Cyber Security Project: Bounty Hunt DVWA

Target Information

- **Application Name:** Damn Vulnerable Web Application (DVWA)
- **Deployment Type:** Localhost (Lab Environment)
- **Testing Type:** Authorized Lab Testing
- **Tester:** Raj Kulkarni
- **Date of Testing:** 17/01/2025

Scope

The scope of testing was limited strictly to the DVWA application hosted locally.
All testing was performed in a **controlled lab environment** for educational purposes only.

In-Scope Assets:

- Web application endpoints exposed by DVWA

Out-of-Scope:

- Any external systems
- Production or real-world targets

Tools Used

- Burp Suite Community Edition
- OWASP ZAP
- SQLMap
- Nmap
- Web Browser (Firefox)

ID	Vulnerability Type	Severity
V-01	Reflected Cross-Site Scripting (XSS)	Medium
V-02	SQL Injection (Authentication Bypass)	High

V-01: Reflected Cross-Site Scripting (XSS)

Severity: Medium

Affected Endpoint: /vulnerabilities/xss_r/

Impact

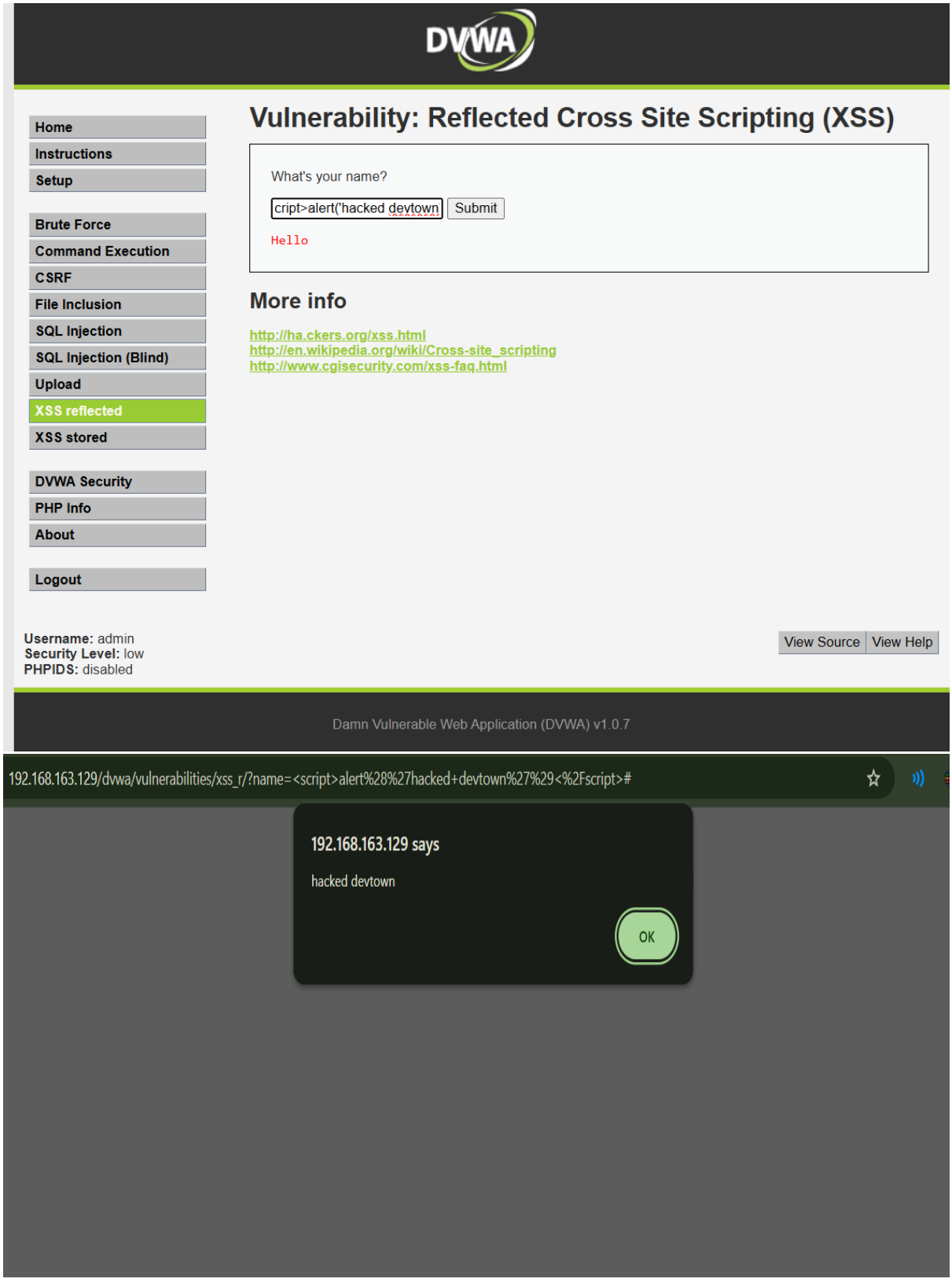
- Allows execution of arbitrary JavaScript
- Can lead to session hijacking
- Enables phishing attacks
- Could be chained with social engineering

