Міністерство освіти і науки України
Харківського національного університету імені В.Н. Каразіна
Навчально-наукового інституту комп’ютерних наук та штучного інтелекту
Спеціальність 125 «Кібербезпека»
Освітня програма «Кібербезпека»

В.о. зав. кафедрою КІСМіТ
Марина ЄСІНА
“Допущено до захисту”
« »_____2025р.

Пояснювальна записка

до кваліфікаційної роботи бакалавра

на тему: «Оцінка захищеності мережі та мережевого з'єднання»

оцінка « _____ »

Голова ЕК
Мичуда Л.З.

Керівник: к.т.н. Мелкозьорова О.М.

Рецензент: д.т.н. Толстолузька О.Г.

Виконавець: студент групи КБ-42

Кравченко Є.В.

Харків 2025

РЕФЕРАТ

Пояснювальна записка до проєкту бакалавра містить 59 сторінок, 11 рисунків, 4 таблиці, 3 додатки, 29 посилань на джерела.

Мета роботи полягає в дослідження методів та інструментів для оцінки захищеності мережі та мережевого з'єднання, а також у практичній перевірці стану інформаційної безпеки за допомогою сучасних інструментів на прикладі тестової моделі локальної комп'ютерної мережі.

Об'єкт дослідження – локальна комп'ютерна мережа як цілісна система передачі обробки та захисту інформації.

Предмет дослідження – основні інструменти та методи оцінки захищеності комп'ютерної мережі, проблематика забезпечення безпеки типових мережевих топологіях.

Основними методами дослідження є аналіз та порівняння сучасних інструментів та методів оцінки захищеності мережі та мережевого з'єднання. Виявлення вразливостей в тестовій мережі, перевірка можливості їх експлуатації та моніторинг трафіку з метою відстеження потенційно небезпечної активності.

У роботі досліджено: сучасні тенденції розвитку у сфері кібербезпеки мереж, найбільш поширені види кібератак на мережеву інфраструктуру, основні критерії захищеності мережі, методи та інструменти для оцінки стану безпеки комп'ютерних мереж, а також проведено практичне тестування моделі локальної мережі з метою виявлення вразливостей, оцінки ризиків та наслідків, формування рекомендацій щодо підвищення рівня її захищеності відповідно до сучасних стандартів.

Результати роботи можуть бути використані у наукових виданнях та/або для кращого розуміння принципів побудови захищених комп'ютерних мереж.

Ключові слова: КІБЕРБЕЗПЕКА, КОМП'ЮТЕРНА МЕРЕЖА, ЗАХИЩЕНІСТЬ МЕРЕЖІ, ТЕСТУВАННЯ НА ПРОНИКНЕННЯ, ВРАЗЛИВОСТІ, МОНІТОРИНГ ТРАФІКУ, АТАКА MITM, DOS-АТАКА, СКАНУВАННЯ ВРАЗЛИВОСТЕЙ, ОЦІНКА РИЗИКІВ, ІНФОРМАЦІЙНА БЕЗПЕКА.

ABSTRACT

The explanatory note to the bachelor's project contains 59 pages, 11 figures, 4 tables, 3 appendices, and 29 references.

The aim of the work is to explore methods and tools for assessing the security of a network and its connections, as well as to conduct a practical evaluation of the information security status using modern tools based on a test model of a local computer network.

Object of the study – a local computer network as an integrated system for information transmission, processing, and protection.

Subject of the study – the main tools and methods for assessing network security, and the challenges of ensuring security in typical network topologies.

The main research methods include the analysis and comparison of modern tools and techniques for assessing network and connection security, identifying vulnerabilities in a test network, testing their exploitability, and monitoring traffic to detect potentially malicious activity.

The work investigates current trends in the field of network cybersecurity, the most common types of cyberattacks targeting network infrastructure, key criteria for network protection, methods and tools for evaluating network security, as well as practical testing of a local network model aimed at identifying vulnerabilities, assessing risks and consequences, and formulating recommendations for improving its security in accordance with modern standards.

The result of the work can be used in scientific publications and/or to better understand the principles of building secure computer networks.

Keywords: CYBERSECURITY, COMPUTER NETWORK, NETWORK SECURITY, PENETRATION TESTING, VULNERABILITIES, TRAFFIC MONITORING, MITM ATTACK, DOS-ATTACK, VULNERABILITY SCANNING, PISK ASSESSMENT, INFORMATION SECURITY.

ЗМІСТ

ПЕРЕЛІК ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

DDoS – Distributed Denial of Service

ПЗ – Програмне забезпечення

AI – Artificial Intelligence

ML – Machine Learning

NLP – Natural Language Processing

IoT – Internet of Things

ZTA – Zero Trust Architecture

MITM – Man In The Middle

SQL – Structured Query Language

URL – Uniform Resource Locator

DNS – Domain Name System

IP – Internet Protocol

CVSS – Common Vulnerabilities Scoring System

IDS – Intrusion Detection System

IPS – Intrusion Prevention System

VPN – Virtual Private Network

HTTP – HyperText Transfer Protocol

FTP – File Transfer Protocol

GUI – Graphical User Interface

ВСТУП

Наш сучасний світ важко уявити без використання комп’ютерних мереж та Інтернету, ці технології стали невід’ємною частиною нашого побуту. Початок розвитку мережевих технологій припав на середину XX століття, саме в цей час вчені всього світу почали замислюватися над спрощенням процесу обміну інформацією між тогочасними обчислювальними пристроями. Переломним моментом в цій галузі стало винайдення ARPANET у 1969 році, який ліг в основі сучасного Інтернету, яким користується майже кожна людина в світі. З розвитком технологій та зростаючою кількістю користувачів мережі виникла необхідність в розробці все більш ефективних та зручних методів передачі інформації між користувачами та забезпеченні безпеки інформації, що передавалася.

З розвитком комп’ютерних мереж та мережевого з’єднання почали траплятися випадки несанкціонованого доступу до даних, які передавалися через мережу. Починаючи з 80-х років XX століття було зареєстровано перші випадки кібер злочинів проти інформаційної безпеки даних такі як, хакерські атаки та використання вірусних програм. Так першим, серйозним, викликом став інцидент у 1989 році, який пізніше назвали «черв'яком Морріса», він виявив вразливість тогочасних мережевих технологій, що дало новий поштовх для їх розвитку в майбутньому та показало важливість забезпечення кібербезпеки в мережах. З часом масштаби кіберзлочинності тільки зростали, починаючи з простих атак на особисті пристрої користувачів такі як персональні комп’ютери, закінчуючи масштабними атаками на великі державні установи, банківські системи та системи держбезпеки по всьому світу. В сучасних реаліях захищеність мереж та мережевого з’єднання стоїть на першому місці у всіх сферах діяльності, з цим пов’язаних, починаючи від малого та великого бізнесу і закінчуючи національною безпекою найрозвинутіших країн світу.

З розвитком інформаційних технологій та зростаючою кількістю випадків кіберзлочинів необхідно відповідально підходити до аналізу захищеності мереж відповідно сучасних критеріїв безпеки, виявлення можливих вразливостей,

попереджувати можливі хакерські атаки. Саме з цією метою було створено безліч різних методів та інструментів, які дозволяють ефективно виявляти проблеми безпеки в мережах, оцінювати пропускну здатність мережі, імітувати новітні методи атак на мережу, для перевірки стійкості, та надають можливість оперативно реагувати на подібні виклики. Саме цій проблемі присвячена дана дипломна робота, в межах якої буде здійснено: Аналіз та дослідження методів та інструментів для оцінки захищеності мережі та мережевого з'єднання, перевірка, на практиці, інформаційної безпеки мережі за допомогою сучасних інструментів для роботи з мережами та мережевим з’єднанням. Основною метою роботи буде виявлення вразливостей в сучасній мережевій інфраструктурі, оцінка ризиків та наслідків з ними пов’язаних, надання рекомендацій щодо їх усунення та покращення захищеності мережі в цілому.

Саме тому дана дипломна робота є актуальною з точки зору сучасної проблематики безпеки в сфері мережевих технологій, оскільки вона дозволяє оцінити сучасний стан безпеки комп’ютерних мереж та сприяє підвищенню загального рівня розуміння сучасних інструментів та методів для оцінки захищеності мережі та мережевого з’єднання. Результати досліджень, що будуть проведені в ході виконання роботи можуть бути використані як і в навчальних цілях, так і для практичного застосування в галузі кібербезпеки, а саме в сфері мережевої безпеки.

1 ТЕНДЕНЦІЇ РОЗВИТКУ КІБЕРЗАХИСТУ МЕРЕЖ, ДОСЛІДЖЕННЯ ПРОБЛЕМАТИКИ БЕЗПЕКИ МЕРЕЖЕВИХ З'ЄДНАНЬ ТА МЕТОДІВ ОЦІНКИ ЗАХИЩЕНОСТІ МЕРЕЖІ

1.1 Виклики інформаційній безпеці в мережах у сучасному світі

Кіберзагрози в сучасному світі є все більш поширеною і масштабною проблемою, вони охоплюють широкий спектр ризиків та вразливостей, що потенційно можуть нанести велику шкоду безпеці цифрових систем, мереж та даних. Загалом вони можуть бути у вигляді різного шкідливого програмного забезпечення, фішинг програм, програм-вимагачів та ще багато інших видів зловмисних атак. Щоб захистити сучасні системи, дані та користувачів від цих загроз необхідністю є використання новітніх методів кіберзахисту таких як, міжмережеві екрани, антивірусні програмні забезпечення, регулярне оновлення системи, тощо [1].

Згідно аналітичними дослідженнями (рис. 1.1) за останні роки серед найбільш актуальних та розповсюджених кіберзагроз у сучасному світі виокремлюють наступні: Програми-вимагачі (Ransomware), DDoS-атаки, порушення безпеки даних (Data), шкідливе ПЗ (Malware), соціальна інженерія (Social engineering), маніпуляція інформацією (Information manipulation), веб-загрози (Web threats), атаки на ланцюги постачання (Supply chain attack) [2].

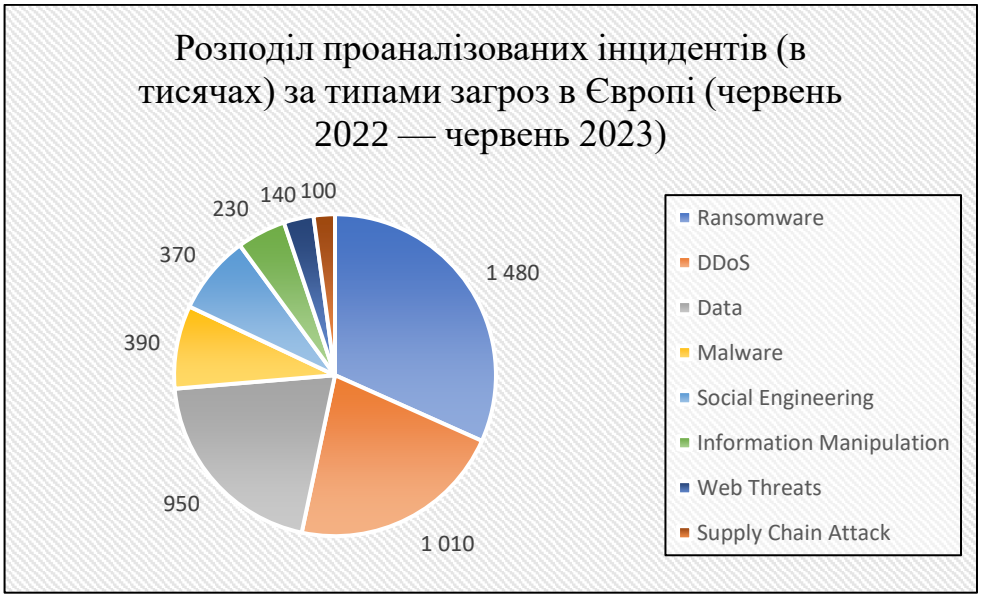


Рисунок 1.1 – Аналітичні дані про кіберінциденти

Однією з найпоширеніших кіберзагроз є DDoS-атаки (Distributed Denial of Service), коли велика кількість запитів спрямовується від пристроїв зловмисників (зламаних пристроїв) напряму до певного сервера через що виникає перевантаження, роблячи сервер недоступним для користувачів. Найчастіше жертвами таких атак стають веб-сайти мережі та онлайн-сервіси, що порушує їхню дієздатність [1].

Ще однією розповсюдженою загрозою є фішинг (один із типів соціальної інженерії), коли зловмисники видають себе за офіційні організації та обманним шляхом змушують людей розкривати конфіденційну інформацію (паролі, дані банківських карток, тощо). Фішингові атаки зазвичай здійснюються через електронні листи, повідомлення, підроблені веб-сайти, тощо [1].

Шкідливе програмне забезпечення (malware) все ще є викликом для сучасної кібербезпеки, воно створене для проникнення в систему, може використовувати інструменти для доступу та викрадення даних або їх знищення з метою заподіяння шкоди компанії та користувачам. Прикладами шкідливого ПЗ є програми-вимагачі, віруси, трояни, всі вони створені з ціллю викрадення інформації або порушення цілісності та дієздатності інформаційних систем [1].

На фоні розвитку інформаційних технологій та їх популяризації серед суспільства, вплив «людського фактору» на безпеку та цілісність даних в мережі стають все більш актуальним і не завжди мова йде про їх забезпечення, а і про підрив інформаційної безпеки [3]. Вплив людського фактору можна умовно поділити на такі дві групи:

- Фактори ризику пов’язані з користувачами: відсутність відповідних знань про роботу з інформацією, недооцінення потенційних загроз безпеці, неправильне використання інформаційних технологій, що може призводити до підриву дієздатності інформаційної системи.
- Фактори ризику пов’язані з управлінням: недостатньо навчений персонал, брак фінансів на забезпечення відповідної безпеки даних, недостатня кількість кваліфікованих спеціалістів у сфері кібербезпеки.

Найбільш вагомими наслідками зловмисних дій і негативного впливу «людського фактору» можуть бути фінансові втрати компанії, витік конфіденційної інформації, втрати репутаційні, всі ці фактори є дуже вагомими і впливають не лише на компанії, які постраждали від кібератак, а і на самих користувачів, які користувалися їх послугами [3].

Найбільшою проблемою є те, що кіберзлочинці постійно вдосконалюють свої методи атак та технологічні засоби, що створює постійну небезпеку навіть для найпрогресивніших засобів захисту інформаційних систем. Тому забезпечення кібербезпеки – це не лише просто питання захисту даних чи збереження фінансів, а стратегічний виклик, який стосується всіх рівнів суспільства.

1.2 Тенденції розвитку кіберзахисту мереж та мережевого з’єднання

Розвиток інформаційних технологій у сучасному світі призводить до виникнення складніших систем та мереж, збільшується кількість користувачів мережі інтернет та компаній, які віддають перевагу діджиталізації, все це супроводжується і розвитком методів та інструментів для забезпечення кіберзахисту більших та складніших систем, більшого обсягу інформації. Провідними технологія у сфері кібербезпеки мереж та загалом, спираючись на сучасні тенденції розвитку цієї галузі, є:

- Використання штучного інтелекту (AI) та машинного навчання (ML);
- Впровадження архітектурної моделі Zero Trust («не довіряй нікому»);
- Використання блокчейн-технологій.

1.2.1 Вплив розвитку технологій штучного інтелекту (AI) на вдосконалення засобів кіберзахисту

У новітньому цифровому світі все більшої популярності набуває використання штучного інтелекту (AI) та машинного навчання (ML), які стали цінними інструментами в боротьбі з сучасними кіберзагрозами. Дослідження фахівців у сфері штучного інтелекту змогли досягти великих успіхів у створенні технологій для виявлення та реагування на атаки. AI дозволяє виконувати задачі, які на сам перед потребують людського інтелекту таких як, ухвалення рішень,

розпізнавання мови, аналіз зображень, тощо. Так штучний інтелект використовує навчальні дані для оцінки можливих загроз та створення методів виявлення та реагування на них. Основною технологією AI в сфері кібербезпеки є машинне навчання, глибоке навчання (Deep leaerning), обробка природної мови (NLP). Використання цих методів дозволяє створювати складні та ефективні автоматизовані системи захисту, які будуть автоматично виявляти нові загрози та використовувати ефективні методи реагування на них, в залежності від типу загрози, оцінки потенційних збитків, тощо [4].

Штучний інтелект на відміну від людського може автоматично аналізувати великий обсяг інформації, таких як мережевий трафік, може виявляти аномалії поведінки користувачів та підозрілі дії в режимі реального часу. Використання таких технологій дозволяє системам кіберзахисту «вчитися» на основі вже існуючих даних про мережеву активність, аналізувати шаблони поведінки користувачів та зловмисників, виявляти вразливості мережі ще до моменту атаки, це дозволяє системі швидко адаптуватися до нових загроз та забезпечити високий рівень захищеності даних в мережі [5].

Інструменти, які вже почали використовувати AI та ML для створення найбільш зручних та ефективних рішень:

- Захист кінцевих точок з використанням штучного інтелекту – завдяки AI ці системи можуть виявляти та усувати загрози на рівні кінцевих пристроїв (персональні комп’ютери, смартфони, тощо).
- Міжмережеві екрани з використанням AI-модулів – нове покоління міжмережевих екранів, що використовує штучний інтелект для швидшого виявлення та швидкої адаптації до нових загроз.
- SIEM інструменти з підтримкою AI – має змогу оброблювати велику кількість стастичної інформації (логів), що дозволяє швидко виявити аномальну поведінку в мережі [5].

1.2.2 Концепція «Zero Trust» як сучасний аналог традиційній моделі безпеки

Ще одним не менш перспективним напрямком розвитку кібербезпеки мереж та мережевого з’єднання є перехід від традиційної моделі безпеки до концепції Zero

Trust («Не довіряй нікому»). В минулому мережева безпека будувалася виключно навколо захисту від зовнішніх загроз, тобто основна увага приділялася контролю доступу ззовні, що створювало ситуації, коли зловмисник, проникнувши всередину системи, мав доступ до внутрішніх корпоративних ресурсів. Збільшенню значимості цієї проблеми сприяв розвиток хмарних технологій, IoT (Internet of Things), робота працівників у режимі онлайн, саме це стало поштовхом до створення новітніх підходів до внутрішньої безпеки мереж та інформаційних систем загалом [6].

Zero Trust Architecture (ZTA) – сучасний підхід до безпеки, який спрямований на захист конкретних ресурсів всередині системи. Ця концепція передбачає, що жоден користувач або пристрій не зможе виконувати дії всередині системи без попередньої автентифікації, незалежно від місця їх розташування та характеру самої дії. Такий підхід дозволяє бути впевненим, що навіть за умови проникнення зловмисника всередину системи, він не матиме змогу отримати доступ до всіх корпоративних ресурсів компанії. Загалом Zero Trust базується на трьох основних принципах [7]:

- Моніторинг та валідація – постійний моніторинг використовуваних ресурсів для виявлення аномальної поведінки, перевірка справжності користувачів шляхом автентифікації та авторизації на основі існуючих даних (місцезнаходження користувача або пристрою, особи користувача, класифікації використовуваних даних, тощо).
- Принцип мінімальних привілеїв – це принцип, що обмежує права доступу до ресурсів системи (даних, застосунків, серверів) користувачів залежно від функцій, які вони повинні виконувати. Реалізується даний принцип за допомогою механізмів доступу Just-in-Time («вчасно») та Just-Enough access («достатньо»), такий підхід дозволяє мінімізувати можливі ризики шкоди від внутрішніх загроз.
- Припущення про можливу компрометацію – принцип за яким система вже передбачає наявність загрози всередині мережі та має забезпечувати постійну перевірку доступів та активність всередині мережі.

1.2.3 Впровадження технологій блокчейн для забезпечення безпеки даних в мережі

Технологія блокчейн стала однією с провідних в світі інформаційних технологій в останні декілька років. Інтеграція блокчейн-технологій в сферу кібербезпеки дозволяє забезпечити ряд ключових аспектів безпеки для захищення даних, що передаються мережею та усунути вразливості традиційних систем безпеки [8]:

- Забезпечення цілісності даних – технологія блокчейн забезпечує незмінність даних, що робить неможливим модифікацію інформації, що передається.
- Принцип децентралізації – відсутність єдиної точки відмови дозволяє зробити систему більш стійкою до атак зловмисників.
- Розширене шифрування – блокчейн використовує криптографічні методи шифрування, що дозволяє гарантувати доступ до інформації лише для авторизованих користувачів.

Прикладами впровадження блокчейн-технологій для забезпечення безпеки мережі та мережевого з'єднання є (рис. 1.2):

Hyperledger – проєкт з відкритим вихідним кодом, що використовує блокчейн-фреймворки та модули для безпеки мережі, таких як Hyperledger Fabric з підтримкою смартконтрактів та приватних каналів; Hyperledger Indy – підтримує концепцію самоврядованої ідентичності та перевірюваних облікових даних; Hyperledger Sawtooth – використовує механізм консенсусу proof-of-elapsed-time (доказ витраченого часу) [9].

Ethereum – публічна блокчейн платформа, що підтримує смартконтракти та децентралізовані застосунки, забезпечуючи прозоре та безпечне виконання мережевої логіки та правил, а також розробку різних рішень для мережевої безпеки.

IOTA – технологія розподіленого реєстру, яка використовує структуру даних Tangle, забезпечуючи швидку та безкомісійну передачу даних в мережі, аналогічно з Ethereum надає інструменти для розробки та розгортання рішень для безпеки мережі [9].

Рисунок 1.2 – Приклади використання блокчейн-технологій в кібербезпеці мереж

З швидким розвитком хмарних технологій, технологій IoT та збільшенням обсягів інформації, що передається мережею, виникла і необхідність створення нових методів захисту цих даних від зловмисників, забезпечення їх цілісності, доступності та конфіденційності. Використання штучного інтелекту, перехід до архітектурної моделі Zero Trust та впровадження інструментів з використання технологій блокчейн дозволяє підвищити рівень захисту від актуальних загроз безпеки, спростити та зробити ефективнішими інструменти виявлення та реагування на загрози. У сучасному світі використання таких технологій є обов’язковим кроком для забезпечення безпеки мереж та мережевого з’єднання.

1.3 Розвиток кіберзлочинності, поширені методи атак на мережі

Кібезлочинність одна із найшвидше зростаючих загроз у сучасному цифровому просторі, з поширенням використання інтернету у суспільстві, частота та складність кібератак на мережі невпинно зростає. Аналіз статистики кіберзлочинів дозволяє побачити, що в найближчому майбутньому очікується збільшення кібератак у всьому світі. За останні роки збитки, завдані кіберзлочинами, зросли в декілька разів (від 3 трильйонів доларів за кілька років тому, до десятків трильйонів доларів у минулому році). Таке стрімке зростання серйозно впливає на економіку окремих громадян, компаній та навіть цілих держав.

Значна частина таких злочинів залишається нерозкритою, оскільки жертви бояться репутаційних втрат або можливих фінансових наслідків, що ускладнює реальну оцінку масштабів проблеми [10].

Аналізуючи частоту кібератак на мережі типу DDoS, на основі даних NETSCOUT, Cloudflare та StormWall (рис. 1.3) можна побачити, що атаки на саме мережеву безпеку стають все більш популярними та завдають збитків не тільки окремим користувачам, а і великим компаніям державного рівня. Розвиток нових технологій не тільки сприяє покращенню захищеності мереж та інших інформаційних систем, а і слугує кіберзлочинцям інструментом для розробки нових методів атак, дозволяючи їм обходити вже існуючи системи безпеки. Все частіше можна побачити роботу не лише окремих хакерів, а цілих угрупувань, які використовують ці технології для масових атак. Тому аналіз сучасних методів атак на мережі є вкрай необхідним для забезпечення мережевої безпеки та запобігання потенційних кіберзлочинів [2].

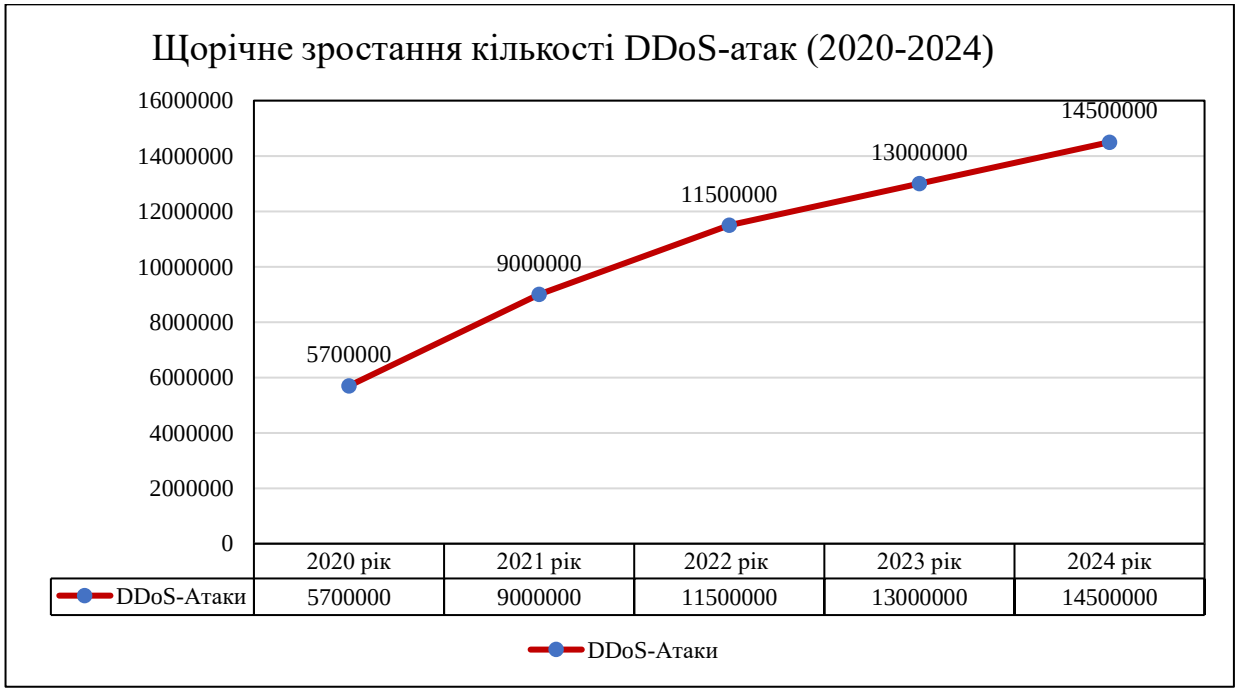


Рисунок 1.3 – Графік зростання кількості DDoS-атак

1.3.1 Найбільш поширені типи кібератак на мережу в сучасному світі та наслідки їх відтворення для компаній та користувачів

Серед найбільш поширених атак на мережі, у сучасних реаліях, можна відзначити наступні:

1) DoS та DdoS атаки – атаки типу відмова в обслуговуванні (Denial-of-Service/DoS), які спрямовані на перевантаження системи великою кількістю запитів до моменту неможливості відповіді на запити легітимних користувачів. Розподілена атака відмови в обслуговуванні (Distributed Denial-of-Service/DdoS) має аналогічну мету перевантаження ресурсів системи, але сама атака здійснюється через велику мережу, заражених шкідливим програмним забезпеченням, пристроїв, що збільшує її ефективність. Свою назву вони отримали через неможливість системи (найчастіше сайту) надавати свої послуги легітимним користувачам. Такі атаки відрізняються від інших тим, що у низ на меті не отримання або розширення доступу до системи, а саме порушення її роботи. DoS-атаки створювати вразливості для інших видів атак через порушення дієздатності системи [11].
2) Атака людина посередині (Man-in-the-Middle) – це тип атаки, коли атакуючий перехоплює та модифікує дані, що передаються між двома сторонами у мережі, в такому випадку зловмисник стає проміжною ланкою між відправником та отримувачем даних. При атаках типу MITM, користувачі вважає, що спілкується між собою та не мають змоги помітити наявність третьої сторони. Частіше всього до такого призводить користування незахищеними публічними мережами в громадських місцях [11].
3) SQL-ін’єкція – поширений метод зламу сайтів, що використовує вразливості баз даних. SQL-атака відбувається шляхом надсилання SQL-запиту від імені користувача до бази даних на сервері сайту, зловмисник використовує вразливості опрацювання таких команд, що дозволяє йому, використовуючи спеціальні команди, давати команди серверу такі як: видалення з БД таблиць з даними, копіювання їх на сторонні ресурси, зміна облікових записів, тощо. Такі атаки часто призводять до витоку конфіденційної інформації, порушення доступності та/або цілісності інформації, порушення роботи всієї системи [11].

4) Інтерпретація URL (або URL poisoning) – це зміна існуючих або створення нових URL-адрес для отримання доступу до конфіденційної інформації. Основний принцип здійснення даної атаки це спроби відгадати/згенерувати правильний формат URL, основною метою цих дій є отримання доступу до внутрішньої частини сайту або адміністративних прав. Найпростіший приклад це додавання до URL сайту /admin, що може перенести зловмисника на сторінку адміністратора і за умови, що стандарті пароль і логін (admin/ admin) не були змінені, він зможу легко отримати повні права доступу до ресурсу [11].
5) DNS-спуфінг – зловмисник змінює записи DNS, перенаправляючи трафік на підроблений сайт, при цьому жертва вважає, що знаходиться на легітимному сайті та вводить свою конфіденційну інформацію, яка потім потрапляє до зловмисника. Частіше всього для таких атак використовуються назви відомих сайтів для скоєння шахрайських дій, що призводить не лише до витоку даних користувачів, а і до репутаційних втрат конкретних компаній.
6) Атаки XSS (Cross-Site Scripting) – це тип атаки, коли зловмисник передає шкідливий сценарій (script) через клікабельний (доступний для натискання користувачем) контент та потім передається та виконується в браузері жертви. Небезпека цієї атаки полягає в тому, що саму дію на сайті виконує легітимний користувач, тому система захисту може пропустити таку загрозу, що призводить до виконання шкідливих дій на серверній частині сайту від імені самого користувача. Наприклад при зміні платіжних даних та інших параметрів (суми переказу) у веб-додатку банку, грошовий переказ може бути виконаний не на рахунок легітимного користувача, а на рахунок зловмисника [11].

1.3.2 Наймасштабніші атаки останніх років, націлені на мережеву безпеку

За останній роки, зі збільшенням кількості кіберзлочинів, збільшилися масштаби кібератак, вони все частіше порушують роботу не тільки однієї компанії, а і цілих корпорацій та мереж державних установ. Серед найбільш серйозних

кібератак останніх років можна виділити: SolarWinds hack (2020), Log4Shell (2021), Mirai Botnet (2016).

Атака SolarWinds (2020) (рисунок 1.4) – це загальноприйнятий термін, що характеризує атаки на ланцюги постачання пов'язаних з системою SolarWinds Orion. Головними підозрюваними, за версію Microsoft, є група хакерів Nobelium, вони отримали доступ до мереж, корпоративних даних та конфіденційної інформації тисяч користувачів SolarWinds. Ця атака вважається однією з найбільш масштабних в своєму роді. Більше декількох десятків тисяч державних і приватних компаній, які використовували систему управління мережею Orion, постраждало від цієї атаки. Причиною атаки стало ненавмисне встановлення, компанією SolarWinds, програмного забезпечення-оновлення для системи Orion, в якому хакери Nobelium заклали backdoor (вразливість в програмному коді) для обходу міжмережевих екранів, що дозволило їм отримати контроль над серверами та даними. Оскільки сама атака відкривала доступ не лише до корпоративних даних компанії SolarWinds, а і до внутрішніх робіт користувачів, доступ до даних та мереж їх клієнтів та партнерів, кількість постраждалих від атаки експоненційно зростала [12].

