (Складна постійна загроза)
Apps	—	Applications (Додатки)
AU	—	Audit and Accountability (Аудит та Підзвітність)
CIA	—	Confidentiality, Integrity, and Availability (Конфіденційність, Цілісність, Доступність)
CISA	—	Cybersecurity and Infrastructure Security Agency (Агентство з кібербезпеки та захисту інфраструктури США)
CVE	—	Common Vulnerabilities and Exposures (Загальні вразливості та ризики)
DCS	—	Distributed Control Systems (Розподілені системи управління)
EDR	—	Endpoint Detection and Response (Виявлення та реагування на кінцевих точках)
ELK Stack	—	Elasticsearch, Logstash, Kibana (Стек технологій для пошуку та аналітики)
EPS	—	Events Per Second (Події за секунду)
EWS	—	Engineering Workstation (Інженерна станція)
FIM	—	File Integrity Monitoring (Моніторинг цілісності файлів)
HMI	—	Human-Machine Interface (Людино-машинний інтерфейс)
IACS	—	Industrial Automation and Control Systems (Системи промислової автоматизації та управління)
ICS	—	Industrial Control Systems (Промислові системи управління)

IEC	—	International Electrotechnical Commission (Міжнародна електротехнічна комісія)
IIoT	—	Industrial Internet of Things (Промисловий інтернет речей)
IT	—	Information Technology (Інформаційні технології)
MITRE ATT&CK	—	(Adversarial Tactics, Techniques, and Common Knowledge) for ICS (Фреймворк тактик та технік зловмисників)
NIST	—	National Institute of Standards and Technology (Національний інститут стандартів і технологій США)
NetFlow	—	(Протокол мережевого моніторингу, розроблений Cisco)
NVD	—	National Vulnerability Database (Національна база даних вразливостей)
OT	—	Operational Technology (Операційні технології)
PID	—	Process Identifier (Ідентифікатор процесу)
PLC	—	Programmable Logic Controller (Програмований логічний контролер)
QFlow	—	QRadar Flow (Компонент аналізу мережевих потоків IBM QRadar)
QVM	—	QRadar Vulnerability Manager (Менеджер вразливостей IBM QRadar)
RTOS	—	Real-Time Operating System (Операційна система реального часу)
RTU	—	Remote Terminal Unit (Виносний термінал / Пристрій телеметрії)
SCADA	—	Supervisory Control and Data Acquisition (Системи диспетчерського управління та збору даних)
SEM	—	Security Event Management (Управління подіями безпеки)
SI	—	System and Information Integrity (Цілісність Системи та Інформації)
SIEM	—	Security Information and Event Management (Управління інформацією про безпеку та подіями)

SIM	—	Security Information Management (Управління інформацією про безпеку)
SIS	—	Safety Instrumented System (Система Інструментальної Безпеки)
SNMP	—	Simple Network Management Protocol (Простий протокол мережевого управління)
SOAR	—	Security Orchestration, Automation and Response (Оркестрація, автоматизація та реагування на загрози безпеки)
SOC	—	Security Operations Center (Центр управління безпекою)
SPAN	—	Switched Port Analyzer (Аналізатор порту комутатора)
SPL	—	Search Processing Language (Мова обробки пошукових запитів Splunk)
TCO	—	Total Cost of Ownership (Загальна вартість володіння)
UEBA	—	User and Entity Behavior Analytics (Аналітика поведінки користувачів та сутностей)
UFA	—	Universal Forwarding Agent (Універсальний агент пересилання Splunk)
UI	—	User Interface (Інтерфейс користувача)
USB	—	Universal Serial Bus (Універсальна послідовна шина)
VD	—	Vulnerability Detection (Виявлення вразливостей)
WMI	—	Windows Management Instrumentation (Інструментарій керування Windows)
XDR	—	Extended Detection and Response (Розширене виявлення та реагування)
X-Pack	—	Пакет розширень для Elastic Stack

ВСТУП

Стрімкий розвиток цифрових технологій та масштабне впровадження автоматизованих систем керування у промисловості, енергетиці, транспорті та сфері критичної інфраструктури призводять до глибокої трансформації сучасних виробничих процесів. Комп’ютерно-інтегровані системи (КІС) стають ключовим елементом цієї трансформації, забезпечуючи високу точність, оперативність та надійність керування технологічними об’єктами. У таких системах забезпечується взаємодія різнорідних інформаційних і технологічних компонентів, об'єднаних у єдину інфраструктуру, що дозволяє досягати високого рівня автоматизації та оптимізації бізнес- і виробничих процесів.

Разом з тим, зростання складності КІС супроводжується збільшенням поверхні атаки та загальної вразливості технологічних середовищ. Сучасні кіберзагрози стають усе більш витонченими, поліморфними та цілеспрямованими, а їх наслідки можуть виходити далеко за межі інформаційної сфери. Атаки на промислові системи здатні впливати не лише на дані, а й на реальні технологічні процеси, що створює ризики порушення виробництва, аварійної зупинки обладнання, фізичних ушкоджень чи навіть загрози життю людей. Такі інциденти, як атаки на енергетичні системи, транспортні вузли чи об'єкти хімічної промисловості, демонструють, наскільки критичною є необхідність створення ефективних механізмів кіберзахисту саме для комп’ютерно-інтегрованих систем.

В умовах постійно зростаючого обсягу даних, високої швидкості кібератак та потреби в безперервному моніторингу традиційні підходи до виявлення та аналізу подій безпеки виявляються недостатніми. Окремі засоби захисту — такі як системи виявлення вторгнень, антивіруси, міжмережеві екрани — генерують величезні масиви подій, але не забезпечують їх комплексного аналізу та кореляції. Внаслідок цього організації стикаються з проблемами інформаційного перевантаження, затримок у виявленні загроз,

складності ідентифікації реальних інцидентів серед значної кількості хибнопозитивних спрацювань.

Одним із найефективніших інструментів вирішення цих проблем є системи управління інформацією та подіями безпеки — SIEM (Security Information and Event Management). Ці системи дозволяють централізовано збирати, зберігати, нормалізувати та аналізувати дані з різних джерел, забезпечуючи єдину панель моніторингу стану кібербезпеки. Вони застосовують методи кореляції подій, поведінкового аналізу, статистичної аналітики для виявлення аномалій і потенційних атак у реальному часі.

У контексті КІС можливості SIEM мають особливе значення, оскільки дозволяють об’єднати інформацію з IT- та OT-сегментів, які часто мають відмінні протоколи, вимоги до доступності та принципи організації.

З огляду на це, розроблення ефективних, масштабованих і адаптивних рішень у сфері моніторингу безпеки є надзвичайно актуальним. Інтеграція SIEM у КІС дозволяє не лише підвищити рівень захищеності технологічної інфраструктури, але й створити передумови для побудови комплексної системи управління кіберризиками, здатної забезпечити безперервність виробництва та відповідність міжнародним стандартам.

РОЗДІЛ 1. АНАЛІЗ ЗАСОБІВ МОНІТОРИНГУ ТА ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ КОМП'ЮТЕРНО-ІНТЕГРОВАНИХ СИСТЕМ

1.1 Загальні принципи моніторингу подій безпеки у комп'ютерно-інтегрованих системах

Моніторинг подій безпеки в комп'ютерно-інтегрованих системах фундаментально відрізняється від традиційного моніторингу в IT-середовищах. Ці відмінності зумовлені унікальними операційними вимогами, пріоритетами та архітектурними обмеженнями промислових мереж.

Основоположна відмінність полягає в пріоритетах безпеки. У класичних IT-системах домінує тріада CIA: Конфіденційність, Цілісність та Доступність. Для операційних технологій ця тріада фактично інвертована та розширена. На перше місце виходить Фізична Безпека (Safety) – гарантія того, що система не завдасть шкоди людям, навколишньому середовищу або обладнанню. Далі йдуть Доступність та Безперервність технологічного процесу, оскільки навіть незначний простій може призвести до мільйонних збитків або порушення роботи критичної інфраструктури [1]. Цілісність даних (наприклад, команд, що віддаються контролеру) є важливою, тоді як конфіденційність часто має найнижчий пріоритет.

Ця інверсія пріоритетів генерує низку практичних викликів для моніторингу:

1. Неприпустимість затримок: системи OT, особливо системи реального часу (RTOS), є надзвичайно чутливими до затримок мережевого трафіку. Інструменти моніторингу, що вносять навіть мілісекундні затримки, можуть порушити синхронізацію технологічного процесу [2].

2. Загроза від активного сканування: традиційні IT-методи моніторингу, такі як активне сканування портів або вразливостей, є неприйнятними в OT-мережах. Вони можуть бути помилково інтерпретовані застарілими пристроями (PLC, RTU) як аномальні запити, що призведе до їх

відмови або перезавантаження [2]. Це змушує покладатися переважно на пасивні методи моніторингу (наприклад, аналіз SPAN-портів).

3. Проблема застарілого обладнання: життєвий цикл обладнання в КІС може сягати 20-30 років. Це означає, що в експлуатації досі перебуває значна кількість пристроїв, які працюють на застарілих операційних системах (напр., Windows XP, Windows 7) і не підлягають оновленню (патчингу). Встановлення на такі системи сучасних агентів безпеки є або технічно неможливим, або заборонено виробником обладнання [3].

1.1.1 Нормативно-правова база та стандартизація моніторингу ОТ

Через високу критичність КІС, їх моніторинг та захист суворо регламентується низкою міжнародних та національних стандартів. Ключовими серед них є NIST SP 800-82 та серія IEC 62443.

NIST Special Publication 800-82

Публікація Національного інституту стандартів і технологій США (NIST) SP 800-82 "Guide to Operational Technology Security" є де-факто "золотим стандартом" для захисту промислових систем [4]. Остання, третя ревізія (r3), опублікована у 2023 році, значно розширила фокус з ICS на весь спектр операційних технологій [5].

Ключовим нововведенням NIST SP 800-82r3 є формалізація концепції "OT overlay" (накладення профілю ОТ) [6]. Цей підхід бере за основу загальний каталог контролів безпеки NIST SP 800-53 і адаптує його для специфічних потреб ОТ-середовищ [7].

Для моніторингу загроз, найбільш релевантними є два сімейства контролів:

1. AU (Audit and Accountability/Аудит та Підзвітність), зокрема, контроль AU-2 "Event Logging" (Журналювання подій).

2. SI (System and Information Integrity/Цілісність Системи та Інформації), зокрема, контроль SI-4 "Information Integrity Monitoring" (Моніторинг цілісності інформації) [5].

Найважливішим внеском стандарту є визнання проблеми застарілих систем та впровадження поняття "Compensating Controls" (Компенсаційних Контролів). Стандарт прямо визнає, що багато OT-пристроїв (наприклад, PLC) технічно не здатні генерувати детальні журнали аудиту, які б відповідали вимогам AU-2 [7].

У таких випадках NIST рекомендує впроваджувати компенсаційні заходи. Замість того, щоб намагатися моніторити "немоніторований" пристрій, моніторинг переноситься на ті компоненти системи, які можна контролювати. Наприклад, для захисту PLC компенсаційні контроли можуть включати [7]:

– Моніторинг журналів брандмауера та комутатора, що контролюють мережевий доступ до цього PLC.

– Моніторинг інженерної станції (EWS) або HMI-сервера, які мають легітимний доступ для програмування чи конфігурування цього PLC.

Цей підхід є критично важливим, оскільки він зміщує фокус моніторингу з неможливого (агент на PLC) на реалістичне (агент на EWS/HMI та аналіз мережевих журналів).

Серія стандартів IEC 62443

Якщо NIST SP 800-82 пропонує каталог контролів, то серія стандартів IEC 62443 (особливо актуальна для Європи) пропонує архітектурний підхід та життєвий цикл безпеки для "Industrial Automation and Control Systems" (IACS) [8].

IEC 62443 впроваджує два фундаментальні принципи:

1. Security Lifecycle (Життєвий цикл безпеки) та Risk-Based Approach (Підхід на основі ризиків): стандарт вимагає, щоб безпека вбудовувалася в систему на етапі проектування, а не додавалася пізніше, і щоб заходи безпеки базувалися на формальній оцінці ризиків [9].

2. Архітектурна концепція "Zones and Conduits" (Зони та Канали) — це найвідоміша частина стандарту [9].

– Зона — це логічна або фізична група активів (пристроїв, систем), які мають спільні вимоги до безпеки. Наприклад, усі контролери, що відповідають за один технологічний процес, можуть бути об'єднані в одну зону.

– Канал — це будь-який шлях комунікації між двома або більше зонами. Важливо, що всі комунікації між зонами мають проходити виключно через визначені та контрольовані канали [9].

Імплікація цього підходу для моніторингу є колосальною. Замість того, щоб намагатися моніторити кожен окремий пристрій у зоні (що може бути неможливо), основні зусилля з моніторингу (наприклад, встановлення мережевих сенсорів, файрволів з функцією Deep Packet Inspection) зосереджуються на "каналах" [10]. Це дозволяє контролювати 100% трафіку, що входить у зону або виходить з неї, реалізуючи таким чином ефективну сегментацію та моніторинг.

1.1.2 Синтез принципів моніторингу КІС

Ефективна стратегія моніторингу КІС не може покладатися виключно на ІТ-підходи і має бути гібридною. Вона повинна поєднувати найкращі практики з обох ключових стандартів.

Ідеальна стратегія починається з архітектурного проектування відповідно до IEC 62443 [9], де система логічно сегментується на "Зони та Канали". Це створює контрольовані точки для моніторингу.

Потім, на цю архітектуру накладаються конкретні контроли з NIST SP 800-82r3 [11]. Для сучасних пристроїв (сервери, HMI, EWS) впроваджуються прямі контроли (журналювання, моніторинг цілісності). Для застарілих пристроїв (PLC, RTU) впроваджуються компенсаційні контроли [7], що включають моніторинг мережевого трафіку в "каналах" (які ми визначили за IEC 62443) та моніторинг тих систем, що мають до них доступ.

Така багатошарова, гібридна стратегія вимагає потужного центрального інструменту, здатного агрегувати та аналізувати дані з усіх цих різнорідних джерел (мережеві журнали, журнали EWS, дані моніторингу

цілісності). Цим центральним інструментом стають системи класу SIEM, які є не просто корисними, а архітектурно необхідними для реалізації сучасної стратегії моніторингу КІС [12].

1.2 Основні види загроз та їх вплив на комп'ютерно-інтегровані системи

Ландшафт загроз для КІС еволюціонував від теоретичних ризиків до доведених на практиці атак з катастрофічними фізичними наслідками. Конвергенція ІТ та ОТ не лише створила нові вектори атак, але й зробила промислові мережі видимими для зловмисників [13].

На відміну від ІТ-середовищ, де наслідком успішної атаки є переважно втрата даних, фінансові збитки або репутаційна шкода, атаки на КІС мають прямий фізичний вимір. Вони можуть призвести до зупинки критичних процесів (наприклад, електропостачання), пошкодження дорогого промислового обладнання та, що найважливіше, до травм або загибелі людей [14].

1.2.1 Ключові вразливості КІС

Промислові системи мають низку притаманних їм вразливостей, що накопичувалися десятиліттями через парадигму "безпеки через ізоляцію". Аналіз звітів CISA [15] та академічних досліджень [16] дозволяє виділити найбільш поширені та небезпечні категорії вразливостей:

1. Неналежна перевірка вхідних даних: багато промислових протоколів та пристроїв не перевіряють належним чином вхідні дані, що робить їх вразливими до атак типу "переповнення буфера", які можуть призвести до віддаленого виконання коду [15].

2. Слабкі механізми автентифікації та управління обліковими даними: найпоширеніша проблема, яка включає використання паролів за замовчуванням, жорстко закодованих паролів у прошивці, відсутність

механізмів автентифікації для критичних команд протоколу, або використання простих паролів, що передаються у відкритому вигляді [17].

3. Помилки конфігурації та слабкі політики безпеки: це включає відсутність належної мережевої сегментації (пласкі мережі), надмірно дозвільні правила брандмауерів, а також підключення інженерних станцій одночасно до ОТ-мережі та інтернету [15].

1.2.2 Класифікація загроз за допомогою MITRE ATT&CK

Для систематизації та аналізу поведінки зловмисників у промислових середовищах використовується спеціалізований фреймворк MITRE ATT&CK. Ця класифікація є критично необхідною для розуміння того, що дії зловмисника в КІС виходять за межі звичайного "злому сервера" і часто спрямовані на маніпуляцію фізичними процесами [18].

У таблиці 1.1 наведено основні тактики та техніки, що демонструють специфіку загроз саме для КІС.

Таблиця 1.1 – Тактики зловмисників у матриці MITRE ATT&CK for ICS

Тактика (Мета)	**Приклади технік (Як?)**	**Опис впливу на КІС**
Первинний доступ (Initial Access)	Експлуатація публічних додатків, компрометація ланцюга постачання, зараження через знімні носії	Проникнення у периметр ОТ, часто через ІТ-системи, заражені USB-носії або компрометацію обладнання постачальника
Виконання (Execution)	Зміна завдань контролера, використання командного рядка	Запуск шкідливого коду або легітимних інструментів на HMI, EWS або безпосередньо в контролері для маніпуляції системою
Закріплення (Persistence)	Модифікація програми; Зміна системної прошивки	Забезпечення того, що атака переживе перезавантаження системи, часто шляхом модифікації прошивки контролера або файлів проекту HMI

Продовження таблиці 1.1

Тактика (Мета)	**Приклади технік (Як?)**	**Опис впливу на КІС**
Ухилення (Evasion)	Підміна звітних повідомлень, руткіти	Приховування шкідливої активності. Наприклад, надсилання хибних (нормальних) значень на HMI оператора, в той час як реальний процес виведено з-під контролю
Горизонтальне переміщення (Lateral Movement)	Експлуатація віддалених сервісів, використання облікових даних за замовчуванням	Переміщення з одного вузла (наприклад, HMI) на інший (наприклад, сервер історії або інший PLC) в межах мережі ОТ
Збір даних (Collection)	Перехоплення мережевого трафіку, вивантаження програми з контролера	Збір інформації про технологічний процес, конфігурацію PLC та архітектуру мережі для підготовки до фази "Вплив"
Блокування функції реагування (Inhibit Response Function)	Придушення тривог, зупинка сервісів, знищення даних	Вимкнення систем безпеки, тривог (alarms) або historian, щоб оператори не могли адекватно зреагувати на інцидент
Порушення управління процесом (Impair Process Control)	Модифікація параметрів, несанкціоновані командні повідомлення	Пряма зміна параметрів технологічного процесу (наприклад, тиску, швидкості обертання, температури) шляхом надсилання шкідливих команд
Вплив (Impact)	Пошкодження майна, втрата безпеки, відмова в управлінні	Кінцевий результат атаки: фізичне руйнування обладнання, зупинка процесу або створення небезпечної ситуації для персоналу

Джерело: складено на основі [19]

1.2.3 Аналіз прецедентів: кіберфізичні атаки на КІС

Аналіз реальних атак є ключовим для розуміння пріоритетів моніторингу. Три атаки: Stuxnet, CrashOverride та Triton — демонструють чітку та тривожну еволюцію загроз. Узагальнена характеристика цих атак наведена у таблиці 1.2.

Таблиця 1.2 – Аналіз ключових кіберфізичних атак на КІС

Характеристика	Stuxnet (2010)	CrashOverride / Industroyer (2016)	Triton / Trisis (2017)
Цільовий об'єкт	Ядерна програма Ірану (завод зі збагачення урану)	Енергосистема України (підстанція "Північна")	Нафтохімічний завод на Близькому Сході
Цільова система	Програмовані логічні контролери (PLC) Siemens S7-300/400	Обладнання підстанцій (керування вимикачами)	Система інструментальної безпеки (SIS) Schneider Electric Triconex
Мета атаки	Саботаж: фізичне руйнування центрифуг для збагачення урану	Переривання сервісу: масштабне відключення електроенергії	Створення аварійної ситуації: нейтралізація систем захисту для спричинення фізичної катастрофи
Механізм проникнення	Заражені USB-носії, експлуатація вразливостей Windows (0-day)	Фішинг, захоплення віддаленого доступу до IT-мережі	Компрометація інженерної станції (EWS) через IT-мережу
Ключова техніка / Особливість	Evasion (Ухилення): Перехоплення даних від PLC до HMI і підміна їх на "нормальні", приховуючи аварію від оператора	Attack on Protocol: Використання легітимних команд промислових протоколів (IEC 61850, IEC 104) для вимкнення світла без використання вразливостей	Attack on Safety: Модифікація логіки контролера безпеки. Атака виглядала як легітимне оновлення прошивки
Наслідки	Фізичне пошкодження близько 1000 центрифуг	Відключення електроенергії у частині Києва на 1 годину	Зупинка заводу (через помилку в коді вірусу), але потенціал для вибуху/жертв був критичним

Джерело: складено на основі [20-28]

Аналіз цих трьох атак демонструє чітку та небезпечну еволюцію цілей зловмисників:

1. Stuxnet (2010) — саботаж технологічного процесу.

2. CrashOverride (2016) — масштабне переривання критичного сервісу.

3. Triton (2017) — нейтралізація системи безпеки для створення умов для фізичної катастрофи та людських жертв [28].

Урок атаки Triton є фундаментальним. Атака не використовувала традиційне "шкідливе ПЗ". Вона виглядала як легітимна зміна конфігурації (завантаження нової логіки в контролер) з авторизованої інженерної станції. Це доводить, що моніторинг, заснований лише на пошуку відомих сигнатур вірусів або аналізі мережевого трафіку, є абсолютно недостатнім. Ефективний моніторинг повинен бути здатним виявляти контекст: хто, коли і які саме конфігураційні файли чи логіку контролера намагається змінити.

1.2.4 Внутрішні загрози

Окрім зовнішніх атак, КІС є надзвичайно вразливими до внутрішніх загроз [29], які поділяються на:

– Зловмисні: саботаж з боку незадоволеного співробітника, промислове шпигунство, зловмисне використання привілеїв [29].

– Ненавмисні: помилка оператора, неправильна конфігурація обладнання, або успішна фішинг-атака на співробітника, що призводить до компрометації його облікового запису [29].

Виявлення таких загроз вимагає не стільки пошуку "вірусів", скільки аналізу поведінки користувачів та пристроїв (UEBA) для виявлення аномалій [30]. Наприклад, якщо інженер, який зазвичай працює з 9 до 17, раптово о 3-й ночі починає завантажувати нову конфігурацію на PLC, — це аномалія, яку система моніторингу повинна негайно ідентифікувати.

1.3 Технологія SIEM: архітектура, принципи роботи, компоненти

Центральним технологічним рішенням для вирішення складних завдань моніторингу, описаних вище, є системи класу SIEM. SIEM — це технологія, що забезпечує комплексний підхід до управління безпекою шляхом централізованого збору, агрегації, зберігання та аналізу подій безпеки з різнорідних джерел в інфраструктурі організації [12].

1.3.1 Еволюція SIEM: від агрегації до інтелекту

Сучасні SIEM-системи є результатом еволюційного розвитку двох окремих класів продуктів [31]:

1. SIM (Security Information Management) — системи, що фокусувалися на довготривалому зберіганні, аналізі та звітності журналів. Їх основною метою було забезпечення відповідності вимогам регуляторів та підтримка розслідувань.

2. SEM (Security Event Management) — системи, що фокусувалися на моніторингу в реальному часі, кореляції подій та генеруванні сповіщень (alerts) про інциденти.

У 2005 році аналітична компанія Gartner об'єднала ці два поняття в єдиний термін "SIEM", визначивши ринок для інтегрованих рішень [32]. Перше покоління SIEM-систем (SIEM 1.0) було орієнтоване переважно на кореляцію на основі жорстких правил та звітність для комплаєнсу.

Проте, ці системи швидко зіткнулися з двома фундаментальними проблемами [33]:

1. "Втома від сповіщень" (Alert Fatigue): величезна кількість подій, що генеруються в сучасних мережах, призводила до генерації тисяч сповіщень на день, більшість з яких були хибно-позитивними. Аналітики SOC (Security Operations Center) фізично не встигали обробляти цей потік і починали ігнорувати сповіщення.

2. Неможливість виявлення невідомих загроз: системи, засновані на правилах, здатні виявляти лише те, що їм відомо (наприклад, "блокувати IP-адресу X"). Вони були "сліпими" до атак нульового дня, складних багатоетапних атак (APT) та інсайдерських загроз, які не порушували чітких правил, але відхилялися від нормальної поведінки.

Для вирішення цих проблем відбулася подальша еволюція SIEM-систем (Next-Gen SIEM), яка інтегрувала нові технології [32]:

● UEBA (User and Entity Behavior Analytics) — інтеграція модулів машинного навчання та статистичного аналізу для побудови "базової лінії" нормальної поведінки для кожного користувача та пристрою в мережі. Це дозволило виявляти аномалії та інсайдерські загрози, які раніше залишалися непоміченими [34].

● SOAR (Security Orchestration, Automation and Response) — інтеграція інструментів для автоматизації процесів реагування на інциденти. SOAR-платформи використовують "playbooks" (сценарії реагування) для автоматичного збагачення даних про інцидент (наприклад, перевірка IP-адреси по базах Threat Intelligence), координації дій між різними системами безпеки (наприклад, SIEM, файрвол, EDR) та управління життєвим циклом інциденту [32].

● XDR (Extended Detection and Response) — нова концепція, що частково конкурує, а частково доповнює SIEM. Якщо SIEM традиційно фокусується на журналах з багатьох джерел, то XDR фокусується на глибокій, інтегрованій телеметрії з обмеженого набору джерел (кінцеві точки, мережа, хмара), пропонуючи більш якісні та контекстно-збагачені дані для виявлення загроз [35]. Сучасні SIEM-системи прагнуть інтегрувати XDR-можливості.

1.3.2 Базова архітектура та компоненти SIEM

Попри відмінності в реалізації, більшість SIEM-систем мають схожу логічну архітектуру, що складається з декількох рівнів [36]:

1. Джерела даних — будь-які пристрої або програмне забезпечення, що генерують події безпеки. У гібридному IT/OT середовищі це включає:

– IT-джерела: фаєрволи, комутатори, маршрутизатори, контролери домену (Active Directory), VPN-концентратори, антивіруси, сервери [37].

– OT-джерела: HMI-станції, інженерні станції (EWS), сервери історії (Historians), промислові фаєрволи, мережеві сенсори OT.

2. Рівень збору — компоненти, що відповідають за збір даних з джерел.

– Агенти — програмне забезпечення, що встановлюється безпосередньо на кінцевий пристрій (наприклад, HMI-сервер) для збору журналів ОС, моніторингу файлів тощо. Прикладом може бути NXLog [38] або спеціалізовані агенти (наприклад, Wazuh Agent).

– Колектори — сервіси, що отримують дані без встановлення агентів, використовуючи стандартні протоколи, такі як Syslog, SNMP, NetFlow, WMI, або через API-інтеграції [39].

3. Рівень обробки — компоненти, що готують дані для аналізу.

– Парсинг — процес розбору неструктурованих або напівструктурованих текстових журналів та виділення з них значущих полів (напр., "IP-джерела", "Порт", "Ім'я користувача").

– Нормалізація — приведення різнорідних даних (наприклад, журнал Windows має "Event ID 4625", а журнал Linux — "Failed password") до єдиного, уніфікованого формату (наприклад, "User.Login.Failed") [38]. Це критично важливий крок, що дозволяє корелювати події з різних типів пристроїв.

4. Рівень аналітики (Analytics Layer) — "Мозок" SIEM-системи.

– Механізм кореляції — компонент, що в реальному часі застосовує набір правил до нормалізованого потоку подій для виявлення складних патернів атак [36]. Наприклад, правило може виглядати так: "Якщо 5 подій 'User.Login.Failed' від одного користувача відбулися протягом 1 хвилини, а потім відбулася 1 подія 'User.Login.Success' з IP-адреси, яка раніше не бачилася, — згенерувати сповіщення 'Можлива атака Brute Force'".

– Аналіз поведінки (UEBA) — модулі машинного навчання, що аналізують поведінку в тривалішому часовому вікні для виявлення аномалій.

5. Рівень зберігання: Бази даних, що використовуються для зберігання оброблених даних. Зазвичай використовується дворівнева система: "гаряче" сховище (наприклад, Elasticsearch, MySQL) для швидкого доступу та аналітики за останні дні/тижні, та "холодне" сховище (архів) для довготривалого зберігання (місяці/роки) з метою комплаєнсу.

6. Рівень представлення — інтерфейс для аналітиків.

– Інформаційні панелі — візуалізація ключових метрик безпеки в реальному часі.

– Система сповіщень — механізми повідомлення аналітиків про виявлені інциденти.

– Система звітності — інструменти для генерації звітів для керівництва або аудиторів.

1.3.3 Принципи роботи: життєвий цикл події у SIEM

Процес кореляції, що є ядром SIEM, являє собою складний багатоетапний конвеєр обробки даних, призначений для перетворення величезного обсягу "сирих" і малоінформативних журналів у невелику кількість значущих та дієвих сповіщень про інциденти [38].

Цей процес включає наступні кроки [38]:

1. Фільтрація — відсіювання очевидного "шуму". Наприклад, інформаційні повідомлення про успішне встановлення з'єднання на фаєрволі, які не несуть цінності для безпеки, можуть бути відфільтровані, щоб зменшити навантаження на систему.

2. Агрегація та дедуплікація — об'єднання однакових або схожих подій, що відбулися за короткий проміжок часу, в одну. Якщо файрвол заблокував 1000 однакових пакетів з однієї IP-адреси за 1 секунду, SIEM не створить 1000 окремих сповіщень, а створить одне: "Подія 'Firewall.Block' сталася 1000 разів".

3. Маскування — тимчасове ігнорування сповіщень, які є очікуваним наслідком відомих дій. Наприклад, якщо адміністратор проводить планове обслуговування сервера з 14:00 до 15:00, система може бути налаштована на маскування сповіщень про "Втрату зв'язку з сервером" у цей проміжок часу, щоб уникнути хибних тривог.

4. Кореляція — власне, аналітичний етап, де система шукає зв'язки між різними нормалізованими подіями з різних джерел, застосовуючи правила або моделі [39].

5. Аналіз першопричини — найбільш складний етап, де система (або аналітик за допомогою інструментів SIEM) намагається визначити не лише що сталося, але й чому. Це може включати побудову діаграм залежностей, щоб прослідкувати ланцюжок подій до початкової першопричини інциденту [38].

1.4 Порівняльний огляд сучасних систем SIEM

Ринок SIEM-рішень є висококонкурентним і включає як потужні комерційні платформи від гігантів індустрії, так і гнучкі рішення з відкритим кодом. Вибір конкретної платформи для автоматизації моніторингу КІС залежить від низки специфічних факторів, що випливають з аналізу, проведеного в попередніх підрозділах.

1.4.1 Критерії оцінки SIEM для середовищ КІС

Для об'єктивного порівняння та обґрунтування вибору платформи, необхідно визначити критерії оцінки, що відображають специфічні виклики захисту КІС (описані в 1.1 та 1.2):

1. Модель ліцензування та вартість (TCO): комерційна (за обсягом даних (GB/day) або кількістю подій за секунду (EPS)) чи Open-Source (безкоштовна ліцензія).

2. Архітектура та гнучкість: можливість розгортання on-premise (що є вимогою для ізольованих OT-мереж), гнучкість та "вага" агентів (для сумісності з legacy-системами), модульність платформи.

3. Основний функціонал (SIEM/EDR/XDR): чи є це чистим SIEM (агрегатор журналів), чи платформа об'єднує функції захисту кінцевих точок (EDR) та розширеного виявлення (XDR)?

4. Критичні функції для OT: наявність вбудованих модулів, що реалізують компенсаційні контроли NIST:

– Моніторинг цілісності файлів (FIM): для відстеження змін у конфігураціях PLC/SCADA (захист від загроз типу Triton).

– Виявлення вразливостей (VD): для інвентаризації та моніторингу вразливостей на застарілих HMI/EWS.

5. Підтримка OT/ICS: наявність вбудованих парсерів для промислових протоколів або можливість інтеграції зі спеціалізованими OT-сенсорами.

6. Масштабованість та спільнота: легкість горизонтального масштабування та наявність активної спільноти (для open-source) або якісної комерційної підтримки.

1.4.2 Порівняльний аналіз ключових рішень

На основі визначених критеріїв, проведемо аналіз чотирьох платформ: двох комерційних лідерів (Splunk, IBM QRadar), визнаних Gartner, та двох провідних рішень з відкритим кодом (Elastic SIEM, Wazuh).

Таблиця 1.3 – Порівняльний аналіз провідних SIEM-рішень

Платформа	**Splunk Enterprise Security**	**IBM QRadar**	**Elastic SIEM (ELK Stack)**	**Wazuh**
Модель ліцензування	Комерційна (дуже висока вартість, на основі обсягу даних)	Комерційна (висока вартість, на основі EPS та потоків)	Open-Source (ELK Stack), з платними комерційними ліцензіями (X-Pack)	Open-Source (повністю безкоштовно)

Продовження таблиці 1.3

Платформа	**Splunk Enterprise Security**	**IBM QRadar**	**Elastic SIEM (ELK Stack)**	**Wazuh**
Основний функціонал	SIEM, Аналітика, Платформа даних	SIEM, Аналітика потоків (QFlow), UEBA	Log Management, SIEM, Observability	SIEM, XDR, EDR (уніфікована платформа)
Моніторинг цілісності	Через додатки (Apps) або Splunk UFA	Через додатки (Apps) або агенти	Через Elastic Agent (Beat)	Вбудовано (ключова функція)
Виявлення вразливостей	Через додатки та інтеграцію	Через окремий платний модуль (QVM)	Обмежено / через інтеграцію	Вбудовано (ключова функція)
Гнучкість агентів	Universal Forwarder (відносно "важкий")	Вимагає адаптерів (DMS) або агентів	Elastic Agent / Beats (гнучкий)	Легкий агент, підтримка legacy ОС
Складність / Підтримка	Висока складність налаштування, відмінна комерційна підтримка	Дуже висока складність, застарілий UI, високі вимоги до обслуговування	Висока складність налаштування та адміністрування кластера	Середня складність, сильна та активна спільнота
Придатність для ОТ (застарілі системи)	Висока (через додатки), але надзвичайно висока вартість	Висока (через додатки), але складна та дорога	Середня (вимагає значної кастомізації)	Висока (легкі агенти, вбудовані FIM/VD, нульова вартість)

Джерело: складено на основі [40]

1.4.3 Аналіз комерційних лідерів

Splunk вже багато років є лідером "Магічного квадранту" Gartner для SIEM [41]. Його беззаперечна перевага — надзвичайна потужність мови пошуку (SPL), гнучкість аналітики та величезна екосистема додатків (Splunkbase) [42]. Однак, його головним недоліком є модель ліцензування, що

базується на обсязі даних, що індексуються на день [43]. У середовищах ОТ, які генерують величезні обсяги телеметрії та журналів, вартість ліцензії Splunk стає астрономічною, що робить його недоступним для багатьох, особливо для академічних досліджень.

IBM QRadar також є багаторічним лідером ринку [40]. Його сильною стороною традиційно вважається потужний механізм кореляції "з коробки" та унікальна здатність аналізувати не лише журнали, але й мережеві потоки (QFlows) [44]. Проте, користувачі та аналітики відзначають "значну складність" системи, високі вимоги до обслуговування та "застарілий і незрозумілий" інтерфейс користувача, що ускладнює швидке реагування та навчання аналітиків [44].

1.4.4 Обґрунтування вибору Wazuh

На основі комплексного аналізу стандартів моніторингу (1.1), специфічних загроз для КІС (1.2) та порівняльного аналізу технологій (1.4), вибір платформи Wazuh для подальшого дослідження та практичного розгортання в рамках даної магістерської дисертації обґрунтовується низкою ключових, взаємопов'язаних аргументів.

Wazuh обрано не тому, що він є "кращим" за Splunk у загальному сенсі, а тому, що його архітектура та набір вбудованих функцій є оптимальними для вирішення специфічних завдань автоматизації моніторингу в гетерогенних, вразливих та критичних середовищах комп’ютерно-інтегрованих систем.

1. Архітектурна відповідність компенсаційним контролям NIST

Як було встановлено в підрозділі 1.1, стандарт NIST SP 800-82r3 визнає неможливість прямого аудиту застарілих ОТ-пристроїв і вимагає впровадження "компенсаційних контролів". Як було встановлено в підрозділі 1.2, найнебезпечніші атаки (Triton) реалізуються через маніпуляцію конфігураційними файлами та проектами на інженерних станціях (EWS) та серверах HMI.

Wazuh надає потужний, вбудований модуль File Integrity Monitoring (FIM) [45]. Встановлення агента Wazuh на HMI-станцію або EWS дозволяє в режимі реального часу відстежувати будь-які зміни (створення, модифікацію, видалення) у критичних файлах конфігурації SCADA або проектах PLC [45].

Таким чином, Wazuh дозволяє ідеально реалізувати компенсаційний контроль SI-4 "Information Integrity", рекомендований NIST, для протидії загрозам найвищого рівня, подібним до Triton. Якщо зловмисник або інсайдер спробує модифікувати логіку контролера, FIM-модуль Wazuh негайно зафіксує зміну у файлі проекту на EWS і згенерує сповіщення, ще до того, як шкідливий код потрапить на контролер.

2. Проактивне управління вразливостями в КІС

Як було показано в 1.2, КІС-середовища насичені застарілим ПЗ, що працює на непідтримуваних ОС (legacy OS) і має безліч відомих вразливостей. Ручна інвентаризація цих вразливостей є неефективною.

Wazuh має вбудований модуль Vulnerability Detection (VD) [46]. Агент Wazuh періодично сканує список встановленого програмного забезпечення на контрольованому вузлі (наприклад, HMI) і відправляє його на сервер. Сервер Wazuh зіставляє цей список з оновлюваними базами даних CVE (Common Vulnerabilities and Exposures), такими як NVD.

Це дозволяє автоматизувати процес безперервного моніторингу вразливостей на критичних вузлах КІС [47]. Система не лише реагує на атаки, але й проактивно вказує на слабкі місця, що дозволяє пріоритезувати заходи із захисту (наприклад, віртуальний патчинг або додаткову сегментацію).

3. Економічна та дослідницька доцільність

Wazuh є повністю безкоштовним рішенням з відкритим кодом [48]. Це має дві ключові переваги для магістерської дисертації:

1) Практична: дозволяє провести повне розгортання, тестування та валідацію рішення в лабораторних умовах без будь-яких фінансових витрат.

2) Академічна: прозорість коду та гнучкість конфігурації (написання власних правил, декодерів) дозволяють глибоко дослідити та модифікувати

логіку кореляції. Це є неможливим у комерційних "чорних скриньках", де логіка роботи прихована від дослідника [49].

4. Уніфікована платформа

На відміну від багатьох інших SIEM, Wazuh не є просто агрегатором журналів. Він з самого початку проектувався як гібридна платформа, що поєднує SIEM (централізований аналіз журналів на сервері) та XDR/EDR (глибокий моніторинг кінцевих точок через агентів) [50].

Це дає значно багатший контекст для виявлення загроз. Традиційний SIEM бачить подію: "Журнал файрвола: IP 192.168.1.10 (HMI) з'єднався зі зловмисним IP 10.20.30.40". Wazuh бачить не лише це, але й одночасно отримує подію від агента на HMI: "Процес 'svchost.exe' (PID 1234) з батьківським процесом 'explorer.exe' ініціював мережеве з'єднання з 10.20.30.40". Це дозволяє миттєво корелювати мережеву подію з конкретним скомпрометованим процесом на кінцевій точці.

1.5 Постановка завдання магістерської дисертації

Відповідно до поставленої мети дослідження, в роботі необхідно вирішити наступні науково-технічні та практичні завдання:

1. Проаналізувати архітектуру та принципи роботи SIEM-системи Wazuh для її впровадження.

2. Здійснити практичне розгортання та конфігурацію компонентів системи: розгорнути гіпервізор VMware ESXi, встановити серверну частину Wazuh на базі Ubuntu та інтегрувати гетерогенні клієнтські системи (Windows, Linux) шляхом встановлення та реєстрації агентів.

3. Налаштувати механізми автоматизації збору журналів та аналізу даних безпеки в розгорнутій інфраструктурі.

4. Розробити сценарії та провести моделювання типових загроз безпеки, зокрема невдалих спроб авторизації, для експериментальної перевірки працездатності системи.

5. Проаналізувати результати експерименту: дослідити спрацьовування правил кореляції в Wazuh, проаналізувати згенеровані журнали подій та надати оцінку ефективності системи щодо виявлення загроз.

6. Визначити перспективи розвитку запропонованого рішення та дослідити можливості його інтеграції у промислові комп’ютерно-інтегровані середовища.

7. Розробити концепцію стартап-проєкту для комерціалізації рішення з автоматизації моніторингу, що включає аналіз ринку, розробку ринкової стратегії та маркетингової програми.

Висновки до розділу 1

У першому розділі магістерської дисертації проведено комплексний аналіз теоретичних та практичних основ моніторингу загроз у комп’ютерно-інтегрованих системах. Аналіз дозволив сформулювати низку ключових висновків, що лягли в основу подальшого дослідження.

1. Встановлено, що моніторинг КІС фундаментально відрізняється від моніторингу ІТ-систем через інверсію пріоритетів безпеки, де фізична безпека та доступність процесу є важливішими за конфіденційність, та через масову наявність застарілого (legacy) обладнання. Ці фактори унеможливлюють пряме застосування багатьох ІТ-підходів, зокрема активного сканування.

2. Сучасні стратегії моніторингу КІС базуються на міжнародних стандартах, зокрема IEC 62443 та NIST SP 800-82r3. Ключовими концепціями, що випливають з цих стандартів, є: 1) архітектурна сегментація мережі на "Зони та Канали" (за IEC 62443) для контролю інформаційних потоків; 2) впровадження "компенсаційних контролів" (за NIST SP 800-82r3) для моніторингу тих пристроїв, які не підтримують аудит, шляхом контролю їхнього оточення (наприклад, HMI та EWS).

3. Ландшафт загроз для КІС еволюціонував від загального шкідливого ПЗ до високоспеціалізованих кіберфізичних атак, розроблених для спричинення фізичних руйнувань. Аналіз прецедентів, зокрема Stuxnet,

CrashOverride та Triton, демонструє найвищий рівень загрози: цілеспрямовану атаку на Системи Інструментальної Безпеки (SIS) з метою нейтралізації захисту та спричинення техногенної катастрофи.

