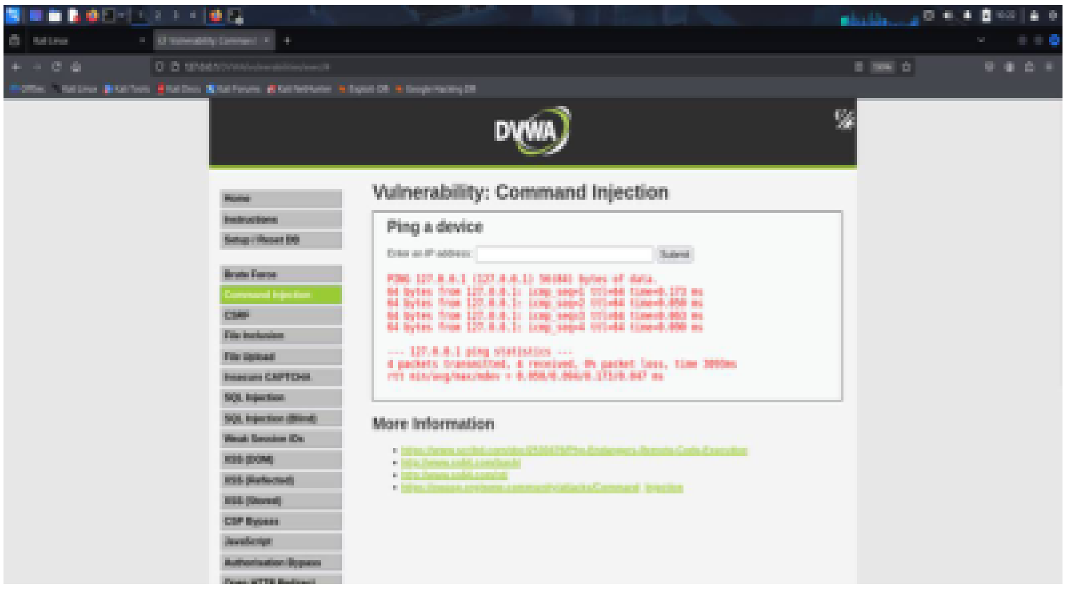
DVWA - Command Injection Exploit

Security Level: Low

Step 1: Navigate to the 'Command Injection' section in DVWA. Enter a valid IP address (e.g., 127.0.0.1) and click Submit to verify functionality.



Vulnerability: Command Injection

Ping a device

Enter an IP address:

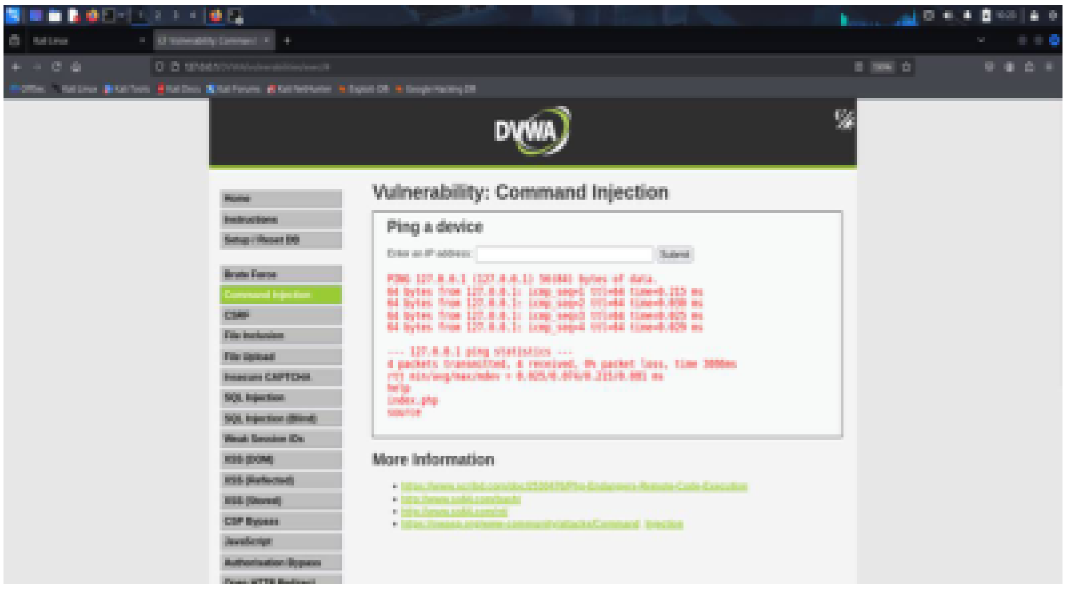
```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.173 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.050 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.063 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.090 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3066ms
rtt min/avg/max/mdev = 0.050/0.094/0.173/0.047 ms
```

More Information

- [illegible]
- [illegible]
- [illegible]
- [illegible]

Step 2: Test command injection by appending '&& help' after the IP address (e.g., 127.0.0.1 && help). This should display a list of commands.



