

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

ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра
на тему: Проєкт системи багаторівневого захисту
мережної інфраструктури для обробки конфіденційних
даних з підсистемою виявлення вторгнень

Виконав:

студент 4-го курсу групи КІ-22-1

спеціальності 123 «Комп'ютерна інженерія»

 Іван СИМАШКО

Керівник: старший викладач кафедри КІ та КБ

 Ольга ДЯЧУК

Рецензент: доцент кафедри КІ та КБ, к.т.н.,

 Олександр ПІРОГ

Житомир – 2026

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Факультет інформаційно-комп'ютерних технологій

Кафедра комп'ютерної інженерії та кібербезпеки

Освітній рівень: бакалавр

Спеціальність 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

комп'ютерної інженерії та

кібербезпеки

 Андрій ЄФІМЕНКО

22 травня 2026 р.

ЗАВДАННЯ

на випускню кваліфікаційну роботу бакалавра

Студент: Сіماشко Іван Борисович

1. Тема роботи: «Проект системи багаторівневого захисту мережної інфраструктури для обробки конфіденційних даних з підсистемою виявлення вторгнень»

Вихідні дані роботи (зазначається предмет і об'єкт дослідження):

Об'єкт дослідження: процеси забезпечення інформаційної безпеки в мережевій інфраструктурі, що обробляє конфіденційні дані.

Предмет дослідження: методи та засоби побудови ешелонованої системи захисту мережі з підсистемою виявлення вторгнень.

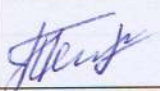
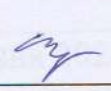
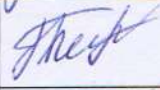
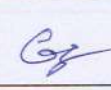
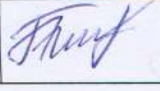
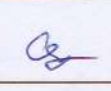
Вихідні дані для роботи: нормативні документи та стандарти у сфері інформаційної безпеки (Закон України «Про захист інформації в інформаційно-комунікаційних системах» № 80/94-ВР, НД ТЗІ 2.5-010-03, НД ТЗІ 2.7-010-09, Регламент GDPR 2016/679, NIST SP 800-30, NIST SP 800-53); методології моделювання загроз і побудови багаторівневого захисту

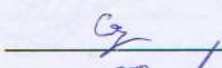
(Defense-in-Depth, STRIDE, PASTA, OWASP Top 10, Cyber Kill Chain); технічна документація використаних open-source рішень (OPNsense, Suricata, Wazuh, PostgreSQL із pgcrypto та pgAudit, WireGuard, pgBackRest); аналітичні звіти щодо кіберінцидентів (Verizon DBIR, ENISA Threat Landscape, Mandiant M-Trends); набори сигнатур Emerging Threats Open.

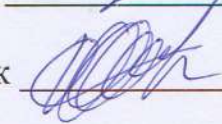
Обладнання та програмне забезпечення:

1. Середовище віртуалізації – Oracle VM VirtualBox;
2. Операційні системи – OPNsense 26.1.6, Ubuntu Server 26.04 LTS, Ubuntu 26.04, Kali Linux;
3. Периметровий захист і мережа – OPNsense (фаєрвол, NAT, маршрутизація), WireGuard (VPN);
4. Виявлення вторгнень – Suricata (NIDS) з наборами правил Emerging Threats Open;
5. СУБД та її захист – PostgreSQL з розширеннями pgcrypto і pgAudit, шифрування TLS;
6. Захист хоста – UFW, AppArmor, auditd, OpenSSH (ключі Ed25519);
7. Моніторинг і реагування – Wazuh (Manager, Indexer, Dashboard, варіант All-in-One) зі стеком Elastic;
8. Резервне копіювання – pgBackRest (шифрування AES-256-CBC) із плануванням через cron;
9. Допоміжні засоби – Nmap, curl, Cisco Packet Tracer (побудова схеми топології).

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

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Олександра ПОКОТИЛО	 2.02.2026	 2.02.2026
2	Олександра ПОКОТИЛО	 2.02.2026	 2.02.2026
3	Олександра ПОКОТИЛО	 2.02.2026	 2.02.2026

Студент  Іван СИМАШКО

Керівник  Ольга ДЯЧУК

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назви етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка завдання	5 квітня 2026	Виконано
2	Аналіз існуючих прикладів мережевих систем багаторівневого запису	6 – 8 квітня 2026	Виконано
3	Пошук та аналіз програмного забезпечення	9 – 10 квітня 2026	Виконано
4	Організація віртуалізації на основному ПК	13 – 14 квітня 2026	Виконано
5	Планування етапів для поступового створення системи	14 – 22 квітня 2026	Виконано
6	Встановлення віртуальних машин	23 – 25 квітня 2026	Виконано
7	Робота з операційними системами, оновлення	26 – 28 квітня 2026	Виконано
8	Налаштування необхідних для багаторівневого захисту технологій	29 квітня 2026 – 1 червня 2026	Виконано
9	Написання пояснювальної записки	1 – 5 червня 2026	Виконано
10	Підготовка презентації	6 – 10 червня 2026	Виконано
11	Отримання відгуку керівника та рецензії на кваліфікаційну роботу	15 червня 2026	Виконано
12	Захист	18 червня 2026	

Студент Сє Іван СІМАШКО

Керівник Ольга ДЯЧУК Ольга ДЯЧУК

АНОТАЦІЯ

Кваліфікаційна робота бакалавра: 120 сторінок, 92 сторінки основного тексту, 68 рисунків, 26 використаних джерел, 2 додатки на 28 сторінках.

Кваліфікаційну роботу присвячено створенню системи багаторівневого (ешелонованого) захисту мережної інфраструктури для обробки конфіденційних даних у базі даних PostgreSQL з підсистемою виявлення вторгнень, побудованої виключно на основі вільного програмного забезпечення (open-source).

У першому розділі виконано аналіз предметної області та постановку задачі: розглянуто сучасний стан інформаційної безпеки мережних інфраструктур, типові вектори атак на конфіденційні дані, нормативно-правову базу, концепцію ешелонованого захисту (Defense-in-Depth) та моделі загроз STRIDE/PASTA, яким протидіятиме проєктована система.

У другому розділі обґрунтовано вибір open-source компонентів захисту (OPNsense, Suricata, Wazuh, PostgreSQL із розширеннями pgcrypto та pgAudit, TLS, WireGuard, pgBackRest), наведено їх порівняння, визначено сфери захисту, які вони покривають, та спроектовано архітектуру їх спільної роботи за рівнями ешелонованості.

У третьому розділі здійснено практичну реалізацію системи на віртуальному стенді: розгорнуто й налаштовано всі компоненти, проведено тестування ефективності захисту (сканування портів Nmap, атаки brute-force, SQL-ін'єкції, перевірка шифрування даних та відновлення з резервної копії) і детальний аналіз роботи кожної підсистеми.

У висновках підбито підсумки проведеної роботи та наведено рекомендації щодо впровадження й масштабування системи. У списку використаних джерел наведено посилання на матеріали, що використовувалися під час теоретичного аналізу.

					ККІКБ.КР.Б – 123 – 13 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Проект системи багаторівневого захисту мережної інфраструктури для обробки конфіденційних даних з підсистемою виявлення вторгнень	Літ.	Арк.	Аркуші
Розроб.		Сіماشко І. Б.	Сш	16.06.26				
Керівник		Дячук О. Ю.	Дячук	16.06.26			5	
Реценз.		Пірог О. В.	Пірог			ФІКТ, Гр. КІ-22-1		
Н. контр.		Дячук О. Ю.	Дячук	16.06.26				
Затверд.		Сфіменко А. А.	Сфіменко	16.06.26				

Додатки містять лістинги конфігураційних файлів розгорнутих компонентів та екранні форми встановлення операційних систем тестового полігону.

Мета кваліфікаційної роботи – розроблення проєкту ешелонованої системи захисту мережної інфраструктури на основі open-source рішень із підсистемою моніторингу та виявлення вторгнень для безпечної обробки конфіденційних даних.

Ключові слова: ЕШЕЛОНОВАНИЙ ЗАХИСТ, ІНФОРМАЦІЙНА БЕЗПЕКА, СИСТЕМА ВИЯВЛЕННЯ ВТОРГНЕНЬ, OPNSENSE, SURICATA, WAZUH, SIEM, POSTGRESQL, ШИФРУВАННЯ ДАНИХ, WIREGUARD, СЕГМЕНТАЦІЯ МЕРЕЖІ, OPEN-SOURCE.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

ANNOTATION

Bachelor's qualification thesis: 120 pages, 92 pages of main text, 68 figures, 26 references, 2 appendices on 28 pages.

The thesis is devoted to the development of a multi-layered (defense-in-depth) protection system for a network infrastructure that processes confidential data stored in a PostgreSQL database, equipped with an intrusion detection subsystem and built entirely on open-source software.

The first chapter analyses the subject area and states the task: the current state of network information security, typical attack vectors against confidential data, the legal framework, the Defense-in-Depth concept and the STRIDE/PASTA threat models. The second chapter substantiates the choice of open-source security components (OPNsense, Suricata, Wazuh, PostgreSQL with pgcrypto and pgAudit, TLS, WireGuard, pgBackRest), compares them and designs the architecture of their joint operation across protection layers. The third chapter covers the practical implementation on a virtual testbed, the deployment and configuration of all components, and the testing of protection effectiveness (Nmap port scanning, brute-force attacks, SQL injections, data encryption and backup recovery), followed by a detailed analysis of each subsystem.

The aim of the thesis is to develop a defense-in-depth network protection system based on open-source solutions with a monitoring and intrusion detection subsystem for the secure processing of confidential data.

Keywords: DEFENSE-IN-DEPTH, INFORMATION SECURITY, INTRUSION DETECTION SYSTEM, OPNSENSE, SURICATA, WAZUH, SIEM, POSTGRESQL, DATA ENCRYPTION, WIREGUARD, NETWORK SEGMENTATION, OPEN-SOURCE.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	9
ВСТУП.....	10
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ.....	11
1.1. Аналіз сучасного стану інформаційної безпеки мережевих інфраструктур....	11
1.2. Концепція ешелонованого захисту (Defense-in-Depth)	13
1.3. Аналіз існуючих рішень та технологій	15
1.4. Постановка задачі проєктування	18
Висновки до розділу 1	20
РОЗДІЛ 2. ПРОЄКТУВАННЯ СИСТЕМИ ЕШЕЛОНОВАНОГО ЗАХИСТУ.....	21
2.1. Розроблення архітектури системи захисту	21
2.2. Проєктування підсистеми периметрового захисту	23
2.3. Проєктування підсистеми виявлення вторгнень.....	27
2.4. Проєктування підсистеми захисту сервера та СКБД	28
2.5. Проєктування підсистеми моніторингу та реагування	30
2.6. Проєктування підсистеми резервного копіювання.....	33
2.7. Модель загроз та оцінка ризиків спроектованої системи	34
Висновки до розділу 2	36
РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ	37
3.1. Розгортання тестового стенду.....	37
3.2. Налаштування периметрового захисту	45
3.3. Налаштування системи виявлення вторгнень	54
3.4. Налаштування захисту сервера Ubuntu та PostgreSQL	59
3.5. Розгортання підсистеми моніторингу	68
3.6. Тестування ефективності системи захисту.....	74
3.7. Рекомендації щодо впровадження та масштабування	83
Висновки до розділу 3	85
ВИСНОВКИ	87
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	90
ДОДАТКИ.....	93

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ACL – Access Control List (список контролю доступу)

AES – Advanced Encryption Standard (вдосконалений стандарт шифрування)

APT – Advanced Persistent Threat (розширена цілеспрямована кіберзагроза)

CSRF – Cross-Site Request Forgery (міжсайтова підробка запиту)

DDoS – Distributed Denial of Service (розподілена відмова в обслуговуванні)

DHCP – Dynamic Host Configuration Protocol (протокол динамічної конфігурації хоста)

EVE JSON – Extensible Event Format JSON (розширений формат подій у вигляді JSON)

GUI – Graphical User Interface (графічний інтерфейс користувача)

IDS – Intrusion Detection System (система виявлення вторгнень)

IP – Internet Protocol (міжмережевий протокол)

IPS – Intrusion Prevention System (система запобігання вторгненням)

JSON – JavaScript Object Notation (текстовий формат обміну даними)

LAN – Local Area Network (локальна обчислювальна мережа)

NAT – Network Address Translation (трансляція мережевих адрес)

NIDS – Network Intrusion Detection System (мережева система виявлення вторгнень)

OSI – Open Systems Interconnection (еталонна модель взаємодії відкритих систем)

SIEM – Security Information and Event Management (система управління інформацією та подіями безпеки)

SSH – Secure Shell (захищений мережевий протокол оболонки)

TCP – Transmission Control Protocol (протокол керування передачею)

TLS – Transport Layer Security (протокол захисту транспортного рівня)

VLAN – Virtual Local Area Network (віртуальна локальна мережа)

VPN – Virtual Private Network (віртуальна приватна мережа)

WAN – Wide Area Network (глобальна обчислювальна мережа)

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

У сучасних умовах глобальної цифровізації та постійного зростання обсягів конфіденційної інформації, що обробляється в корпоративних мережах, забезпечення надійного захисту ІТ-інфраструктури стає критичним завданням для будь-якої організації. Як свідчить статистика кіберінцидентів, постійно зростає кількість цілеспрямованих атак, спроб викрадення даних, експлуатації вразливостей периметру та впливу програм-вимагачів (Ransomware). Традиційного захисту лише на рівні мережевого периметру сьогодні недостатньо для протидії сучасним векторам атак.

Забезпечення конфіденційності, цілісності та доступності даних вимагає переходу до комплексної архітектури ешелонованого захисту (Defense-in-Depth), яка передбачає впровадження кількох незалежних рубежів оборони на мережевому, хостовому рівнях та рівні управління базами даних. Водночас використання виключно комерційних продуктів Enterprise-класу часто є занадто дорогим рішенням. Тому розробка та впровадження систем захисту на базі вільного програмного забезпечення (open-source рішень), які здатні забезпечити високий рівень безпеки та централізований моніторинг інцидентів без значних фінансових витрат на ліцензування, є актуальним і затребуваним науково-практичним завданням.

Мета роботи: розроблення проєкту ешелонованої системи захисту мережної інфраструктури на основі open-source рішень із підсистемою моніторингу та виявлення вторгнень.

Об'єкт дослідження: процеси забезпечення інформаційної безпеки в мережевій інфраструктурі, що обробляє конфіденційні дані.

Предмет дослідження: методи та засоби побудови ешелонованої системи захисту мережі з підсистемою виявлення вторгнень.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Аналіз сучасного стану інформаційної безпеки мережевих інфраструктур

1.1.1. Класифікація загроз інформаційній безпеці мережевих систем

Для побудови ефективної системи захисту необхідно чітко розуміти, які саме загрози можуть впливати на мережеву інфраструктуру. Я використав підхід, що базується на тріаді CIA (конфіденційність, цілісність, доступність), доповнений класифікацією NIST SP 800-30 [1]. За джерелом походження загрози поділяються на:

- Зовнішні – атаки з мережі Інтернет (спроби несанкціонованого доступу, DDoS, сканування портів, експлуатація вразливостей веб-застосунків).
- Внутрішні – дії інсайдерів (навмисне витікання даних, помилки адміністрування, недбале поводження з обліковими записами).
- Техногенні та природні – збої обладнання, перебої електроживлення, стихійні лиха.

За механізмом реалізації атак найчастіше зустрічаються:

1. Шкідливе програмне забезпечення (віруси, трояни, програми-вимагачі), що розповсюджується через фішинг або вразливості.
2. Атаки на веб-додатки: SQL-ін'єкції, міжсайтовий скриптинг (XSS), підробка міжсайтових запитів (CSRF), включені в перелік OWASP Top 10 [2].
3. Атаки на протоколи автентифікації: перехоплення облікових даних, brute-force, підбір паролів.
4. Атаки «людина посередині» (MitM) під час передавання конфіденційної інформації відкритими каналами.
5. Експлуатація вразливостей операційних систем, СКБД та мережевого обладнання.

Особливу небезпеку для систем, що працюють із конфіденційними даними, становлять цілеспрямовані атаки (APT), коли зловмисник намагається закріпитися

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

в мережі, підвищити привілеї та непомітно вилучити інформацію [3]. Саме тому захист має бути багаторівневим і охоплювати всі стадії kill chain [4].

1.1.2. Огляд статистики кіберінцидентів та типових векторів атак на конфіденційні дані

Щоб обґрунтувати актуальність проєкту, було виконано аналіз даних звіту Verizon Data Breach Investigations Report (DBIR) за 2023 рік [5] та щорічного звіту ENISA Threat Landscape [6]. Згідно з ними понад 50% інцидентів пов'язані з використанням викрадених облікових даних. Фішинг залишається основним вектором початкового проникнення (36% випадків). Ransomware-атаки зросли на 13% порівняно з попереднім роком, причому зловмисники часто шифрують не тільки файли, а й бази даних. Злам веб-застосунків є причиною близько 25% витоків даних. Інциденти, пов'язані з помилками конфігурації хмарних сервісів і баз даних, становлять 15% від загальної кількості.

Для мереж, що обробляють конфіденційні дані, критичними є такі типові вектори:

- Пряме підключення до СКБД через незахищені порти (наприклад, PostgreSQL на 5432/tcp без TLS та належного контролю доступу).
- Перехоплення слабо захищеного трафіку VPN (застарілі протоколи PPTP, слабкі шифри).
- Використання вразливостей у веб-інтерфейсах адміністрування мережевого обладнання.
- Недостатнє логування та моніторинг, що унеможливорює швидке виявлення вторгнень.

Ця статистика підтверджує необхідність впровадження комплексного захисту із обов'язковим шифруванням даних, сегментуванням мережі та підсистемою виявлення вторгнень.

1.1.3. Нормативно-правова база захисту інформації в Україні

Проектowana система повинна відповідати вимогам чинного законодавства України. Основним нормативним документом є Закон України «Про захист інформації в інформаційно-комунікаційних системах» № 80/94-ВР [7]. Він

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

визначає, що до інформації, яка є власністю держави, або до інформації з обмеженим доступом висуваються вимоги щодо забезпечення її цілісності, конфіденційності та доступності шляхом створення комплексної системи захисту інформації (КСЗІ). Для автоматизованих систем класу «3» або «2» застосовуються вимоги НД ТЗІ 2.5-010-03 [8] та НД ТЗІ 2.7-010-09 [9]. Хоча мій проєкт орієнтований на open-source рішення, він враховує ці норми в частині побудови ешелонованого захисту, аудиту дій, розмежування доступу та резервного копіювання.

Окрему увагу я приділив Регламенту GDPR (2016/679) [10], оскільки багато українських підприємств працюють із персональними даними громадян ЄС. GDPR вимагає впровадження технічних заходів для захисту персональних даних, зокрема псевдонімізації та шифрування, здатності забезпечити постійну конфіденційність, цілісність, доступність і стійкість систем обробки. У проєкті ці принципи реалізуються через застосування TLS, pgcrypto та суворого контролю доступу.

1.2. Концепція ешелонованого захисту (Defense-in-Depth)

1.2.1. Принципи побудови багаторівневих систем захисту

Концепція Defense-in-Depth, сформульована Агентством національної безпеки США (NSA) та закріплена в NIST SP 800-53 [11], передбачає створення кількох незалежних рубежів оборони, які зловмисник має подолати послідовно. Основні принципи: Багатошаровість – захист на рівні периметру, мережі, хосту, застосунку та даних. Взаємодоповнюваність – компоненти різних рівнів перекривають недоліки один одного. Захист у глибині – припущення, що периметр буде прорвано, тому внутрішні ресурси також захищені. Відмовостійкість та резервування – забезпечення доступності сервісів при відмовах. Безперервний моніторинг – виявлення аномалій і реагування в реальному часі.

Я спроектував систему так, щоб навіть у разі компрометації одного рівня наступний шар стримував атакуючого.

1.2.2. Рівні захисту: периметровий, мережевий, хостовий, прикладний, рівень даних

Керуючись моделлю Defense-in-Depth, я виділив такі рівні:

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

1. Периметровий – фаєрвол / маршрутизатор, який здійснює фільтрацію пакетів, NAT, VPN-термінацію, відкидає небажаний трафік на кордоні мережі.
2. Мережевий – система виявлення вторгнень (NIDS), що аналізує трафік усередині сегментів; VLAN для ізоляції DMZ, внутрішньої зони, зони управління та зони БД.
3. Хостовий – захист окремих серверів: NIDS, локальний фаєрвол (UFW), оновлення, мінімізація служб, контроль цілісності файлів (Wazuh), AppArmor.
4. Прикладний – безпечна конфігурація СКБД: TLS для з'єднань, розмежування ролей, аудит дій (pgAudit), шифрування на рівні застосунку (pgcrypto).

Така декомпозиція дозволяє чітко розподілити функції безпеки між компонентами системи.

1.2.3. Моделі загроз STRIDE та PASTA для мережевих систем

Для систематичного аналізу загроз я застосував моделювання за методикою STRIDE (Microsoft) [12]. Модель класифікує загрози за шістьма категоріями:

- Spoofing (підміна) – підміна IP-адреси, підробка VPN-клієнта.
- Tampering (спотворення) – модифікація пакетів, зміна даних у БД.
- Repudiation (відмова від дії) – відсутність логів дозволяє зловмиснику заперечувати факт атаки.
- Information Disclosure (розкриття інформації) – перехоплення трафіку, витік резервних копій.
- Denial of Service (відмова в обслуговуванні) – DDoS на фаєрвол або VPN, вичерпання ресурсів БД.
- Elevation of Privilege (підвищення привілеїв) – експлуатація вразливостей для отримання root-доступу.

PASTA (Process for Attack Simulation and Threat Analysis) [13] доповнює STRIDE етапами бізнес-аналізу та симуляції атак, однак у моєму проєкті я обмежився картуванням загроз за STRIDE, що буде деталізовано в розділі 2.7.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

1.3. Аналіз існуючих рішень та технологій

1.3.1. Порівняльний аналіз програмних маршрутизаторів і фаєрволів (pfSense, OPNsense, VyOS)

Для периметрового захисту я розглянув три популярні open-source платформи:

1) pfSense [14] – базується на FreeBSD, має веб-інтерфейс, підтримує фаєрвол (pf), шейпінг трафіку, VLAN, VPN (IPsec, OpenVPN, WireGuard), інтегрується із Suricata та pfBlockerNG.

2) OPNsense [15] – форк pfSense з акцентом на сучасний інтерфейс та більш часті оновлення, підтримує аналогічний функціонал.

3) VyOS [16] – Linux-based маршрутизатор із CLI-конфігурацією, орієнтований на мережесімейних інженерів, має менше графічних інструментів для IDS/IPS.

Таблиця 1.1

Порівняльна таблиця характеристик

Характеристика	pfSense	OPNsense	VyOS
Ліцензія	Apache 2.0	BSD 2-Clause	GPL v2
Веб-інтерфейс	Так, зручний	Так, сучасний	Ні (CLI)
Підтримка WireGuard	Так (плагін)	Так (вбудовано)	Так (v1.4+)
Інтеграція IDS/IPS	Так (Suricata)	Так (Suricata, ET Pro)	Ні
Підтримка VLAN	Так	Так	Так
Продуктивність (в базовій конф.)	Висока	Висока	Дуже висока
Спільнота/документація	Велика	Зростаюча	Середня

Зважаючи на широку підтримку пакетів IDS, простоту конфігурації WireGuard через GUI, сучасний інтерфейс та те що продукт є новішим, я обрав OPNSense як основний периметровий компонент.

1.3.2. Огляд систем виявлення вторгнень (Suricata, Snort, Zeek)

Я проаналізував три найпоширеніші NIDS.

Snort [17] – класична сигнатурна система, але має один потік обробки, що обмежує продуктивність на багатоядерних системах.

Suricata [18] – мультипоточна, підтримує апаратне прискорення, аналіз на рівні застосунків (HTTP, TLS, DNS), вивід у форматі EVE JSON, може працювати в режимі IPS inline. Використовує набори правил Emerging Threats (ET Open).

Zeek (колишній Bro) [19] – більше платформа для аналізу мережної безпеки, орієнтована на виявлення аномалій і створення метаданих, а не на сигнатурну детекцію.

Для мого проекту критичною є можливість сигнатурного виявлення відомих атак (SQL-ін'єкції, сканування, експлойти) та простий експорт логів у централізовану систему. Suricata має найкращий баланс продуктивності та функціональності, тому я вибрав її як NIDS.

1.3.3. Аналіз HIDS-рішень та SIEM-систем з відкритим вихідним кодом (Wazuh, OSSEC, ELK Stack)

Для хостового виявлення та централізованого моніторингу я вивчив:

OSSEC [20] – зрілий HIDS, виконує аналіз логів, перевірку цілісності файлів, виявлення rootkit'ів, має активну відповідь. Однак його розвиток сповільнився.

Wazuh [21] – форк OSSEC, який активно розвивається. Додає підтримку XDR, має модулі для аудиту відповідності (GDPR, PCI-DSS), інтегрується із стеками Elastic (Elasticsearch, Kibana), спрощує масштабування та кореляцію подій.

ELK Stack (Elasticsearch, Logstash, Kibana) [22] – потужна платформа для збору, обробки та візуалізації даних, але потребує окремого налаштування правил кореляції та сповіщень, що ускладнює розгортання «з коробки».

Wazuh поєднує можливості HIDS і базового SIEM із готовими правилами відповідності та агентами для Linux. У поєднанні з Elasticsearch та Kibana він

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				16
Змн.	Арк.	№ докум.	Підпис	Дата		

дозволяє отримати повноцінну систему моніторингу, тому саме його я включив до проєкту.

1.3.4. Засоби захисту СКБД PostgreSQL (TLS, pgcrypto, pgAudit)

База даних є основним сховищем конфіденційної інформації, тому я дослідив вбудовані механізми захисту PostgreSQL [23]:

- TLS – забезпечує шифрування трафіку між клієнтами та сервером, запобігає перехопленню даних. Налаштовується через параметри `ssl=on` та сертифікати.
- `pgcrypto` – модуль, який дозволяє шифрувати окремі стовпці таблиць алгоритмами AES, використовуючи симетричні або асиметричні ключі. Це захищає дані навіть у разі витоку дампу БД.
- `pgAudit` – розширення для детального аудиту дій, фіксує SELECT, INSERT, UPDATE, DELETE, DDL-операції, спроби підключення. Логи можна експортувати в Wazuh.

Ці засоби утворюють багаторівневий захист даних, який відповідає вимогам GDPR щодо шифрування та псевдонімізації.

1.3.5. VPN-протоколи для віддаленого доступу (OpenVPN, WireGuard, IPsec)

Для безпечного віддаленого доступу я порівняв три протоколи:

OpenVPN [24] – зрілий, працює в `user space`, підтримує безліч шифрів і аутентифікацію через сертифікати. Налаштування складніше, продуктивність обмежена через роботу через TUN/TAP.

WireGuard [25] – сучасний протокол на основі ядра Linux/FreeBSD, мінімальна кодова база (≈ 4000 рядків), висока продуктивність, прості конфігураційні файли. Підтримує лише сучасні алгоритми (Curve25519, ChaCha20, BLAKE2s).

IPsec (IKEv2) [26] – стандартизований, підтримується більшістю ОС, але конфігурація громіздка, складніше налагодити з NAT.

Я обрав WireGuard через його швидкість, простоту та надійність. У OPNSense підтримується через плагін, що дозволяє легко керувати підключеннями.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		17

1.4. Постановка задачі проєктування

1.4.1. Формулювання вимог до системи захисту

На основі проведеного аналізу я визначив ключові вимоги до майбутньої системи. Передусім вона повинна гарантувати конфіденційність і цілісність даних під час передавання та зберігання, для чого застосовуватимуться наскрізне шифрування та контроль доступу на всіх рівнях.

Мережева інфраструктура має бути логічно сегментована на зони різного рівня довіри з обов'язковою ізоляцією сервера баз даних. Для виявлення та блокування мережевих атак, таких як сканування портів або SQL-ін'єкції, потрібна система виявлення вторгнень (NIDS), що працює на основі актуальних правил. Одночасно хостові агенти (HIDS) повинні контролювати цілісність файлів та системних журналів, фіксуючи підозрілі дії на серверах. Події від усіх компонентів мають збиратися в централізоване сховище з можливістю кореляції та автоматичного сповіщення про інциденти. Віддалений доступ до мережі надаватиметься виключно через захищений VPN-тунель, а резервне копіювання бази даних виконуватиметься автоматично з обов'язковим шифруванням копій. Усі програмні засоби системи повинні належати до вільного програмного забезпечення, не потребувати платних ліцензій та відповідати політиці використання лише open-source компонентів.

1.4.2. Функціональні та нефункціональні вимоги

Функціональні вимоги деталізують поведінку окремих компонентів. Маршрутизатор-фаєрвол має виконувати фільтрацію пакетів на основі IP-адрес, портів і протоколів (ACL), підтримувати NAT, логічну сегментацію через VLAN і термінацію WireGuard-тунелів. Мережева IDS повинна реєструвати підозрілий трафік, аналізуючи його згідно з набором правил ET Open, і за необхідності блокувати загрози в режимі inline. Хостова система HIDS на базі Wazuh зобов'язана відстежувати системні логи, перевіряти цілісність файлів, виявляти brute-force атаки та автоматично блокувати IP-адреси порушників за допомогою механізму активної відповіді. Сервер PostgreSQL працюватиме виключно із з'єднаннями TLS 1.3, з увімкненими розширеннями pgAudit (для повного аудиту дій) та pgcrypto (для

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

шифрування окремих стовпців). Резервне копіювання виконуватиметься щодоби за розкладом, а резервні копії зберігатимуться в зашифрованому вигляді. Централізована консоль Kibana має надавати інтерактивні дашборди для візуалізації поточного стану безпеки.

Нефункціональні вимоги охоплюють продуктивність, сумісність, надійність і безпеку. Так, пропускна здатність фаєрвола повинна бути не нижче 1 Гбіт/с, а втрата пакетів не перевищувати 0,1 %. Усі серверні компоненти розгортаються на Ubuntu Server 26.04 LTS, мережеві – на платформі OPNSense 26.1.6. Час відновлення після збою не має перевищувати 30 хвилин. Архітектура системи повинна передбачати можливість додавання нових сенсорів IDS і агентів Wazuh без зміни ядра. Для криптографічних перетворень дозволяється застосовувати лише сучасні алгоритми: AES-256, Curve25519 та SHA-256. Керування системою спрощується завдяки наявності веб-інтерфейсів (OPNSense, Kibana) і механізму централізованого оновлення правил детекції.

1.4.3. Обмеження проєкту (використання виключно open-source та безкоштовних ресурсів)

Проєкт виконується в лабораторному середовищі з низкою обмежень. Усі програмні компоненти розповсюджуються за вільними ліцензіями (MIT, Apache 2.0, GPL, BSD). Апаратна платформа повністю віртуалізована на базі VirtualBox, фізичні пристрої не залучаються. Мережа працює лише з протоколом IPv4, а кількість одночасних VPN-клієнтів обмежена п'ятьма. Резервне копіювання здійснюється на локальний віртуальний диск, окремий від основного сховища. Ці обмеження дають змогу чітко окреслити границі проєкту та зосередитися на реалізації ключових механізмів захисту.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновки до розділу 1

У першому розділі я виконав всебічний аналіз предметної області. Проведений огляд загроз та статистики кіберінцидентів підтвердив, що мережеві системи, які обробляють конфіденційні дані, потребують багаторівневого захисту. Розглянуто нормативну базу України та GDPR, що визначає вимоги до шифрування, аудиту та контролю доступу. Вивчено принципи Defense-in-Depth та моделі загроз STRIDE/PASTA, які стануть теоретичним фундаментом для проєктування. На основі порівняльного аналізу було обрано конкретні технології: OPNSense як периметровий фаєрвол, Suricata як NIDS, Wazuh у зв'язці з Elasticsearch/Kibana для HIDS та SIEM, PostgreSQL із засобами TLS, pgcrypto, pgAudit для захисту даних, а також WireGuard для VPN. Сформульовано функціональні та нефункціональні вимоги, визначено критерії оцінювання ефективності та обмеження проєкту. Отримані результати створюють підґрунтя для переходу до проєктування архітектури системи в розділі 2.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		20

РОЗДІЛ 2. ПРОЄКТУВАННЯ СИСТЕМИ ЕШЕЛОНОВАНОГО ЗАХИСТУ

2.1. Розроблення архітектури системи захисту

2.1.1. Загальна структурна схема мережної інфраструктури

У межах кваліфікаційної роботи я розробив архітектуру мережі, головна ідея якої полягає у створенні кількох послідовних рубежів оборони. Замість того, щоб покладатися на один захисний пристрій, я спроектував систему, де кожен компонент доповнює інший. На вході в мережу встановлюється головний шлюз безпеки, який контролює абсолютно весь трафік, що надходить із зовнішнього світу або переміщується між внутрішніми серверами. Така схема дозволяє повністю ізолювати важливі дані від прямого доступу з інтернету та забезпечує єдину точку контролю для моніторингу всіх мережевих процесів.

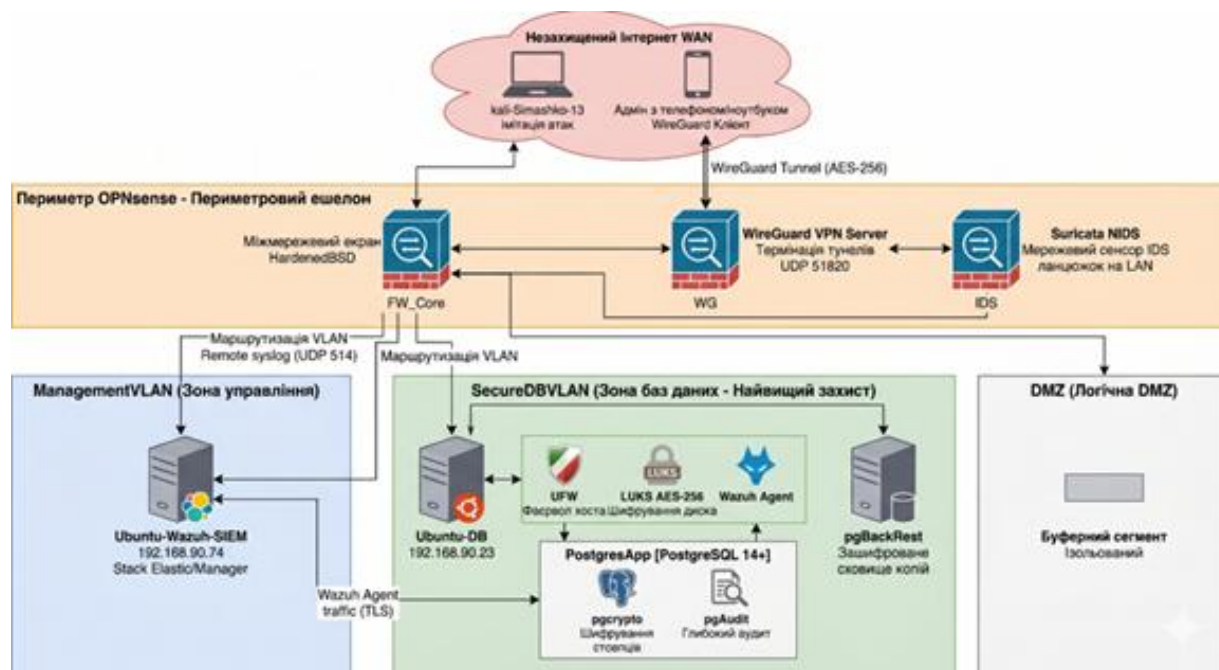


Рисунок 2.1 – Схема мережі

Вся структура побудована так, щоб унеможливити ситуацію, коли зломисник, зламавши один комп'ютер, отримує доступ до всієї мережі. Кожен сервер і кожна служба працюють у своєму власному закритому середовищі. Зовнішні користувачі чи адміністратори взаємодіють із системою виключно через суворо контрольовані канали, що мінімізує ризики випадкового або навмисного витоку конфіденційної інформації.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				21
Змн.	Арк.	№ докум.	Підпис	Дата		

2.1.2. Логічне сегментування мережі: схема VLAN

Для надійної ізоляції різних частин інфраструктури я розділив мережу на кілька окремих віртуальних сегментів за допомогою технології VLAN. Першим важливим бар'єром є демілітаризована зона, яка виконує роль проміжного фільтра. Саме тут підключаються віддалені користувачі, і саме тут відбувається первинна перевірка їхніх запитів, що дозволяє не пускати потенційно небезпечний трафік глибше в систему.

Окремо від інших сегментів я виділив зону управління, яка призначена виключно для технічного обслуговування серверів та збору службових логів. Найбільш захищеним островом у цій структурі є зона баз даних, де функціонує сервер PostgreSQL. Цей сегмент повністю відрізаний від зовнішнього світу, а зв'язок із ним дозволений лише для обмеженого кола перевірених внутрішніх пристроїв, що робить пряму атаку на сховище даних з інтернету технічно неможливою.

2.1.3. Розподіл компонентів захисту за рівнями ешелонованості

Спроектована мною система безпеки нагадує кілька захисних шарів, де кожен рівень має власні інструменти захисту. На самому периметрі інфраструктури працює міжмережевий екран, який відсікає очевидно підозрілі запити. Наступний мережевий рівень контролюється системою виявлення вторгнень, яка аналізує вміст пакетів і шукає сліди хакерської активності безпосередньо в потоці даних.

Глибший контур захисту розгорнутий на рівні самих серверів, де працюють локальні фаєрволи операційної системи та спеціальні програми-агенти, які стежать за цілісністю файлів. Нарешті, останній ешелон захисту реалізовано безпосередньо всередині бази даних за допомогою інструментів шифрування та суворого обмеження прав користувачів. Завдяки такій побудові, навіть якщо нападник подолає зовнішній фаєрвол, він все одно натрапить на системи виявлення атак, шифрування диска та захист самої бази даних.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		22

2.1.4. Схема інформаційних потоків та точок контролю

Під час проектування інформаційних потоків я заклав принцип, за якого кожен крок даних у мережі фіксується та перевіряється. Головними точками контролю виступають інтерфейси маршрутизатора та аналітичні модулі. Коли віддалений адміністратор надсилає запит, цей потік спочатку перевіряється на рівні VPN-сервера, потім проходить крізь сито мережевого аналізатора і лише після цього спрямовується до цільового сервера.

Внутрішні потоки даних між серверами також суворо обмежені, тому комп'ютери з різних зон не можуть вільно спілкуватися між собою без дозволу фаєрвола. Водночас у системі створено окремий закритий потік для службової інформації, по якому всі сервери надсилають свої звіти та логи на центральний вузол моніторингу. Цей потік захищений від втручання, що дозволяє бачити реальну картину того, що відбувається в мережі, навіть під час активної атаки.

2.2. Проектування підсистеми периметрового захисту

2.2.1. Обґрунтування вибору OPNsense як основи маршрутизатора-фаєрвола

Для створення надійного зовнішнього бар'єра я обрав програмний комплекс OPNsense, який є сучасним рішенням із відкритим вихідним кодом. Його головна перевага полягає в тому, що він побудований на базі безпечної операційної системи HardenedBSD, яка початково розроблялася з підвищеними вимогами до захисту від уразливостей. OPNsense поєднує в собі функції потужного міжмережевого екрана, маршрутизатора та платформи для підключення додаткових модулів безпеки.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		23

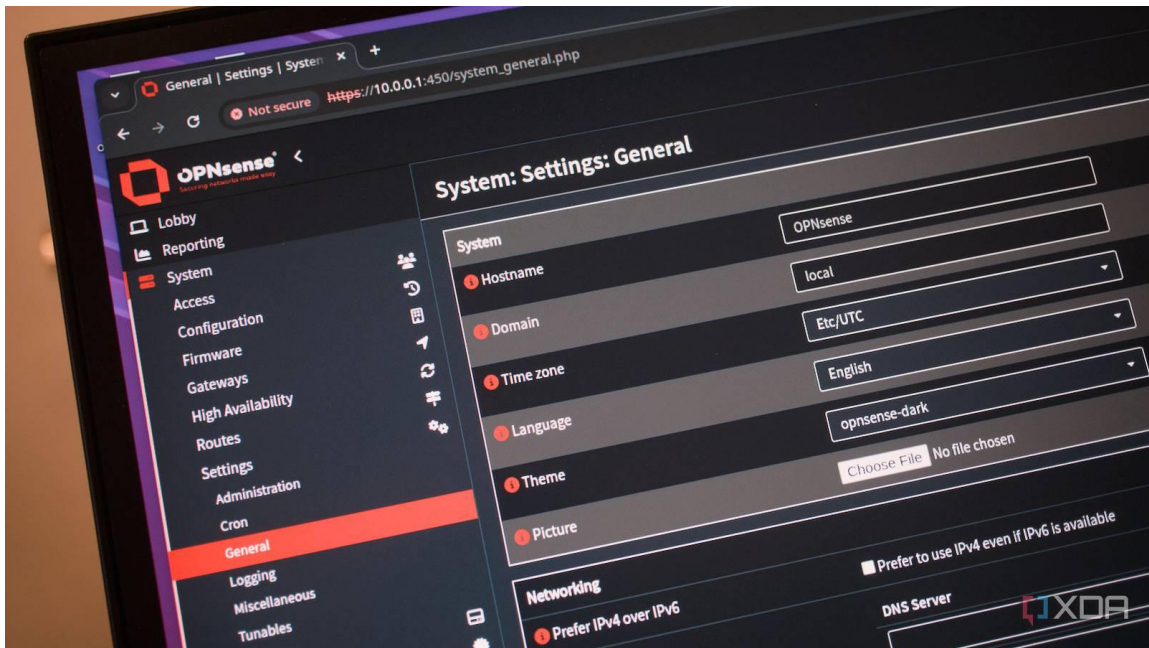


Рисунок 2.2 – Веб-інтерфейс OPNsense

Ця система є дуже стабільною, регулярно отримує оновлення безпеки та дозволяє гнучко налаштовувати правила фільтрації через зручний інтерфейс, що значно знижує ризик помилок через людський фактор. Окрім того, OPNsense не потребує купувати дорогі комерційні ліцензії, але водночас забезпечує швидкість та якість обробки трафіку на рівні корпоративних стандартів.

2.2.2. Проєктування правил фільтрації трафіку (ACL)

Розробляючи списки контролю доступу для фаєрвола, я керувався золотим правилом безпеки, згідно з яким дозволено лише те, що чітко прописано, а все інше – заборонено за замовчуванням. На зовнішньому інтерфейсі, який дивиться в інтернет, буде повністю заблоковано будь-які вхідні запити, окрім одного конкретного порту, необхідного для роботи зашифрованого VPN-тунелю.

Для внутрішніх віртуальних мереж правила фільтрації будуть налаштовані так, щоб максимально обмежити свободу переміщення трафіку. Сервер бази даних отримає дозвіл приймати підключення на свій робочий порт лише з однієї конкретної адреси комп'ютера адміністратора, а будь-які спроби з'єднатися з інших пристроїв фаєрвол просто ігнорує. Крім того, я обмежив право самих серверів самотійно виходити в інтернет, щоб шкідлива програма у разі зараження не могла зв'язатися зі своїм розробником.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		24

2.2.3. Архітектура VPN-з'єднань на базі WireGuard

Для того щоб адміністратор міг безпечно керувати системою з дому чи іншого віддаленого місця, я спроектував підсистему VPN на основі сучасного протоколу WireGuard. Цей протокол використовує новітні та дуже стійкі алгоритми шифрування, які практично неможливо зламати сучасними методами перехоплення даних. WireGuard працює значно швидше за старі аналоги і створює мінімальне навантаження на процесор.

Принцип його роботи в моїй архітектурі полягає в тому, що сервер і клієнт заздалегідь обмінюються спеціальними цифровими ключами. Коли віддалений пристрій підключається до шлюзу OPNsense, йому автоматично видається постійна внутрішня адреса, і весь подальший трафік надійно шифрується. Це дозволяє фаєрволу чітко розуміти, який саме адміністратор зараз перебуває в мережі, і відкривати для нього лише ті ресурси, які потрібні для роботи.

2.2.4. Налаштування NAT та політик переадресації портів

Оскільки тестовий стенд розгорнуто всередині віртуального середовища, виникла технічна проблема подвійного бар'єра, коли віртуальні машини захищені за фаєрволом самого комп'ютера. Щоб вирішити це завдання, я спроектував схему трансляції адрес та переадресації портів, яка дозволяє зовнішнім пакетам знаходити правильну дорогу до нашого шлюзу OPNsense.