4. Технології SIEM є ядром сучасних центрів моніторингу (SOC), оскільки вони дозволяють централізовано збирати та корелювати дані з різнорідних джерел. Вони еволюціонували від простих агрегаторів журналів до інтелектуальних систем, що включають поведінковий аналіз (UEBA) та засоби автоматизації реагування (SOAR).

5. Проведений порівняльний аналіз провідних SIEM-рішень (Splunk, IBM QRadar, Elastic SIEM, Wazuh) та подальше обґрунтування показали, що платформа Wazuh є оптимальним вибором для завдань даної дисертації. Цей вибір обґрунтовується не загальною перевагою, а специфічною архітектурною відповідністю викликам КІС. Зокрема, наявність у Wazuh потужних вбудованих модулів File Integrity Monitoring (FIM) та Vulnerability Detection (VD) дозволяє безпосередньо реалізувати "компенсаційні контроли", необхідні для моніторингу застарілих HMI/EWS та протидії загрозам типу Triton. Модель open-source та уніфікована архітектура (SIEM+XDR) роблять її ідеальною платформою для академічного дослідження та практичної реалізації автоматизованої системи моніторингу.

РОЗДІЛ 2. РОЗГОРТАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ МОНІТОРИНГУ ЗАГРОЗ

2.1 Архітектура та принцип роботи SIEM-системи Wazuh

2.1.1 Загальна характеристика Wazuh як SIEM-платформи

Wazuh є сучасною відкритою платформою класу SIEM та XDR, орієнтованою на забезпечення комплексного моніторингу безпеки, виявлення інцидентів, аналізу журналів та відповідностей нормативним вимогам. Архітектура системи побудована модульно та масштабовано: окремі компоненти відповідають за збір даних, аналітичну обробку, збереження інформації та її подальшу візуалізацію. Така структура дає змогу адаптувати платформу до різних масштабів інфраструктури — від невеликих робочих груп до розподілених корпоративних та інтегрованих середовищ.

Основою роботи системи є взаємодія між агентами, встановленими на контрольованих вузлах, та центральними компонентами — сервером аналізу, індексатором та інформаційною панеллю. Додатково Wazuh підтримує роботу з агентless-джерелами, такими як мережеве обладнання чи системами без можливості встановлення традиційних агентів. Завдяки цьому система охоплює повний спектр активів — від кінцевих пристроїв і серверів до хмарних середовищ та контейнеризованих платформ.

2.1.2 Логічна архітектура системи Wazuh

Архітектура Wazuh складається з чотирьох ключових елементів [51]:

1. Wazuh Agent.
2. Wazuh Server.
3. Wazuh Indexer.
4. Wazuh Dashboard.

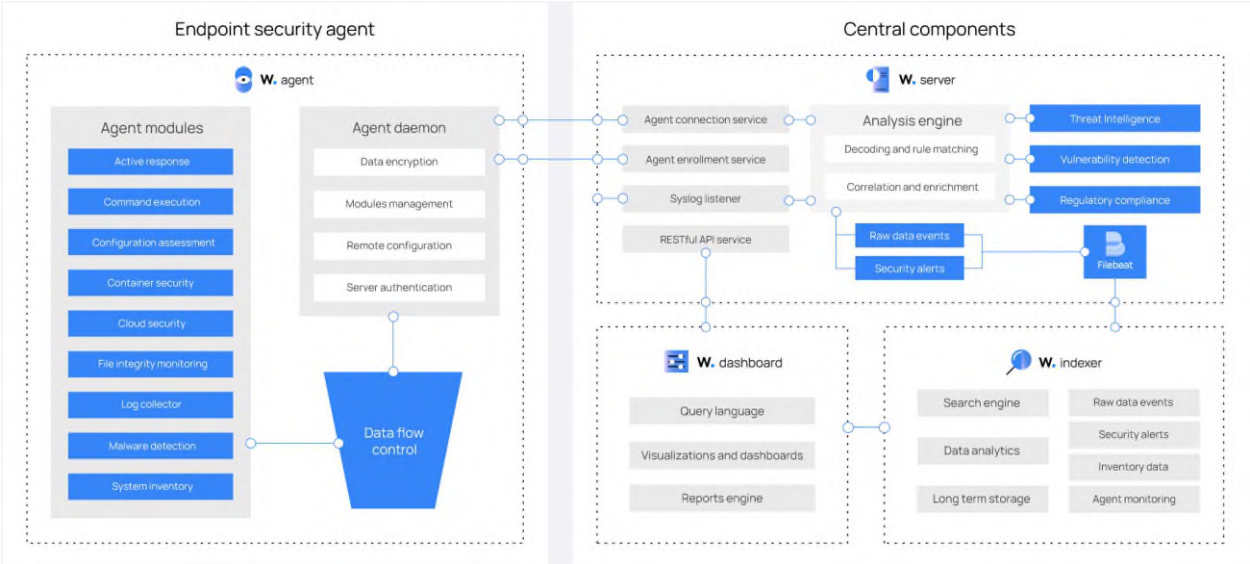


Рисунок 1.1 – Загальна модульна схема архітектури Wazuh [51]

2.1.3 Компонент Wazuh Agent

Wazuh Agent — це клієнтський модуль, який розміщується на широкому спектрі пристроїв: робочих станціях, серверах, контейнерних вузлах, хмарних інстансах тощо [52]. Основна функція агента полягає у постійному зборі інформації про безпековий стан системи.

Загальний набір можливостей агента включає:

– збір логів операційної системи та застосунків;
– виконання інвентаризації апаратного та програмного забезпечення;
– контроль цілісності файлів (FIM);
– моніторинг конфігурацій безпеки (SCA);
– виявлення руткітів та аномалій;
– реакцію на інциденти (Active Response);
– моніторинг контейнерного та хмарного середовища.

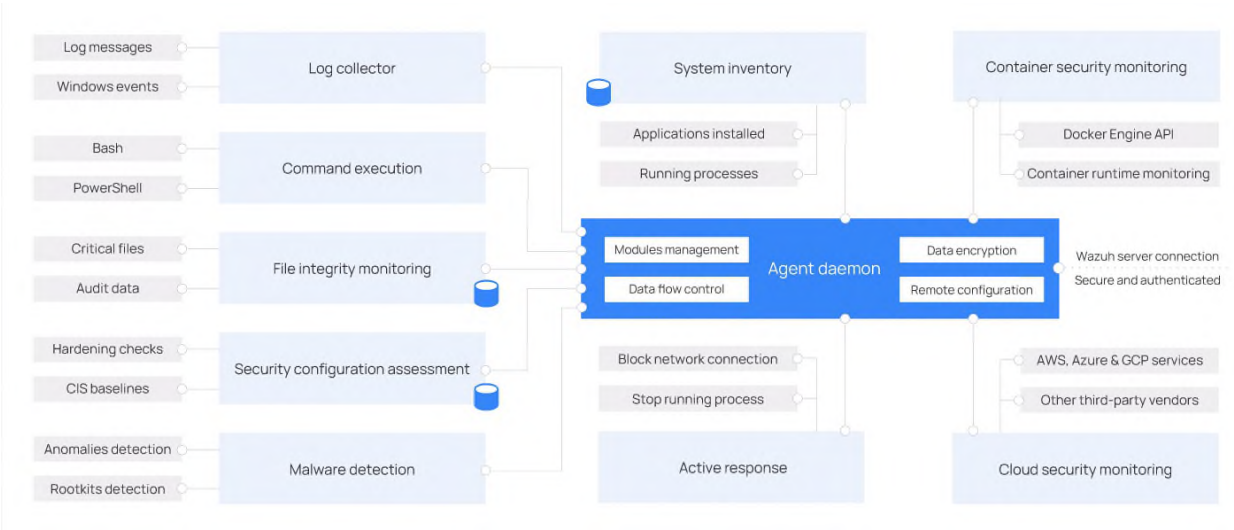


Рисунок 1.2 – Модульна схема архітектури Wazuh Agent [52]

Агент здійснює попередню обробку даних: нормалізує формат, додає метадані (ідентифікатор вузла, часові мітки, тип джерела), виконує локальні перевірки цілісності файлів, аналізує конфігурації на відповідність політикам безпеки. Після цього події шифруються (типово AES з 256-бітними ключами) і передаються на Wazuh Server по захищеному каналу [52].

Wazuh підтримує також agentless-моніторинг для мережевого обладнання й систем, де встановлення агента неможливе (наприклад, деякі апаратні firewall або старі UNIX-системи). У цьому випадку лог-дані надходять через Syslog або SSH безпосередньо на сервер.

2.1.4 Компонент Wazuh Server

Wazuh Server є центральним компонентом — логічним ядром системи, у якому реалізовано [53]:

– службу підключення агентів (agent connection service);

– службу реєстрації та видачі ключів (agent enrollment);

– аналізатор подій (Analysis engine);

– кластерний демон (cluster daemon) для роботи у багатовузловій конфігурації;

– інтегрований Filebeat, що експортує проаналізовані події до Wazuh indexer.

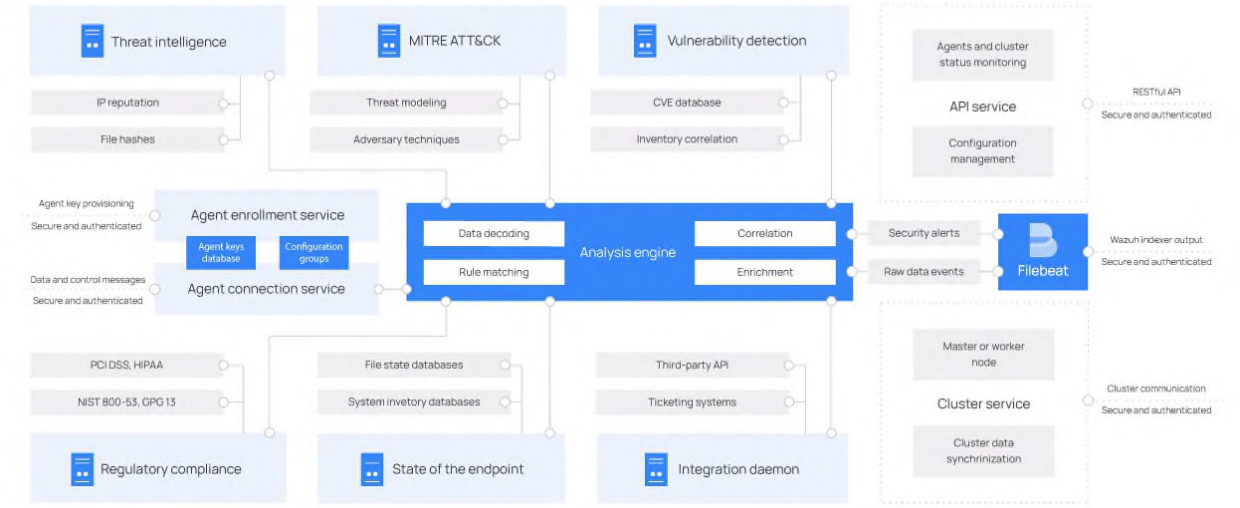


Рисунок 1.3 – Модульна схема архітектури Wazuh Server [53]

Основні функції Wazuh Server:

1. Керування агентами: централізоване додавання, видалення, прив’язка до груп, розповсюдження конфігурацій, оновлення.

2. Приймання та буферизація подій: сервер приймає потік повідомлень від агентів (TCP/UDP 1514 за замовчуванням), поміщає їх у внутрішні черги (FIFO) і розподіляє між потоками обробки.

3. Декодування та нормалізація: за допомогою декодерів події перетворюються у структурований вигляд (поля: джерело, користувач, IP-адреса, статус операції тощо).

4. Застосування правил: до нормалізованих подій застосовується правила, які описують умови виявлення підозрілої активності: невдала аутентифікація, сканування портів, зміна критичних файлів, запуск підозрілих процесів тощо.

5. Генерація сповіщень і активна реакція: при спрацьовуванні правил створюються алерти, які можуть запускати Active response — виконання скриптів блокування IP, завершення процесів, видалення шкідливих файлів та інші дії.

2.1.5 Компонент Wazuh Indexer

Wazuh Indexer — це кластерна система пошуку й аналітики, побудована на основі OpenSearch [54]. Вона отримує потік подій від Wazuh Server через Filebeat та:

– індексує дані у часових індексах;

– забезпечує швидкий пошук, фільтрацію та агрегації;

– служить сховищем для довгострокового зберігання журналів.

Indexer може розгортатися як:

– одновузловий варіант для тестових або невеликих інсталяцій;

– кластер з декількох нод для великих інфраструктур з великою кількістю агентів та високим потоком подій.

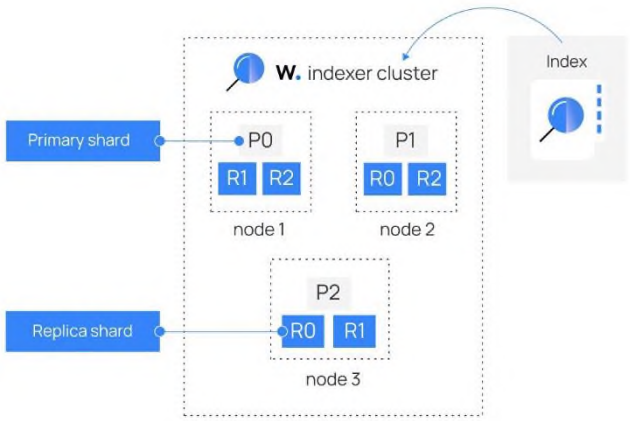


Рисунок 1.4 – Модульна схема архітектури Wazuh Indexer [54]

2.1.6 Компонент Wazuh Dashboard

Wazuh Dashboard — веб-консоль, через яку адміністратор та аналітики SOC взаємодіють із платформою [55].

Основні можливості:

– візуалізація алертів, журналів та метрик безпеки в реальному часі;

– керування агентами: перегляд статусу, включення/відключення модулів, налаштування джерел логів;

– інструменти розробника: тестування власних правил і декодерів, взаємодія з API Wazuh server та Wazuh indexer.

Dashboard отримує дані з Indexer та звертається до API сервера (порт 55000/TCP за замовчуванням) для відображення конфігурацій, статусів і виконання керувальних операцій. Усі з'єднання захищені TLS, а доступ контролюється ролями й обліковими записами [55].

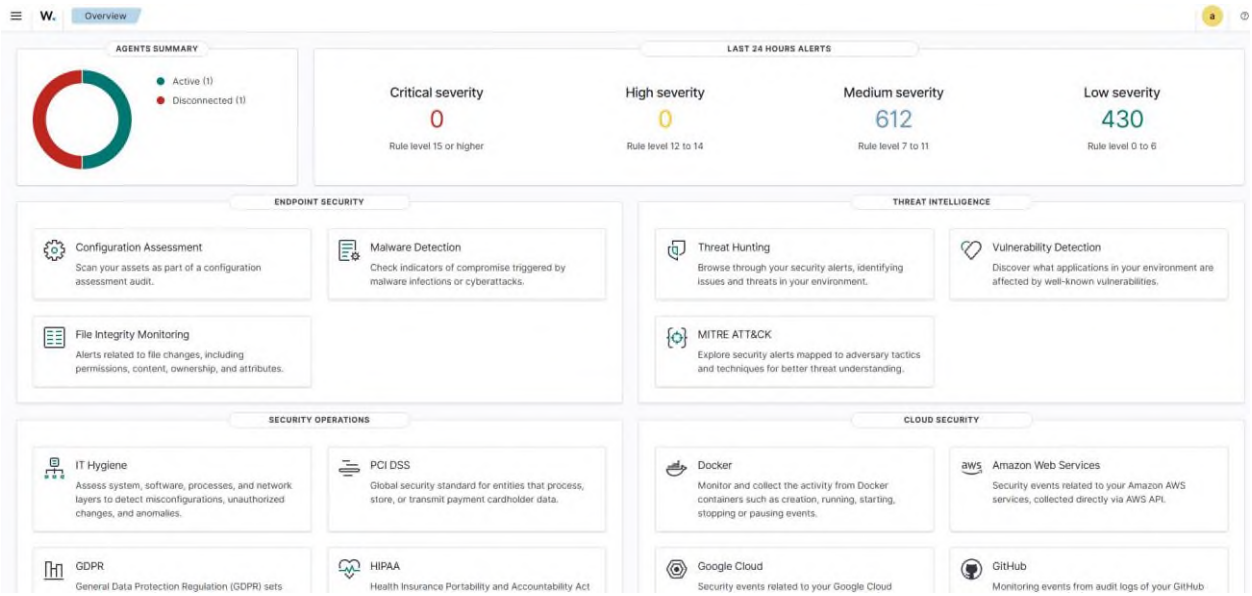


Рисунок 1.5 – Веб-інтерфейс панелі керування Wazuh

2.1.7 Топологія розгортання Wazuh

Wazuh підтримує кілька типових сценаріїв розгортання, що відрізняються за масштабованістю й відмовостійкістю [56]:

1. All-in-one

Усі основні компоненти (server, indexer, dashboard) розгорнуті на одному фізичному або віртуальному сервері. Такий варіант підходить для лабораторій, тестових стендів або невеликих організацій із невеликою кількістю вузлів.

2. Single-node

Кожен компонент розгортається на окремому сервері: один — для Wazuh Server, окремий — для Wazuh Indexer, ще один — для Wazuh Dashboard. Це покращує продуктивність та дозволяє гнучкіше масштабувати елементи.

3. Multi-node

Для великих інфраструктур застосовується багатовузлова архітектура:

– кластер Wazuh Server (master + декілька workers);

– кластер Wazuh Indexer із кількох індекс-нод;

– load balancer перед серверами для рівномірного розподілу навантаження.

У такій конфігурації агенти надсилають події на IP/ім’я балансувальника, а той розподіляє трафік між вузлами кластера. Це дає можливість горизонтально масштабувати обробку подій без зміни конфігурацій на агентах.

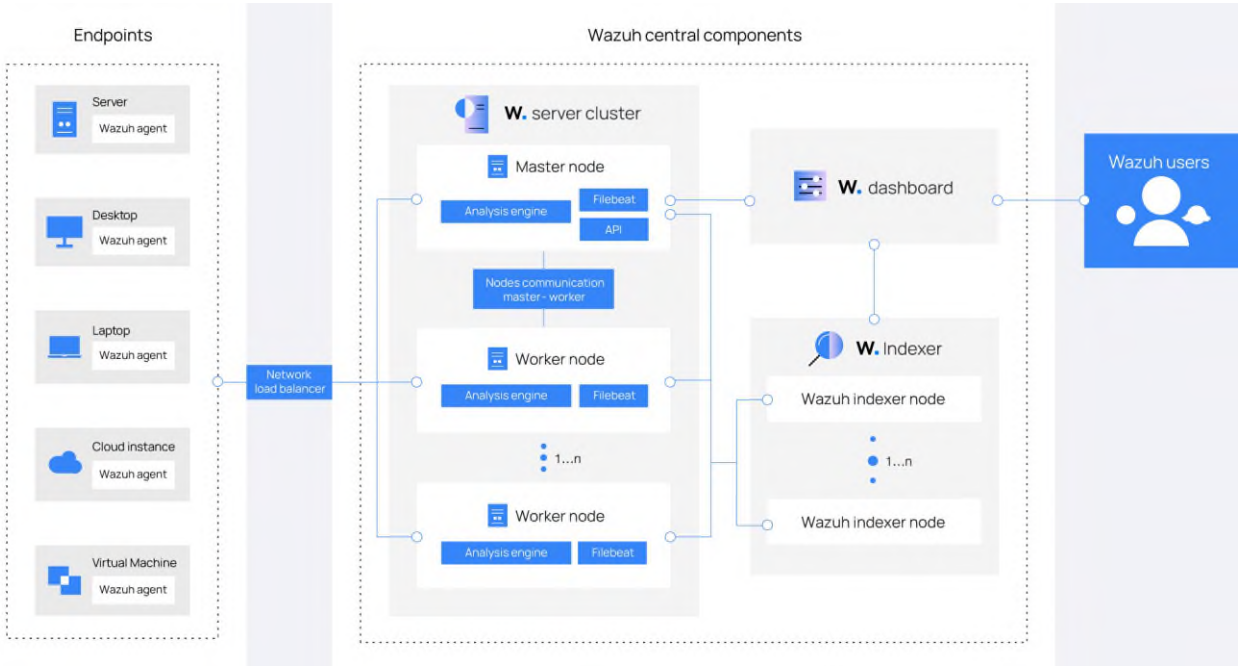


Рисунок 1.6 – Multi-node топологія розгортання Wazuh [56]

2.1.8 Принцип роботи Wazuh як SIEM-системи

Принцип роботи Wazuh як SIEM можна описати у вигляді послідовності етапів:

Етап 1: Збір і передавання подій

1. На кінцевому вузлі agent перехоплює системні події (аутентифікація, доступ до файлів, дії користувачів), а також читає журнали прикладних сервісів.

2. Додаткові модулі агента ведуть:

– моніторинг цілісності файлів (FIM);

– виявлення вразливостей шляхом кореляції встановлених пакетів із базами CVE;

– аналіз конфігурацій на відповідність політикам безпеки.

3. Агент формує події у внутрішньому форматі, шифрує їх і відправляє на Wazuh server по TCP/UDP 1514. При встановленні з'єднання використовується механізм взаємної аутентифікації за ключами, які були видані службою enrollment.

Етап 2: Черги подій та багатопоточна обробка

Отримані події потрапляють до черг (queues) Wazuh Server. Механізм реалізований за принципом FIFO, тобто події обробляються в тому ж порядку, у якому були отримані. Це дозволяє:

– вирівняти навантаження при пікових сплесках;

– уникати втрати подій у разі короткочасних перевантажень;

– розподіляти обробку між кількома потоками аналізу.

Етап 3: Декодування та нормалізація

Кожне вхідне повідомлення проходить через ланцюжок декодерів:

– базові декодери розпізнають загальні формати логів (syslog, JSON, Windows Event Log тощо);

– спеціалізовані – для конкретних застосунків (Apache, Nginx, SSH, файрволи, VPN-сервери, cloud-сервіси);

– результати декодування перетворюються на набір іменованих полів (наприклад, srcip, dstip, user, action, status).

Така нормалізація дає змогу далі застосовувати універсальні правила до подій різних джерел.

Етап 4: Застосування правил та кореляція

На наступному етапі події надходять до двигуна правил (rules engine). Правила Wazuh визначаються в XML-файлах і складаються з умов:

– відповідність значенням певних полів (IP-адреса, ім’я користувача, назва процесу);

– регулярні вирази над текстом повідомлення;

– порівняння рівнів важливості, статусів операцій;

– часові умови (кількість подій за інтервал часу, послідовність дій).

На основі цих умов Wazuh дозволяє реалізувати як прості, так і складні сценарії кореляції, наприклад:

– N невдалих входів у систему протягом t секунд з однієї IP-адреси;

– зміна критичних конфігураційних файлів і одразу після цього – запуск незнайомого бінарного файлу;

– поява нових служб, які слухають мережеві порти, у поєднанні з відомими індикаторами компрометації (IOC) з бази Threat Intelligence.

При спрацьовуванні правила формується алерт із рівнем критичності, класифікацією (наприклад, authentication_failure, privilege_escalation, data_exfiltration_suspected), посиланнями на MITRE ATT&CK тощо.

Етап 5: Збереження та індексація подій

Генеровані алерти та події передаються на Wazuh Indexer через Filebeat. Indexer:

– записує їх в індекси, розбиті за часом (наприклад, добові індекси);

– зберігає оригінальні повідомлення й нормалізовані поля;

– надає API для пошуку й агрегацій.

Важливо, що Wazuh дозволяє налаштовувати політику життєвого циклу даних (ILM): старі індекси можуть архівуватися, видалятися або переміщуватися на дешевші сховища.

Етап 6: Візуалізація, аналітика та розслідування інцидентів

Через Wazuh Dashboard аналітики SOC отримують доступ до:

– дашбордів реального часу: графіки, діаграми, карти, таблиці;

– панелей відповідності стандартам: показ відповідності вимогам PCI DSS, GDPR, HIPAA, NIST 800-53 тощо;

– звітування – побудова регулярних звітів про стан безпеки, активність користувачів, статус патчів.

У процесі розслідування інцидентів аналітик зазвичай:

– отримує алерт високої критичності;

– відкриває відповідний дашборд і переглядає деталізовані події;

– фільтрує події за IP-адресою, хостом, користувачем;

– аналізує контекст: попередні дії з тим самим обліковим записом чи з того ж хоста;

– приймає рішення: ескалювати, блокувати, створювати тікет, запускати автоматизовану реакцію.

Етап 7: Автоматизоване реагування через Active Response

Однією з важливих переваг Wazuh є модуль Active Response, що дозволяє автоматично виконувати дії при спрацюванні певних правил:

– блокування IP-адреси на firewall або у hosts.deny;

– примусове завершення процесу;

– видалення шкідливого файлу;

– запуск зовнішнього скрипта (наприклад, інтеграція з системами тікетингу, оркестрації SOAR, месенджерами).

Active Response може працювати у двох режимах:

– stateless – одноразова дія у відповідь на алерт;

– stateful – пара дій для активації/деактивації (наприклад, блокування IP і подальше розблокування після закінчення часу).

2.1.9 Безпека взаємодії компонентів

Wazuh реалізує кілька механізмів захисту між своїми компонентами:

1. Шифрування трафіку:

– агент ↔ сервер — власний протокол з AES-шифруванням;

– сервер ↔ indexer — TLS (Filebeat → indexer);

– dashboard ↔ сервер/indexer — HTTPS/TLS.

2. Взаємна аутентифікація:

– агенти отримують унікальні ключі при реєстрації;

– доступ до API захищається токенами/обліковими записами.

3. Сегментація мережі:

– відкриваються лише необхідні порти:

- 1514/TCP — передавання даних від агентів;
- 1515/TCP — реєстрація агентів;
- 9200/TCP — REST API індексатора;
- 443/TCP — веб-інтерфейс Dashboard;
- 55000/TCP — API сервера.

– компоненти можуть бути розміщені в окремих VLAN/зонах безпеки.

Завдяки цьому Wazuh не лише контролює стан безпеки інфраструктури, а й сам відповідає вимогам до захисту систем моніторингу.

2.2 Розгортання та конфігурація гіпервізора VMware ESXi

Серед різноманіття рішень на ринку, VMware ESXi займає домінуючу позицію як еталонний гіпервізор корпоративного рівня. Перед тим як розглядати безпосередній процес розгортання цього продукту, важливо сформувати розуміння його внутрішньої архітектури, принципів функціонування ядра VMkernel, механізмів взаємодії з апаратним забезпеченням та специфіки управління ресурсами.

VMware ESXi належить до гіпервізорів першого типу (bare-metal), тобто встановлюється безпосередньо на апаратні ресурси сервера й не потребує попередньо інстальованої операційної системи загального призначення [57]. Фактично ESXi є тонким спеціалізованим програмним шаром, що виконує роль посередника між фізичним "залізом" та віртуальними машинами (ВМ) і забезпечує одночасне функціонування кількох ізольованих середовищ на одному сервері [58].

У термінології VMware ESXi часто позначають як vSphere Hypervisor, оскільки він є базовим компонентом платформи віртуалізації VMware vSphere [59]. Саме на ESXi-хостах запускаються всі ВМ, тоді як інші елементи екосистеми: vCenter Server, інструменти резервного копіювання, моніторингу та автоматизації — забезпечують централізоване керування, координацію й інтеграцію [60]. У сучасних дата-центрах ESXi використовується як стандартна платформа для консолідації серверів, побудови приватних хмар, тестових і навчальних стендів, а також для розміщення критично важливих сервісів [61].

2.2.1 Роль ESXi у віртуалізованій інфраструктурі

Перехід до віртуалізації пов’язаний із прагненням організацій зменшити кількість фізичних серверів, оптимізувати витрати на обладнання й електроенергію та підвищити гнучкість IT-інфраструктури [58]. У традиційній схемі одна апаратна платформа зазвичай обслуговувала одну серверну ОС та один застосунок, що призводило до низького рівня використання ресурсів.

Запровадження ESXi дозволяє [62]:

– консолідувати десятки логічних серверів у вигляді ВМ на меншій кількості фізичних хостів;

– динамічно розподіляти ресурси (процесорний час, оперативну пам’ять, дисковий простір) відповідно до поточного навантаження;

– мінімізувати простої завдяки механізмам міграції віртуальних машин між хостами без зупинки сервісів та використанню кластерів високої доступності;

– спростити адміністрування та оновлення інфраструктури, оскільки значна частина операцій виконується на рівні ВМ, а не фізичних серверів.

2.2.2 Архітектура VMware ESXi

Архітектура ESXi побудована за модульним принципом і включає кілька ключових шарів: ядро VMkernel, процеси користувацького рівня (user worlds), підсистеми зберігання та мережі, а також стек керування й інтеграції [63].

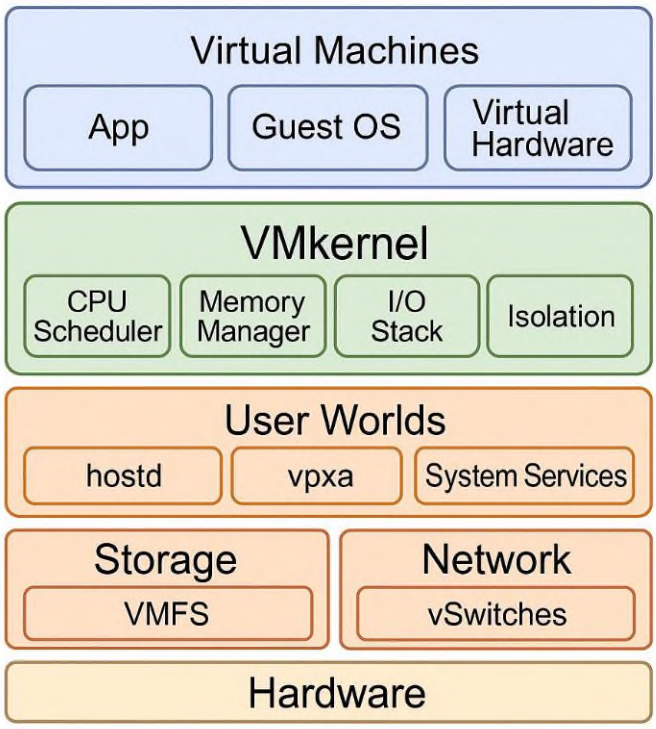


Рисунок 2.1 – Загальна модульна архітектура гіпервізора VMware ESXi

VMkernel як ядро гіпервізора

Центральним елементом ESXi є VMkernel — спеціалізоване ядро, оптимізоване під задачі віртуалізації [60]. Воно виконує функції:

– планувальника процесорного часу для віртуальних процесорів (vCPU) ВМ;

– диспетчера оперативної пам’яті, що розподіляє та оптимізує використання RAM між різними віртуальними машинами;

– менеджера вводу-виводу, відповідального за роботу з дисковими пристроями та мережевими інтерфейсами;

– механізму ізоляції, який гарантує, що збої або атаки у межах однієї ВМ не впливають на роботу інших [63].

VMkernel створює для кожної віртуальної машини стандартний набір віртуальних пристроїв — процесори, мережеві адаптери, контролери дисків, віртуальні порти тощо. Гостьова ОС "бачить" цю віртуальну конфігурацію як звичайний сервер, тоді як VMkernel прозоро відображає її на фізичні ресурси [4].

Worlds та системні процеси

Усередині ESXi усі сутності: віртуальні машини, системні служби, агенти моніторингу — представлені як worlds (аналог процесів) [63]. Для кожної активної ВМ створюється кілька таких worlds, що забезпечують обробку інструкцій гостьової ОС, доступ до пам’яті та роботу з мережевим і дисковим вводом-виводом.

Окрему групу становлять керуючі процеси [59]:

– hostd — локальний демон керування хостом;

– vpxa — агент взаємодії ESXi з сервером централізованого керування vCenter;

– додаткові агенти сторонніх рішень (резервне копіювання, моніторинг, інтеграція з апаратним забезпеченням).

Завдяки такій організації VMware досягає чіткої ізоляції служб та можливості гнучко керувати їхнім пріоритетом на рівні ядра.

Інтерфейси локального та віддаленого керування

Попри мінімалістичний дизайн, ESXi має вбудований Direct Console User Interface (DCUI) — консоль, що відображається на фізичному моніторі сервера [57]. Через DCUI адміністратор виконує початкову конфігурацію: задає пароль root, налаштовує IP-адресу й VLAN для мережі керування, увімкнення/вимкнення служб віддаленого доступу.

Для подальшого адміністрування використовують [61]:

– веб-клієнт Host Client (підключення безпосередньо до ESXi через браузер);

– vSphere Client / vCenter Server, якщо хост входить до кластера;

– ESXi Shell та SSH — для діагностики й виконання скриптів.

Підсистема зберігання даних

ESXi підтримує різні типи сховищ, де зберігаються файли віртуальних машин [62]:

– локальні диски сервера (SATA, SAS, NVMe);

– мережеві SAN-сховища на базі Fibre Channel або iSCSI;

– NFS-сховища, що доступні по мережі як файлові ресурси.

Для спільного доступу кількох хостів до одного LUN використовується файлова система VMFS (VMware File System) — спеціалізована кластерна ФС, що забезпечує блоковий доступ до віртуальних дисків при одночасному монтуванні на кількох ESXi-хостах [60]. У VMkernel реалізовано модуль multipathing, який дозволяє використовувати декілька фізичних шляхів до сховища, підвищуючи відмовостійкість і даючи змогу балансувати навантаження [62].

Мережева підсистема

Мережева частина ESXi складається з віртуальних комутаторів та службових інтерфейсів [58]. Основні поняття:

– Standard vSwitch — віртуальний комутатор, що працює на одному ESXi-хості;

– Distributed vSwitch (VDS) — розподілений комутатор, конфігурація якого централізовано зберігається у vCenter;

– Port Group — логічна група портів на віртуальному комутаторі;

– VMkernel-інтерфейси (vmk) для службового трафіку — керування, vMotion, доступ до iSCSI/NFS-сховищ, реплікації тощо.

ESXi підтримує VLAN-сегментацію, політики обмеження пропускної здатності (traffic shaping) та вбудований брандмауер для контролю доступу до служб гіпервізора [61].

Стек керування та інтеграції

При підключенні до vCenter Server ESXi-хости об’єднуються в кластер, у якому реалізуються розширені функції: планування ресурсів (DRS), кластери високої доступності (HA), централізовані політики безпеки [59]. Взаємодія між хостом і vCenter відбувається через агента vpxa на ESXi та службу vpxd на vCenter [63].

Крім того, ESXi надає API та SDK для інтеграції зі сторонніми рішеннями — системами резервного копіювання, моніторингу, автоматизації (PowerCLI, REST-API тощо) [58].

2.2.3 Ключові можливості та особливості ESXi

Компактна та спеціалізована архітектура

Інсталяційний образ ESXi має порівняно невеликий розмір, оскільки містить лише необхідний мінімум компонентів: VMkernel, базові драйвери, консоль керування та системні служби [60]. Відсутність повноцінної ОС загального призначення зменшує площу потенційної поверхні атак, спрощує оновлення та позитивно впливає на стабільність. ESXi може встановлюватися на традиційні диски, а також на SD-карти або USB-накопичувачі, що звільняє основні дискові масиви для даних віртуальних машин [62].

Продуктивність та масштабованість

VMkernel оптимізовано для роботи з багатоядерними процесорами, великими обсягами пам’яті й високошвидкісними мережами [58]. Сучасні версії ESXi підтримують сотні логічних процесорів та терабайти оперативної пам’яті на один хост, що дозволяє розміщувати широкий спектр навантажень — від легких служб каталогів до продуктивних СУБД [59].

На рівні гіпервізора функціонують механізми [60]:

– Resource Pools — логічні групи ВМ із заданими гарантіями та лімітами ресурсів;

– технології оптимізації пам’яті (ballooning, прозоре спільне використання сторінок, компресія, свопінг);

– пріоритизація доступу до CPU і дисків для різних груп віртуальних машин.

Безпека та надійність

ESXi широко використовується в середовищах із підвищеними вимогами до безпеки – банківських, телекомунікаційних, промислових [57]. На рівні гіпервізора реалізовано:

– Role-Based Access Control (RBAC) — гнучку систему ролей і дозволів;

– централізоване ведення журналів аудиту з можливістю експорту до SIEM-систем;

– підтримку Secure Boot і перевірку цифрових підписів модулів та оновлень;

– шифрування віртуальних дисків та vMotion-трафіку (за наявності відповідних ліцензій vSphere) [61].

Архітектура ESXi забезпечує жорстку ізоляцію ВМ: збій або компрометація однієї машини не впливають на інші, за умови коректного налаштування мережевих політик і розмежування доступу [60].

Інтеграція з корпоративною інфраструктурою

Важливою перевагою ESXi є глибока інтеграція з устаткуванням провідних виробників серверів, які постачають OEM-збірки ESXi з попередньо вбудованими драйверами та інструментами моніторингу [62]. Також гіпервізор підтримує взаємодію з популярними системами резервного копіювання та моніторингу, що використовують спеціалізовані API для створення знімків ВМ без зупинки сервісів [58].

2.2.4 Процес інсталяції та первинної конфігурації VMware ESXi

Перед початком розгортання було виконано підготовку апаратного забезпечення. Критично важливою умовою для функціонування гіпервізора є активація технологій апаратної віртуалізації (Intel VT-x або AMD-V) у

налаштуваннях BIOS/UEFI сервера. Без цієї опції запуск 64-бітних гостьових операційних систем буде неможливим.

Процедура розгортання гіпервізора на підготовленому апаратному забезпеченні складається з двох логічних етапів: безпосередньої інсталяції ядра системи та базового налаштування інтерфейсу керування (DCUI).

Етап 1: Інсталяція гіпервізора

1. Завантаження інсталятора: після вибору пріоритету завантаження (Boot Priority) в BIOS з USB-носія, розпочинається ініціалізація завантажувача ESXi. За замовчуванням системний розділ ESXi займає 120 Гб для зберігання різних системних даних, проте це занадто багато, тому зменшуємо даний розділ. Для цього натискаємо комбінацію клавіш Shift + O, після чого стає доступним командний рядок, в якому прописуємо додатково команду systemMediaSize=min і натискаємо клавішу Enter (рисунок 2.2). Система автоматично пропише мінімальний об’єм системного розділу при подальшому встановленні.

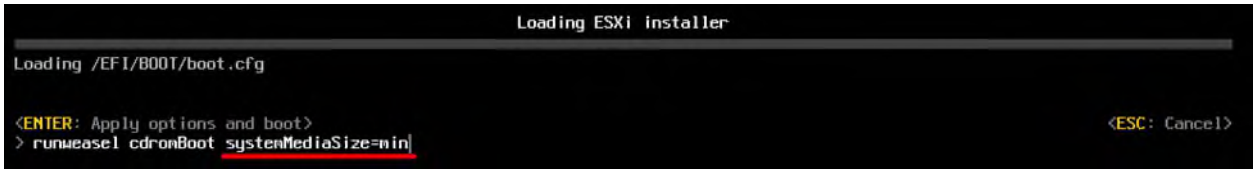


Рисунок 2.2 – Ініціалізація завантажувача ESXi

Далі на екрані відображається процес розпакування модулів ядра в оперативну пам'ять (рисунок 2.3), після чого у верхній частині екрана зазначається версія ядра (Build Number) та конфігурація фізичного процесора і пам'яті (рисунок 2.4).

Рисунок 2.3 – Розпакування модулів ядра в оперативну пам'ять

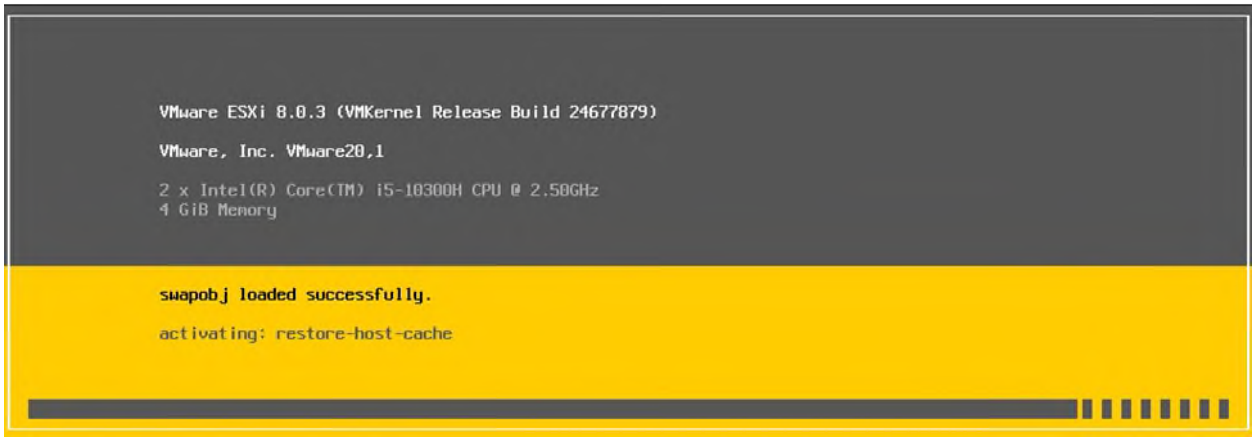


Рисунок 2.4 – Вікно версії ядра та конфігурації фізичного процесора і пам'яті

2. Привітання та перевірка сумісності: з'являється екран привітання (рисунок 2.5). Натискання клавіші Enter ініціює перевірку сумісності обладнання. Якщо виявлено критичні невідповідності (наприклад, відсутність апаратної віртуалізації або несумісний мережевий контролер), інсталятор зупинить роботу.

