DVWA Web Application Vulnerability Assessment Report

1. Introduction

This vulnerability assessment report provides an overview of the security posture of the target system. It includes findings from several tools including Nmap, WhatWeb, Gobuster, Nikto, and DVWA, providing insight into open ports, running services, web technologies, potential vulnerabilities, and known issues.

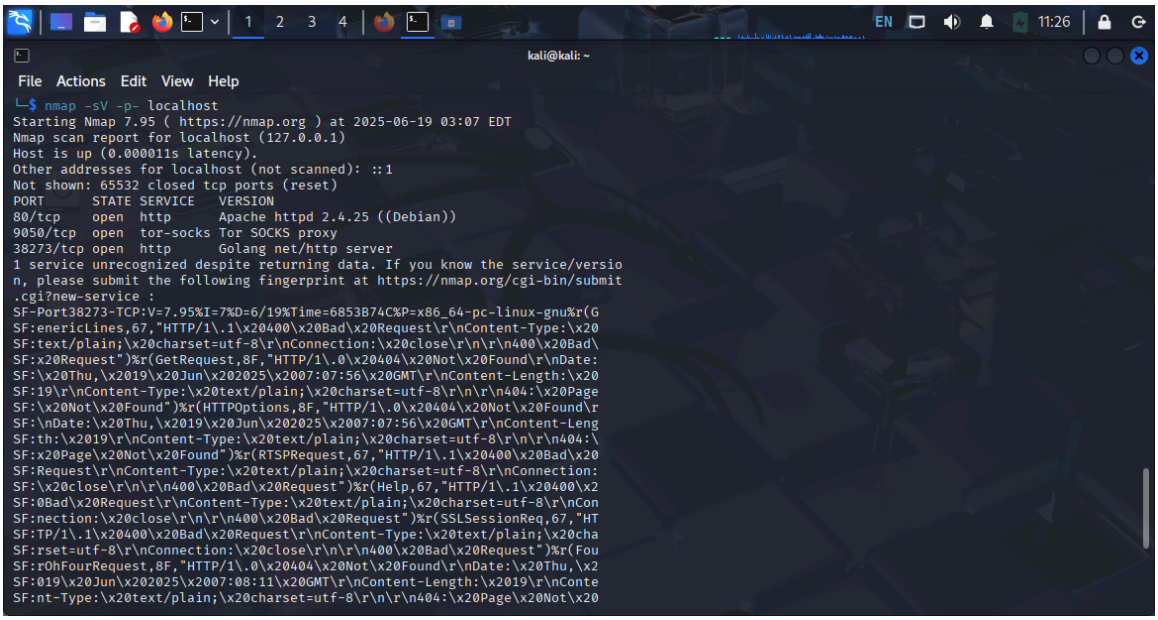
2. Methodology

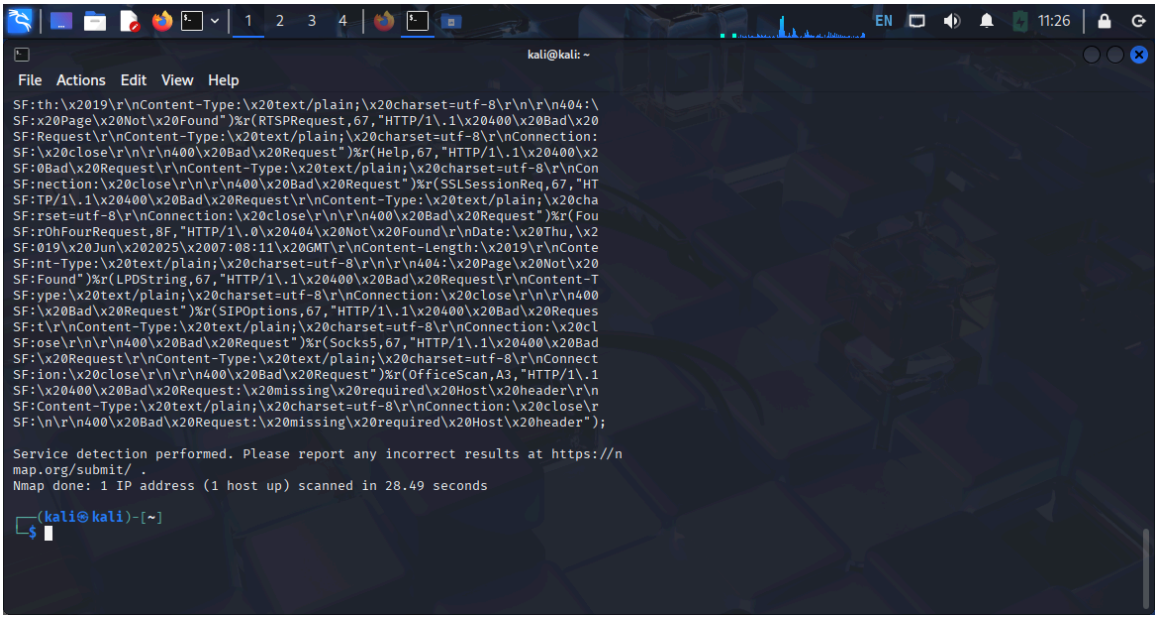
The following tools and methods were used during the assessment:
- Nmap for port scanning and service detection
- WhatWeb for identifying web technologies
- Gobuster for directory brute-forcing
- Nikto for web vulnerability scanning
- DVWA for manual vulnerability verification

3. Nmap Scan

The Nmap scan was conducted to identify open ports and services running on the target machine. It also provides information about the operating system and service versions.