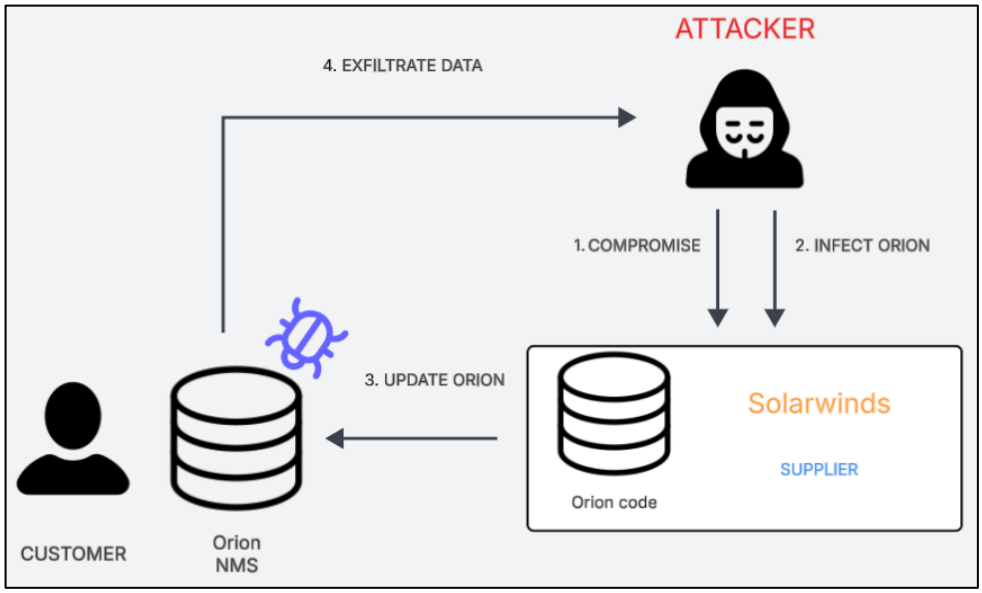


Рисунок 1.4 – Приклад схеми атаки на систему SolarWinds

Log4Shell (2021) (рисунок 1.5) – це назва програмної вразливості java-бібліотеки Apache Log4j 2, яка використовується для логування повідомлень про

помилки в застосунках. Ідентифікатор вразливості CVE-2021-44228, вона дозволяла зловмисникам отримати контроль над пристроєм на якому була встановлена певна версія бібліотеки Log4j 2. В день виявлення цієї вразливості, в 2021 році, мільйони користувачів мережі, в тому числі великі корпорації (Apple, Amazon, Cisco) постраждали від дій зловмисників, які за допомогою виконання певних текстових команд (лог-запитів) могли обходити системи безпеки пристроїв та виконувати шкідливі дії такі як викрадення конфіденційної інформації, порушення роботи застосунків, створення бекдорів в системах, тощо. Ця вразливість отримала CVSS-оцінку 10 з 10 – найвищій рівень критичності, через великий потенціал та масштаб простору її використання зловмисниками. Хоч після інциденту і було випущено декілька патчів для усунення проблеми, вона не була остаточно вирішена та збитки завдані нею ще досі оцінюються [14].

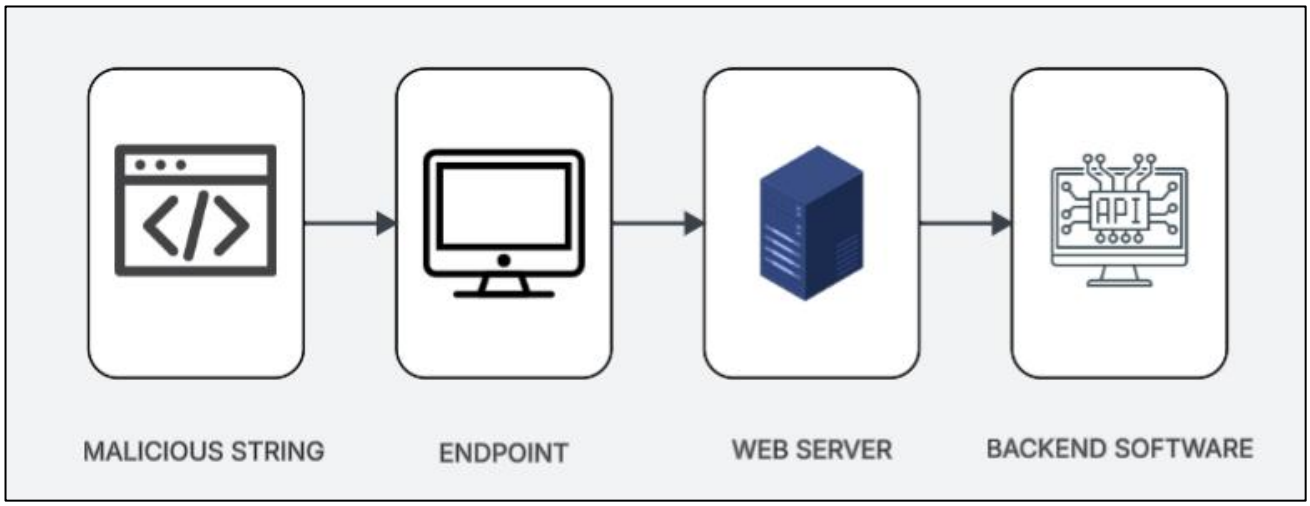


Рисунок 1.5 – Схема вразливості бібліотеки Log4j 2

Mirai Botnet (2016) (рисунок 1.6) – мережа заражених пристроїв створена Джозайєю Вайт (Josiah White), Парасом Джха (Paras Jha), Дальтоном Норманом (Dalton Norman), яка спочатку була написана мовою C для ботів та мовою Go для контролерів і спочатку був використаний для виводу з ладу конкуруючих серверів Minecraft методом атаки відмова в обслуговуванні (DDoS). В подальшому Mirai Botnet почав стрімко поширюватися, інфікуючи тисячі пристроїв Інтернету речей (IoT), набравши потужність для масштабніших кібератак. Перша задокументована атака Mirai відбувалася в вересні 2016 року на французьку компанію OVH та, за приблизними оцінками, використовувала понад сто сорок п’ять тисяч пристроїв.

Після здійснення ще декількох атак, вихідний код Mirai Botnet був злитий в мережу її засновником, що дозволило іншим зловмисникам використовувати його, додаючи нові модулі та методи для збільшення ефективності кібератак. Головною причиною виникнення такої великої мережі інфікованих пристроїв є вразливості безпеки та слабка захищеність в цілому IoT-пристроїв і подібних технологій, що дуже сильно ускладнило процес викорінення проблеми. Mirai Botnet і досі існує у видозмінених формах та використовується для масованих DDoS-атак на мережі, що призводить до великих збитків, не тільки для великих корпорацій, а для маленьких компаній та окремих користувачів [13].

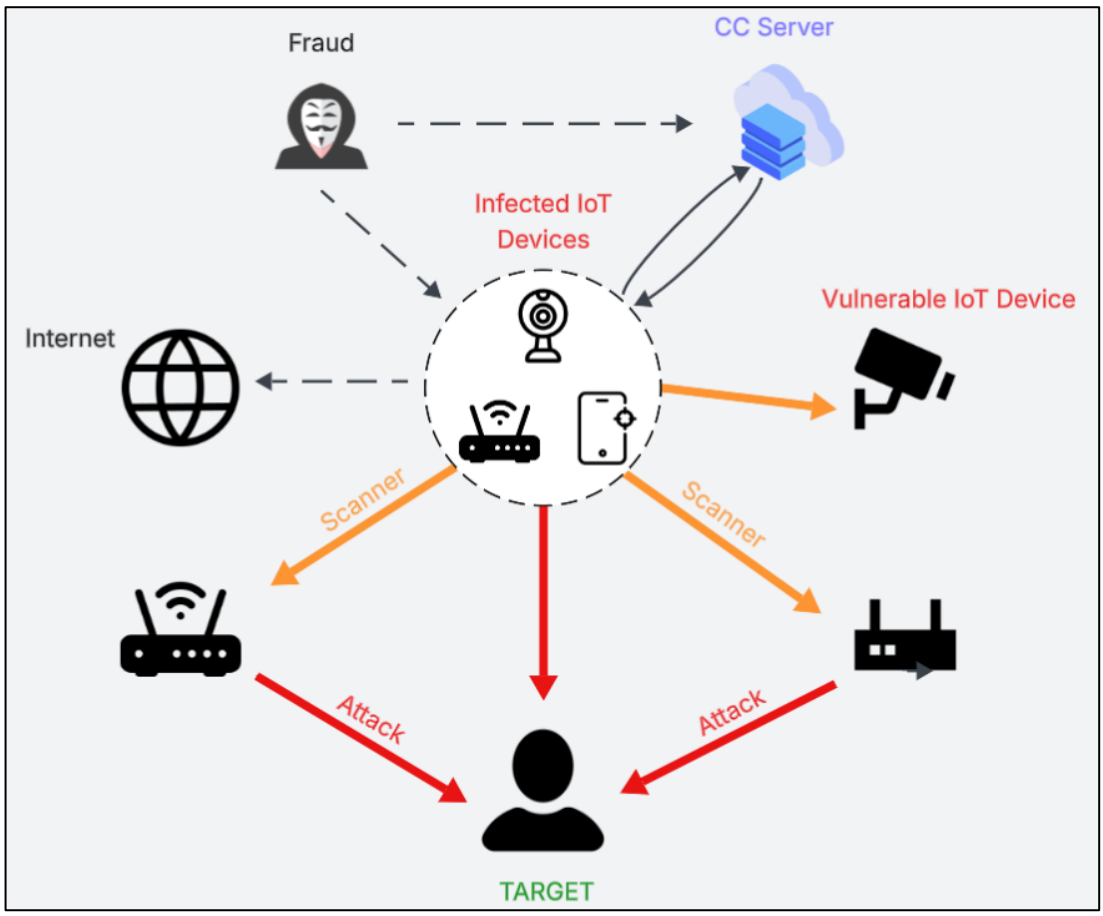


Рисунок 1.6 – Приклад схеми мережі Mirai Botnet

Незважаючи на розвиток сучасних технологій, використання штучного інтелекту та передових методів кіберзахисту, мережі та окремі пристрої не можуть залишатися повністю в безпеці. Кіберзлочинці продовжують розробляти нові види кібератак, використовуючи науковий прогрес у сфері інформаційних технологій, для задоволення власних зловмисних цілей. Аналіз злочинної активності,

інцидентів кібератак на безпеку систем і мереж, дотримання сучасних правил для забезпечення кібербезпеки інформаційних систем дозволяє уникнути як втрат, наприклад, конфіденційної інформації окремим користувачам, так і потенційних фінансових та репутаційних збитків великим та малим організаціям.

1.4 Сучасні стандарти інформаційної безпеки мереж

Для забезпечення достатнього рівня захищеності інформаційних систем, та мереж в тому числі, використовуються спеціальні стандарти безпеки, які вміщають в собі основні вимоги до безпеки та дозволяють стандартизувати всі основні методи та інструменти захисту. Дотримання стандартів безпеки дозволяє організаціям забезпечувати достатній рівень захищеності корпоративних даних та бути впевненими в спроможності реагувати на можливі кібератаки з мінімальними збитками.

Стандарти безпеки мережі – це встановлені протоколи, інструкції та практики, призначені для захисту мережі та її даних від несанкціонованого доступу, витоків і кіберзагроз. Ці стандарти кібербезпеки визначають, як організації повинні захищати свої системи та мережі для забезпечення відповідності нормативним вимогам [15].

Ігнорування, зазначених у стандартах, правил безпеки може призводити до значних проблем в захисті, репутаційним втратам, штрафам та навіть кримінальний відповідальності. Саме через це наявність чіткої політики безпеки є критично важливою, вона повинна встановлювати конкретні правила, процедури та стандарти для забезпечення цілісності, доступності та конфіденційності даних.

Серед ключових аспектів захищеності мереж, які описані в базових стандартах безпеки, можна віднести такі заходи як [15]:

- Віртуальна приватна мережа (VPN) – VPN використовується для захищення підключення між користувачем та мережею, шифруючи IP-адресу користувача. Використання цієї технології є критично важливим при роботі з віддаленими застосунками та роботі в публічній локальній мережі (наприклад публічний Wi-Fi).

- Брандмауери (Firewalls) – брандмауери або міжмережеві екрани використовуються як перша лінія захисту, контролюючи вхідний та вихідний потік трафіку, вони можуть блокувати спроби несанкціонованого доступу до мережі та пропускати лише легітимний трафік.
- Системи виявлення та запобігання вторгненням (IDPS) – це інструменти, що відстежують трафік, що проходить мережею, та виявляють підозрілу активність, блокуючи її тим самим запобігають можливим спробам несанкціонованого доступу або хакерським атакам [29]. Поділяються на два типи: IDS (Intrusion Detection System), що виявляє порушення безпеки, наприклад спробу несанкціонованого доступу; IPS (Intrusion Prevention System), використовуючи вбудовані інструкції реагування на загрози, блокує виявлені атаки до моменту нанесення ними шкоди системі.
- Контроль доступу (Access Control) – система, що визначає хто зможе отримати доступ до мережі та яку саме дію зможе виконати. Механізми роботи включають обмеження доступу до інформації тільки для користувачі, які мають право працювати з певним типом даних в мережі.

Загалом в сучасному світі, в залежності від країни та законодавства, використовується безліч різних стандартів мережевої безпеки, які описують свої конкретні аспекти захищеності системи (управління безпекою, практичні заходи безпеки, захист персональних даних, тощо). Серед найбільш важливих міжнародних стандартів безпеки мереж можна виокремити: ISO/IEC 27001, ISO/IEC 27033, PCI-DSS, IEEE 802.11.

Стандарт ISO 27001 встановлює вимоги до системи управління інформаційною безпекою (ISMS). Він допомагає виявляти ризики, впроваджує нові заходи безпеки, що дозволяє покращити захищеність мережі та забезпечити вищій рівень контролю над системою [15].

ISO/IEC 27033 містить найкращі практики для розробки правильної архітектури мережі, допомагає організаціям виявляти й усувати недоліки та вразливості системи, надає широкий спектр рекомендацій щодо захисту внутрішніх та зовнішніх мереж. Також він визначає інструменти, технології та

протоколи, які необхідно використовувати під час створення мережі, наприклад брандмауери, системи виявлення вторгнень, протоколи шифрування.

PCI-DSS – це стандарт безпеки розроблений для безпеки даних платіжних карт. Він є обов'язковим для організацій, що працюють з платіжними даними (наприклад банківські установи), встановлює вимоги для забезпечення захисту користувачів банківських карт та уникнення можливих маніпуляцій з боку шахраїв. Серед ключових вимог даного стандарту можна відзначити: шифрування даних, що передаються мережею; контроль доступу до інформації; регулярне тестування безпеки; журналювання активності в системи [15].

IEEE 802.11 – це стандарт безпеки, що встановлює вимоги щодо безпеки Wi-Fi мереж, визначає протоколи, що можуть використовуватися для створення безпечного бездротового з'єднання. WPA3 приклад протоколу безпеки, що являє собою покращений стандарт Wi-Fi, який забезпечує вищій рівень захисту від несанкціонованого доступу.

Не менш важливим, забезпечення високого рівня безпеки мережі, є використання правильних протоколів безпеки мережі, що забезпечують захищену передачу даних в мережі та фреймворків інформаційної безпеки – структурований набір документованих процесів, які можуть визначати найкращі методи створення, керування та підтримки заходів безпеки.

До головних протоколів безпеки мережі відносяться:

- Протокол TLS (Transport Layer Protocol) – цей популярний протокол гарантує забезпечення цілісності та конфіденційності даних при передачі їх мережею. Використовуючи методи шифрування даних, наприклад при передачі інформації між двома застосунками, TLS захищає її від перехоплення зловмисником. Це покращена версія протоколу SSL [15].
- Протокол HTTPS (Hypertext Transfer Protocol Secure) – це зашифрована версія HTTP, яка забезпечує захищене з'єднання між браузером та вебсайтом. Для шифрування даних цей протокол використовує TLS (рідше SSL), що захищає інформацію від прослуховування, модифікації або атак типу Man-in-the-

Middle. Префікс https:// показує, що з'єднання знаходиться під захистом саме цього протоколу [15].

- Протокол SSL (Secure Socket Layer) – це оригінальний криптографічний протокол для шифрування чутливих даних та інтернет з'єднань. Хоча його в основному замінено на безпечніший і швидший TLS, цей протокол заклав основу для захищеного каналу зв'язку між користувачем та сервером.
- Протокол SFTP (Secure File Transfer Protocol) – цей протокол забезпечує шифрування при передавання файлів мережею, що дозволяє забезпечити безпечний доступ до потоку даних та можливість керування ним. Зазвичай використовується для передачі файлів між різними системами для гарантування захищеності даних від несанкціонованого доступу [15].

Перелік основних фреймворків мережевої безпеки:

- NIST CSF – цей фреймворк був розроблений Національним інститутом стандартів та технологій США призначений для захисту критичної інфраструктури, зокрема в галузях енергетики, водопостачання, харчової промисловості, зв'язку, охорони здоров'я та транспорту. Він, в першу чергу, зосереджується на аналізі кіберризиків та управління ними, організовуючи заходи безпеки на основі п'яти основних етапів управління ризиками: ідентифікація, захист, виявлення, реагування, відновлення. NIST CSF використовується як для підприємств приватного сектору, так і для компаній та організацій державного рівня [15].
- NIST SP 800-53 – охоплює практично всі основні аспекти систем управління інформаційною безпекою, та приділяє більше уваги хмарним технологіям. Розроблявся для використання федеральними агентствами, але згодом його почали використовувати й в приватному секторі та на його основі було створено інші фреймворки [15].
- CIS (Center for Internet Security) Controls – являє собою критичні інструменти захисту зосереджені на практичних заходах для зменшення можливості виникнення ризиків безпеки даних та загального підвищення стійкості IT-систем. Головна відмінність від NIST CSF, який акцентує увагу на аналізі

ризиків і управління ними, CIS Controls пропонує конкретні дії для зменшення загроз – від захисту від шкідливого програмного забезпечення до проведення тестування на проникнення (Pentest) [15].

- GDPR – це загальний регламент захисту даних, який встановлює стандарти безпеки та конфіденційності для персональних даних користувачів. Серед основних вимог: використання керування доступом за ролями, принцип найменших повноважень, багатофакторна автентифікація. Недотримання вимог, які задає даний стандарт, може призводити до значних штрафів, виникнення серйозних вразливостей системи, тому його використання є критично важливим для компаній, що працюють на міжнародному рівні [15].

Використання сучасних стандартів безпеки та безпекових фреймворків дозволяє забезпечити високий рівень захищеності мережі, конфіденційність та цілісність даних користувачів, захистити систему від можливих кібератак та інших загроз. Ігнорування цих вимог може призводити до великих штрафів, юридичної відповідальності, фінансових та репутаційних збитків для великих та малих компаній у всіх сферах діяльності.

2 СУЧАСНІ МЕТОДИ ОЦІНКИ ЗАХИЩЕНОСТІ МЕРЕЖ, ВИБІР НАЙЕФЕКТИВНІШИХ ІНСТРУМЕНТІВ ДЛЯ ПРОВЕДЕННЯ ПРАКТИЧНОГО ДОСЛІДЖЕННЯ МЕРЕЖЕВОЇ БЕЗПЕКИ

2.1 Основні критерії для оцінки захищеності мережі

В сучасному світі інформаційна безпеки є найвищим пріоритетом для всіх компаній в будь-якому секторі діяльності. Основною метою є запобігання порушенням конфіденційності, несанкціонованому доступу до даних та їх модифікації. Основою інформаційної безпеки є тріада CIA: Конфіденційність (Confidentiality), Цілісність (Integrity), Доступність (Availability).

Принцип конфіденційності передбачає, що доступ до даних системи може отримати лише користувач, який має відповідні повноваження та право працювати з ними. Всі користувачі системи відповідають за коректність роботи систем контролю доступу таких як логічні (паролі від персональних комп'ютерів, облікових записів, тощо) та фізичних (пропускні картки персоналу, закриті серверні кімнати). Саме тому навчання персоналу є дуже важливою частиною забезпечення принципу конфіденційності. Важливо також обмежувати вільне поширення даних в системі та чітко визначати права доступу до інформації відповідно до повноважень конкретних користувачів [16].

Принцип цілісності передбачає використання заходів для забезпечення точності та актуальності даних. Захист даних від модифікації є важливою частиною інформаційної безпеки, оскільки зловмисники можуть спробувати скомпрометувати дані, що передаються, наприклад використовуючи фішінг-атаки з метою зміни номерів банківських рахунків для зміни отримувача коштів. Цілісність даних може бути порушена випадково через можливий збій в роботі системи, або несвоєчасне створення резервних копій даних в системі [16].

Принцип доступності гарантує надійний доступ до інформації, коли він буде потрібен, особам, що мають право на роботу з цією інформацією. Для забезпечення доступності, дані мають зберігатися в правильно організованій та безпечній системі, це дозволяє швидкому та ефективному веденню бізнес-процесів. Найбільш

вразливими є данні медичних та освітніх устав, оскільки у випадках шифрування даних зловмисниками (з використанням програм-вимагачів) можуть постраждати багато людей, через неможливість отримати інформації, наприклад, про діагноз пацієнта і тд [16].

Пересилання даних в мережах (таких як мережа Інтернет) є невід'ємною частиною сучасного цифрового світу, але під час цього процесу зловмисники можуть спробувати отримати доступ до цієї інформації, спробувати змінити їх або просто пошкодити, що може порушувати принципи тріади CIA, саме тому використання правильних практик кібербезпеки захищає не лише суб'єктів даних, інформація яких зберігається та передається мережею, але й активи компаній (запобігання фінансових та репутаційних втрат).

Не менш важливим критерієм захищеності мережі є її стійкість. Стійкість мережі – це здатність мережі протистояти, поглинати та відновлюватися після непередбачуваних подій, таких як хакерські атаки, які можуть вплинути на продуктивність роботи і доступність для користувачів. Стійкість пов'язана з декількома аспектами: надмірність, гнучкість, масштабованість, різноманітність та безпека. Стійка мережа забезпечує ефективну роботи всієї системи та гарантує, що у разі виникнення інцидентів вона зможе продовжувати виконувати свої функції та бути доступною для користувачів [17].

Надмірність (резервування) передбачає дублювання найбільш критичних систем, сервісів та додатків, що дає можливість використовувати їх у разі відмови основних систем. Зазвичай створюється декілька резервних системи в різних центрах (часто розташованих в різних місцях), що робить неможливим вихід з ладу всіх систем. Якщо один центр виходить з ладу, то трафік буде переправлений на резервний центр, для продовження надання послуг. Але використання резервних систем не виключає ті ж ризик, що і для основних систем, вони так само можуть бути ціллю зловмисників, тому потребують відповідного рівня захищеності [17].

Безпека мережі та мережевого з'єднання забезпечується не лише програмними та технічними засобами, а і вчасним проведенням оцінки можливих ризиків для виявлення можливих вразливостей та запобігання можливим

кібератакам з боку зловмисників. Оцінка кіберризиків забезпечує чіткі та практичні дані щодо ефективності існуючих заходів безпеки, вона надає цінну інформацію про вразливості в системі та ймовірність атак на них, пропонує можливі стратегії пом’якшення ризиків, що допомагає приймати правильні рішення під час процесу управління ризиками та забезпечення відповідності міжнародним стандартам безпеки (ISO/IEC 27001, NIST, тощо).

Наприклад, під час оцінки ризиків може з’ясуватися, що компанія надмірно покладається на міжмережеві екрани та рішення з контролю доступу. Тоді зловмисник може використовувати фішинг атаки для того, щоб обійти захист та отримати доступ над системою. В такому випадку оцінка ризиків наддасть рекомендації щодо проведення регулярного пентестування та розширення можливостей реагування на подібні інциденти [18].

Відповідність всіх критеріїв захищеності мережі та мережевого з’єднання сучасним стандартам безпеки дозволяє забезпечити захист даних від витоків, несанкціонованого доступу та захищає компанію від можливих фінансових та репутаційних втрат. Стійка мережа, використання відповідних систем контролю доступу та проведення регулярної оцінки ризиків гарантує ефективність та безперервність роботи всієї системи, безпеку даних користувачів та загальне фінансове та репутаційне благополуччя компанії.

В межах дослідження захищеності мереж, в даній роботі, найбільшу увагу буде приділено саме стійкості мережі до небезпечних подій, таких як несанкціонований доступ до мережі, кібератаки типу MITM та DoS, резервуванню (дозволить, у випадку виходу з ладу одного з блоків, використовувати резервний центр для підтримки працездатності мережі) та проведенню повноцінної оцінки ризиків, складання звіту та надання рекомендацій щодо покращення захищеності мережі.

2.2 Вибір типу комп’ютерної мережі для проведення дослідження

Для прийняття повноцінного та ефективного рішення щодо захисту інформації в мережах необхідно розуміти різні типи мережевих сервісів. До основних типів мереж відносять:

1) LAN (Local Area Network) – це локальна мережа, що обмежується певною територією (в межах будівлі або дата-центру) (рис. 2.1). У LAN кінцеві пристрої взаємодіють між собою та користуються спільними сервісами, обмін даними між пристроями проходить швидко та ефективно, що робить таку мережу ідеальною для використання як домашню мережу. Такий тип мережі дозволяє підключати різноманітні пристрої від ноутбуків та телефонів до принтерів, кількість їх може налічувати тисячі одиниць. Підключення до мережі Інтернет виконується через маршрутизатор, для ефективної маршрутизації даних використовують комутатори. Пристрої під'єднуються за стандартом Ethernet або через бездротове з'єднання (WLAN) [21]. До основних переваг LAN можна віднести: високу швидкість та відносно високу безпеку, легку інтеграцію пристроїв незалежно від їх типу, централізоване зберігання інформації. До основних недоліків можна віднести: обмеженість території використання мережі та залежність від інфраструктури, оскільки, в разі виходу з ладу важливого вузла, може порушуватися вся працездатність системи [20].

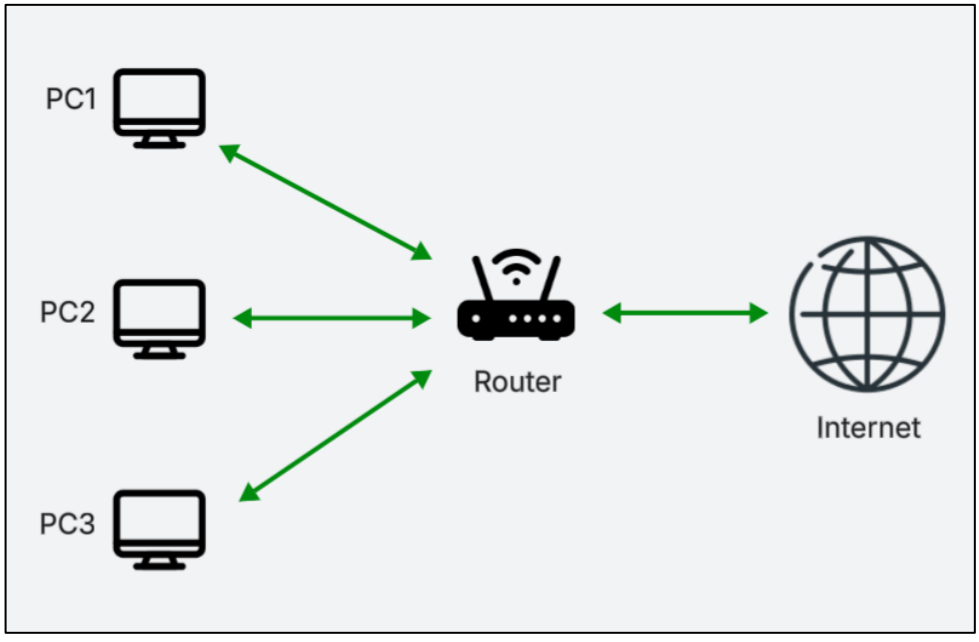


Рисунок 2.1 – Схема-приклад топології мережі LAN

2) WAN (Wide Area Network) – це глобальна комп’ютерна мережа (рис. 2.2), що дозволяє охоплювати великі відстані та може працювати на необмеженій території, використовується для встановлення з’єднань між регіонами, країнами або навіть цілими континентами. Як правило, WAN використовується великими компаніями для передачі даних між офісами (всередині офісів використовується LAN). Базується на технологіях IP/MPLS, PDH, SDH [21]. До переваг WAN можна віднести: покращення комунікації між окремими інформаційними структурами, ефективний обмін даними, гнучкість та масштабованість. До основних недоліків відносяться: велика вартість оренди ліній зв’язку та їх обслуговування, складність адміністрування та відчутні затримки при передачі даних через великі відстані між об’єктами [19].

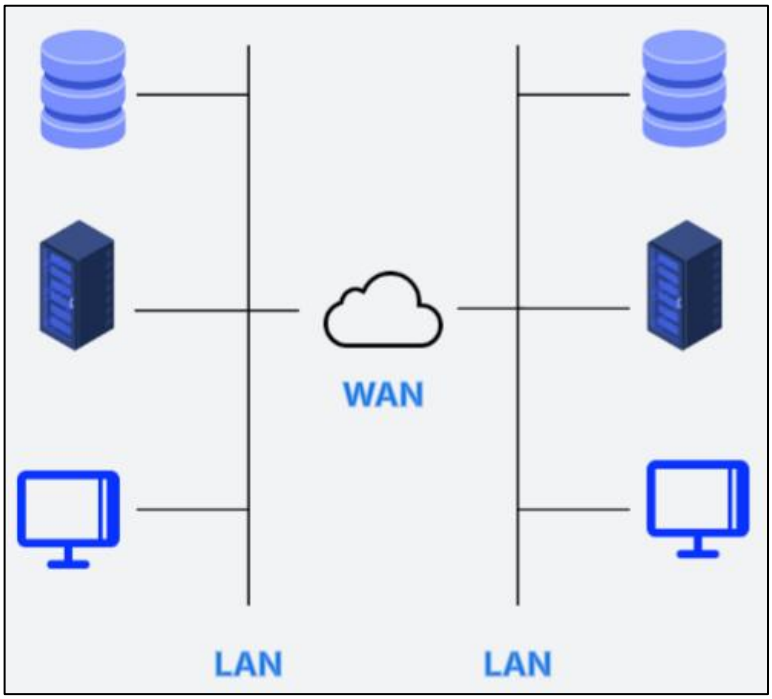


Рисунок 2.2 – Схема-приклад топології мережі WAN

3) WLAN (Wireless Local Area Network) – це вид локальної мережі, який базується на використанні бездротового з’єднання між пристроями (рис. 2.3). Для підключення використовується маршрутизатор та шифрування даних (WPA або WEP) для забезпечення безпеки під час передачі.