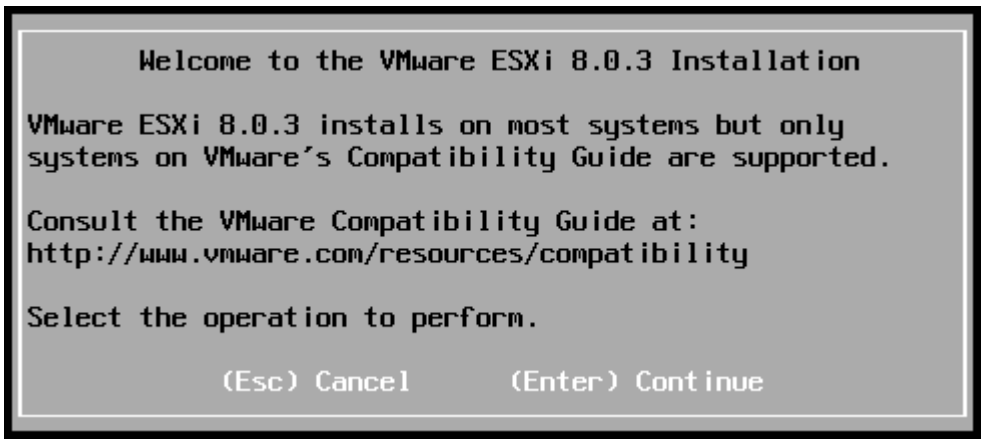


Рисунок 2.5 – Вікно привітання ESXi

3. Ліцензійна угода (EULA): для продовження необхідно прийняти умови ліцензійної угоди (рисунок 2.6), натиснувши клавішу F11.

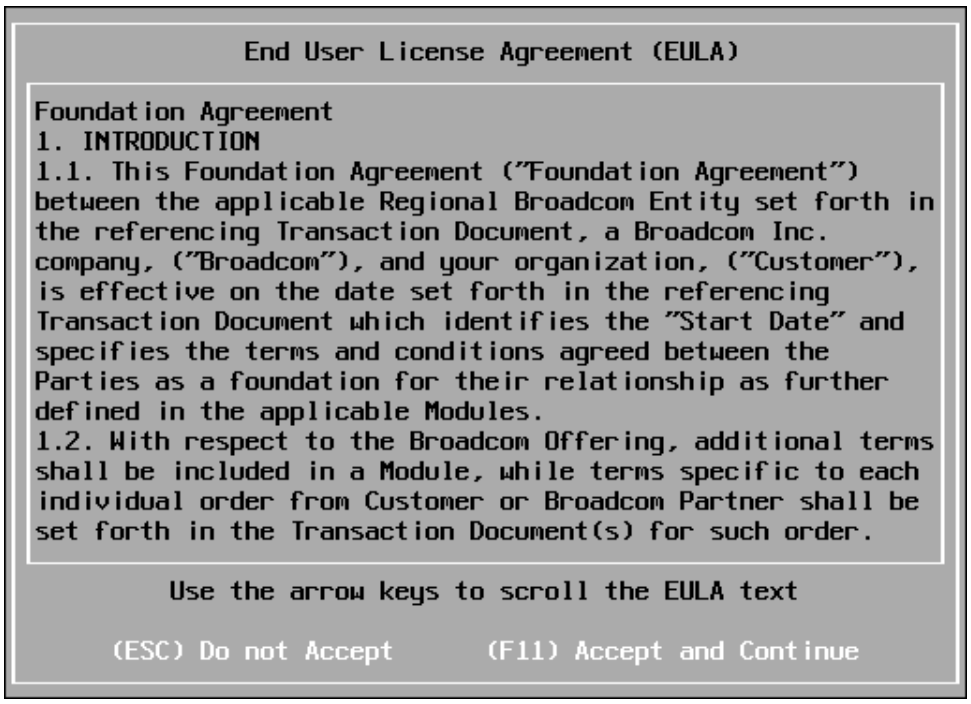


Рисунок 2.6 – Вікно умови ліцензійної угоди (EULA)

4. Вибір дискового простору: інсталятор сканує доступні інтерфейси зберігання даних. Необхідно обрати цільовий диск та натиснути клавішу Enter (рисунок 2.7).

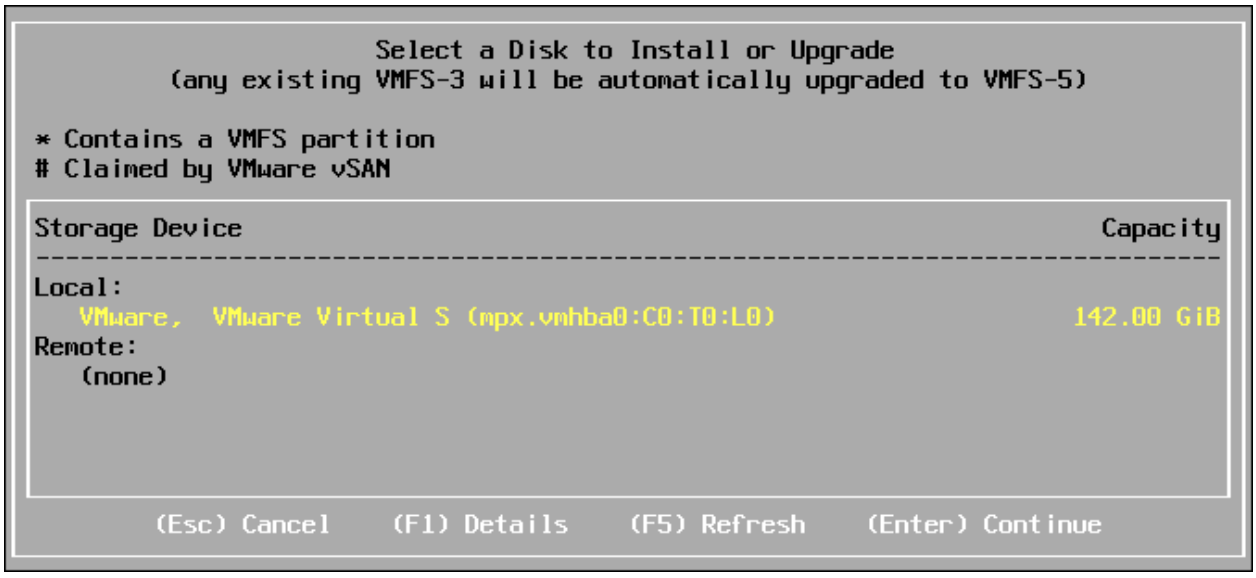


Рисунок 2.7 – Вікно вибору дискового простору

Якщо на диску вже є дані, з'явиться попередження про їх повне знищення, оскільки інсталятор перерозмітить диск під файлову систему VMFS.

5. Налаштування клавіатури: обираємо розкладку клавіатури і натискаєсо клавішу Enter (рисунок 2.8). Для серверних систем стандартом є US Default.

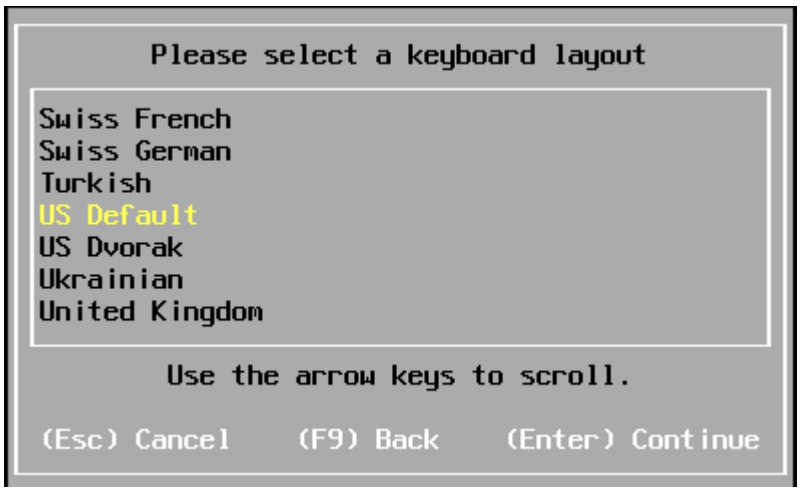


Рисунок 2.8 – Вікно налаштування клавіатури

6. Встановлення пароля адміністратора: задаємо пароль для суперкористувача root та натискаємо клавішу Enter (рисунок 2.9). Система автоматично перевіряє складність пароля. Він має містити мінімум 7 символів та включати три з чотирьох категорій знаків: великі літери, малі літери, цифри та спеціальні символи.

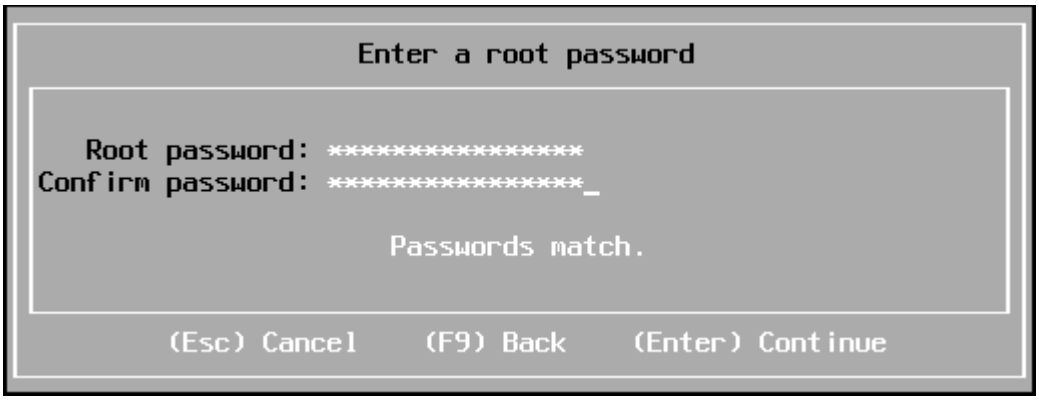


Рисунок 2.9 – Вікно встановлення пароля адміністратора

7. Фіналізація: натискання клавіші F11 (рисунок 2.10) запускає процес копіювання файлів та конфігурації завантажувального сектора (рисунок 2.11). Процес займає кілька хвилин.

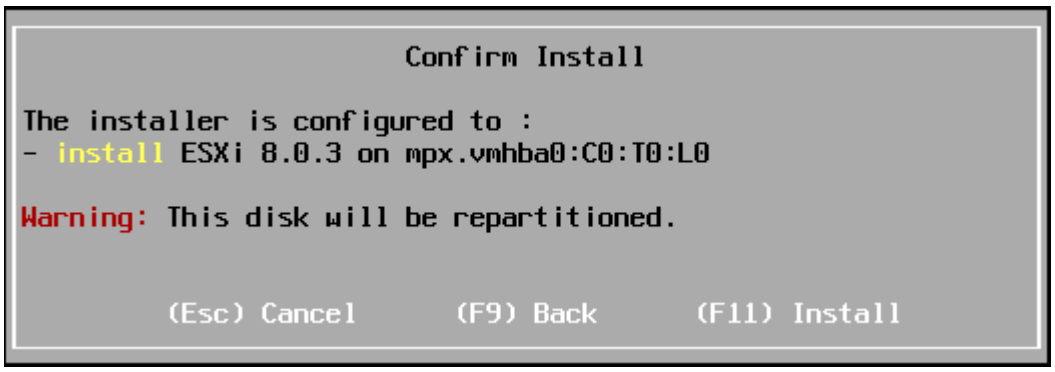


Рисунок 2.10 – Вікно підготовки фінального завантаження

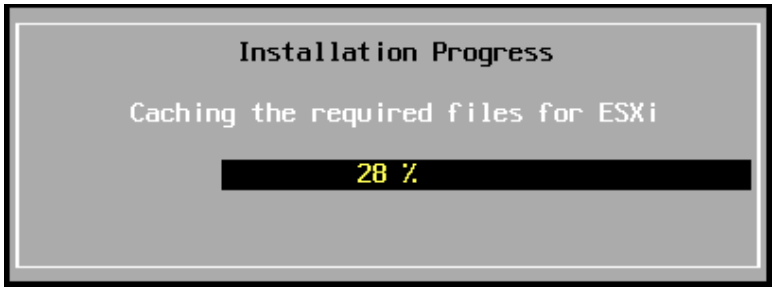


Рисунок 2.11 – Вікно процесу копіювання файлів та конфігурації завантажувального сектора

8. Завершення та перезавантаження: після успішної інсталяції з'являється повідомлення "Installation Complete" (рисунок 2.12). Необхідно витягти інсталяційний носій та натиснути клавішу Enter для перезавантаження ESXi (рисунок 2.13).

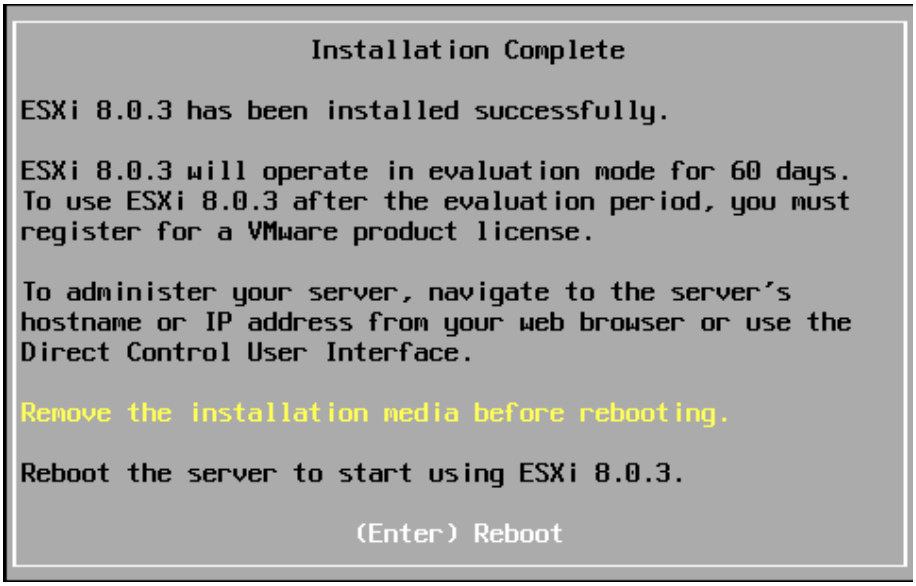


Рисунок 2.12 – Вікно повідомлення про успішну інсталяцію

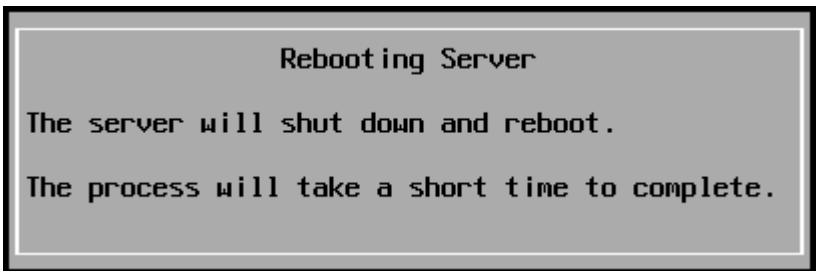


Рисунок 2.13 – Вікно повідомлення про перезавантаження ESXi

Етап 2: Конфігурація мережі керування (DCUI)

Після перезавантаження завантажується Direct Console User Interface (DCUI) — чорно-жовтий консольний інтерфейс (рисунок 2.14).

Рисунок 2.14 – Direct Console User Interface (DCUI)

За замовчуванням ESXi отримав IP-адресу через DHCP, однак для серверної інфраструктури, особливо для систем безпеки, необхідна статична адресація.

Процес налаштування:

1. Вхід у режим налаштування: натискаємо F2 (Customize System/View Logs) та вводимо логін root і створений раніше пароль (рисунок 2.15), після чого натискаємо клавішу Enter.

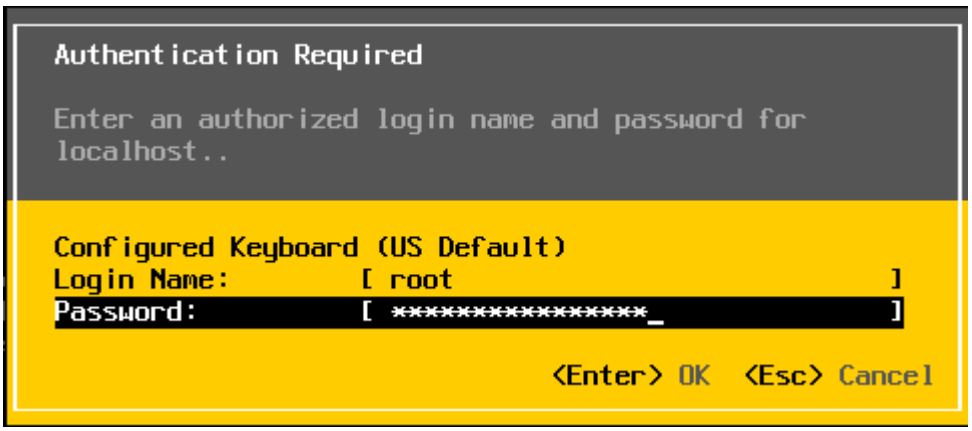


Рисунок 2.15 – Вікно входу у режим налаштування з попередньою автентифікацією

2. Налаштування мережі: у меню обираємо пункт Configure Management Network і натискаємо клавішу Enter (рисунок 2.16). Далі у меню, що розкрилося обираємо IPv4 Configuration та натискаємо клавішу Enter (рисунок 2.17).

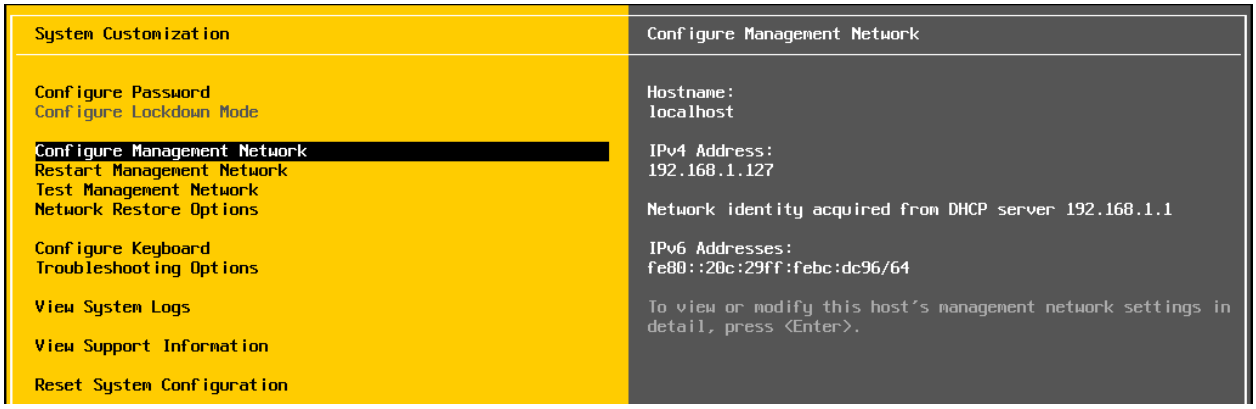


Рисунок 2.16 – Вікно налаштувань System Customization

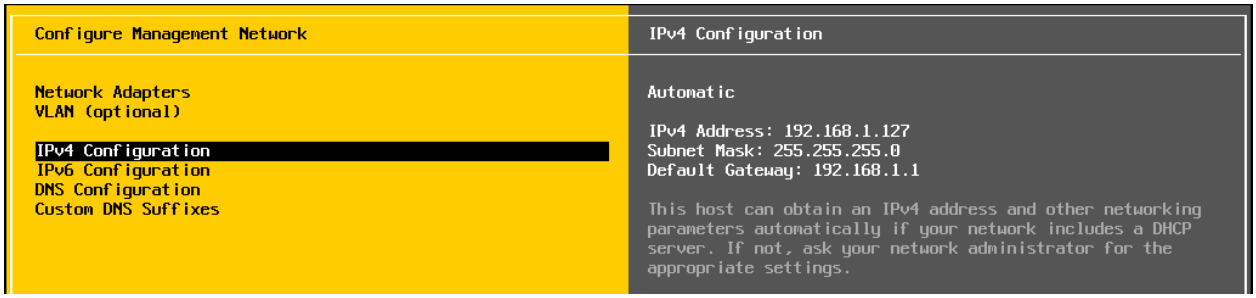


Рисунок 2.17 – Вікно налаштувань мережі Configure Management Network

3. Встановлення статичної IP-адреси: змінюємо перемикач з "Use dynamic IPv4 address" на "Set static IPv4 address and network configuration" (рисунок 2.18).

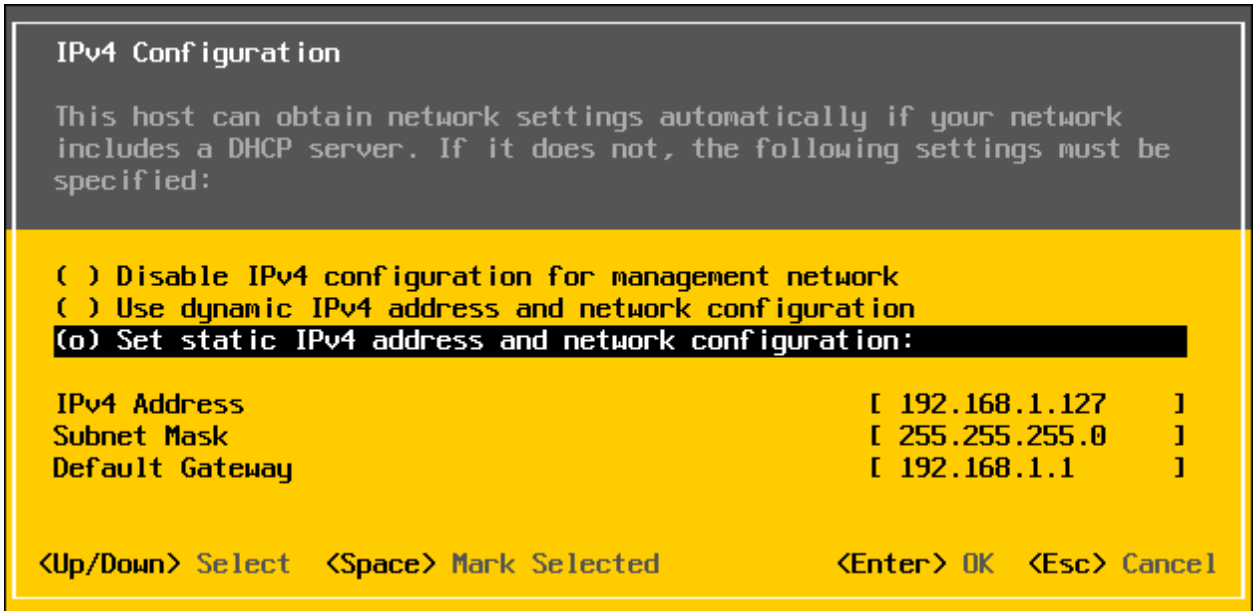


Рисунок 2.18 – Вікно налаштувань IPv4 Configuration

4. Налаштування DNS: у меню DNS Configuration змінюємо перемикач з "Obtain DNS server addresses and a hostname automatically" на "Use the following DNS server addresses and hostname" і вказуємо адресу шлюзу та ім'я хоста (рисунок 2.19).

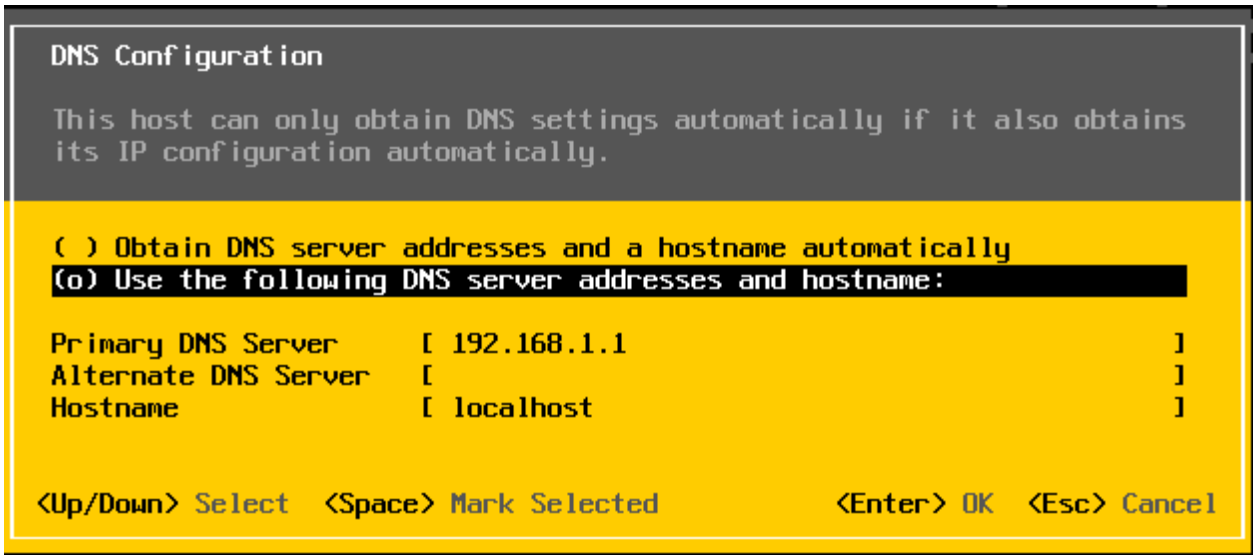


Рисунок 2.19 – Вікно налаштувань DNS Configuration

5. Застосування змін: натискаємо Esc для виходу. Система запитає підтвердження на перезапуск мережевих агентів. Натискаємо Y (рисунок 2.20).

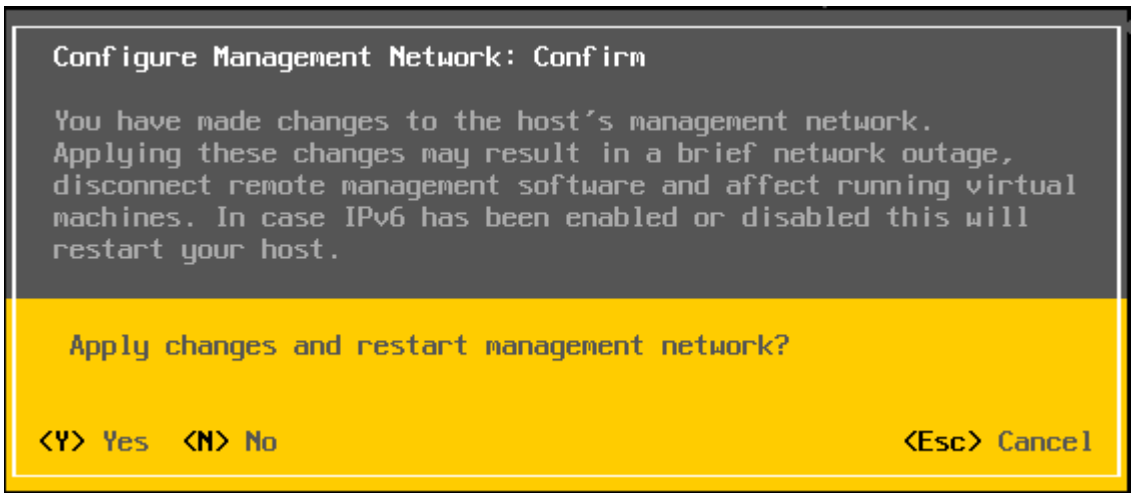


Рисунок 2.20 – Вікно підтвердження мережевих налаштувань

Після перезавантаження можемо побачити оновлену інформацію про конфігурацію ESXi в DCUI (рисунок 2.21).

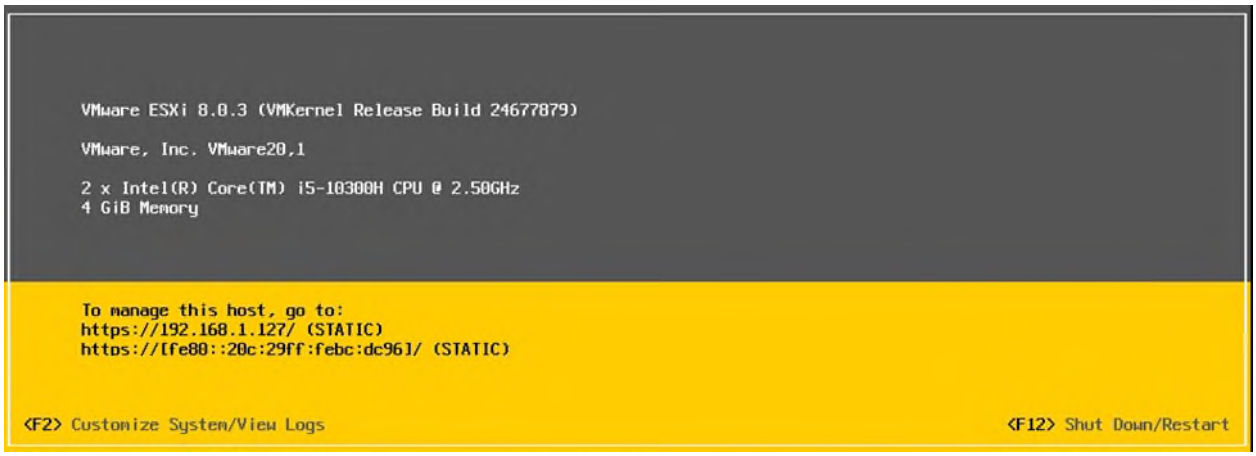


Рисунок 2.21 – DCUI з оновленою інформацією про конфігурацію ESXi

В результаті ми маємо повністю функціональний гіпервізор, який доступний у мережі. Тепер фізичний доступ до сервера більше не потрібен — все подальше керування, створення віртуальних машин та налаштування моніторингу здійснюватиметься віддалено через веб-браузер.

2.2.5 Доступ до інтерфейсу керування VMware Host Client

Після успішної конфігурації мережі керування подальша взаємодія з гіпервізором здійснюється через веб-інтерфейс VMware Host Client. Це повністю функціональний HTML5-клієнт, який дозволяє керувати хостом, віртуальними машинами, сховищем та мережею.

1. Встановлення захищеного з'єднання: для доступу до панелі керування необхідно у веб-браузері робочої станції ввести IP-адресу, задану на попередньому етапі (https://192.168.1.127).

Варто зазначити технічну особливість першого входу: оскільки ESXi використовує самопідписаний SSL-сертифікат (Self-signed certificate), згенерований автоматично під час інсталяції, браузер видасть попередження про "незахищене з'єднання". У рамках локального лабораторного стенду це є нормою, тому необхідно додати виняток безпеки та продовжити завантаження сторінки.

2. Авторизація та огляд інформаційної панелі: для входу використовуються облікові дані суперкористувача root (рисунок 2.22). Після авторизації відкривається головна інформаційна панель, яка надає консолідовану інформацію про стан хоста (рисунок 2.23).

Рисунок 2.22 – Вхід до панелі керування через веб-браузер робочої станції

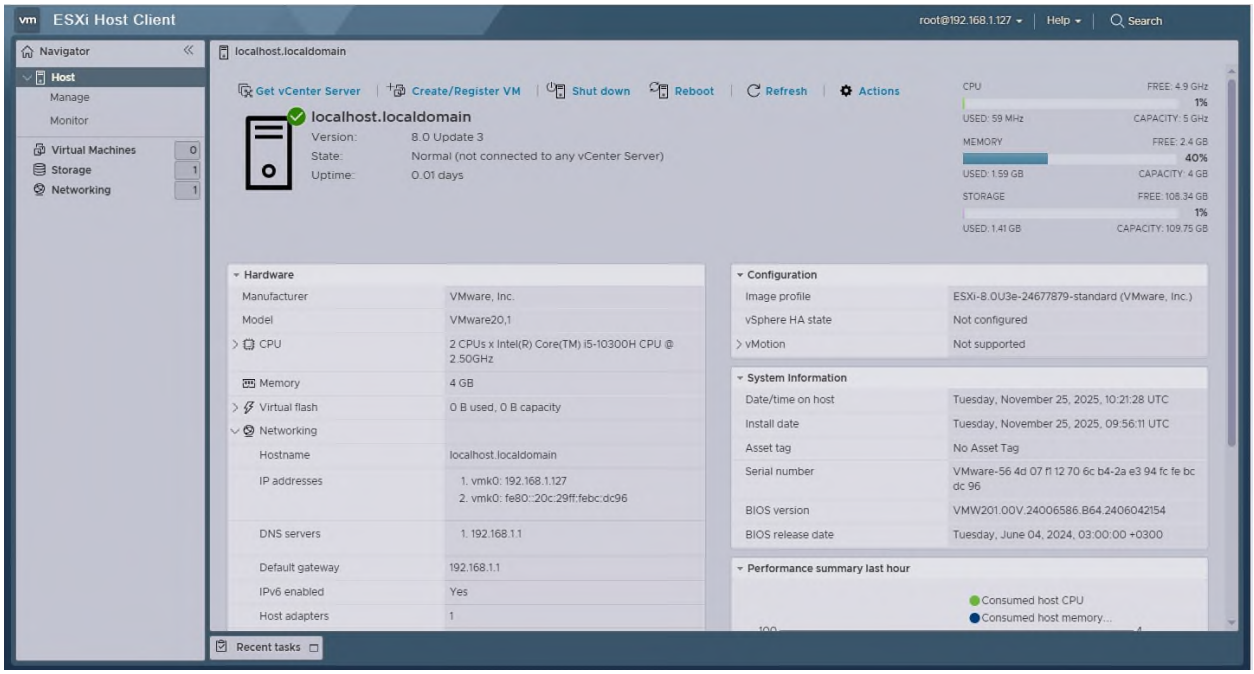


Рисунок 2.23 – Головна інформаційна панель ESXi Host Client

2.2.6 Підготовка сховища даних

Перед створенням віртуальних машин необхідно завантажити образи операційних систем (ISO-файли) на локальне сховище сервера. ESXi використовує логічну структуру зберігання даних, що називається Datastore.

Процес підготовки виконується наступним чином:

1. У навігаційному меню обираємо розділ Storage (рисунок 2.24).

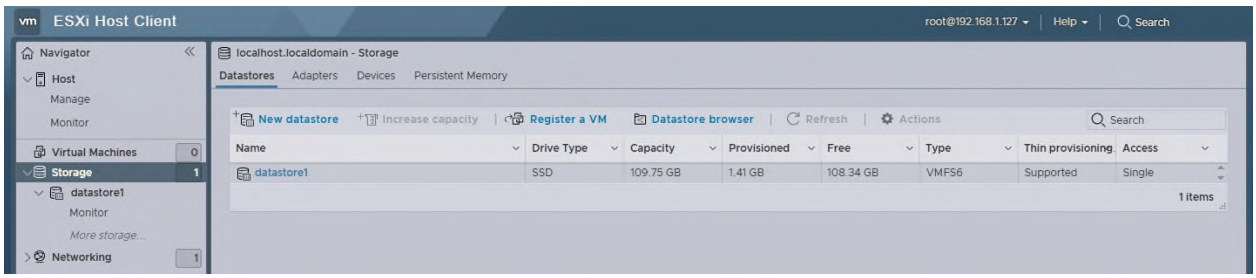


Рисунок 2.24 – Розділ Storage у навігаційному меню ESXi Host Client

2. Відкриваємо вкладку Datastore Browser (рисунок 2.25), куди за допомогою функції Upload завантажуємо необхідні образи.

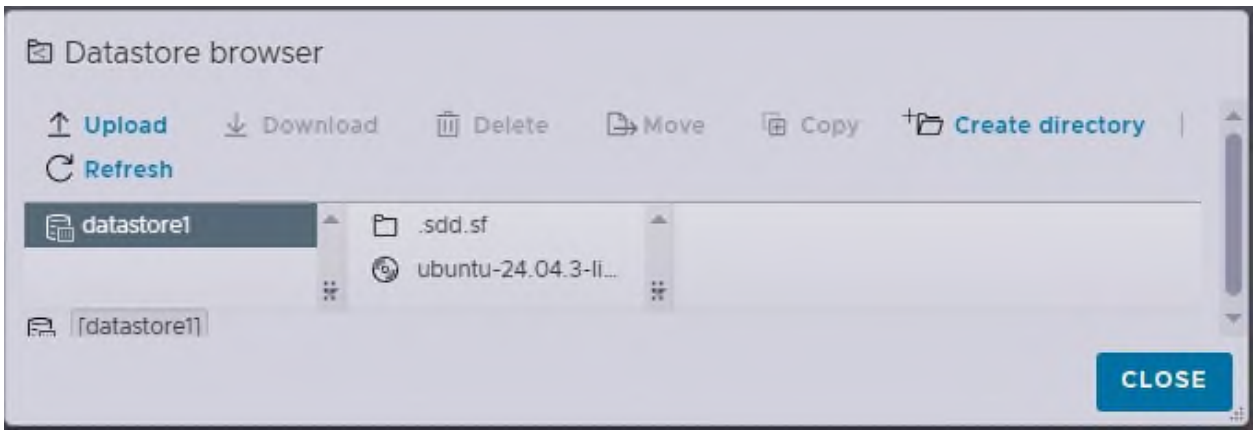


Рисунок 2.25 – Вікно Datastore Browser

Цей крок є завершальним у підготовці платформи. Гіпервізор розгорнуто, мережу налаштовано, інсталяційні файли завантажено. Система готова до створення віртуальної інфраструктури.

2.3 Розгортання та конфігурація Wazuh на базі Ubuntu

Після успішного налаштування гіпервізора наступним етапом є створення віртуальної машини (ВМ), яка виконуватиме роль центрального сервера системи моніторингу Wazuh. Як базову операційну систему було обрано Ubuntu Server 22.04 LTS. Цей вибір зумовлений стабільністю версії (Long Term Support), широкою підтримкою спільноти та повною сумісністю з компонентами Wazuh.

2.3.1 Створення віртуальної машини в VMware Host Client

Процес створення нової сутності у віртуальній інфраструктурі виконується через майстер налаштувань:

1. Ініціалізація: у навігаційній панелі обираємо розділ Virtual Machines та натискаємо Create / Register VM (рисунок 2.26).

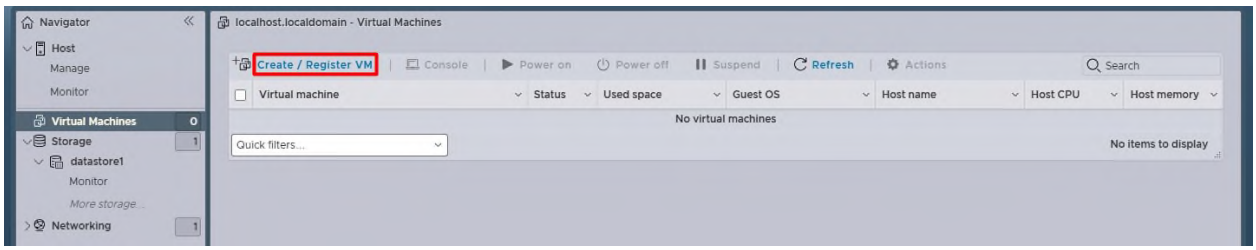


Рисунок 2.26 – Розділ Virtual Machines у навігаційному меню ESXi Host Client

2. Вибір типу створення: у меню обираємо Create a new virtual machine та натискаємо Next (рисунок 2.27).

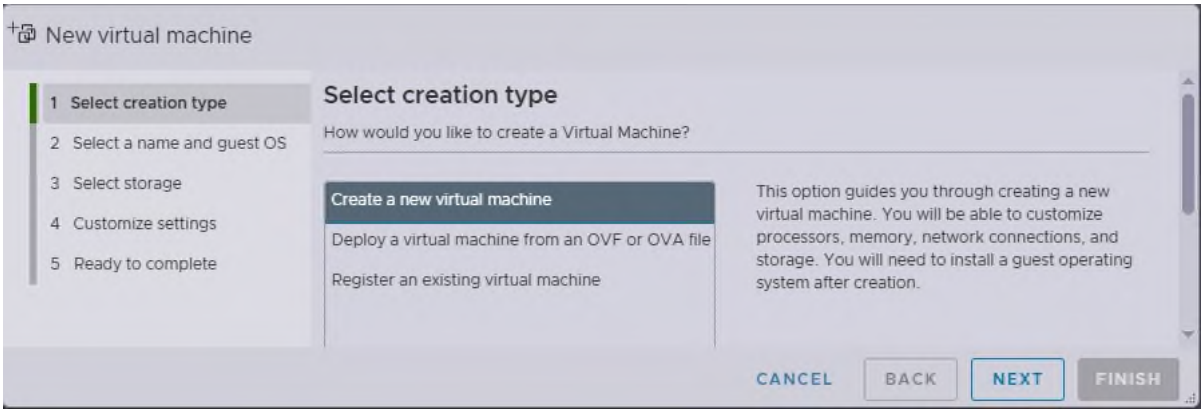


Рисунок 2.27 – Вікно вибору типу створення

3. Вибір імені та гостьової ОС: у полі Name прописуємо Wazuh, у Guest OS family обираємо серед запропонованого списку Linux, а у Guest OS version — Ubuntu Linux (64-bit) та натискаємо Next (рисунок 2.28).

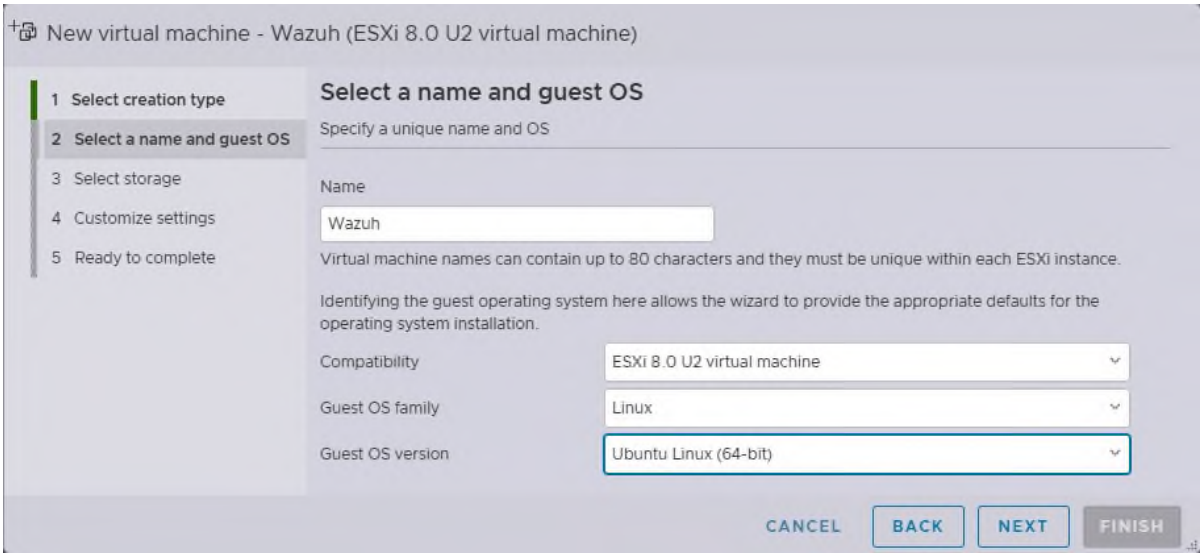


Рисунок 2.28 – Вікно вибору імені та гостьової ОС

4. Вибір сховища: обираємо локальний Datastore, підготовлений на попередніх етапах та натискаємо Next (рисунок 2.29).

