🔍 Introduction

This project explores web application security through practical exercises using vulnerable environments. The goal is to understand and exploit common web vulnerabilities using tools like **Burp Suite** and **OWASP ZAP**. The target application is **DVWA (Damn Vulnerable Web Application)**, and each exercise is mapped to the relevant stage in the **Cyber Kill Chain**.

🛠️ Tools Used

- **Kali Linux** (2024.1)
- **Burp Suite Community Edition**
- **OWASP ZAP**
- **Firefox**
- **DVWA (Damn Vulnerable Web Application)**

🎯 Target Application Description

Application: DVWA
URL: http://localhost/dvwa
Login: admin / password
Security Level: Low
Environment: LAMP stack on local VM

DVWA is designed to help security professionals test their skills in a safe environment. It includes vulnerable code across multiple OWASP Top 10 categories.

🔍 Phase 1: Reconnaissance & Mapping

Cyber Kill Chain Stage: Reconnaissance

✅Tools:

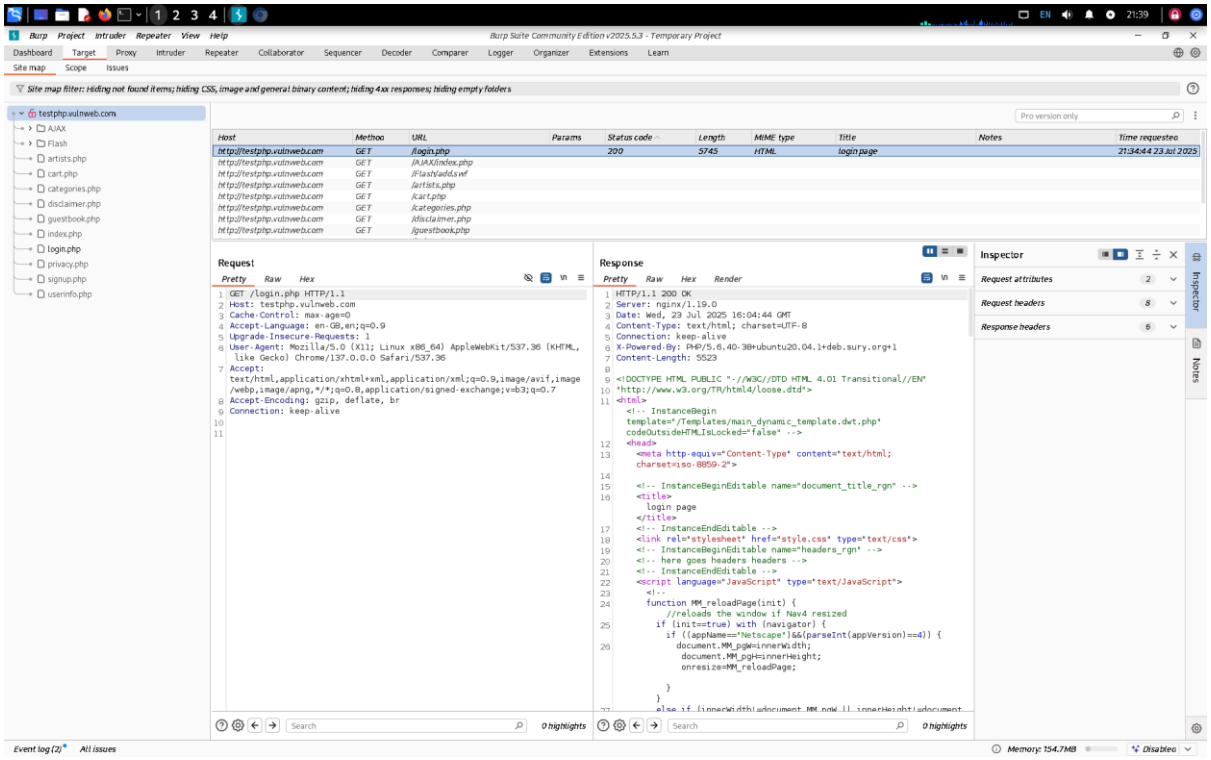
- **Burp Suite** (Target Tab, Proxy)
- **OWASP ZAP** (Spider)

🔍 Tasks Performed:

- Loaded DVWA in Firefox and intercepted traffic in Burp.
- Used **Burp Site Tree** and **ZAP Spider** to map all pages and forms.

