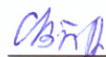


Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра
на тему: «Проект реалізації Honeypot для захисту
публічних портів корпоративної мережі на базі MS Active
Directory»

Виконав: студент 4-го курсу групи КБ-21-1
спеціальності 125 «Кібербезпека»

 Владислав ЦЕВЧУК

Керівник: професор кафедри КІ та КБ, д.т.н., доцент

 Володимир ВОРОТНІКОВ

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

 Олена ГОЛОВНЯ

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

ЗАТВЕРДЖУЮ

Завідувач

кафедри комп'ютерної
інженерії та кібербезпеки

 Андрій Єфіменко
3 березня 2025 р.

ЗАВДАННЯ

на випускню кваліфікаційну роботу бакалавра

Студент: Цевчук Владислав Сергійович

Тема роботи: «Проект реалізації Honeypot для захисту публічних портів
корпоративної мережі на базі MS Active Directory»

затверджена Наказом університету від 3 березня 2025 р. № 80/с

Термін здачі студентом закінченої роботи 24 червня 2025 р.

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

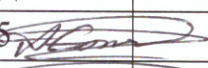
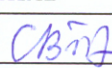
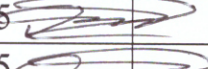
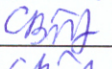
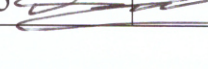
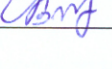
Об'єктом дослідження є процес забезпечення безпеки публічних портів у
корпоративних мережах.

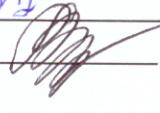
Предметом дослідження є технології побудови Honeypot-підсистем для
виявлення атак на публічні порти.

Обладнання та програмне забезпечення:

1. Гіпервізор Oracle VirtualBox
2. Операційна система Ubuntu Server
3. Операційна система Windows Server
4. Утиліта Cowrie
5. Утиліта Honeyhttpd
6. Kali Linux

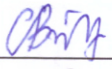
Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Єфіменко А.А.	22.04.2025 	24.04.2025 
2	Єфіменко А.А.	26.04.2025 	28.04.2025 
3	Єфіменко А.А.	01.05.2025 	05.05.2025 

Студент  Владислав ЦЕВЧУК
 Керівник  Володимир ВОРОТНИКОВ

Календарний план

№ з/п	Назви етапів курсової роботи	Строк виконання етапів роботи	Примітки
1	Формулювання мети, завдань і об'єкта дослідження	9.04.2025	Виконано
2	Дослідження літератури	14.04.2025	Виконано
3	Огляд та вибір інструментів реалізації	19.04.2025	Виконано
4	Написання першого розділу	24.04.2025	Виконано
5	Моделювання віртуальної мережевої інфраструктури	03.05.2025	Виконано
6	Написання другого розділу	07.05.2025	Виконано
7	Розгортання Honeypot і Active Directory	10.05.2025	Виконано
8	Тестування безпеки та фіксація атак	27.05.2025	Виконано
9	Аналіз результатів і перевірка працездатності системи	04.06.2025	Виконано
10	Написання третього розділу	14.06.2025	Виконано
11	Оформлення пояснювальної записки	19.06.2025	Виконано
12	Захист кваліфікаційної роботи	24.06.2025	Виконано

Студент  Владислав ЦЕВЧУК
 Керівник  Володимир ВОРОТНИКОВ

АНОТАЦІЯ

Кваліфікаційна робота на тему “Проект реалізації honeypot для захисту публічних портів корпоративної мережі на базі MS Active Directory” складається зі вступу, трьох розділів, висновку, списку використаної літератури та додатку.

Текстова частина складається з 65 сторінок друкованого тексту.

Пояснювальна записка має 1 додаток. Список використаних джерел містить 20 найменувань та займає 2 сторінки. В роботі наведено 59 рисунків. Загальний опис роботи – 65 сторінок.

У першому розділі досліджено технологію та концепцію Honeypot у сфері кібербезпеки, його загальне призначення, типи, переваги, недоліки та роль у корпоративних мережах.

У другому розділі було спроектовано віртуалізовану мережеву інфраструктуру з інтеграцією Honeypot-підсистеми для моніторингу активності на відкритих сервісах, зокрема SSH, Telnet, HTTP та HTTPS.

Третій розділ включає в себе вибір, налаштування та тестування мережі.

Загальною метою курсового проєкту є дослідити концепцію Honeypot та впровадити її у корпоративну мережу для підвищення рівня кібербезпеки.

Ключові слова: Honeypot, кібербезпека, зловмисники, загрози, моніторинг, безпека мережі, Active Directory.

					ККІКБ.КР.Б – 125 – 19 – ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата	Проект реалізації honeypot для захисту публічних портів корпоративної мережі на базі MS Active Directory	Лім.	Арк.	Аркушів	
Розроб.		Цевчук В.С.		25.06.25					
Керівник		Воротніков В.В.		25.06.25					
Реценз.		Головня О.С.		25.06.25					
Н. контр.		Воротніков В.В.		25.06.25					
Затверд.		Єфіменко А.А.		25.06.25		ФІКТ Гр. КБ-21-1			

ANNOTATION

The qualification work on the topic “Project of honeypot implementation for protecting public ports of a corporate network based on MS Active Directory” consists of an introduction, three chapters, a conclusion, a list of references and an appendix.

The text part consists of 65 pages of printed text.

The explanatory note has 1 appendix. The list of references includes 20 titles and occupies 2 pages. The paper contains 59 figures. The total description of the work is 65 pages.

The first section explores the technology and concept of Honeypot in the field of cybersecurity, its general purpose, types, advantages, disadvantages and role in corporate networks.

In the second section, we designed a virtualised network infrastructure with the integration of a Honeypot subsystem for monitoring activity on open services, including SSH, Telnet, HTTP, and HTTPS.

The third section includes network selection, configuration, and testing.

The overall goal of the course project is to explore the concept of Honeypot and implement it in a corporate network to improve cybersecurity.

Keywords: Honeypot, cybersecurity, attackers, threats, monitoring, network security, Active Directory.

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. ТЕОРЕТИЧНІ ПРИНЦИПИ ПОБУДОВИ СИСТЕМ HONEYPOT	8
1.1 Honeypot-системи: концепція, функціональність і класифікація	8
1.2 Роль Honeypot у корпоративних мережах	11
1.3. Переваги та недоліки використання Honeypot у захисті мережі	13
1.4 Сервіси MS Active Directory та їхня взаємодія через публічні порти	16
1.5 Захист публічних портів у корпоративних мережах на основі Honeypot-технологій	18
РОЗДІЛ 2. ПРОЕКТУВАННЯ HONEYPOT-ПІДСИСТЕМИ ДЛЯ ЗАХИСТУ ПУБЛІЧНИХ ПОРТІВ КОРПОРАТИВНОЇ МЕРЕЖІ	20
2.1 Принципи побудови мережевої архітектури з використанням Honeypot	20
2.2 Проектування Honeypot-підсистеми для корпоративної мережної інфраструктури	21
2.3 Проектування Honeypot для сервісів віддаленого доступу	22
2.4 Проектування Honeypot для веб-сервісів	22
РОЗДІЛ 3. ВИБІР ПРОГРАМНОГО РІШЕННЯ ТА НАЛАГОДЖЕННЯ HONEYPOT-ПІДСИСТЕМИ В КОРПОРАТИВНІЙ МЕРЕЖІ	24
3.1 Вибір програмного забезпечення та інструментів для реалізації Honeypot-системи	24
3.2 Налаштування Honeypot-системи в спроектованій мережі	25
3.3. Тестування функціональності Honeypot-підсистеми	54
ВИСНОВКИ	61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	62
Додатки	64

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

AD – Active Directory
API – Application Programming Interface
IP – Internet Protocol
IPS – Intrusion Prevention System
IDS – Intrusion Detection System
DoS – Denial of Service
HTTP – Hypertext Transfer Protocol
HTTPS – Hypertext Transfer Protocol Secure
SSH – Secure Shell
SQL – Structured Query Language
SaaS – Software as a Service
IP – Internet Protocol
VPN – Virtual Private Network
RDP – Remote Desktop Protocol
NAT – Network Address Translation

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

Актуальність теми: Сучасний світ стає все більш залежним від інформаційних технологій, а розвиток цифрових інфраструктур створює нові виклики у сфері кібербезпеки. Зокрема, корпоративні мережі, що забезпечують функціонування критичних бізнес-процесів, постійно зазнають ризику атак з боку кіберзлочинців. Одним із найпоширеніших методів атак є експлуатація вразливих публічних портів мережі, які можуть стати шлюзами для несанкціонованого доступу до внутрішніх ресурсів. У таких умовах використання технологій Honeypot стає ефективним інструментом для виявлення, моніторингу та стримування зловмисників.

Проектування Honeypot для захисту корпоративних мереж є особливо актуальним у контексті роботи з Active Directory – системою, що часто стає ціллю атак через її значення у централізованому управлінні ресурсами. Саме завдяки таким технологіям можна не лише захистити важливі дані, а й отримати безцінну інформацію про стратегії атакуючих, що дозволить удосконалити засоби захисту.

Мета курсової роботи: Метою цієї курсової роботи є розробка проекту реалізації Honeypot для виявлення і блокування зловмисних IP-адрес, що намагаються атакувати публічні порти корпоративної мережі на базі MS Active Directory. Це передбачає дослідження концепцій, методів та інструментів, які дозволяють створити безпечне середовище, що одночасно захищає мережу та зупиняє несанкціоновані спроби доступу.

Об’єкт дослідження: процес забезпечення безпеки публічних портів у корпоративних мережах.

Предметом дослідження є технології побудови Honeypot-підсистем для виявлення атак на публічні порти.

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

РОЗДІЛ 1. ТЕОРЕТИЧНІ ПРИНЦИПИ ПОБУДОВИ СИСТЕМ HONEYROT

1.1 Honeypot-системи: концепція, функціональність і класифікація

Honeypot «медовий горщик» є спеціалізованим інструментом у сфері кіберзахисту, створеним для імітації реальних цифрових середовищ з метою заманювання зломисників. Цей метод передбачає розгортання фальшивих систем, які зовні нагадують справжні інфраструктурні елементи – з характерними сервісами, протоколами, файлами та вразливостями. Такі «принади» виступають у ролі спокусливих цілей, привабливих для кіберзлочинців, які прагнуть отримати несанкціонований доступ до конфіденційних даних або вивести з ладу інформаційні системи.

Наприклад, honeypot може імітувати систему білінгу, що часто стає мішенню атак з метою крадіжки фінансової інформації, зокрема номерів банківських карток. У момент спроби вторгнення всі дії зломисника фіксуються та аналізуються – це дозволяє виявляти нові способи атак, удосконалювати засоби захисту й зміцнювати вразливі ділянки мережевої інфраструктури.

Типовий honeypot налаштовується так, щоб справляти враження слабо захищеної системи – наприклад, із відкритими портами, уразливими сервісами або слабкими паролями. Потрапивши у таку пастку, зломисник несвідомо передає захисникам безцінну інформацію про свої інструменти, техніки й тактики. Зібрані розвіддані (threat intelligence) дозволяють не лише оперативно реагувати на інциденти, а й формувати довгострокові стратегії кіберзахисту [1].

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		8

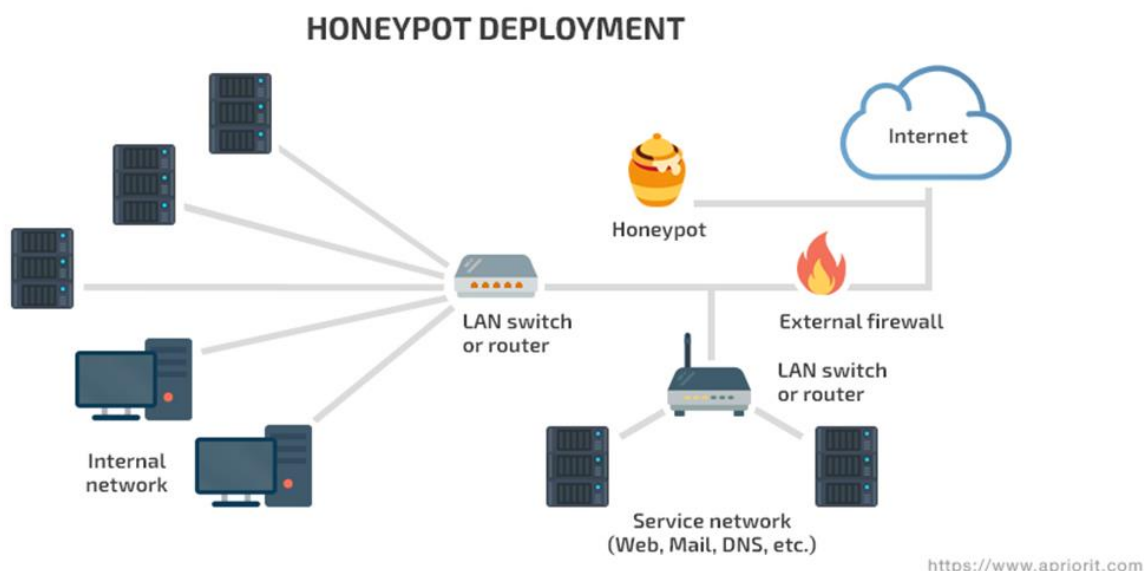


Рисунок 1.1 – Приклад мережі з використанням Honeypot

Honeypot-системи можуть істотно відрізнятися за складністю та функціональними можливостями. Їх класифікують за різними критеріями – від цілей використання до рівня взаємодії.

Із точки зору функціонального призначення можна виділити кілька основних типів:

1. Spam-trap (електронні пастки):

Ці honeypot створюють «приманкові» email-адреси, які не мають жодного функціонального навантаження, але навмисно залишаються відкритими для ботів, що сканують сайти на предмет збирання адрес. Будь-який лист, що надходить на таку скриньку, автоматично класифікується як спам. Це дозволяє виявляти спам-розсилки, формувати чорні списки IP-адрес і вдосконалювати системи фільтрації.

2. Honeypot для баз даних:

Використовуються для виявлення атак, пов'язаних із вразливостями СУБД. Такі пастки імітують структури SQL-баз, дозволяючи зафіксувати спроби SQL-ін'єкцій, ескалації привілеїв або інших способів впливу на серверну логіку.

3. Honeypot для шкідливого ПЗ:

Цей тип моделює середовище, що імітує API, файлову систему або програмну інфраструктуру, аби привабити та активувати зразки шкідливого програмного

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		9

забезпечення. Поведінка ПЗ аналізується задля розробки антивірусного ПЗ, оновлення сигнатур та виправлення вразливостей.

4. Spider honeypot:

Такі системи створюють посилання й сторінки, які недоступні для звичайних користувачів, але привабливі для веб-сканерів (crawler). Завдяки цьому можна виявляти ботів, що індексують контент з рекламною чи шкідливою метою, і блокувати їх [2].

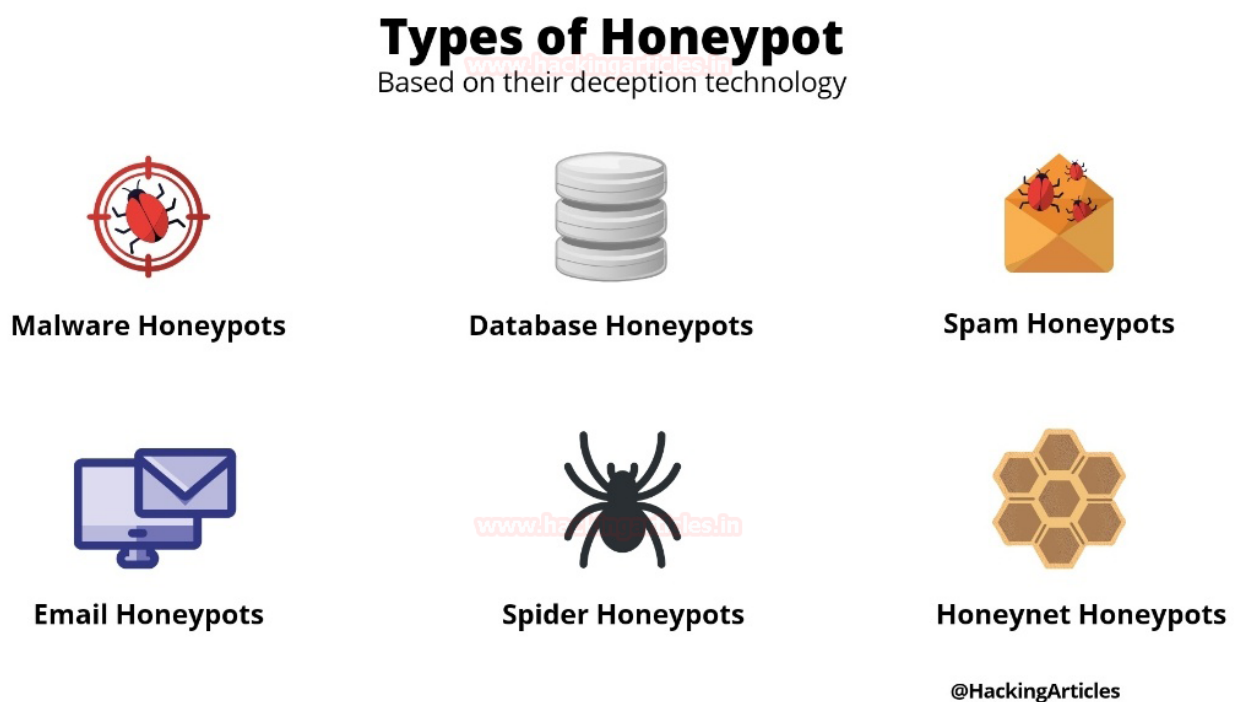


Рисунок 1.2 – Типи Honeypot

З точки зору рівня взаємодії honeypots поділяються на:

1. Низьковзаємодіючі honeypot:

Прості у розгортанні та не потребують значних ресурсів. Вони надають загальну інформацію про типи атак і джерела загроз, імітуючи базові сервіси або протоколи. Основна їхня функція – швидке виявлення активності зловмисника. Проте, обсяг отриманих даних обмежений.

2. Високовзаємодіючі honeypot:

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		10

Це складні інфраструктури, які глибоко імітують реальні системи. Вони дозволяють детально спостерігати за поведінкою атакувальника, збираючи інформацію про способи проникнення, підвищення привілеїв, типи доступу до даних тощо. Такі honeypot можуть включати в себе повноцінні сервери, бази даних і програми, які затримують зловмисника довше, забезпечуючи глибший аналіз. атакуючі.

Однак високовзаємодіючі системи потребують значних обчислювальних ресурсів і людських зусиль для підтримки. Без належного захисту (наприклад, honeypwall – фаєрволу, що обмежує зовнішній трафік із honeypot) вони можуть бути використані як плацдарм для подальших атак або розповсюдження шкідливого ПЗ [3].

1.2 Роль Honeypot у корпоративних мережах

Корпоративна мережа – це складна та багаторівнева інфраструктура, яка охоплює сервери, робочі станції, мережеве обладнання, бази даних, спеціалізоване програмне забезпечення та облікові записи користувачів. Вона забезпечує функціонування бізнес-процесів, збереження конфіденційної інформації, взаємодію працівників і безперешкодний доступ до критично важливих ресурсів. Проте через свою значущість корпоративна мережа є привабливою ціллю для кіберзлочинців. У такому середовищі honeypot виступає як ефективний інструмент виявлення, аналізу та протидії кіберзагрозам [4].

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		11



Рисунок 1.3 – Приклад корпоративної мережі

Honeypot інтегрується у корпоративну мережу для імітації реальних систем – таких як сервери з фінансовими даними або інфраструктурні вузли керування доступом. Вони приваблюють зломисників, спрямовуючи їхню активність у контрольоване середовище, тим самим відволікаючи від реальних ресурсів. Це дозволяє не лише знизити ризик компрометації критичних систем, а й отримати вичерпні відомості про методи та інструменти атаки.

В корпоративній мережі Honeypot виконує кілька важливих функцій:

- виявлення загроз: Honeypot дозволяє оперативно фіксувати спроби сканування портів, brute force-атаки та експлуатацію вразливостей;

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		12

- підвищення загальної стійкості до атак: Зібрані дані дозволяють вдосконалювати налаштування систем IDS/IPS, політики доступу, правила фільтрації трафіку та інші компоненти безпеки;

- моніторинг поведінки зловмисників: Завдяки аналізу дій атакувальників можна виявити їхні цілі, обхідні шляхи та слабкі місця у наявному захисті.

На практиці honeypot у корпоративній мережі застосовується для таких цілей:

- імітація критичних сервісів: Наприклад, баз даних клієнтів або CRM-систем, що дозволяє виявляти реальні спроби вторгнень і зібрати повний контекст атак;

- раннє виявлення інцидентів: Honeypot виявляє первинні ознаки зламу, як-от сканування портів або несанкціоновані спроби авторизації;

- використання honeypot у внутрішньому тренувальному середовищі допомагає підвищити рівень готовності фахівців до реагування на інциденти [5].

1.3. Переваги та недоліки використання Honeypot у захисті мережі

Використання honeypot-систем надає низку стратегічних переваг порівняно з класичними засобами виявлення вторгнень (такими як IDS/IPS), завдяки своїй специфіці – повній відсутності очікуваного легітимного трафіку. Це означає, що будь-яка зафіксована активність, з високою ймовірністю, є спробою сканування, несанкціонованого доступу чи іншої форми атаки.

Однією з ключових переваг honeypot є висока точність виявлення загроз. На відміну від реальних мережевих середовищ, де велика кількість легітимного трафіку ускладнює ідентифікацію шкідливих дій, honeypot функціонує як «чистий аркуш». Зловмисні IP-адреси, а також повторювані патерни атаки (наприклад, сканування з одних і тих самих географічних зон) стають легко впізнаваними. Це дозволяє ефективно ізолювати підозрілу активність та формувати актуальні блокувальні списки.