Vulnerability: Command Injection

Ping a device

Enter an IP address:

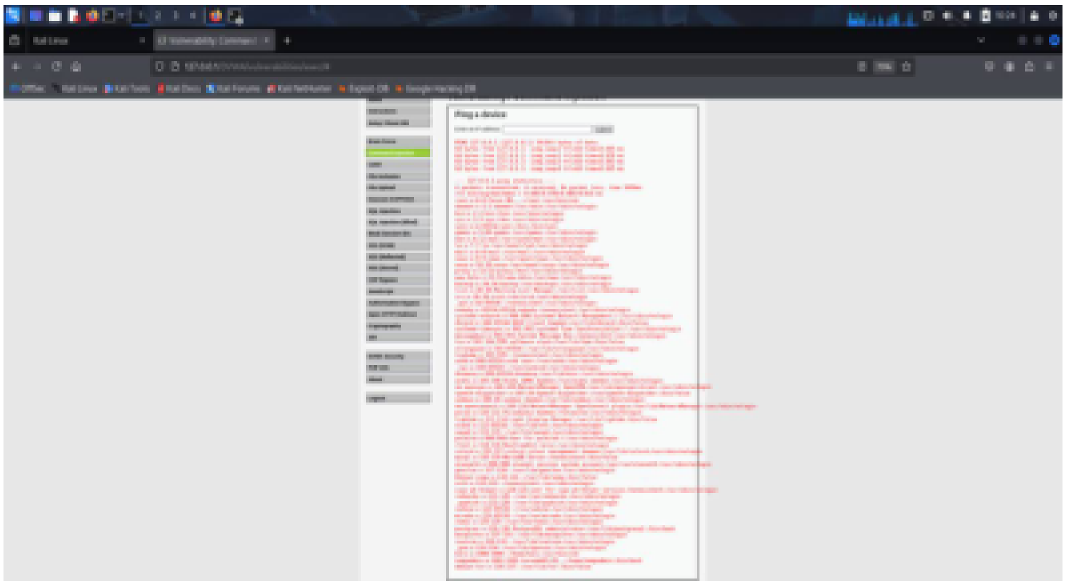
```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.213 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.030 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.025 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.029 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3066ms
rtt min/avg/max/mdev = 0.025/0.076/0.213/0.081 ms
help
index.php
source
```

More Information

- [illegible]
- [illegible]
- [illegible]
- [illegible]

Step 3: Further exploit the vulnerability by using '&& cat /etc/passwd' to read sensitive system files. The output will confirm successful exploitation.



Ping a device

[illegible]

DVWA - Stored Cross Site Scripting (XSS) Report

Objective

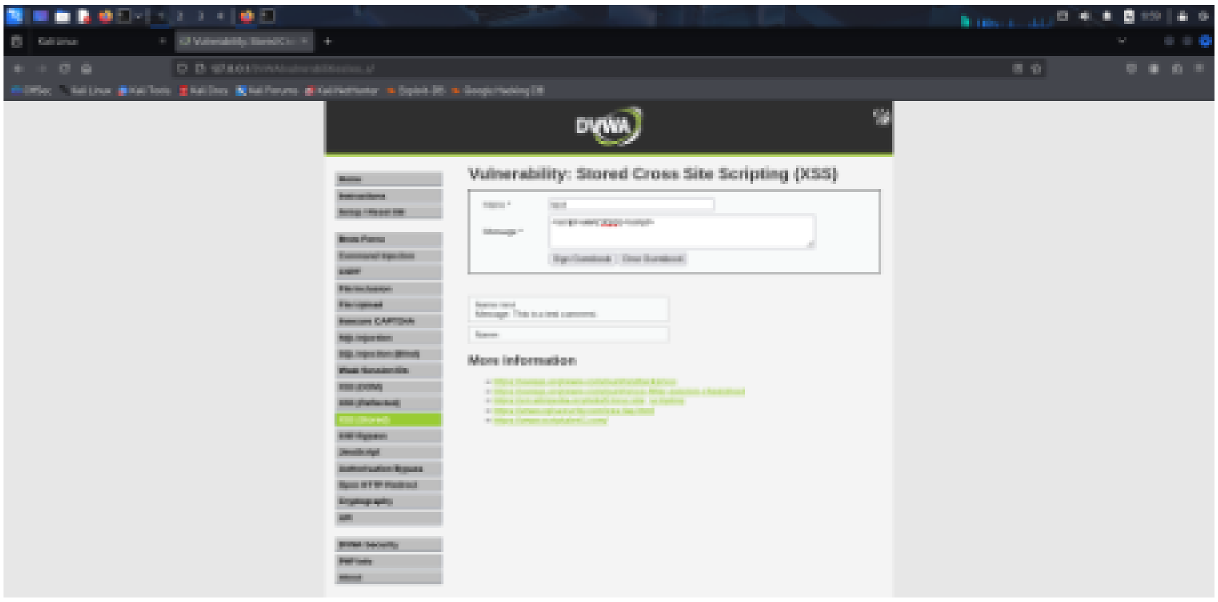
The objective of this task was to demonstrate Stored Cross Site Scripting (XSS) in DVWA under different security levels (Low and Medium). Stored XSS occurs when a malicious script is permanently stored on the target server and served to users whenever they access the stored data.