WLAN працює за стандартом IEEE 802.11 і використовує бездротові точки доступу (WAP), які керують трафіком та сигналом [21]. Основні переваги: гнучкість та мобільність, можливість підключення відразу декількох пристроїв без використання додаткового обладнання (комутаторів та кабелів), висока швидкість передавання даних [20]. Основні недоліки: обмежений радіус дії підключення, зменшення швидкості передачі у випадку завантаженості каналів, вразливість до атак типу MITM та заглушення сигналу .

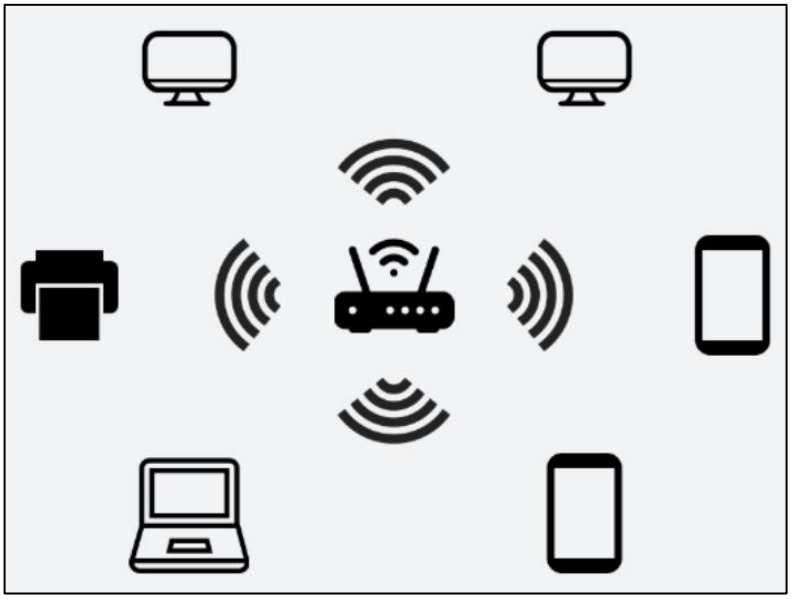


Рисунок 2.3 – Схема-приклад топології мережі WLAN

4) VPN (Virtual Privat Network) (рис. 2.4) – це технологія, які шифрує дані та з’єднання через публічні або приватні мережі. Такий підхід дозволяє забезпечити захищеність даних під час їх передачі та маскує IP-адресу користувача для його конфіденційності в мережі, дозволяє обходити географічні обмеження. Переваги VPN: шифрування даних та захист від модифікації, приховування місцезнаходження користувача, встановлення без фізичного з’єднання [21]. Недоліки VPN: зниження швидкості передачі даних, захищеність даних залежіть від надійності VPN-провайдера, у разі виявлення VPN, у деяких випадках, можуть бути накладені обмеження на користувача [19].

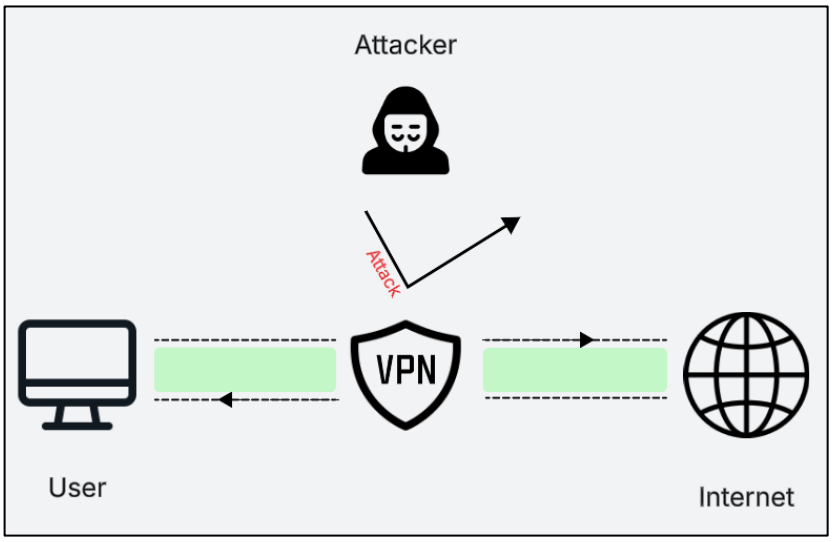


Рисунок 2.4 – Схема-приклад роботи VPN

5) Cloud Networking (Хмарні мережі) (рис. 2.5) – це мережі, що використовують хмарні технології для управління та надання мережевих ресурсів. Вони поєднують в собі гнучкість, масштабованість та ефективність хмарних обчислень для створення ефективної та захищеної мережі. Основні недоліки такого типу мереж: пряма залежність від наявності підключення до мережі Інтернет, залежність від працездатності хмарного провайдера (при порушенні в роботі провайдера, мережа стає недоступною для використання), питання конфіденційності даних, оскільки вони зберігаються на сторонньому сервісі [20].

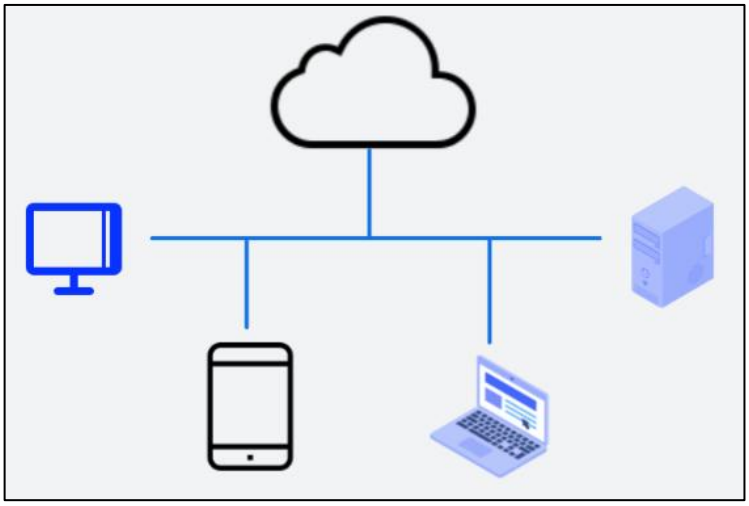


Рисунок 2.5 – Схема-приклад хмарної мережі

Аналізуючи результати дослідження типів комп’ютерних мереж та порівняльну характеристику їх переваг та недоліків (Таб. 2.1), можна зробити висновок, що найкращий варіант, для проведення практичного дослідження, це комбіноване використання LAN та WLAN мережі. Такий підхід дозволить створити повноцінну локальну мережу відповідного до: мети роботи, розміру з визначеними вузлами, різними типами пристроїв та конфігураціями, змоделювати атаки типу MITM, протестувати ефективність роботи брандмауерів та міжмережевих екранів, проаналізувати трафік, що буде проходити мережею.

Таблиця 2.1 – Порівняльна таблиця типів комп’ютерних мереж

Тип мережі	Зона покриття	Переваги	Недоліки	Приклад використання
LAN	Локальна	Висока швидкість, безпека передачі, централізованість	Обмеженість території, інфраструктурна залежність	Домашні та офісні мережі
WAN	Глобальна	Міжрегіональна комунікація, масштабність, гнучкість	Висока вартість обслуговування, складність адміністрування	Корпоративні мережі між філіями компанії
WLAN	Локальна	Мобільність, простота підключення	Обмежений радіус, зменшення швидкості передачі.	Мережа Wi-Fi
VPN	Будь-яка мережа	Шифрування даних, приховування IP	Зниження швидкості, залежність від провайдера, можливе блокування	Захищений доступ до корпоративної мережі
Хмарна мережа	Віртуальна	Гнучкість, масштабованість, відносно низька вартість утримання	Залежність від інтернету та надійності провайдера	Інфраструктура як сервіс (IaaS)

2.3 Визначення основних вразливостей сучасних комп’ютерних мереж для проведення тестування та оцінки ризиків, методи протидії цим загрозам

Для проведення повноцінної оцінки захищеності мережі та мережевого з’єднання дуже важливо визначити, які саме вразливості можуть виникати в безпеці сучасних комп’ютерних мереж, як саме вони можуть впливати на роботу системи та як їм протидіяти у разі виявлення. Серед великої кількості вразливостей різного типу та критичності безпеці даних в мережі можна відзначити найбільш поширені (саме вони найчастіше призводять до інцидентів з НСД та іншими загрозами): застаріле програмне забезпечення, некоректне налаштування міжмережевого екрану (Фаєрвол), слабкі або стандартні паролі для підключення до мережі, незахищені/відкриті точки доступу, відсутність регулярних аудитів безпеки.

Застаріле програмне забезпечення розповсюджена вразливість, яка притаманна багатьом інформаційним системам в тому числі і мережам. Таке ПО містить відомі вразливості, які були виправлені в нових версіях. Використання такого забезпечення не лише унеможливлює застосування функціональних покращень, але і містить недоліки безпеки, які вже використовуються зловмисниками. Несанкціонований доступ до інформації, витоки інформації, використання бекдорів для встановлення шкідливого ПО все це є наслідком не своєчасного оновлення системи. Найбільш прости рішенням такої проблеми є регулярне оновлення та встановлення патчів, які вирішують відомі проблеми безпеки [22].

Некоректне налаштування міжмережевих екранів – це критична вразливість мережевої безпеки, яка виникає через некомпетентність користувача відповідального за безпеку або через банальну неуважність. Фаєрвол – це основний бар’єр проти несанкціонованого доступу і ефективність такого захисту залежить від точного налаштування та своєчасного оновлення. Некоректне налаштування може проявлятися в різних формах, наприклад, неправильне початкове налаштування, відсутність нового патчу, невідповідність налаштування правилам політики безпеки системи. Наслідки використання цієї вразливості зловмисниками можуть бути дуже серйозними та масштабними, через відкриті порти може бути

виконано спроби НСД до мережі, витік даних та можуть виникати інші загрози безпеці. Регулярне оновлення міжмережевих екранів, створення політики безпеки, що буде відповідати реальним потребам безпеки мережі та виконання, спеціалістом з безпеки, правильного налаштування фаєрволу дозволить уникнути проблем з захищеністю системи [22].

Використання слабких або стандартних паролів може призводити до легкого їх зламу та вільного доступу зловмисників до мережі. Якщо зловмисник отримає доступ всередину мережі, він зможе почати перехоплювати пакети даних, що передаються між пристроями, в результаті чого отримає доступ до читання та модифікації конфіденційної інформації. Впровадження політики надійних паролів зможе гарантувати, що доступ до системи та даних зможуть отримати лише легітимні користувачі.

Використання протоколі без шифрування даних. Ця вразливість є дуже критичною та впливає на загальну безпеку даних, що передаються всередині мережі. Протоколи HTTP та FTP не використовують шифрування під час передачі, що робить інформацію вразливою для перехоплення, читання та модифікації. У випадку атаки типу MITM зловмисник отримає доступ до відкритого тексту, який він зможе відразу прочитати тим самим порушивши конфіденційність даних. У разі наявності в системі цієї вразливості та, наприклад, використання слабкого паролю для доступу до мережі, наслідки можливої кібератаки можуть бути критичними і призвести до масштабного витоку даних. Уникнути цього можна, використовуючи протоколи з шифруванням даних HTTP >> HTTPS, FTP >> SFTP.

Незахищені точки доступу до мережі, такі як відкриті Wi-Fi мережі можуть стати серйозною проблемою для мережевої безпеки. Ці точки можуть бути використані як відкриті двері для входу в мережу будь-якому користувачеві, в тому числі і зловмиснику. Це дозволяє виконувати несанкціонований доступ до мережі, впроваджувати шкідливі програми безпосередньо зсередини системи. Особливо небезпечною ця вразливість є для бездротових мереж зі слабким захистом та недостанім контролем гостьових мереж. Рішенням є використання протоколів

шифрування бездротових мереж та використання IDS/IPS для виявлення підозрілої активності в мережі [22].

Відсутність або використання слабких механізмів обмеження трафіку. Наявність такої вразливості в мережі дозволяє зловмисникам реалізовувати атаки типу DoS, відправляючи масовий потік запитів на сервер або сервіс з метою перевантаження його можливостей та виведення з ладу всієї системи. Використання механізму обмеження швидкості дозволяє обмежити кількість запитів, які можуть бути зроблені користувачем за певний проміжок часу. Це дозволяє регулювати та керувати потоком трафіку для забезпечення доступності сервісу [23].

Проведення регулярних аудитів безпеки дозволяє перевірити мережу на відповідність політикам безпеки та визначити загальний рівень захищеності. Проведення аудиту гарантує, що можливі проблеми безпеки будуть вирішені, а дані в мережі будуть захищені від дій потенційних зловмисників.

Саме виявлення перелічених вище вразливостей є основною метою практичного дослідження в даній роботі. На основі виявлених проблем безпеки буде проведено комплексну оцінку захищеності мережі та мережевого з’єднання а також сформульовано рекомендації щодо підвищення рівня інформаційної безпеки.

2.4 Вибір методів та інструментів для практичного дослідження на основі їх ефективності та актуальності

Для проведення практичного дослідження необхідно чітко визначити, які саме інструменти та методи оцінки захищеності мережі слід використовувати задля виявлення, описаних в підрозділі 2.3, поширених вразливостей та перевірки відповідності основних критеріїв захищеності мережі та мережевого з’єднання достатньому рівню безпеки. Існує безліч методів захисту інформації та інструментів для їх реалізації, з них можна виділити найбільш ефективні та актуальні в сучасному світі кібербезпеки: тестування на проникнення (penetration testing), моніторинг безпеки мережі (моніторинг мережевого трафіку), автоматизоване сканування вразливостей.

2.4.1 Тестування на проникнення (penetration testing) – основний метод виявлення вразливостей в мережі

Тестування на проникнення (penetration testing) – це один з найбільш ефективних та інформативних методів оцінки захищеності мережі. Він базується на практиці етичного хакінгу. Суть тестування полягає в тому, щоб етичні хакери (“red teams”), використовуючи спеціалізовані інструменти, імітували реальну кібератаку на систему. Даний вид тестування може бути націлений на веб-застосунки, API, кінцеві точки, тощо. Такі атаки дозволяють виявити актуальні вразливості в мережевій інфраструктурі, такі як: слабкі паролі, нехахищені точки доступу, некоректно налаштовані фаєрволи, протоколи передачі даних без шифрування, відсутність фільтрування та обмеження трафіку. Існує два типи тестів: внутрішні та зовнішні [24].

Внутрішнє тестування мережі – етичні хакери діють як зловмисники, які змогли отримати доступ всередину системи. Основною метою є виявлення вразливостей, які можуть бути використані внутрішнім користувачем (крадіжка інформації, зловживання правами доступу) [24].

Зовнішнє тестування мережі – імітація дій потенційних зловмисників, які намагаються проникнути всередину мережі ззовні. Основною метою є виявлення вразливостей в компонентах, що мають прямий доступ до Інтернету (маршрутизатори, веб-застосунки, тощо).

Процес тестування на проникнення складається з декількох етапів [24]:

1) Збір інформації та планування – проводиться попередня оцінка вразливостей, ознайомлення з відомими вразливостями (за типом мережі, її компонентами, тощо).
2) Виявлення – процес знаходження слабких місць в системі за допопомогою сканування портів (для виявлення відкритих), сканерів вразливостей.
3) Проведення тесту на проникнення в мережу – етап на якому, на практиці, використовують експлойти задля експлуатації, виявлених на попередніх етапах, вразливостей. Мета – проникнути в систему та оцінити, яку саме шкоду можна завдати.

4) Аналіз та звітність – створення звіту про проведене тестування, аналіз отриманих результатів, надання рекомендацій щодо виправлення вразливостей та покращення рівня захищеності мережі.

Використання тесту на проникнення в практичному дослідженні дозволить виявити основні вразливості мережі, наддасть глибоке розуміння загального стану захищеності мережі. Такий метод є дуже інформативним та ефективним, що дозволяє охопити та перевірити більшість аспектів безпеки системи, отримати багато аналітичної інформації щодо слабких місць в мережевій інфраструктурі. Для реалізації тестування на проникнення в практичному дослідженні необхідно визначити найбільш ефективний та доцільний для використання інструмент, який буде відрізнятися функціональністю, зручністю використання, рівнем автоматизації. У таблиці 2.2 наведено порівняльний аналіз найбільш популярних інструментів, що дозволяють реалізувати метод тестування на проникнення в мережу [25].

Таблиця 2.2 – Порівняльний аналіз популярних інструментів тестування на проникнення в мережу

Інструмент	Ціль використання	Основні функції	Переваги	Недоліки
LoadFocus Security Suite	Інтегроване тестування продуктивності	Стрес-тестування, моніторинг API, сканер вразливостей	Масштабованість, продуктивність та безпека	Складність інтеграції в систему
Burp Suite Professional	Оцінка захищеності веб-застосунків	Ручне тестування, обробка сесій, можливість комплексного сканування мережі	Гнучкість, можливість тестування великої кількості застосунків	Необхідність великої кількості ресурсів, платна ліцензія

Продовження таблиці 2.2

Інструмент	Ціль використання	Основні функції	Переваги	Недоліки
Metasploit Framework	Перевірка вразливостей та розробка експлойтів для тестування	Автоматизовані скрипти, генерація payload-ів	Просунуті тести на проникнення, відкритий код	Не підходить для швидких перевірок, складність
Nmap	Сканування мережі та аудит безпеки	Сканування портів та визначення підключених хостів, визначення служб	Надійність, багатофункціональність, швидкість роботи	Командний інтерфейс
Kali Linux	Комплексне середовище для тестування на проникнення в мережі	Комплексний набір необхідних інструментів безпеки	Повна оцінка безпеки, гнучкість, великий обсяг документації	Велика кількість різних утиліт для тестування

Аналізуючи результати порівняння можна зробити висновок, що найбільш доцільним, для проведення практичного дослідження на основі мети роботи, є використання Kali Linux. Цей інструмент надає широкий спектр утиліт для виявлення основних вразливостей мережі, дозволяє протестувати конфігурацію мережі, перевірити стійкість до атак типу “людина посередині” (MITM), DoS, підбору паролів (brute-force). При цьому Kali Linux не потребує великої кількості ресурсів системи та його можна використовувати на мережах довільного розміру (у масштабі обмеженого тестового стенду).

2.4.2 Моніторинг безпеки мережі (NSM)

Моніторинг безпеки мережі (Network Security Monitoring) – це важливий метод оцінки захищеності мережі, який полягає в безперервного аналізу мережевого трафіку та активності в системі для виявлення аномального трафіку або підозрілих дій в мережі та реагування на загрози безпеки в режимі реального часу. Головною метою є виявлення вразливостей в мережевій інфраструктурі та

попередити потенційні кібератаки. До основних компонентів NSM-інструментів відносяться [26]:

1) Аналіз трафіку в режимі реального часу. Постійний нагляд за пакетами, що передаються мережею для вчасного виявлення незвичних шаблонів комунікацій або підозрілого обсягу даних.
2) Управління логами. Це процес збору, зберігання та аналізу логів отриманих від компонентів мережевої інфраструктури (маршрутизаторів, серверів, застосунків, тощо). Це дозволяє відстежувати інциденти порушення безпеки, знайти скомпрометовані системи та виявити підозрілі дії в межах усієї мережі.
3) Виявлення вторгнень. NSM-інструменти надають змогу виявляти спроби несанкціонованого доступу до мережі, порушення політик безпеки та інші ознаки злому. Часто такі системи (IDS/IPS) можуть не тільки виявляти, а й реагувати на подібні інциденти (блокувати трафік, ізолювати певні компоненти інфраструктури для уникнення поширення атаки).
4) Виявлення аномалій. Процес, який передбачає ідентифікацію відхилень поведінки мережі від очікуваної. Він базується на порівнянні з базовим шаблоном звичної активності в системі, що дозволяє виявляти атаки нульового дня або APT (Advanced Persistent Threats).

Використання такого методу оцінки захищеності мережі дозволяє ефективно виявляти вразливості такі як некоректне налаштування міжмережевих екранів, використання протоколів без шифрування, відсутність механізмів обмеження трафіку та використання незахищених точок доступу, що робить його застосування, в межах практичного дослідження, найбільш доцільним для автоматизованого відстеження стану безпеки мережі в реальному часі. Для реалізації цього методу можна використовувати безліч інструментів, найбільш ефективні та актуальні перелічені в порівняльній таблиці 2.3 [26].

Таблиця 2.3 – Порівняльний аналіз найбільш ефективних NSM-інструментів

Інструмент	Основні функції	Переваги	Недоліки	Масштаб використання
Wireshark	Глибокий аналіз пакетів, аналіз на рівні протоколів	Підтримка великої кількості протоколів, висока деталізація, безкоштовне ПЗ	Потребує високого рівня знань в області мережевих технологій, не підходить для масштабного моніторингу	Аудитори безпеки, аналіз інцидентів у малих та середній мережах
Splunk	SIEM-платформа, аналітика великих даних, виявлення загроз з використання ML	Потужна аналітика, масштабованість, ефективне виявлення загроз	Платна ліцензія, складність використання	Великі підприємства з високими вимогами до мережевої безпеки
Suricata	IDS/IPS, виявлення загроз у реальному часі, багатопотокова обробка та аналіз протоколів	Висока продуктивність, гнучкість, масштабованість, відкритий код	Високе навантаження на ресурси, складність налаштування	Великі організації, що потребують швидкого аналізу трафіку
Security Onion	Комплексне рішення з інтегрованими IDS/IPS, стек ELK для логів	Безкоштовне ПЗ, велика кількість інструментів для аналізу	Ресурсомісткість, необхідна достатня технічна підготовка	Середні та великі організації
ELK Stack	Збір, індексаці, аналіз та візуалізація логів	Гнучкість, масштабованість, відкритий код	Складність налаштування, потреба в оптимізації для ефективної роботи	Організації з достатньою кількістю технічних ресурсів

Аналізуючи результати порівняння програмних засобів для моніторингу безпеки мережі, можна побачити, що найбільш доцільним, для використання в практичному дослідженні, є інструмент Suricata. Цей дистрибутив має комплексний набір інструментів для моніторингу та аналізу трафіку в мережі, дозволяє виявити більшість поширених вразливостей мережевої безпеки (використання протоколів без шифрування, наявність відкритих портів, тощо).

Важливою перевагою є гнучкість масштабування, що дає змогу використовувати його в мережах довільного розміру (таких як тестовий стенд). Suricata є відкритим безкоштовним ПЗ, що робить його доступним для використання будь-якому користувачеві.

2.4.3 Сканування вразливостей мережі

Сканування вразливостей – це автоматизований процес пошуку відомих вразливостей у безпеці мережі. Основною метою такого методи оцінки захищеності є виявлення слабких місць у системі до того, як їх використає зловмисник. Процес сканування вразливостей має чітку структуру реалізації, яка характеризується наступними етапами:

1) Ідентифікація та інвентаризація: виконується ідентифікація та каталогізація всіх пристроїв та програмних засобів в межах системи. Це дозволяє сформувати чітку структуру побудови мережевої інфраструктури, її компонентів, для подальшого сканування [27].
2) Виявлення та аналіз: наступним етапом є сканування мережі задля виявлення вразливостей. Цей процес базується на принципі сигнатурного аналізу, коли конфігурація системи, версії ПЗ, мережеві налаштування порівнюються з базою даних відомих вразливостей. Після сканування виявлені вразливості класифікують за системою оцінки CVSS (Common Vulnerability Scoring System), залежно від потенціалу реалізації та критичності (від 0.0 – відсутність ризику, до 10.0 – критична загроза).
3) Після сканування та виявлення вразливостей наступним кроком є усунення, використовуючи оновлення систем та ПЗ або застосування інших змін в мережі. Виконується повторне сканування для підтвердження успішно усунення вразливостей, впроваджується постійний моніторинг для забезпечення безперервного захисту [27].

Існують різні типи сканування вразливостей: активне сканування – передбачає надсилання прямих запитів та зондуючих сигналів до пристроїв; пасивне сканування – відстежує мережевий трафік без взаємодії з системою, внутрішнє сканування – перевірка систем усередині периметру мережі; Зовнішнє

сканування – проводиться оцінка систем, що взаємодіють з мережею Інтернет (хмарні сховища, веб-застосунки); автентифіковане сканування – перевірка системи з використання прав адміністратора (прав привілейованого доступу); неавтентифіковане сканування – перевірка систем без даних для входу в систему.

Використання цього методу виявлення вразливостей в мережі є найбільш доцільним в межах практичного дослідження, оскільки такий підхід дозволяє автоматизовано виявляти широкий спектр потенційних загроз безпеці. Сигнатурний аналіз та система класифікації виявлених вразливостей дозволяє не тільки виявити слабкі місця, але і виставити пріоритети щодо першочерговості їх усунення. Це дозволяє провести повноцінну оцінку захищеності мережі, виявити використання застарілого ПЗ та служб з відомими CVE-вразливостями. Для вибору інструментів для сканування вразливостей в мережі необхідно провести порівняльний аналіз їх переваг та недоліків (таб. 2.4), щоб повноцінно оцінити їх ефективність та доцільність їхнього використання в межах практичного дослідження [28].

Таблиця 2.4 – Порівняльний аналіз інструментів для сканування вразливостей в мережі

Інструмент	Основні функції	Переваги	Недоліки
OpenVAS	Сканування серверів, мережевих пристроїв, є скриптова мова для створення тестів	Безкоштовний доступ, open-source, можливість гнучкого налаштування	Складність налаштування конфігурації
Nessus	Широкий спектр можливостей для виявлення вразливостей ОС та ПЗ (наприклад, неправильна конфігурація), формування інформативних звітів	Має безкоштовну версію (до 16-ти IP-адрес), зручний для використання інтерфейс, надійність роботи	Платна повна версія, відсутність графічних звітів
Intruder	Інтеграція з CI/CD, зовнішній сканер, виявлення нових портів та сервісів	Проста інтеграція, автоматизація процесів виявлення	Неінформативні звіти, відсутність глибокої діагностики

Продовження таблиці 2.4

Інструмент	Основні функції	Переваги	Недоліки
Vulnerability Manager Plus	Сканування вразливостей з точки зору зловмисника, виявлення помилок конфігурації	Глибока аналітика, веб-тестування, сканування нульового дня	Ручне затвердження патчів, висока вартість ліцензії, надлишкова функціональність для малих мереж
Tripwire IP360	Підтримка безагентного та агентного сканування активів, інтеграція з системами управління ризиками	Масштабний сканер, комплексний підхід до управління ризиками та безпеки	Відсутність корпоративної підтримки, висока вартість ліцензії, підходить лише великим компаніям

Аналізуючи результати порівняння популярних інструментів для сканування вразливостей в мережі, можна зробити висновок, що найбільш доцільним варіантом для виконання практичного дослідження є використання інструменту Nessus. Цей сканер є досить зручним у використанні, має широкий спектр можливостей для виявлення критичних вразливостей, що дозволяє ефективно виявляти відкриті порті, помилки конфігурації, використання протоколів без шифрування. Nessus надає безкоштовну версію для використання в мережах розміром до 16-ти IP-адрес та не має надлишкової функціональності, що чудово підходить для малих та середніх тестових мереж.

Для проведення повноцінної оцінки захищеності мережі та мережевого з’єднання необхідно використовувати комплексний підхід. Моніторинг мережевої безпеки дозволить виявити аномальну поведінку в мережі, неправильне налаштування міжмережевих екранів або виявити відкриті порти, відсутність засобів обмеження трафіку. Сканування вразливостей в мережі дозволить автоматизовано виявляти слабкі місця в системі. Тестування на проникнення дозволить протестувати загальний рівень безпеки мережі, перевірити її на стійкість до атак типу MITM та DoS. Проаналізувавши результати, отримані після використання всіх цих методів, можна повноцінно оцінити рівень захищеності мережі та мережевого з’єднання, розробити рекомендації щодо усунення виявлених проблем та недоліків, покращення загального рівня мережевої безпеки.

3 ПРАКТИЧНА ОЦІНКА ОСНОВНИХ КРИТЕРІЇВ ЗАХИЩЕНОСТІ МЕРЕЖІ ВІДПОВІДНО ДО СУЧАСНИХ СТАНДАРТІВ БЕЗПЕКИ

3.1 Проєктування та розгортання тестового середовища, налаштування мережевих вузлів та інструментів для оцінки захищеності мережі

Основною метою практичного дослідження є проведення комплексної оцінки захищеності мережі та мережевого з'єднання, виявлення вразливостей тестової мережі, виконання тестування на проникнення в мережі, проведення аналізу отриманих результатів та надання рекомендацій щодо покращення загального рівня безпеки в мережі. Для запуску тестового середовища буде використано робочу станцію з такими апаратними характеристика:

- Модель: Ноутбук ASUS TUF GAMING A15.
- Процесор: AMD Ryzen 7 7735HS, 8 фізичних ядер, 16 потоків (логічних ядер).
- Оперативна пам'ять: 16 ГБ DDR5 5800Mhz.
- Тип носія: SSD M.2 512 ГБ.
- Ос хосту: Windows 11.

3.1.1 Проєктування мережі: визначення ролі вузлів, необхідних ресурсів для реалізації функціоналу, IP-адрес хостів в підмережі

Для проведення практичної оцінки захищеності мережі було спроєктовано ізольоване тестове середовище, яке імітує локальну мережу (LAN), що буде складатися з 10-ти віртуальних машин. Кожній віртуальній машині буде виділено певні ресурси системи для коректної роботи:

- Одна віртуальна машина на основі дистрибутива Kali Linux – RAM: 4096 МБ, CPU: 5 логічних ядер. Саме таких ресурсів достатньо для використання інструментів тестування захищеності мережі та виконання моніторингу трафіку в мережі.
- Дев'ять віртуальних машин на основі дистрибутива Ubuntu 20.04 – RAM: по 512 МБ, CPU: по 1 логічному ядру. Цих ресурсів достатньо для виконання базових команд Linux.

Тестова мережа повинна містити 10 вузлів, які представляють собою: атакуючий/моніторинг хост (на основі дистрибутива Kali Linux, який інструменти для моніторингу трафіку, сканування вразливостей та тестування на проникнення), три хости-сервери на основі дистрибутиву Ubuntu 20.04 (для демонстрації вразливості використання незашифрованих протоколів передачі даних), шість хостів теж на основі дистрибутиву Ubuntu 20.04, які імітують звичайних користувачів в мережі. Сама мережа буде розгорнута в межах підмережі класу С (192.168.56.0/24), всі вузли матимуть однакові налаштування мережевого адаптера (під'єднаний до: внутрішня мережа, змішаний режим: дозволити всім, встановлено режим: cable connected), такий підхід дозволить обмін трафіком між ними та унеможливлює витік трафіку у зовнішню мережу Інтернет. Кожен вузол буде мати свою IP-адресу та роль в мережі:

- Kali Linux (192.168.56.10) – Атакуючий хост з інструментами для моніторингу трафіку.
- HTTP server (192.168.56.31) – Apache сервер, що буде використовувати протокол HTTP для передачі даних мережею.
- FTP server (192.168.56.32) – Сервер, що буде використовувати FTP незашифрований протокол для передачі даних в мережі.
- SFTP server (192.168.56.33) – Сервер з використанням захищеного SFTP протоколу для передачі даних в мережі.
- Users (192.168.56.34-39) – Звичайні користувачі в мережі, які виконують базові команди Linux.