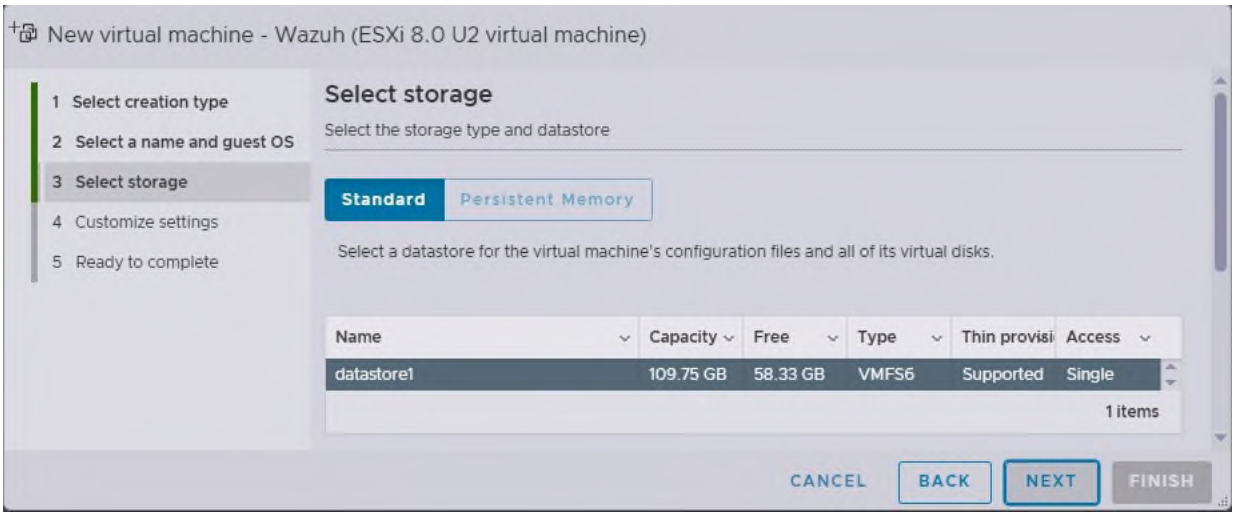
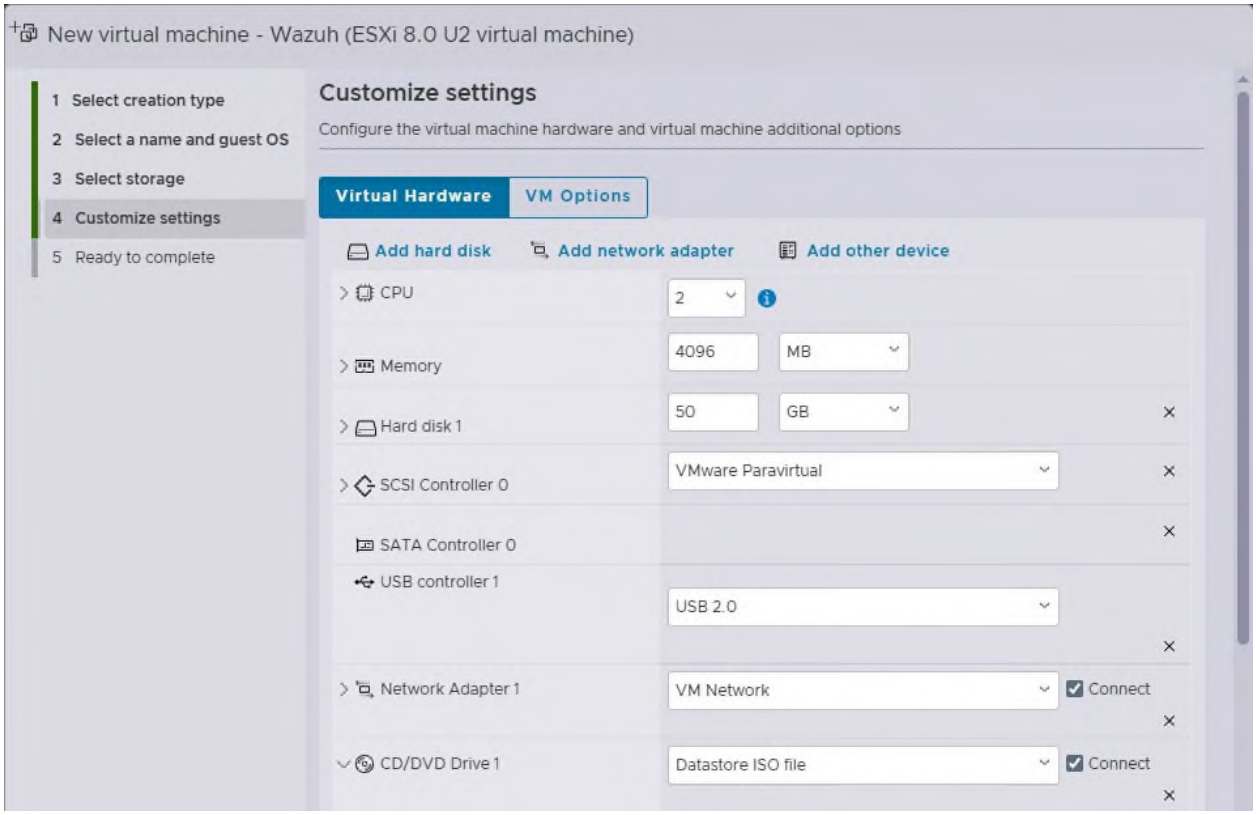


Рисунок 2.29 – Вікно вибору сховища

5. Налаштування ресурсів: виходячи з обчислювальних можливостей розгорнутого гіпервізора ESXi, було виділено наступні ресурси для створення ВМ, які представлені на рисунку 2.30. Після перевірки конфігурації натискаємо Finish. Нова ВМ з'явилася в інвентарі гіпервізора (рисунок 2.31). Запуск ВМ виконується кнопкою Power on



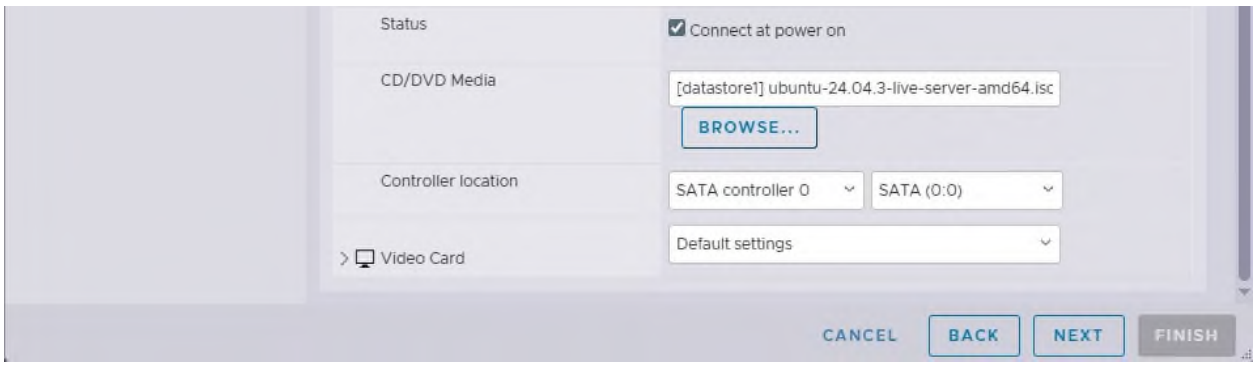


Рисунок 2.30 – Вікно налаштування ресурсів

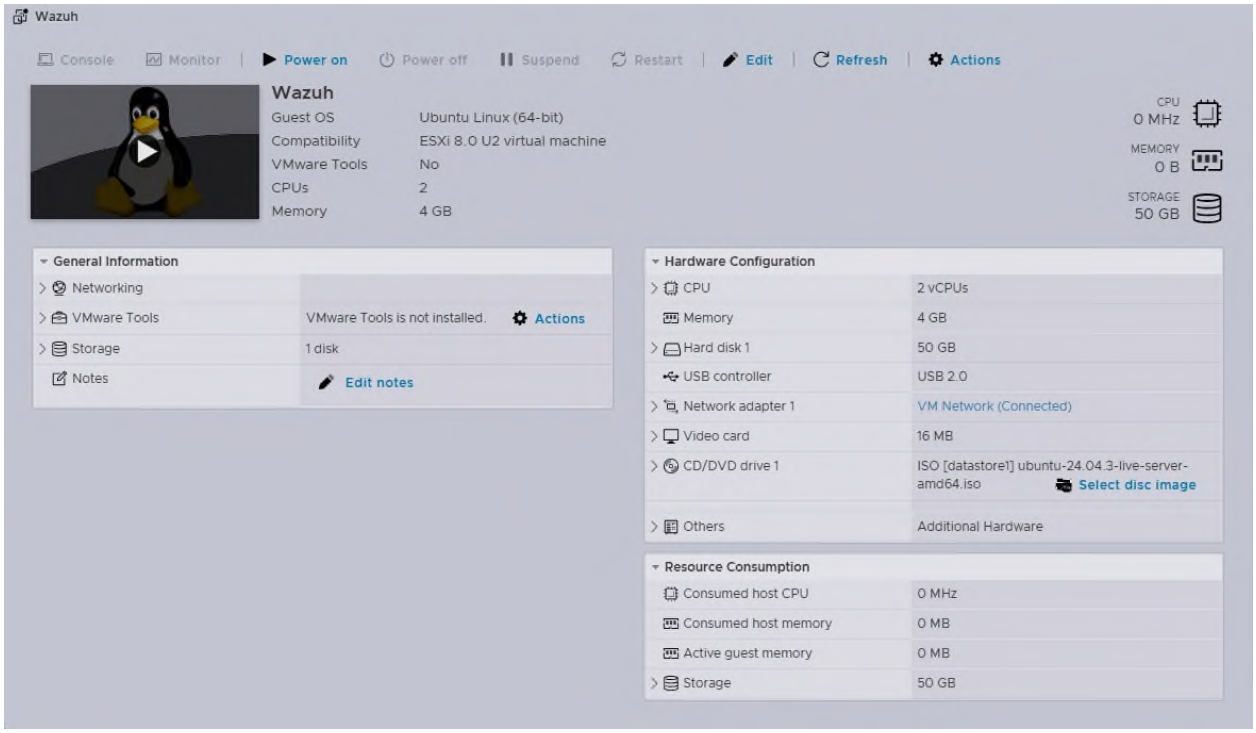


Рисунок 2.31 – Створена ВМ в інвентарі гіпервізора ESXi

2.3.2 Інсталяція та первинна конфігурація Ubuntu Server

Після завершення підготовки виконується завантаження ВМ з ISO-образу, що запускає інсталятор Subiquity, який використовується саме у версії Ubuntu Server 24.04 LTS. Далі Subiquity автоматично ініціалізує драйвери, завантажує модулі ядра та готує текстовий інтерфейс інсталяції (рисунок 2.32). На цьому етапі перевіряється сумісність віртуального обладнання (мережа, диски, контролери).

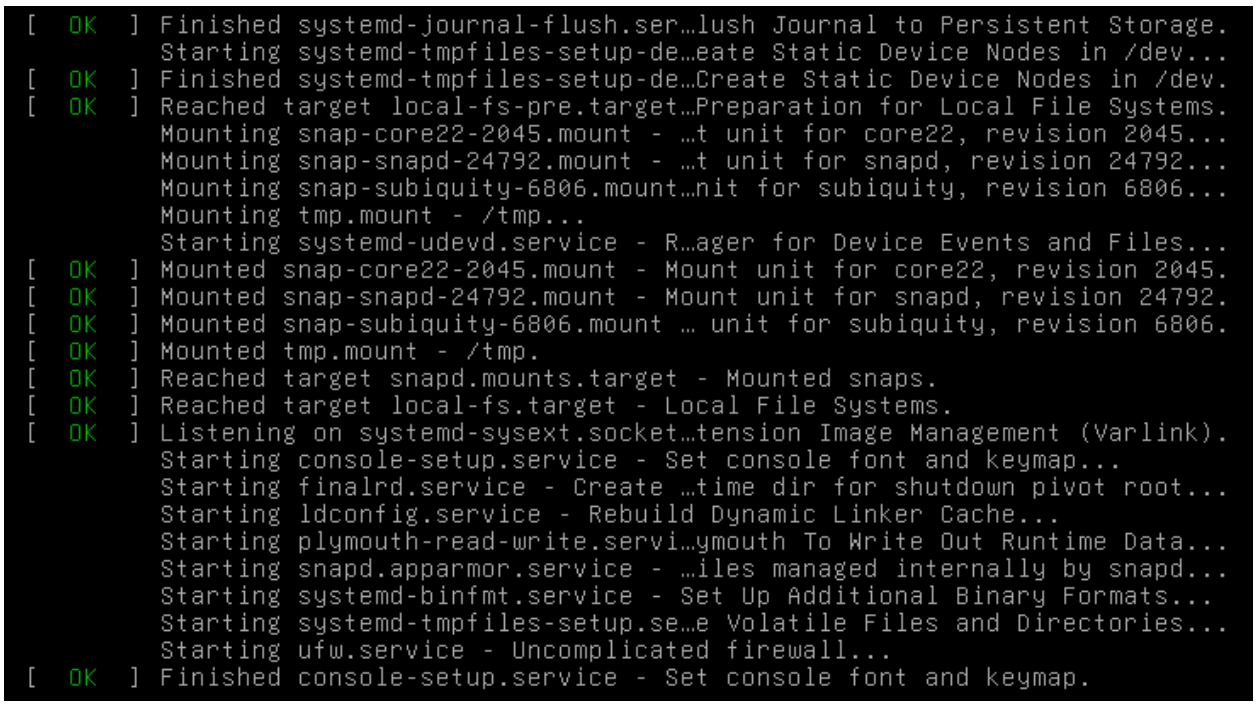


Рисунок 2.32 – Процес ініціалізації драйверів та завантаження модулів ядра

Ключові етапи налаштування під час встановлення:

1. Вибір мови інсталятора та клавіатури: обираємо English (US) для уникнення проблем з кодуванням у логах та терміналі (рисунок 2.33).

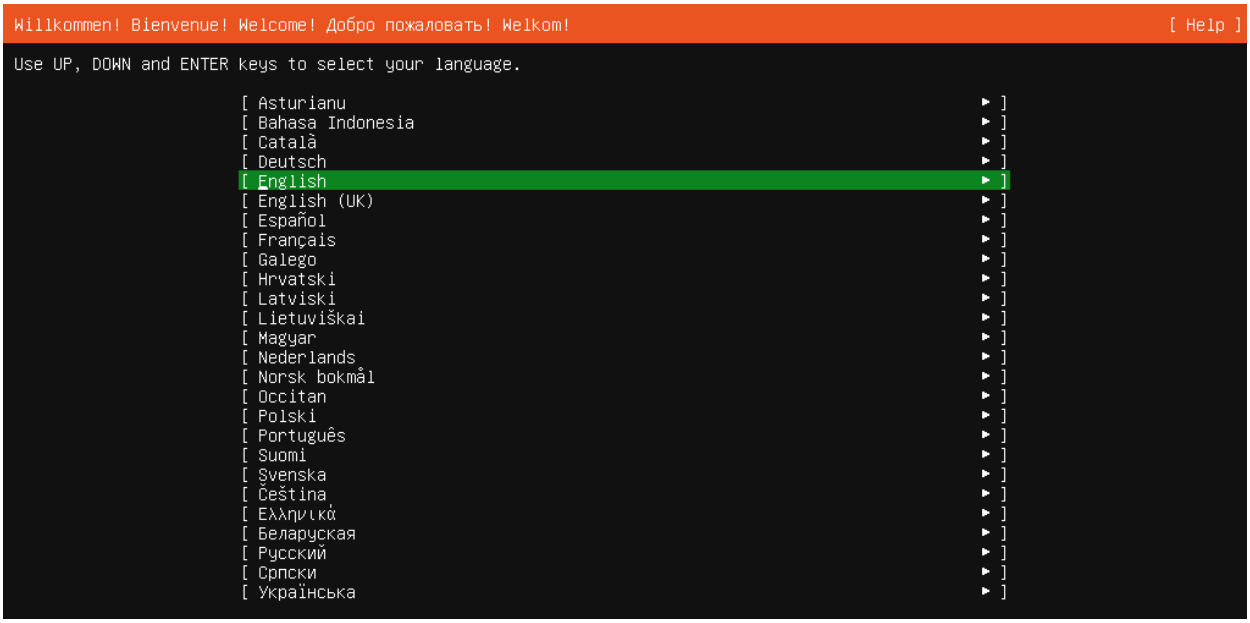


Рисунок 2.33 – Вікно вибору мови інсталятора та клавіатури

2. Вибір типу інсталяції: обираємо Ubuntu Server (рисунок 2.34).

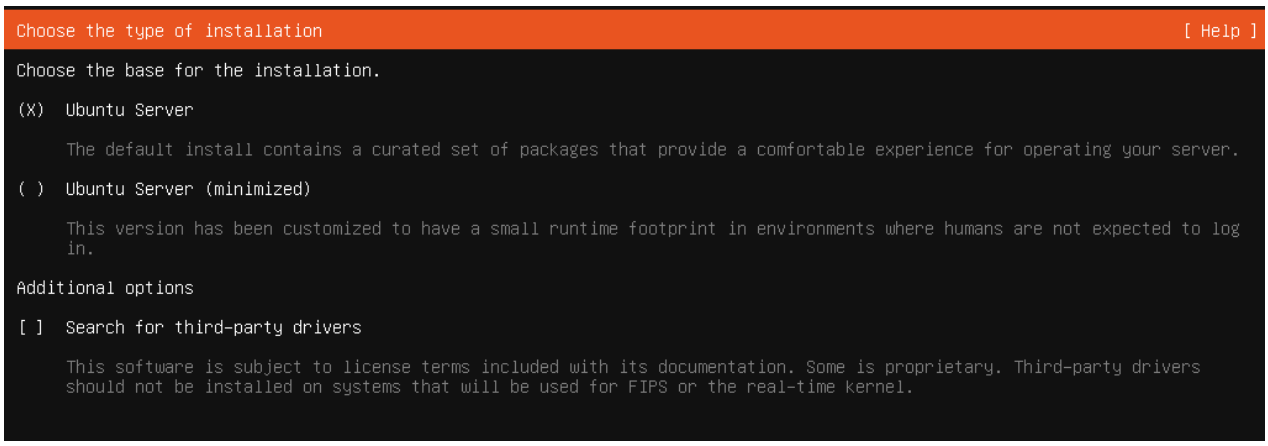


Рисунок 2.34 – Вікно вибору типу інсталяції

3. Налаштування мережевих параметрів: на цьому етапі залишено отримання адреси по DHCP, що представлена на рисунку 2.35 (зміна на статичну адресу буде виконана після інсталяції для демонстрації роботи з конфігураційними файлами)

Рисунок 2.35 – Вікно налаштування мережевих параметрів

4. Налаштування джерела репозиторіїв: пропонується вибір дзеркала репозиторію Ubuntu (рисунок 2.36). Для версії 24.04 використовується дзеркало archive.ubuntu.com, підібраний автоматично. Цей параметр визначає, з яких серверів операційна система надалі завантажуватиме пакети.

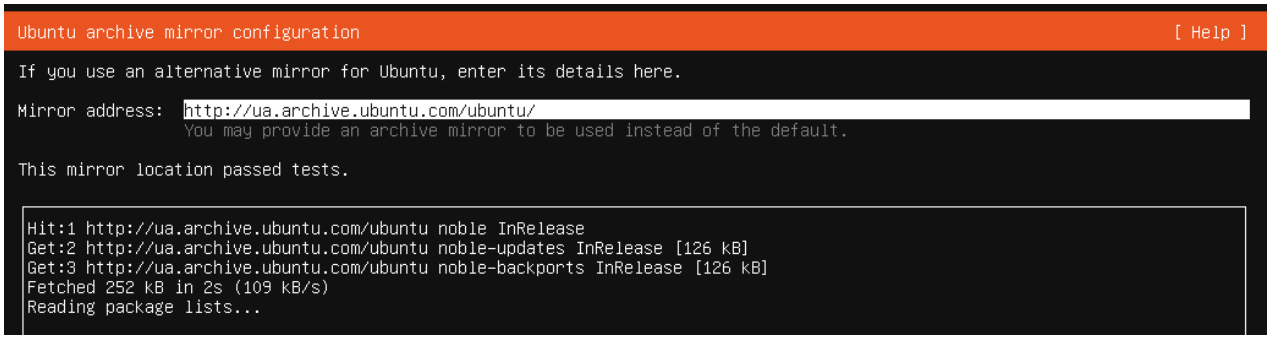


Рисунок 2.36 – Вікно налаштування джерела репозиторіїв

5. Розмітка диску: використано опцію "Use an entire disk" (рисунок 2.37) із створенням LVM-групи для гнучкого керування простором (рисунок 2.38).

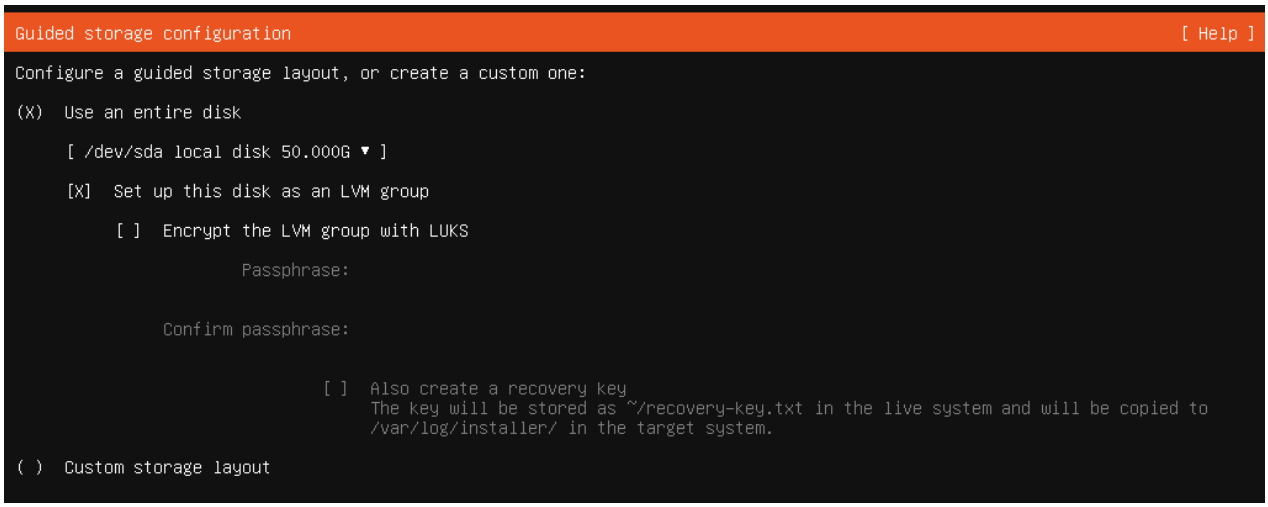


Рисунок 2.37 – Вікно вибору керованої конфігурації сховища

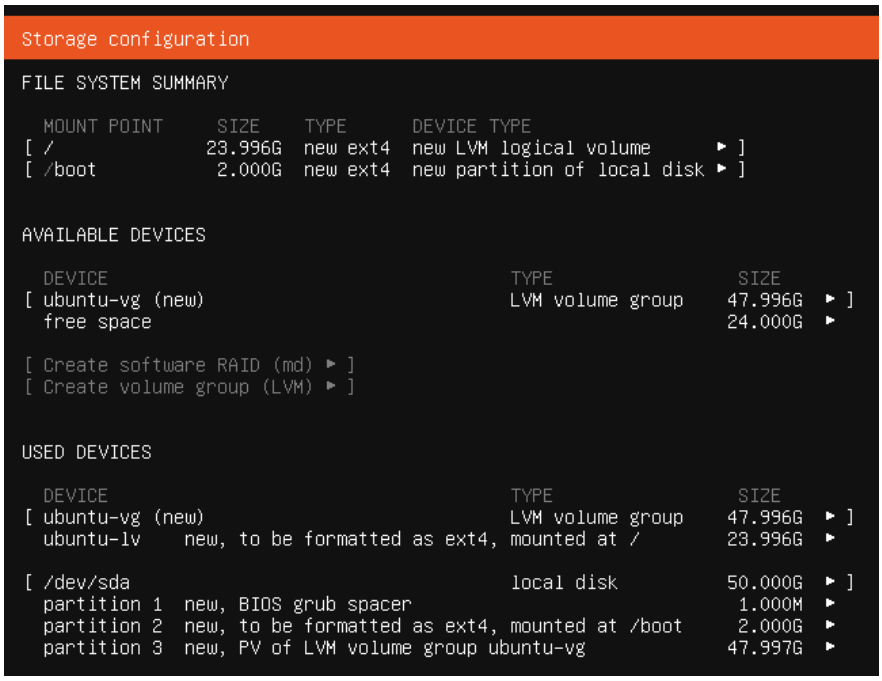


Рисунок 2.38 – Вікно налаштування конфігурації сховища

Після підтвердження (рисунок 2.39) інсталятор виконує форматування та створення файлових систем.

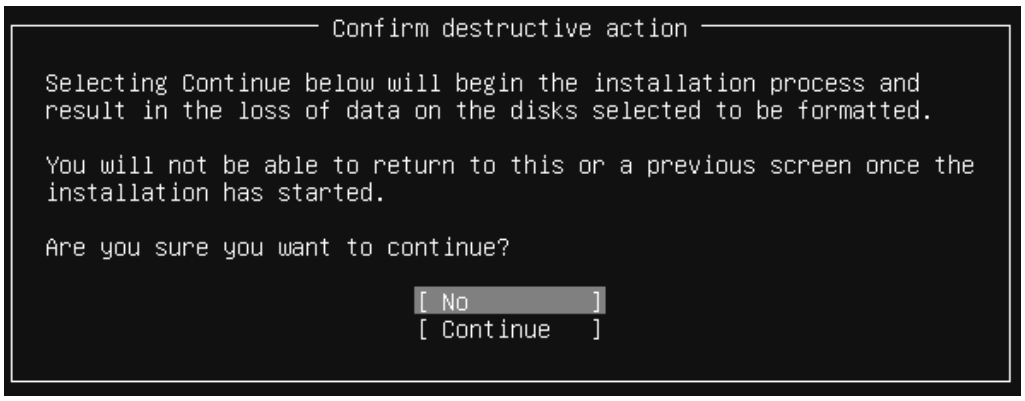


Рисунок 2.39 – Вікно підтвердження налаштування конфігурації сховища

6. Налаштування профіля користувача: створено адміністративного користувача, ім’я сервера та задано пароль (рисунок 2.40).

Рисунок 2.40 – Вікно налаштування профіля користувача

7. Налаштування SSH: активовано опцію Install OpenSSH server, що дозволить підключатися до сервера віддалено одразу після завершення інсталяції (рисунок 2.41).

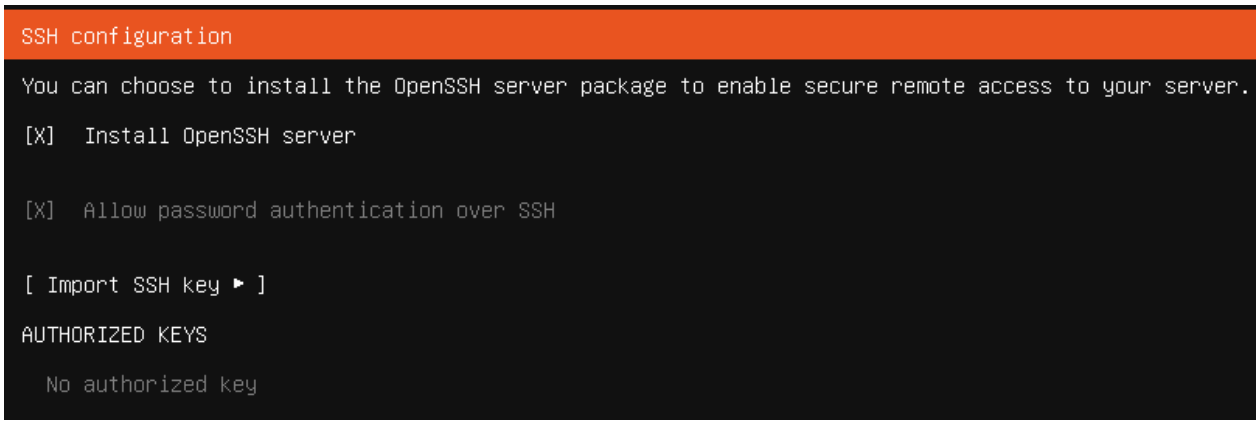


Рисунок 2.41 – Вікно налаштування SSH

8. Копіювання системних файлів та встановлення системи: після підтвердження конфігурації інсталятор переходить до копіювання системних файлів, встановлення ядра Linux, конфігурування завантажувача GRUB, створення файлових систем і розгортання базової структури системи (рисунок 2.42).

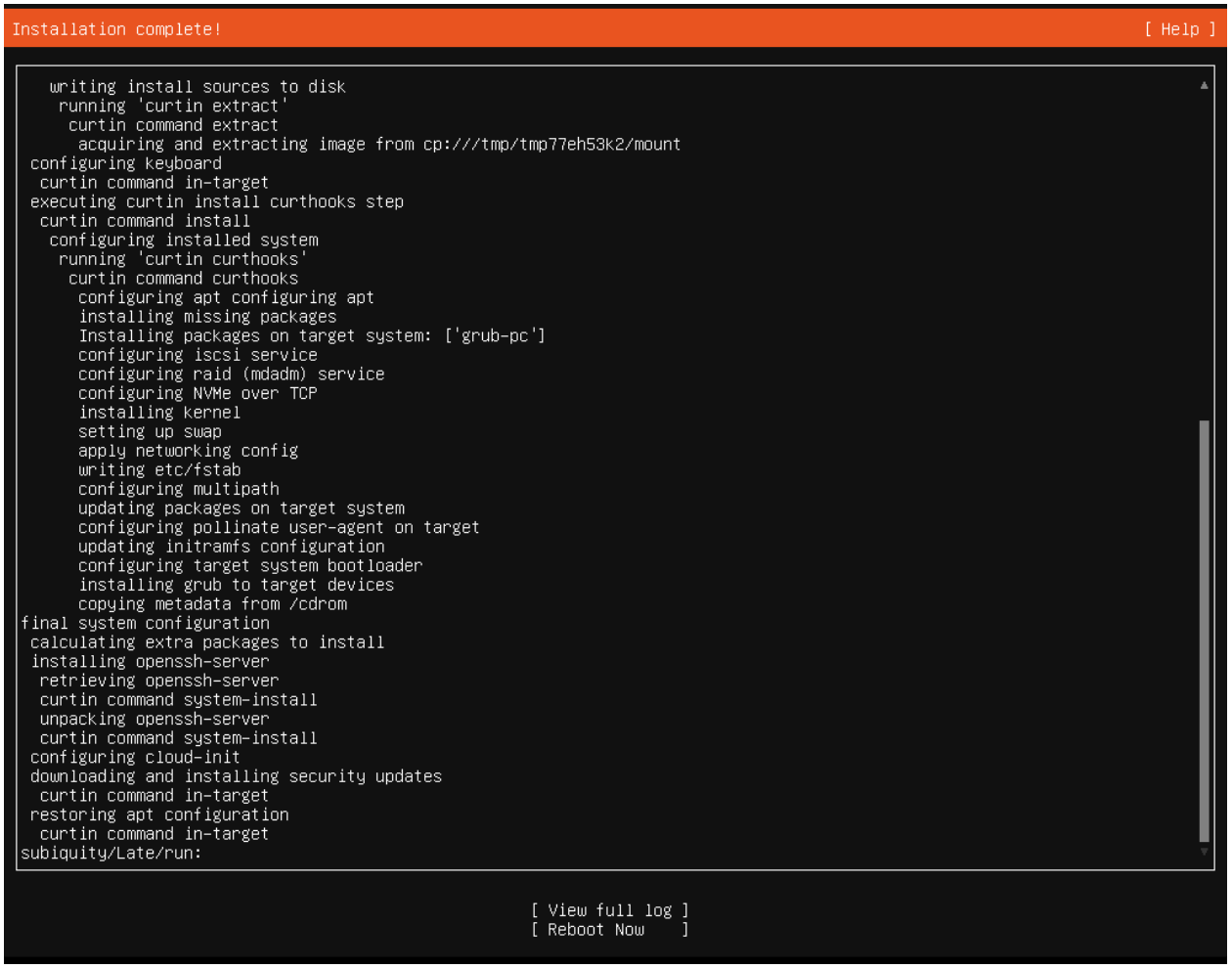


Рисунок 2.42 – Вікно копіювання системних файлів та встановлення системи

Після завершення встановлення інсталятор пропонує перезавантажити ВМ. Перед цим потрібно від’єднати ISO-образ, щоб уникнути повторного запуску інсталятора.

Після перезавантаження запускається нова система Ubuntu Server 24.04. Користувач бачить екран командного рядка для входу, де потрібно авторизуватися, ввівши пароль, який був створений під час налаштування користувача профіля (рисунок 2.43).

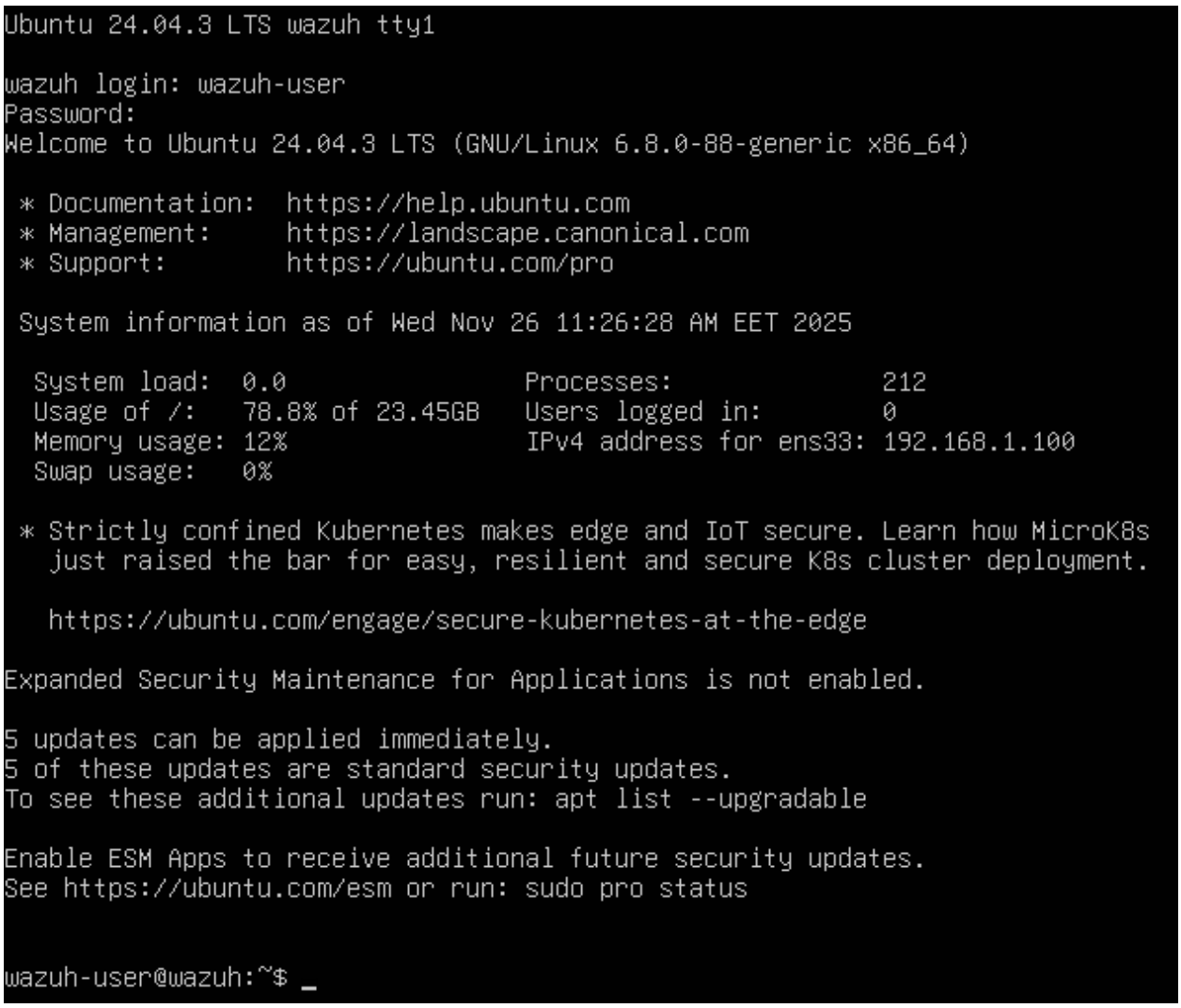


Рисунок 2.43 – Вікно командного рядка для входу з успішною авторизацією

2.3.3 Післяінсталяційне налаштування Ubuntu Server

Перед розгортанням компонентів Wazuh необхідно привести операційну систему до стандартизованого вигляду, що забезпечить стабільність та безпеку.

1. Оновлення пакетної бази: першим кроком є актуалізація списку репозиторіїв та оновлення встановленого програмного забезпечення для закриття відомих вразливостей. На рисунку 2.44 представлено команду, що була введена та процес оновлення.

```
wazuh-user@wazuh:~$ sudo apt update && sudo apt full-upgrade -y && sudo apt autoremove -y
Hit:1 http://ua.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ua.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://ua.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://ua.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1,619 kB]
Get:6 http://ua.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:7 http://ua.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:8 http://ua.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1,500 kB]
Get:9 http://ua.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [377 kB]
Get:10 http://ua.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:11 http://ua.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7,156 B]
Get:12 http://ua.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:13 http://ua.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [10.9 kB]
Get:14 http://ua.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:15 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.6 kB]
Get:16 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:17 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.2 kB]
Get:18 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Fetched 4,144 kB in 2s (1,759 kB/s)
```

Рисунок 2.44 – Оновлення пакетної бази

2. Налаштування статичної IP-адреси: для сервера безпеки критично важливо мати незмінну мережеву адресу. В Ubuntu 22.04 за мережу відповідає утиліта Netplan. Було відредаговано конфігураційний файл у директорії /etc/netplan/50-cloud-init.yaml (рисунок 2.45) та застосовано зміни з перевіркою мережевих налаштувань (рисунок 2.46) за допомогою відповідних команд.

```
 GNU nano 7.2                                  /etc/netplan/50-cloud-init.yaml *
network:
  version: 2
  ethernets:
    ens33:
      dhcp4: false
      addresses:
        - 192.168.1.100/24
      gateway4: 192.168.1.1
      nameservers:
        addresses:
          - 8.8.8.8
          - 1.1.1.1
```

Рисунок 2.45 – Відредагований конфігураційний файл мережевих налаштувань

```
wazuh-user@wazuh:~$ sudo nano /etc/netplan/50-cloud-init.yaml
wazuh-user@wazuh:~$ sudo netplan apply
wazuh-user@wazuh:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:d2:20:c4 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.1.100/24 brd 192.168.1.255 scope global ens33
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fed2:20c4/64 scope link
       valid_lft forever preferred_lft forever
wazuh-user@wazuh:~$ ip route
default via 192.168.1.1 dev ens33 proto static
192.168.1.0/24 dev ens33 proto kernel scope link src 192.168.1.100
wazuh-user@wazuh:~$
```

Рисунок 2.46 – Застосування змін та перевірка мережевих налаштувань

3. Синхронізація часу: для коректної кореляції подій безпеки час на сервері повинен бути ідеально точним. Налаштування часового поясу та перевірка статусу представлені на рисунку 2.47 з відповідним введенням команд.

```
wazuh-user@wazuh:~$ timedatectl
               Local time: Mon 2025-11-24 18:12:13 UTC
           Universal time: Mon 2025-11-24 18:12:13 UTC
                 RTC time: Mon 2025-11-24 18:12:13
                Time zone: Etc/UTC (UTC, +0000)
System clock synchronized: yes
              NTP service: active
          RTC in local TZ: no
wazuh-user@wazuh:~$ sudo timedatectl set-timezone Europe/Kyiv
[sudo] password for wazuh-user:
wazuh-user@wazuh:~$ sudo timedatectl set-timezone Europe/Kyiv
wazuh-user@wazuh:~$ timedatectl
               Local time: Mon 2025-11-24 20:13:46 EET
           Universal time: Mon 2025-11-24 18:13:46 UTC
                 RTC time: Mon 2025-11-24 18:13:46
                Time zone: Europe/Kyiv (EET, +0200)
System clock synchronized: yes
              NTP service: active
          RTC in local TZ: no
wazuh-user@wazuh:~$
```

Рисунок 2.47 – Налаштування часового поясу та перевірка статусу

4. Підключення по SSH: після налаштування мережі подальша робота з сервером виконується через SSH-клієнт, що дозволяє зручно копіювати команди та конфігураційні файли. Для цього необхідно в ESXi у налаштуваннях ВМ натиснути Connect to SSH. У вікні, що відкриється вводимо ім’я користувача та натискаємо Connect (рисунок 2.48).