Ресурсна ефективність – ще один вагомий аргумент на користь honeypot. Такі системи обробляють лише обмежений обсяг трафіку, що дозволяє розгортати їх на

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		13

старому або малопотужному обладнанні. Багато рішень із відкритим кодом доступні в онлайн-репозиторіях (наприклад, Cowrie, Dionaea, T-Pot), що мінімізує потребу в розробці власного ПЗ та знижує витрати на впровадження.

Важливою перевагою honeypot є низький рівень хибнопозитивних спрацьовувань, що суттєво відрізняє їх від традиційних IDS. У той час як IDS можуть генерувати сотні або тисячі сповіщень щодня – багато з яких виявляються помилковими, honeypot фіксує виключно небажану активність. Це дає змогу командам безпеки сфокусуватися на реальних загрозах, зменшуючи аналітичне навантаження та оптимізуючи витрати на реагування.

Ще одна істотна перевага полягає у збірці розвідданих про загрози (cyber threat intelligence). Honeypot здатен фіксувати:

- використовувані вектори атак;
- приклади експлойтів;
- поведінкові шаблони зловмисників;
- типи шкідливого ПЗ;
- спроби спаму та фішингові кампанії (у випадку email honeypots).

Такі дані можуть стати основою для вдосконалення захисної архітектури, оновлення сигнатур у системах виявлення та загального посилення кібергігієни організації.

Окрім технічного застосування, honeypot виконує освітню функцію. У контрольованому середовищі він слугує тренажером для технічного персоналу служби безпеки – даючи змогу на практиці спостерігати поведінку атакувальника, аналізувати наслідки використання різних експлойтів та навчитись оперативно реагувати на вторгнення без ризику для реальної інфраструктури.

Варто також відзначити можливість виявлення внутрішніх загроз. Попри акценту більшості організацій на захисті периметра, саме внутрішні порушники (недобросовісні співробітники, порушення політик доступу, зламані акаунти) становлять особливо небезпечний клас загроз. Розміщення honeypot у

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		14

внутрішньому сегменті мережі дозволяє виявити нетипову активність, що свідчить про компрометацію, а також вказує на слабкі місця у системі контролю доступу.

Додатковий ефект – пасивне стримування атак. Зловмисник витрачає час і ресурси на взаємодію з honeypot, замість того щоб атакувати реальні системи, що дає змогу виграти час для реагування та обмежити збитки. [5].

Водночас honeypot має і низку суттєвих обмежень. Насамперед – він фіксує лише ті загрози, які були прямо на нього спрямовані. Відсутність подій у логах honeypot не гарантує безпеку всієї мережі, тому така система не може бути єдиним елементом оборони.

Ще один критичний аспект – необхідність реалістичної імітації. Успішний обман зловмисника передбачає наявність достовірних повідомлень про вхід, псевдоданих, логотипів і навіть поведінки системи. Якщо ж хакер ідентифікує систему як honeypot, він або припиняє атаку, або направляє зусилля на реальні цілі, ігноруючи пастку. Такий розвиток подій знижує ефективність honeypot як інструмента виявлення.

Більш витончені атаки можуть мати деструктивніші наслідки: зловмисник, виявивши пастку, може свідомо запускати фіктивні атаки з метою відволікання уваги аналітиків, або інжектувати неправдиві дані, ускладнюючи подальший аналіз і виявлення справжньої загрози.

Існує також технічний ризик, що досвідчений атакувальник зможе використати honeypot як плацдарм для подальшого проникнення в інші компоненти мережі. Без належної ізоляції та обмежень доступу honeypot може сам стати точкою компрометації. Для запобігання подібним інцидентам використовують спеціальні механізми контролю, зокрема honeuwall – мережевий фільтр, що блокує вихідний трафік від honeypot, не допускаючи перенаправлення атак на внутрішні системи.

Також варто пам'ятати, що honeypot не є засобом активного захисту. Він не може протистояти атакам або перешкоджати їх здійсненню – лише фіксує вже здійснену активність. У зв'язку з цим, honeypot повинен доповнюватися базовими

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		15

заходами безпеки: міжмережевими екранами, системами виявлення вторгнень (IDS/IPS), антивірусами, сегментацією мережі та шифруванням.

У підсумку, попри наявність певних ризиків і обмежень, використання honeypot залишається виправданим. За умови правильного впровадження та інтеграції в багаторівневу стратегію захисту, honeypot може забезпечити цінну аналітику в реальному часі, дозволити виявити слабкі місця в захисті, а також успішно доповнити традиційні засоби кібербезпеки [6], [7], [8].

1.4 Сервіси MS Active Directory та їхня взаємодія через публічні порти

Active Directory (AD) – це централізована служба каталогів, яка забезпечує автентифікацію, авторизацію та керування доступом до ресурсів у корпоративній мережі. У межах корпоративної інфраструктури AD виконує роль критичного компонента, який об'єднує численні сервіси: контролери домену, DNS-сервери, файлові служби, політики безпеки, механізми групової політики та інші засоби централізованого адміністрування.

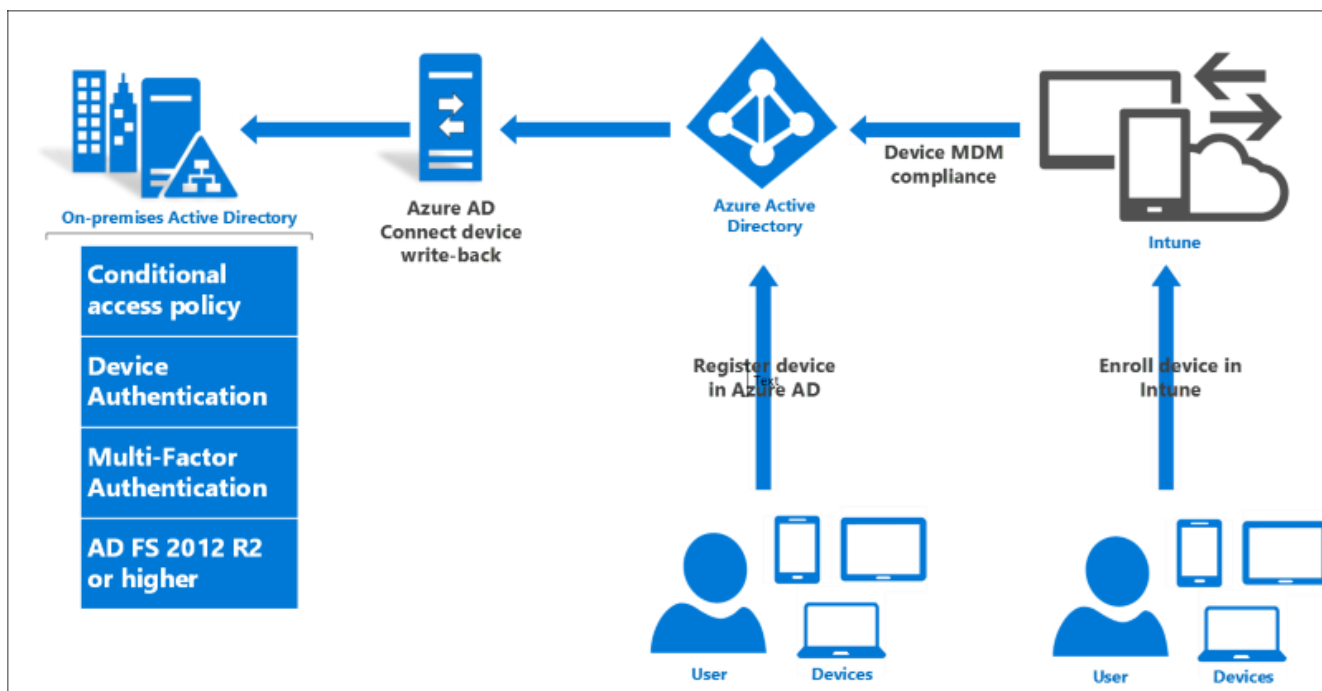


Рисунок 1.4 – Active Directory в корпоративній мережі

Для забезпечення своєї функціональності служби Active Directory активно взаємодіють через низку мережевих портів, деякі з яких можуть бути відкритими

для зовнішнього доступу – наприклад, у випадках, коли організація використовує гібридні або розподілені моделі доступу, або надає адміністративний доступ до ресурсів іззовні. Така взаємодія створює потенційні точки входу для атак.

До найбільш критичних сервісів і портів, пов'язаних із функціонуванням Active Directory, належать:

- TCP/UDP 389 – LDAP: використовується для запитів до служби каталогів, зокрема для пошуку, аутентифікації та синхронізації;
- TCP 636 – LDAPS: захищена версія LDAP із TLS/SSL;
- TCP/UDP 88 – Kerberos: основний протокол аутентифікації;
- TCP 445 – SMB: доступ до файлів, а також передавання політик групи;
- TCP 135, 139, 1025-5000, 49152-65535 – DCOM/RPC: для взаємодії з віддаленими службами, інвентаризації об'єктів, реплікації даних між контролерами домену.

Наявність відкритих портів, пов'язаних з AD, значно підвищує ризики кіберзагроз. Зловмисники можуть використовувати ці порти для:

- виявлення служб через порт-сканування;
- атаки на LDAP з метою обходу аутентифікації або отримання даних із каталогу;
- брутфорс-атак через Kerberos (включаючи Kerberoasting);
- експлуатації RPC/DCOM для отримання доступу до служб адміністрування;
- зловживання SMB для lateral movement у мережі (наприклад, через EternalBlue).

Критичним чинником у контексті кіберзахисту є те, що ці служби часто працюють у непрозорих для зовнішнього спостереження каналах, проте через відкриті порти можуть бути використані як вектори первинного доступу або горизонтального переміщення в інфраструктурі (lateral movement). Саме тому контроль, моніторинг та симуляція поведінки зловмисника (через honeypot-підсистеми) в зоні відкритих портів має особливе значення.

		Цевчук В.С.			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		17

З огляду на це, ефективне проектування захисної архітектури повинно включати моніторинг усіх взаємодій, які відбуваються через зазначені порти, а також застосування honeypot-рішень, здатних симулювати AD-сумісні служби з метою виявлення небажаної активності на ранніх етапах вторгнення [9].

1.5 Захист публічних портів у корпоративних мережах на основі Honeypot-технологій

У корпоративних мережах публічні порти відіграють надзвичайно важливу роль, оскільки забезпечують зв'язок між внутрішніми інформаційними системами організації та зовнішнім середовищем. Саме через них клієнти, партнери, віддалені працівники та зовнішні сервіси отримують доступ до корпоративних ресурсів. Такі порти дозволяють реалізовувати цілий спектр функцій: від надання дистанційного доступу через VPN, RDP або SSH, до обслуговування вебзапитів через HTTP і HTTPS, підключення до баз даних через спеціалізовані порти, наприклад порт 1433 для Microsoft SQL Server, або обміну електронною поштою через SMTP та IMAP.

Втім, кожен відкритий порт – це потенційний вектор атаки. Зловмисники регулярно сканують мережі в пошуках відкритих портів, використовуючи автоматизовані інструменти для виявлення вразливих сервісів. Як правило, на початковому етапі атаки здійснюється сканування портів з метою ідентифікації активних служб. Після цього можуть застосовуватися методи грубої сили для підбору облікових даних, або ж використовуються відомі вразливості у застарілому програмному забезпеченні, що працює на цих портах. У деяких випадках зловмисники можуть ініціювати розподілені атаки типу DDoS, спрямовані на виведення сервісів із ладу шляхом перевантаження мережевого інтерфейсу.

У цьому контексті Honeypot виступає як ефективний інструмент кіберзахисту, що дозволяє не лише виявляти спроби вторгнення, але й зменшувати ризик компрометації основних систем. Суть його роботи полягає у створенні спеціально сконструйованих фальшивих систем або сервісів, які свідомо залишаються доступними ззовні, привертаючи увагу зловмисників. Як тільки відбувається спроба доступу до такого "публічного" ресурсу, всі дії атакуючого

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		18

ретельно фіксуються та аналізуються. Це дозволяє імітувати наявність цікавих для хакерів відкритих портів, таких як SSH або RDP, що стимулює ініціацію атаки саме на ці об'єкти.

Honeypot також дозволяє збирати цінну розвідувальну інформацію: IP-адреси джерел атак, типові техніки та тактики проникнення, використані експлойти. Така аналітика стає основою для подальшого вдосконалення захисної інфраструктури організації – шляхом оновлення правил брандмауера, налаштування систем виявлення та запобігання вторгненням, а також коригування політик безпеки. Окрім цього, використання Honeypot дає змогу ефективно відволікати увагу атакуючих від реальних ресурсів, змушуючи їх витратити зусилля на взаємодію з ізольованими й безпечними середовищами. У поєднанні з сучасними системами моніторингу, Honeypot може бути частиною активного захисту, в якому спроба атаки викликає автоматичну реакцію – наприклад, блокування IP-адреси, яка помічена у ворожій активності [10], [11].

Висновки до першого розділу

Було досліджено теоретичні засади використання Honeypot-систем у кібербезпеці. Визначено їхнє призначення як інструмента дезорієнтації зловмисників, збору розвідувальних даних і підвищення захищеності мережевої інфраструктури. Розкрито класифікацію Honeypot за рівнем інтерактивності та за типами загроз, обґрунтовано їхні переваги і проаналізовано недоліки – зокрема можливість викриття досвідченими хакерами, обмежене охоплення загроз і ризик вторинного використання як плацдарму для подальших атак.

Також було розглянуто сервіси Active Directory та їхню взаємодію через публічні порти. Показано, як симульовані вузли здатні імпіювати реальні системи, фіксувати несанкціоновані спроби доступу та надавати дані для вдосконалення політик безпеки. Отримані теоретичні положення стали основою для побудови моделі корпоративної мережі з Honeypot, орієнтованої на захист публічних портів у середовищі MS Active Directory.

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ.2. ПРОЕКТУВАННЯ HONEYPOT-ПІДСИСТЕМИ ДЛЯ ЗАХИСТУ ПУБЛІЧНИХ ПОРТІВ КОРПОРАТИВНОЇ МЕРЕЖІ

2.1 Принципи побудови мережевої архітектури з використанням Honeypot

У межах сучасної корпоративної інфраструктури реалізація ефективного захисту мережевих сервісів потребує гнучкої архітектури, здатної не лише запобігати атакам, а й виявляти спроби несанкціонованого доступу до критичних ресурсів. Одним із ефективних методів є впровадження Honeypot-підсистеми, яка додається до вже існуючої мережевої структури з публічними портами, службами авторизації, внутрішнім сегментом та зоною DMZ.

Проектування такої архітектури передбачає наявність:

- сегменту для внутрішньої інфраструктури (зокрема, серверів Active Directory, користувацьких машин, службових сервісів);
- периферійного вузла або шлюзу з налаштованими фаєрволами, NAT або port forwarding;
- зовнішнього середовища з підозрілим або ворожим трафіком (імітація атакуючого середовища);
- honeypot-серверів, які логічно або фізично ізольовані від продуктивного середовища, але здатні приймати вхідні запити на певні порти.

Honeypot-підсистема інтегрується на периметр мережі або в DMZ, де вона отримує запити, що надходять на відкриті сервіси, такі як SSH, HTTP, Telnet тощо. За рахунок цього формується додатковий рівень моніторингу та аналізу загроз, що не впливає на реальні вузли мережі. Усі елементи взаємодіють через чітко визначені маршрути, а передача трафіку до Honeypot здійснюється з використанням механізмів портфорвардингу через граничний шлюз.

Таким чином, Honeypot-підсистема є логічно відокремленим, але тісно інтегрованим компонентом архітектури, який не лише фіксує шкідливу активність, а й дозволяє зреагувати на неї через взаємодію з фаєрволом, системою логування чи іншими компонентами безпеки.

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

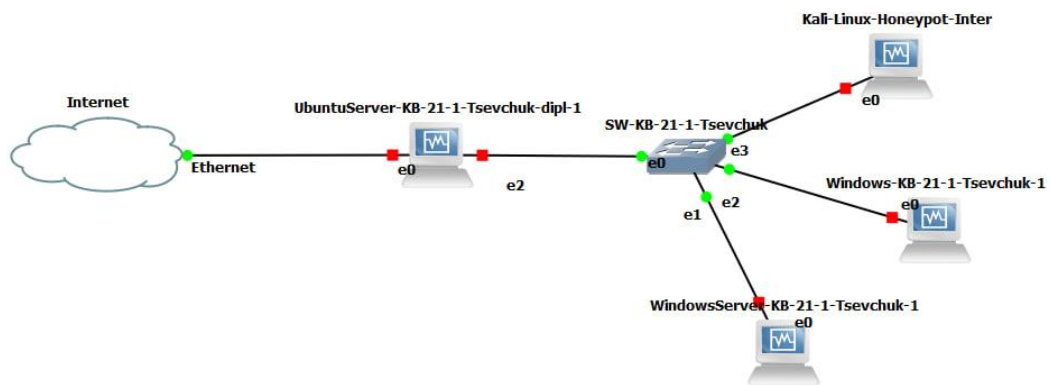


Рисунок 2.1 – Схематичне зображення Honeypot-підсистеми

2.2 Проектування Honeypot-підсистеми для корпоративної мережної інфраструктури

Функціональна структура корпоративної мережі в межах проекту спроектована з використанням віртуального середовища VirtualBox та візуалізована у GNS3. Обрана архітектура відображає ключові елементи корпоративної інфраструктури з акцентом на інтеграцію Honeypot-підсистеми для виявлення загроз на рівні публічних сервісів.

У моделі передбачено сервер служб каталогів Active Directory, що відтворює типове корпоративне середовище й слугує умовною «ціллю» для потенційних атак. На окремому вузлі розміщено Honeypot-сервер, який емулює SSH/Telnet- і HTTP/HTTPS-сервіси завдяки інструментам Cowrie та honeyhttpd. Міжмережевий шлюз із двома логічними інтерфейсами – зовнішнім і внутрішнім – виконує роль фаєрвола, забезпечує порт-форвардинг на honeypot-сервіси та застосовує правила маршрутизації й блокування небажаного трафіку згідно з виявленими індикаторами атаки. Зовнішні вузли, розташовані у відокремленому сегменті, імітують дії потенційного злоумисника: через шлюз вони можуть ініціювати запити на відкриті порти, що скеровуються до honeypot.

		Цевчук В.С.			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		21

Зібрана honeypot-системою інформація використовується для оперативного реагування на загрози, зокрема для динамічного оновлення фільтрів безпеки. Логіка рішення забезпечує чітке розділення ролей: honeypot фіксує активність атакувальників, шлюз контролює взаємодію, а Active Directory створює реалістичний контекст корпоративної мережі.

2.3 Проєктування Honeypot для сервісів віддаленого доступу

Сервіси віддаленого доступу, зокрема SSH і Telnet, є одними з найбільш привабливих цілей для зловмисників. Ці протоколи часто піддаються атакам, спрямованим на підбір облікових даних або використання вразливостей у службах аутентифікації. У контексті проєкту їх було обрано як основні точки спостереження в Honeypot-архітектурі [12].

Метою моделювання є створення фіктивного середовища, що привертатиме увагу потенційних атакувальників і дозволить фіксувати характерні ознаки вторгнень. Емуляція віддаленого доступу забезпечує виявлення спроб підключення, аналіз введених логінів, паролів, команд і сценаріїв поведінки.

Особливий інтерес становлять атаки типу brute force, виявлення інструментів автоматизації зламу, а також дослідження команд, які виконуються після отримання доступу. Це дозволяє зрозуміти, до яких дій готувався зловмисник: сканування внутрішньої мережі, розгортання бекдорів, крадіжка файлів тощо.

Проєктування Honeypot для SSH/Telnet дозволяє імітувати типове середовище Linux-серверів, що є поширеною ціллю атак, і збирати поведінкові ознаки порушень без ризику для продуктивної інфраструктури.

2.4 Проєктування Honeypot для веб-сервісів

Веб-сервіси – одна з основних точок взаємодії в корпоративному середовищі, але водночас і вектор для численних атак. Протоколи HTTP та HTTPS є цільовими для експлойтів, пов'язаних із SQL-ін'єкціями, XSS, directory traversal, фішингом тощо. Моделювання вебсервера як Honeypot дозволяє спостерігати за подібними спробами в безпечному середовищі [13], [14].

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		22

Емуляція HTTP/HTTPS у рамках цього проєкту передбачає створення реалістичних вебсторінок і точок взаємодії, на які може натрапити атакувальний скрипт або вручну спрямована атака. Це можуть бути підроблені панелі адміністрування, форми логіну або посилення, які провокують небажану активність.

Завдяки такому підходу можна збирати журнали запитів, аналізувати їх структуру, виявляти поширені User-Agent, ідентифікувати специфічні сканери або шкідливі шаблони звернень. Такі дані дозволяють глибше зрозуміти характер сучасних web-орієнтованих атак і вдосконалити політики захисту на прикладному рівні.

Висновки до другого розділу

Було здійснено проєктування віртуалізованої мережевої інфраструктури з інтеграцією Honeypot-підсистеми, призначеної для виявлення та аналізу несанкціонованої активності через публічні сервіси. Побудована архітектура охоплює ключові компоненти корпоративного середовища: контролер домену Active Directory, шлюз з фаєрволом і механізмами перенаправлення трафіку, робочу станцію та окремі Honeypot-сервери, орієнтовані на SSH/Telnet- і HTTP/HTTPS-протоколи.

У межах проєкту реалізовано логічне розділення мережі на зовнішній і внутрішній сегменти, що дозволяє моделювати дії зловмисників у контрольованому середовищі. Окрема увага приділена емульованим сервісам віддаленого доступу й вебдодатків, які найчастіше стають об'єктами атак.

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		23

РОЗДІЛ 3. ВИБІР ПРОГРАМНОГО РІШЕННЯ ТА НАЛАГОДЖЕННЯ HONEYPOT-ПІДСИСТЕМИ В КОРПОРАТИВНІЙ МЕРЕЖІ

3.1 Вибір програмного забезпечення та інструментів для реалізації

Honeypot-системи

Для реалізації Honeypot-системи було підібрано комплекс програмних засобів, що дали змогу змодельовати вразливе середовище, інтегрувати його у корпоративну мережу та організувати моніторинг дій потенційно небезпечних суб'єктів.

