Web Application Security Testing Report

Intern Name: Ch.OmkarLakshmi

Project Title: Web Application Security Testing
Tool Used: Damn Vulnerable Web Application (DVWA)
Track id:FUTURE_CS_01

1. Introduction

The purpose of this security assessment was to analyze the **Damn Vulnerable Web Application (DVWA)** for weaknesses that could be exploited by attackers and to suggest effective countermeasures. The evaluation was limited to three common yet high-risk areas: **Brute Force attacks, SQL Injection (both classical and blind), and Cross-Site Scripting (XSS in its DOM-based, Reflected, and Stored forms)**.

To perform the assessment, a set of widely recognized penetration testing tools was used, including **Burp Suite, SQLMap, FoxyProxy, OWASP ZAP,** and a **Kali Linux virtualized environment**.

2. Methodology

The assessment was carried out using the following structured approach:

1. **Environment Setup:** DVWA was installed and configured on a Kali Linux virtual machine, with its database initialized and hosted locally.
2. **Attack Execution:** Simulated attacks were launched against authentication, database, and client-side input components to replicate real-world exploitation.
3. **Vulnerability Verification:** The results were confirmed through analysis of application responses, browser behavior, and database interactions.
4. **Mapping & Mitigation:** Each vulnerability was aligned with the **OWASP Top 10** security categories, and suitable remediation techniques were identified.

3. Key Findings

3.1 Brute Force Attack

- **Observation:** The login mechanism lacked protective measures, enabling automated tools like Burp Suite Intruder to guess valid credentials.
- **Impact:** Exploitation could allow attackers to gain unauthorized access.

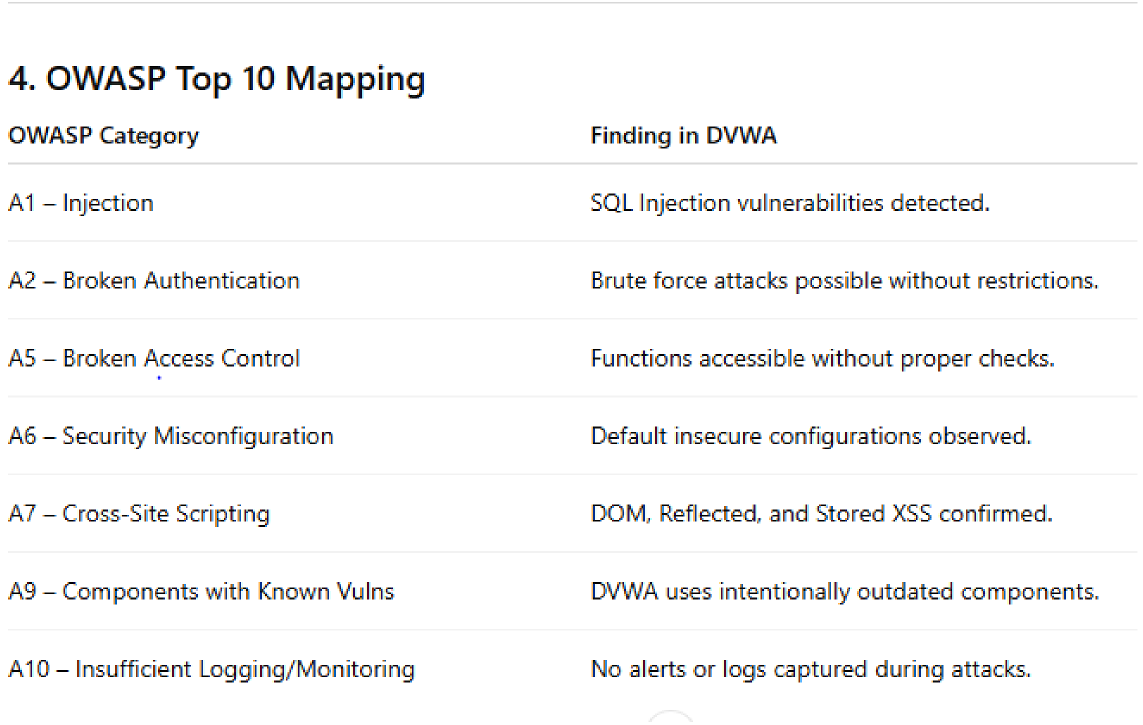
3.2 SQL Injection

- **Normal SQL Injection:** Login could be bypassed using a basic payload (' OR '1'='1).
- **Blind SQL Injection:** SQLMap was able to enumerate the database, extract sensitive data (such as user credentials), and crack hashed passwords.