3.1.2 Практична реалізація та розгортання тестової мережі

Для зручності використання та автоматизації процесу налаштування та запуску мережі було встановлено інструмент Vagrant останньої версії 2.4.5 (рис. Б.1), який дає змогу маніпулювати гіпервізором (наприклад Oracle VirtualBox) за допомогою конфігураційного файлу (vagrantfile), детальніше можна побачити у додатку А. Для практичної реалізації спроєктованої тестової мережі для початку необхідно ініціалізувати Vagrant середовище (рис. Б.2) для створення

конфігураційного файлу (Vagrantfile). Далі виконуємо налаштування конфігураційного файлу відповідно до, описаних вище, функцій вузлів (рис. Б.3).

Командою config.vm.box_check_update = false відключаємо автоматичне оновлення vagrant-боксів після запуску системи. config.vm.synced_folder '.', '/vagrant', disabled: true вимикає синхронізацію хостом та гостьовими ВМ для уникнення автоматичного монтування /vagrant. Далі налаштовуємо наші віртуальні машини: config.vm.define - надає ім'я нашій ВМ; kali.vm.box = "kalilinux/rolling" – вказує, що образом машини буде дистрибутив Kali Linux (аналогічно для інших ВМ буде node.vm.box = "ubuntu/focal64", що вказує на диструбив Ubuntu останньої версії); kali.vm.network "private_network", ip: "192.168.56.10", netmask: "255.255.255.0", virtualbox__intnet: "intnet" – налаштовує приватну внутрішню мережу з IP-вузла 192.168.56.10 в підмережі 192.168.56.0/24, з маскою 255.255.255.0 (аналогічно налаштовуються інші дев'ять віртуальних машин з наданням відповідних IP-адрес «192.168.56.31-39»); kali.vm.provider "virtualbox" do |vb| - налаштовує ресурси машини для VirtualBox, ім'я ВМ, обсяг RAM та кількість логічних ядер (аналогічно налаштовуються інші машини відповідно до потреб). Також виконуємо налаштування мережевих адаптерів для всіх віртуальних машин описане в підрозділі 3.1.1, а саме: під'єднаний до: внутрішня мережа, змішаний режим: дозволити всім, встановлено режим: cable connected (рис. Б.4).

Після налаштування конфігураційного файлу та мережевих адаптерів виконуємо команду vagrant status для перевірки наявності описаних віртуальних машин (рис. Б.5) та виконуємо команду vagrant up для розгортання тестової мережі (рис. Б.6).

В результаті інструмент vagrant виконує запуск одночасно всіх віртуальних машин, які, завдяки правильному налаштуванню конфігураційного файлу та мережевих адаптерів, будуть знаходитися в одній приватній локальній мережі. Для перевірки топології нашої мережі переходимо до Kali Linux, запускаємо інтегровану утіліту Zenmap (офіційний GUI для інструменту Nmap) та проводимо сканування (Quick scan) в підмережі 192.168.56.0/24 для виявлення активних хостів (рис Б.7) та візуального представлення топології мережі (рис. Б.8).

Виконуємо перевірку відповіді хостів на команду ping для цього командою vagrant ssh ubuntu1 (vagrant ssh “ім’я віртуальної машини” надає доступ до консолі запущеної ВМ) переходимо до консолі віртуальної машини ubuntu1 та виконуємо команди: ping 192.168.56.32, ping 192.168.56.10, ping 192.168.56.36. Можна побачити, що команди виконуються успішно (рис. Б.9), тобто всі хости знаходяться в одній локальній мережі, отримують пакети даних та працюють коректно.

3.1.3 Налаштування мережевих вузлів, встановлення та налаштування інструментів для оцінки захищеності мережі

Після успішного запуску тестового стенду та перевірки правильності налаштування конфігурації мережі наступним етапом йде налаштування мережевих вузлів, встановлення та налаштування інструментів для проведення комплексної оцінки захищеності мережі. Для моделювання можливих вразливостей в реальній локальній мережі, таких як використання незахищених протоколів передачі даних та неправильне налаштування фаєрволу, створимо сервери з використання протоколів HTTP та FTP (для демонстрації їх вразливостей до перехоплення даних), сервер на основі SFTP протоколу (для демонстрації різниці в безпеці передачі даних) та налаштуємо один з вузлів так, щоб фаєрвол був вимкнений (що призведе до вільного проходження трафіку через хост без перевірок на наявність можливих загроз).

Спочатку виконаємо розгортання веб-сервера Apache на Ubuntu для цього підключаємося до консолі ВМ ubuntu1 командою vagrant ssh ubuntu1, далі встановлюємо Apache 2 (sudo apt-get install -y apache2) (рис. Б.10). Після успішного встановлення, служба автоматично запускається і сервер починає працювати. Перейшовши до консолі ubuntu6 та виконавши команду curl http://192.168.56.31/ можна переконатися, що сервер дійсно працює і, у відповідь на запит користувача, повертає HTML-код стандартної головної сторінки Apache2 (рис Б.11).

Далі налаштовуємо FTP-сервер на хості ubuntu2, для цього переходимо до консолі ВМ командою vagrant ssh ubuntu2, виконуємо sudo apt-get install -y vsftpd (встановлює FTP-сервер)(рис.Б.12), запускаємо службу (sudo systemctl enable vsftpd –now) та налаштовуємо конфігурацію для надання користувачам дозволу на

підключення по FTP (sudo sed -i ‘s/^#\(local_enable=\)YES/\1YES/’ /etc/vsftpd.conf – дозволяє локальним користувачам входити через FTP, sudo sed -i ‘s/^#\(write_enable=\)YES/\1YES/’ /etc/vsftpd.conf – дозволяє запис файлів) (рис. Б.13), перезавантажуємо службу командою sudo systemctl restart vsftpd для прийняття змін. Для подальшої роботи створюємо нового користувача (sudo adduser testftp –gecos “” –disable-password)(рис. Б.14) та додаємо новий пароль (echo “testftp:password123” | sudo chpasswd)(рис. Б.15). Для перевірка правильності виконаних дій, підключимось до FTP-сервера через Kali Linux. Відкриваємо термінал та виконуємо команду ftp 192.168.56.32, вводимо логін та бачимо, що підключення виконано успішно (рис. Б.16), тобто служба працює корректно.

Аналогічно налаштовуємо SFTP-сервер на хості ubuntu3, встановлюємо openssh-server (рис. Б.17), додаємо нового користувача (рис. Б.18). Командою sudo systemctl status ssh перевіряємо стан сервера та бачимо, що служба запущена успішно (рис. Б.19). Переходимо на хост з Kali Linux, відкриваємо термінал та підключаємося до SFTP-сервера (sftp testsftp@192.168.56.33), з’єднання успішно виконано (рис. Б.20).

Змоделюємо неправильне налаштування фаєрволу (повністю його відключимо) на хості uvuntu4. Командою vagrant ssh ubuntu4 переходимо в консоль відповідної віртуальної машини, перевіримо наявність встановленого фаєрволу (рис. Б.21). Використавши sudo ufw disable, вимкнемо міжмережевий екран та перевіримо результат (рис. Б.22). Після налаштування ми можемо побачити, що все було виконано вірно і тепер змодельовану вразливість можна буде виявити в процесі сканування мережі.

Заключним етапом є налаштування Kali Linux, як атакуючого хосту: встановлення базових інструментів для проведення тестування на проникнення (Ettercap, Bettercap, tcpdump, dsniff), як хосту для моніторингу та сканування мережі: встановлення та налаштування Nessus, Suricata та інструментів для відображення трафіку в графічному вигляді (elasticserarch, kibana, logstash).

Спочатку встановимо основні інструменти пентесту, виконавши команди sudo apt install -y ettercap-text-only bettercap (рис. Б.23), sudo apt install -y tcpdump та

sudo apt install -y dsniff (рис. Б.24), завантажимо консольну версію Ettercap та Bettercap для проведення атак типу MITM та ARP-spoofing, утиліту tcpdump для перехоплення та аналізу мережевих пакетів, набір інструментів dsniff для перехоплення паролів у незашифрованому трафіку.

Перейдемо до встановлення та налаштування системи виявлення вторгнень Suricata. Для початку, командою sudo apt install suricata -y, завантажимо саму систему для аналізу мережевого трафіку (рис. Б.25). Виконавши sudo suricata -i eth1 -c /etc/suricata/suricata.yaml -D (рис. Б.26), запускаємо IDS-систему, вказуємо мережевий інтерфейс для моніторинг (eth1), шлях до конфігураційного файлу (/etc/suricata/suricata.yaml) та запуск у фоновому режимі (-D). Командою ps aux | grep suricata перевіряємо результат виконаних дій (рис. Б.27).

Для графічного виводу логів записаних Suricata, необхідно встановити додаткові інструменти для обробки та візуалізації результатів. Оновлюємо систему (sudo apt update) і виконуємо команду sudo apt install elasticsearch kibana logstash -y (рис. Б.28) для встановлення всіх необхідних утиліт. Далі запускаємо всі служби, використовуючи sudo systemctl enable –now elasticsearch (аналогічно для інших)(рис. Б.29). Відкриваємо конфігураційний файл logstash (sudo nano /etc/logstash/conf.d/suricata.conf)(рис. Б.30) та налаштовуємо (рис. Б.31): input {file{…}} – задаємо лог-файл, який буде читати logstash; output {elasticsearch{…}} – оброблені логи будуть надсилатися на Elasticsearch, який працює на localhost:9200. Перезапускаємо Logstash (sudo systemctl restart logstash).

Після успішного налаштування всіх інструментів можна відкрити GUI для моніторингу мережевого трафіку. Заходимо на Kali Linux в браузер та відкриваємо посилання http://localhost:5601/, бачимо інтерфейс Kibana (рис. Б.32). Далі необхідно перейти в Kibana > Stack Management > Index patterns та створити новий шаблон індексу для перегляду подій із Suricata (рис. Б.33). У полі «Name» вказуємо «suricata-*» (для пошуку відповідних записів в лог файлі), у полі «Timestamp field» вказуємо «@timestamp» (це поле відповідає за побуду графіків і фільтрації даних за часом в Kibana). Після завершення налаштувань переходимо до Kibana > Discover

та обираємо створений раніше шаблон (рис. Б.34). Тепер можна виконувати моніторинг всього трафіку, що проходить мережею, в режимі реального часу.

Останній інструмент, який необхідно встановити для проведення повноцінної оцінки захищеності мережі та мережевого з'єднання, це сканер вразливостей Nessus Essentials. Завантажуємо інсталяційний .deb файл з офіційного сайту Tenable (рис. Б.35). Розпаковуємо та встановлюємо Nessus за допомогою команди sudo dpkg -i Nessus-10.8.4-debian10_amd64.deb (рис. Б.36). Запускаємо службу (sudo systemctl start nessusd) та встановлюємо для неї автозапуск (sudo systemctl enable nessusd) (рис. Б.37). Після успішного виконання всіх дій, заходимо в браузер на Kali Linux та переходимо за посиланням http://localhost:8834/, входимо в обліковий запис Nessus та бачимо веб-інтерфейс для виконання повного сканування вразливостей в мережі (рис. Б.38).

3.2 Практична частина оцінки захищеності мереж: сканування вразливостей мережі, проведення тестування на проникнення, моніторинг трафіку

Практична частина оцінки захищеності мережі та мережевого з'єднання складається з декількох етапів: сканування вразливостей для виявлення слабких місць в службах, ПЗ та налаштуваннях системи; тестування на проникнення, а саме експлуатація знайдених вразливостей для перевірки можливості отримання несанкціонованого доступу до даних, що передаються в мережі; моніторинг мережі IDS-системою для виявлення підозрілих дій в мережі. Ці етапи потрібно виконувати послідовно для отримання точної картини стану безпеки системи.

3.2.1 Комплексне сканування вразливостей мережі

Спочатку необхідно провести комплексне сканування вразливостей створеної тестової мережі. Для цього використовуємо заздалегідь встановлений інструмент, а саме Nessus Essential. Переходимо за посиланням http://localhost:8834 та створюємо нове сканування (натискаємо на кнопку «New scan»), обираємо опцію розширеного сканування (Advanced scan)(рис. Б.39). Налаштовуємо необхідні параметри, задаємо назву «Full network scan», додаємо короткий опис, задаємо ціль для нашого сканування, крім головного хосту Kali Linux, тобто IP-адреси всіх

вузлів в мережі «192.168.56.31 – 192.168.56.39» (рис. Б.40). Далі заходимо у вкладку «Credentials» та надаємо Nessus SSH облікові дані для автентифікованого сканування (метод автентифікації: пароль) хостів через SSH-протокол (рис.Б.41). Переходимо в Settings > Port Scanning (налаштування виявлення відкритих портів) та додаємо TCP-сканування портів, оскільки більшість сервісів працює саме за цим протоколом (рис. Б.42).

Після завершення налаштування, зберігаємо параметри сканування. Новий шаблон з'явиться в папці «My Scans», запускаємо його кнопкою «Launch»(рис. Б.43). Програма повідомить про час завершення сканування та наддасть детальний звіт про результати його виконання (рис. Б.44). Він містить інформацію про всі виявлені вразливості для кожного хосту в мережі (192.168.56.33 – 51, 192.168.56.31 – 23, інші – від 19 до 23) з визначеним рівнем критичності для кожної з них (базуючись на системі оцінки CVSS 3, оцінки варіюються від синього кольору – некритичні до червоного кольору – критичні, які потребують першочергової уваги). Для подальшого детального аналізу експортуємо результати сканування у вигляді PDF-файлу (приклад звіту наведено у додатку В). Аналізуючи результати сканування можна пообачити, що Nessus точно виявив використання HTTP-серверу Apache на хості 192.168.56.31 та сервісу FTP (зокрема вразливий vsftpd) на хості 192.168.56.32. Майже на всіх хостах було виявлено відкриті TCP-порти та можливість віддаленого розпізнавання ОС та сервісів. Наявність цих вразливостей створює потенційну поверхню для реалізації атак типу MITM та DoS.

3.2.2 Тестування на проникнення: ARP-spoofing, DoS атака

Після успішного сканування вразливостей в мережі і виявлення можливих векторів атаки, наступним етапом виконується тестування на проникнення для перевірки реальної можливості їх експлуатації. В нашому випадку доцільним є моделювання атак типу «людина посередині» (MITM) та відмови в обслуговуванні (DoS) як найбільш критичних для мережевої архітектури тестового стенду. Для реалізації зазначених атак використаємо хост з Kali Linux (атакуючий) та кілька вразливих віртуальних машин (FTP-, HTTP-сервер та ВМ, що імітують звичайних користувачів).

Для атаки MITM використовуємо хост Kali (192.168.56.10) як активного перехоплювача трафіку, а в якості жертв – ubuntu6 (192.168.56.36) в ролі користувача, що підключається до HTTP-сервера (ubuntu1 – 192.168.56.31); ubuntu5 (192.168.56.35) в ролі користувача, що підключається до FTP-сервера (ubuntu2 – 192.168.56.32). Для перехоплення HTTP-запитів скористаємося командою urlsnarf. Для запису перехоплених FTP-пакетів використаємо, встановлений раніше tcpdump, а для аналізу даних вбудований інструмент Wireshark.

Перехоплення HTTP-запитів:

На атакуючому хості запускаємо команду sudo urlsnarf -i eth1 (рис. Б.45) для прослуховування мережевого інтерфейсу eth1 (який підключений до локальної мережі), цей інструмент перехоплює HTTP-запити на портах 80, 8080 або 3128. Далі ініціюємо HTTP GET-запит від хосту ubuntu6 (vagrant ssh ubuntu6), використовуючи командний інструмент curl http://192.168.56.31/ (рис. Б.46). Після отримання відповіді від HTTP-сервера, переходимо на Kali та бачимо, що HTTP-запит був успішно перехоплений (рис. Б.47). Результати атаки чітко засвідчили наявність суттєвої вразливості в мережі, пов’язаної з передачею HTTP-трафіку у відкритому вигляді. Зловмисник без автентифікації може отримати доступ до всіх веб-ресурсів, які відкриває користувач мережі.

Перехоплення FTP-пакетів методом ARP-spoofing (підміна ARP-відповідей):

На атакуючому хості Kali, командою sudo tcpdump -i eth1 -w sniff.pcap ‘host 192.168.56.32 and host 192.168.56.35’ (рис. Б.48), запускаємо tcpdump для перехоплення мережевого трафіку, який проходить через інтерфейс eth1 між хостами 192.168.56.32 та 192.168.56.35. Отримані дані будуть записуватися в файл sniff.pcap. Далі використовуємо метод ARP-spoofing, виконавши команди sudo arpspoof -i eth1 -t 192.168.56.35 192.168.56.32 та sudo arpspoof -i eth1 -t 192.168.56.32 192.168.56.35 (рис. Б.49), Kali надсилає підроблені ARP-запити до обох хостів в яких зазначено, що атакуючий хост і є отримувачем пакетів. В результаті таких дій, весь трафік проходить через нього, перш ніж досягти адресата. З хосту користувача (ubuntu5) підключаємося до FTP-сервера (ftp 192.168.56.32) та вводимо облікові дані (рис. Б.50). Після успішного підключення, переходимо до Kali Linux,

зупиняємо всі процеси та відкриваємо файл sniff.pcap через Wireshark (рис. Б.51). Аналізуючи перехоплені FTP-пакети ми можемо чітко побачити облікові дані користувача (логін та пароль) у незашифрованому вигляді. Результати атаки показують вразливість мережі до атак MITM методом ARP-spoofing, зловмисник може легко отримати конфіденційну інформацію, що передається мережею через FTP-протокол.

Фінальним тестом на проникнення є моделювання атаки відмови в обслуговуванні (DoS) на один з хостів мережі. Спочатку зафіксуємо початковий стан цільової машини (ubuntu1 – 192.168.56.31) для подальшого порівняння (рис. Б.52). Реалізуємо класичну SYN flood атаку, виконавши команду sudo hping3 -S -p 80 –flood 192.168.56.31 (рис. Б.53). В даному випадку на IP-адресу цільового вузла (порт 80) надсилається велика кількість TCP-пакетів з встановленим SYN-бітом (для імітації початку з’єднання, --flood надсилає пакети з мінімальною затримкою без очікування відповіді хосту. Виконавши команду top на ubuntu1 (рис. Б.54) можна побачити, що споживання ресурсів процесом ksoftirqd/0 (оброблює програмні переривання в Linux) виросло до максимуму, це є прямим наслідком реалізації DoS атаки. Результати тесту демонструють, що в мережі відсутні механізми обмеження частоти запитів та аномальної активності, це вказує на низьку стійкість мережі до атак відмови в обслуговуванні.

3.2.3 Результати моніторингу мережевого трафіку

В процесі виконання практичної оцінки захищеності мережі, важливою складовою є моніторинг стану для виявлення підозрілої активності та оцінки ефективності систем виявлення загроз. Після завершення тестування на проникнення в мережі, було виконано моніторинг мережевого трафіку за допомогою системи Suricata. Аналізуючи журнали подій за ключовими протоколами (HTTP, FTP)(рис. Б.55-Б.56), IDS-система не виявила перехоплення пакетів методом ARP-Spoofing, це може свідчити про:

- Відсутність відповідних сигнатур у правилах Suricata
- Неналаштовану реакцію системи на підозрілу ARP-активність.

Пошук за типами flow та anomaly (рис. Б.57) показав понад 86000 подій за короткий проміжок часу, що свідчить про аномально високу активність в мережі, спричинену SYN flood атакою. В цьому випадку Suricata згенерувала подію alert.action, тобто система моніторингу успішно виявила на DoS атаку.

Таким чином можна зробити висновок, що система моніторингу виявила лише частину виконаних атак на мережу. Це підкреслює необхідність правильного налаштування правил та створення спеціалізованих сигнатур для виявлення різних типів атак та підозрілої активності в мережі.

3.3 Узагальнення отриманих результатів, аналіз виявлених вразливостей та оцінка їх критичності

В результаті проведення комплексного сканування мережі з використання інструменту Nessus Essentials, можна сформувати загальну картину стану безпеки всіх активних хостів в тестовій мережі. Аналіз віртуальних машин у підмережі 192.168.56.0/24 виявив велику кількість вразливостей, які ранжуються від найменш значущих (інформаційних) до найбільш критичних (високий рівень загрози для безпеки мережі).

Найбільш критичним виявився вузол 192.168.56.33. На ньому було виявлено використання застарілого ядра Linux у якому існує відома вразливість (USN-7516-1), за системою CVSS 3 отримала оцінку 7.8/10 (дуже високий показник критичності), що потенційно дозволяє неавторизованому користувачеві виконувати довільний код. На хостах 192.168.56.31 та 192.168.56.32 було виявлено використання протоколів без шифрування даних таких як, HTTP та FTP відповідно. Наявність таких вразливостей робить ці вузли першочерговою ціллю для атак типу «людина посередині» (MITM). Майже на всіх хостах було виявлено відкриті TCP-порти та можливість віддаленого розпізнавання ОС та сервісів, що робить мережу вразливою для атак типу відмова в обслуговуванні(DoS/DDoS). Відсутність системного управління та слабка сегментація тестової мережі теж сприяють підвищенню ризиків. Ці вразливості створюють цілу низку векторів для атак потенційних зовнішніх та/або внутрішніх зловмисників.

Аналізуючи результати виконаного тестування на проникнення можна побачити, що мережа має слабку стійкість до атак типу «людина посередині» та відмови в обслуговуванні. Це відповідає даним отриманим в процесі сканування вразливостей. Зловмисник зміг легко отримати доступ до конфіденційної інформації користувача, яка передавалася в мережі.

Система моніторингу мережевого трафіку не змогла виявити всі атаки потенційного зловмисника, що свідчить про неправильне або недостатнє налаштування інструментів виявлення підозрілої активності та кібератак в мережі.

Таким чином, результати проведення комплексної оцінки захищеності мережі та мережевого з'єднання демонструють наявність, в тестовій мережі, значних прогалин в безпеці. Наявність відкритих портів, використання протоколів без шифрування даних, відсутність оновлень та засобів обмеження трафіку в мережі створює умови для проведення успішних кібератак з боку потенційних зловмисників. Слабке налаштування системи моніторингу трафіку не дає можливості вчасно виявити та зреагувати на інциденти порушення мережевої безпеки.

3.4 Оцінка ризиків та наслідків, рекомендації щодо покращення інформаційної безпеки мережі

Після проведення оцінки захищеності мережі було виявлено понад 210 вразливостей різних видів на всіх мережевих вузлах. Виконуючи тестування в ролі потенційного зловмисника, вдалося експлуатувати деякі з виявлених вразливостей та отримати несанкціонований доступ до конфіденційної інформації користувачів в мережі. Це говорить про слабкий захист даних в мережі та про високий рівень ризику, пов'язаний як із внутрішніми, так і зовнішніми загрозами. Основні ідентифіковані ризики для тестової моделі мережі

- Ескалація привілеїв через використання застарілих версій ядра Linux, що можуть дозволити потенційному зловмиснику отримати розширені повноваження.
- Виконання довільного коду через бібліотеку Log4J, яка містить в собі відому вразливість для віддаленого запуску команд.

- Витік конфіденційної інформації через протоколи без шифрування даних. Наявність FTP-сервера та застарілих SSH-алгоритмів підвищує ймовірність перехоплення даних зловмисниками.
- Витік технічної інформації через відкриті порти на всіх вузлах в мережі.

Ці ризики можуть призвести до таких потенційних наслідків:

- Втрати конфіденційної інформації користувачів та системних облікових даних.
- Порушення роботи вузлів, які обслуговують критично важливі служби.
- Збільшення площі атаки внаслідок неналежної сегментації.
- Нестабільність та недоступність мережі, якщо вразливості будуть використані для DoS атак.

Враховуючи кількість та критичність виявлених вразливостей, доцільним є впровадження комплексного підходу до покращення інформаційної безпеки мережі. Першочерговим кроком є усунення найнебезпечніших виявлених проблем, а саме вразливість ядра Linux (USN-7516-1, CVSS 3: 7.8) та застарілої бібліотеки Log4J. Необхідно оновити ядро системи до останньої стабільної версії та повністю видалити або оновити вразливу бібліотеку.

Постійно оновлювати систему та програмне забезпечення, регулярно проводити автоматизовані сканування вразливостей з використанням інструменту Nessus Essentials. Такий підхід дозволить унеможливити використання давно відомих вразливостей потенційними зловмисниками.

Також важливою рекомендацією є впровадження повноцінної системи контролю доступу в мережу. Встановити брандмауери з визначеними правилами доступу на всіх вузлах, заборонити доступ до служб, що передають дані у відкритому вигляді.

Для збільшення стійкості мережі до атак типу «людина посередині», необхідно впровадити використання лише сучасних протоколів передачі даних. Першочергово замінити застарілі FTP та HTTP їх покращені версії з функцією шифрування даних та безпечної передачі інформації всередині мережі таких як

SFTP та HTTPS. Вимкнути підтримку SHA-1 та інших слабких алгоритмів у всіх конфігураціях служб (зокрема SSH).

Заключною рекомендацією щодо покращення інформаційної безпеки мережі є налаштування централізованого логування системних та мережевих подій з усіх активних хостів у систему збору логів. Задати правила та сигнатури для швидкого та точного виявлення загроз система IDS/IPS. Створити базові політики реагування на інциденти.

Загалом, реалізація вказаних рекомендацій дозволить значно підвищити рівень інформаційної безпеки тестового середовища, знизить ризики несанкціонованого доступу, витоку даних та порушення цілісності мережевої інфраструктури. Такий підхід відповідає принципам сучасних стандартів безпеки та може бути масштабований для реалізації більш складних мереж.

ВИСНОВКИ

У процесі виконання дипломної роботи було детально досліджено актуальні загрози для комп’ютерних мереж, проаналізовано тенденції розвитку кібербезпеки та ключові стандарти захисту інформації. Було встановлено тісний зв’язок між технологічним прогресом і зростанням складності кібератак, що обумовлює постійну актуальність даної теми. Зокрема, впровадження штучного інтелекту, блокчейн-технологій і концепції Zero Trust суттєво підвищує ефективність сучасних засобів та підходів до кіберзахисту мережі.

Проведено детальний аналіз типів мереж, критеріїв оцінки безпеки та поширених вразливостей. Серед існуючих інструментів було обрано найбільш ефективні, було також обрано відповідні методи тестування, які використовувалися в практичній частині роботи.

Було спроєктовано й розгорнуто тестове середовище, що імітує реальну локальну мережу з декількох пристроїв. Налаштовано ключові вузли та інструменти для сканування вразливостей, тестування на проникнення та моніторингу мережевого трафіку. Проведено імітацію атак, зокрема MITM та DoS, а всі події зафіксовано засобами системи виявлення вторгнень для подальшого аналізу.

Результати оцінки захищеності засвідчили низький рівень стійкості мережі до атак: виявлено значну кількість критичних вразливостей, які в реальних умовах можуть спричинити втрату даних і порушення цілісності мережевої інфраструктури. На основі проведеного аналізу було сформовано низку рекомендацій для підвищення загального рівня інформаційної безпеки мережі.