Ключовим інструментом обрано Cowrie – один із найпоширеніших емуляторів SSH- та Telnet-серверів, здатний імітувати систему з типово відкритими вразливостями, що слугує привабливою ціллю для зломисників. Середовище для розгортання Honeypot побудовано за допомогою Oracle VirtualBox, що дозволило змодельовати гнучку топологію віртуальної мережі. Використання NAT Network забезпечило ізоляцію Honeypot-системи від основної інфраструктури, водночас дозволяючи емулювати реальні умови роботи корпоративної мережі.

Доповненням до симуляції став HoneyHTTPD – Honeypot, орієнтований на протокол HTTP (порт 80) та HTTPS (порт 443). Він емулює роботу реальних веб-додатків, надаючи зломисникам фальшиві сторінки, форми входу та інші елементи, типові для уразливих систем. Це дає змогу отримати ще більше аналітичних даних та краще зрозуміти методи, які застосовують атакувальники у мережі. Cowrie та HoneyHTTPD функціонують як окремі логічні компоненти Honeypot-системи, що обслуговують різні протоколи.

У схемі додатково впроваджено проміжний сервер на базі Ubuntu, який виконує роль мережевого шлюза між зовнішнім і внутрішнім сегментами. Цей вузол оснащений двома мережевими інтерфейсами – публічним та приватним – та виконує функції фаєрвола за допомогою iptables. На ньому реалізовано перенаправлення портів (port forwarding) до Honeypot-сервера, що уможливлює доступ зовнішніх суб'єктів лише до обмеженого набору сервісів (SSH та Telnet), ізоляцію від критичних компонентів внутрішньої мережі та зниження ризиків компрометації.

		Цевчук В.С.			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		24

Окрім цього, на Ubuntu-сервері впроваджено механізм динамічного блокування IP-адрес, з яких фіксуються підозрілі активності. Такий підхід дозволяє автоматизовано формувати чорні списки джерел атак та забезпечувати додатковий рівень захисту на етапі раннього реагування. Логування взаємодії зломисників із Honeypot-сервісами забезпечує базу для подальшого аналізу типових сценаріїв атак. Завдяки цьому Ubuntu-шлюз виконує роль не лише фільтраційного вузла, а й активного елементу системи протидії загрозам.

3.2 Налаштування Honeypot-системи в спроектованій мережі

Початковим етапом конфігурування мережевої інфраструктури стало створення NAT-сегмента з призначенням йому статичного IPv4-префіксу – 192.168.19.128/26, що дозволило чітко визначити діапазон доступних IP-адрес у межах приватної підмережі. Для забезпечення повного контролю над розподілом адрес і запобігання можливим конфліктам між вузлами було відключено службу DHCP, передбачивши ручне налаштування параметрів мережі для кожного пристрою.

На рисунку 3.1 представлено процес створення NAT-мережі. Взаємодію між вузлами та їх підключення до віртуального середовища проілюстровано на рисунках 3.2, 3.3, 3.4, 3.5 та 3.6.



Рисунок 3.1 – Створення NAT-мережі

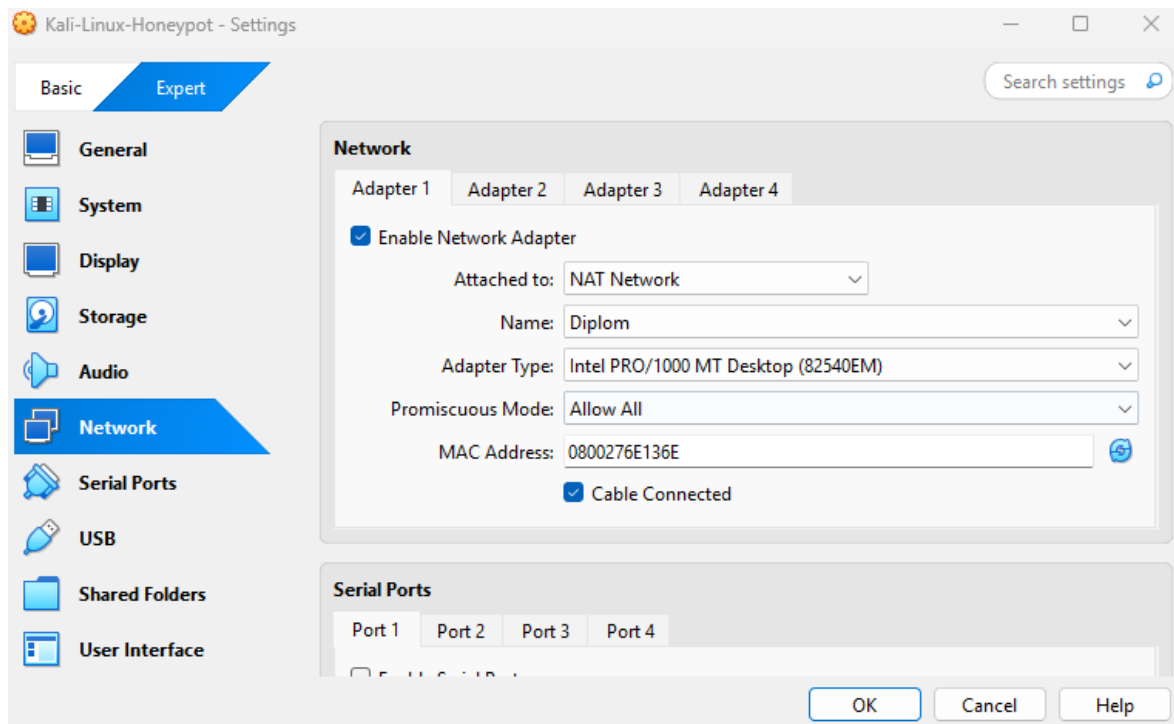


Рисунок 3.2 – Додавання HoneyPot до NAT-мережі

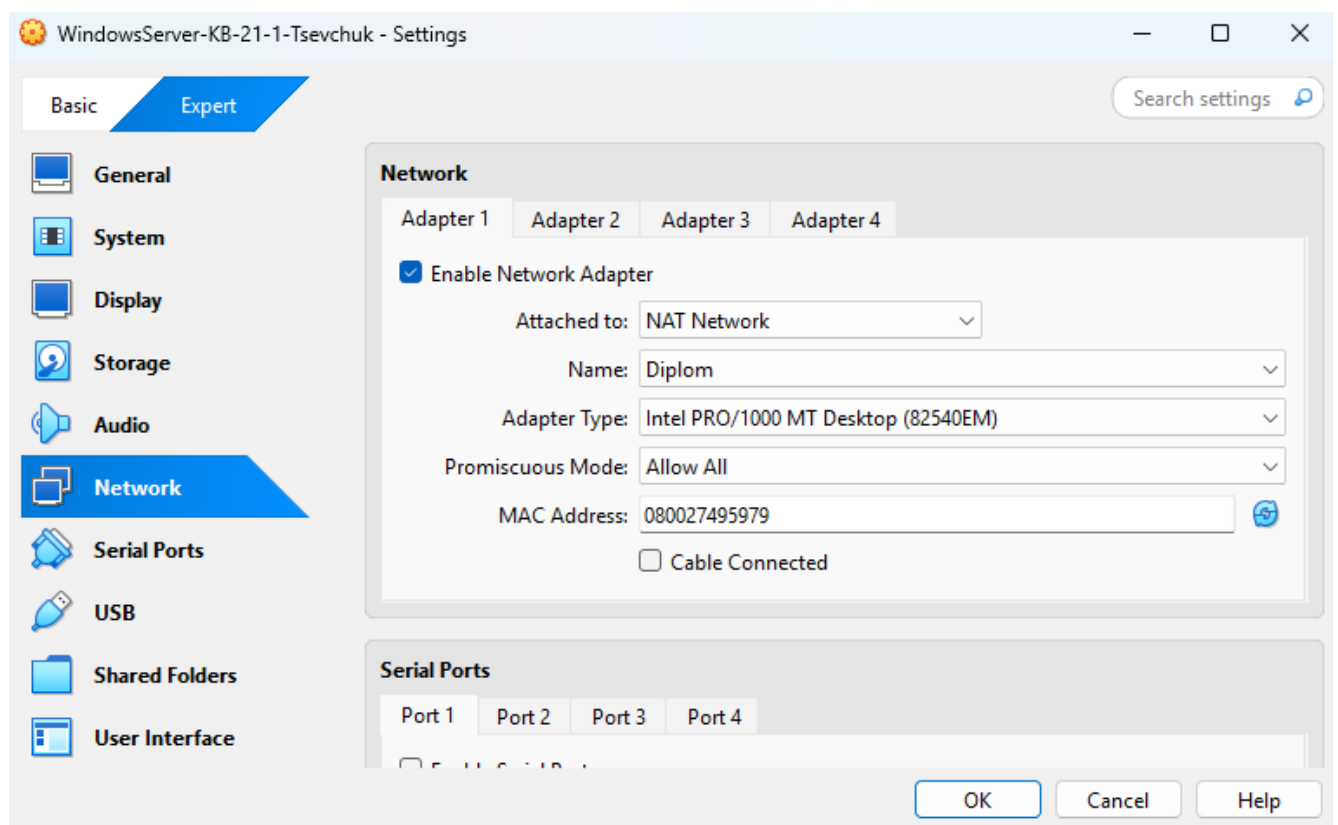


Рисунок 3.3 – Додавання Windows server до NAT-мережі

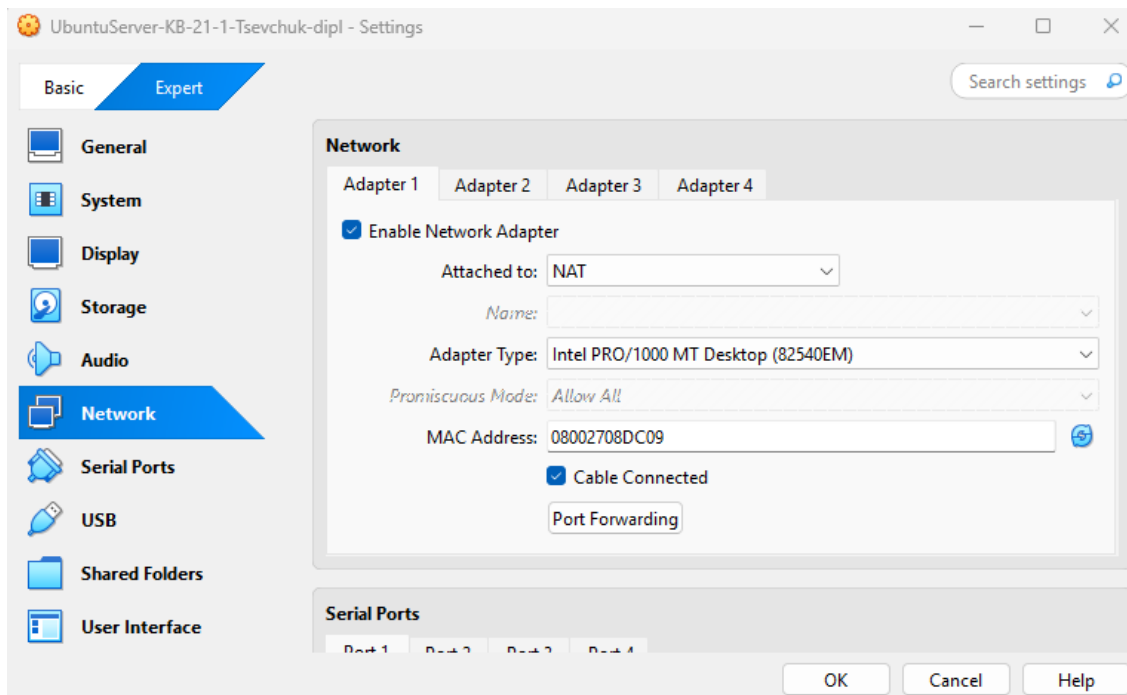


Рисунок 3.4 – Додавання Ubuntu server до NAT-мережі

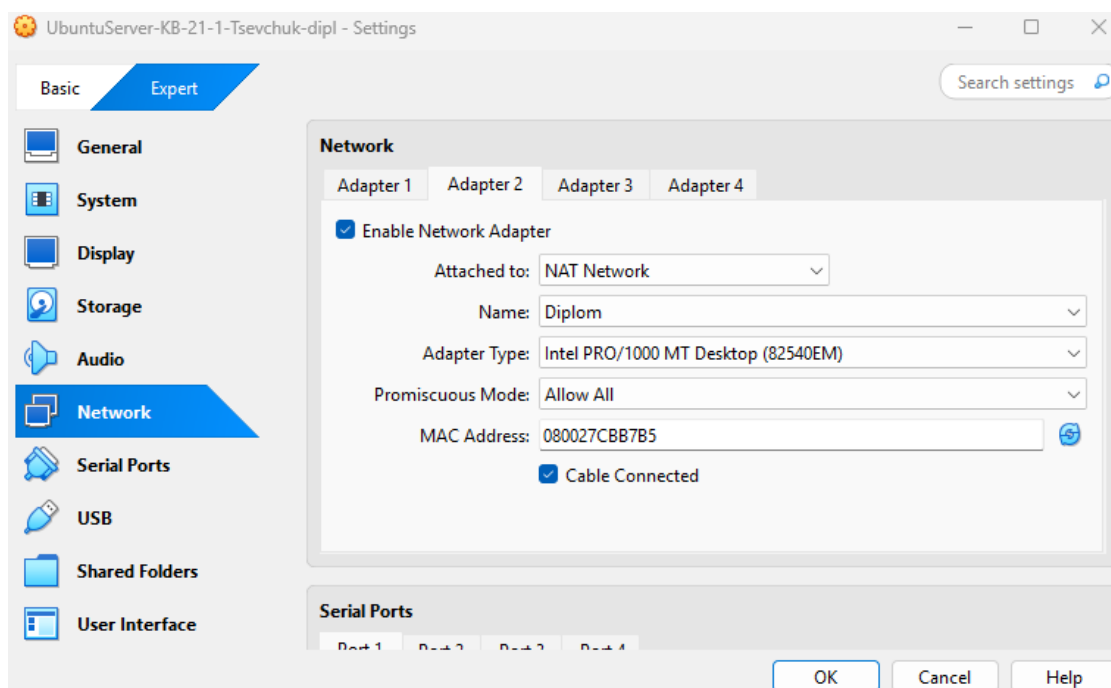


Рисунок 3.5 – Додавання Ubuntu server до NAT-мережі

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		27

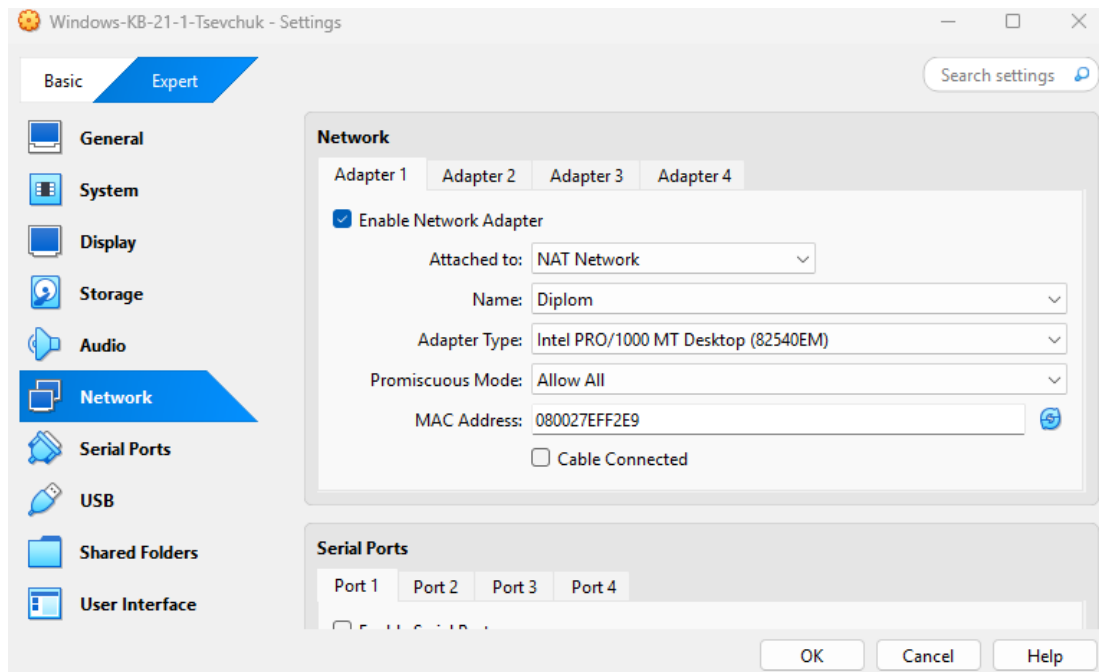


Рисунок 3.6 – Додавання робочої станції до NAT-мережі

Наступним етапом налаштування мережевої інфраструктури стало впровадження механізму Port Forwarding, що забезпечує перенаправлення вхідних з'єднань із хост-машини або зовнішньої мережі на визначені порти honeypot-сервера. Це дозволяє імітувати роботу реального сервера, що приймає зовнішні атаки для подальшого їхнього аналізу. Схематичне відображення налаштувань Port Forwarding наведено на рисунку 3.7 та 3.8 [15].

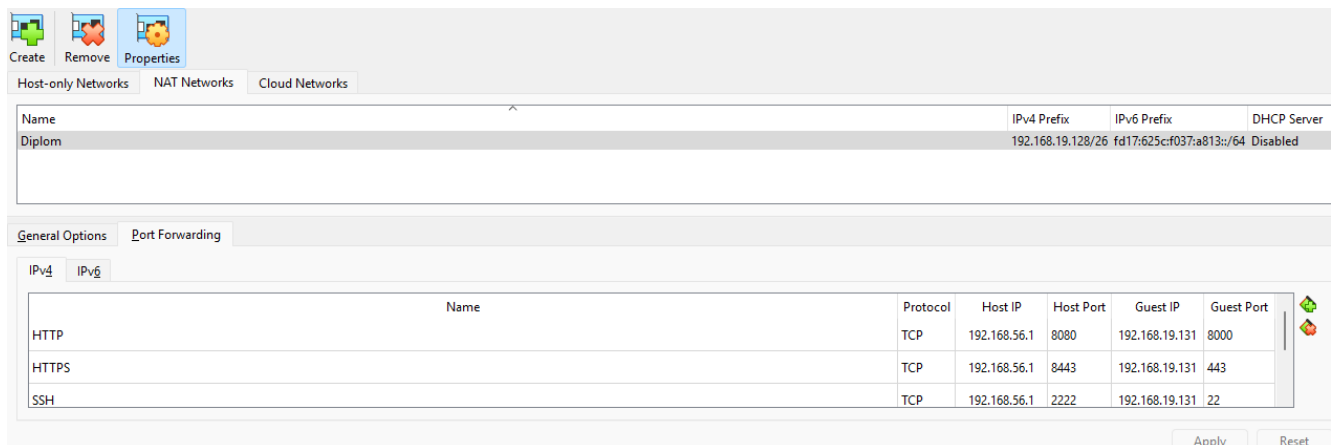


Рисунок 3.7 – Налаштування Port Forwarding

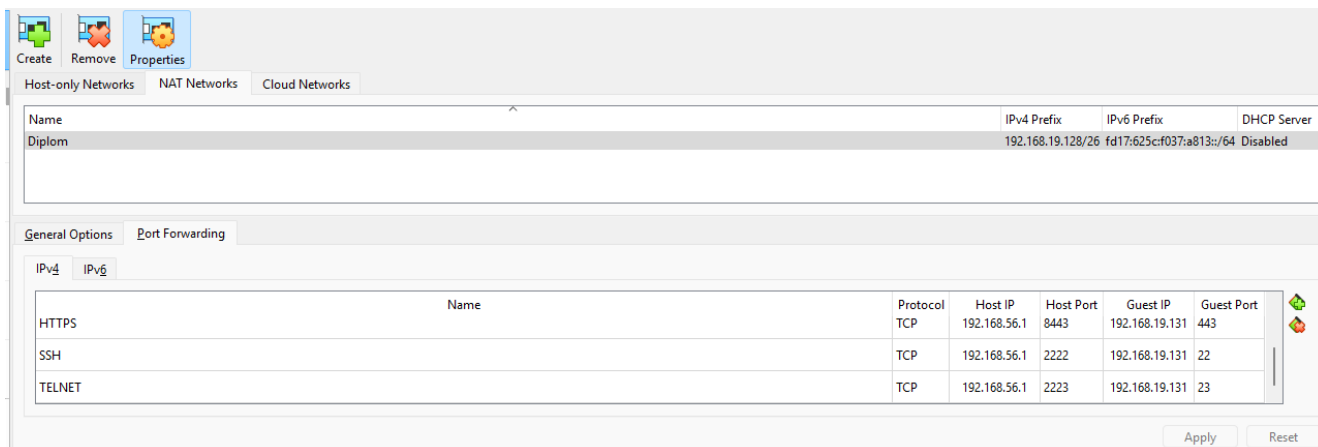


Рисунок 3.8 – Налаштування Port Forwarding

Далі було здійснено налаштування статичної маршрутизації для забезпечення коректної взаємодії пристроїв у межах створеної мережі. Зокрема, серверу Windows Server призначено IP-адресу 192.168.19.130, honeypot-системі – 192.168.19.131, а робочій станції – 192.168.19.132. Додатково до мережі було підключено сервер Ubuntu, конфігурований з двома мережевими інтерфейсами: перший використовує DHCP для отримання IP-адреси з зовнішньої мережі, другий – має статичну IP-адресу 192.168.19.135, що забезпечує його інтеграцію у внутрішній сегмент. Призначення статичних IP-адрес для відповідних вузлів відображено на рисунках 3.9, 3.10, 3.11 та 3.12.