Steps to Reproduce

1. Navigate to the **Reflected XSS** section in DVWA
2. Enter the following payload in the input field:

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

3. Submit the request
4. The JavaScript executes successfully in the browser


DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Vulnerability: Reflected Cross Site Scripting (XSS)
What's your name?
cript>alert('hacked devtown
Submit
Hello
More info
http://ha.ckers.org/xss.html
http://en.wikipedia.org/wiki/Cross-site_scripting
http://www.cgisecurity.com/xss-faq.html
Username: admin
Security Level: low
PHPIDS: disabled
View Source
View Help
Damn Vulnerable Web Application (DVWA) v1.0.7
192.168.163.129/dvwa/vulnerabilities/xss_r/?name=<script>alert%28%27hacked+devtown%27%29<%2Fscript>#
192.168.163.129 says
hacked devtown
OK

V-02: SQL Injection (Authentication Bypass)

Severity: High

Affected Endpoint: http://vulnerabilities/sqli/

Impact

- Complete database compromise
- Unauthorized data access
- Potential data modification or deletion
- Full application takeover

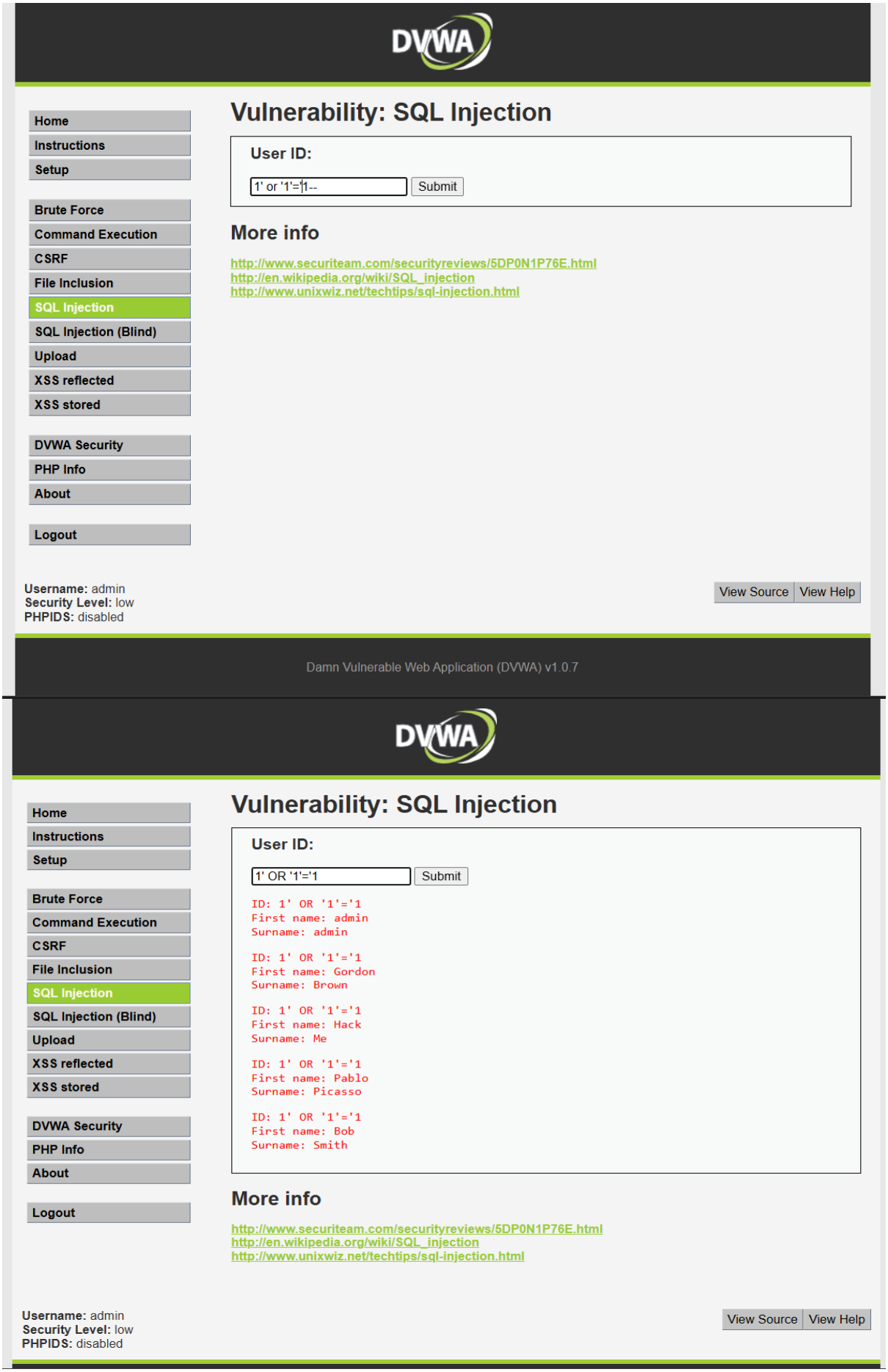
Steps to Reproduce

1. Navigate to the **SQL Injection** section
2. Enter the following payload in the User ID parameter:

1' OR '1'='1

3. Submit the request
4. Application returns user data without proper authorization

5.


DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Vulnerability: SQL Injection
User ID:
1' or '1'='1--
Submit
More info
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html
Username: admin
Security Level: low
PHPIDS: disabled
View Source
View Help
Damn Vulnerable Web Application (DVWA) v1.0.7
DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Vulnerability: SQL Injection
User ID:
1' OR '1'='1
Submit
ID: 1' OR '1'='1
First name: admin
Surname: admin
ID: 1' OR '1'='1
First name: Gordon
Surname: Brown
ID: 1' OR '1'='1
First name: Hack
Surname: Me
ID: 1' OR '1'='1
First name: Pablo
Surname: Picasso
ID: 1' OR '1'='1
First name: Bob
Surname: Smith
More info
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html
Username: admin
Security Level: low
PHPIDS: disabled
View Source
View Help

Overall Risk Assessment

The application contains **multiple critical vulnerabilities** that could allow an attacker to:

- Steal sensitive data
- Bypass authentication
- Execute arbitrary code in a victim's browser

If deployed in a real-world environment, these vulnerabilities could result in **severe security breaches**.

Conclusion

This project successfully demonstrates the **end-to-end bug bounty workflow**, including reconnaissance, vulnerability identification, exploitation, and professional reporting.
All findings were responsibly documented within the defined scope.