- **Impact:** A complete compromise of the database could occur, exposing confidential information.

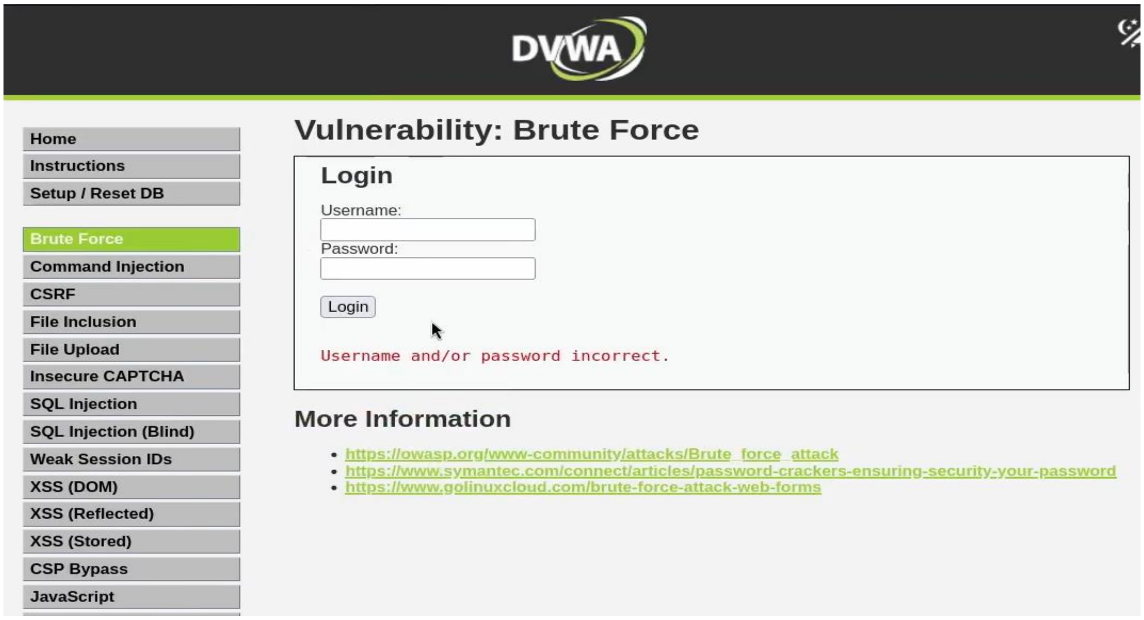
3.3 Cross-Site Scripting (XSS)

- **DOM-based XSS:** Unsanitized JavaScript allowed arbitrary script execution (alert(document.cookie)).
- **Reflected XSS:** Input provided by the user was directly echoed back, resulting in script execution in the browser.
- **Stored XSS:** Malicious input persisted in the database and was executed when other users viewed the page.
- **Impact:** Exploitation could result in session hijacking, impersonation, and leakage of sensitive information.



4. OWASP Top 10 Mapping

OWASP Category	Finding in DVWA
A1 – Injection	SQL Injection vulnerabilities detected.
A2 – Broken Authentication	Brute force attacks possible without restrictions.
A5 – Broken Access Control	Functions accessible without proper checks.
A6 – Security Misconfiguration	Default insecure configurations observed.
A7 – Cross-Site Scripting	DOM, Reflected, and Stored XSS confirmed.
A9 – Components with Known Vulns	DVWA uses intentionally outdated components.
A10 – Insufficient Logging/Monitoring	No alerts or logs captured during attacks.