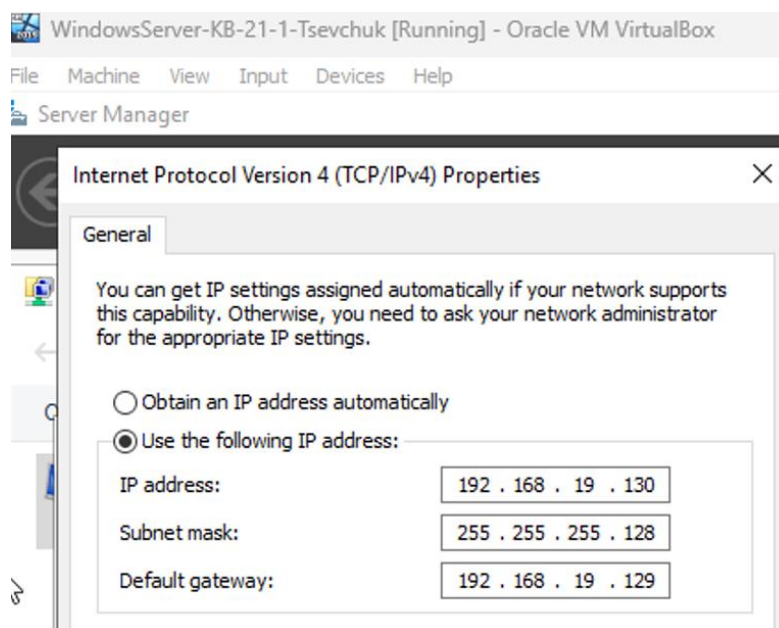


Рисунок 3.9 – Налаштування статичної маршрутизації для Windows Server

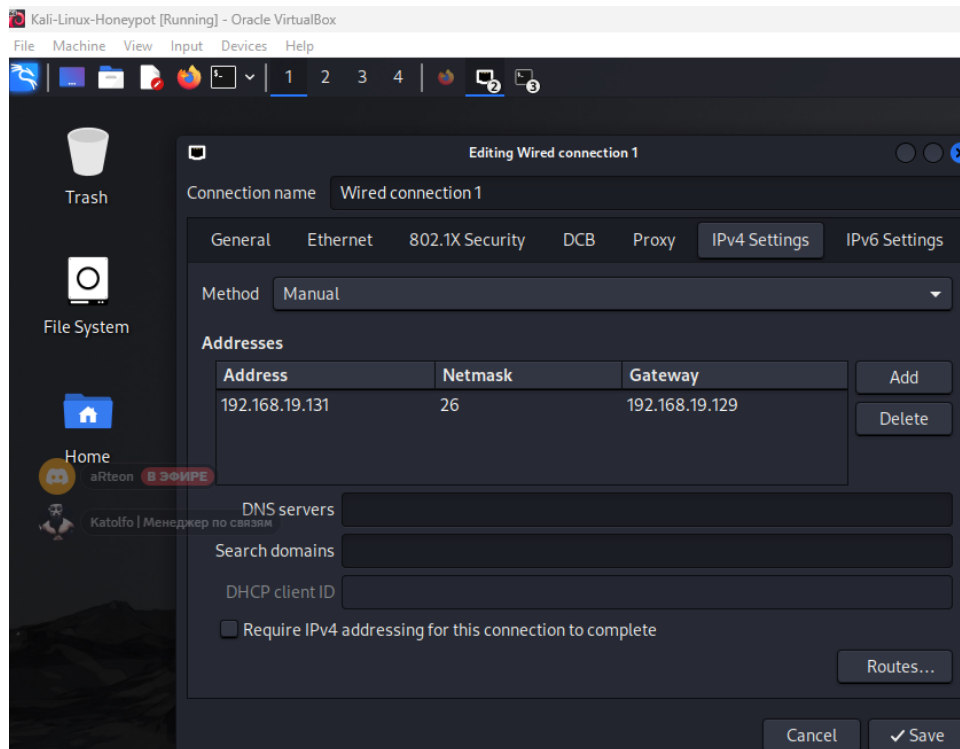


Рисунок 3.10 – Налаштування статичної маршрутизації для Honeypot

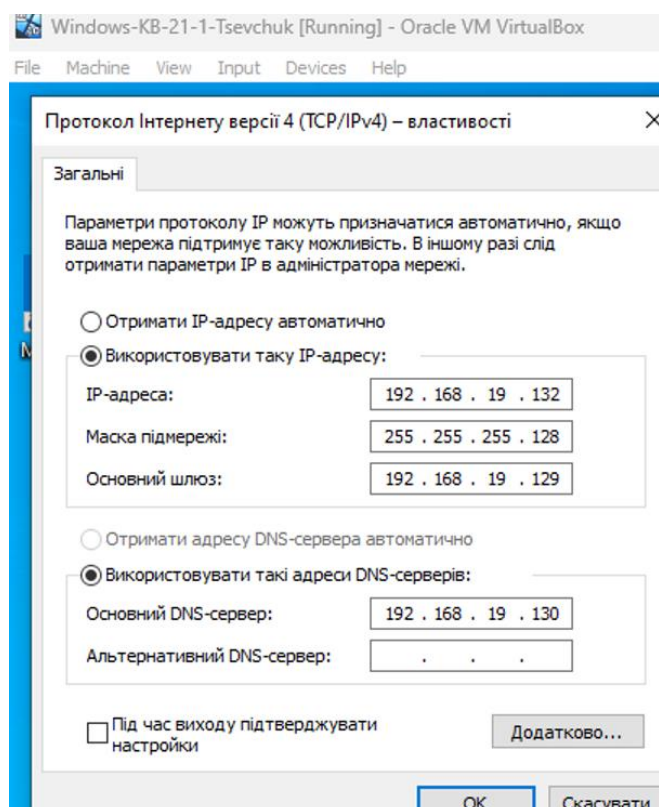


Рисунок 3.11 – Налаштування статичної маршрутизації для робочої станції

```

GNU nano 6.2                                00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      addresses: [192.168.19.135/26]
  version: 2
  
```

Рисунок 3.12 – Налаштування маршрутизації для Ubuntu Server

Наступним етапом проєкту є налаштування Honeypot-системи та встановлення емулятора Cowrie. Першочергово виконано оновлення системи та встановлення необхідних пакетів, що забезпечують коректну роботу Cowrie. Цей етап є критично важливим для створення стабільного середовища. Результат виконаних дій наведено на рисунку 3.13.

```

Kali-Linux-Honeypot [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
PS> kali@kali: /home/kali

(kali@kali)-[/home/kali]
PS> sudo apt-get install git python3-venv libssl-dev libffi-dev build-essential
libpython3-dev python3-minimal authbind virtualenv python3.10-venv
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package virtualenv is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source
However the following packages replace it:
  python3-virtualenv

E: Unable to locate package libssl-dev
E: Unable to locate package authbind
E: Package 'virtualenv' has no installation candidate
E: Unable to locate package python3.10-venv
E: Couldn't find any package by glob 'python3.10-venv'
E: Couldn't find any package by regex 'python3.10-venv'

(kali@kali)-[/home/kali]
PS> sudo apt-get install git python3-venv libssl-dev libffi-dev build-essential
libpython3-dev python3-minimal authbind virtualenv python3.10-venv
  
```

Рисунок 3.13 – Завантаження відповідних пакетів

З метою підвищення безпеки та ізоляції процесів було створено окремий обліковий запис користувача, від імені якого працює honeypot. Приклад створення облікового запису наведено на рисунку 3.14.

```

Kali-Linux-Honeypot [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

cowrie@kali: ~

File Actions Edit View Help

(kali@kali)-[/home/kali]
PS> sudo adduser --disabled-password cowrie
info: Adding user `cowrie' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `cowrie' (1001) ...
info: Adding new user `cowrie' (1001) with group `cowrie (1001)' ...
info: Creating home directory `/home/cowrie' ...
info: Copying files from `/etc/skel' ...
info: Copying files from `/etc/skel' ...
Changing the user information for cowrie
Enter the new value, or press ENTER for the default
    Full Name []: Cowrie
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
info: Adding new user `cowrie' to supplemental / extra groups `users' ...
info: Adding user `cowrie' to group `users' ...

(kali@kali)-[/home/kali]
PS> sudo si - cowrie
sudo: si: command not found

(kali@kali)-[/home/kali]
PS> sudo su - cowrie
(cowrie@kali)-[~]
$
  
```

Рисунок 3.14 – Створення облікового запису

Далі виконано клонування офіційного репозиторію Cowrie з GitHub. Відповідні дії продемонстровано на рисунку 3.15.

The screenshot shows a terminal window titled "cowrie@kali: ~/cowrie". The user is in the "kali" shell at the "/home/kali" directory. They run the command "sudo su - cowrie" to switch to the "cowrie" user. Then, they run "git clone http://github.com/cowrie/cowrie". The terminal output shows the cloning process, including a warning about redirecting to https, enumerating objects, counting objects, compressing objects, and receiving objects. After the clone is complete, the user runs "cd cowrie/" and then "ls". The "ls" command lists the contents of the "cowrie" directory, which includes files like "bin", "CHANGELOG.rst", "CONTRIBUTING.rst", "docker", "docs", "etc", "honeyfs", "INSTALL.rst", "LICENSE.rst", "Makefile", "MANIFEST.in", "pyproject.toml", "README.rst", "requirements-dev.txt", "requirements-output.txt", "requirements-pool.txt", "requirements.txt", "setup.cfg", "setup.py", "src", "tox.ini", and "var".

```
cowrie@kali: ~/cowrie
File Actions Edit View Help

(kali@kali)-[/home/kali]
PS> sudo su - cowrie
(cowrie@kali)-[~]
$ git clone http://github.com/cowrie/cowrie
Cloning into 'cowrie' ...
warning: redirecting to https://github.com/cowrie/cowrie/
remote: Enumerating objects: 18575, done.
remote: Counting objects: 100% (2011/2011), done.
remote: Compressing objects: 100% (336/336), done.
remote: Total 18575 (delta 1881), reused 1675 (delta 1675), pack-reused 16564
(from 3)
Receiving objects: 100% (18575/18575), 10.38 MiB | 9.90 MiB/s, done.
Resolving deltas: 100% (13010/13010), done.

(cowrie@kali)-[~]
$ cd cowrie/

(cowrie@kali)-[~/cowrie]
$ ls
bin          honeyfs      README.rst   setup.py
CHANGELOG.rst  INSTALL.rst  requirements-dev.txt  src
CONTRIBUTING.rst  LICENSE.rst  requirements-output.txt  tox.ini
docker        Makefile     requirements-pool.txt   var
docs          MANIFEST.in  requirements.txt
etc           pyproject.toml  setup.cfg
```

Рисунок 3.15 – Клонування коду з репозиторію

Після завантаження коду було створено віртуальне середовище Python та встановлено всі необхідні залежності згідно з файлом requirements.txt. Приклад реалізації цього кроку представлено на рисунку 3.16.

```

Kali-Linux-Honeypot [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

cowrie@kali: ~/cowrie
File Actions Edit View Help

(cowrie@kali)-[~/cowrie]
$ pwd
/home/cowrie/cowrie

(cowrie@kali)-[~/cowrie]
$ python3 -m venv cowrie-env

(cowrie@kali)-[~/cowrie]
$ source cowrie-env/bin/activate

(cowrie-env)(cowrie@kali)-[~/cowrie]
$ python -m pip install --upgrade pip
Requirement already satisfied: pip in ./cowrie-env/lib/python3.12/site-packag
es (24.3.1)

(cowrie-env)(cowrie@kali)-[~/cowrie]
$ python -m pip install --upgrade -r requirements.txt
Collecting attrs=24.3.0 (from -r requirements.txt (line 1))
  Downloading attrs-24.3.0-py3-none-any.whl.metadata (11 kB)
Collecting bcrypt=4.2.1 (from -r requirements.txt (line 2))
  Downloading bcrypt-4.2.1-cp39-abi3-manylinux_2_28_x86_64.whl.metadata (9.8
kB)
Collecting cryptography=44.0.0 (from -r requirements.txt (line 3))
  Downloading cryptography-44.0.0-cp39-abi3-manylinux_2_28_x86_64.whl.metadat
a (5.7 kB)
Collecting hyperlink=21.0.0 (from -r requirements.txt (line 4))
  Downloading hyperlink-21.0.0-py2.py3-none-any.whl.metadata (1.5 kB)
Collecting idna=3.10 (from -r requirements.txt (line 5))
  Downloading idna-3.10-py3-none-any.whl.metadata (10 kB)

```

Рисунок 3.16 – Завантаження пакетів в середовище

Файли конфігурації для Cowrie розміщені в директорії cowrie/etc і включають cowrie.cfg.dist та cowrie.cfg. Обидва файли завантажуються під час запуску, причому налаштування з cowrie.cfg мають вищий пріоритет. Варто зазначити, що файл cowrie.cfg.dist може бути перезаписаний під час оновлення, тоді як cowrie.cfg залишається незмінним. У нашому випадку необхідно внести певні зміни до файлу cowrie.cfg. Зокрема, слід змінити його назву, щоб уникнути підозри, а також увімкнути функцію telnet. Зміна назва та ввімкнення telnet зображено на рисунку 3.17 та 3.18 відповідно.

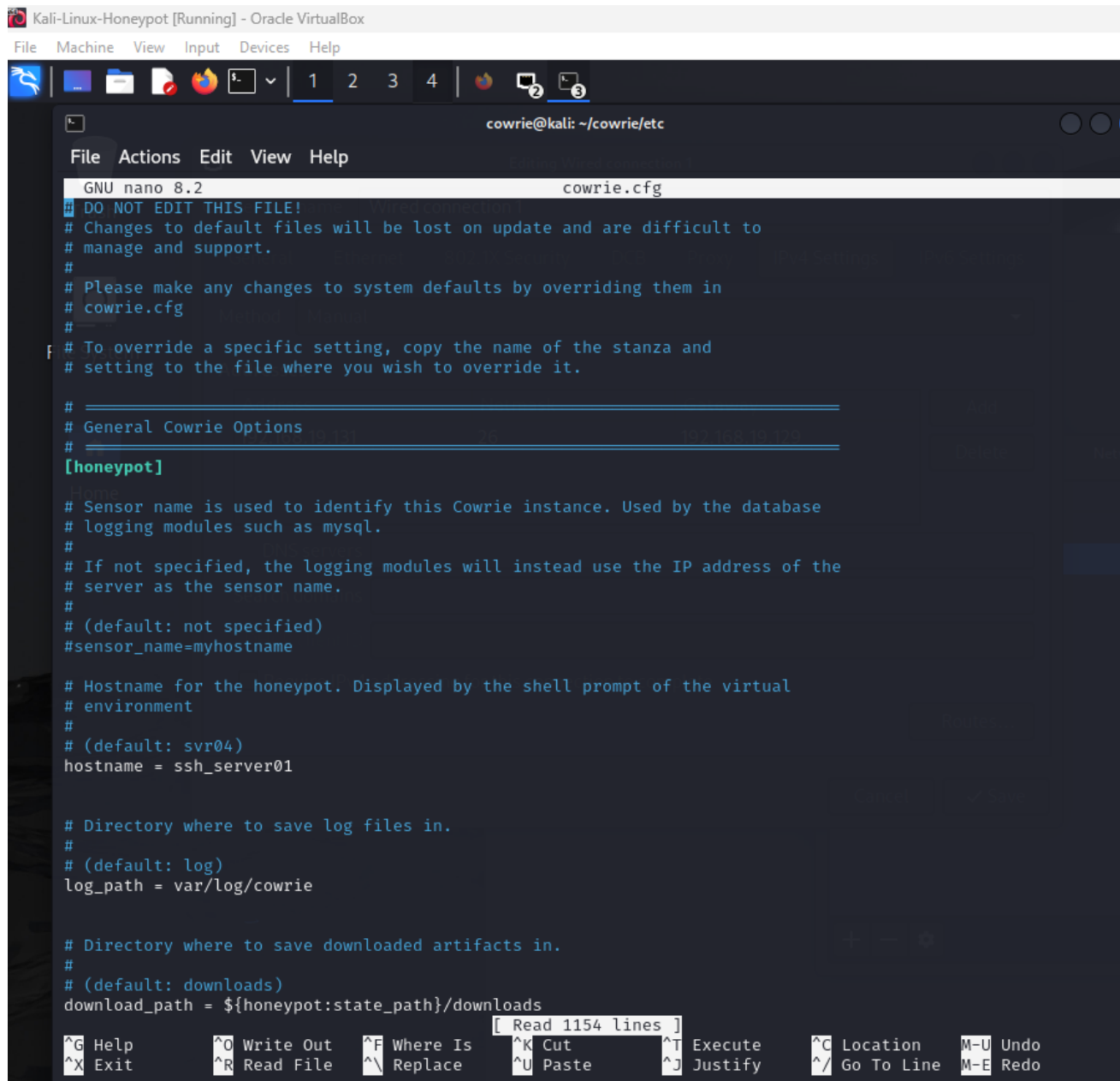
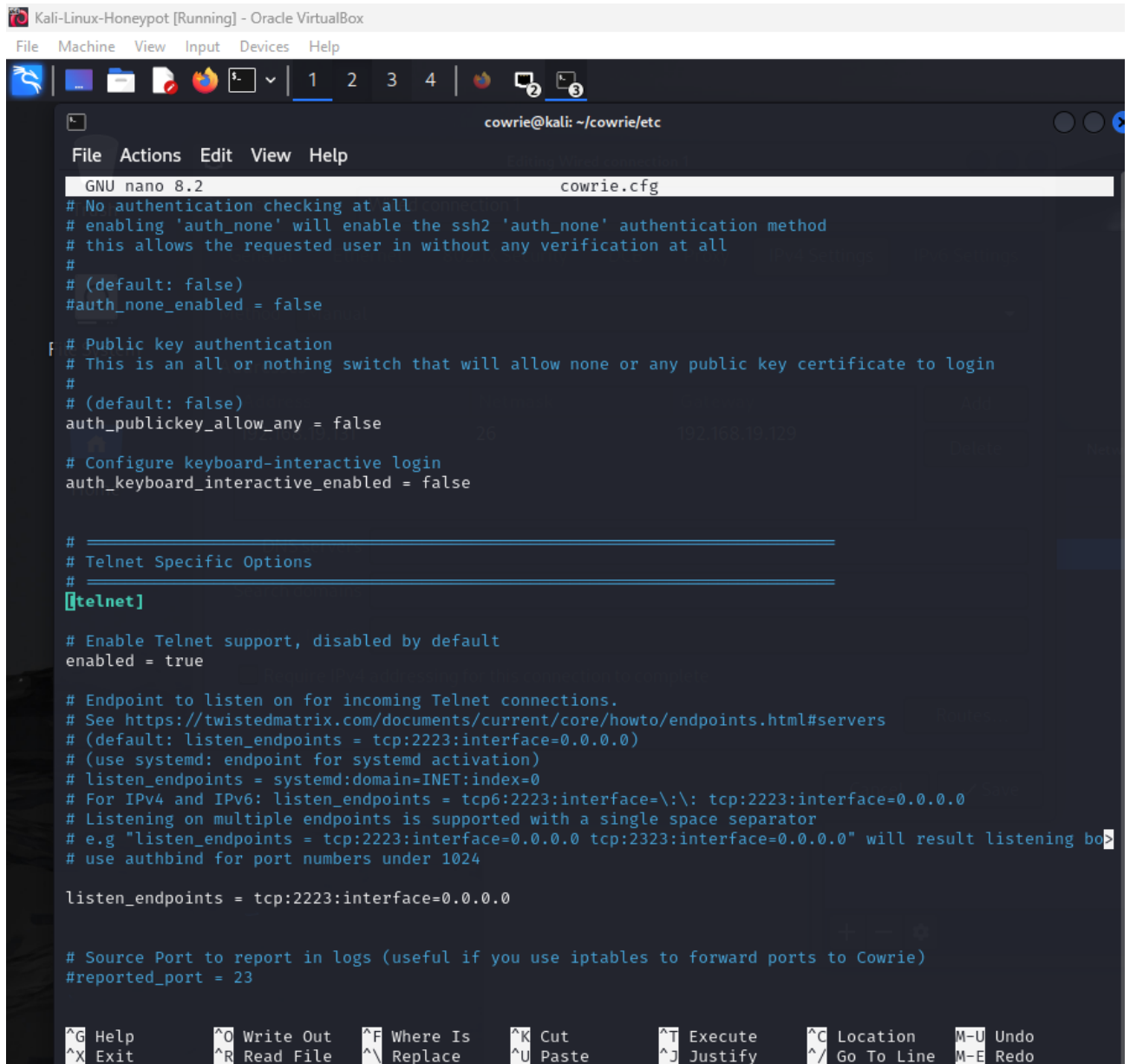


Рисунок 3.17 – Зміна назви

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		35



```
GNU nano 8.2 cowrie.cfg
# No authentication checking at all
# enabling 'auth_none' will enable the ssh2 'auth_none' authentication method
# this allows the requested user in without any verification at all
#
# (default: false)
#auth_none_enabled = false

# Public key authentication
# This is an all or nothing switch that will allow none or any public key certificate to login
#
# (default: false)
auth_publickey_allow_any = false

# Configure keyboard-interactive login
auth_keyboard_interactive_enabled = false

#
# =====
# Telnet Specific Options
# =====

[te]

# Enable Telnet support, disabled by default
enabled = true

# Endpoint to listen on for incoming Telnet connections.
# See https://twistedmatrix.com/documents/current/core/howto/endpoints.html#servers
# (default: listen_endpoints = tcp:2223:interface=0.0.0.0)
# (use systemd: endpoint for systemd activation)
# listen_endpoints = systemd:domain=INET:index=0
# For IPv4 and IPv6: listen_endpoints = tcp6:2223:interface=\::: tcp:2223:interface=0.0.0.0
# Listening on multiple endpoints is supported with a single space separator
# e.g "listen_endpoints = tcp:2223:interface=0.0.0.0 tcp:2323:interface=0.0.0.0" will result listening bo
# use authbind for port numbers under 1024

listen_endpoints = tcp:2223:interface=0.0.0.0

# Source Port to report in logs (useful if you use iptables to forward ports to Cowrie)
#reported_port = 23

^G Help      ^O Write Out ^F Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line M-E Redo
```