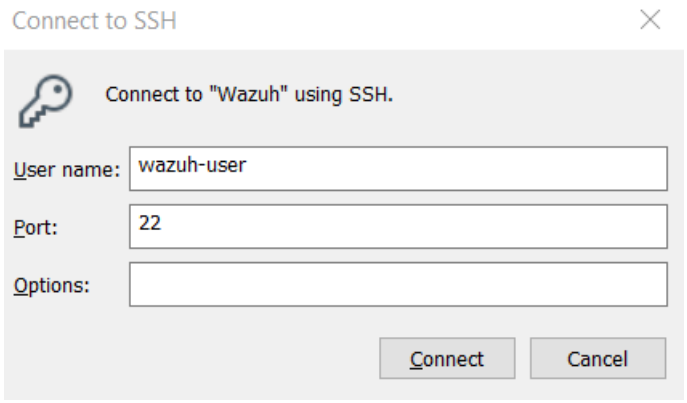


Рисунок 2.48 – Підключення через SSH-клієнт до серверу

Далі у терміналі Windows, що відкриється погоджуємося з продовженням з'єднання, прописуючи yes в командному рядку та вводимо пароль профіля користувача (рисунок 2.49). Після успішної авторизації з'являється екран привітання (див. рисунок 2.43).

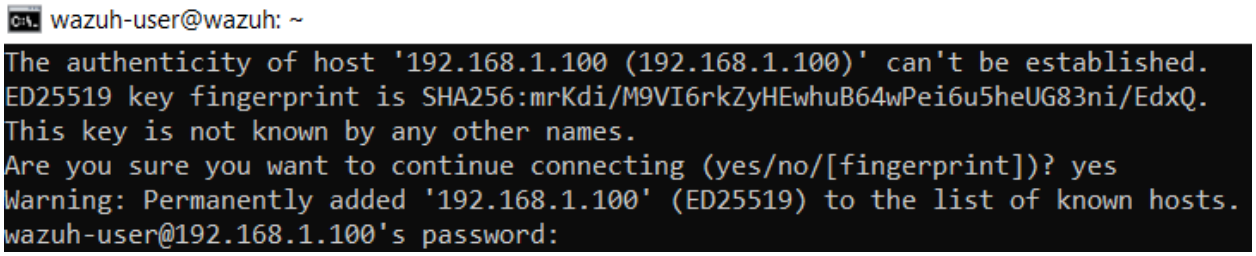


Рисунок 2.49 – Завершення підключення через SSH-клієнт до серверу у терміналі Windows

2.3.4 Покомпонентне розгортання платформи Wazuh

Для забезпечення глибокого контролю над конфігурацією та розуміння архітектурних зв'язків було обрано метод покомпонентної інсталяції і розгорнуто топологію All-in-one. Цей підхід передбачає встановлення всіх трьох ключових компонентів (Indexer, Server, Dashboard) на одному хості.

Розгортання виконувалося у послідовності, рекомендованій офіційною документацією розробника, з адаптацією конфігураційних файлів під специфіку розробленої віртуальної мережі.

Розгортання та конфігурація Wazuh Indexer

Розгортання компонента Wazuh Indexer передбачає виконання низки послідовних дій, спрямованих на встановлення пошукового рушія OpenSearch, генерацію сертифікатів та налаштування безпечної взаємодії між елементами системи Wazuh.

На першому етапі здійснюється завантаження службових скриптів Wazuh (wazuh-certs-tool.sh та config.yml), що забезпечують автоматизовану генерацію TLS-сертифікатів для всіх вузлів архітектури. Після корегування конфігураційного файлу відбувається формування кореневого сертифіката, ключів для Indexer, Dashboard, Filebeat та адміністративного доступу.

Наступним кроком є додавання офіційного репозиторію Wazuh, імпорт GPG-ключа підпису та встановлення пакунка wazuh-indexer. Після інсталяції компоненту виконується конфігурація OpenSearch, зокрема визначення IP-адреси вузла, назви кластера, початкових master-вузлів, шляхів до службових каталогів та підключення сертифікатів, згенерованих на попередньому етапі.

Після копіювання сертифікатів у службовий каталог /etc/wazuh-indexer/certs і встановлення необхідних атрибутів доступу відбувається активація служби Indexer через systemctl (рисунок 2.50).

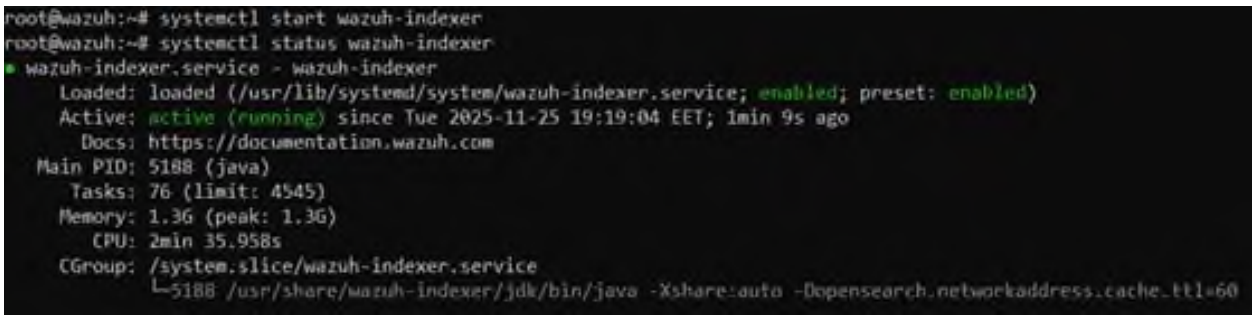


Рисунок 2.50 – Підтвердження активного статусу служби Wazuh Indexer

Додатково виконується запуск скрипта indexer-security-init.sh, який створює індекс безпеки OpenSearch та застосовує стандартні конфігурації доступу (ролі, мепінги, політики).

Фінальним етапом є перевірка працездатності кластера за допомогою HTTP-запитів до порту 9200. Отримання інформації про вузол та стан кластера

свідчить про коректне налаштування та готовність Indexer до інтеграції з іншими складовими Wazuh.

Усі команди, конфігураційні файли та службові параметри наведені у Додатку А.

Розгортання та конфігурація Wazuh Manager

На початковому етапі до системи додається GPG-ключ розробника та репозиторій Wazuh, після чого здійснюється оновлення списку пакунків і встановлення сервісу wazuh-manager разом із необхідними залежностями. Це створює базове середовище SIEM, здатне приймати та обробляти події від агентів.

Наступним кроком є налаштування компонента Filebeat, який виконує роль транспортного рівня між Wazuh Manager та Wazuh Indexer (OpenSearch). На сервері встановлюється filebeat, завантажується типовий конфігураційний файл для інтеграції з Wazuh, а також шаблон індексу та модуль wazuh. Для забезпечення безпечного обміну даними використовується TLS: на базі раніше згенерованих сертифікатів створюються окремі файли для Filebeat, налаштовуються шляхи до сертифікатів і ключів, а облікові дані для доступу до Indexer зберігаються у захищеному сховищі (keystore).

Ключовим елементом розгортання є конфігурація самого Wazuh Manager у файлі ossec.conf. У цьому файлі задаються глобальні параметри логування та оповіщень, політика моніторингу цілісності файлів (модуль Syscheck), перевірка системи на наявність руткітів і небажаних змін (Rootcheck), збір інвентаризаційних даних (Syscollector), а також функції виявлення вразливостей та перевірки конфігурації (Vulnerability detection, SCA). Окремий блок конфігурації відповідає за інтеграцію з Wazuh Indexer: визначаються адреса вузла, протокол HTTPS, а також сертифікати та ключі, спільні з Filebeat.

Після завершення налаштування конфігураційних файлів виконується активація служб Wazuh Manager (рисунок 2.51) та Filebeat (рисунок 2.52) через systemctl і перевірка їх працездатності.

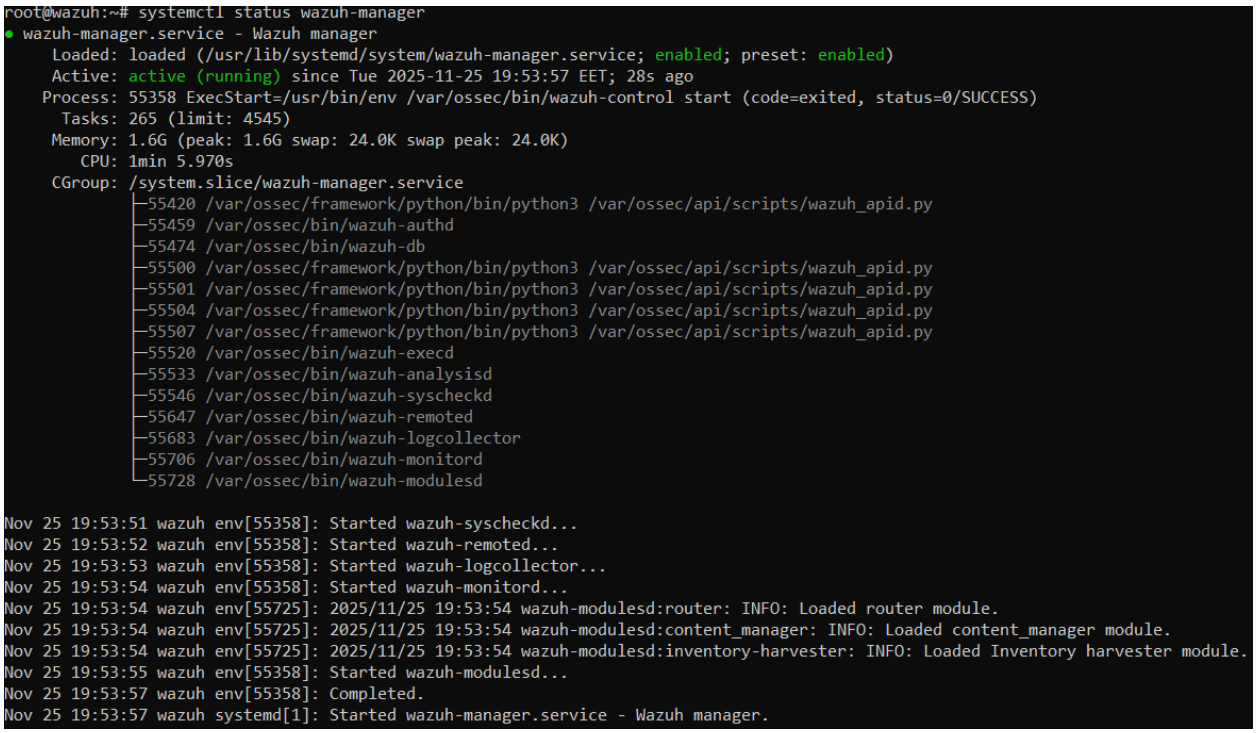


Рисунок 2.51 – Підтвердження активного статусу служби Wazuh Manager

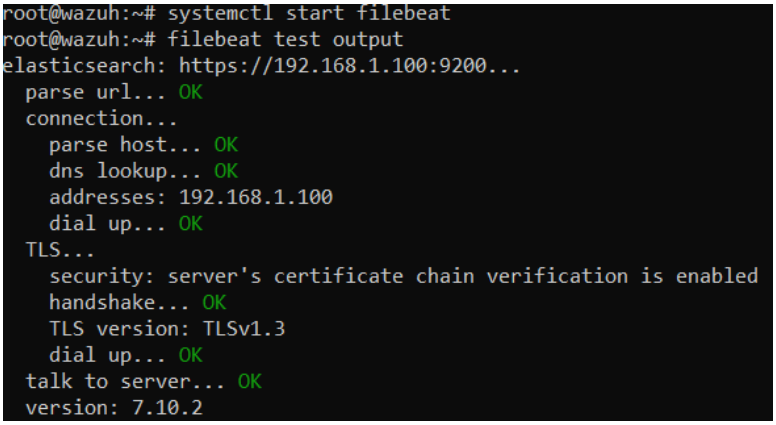


Рисунок 2.52 – Підтвердження активного статусу служби Filebeat

Запуск wazuh-manager забезпечує роботу основних служб (збір логів, кореляція подій, обробка правил, управління агентами), тоді як Filebeat відповідає за передачу згенерованих журналів безпеки до Indexer для подальшої індексації та візуалізації.

Деталізований лістинг команд розгортання, а також фрагменти конфігураційних файлів filebeat.yml і ossec.conf наведені у Додатку Б.

Розгортання та конфігурація Wazuh Dashboard

На першому етапі до системи імпортується GPG-ключ та додається офіційний репозиторій Wazuh, що забезпечує доступ до актуальних пакунків Dashboard. Після оновлення індексу пакетів виконується встановлення компонента wazuh-dashboard, у результаті чого створюється ізольоване середовище на базі Node.js з інтегрованими модулями взаємодії з OpenSearch.

Наступним кроком є налаштування захищеного доступу, який реалізується через TLS-сертифікати, згенеровані на попередніх етапах розгортання SIEM-системи. Сертифікати dashboard.pem, dashboard-key.pem та root-ca.pem додаються до службової директорії Dashboard, обмежуються у правах доступу та використовуються для шифрування з'єднань на порту 443. Це гарантує захищений доступ до веб-інтерфейсу та виключає можливість перехоплення даних.

У конфігураційному файлі opensearch_dashboards.yml визначаються ключові параметри роботи Dashboard, серед яких:

– IP-адреса та порт веб-сервера,
– адреса Wazuh Indexer для виконання запитів до OpenSearch,
– шлях до сертифікатів TLS,
– налаштування тривалості сесії користувача.

Після цього налаштовується файл wazuh.yml, який визначає спосіб підключення Dashboard до Wazuh Manager через його REST API на порту 55000. У цьому файлі задаються URL, порт та облікові дані користувача wazuh-wui, що мають доступ до API Manager.

Після застосування конфігурації сервіс Wazuh Dashboard активується через systemctl (рисунок 2.53), додається до автозавантаження та запускається як окрема служба.

```
root@wazuh:~# systemctl status wazuh-dashboard
● wazuh-dashboard.service - wazuh-dashboard
     Loaded: loaded (/etc/systemd/system/wazuh-dashboard.service; enabled; preset: enabled)
     Active: active (running) since Tue 2025-11-25 20:42:23 EET; 2min 3s ago
   Main PID: 64873 (node)
      Tasks: 11 (limit: 4545)
     Memory: 222.5M (peak: 302.4M)
        CPU: 11.146s
     CGroup: /system.slice/wazuh-dashboard.service
             └─64873 /usr/share/wazuh-dashboard/node/bin/node --no-warnings --max-http-header-size=65536 --ur

Nov 25 20:42:49 wazuh opensearch-dashboards[64873]: {"type":"log","@timestamp":"2025-11-25T18:42:49Z","tags":
Nov 25 20:42:49 wazuh opensearch-dashboards[64873]: {"type":"log","@timestamp":"2025-11-25T18:42:49Z","tags":
Nov 25 20:42:50 wazuh opensearch-dashboards[64873]: {"type":"log","@timestamp":"2025-11-25T18:42:50Z","tags":
Nov 25 20:42:50 wazuh opensearch-dashboards[64873]: {"type":"log","@timestamp":"2025-11-25T18:42:50Z","tags":
Nov 25 20:42:50 wazuh opensearch-dashboards[64873]: {"type":"log","@timestamp":"2025-11-25T18:42:50Z","tags":
Nov 25 20:42:50 wazuh opensearch-dashboards[64873]: {"type":"log","@timestamp":"2025-11-25T18:42:50Z","tags":
Nov 25 20:42:50 wazuh opensearch-dashboards[64873]: {"type":"log","@timestamp":"2025-11-25T18:42:50Z","tags":
Nov 25 20:42:50 wazuh opensearch-dashboards[64873]: {"type":"log","@timestamp":"2025-11-25T18:42:50Z","tags":
Nov 25 20:42:51 wazuh opensearch-dashboards[64873]: {"type":"log","@timestamp":"2025-11-25T18:42:51Z","tags":
Nov 25 20:42:51 wazuh opensearch-dashboards[64873]: {"type":"log","@timestamp":"2025-11-25T18:42:51Z","tags":
lines 1-20/20 (END)
```

Рисунок 2.53 – Підтвердження активного статусу служби Wazuh Dashboard

Усі команди встановлення, а також налаштовані конфігураційні файли наведено у Додатку В.

Для перевірки працездатності системи у веб-браузері було відкрито адресу https://192.168.1.100. Після введення логіну admin та згенерованого пароля (рисунок 2.54) відкривається головна панель Wazuh (див. рисунок 1.5). Система повідомляє про активний статус усіх компонентів та готовність до підключення агентів.

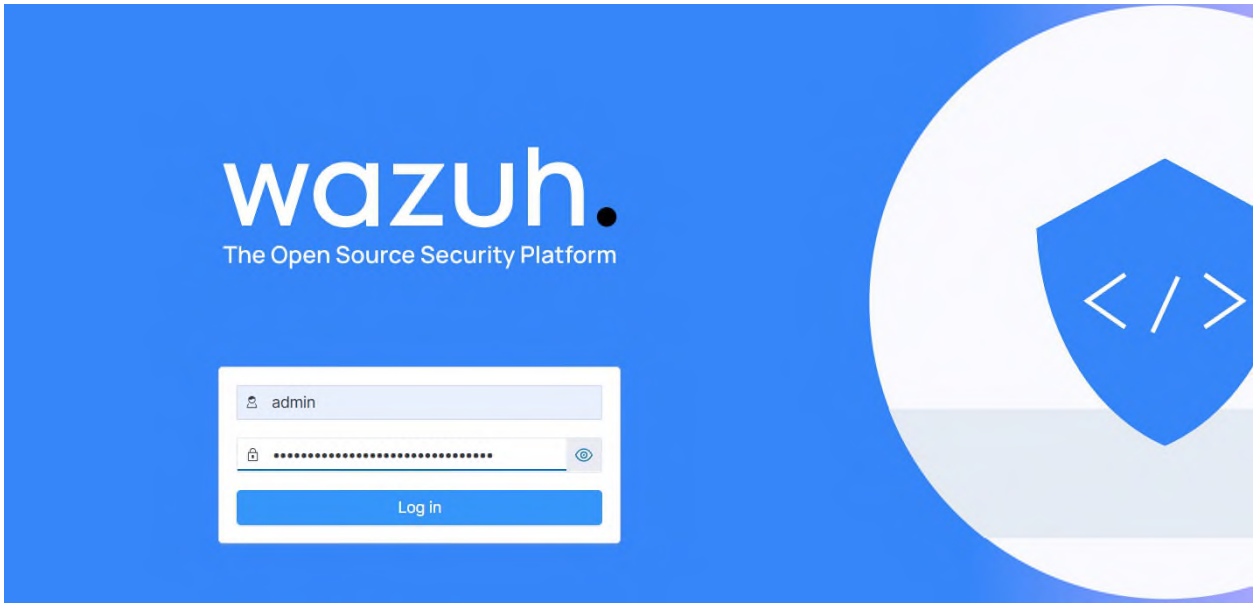


Рисунок 2.54 – Вікно введення облікових даних для авторизації у Wazuh

2.4 Встановлення та реєстрація агентів на Windows та Linux системах

Однією з переваг платформи Wazuh є наявність вбудованого майстра розгортання (Deploy new agent) у веб-інтерфейсі, який автоматизує процес створення інсталяційних команд для різних операційних систем. Це мінімізує ризик людської помилки при введенні конфігураційних даних.

Процес генерації сценарію виконується у декілька кроків через меню Agents => Deploy new agent (рисунок 2.55):

1. Вибір операційної системи: вказується тип ОС (Windows, Linux, macOS) та архітектура (x86_64, arm64).

2. Налаштування підключення: у полі Server Address вказується IP-адреса або доменне ім'я менеджера (у нашому випадку — 192.168.1.100).

3. Присвоєння імені (опціонально): задається унікальне ім'я агента для ідентифікації в системі.

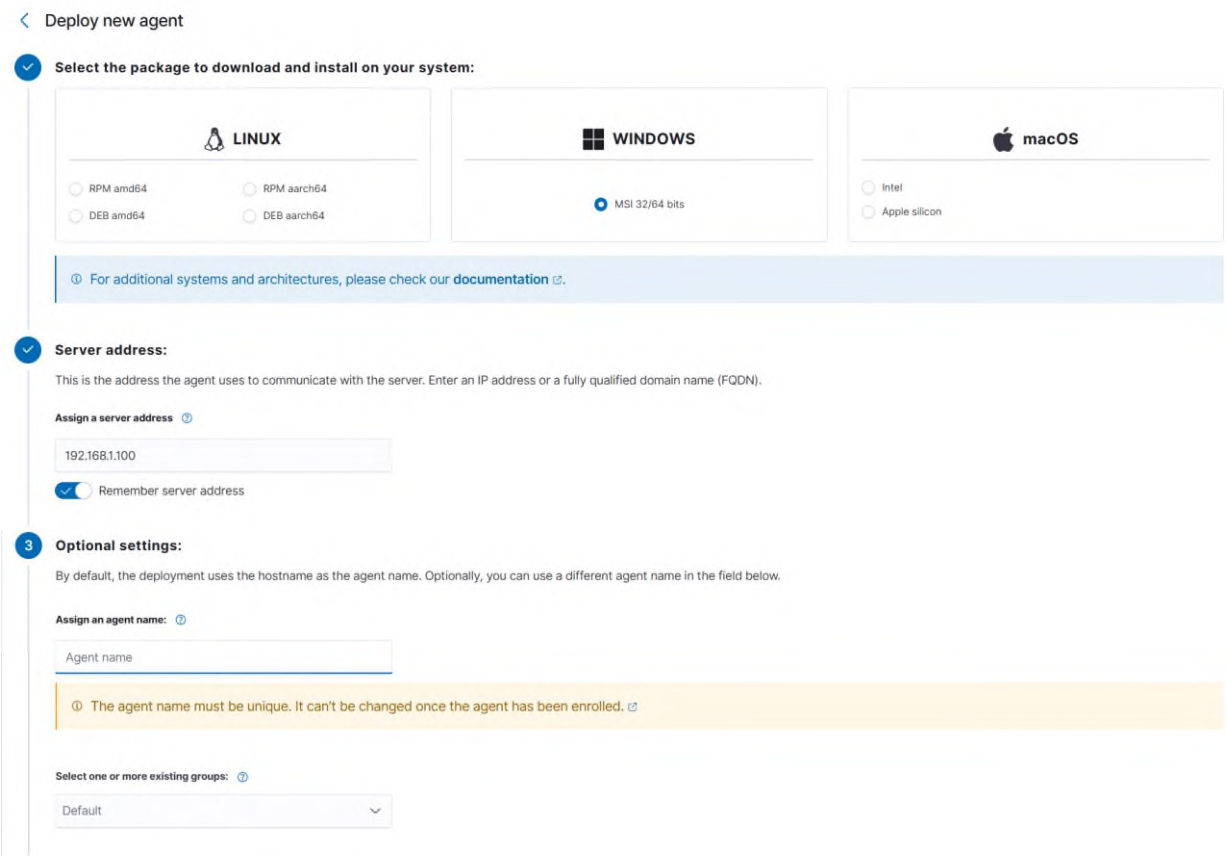


Рисунок 2.55 – Інтерфейс майстра розгортання агентів у Wazuh Dashboard

В результаті система генерує готовий рядок коду, який містить команду завантаження, інсталяції та реєстрації агента з необхідними ключами (рисунок 2.56). Цей підхід було використано як базовий для подальшого розгортання.

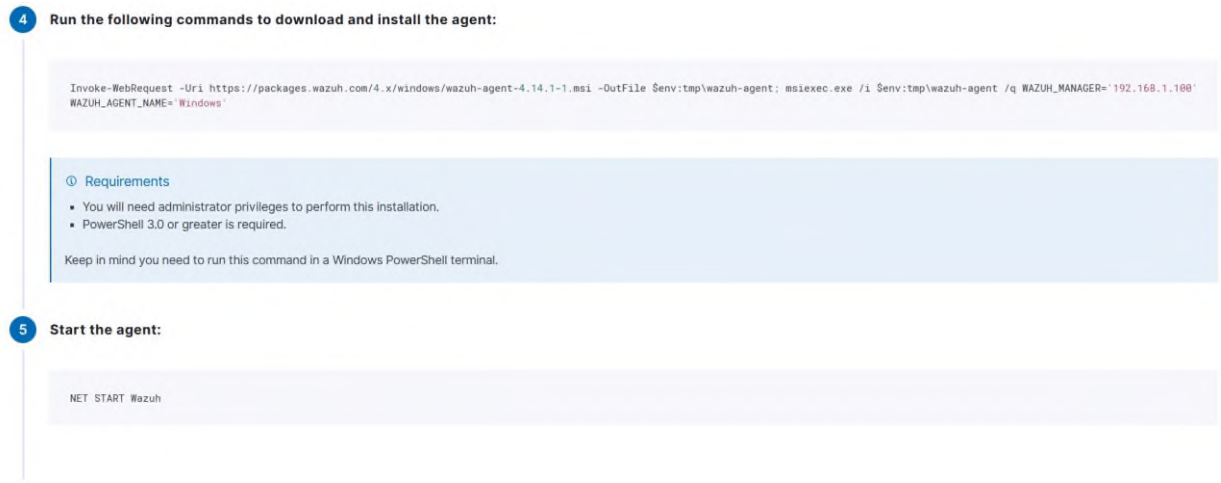


Рисунок 2.56 – Згенерований рядок коду для завантаження, інсталяції та реєстрації агента у ОС Windows

Процес розгортання агента у ОС Windows було виконано через інтерфейс командного рядка PowerShell з правами адміністратора (рисунок 2.57), що дозволило автоматизувати передачу параметрів конфігурації. Запуск служби агента виконано командою NET START Wazuh. Для підтвердження успішного запуску та реєстрації використано команду Get-Service WazuhSvc.

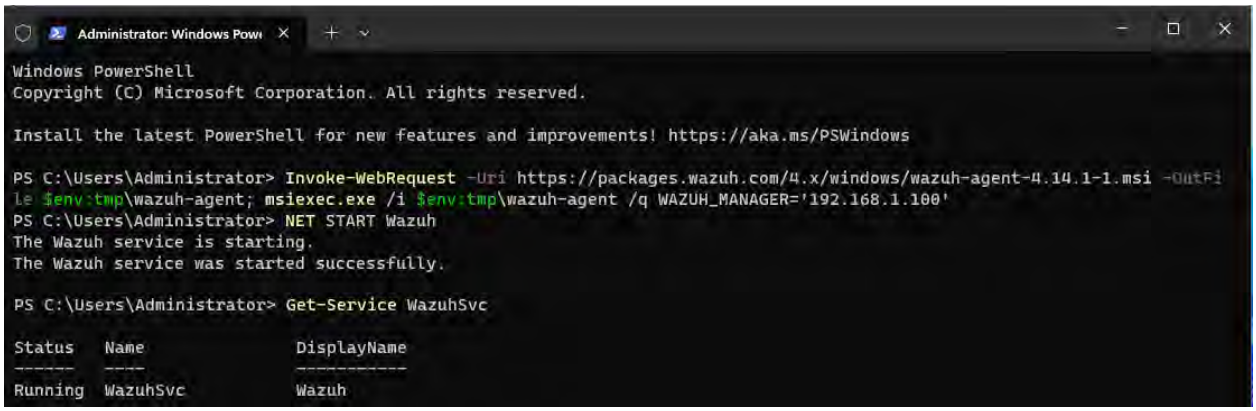


Рисунок 2.57 – Розгортання агента у ОС Windows через інтерфейс командного рядка PowerShell з правами адміністратора

Як видно з рисунку 2.57, служба має статус Running. Це свідчить про те, що агент успішно з'єднався з менеджером, отримав унікальний ключ шифрування та розпочав передачу подій на сервер.

Для ОС на базі Linux (Ubuntu 22.04) було згенеровано сценарій, який також був отриманий через інтерфейс майстра розгортання агентів у Wazuh Dashboard.

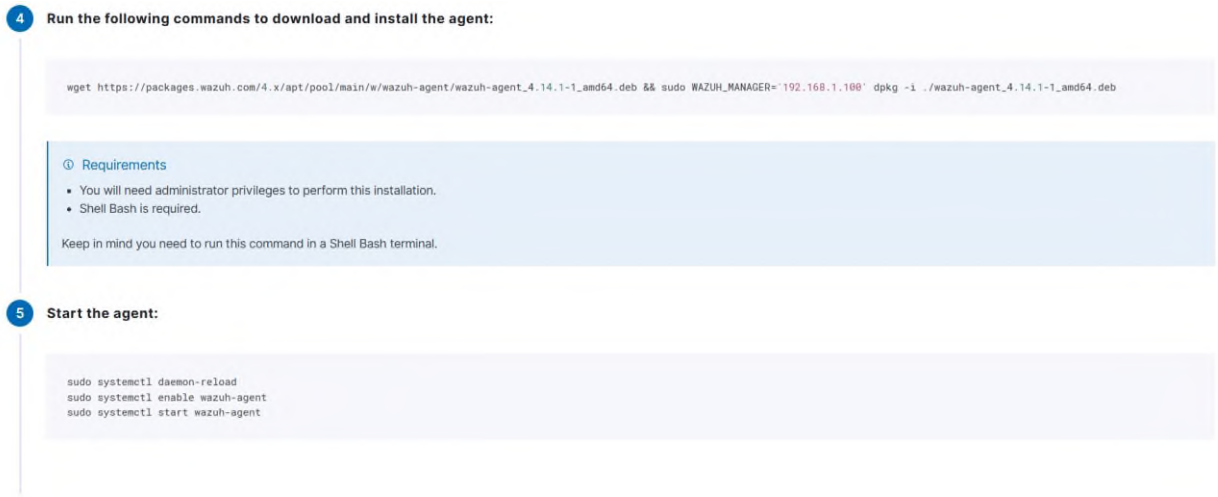


Рисунок 2.58 – Згенерований рядок коду для завантаження, інсталяції та реєстрації агента у ОС Linux

Процес розгортання агента у ОС Linux виконувався у терміналі з правами суперкористувача root (рисунок 2.59). Згенерована попередньо команда автоматично додає репозиторій, завантажує пакет та редагує файл конфігурації /var/ossec/etc/ossec.conf, прописуючи адресу менеджера.

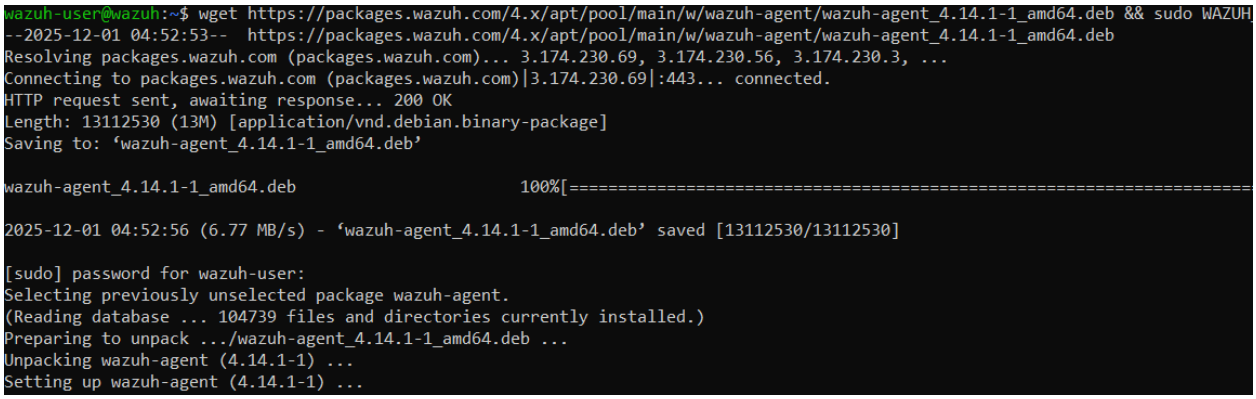


Рисунок 2.59 – Розгортання агента у ОС Linux через термінал з правами суперкористувача

Після виконання інсталяції службу було активовано та запущено. Контроль працездатності виконано шляхом перевірки статусу системного юніта. Відсутність помилок у виводі команди підтверджує стабільну роботу процесу збору логів (рисунок 2.60).

```
wazuh-user@wazuh:~$ sudo systemctl daemon-reload
wazuh-user@wazuh:~$ sudo systemctl enable wazuh-agent
Created symlink /etc/systemd/system/multi-user.target.wants/wazuh-agent.service → /usr/lib/systemd/system/wazuh-agent.service.
wazuh-user@wazuh:~$ sudo systemctl start wazuh-agent
wazuh-user@wazuh:~$ sudo systemctl status wazuh-agent
● wazuh-agent.service - Wazuh agent
     Loaded: loaded (/usr/lib/systemd/system/wazuh-agent.service; enabled; preset: enabled)
     Active: active (running) since Mon 2025-12-01 04:54:17 EET; 13s ago
    Process: 2958 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=exited, status=0/SUCCESS)
      Tasks: 29 (limit: 2207)
     Memory: 36.8M (peak: 42.4M)
        CPU: 3.429s
     CGroup: /system.slice/wazuh-agent.service
             ├─2982 /var/ossec/bin/wazuh-execd
             ├─2991 /var/ossec/bin/wazuh-agentd
             ├─3004 /var/ossec/bin/wazuh-syscheckd
             ├─3017 /var/ossec/bin/wazuh-logcollector
             └─3034 /var/ossec/bin/wazuh-modulesd
```

Рисунок 2.60 – Статус служби Wazuh Agent у середовищі Linux

Для остаточного підтвердження того, що обидва агенти (Windows та Linux) коректно інтегровані в систему безпеки, було виконано перевірку списку активних агентів через веб-інтерфейс Wazuh Dashboard (рисунок 2.61).

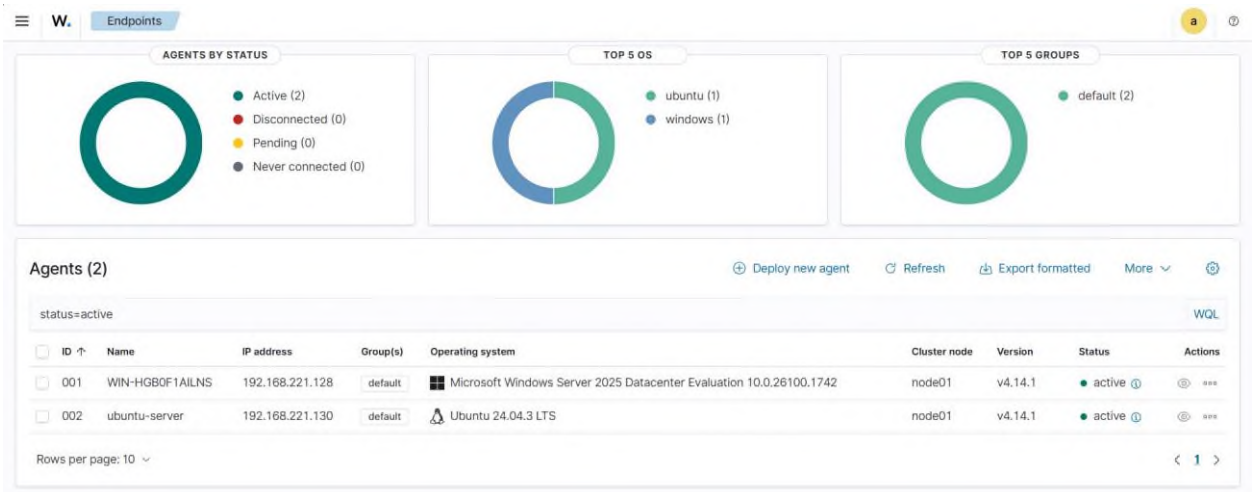


Рисунок 2.61 – Список активних агентів, зареєстрованих на сервері Wazuh

На рисинку 2.61 відображено, що агенти з ID 001 (Windows) та 002 (Linux) мають статус Active. Це завершує етап побудови інфраструктури моніторингу. Система готова до проведення експериментальної частини з моделювання кіберінцидентів.

Висновки до розділу 2

У другому розділі магістерської дисертації було виконано практичну реалізацію автоматизованої системи моніторингу загроз на базі SIEM-платформи Wazuh. В ході виконання роботи було досягнуто наступних результатів:

1. Досліджено архітектуру та принципи функціонування SIEM-системи. Проведено детальний аналіз компонентної бази платформи Wazuh (Indexer, Server, Dashboard, Agents). Визначено, що модульна структура та використання шифрованих каналів зв'язку забезпечують високий рівень масштабованості та захищеності. Обґрунтовано вибір топології розгортання, яка відповідає вимогам до сучасних систем виявлення вторгнень та забезпечує цілісність даних на всіх етапах обробки.

2. Розгорнуто віртуалізовану інфраструктуру. В якості основи для лабораторного стенду було обрано та інстальовано гіпервізор першого типу VMware ESXi. Це дозволило забезпечити ізоляцію обчислювальних ресурсів та наблизити умови експерименту до реального промислового.

3. Реалізовано серверну частину системи безпеки. На базі операційної системи Ubuntu Server розгорнуто центральні компоненти платформи у конфігурації All-in-one. Забезпечено захист каналів управління та передачі даних шляхом генерації та впровадження SSL-сертифікатів, що гарантує конфіденційність зібраної інформації.

4. Побудовано мережу моніторингу. Здійснено успішну інтеграцію різнорідних кінцевих точок у єдиний контур безпеки. Реалізовано встановлення та реєстрацію агентів Wazuh для операційних систем Windows та Linux із застосуванням методів автоматизації (PowerShell, Bash), що дозволило централізувати збір журналів подій з усієї інфраструктури.

РОЗДІЛ 3. ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ АВТОМАТИЗОВАНОЇ СИСТЕМИ МОНІТОРИНГУ ЗАГРОЗ

3.1 Моделювання сценаріїв невдалих авторизацій

Одним із найбільш поширених векторів атак на інформаційні системи є спроби несанкціонованого доступу через підбір паролів (Brute-force) або використання компрометованих облікових даних. Згідно з матрицею MITRE ATT&CK, ця техніка класифікується як T1110 (Brute Force).

Метою даного етапу дослідження є перевірка здатності системи Wazuh детектувати спроби невдалої автентифікації на різнорідних операційних системах (Windows та Linux) за допомогою стандартного набору правил

3.1.1 Сценарій атаки на Windows-агент

Для моделювання інциденту було використано робочу станцію під управлінням Windows 10, на якій попередньо встановлено та зареєстровано агент Wazuh.

Хід експерименту:

1. Здійснено спробу локального входу в систему з використанням існуючого облікового запису користувача, але з введенням завідомо неправильного пароля (рисунок 3.1).

2. Дії повторювалися з інтервалом у 10-15 секунд, щоб імітувати "ручні" спроби підбору.

На рівні операційної системи ці дії генерують подію з кодом Event ID 4625 ("An account failed to log on"). Агент Wazuh зчитує цю подію в реальному часі та передає її на сервер для аналізу і застосовує власне правило для класифікації цієї події з кодом Event ID 60122 ("Logon Failure - Unknown user or bad password").

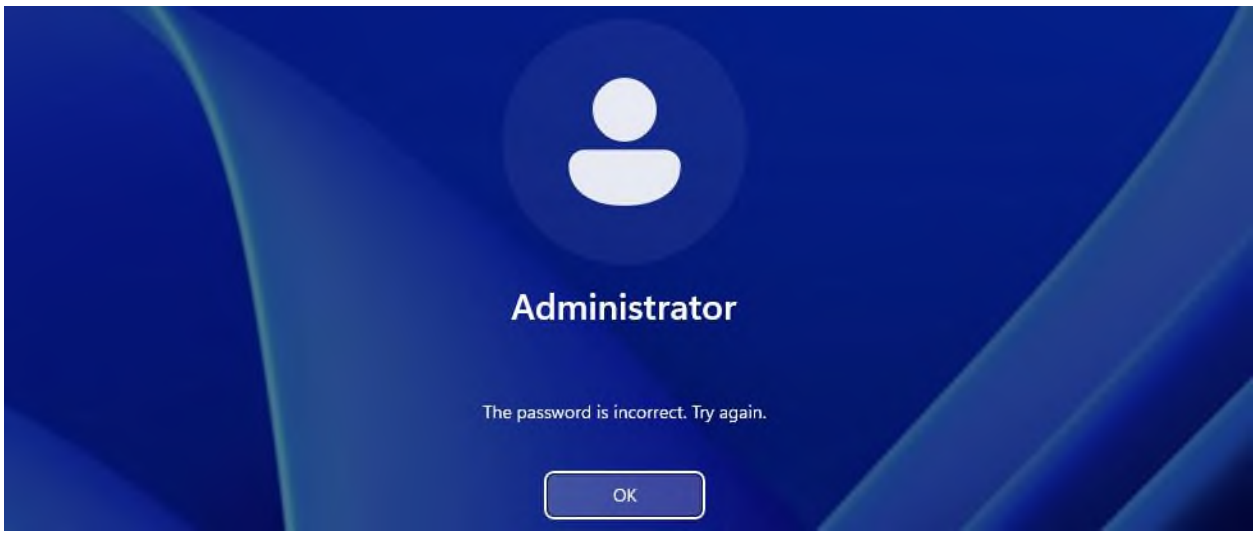


Рисунок 3.1 – Імітація невдалої спроби входу в середовищі Windows

3.1.2 Сценарій атаки на Linux -агент

Аналогічний експеримент було проведено на сервері під управлінням Ubuntu 22.04 LTS. Оскільки Linux-сервери найчастіше адмініструються віддалено, моделювалася спроба підключення через протокол SSH.

Хід експерименту:

1. Ініційовано SSH-з'єднання з хостової машини: ssh administrator@192.168.221.130.

2. Введено неодноразово неправильний пароль для існуючого користувача (рисунок 3.2.).

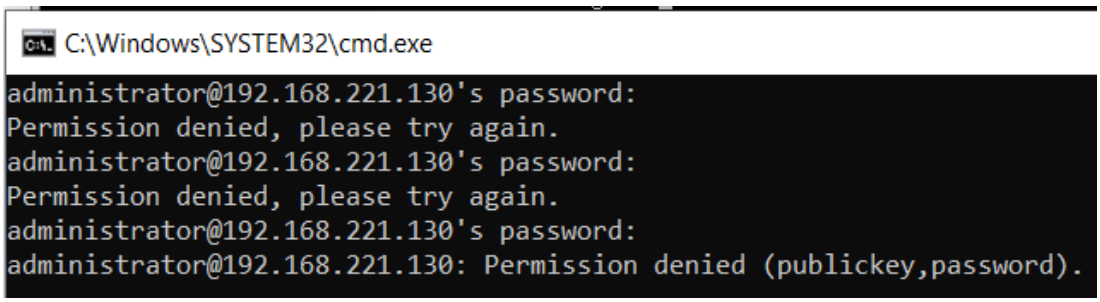


Рисунок 3.2 – Імітація невдалої спроби входу в середовищі Linux

Ці дії фіксуються локальним демоном sshd та записуються у файл /var/log/auth.log. Агент Wazuh моніторить цей файл і при появі записів типу "Failed password" ініціює створення алерту.