Vulnerability: Brute Force

Login

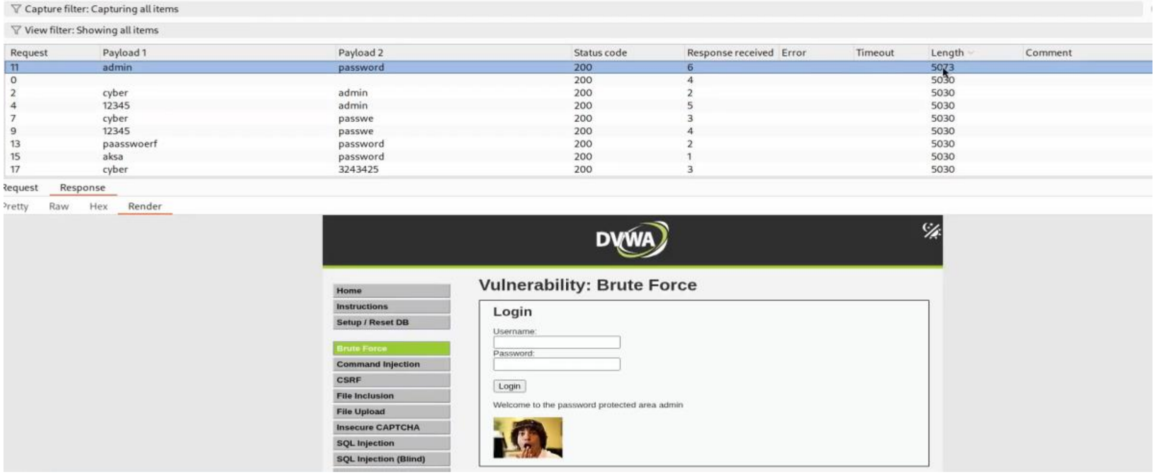
Username:

Password:

Username and/or password incorrect.

More Information

- https://owasp.org/www-community/attacks/Brute_force_attack
- https://www.symantec.com/connect/articles/password-crackers-ensuring-security-your-password
- https://www.golinuxcloud.com/brute-force-attack-web-forms



Capture filter: Capturing all items

View filter: Showing all items

Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	Comment
11	admin	password	200	6			5073	
0			200	4			5030	
2	cyber	admin	200	2			5030	
4	12345	admin	200	5			5030	
7	cyber	passwe	200	3			5030	
9	12345	passwe	200	4			5030	
13	paasswoerf	password	200	2			5030	
15	aksa	password	200	1			5030	
17	cyber	3243425	200	3			5030	

Vulnerability: Brute Force

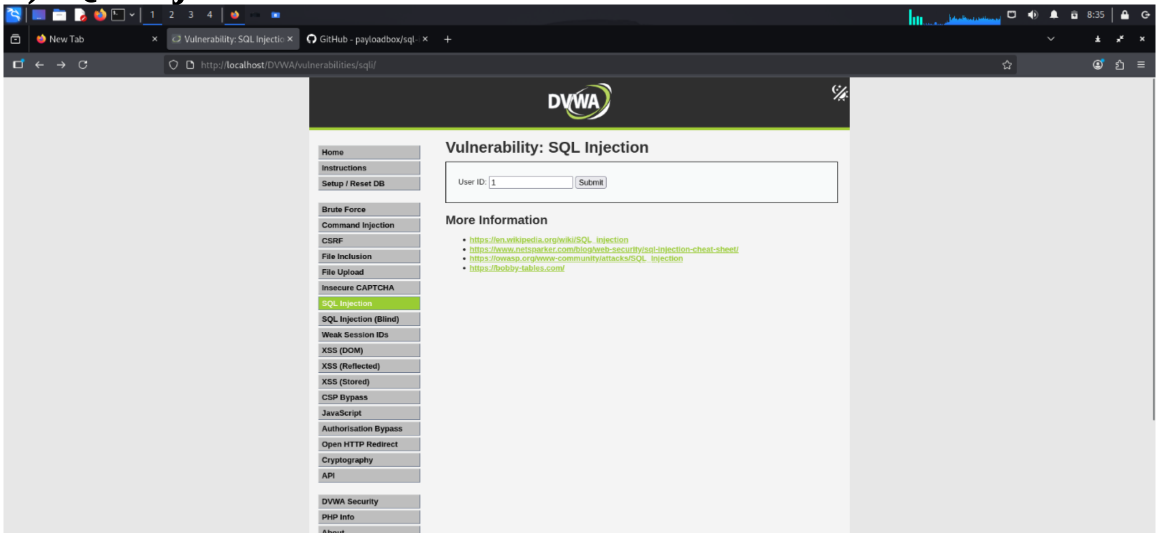
Login

Username:

Password:

Welcome to the password protected area admin

b)SQL injection



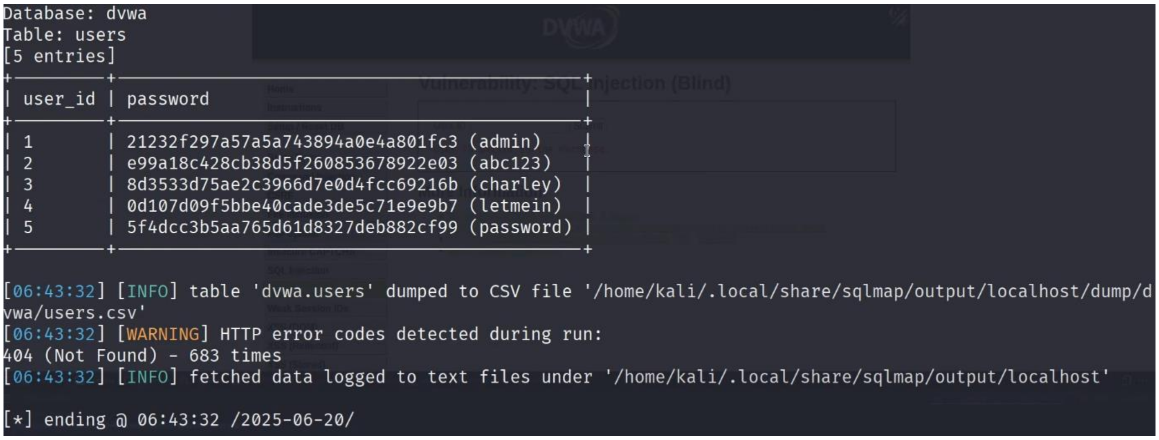
http://localhost/DVWA/vulnerabilities/sqli/

Vulnerability: SQL Injection

User ID: 1

More Information

- https://en.wikipedia.org/wiki/SQL_injection
- https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/
- https://owasp.org/www-community/attacks/SQL_injection
- https://bobby-tables.com/

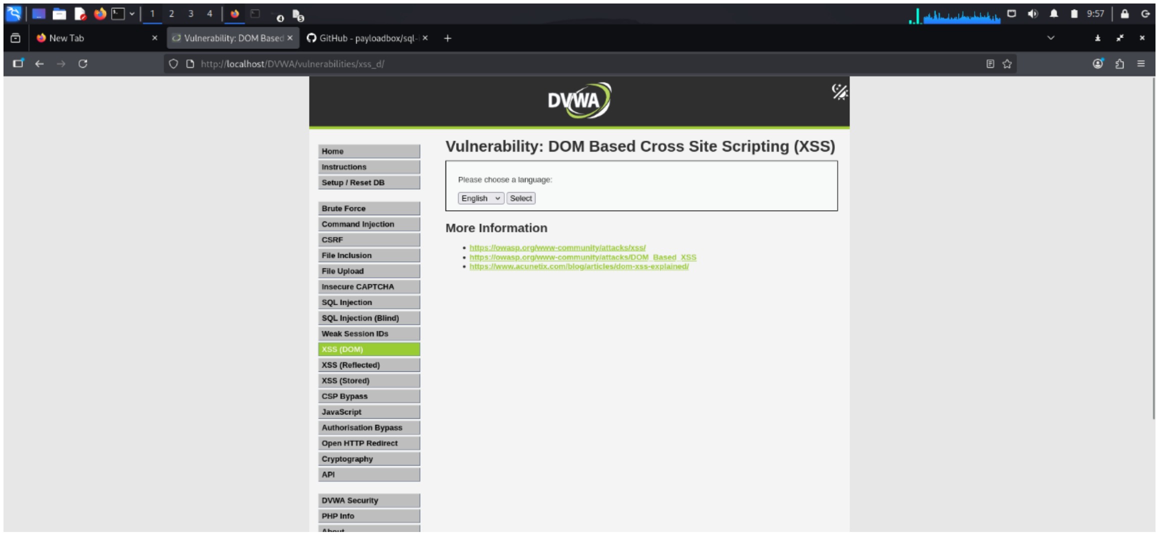


```
Database: dvwa
Table: users
[5 entries]
+---------+---------------------------------------------+
| user_id | password                                    |
+---------+---------------------------------------------+
| 1       | 21232f297a57a5a743894a0e4a801fc3 (admin)    |
| 2       | e99a18c428cb38d5f260853678922e03 (abc123)   |
| 3       | 8d3533d75ae2c3966d7e0d4fcc69216b (charley)  |
| 4       | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein)  |
| 5       | 5f4dcc3b5aa765d61d8327deb882cf99 (password) |
+---------+---------------------------------------------+

[06:43:32] [INFO] table 'dvwa.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/localhost/dump/d
vwa/users.csv'
[06:43:32] [WARNING] HTTP error codes detected during run:
404 (Not Found) - 683 times
[06:43:32] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/localhost'

[*] ending @ 06:43:32 /2025-06-20/
```