Рисунок 3.18 – Ввімкнення Telnet

Для імітації реального сервера необхідно, щоб Cowrie був доступним на стандартних портах – 22 (SSH) та 23 (Telnet). Налаштування портів здійснено згідно з рисунком 3.19.

```

PS> kali@kali: /home/kali

File Actions Edit View Help

└─$ sudo iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRECT --to-port 2022
[sudo] password for cowrie:
Sorry, try again.
[sudo] password for cowrie:
Sorry, try again.
[sudo] password for cowrie:
sudo: 2 incorrect password attempts

(cowrie-env)(cowrie@kali)-[~/cowrie/etc]
└─$ exit
logout

(kali@kali)-[/home/kali]
└─PS> sudo iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRECT --to-port 2022

(kali@kali)-[/home/kali]
└─PS> sudo iptables -t nat -A PREROUTING -p tcp --dport 23 -j REDIRECT --to-port 2023

(kali@kali)-[/home/kali]
└─PS>

```

Рисунок 3.19 – Налаштування портів

Після завершення всіх етапів конфігурації cowrie запускається відповідною командою, яка представлена на рисунку 3.20. З цього моменту система готова до фіксації спроб несанкціонованого доступу [16][17].

```

cowrie@kali: ~/cowrie

File Actions Edit View Help

CHANGELOG.rst  honeyfs  README.rst  setup.py
CONTRIBUTING.rst  INSTALL.rst  requirements-dev.txt
cowrie-env  LICENSE.rst  requirements-output.txt  src
docker  Makefile  requirements-pool.txt  tox.ini
docs  MANIFEST.in  requirements.txt  var

└─(cowrie@kali)-[~/cowrie]
└─$ bin/cowrie start

Join the Cowrie community at: https://www.cowrie.org/slack/

Using default Python virtual environment "/home/cowrie/cowrie/cowrie-env"
Starting cowrie: [twistd --umask=0022 --pidfile=var/run/cowrie.pid --logger cowrie.python.logfile.logger cowrie ]...
/home/cowrie/cowrie/cowrie-env/lib/python3.12/site-packages/twisted/conch/ssh/transport.py:105: CryptographyDeprecationWarning: TripleDES has been moved to cryptography.hazmat.decrepit.ciphers.algorithms.TripleDES and will be removed from cryptography.hazmat.primitives.ciphers.algorithms in 48.0.0.
  b"3des-cbc": (algorithms.TripleDES, 24, modes.CBC),
/home/cowrie/cowrie/cowrie-env/lib/python3.12/site-packages/twisted/conch/ssh/transport.py:112: CryptographyDeprecationWarning: TripleDES has been moved to cryptography.hazmat.decrepit.ciphers.algorithms.TripleDES and will be removed from cryptography.hazmat.primitives.ciphers.algorithms in 48.0.0.
  b"3des-ctr": (algorithms.TripleDES, 24, modes.CTR),

└─(cowrie@kali)-[~/cowrie]
└─$

```

Рисунок 3.20 – Запуск Cowrie

Наступним етапом реалізації Honeypot-системи є встановлення та налаштування веб-емулятора honeyhttpd. Для початку потрібно оновити системи та встановлення необхідних пакетів, зокрема Python3, pip та git. Все це було зроблено раніше. Далі завантажено код проекту з офіційного репозиторію GitHub. Результат наведено на рисунку 3.21.

```

kali@kali: ~/honeyhttpd
File Actions Edit View Help
(kali@kali)-[~]
$ cd honeyhttpd

(kali@kali)-[~/honeyhttpd]
$ ls
config.json      config-test.json  honeyhttpd      LICENSE.txt  README.md  servers  util
config.json.default  docs             honeyhttpd.png  logs        requirements.txt  start.py  venv

(kali@kali)-[~/honeyhttpd]
$ git clone https://github.com/bocajsppear1/honeyhttpd.git

```

Рисунок 3.21 – Клонування Honeyhttpd з GitHub

Далі для налаштування Honeypot-сервера було скопійовано стандартний конфігураційний файл config.json.default із наступним перейменуванням в config.json та редагуванням для адаптації параметрів роботи сервера, зокрема портів та шляхом до сертифікатів TLS. Відредагований конфігураційний файл зі створеним сертифікатом для HTTPS наведено на рисунку 3.22.

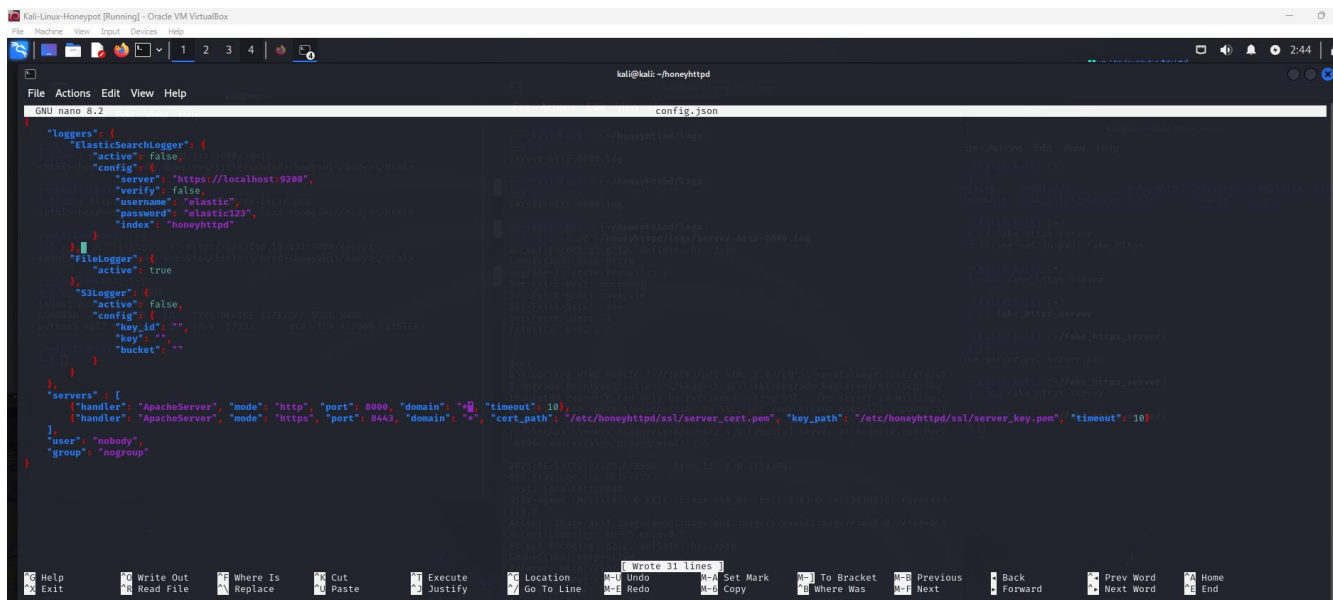


Рисунок 3.22 – Редагування конфігураційного файлу

Завершальним етапом є запуск сервера Honeyhttpd, що дозволяє розпочати роботу та моніторинг вхідних запитів. Запуск сервера наведено на рисунку 3.23 [18].

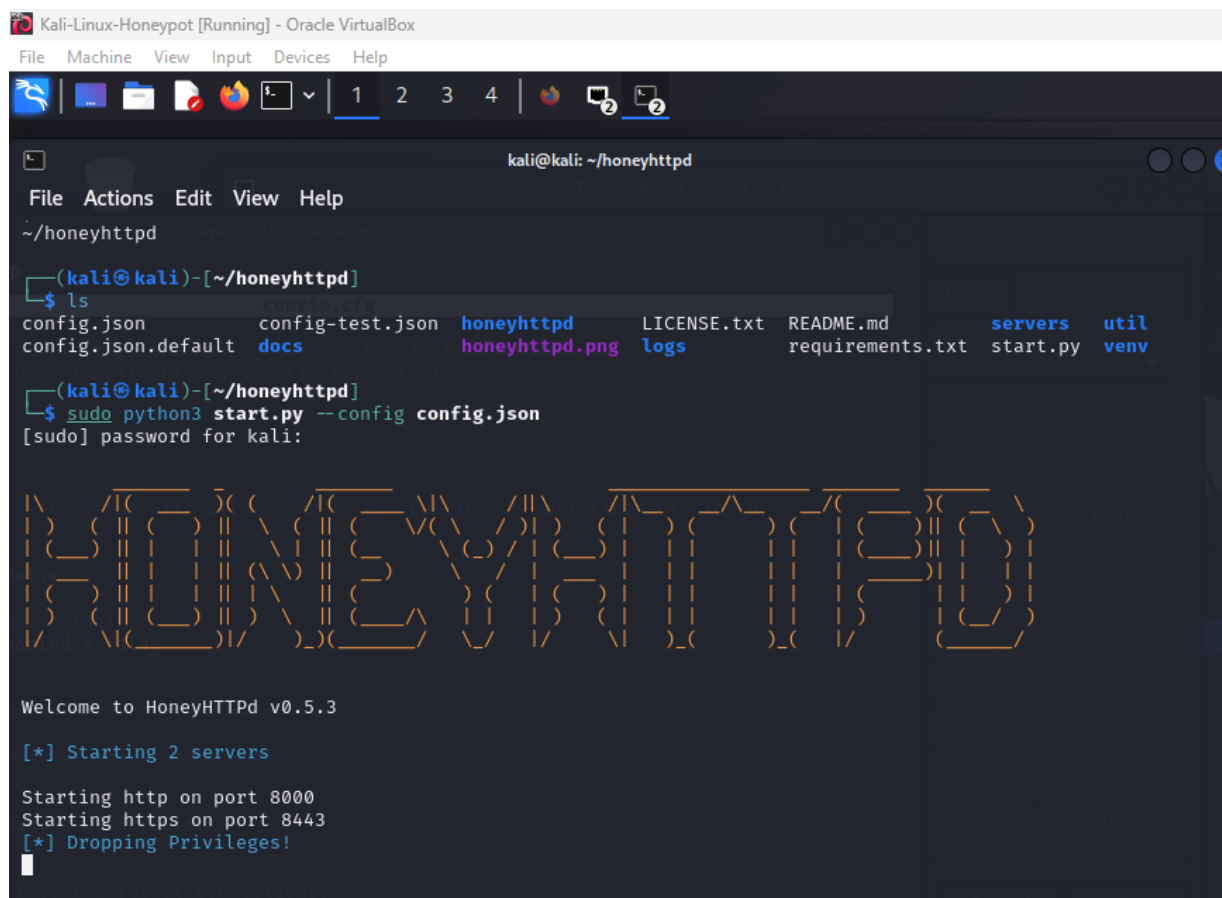


Рисунок 3.23 – Запуск Honeyhttpd

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				39
Змн.	Арк.	№ докум.	Підпис	Дата		

Після завершення налаштування Honeypot-системи наступним етапом є встановлення та конфігурація служби Active Directory на базі Windows Server.

На початковому етапі потрібно встановити необхідний пакет компонентів для розгортання доменної інфраструктури. Процес інсталяції представлено на рисунках 3.24 та 3.25.

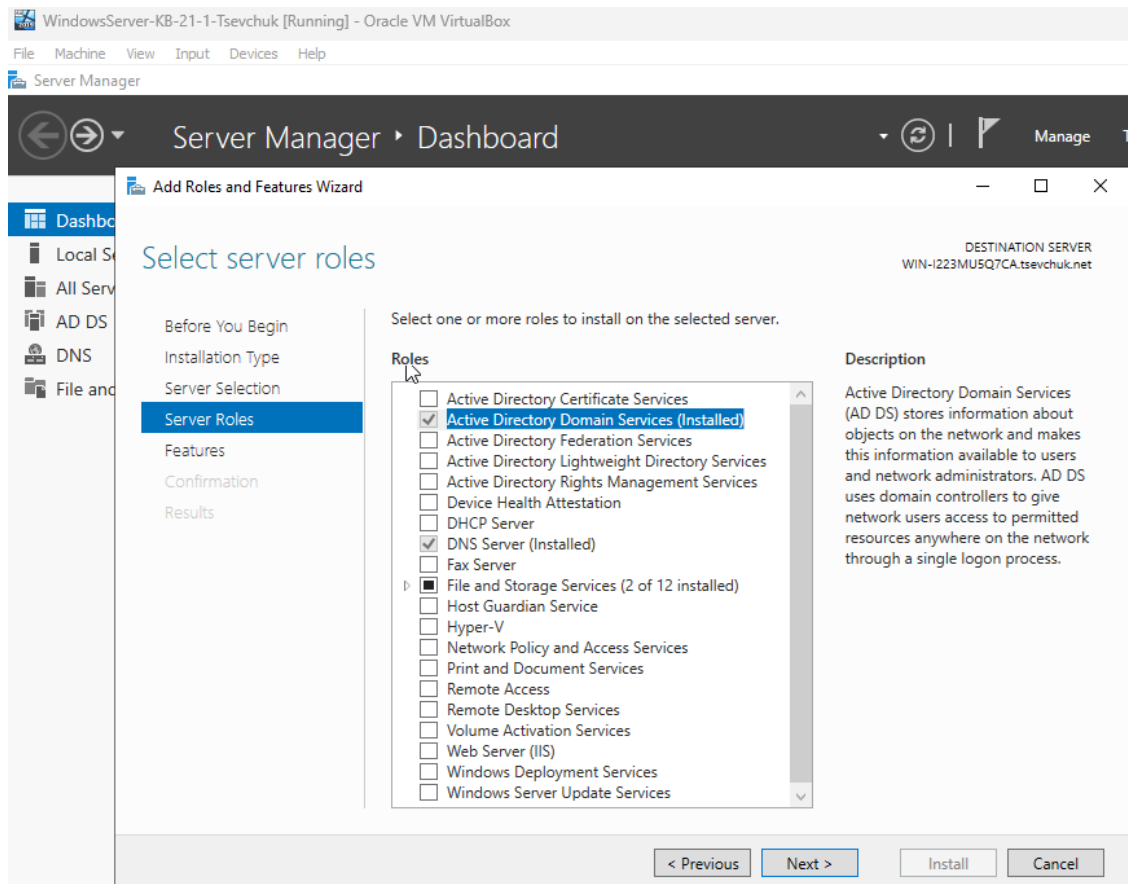


Рисунок 3.24 – Встановлення Active Directory

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				40
Змн.	Арк.	№ докум.	Підпис	Дата		

У межах подальших дій було створено три організаційні одиниці, до яких додано користувачів. Відповідні етапи показано на рисунках 3.27–3.30.

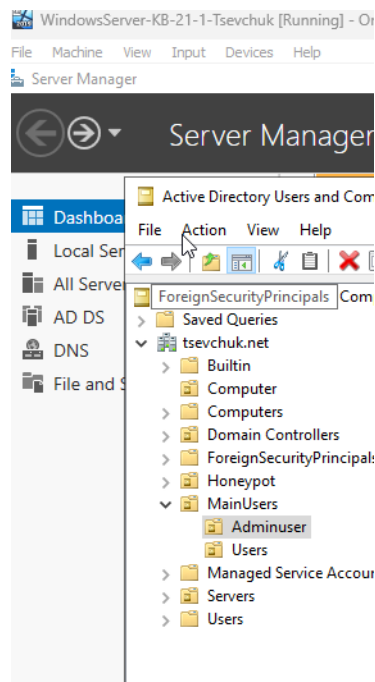


Рисунок 3.27 – Створення організаційних одиниць

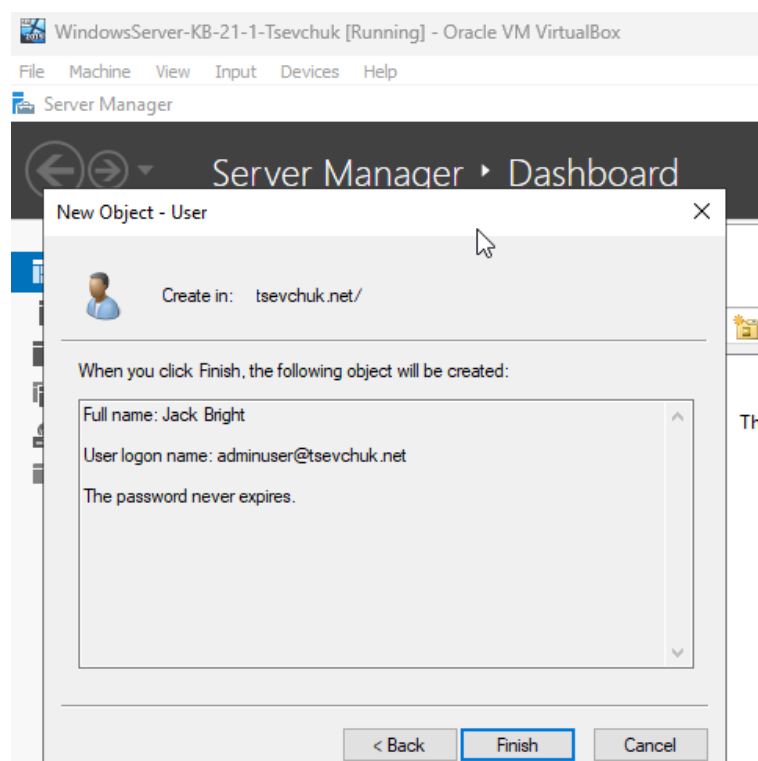


Рисунок 3.28 – Створення користувачів

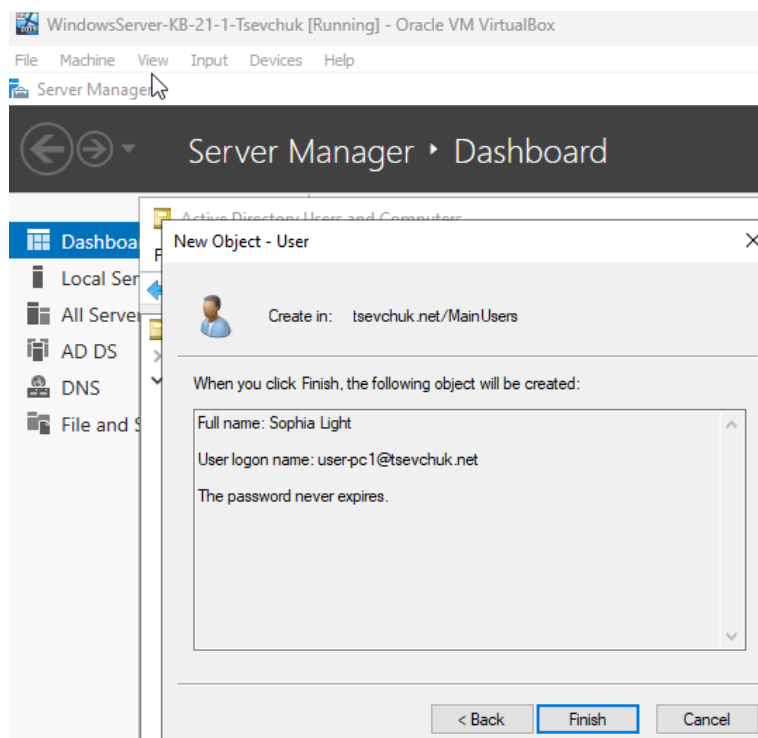


Рисунок 3.29 – Створення користувачів

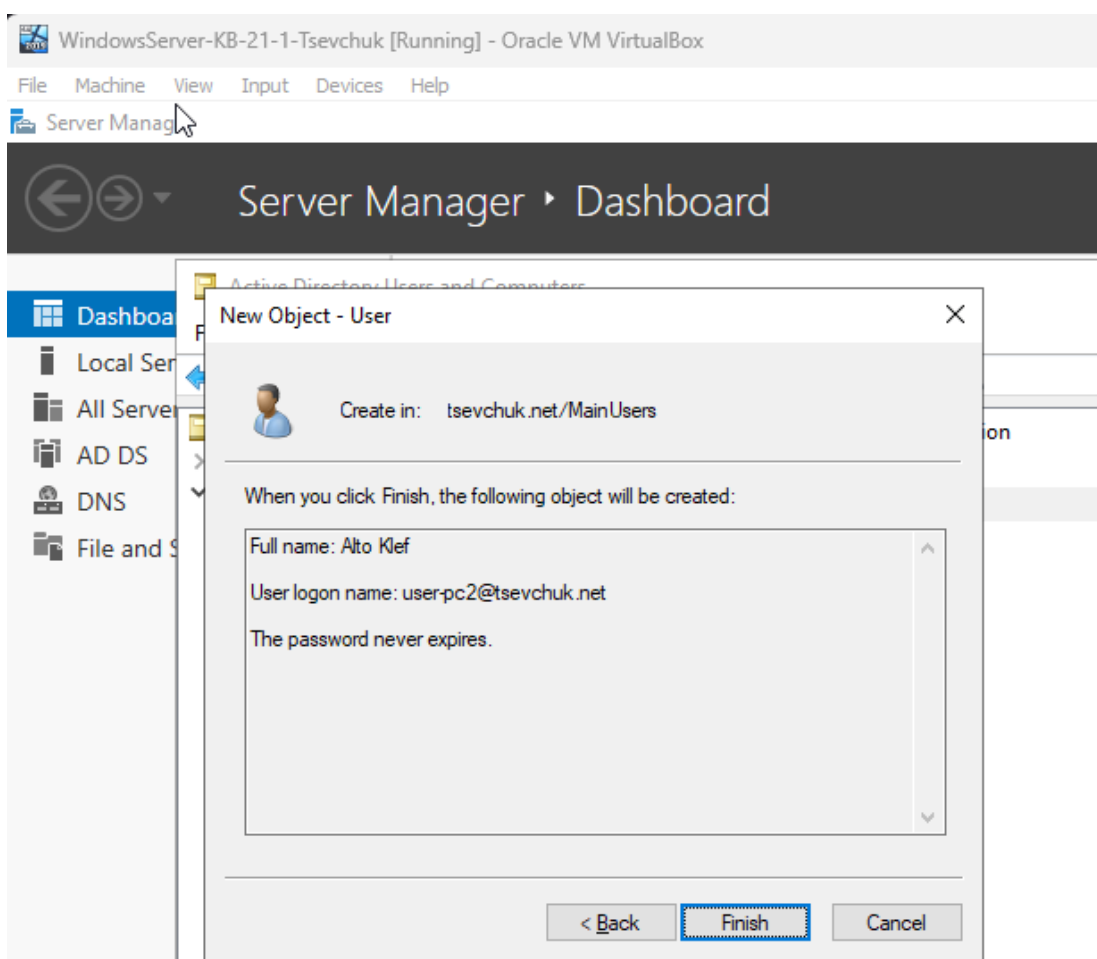


Рисунок 3.30 – Створення користувачів

Наступним кроком стало формування політик безпеки для доменних користувачів. Спершу було визначено політику паролів, зокрема мінімальну довжину, термін дії, термін життя та вимоги до складності. Відповідні налаштування зафіксовані на рисунках 3.31–3.34.