Low Security Level

1. Navigate to 'XSS (Stored)' in DVWA.
2. Enter a test payload in the 'Message' field.
3. Payload used:

<script>alert('XSS')</script>

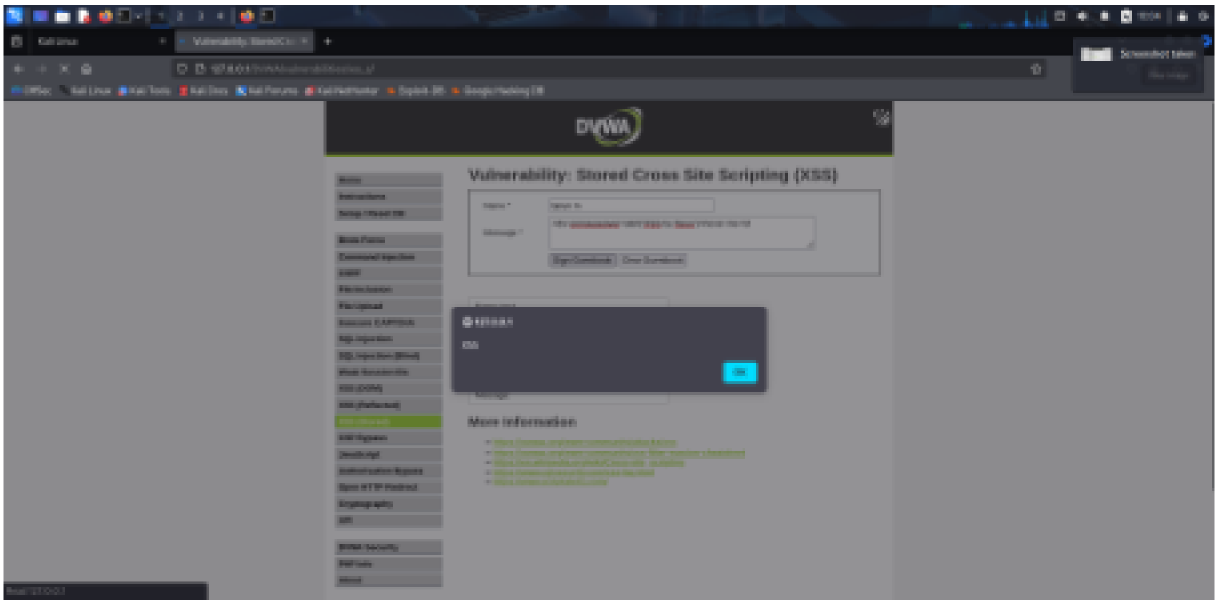
4. After submitting, the alert popup confirms the stored XSS execution.



Vulnerability: Stored Cross Site Scripting (XSS)

More Information

Screenshot: Entering the XSS payload in Low Security mode.



Vulnerability: Stored Cross Site Scripting (XSS)

More Information

DVWA - Stored Cross Site Scripting (XSS) Report

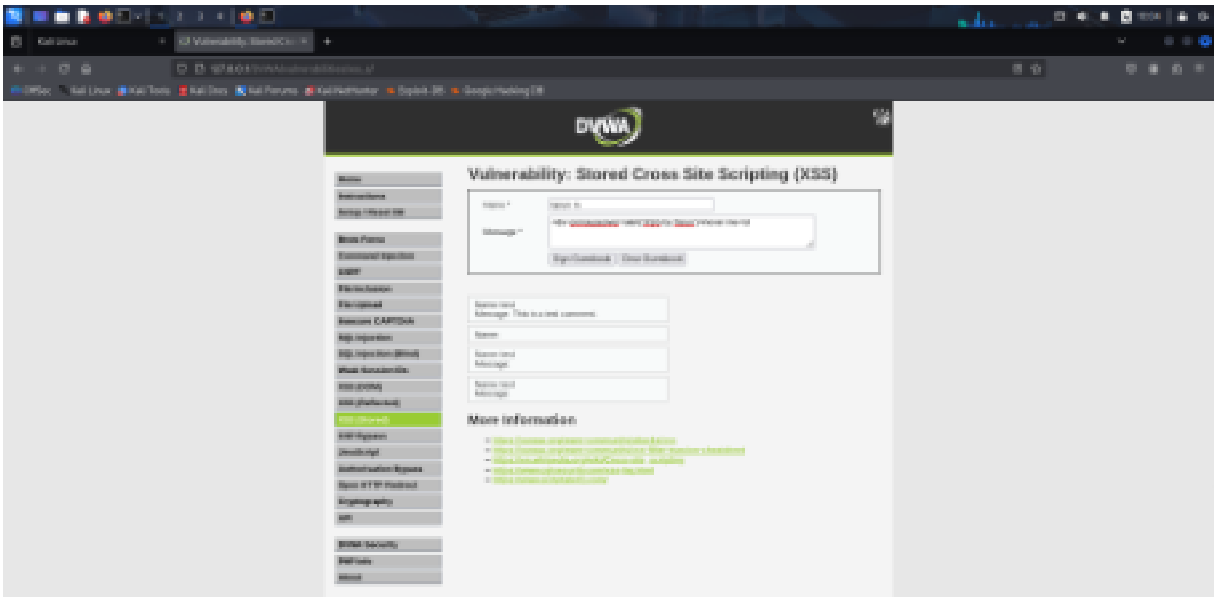
Screenshot: Alert popup triggered in Low Security mode.

Medium Security Level

1. Navigate to 'XSS (Stored)' in DVWA with security set to Medium.
2. The application may apply basic filtering; therefore, a different payload was used.
3. Payload used:

<div onmouseover=alert('XSS by Tarun')>hover me</div>

4. The payload triggers when the user hovers over the text.

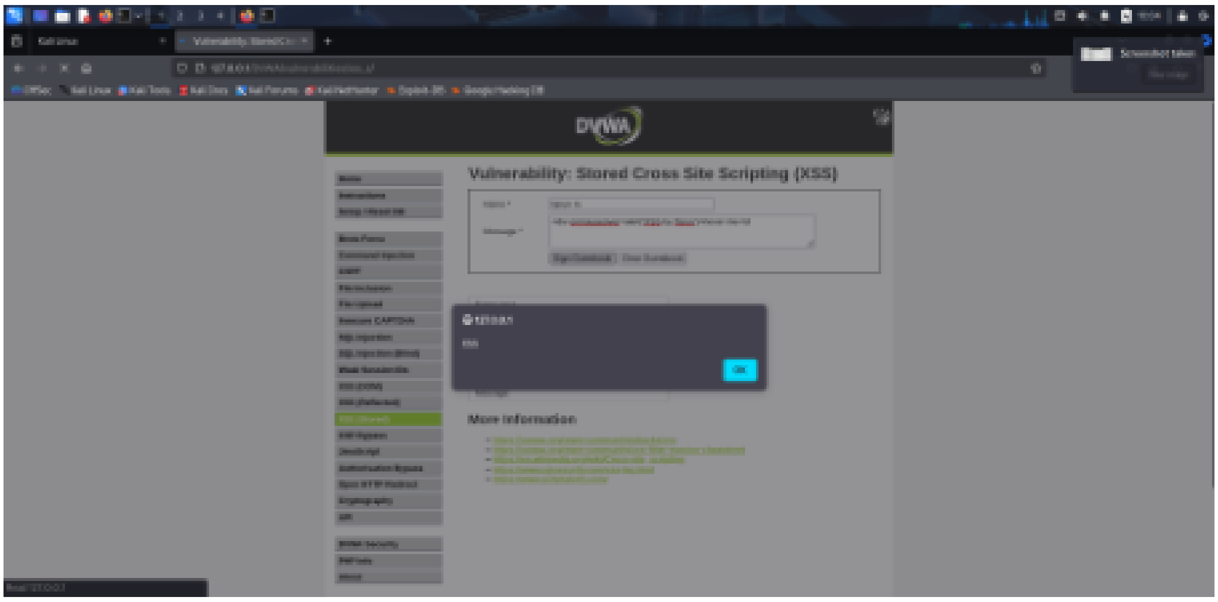


Vulnerability: Stored Cross Site Scripting (XSS)

Message: This is a test comment.

More Information

Screenshot: Entering the mouseover XSS payload in Medium Security mode.

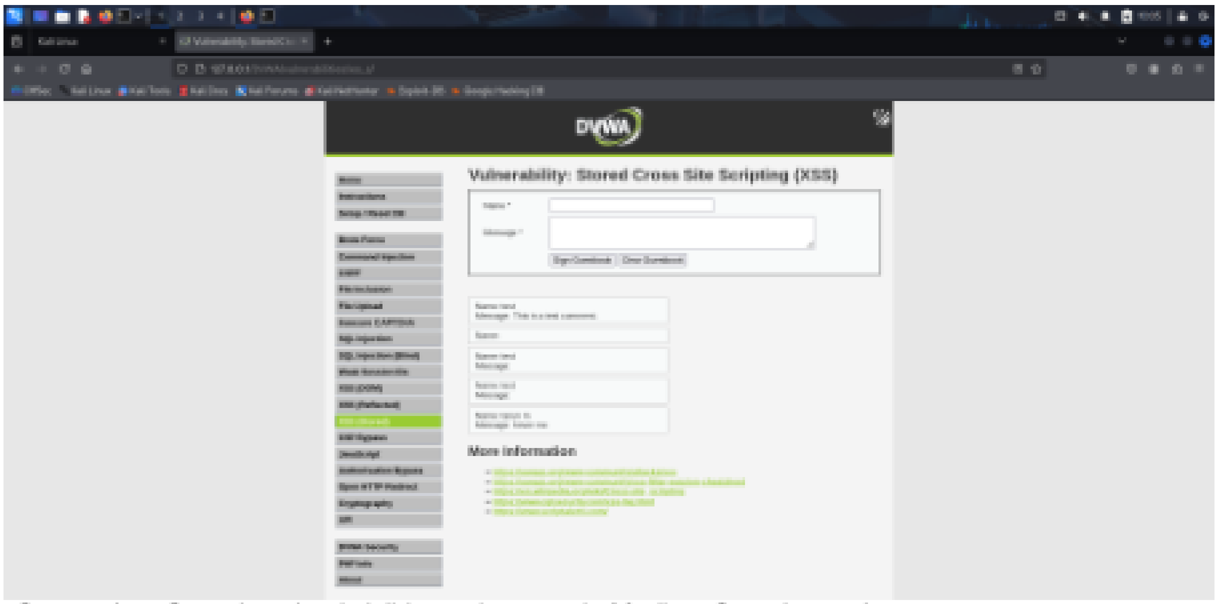


Vulnerability: Stored Cross Site Scripting (XSS)

More Information

Screenshot: Hovering over text triggers the XSS popup.

DVWA - Stored Cross Site Scripting (XSS) Report


Vulnerability: Stored Cross Site Scripting (XSS)

Name: test
Message: This is a test comment.

Name:

Name: test
Message:

Name: test
Message:

Name: [illegible]
Message: [illegible]

More Information

Screenshot: Stored payload visible on the page in Medium Security mode.