Головна увага була приділена порту WireGuard, для якого налаштовано наскрізне пересилання пакетів. Важливою особливістю обраного протоколу є те, що він працює абсолютно непомітно: якщо на порт приходить запит без правильного цифрового ключа, сервер просто мовчить і ніяк не реагує. Це захищає систему від автоматичних сканерів хакерів, оскільки для зовнішнього спостерігача цей порт виглядає повністю закритим.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		25

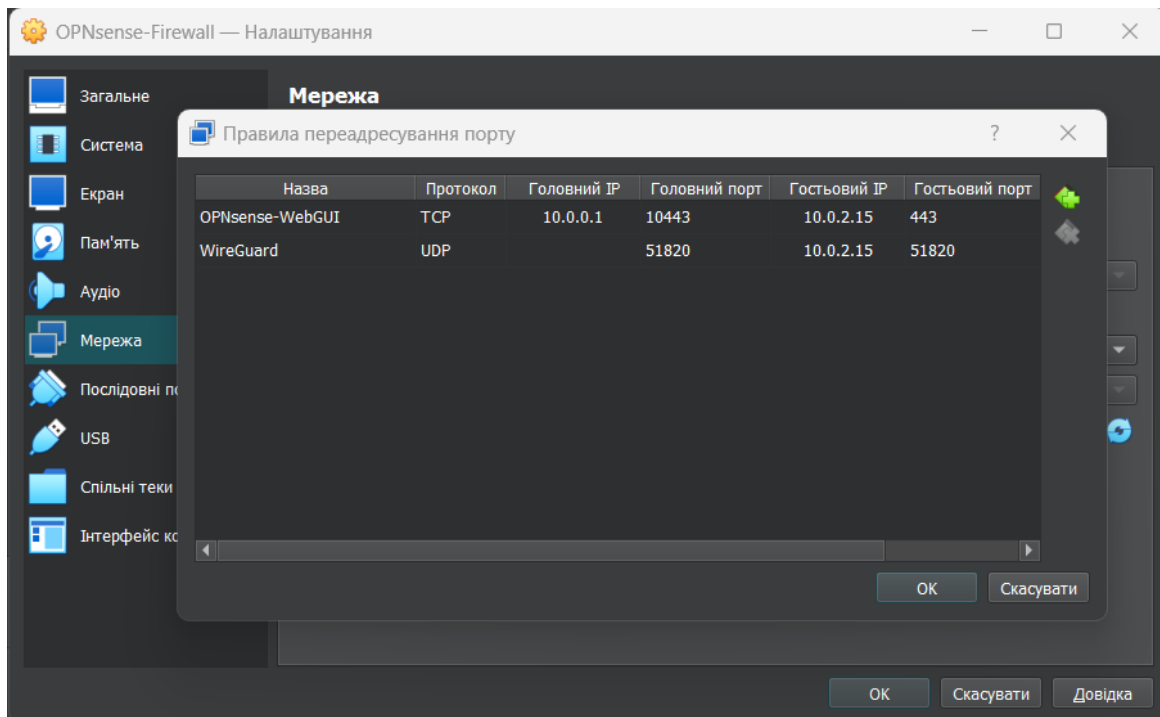


Рисунок 2.3 – Переадресування портів в VirtualBox

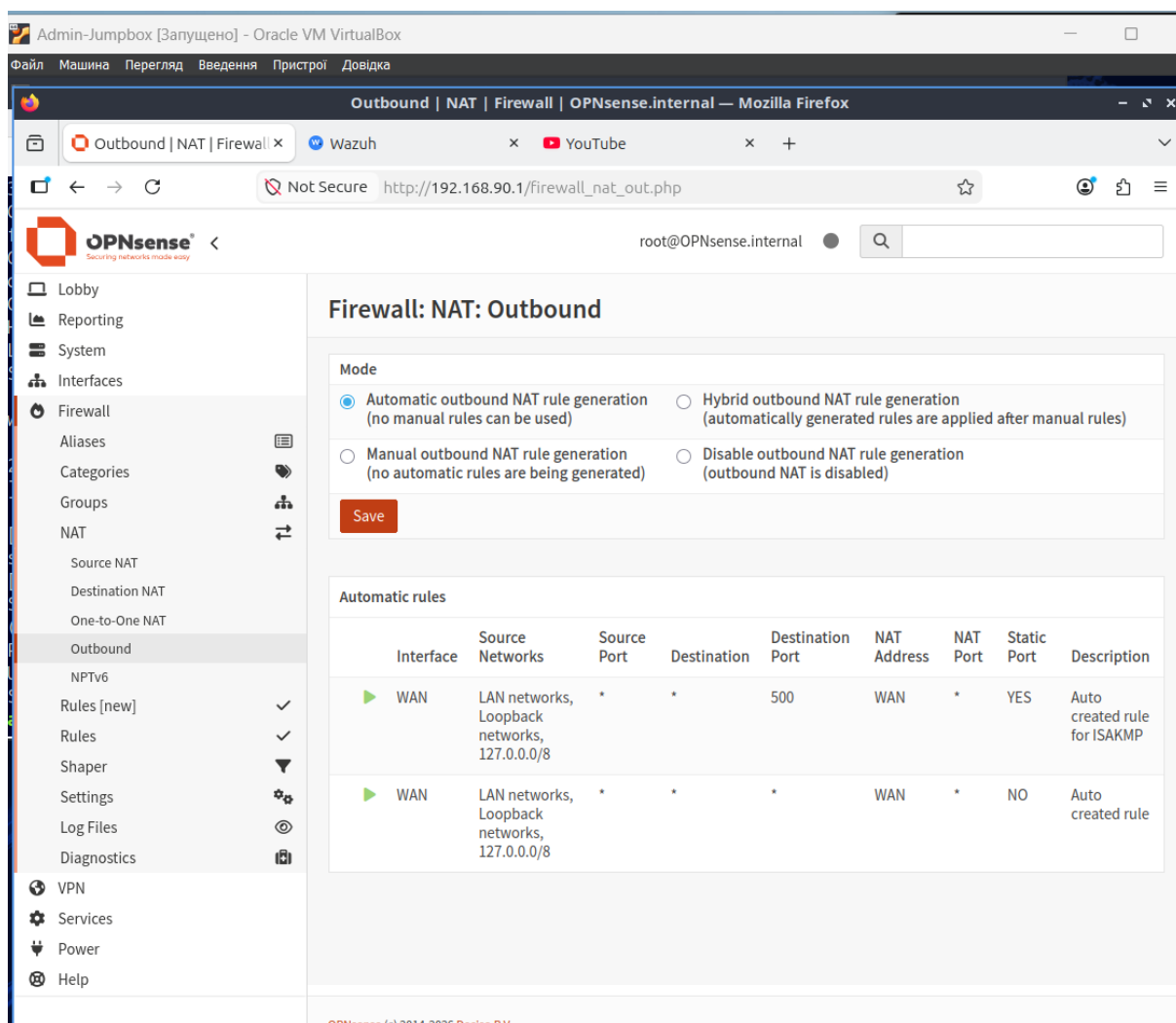


Рисунок 2.4 – Налаштований NAT на OPNsense маршрутизаторі

2.3. Проектування підсистеми виявлення вторгнень

2.3.1. Обґрунтування вибору Suricata як NIDS

Для виявлення підозрілої активності всередині мережеских потоків я обрав систему Suricata, яка виконує роль мережевого датчика вторгнень. На відміну від старіших систем, Suricata вміє працювати з багатьма ядрами процесора одночасно, що дозволяє їй аналізувати великі обсяги інформації без затримок і втрати пакетів. Це вкрай важливо для сучасних мереж, де швидкість передачі даних є дуже високою.



Рисунок 2.5 – Suricata

Крім того, ця система здатна глибоко аналізувати вміст пакетів на рівні програм, розуміючи структуру вебзапитів чи поштових повідомлень. Вона вміє зберігати всі звіти у зручному та чіткому форматі JSON, що дозволяє легко передавати інформацію про знайдені загрози до єдиного центру моніторингу для автоматичного аналізу.

2.3.2. Розміщення сенсорів IDS у топології мережі

Щоб система виявлення вторгнень працювала ефективно, її датчик потрібно встановити в правильному місці, де проходить найбільше трафіку. У своєму проєкті я інтегрував сенсор Suricata безпосередньо всередину шлюзу OPNsense і підключив його до внутрішнього інтерфейсу локальної мережі.

Завдяки такому рішенню Suricata отримує можливість бачити абсолютно всі пакети, які рухаються між віртуальними мережами, а також трафік, який приходить від віддалених адміністраторів через VPN. Датчик працює в режимі копіювання

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				27
Змн.	Арк.	№ докум.	Підпис	Дата		

пакетів, тобто він просто "слухає" мережу і аналізує копії даних, взагалі не уповільнюючи швидкість роботи основних серверів та користувачів.

2.3.3. Проєктування набору правил детекції (Emerging Threats, ET Open)

Щоб Suricata знала, які саме мережеві пакети є небезпечними, їй потрібна база сигнатур, тобто цифрових портретів відомих атак. Я спроектував оптимальний набір таких правил на основі відкритої бази Emerging Threats Open, виключивши зайві категорії, щоб не перевантажувати пам'ять віртуальної машини.

До конфігурації увійшли правила для розпізнавання хакерського сканування мережі, сигнатури для пошуку спроб зламати операційну систему, а також шаблони поведінки відомих комп'ютерних вірусів. Особливий акцент було зроблено на виявленні специфічних атак на бази даних, таких як SQL-ін'єкції, що дозволяє системі миттєво піднімати тривогу, як тільки хтось намагається надіслати підозрілий запит до нашого сховища конфіденційних даних.

2.3.4. Інтеграція Suricata з системою централізованого логування

Саме по собі виявлення атаки не має сенсу, якщо адміністратор про це не дізнається або якщо лог події буде знищено зловмисником. Тому в проєкті я реалізував схему, за якої Suricata миттєво відправляє всі повідомлення про небезпеку на Wazuh через захищений мережевий протокол Syslog.

Це гарантує безпеку самих журналів аудиту, адже навіть якщо хакер зможе якимось чином отримати повний контроль над головним шлюзом мережі, він фізично не зможе стерти записи про свій злам, оскільки вони вже збережені на іншому, ізольованому сервері, до якого у нього немає доступу.

2.4. Проєктування підсистеми захисту сервера та СКБД

2.4.1. Архітектура захисту Ubuntu Server 26.04 LTS

Для безпосереднього розгортання бази даних я обрав операційну систему Ubuntu Server 26.04 LTS і спроектував її захист за принципом максимального зменшення зон ризику. Першим кроком стало повне видалення або вимкнення всіх стандартних програм і служб, які не потрібні для роботи самої бази даних, щоб у хакерів не було за що зачепитися.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		28

Для контролю мережі на самому сервері я налаштував вбудований локальний фаєрвол UFW, який став ще одним ешелonom оборони після OPNsense. Цей фаєрвол налаштований так, що він закриває всі входи до сервера, залишаючи відкритими лише порт для моніторингу стану системи та порт 5432, через який працює база даних PostgreSQL, причому пускає туди пакети тільки з дозволених адрес.

2.4.2. Модель розмежування доступу

Модель безпеки сервера повністю виключає можливість входу в систему з правами головного суперкористувача root через мережу, оскільки саме цей обліковий запис хакери намагаються зламати в першу чергу. Замість цього створено звичайного обмеженого користувача, а право виконувати технічні команди надається йому лише через систему індивідуальних дозволів sudo-політик.

Вхід на сервер по протоколу SSH налаштований виключно за допомогою спеціальних цифрових ключів, а звичайні паролі повністю вимкнені, що робить марними будь-які спроби зломисників підібрати пароль автоматом. Додатково для захисту критичних процесів операційної системи застосовано систему AppArmor, яка жорстко обмежує кожен програму в межах її робочих файлів, не дозволяючи їй втручатися в роботу ядра системи у разі збою.

2.4.3. Проектування захисту PostgreSQL: TLS-з'єднання, pg_hba.conf, ролі та привілеї

Захист даних під час їх руху мережею реалізовано через обов'язкове шифрування всіх підключень до PostgreSQL за допомогою сучасного криптографічного протоколу TLS 1.3. Це повністю захищає інформацію від перехоплення всередині локальної мережі. У спеціальному файлі налаштувань доступу розроблено сувору таблицю правил, яка дозволяє з'єднання з базою лише для чітко визначених IP-адрес із обов'язковим використанням стійкого шифрування паролів.

Права всередині самої бази даних спроектовані так, щоб звичайні програми мали доступ лише до тих таблиць, які їм потрібні для щоденної роботи, і не могли

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				29
Змн.	Арк.	№ докум.	Підпис	Дата		

жодним чином змінити системні налаштування чи переглянути чужі конфіденційні записи.

2.4.4. Шифрування даних at-rest та at-application-level

Даний рівень захисту працює безпосередньо всередині самої бази даних за допомогою вбудованого модуля pgcrypto. Найбільш важливі поля таблиць, такі як персональні дані клієнтів, шифруються ще на етапі запису спеціальними ключами додатка, тому навіть адміністратор сервера з повними правами доступу до файлів не зможе прочитати вміст цих полів без криптографічного ключа.

```
admin1@ubuntu-db:~$ sudo -u postgres psql -d secure_data
psql (18.3 (Ubuntu 18.3-1))
Type "help" for help.

secure_data=# SELECT * FROM clients_confidential;
 id | full_name | secret_data
-----+-----+-----
 1 | Simashko Ivan | \xc30d040703020d2db8411c36d6d66cd25401ae8f04d0e1632d4cb560fbb9f45f3984883a6685230079721e5a233ef09fab64a2bd1b5d5502443dc8fa15ff78c6a24d39
2c6992033e66e9810ba73a783bd66d1b63b833bcd3f76e43dcd4468ccdadbd3d5a32
 2 | Podro Homenk | \xc30d04070302506b6a4903a37b617dd25a016bde83ed559eb9706c41e2be1eb55379b032f2e85b84e9e0374ce4d45fee04e06ca20d3d504d7d68271219ac0f78bda17b
c887eb7a2d53a66ac413c159fc5b62d4c3c888b6e7e2b9bd61a3a1c6ff56db03152325dade941c01
 3 | TheyKilledKenny | \xc30d0407030243751f7f9f76d45764d25e019a437661f35be0d18426f3b8d154c566fc4893e0da5581c59a154cffe46f4a20bca0c9899512cedb470b8ab28553a4ab87
e4fcd489076f06d16ecaddb73eedab997f97c5c561e6e06d1022b01185e9cc5da9e4890362ce2108734fdea
(3 rows)
```

Рисунок 2.6 – Зашифрована інформація в бд

2.4.5. Підсистема аудиту БД на основі pgAudit

Оскільки стандартні звіти PostgreSQL показують лише загальну інформацію про роботу системи, я інтегрував у проєкт спеціалізоване розширення pgAudit, яке забезпечує повноцінний безпековий аудит. Цей інструмент дозволяє детально записувати кожну дію користувачів чи програм всередині бази даних.

Підсистема налаштована так, що вона фіксує будь-які спроби прочитати конфіденційні таблиці, змінити структуру бази або видалити інформацію. Усі ці записи миттєво зберігаються у вигляді структурованих повідомлень JSON, що дозволяє чітко бачити, хто саме, коли і який конкретно SQL-запит намагався виконати стосовно захищених даних.

2.5. Проєктування підсистеми моніторингу та реагування

2.5.1. Архітектура HIDS на базі Wazuh-агента

Для постійного контролю безпеки безпосередньо всередині серверів я спроєктував систему моніторингу на основі рішень Wazuh. На кожну операційну

систему встановлюється спеціальний легкий агент, який працює непомітно і практично не витрачає ресурси процесора чи оперативної пам'яті.

Цей агент безперервно стежить за внутрішнім станом хоста: він контролює цілісність важливих системних файлів, перевіряє запущені процеси, шукає сліди хакерських утиліт та аналізує локальні журнали подій. Уся зібрана інформація миттєво відправляється по зашифрованому каналу зв'язку на головний сервер аналітики для подальшої обробки.

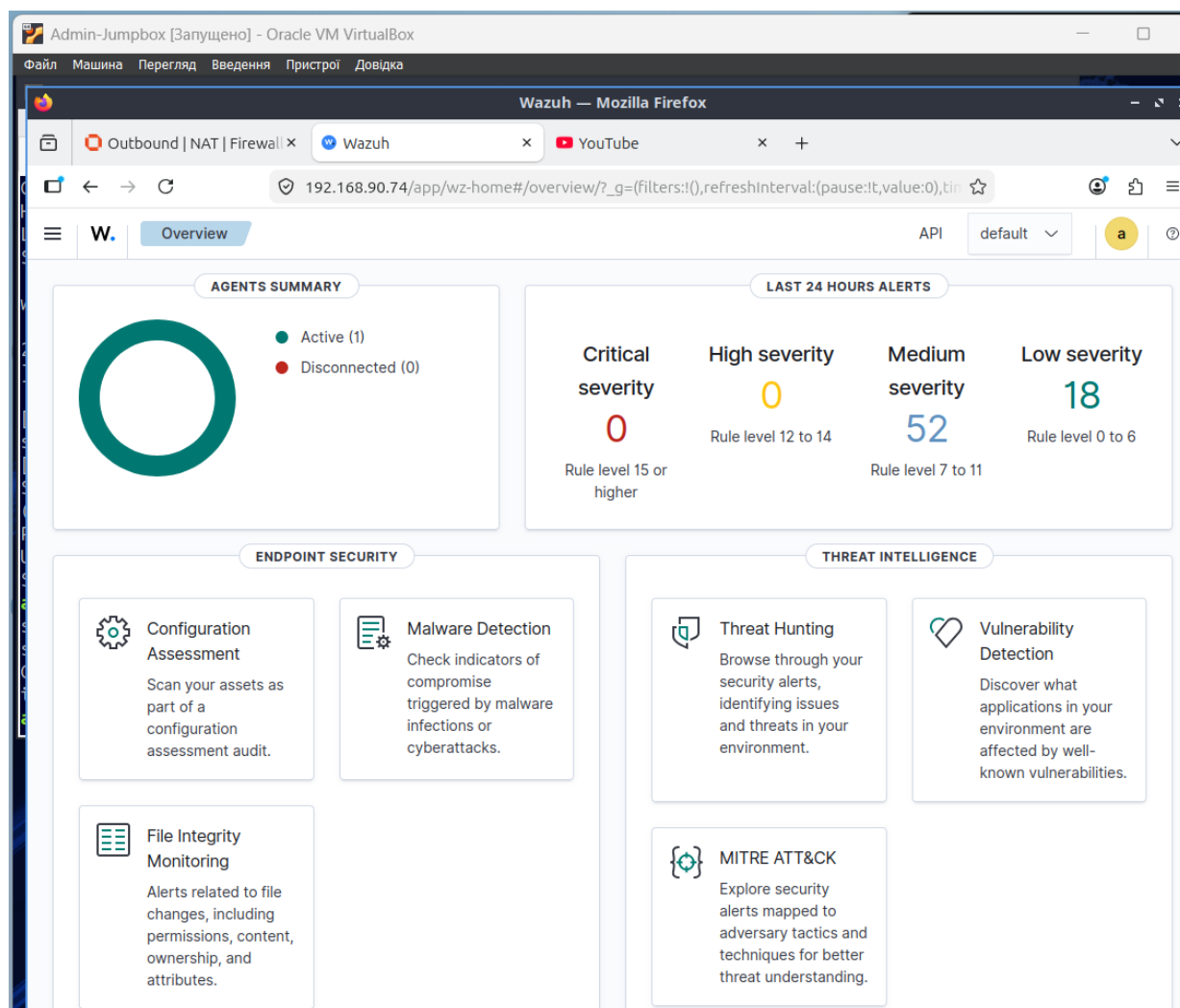


Рисунок 2.7 – Головна сторінка Wazuh

2.5.2. Централізований збір та кореляція подій

Центральною частиною всієї системи моніторингу є аналітичний сервер Wazuh Manager, який об'єднаний із базою даних Elasticsearch та графічною панеллю Kibana. Головний сервер приймає величезні потоки звітів від усіх пристроїв у мережі, сортує їх та розкладає по полицях у швидкісній базі даних. Це

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				31
Змн.	Арк.	№ докум.	Підпис	Дата		

дозволяє адміністратору миттєво знаходити будь-яку подію чи переглядати історію інцидентів за минулі місяці.

Для зручного контролю за станом безпеки налаштовано графічні дашборди Kibana, де у вигляді наочних графіків та діаграм у реальному часі відображаються всі підозрілі події, що відбуваються в інфраструктурі.

2.5.3. Інтеграція логів OPNsense, Suricata, PostgreSQL та системних логів Ubuntu

Головна цінність створеної мною системи моніторингу полягає в тому, що вона збирає докупи абсолютно різні інформаційні потоки. Система одночасно приймає звіти про заблоковані пакети з фаєрвола OPNsense, попередження про мережеві атаки від Suricata, логи глибокого аудиту з бази даних PostgreSQL, а також записи про вхід користувачів з операційної системи Ubuntu.

Для реалізації цього, в конфігураційних файлах агентів прописуються шляхи до журналів усіх цих служб. Завдяки такому підходу сервер бачить повну картину подій і може пов'язати підозрілий клік на зовнішньому шлюзі із раптовою зміною файлів всередині сервера бази даних.

2.5.4. Проєктування правил кореляції та сповіщень про інциденти

Щоб система могла самостійно знаходити складні хакерські атаки без участі людини, я розробив набір спеціальних правил кореляції подій. Ці правила описують логічні ланцюжки між різними діями, які окремо можуть здаватися звичайними помилками, але разом свідчать про спробу зламу.

Наприклад, правило спрацює, якщо система зафіксує кілька невдалих спроб вгадати пароль до SSH, і одночасно з цим мережевий сенсор Suricata помітить підозрілий трафік з цієї ж адреси. Кожній такій події присвоюється рівень небезпеки, і якщо цей рівень є критичним, система негайно б'є на сполох та виводить попередження на екран адміністратора.

2.5.5. Автоматичне реагування: active response Wazuh

Для того щоб система могла захистити себе у перші ж секунди атаки, коли адміністратор ще не встиг відреагувати, я налаштував модуль автоматичної

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		32

протидії Active Response. Цей інструмент дозволяє аналітичному серверу давати миттєві команди локальним фаєрволам на ізоляцію порушників.

Зокрема, було спроектовано сценарій автоматичного блокування, який реагує на спроби підбору паролів. Як тільки система бачить атаку, вона миттєво витягує IP-адресу хакера з логів і додає її в чорний список фаєрвола на 60 секунд. Цього часу цілком достатньо, щоб зупинити автоматичний підбір паролів і повністю заблокувати зловмисника ще на підступах до критичних даних.

2.6. Проектування підсистеми резервного копіювання

2.6.1. Стратегія резервного копіювання БД (pgBackRest)

Щоб гарантувати збереження конфіденційної інформації у разі поломки обладнання або серйозного збою, я спроектував підсистему резервного копіювання на основі утиліти pgBackRest. Цей інструмент спеціально створений для надійної роботи з великими базами даних і вміє робити копії дуже швидко й економно.

На мою думку оптимальним графіком резервного копіювання буде що повний дублікат усієї бази даних створюється автоматично раз на тиждень, а кожного дня до нього додаються лише ті зміни, які відбулися за добу. Крім того, система безперервно копіює службові журнали операцій у реальному часі. Це дозволяє реалізувати технологію відновлення на будь-який момент часу, тобто адміністратор може повернути базу в той стан, у якому вона була, наприклад, рівно п'ять хвилин тому перед збоєм.

2.6.2. Шифрування та зберігання резервних копій

Оскільки резервні копії містять точні дублікати всієї конфіденційної інформації, вони самі по собі потребують серйозного захисту, щоб їх ніхто не міг викрасти. Тому в налаштуваннях pgBackRest я увімкнув обов'язкове шифрування всіх архівних файлів за допомогою стійкого алгоритму AES-256-CBC.

Усі зашифровані копії автоматично стискаються для економії місця на накопичувачі та зберігаються в окремій закритій директорії на ізольованому диску. Доступ до цих файлів мають лише самі системні процеси бекапу, що повністю захищає архіви від випадкового видалення або модифікації сторонніми особами чи шкідливими програмами.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		33

2.7. Модель загроз та оцінка ризиків спроектованої системи

2.7.1. Ідентифікація активів та загроз за моделлю STRIDE

Для ретельної перевірки надійності розробленої архітектури я застосував міжнародну модель аналізу загроз STRIDE, яка ділить усі потенційні небезпеки на шість чітких категорій. Перша категорія розглядає ризики підміни користувача, коли хакер намагається видати себе за легітимного сисадміна. Друга категорія аналізує загрози зміни даних, наприклад, спроби таємно підправити файли конфігурації фаєрвола чи записи в таблицях. Третя категорія вивчає ризики приховування дій, коли зловмисник намагається видалити системні логи, щоб замісти сліди.

Четверта категорія – це витік інформації, що є найголовнішим ризиком для нашого проєкту, оскільки вона описує загрозу крадіжки чи перехоплення конфіденційних даних клієнтів. П'ята категорія розглядає атаки типу "відмова в обслуговуванні", які намагаються завалити сервери смітєвими запитами і паралізувати їх роботу. Остання категорія аналізує ризики несанкціонованого підвищення прав, коли обмежений користувач намагається отримати повний контроль над операційною системою.

2.7.2. Карта покриття загроз компонентами системи захисту

На основі проведеного аналізу STRIDE я розробив таблицю відповідності, яка наочно показує, як саме створені мною рубежі оборони протидіють кожній з виявлених небезпек. Ризики підміни користувача повністю нівелюються використанням цифрових ключів замість паролів у WireGuard та SSH, що робить підбір ключів технічно неможливим. Спроби таємно змінити дані чи видалити логи миттєво виявляються системою моніторингу Wazuh, яка копіює всі записи на незалежний сервер, де хакер не може їх дістати чи відредагувати.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		34

Покриття загроз

Категорія загрози (STRIDE)	Потенційний вектор атаки в інфраструктурі	Компонент системи захисту	Механізм нейтралізації загрози
Spoofing (Підміна особи)	Спроба несанкціонованого віддаленого входу під виглядом легітимного адміністратора.	WireGuard VPN, OpenSSH	Автентифікація виключно за допомогою криптографічних ключів (Curve25519, Ed25519). Парольний доступ повністю відключено.
Tampering (Порушення цілісності)	Модифікація зловмисником конфігураційних файлів шлюзу або бази даних.	Wazuh Agent (FIM), pgBackRest	Модуль File Integrity Monitoring миттєво фіксує зміну хешу файлів. Відновлення цілісності відбувається із зашифрованого бекапу.
Repudiation (Відмова від дій)	Зловмисник (або інсайдер) видаляє системні логи, щоб приховати факт викрадення даних.	pgAudit, Remote Syslog (SIEM)	Детальний аудит дій у БД відправляється на ізольований сервер (Wazuh). Зловмисник не має доступу до центрального сховища логів.
Information Disclosure (Розголошення інформації)	Перехоплення трафіку в мережі (Sniffing) або фізичне викрадення диска з сервером БД.	pgcrypto, TLS 1.3	Дані зашифровані в русі (TLS), на рівні диска (AES-256) та на рівні чутливих стовпців таблиць (pgcrypto).
Denial of Service (Відмова в обслуговуванні)	Перевантаження сервера бази даних сміттєвими запитами або масовий брутфорс SSH.	OPNsense (ACL), Wazuh Active Response	Фаєрвол відсікає нелегітимні IP-адреси. SIEM-система при виявленні атаки автоматично додає IP хакера в блок-лист (firewall-drop).
Elevation of Privilege (Підвищення привілеїв)	Експлуатація вразливості в ПЗ бази даних для отримання прав суперкористувача (root).	Suricata NIDS, AppArmor	NIDS блокує відомі мережеві експлойти на льоту. AppArmor ізолює процес СКБД, не дозволяючи йому вийти за межі своєї директорії.

Висновки до розділу 2

У другому розділі кваліфікаційної роботи я провів теоретичне проектування та розробив архітектуру комплексної системи захисту. На основі аналізу вимог безпеки було створено логічну структуру мережі із чітким розподілом на ізольовані віртуальні сегменти, включаючи організацію демілітаризованої зони для надійної фільтрації вхідних потоків даних.

Я повністю обґрунтував вибір та спроектував інтеграцію ефективних програмних рішень із відкритим вихідним кодом, що дозволило створити надійну Enterprise-архітектуру без залучення високовартісних комерційних ліцензій.

У межах цього етапу було детально розроблено правила міжмережевого екранування, архітектуру VPN-мережі на базі сучасного криптографічного протоколу WireGuard, принципи функціонування багатопотокової системи виявлення вторгнень Suricata, а також комплексну модель підсилення безпеки операційної системи хоста та СКБД PostgreSQL.

Додатково було інтегровано підсистеми детального аудиту даних через pgAudit, надійного шифрованого резервного копіювання на основі pgBackRest та централізованого моніторингу подій безпеки на базі Wazuh SIEM. Розроблена модель загроз STRIDE та побудована карта покриття довели високу ефективність спроектованої архітектури, яка забезпечує надійний захист конфіденційної інформації на всіх рівнях її обробки та зберігання, створюючи міцний фундамент для подальшої практичної реалізації системи на тестовому стенді.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		36

РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ

3.1. Розгортання тестового стенду

3.1.1. Опис апаратно-програмної платформи

Для практичної реалізації спроектованої архітектури ешелонованого захисту та проведення експериментального тестування мною було розгорнуто комплексний тестовий стенд. Як базову апаратно-програмну платформу для побудови стенда я обрав систему віртуалізації Oracle VM VirtualBox. Вибір цього гіпервізора другого типу зумовлений його високою гнучкістю у налаштуванні складних мережесхем топологій, повною підтримкою ізольованих віртуальних комутаторів (Internal Networks) та можливістю детального розподілу апаратних ресурсів хостової машини. Завдяки VirtualBox вдалося відтворити повноцінну інфраструктуру підприємства з декількома логічними сегментами на базі одного фізичного комп'ютера.

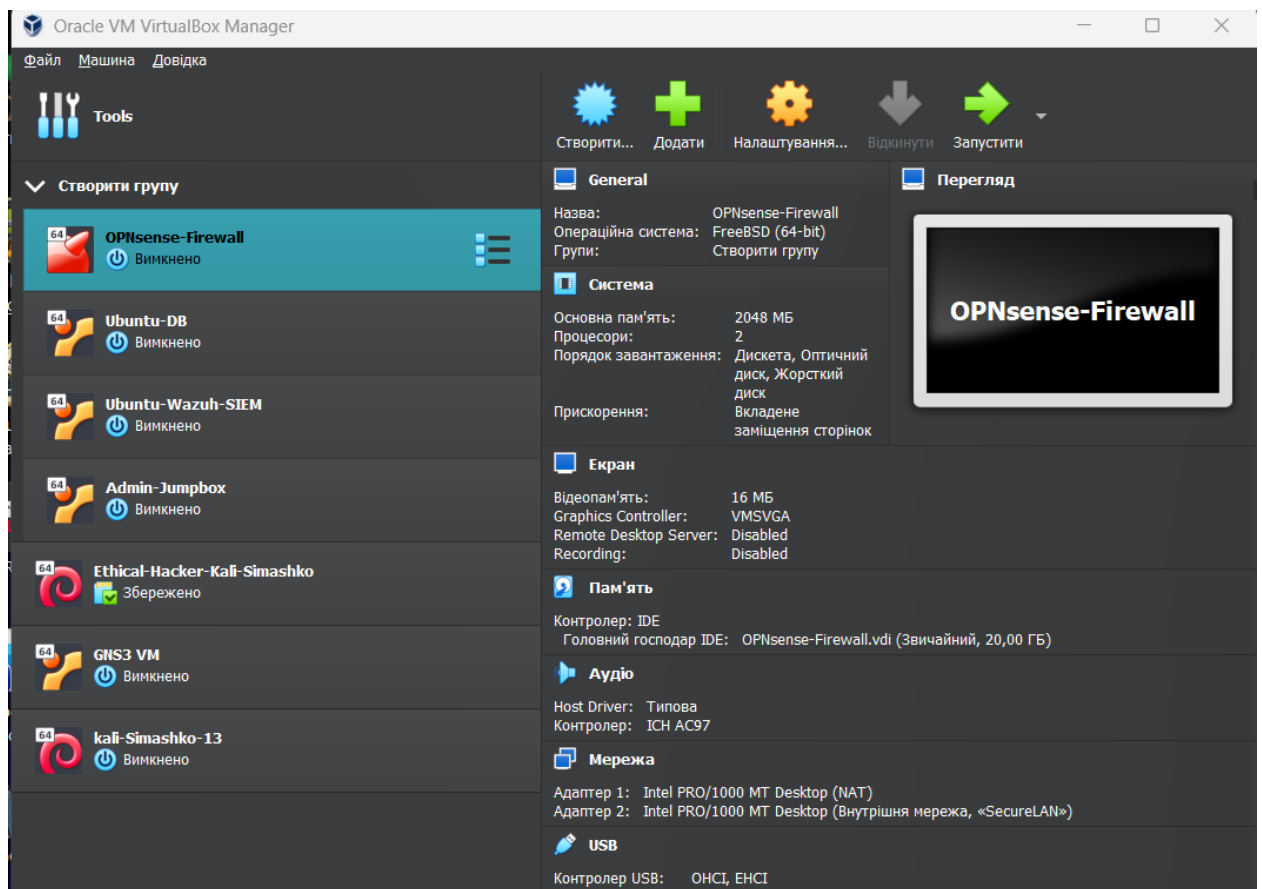
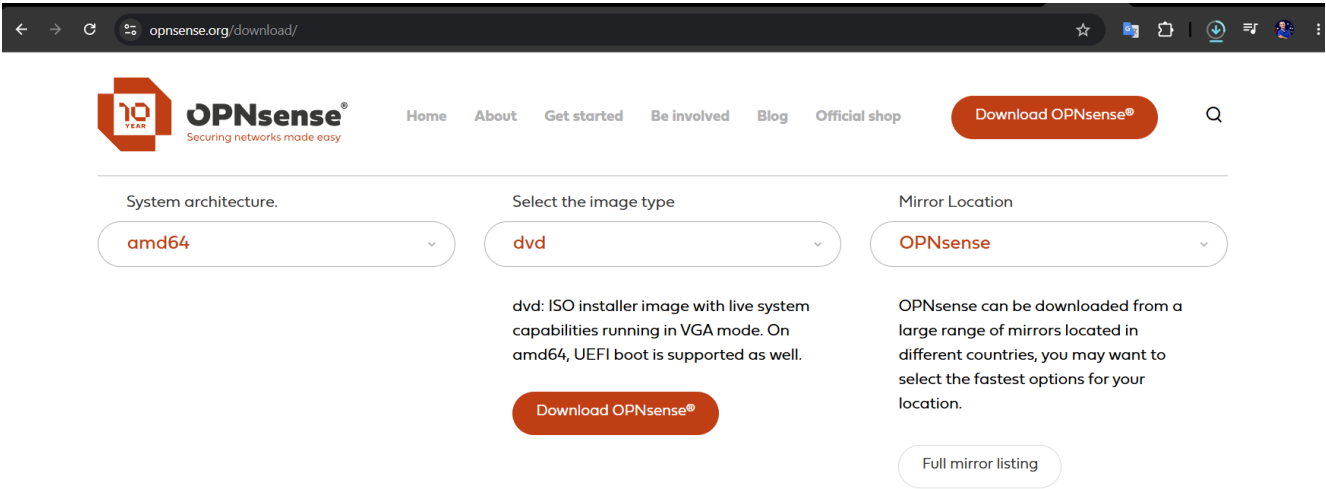


Рисунок 3.1 – Інтерфейс Oracle VM VirtualBox

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				37
Змн.	Арк.	№ докум.	Підпис	Дата		

Особливу увагу під час підготовки стенда я приділив вибору версій операційних систем, оскільки від їхньої стабільності та архітектури залежить безпека всієї інфраструктури. Для організації головного шлюзу безпеки та периметрового фаєрвола було використано актуальний образ OPNsense-26.1.6-dvd-amd64.iso. Ця версія базується на захищеній операційній системі FreeBSD і містить найновіші патчі безпеки, оновлені криптографічні бібліотеки для роботи WireGuard та оптимізоване ядро для високошвидкісної обробки пакетів системою виявлення вторгнень Suricata.



Checksum verification

Рисунок 3.2 – Офіційна сторінка завантаження

Серверну інфраструктуру, що включає вузли баз даних та систему моніторингу SIEM, я розгорнув на базі ubuntu-26.04-live-server-amd64.iso. Вибір серверної редакції Ubuntu версії LTS (Long Term Support) є галузевим стандартом, що гарантує довгострокову стабільність, відсутність зайвих графічних компонентів (що суттєво зменшує поверхню атаки) та ідеальну сумісність із необхідним захисним програмним забезпеченням, таким як AppArmor, UFW та агенти Wazuh.

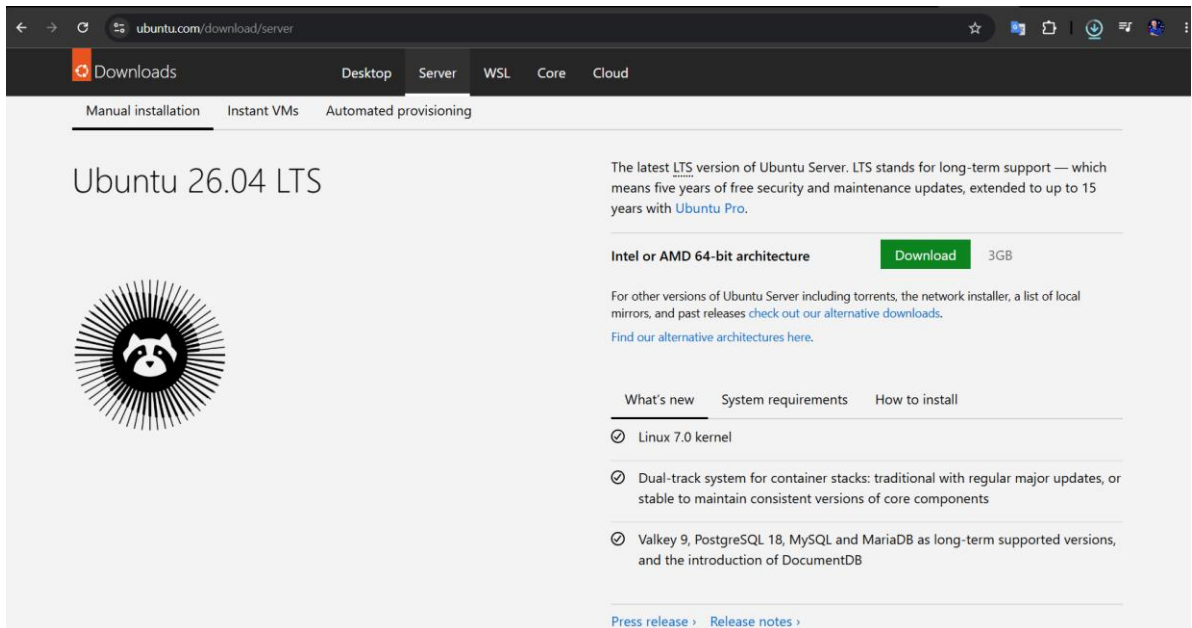


Рисунок 3.3 – Офіційна сторінка продукту

Для імітації робочих станцій клієнтів та тестування доступу всередині мережі мною було інтегровано образ `lubuntu-26.04-desktop-amd64.iso`. Цей дистрибутив відрізняється використанням надзвичайно легкого графічного середовища LXQt. Його застосування дозволило повноцінно симулювати поведінку користувача в системі, тестувати веб-інтерфейси та роботу VPN-клієнтів, витрачаючи при цьому мінімальний обсяг оперативної пам'яті та потужностей процесора, зберігаючи ресурси для критично важливих серверів захисту.

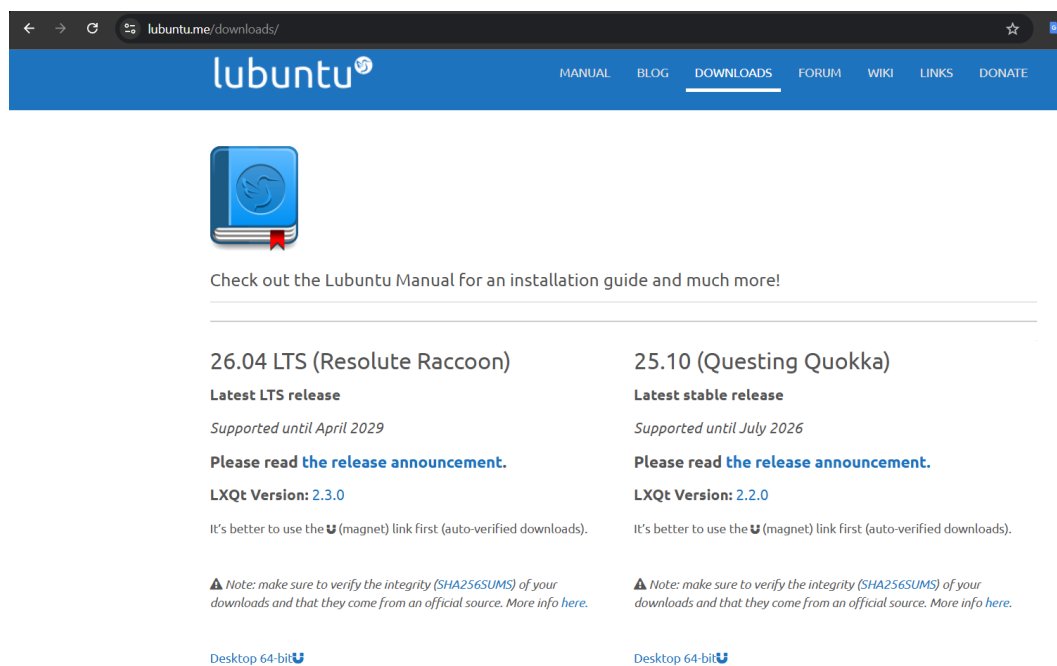


Рисунок 3.4 – Сторінка завантаження lubuntu

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				39
Змн.	Арк.	№ докум.	Підпис	Дата		

3.1.2. Специфікація віртуальних машин та параметрів мережі

Процес створення тестового полігону розпочався з ініціалізації та конфігурування базових параметрів віртуальних машин. Першим елементом інфраструктури став головний міжмережевий екран OPNsense. Під час створення цієї віртуальної машини у менеджері VirtualBox, мною було вказано тип операційної системи BSD із версією FreeBSD (64-bit), що забезпечує коректну підтримку ядра OPNsense гіпервізором.

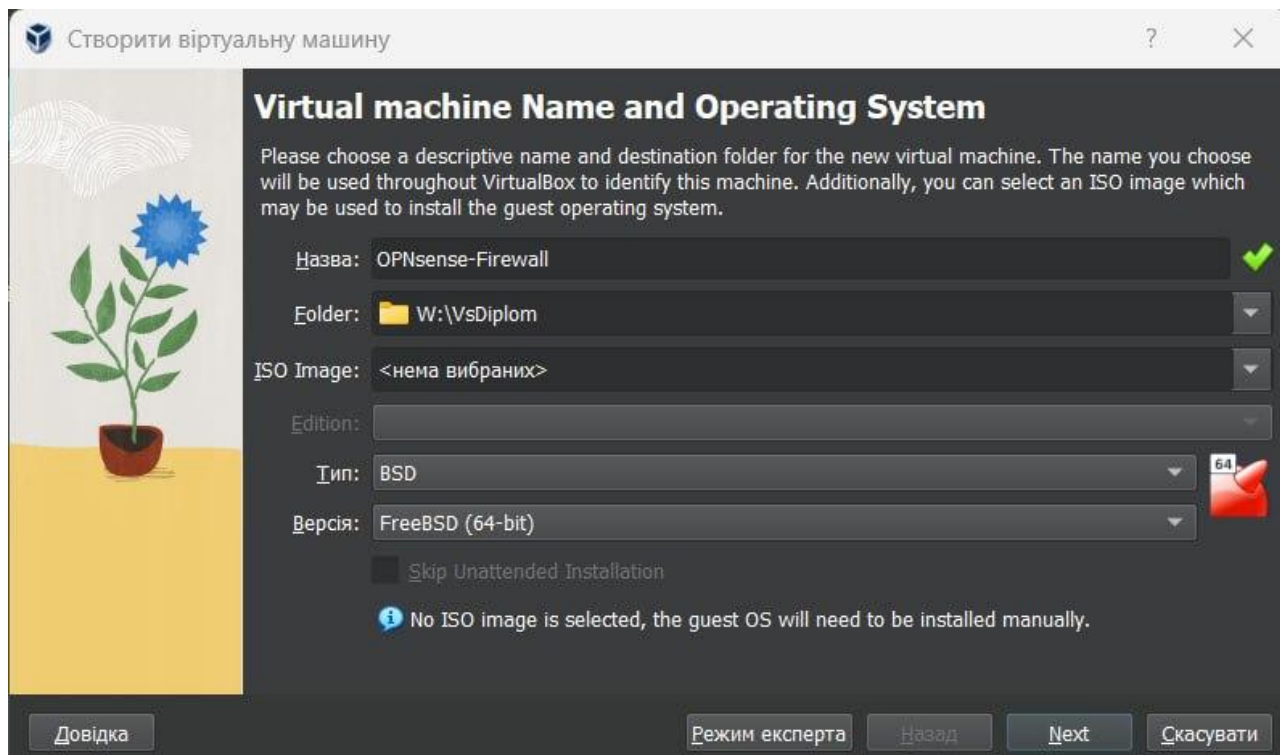


Рисунок 3.5 – Початок встановлення віртуальних машин

Для того щоб віртуальна машина OPNsense справлялась з функціями міжмережевого екрана, фаєрвола, тримала на собі NIDS Suricata, маршрутизатору було виділено два обчислювальні ядра процесора та 2048 мегабайтів оперативної пам'яті.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				40
Змн.	Арк.	№ докум.	Підпис	Дата		

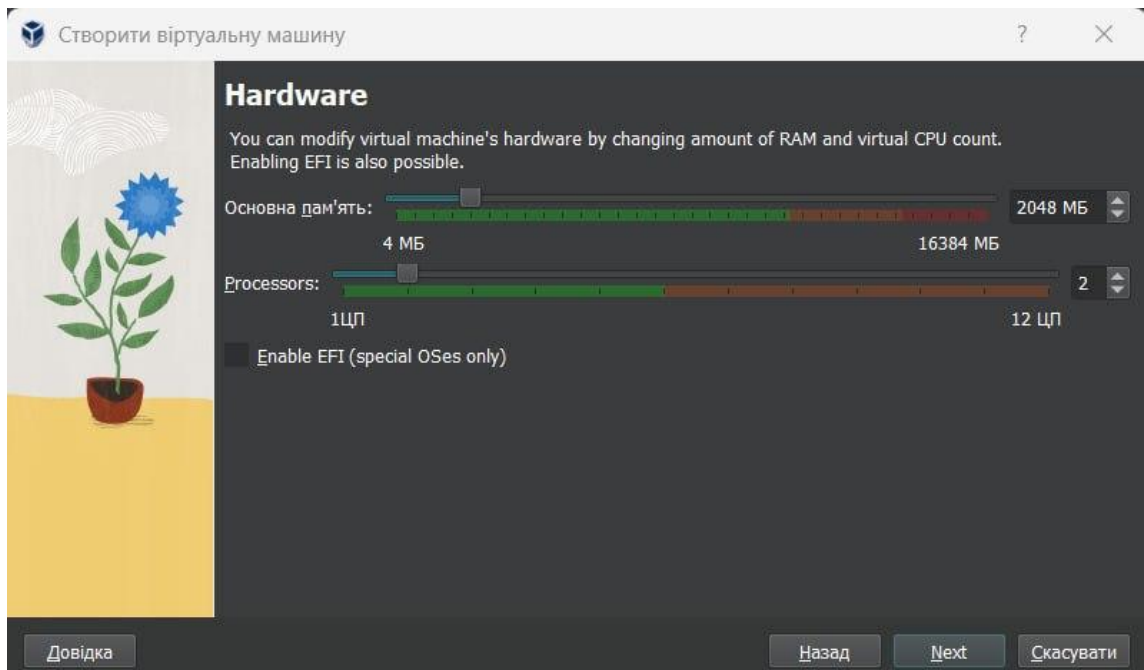


Рисунок 3.6 – Виділення ОЗУ та кількість ядер

Як носій інформації для встановлення системи я створив динамічний віртуальний жорсткий диск формату VDI об'ємом 20 гігабайтів, чого цілком достатньо для операційної системи та зберігання локальних журналів подій перед їх відправкою на центральний SIEM-сервер. Завантаження інсталяційних файлів здійснювалося шляхом монтування завантаженого ISO-образу у віртуальний оптичний привід машини.

