Vulnerability Assessment and Exploitation Report |Oluwaseun Quadri

1. Introduction

This assessment was conducted in a controlled laboratory environment to identify vulnerabilities present in the target system hosting the Damn Vulnerable Web Application (DVWA). The objective was to perform vulnerability scanning, exploit at least one discovered vulnerability, and provide remediation recommendations.

2. Tools Used

The following tools were used during the assessment:

- **Nmap** – for network discovery, port scanning, and service identification
- **Nikto** – for web server vulnerability scanning
- **Metasploit Framework** – for exploitation validation
- **DVWA (Damn Vulnerable Web Application)** – intentionally vulnerable test environment

Reason for Using Nmap and Nikto Instead of OpenVAS

OpenVAS was initially selected for vulnerability scanning; however, the vulnerability feed synchronization process did not complete after several hours of waiting. Since OpenVAS requires updated feeds to perform scans effectively, alternative scanning tools (Nmap and Nikto) were used to proceed with the vulnerability assessment while maintaining the project timeline. These tools provided sufficient vulnerability discovery capability for the controlled lab environment.

3. Nmap Scan Findings

An Nmap scan was conducted to identify open ports and running services on the target host.

Command used:

```
nmap -sV --script vuln 192.168.56.102
```

Key findings:

- Port 80 (HTTP) open – Apache Web Server detected
- Service version disclosure identified
- Potential web-based vulnerabilities detected through script scanning

This information helped identify the web server as the primary attack surface

4. Nikto Web Vulnerability Scan Findings

A Nikto scan was performed to identify web server misconfigurations and vulnerabilities.

Command used:

nikto -h http://192.168.56.102/DVWA

Key vulnerabilities identified:

- Missing X-Frame-Options header (Clickjacking risk)
- Missing X-Content-Type-Options header
- Directory indexing enabled in multiple directories
- Exposure of configuration and repository files (.git directory)
- Web application login interface exposed

These issues indicate insecure server configuration and potential information disclosure risks.

5. Exploitation Performed

A command injection vulnerability in DVWA was successfully exploited using the application's vulnerable input field. By injecting operating system commands into the application input, unauthorized command execution was achieved, confirming the presence of an input validation vulnerability.

Additionally, Metasploit Framework was used to interact with the target system to validate exploitability within the penetration testing workflow.

6. Risk Impact

Successful exploitation of the identified vulnerabilities could allow attackers to:

- Execute arbitrary commands on the server
- Access sensitive configuration data
- Enumerate system information
- Potentially gain further system control

7. Mitigation and Remediation Recommendations

To reduce the identified risks, the following actions are recommended:

1. **Input Validation**
    - Implement strict server-side input validation and sanitization

- Use parameterized commands and avoid direct command execution from user input

2. **Secure HTTP Headers**
    - Enable security headers including:
        - X-Frame-Options
        - X-Content-Type-Options
        - Content-Security-Policy
3. **Disable Directory Indexing**
    - Prevent directory browsing in the web server configuration
4. **Remove Sensitive Files**
    - Remove exposed repository files (.git directories)
    - Restrict access to configuration folders

8. Conclusion

The vulnerability assessment demonstrated that the target system contains multiple web application misconfigurations and input validation weaknesses that allow command execution. Implementing proper input validation, secure server configuration, and regular patch management will significantly reduce the risk of compromise.