Таким чином, усі завдання роботи були успішно виконані. Отримані результати підтверджують важливість регулярної оцінки захищеності мережі для вчасного виявлення загроз і запобігання їх негативним наслідкам в разі експлуатації потенційними зловмисниками.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Cyber security threats - all you need to know about types and sources – [Електронний ресурс]. Режим доступу: https://www.dataguard.com/cyber-security/threats/
2. +95 Cyber Security Breach Statistics 2025 – [Електронний ресурс]. Режим доступу: https://www.stationx.net/cyber-security-breach-statistics/
3. HUMAN FACTORS IN CYBERSECURITY: RISKS AND IMPACTS – [Електронний ресурс]. Режим доступу: https://zagrebsecurityforum.com/Portals/0/SecurityScienceJournal/SSJ%202_2_4%20HUMAN%20FACTORS%20IN%20CYBERSECURITY%20RISKS%20AND%20IMPACTS.PDF
4. Artificial Intelligence and Machine Learning in Cyber Security AI & ML in Cyber Security – [Електронний ресурс]. Режим доступу: https://www.aalpha.net/articles/artificial-intelligence-and-machine-learning-in-cyber-security/
5. AI in Cybersecurity: Key Benefits, Defense Strategies, & Future Trends – [Електронний ресурс]. Режим доступу: https://www.fortinet.com/resources/cyberglossary/artificial-intelligence-in-cybersecurity
6. IMPLEMENTING A ZERO TRUST ARCHITECTURE – [Електронний ресурс]. Режим доступу: https://www.nccoe.nist.gov/sites/default/files/legacy-files/zta-project-description-final.pdf
7. What is Zero Trust Architecture (ZTA)? – [Електронний ресурс]. Режим доступу: https://www.paloaltonetworks.com.au/cyberpedia/what-is-a-zero-trust-architecture
8. Blockchain in Cybersecurity: Applications and Challenges – [Електронний ресурс]. Режим доступу:

https://globalcybersecuritynetwork.com/blog/blockchain-in-cybersecurity-applications-and-challenges/

9. How can you use blockchain for network security? – [Електронний ресурс]. Режим доступу: https://www.linkedin.com/advice/0/how-can-you-use-blockchain-network-security-skills-computer-science
10. Cyber Crime Statistics and Trends – [Електронний ресурс]. Режим доступу: https://www.go-globe.com/cyber-crime-statistics-and-trends-infographic/
11. Types Of Cyber Attacks – [Електронний ресурс]. Режим доступу: https://www.fortinet.com/resources/cyberglossary/types-of-cyber-attacks
12. SolarWinds hack explained: Everything you need to know – [Електронний ресурс]. Режим доступу: https://www.techtarget.com/whatis/feature/SolarWinds-hack-explained-Everything-you-need-to-know
13. The Mirai Botnet - Threats and Mitigations – [Електронний ресурс]. Режим доступу: https://www.cisecurity.org/insights/blog/the-mirai-botnet-threats-and-mitigations
14. What is Log4Shell? The Log4j vulnerability explained (and what to do about it) – [Електронний ресурс]. Режим доступу: https://www.dynatrace.com/news/blog/what-is-log4shell/
15. Network security standards: Everything you need to know – [Електронний ресурс]. Режим доступу: https://nordvpn.com/ru/blog/network-security-standards/
16. What are Confidentiality, Integrity and Availability in Information Security? – [Електронний ресурс]. Режим доступу: https://vinciworks.com/blog/what-are-confidentiality-integrity-and-availability-in-information-security/
17. Network Resilience vs Redundancy vs Backups – [Електронний ресурс]. Режим доступу:https://zpesystems.com/network-resilience-vs-redundancy-zs/
18. How to Perform a Network Security Risk Assessment in 6 Steps – [Електронний ресурс]. Режим доступу: https://www.algosec.com/blog/network-security-risk-assessment

19. Types of Networks: LAN, WAN, MAN & More Explained – [Електронний ресурс]. Режим доступу: https://www.simplilearn.com/tutorials/networking-tutorial/importance-of-types-of-networks-lan-man-wan
20. Understanding Different types of networking services – [Електронний ресурс]. Режим доступу: https://blog.lumen.com/understanding-different-types-of-networking-services/
21. Network types decrypted: Your comprehensive guide to WAN, LAN, WLAN, VLAN, CAN, GAN, MAN, PAN and VPN – [Електронний ресурс]. Режим доступу: https://blog.it-planet.com/en/network-types-decrypted-your-comprehensive-guide-to-wan-lan-wlan-vlan-can-gan-man-pan-and-vpn/
22. Common Network Security Vulnerabilities – [Електронний ресурс]. Режим доступу: https://www.cobalt.io/blog/common-network-security-vulnerabilities
23. What is rate limiting and how does it work? – [Електронний ресурс]. Режим доступу: https://www.radware.com/cyberpedia/bot-management/rate-limiting/
24. What is network penetration testing? – [Електронний ресурс]. Режим доступу: https://www.ibm.com/think/topics/network-penetration-testing
25. 5 Most Popular Penetration Testing Tools In 2025 – [Електронний ресурс]. Режим доступу: https://loadfocus.com/blog/comparisons/penetration-testing-tools/
26. Top 10 Network Security Monitoring Tools for 2025 – [Електронний ресурс]. Режим доступу: https://signoz.io/comparisons/network-security-monitoring-tools/
27. What is Vulnerability Scanning? – [Електронний ресурс]. Режим доступу: https://www.balbix.com/insights/what-is-vulnerability-scanning/
28. Top Vulnerability Assessment tools for 2024 – [Електронний ресурс]. Режим доступу: https://www.secopsolution.com/blog/top-vulnerability-assessment-tools-for-2023
29. IDS vs. IPS: Definitions, Comparisons & Why You Need Both – [Електронний ресурс]. Режим доступу: https://www.okta.com/identity-101/ids-vs-ips/

ДОДАТОК А

ВИХІДНИЙ КОД КОНФІГУРАЦІЙНОГО ФАЙЛУ VAGRANT

```
Vagrant.configure("2") do |config|
    # 1. Налаштування за замовчуванням
    config.vm.box_check_update = false # не перевіряти оновлення боксів при vagrant up
    config.vm.synced_folder '.', '/vagrant', disabled: true # вимкнути синхронізацію
поточної папки
    # 2. Визначення вузла Kali (атакувальний хост)
    config.vm.define "kali" do |kali|
        kali.vm.box = "kalilinux/rolling"
    # офіційний образ Kali Linux
        kali.vm.hostname = "kali"
        kali.vm.network "private_network", ip: "192.168.56.10", netmask: "255.255.255.0",
virtualbox__intnet: "intnet"
        # Ресурси для Kali
        kali.vm.provider "virtualbox" do |vb|
            vb.name = "lab_kali"
            vb.memory = 4096
            vb.cpus = 5
        end
    end

    # 3. Визначення 9 Ubuntu-хостів внутрішньої мережі
    (1..9).each do |i|
    config.vm.define "ubuntu#{i}" do |node|
        node.vm.box = "ubuntu/focal64"
        node.vm.hostname = "ubuntu#{i}"
        # Приватна мережа (Host-Only) з присвоєнням IP для кожного хоста
        node.vm.network "private_network", ip: "192.168.56.#{30 + i}", netmask:
"255.255.255.0", virtualbox__intnet: "intnet"
        node.vm.provider "virtualbox" do |vb|
            vb.name = "lab_ubuntu#{i}"
            vb.memory = 512
            vb.cpus = 1
            end
        end
    end
end
```

ДОДАТОК Б

СКРІНШОТИ ВИКОНАННЯ ПРАКТИЧНОЇ ЧАСТИНИ ДОСЛІДЖЕННЯ

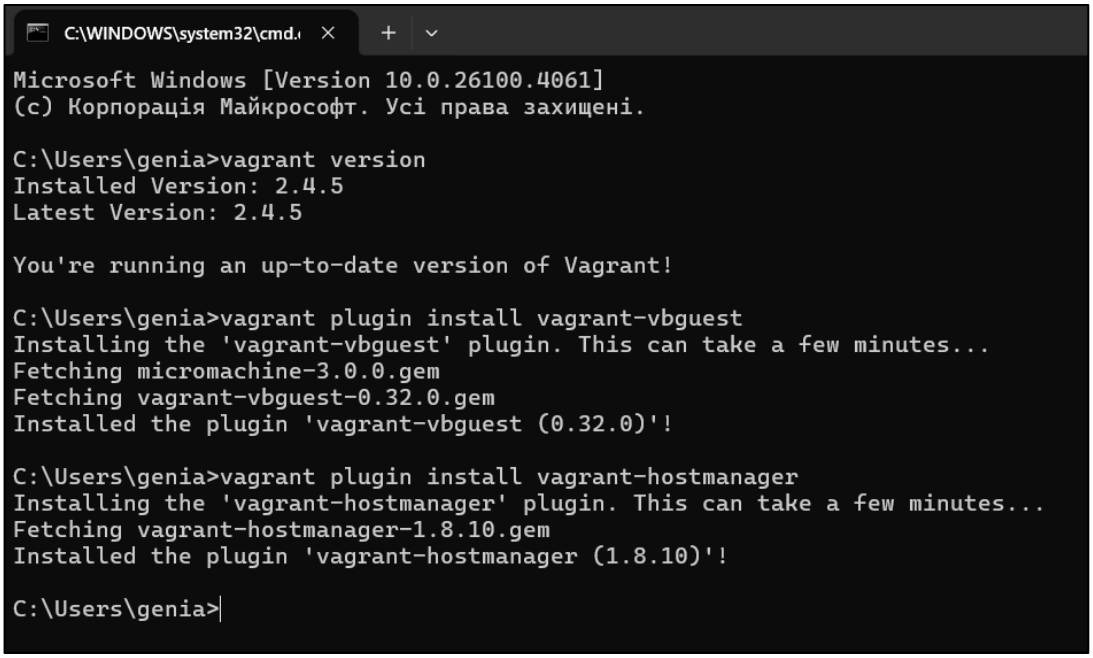


Рисунок Б.1 – Встановлення інструменту Vagrant

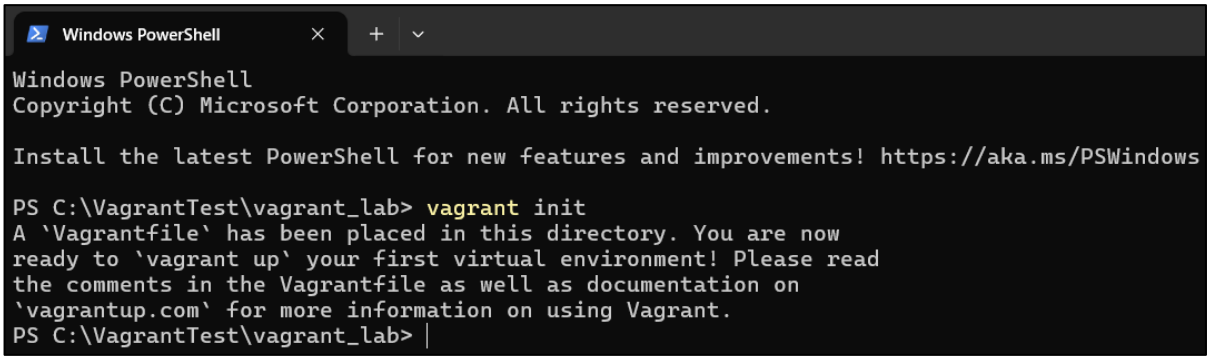


Рисунок Б.2 – Ініціалізація vagrant середовища

Vagrantfile

```
Vagrant.configure("2") do |config|
    # 1. Налаштування за замовчуванням
    config.vm.box_check_update = false # не перевіряти оновлення боксів при vagrant up
    config.vm.synced_folder '.', '/vagrant', disabled: true # вимкнути синхронізацію поточної папки
    # 2. Визначення вузла Kali (атакувальний хост)
    config.vm.define "kali" do |kali|
        kali.vm.box = "kalilinux/rolling"
    # офіційний образ Kali Linux
        kali.vm.hostname = "kali"
        kali.vm.network "private_network", ip: "192.168.56.10", netmask: "255.255.255.0", virtualbox__intnet: "intnet"
        # Ресурси для Kali
        kali.vm.provider "virtualbox" do |vb|
            vb.name = "lab_kali"
            vb.memory = 4096
            vb.cpus = 5
        end
    end

    # 3. Визначення 9 Ubuntu-хостів внутрішньої мережі
    (1..9).each do |i|
    config.vm.define "ubuntu#{i}" do |node|
        node.vm.box = "ubuntu/focal64"
        node.vm.hostname = "ubuntu#{i}"
        # Приватна мережа (Host-Only) з присвоєнням IP для кожного хоста
        node.vm.network "private_network", ip: "192.168.56.#{30 + i}", netmask: "255.255.255.0", virtualbox__intnet: "intnet"
        node.vm.provider "virtualbox" do |vb|
            vb.name = "lab_ubuntu#{i}"
            vb.memory = 512
            vb.cpus = 1
            end
        end
    end
end
```

Рисунок Б.3 – Вихідний код конфігураційного файл

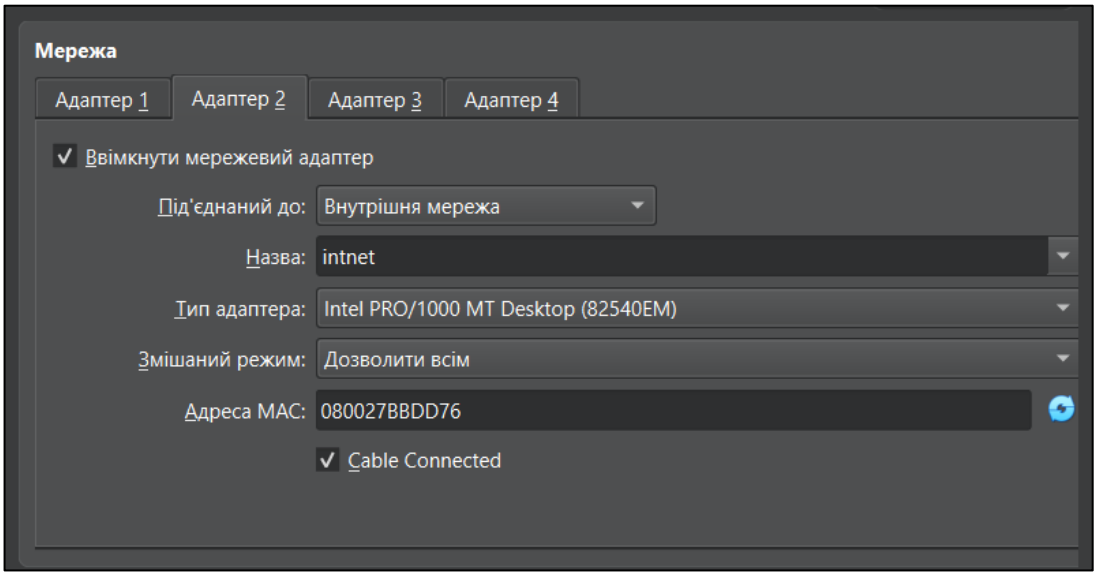


Рисунок Б.4 – Налаштування мережевого адаптеру

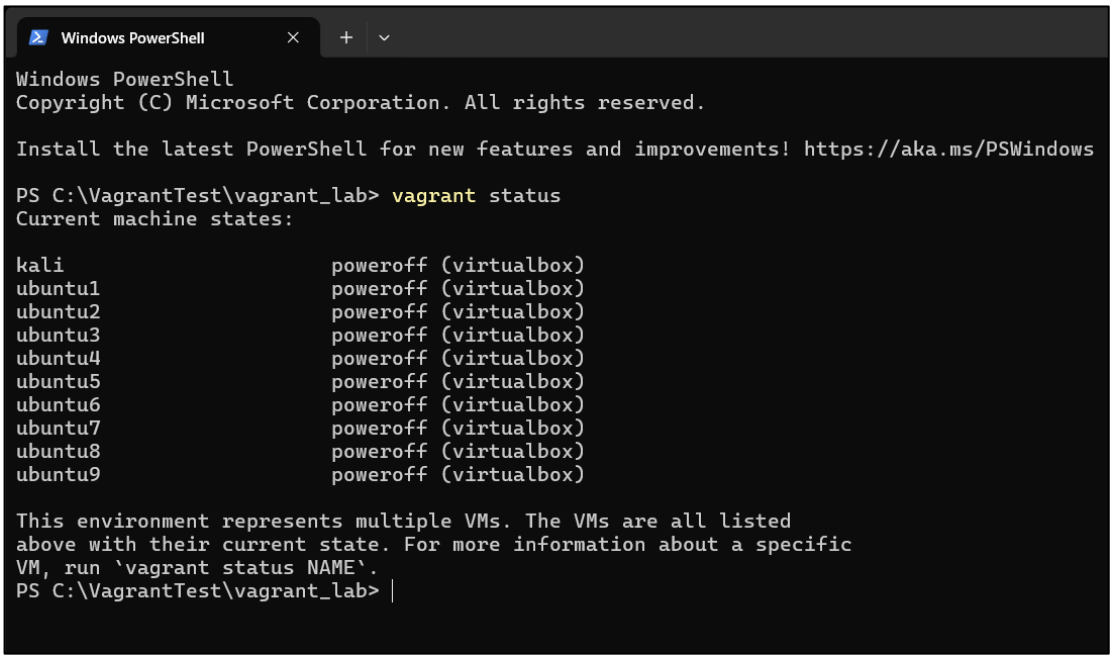


Рисунок Б.5 – Перевірка правильності налаштування конфігураційного файлу

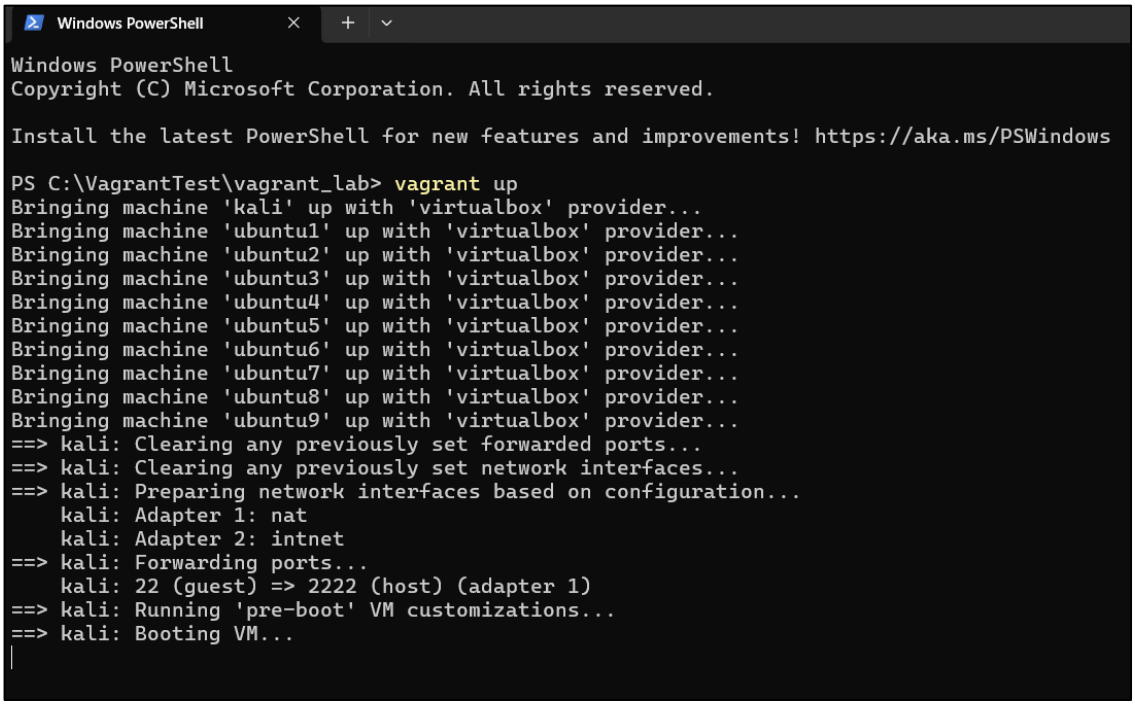


Рисунок Б.6 – Розгортання тестової мережі

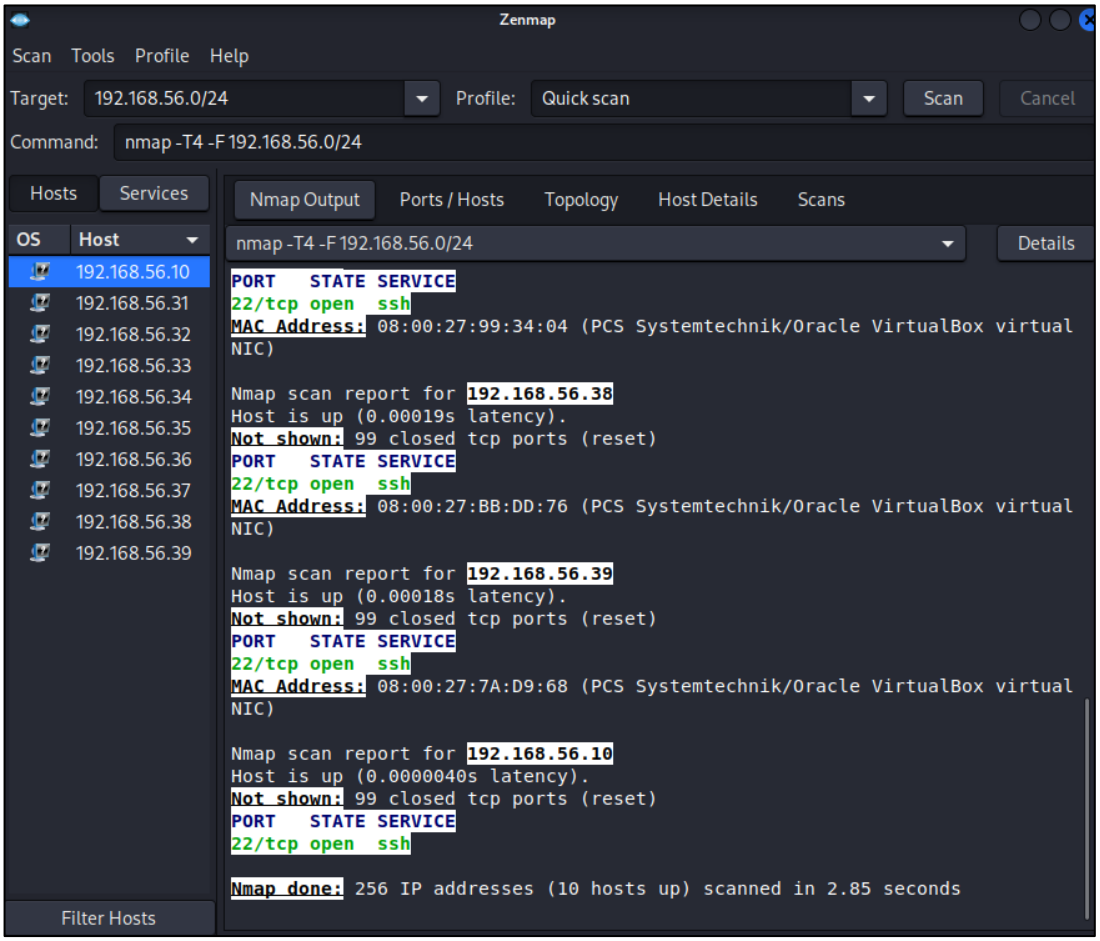


Рисунок Б.7 – Сканування підмережі 192.168.56.0/24

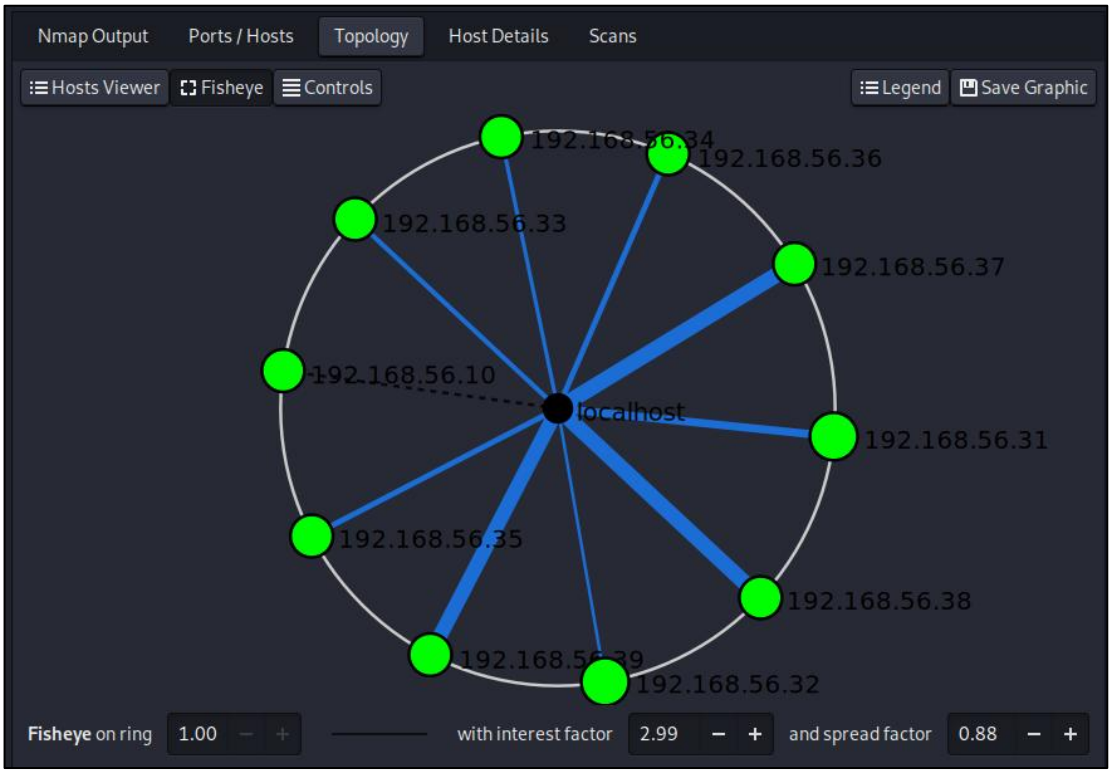


Рисунок Б.8 – Графічне представлення топології тестової мережі

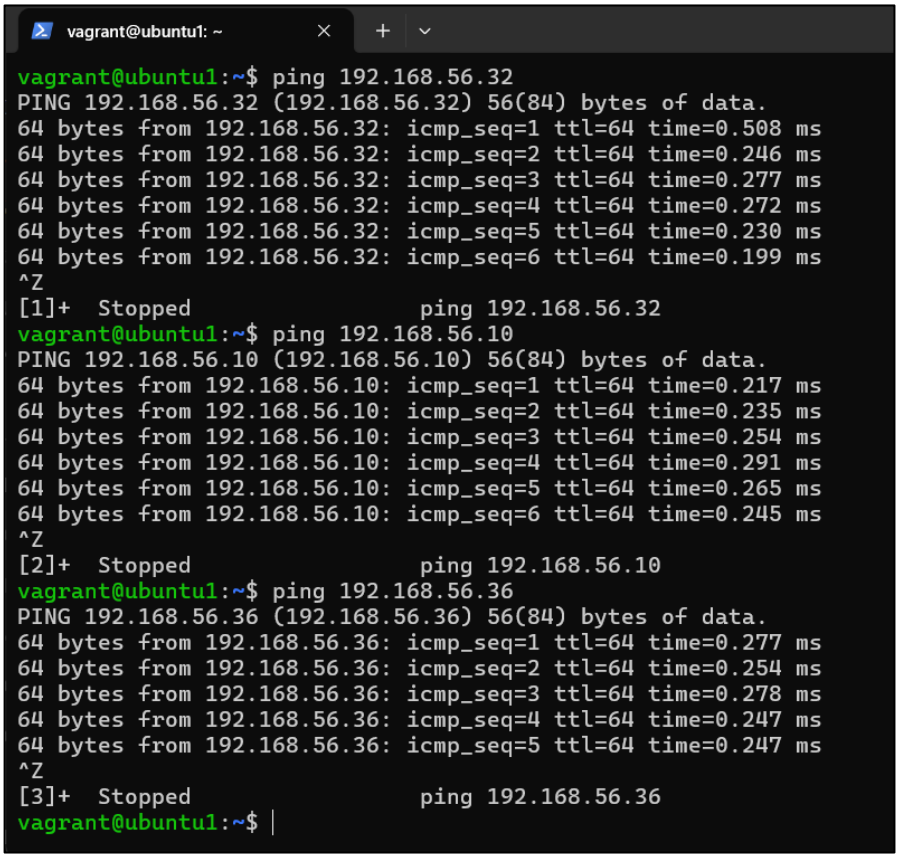


Рисунок Б.9 – Результати виконання команд ping

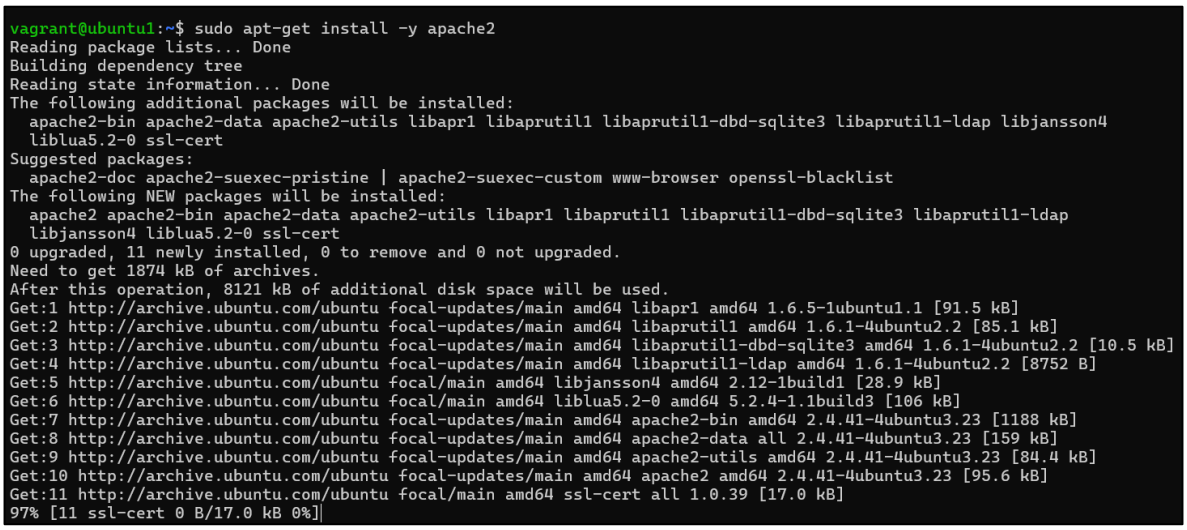


Рисунок Б.10 – Встановлення веб-сервера Apache

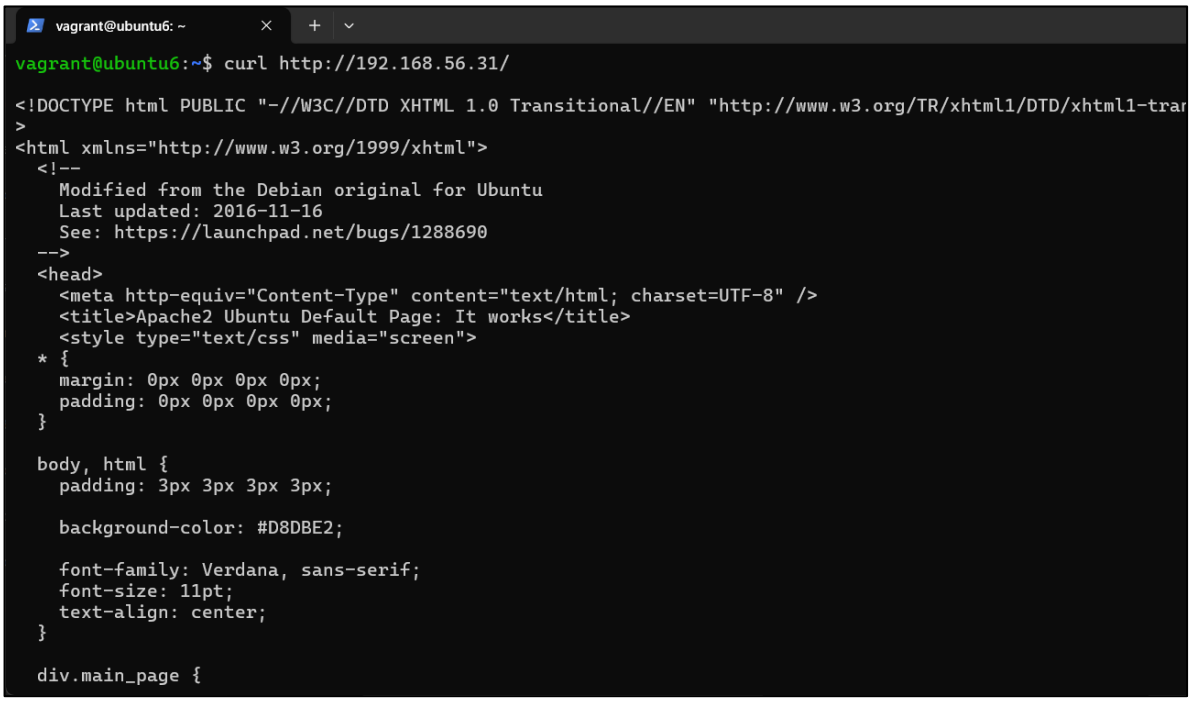


Рисунок Б.11 – Демонстрація роботи HTTP-сервера

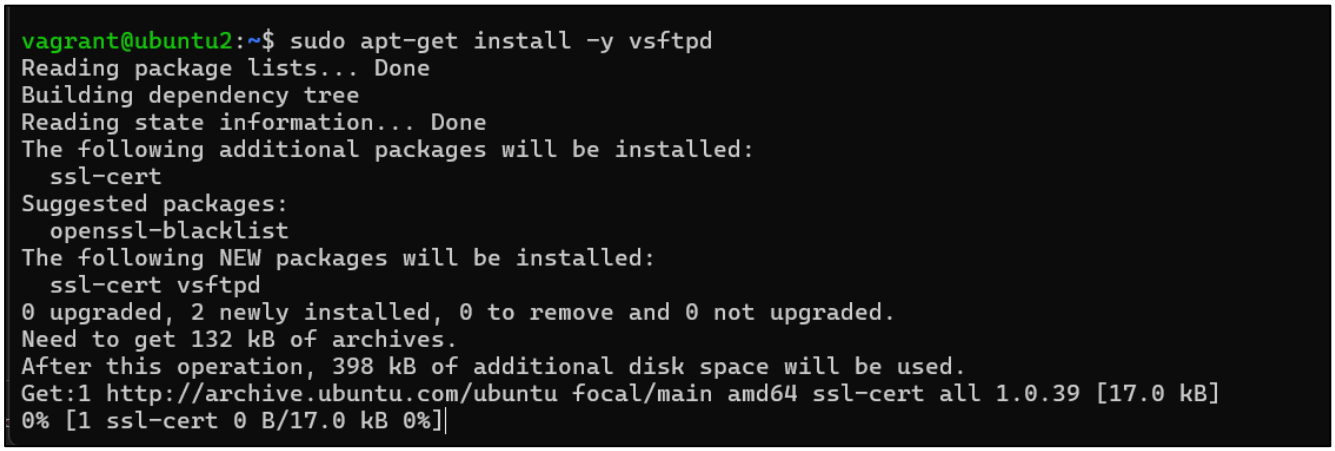


Рисунок Б.12 – Процес встановлення FTP-сервера

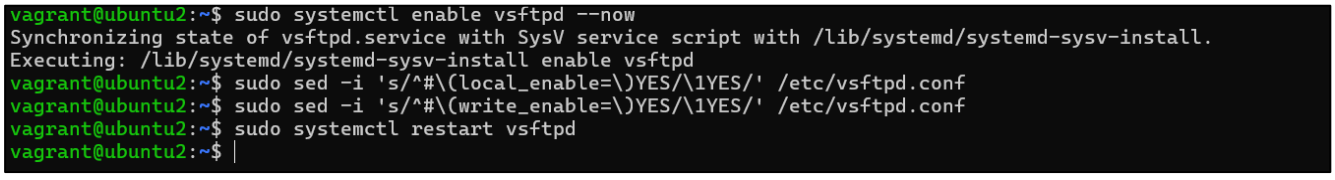


Рисунок Б.13 – Процес запуску та налаштування FTP-сервера

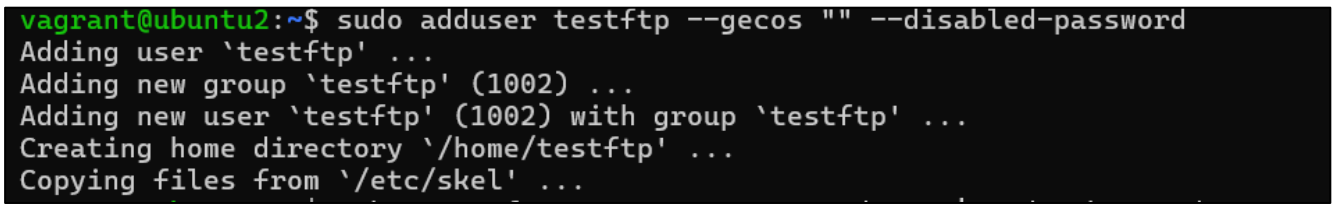


Рисунок Б.14 – Додавання нового користувача

```
vagrant@ubuntu2:~$ echo "testftp:password123" | sudo chpasswd
vagrant@ubuntu2:~$
```