3.1.3 Аналіз результатів у Wazuh Dashboard

Одразу після виконання маніпуляцій у веб-інтерфейсі Wazuh Dashboard розділу Threat Hunting було зафіксовано сплеск активності (рисунок 3.3).

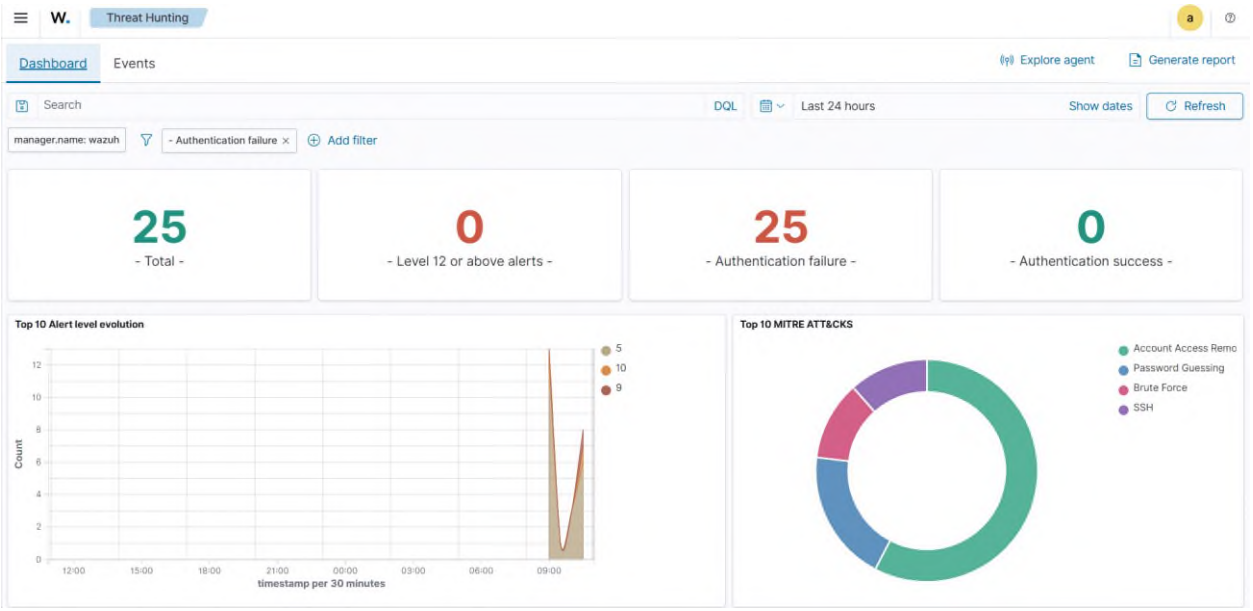
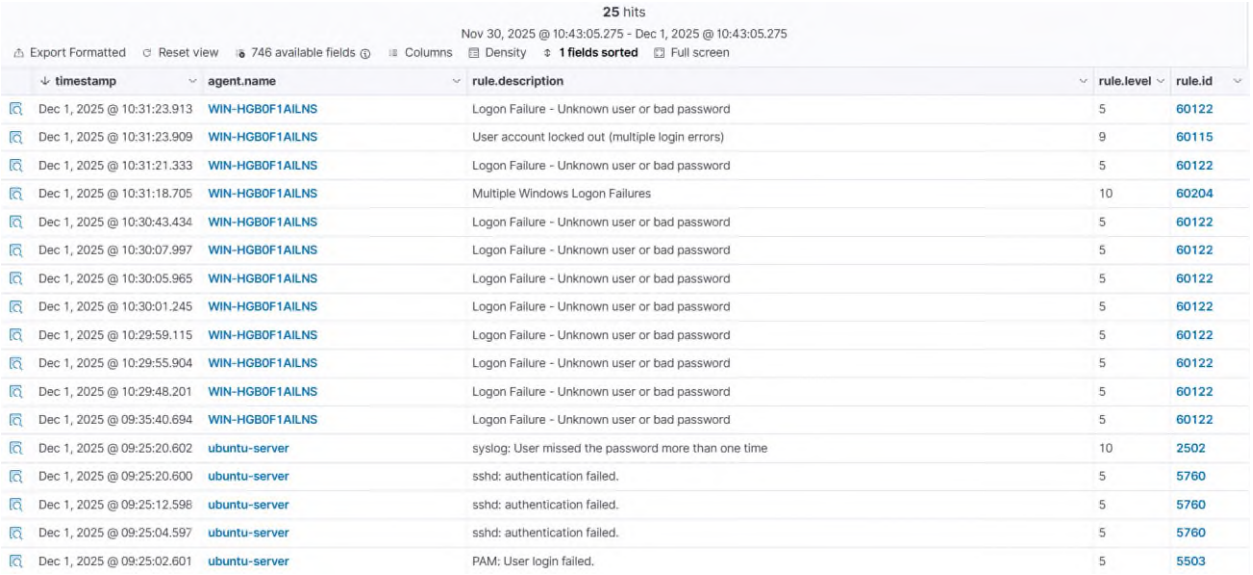


Рисунок 3.3 – Відображення подій невдалої авторизації в Wazuh Dashboard

Для перегляду деталей інциденту було використано фільтри Events, що дозволило виокремити події за рівнем критичності та ідентифікатором правила (рисунок 3.4).



25 hits

Nov 30, 2025 @ 10:43:05.275 - Dec 1, 2025 @ 10:43:05.275

Export Formatted | Reset view | 746 available fields | Columns | Density | 1 fields sorted | Full screen

timestamp	agent.name	rule.description	rule.level	rule.id
Dec 1, 2025 @ 10:31:23.913	WIN-HGB0F1AILNS	Logon Failure - Unknown user or bad password	5	60122
Dec 1, 2025 @ 10:31:23.909	WIN-HGB0F1AILNS	User account locked out (multiple login errors)	9	60115
Dec 1, 2025 @ 10:31:21.333	WIN-HGB0F1AILNS	Logon Failure - Unknown user or bad password	5	60122
Dec 1, 2025 @ 10:31:18.705	WIN-HGB0F1AILNS	Multiple Windows Logon Failures	10	60204
Dec 1, 2025 @ 10:30:43.434	WIN-HGB0F1AILNS	Logon Failure - Unknown user or bad password	5	60122
Dec 1, 2025 @ 10:30:07.997	WIN-HGB0F1AILNS	Logon Failure - Unknown user or bad password	5	60122
Dec 1, 2025 @ 10:30:05.965	WIN-HGB0F1AILNS	Logon Failure - Unknown user or bad password	5	60122
Dec 1, 2025 @ 10:30:01.245	WIN-HGB0F1AILNS	Logon Failure - Unknown user or bad password	5	60122
Dec 1, 2025 @ 10:29:59.115	WIN-HGB0F1AILNS	Logon Failure - Unknown user or bad password	5	60122
Dec 1, 2025 @ 10:29:55.904	WIN-HGB0F1AILNS	Logon Failure - Unknown user or bad password	5	60122
Dec 1, 2025 @ 10:29:48.201	WIN-HGB0F1AILNS	Logon Failure - Unknown user or bad password	5	60122
Dec 1, 2025 @ 09:35:40.694	WIN-HGB0F1AILNS	Logon Failure - Unknown user or bad password	5	60122
Dec 1, 2025 @ 09:25:20.602	ubuntu-server	syslog: User missed the password more than one time	10	2502
Dec 1, 2025 @ 09:25:20.600	ubuntu-server	sshd: authentication failed.	5	5760
Dec 1, 2025 @ 09:25:12.598	ubuntu-server	sshd: authentication failed.	5	5760
Dec 1, 2025 @ 09:25:04.597	ubuntu-server	sshd: authentication failed.	5	5760
Dec 1, 2025 @ 09:25:02.601	ubuntu-server	PAM: User login failed.	5	5503

Рисунок 3.4 – Відображення подій невдалої авторизації в Wazuh Dashboard

Як видно з рисунку 3.2, система успішно ідентифікувала спроби входу на обох платформах. Кожна подія отримала відповідний рівень небезпеки, що привертає увагу адміністратора.

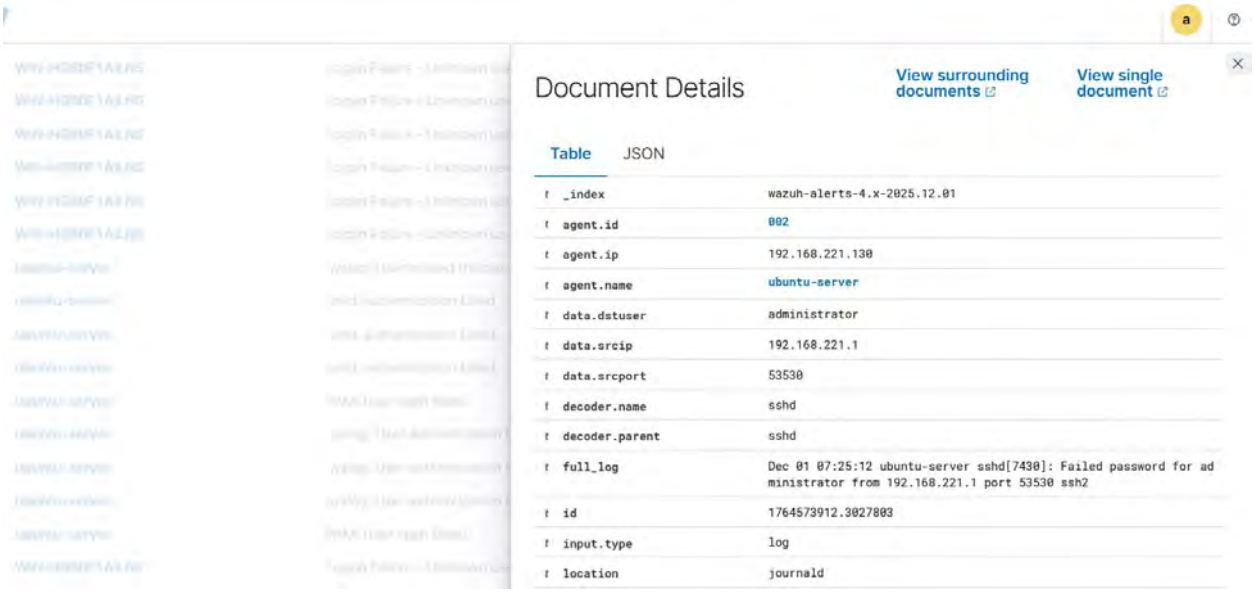
3.2 Аналіз спрацьовування правил та журналів подій

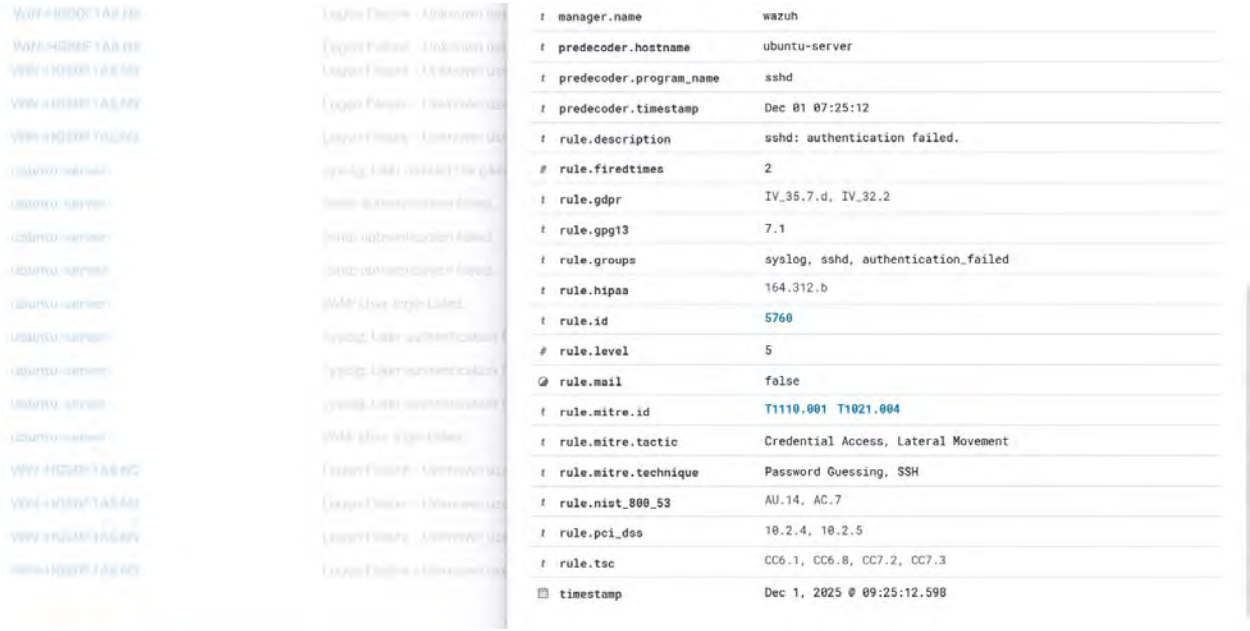
Ефективність SIEM-системи визначається не лише фактом збору інформації, а й здатністю інтерпретувати події, виявляючи приховані загрози серед великого масиву даних. У цьому підрозділі проведено детальний аналіз механізмів обробки інцидентів у системі Wazuh на прикладі сценаріїв, змодельованих у п. 3.1. Дослідження фокусується на двох ключових процесах: нормалізації "сирих" логів та кореляції взаємопов’язаних подій.

3.2.1 Нормалізація даних та аналіз одиничних подій

Першим етапом обробки даних у Wazuh є нормалізація. Це процес, під час якого декодери розбирають неструктуровані системні повідомлення та перетворюють їх у єдиний формат JSON. Це дозволяє уніфікувати аналіз подій, отриманих з різних джерел (Linux, Windows, мережеве обладнання).

Розглянемо структуру події, зафіксованої під час спроби невдалої авторизації на сервері Ubuntu (див. п. 3.1.2). На рисунку 3.5 наведено реальний лог алерту, сформованого системою.





manager.name	wazuh
predecoder.hostname	ubuntu-server
predecoder.program_name	sshd
predecoder.timestamp	Dec 01 07:25:12
rule.description	sshd: authentication failed.
rule.firedtimes	2
rule.gdpr	IV_35.7.d, IV_32.2
rule.gpg13	7.1
rule.groups	syslog, sshd, authentication_failed
rule.hipaa	164.312.b
rule.id	5760
rule.level	5
rule.mail	false
rule.mitre.id	T1110.001 T1021.004
rule.mitre.tactic	Credential Access, Lateral Movement
rule.mitre.technique	Password Guessing, SSH
rule.nist_800_53	AU.14, AC.7
rule.pci_dss	10.2.4, 10.2.5
rule.tsc	CC6.1, CC6.8, CC7.2, CC7.3
timestamp	Dec 1, 2025 @ 09:25:12.598

Рисунок 3.5 – Деталізована картка критичного інциденту невдалої авторизації SSH

Аналіз ключових полів логу:

1. Ідентифікація та класифікація (rule):

– id: 5760: система ідентифікувала подію за стандартним правилом для SSH-сервісу ("Authentication failed").

– mitre: автоматично визначено техніки атаки згідно з матрицею MITRE ATT&CK — T1110.001 (Password Guessing) та T1021.004 (SSH). Це вказує на спробу отримання доступу через підбір пароля та можливе горизонтальне переміщення зловмисника мережею.

– Відповідність стандартам: подія автоматично співставлена з вимогами регуляторів — PCI DSS (вимоги 10.2.4, 10.2.5 щодо моніторингу доступу), GDPR (захист персональних даних) та NIST 800-53.

2. Контекстні дані (data):

– srcip: 192.168.221.1 — декодер успішно виокремив IP-адресу джерела атаки.

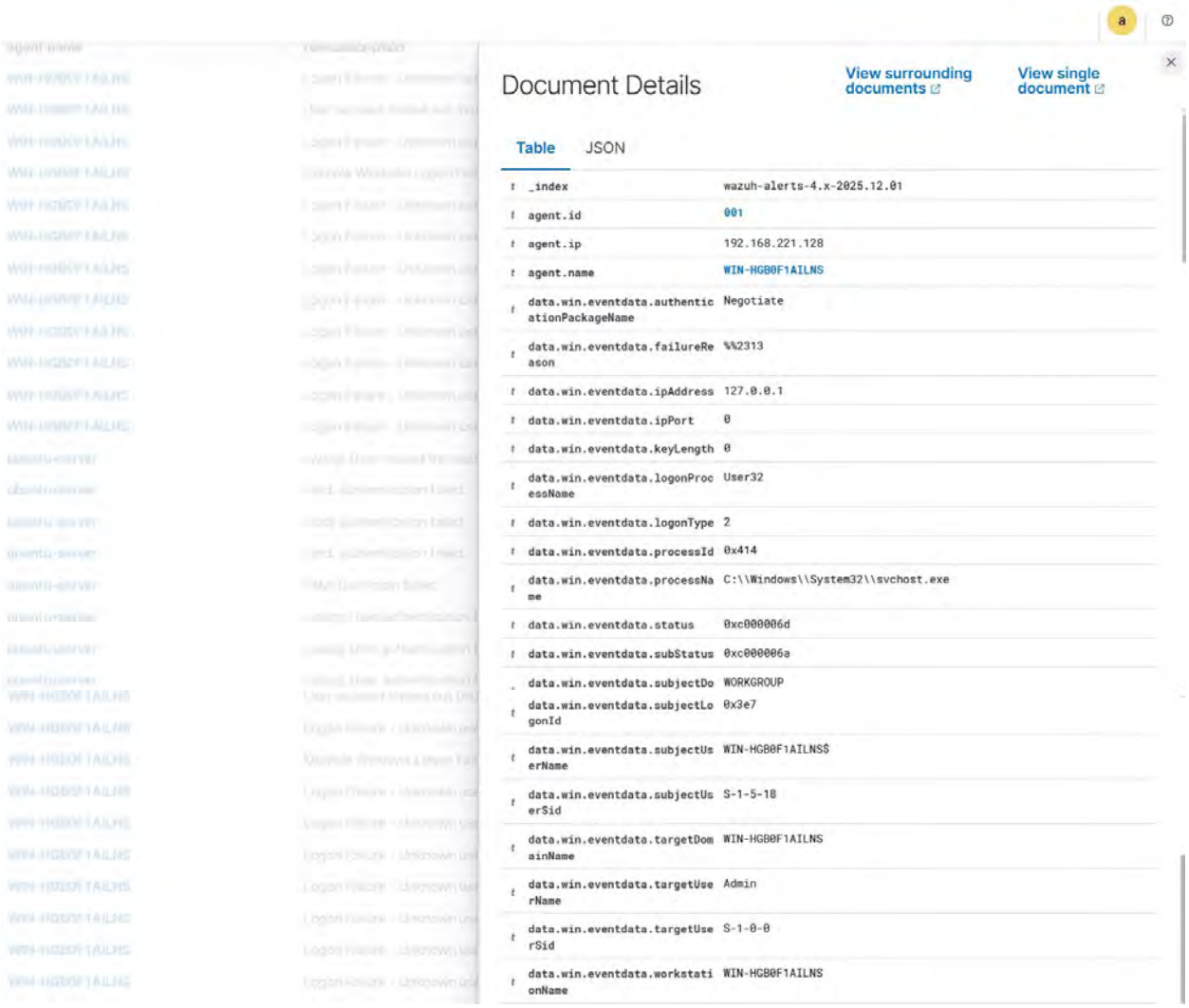
– dstuser: administrator — зафіксовано ім'я користувача, який намагався використати атакуючий.

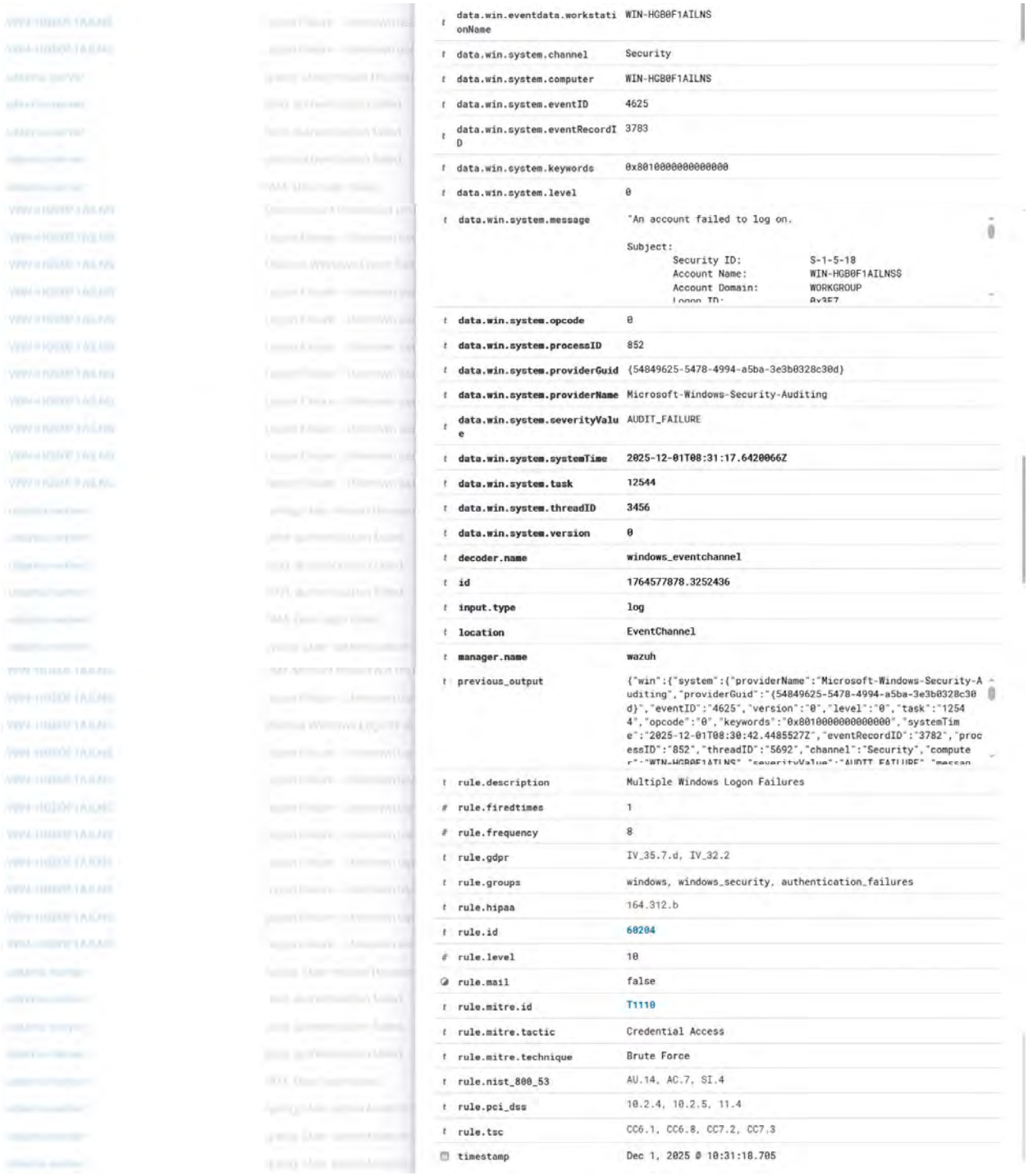
Цей приклад демонструє, як Wazuh збагачує "сирий" технічний лог аналітичною інформацією, необхідною для розслідування інцидентів та проведення аудиту.

3.2.2 Механізм кореляції подій

Якщо одинична помилка може бути наслідком людського фактору, то серія таких помилок є ознакою цілеспрямованої атаки. Для виявлення подібних патернів Wazuh використовує механізм кореляції з параметрами частоти (frequency) та часового вікна (timeframe).

Для дослідження цього механізму було проаналізовано критичний інцидент, що виник внаслідок серії невдалих спроб входу на Windows-агенті (див. п. 3.1.1). На рисунку 3.6 наведено реальний лог алерту високого рівня небезпеки, сформованого системою.





data.win.eventdata.workstationName	WIN-HGB0F1AILNS
data.win.system.channel	Security
data.win.system.computer	WIN-HGB0F1AILNS
data.win.system.eventID	4625
data.win.system.eventRecordID	3783
data.win.system.keywords	0x8010000000000000
data.win.system.level	0
data.win.system.message	"An account failed to log on. Subject: Security ID: S-1-5-18 Account Name: WIN-HGB0F1AILNS$ Account Domain: WORKGROUP
data.win.system.opcode	0
data.win.system.processID	852
data.win.system.providerGuid	{54849625-5478-4994-a5ba-3e3b0328c30d}
data.win.system.providerName	Microsoft-Windows-Security-Auditing
data.win.system.severityValue	AUDIT_FAILURE
data.win.system.systemTime	2025-12-01T08:31:17.6420066Z
data.win.system.task	12544
data.win.system.threadID	3456
data.win.system.version	0
decoder.name	windows_eventchannel
id	1764577878.3252436
input.type	log
location	EventChannel
manager.name	wazuh
previous_output	{"win":{"system":{"providerName":"Microsoft-Windows-Security-Auditing","providerGuid":"{54849625-5478-4994-a5ba-3e3b0328c30d}","eventID":"4625","version":"0","level":"0","task":"12544","opcode":"0","keywords":"0x8010000000000000","systemTime":"2025-12-01T08:30:42.4485527Z","eventRecordID":"3782","processID":"852","threadID":"5692","channel":"Security","compute
rule.description	Multiple Windows Logon Failures
rule.firedtimes	1
rule.frequency	8
rule.gdpr	IV_35.7.d, IV_32.2
rule.groups	windows, windows_security, authentication_failures
rule.hipaa	164.312.b
rule.id	60204
rule.level	10
rule.mail	false
rule.mitre.id	T1110
rule.mitre.tactic	Credential Access
rule.mitre.technique	Brute Force
rule.nist_800_53	AU.14, AC.7, SI.4
rule.pci_dss	10.2.4, 10.2.5, 11.4
rule.tsc	CC6.1, CC6.8, CC7.2, CC7.3
timestamp	Dec 1, 2025 @ 10:31:18.705

Рисунок 3.6 – Деталізована картка критичного інциденту "Multiple Windows Logon Failures"

Аналіз механізму виявлення атаки:

1. Агрегація подій (rule.frequency: 8): ключовим полем у даному алерті є параметр частоти. Правило ID 60204 (Level 10) активувалося лише після накопичення 8 подій-тригерів (поодиноких помилок EventID 4625) за

короткий проміжок часу. Це дозволяє системі фільтрувати випадкові помилки користувачів і реагувати лише на аномальну активність.

2. Доказова база (previous_output): лог містить поле previous_output, в якому зберігається повна історія попередніх 8 подій, що призвели до спрацювання правила. Це критично важливо для цифрової криміналістики, оскільки дозволяє відновити точну хронологію атаки.

3. Аналіз вектору атаки:

– LogonType: 2 вказує на тип входу "Interactive". Це свідчить про те, що спроби підбору пароля здійснювалися локально (через фізичну консоль або консоль гіпервізора), а не через мережевий протокол SMB/RDP.

– targetUserName: Admin: атака була спрямована на привілейований обліковий запис.

4. Ескалація рівня загрози — система автоматично підвищила рівень критичності інциденту до Level 10 (High), що вимагає негайної реакції адміністратора. Подія також класифікована як T1110 (Brute Force) згідно з MITRE ATT&CK.

Проведений аналіз підтверджує, що розгорнута система Wazuh успішно реалізує функції SIEM. Завдяки механізмам нормалізації, система приводить різнорідні дані (Syslog Linux, Windows Event Channel) до єдиного формату, а механізм кореляції дозволяє автоматично виявляти складні атаки, базуючись на частотному аналізі подій, та надавати вичерпну інформацію для реагування.

3.3 Оцінювання ефективності системи Wazuh щодо виявлення загроз

На основі проведених у попередніх підрозділах експериментів з моделювання кіберінцидентів, було проведено комплексне оцінювання ефективності розгорнутої SIEM-системи Wazuh.

Оцінка здійснювалася за чотирма ключовими критеріями, які є визначальними для систем класу EDR/XDR: швидкість детектування, точність

класифікації, повнота контекстних даних та відповідність регуляторним вимогам.

3.3.1 Аналіз швидкості та точності реакції

В ході експерименту з атакою Brute Force (п. 3.2) система продемонструвала здатність працювати в режимі реального часу. Час між генерацією події на кінцевій точці (Windows/Linux) та появою алерту на інформаційній панелі склав менше 1 секунди. Це підтверджує, що архітектура на базі агентів забезпечує мінімальні затримки передачі даних, що критично важливо для оперативного реагування.

Показник точності був підтверджений коректною роботою механізму кореляції. Система успішно розрізнила два типи активності:

– Поодинокі помилки (Level 5): інтерпретовані, як потенційні помилки користувачів.

– Серійні помилки (Level 10): інтерпретовані, як цілеспрямована атака.

Така багаторівнева логіка дозволяє мінімізувати кількість хибних спрацювань, не перевантажуючи адміністратора зайвими сповіщеннями про рутинні події, але гарантовано сповіщаючи про критичні інциденти.

3.3.2 Оцінка інформативності та контекстуалізації

Ефективність системи моніторингу залежить не лише від факту виявлення загрози, а й від якості наданої інформації для її розслідування. Проведений аналіз JSON-логів показав, що Wazuh автоматично збагачує події метаданими, які значно спрощують роботу аналітика SOC:

– Атрибуція атаки: чітка фіксація джерела (srcip), об'єкта атаки (dstuser) та вектору (logonType).

– Класифікація MITRE ATT&CK: автоматичне зіставлення інциденту з тактиками T1110 (Brute Force) та T1021 (Lateral Movement) дозволяє зрозуміти стратегію зловмисника.

– Хронологічний контекст: наявність поля previous_output у корельованих алертах дозволяє відновити повну хронологію атаки без необхідності ручного пошуку в архівах логів.

3.3.3 Оцінка інформативності та контекстуалізації

Результати тестування дозволяють стверджувати, що розгорнута конфігурація Wazuh забезпечує високий рівень захищеності інформаційної інфраструктури. Зведена оцінка ефективності системи за результатами дослідження наведена в таблиці 3.1.

Таблиця 3.1 – Оцінка ефективності SIEM-системи Wazuh

Критерій оцінювання	**Результат**	**Характеристика**
Оперативність	Висока	Обробка та візуалізація подій відбувається в режимі near real-time
Масштабованість	Висока	Агентська архітектура дозволяє підключати різнорідні ОС (Windows, Linux, MacOS) без зміни логіки сервера
Відповідність	Висока	Автоматичний мапінг подій на вимоги PCI DSS, GDPR, NIST спрощує проходження аудитів
Аналітичні можливості	Високі	Вбудовані правила кореляції ефективно виявляють відомі патерни атак, проте для специфічних загроз необхідне написання кастомних правил

Отримані результати свідчать про те, що впровадження системи Wazuh дозволяє автоматизувати процес виявлення інцидентів інформаційної безпеки. Система довела свою ефективність у детектуванні атак на етапі спроби отримання доступу, надаючи вичерпну доказову базу для проведення розслідувань. Функціональні можливості платформи повністю відповідають вимогам до сучасних систем моніторингу загроз у КІС.

3.4 Перспективи розвитку та можливості інтеграції Wazuh у інші середовища

Досліджена у роботі система моніторингу на базі Wazuh має архітектурну гнучкість, необхідну для захисту кіберфізичних систем. В умовах переходу до концепції Індустрії 4.0, де межі між інформаційними та операційними технологіями стираються, впроваджене рішення відкриває значні перспективи для захисту автоматизованих виробничих комплексів.

3.4.1 Глибока інспекція промислових протоколів

Специфіка комп'ютерно-інтегрованих систем полягає у використанні спеціалізованих протоколів обміну даними. Перспективним напрямком розвитку системи є інтеграція Wazuh з модулями глибокого аналізу пакетів (DPI) для моніторингу трафіку в мережах Industrial Ethernet.

Це дозволяє реалізувати контроль на рівні технологічного процесу:

– Аналіз протоколів: виявлення аномалій у стандартах Modbus TCP, PROFINET, EtherCAT та OPC UA.

– Валідація команд: система здатна відрізняти легітимні команди зчитування даних від несанкціонованих спроб запису в регістри ПЛК, що можуть призвести до зупинки конвеєра або аварії.

– Моніторинг HMI-панелей: встановлення агентів на операторські станції дозволяє фіксувати нетипові дії персоналу або спроби перехоплення керування інтерфейсом.

3.4.2 Захист робототехнічних комплексів

Сучасна робототехніка базується на відкритих стандартах, таких як ROS (Robot Operating System), які часто не мають вбудованих механізмів шифрування та автентифікації. Інтеграція Wazuh у контур керування промисловими маніпуляторами та мобільними роботами (AMR) забезпечить:

– Цілісність прошивок: використання модуля FIM для контролю незмінності виконуваного коду та конфігураційних файлів робота. Це унеможливлює атаки типу "підміна логіки руху", які можуть призвести до фізичного пошкодження обладнання.

– Моніторинг флоту (Fleet Management): централізований збір логів безпеки з десятків мобільних роботів, що дозволяє виявляти скомпрометовані вузли, які намагаються надсилати хибні навігаційні дані іншим учасникам рою.

3.4.3 Моніторинг сегменту промислового Інтернету речей

У розгалужених системах автоматизації використовується велика кількість датчиків та виконавчих механізмів з обмеженими обчислювальними ресурсами. Для таких пристроїв Wazuh пропонує режим "Agentless monitoring" (безагентний моніторинг).

Перспективи використання:

– Контроль периферії: моніторинг шлюзів IoT, які агрегують дані з датчиків перед відправкою в хмару або SCADA.

– Виявлення аномалій телеметрії: аналіз потоків даних (наприклад, через протокол MQTT) для виявлення підміни значень датчиків, що є критичним для систем автоматичного регулювання.

3.4.4 Забезпечення безперервності технологічних процесів

На відміну від корпоративних мереж, де пріоритетом є конфіденційність, у системах автоматизації головним є доступність та безпека персоналу. Адаптація правил реагування (Active Response) під вимоги OT дозволяє:

– Ізоляцію замість блокування: при виявленні загрози система може не блокувати контролер (що небезпечно для процесу), а логічно ізолювати його від зовнішньої мережі, залишаючи локальне керування.

– Інтеграцію з системами ПАЗ: взаємодія з системами протиаварійного захисту для автоматичного переведення технологічного процесу в безпечний стан при виявленні критичної кібератаки.

Розгорнута SIEM-система є універсальною платформою, придатною для захисту критичної інфраструктури. Її можливості виходять за межі класичної IT-безпеки, дозволяючи реалізувати комплексний захист рівнів управління, контролю та виконання у сучасних автоматизованих та робототехнічних системах.

Висновки до розділу 3

У третьому розділі магістерської дисертації проведено експериментальне дослідження ефективності розгорнутої системи моніторингу загроз у середовищі, наближеному до реальних умов експлуатації комп'ютерно-інтегрованих систем. За результатами практичної частини зроблено наступні висновки:

1. Змодельовано сценарії кібератак на гетерогенну інфраструктуру. Відтворено спроби несанкціонованого доступу до серверів управління на базі ОС Windows Linux, а також змодельовано атаку підбору паролів для перевірки реакції системи на інтенсивні запити автентифікації.

2. Підтверджено ефективність алгоритмів кореляції подій. Аналіз журналів продемонстрував здатність системи автоматично виявляти складні патерни атак. Механізм частотного аналізу (дозволив успішно відфільтрувати поодинокі помилки входу та ідентифікувати цілеспрямовану атаку, автоматично ескалувавши рівень загрози до критичного (Level 10) із збереженням історії інциденту.

3. Проведено оцінювання ефективності системи. На основі аналізу швидкості та точності детектування встановлено, що розгорнуте рішення забезпечує обробку подій у режимі реального часу. Система продемонструвала високу інформативність, автоматично збагачуючи алерти

контекстними даними (мапінг на базу знань MITRE ATT&CK та вимоги стандартів PCI DSS, GDPR, NIST), що значно скорочує час на розслідування.

4. Визначено стратегію інтеграції в промислові системи. Обґрунтовано доцільність використання платформи Wazuh для захисту кіберфізичних систем та робототехнічних комплексів. Проаналізовано можливості масштабування архітектури для моніторингу промислових протоколів, захисту SCADA-систем та забезпечення безперервності технологічних процесів.

РОЗДІЛ 4. РОЗРОБКА СТАРТАП-ПРОЄКТУ "SECUREVISION"

4.1 Опис та технологічний аудит ідеї стартап-проєкту

Стартап-проєкт "SecureVision" спрямований на створення комплексної автоматизованої системи для моніторингу подій безпеки у гетерогенних мережах, що поєднує функціональність SIEM Wazuh та можливості віртуалізаційної інфраструктури VMware ESXi. Продукт орієнтований на організації, що потребують бюджетного, але професійного SOC (Security Operations Center), здатного інтегрувати дані як з класичних IT-мереж, так і з промислових систем.

Таблиця 4.1 подає базовий опис ідеї стартапу, напрями застосування платформи та ключові вигоди для різних груп користувачів. Це дозволяє сформувати початкове бачення продукту та визначити його цінність.

Таблиця 4.1 – Опис ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Створення автоматизованої платформи моніторингу та реагування на кіберзагрози на базі SIEM Wazuh та VMware ESXi	Корпоративні IT-інфраструктури	Зменшення ризиків, централізація управління безпекою
	Державні установи	Відповідність стандартам (ISO 27001, NIST), підвищення контролю
	Промислові об’єкти та IoT	Безперервний моніторинг, мінімізація впливу атак
	Хмарні провайдери та дата-центри	Масштабованість, оптимізація витрат

Подана характеристика підтверджує, що продукт має широкий спектр застосування та здатний задовольнити потреби різних сегментів ринку. Зокрема, інноваційність рішення полягає в поєднанні автоматизації, масштабованості та простоти впровадження. Ідея стартапу відзначається

високою комерційною перспективністю та можливістю швидкої адаптації до змін ринку.

Для визначення конкурентоспроможності проєкту необхідно провести порівняльний аналіз його техніко-економічних властивостей з існуючими на ринку аналогами. Аналіз сфокусовано на трьох групах конкурентів: дорогі Enterprise-рішення (Splunk), спеціалізовані системи для OT та "самозбірні" Open-Source інсталяції. Порівняння представлено у таблиці 4.2.

Таблиця 4.2 – Порівняльна характеристика з аналогами

Критерій	**Проєкт "SecureVision"**	**Enterprise SIEM (Splunk, QRadar)**	**Спеціалізовані OT Security (Nozomi, Claroty)**
Основна мета продукту	Універсальний моніторинг IT/OT середовищ для широкого ринку	Глибока аналітика великих даних для головних офісів корпорацій	Глибокий аналіз промислових протоколів
Ціновий сегмент	Низький/Середній (оплата за впровадження)	Преміум (висока вартість ліцензій/підписки)	Преміум (висока вартість апаратних сенсорів)
Архітектура	Віртуалізована (VMware ESXi + Ubuntu)	Хмарна (SaaS) або важке On-Premise	Апаратно-програмні комплекси (Network Taps)
Складність впровадження	Низька (імпорт OVA-образу)	Висока (потребує сертифікованих інженерів)	Середня (потребує зміни мережевої топології)
Ключова відмінність (УТП)	Гнучкість Linux та Wazuh, упакована у зручний формат для ESXi	Потужна екосистема, AI-аналітика, глобальна підтримка	Розуміння специфіки контролерів "з коробки"

Аналіз показує, що проєкт "SecureVision" займає вільну нішу між дорогими комерційними продуктами та складними у налаштуванні "голими" Open-Source інструментами. Головною перевагою є баланс між вартістю володіння та готовністю до роботи в середовищі віртуалізації. Продукт виграє за рахунок універсальності та можливості моніторингу всієї супутньої інфраструктури, не вимагаючи при цьому щорічних ліцензійних платежів. Це є вирішальним фактором під час бюджетних скорочень у державному секторі та промисловості.

Наступним кроком є визначення сильних, слабких та нейтральних сторін проєкту в порівнянні з конкурентами (Enterprise та DIY-рішення). Це дозволить сформувати унікальну торговельну пропозицію (УТП) та зрозуміти, на яких характеристиках варто робити акцент у маркетинговій стратегії. Результати аудиту характеристик наведено в табл. 4.3.

Таблиця 4.3 – Визначення сильних, слабких та нейтральних характеристик ідеї проєкту

№ п/п	Техніко-економічна характеристика	Мій проєкт	Конкурент 1 (Splunk)	Конкурент 2 (самостійна збірка)	W	N	S
1.	Вартість ліцензування ПЗ	Безкоштовно	Дуже висока	Безкоштовно	–	–	+
2.	Швидкість розгортання	Висока (Ready Image)	Середня	Низька	–	–	+
3.	Рівень технічної підтримки	Обмежена (Community)	Повна (24/7 Vendor)	Відсутня	+	–	–
4.	Функціонал "з коробки"	Базовий +	Розширений	Базовий	–	+	–
5.	Вимоги до "заліза"	Середні (Ubuntu)	Високі	Варіативні	–	+	–
6.	Адаптація під віртуалізацію	Оптимізовано (ESXi)	Низька (універсал)	Низька	–	–	+

Проведений аналіз демонструє, що проєкт має чітко виражені сильні сторони: відсутність ліцензійних платежів та оптимізацію під віртуальне середовище VMware. Це є конкурентною перевагою для державних тендерів та промислових замовників, які прагнуть мінімізувати операційні витрати (OPEX). Слабка сторона — відсутність вендорської підтримки 24/7, яка може бути компенсована наданням власних послуг з супроводу, що перетворить недолік на джерело додаткового доходу.

Для підтвердження можливості реалізації проєкту необхідно провести технологічний аудит обраних інструментів. Важливо переконатися, що всі складові рішення (Wazuh, Ubuntu, VMware) сумісні між собою та доступні для використання без юридичних або технічних обмежень. У таблиці 4.4 наведено аналіз технологічної здійсненності.