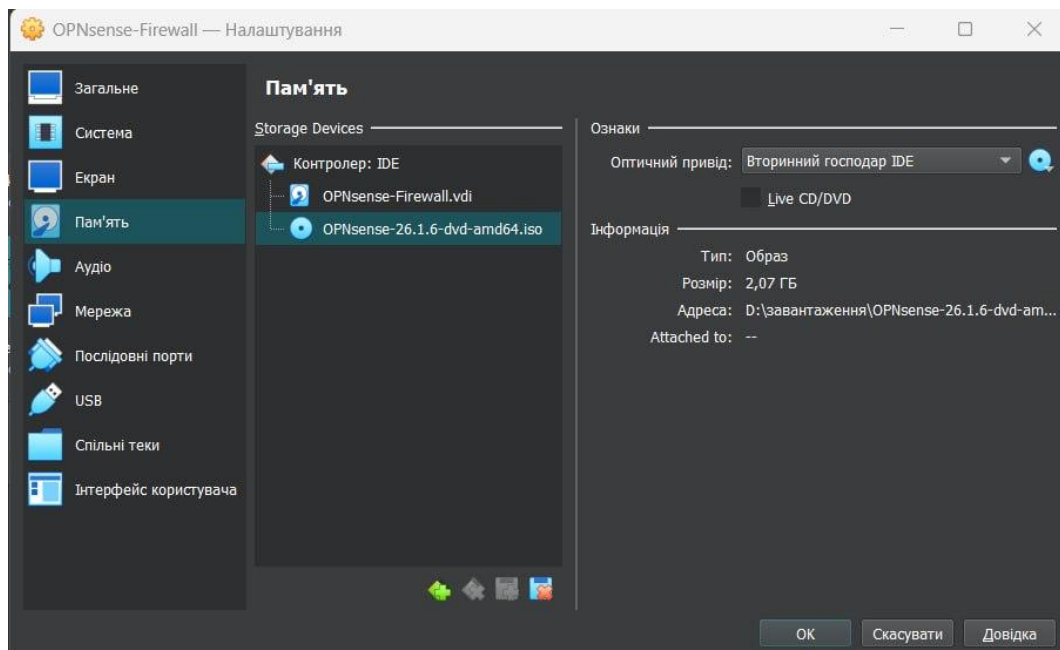


Рисунок 3.7 – Завантаження образу у віртуальний оптичний привід

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				41
Змн.	Арк.	№ докум.	Підпис	Дата		

Найважливішим етапом специфікації стало конфігурування мережевих інтерфейсів, які формують логічну топологію нашого стенда. Для віртуальної машини OPNsense я задіяв два мережеві адаптери типу Intel PRO/1000 MT Desktop, що імітують фізичні мережеві карти. Перший адаптер був налаштований у режимі трансляції мережевих адрес (NAT). Цей інтерфейс виконує роль зовнішнього WAN-підключення і автоматично отримав від гіпервізора IP-адресу 10.0.2.15/24. Саме через цей інтерфейс маршрутизатор має вихід у глобальну мережу Інтернет та приймає вхідні зашифровані тунелі від віддалених адміністраторів.

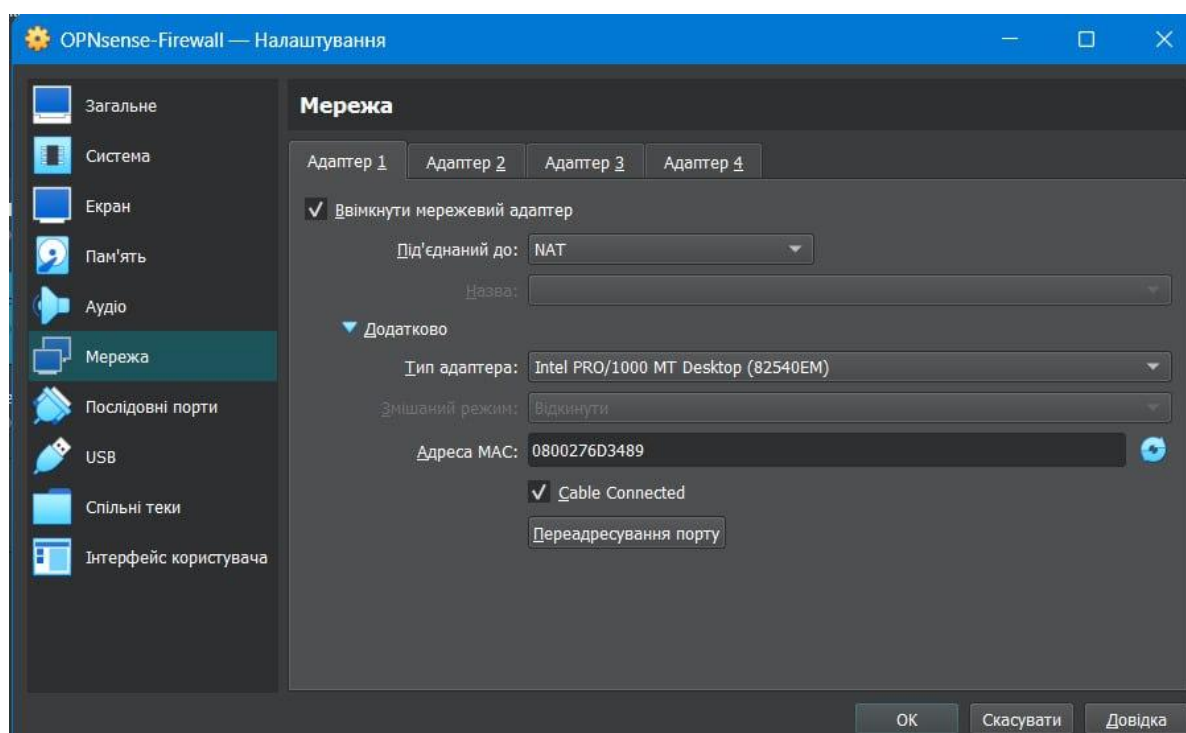


Рисунок 3.8 – Адаптер NAT для OPNsense

Другий мережевий адаптер був переведений у режим внутрішньої мережі (Internal Network) та отримав назву SecureLAN. Цей інтерфейс формує повністю ізольований від фізичного комп'ютера комутатор, до якого в подальшому були підключені сервери баз даних та моніторингу. Під час первинного завантаження та конфігурації OPNsense через консольний інтерфейс, цьому внутрішньому адаптеру (LAN) була жорстко призначена статична IP-адреса 192.168.90.1/24, яка стала основним шлюзом за замовчуванням для всієї внутрішньої серверної інфраструктури.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		42

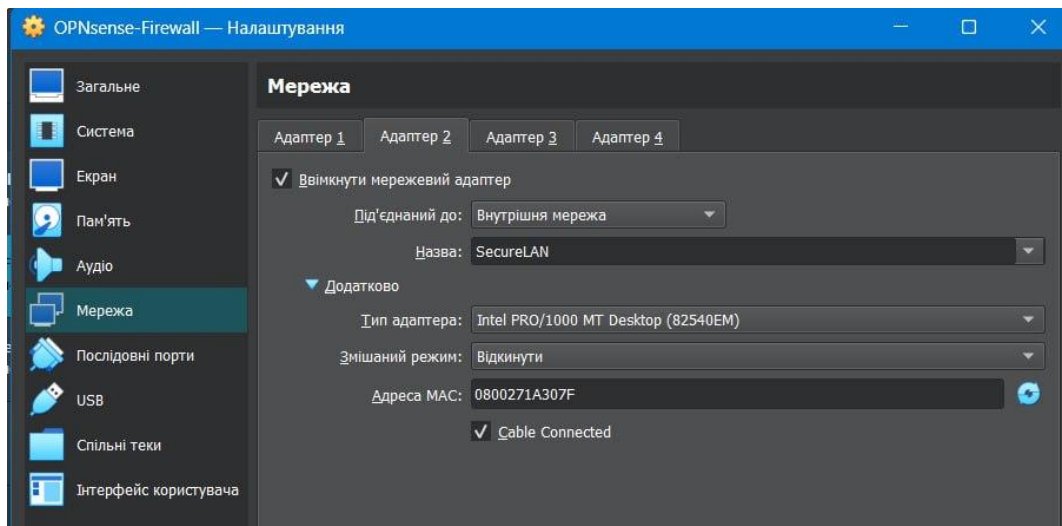


Рисунок 3.9 – Внутрішній адаптер LAN

Усі інші сервери (Ubuntu-DB, Ubuntu-Wazuh-SIEM) та клієнтські станції були створені в менеджері віртуальних машин за аналогічним принципом виділення ресурсів, однак їхні мережеві адаптери підключалися виключно до створеного ізольованого сегмента SecureLAN. Такий підхід до специфікації мережі на рівні гіпервізора повністю відповідає спроектованій у другому розділі архітектури ешелонованого захисту, унеможливлюючи прямий фізичний доступ зовнішніх пристроїв до критично важливих вузлів інфраструктури в обхід головного фаєрвола.

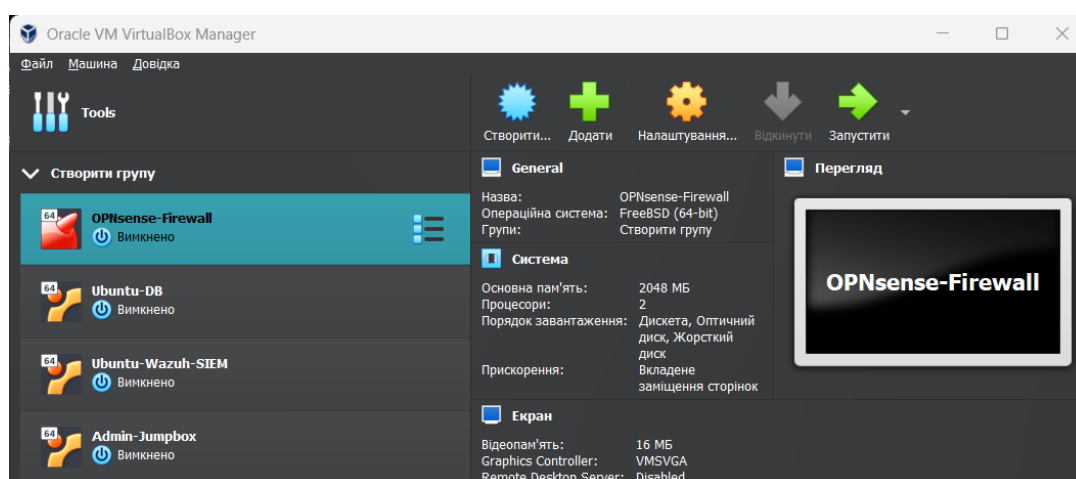


Рисунок 3.10 – Список налаштованих станцій

Детальні налаштування кожної робочої станції є в додатках до кваліфікаційної роботи.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				43
Змн.	Арк.	№ докум.	Підпис	Дата		

3.1.3. Загальна схема тестового полігону

Для візуалізації всіх машин та побудованого середовища мною було розроблено загальну структурну схему тестового полігону в cisco packet tracer яка повністю відображає наявні робочі станції, зони в яких вони знаходяться та їх призначення. Центральним вузлом топології виступає програмний маршрутизатор OPNsense, який виконує функцію єдиної точки входу та контролю над усіма мережевими потоками. Його зовнішній інтерфейс підключено до віртуального NAT-адаптера VirtualBox, який імітує підключення до глобальної мережі Інтернет та отримує динамічну IP-адресу 10.0.2.15. Саме через цей інтерфейс зазвичай можуть здійснюватися кібератаки з машин таких як Kali Linux та ініціюються легітимні запити на встановлення зашифрованих VPN-з'єднань від віддалених робочих станцій адміністратора.

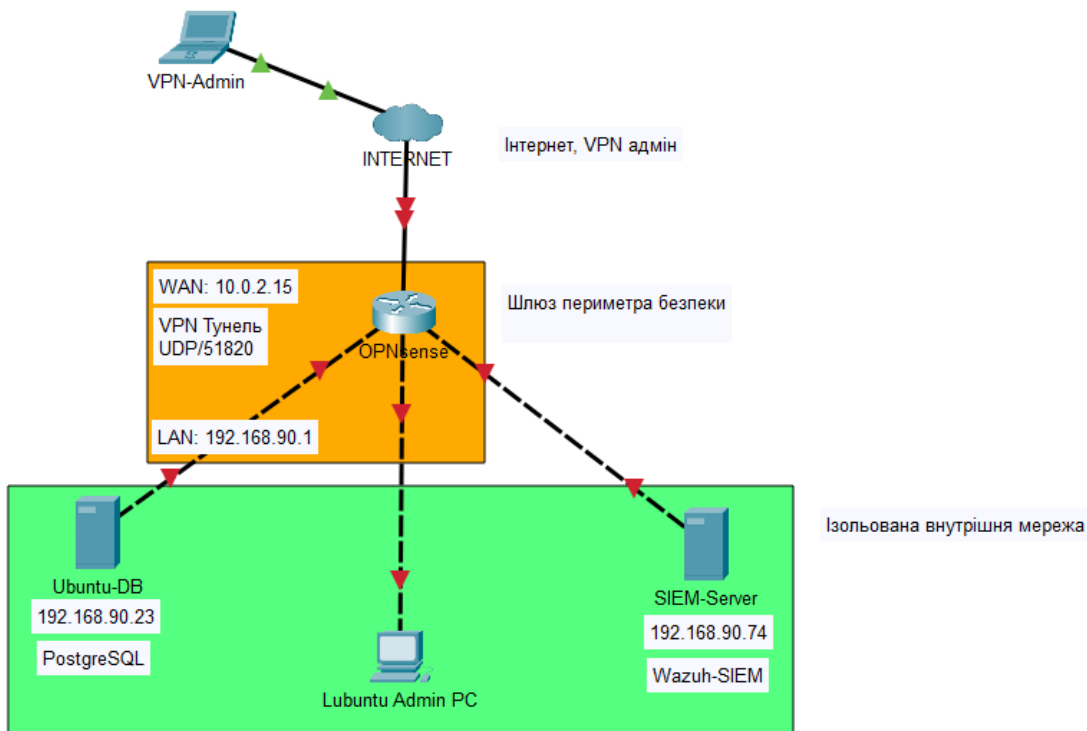


Рисунок 3.11 – Схема мережі

Внутрішня інфраструктура полігону ізольована від зовнішнього світу за допомогою віртуального комутатора типу «Внутрішня мережа» (Internal Network) під назвою SecureLAN. До цього сегмента підключено локальний інтерфейс шлюзу безпеки зі статичною адресою 192.168.90.1, який виступає основним маршрутизатором для внутрішніх вузлів. У цьому ж сегменті функціонують два

критично важливі сервери: сервер бази даних (Ubuntu-DB) з адресою 192.168.90.23 та сервер системи управління інформаційною безпекою (Ubuntu-Wazuh-SIEM) з адресою 192.168.90.74. Для забезпечення безпечного віддаленого доступу на шлюзі додатково розгорнуто віртуальний тунельний інтерфейс WireGuard (wg0), який виділяє клієнтам адреси з підмережі 10.0.0.0/24 та забезпечує їхню логічну маршрутизацію до внутрішніх ресурсів після успішної криптографічної автентифікації.

Спроектowana схема наочно демонструє рух інформаційних потоків та подій безпеки. Увесь транзитний трафік, який перетинає межу між зовнішнім та внутрішнім сегментами, проходить обов'язковий аналіз підсистемою виявлення вторгнень Suricata безпосередньо на шлюзі. Водночас системні журнали маршрутизатора, події аудиту бази даних PostgreSQL та сповіщення від хостових агентів безпеки збираються в єдиний інформаційний потік і спрямовуються по протоколах Syslog та TLS до центрального сервера аналітики Wazuh-SIEM, що формує завершену ешелоновану архітектуру моніторингу та захисту тестового полігону.

3.2. Налаштування периметрового захисту

3.2.1. Інсталяція та базове конфігурування OPNsense

Інсталяція програмного маршрутизатора OPNsense здійснювалася з офіційного інсталяційного образу версії 26.1.6 на попередньо виділену віртуальну машину. Процес встановлення включав автоматичну розмітку віртуального жорсткого диска та розгортання базової файлової системи на основі ядра HardenedBSD. Після першого перезавантаження системи мною було виконано критично важливий етап первинної ініціалізації, який полягав у правильному розподілі мережевих інтерфейсів для створення ешелонованої структури.

У консольному меню управління OPNsense інтерфейс em0, який на рівні гіпервізора підключений до адаптера трансляції мережевих адрес (NAT), був призначений як зовнішній канал (WAN). Для цього інтерфейсу було активовано режим автоматичного отримання параметрів по протоколу DHCP, що дозволило йому отримати адресу 10.0.2.15. Інтерфейс em1, який фізично ізольований у

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		45

віртуальній мережі SecureLAN, був налаштований як внутрішній шлюз (LAN). Йому було жорстко призначено статичну IPv4-адресу 192.168.90.1 із маскою підмережі /24. Саме ця конфігурація сформувала початковий бар'єр між імітованою глобальною мережею та захищеним внутрішнім сегментом тестового полігону.

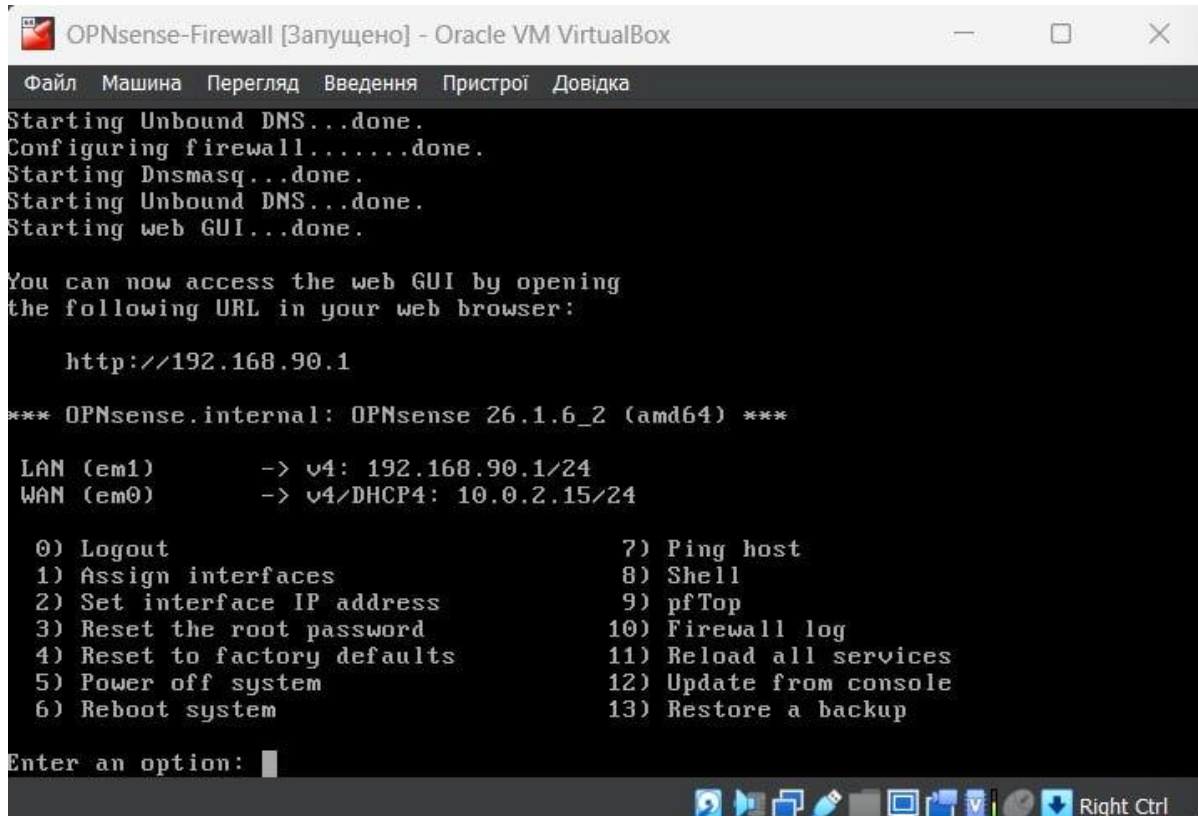


Рисунок 3.12 – Налаштовані інтерфейси на OPNsense

Після успішної ініціалізації мережевих адаптерів подальше конфігурування системи здійснювалося через графічний вебінтерфейс. Доступ до панелі управління було отримано за допомогою браузера з робочої станції, підключеної до локальної мережі. У процесі базового налаштування через майстер початкової конфігурації мною було задано унікальне ім'я хоста шлюзу, зробив регіональні налаштування та синхронізацію точного часу (NTP), що є обов'язковою вимогою для правильної роботи криптографічних протоколів та достовірної фіксації часу подій у журналах аудиту безпеки.

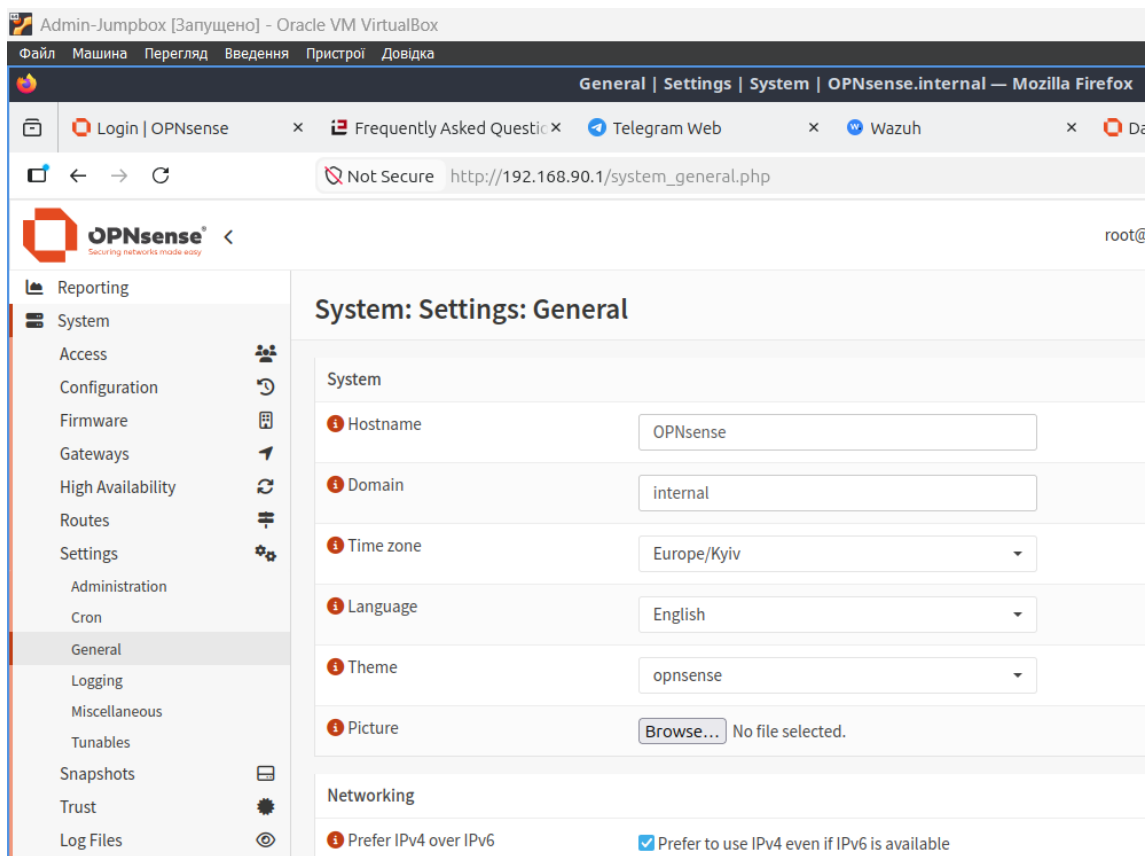


Рисунок 3.13 – Базові налаштування

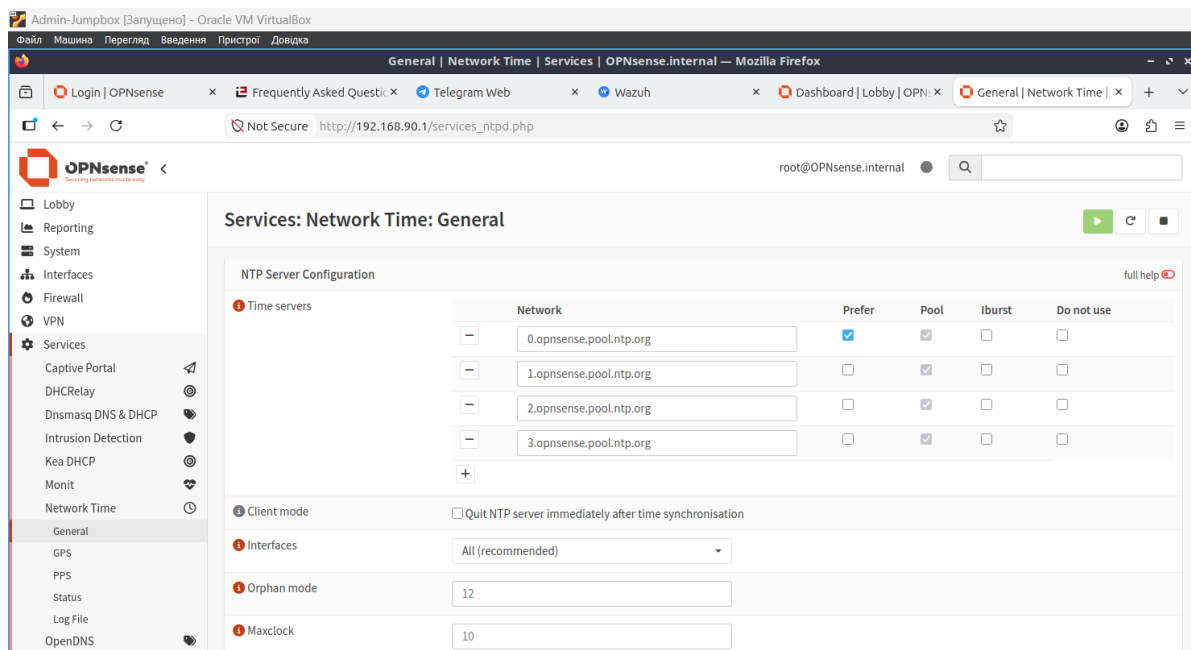


Рисунок 3.14 – Налаштування NTP синхронізації часу

3.2.2. Конфігурація інтерфейсів, VLAN та правил фаєрвола

Наступним кроком практичної реалізації стало детальне налаштування мережної взаємодії та політик безпеки на головному міжмережевому екрані

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				47
Змн.	Арк.	№ докум.	Підпис	Дата		

OPNsense. Процес конфігурації розпочався з логічного прив'язування віртуальних мережеских адаптерів гіпервізора до системних інтерфейсів шлюзу (рис. 3.15). Адаптер em0 був призначений як зовнішній інтерфейс (WAN) для зв'язку з імітованою глобальною мережею, а адаптер em1 – як внутрішній шлюз (LAN) для захищеної мережі.

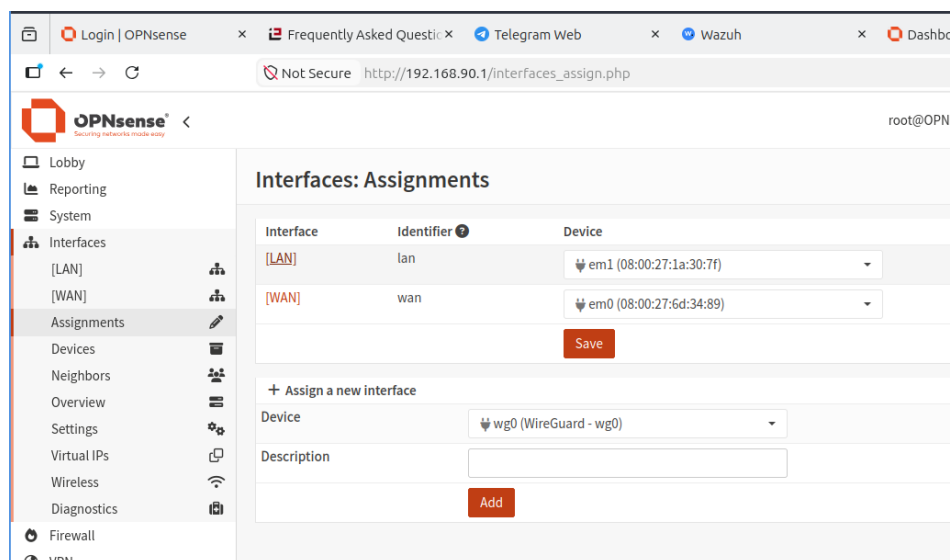


Рисунок 3.15 – Налаштовані інтерфейси

Відповідно до архітектури проєкту, сегментація інфраструктури на фізичному обладнанні передбачає використання стандарту тегування трафіку IEEE 802.1Q (VLAN). Однак, враховуючи специфіку розгортання тестового полігону у середовищі VirtualBox, логічне сегментування мережі було реалізовано засобами самого гіпервізора. Роль ізольованих VLAN у стенді виконує тип підключення «Внутрішня мережа» (Internal Network) з ідентифікатором SecureLAN. Цей механізм забезпечує повну апаратну ізоляцію широкомовних доменів на другому рівні моделі OSI. Саме тому додаткове створення VLAN-інтерфейсів безпосередньо в панелі управління OPNsense не виконувалося, оскільки необхідний рівень сегментації вже забезпечено архітектурою віртуального комутатора.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				48
Змн.	Арк.	№ докум.	Підпис	Дата		

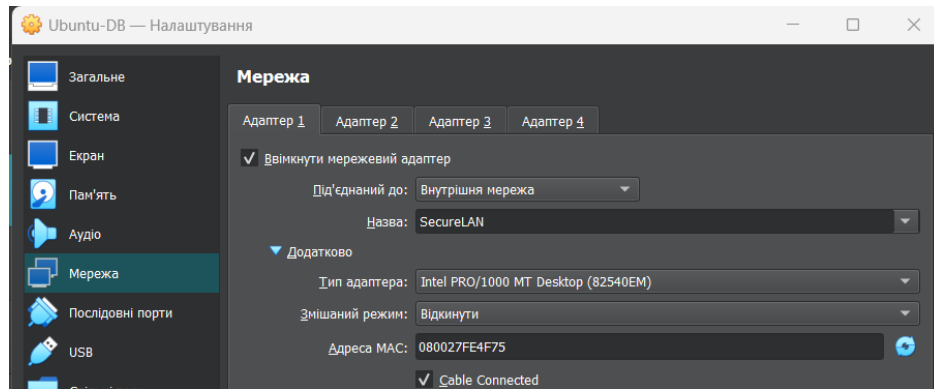


Рисунок 3.16 – Ізольована LAN

Ключовим етапом побудови ешелонованої системи стало проектування списків контролю доступу (ACL) у модулі фаєрвола. Для реалізації принципу мінімальних привілеїв, політика зовнішнього інтерфейсу WAN була налаштована на повне блокування всього вхідного трафіку (Default Deny). Єдиним винятком стало створення дозволяючого правила для протоколу UDP на порт 51820. Цей порт необхідний для термінації вхідних підключень до захищеного VPN-тунелю WireGuard. Усі інші спроби сканування або встановлення з'єднань із зовнішньої мережі автоматично відкидаються міжмережним екраном без надсилання відповіді, що робить інфраструктуру невидимою для сканерів розвідки.

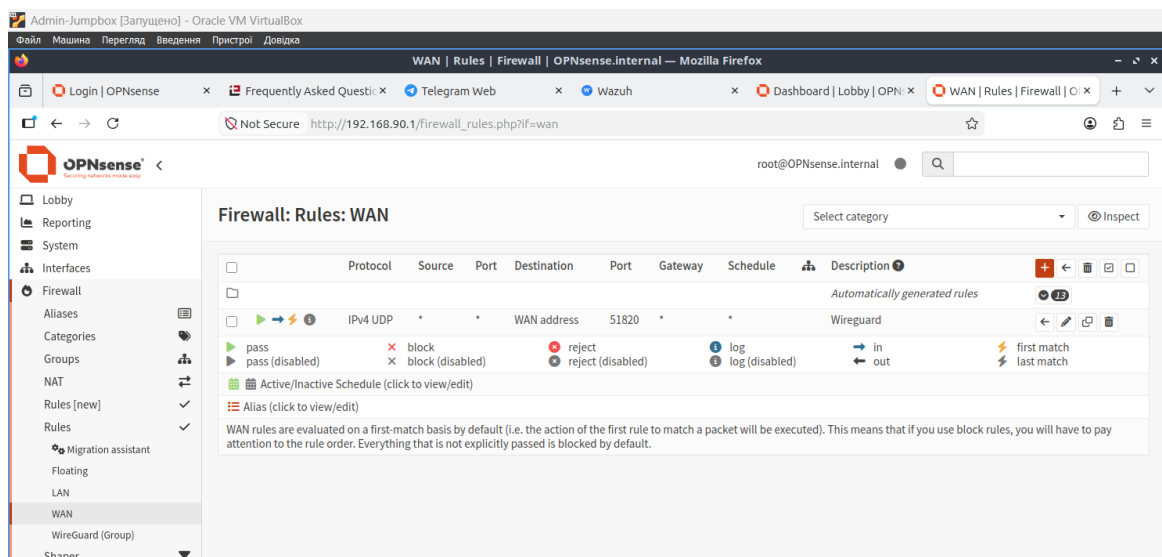


Рисунок 3.17 – Фаєрвол на WAN

Для внутрішнього сегмента LAN були створені правила, які суворо обмежують міжмережну взаємодію. Фаєрвол регламентує обмін трафіком між серверами, гарантуючи, що у випадку компрометації одного з вузлів, зловмисник

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				49
Змн.	Арк.	№ докум.	Підпис	Дата		

не зможе вільно здійснювати горизонтальне переміщення та атакувати суміжні сервіси без проходження жорсткої перевірки на рівні шлюзу безпеки.

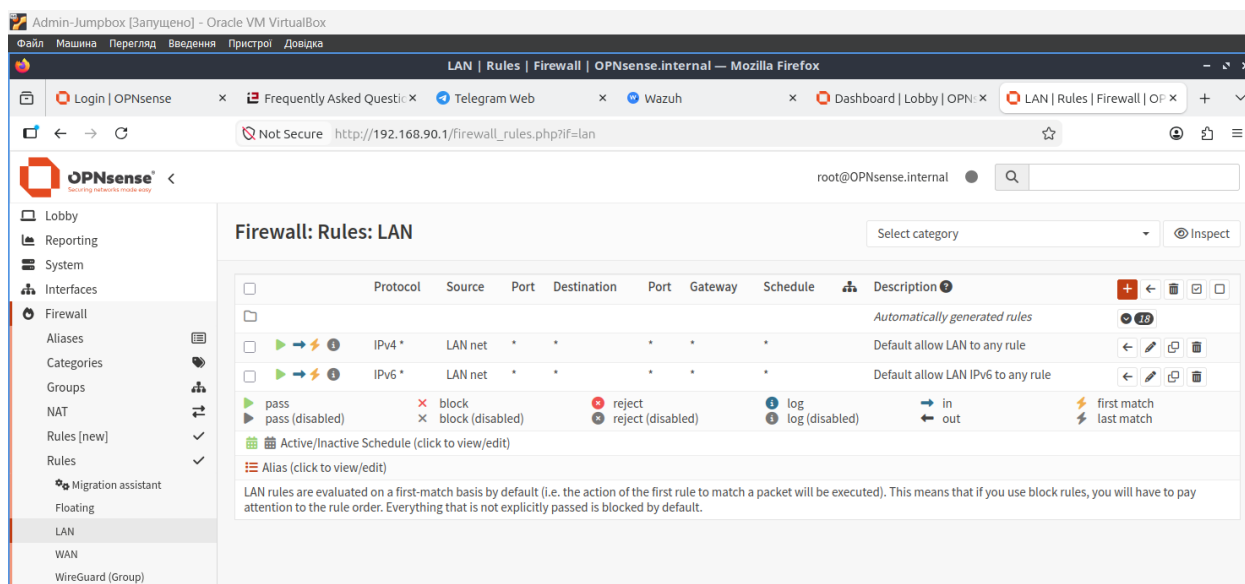


Рисунок 3.18 – Фаєрвол на LAN

3.2.3. Розгортання WireGuard-сервера та генерація ключів клієнтів

Оскільки тестовий полігон розгорнуто у віртуалізованому середовищі гіпервізора VirtualBox, зовнішній інтерфейс маршрутизатора знаходиться за механізмом трансляції адрес (NAT). Тому першим технічним кроком стала конфігурація переадресації портів (Port Forwarding) на рівні самої віртуальної машини. Для того, щоб зовнішні VPN-підключення з фізичної мережі могли безперешкодно досягати нашого шлюзу, було створено правило, яке прокидає вхідний трафік по протоколу UDP з порту 51820 хостової машини безпосередньо на зовнішню WAN-адресу OPNsense – 10.0.2.15. Це дозволило успішно обійти обмеження подвійного NAT у лабораторних умовах.

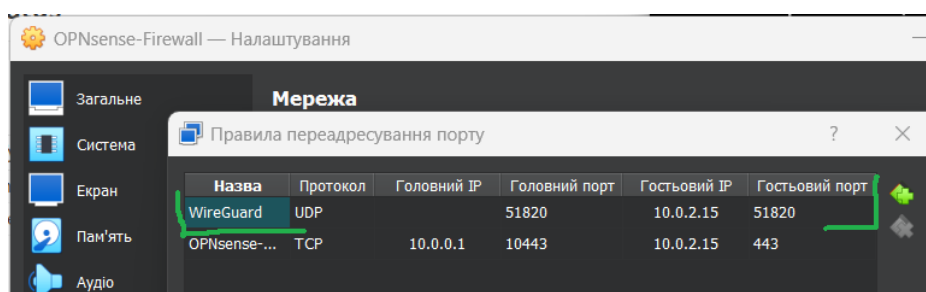


Рисунок 3.19 – Налаштування перекидання порта для WireGuard

Наступним етапом стало створення серверного вузла (Local Instance) безпосередньо у вебінтерфейсі OPNsense. Новому віртуальному тунелю було

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				50
Змн.	Арк.	№ докум.	Підпис	Дата		

присвоєно системний ідентифікатор wg0. Під час ініціалізації система автоматично згенерувала унікальну пару криптографічних ключів (публічний та приватний), які виконують роль своєрідного цифрового паспорта сервера. Для маршрутизації клієнтського трафіку цьому серверному екземпляру було виділено окрему внутрішню підмережу, а самому інтерфейсу призначено IP-адресу 10.0.0.1/24. Прослуховування очікуваних вхідних з'єднань було налаштовано на стандартному порту 51820.

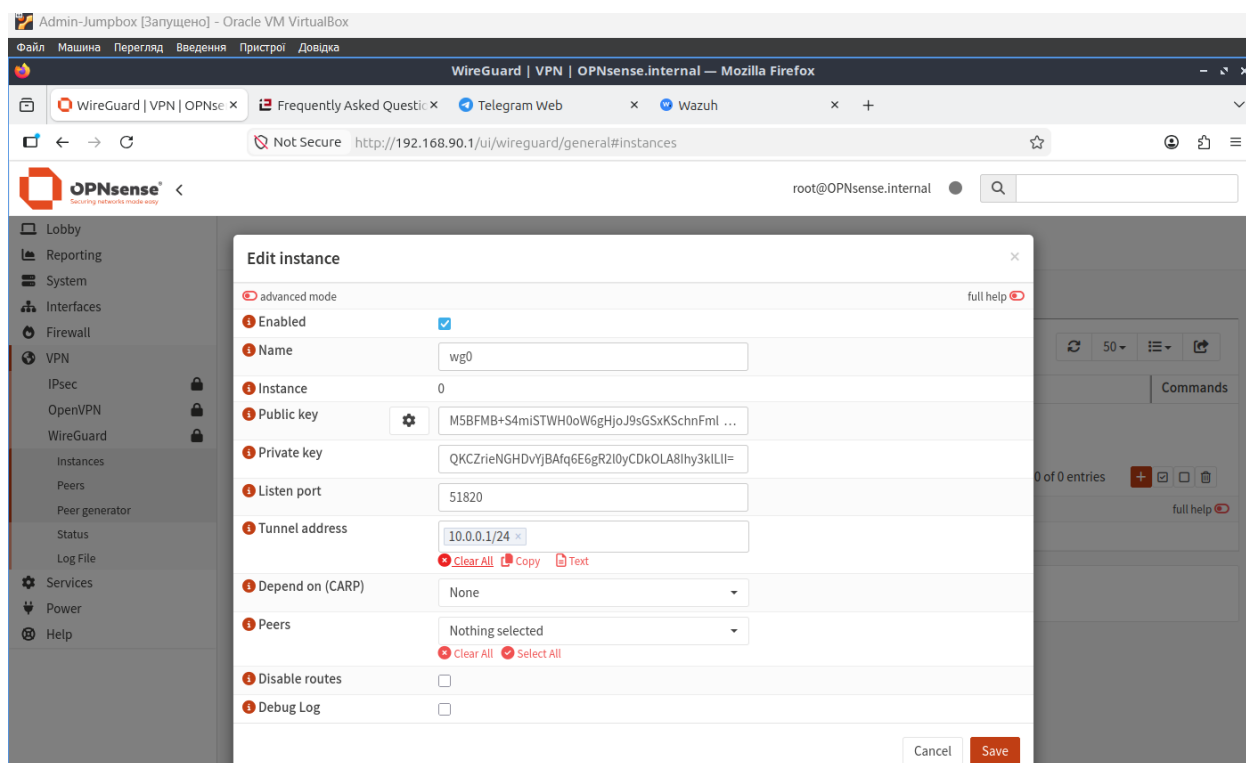


Рисунок 3.20 – Створення серверного вузла

Після підготовки серверної частини було розроблено конфігурацію для довіреного клієнта (Peer). У системі було створено обліковий запис із назвою Admin_Laptop_Simashko, який імітує робочу станцію адміністратора безпеки. Для цього пристрою була жорстко зарезервована статична IP-адреса всередині VPN-тунелю – 10.0.0.2/32. Фіксація адреси є критично важливою вимогою, оскільки вона дозволяє фаєрволу точно розпізнавати адміністратора та застосовувати до нього відповідні правила доступу. Для зручного перенесення налаштувань на кінцевий пристрій було використано вбудований генератор конфігурацій. Цей модуль автоматично сформував готовий набір параметрів та згенерував QR-код, що дозволяє миттєво імпортувати криптографічні ключі та мережеві маршрути на

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				51
Змн.	Арк.	№ докум.	Підпис	Дата		

смартфон або ноутбук, уникаючи ризику перехоплення приватного ключа під час ручного копіювання.