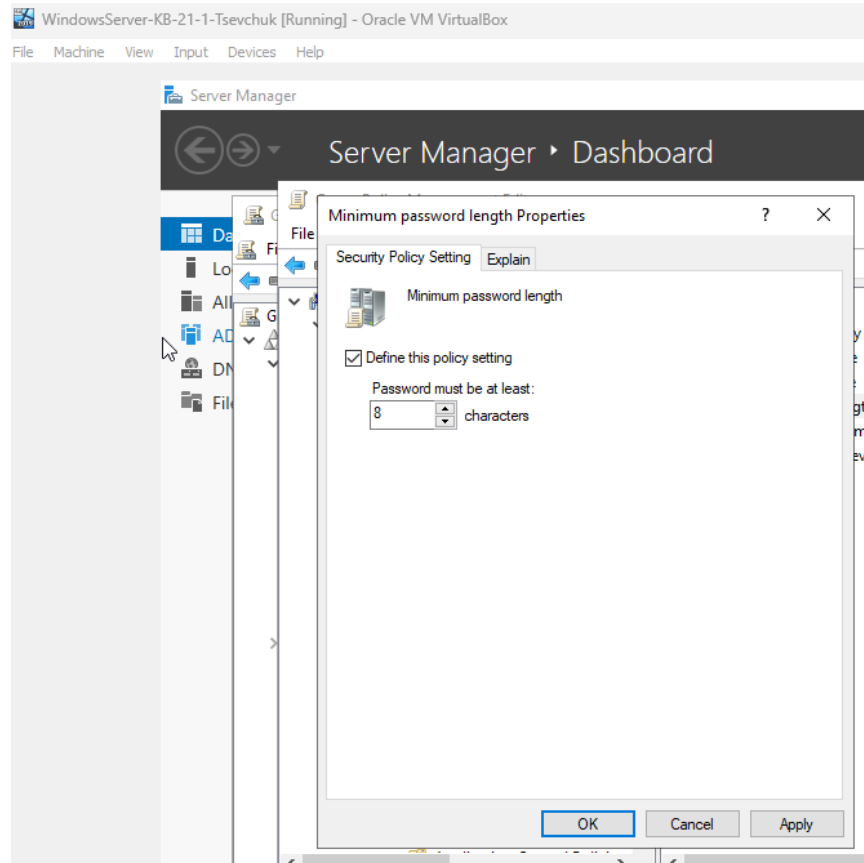


Рисунок 3.31 – Встановлення мінімального паролю

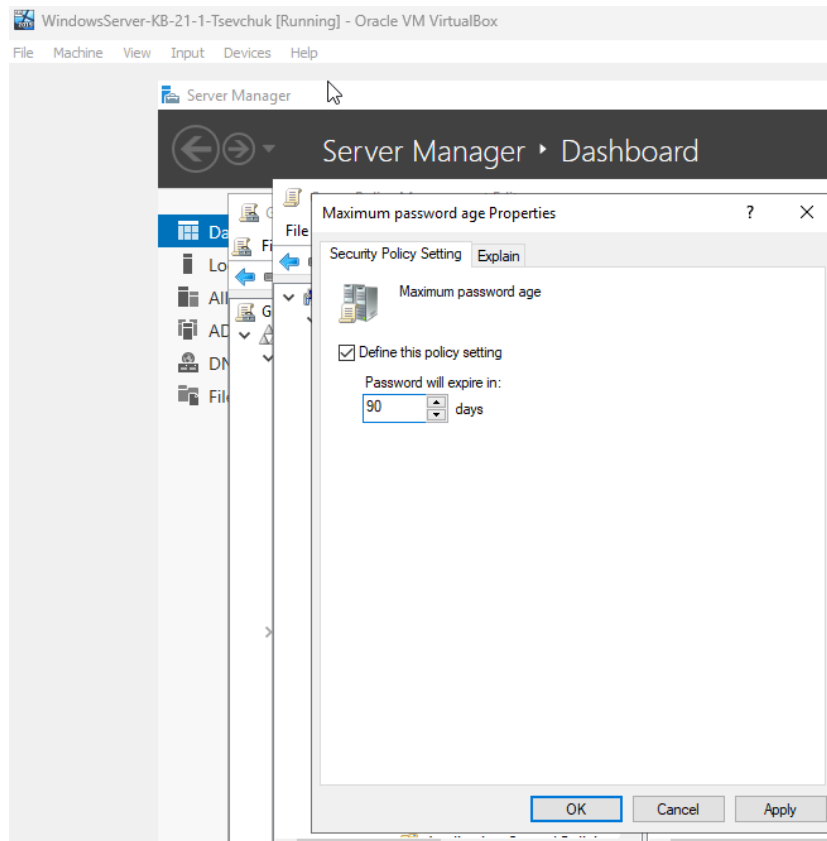


Рисунок 3.32 – Встановлення максимального терміну дії паролю

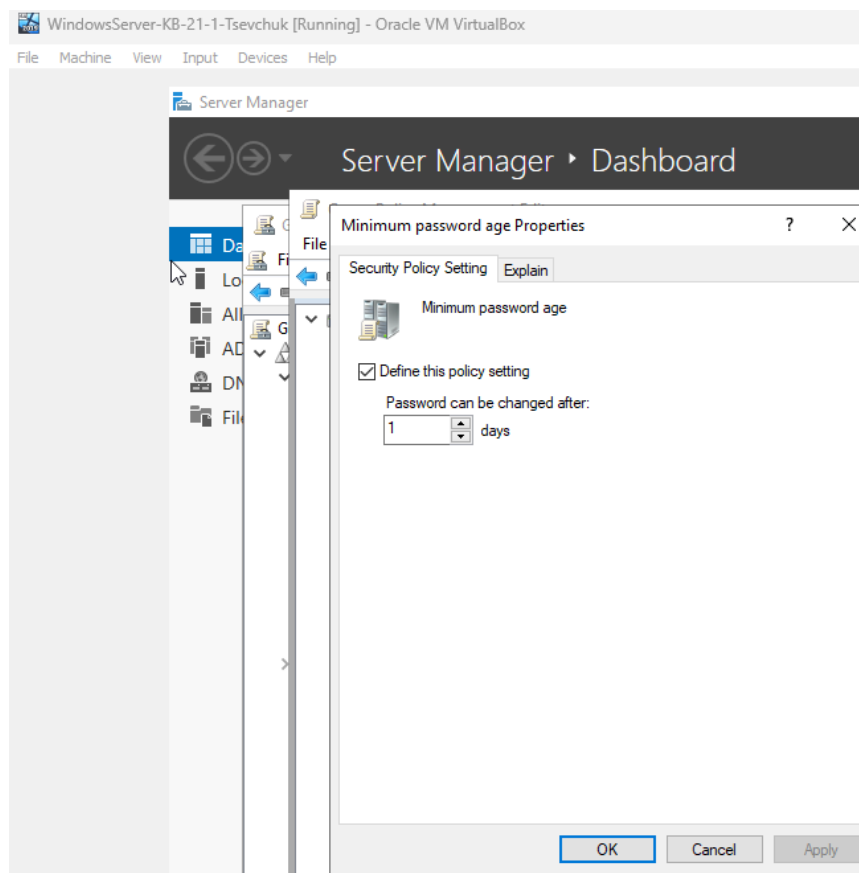


Рисунок 3.33 – Встановлення терміну життя паролю

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		45

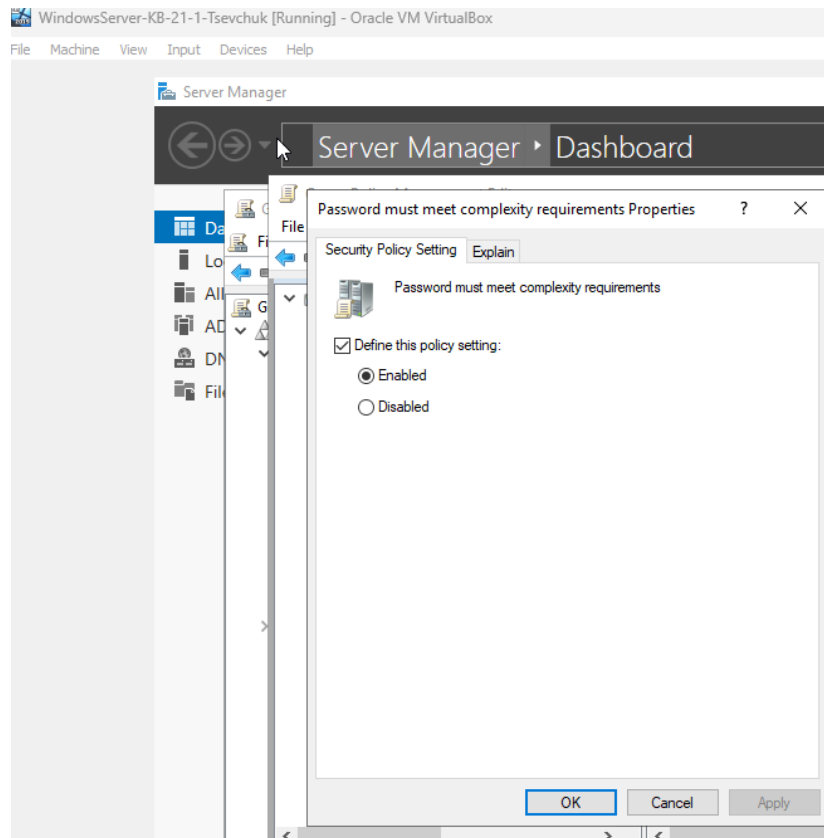


Рисунок 3.34 – Встановлення складності паролю

Також було впроваджено політику блокування облікового запису після п'яти невдалих спроб входу. Налаштування наведені на рисунках 3.35 та 3.36.

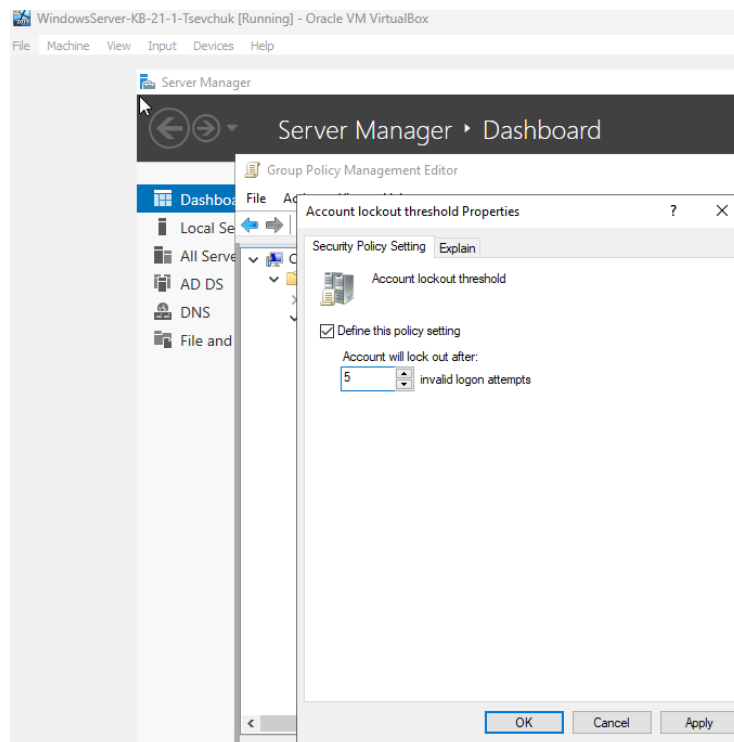


Рисунок 3.35 – Встановлення політики блокування профілю

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				46
Змн.	Арк.	№ докум.	Підпис	Дата		

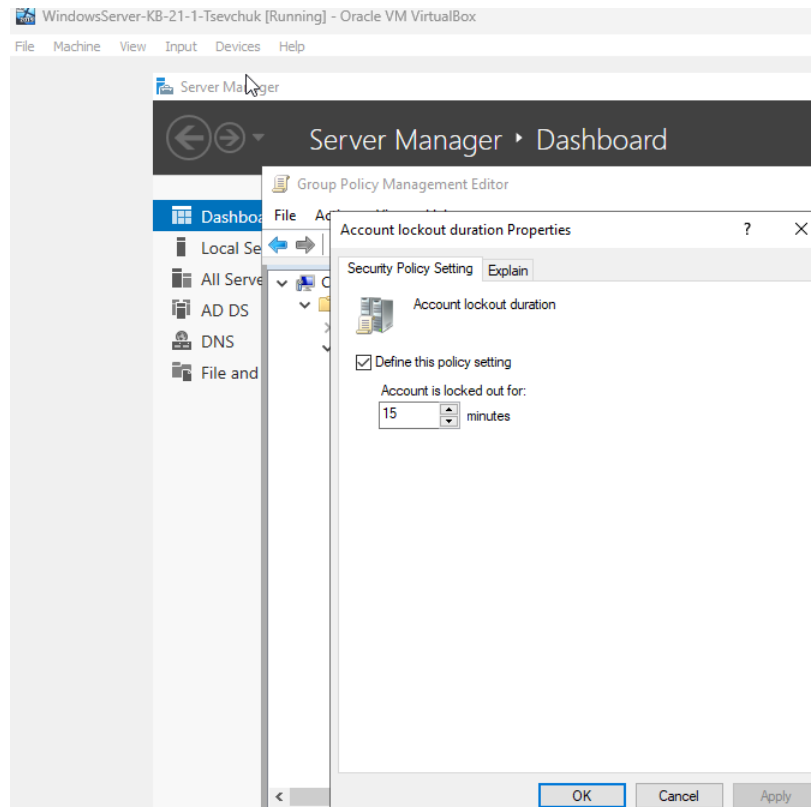


Рисунок 3.36 – Тривалість блокування профілю

Додатково було налаштовано аудит подій входу та виходу користувачів, що представлено на рисунках 3.37 і 3.38.

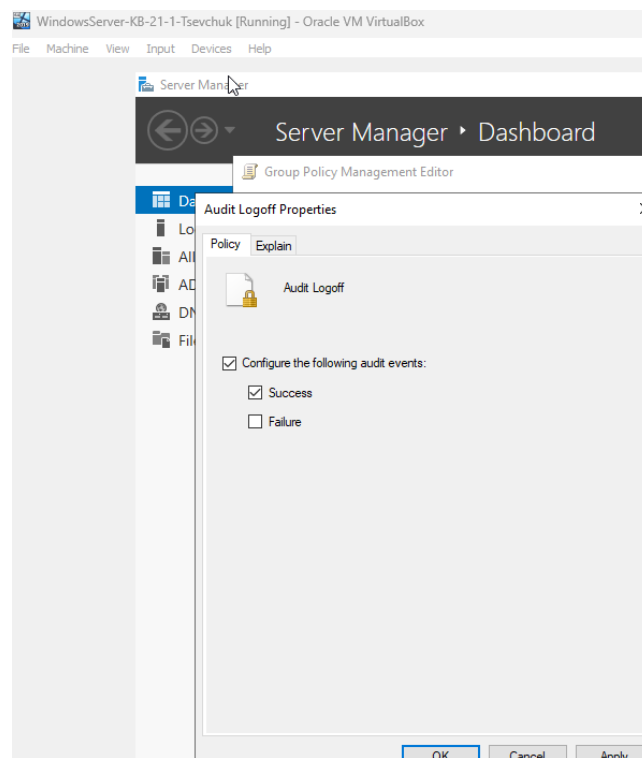


Рисунок 3.37 – Ввімкнення аудиту виходу

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				47
Змн.	Арк.	№ докум.	Підпис	Дата		

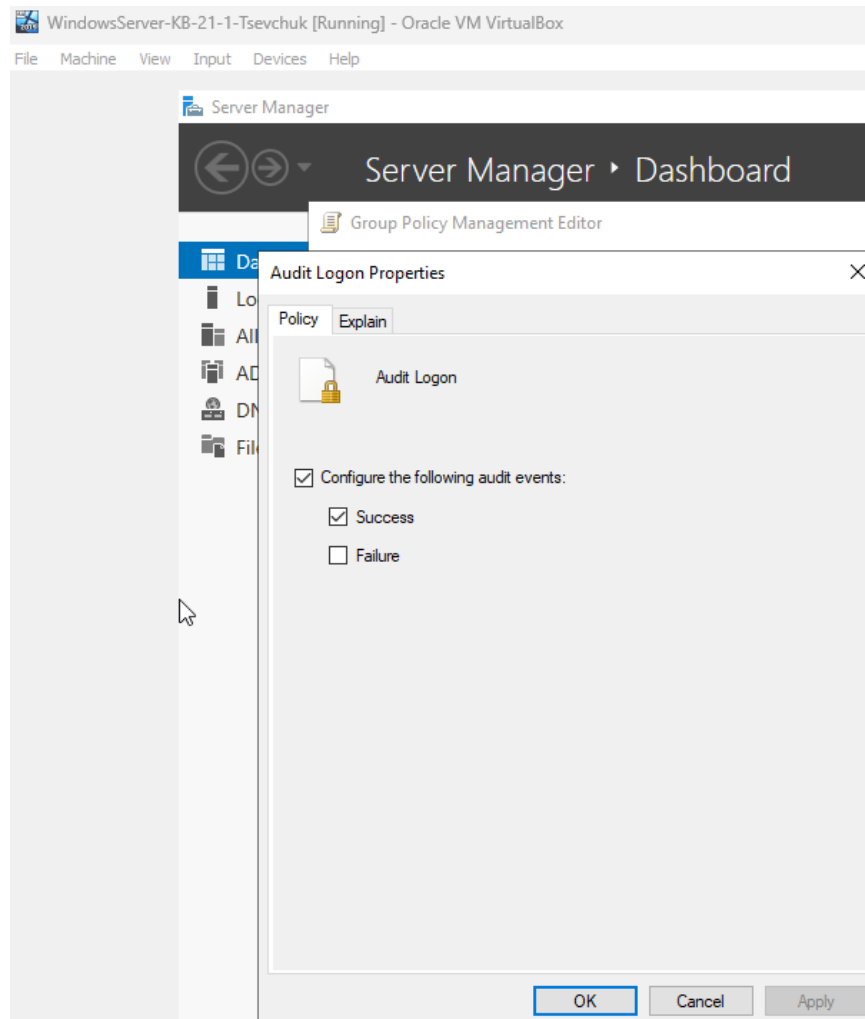


Рисунок 3.38 – Ввімкнення аудиту входу

Далі активовано політику екрана блокування із встановленням таймера очікування. Відповідні параметри показані на рисунках 3.39 та 3.40.