a) Vulnerability: DOM Based XSS



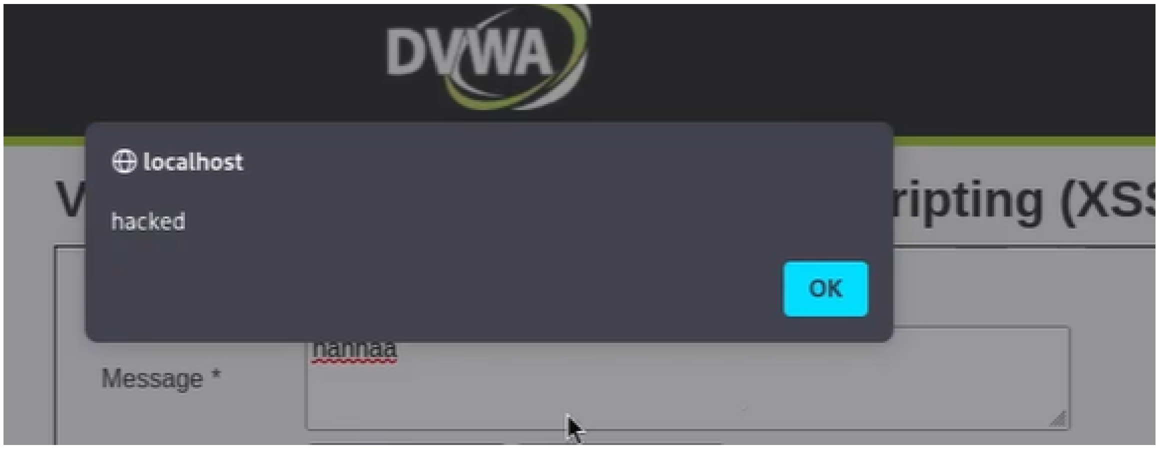
Vulnerability: DOM Based Cross Site Scripting (XSS)

Please choose a language:

English Select

More Information

- https://owasp.org/www-community/attacks/xss/
- https://owasp.org/www-community/attacks/DOM_Based_XSS
- https://www.acunetix.com/blog/articles/dom-xss-explained/