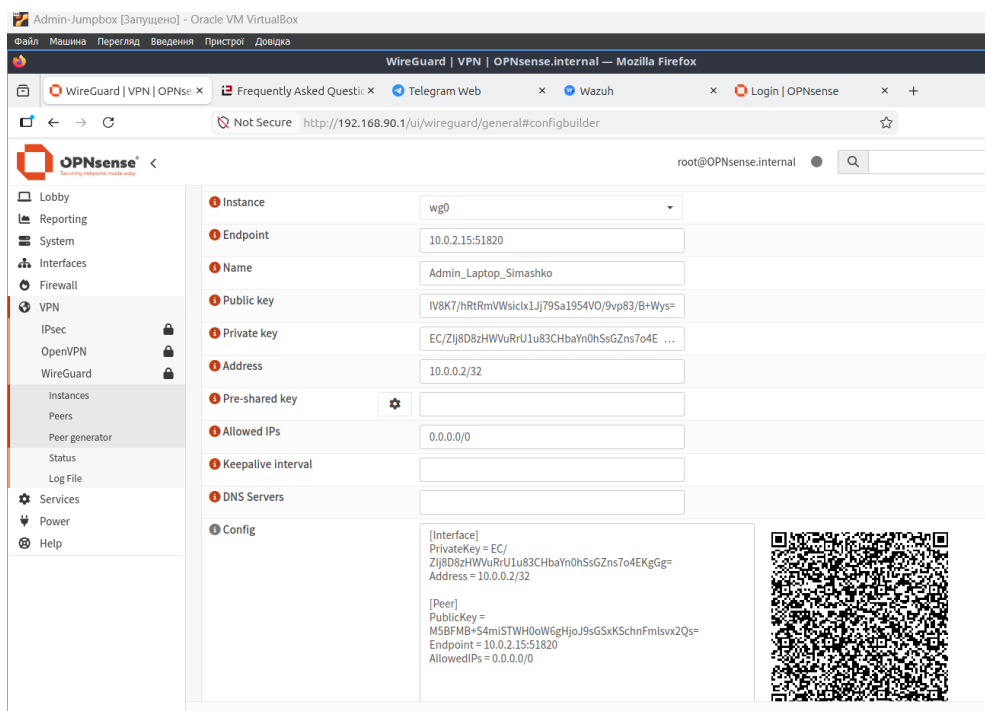


Рисунок 3.21 – Конфігурація для довіреного клієнта

Заключним кроком для забезпечення працездатності системи стало налаштування правил фільтрації трафіку. Оскільки зовнішній інтерфейс WAN за замовчуванням відкидає будь-які ініціалізовані ззовні підключення, у модулі фаєрвола було створено нове дозволяюче правило (Pass). Це правило чітко вказує міжмережевому екрану приймати вхідні пакети по протоколу UDP з будь-якої зовнішньої адреси, але виключно за умови, що вони спрямовані на порт 51820 інтерфейсу WAN. Лише після збереження та застосування цього правила шлюз почав коректно приймати зашифровані запити на підключення, формуючи повноцінну та безпечну точку входу до захищеної мережної інфраструктури.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				52
Змн.	Арк.	№ докум.	Підпис	Дата		

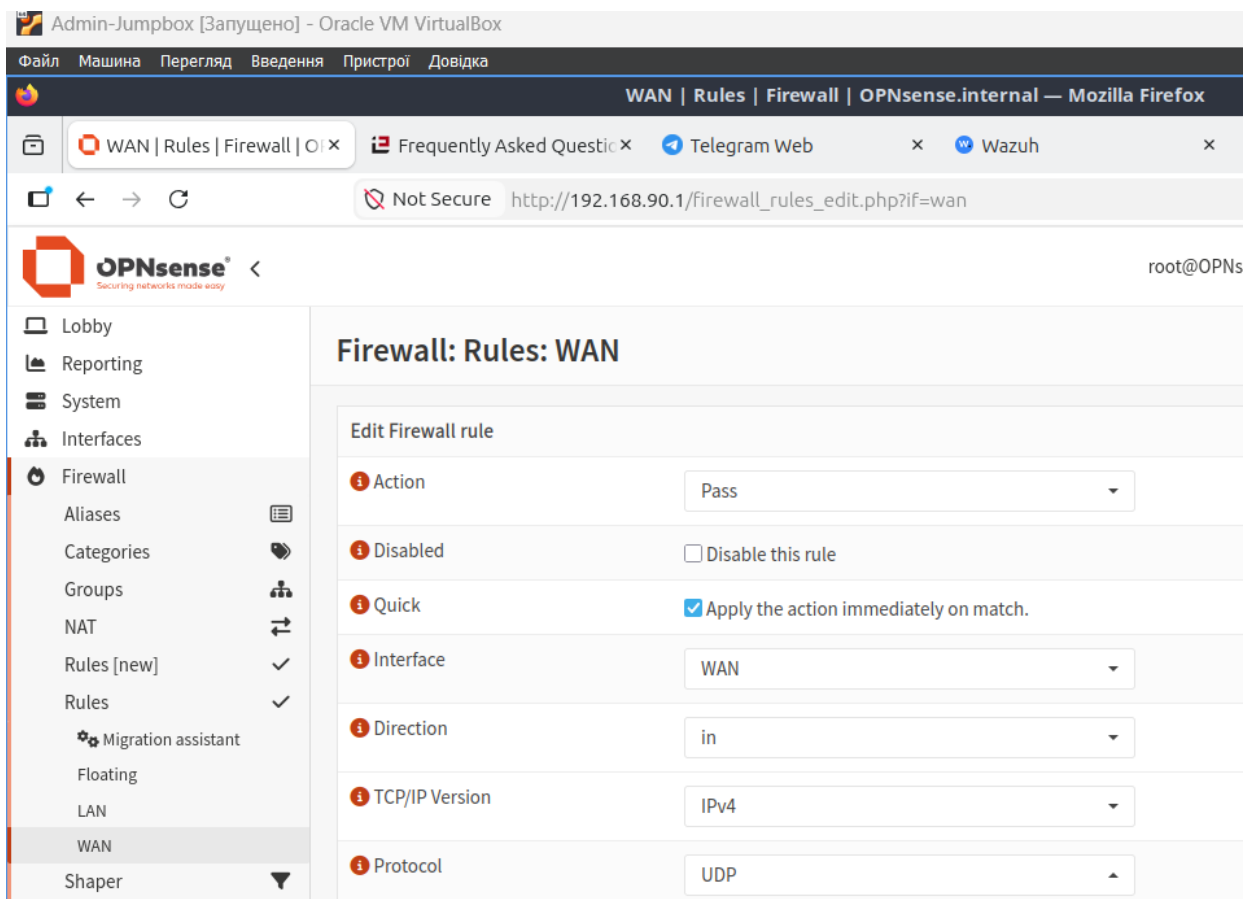


Рисунок 3.22 – Налаштування фільтрації

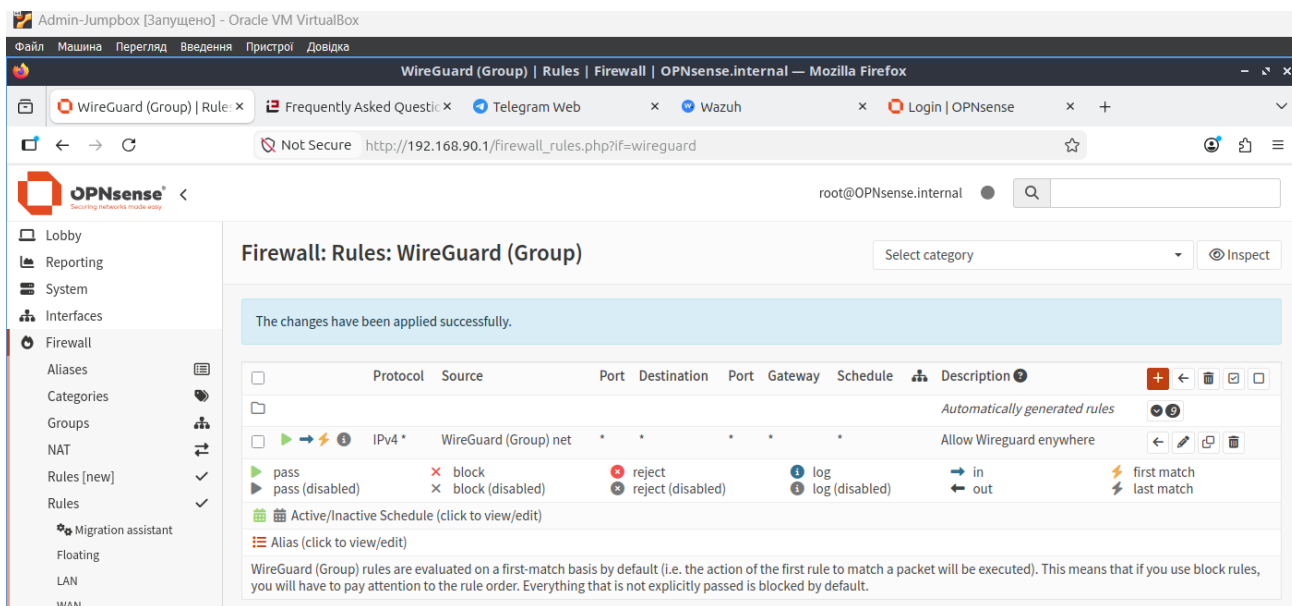


Рисунок 3.23 – Налаштований фаєрвол

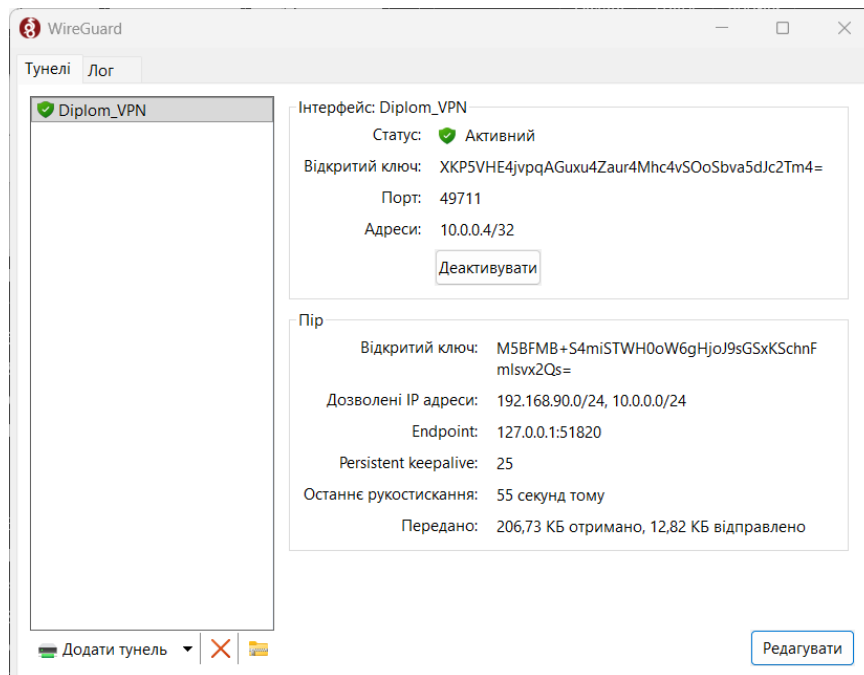


Рисунок 3.24 – Успішне підключення

3.3. Налаштування системи виявлення вторгнень

3.3.1. Ініціалізація та базове налаштування NIDS Suricata

Наступним кроком практичної реалізації стала розбудова внутрішнього ешелону безпеки – мережної системи виявлення вторгнень (NIDS). Відповідно до спроектованої архітектури, як основний аналітичний рушій було обрано систему Suricata. Важливою архітектурною особливістю програмного маршрутизатора OPNsense є те, що на відміну від багатьох інших платформ, де механізми IDS/IPS встановлюються як додаткові сторонні пакети, тут Suricata є нативно інтегрованою частиною базової операційної системи. Це забезпечує глибоку взаємодію з мережним стеком ядра HardenedBSD та значно підвищує продуктивність обробки пакетів завдяки використанню технології Netmap.

Зважаючи на цю особливість, етап розгортання полягав не у фізичному встановленні пакета, а в правильній ініціалізації інтегрованого модуля та його прив'язці до топології тестового полігону. У розділі управління виявленням вторгнень (Intrusion Detection) вебінтерфейсу OPNsense службу Suricata було переведено в активний стан.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		54

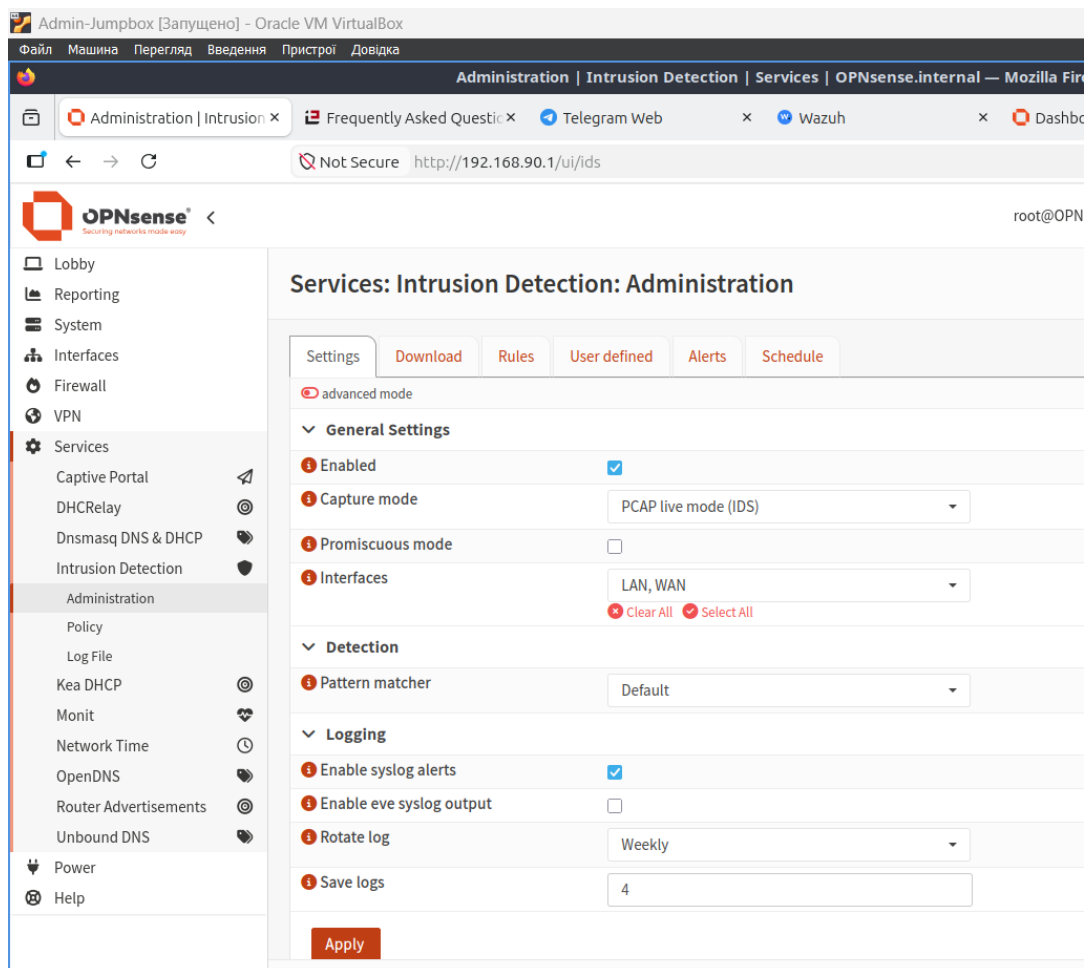


Рисунок 3.25 – Переведення служби в активний стан

Ключовим рішенням на цьому етапі став вибір точок перехоплення трафіку (сенсорів). Для забезпечення максимальної видимості мережевих подій, рушій Suricata було налаштовано на одночасне прослуховування обох логічних інтерфейсів – зовнішнього (WAN) та внутрішнього (LAN). Аналіз трафіку на зовнішньому інтерфейсі дозволяє виявляти спроби сканування периметра, брутфорс-атаки та розвідку ще до того, як вони будуть заблоковані правилами фаєрвола (Default Deny). Водночас прослуховування внутрішнього інтерфейсу LAN є критично необхідним для фіксації можливого горизонтального переміщення зломисника (Lateral Movement) у разі компрометації одного з вузлів, а також для контролю вихідного трафіку від захищених серверів до глобальної мережі. Для коректного захоплення всіх пакетів, що проходять через віртуальний комутатор, мережеві адаптери були автоматично переведені в нерозбірливий режим (Promiscuous mode).

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				55
Змн.	Арк.	№ докум.	Підпис	Дата		

3.3.2. Завантаження та налаштування правил ET Open

Процес інтеграції здійснювався через вбудований менеджер завантажень у модулі Intrusion Detection вебінтерфейсу OPNsense. Для оптимізації використання апаратних ресурсів (зокрема оперативної пам'яті) та мінімізації кількості хибнопозитивних спрацьовувань (False Positives), мною було прийнято рішення не завантажувати абсолютно всі доступні правила, а здійснити вибіркове налаштування. Було активовано лише ті категорії сигнатур, які безпосередньо відповідають архітектурі нашого ешелонованого стенду та профілю потенційних загроз.

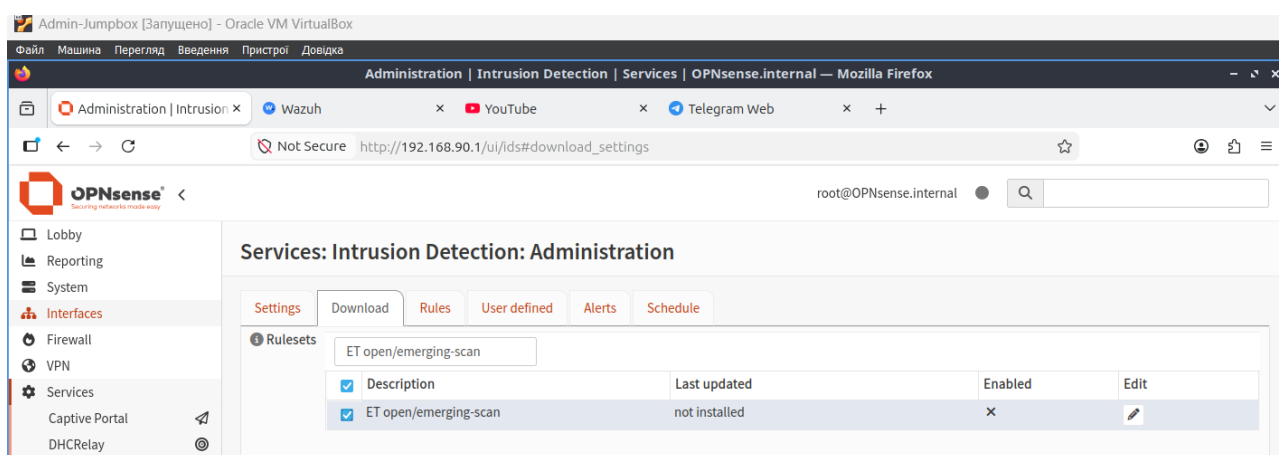


Рисунок 3.26 – Завантаження ET emerging-scan

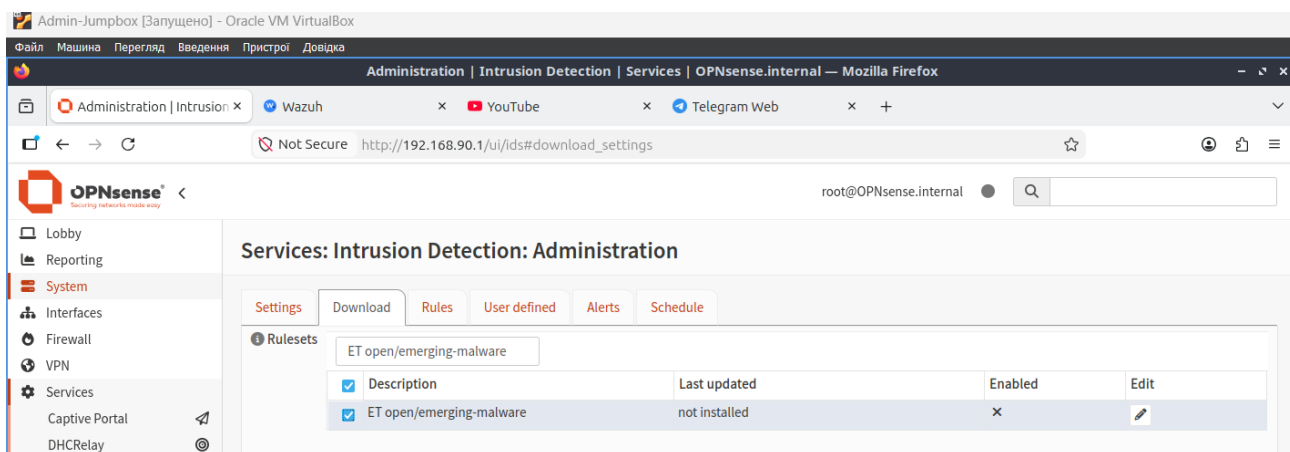


Рисунок 3.27 – Завантаження ET emerging-malware

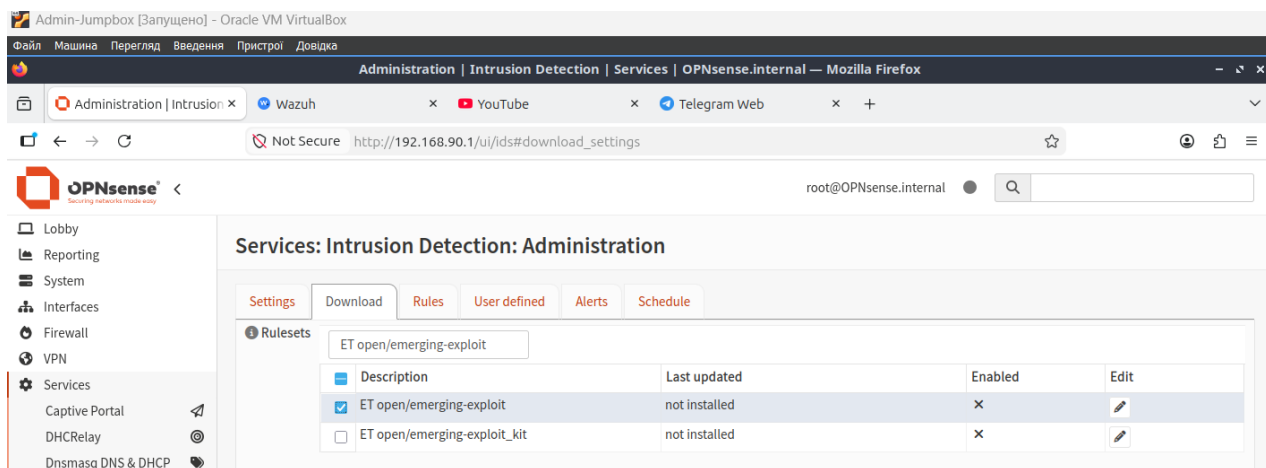


Рисунок 3.28 – Завантаження ET emerging-exploit

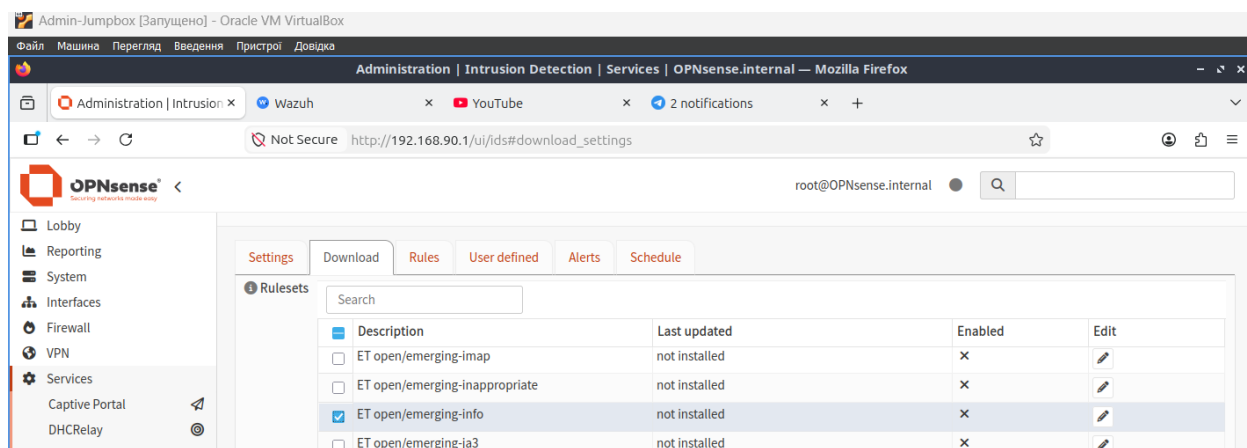


Рисунок 3.29 – Завантаження ET emerging-info

До завантаження та постійного оновлення було включено наступні критичні набори правил (Rulesets):

- **ET open/emerging-scan** – набір сигнатур, орієнтований на виявлення спроб мережної розвідки. Він дозволяє фіксувати активність таких інструментів як Nmap, Masscan або Zmap, ідентифікуючи нетипові патерни TCP-рукоштовань та спроби сканування портів на зовнішньому інтерфейсі шлюзу.

- **ET open/emerging-exploit** – база даних для виявлення спроб експлуатації відомих вразливостей (CVE) у мережевих протоколах та вебдодатках. Ці правила аналізують корисне навантаження (Payload) пакетів на наявність специфічних послідовностей байтів, характерних для відомих експлойтів.

- **ET open/emerging-malware** – набір для ідентифікації активності шкідливого програмного забезпечення, включаючи спроби комунікації локальних вузлів із командними центрами ботнетів (C&C серверами), а також виявлення

сигнатур відомих троянів та програм-вимагачів (Ransomware) у транзитному трафіку.

– **ET open/emerging-info** – категорія інформаційних правил, яка фіксує нетипові або підозрілі події, що самі по собі не є атакою, але можуть свідчити про порушення корпоративних політик безпеки (наприклад, використання нестандартних протоколів або витік чутливих даних).

Після активації обраних категорій було ініційовано процес завантаження баз даних (Download & Update Rules). Система успішно зв'язалася з серверами Proofpoint (розробниками ET Open), завантажила останні версії списків та автоматично розпакувала їх у робочу директорію рушія Suricata. Це забезпечило нашу систему виявлення вторгнень актуальною аналітичною базою для здійснення глибокого інспектування пакетів (DPI) у режимі реального часу.

3.3.3. Конфігурація логування у форматі EVE JSON

За замовчуванням більшість NIDS генерують події у форматі звичайного неструктурованого тексту (Fast log), що суттєво ускладнює їх подальший автоматизований аналіз. Враховуючи те, що архітектура нашого тестового полігону передбачає подальшу інтеграцію шлюзу із системою управління інформацією та подіями безпеки (SIEM), класичний підхід до логування є неефективним.

Для вирішення цього завдання я активував та налаштував логування подій Suricata у спеціалізованому форматі EVE JSON (Extensible Event Format). На відміну від звичайних текстових файлів, EVE представляє кожен мережевий подію як структурований JSON-об'єкт із чіткою ієрархією пар "ключ-значення". Цей формат фіксує не лише сам факт спрацювання сигнатури (Alert), але й глибокий контекст мережної взаємодії: IP-адреси джерела та призначення, порти, протоколи прикладного рівня, вміст HTTP-заголовків, DNS-запити та хеші переданих файлів.

У рамках налаштування програмного маршрутизатора OPNsense, механізм генерації Eve Json забезпечує запис ідентифікованих аномалій до локального файлу /var/log/suricata/eve.json. Такий підхід до стандартизації логів створює надійний фундамент для побудови централізованого моніторингу. Згенеровані у форматі

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		58

JSON події ідеально підходять для автоматичного парсингу та нормалізації агентами збору логів (Log Shippers), що дозволяє миттєво експортувати дані безпеки до аналітичного ядра SIEM-системи без необхідності написання складних регулярних виразів для розбору текстових рядків. Таким чином, конфігурація Eve Jsn є обов'язковою сполучною ланкою між ешелонам периметрового захисту та ешелонам моніторингу і реагування на інциденти.

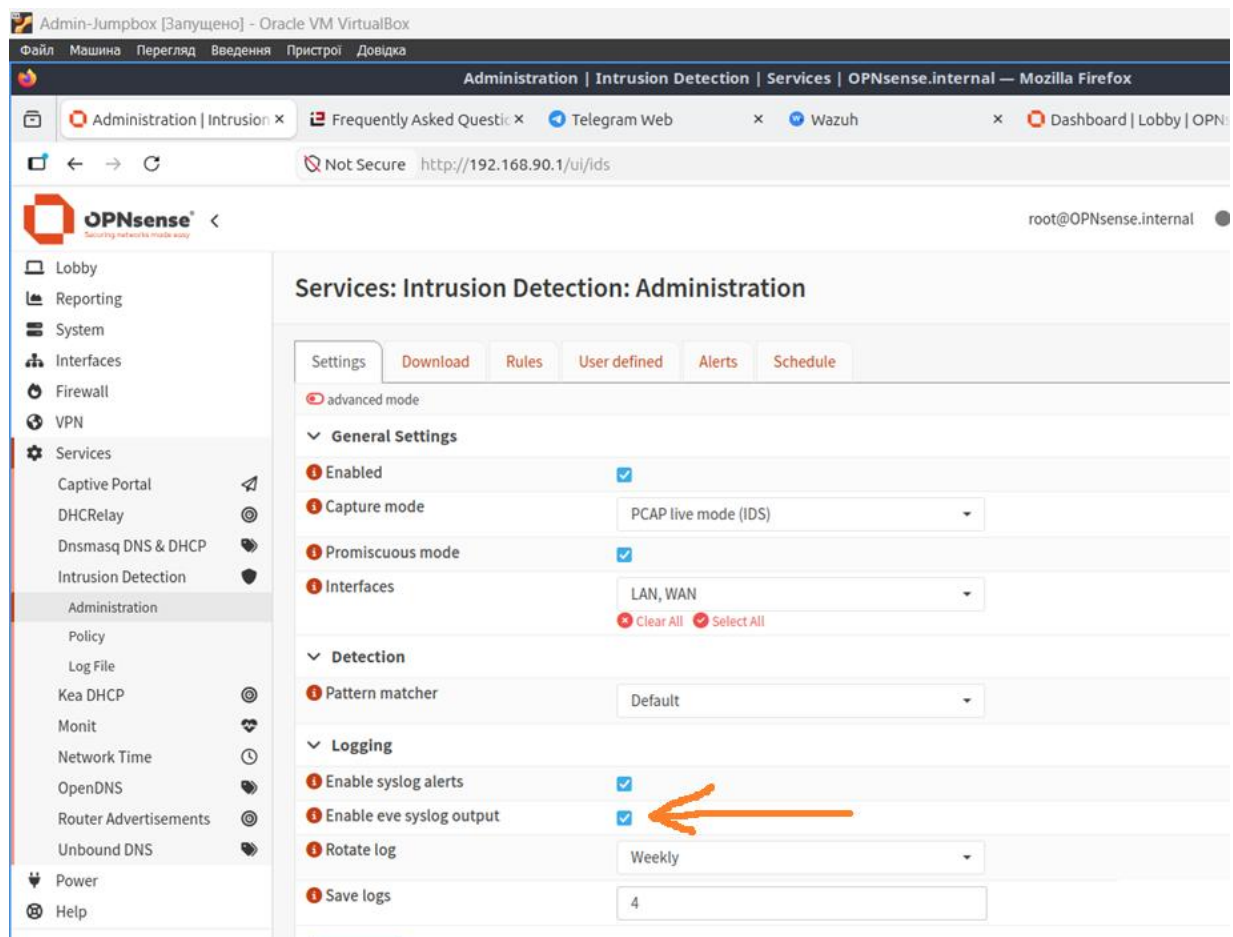


Рисунок 3.30 – Увімкнення логування EVE Jsn

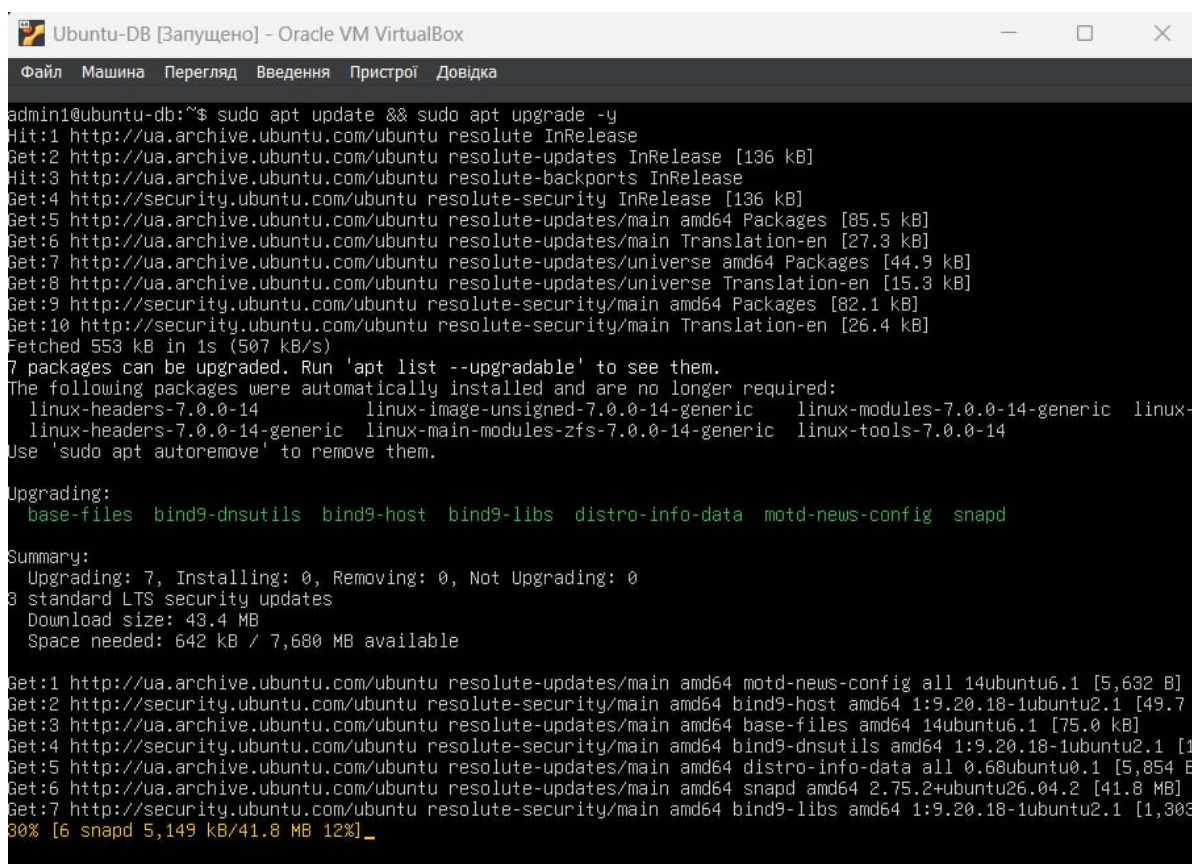
3.4. Налаштування захисту сервера Ubuntu та PostgreSQL

Після успішного розгортання периметрового ешелону захисту на базі OPNsense, наступним кроком стала реалізація принципу глибоко ешелонованого захисту (Defense-in-Depth). Цей підхід передбачає, що безпека не повинна обмежуватися лише зовнішнім шлюзом: кожен внутрішній вузол інфраструктури повинен мати власні механізми самозахисту. Основним об'єктом для налаштування став сервер баз даних під управлінням операційної системи Ubuntu Server.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				59
Змн.	Арк.	№ докум.	Підпис	Дата		

3.4.1. Базове підсилення Ubuntu Server (Hardening)

Процес підсилення (Hardening) цільового сервера розпочався з оновлення системних компонентів. Використання застарілого програмного забезпечення є одним із найпоширеніших векторів атак, тому першочерговим завданням стало синхронізація локальних репозиторіїв та встановлення актуальних патчів безпеки для ядра ОС і системних бібліотек.



```
admini@ubuntu-db:~$ sudo apt update && sudo apt upgrade -y
Hit:1 http://ua.archive.ubuntu.com/ubuntu resolute InRelease
Get:2 http://ua.archive.ubuntu.com/ubuntu resolute-updates InRelease [136 kB]
Hit:3 http://ua.archive.ubuntu.com/ubuntu resolute-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu resolute-security InRelease [136 kB]
Get:5 http://ua.archive.ubuntu.com/ubuntu resolute-updates/main amd64 Packages [85.5 kB]
Get:6 http://ua.archive.ubuntu.com/ubuntu resolute-updates/main Translation-en [27.3 kB]
Get:7 http://ua.archive.ubuntu.com/ubuntu resolute-updates/universe amd64 Packages [44.9 kB]
Get:8 http://ua.archive.ubuntu.com/ubuntu resolute-updates/universe Translation-en [15.3 kB]
Get:9 http://security.ubuntu.com/ubuntu resolute-security/main amd64 Packages [82.1 kB]
Get:10 http://security.ubuntu.com/ubuntu resolute-security/main Translation-en [26.4 kB]
Fetched 553 kB in 1s (507 kB/s)
7 packages can be upgraded. Run 'apt list --upgradable' to see them.
The following packages were automatically installed and are no longer required:
  linux-headers-7.0.0-14          linux-image-unsigned-7.0.0-14-generic  linux-modules-7.0.0-14-generic  linux-
  linux-headers-7.0.0-14-generic  linux-main-modules-zfs-7.0.0-14-generic  linux-tools-7.0.0-14
Use 'sudo apt autoremove' to remove them.

Upgrading:
  base-files  bind9-dnsutils  bind9-host  bind9-libs  distro-info-data  motd-news-config  snapd

Summary:
  Upgrading: 7, Installing: 0, Removing: 0, Not Upgrading: 0
  3 standard LTS security updates
  Download size: 43.4 MB
  Space needed: 642 kB / 7,680 MB available

Get:1 http://ua.archive.ubuntu.com/ubuntu resolute-updates/main amd64 motd-news-config all 14ubuntu6.1 [5,632 B]
Get:2 http://security.ubuntu.com/ubuntu resolute-security/main amd64 bind9-host amd64 1:9.20.18-1ubuntu2.1 [49.7 kB]
Get:3 http://ua.archive.ubuntu.com/ubuntu resolute-updates/main amd64 base-files amd64 14ubuntu6.1 [75.0 kB]
Get:4 http://security.ubuntu.com/ubuntu resolute-security/main amd64 bind9-dnsutils amd64 1:9.20.18-1ubuntu2.1 [1.1 kB]
Get:5 http://ua.archive.ubuntu.com/ubuntu resolute-updates/main amd64 distro-info-data all 0.68ubuntu0.1 [5,854 B]
Get:6 http://ua.archive.ubuntu.com/ubuntu resolute-updates/main amd64 snapd amd64 2.75.2+ubuntu26.04.2 [41.8 MB]
Get:7 http://security.ubuntu.com/ubuntu resolute-security/main amd64 bind9-libs amd64 1:9.20.18-1ubuntu2.1 [1,303
30% [6 snapd 5,149 kB/41.8 MB 12%]
```

Рисунок 3.31 – Оновлення ОС

Після оновлення системи було встановлено набір базових утиліт та сервісів безпеки, серед яких ключову роль відіграє служба auditd (Linux Audit Daemon). Встановлення цього компонента є критично важливим для подальшої інтеграції із SIEM-системою, оскільки він забезпечує низькорівневий моніторинг системних викликів, цілісності файлів (FIM) та дій користувачів на рівні ядра операційної системи.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				60
Змн.	Арк.	№ докум.	Підпис	Дата		

```

admin1@ubuntu-db:~$ sudo apt install ufw auditd curl net-tools -y
ufw is already the newest version (0.36.2-9build1).
ufw set to manually installed.
curl is already the newest version (8.18.0-1ubuntu2.1).
curl set to manually installed.
The following packages were automatically installed and are no longer required:
  linux-headers-7.0.0-14      linux-image-unsigned-7.0.0-14-generic  linux-modules-7.0.0-14-generic  linux-t
  linux-headers-7.0.0-14-generic  linux-main-modules-zfs-7.0.0-14-generic  linux-tools-7.0.0-14
Use 'sudo apt autoremove' to remove them.

Installing:
  auditd net-tools

Installing dependencies:
  libauparse0t64 libauplugin1

Suggested packages:
  audispd-plugins
  
```

Рисунок 3.32 – Встановлення служби auditd

Для забезпечення нормальної мережної ізоляції сервера бази даних було активовано вбудований міжмережевий екран ОС Linux – UFW (Uncomplicated Firewall). Незважаючи на наявність апаратного фаєрвола OPNsense на периметрі мережі, використання локального UFW реалізує концепцію глибоко ешелонованого захисту (Defense in Depth) та гарантує захист від внутрішніх загроз у разі горизонтального переміщення злоумисника (Lateral Movement) всередині сегмента SecureLAN.

Політика UFW була налаштована за принципом мінімальних привілеїв (Default Deny): весь вхідний трафік блокується за замовчуванням, за винятком явно дозволених портів – 22/TCP для захищеного SSH-доступу адміністратора та 5432/TCP для роботи служби СУБД.

```

admin1@ubuntu-db:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
admin1@ubuntu-db:~$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
admin1@ubuntu-db:~$ sudo ufw allow 22/tcp
Rules updated
Rules updated (v6)
admin1@ubuntu-db:~$ sudo ufw allow 5432/tcp
Rules updated
Rules updated (v6)
admin1@ubuntu-db:~$ sudo ufw enable
Firewall is active and enabled on system startup
admin1@ubuntu-db:~$
  
```

Рисунок 3.33 – Налаштування UFW

Під час подальшого аудиту конфігурації фаєрвола за допомогою команди `ufw status numbered` було виявлено, що стандартне правило для порту 5432 відкриває доступ з будь-якої IP-адреси (Anywhere). Для мінімізації поверхні атаки (Attack Surface) це глобальне правило було видалено. Натомість було створено суворе цільове правило, яке дозволяє підключення до бази даних виключно з довіреної адміністративної підмережі 192.168.90.0/24.

```

admin1@ubuntu-db: ~
22/tcp                ALLOW IN    Anywhere
5432/tcp              ALLOW IN    Anywhere
5432/tcp              ALLOW IN    192.168.90.0/24
22/tcp (v6)          ALLOW IN    Anywhere (v6)
5432/tcp (v6)         ALLOW IN    Anywhere (v6)

admin1@ubuntu-db:~$ sudo ufw status numbered
Status: active

      To      Action     From
      --      -
[ 1] 22/tcp    ALLOW IN   Anywhere
[ 2] 5432/tcp  ALLOW IN   Anywhere
[ 3] 5432/tcp  ALLOW IN   192.168.90.0/24
[ 4] 22/tcp (v6) ALLOW IN   Anywhere (v6)
[ 5] 5432/tcp (v6) ALLOW IN   Anywhere (v6)

admin1@ubuntu-db:~$ sudo ufw delete 5
Deleting:
allow 5432/tcp
Proceed with operation (y/n)? y
Rule deleted (v6)
admin1@ubuntu-db:~$ sudo ufw delete 2
Deleting:
allow 5432/tcp
Proceed with operation (y/n)? y
Rule deleted
admin1@ubuntu-db:~$ sudo ufw status numbered
Status: active

      To      Action     From
      --      -
[ 1] 22/tcp    ALLOW IN   Anywhere
[ 2] 5432/tcp  ALLOW IN   192.168.90.0/24
[ 3] 22/tcp (v6) ALLOW IN   Anywhere (v6)

```

Рисунок 3.34 – Аудит та посилення політик UFW: обмеження доступу до СУБД конкретною підмережею

Для верифікації ефективності налаштованих мережеских політик було проведено контрольне сканування портів сервера Ubuntu-DB з вузла адміністратора за допомогою популярного мережевого сканера Nmap.

Результати сканування практично довели правильність конфігурації фаєрвола. Утиліта Nmap підтвердила, що легітимні порти (22 та 5432) відкриті для адміністративної підмережі, тоді як усі інші несанкціоновані запити (наприклад, до портів 80 або 3389) жорстко блокуються фаєрволом, про що свідчить статус `filtered` (пакети відкидаються без відповіді атакуючому).

```

admin@admin-jumpbox:~$ nmap -Pn -p 22,80,3389,5432 192.168.90.23
Starting Nmap 7.98 ( https://nmap.org ) at 2026-06-03 11:32 +0300
Nmap scan report for 192.168.90.23
Host is up (0.00039s latency).

PORT      STATE      SERVICE
22/tcp    open       ssh
80/tcp    filtered   http
3389/tcp   filtered   ms-wbt-server
5432/tcp   open       postgresql

Nmap done: 1 IP address (1 host up) scanned in 1.73 seconds
admin@admin-jumpbox:~$

```

Рисунок 3.35 – Перевірка для внутрішнього користувача

```

admin@admin-jumpbox:~$ nmap -Pn -p 22,80,3389,5432 192.168.90.23
Starting Nmap 7.98 ( https://nmap.org ) at 2026-06-03 11:34 +0300
Nmap scan report for 192.168.90.23
Host is up (0.00039s latency).

PORT      STATE      SERVICE
22/tcp    open       ssh
80/tcp    filtered   http
3389/tcp   filtered   ms-wbt-server
5432/tcp   filtered   postgresql

Nmap done: 1 IP address (1 host up) scanned in 1.73 seconds
admin@admin-jumpbox:~$

```

Рисунок 3.36 – Перевірка для зовнішнього користувача

3.4.2. Встановлення та захист СУБД PostgreSQL, TLS

Наступним етапом стало розгортання самої системи управління базами даних PostgreSQL. Після встановлення базових пакетів (postgresql та postgresql-contrib) було проведено конфігурацію системи з урахуванням сучасних стандартів безпеки.

```

admin@ubuntu-db:~$ sudo apt install postgresql postgresql-contrib -y
Note, selecting 'postgresql' instead of 'postgresql-contrib'
The following packages were automatically installed and are no longer required:
  linux-headers-7.0.0-14 linux-image-unsigned-7.0.0-14-generic linux-modules-7.0.0-14-generic linux-headers-7.0.0-14-generic linux-main-modules-zfs-7.0.0-14-generic linux-tools-7.0.0-14
Use 'sudo apt autoremove' to remove them.

Installing:
  postgresql

Installing dependencies:
  libcommon-sense-perl libjson-xs-perl libtypes-serialiser-perl postgresql-18 postgresql-client-18
  libjson-perl libpq5 liburing2 postgresql-18-jit postgresql-client-common

```

Рисунок 3.37 – Встановлення postgresql

Для уникнення використання стандартного системного профілю postgres під час роботи додатків, у системі було створено окрему базу даних secure_data та виділеного адміністративного користувача db_admin. Цьому користувачеві було

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				63
Змн.	Арк.	№ докум.	Підпис	Дата		

делеговано гранулярні права доступу виключно до створеної бази, що мінімізує ризики у випадку гіпотетичної SQL-ін'єкції або компрометації облікових даних.

```

admin1@ubuntu-db:~$ sudo -u postgres psql
psql (18.3 (Ubuntu 18.3-1))
Type "help" for help.

postgres=# CREATE DATABASE secure_data;
CREATE DATABASE
postgres=# CREATE USER db_admin WITH PASSWORD 'SuperSecretPass123!';
CREATE ROLE
postgres=# GRANT ALL PRIVILEGES ON DATABASE secure_data TO db_admin
postgres-# ;
GRANT
postgres=# \q
admin1@ubuntu-db:~$ _
  
```

Рисунок 3.38 – Створення бази даних

Особлива увага була приділена мережевій конфігурації СУБД. У головному конфігураційному файлі `postgresql.conf` параметр `listen_addresses` був налаштований для прослуховування мережеских інтерфейсів. Однак, щоб унеможливити несанкціонований доступ, жорсткі обмеження були застосовані на рівні файлу контролю доступу клієнтів `pg_hba.conf` (Host-Based Authentication).

Окремим і надзвичайно важливим етапом захисту бази даних, який я реалізував у межах роботи, стало налаштування примусового шифрування мережевого трафіку (TLS). Навіть за умови використання надійних паролів та мережеских екранів, дані між клієнтом та СУБД (включаючи самі SQL-запити та результати їх виконання) за замовчуванням можуть передаватися локальною мережею у відкритому вигляді. Це створює ризик перехоплення інформації у випадку компрометації будь-якого вузла всередині сегмента.

Щоб унеможливити такі атаки, я налаштував базу даних таким чином, щоб вона відхиляла будь-які запити без шифрування. Для цього я вніс зміни до конфігураційного файлу політик доступу `pg_hba.conf`. Стандартну директиву `host`, яка дозволяє як зашифровані, так і звичайні з'єднання, я замінив на `hostssl` для цільової підмережі адміністратора. Після цього я перезапустив службу PostgreSQL, щоб застосувати нові правила безпеки.

```
# Allow local connections:
host all all ::1/128 scram-sha-256
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all peer
host replication all 127.0.0.1/32 scram-sha-256
host replication all ::1/128 scram-sha-256
hostssl secure_data db_admin 192.168.90.0/24 scram-sha-256
```

Рисунок 3.39 – Результат зміни конфігураційного файлу

3.4.3. Зміцнення безпеки віддаленого адміністрування (SSH Hardening) та впровадження криптографічної автентифікації

Для безпечного управління інфраструктурою традиційна автентифікація за допомогою паролів вважається вразливою до атак типу Brute-Force (перебір словником) та перехоплення (Sniffing). З метою забезпечення найвищого рівня безпеки адміністративного доступу було прийнято рішення повністю перевести взаємодію між адміністративним вузлом (Admin-Jumpbox) та цільовими серверами (зокрема Ubuntu-DB) на криптографічну автентифікацію за допомогою відкритих ключів (Public Key Authentication).

Фінальна перевірка продемонструвала бездоганну роботу механізму: при спробі підключення з вузла Admin-Jumpbox до сервера Ubuntu-DB авторизація пройшла миттєво і безпечно, використовуючи створений ключ Ed25519, без жодних запитів на введення текстового пароля.