DVWA — SQL Injection Walkthrough

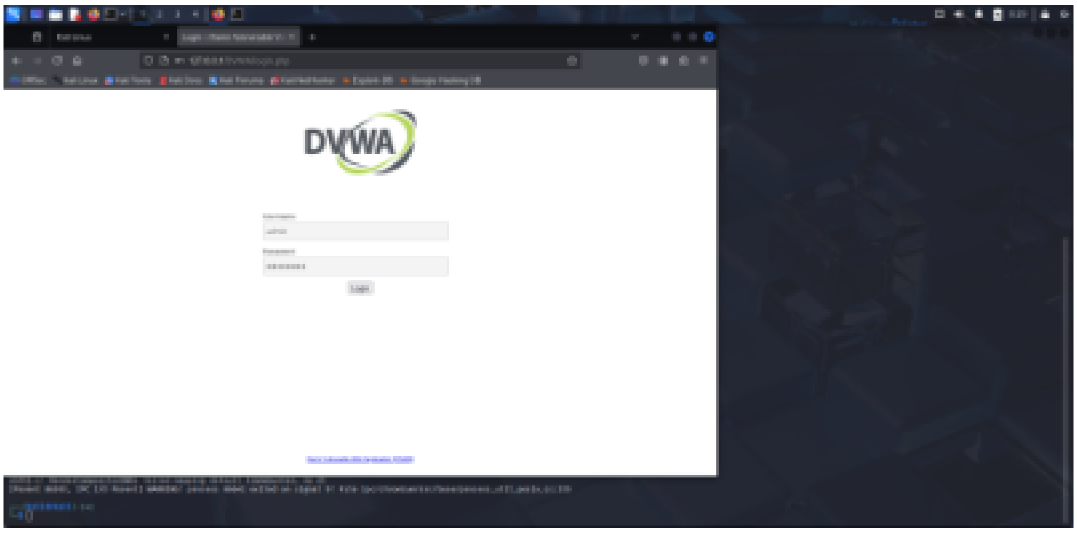
Prepared By: Tarun M
Date: 10/08/2025

Introduction

This document demonstrates a SQL Injection vulnerability test performed on DVWA (Damn Vulnerable Web Application) with security level set to LOW. Screenshots are presented in the correct step-by-step sequence along with explanations.

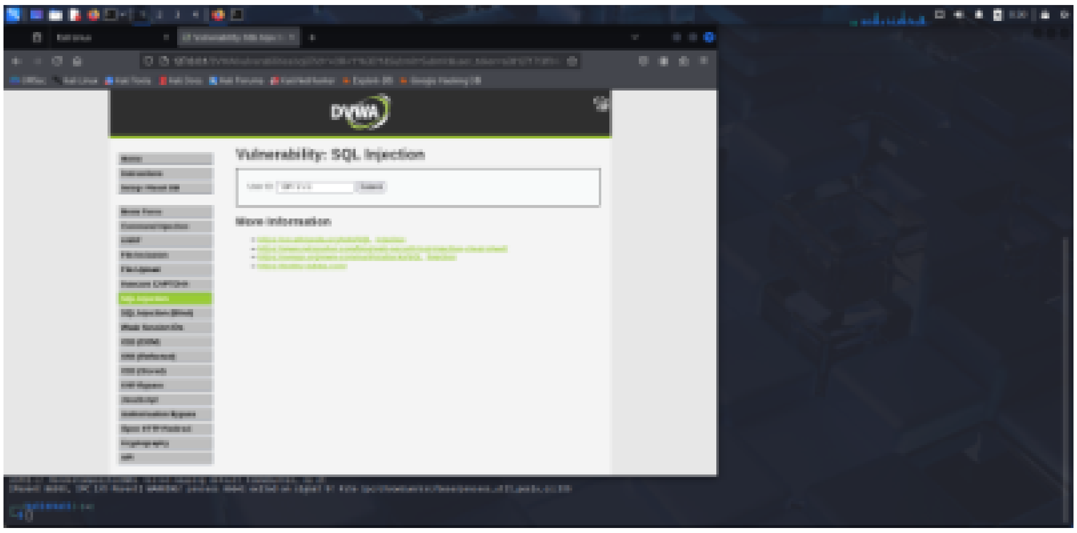
Step 1 — Injection Results

After submitting the payload, the application returned database contents, confirming successful SQL Injection.



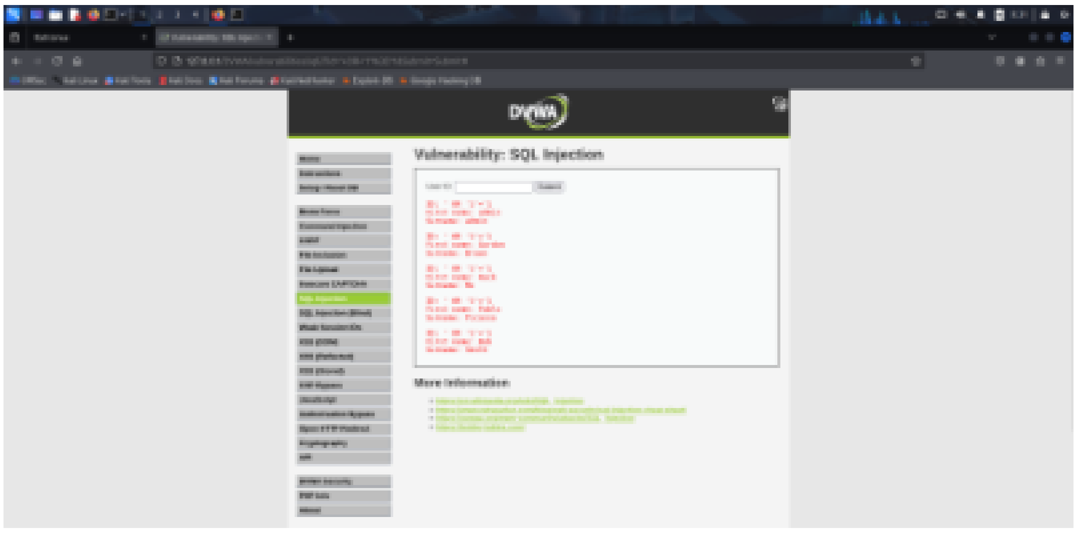
Step 2 — SQL Injection Payload

In the 'SQL Injection' module, the payload ' OR '1'='1 was entered to bypass authentication logic and retrieve all rows.