🔍 Observations:

- Number of pages/endpoints discovered: *[e.g., 12]*
- Notable features:
    - Login form
    - SQL Injection page
    - XSS (Stored & Reflected)
    - Command Execution



🔍 Phase 2: Scanning & Vulnerability Discovery

Cyber Kill Chain Stages: Weaponization, Delivery

✅Tools:

- **OWASP ZAP Active Scan**
- **Burp Suite Repeater**

🧪 Tasks Performed:

- Performed **Active Scan** using ZAP
- Captured login request in Burp and tested injection payloads

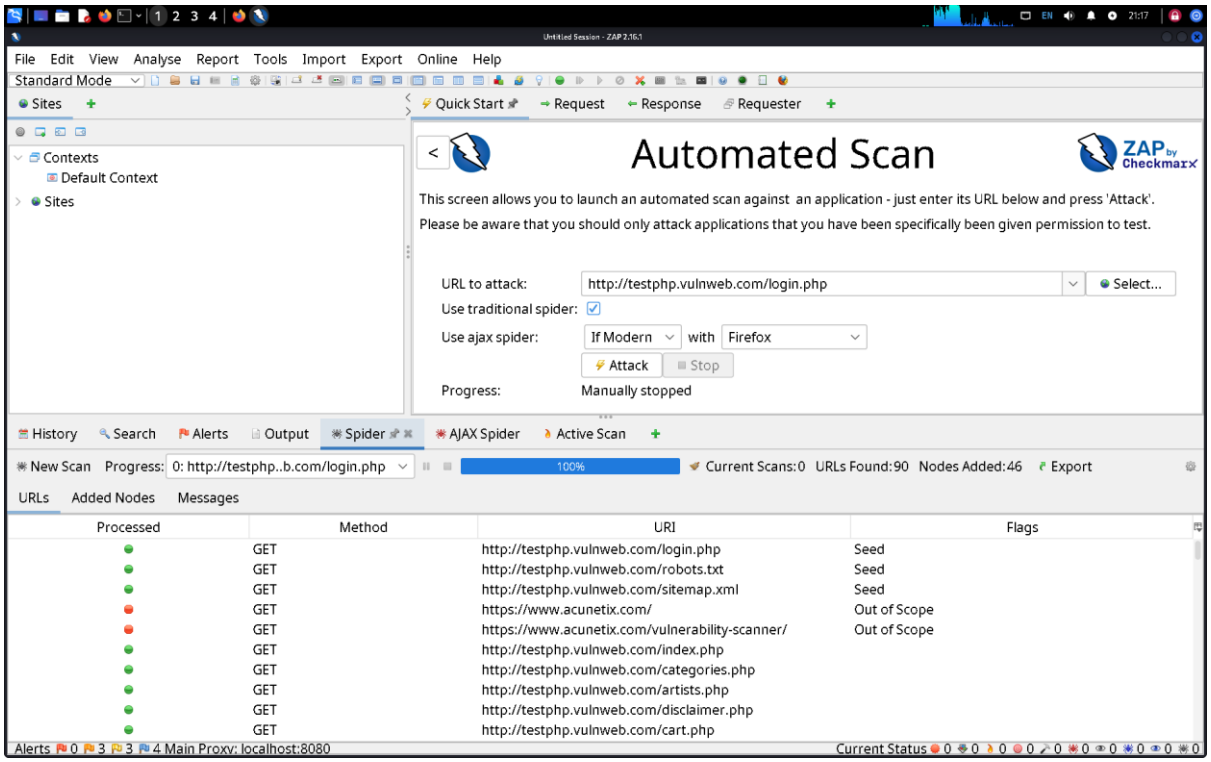
🧪 *Payloads Tried:*

sql

CopyEdit

admin' OR '1'='1

; ls



? Vulnerabilities Found:

Vulnerability	Description	OWASP Category
SQL Injection	Authentication bypass via payload	A1: Injection
Command Injection	System command execution	A1: Injection
Stored XSS	JS execution from comment form	A7: XSS

? Phase 3: Exploitation

Cyber Kill Chain Stage: Exploitation

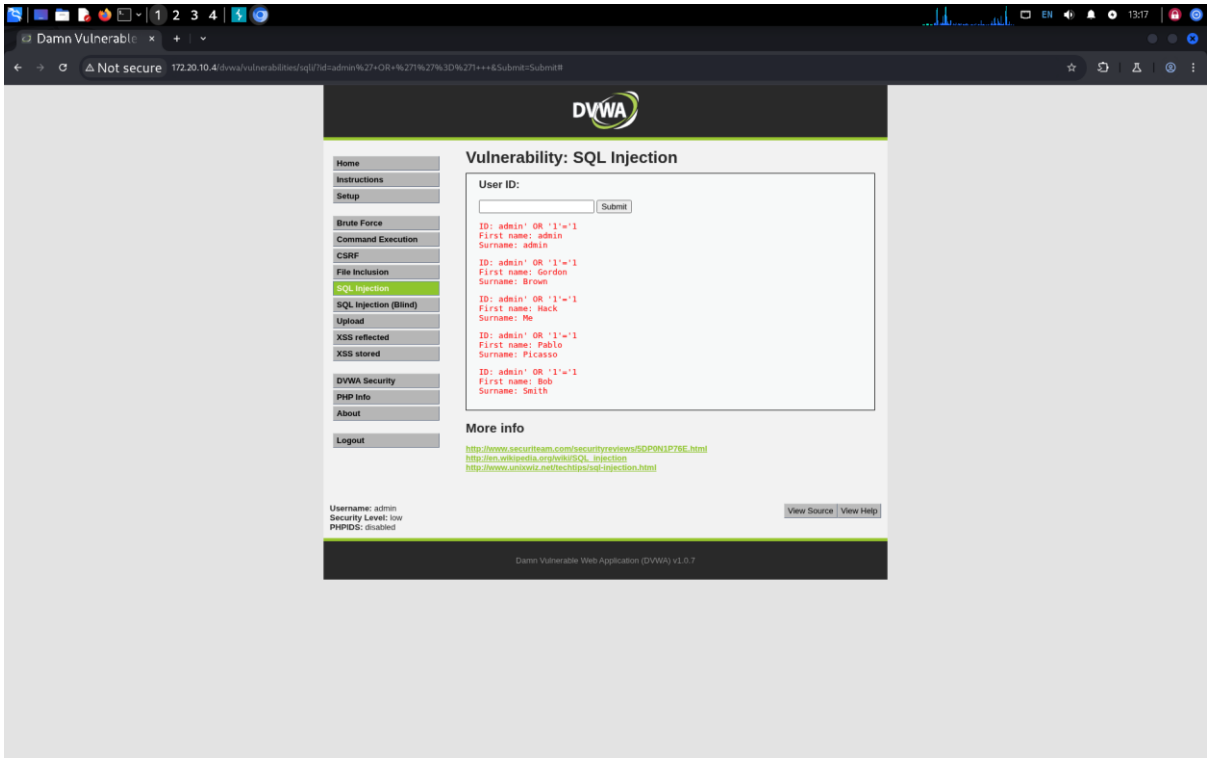
✅SQL Injection

- **Page:** Login
- **Payload:**

pgsql

CopyEdit
Username: admin' OR '1'='1
Password: anything

- **Result:** Logged in as admin without credentials.

