Web Application Vulnerability Assessment on DVWA

Objective

The objective of this lab was to test a vulnerable web application (DVWA) using industry-standard tools (Burp Suite, sqlmap, and OWASP ZAP) and manual techniques to identify OWASP Top 10 vulnerabilities, document results, and recommend remediation strategies.

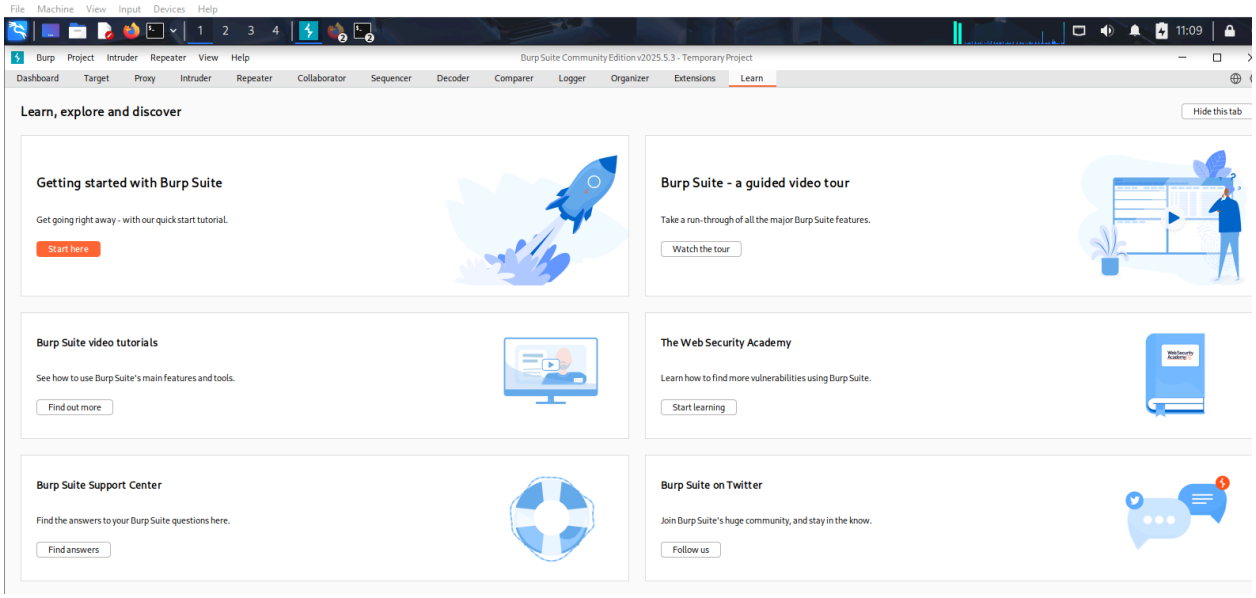
Test Setup

- **Target:** DVWA running on VM at http://192.168.1.200/
- **Tools Used:**
    - Burp Suite (interception, manual payloads)
    - sqlmap (automated SQL injection)
    - Manual scripting (XSS payloads)
- **Security Level:** Low (to allow vulnerability demonstration)

Test ID	Vulnerability	Severity	Target URL	Tool Used	Status
001	SQL Injection	Critical	http://192.168.1.200/login	Burp/sqlmap	Success
002	Reflected XSS	Medium	http://192.168.1.200/form	Manual	Success
003	Weak Authentication	High	http://192.168.1.200/login	Burp Suite	Success

Testing Process

I intercepted the login request using Burp Suite and manipulated the parameters with payloads such as:

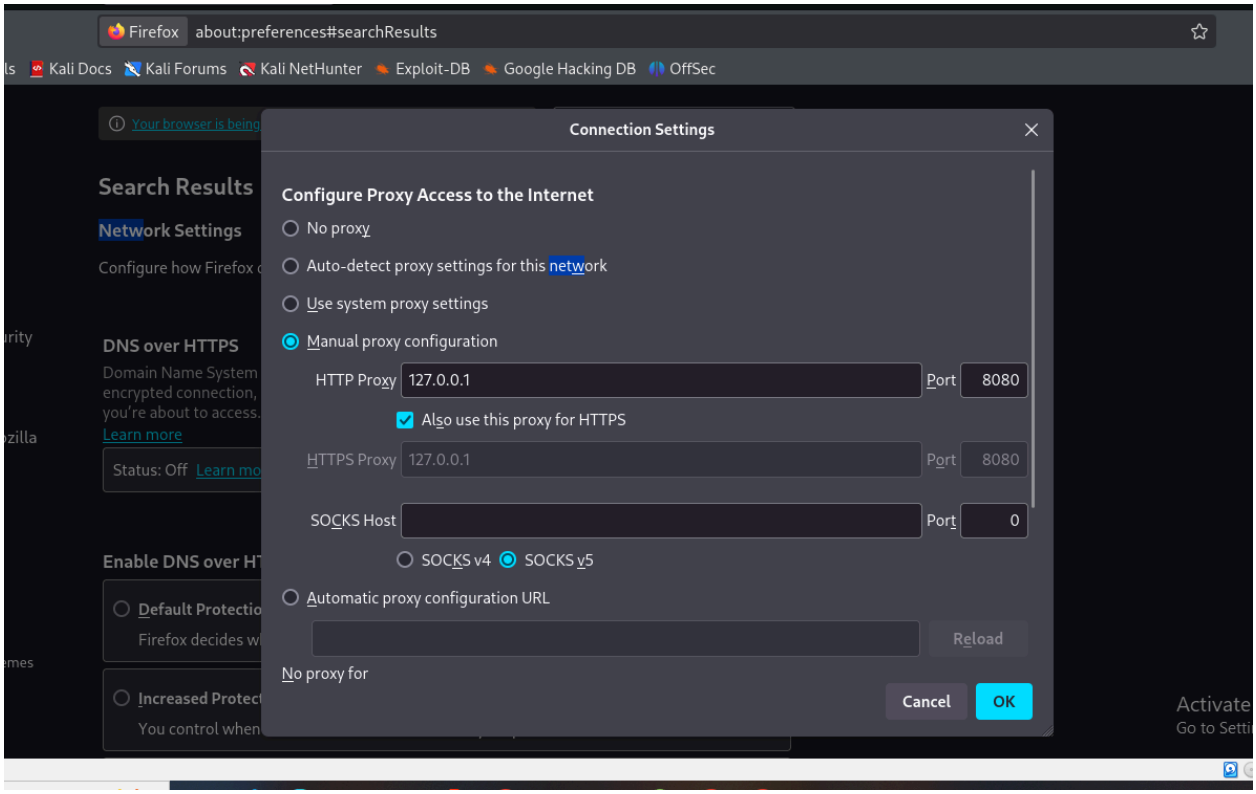


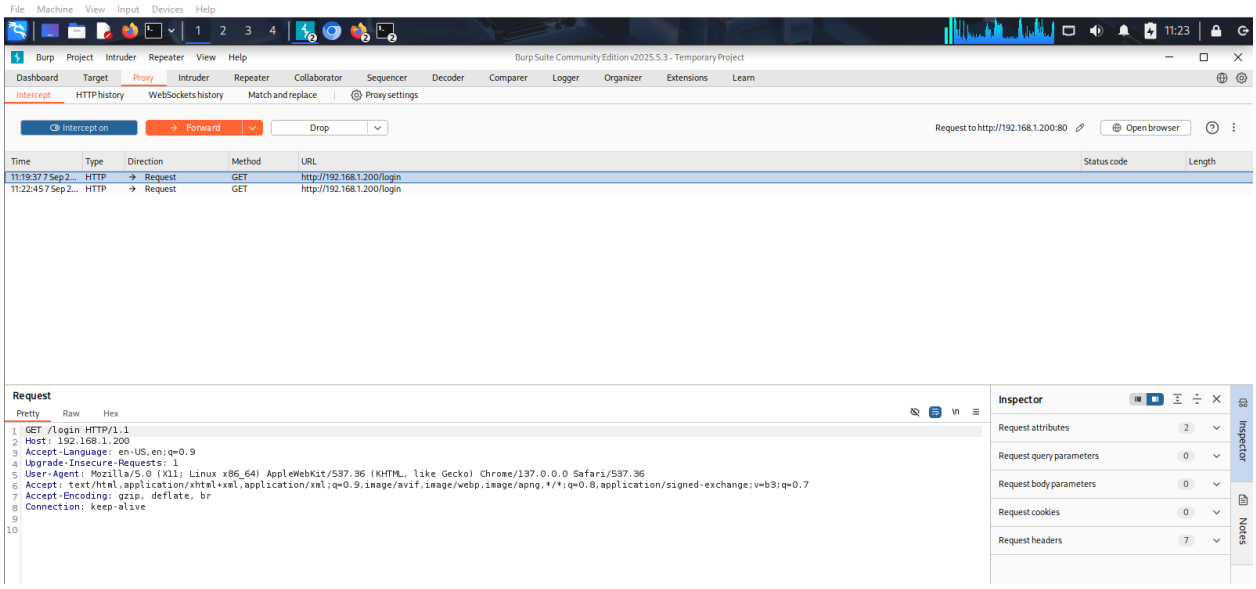
Cross-Site Scripting (XSS)

The payload executed in the browser, confirming Reflected XSS. Severity was rated as Medium, since it required user interaction but could still be used for session hijacking.

Authentication Testing

Using Burp Suite, I tested login credentials and session handling. I was able to log in using weak/default credentials (`admin:admin`). Additionally, session tokens lacked complexity, making brute force or replay attacks feasible. This confirmed a High severity Weak Authentication issue


Firefox
about:preferences#searchResults
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Search Results
Network Settings
DNS over HTTPS
Connection Settings
Configure Proxy Access to the Internet
No proxy
Auto-detect proxy settings for this network
Use system proxy settings
Manual proxy configuration
HTTP Proxy
127.0.0.1
Port
8080
Also use this proxy for HTTPS
HTTPS Proxy
127.0.0.1
Port
8080
SOCKS Host
Port
0
SOCKS v4
SOCKS v5
Automatic proxy configuration URL
Reload
No proxy for
Cancel
OK


Burp Suite Community Edition v2025.5.3 - Temporary Project
Intercept on
Forward
Drop
Request
Pretty
Raw
Hex
GET /login HTTP/1.1
Host: 192.168.1.200
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 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
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Inspector
Request attributes
Request query parameters
Request body parameters
Request cookies
Request headers

Summary

I conducted penetration testing on DVWA using Burp Suite, sqlmap, and manual payloads to assess OWASP Top 10 vulnerabilities. The tests revealed critical SQL injection, medium severity reflected XSS, and weak authentication issues. These findings highlight insecure coding practices and emphasize the importance of strong authentication, input sanitization, and regular vulnerability testing.

Remediation Recommendations

1. **SQL Injection:**

- ❖ Use prepared statements and parameterized queries.
- ❖ Implement strict server-side input validation.

2. **XSS:**

- ❖ Encode output before rendering in the browser.
- ❖ Sanitize user inputs at all form fields.

3. **Authentication:**

- ❖ Disable default accounts and enforce strong password policies.
- ❖ Implement secure session management with unpredictable tokens.