Step 3 — DVWA Login

Logged into DVWA using default credentials (admin / password) to access the vulnerable modules.



Impact

SQL Injection allows attackers to retrieve sensitive data, bypass authentication, and potentially compromise the entire database.

Mitigation

- Use parameterized queries (prepared statements) - Validate and sanitize all user inputs - Apply least privilege to database accounts - Deploy a Web Application Firewall (WAF) for extra protection

Penetration Testing Report

Prepared By: Tarun M
Date: 10/08/2025

1. Executive Summary

A penetration test was conducted against scanme.nmap.org (45.33.32.156) to identify open ports, running services, and potential vulnerabilities. The assessment revealed outdated services and unnecessary open ports. Remediation steps have been recommended to improve security posture.

2. Scope of Work

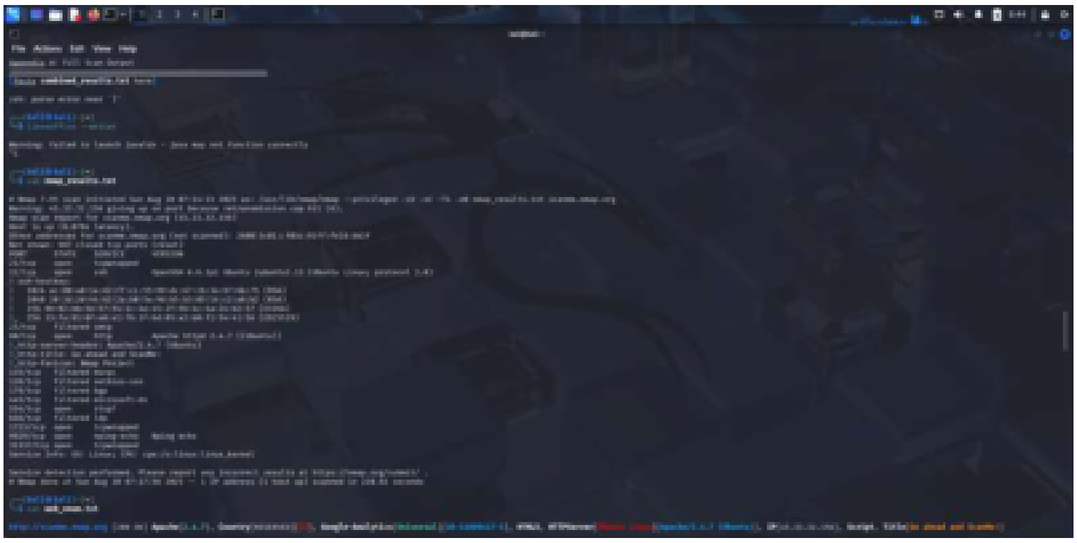
Target Host: scanme.nmap.org (45.33.32.156) Date of Testing: 10/08/2025 Testing Methodology: External reconnaissance and service enumeration Tools Used: Nmap, WhatWeb, Nikto, Gobuster, Searchsploit

3. Findings Summary

Port	State	Service	Version/Notes
22	Open	SSH	OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (outdated)
80	Open	HTTP	Apache httpd 2.4.7 ((Ubuntu)) – outdated
1723	Open	tcpwrapped	Service responded but details hidden
9929	Open	nping-echo	Nmap testing service
31337	Open	tcpwrapped	Service responded but details hidden

4. Detailed Methodology

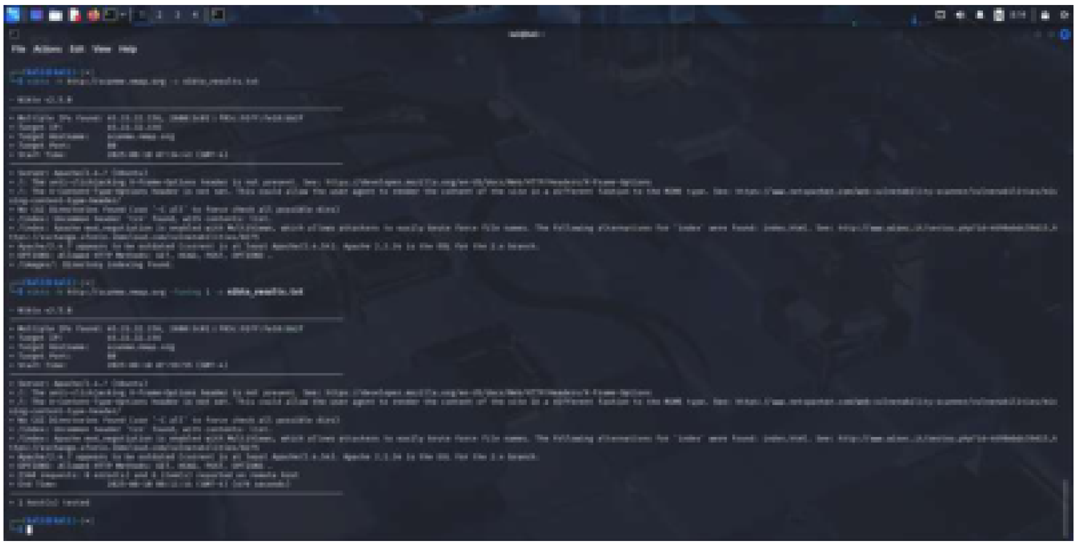
Step 1 – Nmap Service & Port Scan
Nmap was used to identify open ports, running services, and their versions. Command used: nmap -sV -sC -T4 scanme.nmap.org



```
[illegible]
```

Results indicated multiple open ports including SSH (OpenSSH 6.6.1p1) and HTTP (Apache 2.4.7), both outdated. Some ports were filtered, indicating firewall rules in place.