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				48
Змн.	Арк.	№ докум.	Підпис	Дата		

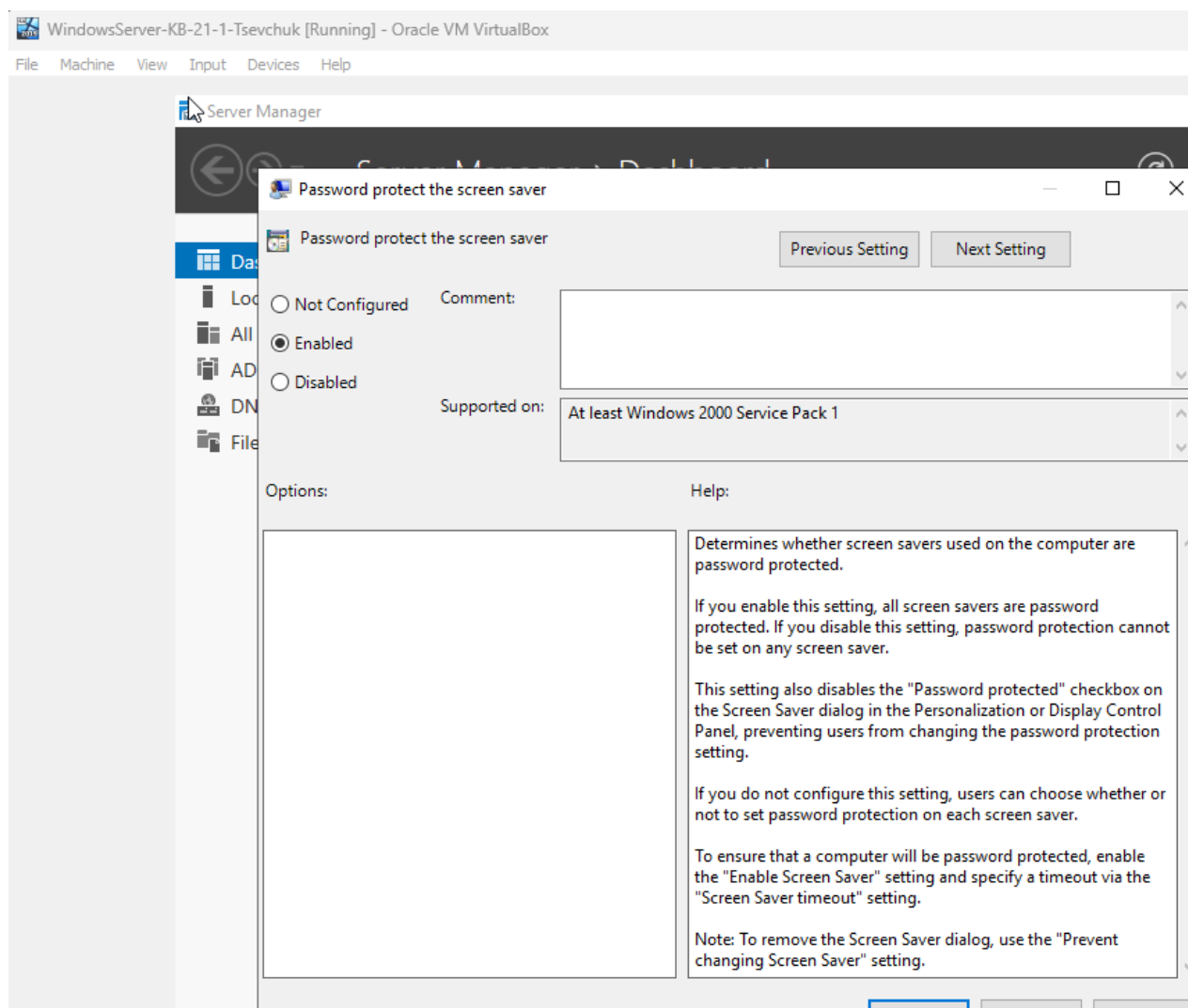


Рисунок 3.39 – Налаштування політики екрана заставки

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				49
Змн.	Арк.	№ докум.	Підпис	Дата		

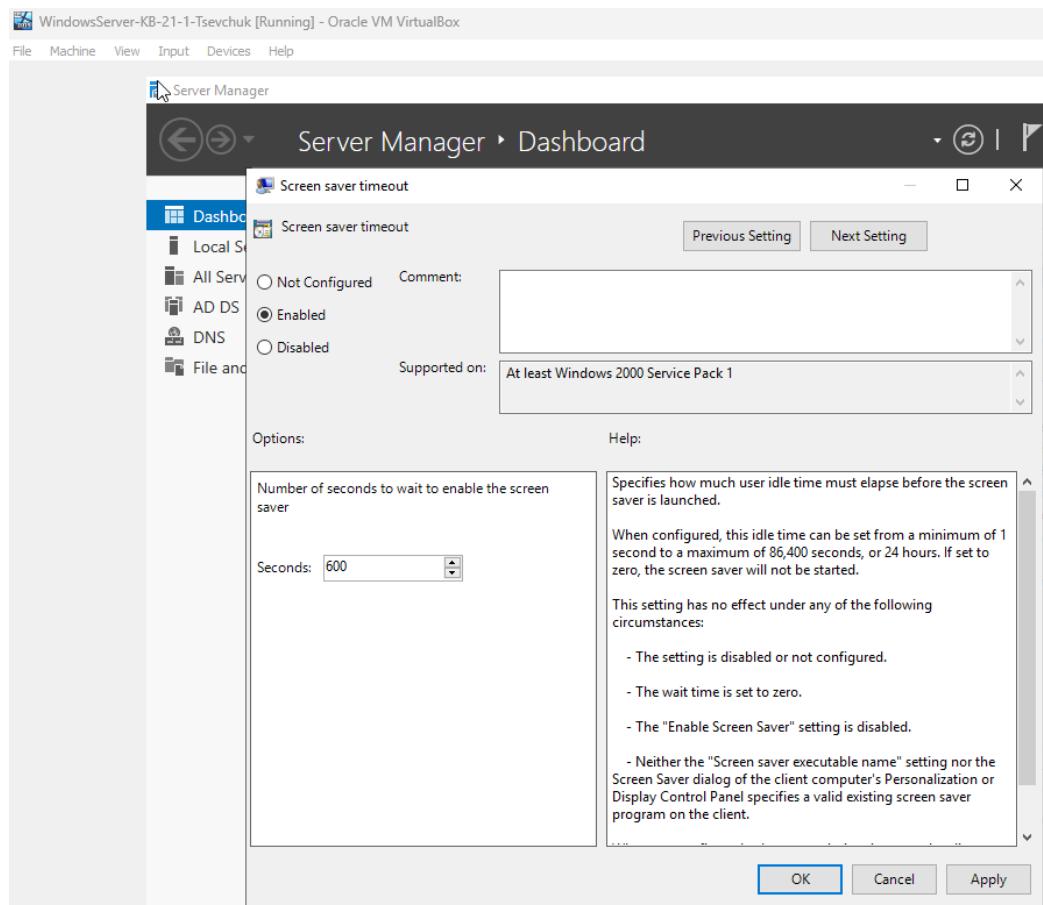


Рисунок 3.40 – Налаштування політики екрана заставки

На завершення, для застосування створених політик було виконано оновлення політик через відповідну команду, як показано на рисунку 3.41.

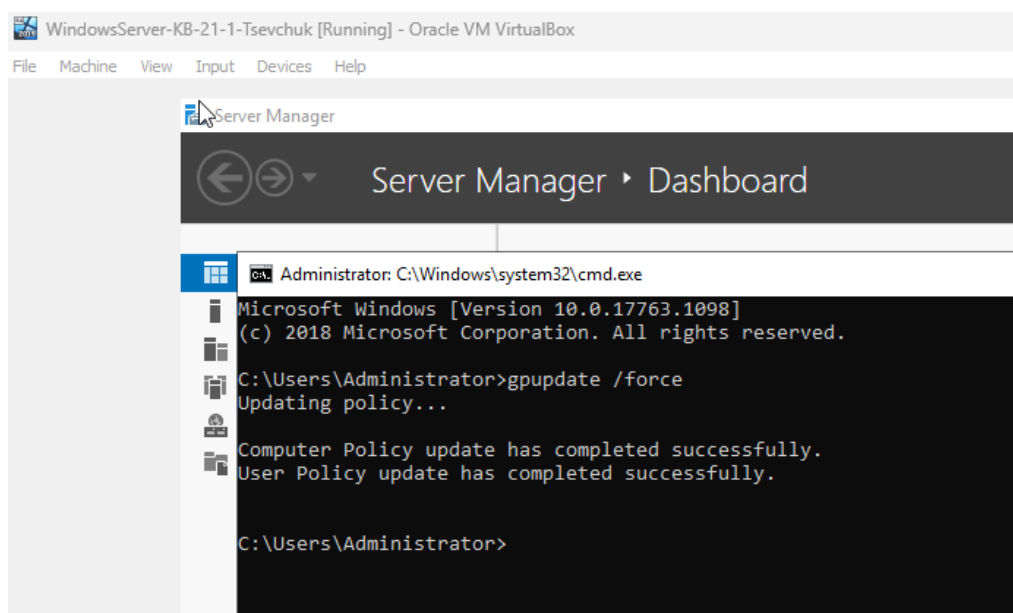


Рисунок 3.41 – Оновлення політик

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		50

Після налаштування ключових елементів внутрішньої інфраструктури наступним етапом стало розгортання проміжного вузла безпеки, який забезпечує контроль і перенаправлення вхідного трафіку. Було встановлено та налаштовано Ubuntu-сервер із двома мережевими інтерфейсами: публічним та приватним.

Першим кроком було увімкнено функцію маршрутизації пакетів між інтерфейсами, що дозволяє серверу передавати трафік з одного сегмента мережі до іншого. Це забезпечує базову можливість реалізації міжмережевої маршрутизації.

Після цього налаштовано правила для переадресації вхідного трафіку з публічного інтерфейсу на конкретний вузол Honeypot. Зокрема, весь трафік, що надходить на порт 22 (SSH), 23 (Telnet), 80 (HTTP), 443 (HTTPS) зі сторони зовнішньої мережі, автоматично переспрямовується на відповідний порт внутрішньої IP-адреси Honeypot-сервера. Відповідні налаштування наведені на рисунку 3.42 та 3.43.

```

3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:cb:b7:b5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.19.135/26 brd 192.168.19.191 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:feeb:b7b5/64 scope link
        valid_lft forever preferred_lft forever
tsevchuk@ununtuserv-tsevchuk:~$ sudo sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
tsevchuk@ununtuserv-tsevchuk:~$ echo 'net.ipv4.ipforward=1' | sudo tee -a /etc/sysctl.conf
net.ipv4.ipforward=1
tsevchuk@ununtuserv-tsevchuk:~$ sudo iptables -t net -A PREROUTING -i enp0s3 -p tcp --dport 22 -j MASQUERADE
iptables v1.8.7 (nf_tables): table 'net' does not exist
Perhaps iptables or your kernel needs to be upgraded.
tsevchuk@ununtuserv-tsevchuk:~$ sudo iptables -t nat -A PREROUTING -i enp0s3 -p tcp --dport 22 -j MASQUERADE
iptables v1.8.7 (nf_tables): table 'nat' does not exist
Perhaps iptables or your kernel needs to be upgraded.
tsevchuk@ununtuserv-tsevchuk:~$ sudo iptables -t nat -A PREROUTING -i enp0s3 -p tcp --dport 22 -j MASQUERADE
iptables v1.8.7 (nf_tables): table 'nat' does not exist
Perhaps iptables or your kernel needs to be upgraded.
tsevchuk@ununtuserv-tsevchuk:~$ sudo iptables -A FORWARD -i enp0s3 -o enp08 -p tcp --dport 22 -d 192.168.19.131 -j ACCEPT
tsevchuk@ununtuserv-tsevchuk:~$ sudo iptables -t nat -A POSTROUTING -o enp0s8 -j MASQUERADE
tsevchuk@ununtuserv-tsevchuk:~$
tsevchuk@ununtuserv-tsevchuk:~$

```

Рисунок 3.42 – Налаштування iptables

```

inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
inet6 ::1/128 scope host
    valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:08:dc:09 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 83936sec preferred_lft 83936sec
    inet6 fd17:625c:f037:2:a00:27ff:fe08:dc09/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 86198sec preferred_lft 14198sec
    inet6 fe80::a00:27ff:fe08:dc09/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:cb:b7:b5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.19.135/26 brd 192.168.19.191 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe08:b7b5/64 scope link
        valid_lft forever preferred_lft forever
tsevchuk@ununtuserv-tsevchuk:~$ sudo tcpdump -i enp0s8 icmp
[sudo] password for tsevchuk:
sudo: a password is required
tsevchuk@ununtuserv-tsevchuk:~$ sudo iptables -t nat -A PREROUTING -i enp0s3 -p tcp --dport 2223 -j DNAT --to-destination 192.168.19.131:23
[sudo] password for tsevchuk:
tsevchuk@ununtuserv-tsevchuk:~$ sudo iptables -A FORWARD -p tcp -d 192.168.19.131 --dport 23 -j ACCEPT
tsevchuk@ununtuserv-tsevchuk:~$ sudo iptables -t nat -A PREROUTING -i enp0s3 -p tcp --dport 8080 -j DNAT --to-destination 192.168.19.131:8000
tsevchuk@ununtuserv-tsevchuk:~$ sudo iptables -A FORWARD -p tcp -d 192.168.19.131 --dport 8000 -j ACCEPT
tsevchuk@ununtuserv-tsevchuk:~$ sudo iptables -t nat -A PREROUTING -i enp0s3 -p tcp --dport 8443 -j DNAT --to-destination 192.168.19.131:443
tsevchuk@ununtuserv-tsevchuk:~$ sudo iptables -A FORWARD -p tcp -d 192.168.19.131 --dport 443 -j ACCEPT
tsevchuk@ununtuserv-tsevchuk:~$

```

Рисунок 3.43 – Налаштування iptables

Було так же впроваджено інструмент Fail2Ban, який автоматично блокує IP-адреси після кількох невдалих спроб входу. Для цього було налаштовано конфігураційний файл jail.local, де активовано відповідний jail для служби sshd. Конфігураційний файл наведено на рисунку 3.43.

```

GNU nano 6.2 /etc/fail2ban/jail.local *
# JAILS
#
#
# SSH servers
#
[sshd]
# To use more aggressive sshd modes set filter parameter "mode" in jail.local:
# normal (default), ddos, extra or aggressive (combines all).
# See "tests/files/logs/ssh" or "filter.d/ssh.conf" for usage example and details.
#mode = normal
port = 22
filter = sshd
logpath = /var/log/auth.log
maxretry = 3
bantime = 1h
backend = %(sshd_backend)s

[dropbear]
port = ssh
logpath = %(dropbear_log)s
backend = %(dropbear_backend)s

[selinux-ssh]
port = ssh
logpath = %(auditd_log)s
  
```

Рисунок 3.44 – Налаштування конфігураційного файлу jail.local

Останнім етапом стало збереження налаштувань після перезавантаження, які наведені на рисунку 3.45.

```

tsevchuk@ununtuserv-tsevchuk:~$
tsevchuk@ununtuserv-tsevchuk:~$ sudo apt install iptables-persistent
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
iptables-persistent is already the newest version (1.0.16).
оновлено 0, встановлено 0 нових, 0 відмічено для видалення і 86 не оновлено.
tsevchuk@ununtuserv-tsevchuk:~$ sudo netfilter-persistent save
sudo: netfilter-persistent: command not found
tsevchuk@ununtuserv-tsevchuk:~$ sudo netfilter-persistent save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables save
tsevchuk@ununtuserv-tsevchuk:~$
  
```

Рисунок 3.45 – Збереження налаштувань

3.3. Тестування функціональності Honeypot-підсистеми

Передусім необхідно перевірити працездатність встановленої системи Cowrie на Honeypot-сервері, зокрема фіксацію дій потенційного зловмисника у журналах подій. Було здійснено серію тестових дій, що імітують поведінку потенційного зловмисника. Зокрема, виконано сканування відкритих портів, а також здійснено спробу входу з використанням типових облікових даних за замовчуванням. Після встановлення з'єднання були введені команди типу whoami та cat /etc/passwd, що характерно для початкового етапу розвідки в межах реальної атаки. Відповідні результати журналювання наведено на рисунках 3.46 та 3.35.

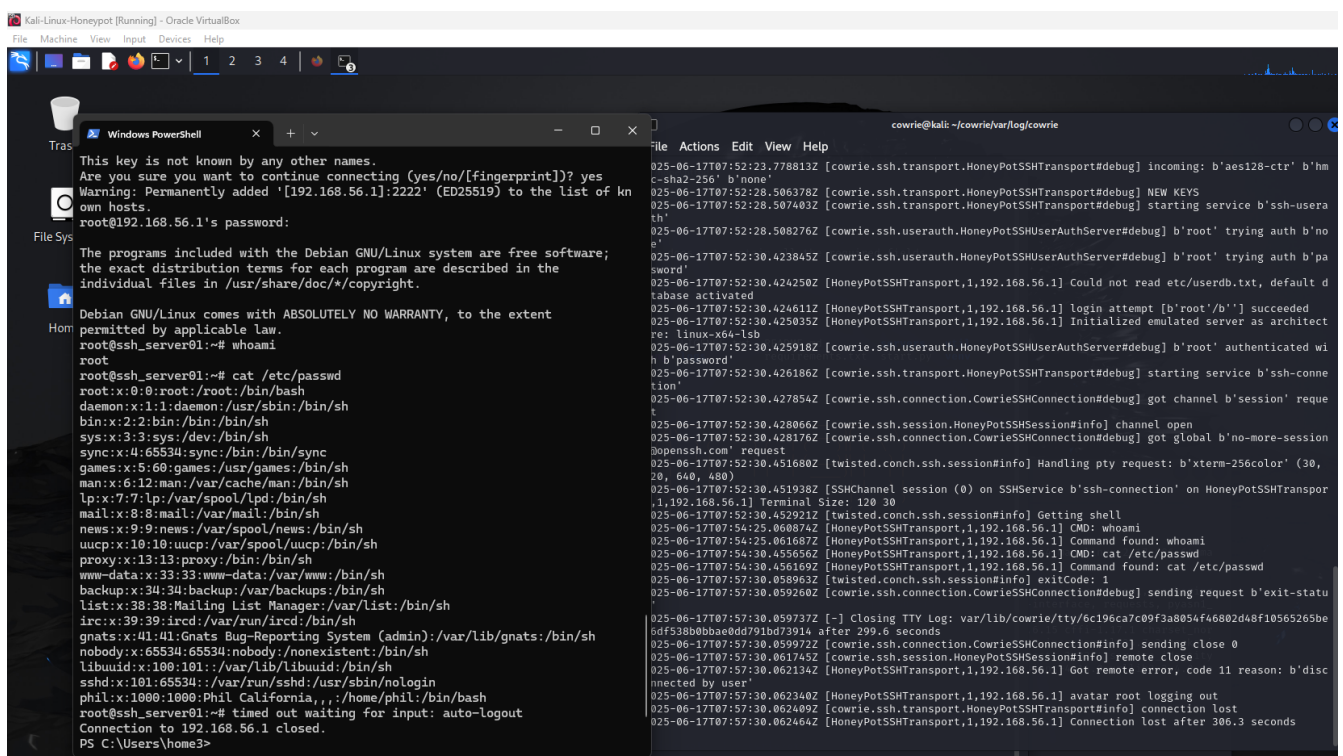


Рисунок 3.46 – Перевірка логувань дій зловмисника

Після успішної перевірки роботи емулятора Cowrie та фіксації дій, наступним етапом стало тестування HTTP/HTTPS-складової Honeypot-системи. Для цього було проведено перевірку працездатності компонента HoneypotHTTPD, що відповідає за імітацію вебсервера та фіксацію спроб взаємодії з відкритими портами. Результат роботи наведено на рисунках 3.47, 3.48.

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		54

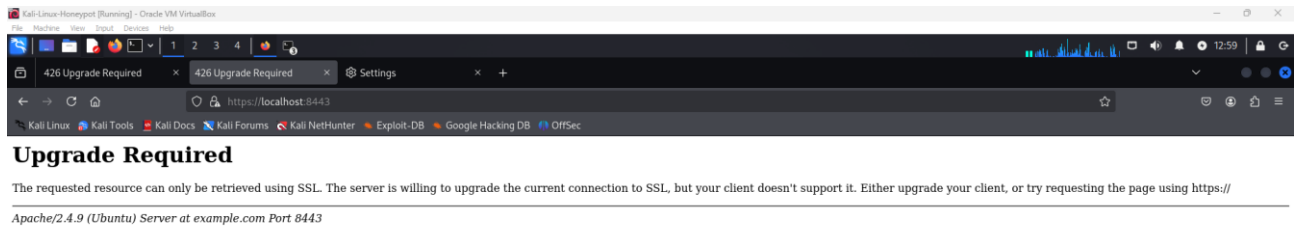


Рисунок 3.47 – Емітація веб-сервера портів HTTPS

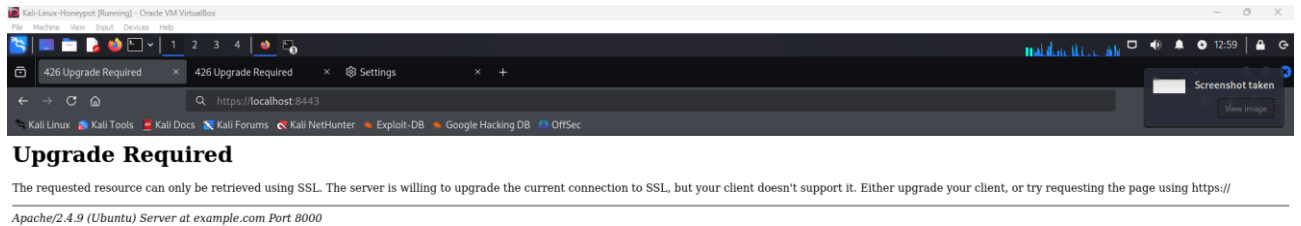


Рисунок 3.48 – Емітація веб-сервера з портом HTTP

Далі було змодельовано декілька типових сценаріїв атак на вебсервер. Зокрема, було здійснено сканування поширених потенційно вразливих URL-адрес, зокрема `admin.php` та `wp-login.php`, з метою перевірки наявності панелі адміністратора чи CMS-інтерфейсу. Цей етап імітує автоматизоване сканування директорій, яке часто передуює спробам несанкціонованого доступу.

Також було здійснено спробу SQL-ін'єкції за допомогою модифікованого запиту до сторінки `index.php?id=1' OR '1'=1`, що дозволило перевірити здатність HoneyHTTPD фіксувати приклади атак на рівні параметрів HTTP-запитів. Усі спроби були успішно зафіксовані в логах HoneyHTTPD, що підтверджує рисунок 3.49.

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		55

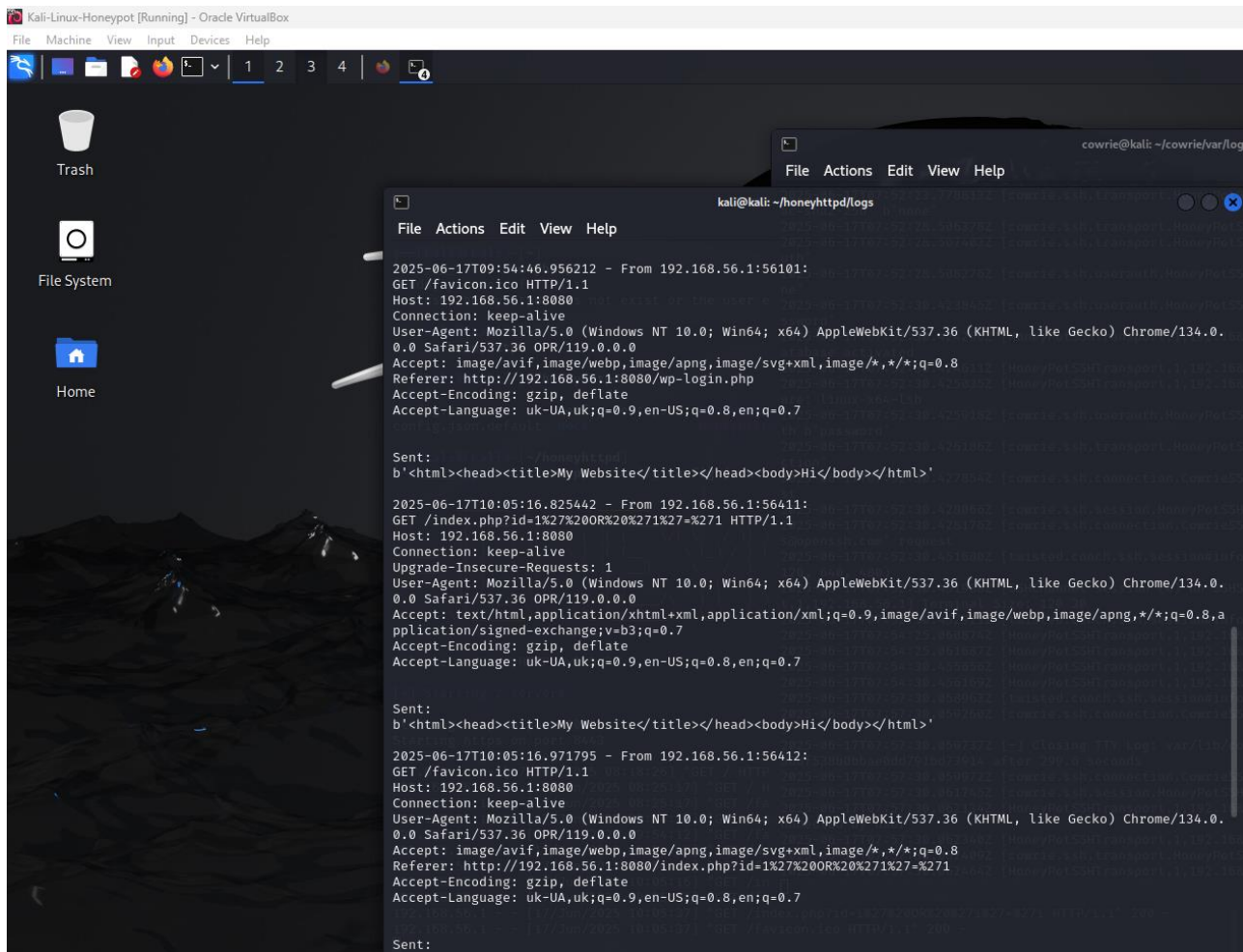


Рисунок 3.49 – Перевірка логувань дій зловмисника

Для перевірки коректності функціонування служби Active Directory було проведено низку тестів. На першому етапі перевірено, чи була робоча станція успішно додана до домену. Відповідний результат представлено на рисунку 3.50.