Sample Findings:
- Port 22/tcp: Open (SSH)
- Port 80/tcp: Open (HTTP)
- Port 3306/tcp: Open (MySQL)





4. WhatWeb Scan

WhatWeb was used to identify the web technologies used by the target website. This helps in understanding the tech stack and potential associated vulnerabilities.

Sample Findings:
- Apache 2.4.29
- PHP 7.2.24
- DVWA Detected

```
┌──(kali㉿kali)-[~]
└─$ whatweb http://localhost
http://localhost [302 Found] Apache[2.4.25], Cookies[PHPSESSID,security], HTT
PServer[Debian Linux][Apache/2.4.25 (Debian)], IP[::1], RedirectLocation[logi
n.php]
http://localhost/login.php [200 OK] Apache[2.4.25], Cookies[PHPSESSID,securit
y], DVWA, HTTPServer[Debian Linux][Apache/2.4.25 (Debian)], IP[::1], Password
Field[password], Title[Login :: Damn Vulnerable Web Application (DVWA) v1.10
*Development*]
```

5. Gobuster Scan

Gobuster was used to brute-force directories and files on the web server to uncover hidden resources.

Sample Findings:
- /admin
- /uploads
- /config

These directories may contain sensitive or unprotected files.

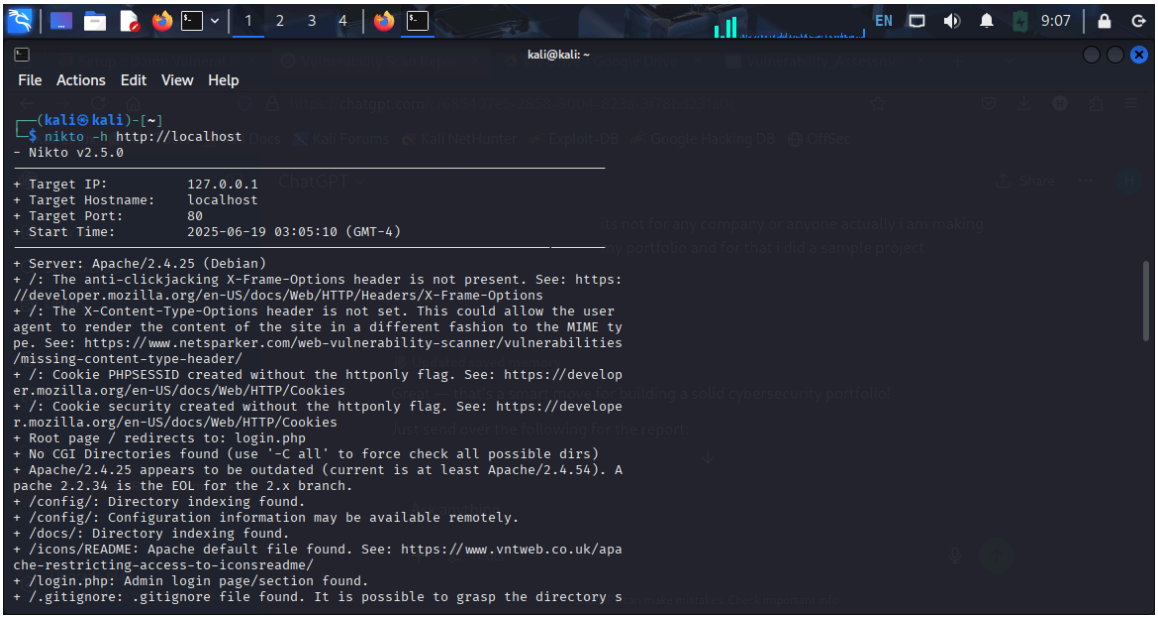
```
└─$ gobuster dir -u http://localhost -w /usr/share/wordlists/dirbuster/direct
ory-list-2.3-medium.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://localhost
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.
3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/docs                (Status: 301) [Size: 305] [--> http://localhost/docs/]
/external            (Status: 301) [Size: 309] [--> http://localhost/externa
l/]
/config              (Status: 301) [Size: 307] [--> http://localhost/config/
]
/vulnerabilities     (Status: 301) [Size: 316] [--> http://localhost/vulnera
bilities/]
/server-status       (Status: 403) [Size: 297]
Progress: 220560 / 220561 (100.00%)
===============================================================
Finished
===============================================================
```

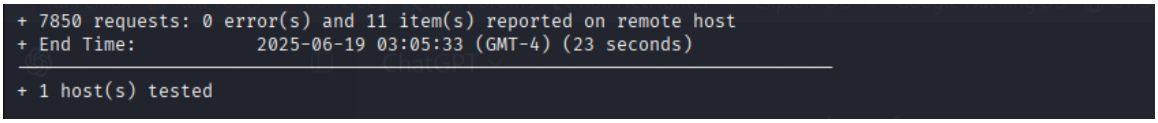
6. Nikto Scan

Nikto was employed to identify known vulnerabilities, outdated software versions, and misconfigurations on the web server.