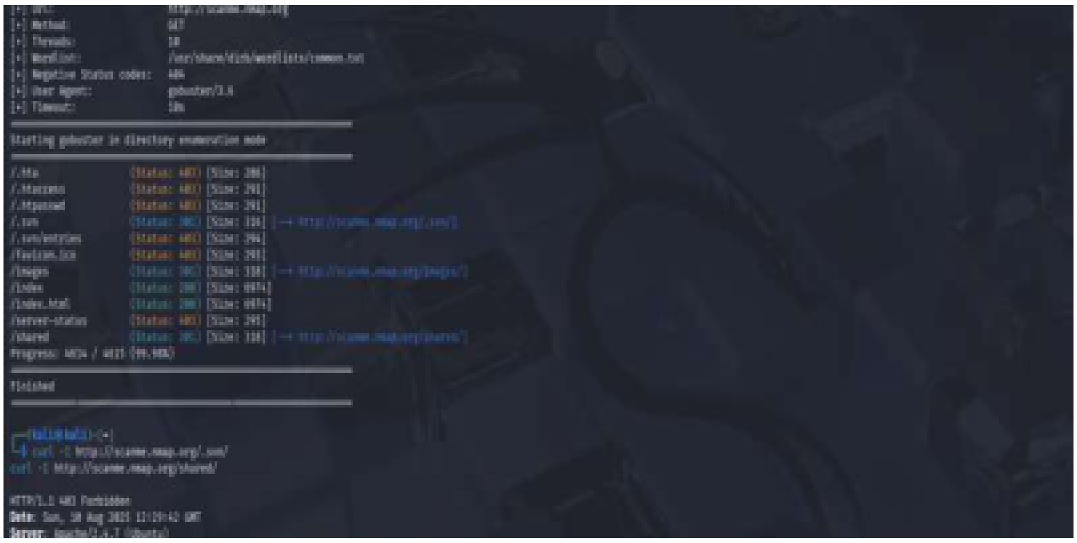
Step 2 – Web Vulnerability Scan (Nikto)
Nikto was used to identify web vulnerabilities and outdated software versions. Command used: nikto -h http://scanme.nmap.org



```
[illegible]
```

Nikto identified outdated Apache version and missing security headers (X-Frame-Options, X-Content-Type-Options). These issues may allow clickjacking or MIME-based attacks.

Step 3 – Directory Enumeration (Gobuster)
Gobuster was used to discover hidden directories and files. Command used: gobuster dir -u http://scanme.nmap.org -w /usr/share/wordlists/dirb/common.txt



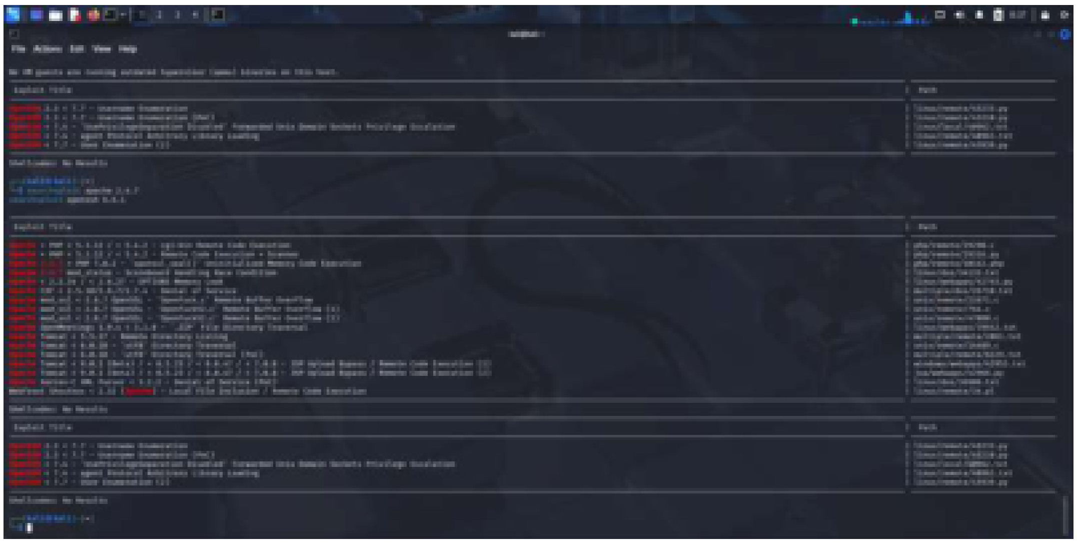
```
[+] Url:                     http://scanme.nmap.org
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/dirb/wordlists/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.hta                 (Status: 403) [Size: 288]
/.htaccess            (Status: 403) [Size: 291]
/.htpasswd            (Status: 403) [Size: 291]
/.svn                 (Status: 301) [Size: 316] [--> http://scanme.nmap.org/.svn/]
/.svn/entries         (Status: 403) [Size: 294]
/favicon.ico          (Status: 403) [Size: 291]
/images               (Status: 301) [Size: 318] [--> http://scanme.nmap.org/images/]
/index                (Status: 200) [Size: 6974]
/index.html           (Status: 200) [Size: 6974]
/server-status        (Status: 403) [Size: 295]
/shared               (Status: 301) [Size: 318] [--> http://scanme.nmap.org/shared/]
Progress: 4614 / 4615 (99.98%)
===============================================================
Finished
===============================================================

┌──(kali㉿kali)-[~]
└─$ curl -I http://scanme.nmap.org/.svn/
curl -I http://scanme.nmap.org/shared/

HTTP/1.1 403 Forbidden
Date: Sun, 10 Aug 2025 11:19:42 GMT
Server: Apache/2.4.7 (Ubuntu)
```

Directories such as /images/, /shared/, and /.svn/ were found but returned 403 Forbidden, indicating restricted access. These could leak data if misconfigured.