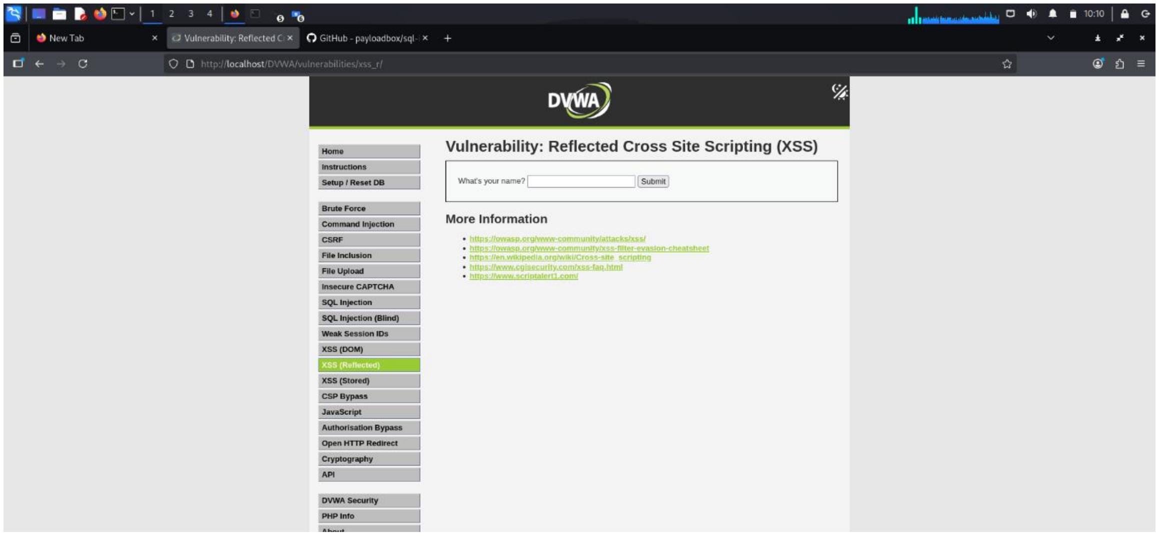


localhost

hacked

Message * hannaa

b) Vulnerability: Reflected XSS



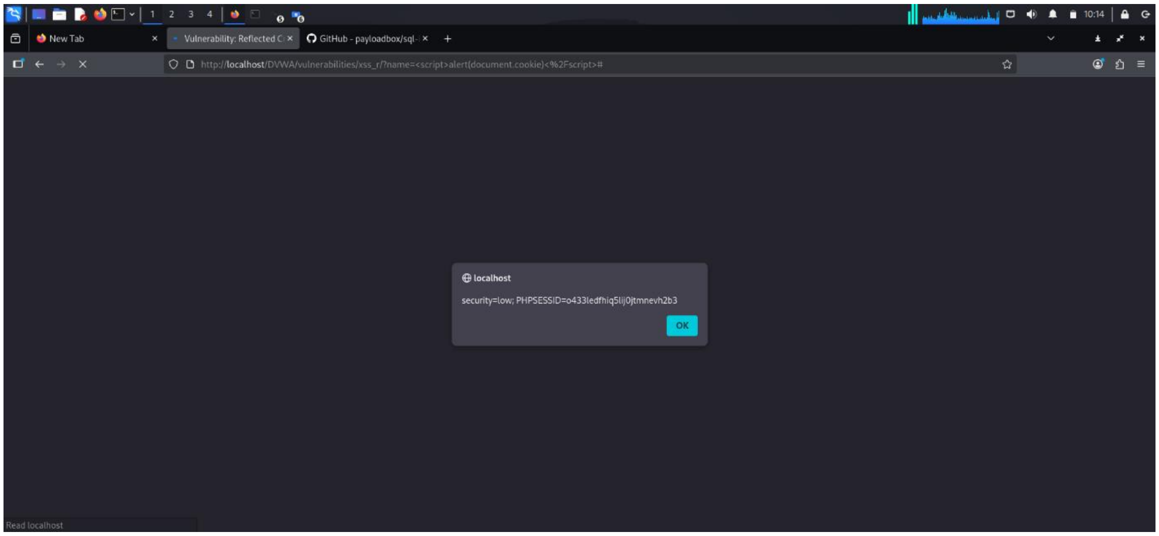
http://localhost/DVWA/vulnerabilities/xss_r/

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

More Information

- https://owasp.org/www-community/attacks/xss/
- https://owasp.org/www-community/xss-filter-evasion-cheatsheet
- https://en.wikipedia.org/wiki/Cross-site_scripting
- https://www.cgisecurity.com/xss-faq.html
- https://www.scriptalert1.com/