Таблиця 4.4 – Технологічна здійсненність ідеї проєкту

№ з/п	Ідея проєкту	Технології реалізації	Наявність технологій	Доступність технологій
1.	Середовище віртуалізації	Гіпервізор **VMware ESXi**	Наявна. Індустріальний стандарт віртуалізації	Доступна (Free/Trial версії, корпоративні ліцензії).
2.	Операційна система	**Ubuntu Server LTS**	Наявна. Стабільна та безпечна ОС	Доступна (Open Source, безкоштовна)
3.	Система моніторингу (SIEM)	**Wazuh** (Manager, Indexer, Dashboard)	Наявна. Потужний Fork OSSEC + Elastic Stack	Доступна (ліцензія GPLv2, відкритий код)
4.	Збір даних та реагування	Агенти Wazuh, скрипти Active Response	Наявна. Розроблено в рамках магістерської роботи	Доступна. Не потребує додаткового обладнання

Технологічна реалізація проєкту є повністю здійсненною, оскільки базується на зрілих та перевірених технологіях з відкритим кодом. Всі ключові компоненти вже інтегровані та протестовані в рамках практичної частини магістерської дисертації. Гібридне використання Ubuntu та VMware створює надійну основу для комерційного продукту, що не вимагає винаходу нових технологій, а лише їх правильної конфігурації під потреби ринку.

4.2 Аналіз ринкових можливостей запуску стартап-проєкту

Аналіз ринкових можливостей є ключовим етапом оцінки перспективності стартапу, оскільки саме ринкові умови визначають успіх впровадження інноваційних рішень. Проведемо комплексний аналіз, спрямований на вивчення попиту, пропозиції, цільових клієнтів, а також можливих загроз і переваг.

Метою цього аналізу є визначення специфічних факторів, що можуть впливати на успішність проєкту. Результати аналізу дозволять сформувати рекомендації для ефективного ринкового впровадження та забезпечення конкурентних переваг.

У таблиці 4.5 наведено ключові характеристики потенційного ринку для системи "SecureVision".

Таблиця 4.5 – Попередня характеристика потенційного ринку стартап-проєкту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	6
2	Загальний обсяг ринку SIEM-рішень, млрд. доларів	6,5
3	Динаміка ринку (якісна оцінка)	Стрімко зростає
4	Наявність обмежень для входу	Висока конкуренція та необхідність сертифікації
5	Специфічні вимоги до стандартизації та сертифікації	Поширені стандарти ISO 27001, NIST, GDPR
6	Середня норма рентабельності в галузі (або ринку), %	18-25

Ринок SIEM-рішень демонструє значне та стабільне зростання, що підтверджує перспективність виходу нового продукту. Незважаючи на наявність потужних конкурентів, стрімке збільшення кіберзагроз створює підвищений попит на нові інструменти моніторингу. Високий рівень стандартизації підвищує вимоги до продукту, але водночас створює можливість для його диференціації. Рентабельність ринку робить інвестиції привабливими.

Для визначення ринкового потенціалу проєкту необхідно детально проаналізувати групи споживачів, які мають потребу в системах моніторингу загроз. Оскільки проєкт орієнтований на нішу доступних та безпечних рішень (On-Premise), ми виділяємо три ключові сегменти: державний сектор, промисловий бізнес та IT-посередників. У таблиці 4.6 наведено детальну характеристику цих груп, їхні поведінкові особливості та специфічні вимоги до продукту "SecureVision".

Таблиця 4.6 – Характеристика потенційних клієнтів стартап-проєкту

№ п/п	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних потенційних цільових груп	Вимоги споживачів до товару
1	Необхідність виконання законодавчих вимог (КСЗІ) та імпортозаміщення ПЗ	**Державні установи та об'єкти критичної інфраструктури (G2B)**	Цикл купівлі довгий, регламентований процедурами тендерів. Висока чутливість до наявності документації та сертифікатів. Рішення приймається колегіально.	**Суверенітет даних:** повна ізоляція (On-Premise), відсутність хмарних компонентів. **Ціна:** нульова вартість ліцензій. **Локалізація:** підтримка української мови у звітах.
2	Потреба у захисті безперервних технологічних процесів від кібератак та простоїв	**Середні промислові підприємства та агрохолдинги (B2B)**	Прагматична поведінка. Вимагають пілотного тестування (PoC) перед покупкою. Бояться, що активне сканування зупинить виробництво.	**Стабільність:** пасивний моніторинг без навантаження на мережу. **Сумісність:** підтримка застарілих ОС (Legacy), що керують верстатами. **Надійність:** відмовостійкість на рівні ESXi.
3	Потреба в інструменті для надання платних послуг кібербезпеки малим клієнтам	**MSP-провайдери та IT-аутсорсери (B2B2C)**	Швидкий цикл прийняття рішень. Шукають інструменти, що дозволяють обслуговувати багато клієнтів малим штатом інженерів. Фокус на маржинальності.	**Ізоляція даних клієнтів:** можливість безпечно розділяти доступ для різних замовників в одній системі. **Швидкість:** розгортання нового клієнта за хвилини (через OVA). **Економія:** низька собівартість обслуговування.

Аналіз цільової аудиторії показує, що хоча всі групи потребують моніторингу загроз, їхні мотиви кардинально відрізняються: держсектор керується вимогами закону, промисловість — страхом збитків, а MSP — бажанням заробити. Це вимагає від стартапу гнучкості: для першої групи

необхідно готувати пакет документів для тендерів, для другої — демонструвати стабільність, а для третьої — зручність масштабування.

Наступним кроком є аналіз загроз і можливостей, що дозволить розробити стратегії для зменшення ризиків та максимального використання ринкових переваг.

Аналіз зовнішнього середовища дозволяє виявити фактори, що можуть негативно вплинути на успіх проєкту. Для стартапу в сфері кібербезпеки критичними є загрози, пов'язані з сумісністю, кадрами та конкуренцією з хмарами. У таблиці 4.7 визначено 5 ключових факторів загроз.

Таблиця 4.7 – Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Складність інтеграції з Legacy-системами	Старі державні системи або ПЛК можуть не підтримувати сучасні агенти	Розробка спеціалізованих Linux-шлюзів для збору логів без агентів
2	Бюрократія в держзакупівлях	Довгі процедури тендерів та високі вимоги до документації	Партнерство з досвідченими інтеграторами, що вже мають акредитацію.
3	"Втома від сповіщень" (Alert Fatigue)	Персонал клієнта ігноруватиме систему через хибні спрацювання	Попереднє тонке налаштування правил кореляції під типові профілі клієнтів.
4	Конкуренція з хмарними SIEM	Microsoft/Google пропонують легший старт, але зберігають дані у хмарі	Акцент на безпеці даних: "Ваші логи залишаються у вашому периметрі", що критично для держсектору
5	Шифрування трафіку	Неможливість аналізу зашифрованих пакетів	Використання агентів на кінцевих точках для перехоплення подій до шифрування

Ідентифіковані загрози вимагають проактивної стратегії, особливо в роботі з державним сектором. Загроза бюрократії нівелюється через партнерську мережу, а конкуренція з хмарами перетворюється на перевагу завдяки архітектурі On-Premise. Основний акцент слід робити на адаптації продукту під застарілі системи, які все ще масово використовуються в Україні.

Поряд із загрозами ринок пропонує значні можливості, зумовлені глобальними трендами та локальною ситуацією. Використання цих

можливостей дозволить проєкту швидше масштабуватися та закріпитися у ніші. У таблиці 4.8 наведено 5 факторів можливостей.

Таблиця 4.8 – Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Державні програми кіберзахисту	Фінансування захисту критичної інфраструктури з бюджету та грантів	Участь у тендерах Prozorro з пропозицією "найкраща ціна"
2	Тренд на імпортозаміщення	Відмова від російського та дорогого західного ПЗ	Позиціонування як українського продукту на базі відкритого коду
3	Industry 4.0 та цифровізація	Підключення заводів до мережі створює нові вектори атак	Розробка модулів для моніторингу IoT/OT протоколів через Wazuh
4	Популярність віртуалізації	Масовий перехід на VMware ESXi в держсекторі та бізнесі	Постачання продукту виключно як готового OVA-аплаєнсу
5	Оптимізація бюджетів	Криза змушує шукати дешевші аналоги Splunk/QRadar	Пропозиція функціонального аналога за 10-20% вартості конкурентів

Найбільш перспективною можливістю є державні програми кіберзахисту та тренд на імпортозаміщення. "SecureVision" ідеально вписується в цю нішу, пропонуючи легальне, безпечне та доступне рішення. Використання грантових коштів може стати драйвером для пілотних впроваджень у промисловості та держсекторі.

Аналіз конкурентного середовища є необхідним етапом для вибору правильної ринкової ніші та стратегії поведінки. Він дозволяє зрозуміти, за якими саме правилами відбувається боротьба на ринку систем кібербезпеки. У таблиці 4.9 наведено ступеневий аналіз конкуренції, що визначає характер взаємодії між ключовими гравцями.

Таблиця 4.9 – Ступеневий аналіз конкуренції на ринку

№	Особливості конкурентного середовища	У чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1	**Тип конкуренції**	**Монополістична конкуренція.** На ринку присутні глобальні лідери, але існує безліч нішевих інтеграторів та Open-Source рішень	Необхідність чіткого позиціонування (УТП). Компанія повинна уникати прямого порівняння з гігантами, фокусуючись на вузькій спеціалізації
2	**Рівень конкурентної боротьби**	**Національний (Локальний).** Основна боротьба розгортається за бюджети українських державних установ та промислових підприємств	Адаптація продукту під локальні реалії: повна підтримка української мови у звітах, відповідність вимогам НБУ/ДССЗЗІ
3	**За галузевою ознакою**	**Внутрішньогалузева.** Конкуренція відбувається між різними засобами захисту (SIEM vs EDR vs Firewall)	Проводити просвітницьку роботу з клієнтами, пояснюючи, що SIEM — це верхній рівень аналітики, який об'єднує дані, а не замінює інші засоби
4	**Конкуренція за видами товарів**	**Товарно-видова.** Вибір між комерційними ліцензіями (CAPEX) та сервісними моделями на базі вільного ПЗ (OPEX)	Пропонувати гнучку фінансову модель: клієнт платить за налаштовану систему та експертну підтримку, а не за ліцензії
5	**За характером конкурентних переваг**	**Нецінова (якісна).** Клієнти обирають продукт за рівнем довіри, прозорості коду та гарантії суверенітету даних	Акцентувати увагу на безпеці рішення (Supply Chain Security): відкритий код дозволяє провести аудит та гарантувати відсутність "закладок"
6	**За інтенсивністю**	**Середня (у обраній ніші).** У сегменті Enterprise конкуренція надвисока, але в ніші бюджетних систем для промисловості та держсектору вона помірна	Використовувати стратегію "Блакитного океану": заходити в ті сегменти (застарілі виробництва, бюджетні установи), які ігноруються великими вендорами

Проведений аналіз свідчить, що для стартапу "SecureVision" найбільш вигідною є стратегія уникнення прямої конфронтації з глобальними брендами. Конкурентна боротьба має вестися не у площині функціональних перегонів, а у площині локалізації та довіри. Фокус на національному рівні конкуренції дозволяє перетворити недоліки Open-Source рішення (відсутність великого бренду) на ключові переваги для державних замовників.

Після оцінки конкуренції виконаємо більш поглиблений аналіз конкурентних умов у галузі за методом Портера. Результати цього аналізу наведено в таблиці 4.10.

Таблиця 4.10 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Характеристика ситуації в галузі	Висновки (Вплив на проєкт)
Прямі конкуренти в галузі	Ринок розділений між глобальними вендорами (Splunk) та локальними інтеграторами. Глобальні гравці пропонують потужний, але дорогий продукт	**Інтенсивність: середня.** Ми не конкуруємо з гігантами через ціну. Основна боротьба йде за увагу клієнта з іншими інтеграторами. Наша перевага — спеціалізація на ESXi/Linux
Потенційні конкуренти	Внутрішні IT-відділи (DIY), які можуть спробувати налаштувати Wazuh самостійно. Бар'єр входу для нових гравців середній	**Загроза: висока.** Необхідно довести клієнту, що купівля готового образу "SecureVision" дешевша, ніж оплата годин роботи їхніх власних інженерів на розробку з нуля
Постачальники	Розробники Wazuh Inc, Ubuntu та VMware. Код Wazuh відкритий, його неможливо відкликати. VMware є стандартом, але замінним	**Вплив: низький.** Стартап має високу незалежність. Ми не залежимо від зміни цін на ліцензії, оскільки використовуємо відкрите ПЗ. Це дозволяє гнучко керувати маржею
Клієнти	Державні установи та промислові підприємства. Мають високі вимоги до безпеки даних, суверенітету та відповідності законодавству	**Вплив: високий.** Клієнти диктують умови щодо ТЗ та процедури оплати (тендери). Необхідно адаптувати продукт під їхні жорсткі вимоги
Товари-замінники	Антивіруси (EDR) та Firewall. Вони частково закривають потребу в безпеці, але не дають повної картини інфраструктури	**Загроза: середня.** Замінники вирішують проблему лише частково. Наша задача — пояснити, що SIEM бачить картину в цілому, чого не може зробити звичайний антивірус

Аналіз за моделлю Портера засвідчує, що найбільш значущими факторами впливу є сила клієнтів та загроза DIY. Клієнти мають сильну переговорну позицію, вимагаючи високої якості за низьку ціну. Водночас, низька залежність від постачальників є критичною перевагою, яка забезпечує

економічну безпеку стартапу та дозволяє утримувати високу маржинальність послуг.

На основі аналізу конкурентів та вимог ринку необхідно сформулювати фактори конкурентоспроможності проєкту. Це ті ключові атрибути, які переконають клієнта обрати саме "SecureVision". Обґрунтування цих факторів наведено в табл. 4.11.

Таблиця 4.11 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	**Нульова вартість ліцензій**	Критичний фактор для держсектору та українського бізнесу. Відсутність щорічних платежів за ПЗ.
2	**Адаптивність до КІС**	Попередньо налаштовані правила для промислових систем, чого немає у "універсальних" SIEM.
3	**Суверенітет даних**	100% On-Premise архітектура. Дані не покидають периметр, що є вимогою закону.
4	**Швидкість розгортання**	Формат OVA-образу для ESXi дозволяє запустити SOC за 1 годину замість тижнів.
5	**Незалежність від вендора**	Відкритий код дозволяє пройти будь-який аудит безпеки (важливо для військових/держ структур).

Визначені фактори формують потужну ціннісну пропозицію. Поєднання нульової вартості ліцензій (фактор 1) та повного контролю над даними (фактор 3) робить продукт безальтернативним для багатьох державних та режимних об'єктів в Україні. Це дозволяє вигравати конкурентну боротьбу не за рахунок маркетингових бюджетів, а за рахунок відповідності реальним потребам замовників.

Після визначення основних факторів, які впливають на конкурентоспроможність системи стартапу, здійснюється аналіз, що дозволяє ідентифікувати її сильні та слабкі сторони. Результати наведено в таблиці 4.12.

Таблиця 4.12 – Порівняльний аналіз сильних та слабких сторін "SecureVision"

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Фактор конкурентоспроможності у порівнянні з Splunk						
			-3	-2	-1	0	1	2	3
1	Економічна доступність	20							+
2	Суверенітет даних	20							+
3	Адаптація до законодавства України	15						+	
4	Прозорість та безпека коду	20							+
5	Спеціалізація під VMware ESXi	12						+	
6	Гнучкість інтеграції з Legacy	15						+	

Побудований профіль яскраво демонструє, що стартап-проєкт має абсолютні переваги (+3) у ключових для цільового ринку аспектах: ціна, юридична відповідність та безпека даних. Фактори, оцінені у +2 бали, підкреслюють технічну унікальність рішення. Проєкт перемагає у чітко визначеній ніші завдяки спеціалізації, замість прямої конкуренції по функціоналу.

Наступний крок передбачає виконання SWOT-аналізу стартапу. Результати цього аналізу, наведені у таблиці 4.13, слугуватимуть основою для розробки стратегічних напрямків розвитку проєкту.

Таблиця 4.13 – SWOT-аналіз стартап-проєкту

Сильні сторони	Слабкі сторони
1. Нульова вартість ліцензій (Open Source) 2. Готова конфігурація під КІС 3. Повний контроль даних (On-Premise) 4. Незалежність від апаратного забезпечення 5. Легка інтеграція в середовище VMware	1. Відсутність сертифікату ДССЗЗІ на старті 2. Залежність від спільноти Wazuh 3. Відсутність бренду та історії 4. Обмежені ресурси на маркетинг 5. Малий штат технічної підтримки

Можливості	Загрози
1. Державні тендери на кіберзахист 2. Імпортозаміщення ПЗ 3. Партнерство з MSP-провайдерами 4. Посилення регуляторних вимог 5. Гранти на цифровізацію	1. Бюрократія в тендерах 2. Дефіцит кадрів у замовника 3. Зміна ліцензії VMware 4. Агресивний демпінг хмарних гігантів 5. Зміна законодавства

У ході проведення SWOT-аналізу стартап-проєкту були ідентифіковані основні сильні та слабкі сторони, а також можливості й загрози, які можуть впливати на його розвиток та реалізацію. На основі отриманих даних здійснимо аналіз та розгляд альтернатив для ринкового впровадження стартап-проєкту. Результати цього аналізу представлені в таблиці 4.14.

Таблиця 4.14 – Альтернативи ринкового впровадження стартап-проєкту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Продаж готового, налаштованого OVA-образу віртуальної машини через сайт. Орієнтація на масовий ринок та IT-адміністраторів, які хочуть заощадити час на налаштуванні.	**Висока.** Продукт фактично готовий (результат дипломної роботи). Витрати лише на хостинг та рекламу.	**1-2 місяці**
2	Впровадження системи "під ключ" для великих заводів та держустанов. Включає виїзд інженерів, аудит інфраструктури, навчання персоналу та річну підтримку.	**Середня.** Потребує формування кваліфікованої команди інженерів та отримання акредитації для участі в тендерах.	**3-6 місяців**
3	Надання платформи системним інтеграторам та аутсорсерам для обслуговування їхніх клієнтів (малого бізнесу). Монетизація через підписку за кожного підключеного клієнта.	**Середня.** Потребує доробки білінгової системи та API для партнерів, а також юридичної бази.	**3-6 місяців**
4	Безкоштовне поширення базової версії для популяризації бренду. Монетизація за рахунок продажу платних пакетів правил та розширеної підтримки.	**Висока.** Легко реалізувати технічно, але ризиковано фінансово (потрібен запас міцності до перших доходів).	**6-12 місяців**
5	Продаж фізичних міні-серверів із попередньо встановленим та налаштованим ПЗ "SecureVision" для промислових об'єктів.	**Низька.** Потребує значних оборотних коштів на закупівлю "заліза" та організацію логістики/складу.	**4-8 місяців**

За результатами аналізу, найбільш доцільною на початковому етапі є комбінація альтернатив 1 та 2. Продуктова модель (продаж OVA) забезпечує швидкий вхід на ринок і перші обігові кошти, тоді як сервісна модель дозволяє працювати з високомаржинальними державними та промисловими замовленнями.

4.3 Розроблення ринкової стратегії та маркетингової програми проєкту

Для успішного впровадження стартап-проєкту "SecureVision" на ринок важливо визначити основні цільові групи потенційних споживачів. Це дозволить зосередити зусилля на тих сегментах, які мають найвищий попит на продукт і демонструють готовність до його використання. Враховуючи результати аналізу ринку, клієнтських потреб та конкурентного середовища, було обрано кілька ключових груп споживачів. Їхній профіль, готовність до впровадження продукту, обсяг потенційного попиту та інші характеристики наведені у таблиці 4.15.

Таблиця 4.15 – Вибір цільових груп потенційних споживачів

№ п/п	**Опис профілю цільової групи потенційних клієнтів**	**Готовність споживачів сприйняти продукт**	**Орієнтовний попит в межах цільової групи (сегменту)**	**Інтенсивність конкуренції в сегменті**	**Простота входу у сегмент**
1	Середній бізнес	Висока	Високий	Середня	Середня
2	Держсектор	Висока	Високий	Низька	Низька
3	Малий бізнес	Низька	Середній	Висока	Висока
4	Великі корпорації	Середня	Низький	Висока	Висока
Цільовими групами обрано: держсектор та середній бізнес					

Аналіз вибору цільових груп потенційних споживачів показав, що основними сегментами для ринкового впровадження стартапу "SecureVision"

є держсектор та середній бізнесі. Малий бізнес охоплюється опосередковано через MSP-партнерів. Це забезпечує баланс між стабільністю (держзамовлення) та прибутковістю (комерційний сектор).

Для забезпечення ефективного розвитку стартапу "SecureVision" важливо визначити базову стратегію, яка дозволить максимально реалізувати сильні сторони продукту та врахувати потреби ринку. На основі аналізу конкурентоспроможності, ринкових можливостей та загроз сформовано стратегію, спрямовану на досягнення довгострокових цілей проєкту. Основні аспекти обраної стратегії наведено у таблиці 4.16.

Таблиця 4.16 – Визначення базової стратегії розвитку

№ п/п	**Обрана альтернатива розвитку проєкту**	**Стратегія охоплення ринку**	**Ключові конкурентоспроможні позиції відповідно до обраної альтернативи**	**Базова стратегія розвитку**
1	Гібридна (продукт + інтеграція)	Диференційований маркетинг	Експертиза в Linux/ESXi та розуміння вимог держсектору/промисловості	Стратегія диференціації

Стратегія диференціації дозволяє виділитися серед "сірої маси" інтеграторів. Ми продаємо не просто SIEM, а "Систему захисту КІС на базі відкритого коду", що одразу резонує з болями цільової аудиторії. Диференціація акцентує увагу на унікальних характеристиках продукту, що забезпечує стартапу конкурентні позиції та довгострокову перспективу розвитку.

Визначення стратегії конкурентної поведінки окреслює підхід компанії до взаємодії з конкурентами, залучення клієнтів та утвердження своїх позицій серед існуючих гравців ринку. опис обраного підходу наведено у таблиці 4.17.

Таблиця 4.17 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проєкт "першопрохідцем" на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Так, у ніші	Шукати нових	Частково. Копіюються загальноприйняті стандарти індустрії: логіка кореляції подій, візуалізація дашбордів та структура звітів, щоб забезпечити звичний інтерфейс користувача.	Стратегія зайняття конкурентної ніші
2	Ні, у галузі	Забирати існуючих	Ні, не буде копіювати. Ми уникаємо копіювання бізнес-моделі (ліцензії на обсяг даних) та закритої архітектури. Натомість пропонуємо прозорість коду та фіксовану вартість інтеграції, що є протилежністю підходу конкурентів.	Стратегія диференціації

Обрана стратегія підтверджує, що "SecureVision" діє як "нішер", уникаючи прямої конфронтації з глобальними лідерами. Компанія фокусується на створенні нового попиту серед клієнтів, які раніше були "невидимими" для великих вендорів через малі бюджети (стратегія пошуку нових споживачів). Водночас, відмова від копіювання ліцензійної моделі конкурентів та акцент на технологічній відкритості дозволяє ефективно працювати на ринку імпортозаміщення, пропонуючи державним та промисловим замовникам безпечну альтернативу (стратегія диференціації).

Виходячи з результатів попереднього аналізу у та враховуючи обрану стратегію розвитку і підхід до конкурентної поведінки, необхідно визначити стратегію позиціонування продукту. Її мета – створити унікальний образ продукту в свідомості цільової аудиторії, який підкреслює його конкурентні переваги та відповідає очікуванням клієнтів. Основні аспекти розробленої стратегії позиціонування представлені в таблиці 4.18.

Таблиця 4.18 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проєкту	Вибір асоціацій, які мають формувати комплексну позицію власного проєкту
1	**Державні установи та об'єкти критичної інфраструктури (G2B):** повний суверенітет даних (On-Premise), відповідність законодавству (КСЗІ), відсутність валютних ліцензійних витрат.	**Стратегія диференціації**	Гарантія зберігання даних у периметрі замовника, відкритий код для аудиту безпеки, адаптація звітів під українське законодавство.	**1. Суверенітет** (Ваші дані тільки у Вас). **2. Законність** (відповідність вимогам регуляторів). **3. Незалежність** (відсутність "голки" ліцензій).
2	**Промислові підприємства (Industry/B2B):** стабільність технологічних процесів, підтримка застарілого обладнання, захист від простоїв виробництва.	**Стратегія зайняття конкурентної ніші**	Спеціалізація на пасивному моніторингу КІС, гнучкість інтеграції з Legacy-системами через кастомні агенти.	**1. Безпека виробництва** (захист без зупинки процесів). **2. Надійність** (стабільна робота на ESXi). **3. Контроль** (повна видимість інфраструктури).
3	**Середній бізнес та MSP-провайдери:** економія бюджету, швидкий старт, простота масштабування та обслуговування.	**Стратегія диференціації**	Готовий до розгортання образ (OVA) за ціною інтеграції, відсутність капітальних витрат (CAPEX) на ПЗ.	**1. Економія** (enterprise-рівень безкоштовно). **2. Швидкість** (розгортання за 1 годину). **3. Ефективність** (максимальний ROI з першого дня).

Розроблена стратегія позиціонування формує цілісний образ продукту, що поєднує в собі підхід до суверенітету даних та економічної ефективності. Ключова асоціація для всього проєкту — "безпечна незалежність". Це дозволяє стартапу "SecureVision" дистанціюватися від дорогих іноземних

вендорів, пропонуючи клієнтам не просто софт, а гарантію контролю над власною інфраструктурою та бюджетом.

Одним із пріоритетних завдань у формуванні маркетингової програми є визначення основних переваг продукту для цільової аудиторії. Це дозволяє зрозуміти, як саме товар відповідає потребам ринку та чим він може перевершити конкурентів. У процесі аналізу було виділено ключові вигоди, які система "SecureVision" пропонує споживачам, а також її основні переваги на ринку. Результати цього аналізу представлені в таблиці 4.19.

Таблиця 4.19 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
1	Виконання вимог закону	Готові шаблони звітів для перевірок регуляторів	Адаптація під українське законодавство (НБУ, ДССЗЗІ) "з коробки", чого немає у західних аналогів
2	Захист виробництва	Пасивний метод збору логів (без активного сканування)	Гарантія безпеки для технологічних процесів (SCADA), усунення ризику зупинки обладнання.
3	Економія бюджету	Відсутність капітальних витрат (CAPEX) на ліцензії	Модель "Pay only for Service" — оплата лише за налаштування, що в рази дешевше річних підписок
4	Суверенітет даних	Архітектура On-Premise (дані зберігаються локально)	Повна ізоляція чутливої інформації від зовнішніх хмарних провайдерів (на відміну від SaaS-рішень)
5	Незалежність від вендора	Відкритий вихідний код (Open Source)	Відсутність ризику "Vendor Lock-in" (прив'язки до постачальника) та можливість незалежного аудиту коду

Сформульована концепція товару чітко відповідає на ключові виклики ринку: необхідність економії, юридичну відповідність та безпеку даних. Додавання пунктів про суверенітет даних та незалежність від вендора перетворює "SecureVision" з просто "дешевого рішення" на стратегічно безпечний вибір. Це дозволяє ефективно конкурувати з хмарними гігантами, граючи на полі національної безпеки та корпоративної незалежності.

Таблиця 4.20 – Опис трьох рівнів моделі товару

<table>
<tr><th>Рівні товару</th><th colspan="3">Сутність та складові</th></tr>
<tr><td>І. Товар за задумом</td><td colspan="3">Автоматизована система моніторингу кіберзагроз для комп'ютерно-інтегрованих систем, що забезпечує централізований збір подій, виявлення атак та відповідність законодавчим вимогам (КСЗІ) без капітальних витрат на ліцензії</td></tr>
<tr><td rowspan="8">ІІ. Товар у реальному виконанні</td><td>Властивості/характеристики</td><td>М/Нм</td><td>Вр/Тх/Тл/Ор</td></tr>
<tr><td>1. Технологічна готовність: постачання у вигляді налаштованого образу віртуальної машини (OVA), що не потребує інсталяції ОС</td><td>Нм</td><td>Тл</td></tr>
<tr><td>2. Економічна ефективність: відсутність ліцензійних платежів завдяки Open Source архітектурі (Wazuh, Ubuntu)</td><td>Нм</td><td>Е</td></tr>
<tr><td>3. Адаптивність до КІС: наявність спеціалізованих правил кореляції для промислових протоколів та Legacy-систем.</td><td>Нм</td><td>Тх</td></tr>
<tr><td>4. Суверенітет: архітектура гарантує збереження даних виключно в межах периметра організації</td><td>Нм</td><td>Ор</td></tr>
<tr><td>5. Надійність: забезпечення відмовостійкості засобами середовища віртуалізації VMware ESXi (Snapshots).</td><td>Нм</td><td>Тх</td></tr>
<tr><td colspan="3">Якість: відповідність стандартам стабільності LTS-версій Ubuntu та верифіковані правила детектування загроз (без помилкових спрацювань)</td></tr>
<tr><td colspan="3">Пакування: цифровий дистрибутив (захищений архів або посилання на сховище) з контрольною сумою (SHA256) та інструкцією</td></tr>
<tr><td colspan="3">Марка: SecureVision</td></tr>
<tr><td rowspan="2">ІІІ. Товар з підкріпленням</td><td colspan="3">До продажу: демонстрація можливостей на інфраструктурі замовника, аудит сумісності обладнання</td></tr>
<tr><td colspan="3">Після продажу: технічна підтримка інтеграції, регулярні оновлення бази правил виявлення загроз, доступ до закритого репозиторію скриптів</td></tr>
<tr><td colspan="4">За рахунок чого потенційний товар буде захищено від копіювання: патент</td></tr>
</table>

Аналіз трирівневої моделі товару засвідчує, що "SecureVision" позиціонується не просто як програмне забезпечення, а як комплексний сервіс

із високою технологічною готовністю. Поєднання економічної ефективності за рахунок Open Source та організаційної гарантії суверенітету даних створює унікальну ціннісну пропозицію для державних та промислових замовників. Наявність третього рівня (товару з підкріпленням) є критично важливою, оскільки саме сервісна підтримка та оновлення правил дозволяють монетизувати проєкт і захистити його від прямого копіювання конкурентами.

Правильне ціноутворення впливає на конкурентоспроможність продукту та його привабливість для споживачів. Для визначення оптимальної ціни було проаналізовано рівні вартості товарів-замінників і товарів-аналогів, а також враховано рівень доходів цільової аудиторії. Результати дослідження дозволяють встановити верхню та нижню межі ціни, які забезпечать баланс між цінністю продукту та доступністю для клієнтів. Деталі розрахунків наведено в таблиці 4.21.

Таблиця 4.21 – Визначення меж встановлення ціни

№ п/п	**Рівень цін на товари-замінники**	**Рівень цін на товари-аналоги**	**Рівень доходів цільової групи споживачів**	**Верхня та нижня межі встановлення ціни на товар/послугу**
1	400 – 1 000	25 000 – 50 000	1 500 – 8 000	500 – 3 000

Розрахунок цінових меж для "SecureVision" демонструє, що обраний середній сегмент дозволяє запропонувати конкурентну вартість продукту, зберігаючи його технічні переваги. Орієнтація на аналіз вартості аналогів і замінників, а також доходів клієнтів, забезпечує баланс між прибутковістю стартапу та доступністю продукту для цільової аудиторії. Такий підхід сприяє розширенню клієнтської бази і закріпленню позицій на ринку.

Після визначення меж ціноутворення стартапу важливо розробити ефективну систему збуту, яка забезпечить успішне виведення продукту "SecureVision" на ринок. Формування системи збуту враховує особливості закупівельної поведінки цільових клієнтів, їхні потреби у технічній підтримці

та налаштуванні продукту. Оптимальний підхід до організації збуту наведено в таблиці 4.22.

Таблиця 4.22 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	**Держсектор (G2B):** довгий цикл прийняття рішень, сувора регламентація через тендери, вимога повної відповідності ТЗ та наявності офіційної документації	Моніторинг тендерних майданчиків, підготовка юридично коректної документації, офіційне листування, надання гарантійних листів	**Канал нульового рівня** (прямий маркетинг): Розробник => Державний замовник (через E-Tender).	**Власна (пряма):** участь у публічних закупівлях без посередників для забезпечення конкурентної ціни
2	**Промисловість (B2B):** раціональна поведінка, вимога попереднього тестування, потреба в консультаціях щодо безпеки виробництва	Організація презентацій та пілотних впроваджень, технічний консалтинг, налаштування інтеграції з наявним обладнанням (SCADA/ПЛК)	**Канал нульового рівня**: Розробник => Підприємство	**Власна (пряма):** особисті продажі через переговори з технічними директорами (CTO/CISO)
3	**Малий бізнес через MSP (B2B2C):** Потреба в готовому рішенні "під ключ", яке обслуговує стороння IT-компанія. Низький чек, але масовість замовлень	Технічна підтримка партнерів, навчання персоналу партнерів, забезпечення стабільності API та оновлень	**Канал одного рівня** (через посередника): Розробник => MSP-провайдер => Кінцевий клієнт	**Залучена (непряма):** побудова партнерської мережі системних інтеграторів

Розроблена система збуту для стартапу "SecureVision" гібридну систему збуту. Для стратегічно важливих клієнтів (держава та велика промисловість) використовується канал нульового рівня, що дозволяє компанії повністю контролювати процес впровадження та якість послуг. Для масштабування на масовий ринок малого бізнесу залучається канал одного рівня (партнери), що дозволяє охопити велику кількість клієнтів без необхідності розширювати власний штат відділу продажів.

Після формування ефективної системи збуту для стартапу "SecureVision", наступним кроком є розроблення концепції маркетингових комунікацій. Ця концепція спрямована на формування стійкого іміджу продукту, підвищення його впізнаваності та донесення ключових переваг до цільової аудиторії. Враховуючи специфіку ринку, було визначено основні канали комунікації, ключові позиції для позиціонування продукту та завдання рекламних звернень, які представлені у таблиці 4.23.

Таблиця 4.23 – Концепція маркетингових комунікацій

№ п/п	**Специфіка поведінки цільових клієнтів**	**Канали комунікації, якими користуються цільові клієнти**	**Ключові позиції, обрані для позиціонування**	**Завдання рекламного повідомлення**	**Концепція рекламного звернення**
1	**Держслужбовці (G2B):** шукають відповідність нормативним документам (накази НБУ, ДССЗЗІ), уникають ризиків, орієнтуються на офіційні рекомендації	Офіційне ділове листування, галузеві наради та семінари з кібербезпеки, майданчики публічних закупівель	**Суверенітет та законність:** імпортозаміщення та повна локалізація даних	Інформування про наявність легальної, безкоштовної у ліцензуванні альтернативи іноземному ПЗ, що відповідає КСЗІ	*"Державна безпека без залежності від ліцензій: українська платформа моніторингу, що гарантує суверенітет даних"*
2	**Керівники промисловості (B2B):** прагматики, які бояться зупинки виробництва. Читають технічну аналітику, довіряють кейсам та рекомендаціям партнерів	Професійна мережа LinkedIn, спеціалізовані технічні форуми, профільні промислові виставки та конференції	**Надійність та економія:** захист безперервних процесів (SCADA) без капітальних витрат	Подолання страху перед впровадженням нового ПЗ, демонстрація безпечності пасивного моніторингу для обладнання	*"Ваше виробництво під захистом 24/7. Професійний SOC, який не потребує мільйонних бюджетів та не зупиняє конвеєр"*

Продовження таблиці 4.23

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікації, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
3	**IT-адміністратори та MSP (Tech):** технічні ентузіасти, які шукають готові рішення для економії часу. Довіряють коду, а не маркетинговим гаслам.	GitHub (репозиторії), YouTube (технічні огляди та туторіали), Telegram-канали про InfoSec та DevOps	**Швидкість та технологічність:** готовий інструмент на базі Linux/ESXi для профі	Стимулювання до завантаження та тестування продукту, демонстрація простоти розгортання через OVA	*"Припиніть витрачати тижні на налаштування Linux. Розгорніть готовий SIEM на ESXi за 1 годину та контролюйте все"*

Розроблена комунікаційна стратегія базується на принципі вузького таргетингу. Для державного сектору використовується офіційно-діловий стиль із наголосом на патріотизмі та законності. Для промисловості — прагматичний підхід з акцентом на фінансовій вигоді та стабільності. Для технічної спільноти — пряма демонстрація можливостей продукту через туторіали та код. Такий підхід дозволяє максимально ефективно використовувати маркетинговий бюджет.

4.4 Бізнес-модель реалізації стартап-проекту та оцінювання його економічної ефективності

Для цілісного розуміння логіки функціонування стартапу та механізмів отримання прибутку розроблено бізнес-модель за методологією Business Model Canvas. Ця модель візуалізує, як саме технічне рішення на базі Wazuh та ESXi перетворюється на комерційний продукт, визначаючи ключових партнерів, структуру витрат та потоки доходів. Деталізована бізнес-модель проєкту "SecureVision" представлена в табл. 4.24.

Таблиця 4.24 – Бізнес-модель Canvas стартап-проєкту "SecureVision"

<table>
<tr><th>Ключові партнери</th><th>Ключові види діяльності</th><th>Ціннісна пропозиція</th><th>Взаємовідносини з клієнтами</th><th>Сегменти споживачів</th></tr>
<tr><td rowspan="3">1. Системні інтегратори: компанії, що впроваджують IT-інфраструктуру "під ключ" і потребують модуля безпеки.
2. Державні регулятори: взаємодія для отримання експертних висновків відповідності КСЗІ.
3. VMware/Broadcom: технологічне партнерство для сертифікації OVA-образу.
4. Хостинг-провайдери: для розміщення репозиторіїв оновлень та демо-стендів.</td><td>1. R&D та розробка: постійне оновлення правил кореляції (Wazuh Ruleset) під нові загрози.
2. Інженерія: збірка та оптимізація віртуальних образів для ESXi.
3. Продажі та тендери: підготовка документації для Prozorro та переговори з бізнесом.
4. Підтримка: технічний консалтинг та реагування на інциденти клієнтів.</td><td rowspan="3">Для Держсектору:
1. Повний суверенітет даних (On-Premise).
2. Відсутність валютних витрат на ліцензії.
3. Відповідність законодавству.
Для Промисловості:
1. Захист SCADA без зупинки виробництва.
2. Швидке розгортання.
Для Бізнесу:
1. Економія бюджету (на 80% дешевше Splunk).
2. Незалежність від вендора (Open Source).</td><td>1. Персональна підтримка: для великих замовників (виділений інженер).
2. Самообслуговування: детальна документація, Wiki-база знань, відео-інструкції.
3. Спільнота: закритий форум для клієнтів для обміну досвідом налаштування.
4. Автоматизація: регулярна розсилка оновлень правил безпеки.</td><td rowspan="3">1. Державні установи (G2B): міністерства, відомства, комунальні підприємства.
2. Середні промислові підприємства: заводи, агрохолдинги, логістичні центри.
3. MSP-провайдери: IT-компанії, що обслуговують малий бізнес.
4. Великі корпорації: для захисту віддалених філій та ізольованих сегментів мережі.</td></tr>
<tr><td>Ключові ресурси</td><td>Канали збуту</td></tr>
<tr><td>1. Інтелектуальна власність: унікальні скрипти конфігурації, набір правил виявлення загроз.
2. Людські ресурси: команда DevSecOps та аналітиків SOC.
3. Технологічна платформа: лабораторний стенд на базі ESXi для тестування.</td><td>1. Електронні майданчики: для участі в державних тендерах.
2. Прямі продажі: особисті контакти, презентації, пілотні проєкти.
3. Партнерська мережа: продаж ліцензій через інтеграторів.</td></tr>
<tr><td colspan="2">Структура витрат</td><td colspan="3">Потоки доходів</td></tr>
<tr><td colspan="2">1. Фонд оплати праці: зарплати інженерів та менеджерів (основна стаття).
2. R&D: витрати на дослідження нових вразливостей.
3. Маркетинг: участь у виставках, підтримка сайту.
4. Адміністративні: юридичний супровід, податки, оренда серверів.</td><td colspan="3">1. Плата за впровадження: разовий платіж за розгортання системи.
2. Сервісна підписка: щомісячна/щорічна плата за оновлення правил та консультації.
3. Кастомна розробка: плата за написання конекторів до специфічного обладнання.</td></tr>
</table>

Розроблена бізнес-модель Canvas демонструє, що стартап "SecureVision" будує свою стійкість на сервісній моделі монетизації. Оскільки ядро продукту є безкоштовним (Open Source), основні потоки доходів генеруються за рахунок експертизи (впровадження, підтримка, адаптація), що мінімізує залежність від зовнішніх постачальників ПЗ. Ключовим активом компанії є не програмний код сам по собі, а інтелектуальна власність у вигляді спеціалізованих правил кореляції та компетенції команди, що дозволяє ефективно обслуговувати специфічні сегменти (G2B, Industry).

Для ефективної реалізації бізнес-моделі необхідно сформувати команду, яка володіє як технічними компетенціями (DevSecOps, аналіз загроз), так і навичками B2B-продажів. Оскільки проєкт орієнтований на роботу з державним сектором та промисловістю, критично важливою є роль керівника, здатного вести переговори та працювати з тендерною документацією. У таблиці 4.25 наведено оптимальну організаційну структуру стартапу.

Таблиця 4.25 – Організаційна структура та команда проєкту

Роль	Ключові функції та обов'язки	Обґрунтування
1. Керівник проєкту (CEO)	Стратегічне планування, управління фінансами, робота з тендерними майданчиками (Prozorro), переговори з ключовими клієнтами (G2B/Enterprise), юридичний супровід	Власник продукту. Відповідає за монетизацію та проходження бюрократичних процедур у держсекторі, що є критичним для обраної бізнес-моделі
2. Технічний лід (DevSecOps Engineer)	Архітектура рішення, налаштування та оптимізація ядра (Ubuntu + Wazuh), підготовка OVA-образів для VMware ESXi, автоматизація оновлень, керування репозиторіями	Ключова технічна фігура. Забезпечує стабільність роботи продукту та його сумісність з інфраструктурою клієнтів
3. Аналітик безпеки (SOC Analyst)	Розробка правил кореляції подій, створення шаблонів звітів (Compliance), написання інструкцій та скриптів реагування (Active Response), технічна підтримка L2/L3	Створює основну додану вартість продукту — експертний контент (правила та звіти), який відрізняє "SecureVision" від "пустого" Wazuh.
4. Маркетолог / Менеджер по роботі з партнерами	Ведення сайту та блогу, створення технічного контенту (кейси, статті), комунікація з MSP-партнерами, організація вебінарів	Забезпечує вхідний потік лідів та будує довіру до бренду через демонстрацію експертизи

Запропонована структура команди є компактною та збалансованою. Вона покриває всі критичні зони відповідальності: управлінську, інженерну, продуктову та комунікаційну. Такий склад дозволяє мінімізувати постійні витрати на етапі запуску, зберігаючи при цьому здатність надавати якісний сервіс та технічну підтримку, що є головною конкурентною перевагою перед DIY-рішеннями.