Step 4 – Exploit Search (Searchsploit)
Searchsploit was used to identify public exploits for detected software versions. Commands used: searchsploit apache 2.4.7 searchsploit openssh 6.6.1



```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
---------------------------------------------------------------------------------------------------- ---------------------------------
 Exploit Title                                                                                      |  Path
---------------------------------------------------------------------------------------------------- ---------------------------------
OpenSSH 2.3 < 7.7 - Username Enumeration                                                            | linux/remote/45233.py
OpenSSH 2.3 < 7.7 - Username Enumeration (PoC)                                                      | linux/remote/45210.py
OpenSSH < 7.4 - 'UsePrivilegeSeparation Disabled' Forwarded Unix Domain Sockets Privilege Escalation | linux/local/40962.txt
OpenSSH < 7.4 - agent Protocol Arbitrary Library Loading                                            | linux/remote/40963.txt
OpenSSH < 7.7 - User Enumeration (2)                                                                | linux/remote/45939.py
---------------------------------------------------------------------------------------------------- ---------------------------------
Shellcodes: No Results

┌──(kali㉿kali)-[~]
└─$ searchsploit apache 2.4.7
searchsploit openssh 6.6.1

---------------------------------------------------------------------------------------------------- ---------------------------------
 Exploit Title                                                                                      |  Path
---------------------------------------------------------------------------------------------------- ---------------------------------
Apache + PHP < 5.3.12 / < 5.4.2 - cgi-bin Remote Code Execution                                     | php/remote/29290.c
Apache + PHP < 5.3.12 / < 5.4.2 - Remote Code Execution + Scanner                                   | php/remote/29316.py
Apache 2.4.7 + PHP 7.0.2 - 'openssl_seal()' Uninitialized Memory Code Execution                     | php/remote/40142.php
Apache 2.4.7 mod_status - Scoreboard Handling Race Condition                                         | linux/dos/34133.txt
Apache < 2.2.34 / < 2.4.27 - OPTIONS Memory Leak                                                    | linux/webapps/42745.py
Apache CXF < 2.5.10/2.6.7/2.7.4 - Denial of Service                                                 | multiple/dos/26710.txt
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuck.c' Remote Buffer Overflow                                | unix/remote/21671.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (1)                          | unix/remote/764.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (2)                          | unix/remote/47080.c
Apache OpenMeetings 1.9.x < 3.1.0 - '.ZIP' File Directory Traversal                                 | linux/webapps/39642.txt
Apache Tomcat < 5.5.17 - Remote Directory Listing                                                   | multiple/remote/2061.txt
Apache Tomcat < 6.0.18 - 'utf8' Directory Traversal                                                 | unix/remote/14489.c
Apache Tomcat < 6.0.18 - 'utf8' Directory Traversal (PoC)                                           | multiple/remote/6229.txt
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Remote Code Execution (1) | windows/webapps/42953.txt
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Remote Code Execution (2) | jsp/webapps/42966.py
Apache Xerces-C XML Parser < 3.1.2 - Denial of Service (PoC)                                        | linux/dos/36906.txt
Webfroot Shoutbox < 2.32 (Apache) - Local File Inclusion / Remote Code Execution                    | linux/remote/34.pl
---------------------------------------------------------------------------------------------------- ---------------------------------
Shellcodes: No Results
---------------------------------------------------------------------------------------------------- ---------------------------------
 Exploit Title                                                                                      |  Path
---------------------------------------------------------------------------------------------------- ---------------------------------
OpenSSH 2.3 < 7.7 - Username Enumeration                                                            | linux/remote/45233.py
OpenSSH 2.3 < 7.7 - Username Enumeration (PoC)                                                      | linux/remote/45210.py
OpenSSH < 7.4 - 'UsePrivilegeSeparation Disabled' Forwarded Unix Domain Sockets Privilege Escalation | linux/local/40962.txt
OpenSSH < 7.4 - agent Protocol Arbitrary Library Loading                                            | linux/remote/40963.txt
OpenSSH < 7.7 - User Enumeration (2)                                                                | linux/remote/45939.py
---------------------------------------------------------------------------------------------------- ---------------------------------
Shellcodes: No Results

┌──(kali㉿kali)-[~]
└─$
```

Multiple exploits were available for both Apache and OpenSSH versions, emphasizing the need for patching and upgrades.

5. Recommendations

- Update Apache HTTP Server to the latest stable release. - Upgrade OpenSSH to a supported version. - Close unused ports at the firewall. - Implement HTTP security headers. - Conduct regular vulnerability scans and patch management.

6. Skills Demonstrated

- Network scanning & enumeration (Nmap) - Web vulnerability scanning (Nikto) - Directory brute forcing (Gobuster) - Web technology fingerprinting (WhatWeb) - Exploit research (Searchsploit) - Report writing and documentation