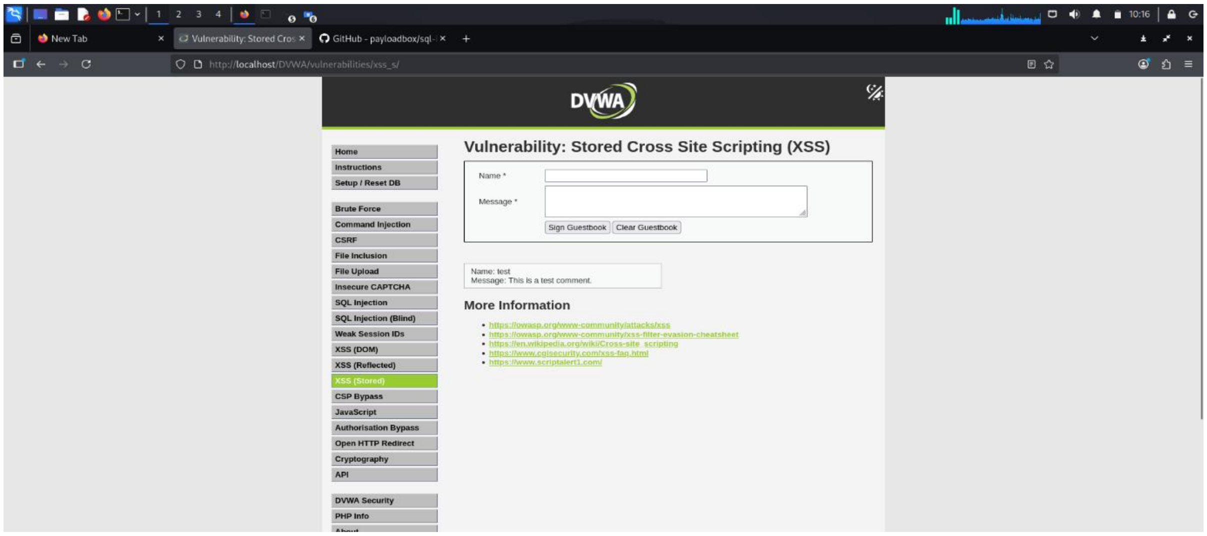


http://localhost/DVWA/vulnerabilities/xss_r/?name=<script>alert(document.cookie)<%2Fscript>#

localhost

security=low; PHPSESSID=o433ledfhiq5lij0jtmnevh2b3

c) Vulnerability: Stored XSS



Vulnerability: Stored Cross Site Scripting (XSS)

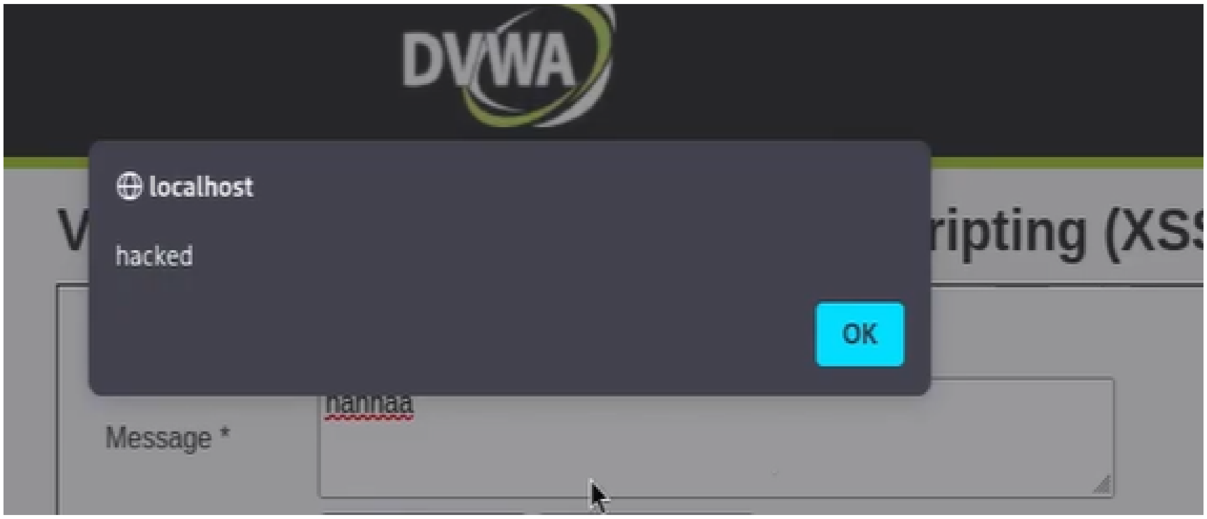
Name *

Message *

Name: test
Message: This is a test comment.

More Information

- https://owasp.org/www-community/attacks/xss
- https://owasp.org/www-community/xss-filter-evasion-cheatsheet
- https://en.wikipedia.org/wiki/Cross-site_scripting
- https://www.cgisecurity.com/xss-faq.html
- https://www.scriptalert1.com/



localhost

hacked

Message * nannaa

4. Consolidated Mitigation Strategies

Brute Force

- ✓ Apply account lockout after consecutive failed attempts.
- ✓ Use CAPTCHA to detect and prevent automated scripts.
- ✓ Establish strong password policies.

SQL Injection

- ✓ Use parameterized queries and prepared statements exclusively.
- ✓ Conduct thorough input validation and sanitization.
- ✓ Employ a Web Application Firewall (WAF) to detect injection attempts.

Cross-Site Scripting (XSS)

- ✓ Encode all user input before displaying it.

- ✓ Enforce strict Content Security Policy (CSP) rules.
- ✓ Apply strong validation and sanitization for all user inputs.

5. Conclusion

The assessment of DVWA demonstrated the presence of multiple high-severity vulnerabilities, many of which align with the **OWASP Top 10 security risks**. Although DVWA is intentionally designed to be insecure, the findings serve as a reminder of the dangers posed by poor coding practices and insufficient security controls.