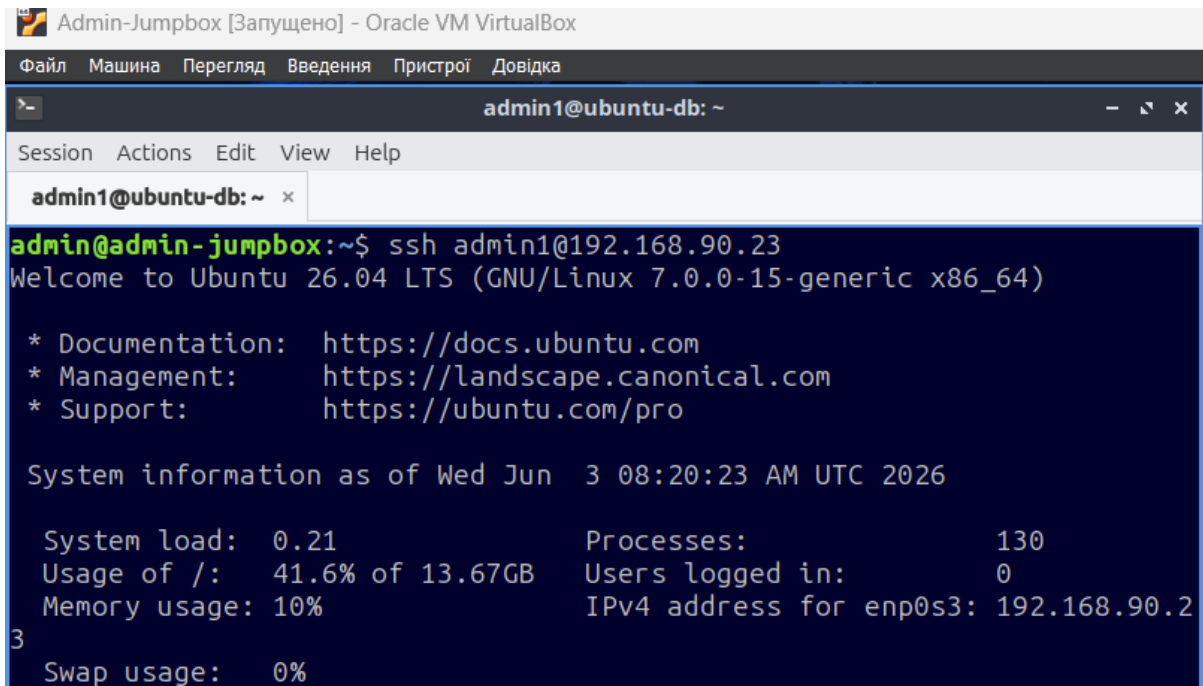
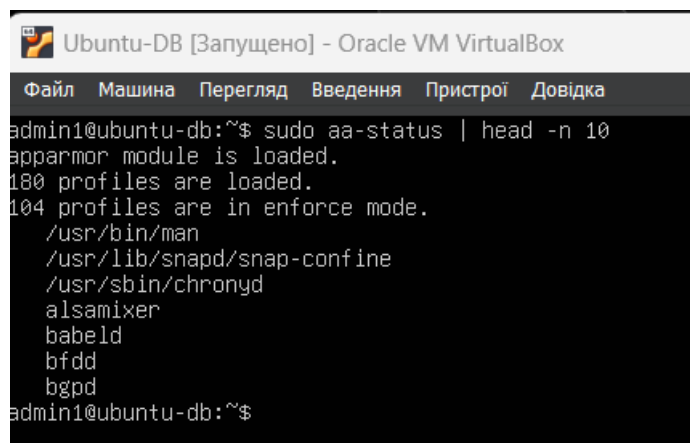


Рисунок 3.40 – Успішна криптографічна автентифікація та отримання віддаленого доступу без використання пароля

3.4.4. Налаштування механізмів мандатного контролю доступу AppArmor та підсистеми аудиту auditd

Для забезпечення проактивного захисту цільового сервера на рівні ядра операційної системи було впроваджено механізми мандатного контролю доступу (Mandatory Access Control, MAC) та деталізованого системного логування.



```
admin1@ubuntu-db:~$ sudo aa-status | head -n 10
apparmor module is loaded.
180 profiles are loaded.
104 profiles are in enforce mode.
/usr/bin/man
/usr/lib/snapd/snap-confine
/usr/sbin/chronyd
alsamixer
babeld
bfd
bgpd
admin1@ubuntu-db:~$
```

Рисунок 3.41 – Налаштований AppArmor

Фінальна перевірка утилітою `auditctl -l` підтвердила успішне завантаження створених правил у пам'ять ядра. Тепер усі зафіксовані аномалії зберігаються у структурованому форматі, повністю підготовленому для збору та нормалізації агентом безпеки Wazuh.

3.4.5. Підключення pgcrypto для шифрування чутливих стовпців

У сучасних системах управління базами даних захист інформації лише на рівні мережевого транспорту (TLS) та контролю доступу до файлової системи (AppArmor) не є вичерпним. Для забезпечення безпеки даних у стані спокою (Data at Rest) було застосовано механізм криптографічного перетворення інформації безпосередньо на рівні стовпців таблиць. З цією метою в СУБД PostgreSQL було активовано офіційне розширення `pgcrypto`.

Для легітимного отримання доступу до інформації необхідно виконати зворотне криптографічне перетворення на стороні СУБД. Це було продемонстровано за допомогою функції `pgp_sym_decrypt`. Запит був сформований таким чином, щоб розшифрувати конкретний рядок за умови передачі правильного симетричного ключа, який був використаний під час створення запису.

```
secure_data=# SELECT
secure_data=# id,
secure_data=# full_name,
secure_data=# pgp_sym_decrypt(secret_data, 'UareBreakingMyBales') AS decrypted_info
secure_data=# FROM clients_confidential
secure_data=# WHERE full_name = 'TheyKilledKenny';
 id |      full_name      |      decrypted_info
-----+-----+-----
  3 | TheyKilledKenny | Medical_Info: BloodType -, Diagnos: NotHelthy
(1 row)

secure_data=#
```

Рисунок 3.42 – Читання чутливої інформації

3.4.6. Активація pgAudit та налаштування рівнів аудиту

Для забезпечення повноцінної видимості подій безпеки всередині бази даних стандартного системного аудиту ОС було недостатньо. З метою глибокого моніторингу сесій, доступу до таблиць та виконання SQL-запитів мною було розгорнуто офіційне розширення PostgreSQL Audit Extension (pgAudit).

```
wazuh@ubuntu-wazuh-siem:~$ sudo cat /var/ossec/logs/alerts/alerts.log | grep "100002"
Rule: 100002 (level 5) -> 'PostgreSQL Security: Alert, Zvernennya do konfedenciynih dannyh (pgAudit)'
Rule: 100002 (level 5) -> 'PostgreSQL Security: Alert, Zvernennya do konfedenciynih dannyh (pgAudit)'
Rule: 100002 (level 5) -> 'PostgreSQL Security: Alert, Zvernennya do konfedenciynih dannyh (pgAudit)'
Rule: 100002 (level 5) -> 'PostgreSQL Security: Alert, Zvernennya do konfedenciynih dannyh (pgAudit)'
Rule: 100002 (level 5) -> 'PostgreSQL Security: Alert, Zvernennya do konfedenciynih dannyh (pgAudit)'
Rule: 100002 (level 5) -> 'PostgreSQL Security: Зафіксовано звернення до конфіденційних даних (pgAudit)'
Rule: 100002 (level 5) -> 'PostgreSQL Security: Зафіксовано звернення до конфіденційних даних (pgAudit)'
Rule: 100002 (level 5) -> 'PostgreSQL Security: Зафіксовано звернення до конфіденційних даних (pgAudit)'
Rule: 100002 (level 5) -> 'PostgreSQL Security: Зафіксовано звернення до конфіденційних даних (pgAudit)'
Rule: 100002 (level 5) -> 'PostgreSQL Security: Зафіксовано звернення до конфіденційних даних (pgAudit)'
Rule: 100002 (level 5) -> 'PostgreSQL Security: Зафіксовано звернення до конфіденційних даних (pgAudit)'
wazuh@ubuntu-wazuh-siem:~$
```

Рисунок 3.43 – Робота pgaudit, приклад отримання логів

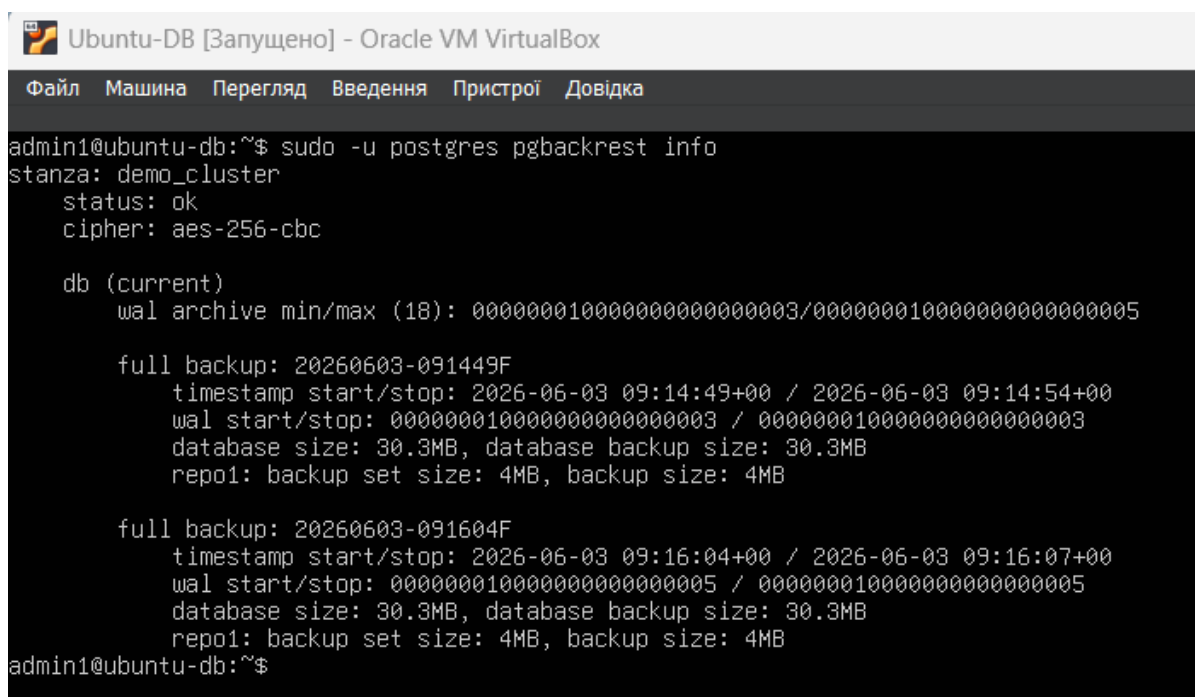
Демонстрація роботи та взаємодії з Wazuh буде в розділі про роботу самої SIEM-системи та Threat Hunting

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		67

3.4.7. Конфігурація pgBackRest для резервного копіювання

Для забезпечення безперервності процесів та захисту інформації від втрати внаслідок апаратних збоїв або атак програм-вимагачів (Ransomware), систему захисту СУБД було доповнено механізмом регулярного резервного копіювання. З цією метою мною було обрано спеціалізовану утиліту pgBackRest, яка підтримує паралельну обробку даних, інкрементальні копії та, найголовніше, криптографічний захист архівів.

Для перевірки результатів було виконано діагностичну команду `pgbackrest info`. Вивід системи підтвердив успішне створення архівів, їх розмір, а також активний статус алгоритму шифрування `aes-256-cbc`. Налаштована інфраструктура дозволяє гарантовано відновити інформацію навіть у випадку повного знищення оригінальних файлів бази даних зломисником.



```
admin1@ubuntu-db:~$ sudo -u postgres pgbackrest info
stanza: demo_cluster
status: ok
cipher: aes-256-cbc

db (current)
wal archive min/max (18): 000000010000000000000003/000000010000000000000005

full backup: 20260603-091449F
timestamp start/stop: 2026-06-03 09:14:49+00 / 2026-06-03 09:14:54+00
wal start/stop: 000000010000000000000003 / 000000010000000000000003
database size: 30.3MB, database backup size: 30.3MB
repo1: backup set size: 4MB, backup size: 4MB

full backup: 20260603-091604F
timestamp start/stop: 2026-06-03 09:16:04+00 / 2026-06-03 09:16:07+00
wal start/stop: 000000010000000000000005 / 000000010000000000000005
database size: 30.3MB, database backup size: 30.3MB
repo1: backup set size: 4MB, backup size: 4MB
admin1@ubuntu-db:~$
```

Рисунок 3.44 – Верифікація створених резервних копій та статусу їх шифрування

3.5. Розгортання підсистеми моніторингу

Для забезпечення централізованого збору, нормалізації та аналізу подій безпеки (Log Management & Threat Detection) з усіх вузлів інфраструктури мною було обрано рішення з відкритим вихідним кодом Wazuh. Ця платформа поєднує в собі можливості XDR (Extended Detection and Response) та SIEM, що дозволяє

ефективно виявляти аномалії, контролювати цілісність файлів та реагувати на загрози в режимі реального часу.

3.5.1. Інсталяція Wazuh Manager та стека Elastic (Wazuh All-in-One)

Зважаючи на масштаби розробленої тестової інфраструктури, для розгортання центрального сервера моніторингу було обрано архітектуру типу «All-in-One» (Все-в-одному). Цей підхід передбачає встановлення всіх ключових компонентів системи – сервера управління (Wazuh Manager), системи індексації даних (Wazuh Indexer) та графічного інтерфейсу (Wazuh Dashboard) – на єдиний виділений віртуальний вузол Ubuntu-Wazuh-SIEM.

Процес розгортання розпочався з базової підготовки операційної системи. Мною було перевірено мережеву конфігурацію сервера (йому призначено статичну IP-адресу 192.168.90.74 у сегменті управління), після чого виконано комплексне оновлення пакетної бази системи за допомогою утиліти apt, а також встановлено необхідний пакет curl для завантаження інсталяційних скриптів.

Успішність розгортання була підтверджена шляхом доступу до вебінтерфейсу системи через браузер з машини адміністратора (Admin-Jumpbox). З'єднання було встановлено за захищеним протоколом HTTPS за адресою сервера моніторингу.

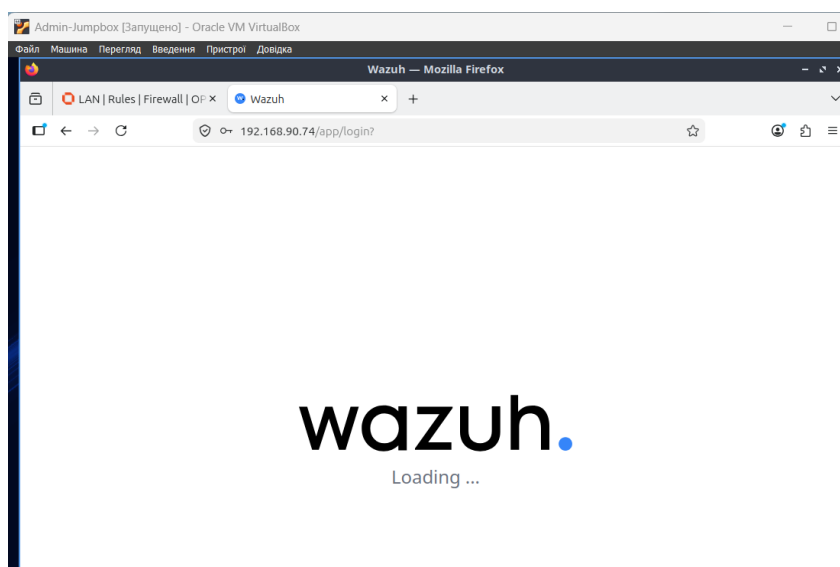


Рисунок 3.45 – Завантаження веб інтерфейсу

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		69

3.5.2. Встановлення Wazuh-агентів на захищувані вузли

Для того щоб центральний сервер управління (Wazuh Manager) міг отримувати та аналізувати події з цільового сервера бази даних, необхідно було розгорнути на ньому спеціалізоване програмне забезпечення – Wazuh Agent. Цей легковаговий процес (демон) працює у фоновому режимі, локально збирає системні журнали, дані про цілісність файлів (FIM), аномалії та події безпеки (зокрема від налаштованих раніше auditd та pgAudit), після чого через зашифрований канал зв'язку передає їх на центральний сервер.

Для мінімізації можливих помилок під час конфігурації я використав майстер автоматизованого розгортання агентів (Deploy new agent) безпосередньо у вебінтерфейсі Wazuh Dashboard. Задавши параметри цільового вузла – архітектуру операційної системи (Linux DEB amd64), IP-адресу менеджера (192.168.90.74) та ідентифікатор захищуваного сервера (Ubuntu-DB), – система автоматично згенерувала готовий скрипт для встановлення.

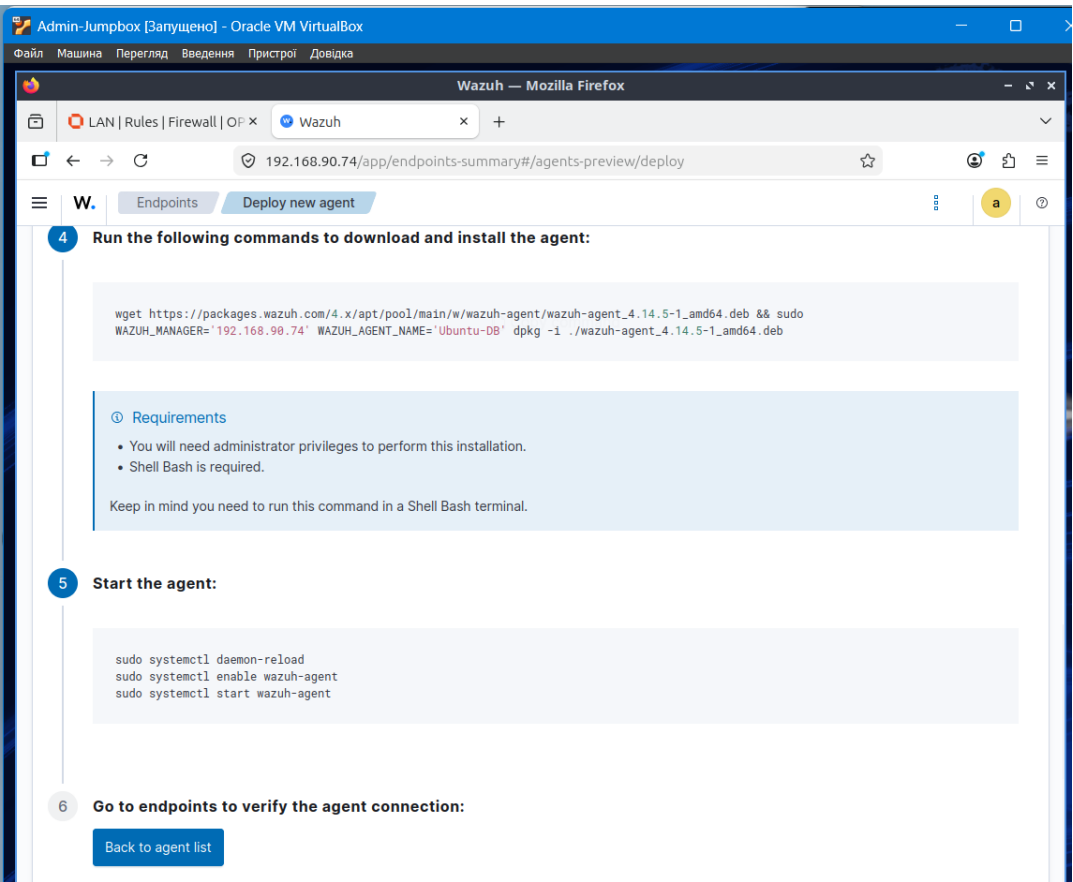


Рисунок 3.46 – Конфігурація параметрів та генерація інсталяційної команди у вебінтерфейсі Wazuh

Відразу після запуску служби агент згенерував власні криптографічні ключі, успішно пройшов процес авторизації на сервері управління Wazuh Manager та розпочав активний моніторинг цільової системи.

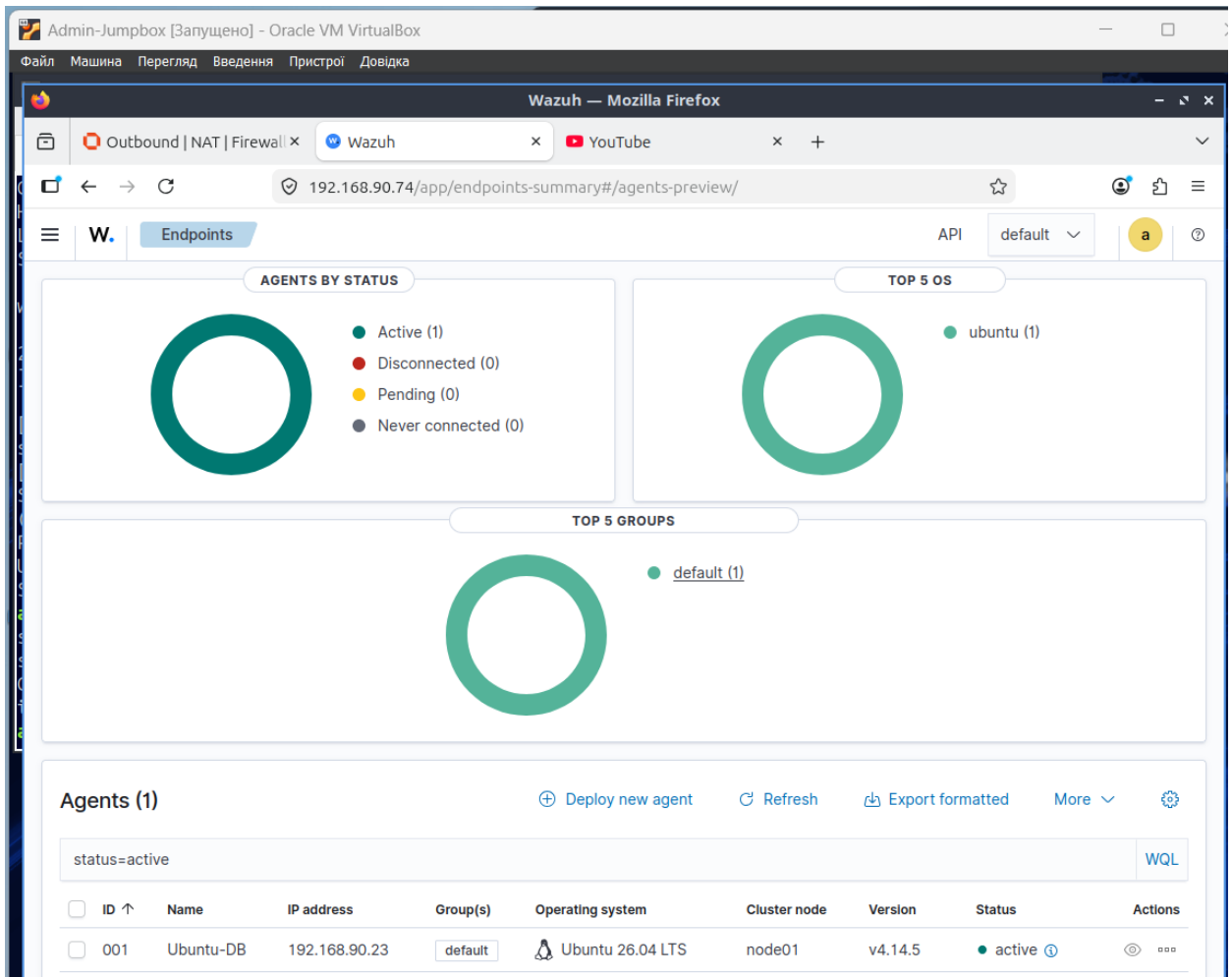


Рисунок 3.47 – Веб сторінка Wazuh

3.5.3. Розширений збір та нормалізація подій аудиту PostgreSQL у Wazuh за допомогою JSON та власних правил кореляції (pgAudit)

Механізм логування цільової бази даних було переведено у формат JSON. Це забезпечує точне розпізнавання окремих полів (дата, користувач, тип SQL-запиту, об'єкт доступу) на стороні Wazuh Manager. Одночасно, з міркувань безпеки, права доступу до файлів журналу були суворо обмежені на рівні операційної системи, щоб унеможливити їх читання неавторизованими локальними користувачами.

Ключовим етапом інтеграції стала розробка власних правил виявлення загроз (Custom Rules) на рівні SIEM-системи. Оскільки розширення pgAudit генерує специфічні події зі своїми унікальними маркерами, стандартні правила Wazuh не

зможли б їх коректно класифікувати. Для вирішення цього завдання було створено логічну ієрархію з двох взаємопов'язаних правил. Базове правило відповідає за декодування вхідного JSON-потoku та його первинну класифікацію як системного логу СУБД. Друге, дочірнє правило, здійснює глибокий аналіз повідомлення і шукає специфічні маркери спрацювання аудиту. У разі їх виявлення система автоматично підвищує рівень критичності події та формує алерт із детальним описом порушення конфіденційності.

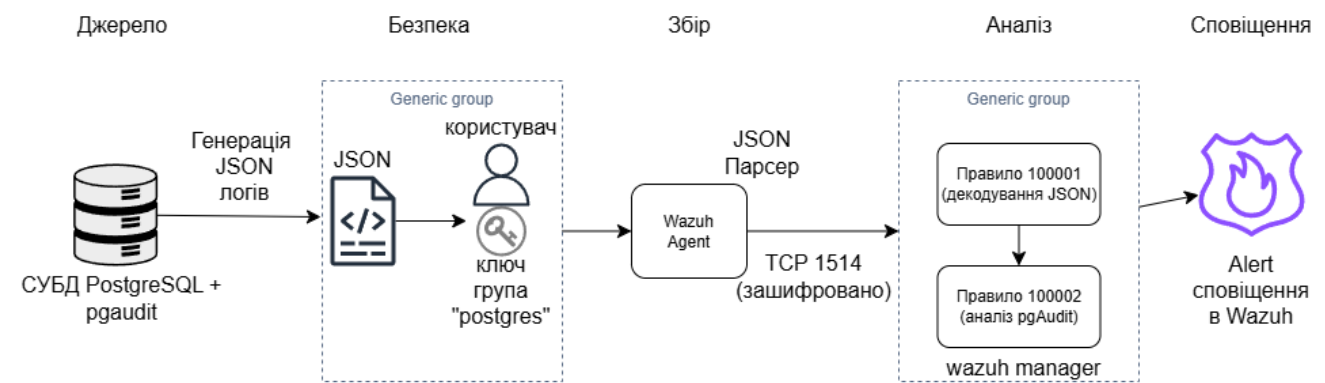


Рисунок 3.48 – Зображення діаграми роботи аудиту



Рисунок 3.49 – Створення власного правила

3.5.4. Інтеграція мережевого екрана OPNsense та IDS Suricata із системою централізованого моніторингу

Для забезпечення повної видимості інфраструктури (Full Stack Visibility) необхідно об'єднати події, що відбуваються на рівні кінцевих хостів (HIDS), із подіями мережевого рівня (NIDS). З цією метою було реалізовано архітектурну

інтеграцію головного мережевого екрана OPNsense та вбудованої в нього системи виявлення вторгнень Suricata із центральним сервером моніторингу Wazuh SIEM.

Оскільки закрита архітектура операційної системи мережевого екрана не передбачає встановлення стандартного програмного агента Wazuh, для надійної передачі журналів аудиту було використано класичний протокол системного логування Syslog. З боку сервера управління SIEM було налаштовано виділений канал зв'язку, який ініціює прослуховування мережевого порту та дозволяє приймати вхідний потік подій виключно від довіреної IP-адреси шлюзу.

Відповідні дзеркальні налаштування маршрутизації логів були виконані і на самому OPNsense. Критично важливим етапом цієї конфігурації стала сувора фільтрація вихідного трафіку: щоб не перевантажувати систему моніторингу зайвим «інформаційним шумом» від рутинних мережевих операцій, для експорту на SIEM-сервер були обрані виключно події блокування пакетів фаєрволом та сповіщення про аномалії від підсистеми виявлення вторгнень.

Оскільки система Suricata генерує специфічні сповіщення у власному форматі, для їх коректної класифікації та відображення на стороні Wazuh Manager було розроблено набір спеціалізованих правил кореляції (Custom Rules). Логіка їхньої роботи побудована за ієрархічним принципом: спочатку базове правило ідентифікує та відфільтровує всі вхідні Syslog-повідомлення, марковані тегамі IDS. Далі дочірні правила здійснюють глибокий аналіз (Deep Packet Inspection) тексту сповіщення, здійснюючи пошук критичних патернів. Наприклад, у разі фіксації маркерів, що свідчать про відповідь внутрішнього сервера на успішну експлуатацію вразливості ззовні (відомі сигнатури атак), система автоматично класифікує таку подію найвищим рівнем загрози.

Такий підхід дозволяє консолідувати розрізнені мережеві журнали в єдиній панелі управління та забезпечує аналітиків безпеки структурованою інформацією для миттєвого реагування на атаки, спрямовані на периметр інфраструктури.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				73
Змн.	Арк.	№ докум.	Підпис	Дата		

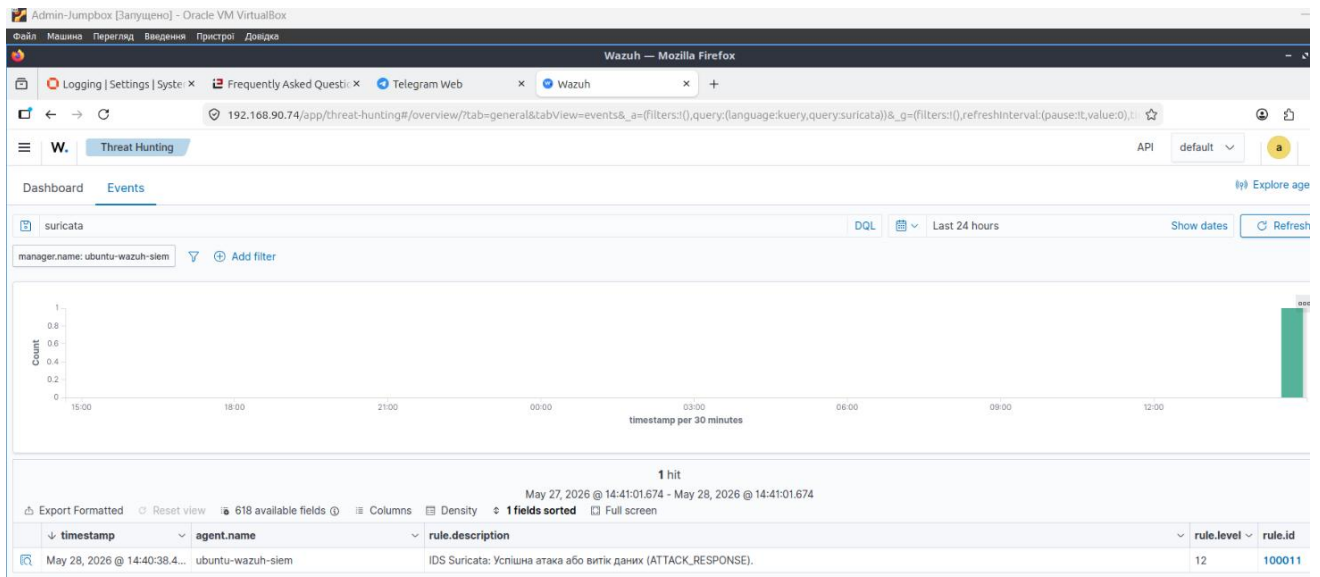


Рисунок 3.50 – Приклад налаштованого та працюючого правила

3.6. Тестування ефективності системи захисту

3.6.1. Сканування портів і виявлення розвідки (Nmap) – реакція Suricata та OPNSense

Для комплексної перевірки надійності зовнішнього периметра мережі було проведено імітацію кібератаки з використанням методів активної розвідки. Головною метою тестування була перевірка здатності міжмережевого екрана OPNSense блокувати несанкціоновані запити, а також ефективності підсистеми виявлення вторгнень Suricata та Wazuh SIEM у розпізнаванні патернів сканування.

Щоб максимально наблизити симуляцію до реальних умов (загроза з боку глобальної мережі Інтернет), атакуючий тестовий вузол було тимчасово виведено із захищеного внутрішнього сегмента LAN. За допомогою налаштувань гіпервізора тип мережевого адаптера було змінено на проміжний (Bridged Adapter). Це дозволило машині отримати зовнішню IP-адресу та взаємодіяти безпосередньо з WAN-інтерфейсом шлюзу безпеки

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				74
Змн.	Арк.	№ докум.	Підпис	Дата		

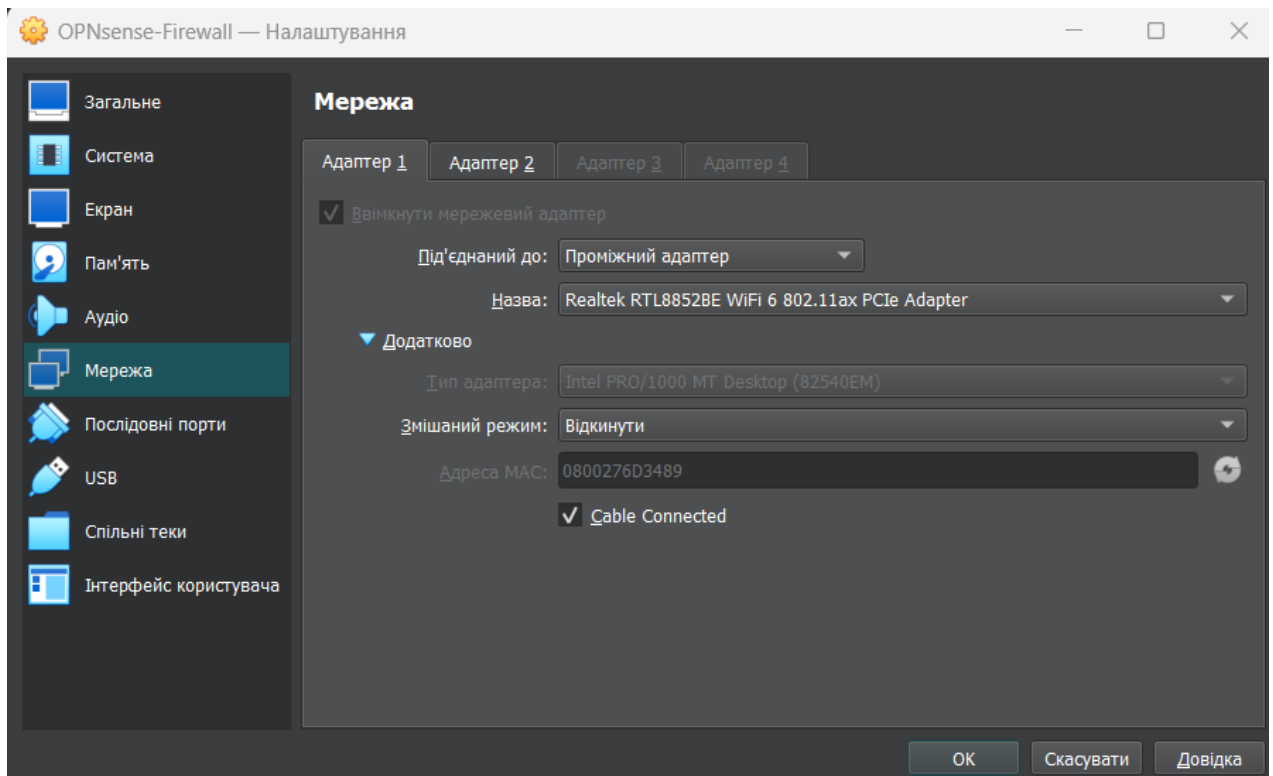


Рисунок 3.51 – Налаштування мережевих адаптерів для симуляції зовнішньої атаки

Зовнішня IP-адреса цільового мережевого екрана визначалася як 192.168.0.153. З атакуючого вузла за допомогою утиліти Nmap було ініційовано агресивне сканування транспортних портів діапазону 1–10000. Для створення максимального мережевого навантаження використовувалися параметри визначення операційної системи та версій сервісів (-A), а також підвищена швидкість сканування (-T4).

Результати роботи сканера з боку атакуючого вузла підтвердили абсолютну стійкість фаєрвола: усі 10 000 перевірених портів отримали статус filtered (відфільтровано). Це означає, що мережевий екран успішно застосовує сувору політику заборони за замовчуванням (Default Deny), просто "скидаючи" пакети без відправлення жодної відповіді зломиснику. Така конфігурація унеможливорює складання топології мережі методами зовнішньої розвідки

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				75
Змн.	Арк.	№ докум.	Підпис	Дата		

```

Admin-Jumpbox [Запущено] - Oracle VM VirtualBox
Файл  Машина  Перегляд  Введення  Пристрої  Довідка
admin@admin-jumpbox: ~
Session  Actions  Edit  View  Help
admin@admin-jumpbox: ~ x
admin@admin-jumpbox:~$ sudo nmap -A -T4 -p 1-10000 192.168.0.153
[sudo: authenticate] Password:
Starting Nmap 7.98 ( https://nmap.org ) at 2026-06-14 21:19 +0300
Stats: 0:00:03 elapsed; 0 hosts completed (0 up), 1 undergoing ARP Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Stats: 0:00:08 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 1.29% done; ETC: 21:23 (0:03:50 remaining)
Stats: 0:00:23 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 8.50% done; ETC: 21:23 (0:03:14 remaining)
Nmap scan report for 192.168.0.153
Host is up (0.00037s latency).
All 10000 scanned ports on 192.168.0.153 are in ignored states.
Not shown: 10000 filtered tcp ports (no-response)
MAC Address: 08:00:27:6D:34:89 (Oracle VirtualBox virtual NIC)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

TRACEROUTE
HOP RTT      ADDRESS
1   0.37 ms  192.168.0.153

OS and Service detection performed. Please report any incorrect results
at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 214.92 seconds
admin@admin-jumpbox:~$

```

Рисунок 3.52 – Результати агресивного сканування Nmap

Після завершення фази розвідки атакуючий вузол було повернено до внутрішньої мережі для доступу до центрального сервера моніторингу Wazuh. Аналіз зібраних журналів аудиту в модулі Threat Hunting продемонстрував, що система захисту миттєво відреагувала на загрозу. Панель управління інцидентами зафіксувала масований потік подій (понад 10 сповіщень за короткий проміжок часу), чітко ідентифікувавши джерело атаки.

Admin-Jumpbox [Запущено] - Oracle VM VirtualBox				
Wazuh — Mozilla Firefox				
Traffic Reporting OPN: x Frequently Asked Questio: x Telegram Web x Wazuh x Dashboard Lobby OPN: x WireGuard VPN OPN: x + v				
Not Secure 192.168.90.74/app/threat-hunting#/overview/?tab=general&tabView=events&_a=(filters:(),query:(language:query,query:))				
Threat Hunting API default a				
timestamp per 12 hours				
728 hits				
May 28, 2026 @ 02:56:07.624 - Jun 14, 2026 @ 21:34:08.179				
Export Formatted Reset view 654 available fields Columns Density 1 fields sorted Full screen				
	timestamp	agent.name	rule.description	rule.level rule.id
	Jun 14, 2026 @ 21:19:56.7...	ubuntu-wazuh-siem	Multiple pfSense firewall blocks events from same source.	10 87702
	Jun 14, 2026 @ 21:14:02.1...	ubuntu-wazuh-siem	IDS Suricata: Виявлено підозрілу мережеву активність.	10 100010
	Jun 14, 2026 @ 21:14:01.7...	ubuntu-wazuh-siem	IDS Suricata: Виявлено підозрілу мережеву активність.	10 100010
	Jun 14, 2026 @ 21:14:01.7...	ubuntu-wazuh-siem	IDS Suricata: Виявлено підозрілу мережеву активність.	10 100010
	Jun 14, 2026 @ 21:14:01.7...	ubuntu-wazuh-siem	IDS Suricata: Виявлено підозрілу мережеву активність.	10 100010
	Jun 14, 2026 @ 21:14:01.7...	ubuntu-wazuh-siem	IDS Suricata: Виявлено підозрілу мережеву активність.	10 100010
	Jun 14, 2026 @ 21:14:01.7...	ubuntu-wazuh-siem	IDS Suricata: Виявлено підозрілу мережеву активність.	10 100010
	Jun 14, 2026 @ 21:14:01.7...	ubuntu-wazuh-siem	IDS Suricata: Виявлено підозрілу мережеву активність.	10 100010
	Jun 14, 2026 @ 21:14:01.7...	ubuntu-wazuh-siem	IDS Suricata: Виявлено підозрілу мережеву активність.	10 100010
	Jun 14, 2026 @ 21:14:01.7...	ubuntu-wazuh-siem	IDS Suricata: Виявлено підозрілу мережеву активність.	10 100010
	Jun 14, 2026 @ 21:14:01.7...	ubuntu-wazuh-siem	IDS Suricata: Виявлено підозрілу мережеву активність.	10 100010
	Jun 14, 2026 @ 21:14:01.6...	ubuntu-wazuh-siem	IDS Suricata: Виявлено підозрілу мережеву активність.	10 100010
	Jun 14, 2026 @ 21:14:01.6...	ubuntu-wazuh-siem	IDS Suricata: Виявлено підозрілу мережеву активність.	10 100010
	Jun 14, 2026 @ 21:14:01.6...	ubuntu-wazuh-siem	IDS Suricata: Виявлено підозрілу мережеву активність.	10 100010
Rows per page: 15 < 1 2 3 4 5 ... 49 >				

Рисунок 3.53 – Відображення перехопленої мережної атаки на панелі інцидентів Wazuh

Процес кореляції подій у SIEM-системі успішно відпрацював за двома ключовими векторами

1. **Відмова в доступі на рівні фаєрвола:** Спрацювало правило з ідентифікатором **87702** (*Multiple pfSense firewall blocks events from same source*). Рівень критичності – 10. Це підтверджує автоматичне блокування серії пакетів від єдиного зовнішнього джерела підсистемою фільтрації пакетів.

2. **Виявлення аномалії на рівні IDS:** Розроблене під час архітектурного налаштування кастомне правило **100010** (*IDS Suricata: Виявлено підозрілу мережеву активність*) також спрацювало з рівнем критичності 10. Механізм глибокої інспекції пакетів (DPI) Suricata успішно розпізнав специфічні сигнатури мережевого сканера та згенерував відповідні сповіщення.

Проведене тестування наочно доводить, що спроектована архітектура захисту периметра функціонує коректно. Інфраструктура не лише ефективно відбиває спроби несанкціонованого зовнішнього доступу на мережевому рівні, але

й забезпечує адміністраторів безпеки вичерпною аналітикою інцидентів у режимі реального часу.

3.6.2. Верифікація детектування системних атак типу Brute-Force

Верифікація налаштованого механізму проводилася шляхом імітації атаки підбору паролів на сервер бази даних (Ubuntu-DB). Аналіз панелі подій Wazuh Dashboard продемонстрував ідеальне спрацювання ланцюга автоматизованого захисту:

1. Система зафіксувала спробу входу з неіснуючим іменем користувача (правило 5710).
2. Модуль Active Response миттєво ініціював команду firewall-drop, про що свідчить генерація події з ідентифікатором 651 (*Host Blocked by firewall-drop Active Response*). Доступ атакуючого до сервера був повністю заблокований на рівні ядра ОС.
3. Рівно через 60 секунд, згідно з налаштованим таймаутом, система автоматично виконала розблокування адреси, згенерувавши контрольну подію 652 (*Host Unblocked*).

Рисунок 3.54 – Симуляція атаки підбору паролів до служби SSH

Admin-Jumpbox [Запущено] - Oracle VM VirtualBox				
Wazuh — Mozilla Firefox				
Administration Intrusion x Frequently Asked Questio Telegram Web x Wazuh x +				
192.168.90.74/app/threat-hunting#/overview/?tab=general&tabView=events&a=(filters:(),query:(language:kuery,query				
Threat Hunting				
timestamp per 3 hours				
420 hits				
May 28, 2026 @ 02:56:07.624 - Jun 3, 2026 @ 11:49:27.349				
Export Formatted Reset view 619 available fields Columns Density 1 fields sorted Full screen				
timestamp	agent.name	rule.description	rule.level	rule.id
Jun 3, 2026 @ 11:49:20.229	Ubuntu-DB	Host Unblocked by firewall-drop Active Response	3	652
Jun 3, 2026 @ 11:48:19.799	Ubuntu-DB	Host Blocked by firewall-drop Active Response	3	651
Jun 3, 2026 @ 11:48:17.316	Ubuntu-DB	sshd: Attempt to login using a non-existent user	5	5710
Jun 3, 2026 @ 11:48:15.330	Ubuntu-DB	PAM: User login failed.	5	5503
Jun 3, 2026 @ 11:48:08.377	Ubuntu-DB	Host Unblocked by firewall-drop Active Response	3	652
Jun 3, 2026 @ 11:47:07.705	Ubuntu-DB	Host Blocked by firewall-drop Active Response	3	651
Jun 3, 2026 @ 11:47:06.658	Ubuntu-DB	sshd: Attempt to login using a non-existent user	5	5710

Рисунок 3.55 – Результат перебору паролів

Реалізація такого підходу дозволила трансформувати пасивну систему моніторингу (SIEM) у повноцінну платформу класу XDR (Extended Detection and Response), здатну самостійно зупиняти кібератаки на ранніх етапах без втручання адміністратора.

3.6.3. Емуляція SQL-ін'єкцій та шкідливого трафіку – детекція IDS

Для перевірки здатності підсистеми виявлення вторгнень (IDS Suricata) аналізувати транзитний трафік на рівні застосунків (Application Layer) та виявляти шкідливі корисні навантаження (payloads), було проведено симуляцію мережевих атак із внутрішнього сегмента мережі. Метою тестування була перевірка механізму глибокої інспекції пакетів (DPI) та коректності роботи ланцюга передачі сповіщень від шлюзу OPNsense до центральної SIEM-системи Wazuh.

Оскільки мережеві атаки типу SQL-ін'єкцій та ексфільтрації даних часто здійснюються через стандартні веб-порти, для тестування використовувалися нешифровані HTTP-запити до спеціалізованих тестових ресурсів. З віртуальної машини адміністратора було виконано запит за допомогою утиліти curl, який імітує ситуацію успішної експлуатації вразливості та отримання доступу до командної оболонки з правами суперкористувача (маркер uid=0(root) gid=0(root) groups=0(root) у тілі відповіді).