Рисунок Б.15 – Додавання нового паролю користувача

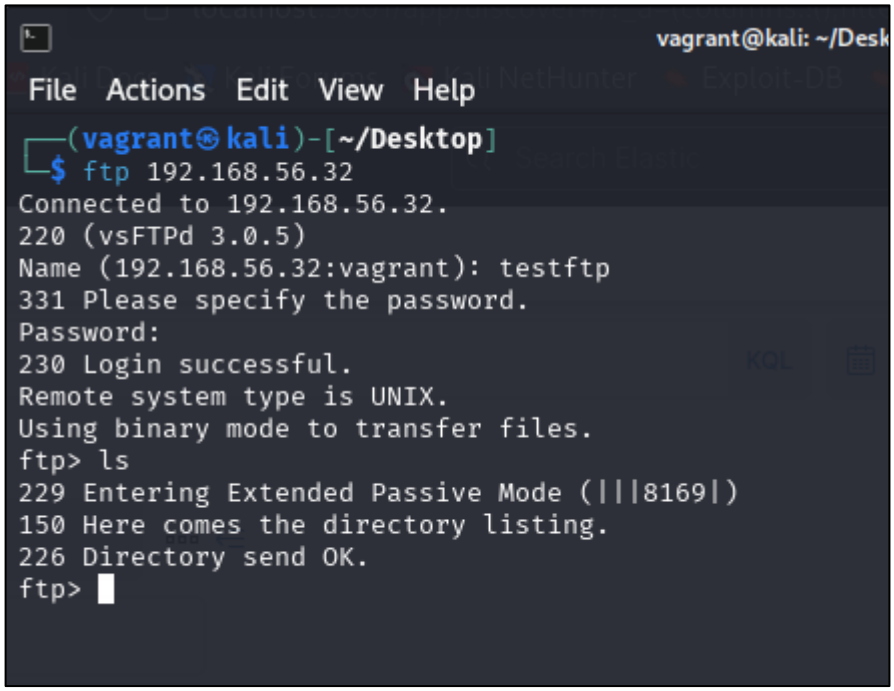


Рисунок Б.16 – Підключення до створеного FTP-сервера

```
vagrant@ubuntu3:~$ sudo apt install -y openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:8.2p1-4ubuntu0.13).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
vagrant@ubuntu3:~$
```

Рисунок Б.17 – Встановлення SFTP-сервера

```
vagrant@ubuntu3:~$ sudo adduser testsftp --gecos ""
Adding user `testsftp' ...
Adding new group `testsftp' (1002) ...
Adding new user `testsftp' (1002) with group `testsftp' ...
Creating home directory `/home/testsftp' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
vagrant@ubuntu3:~$
```

Рисунок Б.18 – Додавання нового користувача

```
vagrant@ubuntu3:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2025-05-19 06:41:45 UTC; 2h 19min ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 666 (sshd)
      Tasks: 1 (limit: 513)
     Memory: 6.6M
     CGroup: /system.slice/ssh.service
             └─666 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

May 19 06:41:44 ubuntu3 systemd[1]: Starting OpenBSD Secure Shell server...
May 19 06:41:45 ubuntu3 sshd[666]: Server listening on 0.0.0.0 port 22.
May 19 06:41:45 ubuntu3 sshd[666]: Server listening on :: port 22.
May 19 06:41:45 ubuntu3 systemd[1]: Started OpenBSD Secure Shell server.
May 19 06:41:50 ubuntu3 sshd[749]: Accepted publickey for vagrant from 10.0.2.2 port 52166 ssh2: ED25519 SHA256:GsQwjq+>
May 19 06:41:50 ubuntu3 sshd[749]: pam_unix(sshd:session): session opened for user vagrant by (uid=0)
May 19 08:58:54 ubuntu3 sshd[1586]: Accepted publickey for vagrant from 10.0.2.2 port 57377 ssh2: ED25519 SHA256:GsQwjq>
May 19 08:58:54 ubuntu3 sshd[1586]: pam_unix(sshd:session): session opened for user vagrant by (uid=0)
lines 1-19/19 (END)
```

Рисунок Б.19 – Перевірка стану сервера

```
┌──(vagrant㉿kali)-[~/Desktop]
└─$ sftp testsftp@192.168.56.33
testsftp@192.168.56.33's password:
Permission denied, please try again.
testsftp@192.168.56.33's password:
Connected to 192.168.56.33.
sftp> ls
```

Рисунок Б.20 – Підключення до SFTP-сервера

```
vagrant@ubuntu4:~$ sudo apt-get install -y ufw
Reading package lists... Done
Building dependency tree
Reading state information... Done
ufw is already the newest version (0.36-6ubuntu1.1).
ufw set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
vagrant@ubuntu4:~$
```

Рисунок Б.21 – Перевірка наявності міжмережевого екрану

```
vagrant@ubuntu4:~$ sudo ufw disable
Firewall stopped and disabled on system startup
vagrant@ubuntu4:~$ sudo ufw status verbose
Status: inactive
vagrant@ubuntu4:~$
```

Рисунок Б.22 – Вимкнення міжмережевого екрану

```
┌──(vagrant㉿kali)-[~/Desktop]
└─$ sudo apt install -y ettercap-text-only bettercap
Installing:
  bettercap  ettercap-text-only

Installing dependencies:
  bettercap-caplets

Suggested packages:
  bettercap-ui
```

Рисунок Б.23 – Встановлення Ettercap та Bettercap

```
┌──(vagrant㉿kali)-[~/Desktop]
└─$ sudo apt install -y tcpdump
tcpdump is already the newest version (4.99.5-2).
tcpdump set to manually installed.
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0

┌──(vagrant㉿kali)-[~/Desktop]
└─$ sudo apt install -y dsniff
dsniff is already the newest version (2.4b1+debian-34).
dsniff set to manually installed.
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0

┌──(vagrant㉿kali)-[~/Desktop]
└─$
```

Рисунок Б.24 – Встановлення tcpdump та dsniff

```
┌──(vagrant㉿kali)-[~/Desktop]
└─$ sudo apt install suricata -y

Installing:
  suricata

Installing dependencies:
  isa-support         librte-ip-frag25   librte-ring25
  libfdt1             librte-kvargs25    librte-sched25
  libhtp2             librte-log25       librte-telemetry25
  libhyperscan5       librte-mbuf25      libxdp1
  libnetfilter-log1   librte-mempool25   oinkmaster
  librte-bus-pci25    librte-meter25     snort-rules-default
  librte-bus-vdev25   librte-net-bond25  sse3-support
  librte-eal25        librte-net25       sse4.2-support
  librte-ethdev25     librte-pci25       suricata-update
  librte-hash25       librte-rcu25

Suggested packages:
  snort  | snort-pgsql  | snort-mysql  libtcmalloc-minimal4

Summary:
  Upgrading: 0, Installing: 30, Removing: 0, Not Upgrading: 1006
  Download size: 3,898 kB / 6,991 kB
  Space needed: 32.1 MB / 21.9 GB available
```

Рисунок Б.25 – Встановлення IDS-системи Suricata

Рисунок Б.26 – Запуск Suricata

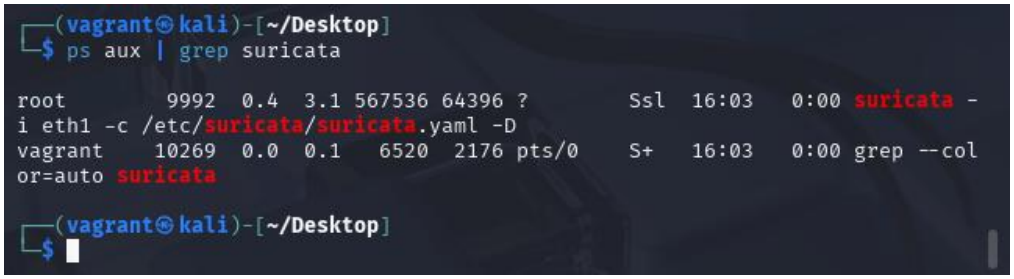


Рисунок Б.27 – Перевірка роботи системи

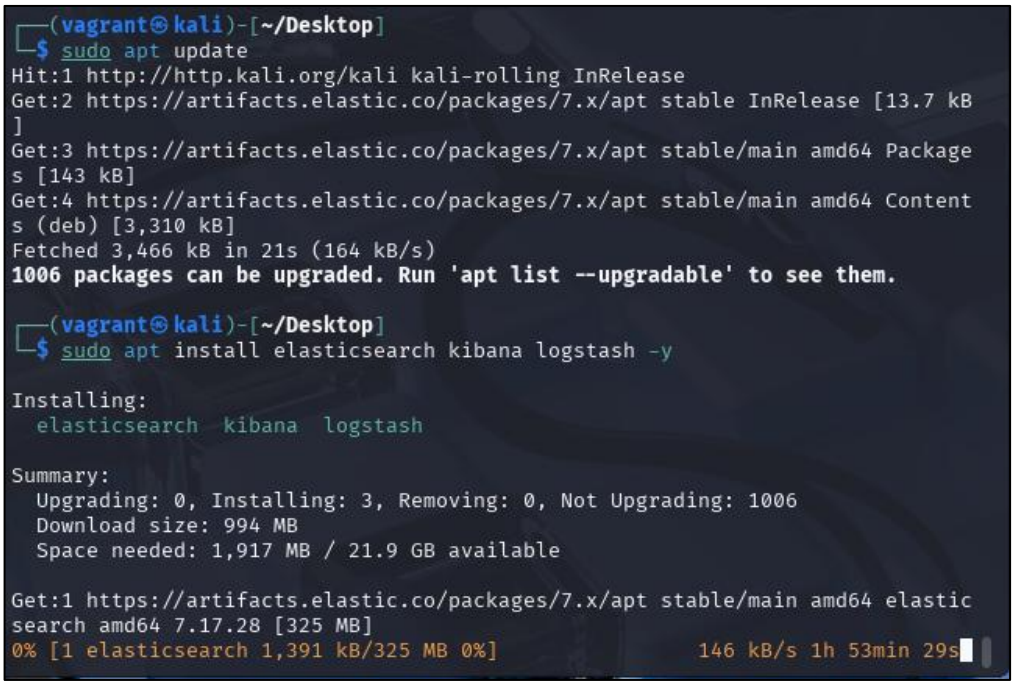


Рисунок Б.28 – Встановлення elasticsearch kibana logstash

```
┌──(vagrant㉿kali)-[~/Desktop]
└─$ sudo systemctl enable --now elasticsearch
Synchronizing state of elasticsearch.service with SysV service script with /u
sr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable elasticsearch
Created symlink '/etc/systemd/system/multi-user.target.wants/elasticsearch.se
rvice' → '/usr/lib/systemd/system/elasticsearch.service'.

┌──(vagrant㉿kali)-[~/Desktop]
└─$ sudo systemctl enable --now kibana
Synchronizing state of kibana.service with SysV service script with /usr/lib/
systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable kibana
Created symlink '/etc/systemd/system/multi-user.target.wants/kibana.service'
→ '/etc/systemd/system/kibana.service'.

┌──(vagrant㉿kali)-[~/Desktop]
└─$ sudo systemctl enable --now logstash
Created symlink '/etc/systemd/system/multi-user.target.wants/logstash.service
' → '/etc/systemd/system/logstash.service'.

┌──(vagrant㉿kali)-[~/Desktop]
└─$
```

Рисунок Б.29 – Запуск elasticsearch kibana logstash

```
┌──(vagrant㉿kali)-[~/Desktop]
└─$ sudo nano /etc/logstash/conf.d/suricata.conf

┌──(vagrant㉿kali)-[~/Desktop]
└─$ sudo systemctl restart logstash
```

Рисунок Б.30 – Відкриття конфігураційного файлу logstash

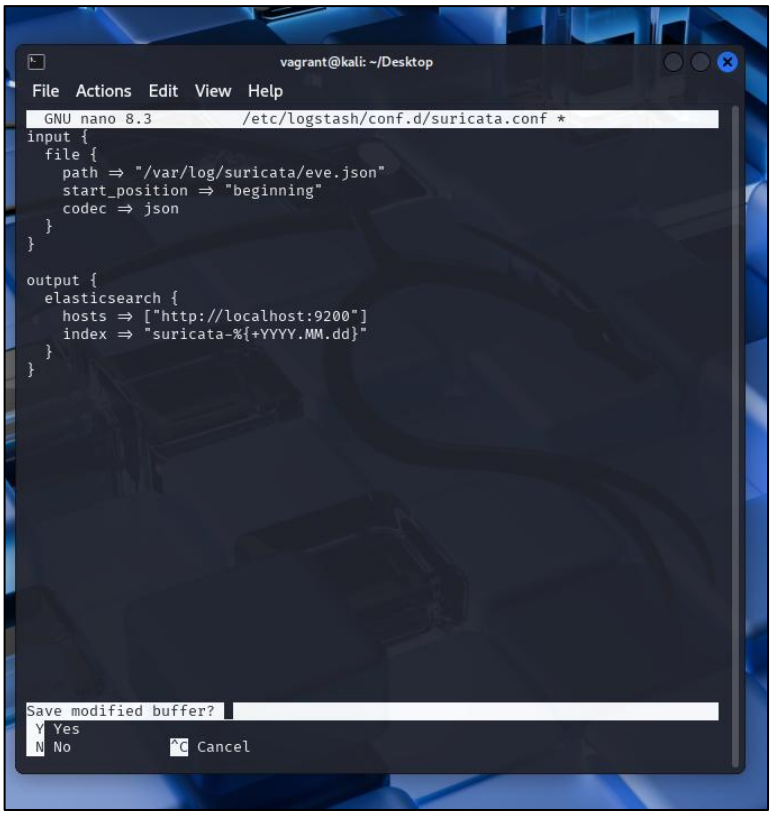


Рисунок Б.31 – Налаштування конфігураційного файлу

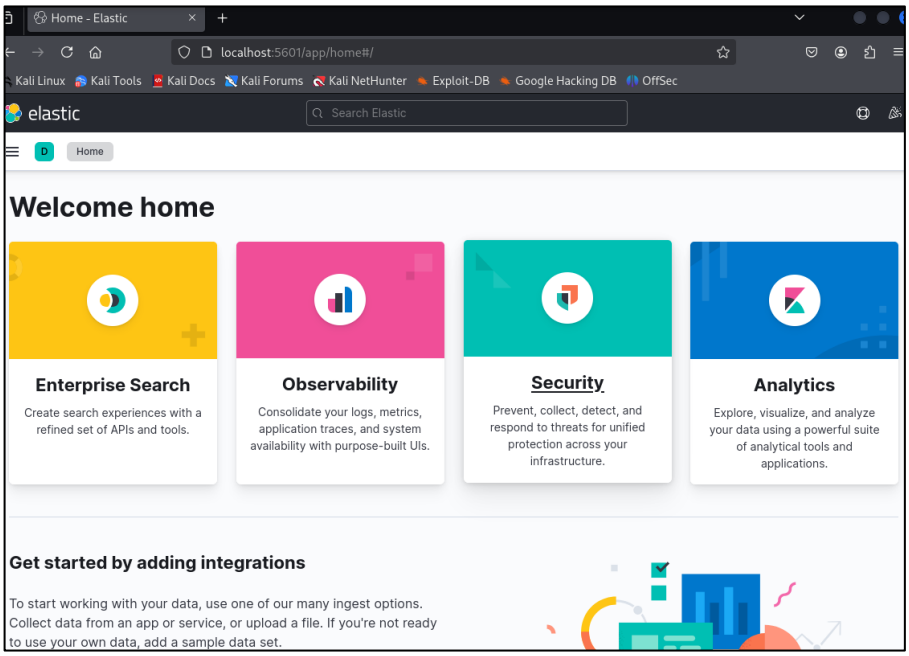


Рисунок Б.32 – Інтерфейс для моніторингу мережевого трафіку

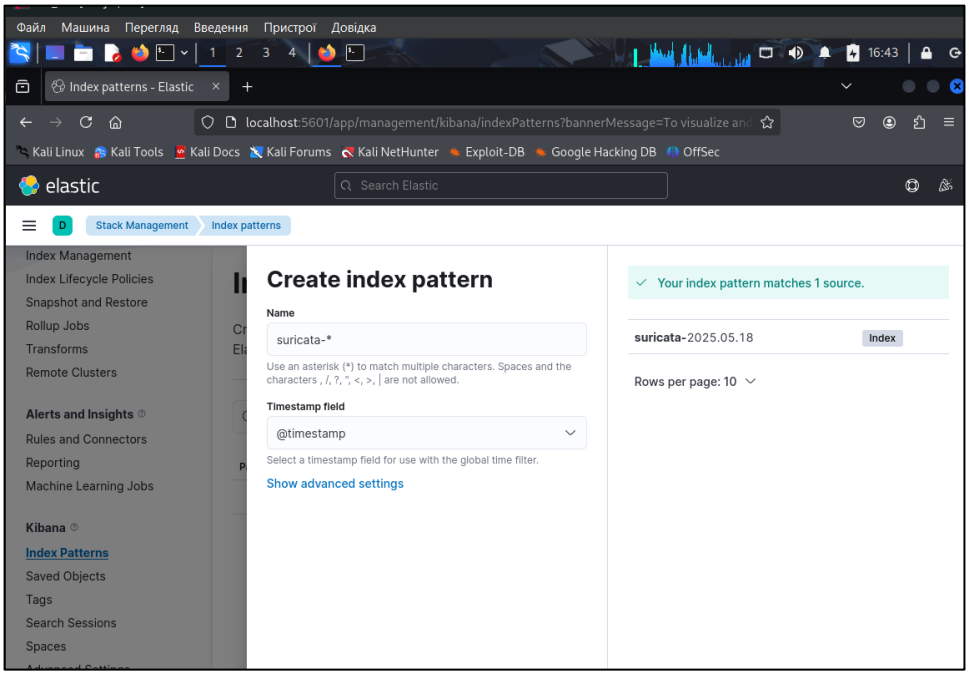


Рисунок Б.33 – Створення нового шаблону індексу (index pattern)