SCREENSHOTS

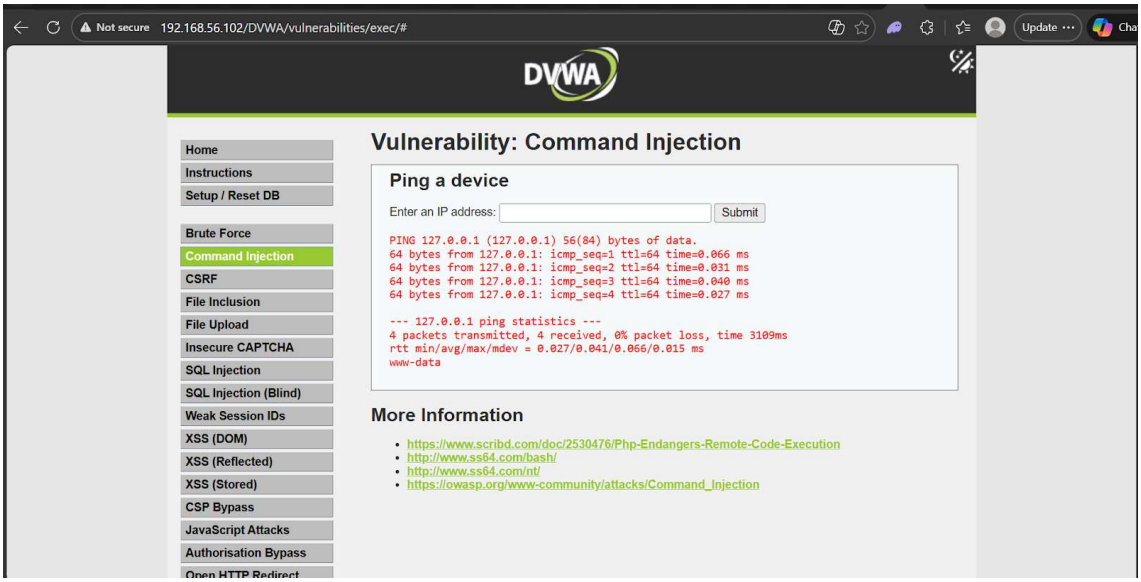

Fig.1 image indicating running of a vulnerability with DVWA

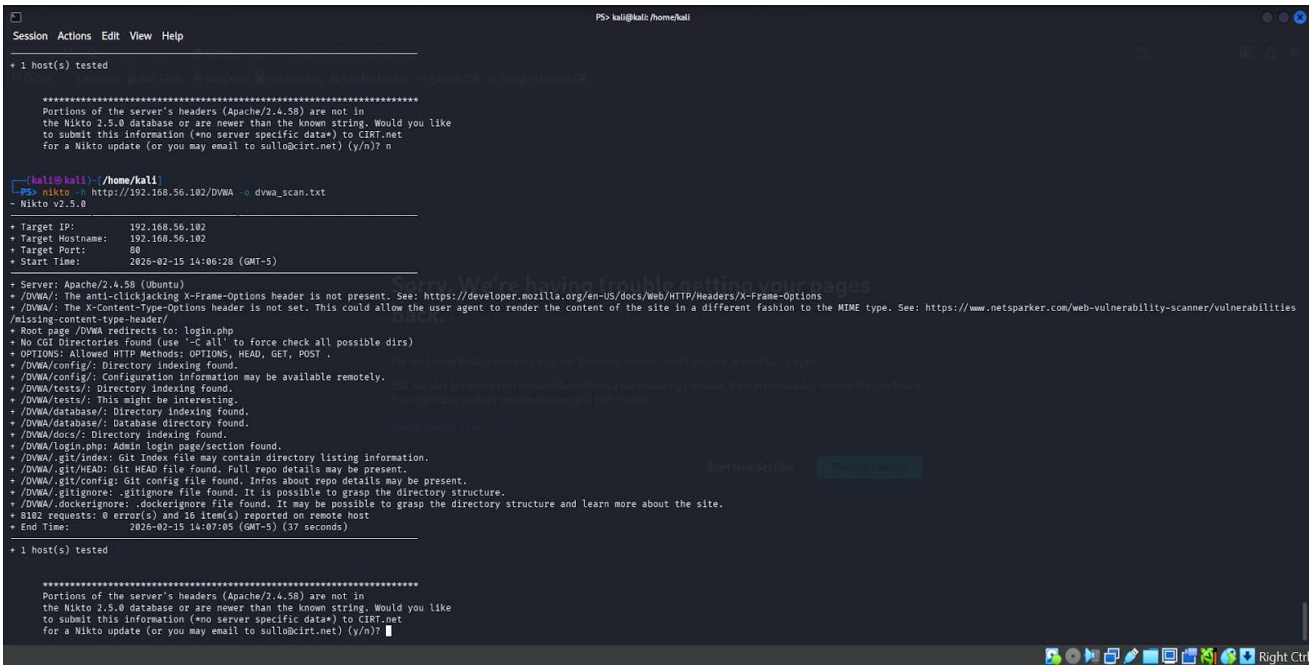

Fig.2 image indicating NIkto scanning sql headers

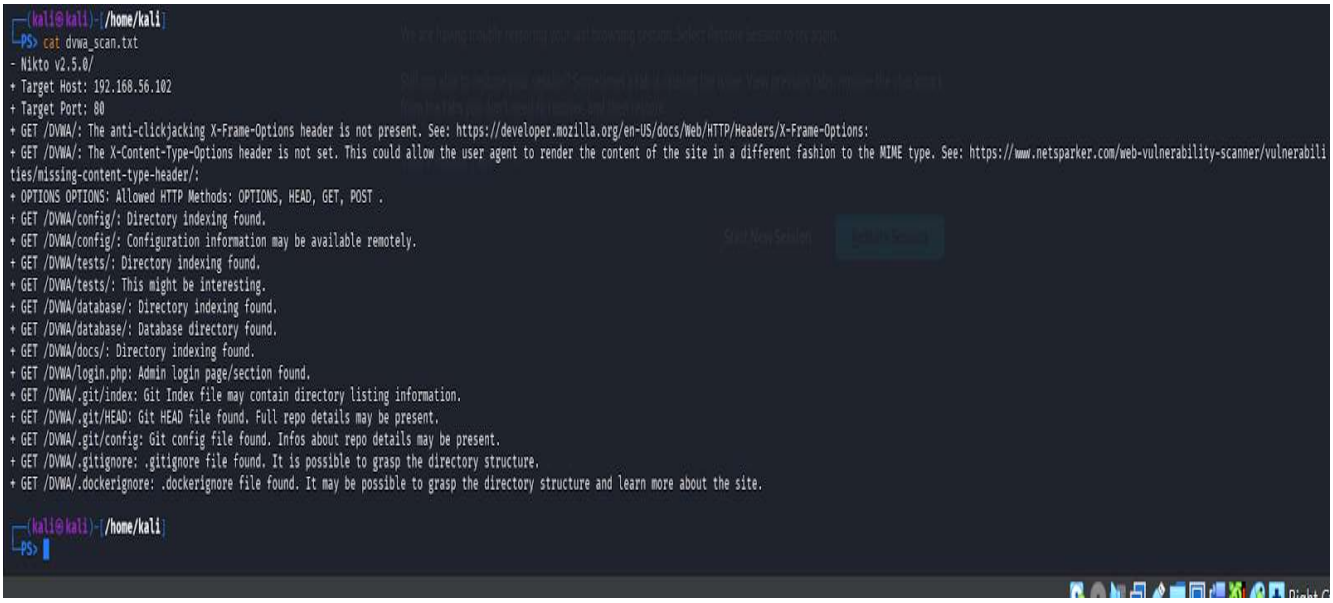