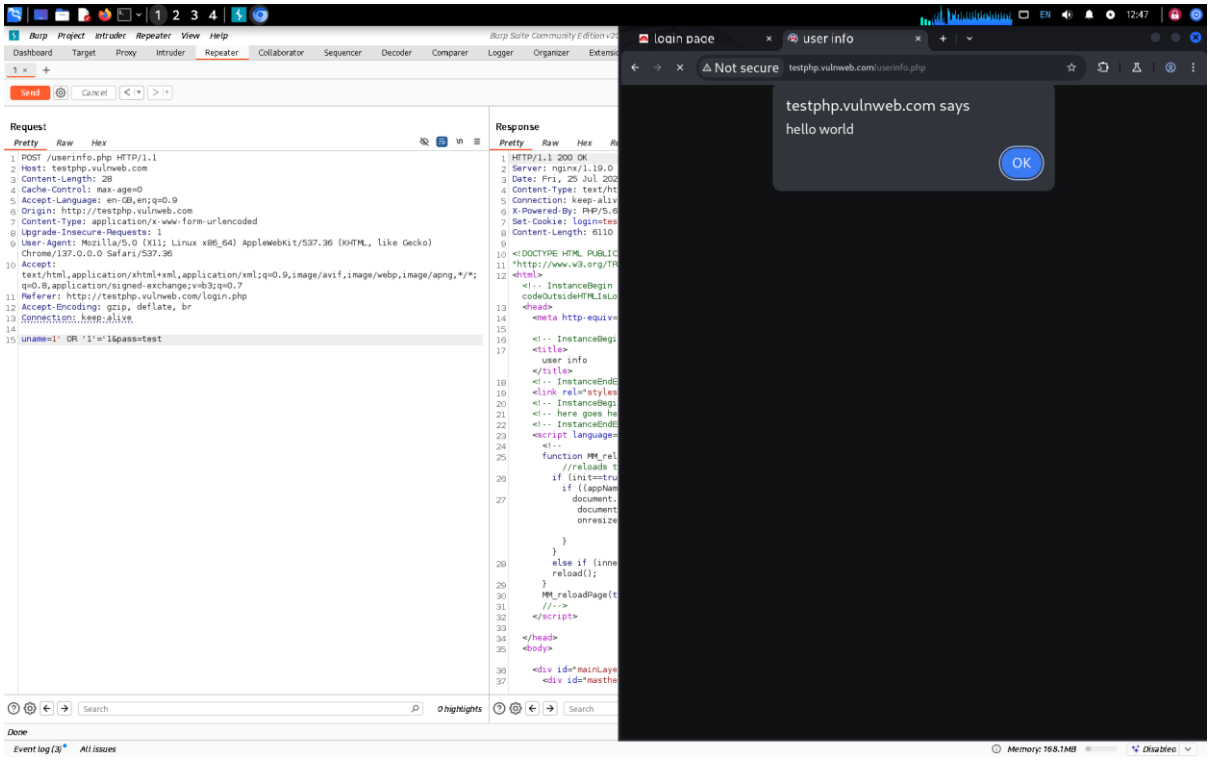


✅Stored XSS

- **Page:** Stored XSS
- **Payload:**

html
CopyEdit
<script>alert('XSS')</script>

- **Result:** JavaScript alert triggered when visiting the page.