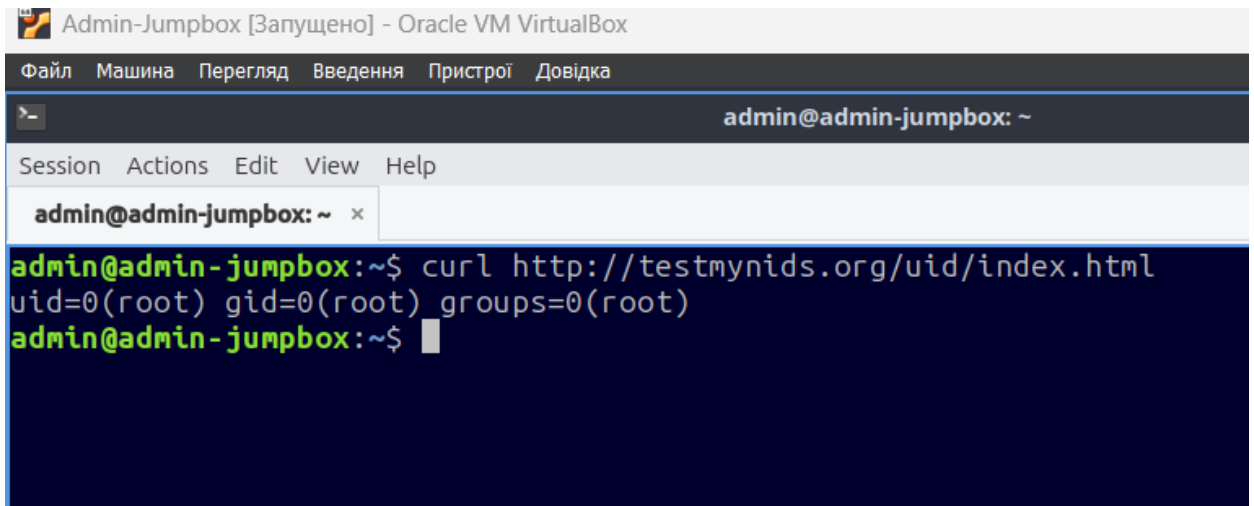


Рисунок 3.56 – Виконання HTTP-запиту з імітацією шкідливого мережевого навантаження

Підсистема IDS Suricata, що функціонує на міжмережевому екрані OPNsense, успішно здійснила перехоплення та аналіз цього транзитного трафіку. Звіривши вміст пакетів із базами сигнатур відомих загроз, система виявила точний збіг із правилом SID 2100498. Як наслідок, у локальному інтерфейсі керування мережевим екраном було згенеровано попередження «GPL ATTACK_RESPONSE id check returned root», що підтверджує дієвість підсистеми у розпізнаванні патернів успішних кібератак.

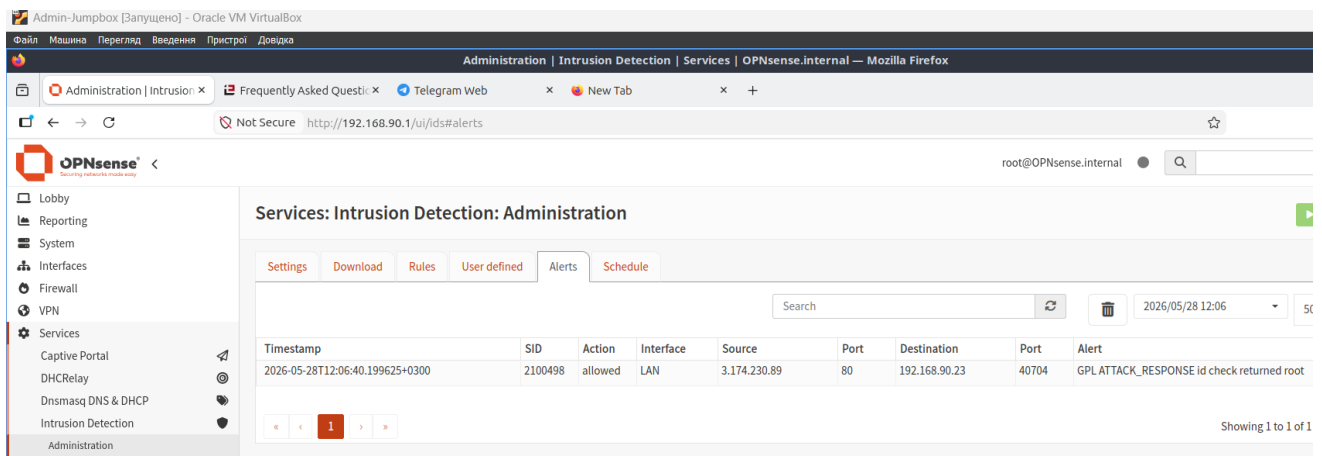


Рисунок 3.57 – Фіксація шкідливого трафіку та спрацювання сигнатури в інтерфейсі OPNsense

Згідно з налаштованою раніше архітектурою маршрутизації логів, це сповіщення було автоматично експортовано через протокол Syslog на сервер управління Wazuh Manager. Модуль аналізу SIEM-системи обробив вхідне повідомлення та застосував ієрархію кастомних правил кореляції. Система

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				80
Змн.	Арк.	№ докум.	Підпис	Дата		

розпізнала маркер «ATTACK_RESPONSE», після чого правило з ідентифікатором 100011 автоматично підвищило статус інциденту. На дашборді модулю Threat Hunting було згенеровано критичний алерт найвищого рівня загрози (Level 12) із детальним описом події.

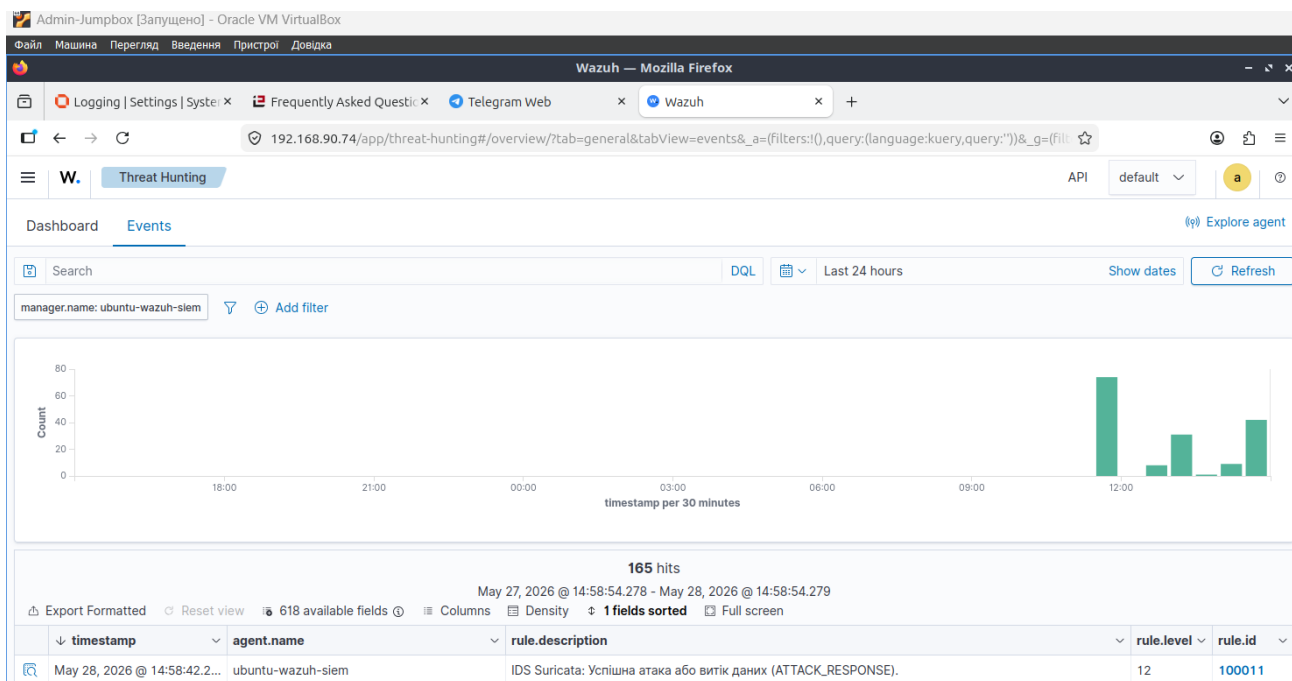


Рисунок 3.58 – Відображення перехопленої атаки з присвоєнням критичного рівня загрози на панелі інцидентів Wazuh

3.6.4. Перевірка цілісності шифрування даних у БД

Для проведення експерименту за допомогою розширення pgcrypto було створено тестову таблицю `clients_confidential`, яка містить ідентифікатор, ім'я клієнта та стовпець `secret_data` формату BYTEA для зберігання зашифрованого бінарного навантаження. До бази було внесено імітаційні записи з різними категоріями чутливих даних: персональною інформацією (паспорт, ІПН), фінансовими даними (номери карток, PIN) та медичними відомостями. Шифрування здійснювалося безпосередньо під час запису за допомогою функції `pgp_sym_encrypt` із застосуванням унікальних симетричних паролів.

```

secure_data=# CREATE TABLE clients_confidential (
secure_data=# id SERIAL PRIMARY KEY,
secure_data=# full_name VARCHAR(100) NOT NULL,
secure_data=# secret_data BYTEA NOT NULL
secure_data=# );
CREATE TABLE
secure_data=# INSERT INTO clients_confidential (full_name, secret_data)
secure_data=# VALUES
secure_data=# ('Simashko Ivan', pgp_sym_encrypt('Passport: KI 123456, IPN: 312398742', 'SimashkoPassword1337')),
secure_data=# ('Podro Homenk', pgp_sym_encrypt('DebitCard: 8594-1233-1233-1233, PIN: 2100', 'KAVA_o_2100_password')),
secure_data=# ('TheyKilledKenny', pgp_sym_encrypt('Medical_Info: BloodType -, Diagnos: NotHelthy', 'UareBreakingMyBales'));
INSERT 0 3
secure_data=# SELECT id, full_name, secret_data FROM clients_confidential
secure_data=# ;
 id | full_name | secret_data
-----+-----+-----
1 | Simashko Ivan | \xc30d040703020d2db8411c36d6d66cd25401ae8f04d0e1632d4cb560fbb9f45f3984883a6685230879721e5a233ef09fab64a2bd1b5d5502443dc8fa15ff78c6a24d3f2c6992033e66e9810ba73a783bd66d1b63b833bcd3f76e43dcd4468ccdadbd3d5a32
2 | Podro Homenk | \xc30d04070302506b6a4903a37b617dd25a016bde83ed559eb9706c41e2be1eb55379b032f2e85b84e9e0374ce4d45fee04e86ca20d3d504d7d68271219ac0f78bda17fc887eb7a2d53a66ac413c159fc5b62d4c3c888b6e7e2b9bd61a3a1c6ff56db03152325dade941c01
3 | TheyKilledKenny | \xc30d0407030243751f7f9f76d45764d25e019a437661f35be0d18426f3b8d154c566fc4893e0da5581c59a154cffe46f4a20bca0c9899512cedb470b8ab28553a4ab87e4fcd489076f06d16ecaddb73eedab997f97c5c561e6e06d1022b01185e9cc5da9e4890362ce2108734fdea
(3 rows)
(END)

```

Рисунок 3.59 – Створення таблиці та шифрування

Виконання стандартного SQL-запиту SELECT для читання вмісту таблиці продемонструвало ефективність застосованого захисту. Як видно з результатів виводу, замість відкритого тексту база даних повертає виключно шістнадцятковий шифротекст (hex-рядок, що починається з \x...). Це підтверджує, що у разі успішної компрометації сервера баз даних, викрадення дамів пам'яті або несанкціонованого доступу до таблиць через SQL-ін'єкцію, зловмисник не зможе прочитати конфіденційну інформацію без знання відповідних криптографічних ключів.

Наступним кроком стала перевірка процесу легітимного доступу та оцінка цілісності даних після їх розшифрування. Для імітації запиту від авторизованого застосунку було використано функцію `pgp_sym_decrypt` із передачею правильного секретного ключа для цільового запису.

```

secure_data=# SELECT
secure_data=# id,
secure_data=# full_name,
secure_data=# pgp_sym_decrypt(secret_data, 'UareBreakingMyBales') AS decrypted_info
secure_data=# FROM clients_confidential
secure_data=# WHERE full_name = 'TheyKilledKenny';
 id | full_name | decrypted_info
-----+-----+-----
3 | TheyKilledKenny | Medical_Info: BloodType -, Diagnos: NotHelthy
(1 row)
secure_data=#

```

Рисунок 3.60 – Успішне розшифрування інформації

Проведений тест підтверджує, що впроваджена модель криптографічного захисту даних повністю працездатна. Вона надійно ізолює конфіденційні відомості від несанкціонованого перегляду на рівні зберігання, водночас гарантуючи

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		82

безперебійну доступність та цілісність інформації для авторизованих процесів, що володіють відповідними ключами доступу.

3.6.5. Тестування процедури відновлення з резервної копії

Для підтвердження надійності розгорнутої підсистеми резервного копіювання було проведено практичне тестування процедури аварійного відновлення (Disaster Recovery Simulation). Цей етап є критично важливим, оскільки наявність бекапів без перевірки їхньої здатності до відновлення не гарантує безпеки даних.

```
admin1@ubuntu-db:~$ sudo ls /var/lib/postgresql/18/main
PG_VERSION  current_logfiles  log                pg_dynshmem  pg_multixact  pg_replslot  pg_snapshots  pg_stat_tmp  pg_tblspc  pg_wal  postgresql.auto.conf
base        global            pg_commit_ts      pg_logical   pg_notify     pg_serial    pg_stat       pg_subtrans  pg_twophase pg_xact  postmaster.opts

admin1@ubuntu-db:~$ sudo rm -rf /var/lib/postgresql/18/main
admin1@ubuntu-db:~$ sudo mkdir -p /var/lib/postgresql/18/main
admin1@ubuntu-db:~$ sudo chown postgres:postgres /var/lib/postgresql/18/main
admin1@ubuntu-db:~$ sudo chmod 700 /var/lib/postgresql/18/main
admin1@ubuntu-db:~$ sudo ls /var/lib/postgresql/18/main
admin1@ubuntu-db:~$ sudo -u postgres pgbackrest --stanza=demo_cluster restore
admin1@ubuntu-db:~$
admin1@ubuntu-db:~$ sudo ls /var/lib/postgresql/18/main
PG_VERSION  current_logfiles  pg_commit_ts  pg_multixact  pg_serial  pg_stat_tmp  pg_twophase  postgresql.auto.conf
backup_label global          pg_logical    pg_notify     pg_replslot pg_subtrans  pg_wal       recovery.signal
base        log             pg_logical    pg_replslot   pg_stat    pg_tblspc    pg_xact

admin1@ubuntu-db:~$ sudo systemctl start postgresql
admin1@ubuntu-db:~$ sudo systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/usr/lib/systemd/system/postgresql.service; enabled; preset: enabled)
   Active: active (exited) since Wed 2026-06-03 09:29:25 UTC; 16s ago
 Invocation: c5116d210ac1493dbb73378795acc968
   Process: 6591 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
  Main PID: 6591 (code=exited, status=0/SUCCESS)
     Mem peak: 2M
        CPU: 6ms

Jun 03 09:29:24 ubuntu-db systemd[1]: Starting postgresql.service - PostgreSQL RDBMS...
Jun 03 09:29:25 ubuntu-db systemd[1]: Finished postgresql.service - PostgreSQL RDBMS.
admin1@ubuntu-db:~$ sudo -u postgres psql -d secure_data
psql (18.3 (Ubuntu 18.3-1))
Type "help" for help.

secure_data=# SELECT id, full_name, secret_data FROM clients_confidential;
 id | full_name | secret_data
-----+-----+-----
 1 | Simashko Ivan | \xc30d040703020d2db8411c36d6d66cd25401ae8f04d0e1632d4cb560fbb9f45f3984883a6685230879721e5a233ef09fab64a2bd1b5d5502443dc8fa15ff78c6a24d39
2c6992033e66e9810ba73a783bd66d1b63b833bcd3f76e43dcd4468ccdadbd3d5a32
 2 | Podro Homek | \xc30d04070302506b6a4903a37b617dd25a016bde83ed559eb9706c41e2be1eb55379b032f2e85b84e9e0374ce4d45fee04e86ca20d3d504d7d68271219ac0f78bda17b
c887eb7a2d53a66ac413c159fc5b62d4c3c888b6e7e2b9bd61a3a1c6ff56db03152325dade941c01
 3 | TheyKilledKenny | \xc30d0407030243751f7f9f76d45764d25e019a437661f35be0d18426f3b8d154c566fc4893e0da5581c59a154cffe46f4a20bca0c9899512cedb470b8ab28553a4ab87
e4fcd489076f06d16ecaddd73eedab997f97c5c561e6e06d1022b01185e9cc5da9e4890362ce2108734fdea
(3 rows)

(END)
```

Рисунок 3.61 – Повне видалення та відновлення БД

3.7. Рекомендації щодо впровадження та масштабування

Розроблена та протестована в межах даної дипломної роботи інфраструктура продемонструвала високу ефективність у виявленні та блокуванні кіберзагроз. Однак, оскільки реалізація виконувалася на базі віртуального тестового стенда (Testbed), впровадження цього рішення в реальне корпоративне (Enterprise) середовище вимагає врахування низки архітектурних та експлуатаційних особливостей.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		83

3.7.1. Особливості переходу від тестового стенду до промислового впровадження

Перехід від експериментального середовища до промислової експлуатації (Production) вимагає трансформації архітектури з фокусом на високу доступність (High Availability, HA) та відмовостійкість (Fault Tolerance).

Основні рекомендації щодо промислового впровадження:

1. **Масштабування бази даних:** Використання єдиного сервера бази даних (Ubuntu-DB) створює єдину точку відмови (Single Point of Failure). У промисловому середовищі необхідно впровадити кластеризацію PostgreSQL (наприклад, за допомогою Patroni або Repmgr) із налаштуванням потокової реплікації (Master-Slave) та автоматичного перемикавання (Auto-failover). Для оптимізації навантаження доцільно використовувати балансувальники з'єднань, такі як PgBouncer.

2. **Розподілена архітектура SIEM:** Архітектура Wazuh «All-in-One» підходить лише для невеликих мереж або тестування. Для обробки тисяч подій на секунду (EPS) необхідно розгорнути розподілений кластер: виділити окремі сервери для Wazuh Manager Cluster (для парсингу та правил), окремий кластер індексаторів Wazuh Indexer (OpenSearch/Elasticsearch) для зберігання логів та балансувальники навантаження (Nginx/HAProxy) перед ними.

3. **Автоматизація розгортання (Infrastructure as Code):** Ручне конфігураційне управління, яке застосовувалося на стенді, не масштабується. Для розгортання сотень агентів та налаштування серверів слід використовувати інструменти IaC, такі як Ansible, Terraform або Chef, що дозволить підтримувати ідентичність конфігурацій (Idempotency) та уникати помилок людського фактора.

4. **Сховища резервних копій:** Замість локального зберігання бекапів pgBackRest, у промисловому середовищі необхідно використовувати об'єктні хмарні сховища (сумісні з Amazon S3) або виділені NAS-сервери в інших географічних зонах (за правилом бекапів 3-2-1).

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				84
Змн.	Арк.	№ докум.	Підпис	Дата		

3.7.2. Можливі напрями розвитку системи

Кібербезпека є безперервним процесом, тому розроблена система має значний потенціал для подальшої модернізації та посилення:

1. **Впровадження платформ SOAR (Security Orchestration, Automation, and Response):** Поточний модуль Active Response ефективно вирішує базові завдання блокування. Інтеграція Wazuh із SOAR-платформами (наприклад, TheHive, Shuffle) дозволить створювати складні автоматизовані сценарії реагування (Playbooks) із залученням різних ІТ-систем та оповіщенням адміністраторів через месенджери чи системи тикетів.

2. **Інтеграція Threat Intelligence (TI):** Підключення зовнішніх джерел даних про кіберзагрози (MISP, VirusTotal, AlienVault OTX) до SIEM-системи дозволить виявляти шкідливу активність на основі актуальних індикаторів компрометації (IoC), таких як відомі шкідливі IP-адреси, хеші файлів та домени.

3. **Управління секретами (Secret Management):** Замість статичного зберігання ключів та паролів доцільно впровадити системи динамічного управління секретами, такі як HashiCorp Vault. Це дозволить генерувати тимчасові облікові дані для доступу до бази даних та автоматично ротувати криптографічні ключі.

4. **Контроль доступу з нульовою довірою (Zero Trust Network Access):** Подальший розвиток мережевого захисту передбачає перехід від VPN до ZTNA-рішень та мікросегментації мережі, де доступ надається не до всього сегмента, а лише до конкретних додатків після багатофакторної автентифікації (MFA) та перевірки стану безпеки пристрою користувача.

Висновки до розділу 3

У третьому розділі дипломної роботи було успішно вирішено комплексне практичне завдання з проектування, розгортання та тестування захищеної ІТ-інфраструктури підприємства. На базі технологій віртуалізації було створено ізольований мережевий сегмент, який включає маршрутизатор з функціями IDS/IPS (OPNsense), сервер бази даних (PostgreSQL) та систему централізованого моніторингу (Wazuh SIEM).

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		85

У ході виконання практичної частини було реалізовано наступні завдання:

– **Побудова глибоко ешелонованого захисту СУБД:** Виконано комплексне зміцнення операційної системи (Hardening). Впроваджено суворі політики міжмережевого екрана (UFW), налаштовано криптографічну автентифікацію SSH (Ed25519). На рівні самої бази даних реалізовано примусове шифрування мережевого трафіку (TLS 1.3) та застосовано модуль pgcrypto для симетричного шифрування чутливих даних клієнтів безпосередньо у таблицях (Data at Rest).

– **Забезпечення відмовостійкості:** Налаштовано автоматизовану систему резервного копіювання за допомогою pgBackRest. Резервні копії захищені алгоритмом AES-256-CBC, а успішність механізмів Disaster Recovery підтверджена шляхом симуляції критичного збою та повного відновлення бази даних.

– **Централізований моніторинг (SIEM):** Розгорнуто платформу Wazuh, до якої інтегровано події з системних журналів ОС, систем аудиту (auditd, pgAudit) та мережевого екрана периметра (Suricata IDS / OPNsense). Розроблено власні правила кореляції (Custom Rules) для парсингу JSON-логів PostgreSQL та виявлення аномальних звернень до конфіденційної інформації.

– **Верифікація та реагування:** Проведено практичне тестування ефективності системи шляхом симуляції реальних кібератак (мережеве сканування Nmap, спроби несанкціонованого доступу, підбір паролів). Система продемонструвала здатність не лише виявляти загрози, але й проактивно блокувати атакуючі вузли в автоматичному режимі за допомогою модуля Wazuh Active Response.

Таким чином, розроблена інфраструктура повністю відповідає сучасним вимогам інформаційної безпеки, забезпечує конфіденційність, цілісність та доступність даних, а також може слугувати надійною основою для впровадження у корпоративному середовищі з подальшим масштабуванням.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				86
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Під час роботи над кваліфікаційною роботою було успішно розв'язано актуальне науково-практичне завдання щодо розроблення та практичної реалізації проекту комплексної системи багаторівневого захисту мережної інфраструктури, призначеної для безпечної обробки та зберігання конфіденційних даних. На основі парадигми Defense-in-Depth (захист у глибину) було доведено, що використання виключно вільного програмного забезпечення (open-source рішень) дозволяє побудувати інфраструктуру рівня Enterprise, здатну ефективно протистояти сучасним кіберзагрозам.

Загальні підсумки роботи підтверджують, що розроблена архітектура повністю задовольняє поставлені на початку проектування функціональні та нефункціональні вимоги.

Проведено системний аналіз предметної області та профілювання загроз. Досліджено статистику кіберінцидентів та типові вектори атак на бази даних. За допомогою методології STRIDE було побудовано комплексну модель загроз, яка охопила ризики підміни користувача, порушення цілісності, відмови від дій, витоку інформації, відмови в обслуговуванні (DoS) та підвищення привілеїв.

Розроблено логічну топологію мережі з поділом на ізольовані віртуальні сегменти (WAN, LAN, Management VLAN, SecureDB VLAN), що унеможливорює прямий доступ до бази даних із глобальної мережі.

На базі програмного шлюзу OPNsense розгорнуто міжмережевий екран із політикою Default Deny. Налаштовано захищений віддалений доступ для адміністраторів через сучасний VPN-протокол WireGuard із криптографічною автентифікацією. Для проактивного виявлення загроз на мережевому рівні в OPNsense інтегровано NIDS Suricata з актуальними базами сигнатур Emerging Threats Open та логуванням у структурованому форматі EVE JSON.

Виконано комплексне підсилення (Hardening) операційної системи Ubuntu Server: налаштовано локальний фаєрвол UFW, систему мандатного контролю доступу AppArmor, службу низькорівневого логування auditd та впроваджено доступ по SSH виключно за ключами Ed25519. Захист PostgreSQL реалізовано

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				87
Змн.	Арк.	№ докум.	Підпис	Дата		

шляхом примусового шифрування транспорту (TLS 1.3), застосування розширення pgcrypto для шифрування конфіденційних даних на рівні стовпців таблиць (Data at Rest) та активації деталізованого аудиту подій через pgAudit. Забезпечено відмовостійкість за допомогою автоматизованого створення зашифрованих резервних копій утилітою pgBackRest.

Побудовано централізовану систему моніторингу та реагування (SIEM/XDR): Розгорнуто платформу Wazuh (All-in-One), до якої підключено агенти з серверів баз даних та налаштовано отримання Syslog-повідомлень від OPNsense. Розроблено ієрархію власних правил кореляції (Custom Rules) для нормалізації подій від pgAudit та Suricata, що дозволило агрегувати розрізнені логи в єдиній консолі управління (Threat Hunting Dashboard).

Проведено практичне тестування ефективності: Тестовий полігон у середовищі VirtualBox успішно витримав серію імітованих кібератак. Доведено здатність системи блокувати спроби зовнішнього сканування портів (Nmap), автоматично припиняти атаки підбору паролів (Brute-Force) за допомогою модуля Wazuh Active Response (блокування IP-адреси зловмисника на 60 секунд), виявляти мережеві SQL-ін'єкції завдяки DPI-аналізу Suricata та гарантувати збереження конфіденційності даних у БД через успішні процедури шифрування/розшифрування.

Практична цінність одержаних результатів полягає у створенні готової до впровадження моделі захищеної IT-інфраструктури, яка не потребує капітальних інвестицій у комерційне програмне забезпечення (Vendor Lock-in). Самостійно розроблені конфігураційні профілі для фаєрволів, адаптовані правила кореляції для SIEM-системи Wazuh (зокрема, для ідентифікації подій розширення pgAudit та транзитних сповіщень Suricata) та підходи до інтеграції криптографічного захисту СУБД становлять готовий інструментарій для системних адміністраторів та фахівців з кібербезпеки. Проєкт доводить, що наявність open-source рішень дозволяє трансформувати пасивну систему моніторингу в повноцінну XDR-платформу, здатну автоматично реагувати на загрози.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				88
Змн.	Арк.	№ докум.	Підпис	Дата		

Практичне значення роботи полягає в тому, що спроектована та протестована на віртуальному полігоні архітектура може бути безпосередньо адаптована та впроваджена в реальних корпоративних мережах підприємств малого та середнього бізнесу (SMB), державних установах чи медичних закладах, що здійснюють обробку критичної або конфіденційної інформації. Наведена у роботі детальна методологія розгортання, від конфігурації VPN та VLAN до глибокого інспектування пакетів і шифрування резервних копій, є практичним посібником для побудови надійного фундаменту інформаційної безпеки організації.

Сформульовані в останньому розділі рекомендації щодо кластеризації PostgreSQL, переходу до розподіленої архітектури SIEM та інтеграції з платформами SOAR забезпечують чіткий вектор для подальшого масштабування системи до рівня великих Enterprise-інфраструктур.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		89

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. NIST Special Publication 800-30 Rev. 1. Guide for Conducting Risk Assessments. – Gaithersburg: National Institute of Standards and Technology, 2012. – 95 p.
2. OWASP Foundation. OWASP Top 10 – 2021: The Ten Most Critical Web Application Security Risks [Електронний ресурс]. – 2021. – Режим доступу: <https://owasp.org/www-project-top-ten/> (дата звернення: 15.05.2026).
3. Mandiant. M-Trends 2024: Insights into Today's Cyber Threats and Defender Strategies. – Mandiant, 2024. – 56 p.
4. Lockheed Martin. The Cyber Kill Chain® [Електронний ресурс]. – Режим доступу: <https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html> (дата звернення: 15.05.2026).
5. Verizon. 2023 Data Breach Investigations Report. – Verizon Business, 2023. – 108 p.
6. ENISA. ENISA Threat Landscape 2023. – European Union Agency for Cybersecurity, 2023. – 124 p.
7. Про захист інформації в інформаційно-комунікаційних системах : Закон України від 05.07.1994 № 80/94-ВР (зі змінами) // Відомості Верховної Ради України. – 1994. – № 31. – Ст. 286.
8. НД ТЗІ 2.5-010-03. Захист інформації на об'єктах інформаційної діяльності. Вимоги до комплексної системи захисту інформації (КСЗІ) [Текст]. – Київ : ДССЗІ, 2003.
9. НД ТЗІ 2.7-010-09. Захист інформації на об'єктах інформаційної діяльності. Методика оцінки захищеності державних інформаційних ресурсів [Текст]. – Київ : Адміністрація Держспецзв'язку, 2009.
10. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation) // Official Journal of the European Union. – 2016. – L 119. – P. 1–88.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				90
Змн.	Арк.	№ докум.	Підпис	Дата		

11. NIST Special Publication 800-53 Rev. 5. Security and Privacy Controls for Information Systems and Organizations. – Gaithersburg: National Institute of Standards and Technology, 2020. – 492 p.
12. Microsoft. The STRIDE Threat Model [Електронний ресурс]. – Режим доступу: [https://learn.microsoft.com/en-us/previous-versions/commerce-server/ee823878\(v=cs.20\)](https://learn.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20)) (дата звернення: 15.05.2026).
13. Uceda Velez T., Morana M. M. Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis. – Hoboken : John Wiley & Sons, 2015. – 696 p.
14. Netgate. pfSense® Documentation [Електронний ресурс]. – Режим доступу: <https://docs.netgate.com/pfsense/en/latest/> (дата звернення: 15.05.2026).
15. OPNsense® Project. OPNsense Documentation [Електронний ресурс]. – Режим доступу: <https://docs.opnsense.org/> (дата звернення: 15.05.2026).
16. VyOS Project. VyOS Documentation [Електронний ресурс]. – Режим доступу: <https://docs.vyos.io/> (дата звернення: 15.05.2026).
17. Cisco. Snort® Users Manual 3.1. – Cisco Systems, 2023. – 250 p.
18. Open Information Security Foundation. Suricata User Guide [Електронний ресурс]. – Режим доступу: <https://docs.suricata.io/> (дата звернення: 15.05.2026).
19. The Zeek Project. Zeek Documentation [Електронний ресурс]. – Режим доступу: <https://docs.zeek.org/> (дата звернення: 15.05.2026).
20. Atomicorp. OSSEC Documentation [Електронний ресурс]. – Режим доступу: <https://www.ossec.net/docs/> (дата звернення: 15.05.2026).
21. Wazuh Inc. Wazuh Documentation (v4.7) [Електронний ресурс]. – Режим доступу: <https://documentation.wazuh.com/> (дата звернення: 15.05.2026).
22. Elasticsearch B.V. Elastic Stack and Product Documentation [Електронний ресурс]. – Режим доступу: <https://www.elastic.co/guide/index.html> (дата звернення: 15.05.2026).
23. The PostgreSQL Global Development Group. PostgreSQL 18 Documentation [Електронний ресурс]. – Режим доступу: <https://www.postgresql.org/docs/current/index.html> (дата звернення: 15.05.2026).

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				91
Змн.	Арк.	№ докум.	Підпис	Дата		

24. OpenVPN Inc. OpenVPN Community Resources [Електронний ресурс]. – Режим доступу: <https://openvpn.net/community-resources/> (дата звернення: 15.05.2026).

25. Donenfeld J. A. WireGuard: fast, modern, secure VPN tunnel // Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. – ACM, 2017.

26. Kent S., Seo K. Security Architecture for the Internet Protocol : RFC 4301. – IETF, 2005.

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				92
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				93
Змн.	Арк.	№ докум.	Підпис	Дата		

Лістинги конфігураційних файлів та додаткові етапи налаштування

Тестування VPN- та підключення правил фільтрації

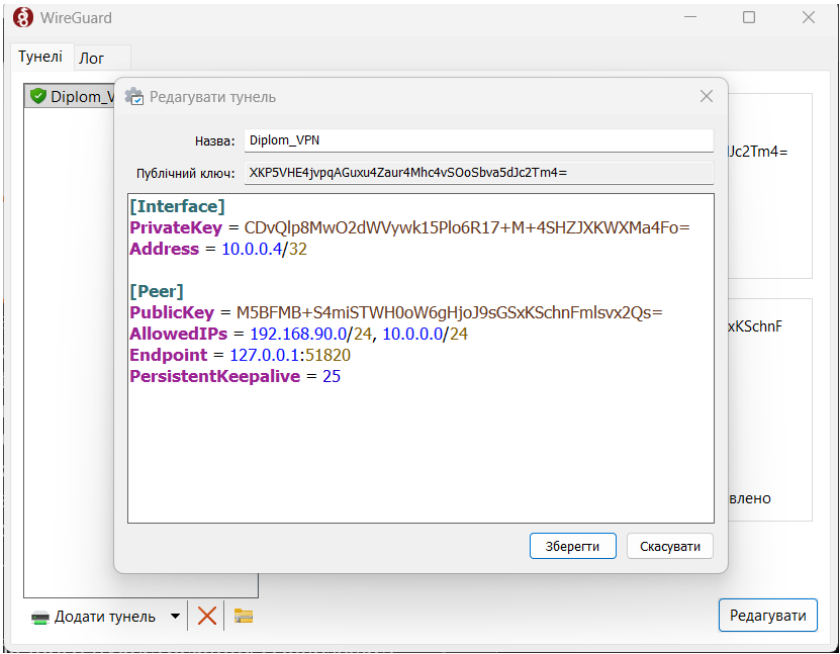


Рисунок А.1 – Налаштування тунелю на стороні користувача

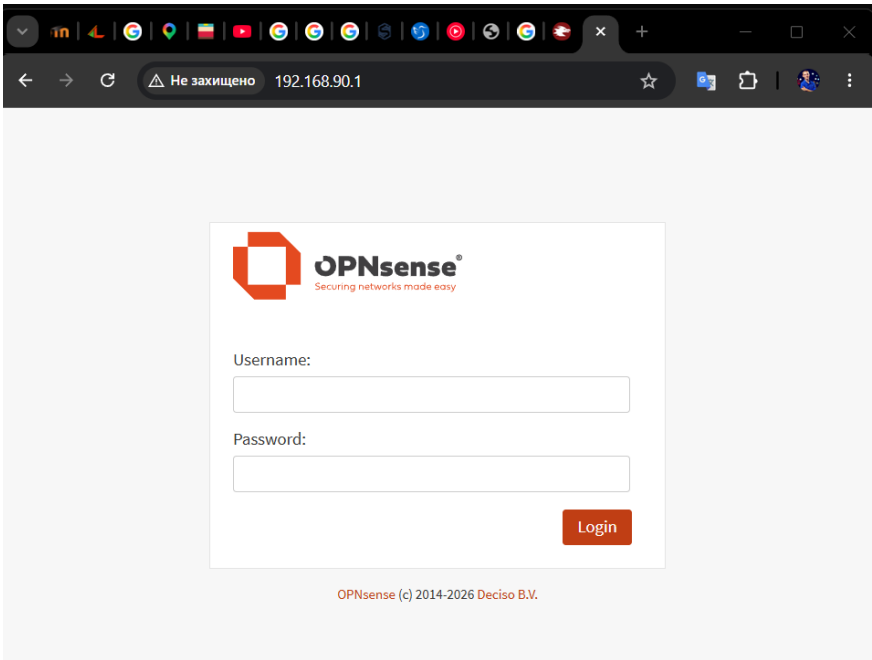


Рисунок А.2 – Вхід на сторінку OPNsense через основний комп’ютер підключений до VPN

Налаштування TLS

У конфігураційному файлі postgresql було створено правило, яке дозволяє підключення до бази secure_data користувачу db_admin виключно з IP-адрес внутрішньої підмережі 192.168.90.0/24. Будь-які спроби підключення з інших сегментів або з глобальної мережі автоматично відхиляються на рівні самої СУБД. Крім того, для захисту процесу автентифікації від атак перехоплення (MitM), механізм перевірки паролів був переведений на використання сучасного криптографічного алгоритму scram-sha-256, який є стійким до атак за словником та є значно безпечнішим за застарілий алгоритм MD5.

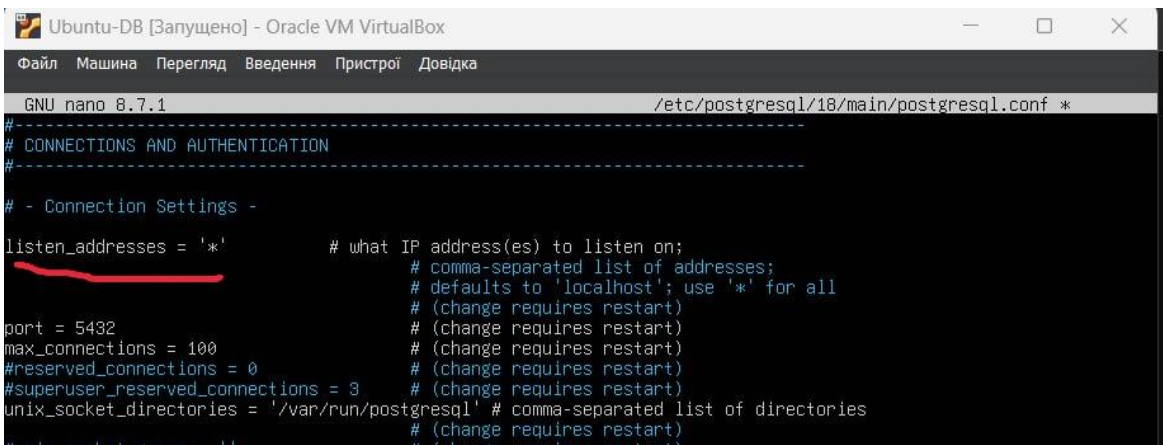


Рисунок А.3 – Базова мережева конфігурація служби PostgreSQL

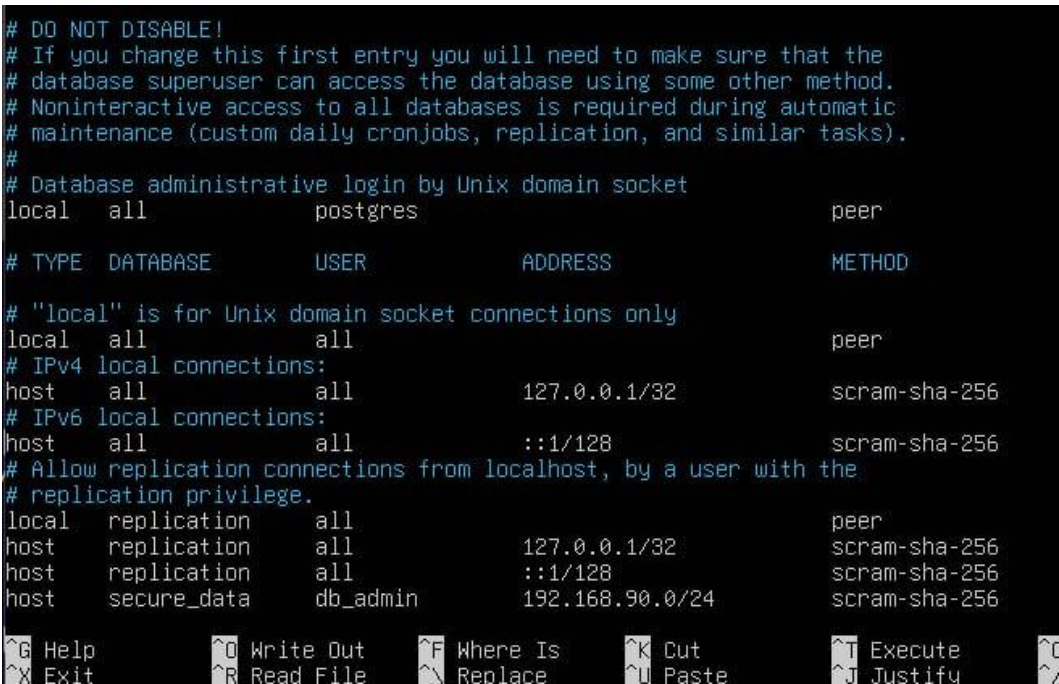


Рисунок А.4 – Створено правило для бд та користувача

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				95
Змн.	Арк.	№ докум.	Підпис	Дата		

```

# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all peer
# IPv4 local connections:
host all all 127.0.0.1/32 scram-sha-256
# IPv6 local connections:
host all all ::1/128 scram-sha-256
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all peer
host replication all 127.0.0.1/32 scram-sha-256
host replication all ::1/128 scram-sha-256
hostssl secure_data db_admin 192.168.90.0/24 scram-sha-256

```

Рисунок А.5 – Редагування конфігураційного файлу

```

admin1@ubuntu-db:~$ sudo systemctl restart postgresql
admin1@ubuntu-db:~$

```

Рисунок А.6 – Перезапуск служби

```

admin@admin-jumpbox:~$ psql -h 192.168.90.23 -U db_admin -d secure_data
Password for user db_admin:
psql (18.4 (Ubuntu 18.4-0ubuntu0.26.04.1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off
, ALPN: postgresql)
Type "help" for help.

secure_data=> \conninfo
          Connection Information
Parameter | Value
-----|-----
Database | secure_data
Client User | db_admin
Host | 192.168.90.23
Server Port | 5432
Options |
Protocol Version | 3.0
Password Used | true
GSSAPI Authenticated | false
Backend PID | 2310
SSL Connection | true
SSL Library | OpenSSL
SSL Protocol | TLSv1.3
SSL Key Bits | 256
SSL Cipher | TLS_AES_256_GCM_SHA384
SSL Compression | false
ALPN | postgresql
Superuser | off
Hot Standby | off
(18 rows)

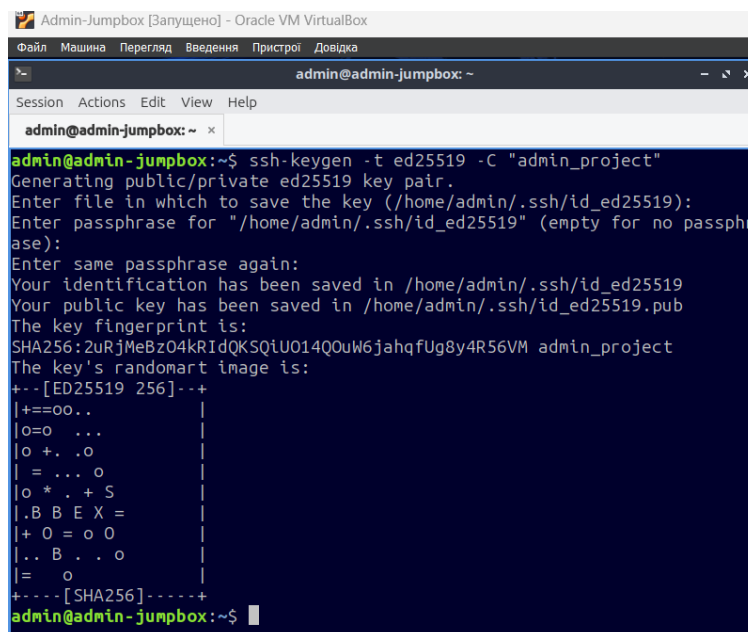
secure_data=>

```

Рисунок А.7 – Успішна автентифікація

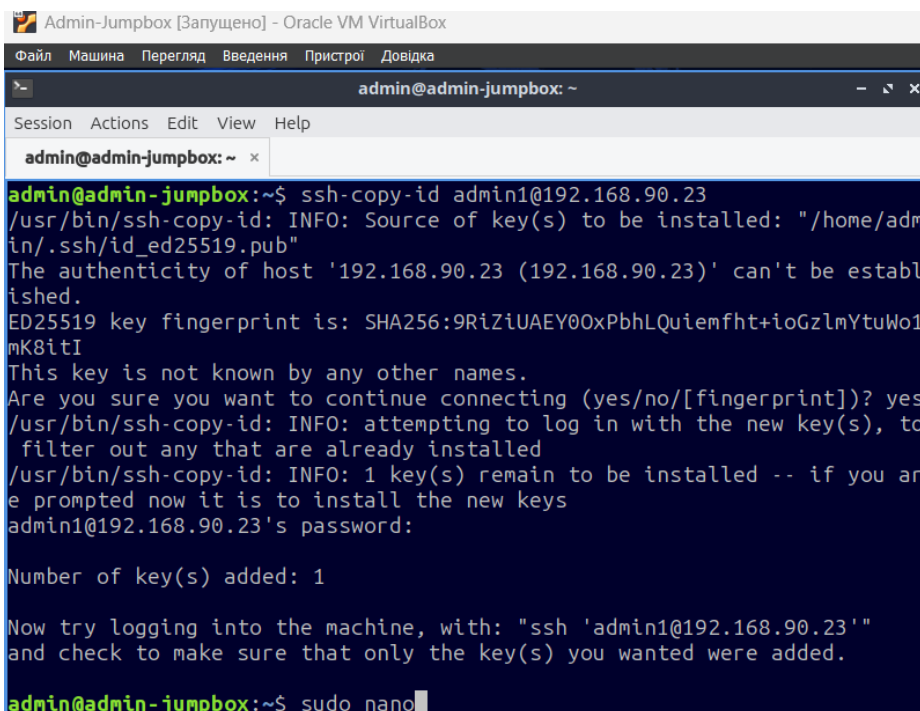
Генерація та обмін SSH ключів

Використано сучасний алгоритм на еліптичних кривих **Ed25519** (ssh-keygen -t ed25519).



```
Admin-Jumpbox [Запущено] - Oracle VM VirtualBox
Файл  Машина  Перегляд  Введення  Пристрої  Довідка
admin@admin-jumpbox: ~
Session  Actions  Edit  View  Help
admin@admin-jumpbox: ~ x
admin@admin-jumpbox:~$ ssh-keygen -t ed25519 -C "admin_project"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/admin/.ssh/id_ed25519):
Enter passphrase for "/home/admin/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/admin/.ssh/id_ed25519
Your public key has been saved in /home/admin/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:2uRjMeBz04kRIdQKSQiU014QOuW6jahqfUg8y4R56VM admin_project
The key's randomart image is:
+--[ED25519 256]--+
|+==00..          |
|o=o   ...        |
|o+.  .o          |
| = .. o          |
|o * . + S        |
|.B B E X =       |
|+ O = o O        |
|.. B . . o       |
| = o             |
+----[SHA256]-----+
admin@admin-jumpbox:~$
```

Рисунок А.8 – Генерація криптографічної пари ключів



```
Admin-Jumpbox [Запущено] - Oracle VM VirtualBox
Файл  Машина  Перегляд  Введення  Пристрої  Довідка
admin@admin-jumpbox: ~
Session  Actions  Edit  View  Help
admin@admin-jumpbox: ~ x
admin@admin-jumpbox:~$ ssh-copy-id admin1@192.168.90.23
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/admin/.ssh/id_ed25519.pub"
The authenticity of host '192.168.90.23 (192.168.90.23)' can't be established.
ED25519 key fingerprint is: SHA256:9RiZiUAEY00xPbhlQuiemfht+ioGzlnYtuWo1mK8iti
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
admin1@192.168.90.23's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'admin1@192.168.90.23'"
and check to make sure that only the key(s) you wanted were added.
admin@admin-jumpbox:~$ sudo nano
```

Рисунок А.9 – Експорт відкритого ключа на захищений сервер

```

Admin-Jumpbox [Запущено] - Oracle VM VirtualBox
Файл  Машина  Перегляд  Введення  Пристрої  Довідка
admin1@ubuntu-db: ~
Session  Actions  Edit  View  Help
admin1@ubuntu-db: ~ x
GNU nano 8.7.1 /etc/ssh/sshd_config *

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

```

Рисунок А.10 – Налаштування політик безпеки у конфігураційному файлі служби SSH

- PermitRootLogin no – жорстка заборона прямого віддаленого входу для суперкористувача (root), що є обов'язковою вимогою стандартів безпеки (наприклад, PCI DSS).
- PubkeyAuthentication yes – явний дозвіл на використання криптографічних ключів для входу.