Виробнича стратегія проєкту базується на принципах DevSecOps та цифрової дистрибуції. Оскільки продуктом є програмний код та налаштування, "виробництвом" вважається процес створення та тестування еталонного образу системи, а "логістикою" — забезпечення безпечної передачі цього образу замовнику. У таблиці 4.26 описано ключові етапи організації цих процесів.

Таблиця 4.26 – Організація розробки та дистрибуції продукту

Етап	Технологія / Інструмент	Обґрунтування (для стратегії диференціації)
1. R&D та розробка (Development)	**In-house Engineering:** використання власного лабораторного стенда на базі VMware ESXi для конфігурації ядра системи (Ubuntu + Wazuh) та пакування у формат OVA	Забезпечення повного контролю над архітектурою рішення. Створення "золотого образу" (Golden Image), який є еталоном безпеки та надійності для клієнта
2. Забезпечення якості (QA & Testing)	**Cyber Range Testing:** автоматизована перевірка правил кореляції на віртуальному полігоні з симуляцією реальних векторів атак перед кожним релізом	Гарантія стабільності роботи ("Zero Bugs" політика). Мінімізація помилкових спрацювань, що критично важливо для промислових середовищ
3. Цифрова дистрибуція (Delivery)	**Secure Cloud Storage:** надання клієнту тимчасового захищеного посилання на завантаження образу із хмарного сховища з перевіркою цілісності (SHA256)	Миттєва доставка продукту без логістичних витрат. Криптографічний захист гарантує, що клієнт отримує оригінальний, не модифікований хакерами продукт
4. Підтримка та оновлення (Maintenance)	**Private Repository:** доставка оновлень правил безпеки (Detection Rules) через підключення до приватного Git-репозиторію компанії	Формування постійного потоку цінності: продукт стає "розумнішим" з кожним оновленням, адаптуючись до нових загроз без участі клієнта

Запропонована модель організації процесів дозволяє досягти максимальної операційної ефективності. Заміна фізичного виробництва на цифрову розробку зводить змінні витрати до мінімуму, перетворюючи їх на фіксовані витрати на R&D. Використання захищених каналів дистрибуції підтверджує імідж надійного постачальника кібербезпеки та дозволяє масштабувати бізнес на всю територію країни без географічних обмежень.

Для успішного запуску проєкту критично важливо мати чіткий план дій, прив'язаний до часових рамок. Календарний графік дозволяє синхронізувати технічну доробку продукту з маркетинговими активностями та юридичними процедурами. У таблиці 4.27 розроблено поетапний план виведення системи "SecureVision" на ринок, розрахований на перші 6 місяців.

Таблиця 4.27 – Організація розробки та дистрибуції продукту

Етап	Місяці	Ключові завдання	Результат
1. Організаційний та R&D	1-2	Реєстрація суб'єкта господарювання, фіналізація "Golden Image" (OVA) на лабораторному стенді, отримання авторських прав на код	Юридично оформлений бізнес, готовий до продажу продукт версії v1.0, захищена інтелектуальна власність
2. Пілотне впровадження	3	Проведення безкоштовного впровадження на базі дружнього підприємства або ВНЗ, збір метрик ефективності та відгуків	Перший успішний кейс, виправлення багів, отримання реальних даних для маркетингу
3. Маркетинг та підготовка до тендерів	4	Запуск веб-сайту, акредитація на майданчиках Prozorro/E-Tender, публікація статей у профільних медіа	Сформована впізнаваність бренду, доступ до ринку державних закупівель, перші вхідні ліди
4. Активні продажі	5-6	Участь у тендерах, прямі переговори з промисловими клієнтами, укладання перших комерційних контрактів	Отримання першого прибутку, досягнення точки беззбитковості, перші 3-5 платних клієнтів

Розроблений графік підтверджує можливість швидкого виходу на ринок. План передбачає отримання перших доходів вже на 5-6 місяць, що є оптимальним показником для стартапу та знижує ризик касових розривів.

Оцінювання потреби у початковому капіталі є фундаментом фінансової моделі стартапу. Оскільки проєкт є програмним продуктом, він не вимагає закупівлі сировини чи верстатів, а основні витрати пов'язані з технічним забезпеченням та інтелектуальною працею. У таблиці 4.28 наведено розрахунок стартових інвестицій, необхідних для запуску проєкту.

Таблиця 4.28 – Витрати та інвестиції

Стаття витрат	**Характеристика**	**Орієнтовні початкові витрати, $**
1. Обладнання	Придбання потужного сервера (або оренда Dedicated Server) для розгортання середовища VMware ESXi та тестування навантажень	$1,500
2. Юридичні витрати	Реєстрація компанії, консультації щодо авторського права, оплата доступу до тендерних майданчиків	$500
3. Маркетинг	Розробка Landing Page, контент-маркетинг, участь у профільних конференціях	$500
4. Оборотні кошти (R&D)	Оціночна вартість інтелектуальної праці засновників на етапі доробки продукту (3 міс.)	$3,000
РАЗОМ	**Загальні початкові інвестиції**	**$5,500**

Розрахунок показує, що проєкт відноситься до категорії низькозатратних. Сума інвестицій у $5,500 є доступною для запуску власними силами без залучення зовнішніх інвесторів, що дозволяє засновникам зберегти 100% контролю над компанією. Основна частка інвестицій — це інтелектуальний внесок команди, що мінімізує реальні фінансові ризики.

Фінальним етапом є розрахунок ключових показників, що свідчать про економічну доцільність бізнесу. На основі планових витрат та визначеної ринкової ціни визначено точку беззбитковості та рентабельність інвестицій. У таблиці 4.29 наведено економічні показники ефективності проєкту "SecureVision".

Таблиця 4.29 – Показники економічної ефективності

Показник	Орієнтовна оцінка	Обґрунтування
Собівартість продукту	**$150**	Змінні витрати на одного клієнта, що включають: оплату 3-4 годин роботи інженера на розгортання та первинне налаштування системи, а також витрати на хмарну інфраструктуру (репозиторії, трафік)
Середня ціна продажу	**$1,500**	Середнє значення ціни контракту, що включає вартість продукту та базові послуги з інтеграції
Валова маржа	**≈90%**	Висока маржинальність ($1,350 з кожного продажу) досягається за рахунок використання безкоштовних Open Source компонентів, що мінімізує ліцензійні відрахування
Точка беззбитковості	**5 угод**	Для покриття стартових інвестицій ($5,500) необхідно реалізувати лише 5 проєктів (при чистому прибутку $1,350 з одного клієнта)
Термін окупності	**4-6 місяців**	При песимістичному сценарії продажів (1 клієнт на місяць) інвестиції окупляться протягом першого півріччя роботи

Фінансова модель проєкту демонструє високу ефективність. Собівартість у $150 (сформована переважно витратами часу на впровадження) при ціні продажу $1,500 забезпечує валову маржу на рівні 90%. Це дозволяє компанії швидко генерувати вільні кошти для покриття постійних витрат та маркетингу. Точка беззбитковості у 5 клієнтів є легко досяжною, враховуючи потенціал ринку державних та промислових закупівель.

Висновки до розділу 4

У четвертому розділі магістерської дисертації проведено комплексне розроблення стартап-проєкту "SecureVision", присвяченого впровадженню системи автоматизованого моніторингу загроз у комп’ютерно-інтегрованих системах. Проведений маркетинговий, технологічний та економічний аудит підтвердив, що розроблене технічне рішення має високий потенціал для комерціалізації та здатне зайняти стійку позицію на ринку інформаційної безпеки України. Аналіз ринкового середовища засвідчив наявність гострого попиту на доступні та надійні інструменти кіберзахисту, зумовленого посиленням регуляторних вимог до державних установ та зростанням

кіберзагроз для промислового сектору. Виявлено, що на ринку існує вільна ніша між дорогими комерційними SIEM-системами, які вимагають значних валютних витрат на ліцензії, та складними у налаштуванні Open-Source рішеннями, що потребують високої кваліфікації персоналу.

Технологічний аудит підтвердив здійсненність ідеї створення універсального віртуального аплаєнсу на базі операційної системи Ubuntu та платформи моніторингу Wazuh, оптимізованого для роботи у середовищі віртуалізації VMware ESXi. Таке поєднання технологій дозволяє забезпечити високу надійність та масштабованість системи при нульовій вартості ліцензійних відрахувань. Розроблена концепція товару передбачає створення ціннісної пропозиції, що базується на трьох ключових перевагах: економічній доступності, повному суверенітеті даних завдяки локальному зберіганню та адаптації під специфічні вимоги вітчизняного законодавства і застарілого промислового обладнання. Це дозволяє стартапу ефективно конкурувати з глобальними вендорами, пропонуючи замовникам не просто програмне забезпечення, а гарантію безпеки та відповідності нормативним вимогам.

В рамках стратегічного планування було обрано модель поведінки "нішера", що передбачає фокусування на специфічних сегментах ринку, які недостатньо охоплені великими гравцями, а саме на державному секторі та середніх промислових підприємствах. Для реалізації цієї стратегії розроблено гібридну бізнес-модель, яка поєднує прямі продажі через участь у державних тендерах та партнерську мережу системних інтеграторів для охоплення малого бізнесу. Позиціонування продукту будується на асоціаціях з безпекою, незалежністю від вендорів та економічною ефективністю, що резонує з актуальними потребами клієнтів в умовах необхідності імпортозаміщення та оптимізації бюджетів. Комунікаційна стратегія передбачає диференційований підхід до кожної групи клієнтів, акцентуючи увагу на юридичній відповідності для держслужбовців та стабільності технологічних процесів для керівників промисловості.

Економічний аналіз проєкту продемонстрував його високу інвестиційну привабливість та фінансову стійкість. Розрахунки показали, що для запуску стартапу необхідний відносно низький обсяг початкових інвестицій, основну частину яких складає інтелектуальний внесок розробників. Завдяки використанню відкритого програмного забезпечення та цифровій дистрибуції продукту, компанія досягає високої валової маржі. Собівартість продукту формується переважно за рахунок витрат часу інженерів на впровадження та підтримку, що дозволяє гнучко керувати ціноутворенням. Розрахована точка беззбитковості знаходиться на рівні п'яти угод, що є легко досяжним показником навіть на ранніх етапах діяльності, а термін окупності проєкту становить від чотирьох до шести місяців залежно від темпів продажів.

Організаційна структура стартапу розроблена з урахуванням потреби у швидкому прийнятті рішень та мінімізації постійних витрат, включаючи ролі керівника, технічного лідера та аналітика безпеки. Виробничий план базується на сучасних практиках DevSecOps, що забезпечує постійне оновлення правил виявлення загроз та автоматизований контроль якості перед кожним релізом. Календарний графік реалізації передбачає вихід на отримання першого прибутку вже через пів року після старту, включаючи етапи юридичного оформлення, пілотного впровадження та активної участі у тендерних закупівлях.

Узагальнюючи результати четвертого розділу, можна стверджувати, що стартап-проєкт "SecureVision" є повністю готовим до практичної реалізації. Він має чітко окреслену цільову аудиторію, переконливі конкурентні переваги та життєздатну економічну модель. Впровадження цього проєкту не лише принесе комерційний успіх засновникам, але й сприятиме підвищенню рівня кіберстійкості критичної інфраструктури, забезпечуючи надійний захист інформаційних систем від сучасних загроз.

ВИСНОВКИ

У магістерській дисертації вирішено актуальне науково-прикладне завдання підвищення ефективності захисту комп’ютерно-інтегрованих систем шляхом розробки та впровадження автоматизованої системи моніторингу загроз на базі технології SIEM.

На основі проведених теоретичних досліджень, практичної реалізації та експериментальної перевірки отримано наступні результати:

1. Проаналізовано специфіку та ландшафт загроз сучасних КІС. Встановлено, що підходи до безпеки в промислових середовищах фундаментально відрізняються від класичних IT-систем пріоритетністю доступності технологічного процесу над конфіденційністю даних. Визначено, що наявність застарілого обладнання та специфічних протоколів унеможливлює використання активного сканування мережі. Аналіз прецедентів кіберфізичних атак (Stuxnet, Triton) показав, що критичні вектори атак спрямовані на системи протиаварійного захисту, що вимагає застосування пасивних методів моніторингу та "компенсаційних контролів" згідно зі стандартами IEC 62443 та NIST SP 800-82r3.

2. Обґрунтовано вибір технологічного стеку для побудови системи моніторингу. За результатами порівняльного аналізу провідних SIEM-рішень визначено платформу Wazuh як оптимальний інструмент для захисту КІС. Її архітектура, що поєднує функції SIEM та XDR, дозволяє реалізувати необхідні компенсаційні контроли (моніторинг цілісності файлів, виявлення вразливостей) без додаткового навантаження на промислову мережу. Використання Open-Source моделі забезпечує економічну ефективність та незалежність від вендорів, що є критичним фактором для стратегічних об’єктів інфраструктури.

3. Реалізовано автоматизовану систему моніторингу. Створено віртуалізований полігон на базі гіпервізора першого типу VMware ESXi, що дозволило емулювати умови реального промислового середовища з ізоляцією

обчислювальних ресурсів. У межах цієї інфраструктури розгорнуто серверну частину Wazuh на базі Ubuntu Server у конфігурації All-in-one із забезпеченням захисту каналів зв’язку SSL-сертифікацією. Така архітектура гарантує цілісність даних при їх передачі від розподілених агентів до центрального сервера аналізу.

4. Здійснено інтеграцію гетерогенних компонентів у єдиний контур безпеки. Реалізовано розгортання та реєстрацію агентів моніторингу на різнорідних операційних системах (Windows та Linux). Це дозволило централізувати збір журналів подій з усієї інфраструктури, подолавши проблему фрагментації даних, яка є типовою для складних комп’ютерно-інтегрованих систем.

5. Експериментально підтверджено ефективність системи виявлення загроз. Шляхом моделювання сценаріїв кібератак (зокрема, brute-force атак та несанкціонованого доступу) доведено працездатність механізмів кореляції подій. Система продемонструвала здатність у реальному часі фільтрувати інформаційний шум, ідентифікувати аномальну активність та автоматично присвоювати інцидентам високий рівень критичності. Це підтверджує, що розроблене рішення здатне значно скоротити час реакції на інциденти.

6. Розроблено та економічно обґрунтовано стартап-проєкт "SecureVision". Запропоновано концепцію комерціалізації розробленого рішення у вигляді віртуального аплаєнсу. Маркетинговий та економічний аналіз підтвердив наявність вільної ринкової ніші для доступних систем кіберзахисту, що забезпечують суверенітет даних. Обрана стратегія "нішера" та відсутність ліцензійних відрахувань дозволяють проєкту конкурувати з дорогими пропрієтарними рішеннями.

Таким чином, у магістерській дисертації розв’язано задачу автоматизації процесів моніторингу загроз у комп’ютерно-інтегрованих системах. Запропоноване рішення є масштабованим, економічно вигідним та готовим до практичного впровадження як на рівні окремих технологічних ліній, так і в масштабах підприємства.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Securing Industrial Control Systems: Components, Cyber Threats, and Machine Learning-Driven Defense Strategies – PubMed [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/37960539/.

2. Cyber Security: Perspective of Challenges in Operational Technology Systems in Power Sector – CPRI Journal. [Online]. Available: https://cprijournal.in/index.php/pr/article/view/1168.

3. Guide to Operational Technology Security – NIST Technical Series Publications. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r3.pdf.

4. SP 800-82 Rev. 2, Guide to Industrial Control Systems Security | CSRC. [Online]. Available: https://csrc.nist.gov/pubs/sp/800/82/r2/final.

5. NIST SP 800-82r3: Enhancing OT Security with Dragos and NP-View. [Online]. Available: https://www.dragos.com/blog/nist-sp-800-82r3-enhancing-ot-security-with-dragos-and-np-view.

6. SP 800-82 Rev. 3, Guide to Operational Technology Security | CSRC. [Online]. Available: https://csrc.nist.gov/pubs/sp/800/82/r3/ipd.

7. Guide to Operational Technology Security – NIST Technical Series Publications. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r3.pdf.

8. Guide to Industrial Control Systems Security – NIST Technical Series Publications [Online]. Available: https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-82r2.pdf.

9. ISA/IEC 62443 Explained: OT Cybersecurity Standards | Dragos. [Online]. Available: https://www.dragos.com/blog/isa-iec-62443-concepts.

10. Practical Overview of Implementing IEC 62443 Security Levels in Industrial Control Applications – InstMC. [Online]. Available: https://www.instmc.org/_userfiles/pages/groupfiles/cyber_security/wp_implementing_iec_62443_security_levels_in_industrial_control_applications_1.pdf.

11. NIST Publishes SP 800-82, Revision 3 | CSRC. [Online]. Available: https://csrc.nist.gov/News/2023/nist-publishes-sp-800-82-revision-3.

12. A Security Information and Event Management Pattern. [Online]. Available: https://epub.uni-regensburg.de/41139/1/A%20Security%20Information%20and%20Event%20Management%20Pattern.pdf.

13. The convergence of IT and OT – Cybersecurity Related Challenges and Best Practices. [Online]. Available: https://www.ijcttjournal.org/Volume-69%20Issue-2/IJCTT-V69I2P113.pdf.

14. Industrial control systems: The biggest cyber threat – ResearchGate. [Online]. Available: https://www.researchgate.net/publication/348268112_Industrial_control_systems_The_biggest_cyber_threat.

15. Common Cybersecurity Vulnerabilities in Industrial Control – CISA. [Online]. Available: https://www.cisa.gov/sites/default/files/recommended_practices/DHS_Common_Cybersecurity_Vulnerabilities_ICS_2010.pdf.

16. Дипломний проект – CORE. [Online]. Available: https://core.ac.uk/download/pdf/323529362.pdf.

17. Industrial Control Systems: The Biggest Cyber Threat – Biomedical Journal of Scientific & Technical Research. [Online]. Available: https://biomedres.us/pdfs/BJSTR.MS.ID.005143.pdf.

18. What Is MITRE ATT&CK Framework? – Palo Alto Networks. [Online]. Available: https://www.paloaltonetworks.com/cyberpedia/what-is-mitre-attack.

19. Matrix – ICS | MITRE ATT&CK®. [Online]. Available: https://attack.mitre.org/matrices/ics/.

20. Cybersafety: A System-theoretic Approach to Identify Cyber-vulnerabilities & Mitigation Requirements in Industrial Control Systems – IEEE Xplore. [Online]. Available: https://ieeexplore.ieee.org/document/9468393/.

21. Cyber Risk to Mission Case Study: Triton – DTIC. [Online]. Available: https://apps.dtic.mil/sti/trecms/pdf/AD1183008.pdf.

22. What is OT malware? – NCSC.GOV.UK. [Online]. Available: https://www.ncsc.gov.uk/blog-post/what-is-ot-malware.

23. A Taxonomy-based Analysis of Attacks on Industrial Control Systems – St. Cloud State Repository. [Online]. Available: https://repository.stcloudstate.edu/cgi/viewcontent.cgi?article=1138&context=msia_etds.

24. Cyber-Attacks on Energy Infrastructure — A Literature Overview – MDPI. [Online]. Available: https://www.mdpi.com/2076-3417/15/17/9233.

25. Industrial and Critical Infrastructure Security: Technical Analysis of Real-Life Security Incidents – IEEE Xplore. [Online]. Available: https://ieeexplore.ieee.org/iel7/6287639/9312710/09638617.pdf.

26. An overview of Cyber Attack to Industrial Control System – Aidic. [Online]. Available: https://www.aidic.it/cet/19/77/152.pdf.

27. TRITON: The First ICS Cyberattack on Safety Instrument – Black Hat. [Online]. Available: https://i.blackhat.com/us-18/Wed-August-8/us-18-Carcano-TRITON-How-It-Disrupted-Safety-Systems-And-Changed-The-Threat-Landscape-Of-Industrial-Control-Systems-Forever-wp.pdf.

28. Triton Malware Spearheads Latest Attacks on Industrial Systems | McAfee Blogs – Trellix. [Online]. Available: https://www.trellix.com/blogs/research/triton-malware-spearheads-latest-generation-of-attacks-on-industrial-systems1/.

29. Impact and Key Challenges of Insider Threats on Organizations – MDPI. [Online]. Available: https://www.mdpi.com/2079-9292/9/9/1460.

30. Cybersecurity Advances in SCADA Systems – SAI Organization. [Online]. Available: https://thesai.org/Downloads/Volume14No8/Paper_35-Cybersecurity_Advances_in_SCADA_Systems.pdf.

31. SOAR vs. SIEM: What Is the Difference? – Palo Alto Networks. [Online]. Available: https://www.paloaltonetworks.com/cyberpedia/what-is-soar-vs-siem.

32. The Evolution of SIEM Solutions: From Log Management to AI – Gurucul. [Online]. Available: https://gurucul.com/blog/siem-solutions-evolution-to-next-gen-security-solutions/.

33. The Path to Choosing a SIEM System – Systematic Literature Review – FH Wedel. [Online]. Available: https://www.fh-wedel.de/fileadmin/Mitarbeiter/Records/Hase_2024_-_The_Path_to_Choosing_a_SIEM_System_-_A_Systematic_Literature_Review.pdf.

34. What is SIEM? – IBM. [Online]. Available: https://www.ibm.com/think/topics/siem.

35. XDR vs. SIEM vs. SOAR: What's the Difference? – CrowdStrike. [Online]. Available: https://www.crowdstrike.com/en-us/cybersecurity-101/next-gen-siem/xdr-vs-siem-vs-soar/.

36. Security Information and Event Management: Analysis, Trends, and Usage in Critical Infrastructures – ResearchGate. [Online]. Available: https://www.researchgate.net/publication/353214895_Security_Information_and_Event_Management_SIEM_Analysis_Trends_and_Usage_in_Critical_Infrastructures.

37. Security Information and Event Management (SIEM): Analysis – PMC. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC8309804/.

38. Revolutionizing SIEM Security: An Innovative Correlation Engine – PMC. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC11314677/.

39. Systematic Literature Review of Security Event Correlation Methods – IEEE Xplore. [Online]. Available: https://ieeexplore.ieee.org/iel7/6287639/6514899/09760432.pdf.

40. Best Security Information and Event Management Reviews 2025 – Gartner Peer Insights. [Online]. Available: https://www.gartner.com/reviews/market/security-information-event-management.

41. 2025 Gartner® Magic Quadrant™ for SIEM – Splunk. [Online]. Available: https://www.splunk.com/en_us/form/gartner-siem-magic-quadrant.html.

42. Splunk Enterprise vs Wazuh – The Open Source Security Platform 2025 – Gartner. [Online]. Available: https://www.gartner.com/reviews/market/security-information-event-management/compare/product/splunk-enterprise-vs-wazuh-the-open-source-security-platform.

43. About WAZUH vs SPLUNK FOR SIEM – Reddit. [Online]. Available: https://www.reddit.com/r/Splunk/comments/1igixcl/about_wazuh_vs_splunk_for_siem/.

44. IBM Security QRadar SIEM vs Wazuh – Gartner. [Online]. Available: https://www.gartner.com/reviews/market/security-information-event-management/compare/product/ibm-security-qradar-siem-vs-wazuh-the-open-source-security-platform.

45. Monitoring Rapid SCADA with Wazuh | Wazuh. [Online]. Available: https://wazuh.com/blog/monitoring-rapid-scada-with-wazuh/.

46. Overview | Wazuh. [Online]. Available: https://wazuh.com/platform/overview/.

47. Wazuh SIEM for Cyber Security and Threat Mitigation in Apparel Industries. [Online]. Available: https://deerhillpublishing.com/index.php/ijemm/article/download/279/262/2277.

48. IBM QRadar SIEM vs. Splunk Enterprise vs. Wazuh Comparison – SourceForge. [Online]. Available: https://sourceforge.net/software/compare/IBM-QRadar-SIEM-vs-Splunk-vs-Wazuh/.

49. Case Study–iSecNG–Strengthening Cybersecurity with Wazuh. [Online]. Available: https://wazuh.com/uploads/2025/04/Case-Study-iSecNG-Strengthening-Cybersecurity-with-Wazuh.pdf.

50. The top free and open source SIEM tools for 2025 – Red Canary. [Online]. Available: https://redcanary.com/cybersecurity-101/security-operations/top-free-siem-tools/.

51. Wazuh Documentation. Components. [Online] Available: https://documentation.wazuh.com/current/getting-started/components/index.html?utm_source=chatgpt.com

52. Wazuh Documentation. Wazuh Agent. [Online] Available: https://documentation.wazuh.com/current/getting-started/components/wazuh-agent.html

53. Wazuh Documentation. Wazuh Server. [Online] Available: https://documentation.wazuh.com/current/getting-started/components/wazuh-server.html

54. Wazuh Documentation. Wazuh Indexer. [Online] Available: https://documentation.wazuh.com/current/getting-started/components/wazuh-indexer.html

55. Wazuh Documentation. Wazuh Dashboard. [Online] Available: https://documentation.wazuh.com/current/getting-started/components/wazuh-dashboard.html

56. Wazuh Documentation. Wazuh Architecture. [Online] Available: https://documentation.wazuh.com/current/getting-started/architecture.html

57. VMware, Inc., “VMware vSphere | Virtualization Platform,” VMware, 2025. [Online]. Available: https://www.vmware.com/products/cloud-infrastructure/vsphere.

58. VMware, Inc., VMware ESXi Installation and Setup, VMware vSphere 6.7 Documentation, Broadcom, 2025. [Online]. Available: https://techdocs.broadcom.com/us/en/vmware-cis/vsphere/vsphere/6-7/esxi-installation-and-setup-6-7.html.

59. R. Sheldon and B. Posey, “What is VMware vSphere? vSphere vs. ESXi vs. vCenter,” TechTarget – SearchVMware, Jul. 30, 2024. [Online]. Available: https://www.techtarget.com/searchvmware/definition/VMware-vSphere.

60. VMware, Inc., Security of the VMware vSphere Hypervisor, white paper, updated Jan. 2014. [Online]. Available: https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/whitepaper/techpaper/vmw-white-paper-secrty-vsphr-hyprvsr-uslet-101.pdf.

61. M. Aleksic, “VMware ESXi: Installation and Setup,” PhoenixNAP Knowledge Base, Oct. 1, 2025. [Online]. Available: https://phoenixnap.com/kb/vmware-esxi.

62. K. Kumar and C. Stankowic, VMware vSphere 5.5 Essentials. Packt Publishing, 2015. [Online]. Available: https://www.oreilly.com/library/view/vmware-vsphere-5-5/9781784398750/.

63. VMware, Inc., VMware ESXi 8.0 Update 3e Security Target, 2025. [Online]. Available: https://www.vmware.com/docs/vmware-esxi-8-0-update-3e-security-target.

64. Розроблення стартап-проєкту [Електронний ресурс] : Методичні рекомендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей / За заг. ред. О.А. Гавриша. – Київ : НТУУ «КПІ», 2016. – 28 с.

65. Гавриш О. А., Бояринова К. О., Копішинська К. О. Розробка стартап-проектів. Конспект лекцій : навчальний посібник для студентів спеціальностей 151 – «Автоматизація та комп’ютерно-інтегровані технології» та 152 – «Метрологія та інформаційно-вимірювальна техніка»; КПІ ім. Ігоря Сікорського. Електронні текстові данні (1 файл: 2,88 Мбайт). Київ : КПІ ім. Ігоря Сікорського, 2019. 188 с.

66. Гавриш О. А., Бояринова К. О., Копішинська К. О. Розробка стартап-проектів: практикум: навчальний посібник для студентів спеціальностей 151 – «Автоматизація та комп’ютерно-інтегровані технології» та 152 – «Метрологія та інформаційно-вимірювальна техніка»; КПІ ім. Ігоря Сікорського. Електронні текстові данні (1 файл: 2,11 Мбайт). Київ : КПІ ім. Ігоря Сікорського, 2019. 116 с.

ДОДАТОК А

Лістинг розгортання Wazuh Indexer на Ubuntu Server

А.1 Завантаження інструмента для генерації сертифікатів та базового конфігураційного файлу

```
curl -sO https://packages.wazuh.com/4.14/wazuh-certs-tool.sh
curl -sO https://packages.wazuh.com/4.14/config.yml
nano ./config.yml
```

А.2 Генерація сертифікатів для Wazuh-компонентів

```
bash ./wazuh-certs-tool.sh -A
```

А.3 Архівація та підготовка сертифікатів

```
tar -cvf ./wazuh-certificates.tar -C ./wazuh-certificates/ .
rm -rf ./wazuh-certificates
```

А.4 Встановлення необхідних системних компонентів

```
apt-get install debconf adduser procps
apt-get install gnupg apt-transport-https
```

А.5 Додавання GPG-ключа та репозиторію Wazuh

```
curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH \
  | gpg --no-default-keyring --keyring gnupg-
ring:/usr/share/keyrings/wazuh.gpg --import
chmod 644 /usr/share/keyrings/wazuh.gpg

echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg]
https://packages.wazuh.com/4.x/apt/ stable main" \
  | tee -a /etc/apt/sources.list.d/wazuh.list

apt-get update
```

А.6 Встановлення Wazuh Indexer

```
apt-get -y install wazuh-indexer
```

А.7 Підготовка конфігурації Indexer та розміщення сертифікатів

```
nano /etc/wazuh-indexer/opensearch.yml
NODE_NAME=wazuh

mkdir /etc/wazuh-indexer/certs

tar -xf ./wazuh-certificates.tar -C /etc/wazuh-indexer/certs/ \
  ./wazuh.pem ./wazuh-key.pem ./admin.pem ./admin-key.pem ./root-
ca.pem

mv -n /etc/wazuh-indexer/certs/wazuh.pem /etc/wazuh-
indexer/certs/indexer.pem
mv -n /etc/wazuh-indexer/certs/wazuh-key.pem /etc/wazuh-
indexer/certs/indexer-key.pem

chmod 500 /etc/wazuh-indexer/certs
```

```
chmod 400 /etc/wazuh-indexer/certs/*
chown -R wazuh-indexer:wazuh-indexer /etc/wazuh-indexer/certs
```

А.8 Активація служби Indexer

```
systemctl daemon-reload
systemctl enable wazuh-indexer
systemctl start wazuh-indexer
systemctl status wazuh-indexer
```

А.9 Вимкнення репозиторію Wazuh після встановлення

```
sed -i "s/^deb /#deb /" /etc/apt/sources.list.d/wazuh.list
apt update
```

А.10 Ініціалізація безпекової конфігурації OpenSearch

```
nano /usr/share/wazuh-indexer/bin/indexer-security-init.sh
```

А.11 Перевірка стану Indexer та вузла кластера

```
curl -k -u admin https://192.168.1.100:9200
curl -k -u admin https://192.168.1.100:9200/_cat/nodes?v
```

А.12 Конфігураційний файл config.yml

```
nodes:
  # Wazuh indexer nodes
  indexer:
    - name: wazuh
      ip: 192.168.1.100
    #- name: node-2
    #  ip: "<indexer-node-ip>"
    #- name: node-3
    #  ip: "<indexer-node-ip>"

  # Wazuh server nodes
  server:
    - name: wazuh-1
      ip: 192.168.1.100
    #  node_type: master
    #- name: wazuh-2
    #  ip: "<wazuh-manager-ip>"
    #  node_type: worker
    #- name: wazuh-3
    #  ip: "<wazuh-manager-ip>"
    #  node_type: worker

  # Wazuh dashboard nodes
  dashboard:
    - name: dashboard
      ip: 192.168.1.100
```

А.13 Конфігураційний файл /etc/wazuh-indexer/opensearch.yml

```
network.host: "192.168.1.100"
node.name: "wazuh"
cluster.initial_master_nodes:
- "wazuh"
cluster.name: "wazuh-cluster"

node.max_local_storage_nodes: "3"
path.data: /var/lib/wazuh-indexer
path.logs: /var/log/wazuh-indexer
```

```
plugins.security.ssl.http.pemcert_filepath: /etc/wazuh-
indexer/certs/indexer.pem
plugins.security.ssl.http.pemkey_filepath: /etc/wazuh-
indexer/certs/indexer-key.pem
plugins.security.ssl.http.pemtrustedcas_filepath: /etc/wazuh-
indexer/certs/root-ca.pem

plugins.security.ssl.transport.pemcert_filepath: /etc/wazuh-
indexer/certs/indexer.pem
plugins.security.ssl.transport.pemkey_filepath: /etc/wazuh-
indexer/certs/indexer-key.pem
plugins.security.ssl.transport.pemtrustedcas_filepath: /etc/wazuh-
indexer/certs/root-ca.pem

plugins.security.ssl.http.enabled: true
plugins.security.ssl.transport.enforce_hostname_verification: false
plugins.security.ssl.transport.resolve_hostname: false

plugins.security.authcz.admin_dn:
- "CN=admin,OU=Wazuh,O=Wazuh,L=California,C=US"

plugins.security.check_snapshot_restore_write_privileges: true
plugins.security.enable_snapshot_restore_privilege: true

plugins.security.nodes_dn:
- "CN=wazuh,OU=Wazuh,O=Wazuh,L=California,C=US"

plugins.security.restapi.roles_enabled:
- "all_access"
- "security_rest_api_access"

plugins.security.system_indices.enabled: true
plugins.security.system_indices.indices: [
  ".plugins-ml-model",
  ".plugins-ml-task",
  ".opendistro-alerting-config",
  ".opendistro-alerting-alert*",
  ".opendistro-anomaly-results*",
  ".opendistro-anomaly-detector*"
]

compatibility.override_main_response_version: true
```

ДОДАТОК Б

Лістинг розгортання Wazuh Manager на Ubuntu Server

Б.1 Додавання репозиторію Wazuh та оновлення списку пакунків

```
apt-get install gnupg apt-transport-https

curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH \
  | gpg --no-default-keyring --keyring gnupg-
ring:/usr/share/keyrings/wazuh.gpg --import

chmod 644 /usr/share/keyrings/wazuh.gpg

echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg]
https://packages.wazuh.com/4.x/apt/ stable main" \
  | tee -a /etc/apt/sources.list.d/wazuh.list

apt-get update
```

Б.2 Встановлення Wazuh Manager

```
apt-get -y install wazuh-manager
systemctl status wazuh-manager
```

Б.3 Встановлення Filebeat та базової конфігурації

```
apt-get -y install filebeat

curl -so /etc/filebeat/filebeat.yml \
  https://packages.wazuh.com/4.14/tpl/wazuh/filebeat/filebeat.yml

nano /etc/filebeat/filebeat.yml
```

Б.4 Створення keystore Filebeat та збереження облікових даних Indexer

```
filebeat keystore create

echo admin | filebeat keystore add username --stdin --force
echo admin | filebeat keystore add password --stdin --force
```

Б.5 Завантаження шаблону індексу та модуля Wazuh для Filebeat

```
curl -so /etc/filebeat/wazuh-template.json \

https://raw.githubusercontent.com/wazuh/wazuh/v4.14.1/extensions/elast
icsearch/7.x/wazuh-template.json

chmod go+r /etc/filebeat/wazuh-template.json

curl -s https://packages.wazuh.com/4.x/filebeat/wazuh-filebeat-
0.4.tar.gz \
  | tar -xvz -C /usr/share/filebeat/module
```

Б.6 Підготовка сертифікатів для Filebeat

```
NODE_NAME=wazuh-1

mkdir /etc/filebeat/certs

tar -xf ./wazuh-certificates.tar -C /etc/filebeat/certs/ \
```

```
  ./wazuh-1.pem ./wazuh-1-key.pem ./root-ca.pem

mv -n /etc/filebeat/certs/wazuh-1.pem /etc/filebeat/certs/filebeat.pem
mv -n /etc/filebeat/certs/wazuh-1-key.pem
/etc/filebeat/certs/filebeat-key.pem

chmod 500 /etc/filebeat/certs
chmod 400 /etc/filebeat/certs/*
chown -R root:root /etc/filebeat/certs
```

Б.7 Налаштування облікових даних Indexer у Wazuh Manager

```
echo admin | /var/ossec/bin/wazuh-keystore -f indexer -k username
echo admin | /var/ossec/bin/wazuh-keystore -f indexer -k password
```

Б.8 Редагування основного конфігураційного файлу Wazuh Manager

```
nano /var/ossec/etc/ossec.conf
```

Б.9 Активація служби Wazuh Manager

```
systemctl daemon-reload
systemctl enable wazuh-manager
systemctl start wazuh-manager
systemctl status wazuh-manager
```

Б.10 Активація та перевірка Filebeat

```
systemctl daemon-reload
systemctl enable filebeat
systemctl start filebeat

filebeat test output
```

Б.11 Вимкнення репозиторію Wazuh після розгортання

```
sed -i "s/^deb /#deb /" /etc/apt/sources.list.d/wazuh.list
apt update
```

Б.12 Конфігураційний файл /etc/filebeat/filebeat.yml

```
# Wazuh - Filebeat configuration file
output.elasticsearch:
  hosts: ["192.168.1.100:9200"]
  protocol: https
  username: ${username}
  password: ${password}
  ssl.certificate_authorities:
    - /etc/filebeat/certs/root-ca.pem
  ssl.certificate: "/etc/filebeat/certs/filebeat.pem"
  ssl.key: "/etc/filebeat/certs/filebeat-key.pem"

setup.template.json.enabled: true
setup.template.json.path: '/etc/filebeat/wazuh-template.json'
setup.template.json.name: 'wazuh'
setup.ilm.overwrite: true
setup.ilm.enabled: false

filebeat.modules:
  - module: wazuh
    alerts:
      enabled: true
    archives:
      enabled: false

logging.level: info
```

```
logging.to_files: true
logging.files:
  path: /var/log/filebeat
  name: filebeat
  keepfiles: 7
  permissions: 0644

logging.metrics.enabled: false

seccomp:
  default_action: allow
  syscalls:
  - action: allow
    names:
    - rseq
```

Б.13 Фрагмент конфігураційного файлу /var/ossec/etc/ossec.conf

```
<indexer>
  <enabled>yes</enabled>
  <hosts>
    <host>https://192.168.1.100:9200</host>
  </hosts>
  <ssl>
    <certificate_authorities>
      <ca>/etc/filebeat/certs/root-ca.pem</ca>
    </certificate_authorities>
    <certificate>/etc/filebeat/certs/filebeat.pem</certificate>
    <key>/etc/filebeat/certs/filebeat-key.pem</key>
  </ssl>
</indexer>
```

ДОДАТОК В

Лістинг розгортання Wazuh Dashboard на Ubuntu Server

В.1 Додавання репозиторію Wazuh та імпорт GPG-ключа

```
apt-get install debhelper tar curl libcap2-bin
apt-get install gnupg apt-transport-https

curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH \
  | gpg --no-default-keyring --keyring gnupg-
ring:/usr/share/keyrings/wazuh.gpg --import

chmod 644 /usr/share/keyrings/wazuh.gpg

echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg]
https://packages.wazuh.com/4.x/apt/ stable main" \
  | tee -a /etc/apt/sources.list.d/wazuh.list

apt-get update
```

В.2 Встановлення Wazuh Dashboard

```
apt-get -y install wazuh-dashboard
```

В.3 Підготовка TLS-сертифікатів для Dashboard

```
NODE_NAME=dashboard

mkdir /etc/wazuh-dashboard/certs

tar -xf ./wazuh-certificates.tar -C /etc/wazuh-dashboard/certs/ \
  ./dashboard.pem ./dashboard-key.pem ./root-ca.pem

chmod 500 /etc/wazuh-dashboard/certs
chmod 400 /etc/wazuh-dashboard/certs/*
chown -R wazuh-dashboard:wazuh-dashboard /etc/wazuh-dashboard/certs
```

В.4 Застосування конфігурації Dashboard (opensearch_dashboards.yml)

```
nano /etc/wazuh-dashboard/opensearch_dashboards.yml
```

В.5 Додавання конфігурації взаємодії Dashboard ↔ Manager

```
nano /usr/share/wazuh-dashboard/data/wazuh/config/wazuh.yml
```

В.6 Активація та запуск служби Dashboard

```
systemctl daemon-reload
systemctl enable wazuh-dashboard
systemctl start wazuh-dashboard
systemctl status wazuh-dashboard
```

В.7 Вимкнення репозиторію Wazuh після встановлення

```
sed -i "s/^deb /#deb /" /etc/apt/sources.list.d/wazuh.list
apt update
```

В.8 Конфігураційний файл

/etc/wazuh-dashboard/opensearch_dashboards.yml

```
server.host: 192.168.1.100
server.port: 443

opensearch.hosts: https://192.168.1.100:9200
opensearch.ssl.verificationMode: certificate
#opensearch.username:
#opensearch.password:
opensearch.requestHeadersAllowlist: ["securitytenant","Authorization"]

opensearch_security.multitenancy.enabled: false
opensearch_security.readonly_mode.roles: ["kibana_read_only"]

server.ssl.enabled: true
server.ssl.key: "/etc/wazuh-dashboard/certs/dashboard-key.pem"
server.ssl.certificate: "/etc/wazuh-dashboard/certs/dashboard.pem"

opensearch.ssl.certificateAuthorities:
  ["/etc/wazuh-dashboard/certs/root-ca.pem"]

uiSettings.overrides.defaultRoute: /app/wz-home

# Session expiration settings
opensearch_security.cookie.ttl: 900000
opensearch_security.session.ttl: 900000
opensearch_security.session.keepalive: true
```

В.9 Конфігураційний файл

/usr/share/wazuh-dashboard/data/wazuh/config/wazuh.yml

```
hosts:
  - default:
      url: https://192.168.1.100
      port: 55000
      username: wazuh-wui
      password: wazuh-wui
      run_as: false
```