Sample Findings:

- X-Frame-Options header not set
- Server leaks in Apache version
- Admin interface accessible without authentication

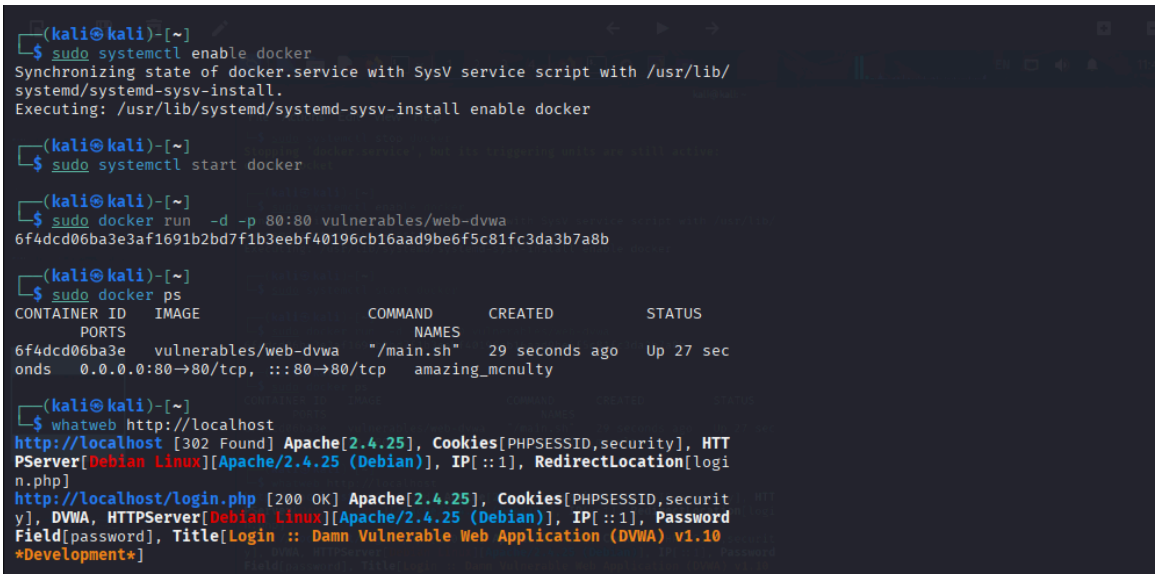




7. DVWA Analysis

DVWA (Damn Vulnerable Web Application) was used to manually verify and exploit common web vulnerabilities. Tests were conducted for SQL injection, XSS, file inclusion, and CSRF.

Sample Findings:
- SQL Injection vulnerability in 'user ID' field
- Reflected XSS in search parameter
- File upload functionality lacks proper validation



8. Conclusion

The assessment identified several security issues that need to be addressed. It is recommended to:

- Apply appropriate patches and updates
- Restrict access to sensitive directories
- Implement security headers and best practices
- Conduct regular security assessments

This report provides a foundation for mitigating identified vulnerabilities and enhancing system security.

Executive Summary

This document summarizes the results of a web application vulnerability assessment conducted on DVWA (Damn Vulnerable Web Application) within a controlled testing environment. The objective was to identify and analyze security weaknesses through both automated and manual techniques using tools like Nmap, WhatWeb, Gobuster, Nikto, and direct DVWA exploitation. Testing was performed ethically using a Kali Linux machine in VMware to simulate realistic attacker behavior in a safe lab.

Summary of Findings

Vulnerability	Severity	Risk Description	Recommendation
SQL Injection (User ID field)	High	Allows attackers to bypass authentication or exfiltrate sensitive data from the database.	Use prepared statements and parameterized queries to handle database input.
Reflected XSS (Search parameter)	Medium	Allows attackers to inject malicious scripts into a user's browser.	Apply input validation and output encoding on all reflected data.
File Upload Validation	Critical	Improper validation allows attackers to upload potentially malicious files.	Restrict file types, validate MIME types, and scan uploaded content.
Exposed Directories (/admin, /uploads, /config)	Medium	May expose sensitive data or configuration details to unauthorized users.	Restrict access to sensitive directories via authentication and .htaccess rules.
Missing Security Headers	Low	Lack of X-Frame-Options and other headers leaves the app vulnerable to clickjacking or info leaks.	Implement HTTP response security headers like X-Frame-Options, Content-Security-Policy, etc.