Рисунок Б.34 – Результат налаштування IDS-системи

Рисунок Б.35 – Завантаження інсталяційного .deb файлу

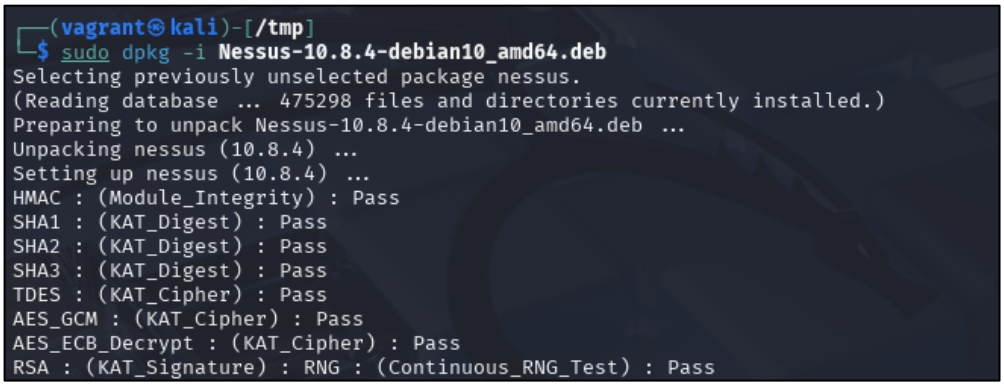


Рисунок Б.36 – Розпаковка інсталяційного файлу та встановлення Nessus

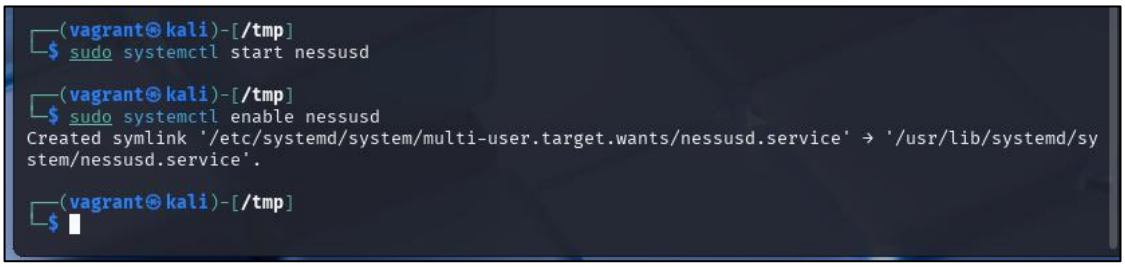


Рисунок Б.37 – Запуск служби Nessus

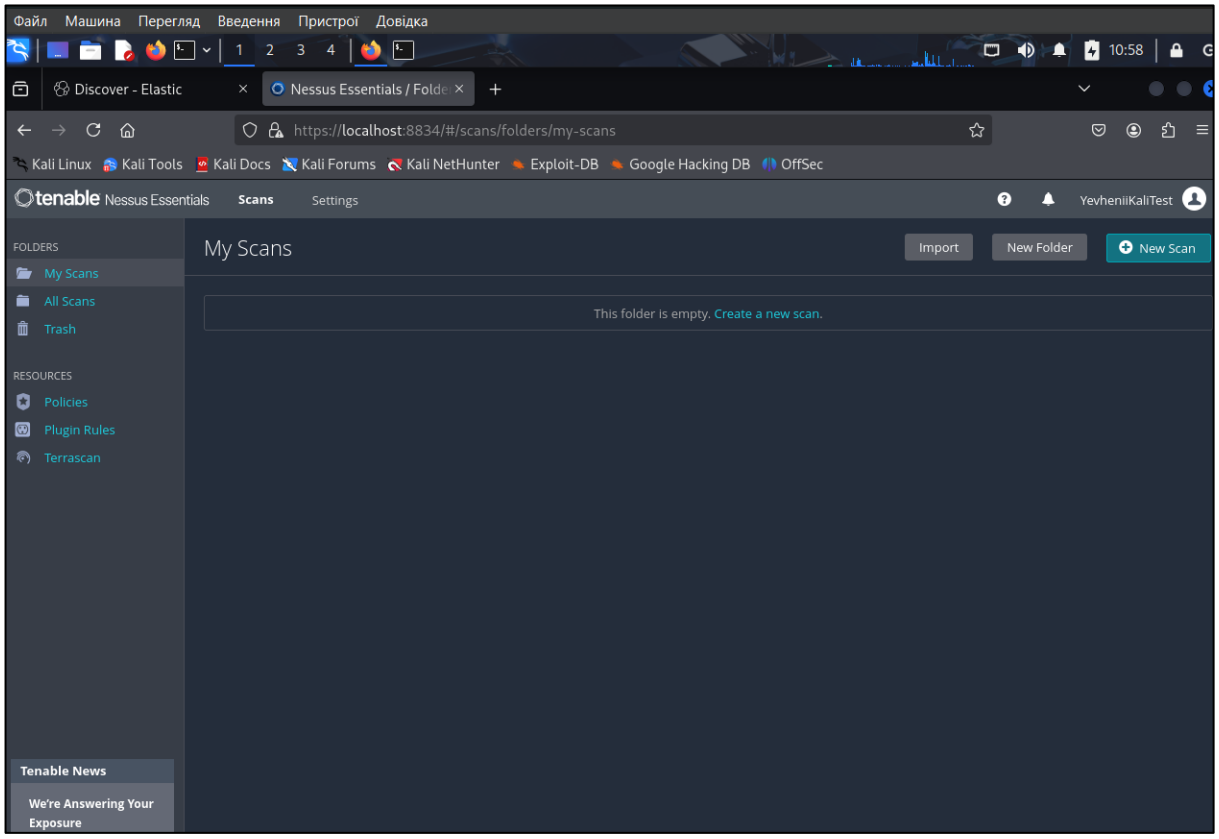


Рисунок Б.38 – Веб-інтерфейс Nessus Essentials

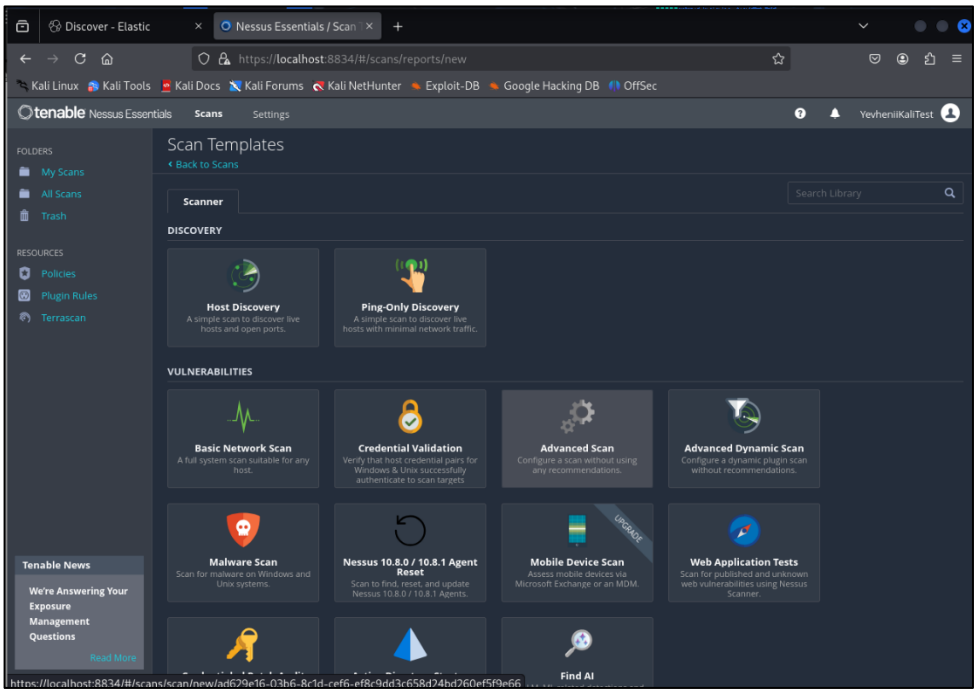


Рисунок Б.39 – Створення розширеного сканування

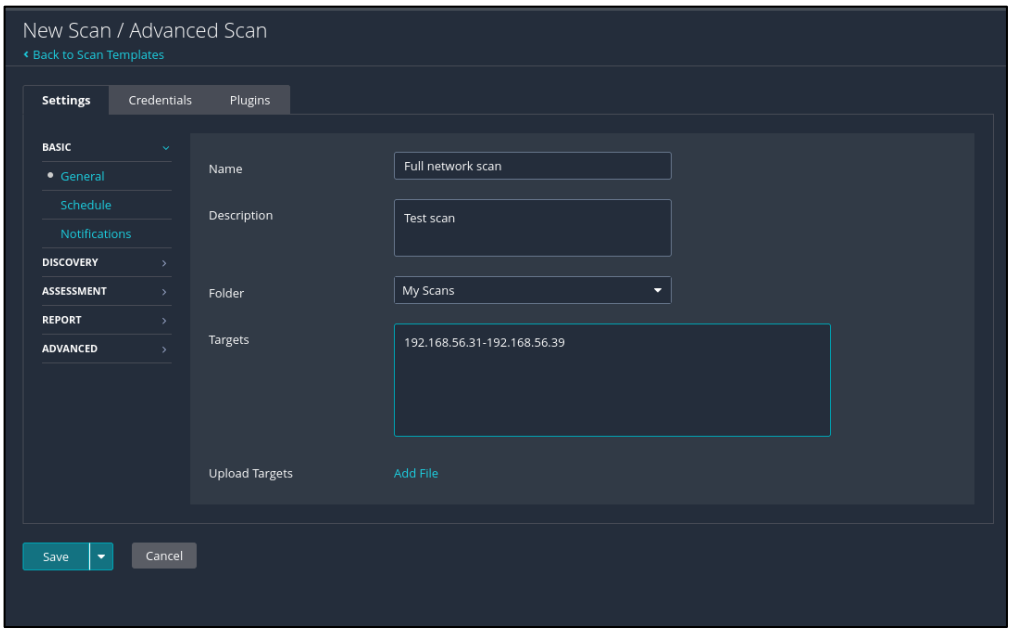


Рисунок Б.40 – Налаштування базових параметрів

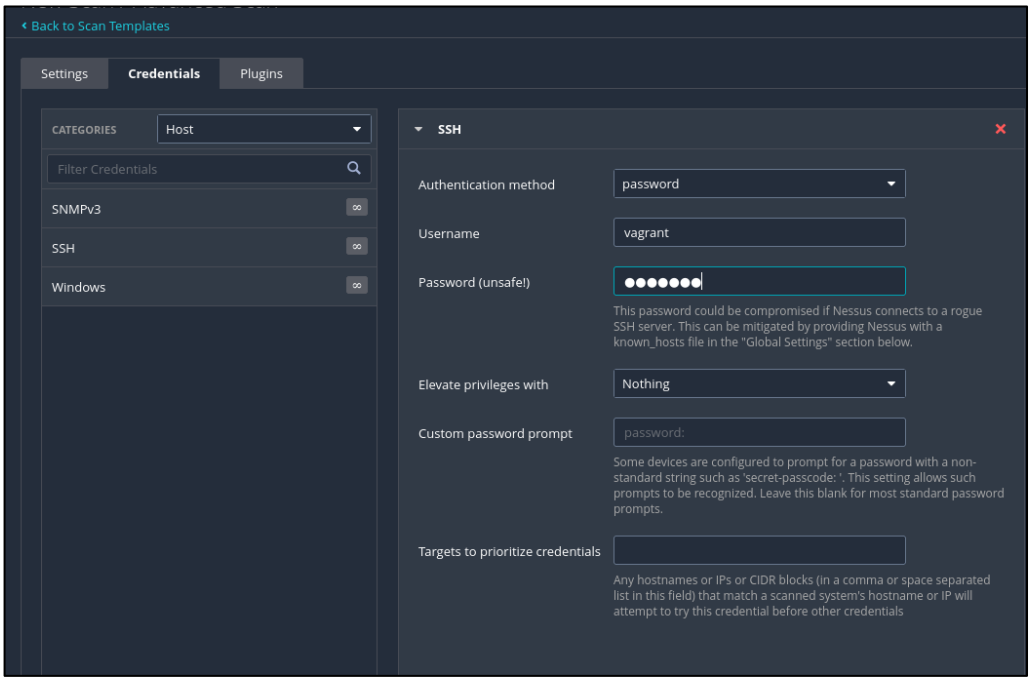


Рисунок Б.41 – Налаштування автентифікованого сканування

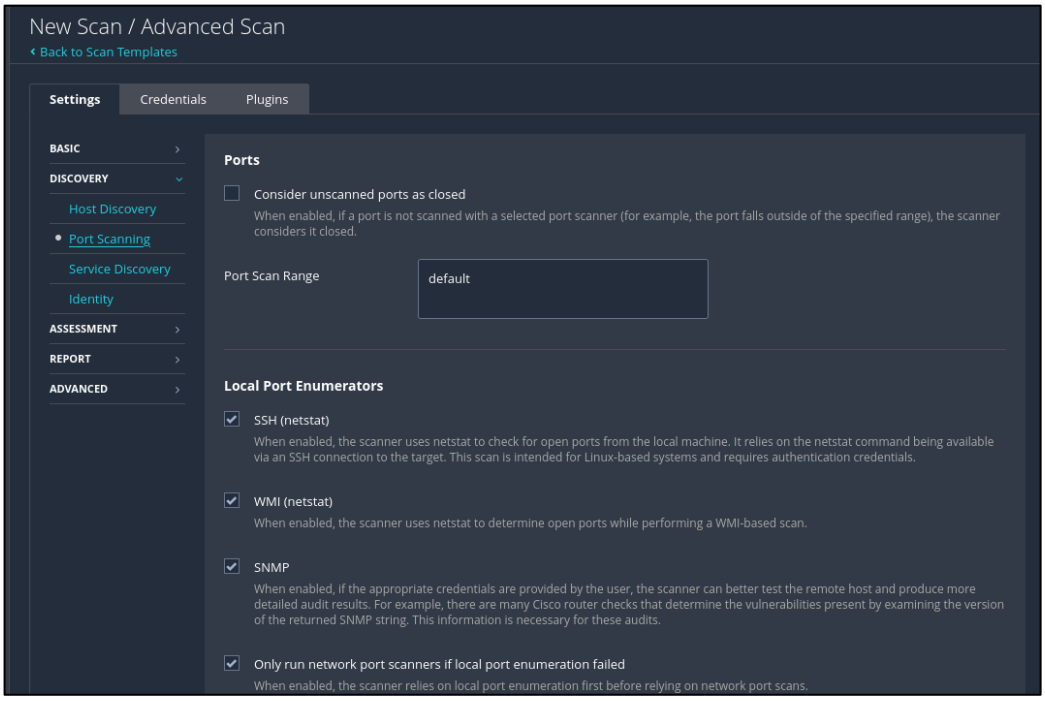


Рисунок Б.42 – Налаштування сканування портів

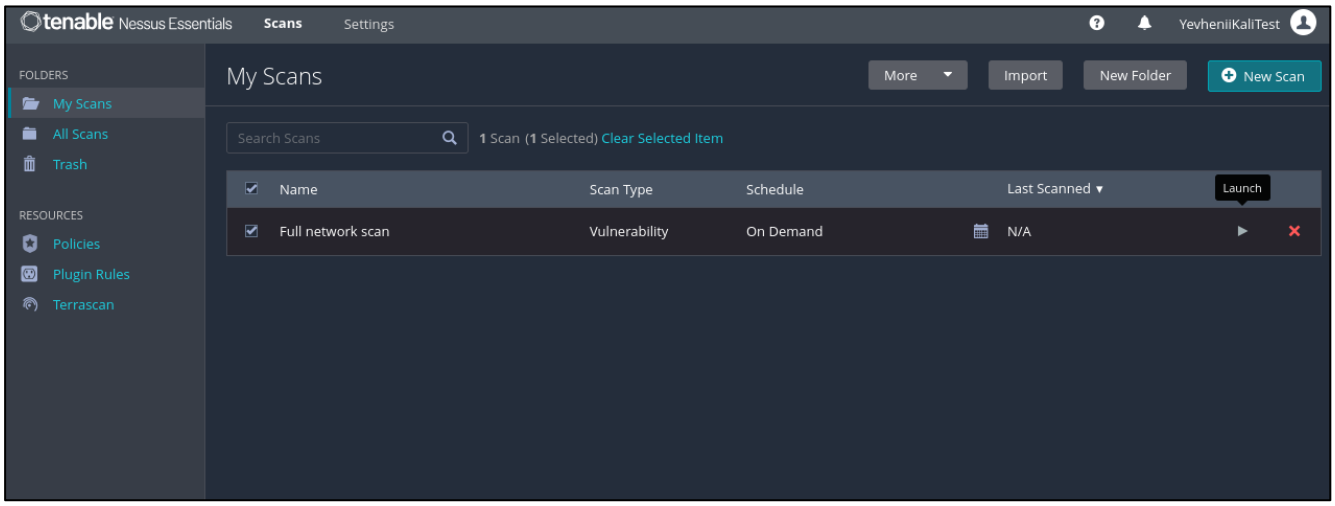


Рисунок Б.43 – Запуск сканування вразливостей мережі

Рисунок Б.44 – Результат виконаного сканування

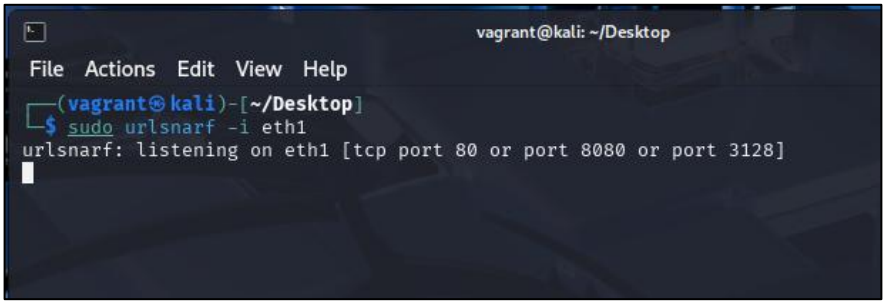


Рисунок Б.45 – Запуск програми для прослуховування мережевого інтерфейсу

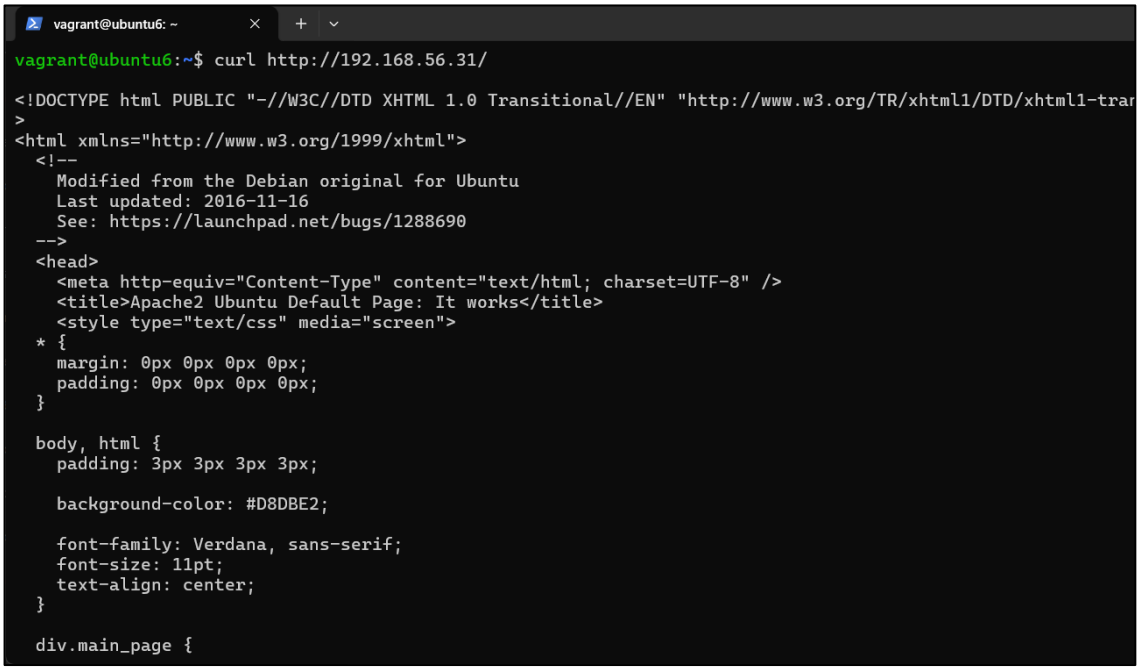


Рисунок Б.46 – Підключення до HTTP-серверу

Рисунок Б.47 – Перехоплення HTTP-запиту

Рисунок Б.48 – Запуск tcpdump для перехоплення трафіку

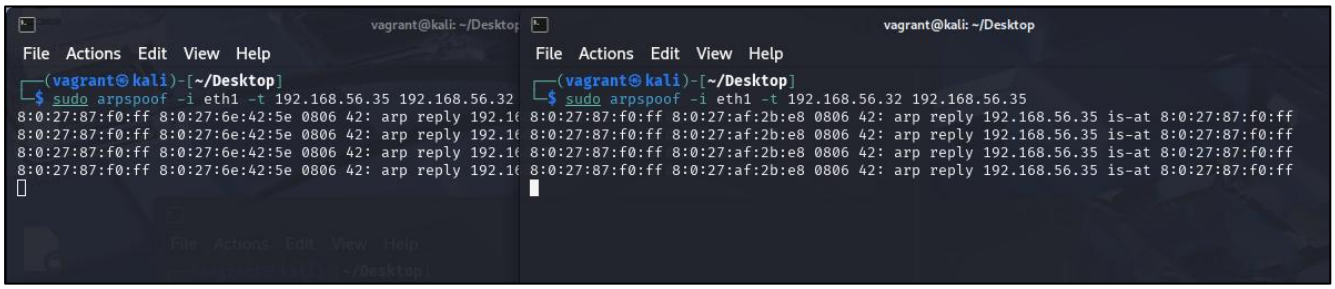


Рисунок Б.49 – Надсилання підроблених ARP-відповідей

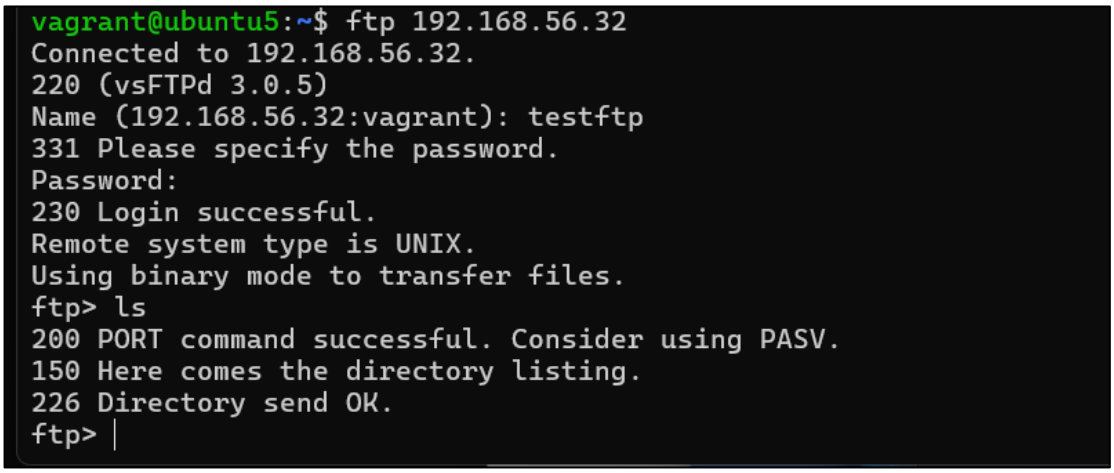


Рисунок Б.50 – Підключення до FTP-сервера

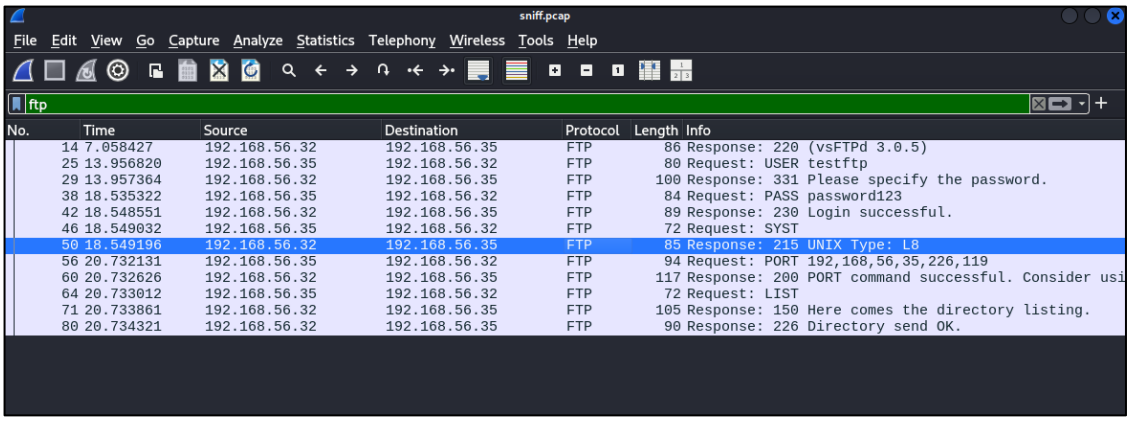


Рисунок Б.51 – Аналіз перехоплених пакетів

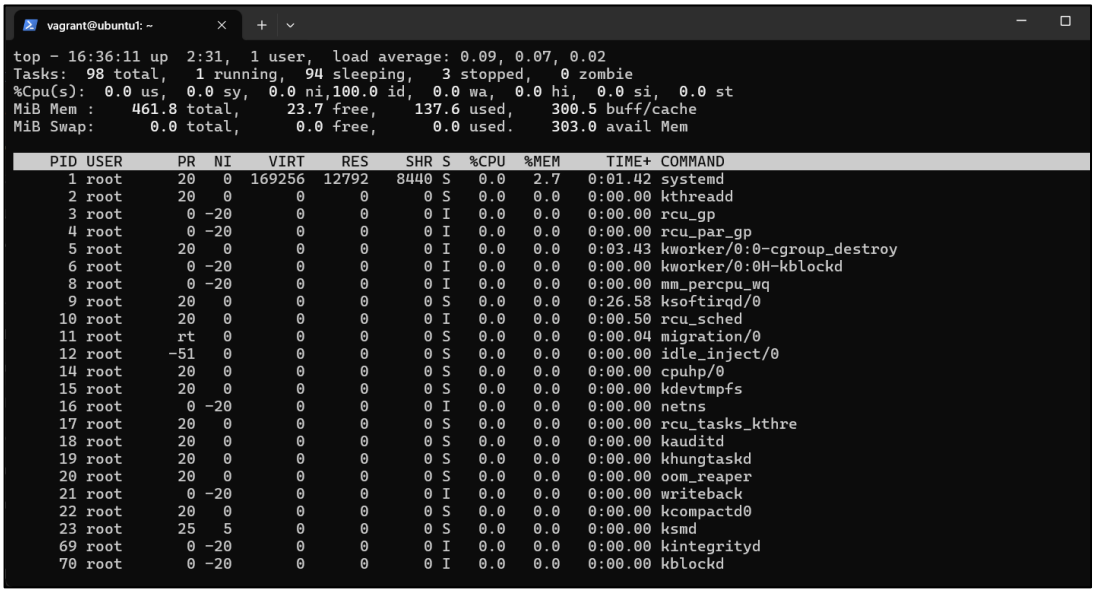


Рисунок Б.52 – Споживання ресурсів системи до реалізації атаки

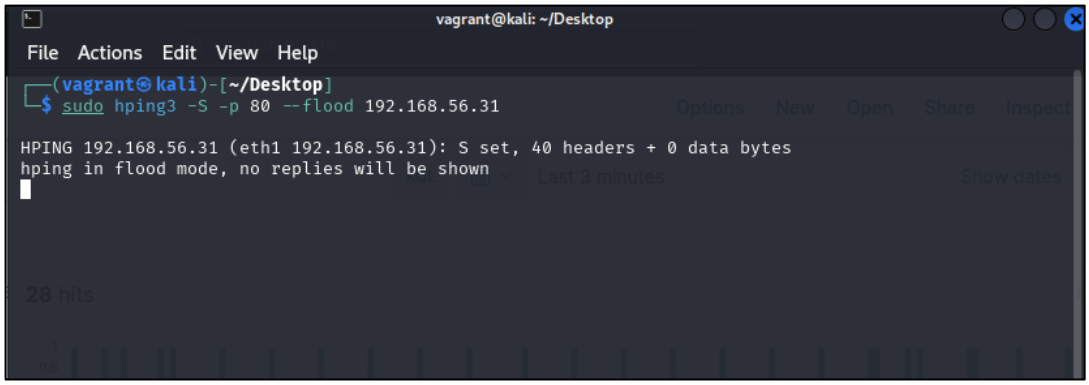


Рисунок Б.53 – Реалізація DoS атаки

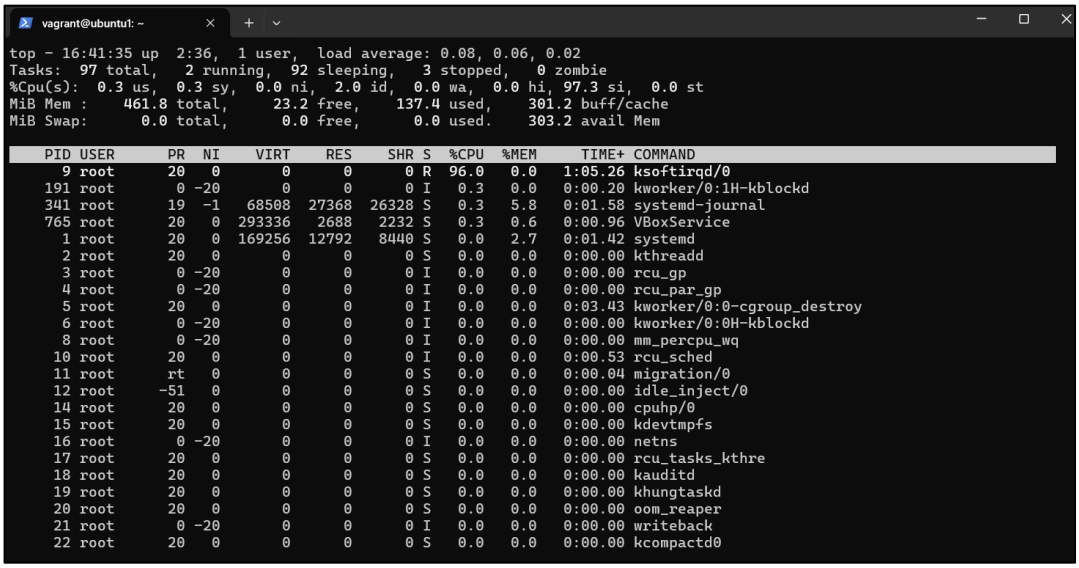


Рисунок Б.54 – Споживання ресурсів системи після реалізації атаки

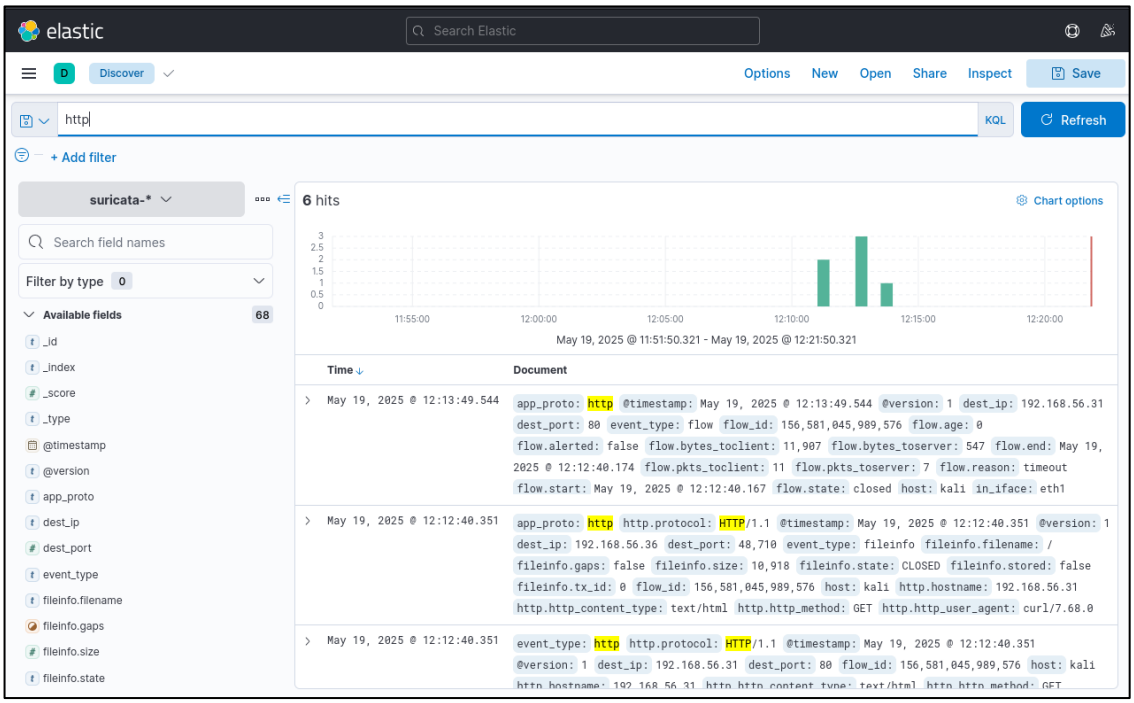


Рисунок Б.55 – Пошук протоколу HTTP

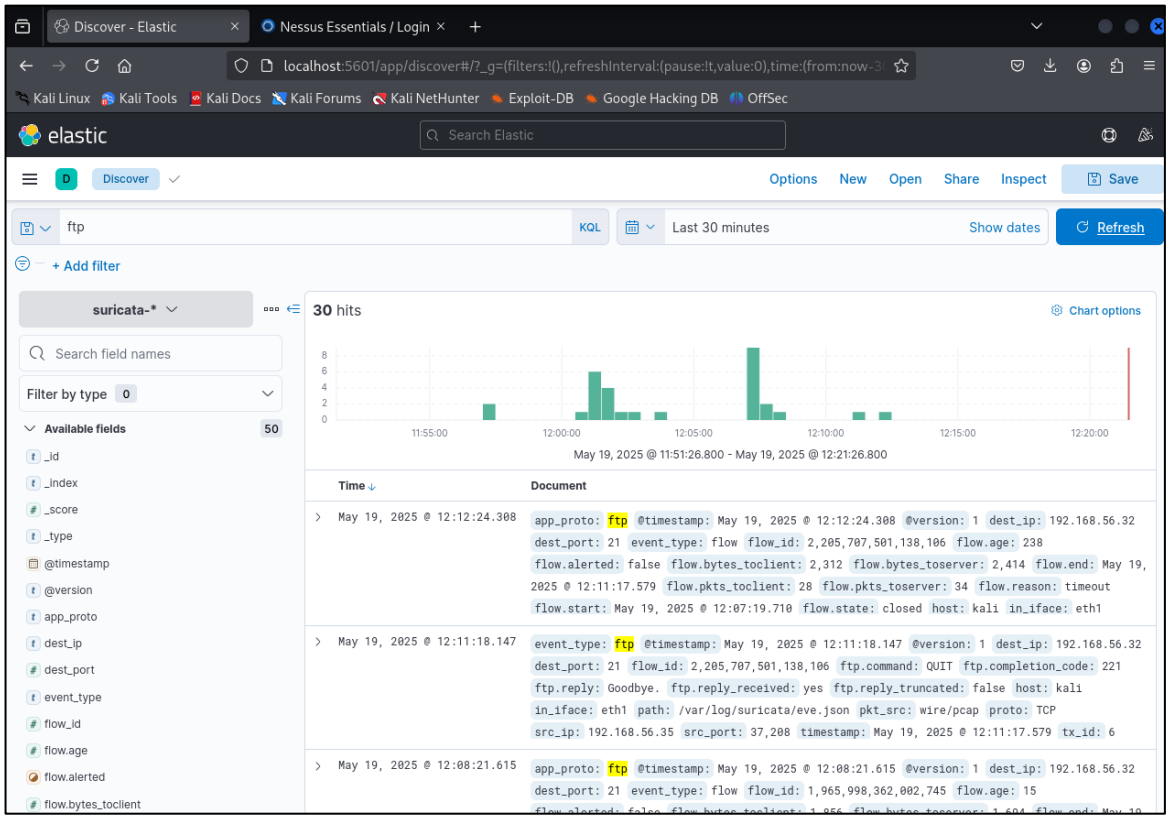


Рисунок Б.56 – Пошук протоколу FTP

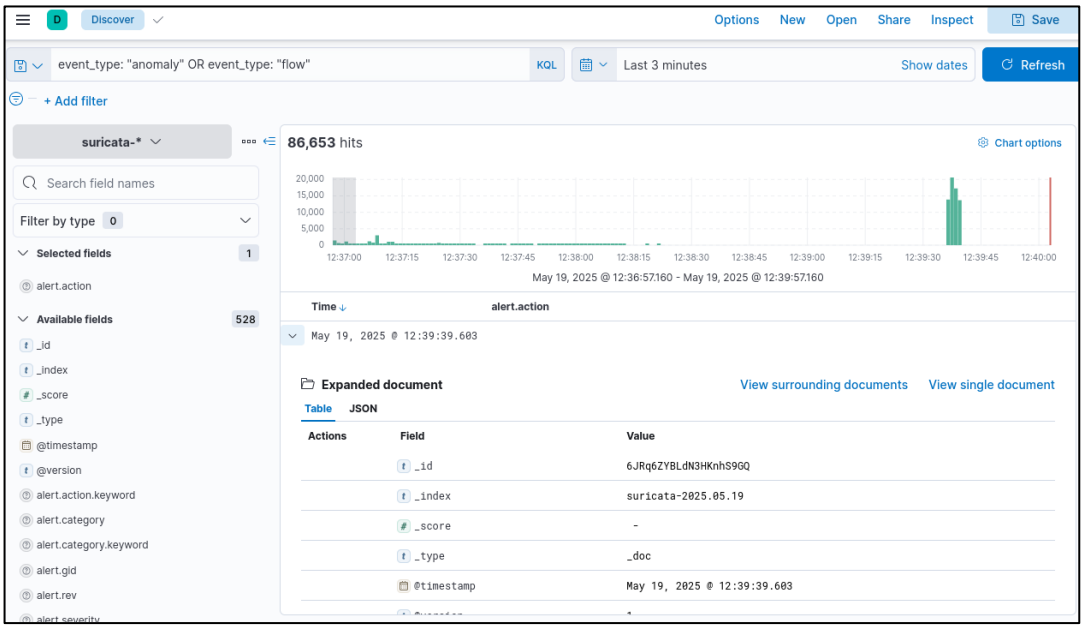
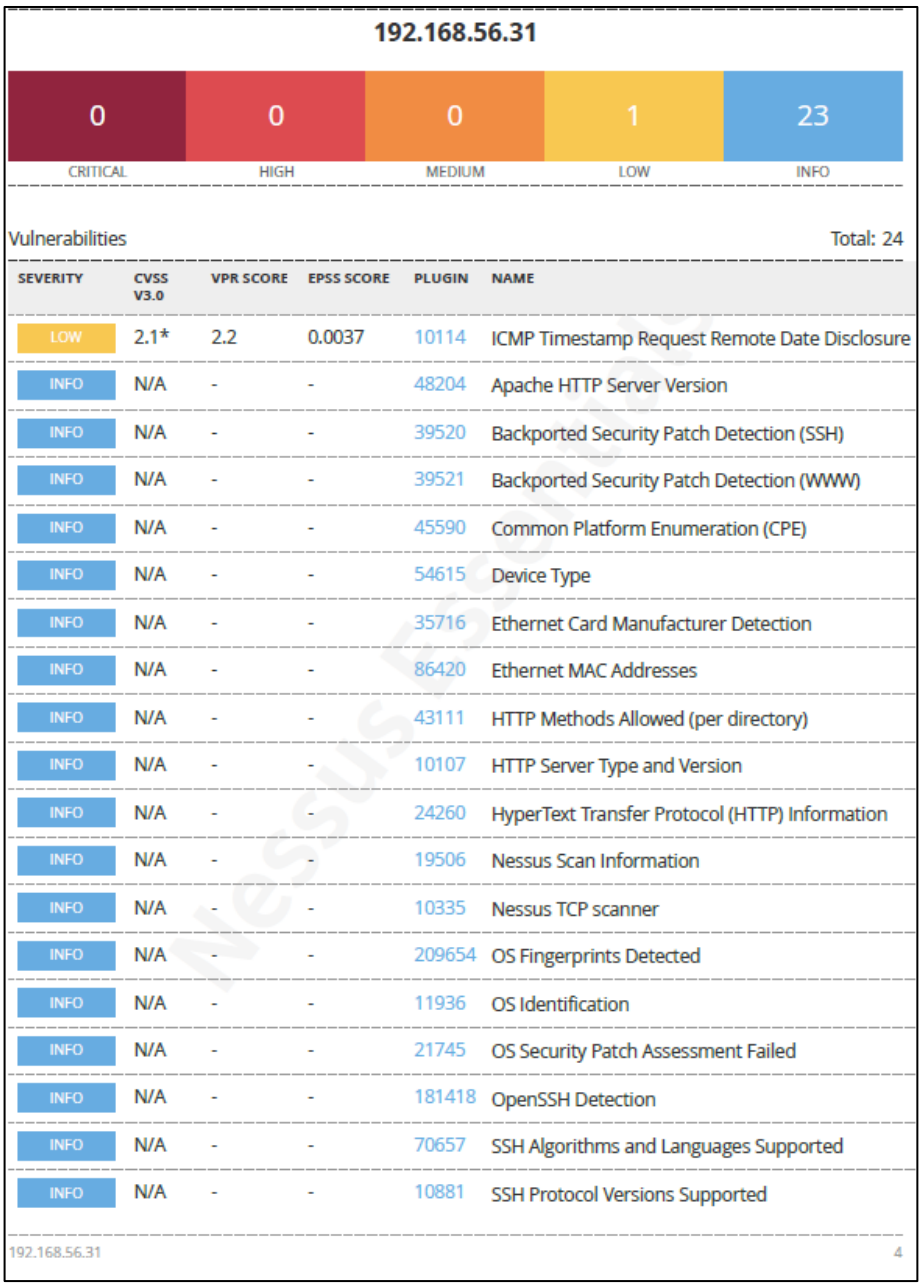