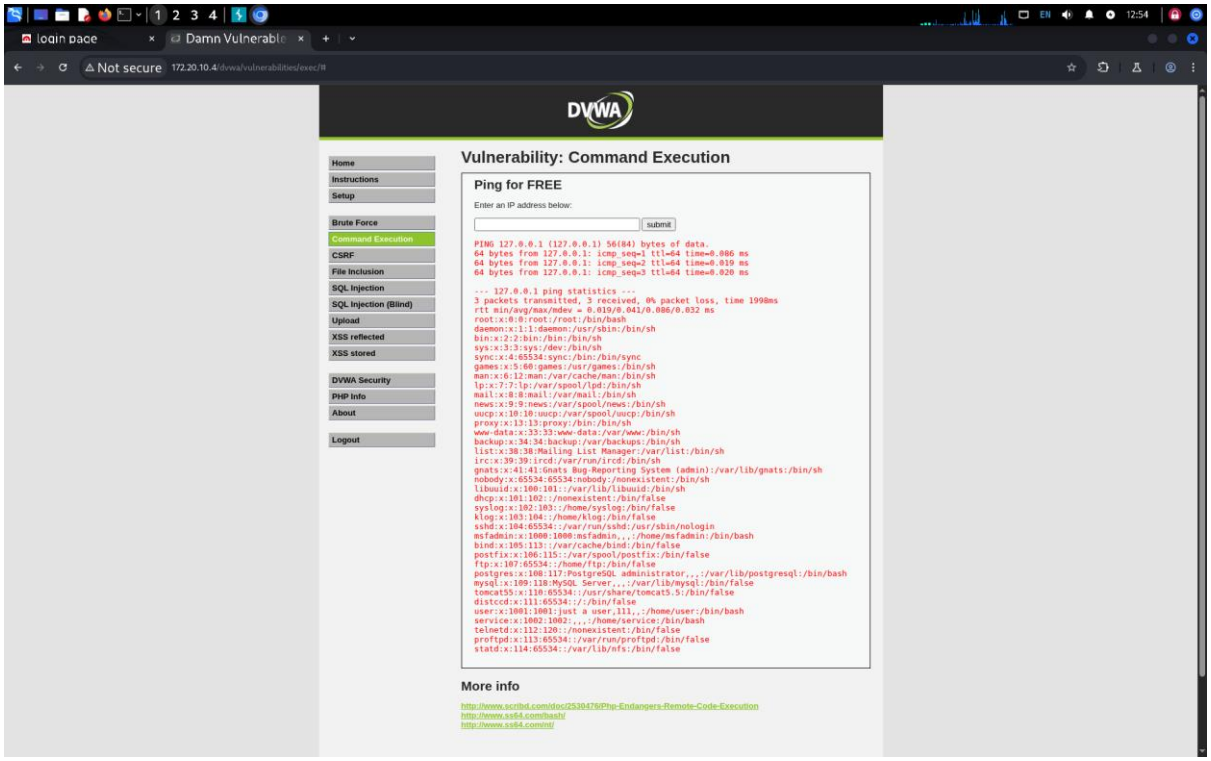


✅Command Injection

- **Page:** Command Execution
- **Payload:**

bash
CopyEdit
127.0.0.1 ; cat /etc/passwd

- **Result:** System file contents displayed.



? Cyber Kill Chain Mapping

Stage	Action Taken
Reconnaissance	Mapped app structure using ZAP and Burp
Weaponization	Prepared payloads for injection
Delivery	Sent malicious input via form fields
Exploitation	Gained unauthorized access, executed commands
Installation	(Not applicable in DVWA)
Command & Control	(Not applicable — local lab only)
Actions on Objective	Viewed sensitive data, proved exploitation success

? Recommendations

To secure applications against the vulnerabilities found:

- Use **parameterized queries** to prevent SQL injection
- Filter and encode all user input to mitigate XSS
- Avoid executing system commands from user input
- Set up **Web Application Firewalls (WAF)** for added protection
- Regularly scan web apps using tools like ZAP and Burp Suite

✅Conclusion

This project demonstrated practical web application exploitation using **Burp Suite**, **OWASP ZAP**, and **DVWA**. The hands-on tasks helped reinforce concepts from the **OWASP Top 10** and the **Cyber Kill Chain**, enhancing both offensive and defensive understanding of web security.