Fig.3 image indicating Nikto scanning sql headers

Fig.4 image indicating the use of Nmap for scanning of the ports

Fig. 5. Image indicating Nmap scanning the ports

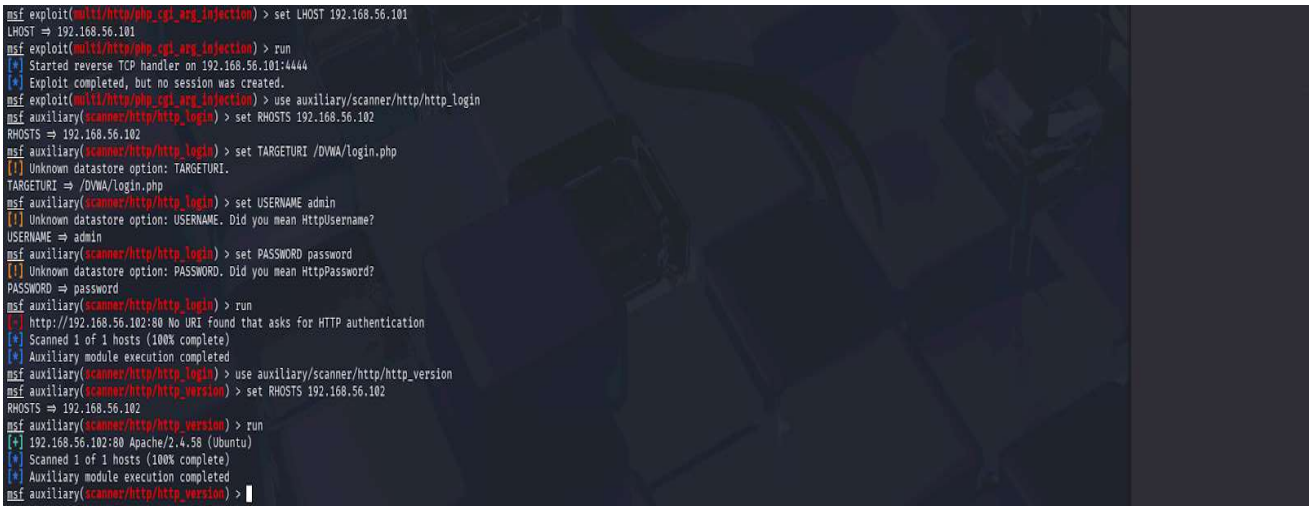

Fig.6 image Indicating metasploit exploitation validation