Рисунок Б.57 – Пошук подій flow та anomaly

ДОДАТОК В

ПРИКЛАД ВМІСТУ ЗВІТУ ПРО ВИЯВЛЕНІ ВРАЗЛИВОСТІ



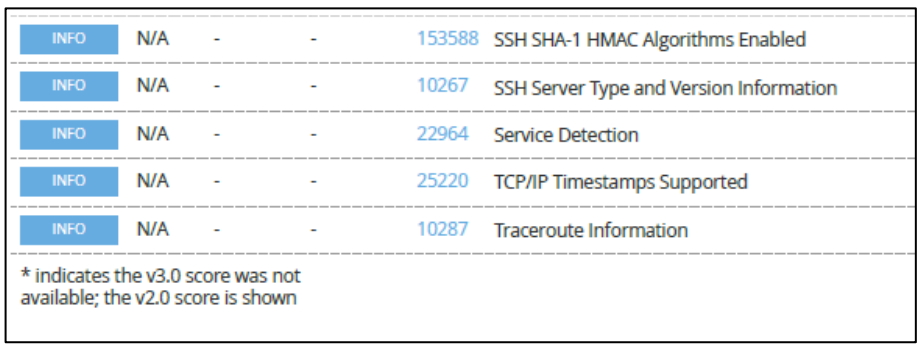
192.168.56.31

CRITICAL	HIGH	MEDIUM	LOW	INFO
0	0	0	1	23

Vulnerabilities — Total: 24

SEVERITY	CVSS V3.0	VPR SCORE	EPSS SCORE	PLUGIN	NAME
LOW	2.1*	2.2	0.0037	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	-	-	48204	Apache HTTP Server Version
INFO	N/A	-	-	39520	Backported Security Patch Detection (SSH)
INFO	N/A	-	-	39521	Backported Security Patch Detection (WWW)
INFO	N/A	-	-	45590	Common Platform Enumeration (CPE)
INFO	N/A	-	-	54615	Device Type
INFO	N/A	-	-	35716	Ethernet Card Manufacturer Detection
INFO	N/A	-	-	86420	Ethernet MAC Addresses
INFO	N/A	-	-	43111	HTTP Methods Allowed (per directory)
INFO	N/A	-	-	10107	HTTP Server Type and Version
INFO	N/A	-	-	24260	HyperText Transfer Protocol (HTTP) Information
INFO	N/A	-	-	19506	Nessus Scan Information
INFO	N/A	-	-	10335	Nessus TCP scanner
INFO	N/A	-	-	209654	OS Fingerprints Detected
INFO	N/A	-	-	11936	OS Identification
INFO	N/A	-	-	21745	OS Security Patch Assessment Failed
INFO	N/A	-	-	181418	OpenSSH Detection
INFO	N/A	-	-	70657	SSH Algorithms and Languages Supported
INFO	N/A	-	-	10881	SSH Protocol Versions Supported

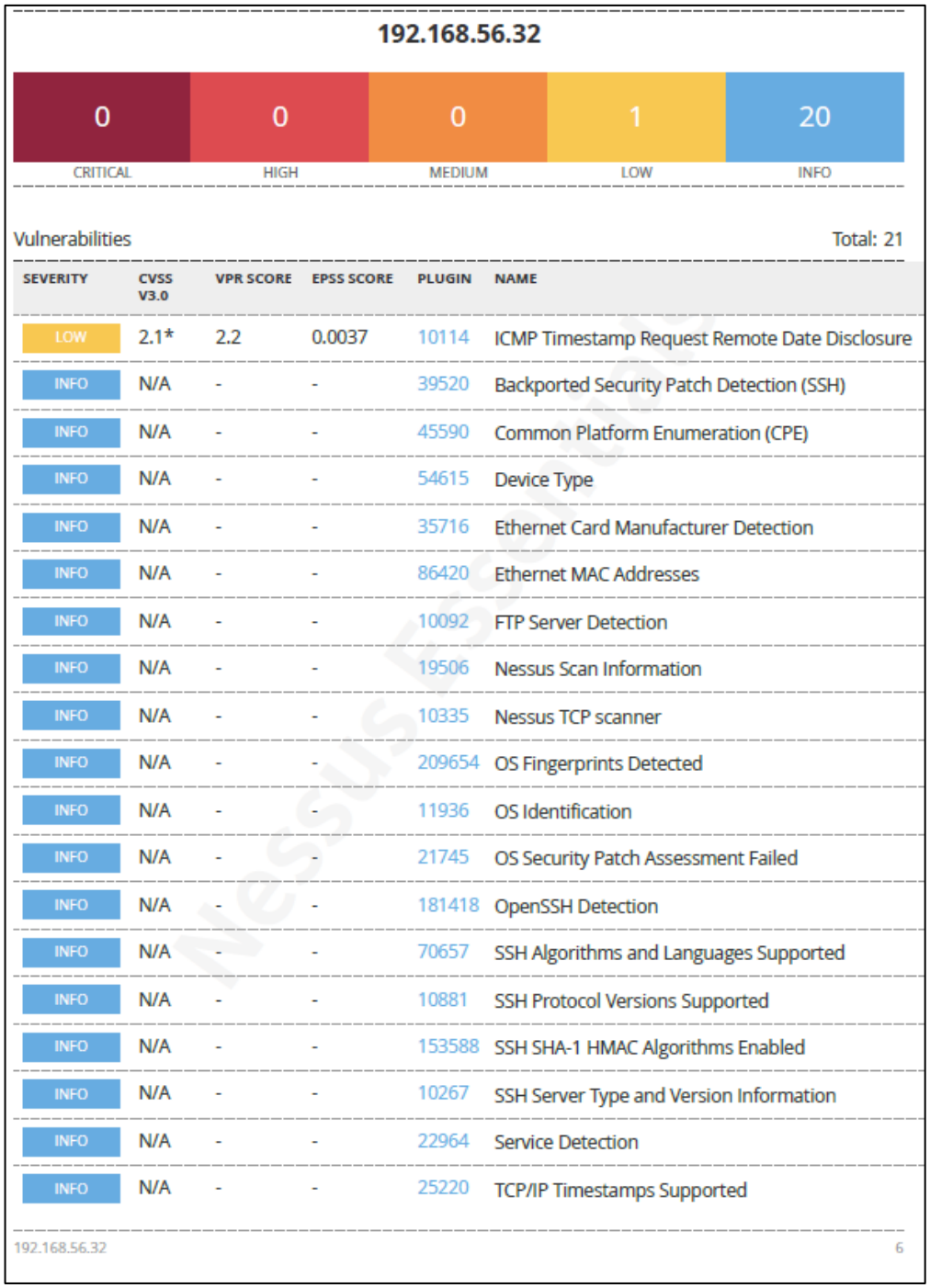
Рисунок В.1 – Результати сканування хосту 192.168.56.31



INFO	N/A	-	-	153588	SSH SHA-1 HMAC Algorithms Enabled
INFO	N/A	-	-	10267	SSH Server Type and Version Information
INFO	N/A	-	-	22964	Service Detection
INFO	N/A	-	-	25220	TCP/IP Timestamps Supported
INFO	N/A	-	-	10287	Traceroute Information

* indicates the v3.0 score was not available; the v2.0 score is shown

Рисунок В.2 – Результати сканування хосту 192.168.56.31



192.168.56.32

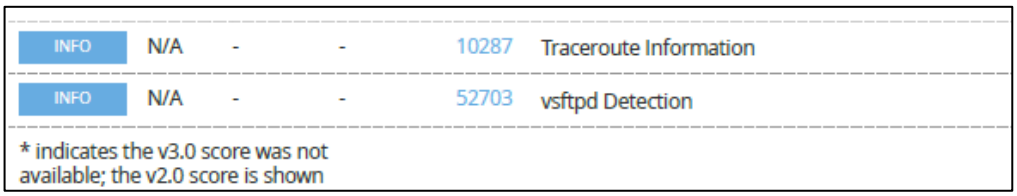
0	0	0	1	20
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities Total: 21

SEVERITY	CVSS V3.0	VPR SCORE	EPSS SCORE	PLUGIN	NAME
LOW	2.1*	2.2	0.0037	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	-	-	39520	Backported Security Patch Detection (SSH)
INFO	N/A	-	-	45590	Common Platform Enumeration (CPE)
INFO	N/A	-	-	54615	Device Type
INFO	N/A	-	-	35716	Ethernet Card Manufacturer Detection
INFO	N/A	-	-	86420	Ethernet MAC Addresses
INFO	N/A	-	-	10092	FTP Server Detection
INFO	N/A	-	-	19506	Nessus Scan Information
INFO	N/A	-	-	10335	Nessus TCP scanner
INFO	N/A	-	-	209654	OS Fingerprints Detected
INFO	N/A	-	-	11936	OS Identification
INFO	N/A	-	-	21745	OS Security Patch Assessment Failed
INFO	N/A	-	-	181418	OpenSSH Detection
INFO	N/A	-	-	70657	SSH Algorithms and Languages Supported
INFO	N/A	-	-	10881	SSH Protocol Versions Supported
INFO	N/A	-	-	153588	SSH SHA-1 HMAC Algorithms Enabled
INFO	N/A	-	-	10267	SSH Server Type and Version Information
INFO	N/A	-	-	22964	Service Detection
INFO	N/A	-	-	25220	TCP/IP Timestamps Supported

192.168.56.32 6

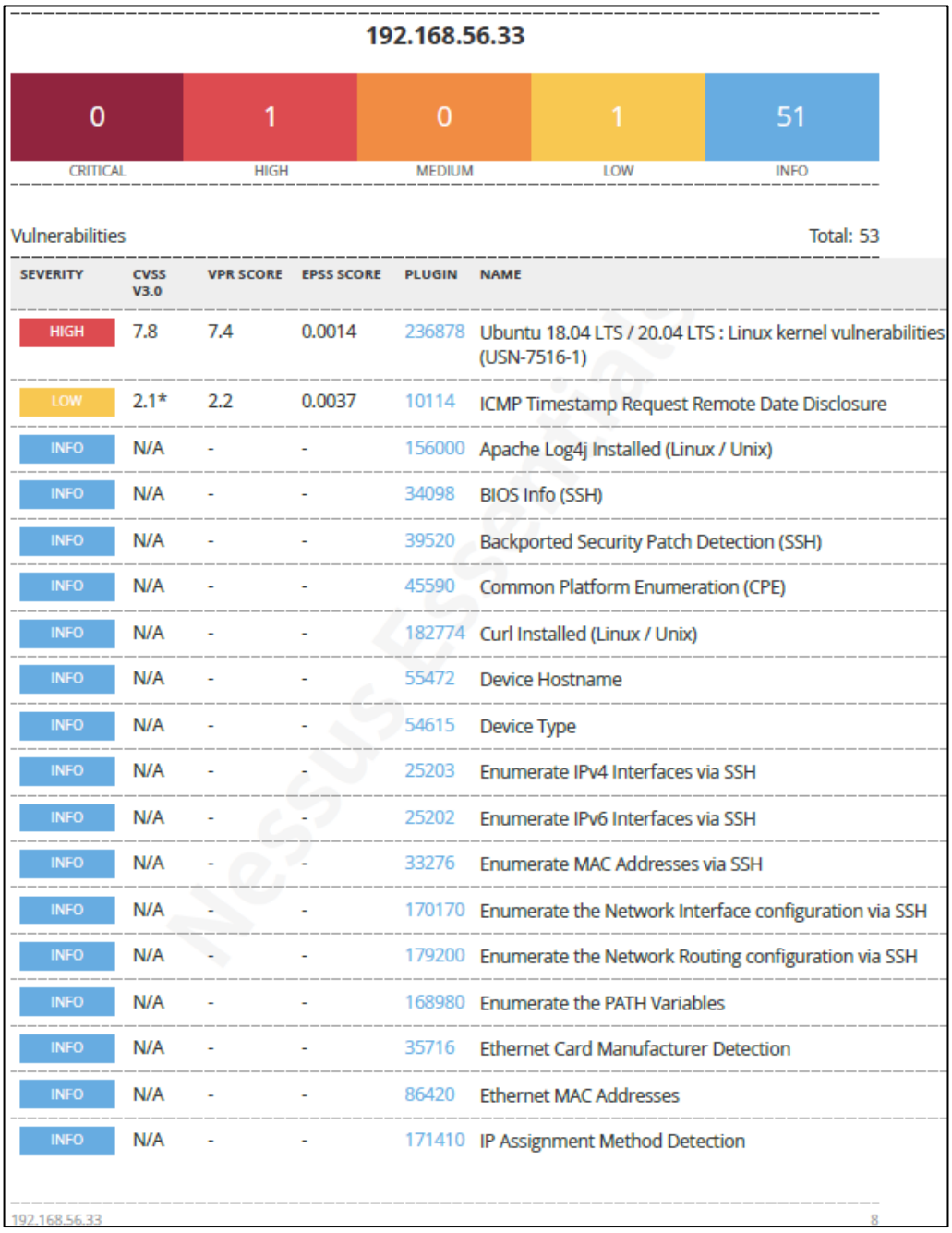
Рисунок В.3 – Результати сканування хосту 192.168.56.32



INFO	N/A	-	-	10287	Traceroute Information
INFO	N/A	-	-	52703	vsftpd Detection

* indicates the v3.0 score was not available; the v2.0 score is shown

Рисунок В.4 – Результати сканування хосту 192.168.56.32



192.168.56.33

0	1	0	1	51
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities Total: 53

SEVERITY	CVSS V3.0	VPR SCORE	EPSS SCORE	PLUGIN	NAME
HIGH	7.8	7.4	0.0014	236878	Ubuntu 18.04 LTS / 20.04 LTS : Linux kernel vulnerabilities (USN-7516-1)
LOW	2.1*	2.2	0.0037	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	-	-	156000	Apache Log4j Installed (Linux / Unix)
INFO	N/A	-	-	34098	BIOS Info (SSH)
INFO	N/A	-	-	39520	Backported Security Patch Detection (SSH)
INFO	N/A	-	-	45590	Common Platform Enumeration (CPE)
INFO	N/A	-	-	182774	Curl Installed (Linux / Unix)
INFO	N/A	-	-	55472	Device Hostname
INFO	N/A	-	-	54615	Device Type
INFO	N/A	-	-	25203	Enumerate IPv4 Interfaces via SSH
INFO	N/A	-	-	25202	Enumerate IPv6 Interfaces via SSH
INFO	N/A	-	-	33276	Enumerate MAC Addresses via SSH
INFO	N/A	-	-	170170	Enumerate the Network Interface configuration via SSH
INFO	N/A	-	-	179200	Enumerate the Network Routing configuration via SSH
INFO	N/A	-	-	168980	Enumerate the PATH Variables
INFO	N/A	-	-	35716	Ethernet Card Manufacturer Detection
INFO	N/A	-	-	86420	Ethernet MAC Addresses
INFO	N/A	-	-	171410	IP Assignment Method Detection

192.168.56.33 8

Рисунок В.5 – Результати сканування хосту 192.168.56.33

INFO	N/A	-	-	151883	Libgcrypt Installed (Linux/UNIX)
INFO	N/A	-	-	157358	Linux Mounted Devices
INFO	N/A	-	-	193143	Linux Time Zone Information
INFO	N/A	-	-	95928	Linux User List Enumeration
INFO	N/A	-	-	19506	Nessus Scan Information
INFO	N/A	-	-	64582	Netstat Connection Information
INFO	N/A	-	-	14272	Netstat Portscanner (SSH)
INFO	N/A	-	-	209654	OS Fingerprints Detected
INFO	N/A	-	-	11936	OS Identification
INFO	N/A	-	-	97993	OS Identification and Installed Software Enumeration over SS (Using New SSH Library)
INFO	N/A	-	-	117887	OS Security Patch Assessment Available
INFO	N/A	-	-	181418	OpenSSH Detection
INFO	N/A	-	-	168007	OpenSSL Installed (Linux)
INFO	N/A	-	-	179139	Package Manager Packages Report (nix)
INFO	N/A	-	-	66334	Patch Report
INFO	N/A	-	-	70657	SSH Algorithms and Languages Supported
INFO	N/A	-	-	102094	SSH Commands Require Privilege Escalation
INFO	N/A	-	-	149334	SSH Password Authentication Accepted
INFO	N/A	-	-	10881	SSH Protocol Versions Supported
INFO	N/A	-	-	90707	SSH SCP Protocol Detection
INFO	N/A	-	-	153588	SSH SHA-1 HMAC Algorithms Enabled
INFO	N/A	-	-	10267	SSH Server Type and Version Information
INFO	N/A	-	-	22964	Service Detection
INFO	N/A	-	-	22869	Software Enumeration (SSH)
INFO	N/A	-	-	25220	TCP/IP Timestamps Supported

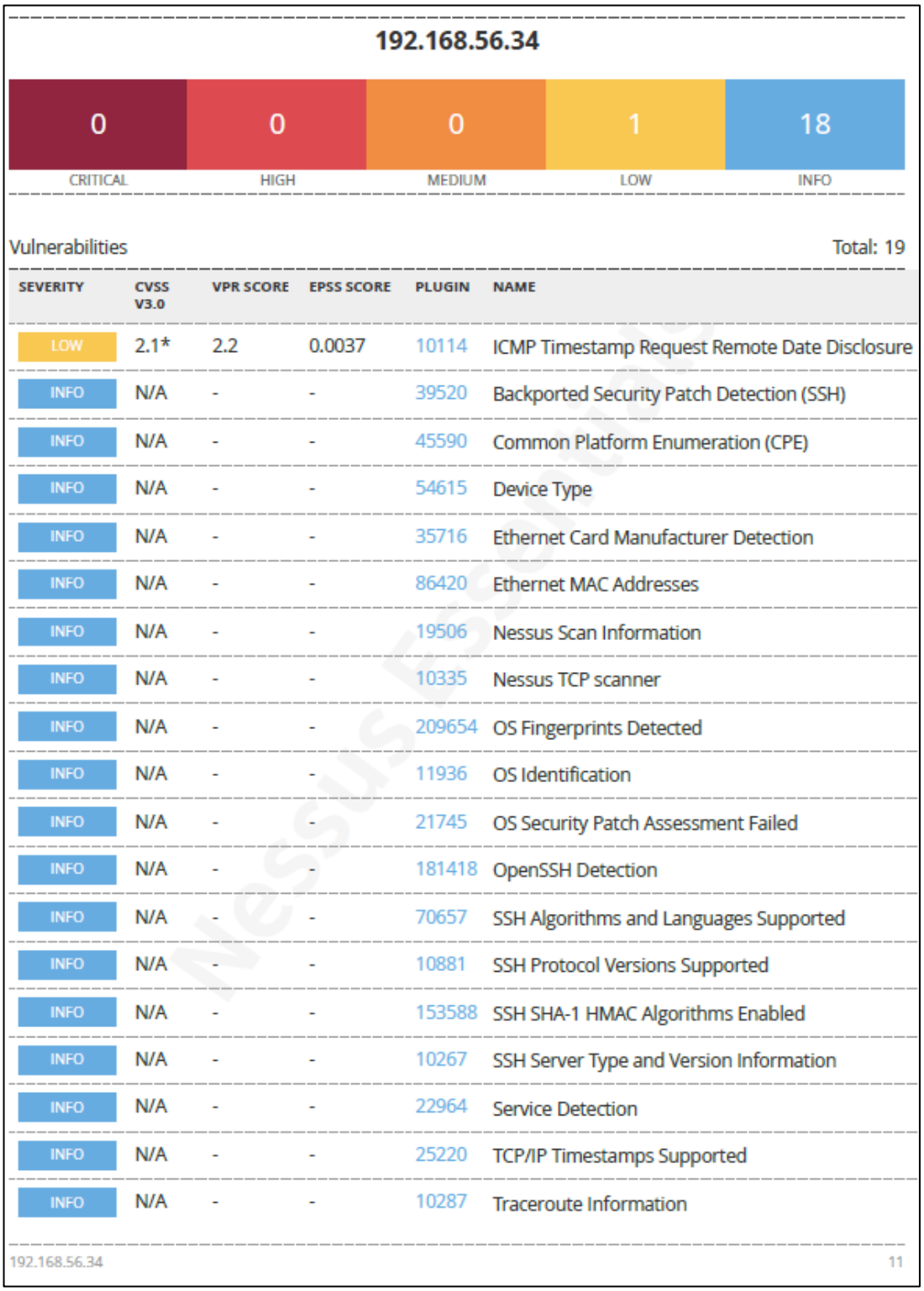
192.168.56.33 9

Рисунок В.6 – Результати сканування хосту 192.168.56.33

INFO	N/A	-	-	110385	Target Credential Issues by Authentication Protocol - Insufficie Privilege
INFO	N/A	-	-	141118	Target Credential Status by Authentication Protocol - Valid Credentials Provided
INFO	N/A	-	-	56468	Time of Last System Startup
INFO	N/A	-	-	10287	Traceroute Information
INFO	N/A	-	-	192709	Tukaani XZ Utils Installed (Linux / Unix)
INFO	N/A	-	-	110483	Unix / Linux Running Processes Information
INFO	N/A	-	-	152743	Unix Software Discovery Commands Not Available
INFO	N/A	-	-	186361	VMWare Tools or Open VM Tools Installed (Linux)
INFO	N/A	-	-	189731	Vim Installed (Linux)
INFO	N/A	-	-	182848	libcurl Installed (Linux / Unix)

* indicates the v3.0 score was not available; the v2.0 score is shown

Рисунок В.7 – Результати сканування хосту 192.168.56.33



192.168.56.34

0	0	0	1	18
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities Total: 19

SEVERITY	CVSS V3.0	VPR SCORE	EPSS SCORE	PLUGIN	NAME
LOW	2.1*	2.2	0.0037	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	-	-	39520	Backported Security Patch Detection (SSH)
INFO	N/A	-	-	45590	Common Platform Enumeration (CPE)
INFO	N/A	-	-	54615	Device Type
INFO	N/A	-	-	35716	Ethernet Card Manufacturer Detection
INFO	N/A	-	-	86420	Ethernet MAC Addresses
INFO	N/A	-	-	19506	Nessus Scan Information
INFO	N/A	-	-	10335	Nessus TCP scanner
INFO	N/A	-	-	209654	OS Fingerprints Detected
INFO	N/A	-	-	11936	OS Identification
INFO	N/A	-	-	21745	OS Security Patch Assessment Failed
INFO	N/A	-	-	181418	OpenSSH Detection
INFO	N/A	-	-	70657	SSH Algorithms and Languages Supported
INFO	N/A	-	-	10881	SSH Protocol Versions Supported
INFO	N/A	-	-	153588	SSH SHA-1 HMAC Algorithms Enabled
INFO	N/A	-	-	10267	SSH Server Type and Version Information
INFO	N/A	-	-	22964	Service Detection
INFO	N/A	-	-	25220	TCP/IP Timestamps Supported
INFO	N/A	-	-	10287	Traceroute Information

192.168.56.34 11

Рисунок В.8 – Результати сканування хосту 192.168.56.34

192.168.56.35

0	0	0	1	18
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities — Total: 19

SEVERITY	CVSS V3.0	VPR SCORE	EPSS SCORE	PLUGIN	NAME
LOW	2.1*	2.2	0.0037	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	-	-	39520	Backported Security Patch Detection (SSH)
INFO	N/A	-	-	45590	Common Platform Enumeration (CPE)
INFO	N/A	-	-	54615	Device Type
INFO	N/A	-	-	35716	Ethernet Card Manufacturer Detection
INFO	N/A	-	-	86420	Ethernet MAC Addresses
INFO	N/A	-	-	19506	Nessus Scan Information
INFO	N/A	-	-	10335	Nessus TCP scanner
INFO	N/A	-	-	209654	OS Fingerprints Detected
INFO	N/A	-	-	11936	OS Identification
INFO	N/A	-	-	21745	OS Security Patch Assessment Failed
INFO	N/A	-	-	181418	OpenSSH Detection
INFO	N/A	-	-	70657	SSH Algorithms and Languages Supported
INFO	N/A	-	-	10881	SSH Protocol Versions Supported
INFO	N/A	-	-	153588	SSH SHA-1 HMAC Algorithms Enabled
INFO	N/A	-	-	10267	SSH Server Type and Version Information
INFO	N/A	-	-	22964	Service Detection
INFO	N/A	-	-	25220	TCP/IP Timestamps Supported
INFO	N/A	-	-	10287	Traceroute Information

192.168.56.35 13

Рисунок В.9 – Результати сканування хосту 192.168.56.35

192.168.56.36

0	0	0	1	18
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities — Total: 19

SEVERITY	CVSS V3.0	VPR SCORE	EPSS SCORE	PLUGIN	NAME
LOW	2.1*	2.2	0.0037	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	-	-	39520	Backported Security Patch Detection (SSH)
INFO	N/A	-	-	45590	Common Platform Enumeration (CPE)
INFO	N/A	-	-	54615	Device Type
INFO	N/A	-	-	35716	Ethernet Card Manufacturer Detection
INFO	N/A	-	-	86420	Ethernet MAC Addresses
INFO	N/A	-	-	19506	Nessus Scan Information
INFO	N/A	-	-	10335	Nessus TCP scanner
INFO	N/A	-	-	209654	OS Fingerprints Detected
INFO	N/A	-	-	11936	OS Identification
INFO	N/A	-	-	21745	OS Security Patch Assessment Failed
INFO	N/A	-	-	181418	OpenSSH Detection
INFO	N/A	-	-	70657	SSH Algorithms and Languages Supported
INFO	N/A	-	-	10881	SSH Protocol Versions Supported
INFO	N/A	-	-	153588	SSH SHA-1 HMAC Algorithms Enabled
INFO	N/A	-	-	10267	SSH Server Type and Version Information
INFO	N/A	-	-	22964	Service Detection
INFO	N/A	-	-	25220	TCP/IP Timestamps Supported
INFO	N/A	-	-	10287	Traceroute Information

Рисунок В.10 – Результати сканування хосту 192.168.56.36

192.168.56.37

0	0	0	1	18
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities — Total: 19

SEVERITY	CVSS V3.0	VPR SCORE	EPSS SCORE	PLUGIN	NAME
LOW	2.1*	2.2	0.0037	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	-	-	39520	Backported Security Patch Detection (SSH)
INFO	N/A	-	-	45590	Common Platform Enumeration (CPE)
INFO	N/A	-	-	54615	Device Type
INFO	N/A	-	-	35716	Ethernet Card Manufacturer Detection
INFO	N/A	-	-	86420	Ethernet MAC Addresses
INFO	N/A	-	-	19506	Nessus Scan Information
INFO	N/A	-	-	10335	Nessus TCP scanner
INFO	N/A	-	-	209654	OS Fingerprints Detected
INFO	N/A	-	-	11936	OS Identification
INFO	N/A	-	-	21745	OS Security Patch Assessment Failed
INFO	N/A	-	-	181418	OpenSSH Detection
INFO	N/A	-	-	70657	SSH Algorithms and Languages Supported
INFO	N/A	-	-	10881	SSH Protocol Versions Supported
INFO	N/A	-	-	153588	SSH SHA-1 HMAC Algorithms Enabled
INFO	N/A	-	-	10267	SSH Server Type and Version Information
INFO	N/A	-	-	22964	Service Detection
INFO	N/A	-	-	25220	TCP/IP Timestamps Supported
INFO	N/A	-	-	10287	Traceroute Information

Рисунок В.11 – Результати сканування хосту 192.168.56.37

192.168.56.38

0	0	0	1	18
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities — Total: 19

SEVERITY	CVSS V3.0	VPR SCORE	EPSS SCORE	PLUGIN	NAME
LOW	2.1*	2.2	0.0037	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	-	-	39520	Backported Security Patch Detection (SSH)
INFO	N/A	-	-	45590	Common Platform Enumeration (CPE)
INFO	N/A	-	-	54615	Device Type
INFO	N/A	-	-	35716	Ethernet Card Manufacturer Detection
INFO	N/A	-	-	86420	Ethernet MAC Addresses
INFO	N/A	-	-	19506	Nessus Scan Information
INFO	N/A	-	-	10335	Nessus TCP scanner
INFO	N/A	-	-	209654	OS Fingerprints Detected
INFO	N/A	-	-	11936	OS Identification
INFO	N/A	-	-	21745	OS Security Patch Assessment Failed
INFO	N/A	-	-	181418	OpenSSH Detection
INFO	N/A	-	-	70657	SSH Algorithms and Languages Supported
INFO	N/A	-	-	10881	SSH Protocol Versions Supported
INFO	N/A	-	-	153588	SSH SHA-1 HMAC Algorithms Enabled
INFO	N/A	-	-	10267	SSH Server Type and Version Information
INFO	N/A	-	-	22964	Service Detection
INFO	N/A	-	-	25220	TCP/IP Timestamps Supported
INFO	N/A	-	-	10287	Traceroute Information

192.168.56.38 19

Рисунок В.12 – Результати сканування хосту 192.168.56.38

192.168.56.39

0	0	0	1	18
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities — Total: 19

SEVERITY	CVSS V3.0	VPR SCORE	EPSS SCORE	PLUGIN	NAME
LOW	2.1*	2.2	0.0037	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	-	-	39520	Backported Security Patch Detection (SSH)
INFO	N/A	-	-	45590	Common Platform Enumeration (CPE)
INFO	N/A	-	-	54615	Device Type
INFO	N/A	-	-	35716	Ethernet Card Manufacturer Detection
INFO	N/A	-	-	86420	Ethernet MAC Addresses
INFO	N/A	-	-	19506	Nessus Scan Information
INFO	N/A	-	-	10335	Nessus TCP scanner
INFO	N/A	-	-	209654	OS Fingerprints Detected
INFO	N/A	-	-	11936	OS Identification
INFO	N/A	-	-	21745	OS Security Patch Assessment Failed
INFO	N/A	-	-	181418	OpenSSH Detection
INFO	N/A	-	-	70657	SSH Algorithms and Languages Supported
INFO	N/A	-	-	10881	SSH Protocol Versions Supported
INFO	N/A	-	-	153588	SSH SHA-1 HMAC Algorithms Enabled
INFO	N/A	-	-	10267	SSH Server Type and Version Information
INFO	N/A	-	-	22964	Service Detection
INFO	N/A	-	-	25220	TCP/IP Timestamps Supported
INFO	N/A	-	-	10287	Traceroute Information

Рисунок В.13 – Результати сканування хосту 192.168.56.39