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		56

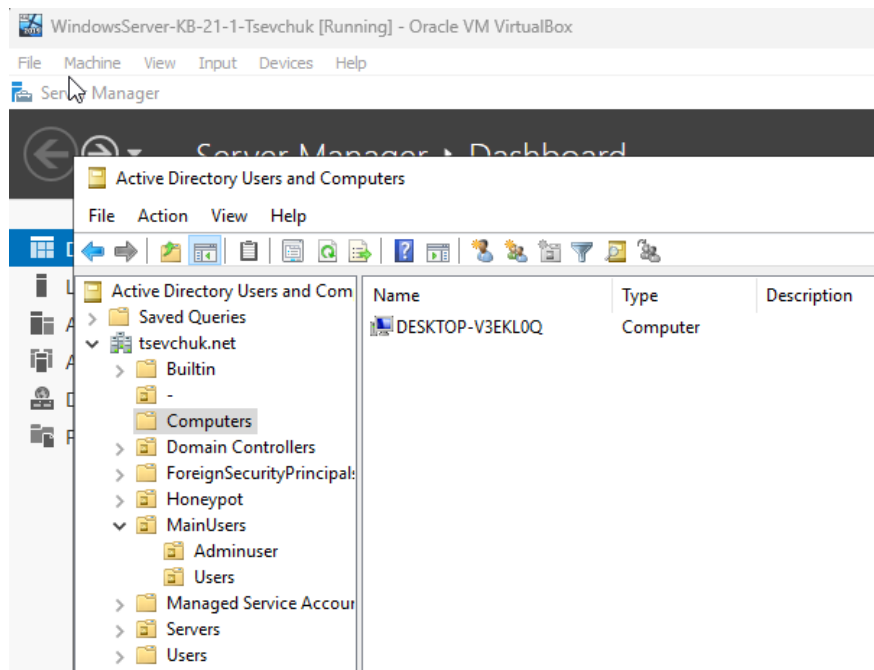


Рисунок 3.50 – Перевірка чи додана станція до домену

Наступним кроком стала перевірка авторизації створених користувачів у середовищі домену – зокрема, можливість входу на робочу станцію під їхніми обліковими записами. Результати цієї перевірки наведено на рисунках 3.51 та 3.52.

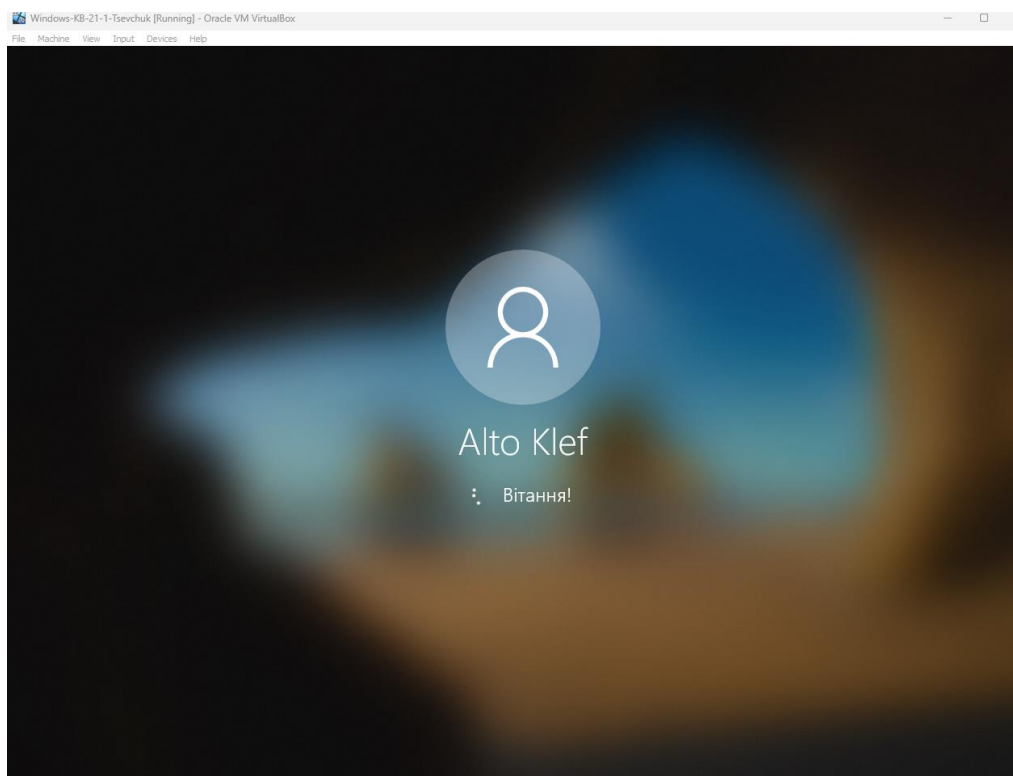


Рисунок 3.51 – Успішний вхід з першого користувача

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				57
Змн.	Арк.	№ докум.	Підпис	Дата		

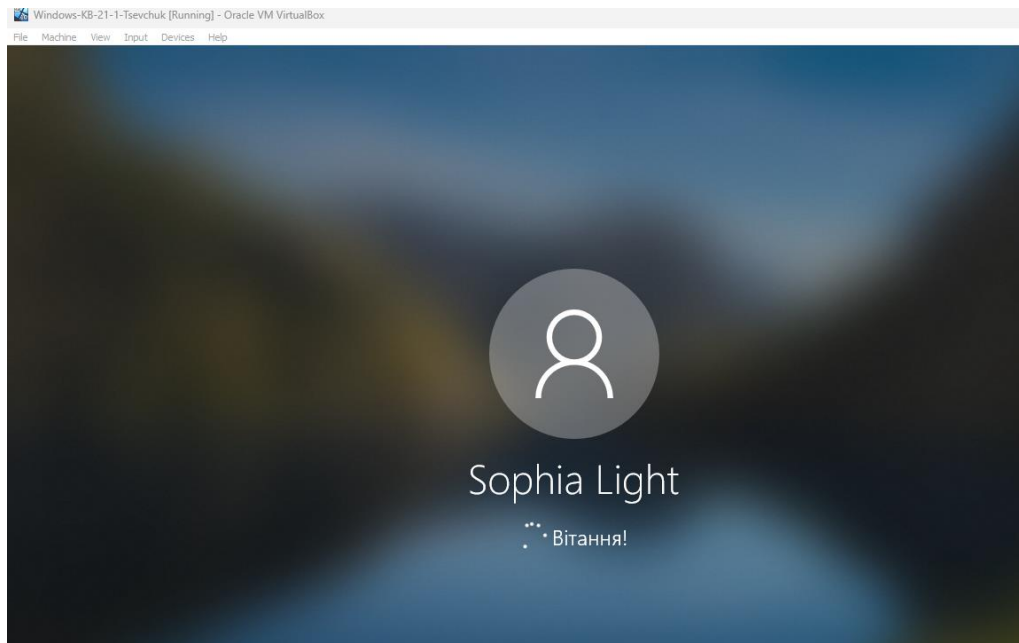


Рисунок 3.52 – Успішний вхід з другого користувача

Остаточним етапом перевірки стала перевірка налаштувань політик паролів та аудиту входу і виходу користувачів. Результати наведено на рисунку 3.53 та 3.54.

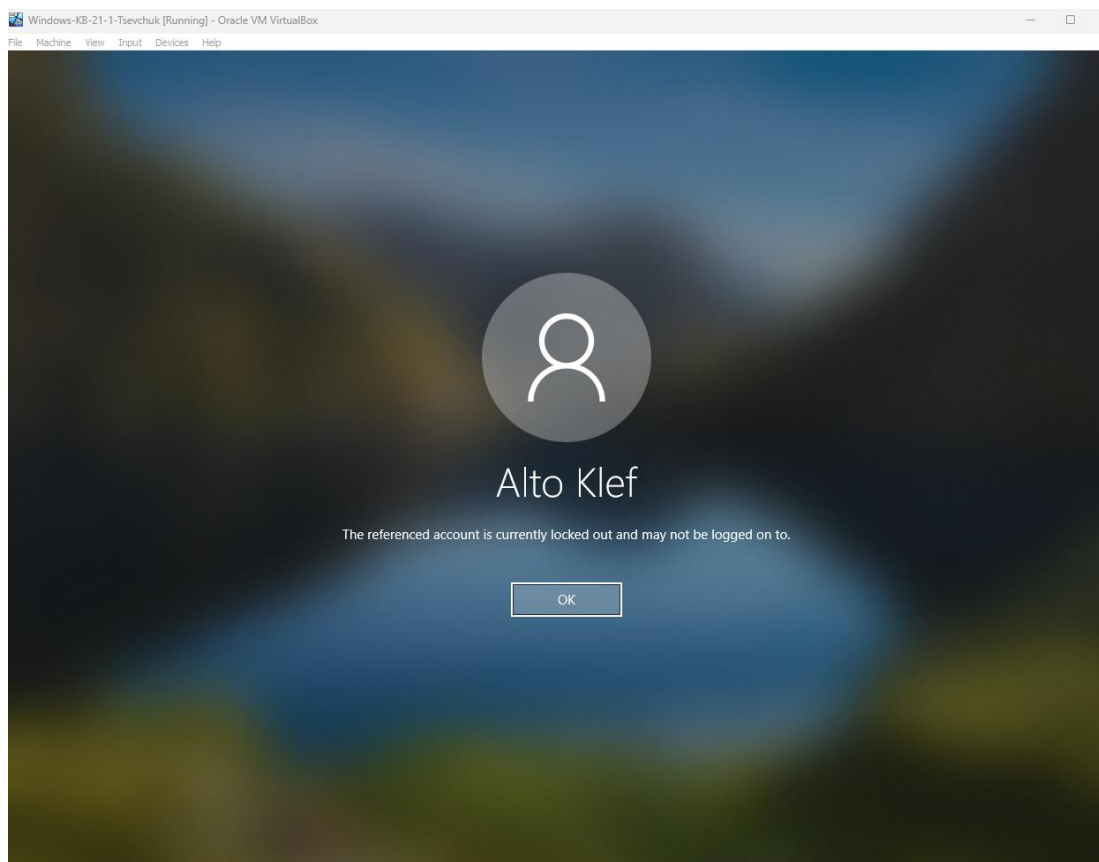


Рисунок 3.53 – Блокування при невдалих спробах

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		58

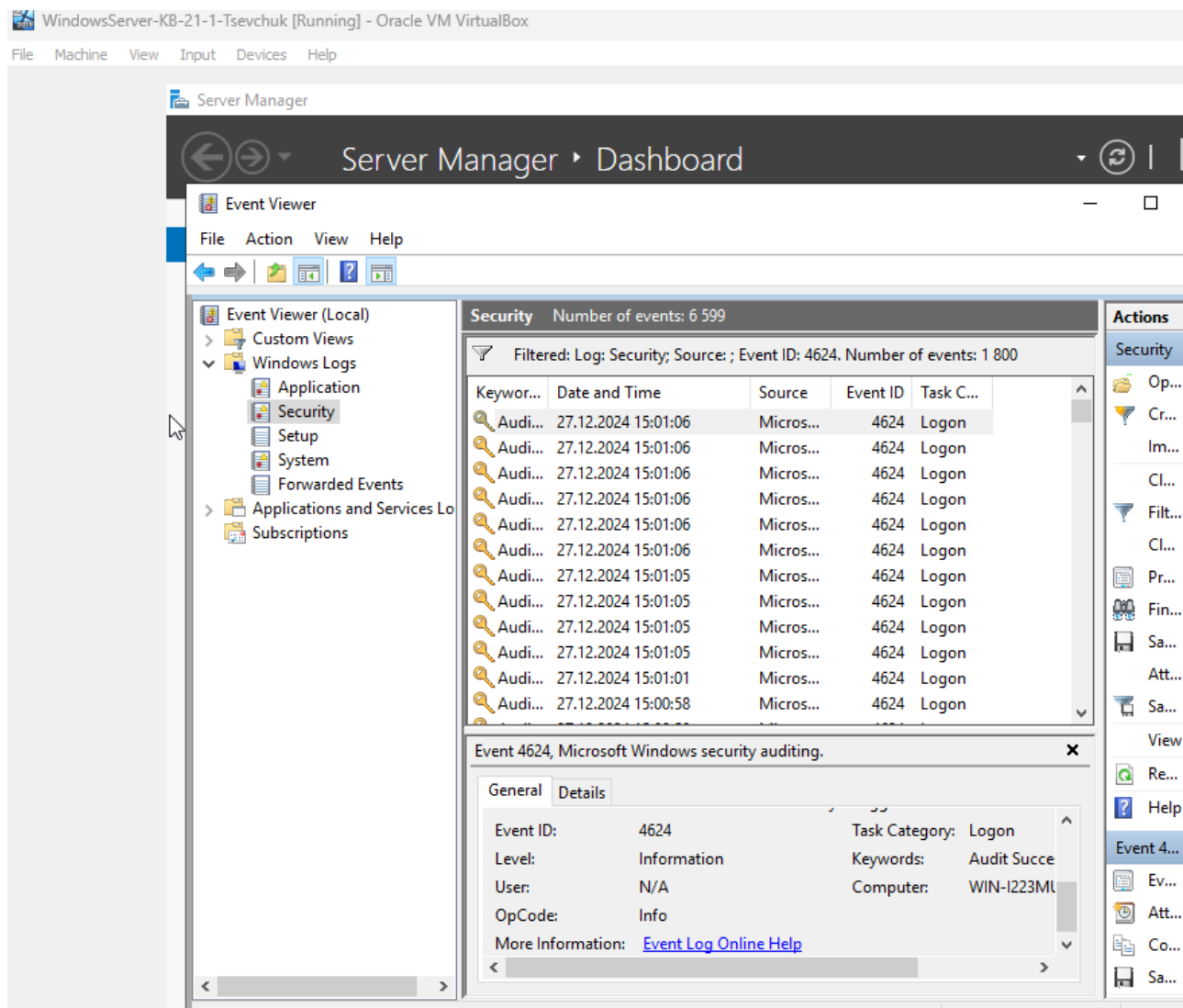


Рисунок 3.54 – Перевірка аудиту входу виходу

Висновки до третього розділу

Було детально описано практичне розгортання та перевірку Honeypot-системи в корпоративному середовищі. Як основний інструмент обрано емулятор Cowrie, що дав змогу створити контрольоване вразливе середовище для фіксації шкідливої активності та Honeypotd для імітації веб-сервера. Honeypot інтегровано у віртуальну інфраструктуру, сформовану засобами Oracle VirtualBox із використанням NAT-мережі. Конфігурація охопила налаштування статичних IP-адрес, правила перенаправлення портів і ізоляцію внутрішніх ресурсів.

Паралельно було розгорнуто Ubuntu-сервер, що має два мережеві інтерфейси – публічний і приватний. Він виконує функції шлюзу та фаєрвола, забезпечує port

forwarding до Honeypot та інших внутрішніх вузлів, а також реалізує механізм блокування IP-адрес атакуючих джерел через iptables. Це дозволяє забезпечити надійне логічне розділення зовнішньої і внутрішньої мережі без потреби у фізичних ізоляторах.

Також у межах інфраструктури було встановлено Active Directory на Windows Server, створено домен та приєднано клієнтську станцію. Налаштовано політики безпеки – зокрема, вимоги до складності паролів, блокування облікового запису після кількох невдалих спроб входу, а також аудит подій автентифікації користувачів.

Реалізована архітектура демонструє ефективний підхід до захисту публічних портів корпоративної мережі на базі MS Active Directory з інтеграцією засобів активної оборони.

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		60

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було комплексно досліджено можливості застосування Honeypot-систем для захисту публічних портів у корпоративних мережах. Розглянуто їхнє призначення, класифікацію, переваги та обмеження, а також особливості інтеграції в інфраструктуру, побудовану на базі Microsoft Active Directory.

На основі отриманих теоретичних положень було спроектовано та реалізовано модель корпоративної мережі із віртуалізованим середовищем, логічним розділенням на зони та засобами активного захисту. Для цього використано емулятори Cowrie та Honeyhttpd, що забезпечили моніторинг несанкціонованої активності на портах SSH, Telnet, HTTP і HTTPS.

Практична реалізація охопила налаштування мережевої інфраструктури, Ubuntu-сервера, інструментів маршрутизації та фільтрації трафіку, а також засобів адміністрування домену в середовищі Windows Server. Проведене тестування підтвердило ефективність обраного рішення для виявлення і блокування потенційних загроз.

Результати роботи демонструють доцільність використання Honeypot як компонента активної оборони в корпоративному середовищі та підтверджують його роль як інструмента аналітики і підвищення безпеки відкритих сервісів.

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		61

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. What is a honeypot? How honeypots help security. /. URL: <https://www.kaspersky.com/resource-center/threats/what-is-a-honeypot>.
2. What is a honeypot and how does it work?. Norton AntiVirus, Privacy, Identity, & Advanced Scam Protection. URL: <https://us.norton.com/blog/iot/what-is-a-honeypot>.
3. Honeypot Basics. URL: <https://www.rapid7.com/fundamentals/honeypots/>.
4. What is an enterprise network?. Cisco. URL: [https://www.cisco.com/site/us/en/learn/topics/networking/what-is-an-enterprise-network.html#:~:text="Enterprise%20network"%20denotes%20the%20IT,users,%20devices,%20and%20applications](https://www.cisco.com/site/us/en/learn/topics/networking/what-is-an-enterprise-network.html#:~:text=).
5. What is honeypot? Definition & uses. SentinelOne. URL: <https://www.sentinelone.com/cybersecurity-101/threat-intelligence/honeypot-cyber-security>.
6. Journal D. What are honeypots: everything you need to know. Medium. URL: <https://medium.com/@datajournal/what-are-honeypots-f3a4324d2776>.
7. What is a honeypot? Meaning, types, benefits, and more | fortinet. Fortinet. URL: <https://www.fortinet.com/resources/cyberglossary/what-is-honeypot#:~:text=There%20are%20two%20primary%20kinds,the%20same%20kinds%20of%20services>.
8. Why a honeypot is not a comprehensive security solution. International Journal of Computer Networks And Applications (IJCNA). URL: <https://ijcna.org/Manuscripts/Volume-2/Issue-5/Vol-2-issue-5-M-01.pdf>.
9. General information about Active Directory domain services. Microsoft Learn: Build skills that open doors in your career. URL: <https://learn.microsoft.com/ru-ru/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview>.
10. Open ports and their vulnerabilities. Specops Software. URL: <https://specopssoft.com/blog/open-ports-and-their-vulnerabilities/>.

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		62

11. Public-Facing infrastructure – threatng security - external attack surface management (EASM) - digital risk protection - security ratings. ThreatNG Security. URL: <https://www.threatngsecurity.com/glossary/public-facing-infrastructure>.
12. Gillis A. S., Loshin P., Cobb M. What is SSH (secure shell) and how does it work? | techtarget. Search Security. URL: <https://www.techtarget.com/searchsecurity/definition/Secure-Shell>.
13. Overview of HTTP - HTTP | MDN. MDN Web Docs. URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Guides/Overview>.
14. HTTP и HTTPS: what it is and what the difference is | HOSTiQ Wiki. HOSTiQ Wiki. URL: <https://hostiq.ua/wiki/http-https/>.
15. Port forwarding в VirtualBox. evlanoff. URL: <https://evlanoff.wordpress.com/2017/11/23/port-forwarding-in-virtualbox/>.
16. GitHub - cowrie/cowrie: cowrie ssh/telnet honeypot. GitHub. URL: <https://github.com/cowrie/cowrie>.
17. Installing Cowrie in seven steps – cowrie 2.6.1 documentation. Cowrie’s documentation! – cowrie 2.6.1 documentation. URL: <https://docs.cowrie.org/en/latest/INSTALL.html>.
18. GitHub - bocajsppear1/honeyhttpd: HoneyHTTPD - Python-based web server honeypot. GitHub. URL: <https://github.com/bocajsppear1/honeyhttpd>.
19. Honeypots: tracking hackers. Boston : Addison-Wesley, 2002. 452 с.
20. Blue team handbook: incident response edition : a condensed field guide for the cyber security incident responder. 2-ге вид. CreateSpace Independent Publishing Platform, 2014. 146 с.

Додаток

		Цевчук В.С			ККІКБ.КР.Б – 125 – 19 – ПЗ	Арк.
		Воротніков В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		64

Проект Noneurot-підсистеми побудованої мережі