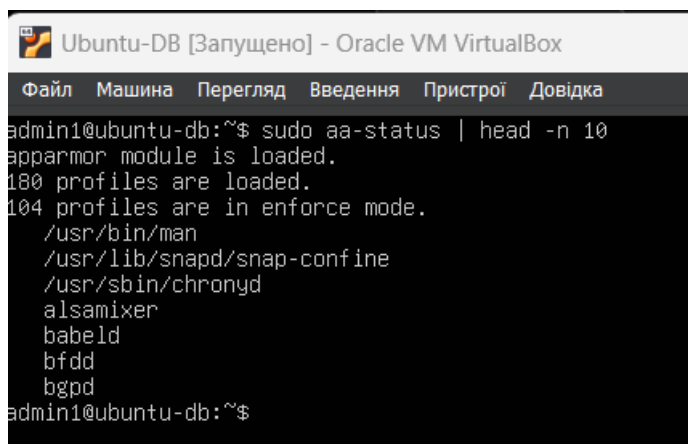
```

Admin-Jumpbox [Запущено] - Oracle VM VirtualBox
Файл  Машина  Перегляд  Введення  Пристрої  Довідка
admin1@ubuntu-db: ~
Session  Actions  Edit  View  Help
admin1@ubuntu-db: ~ x
admin1@ubuntu-db:~$ sudo systemctl restart ssh
admin1@ubuntu-db:~$

```

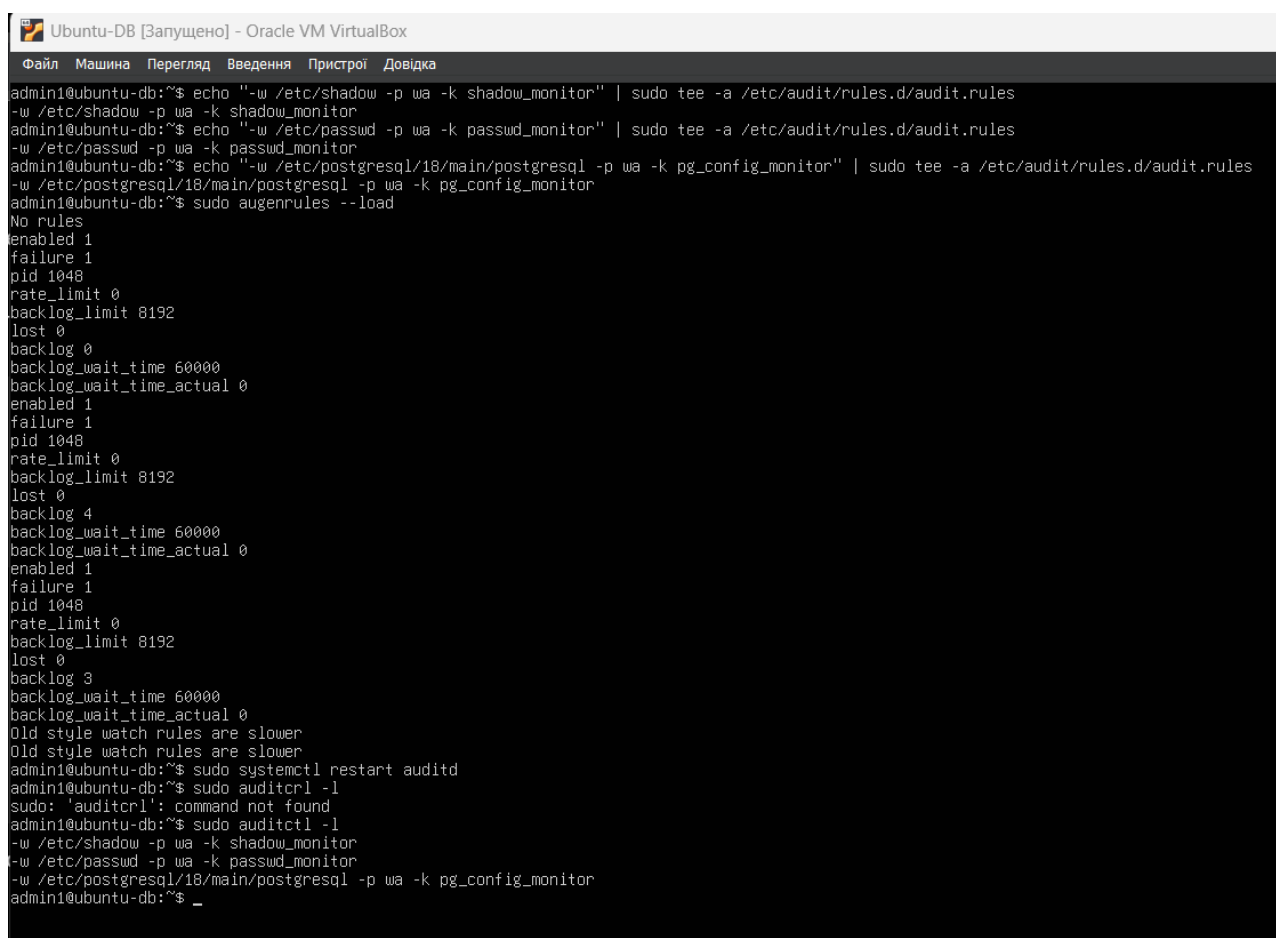
Рисунок А.11 – Перезапуск служби

AppArmor та AUDITD



```
admini@ubuntu-db:~$ sudo aa-status | head -n 10
apparmor module is loaded.
180 profiles are loaded.
104 profiles are in enforce mode.
/usr/bin/man
/usr/lib/snapd/snap-confine
/usr/sbin/chronyd
alsamixer
babeld
bfdd
bgpd
admini@ubuntu-db:~$
```

Рисунок А.12 – Налаштований AppArmor

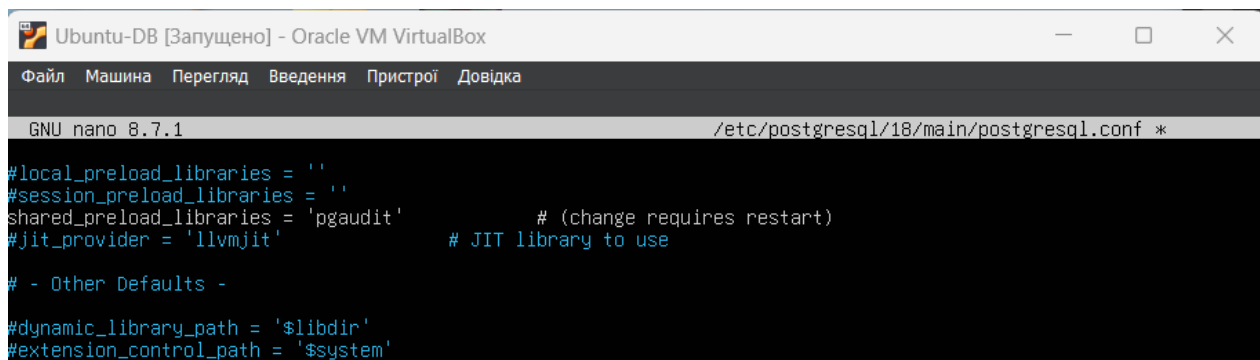


```
admini@ubuntu-db:~$ echo "-w /etc/shadow -p wa -k shadow_monitor" | sudo tee -a /etc/audit/rules.d/audit.rules
-w /etc/shadow -p wa -k shadow_monitor
admini@ubuntu-db:~$ echo "-w /etc/passwd -p wa -k passwd_monitor" | sudo tee -a /etc/audit/rules.d/audit.rules
-w /etc/passwd -p wa -k passwd_monitor
admini@ubuntu-db:~$ echo "-w /etc/postgresql/18/main/postgresql -p wa -k pg_config_monitor" | sudo tee -a /etc/audit/rules.d/audit.rules
-w /etc/postgresql/18/main/postgresql -p wa -k pg_config_monitor
admini@ubuntu-db:~$ sudo augenrules --load
No rules
enabled 1
failure 1
pid 1048
rate_limit 0
backlog_limit 8192
lost 0
backlog 0
backlog_wait_time 60000
backlog_wait_time_actual 0
enabled 1
failure 1
pid 1048
rate_limit 0
backlog_limit 8192
lost 0
backlog 4
backlog_wait_time 60000
backlog_wait_time_actual 0
enabled 1
failure 1
pid 1048
rate_limit 0
backlog_limit 8192
lost 0
backlog 3
backlog_wait_time 60000
backlog_wait_time_actual 0
Old style watch rules are slower
Old style watch rules are slower
admini@ubuntu-db:~$ sudo systemctl restart auditd
admini@ubuntu-db:~$ sudo auditctl -l
sudo: 'auditctl': command not found
admini@ubuntu-db:~$ sudo auditctl -l
-w /etc/shadow -p wa -k shadow_monitor
-w /etc/passwd -p wa -k passwd_monitor
-w /etc/postgresql/18/main/postgresql -p wa -k pg_config_monitor
admini@ubuntu-db:~$ _
```

Рисунок А.13 – Налаштований auditd

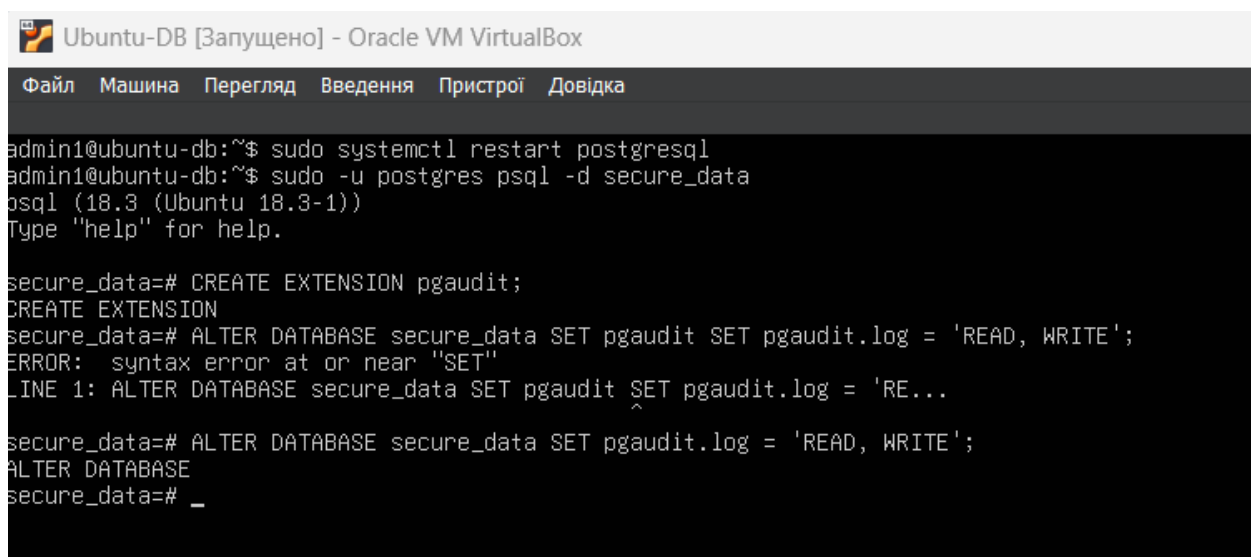
Налаштування Pgcrypto

На першому етапі було встановлено відповідний пакет postgresql-18-pgaudit. Для того, щоб ядро СУБД могло використовувати цей модуль, його було додано до параметра `shared_preload_libraries` у головному конфігураційному файлі `postgresql.conf`.



```
Ubuntu-DB [Запущено] - Oracle VM VirtualBox
Файл  Машина  Перегляд  Введення  Пристрої  Довідка
GNU nano 8.7.1 /etc/postgresql/18/main/postgresql.conf *
#local_preload_libraries = ''
#session_preload_libraries = ''
shared_preload_libraries = 'pgaudit'          # (change requires restart)
#jit_provider = 'llvmjit'                    # JIT library to use
# - Other Defaults -
#dynamic_library_path = '$libdir'
#extension_control_path = '$system'
```

Рисунок А.14 – Додавання параметру



```
Ubuntu-DB [Запущено] - Oracle VM VirtualBox
Файл  Машина  Перегляд  Введення  Пристрої  Довідка
admin1@ubuntu-db:~$ sudo systemctl restart postgresql
admin1@ubuntu-db:~$ sudo -u postgres psql -d secure_data
psql (18.3 (Ubuntu 18.3-1))
Type "help" for help.

secure_data=# CREATE EXTENSION pgaudit;
CREATE EXTENSION
secure_data=# ALTER DATABASE secure_data SET pgaudit SET pgaudit.log = 'READ, WRITE';
ERROR:  syntax error at or near "SET"
LINE 1: ALTER DATABASE secure_data SET pgaudit SET pgaudit.log = 'RE...
              ^
secure_data=# ALTER DATABASE secure_data SET pgaudit.log = 'READ, WRITE';
ALTER DATABASE
secure_data=# _
```

Рисунок А.15 – Налаштування логування


```

GNU nano 8.7.1 /etc/pgbackrest/pgbackrest.conf *
[global]
repo1-path=/var/lib/pgbackrest
repo1-retention-full=2
repo1-cipher-type=aes-256-cbc
repo1-cipher-pass=SimashkoBackupKey2026!

[demo_cluster]
pg1-path=/var/lib/postgresql/18/main

```

Рисунок А.19 – Конфігурація репозиторію pgBackRest із застосуванням шифрування AES-256-CBC

```

GNU nano 8.7.1 /etc/postgresql/18/main/postgresql.conf *
# - Archiving -
archive_mode = on                # enables archiving; off, on, or always
                                # (change requires restart)
#archive_library = ''            # library to use to archive a WAL file
                                # (empty string indicates archive_command should
                                # be used)
archive_command = 'pgbackrest --stanza=demo_cluster archive-push %p'    # command to use to archive a WAL file
                                # placeholders: %p = path of file to archive
                                #           %f = file name only
                                # e.g. 'test ! -f /mnt/server/archivedir/%f && cp %p /mnt/server/archivedir/%f'
#archive_timeout = 0             # force a WAL file switch after this
                                # number of seconds; 0 disables

```

Рисунок А.20 – Активація режиму архівування WAL-журналів у конфігурації PostgreSQL

Резервне копіювання - щоденне

Використовуючи стандартний системний планувальник завдань cron операційної системи Linux (від імені користувача postgres), я налаштував виконання фонового завдання.

```

Ubuntu-DB [Запущено] - Oracle VM VirtualBox
Файл  Машина  Перегляд  Введення  Пристрої  Довідка
GNU nano 8.7.1                                     /tmp/crontab.RjHIN9/crontab *
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
0 2 * * * pgbackrest --stanza=demo_cluster --type=full backup >> /tmp/pgbackrest_cron.log 2>&1_

```

Рисунок А.21 – Редагування системного планувальника завдань cron для автоматизації процесу

```

Ubuntu-DB [Запущено] - Oracle VM VirtualBox
Файл  Машина  Перегляд  Введення  Пристрої  Довідка
admin1@ubuntu-db:~$ sudo -u postgres crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
0 2 * * * pgbackrest --stanza=demo_cluster --type=full backup >> /tmp/pgbackrest_cron.log 2>&1
admin1@ubuntu-db:~$

```

Рисунок А.22 – Збережене завдання щоденного створення резервних копій у планувальнику

```
Ubuntu-DB [Запущено] - Oracle VM VirtualBox
Файл  Машина  Перегляд  Введення  Пристрої  Довідка

admin1@ubuntu-db:~$ sudo ls /var/lib/postgresql/18/main
PG_VERSION  current_logfiles  log                pg_dynshmem  pg_multixact  pg_repslot  pg_snapshots  pg_stat_tmp  pg_tblspc  pg_wal  postgresql.auto.conf
base        global            pg_commit_ts      pg_logical   pg_notify     pg_serial   pg_stat       pg_subtrans  pg_twophase pg_xact  postmaster.opts
admin1@ubuntu-db:~$ sudo rm -rf /var/lib/postgresql/18/main
admin1@ubuntu-db:~$ sudo mkdir -p /var/lib/postgresql/18/main
admin1@ubuntu-db:~$ sudo chown postgres:postgres /var/lib/postgresql/18/main
admin1@ubuntu-db:~$ sudo chmod 700 /var/lib/postgresql/18/main
admin1@ubuntu-db:~$ sudo ls /var/lib/postgresql/18/main
admin1@ubuntu-db:~$
admin1@ubuntu-db:~$ sudo -u postgres pgbackrest --stanza=demo_cluster restore
admin1@ubuntu-db:~$
admin1@ubuntu-db:~$ sudo ls /var/lib/postgresql/18/main
PG_VERSION  current_logfiles  pg_commit_ts  pg_multixact  pg_notify     pg_repslot  pg_stat_tmp  pg_twophase  postgresql.auto.conf
backup_label global          pg_dynshmem  pg_logical    pg_notify     pg_snapshots pg_subtrans  pg_wal       recovery.signal
base        log             pg_dynshmem  pg_repslot    pg_notify     pg_stat      pg_tblspc    pg_xact
admin1@ubuntu-db:~$
admin1@ubuntu-db:~$ sudo systemctl start postgresql
admin1@ubuntu-db:~$ sudo systemctl status postgresql
• postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/usr/lib/systemd/system/postgresql.service; enabled; preset: enabled)
   Active: active (exited) since Wed 2026-06-03 09:29:25 UTC; 16s ago
   Invocation: c5116d210ac1493dbb73378795acc968
   Process: 6591 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 6591 (code=exited, status=0/SUCCESS)
   Mem peak: 2M
   CPU: 6ms

Jun 03 09:29:24 ubuntu-db systemd[1]: Starting postgresql.service - PostgreSQL RDBMS...
Jun 03 09:29:25 ubuntu-db systemd[1]: Finished postgresql.service - PostgreSQL RDBMS.
admin1@ubuntu-db:~$ sudo -u postgres psql -d secure_data
psql (18.3 (Ubuntu 18.3-1))
Type "help" for help.

secure_data=# SELECT id, full_name, secret_data FROM clients_confidential;
 id | full_name | secret_data
-----+-----+-----
 1 | Simashko Ivan | \xc30d040703020d2db8411c36d6d66cd25401ae8f04d0e1632d4cb560fbb9f45f3984883a6685230879721e5a233ef09fab64a2bd1b5d5502443dc8fa15ff78c6a24d39
2c6992033e66e9810ba73a783bd66d1b63b833bcd3f76e43dcd4468ccdadb3d5a32
 2 | Podro Homenk | \xc30d04070302506b6a4903a37b617dd25a016bde83ed559eb9706c41e2be1eb55379b032f2e85b84e9e0374ce4d45fee04e86ca20d3d504d7d68271219ac0f78bda17b
c887eb7a2d53a66ac413c159fc5b62d4c3c888b6e7e2b9bd61a3a1c6ff56db03152325dade941c01
 3 | TheyKilledKenny | \xc30d0407030243751f7f9f76d45764d25e019a437661f35be0d18426f3b8d154c566fc4893e0da5581c59a154cffe46f4a20bca0c9899512cedb470b8ab28553a4ab87
e4fcd489076f06d16ecaddb73eedab997f97c5c561e6e06d1022b01185e9cc5da9e4890362ce2108734fdea
(3 rows)

(END)
```

Рисунок А.23 – Видалення бази даних та відновлення

```
Ubuntu-DB [Запущено] - Oracle VM VirtualBox
Файл  Машина  Перегляд  Введення  Пристрої  Довідка

admin1@ubuntu-db:~$ sudo -u postgres pgbackrest info
stanza: demo_cluster
status: ok
cipher: aes-256-cbc

db (current)
wal archive min/max (18): 0000000100000000000000003/000000010000000000000005

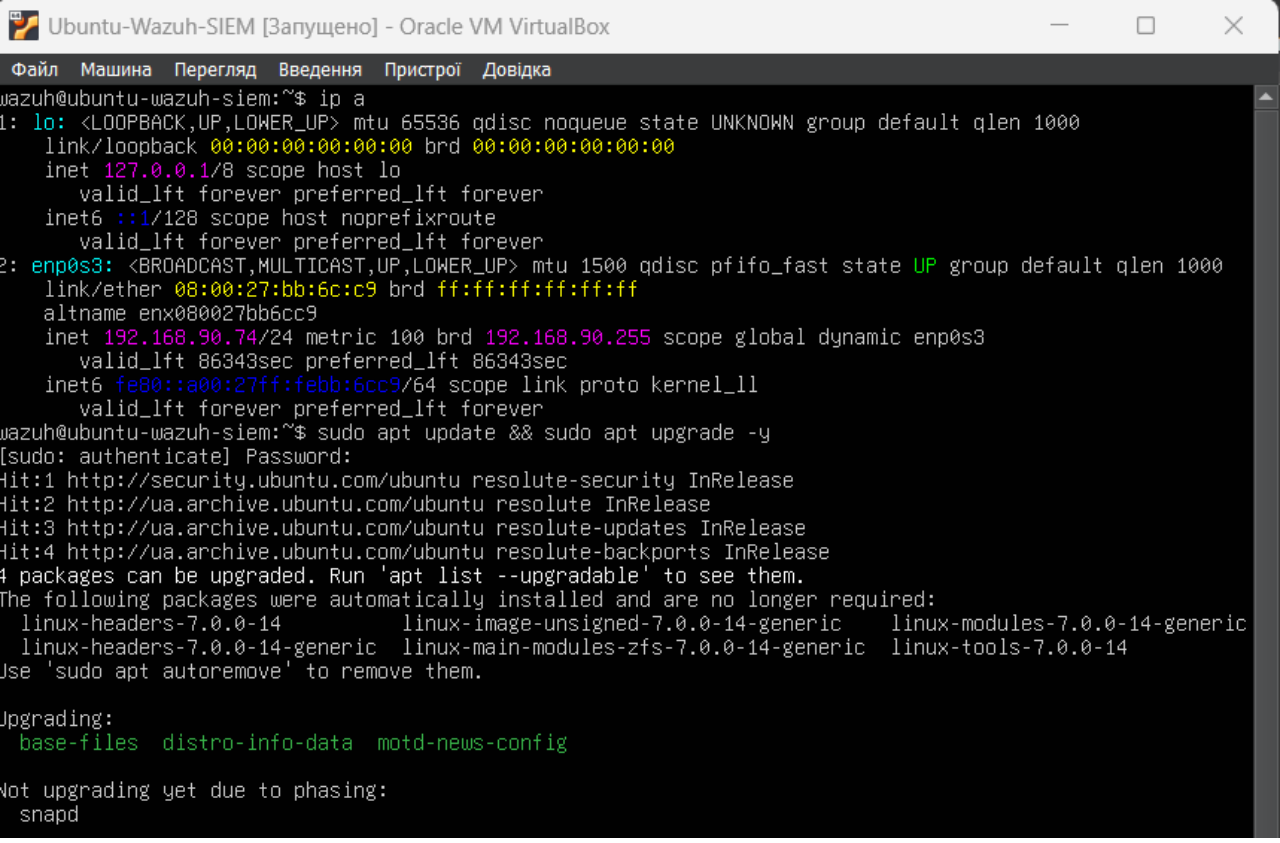
full backup: 20260603-091449F
timestamp start/stop: 2026-06-03 09:14:49+00 / 2026-06-03 09:14:54+00
wal start/stop: 0000000100000000000000003 / 000000010000000000000003
database size: 30.3MB, database backup size: 30.3MB
repo1: backup set size: 4MB, backup size: 4MB

full backup: 20260603-091604F
timestamp start/stop: 2026-06-03 09:16:04+00 / 2026-06-03 09:16:07+00
wal start/stop: 0000000100000000000000005 / 000000010000000000000005
database size: 30.3MB, database backup size: 30.3MB
repo1: backup set size: 4MB, backup size: 4MB

admin1@ubuntu-db:~$
```

Рисунок А.24 – Інформація про бекап

Встановлення WAZUH

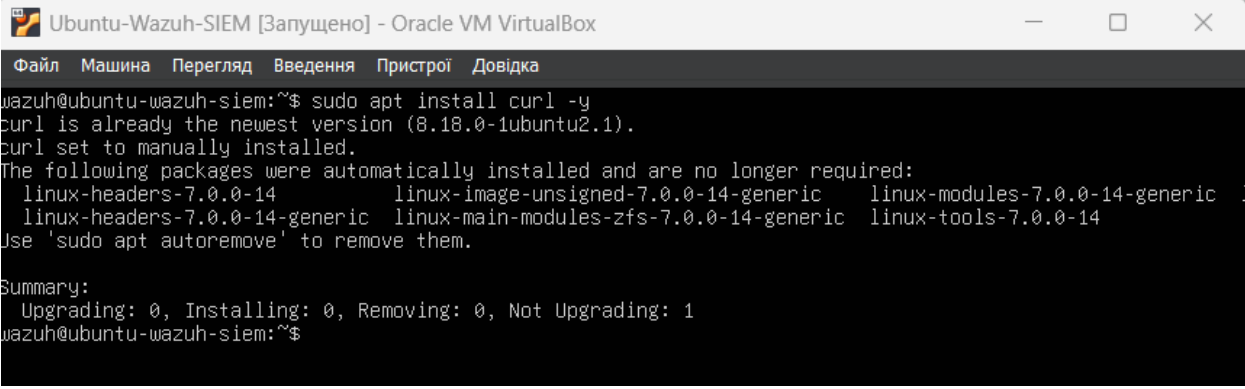


```
Ubuntu-Wazuh-SIEM [Запущено] - Oracle VM VirtualBox
Файл  Машина  Перегляд  Введення  Пристрої  Довідка
wazuh@ubuntu-wazuh-siem:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:bb:6c:c9 brd ff:ff:ff:ff:ff:ff
    altname enx080027bb6cc9
    inet 192.168.90.74/24 metric 100 brd 192.168.90.255 scope global dynamic enp0s3
        valid_lft 86343sec preferred_lft 86343sec
    inet6 fe80::a00:27ff:febb:6cc9/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever
wazuh@ubuntu-wazuh-siem:~$ sudo apt update && sudo apt upgrade -y
[sudo: authenticate] Password:
Hit:1 http://security.ubuntu.com/ubuntu resolute-security InRelease
Hit:2 http://ua.archive.ubuntu.com/ubuntu resolute InRelease
Hit:3 http://ua.archive.ubuntu.com/ubuntu resolute-updates InRelease
Hit:4 http://ua.archive.ubuntu.com/ubuntu resolute-backports InRelease
4 packages can be upgraded. Run 'apt list --upgradable' to see them.
The following packages were automatically installed and are no longer required:
  linux-headers-7.0.0-14          linux-image-unsigned-7.0.0-14-generic  linux-modules-7.0.0-14-generic
  linux-headers-7.0.0-14-generic  linux-main-modules-zfs-7.0.0-14-generic  linux-tools-7.0.0-14
Use 'sudo apt autoremove' to remove them.

Upgrading:
  base-files  distro-info-data  motd-news-config

Not upgrading yet due to phasing:
  snapd
```

Рисунок А.25 – Перевірка мережної конфігурації та оновлення пакетної бази сервера управління SIEM



```
Ubuntu-Wazuh-SIEM [Запущено] - Oracle VM VirtualBox
Файл  Машина  Перегляд  Введення  Пристрої  Довідка
wazuh@ubuntu-wazuh-siem:~$ sudo apt install curl -y
curl is already the newest version (8.18.0-1ubuntu2.1).
curl set to manually installed.
The following packages were automatically installed and are no longer required:
  linux-headers-7.0.0-14          linux-image-unsigned-7.0.0-14-generic  linux-modules-7.0.0-14-generic
  linux-headers-7.0.0-14-generic  linux-main-modules-zfs-7.0.0-14-generic  linux-tools-7.0.0-14
Use 'sudo apt autoremove' to remove them.

Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 1
wazuh@ubuntu-wazuh-siem:~$
```

Рисунок А.26 – Встановлено пакет curl

```
Ubuntu-Wazuh-SIEM [Запущено] - Oracle VM VirtualBox
Файл Машина Перегляд Введення Пристрої Довідка
.04, 22.04; Rocky Linux 9.4.
22/05/2026 18:04:38 WARNING: The current system does not match with the list of recommended systems. The installation may not work properly.
22/05/2026 18:04:45 WARNING: Hardware checks ignored.
22/05/2026 18:04:45 INFO: Wazuh web interface port will be 443.
22/05/2026 18:04:52 INFO: Wazuh repository added.
22/05/2026 18:04:52 INFO: --- Configuration files ---
22/05/2026 18:04:52 INFO: Generating configuration files.
22/05/2026 18:04:52 INFO: Generating the root certificate.
22/05/2026 18:04:52 INFO: Generating Admin certificates.
22/05/2026 18:04:52 INFO: Generating Wazuh indexer certificates.
22/05/2026 18:04:53 INFO: Generating Filebeat certificates.
22/05/2026 18:04:53 INFO: Generating Wazuh dashboard certificates.
22/05/2026 18:04:53 INFO: Created wazuh-install-files.tar. It contains the Wazuh cluster key, certificates, and passwords necessary for installation.
22/05/2026 18:04:53 INFO: --- Wazuh indexer ---
22/05/2026 18:04:53 INFO: Starting Wazuh indexer installation.
22/05/2026 18:05:09 INFO: Wazuh Indexer installation finished.
22/05/2026 18:05:09 INFO: Wazuh indexer post-install configuration finished.
22/05/2026 18:05:09 INFO: Starting service wazuh-indexer.
22/05/2026 18:05:29 INFO: wazuh-indexer service started.
22/05/2026 18:05:29 INFO: Initializing Wazuh indexer cluster security settings.
22/05/2026 18:05:33 INFO: Wazuh indexer cluster security configuration initialized.
22/05/2026 18:05:33 INFO: Wazuh indexer cluster initialized.
22/05/2026 18:05:33 INFO: --- Wazuh server ---
22/05/2026 18:05:33 INFO: Starting the Wazuh manager installation.
22/05/2026 18:07:52 INFO: Wazuh manager installation finished.
22/05/2026 18:07:52 INFO: Wazuh manager vulnerability detection configuration finished.
22/05/2026 18:07:52 INFO: Starting service wazuh-manager.
22/05/2026 18:08:05 INFO: wazuh-manager service started.
22/05/2026 18:08:05 INFO: Starting Filebeat installation.
22/05/2026 18:08:12 INFO: Filebeat installation finished.
22/05/2026 18:08:12 INFO: Filebeat post-install configuration finished.
22/05/2026 18:08:12 INFO: Starting service filebeat.
22/05/2026 18:08:14 INFO: filebeat service started.
22/05/2026 18:08:14 INFO: --- Wazuh dashboard ---
22/05/2026 18:08:14 INFO: Starting Wazuh dashboard installation.
22/05/2026 18:09:57 INFO: Wazuh dashboard installation finished.
22/05/2026 18:09:58 INFO: Wazuh dashboard post-install configuration finished.
22/05/2026 18:09:58 INFO: Starting service wazuh-dashboard.
22/05/2026 18:09:58 INFO: wazuh-dashboard service started.
22/05/2026 18:10:00 INFO: Updating the internal users.
22/05/2026 18:10:05 INFO: A backup of the internal users has been saved in the /etc/wazuh-indexer/internalusers-backup folder.
22/05/2026 18:10:16 INFO: The filebeat.yml file has been updated to use the Filebeat Keystore username and password.
22/05/2026 18:10:44 INFO: Initializing Wazuh dashboard web application.
22/05/2026 18:10:46 INFO: Wazuh dashboard web application initialized.
22/05/2026 18:10:46 INFO: --- Summary ---
22/05/2026 18:10:46 INFO: You can access the web interface https://<wazuh-dashboard-ip>:443
User: admin
Password: cYk?qgBV1pr21pbXrMG00szI76kgnPi
22/05/2026 18:10:46 INFO: Installation finished.
wazuh@ubuntu-wazuh-siem:~$
```

Рисунок А.27 – Процес автоматизованого розгортання компонентів Wazuh All-in-One

```
Admin-Jumpbox [Запущено] - Oracle VM VirtualBox
Файл Машина Перегляд Введення Пристрої Довідка
admin1@ubuntu-db: ~
Session Actions Edit View Help
admin1@ubuntu-db: ~ x
admin1@ubuntu-db:~$ wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.14.5-1_amd64.deb && sudo WAZUH_MANAGER='192.168.90.74' WAZUH_AGENT_NAME='Ubuntu-DB' dpkg -i ./wazuh-agent_4.14.5-1_amd64.deb
--2026-05-27 06:23:04-- https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.14.5-1_amd64.deb
Resolving packages.wazuh.com (packages.wazuh.com)... 3.174.230.58, 3.174.230.69, 3.174.230.3, ...
Connecting to packages.wazuh.com (packages.wazuh.com)[3.174.230.58]:443...
failed: Connection timed out.
Connecting to packages.wazuh.com (packages.wazuh.com)[3.174.230.69]:443... failed: Connection timed out.
Connecting to packages.wazuh.com (packages.wazuh.com)[3.174.230.3]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 13214566 (13M) [application/vnd.debian.binary-package]
Saving to: 'wazuh-agent_4.14.5-1_amd64.deb'

wazuh-agent_4.14.5-1_a 100%[=====] 12.60M 5.10MB/s in 2.5s

2026-05-27 06:27:38 (5.10 MB/s) - 'wazuh-agent_4.14.5-1_amd64.deb' saved [13214566/13214566]

[sudo: authenticate] Password:
sudo: Authentication failed, try again.
[sudo: authenticate] Password:
```

Рисунок А.28 – Завантаження пакетів на станцію з бд

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				106
Змн.	Арк.	№ докум.	Підпис	Дата		

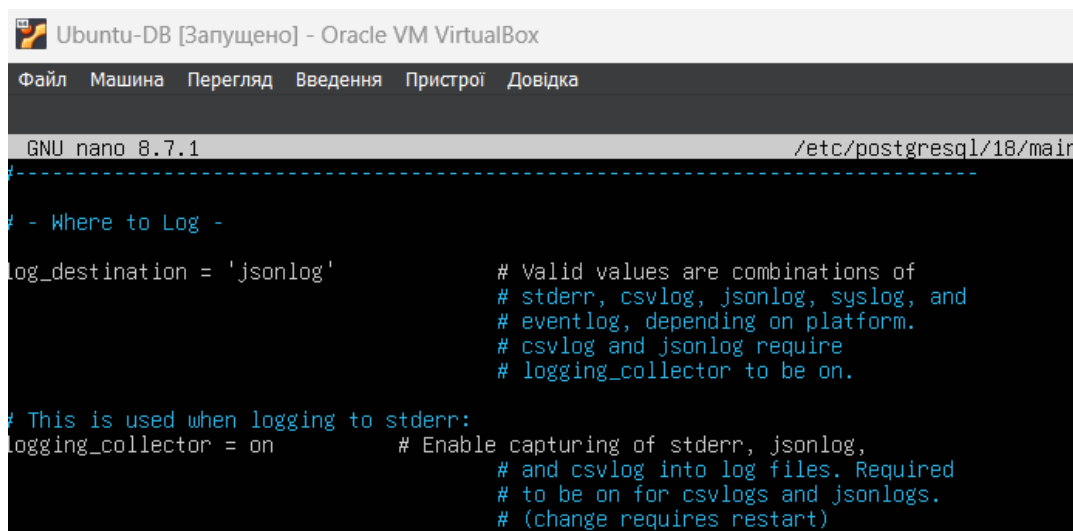
```

admin1@ubuntu-db:~$ sudo systemctl daemon-reload
sudo systemctl enable wazuh-agent
sudo systemctl start wazuh-agent
Created symlink '/etc/systemd/system/multi-user.target.wants/wazuh-agent.service' -> '/usr/lib/systemd/system/wazuh-agent.service'.
admin1@ubuntu-db:~$

```

Рисунок А.29 – Активація демона

Розширений збір та нормалізація подій аудиту PostgreSQL у Wazuh за допомогою JSON та власних правил кореляції (pgAudit)



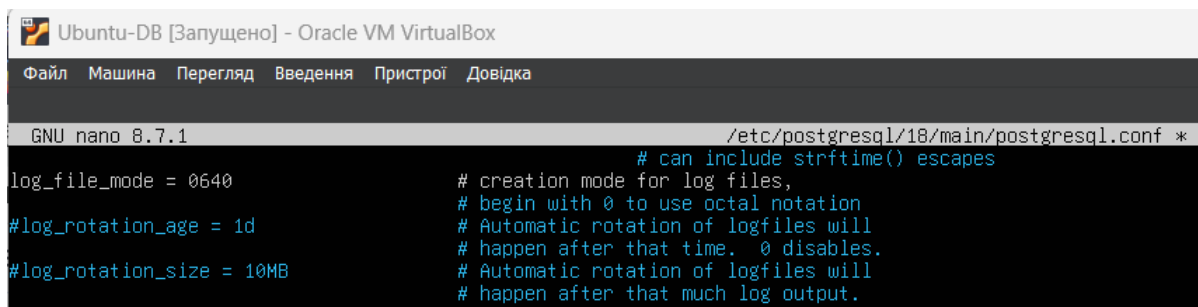
```

GNU nano 8.7.1 /etc/postgresql/18/main/
#-----
# - Where to Log -
log_destination = 'jsonlog'           # Valid values are combinations of
                                       # stderr, csvlog, jsonlog, syslog, and
                                       # eventlog, depending on platform.
                                       # csvlog and jsonlog require
                                       # logging_collector to be on.

# This is used when logging to stderr:
logging_collector = on                # Enable capturing of stderr, jsonlog,
                                       # and csvlog into log files. Required
                                       # to be on for csvlogs and jsonlogs.
                                       # (change requires restart)

```

Рисунок А.30 – Налаштування конфігураційного файлу

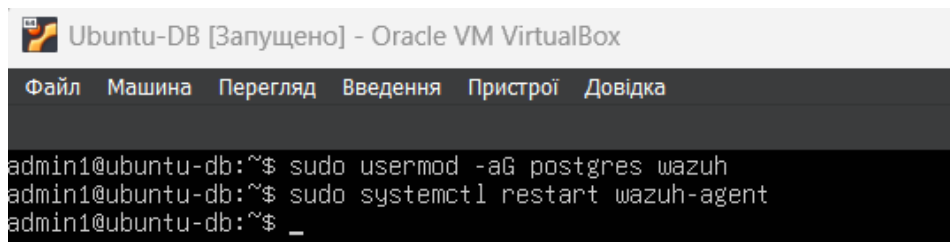


```

GNU nano 8.7.1 /etc/postgresql/18/main/postgresql.conf *
# can include strftime() escapes
log_file_mode = 0640                 # creation mode for log files,
                                       # begin with 0 to use octal notation
#log_rotation_age = 1d                # Automatic rotation of logfiles will
                                       # happen after that time. 0 disables.
#log_rotation_size = 10MB             # Automatic rotation of logfiles will
                                       # happen after that much log output.

```

Рисунок А.31 – Зміна прав доступу



```

admin1@ubuntu-db:~$ sudo usermod -aG postgres wazuh
admin1@ubuntu-db:~$ sudo systemctl restart wazuh-agent
admin1@ubuntu-db:~$ _

```

Рисунок А.32 – Вирішення проблеми читання

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				107
Змн.	Арк.	№ докум.	Підпис	Дата		

Ubuntu-DB [Запущено] - Oracle VM VirtualBox

Файл Машина Перегляд Введення Пристрої Довідка

GNU nano 8.7.1 /var/ossec/etc/ossec.conf *

```
<!-- Log analysis -->
<localfile>
  <log_format>command</log_format>
  <command>df -P</command>
  <frequency>360</frequency>
</localfile>

<localfile>
  <log_format>full_command</log_format>
  <command>netstat -tuln | sed 's/\([[:alnum:]]\+\) \+([[:digit:]]\+\) \+([[:digit:]]\+\) \+\.(\.*)\([[:digit:]]*\)\ \+([0-9]'
  <alias>netstat listening ports</alias>
  <frequency>360</frequency>
</localfile>

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

<localfile>
  <log_format>json</log_format>
  <location>/var/lib/postgresql/18/main/log/*.json</location>
</localfile>
```

The image shows a web browser window with the Wazuh interface. The browser has several tabs open: 'Health | Reporting | OPN: x', 'Wazuh', 'YouTube', and 'Telegram Web'. The address bar displays the URL '192.168.90.74/app/rules#/manager/?tab=ruleset'. The page title is 'Wazuh — Mozilla Firefox'. The main content area is titled 'local_rules.xml' and contains an XML configuration for rules. The XML defines two rules: Rule 100001 (level 3) for PostgreSQL system logs and Rule 100002 (level 5) for PostgreSQL security events. The interface includes a 'Ruleset Test' button and a 'Save' button.

Налаштування пересилання логів мережевого екрана OPNsense та IDS Suricata у Wazuh

		Сімашко І. Б.			<i>ККІКБ.КР.Б – 123 – 13 – ПЗ</i>	Арк.
		Дячук О.Ю.				108
Змн.	Арк.	№ докум.	Підпис	Дата		

Налаштування розпочалося зі сторони Wazuh Manager: у головному конфігураційному файлі `/var/ossec/etc/ossec.conf` було додано новий блок `<remote>`. Це дозволило серверу управління відкрити мережевий порт 514 (UDP) та приймати вхідний потік логів виключно від довіреної IP-адреси шлюзу 192.168.90.1.

```

GNU nano 8.7.1

<remote>
  <connection>secure</connection>
  <port>1514</port>
  <protocol>tcp</protocol>
  <queue_size>131072</queue_size>
</remote>

<remote>
  <connection>syslog</connection>
  <port>514</port>
  <protocol>udp</protocol>
  <allowed-ips>192.168.90.1</allowed-ips>
</remote>
  
```

Рисунок А.35 – Налаштування Wazuh Manager для прийому мережевих подій через протокол Syslog

```

GNU nano 8.7.1 /var/ossec/etc/rules/local_rules.xml

<if_sid>100001</if_sid>
  <field name="message" type="pcre2">AUDIT: SESSION</field>
  <description>PostgreSQL Security: Зафіксовано звернення до конфіденційних даних (pgAudit)</description>
  <group>database_audit, gdpr, pci_dss_10.2,</group>
</rule>

<rule id="100010" level="10">
  <program_name>suricata</program_name>
  <description>IDS Suricata: Виявлено підозрілу мережеву активність.</description>
  <group>ids,suricata</group>
</rule>

<rule id="100011" level="12">
  <if_sid>100010</if_sid>
  <match>GPL ATTACK_RESPONSE</match>
  <description>IDS Suricata: Успішна атака або витік даних (ATTACK_RESPONSE).</description>
  <group>ids,suricata,attack</group>
</rule>
</group>
  
```

Рисунок А.36 – Розробка правил кореляції для ідентифікації мережевих атак із логів Suricata

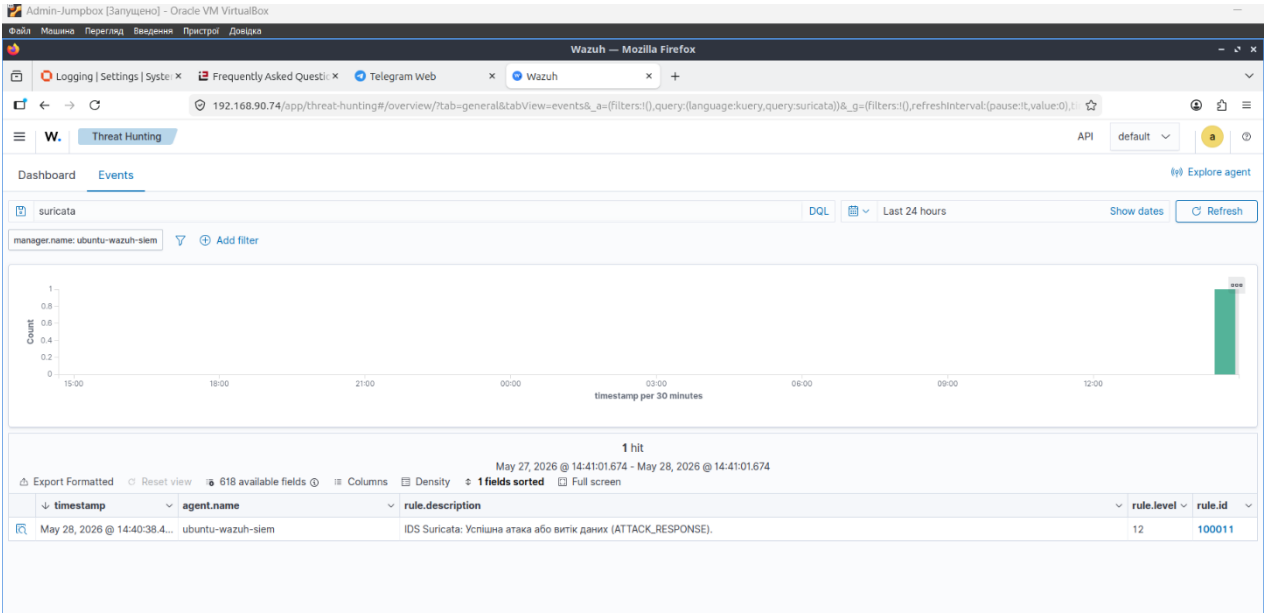


Рисунок А.37 – Відображення перехопленої мережної атаки на панелі інцидентів Wazuh

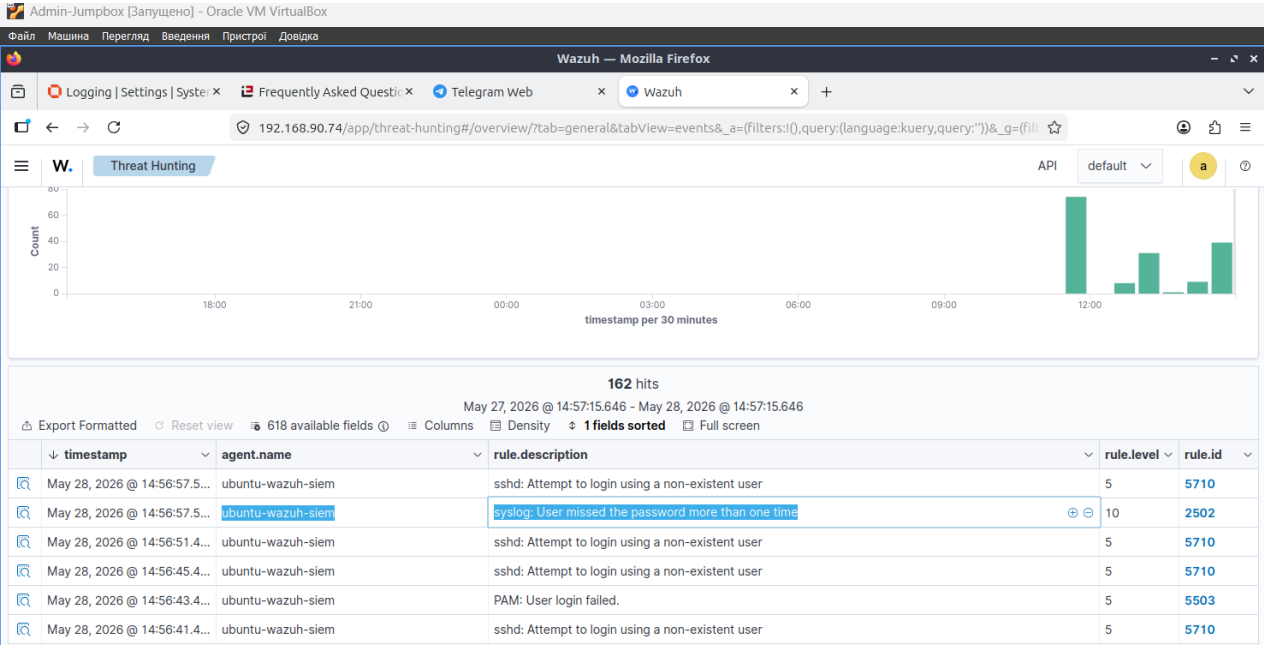


Рисунок А.38 – Детекція аномалій автентифікації та генерація відповідних сповіщень безпеки у Wazuh

```

GNU nano 8.7.1 /var/ossec/etc/ossec.conf *
<!--
<active-response>
  active-response options here
</active-response>
-->

```xml
<active-response>
 <command>firewall-drop</command>
 <location>local</location>
 <rules_id>5710, 5712, 5716, 5720</rules_id>
 <timeout>60</timeout>
</active-response>

```

Рисунок А.39 – Конфігурація модуля Active Response для автоматичного блокування мережевих атак

```

wazuh@ubuntu-wazuh-siem:~$ sudo nano /var/ossec/etc/ossec.conf
[sudo: authenticate] Password:
wazuh@ubuntu-wazuh-siem:~$ sudo systemctl restart wazuh-manager
wazuh@ubuntu-wazuh-siem:~$

```

Рисунок А.40 – Перезапуск служби

## Налаштування Port Forwarding

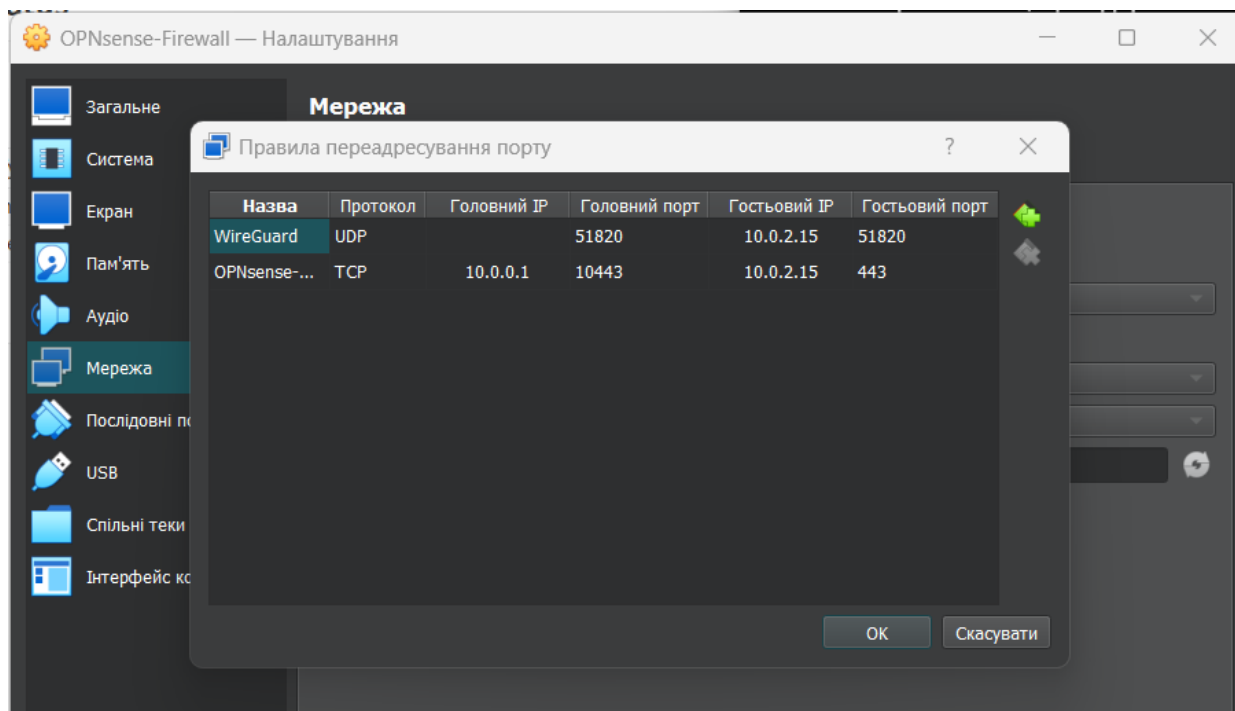


Рисунок А.41 – Налаштовано перекидання портів на OPNsense машині

## Встановлення операційних систем Linux

## Віртуальна машина Ubuntu-DB

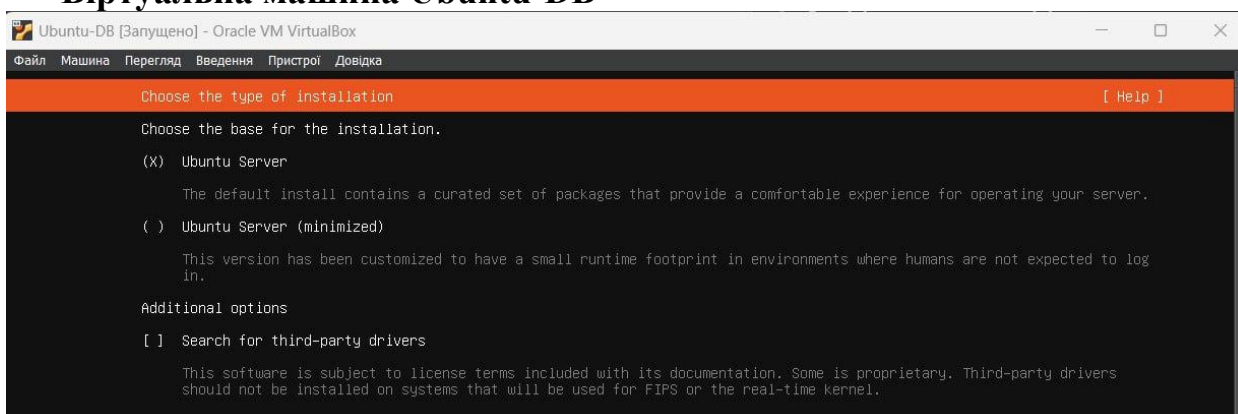


Рисунок Б.1 – Вибир типу інсталяції сервера

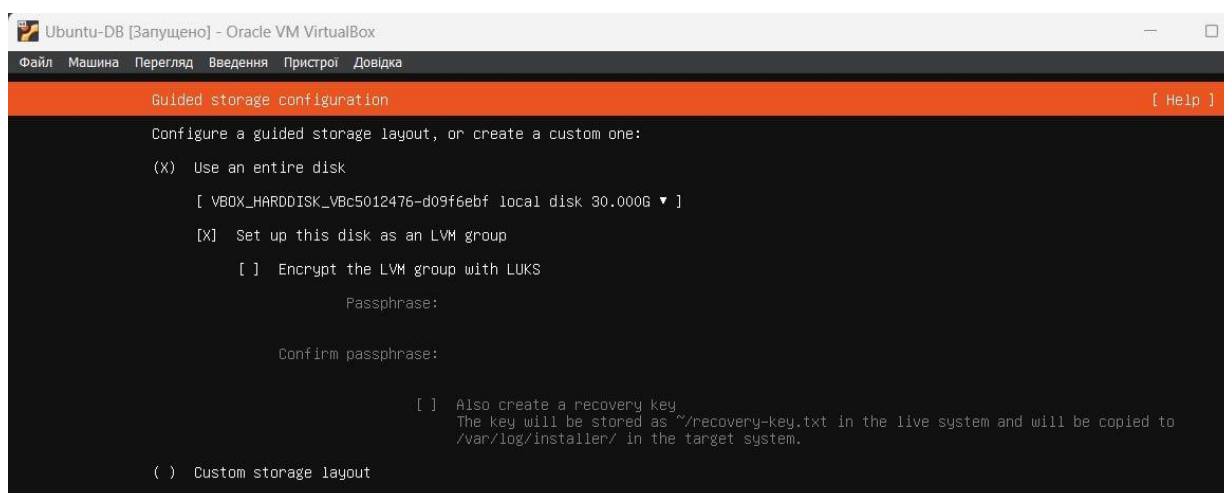


Рисунок Б.2 – Налаштування диску на Ubuntu-DB

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				113
Змн.	Арк.	№ докум.	Підпис	Дата		

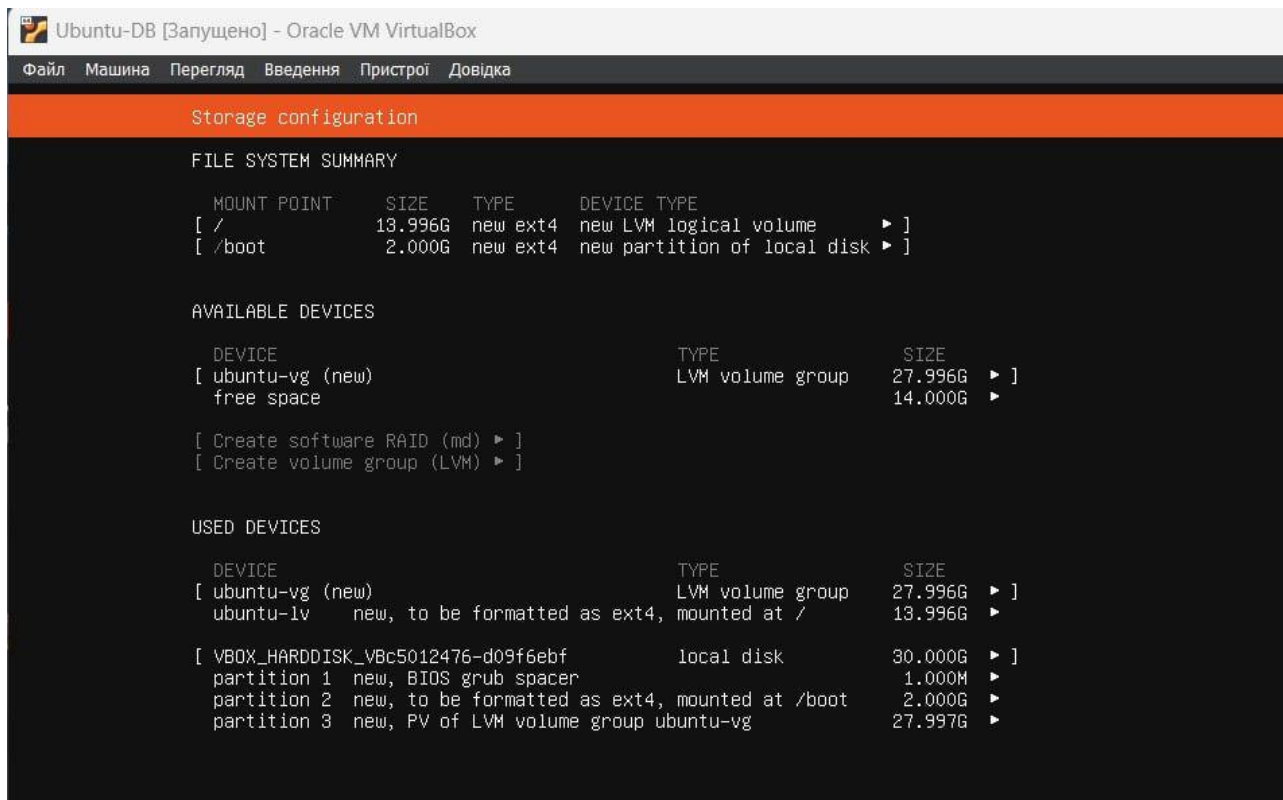


Рисунок Б.3 – Розширена конфігурація диску на Ubuntu-DB

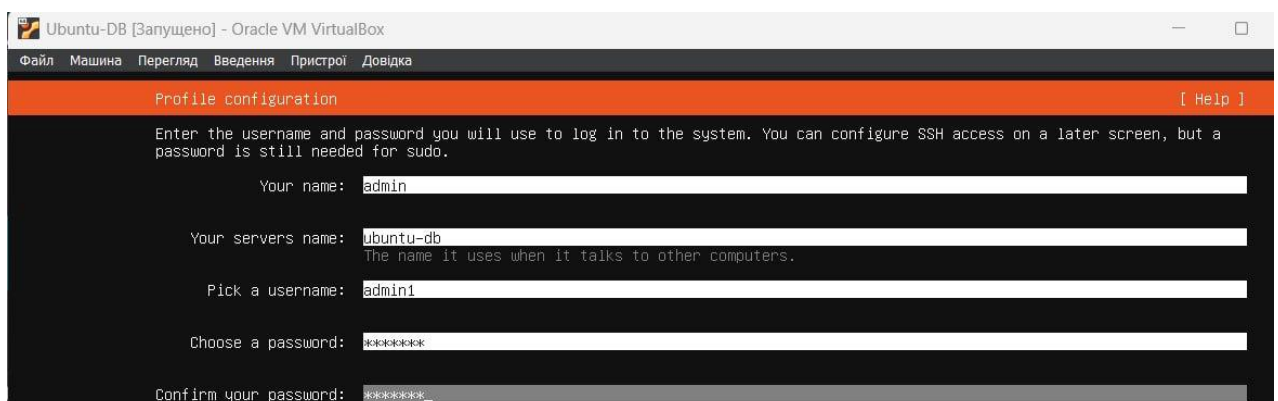


Рисунок Б.4 – Створення користувача

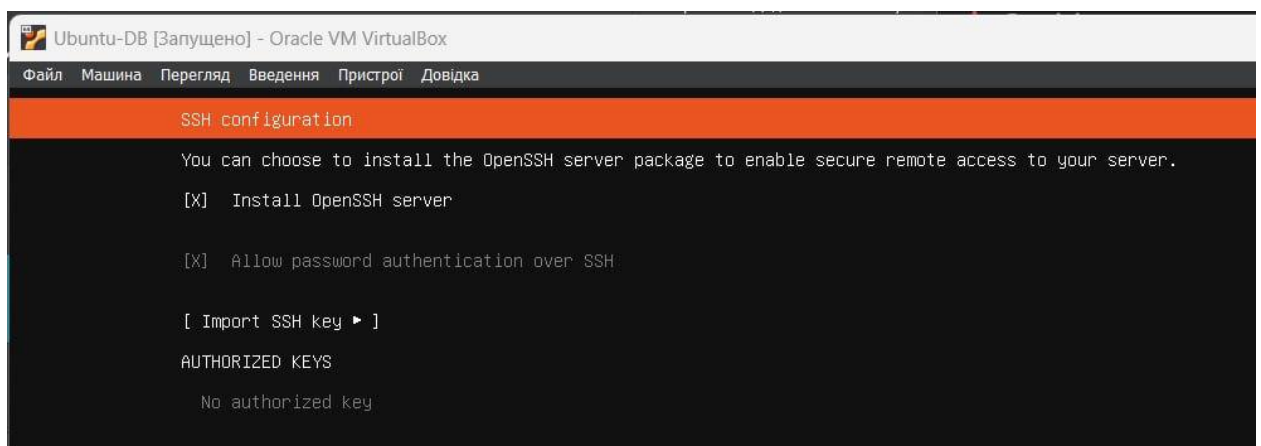


Рисунок Б.5 – Встановлення пакунку SSH

```

Ubuntu-DB [Запущено] - Oracle VM VirtualBox
Файл Машина Перегляд Введення Пристрої Довідка
[135.502051] cloud-init[1251]: | u s b . . |
[135.503086] cloud-init[1251]: | = = +0. . |
[135.504947] cloud-init[1251]: | . o . . |
[135.505412] cloud-init[1251]: | . +E . |
[135.505643] cloud-init[1251]: | .Bo |
[135.505810] cloud-init[1251]: +----[SHA256]-----+
[135.506269] cloud-init[1251]: Generating public/private ecdsa key pair.

[135.509264] cloud-init[1251]: Your public key has been saved in /etc/ssh/ssh_host_ecdsa_key.pub
[135.511544] cloud-init[1251]: The key fingerprint is:
[135.512229] cloud-init[1251]: SHA256:olzaEvb/znlcxN12Ku0uicIHE33xG+dR0luq8RD6KAs root@ubuntu-db
[135.512428] cloud-init[1251]: The key's randomart image is:
[135.512564] cloud-init[1251]: +---[ECDSA 256]---+
[135.512859] cloud-init[1251]: |
[135.513413] cloud-init[1251]: | . . . + |
[135.513615] cloud-init[1251]: | . + o =+ |
[135.513935] cloud-init[1251]: | . . o = * = |
[135.514163] cloud-init[1251]: | o o.S. o @ +. |
[135.514848] cloud-init[1251]: | o BE. . * = |
[135.516684] cloud-init[1251]: | =.o+ o o + |
[135.516902] cloud-init[1251]: | .o.o o+ |
[135.517042] cloud-init[1251]: | oooB+. |
[135.517199] cloud-init[1251]: +----[SHA256]-----+
[135.517436] cloud-init[1251]: Generating public/private ed25519 key pair.
[135.517582] cloud-init[1251]: Your identification has been saved in /etc/ssh/ssh_host_ed25519_key
[135.517717] cloud-init[1251]: Your public key has been saved in /etc/ssh/ssh_host_ed25519_key.pub
[135.517844] cloud-init[1251]: The key fingerprint is:
[135.517979] cloud-init[1251]: SHA256:9RiZiUAEY00xPbhlQuiemfht+ioGzlmYtuWo1mK8itI root@ubuntu-db
[135.518227] cloud-init[1251]: The key's randomart image is:
[135.518421] cloud-init[1251]: +---[ED25519 256]---+
[135.518780] cloud-init[1251]: | .B=+ |
[135.518918] cloud-init[1251]: | . . * . . + |
[135.520407] cloud-init[1251]: | . . o o . * |
[135.520811] cloud-init[1251]: | + |
[135.521004] cloud-init[1251]: | . + o S . . |
[135.521353] cloud-init[1251]: | +++= . |
[135.521770] cloud-init[1251]: | B*0 . |
[135.521980] cloud-init[1251]: | +&E+. |
[135.522283] cloud-init[1251]: | @o*=o |
[135.523001] cloud-init[1251]: +----[SHA256]-----+

Ubuntu 26.04 LTS ubuntu-db tty1
ubuntu-db login: ud-init[1251]: Your identification has been saved in /etc/ssh/ssh_host_ecdsa_key
ubuntu-db login:

```

Рисунок Б.6 – Успішний запуск та генерація SSH ключа на Ubuntu-DB

## Віртуальна машина Ubuntu-Wazuh

```

Ubuntu-Wazuh-SIEM [Запущено] - Oracle VM VirtualBox
Файл Машина Перегляд Введення Пристрої Довідка
Network configuration [Help]

Configure at least one interface this server can use to talk to other machines, and which preferably provides sufficient
access for updates.

NAME TYPE NOTES
[enp0s3 eth -
 DHCPv4 192.168.90.74/24
 DHCPv6 -
 08:00:27:bb:6c:c9 / Intel Corporation / 82540EM Gigabit Ethernet Controller (PRO/1000 MT Desktop Adapter)
[Create bond ►]

```

Рисунок Б.7 – Налаштування інтерфейсу, IP

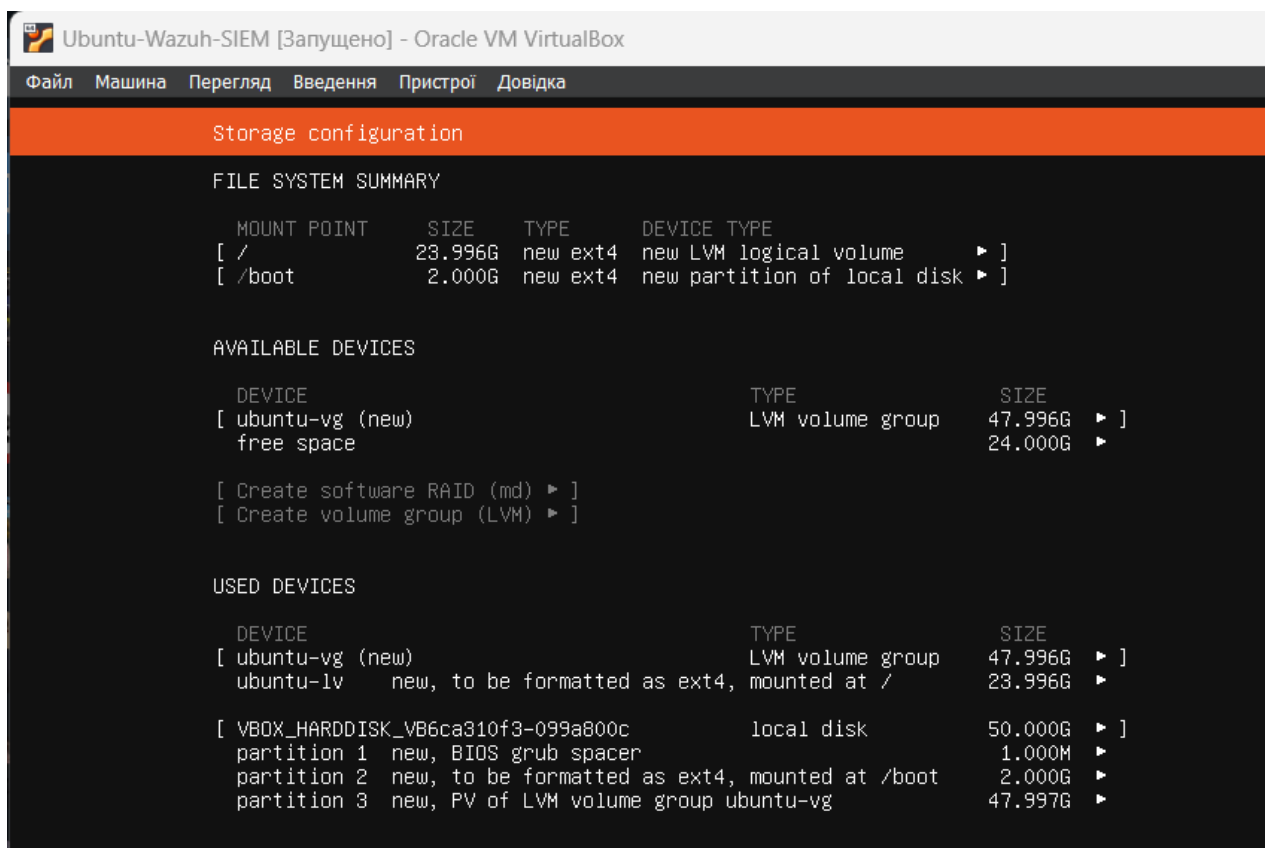


Рисунок Б.8 – Налаштування диску

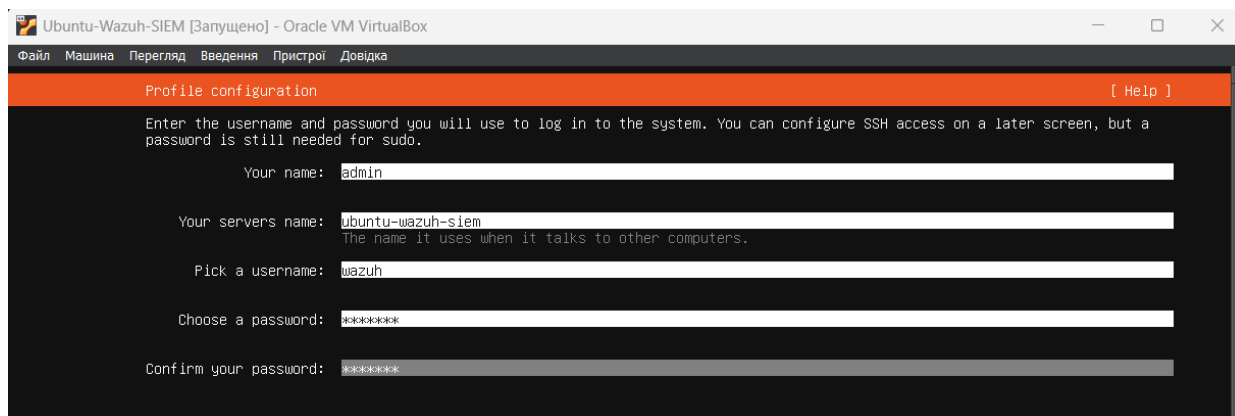


Рисунок Б.9 – Створення користувача

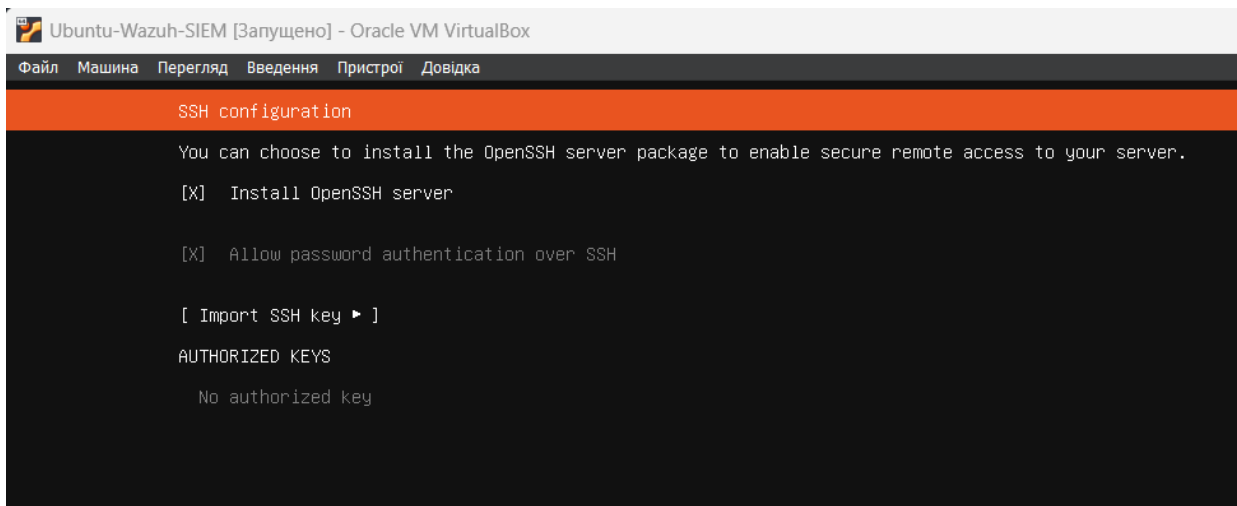


Рисунок Б.10 – Встановлення SSH пакету

## Віртуальна машина Admin-Jumpbox (lubuntu)

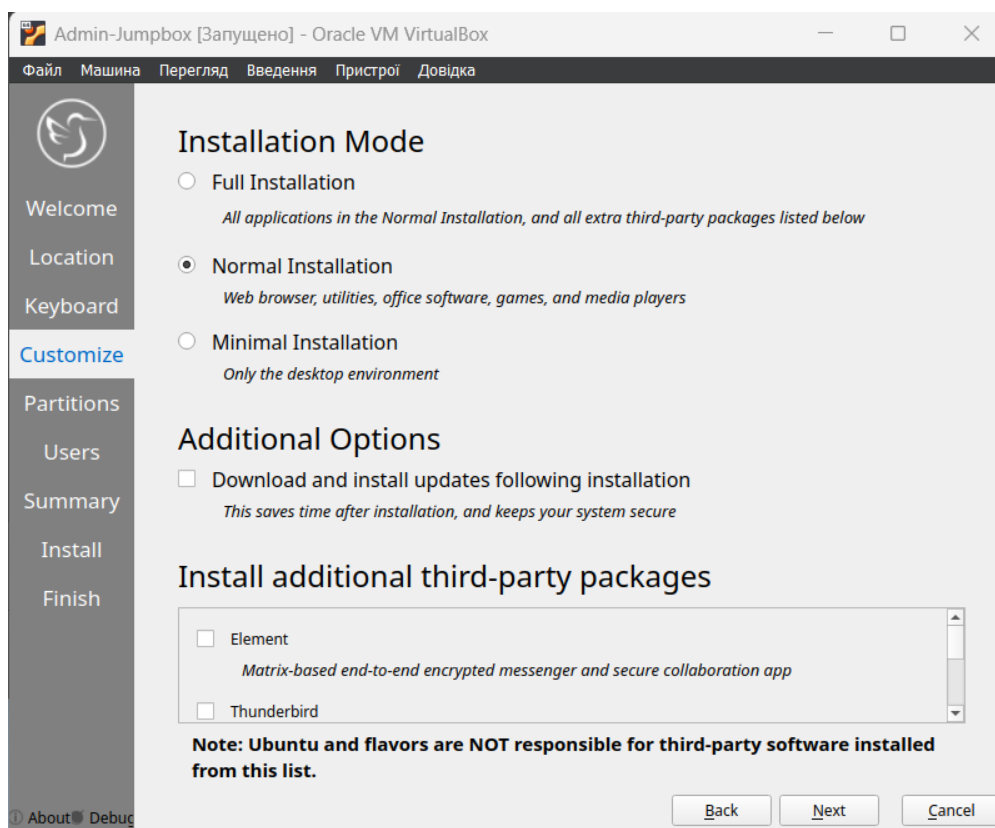


Рисунок Б.11 – Режим інсталяції lubuntu

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				117
Змн.	Арк.	№ докум.	Підпис	Дата		

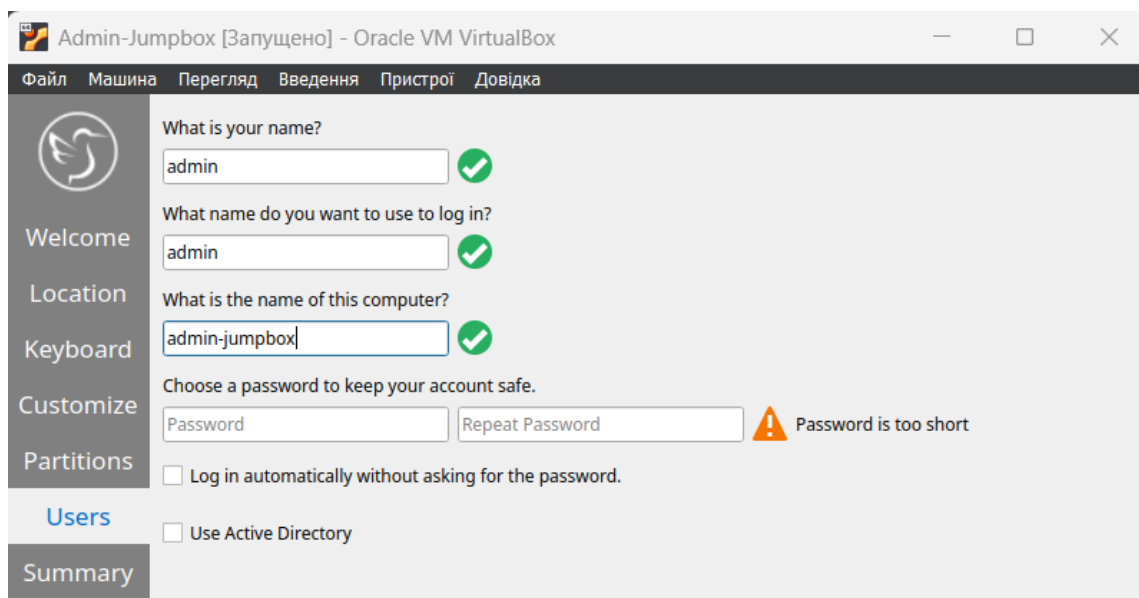


Рисунок Б.12 – Створення користувача

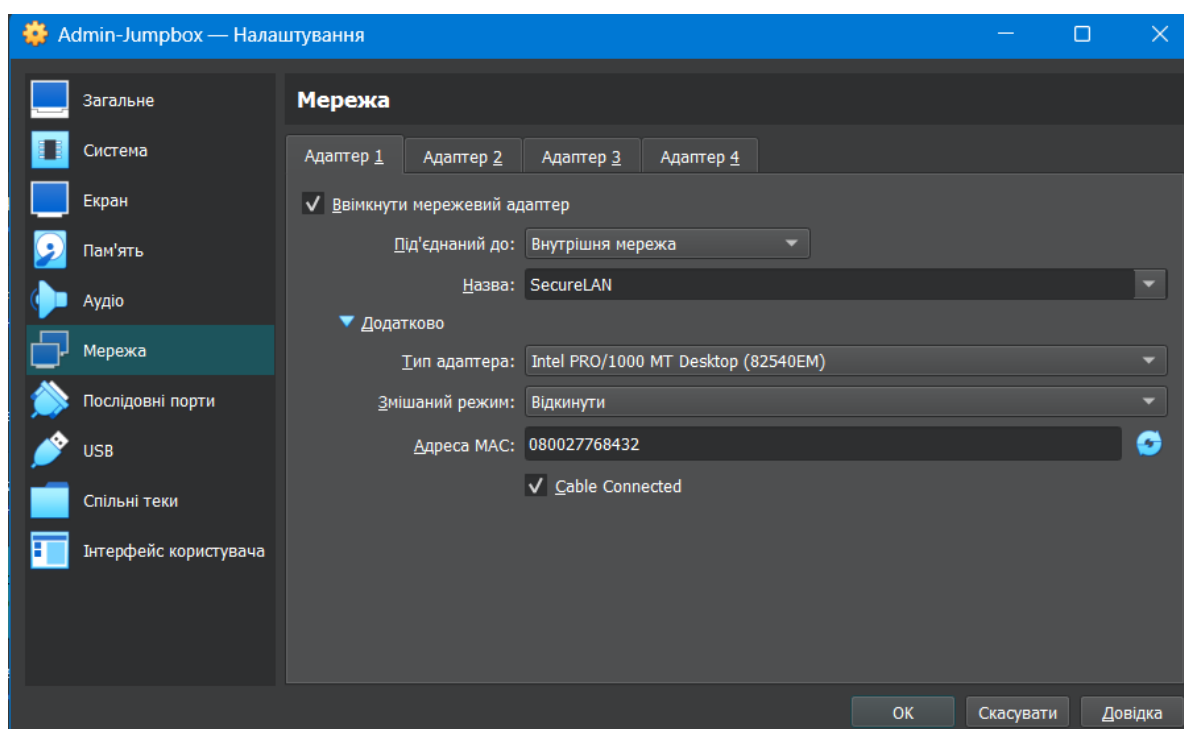


Рисунок Б.13 – Налаштування мережевого адаптера

OPNsense

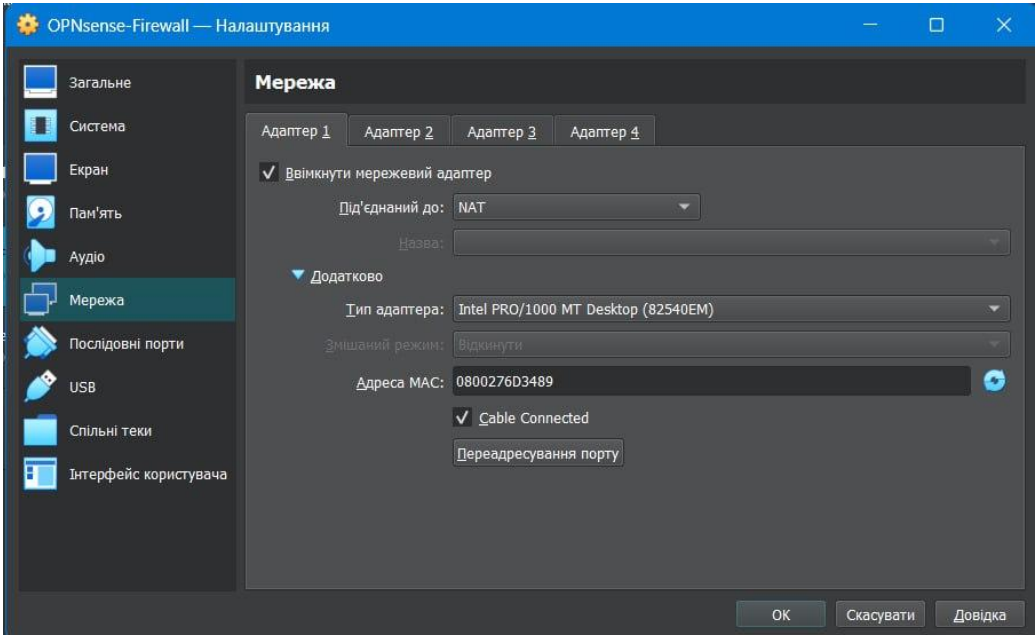


Рисунок Б.14 – Налаштування мережевого адаптера WAN

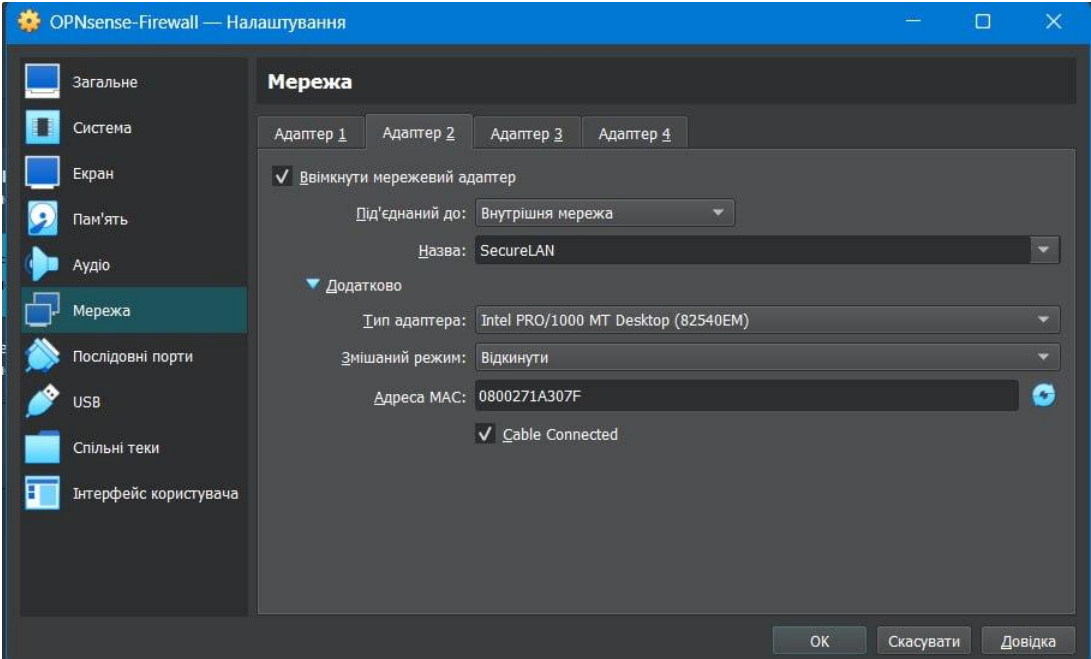


Рисунок Б.15 – Налаштування мережевого адаптера LAN

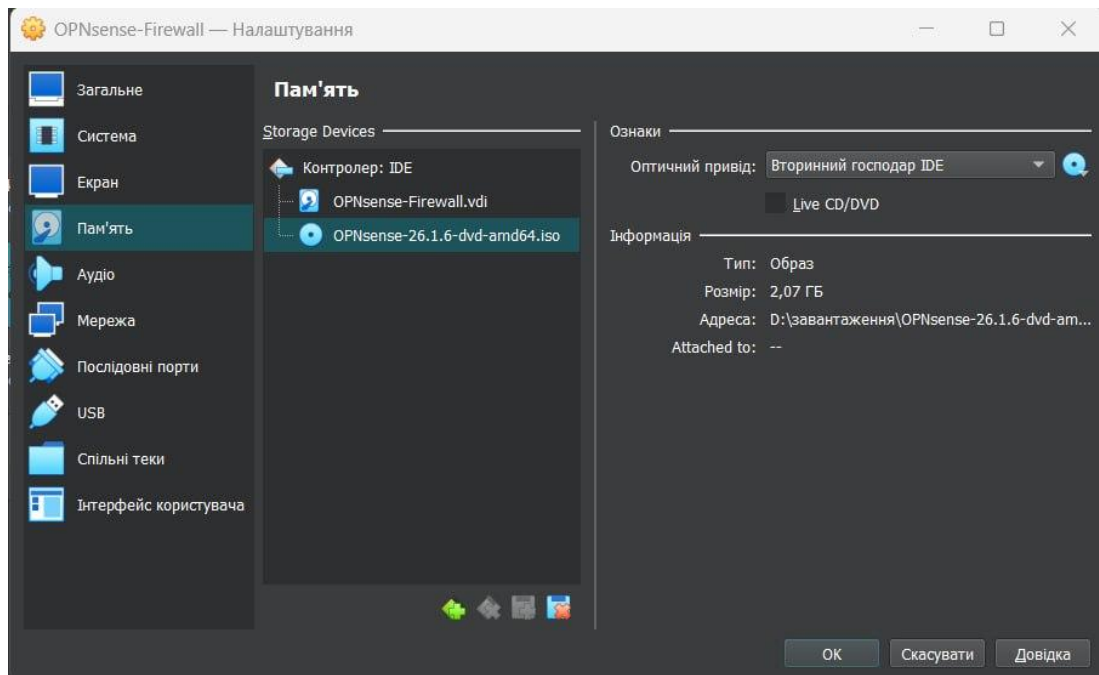


Рисунок Б.16 – Обрано диск для встановлення образу

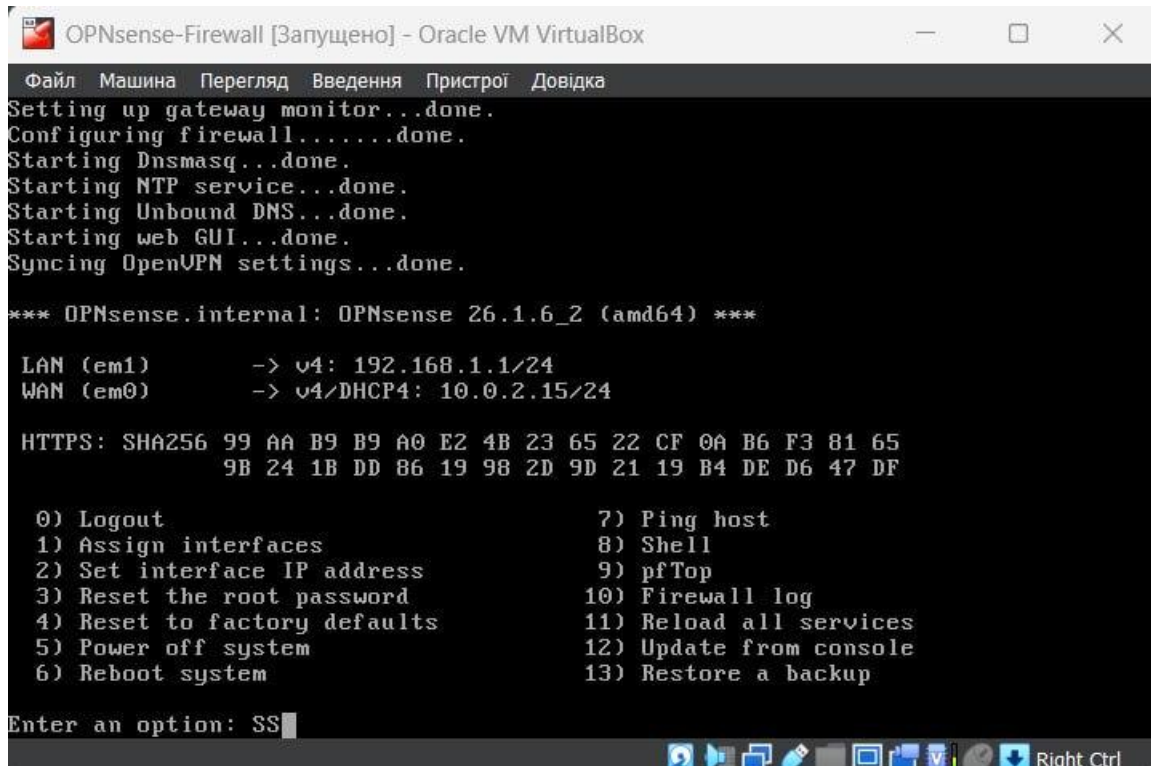


Рисунок Б.17 – Успішний запуск системи

		Сімашко І. Б.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Дячук О.Ю.				120
Змн.	Арк.	№ докум.	Підпис	Дата		