SECURING A VULNERABLE WEB APPLICATION

OWASP Juice Shop Security Testing Project

1. Abstract

Web applications are frequently targeted by attackers due to improper security controls, insecure coding practices, and misconfigurations. This project focuses on identifying and understanding common web application vulnerabilities using **OWASP Juice Shop**, an intentionally vulnerable web application developed for security training purposes.

The project involves setting up a controlled lab environment using **VirtualBox**, **Ubuntu Server**, **Docker**, and **Kali Linux**, followed by identifying vulnerabilities, exploiting them in a safe environment, and understanding mitigation techniques. This study enhances practical knowledge of web application security and aligns with OWASP Top 10 vulnerability categories.

2. Introduction

Web application security is a critical and rapidly evolving domain within cybersecurity, as modern organizations increasingly rely on web-based systems to deliver services, store sensitive data, and support business operations. With the widespread adoption of cloud platforms, APIs, and web applications, attackers now have a larger attack surface to exploit. As a result, a significant number of real-world data breaches and cyber incidents occur due to insecure web application design, improper implementation, and insufficient security testing.

Common vulnerabilities such as **SQL Injection**, **Authentication and Session Management Flaws**, **Broken Access Control**, **Cross-Site Scripting (XSS)**, and **Insecure Direct Object References (IDOR)** continue to be among the leading causes of security compromise. These weaknesses allow attackers to bypass authentication mechanisms, manipulate backend databases, escalate privileges, or gain unauthorized access to sensitive information, leading to financial loss, reputational damage, and legal consequences for organizations.

OWASP Juice Shop is a deliberately insecure, modern web application developed by the Open Web Application Security Project (OWASP) to help learners, developers, and security professionals understand and identify these vulnerabilities in a safe and controlled environment. It incorporates real-world security flaws aligned with the OWASP Top 10, making it an effective platform for hands-on penetration testing practice and security awareness training.

This project aims to simulate a real-world web application penetration testing scenario within an isolated laboratory setup, following ethical hacking principles and legal guidelines. By identifying, exploiting, and analyzing vulnerabilities in OWASP Juice Shop, the project provides practical exposure to web application attack methodologies, vulnerability assessment techniques, and risk understanding. The overall objective is to strengthen foundational knowledge in web application security, promote secure development practices, and enhance defensive and offensive cybersecurity skills strictly for educational and ethical learning purposes only.

3. Objectives

- To understand common web application vulnerabilities
- To deploy a vulnerable application securely using Docker
- To perform ethical security testing using Kali Linux tools
- To analyze vulnerabilities mapped to OWASP Top 10
- To document findings and recommend security mitigations

4. Scope of the Project

- The testing is conducted **only in a local lab environment**
- No real systems or external networks are targeted
- Focus is on **learning, analysis, and documentation**
- Exploitation is limited to demonstration and validation only

5. Tools & Technologies Used

Category	Tools
Virtualization	Oracle VirtualBox
Target OS	Ubuntu Server 24.04
Attacker OS	Kali Linux 2025
Containerization	Docker
Vulnerable App	OWASP Juice Shop
Testing Tools	Browser DevTools, Kali Linux utilities

6. System Architecture

- **Host Machine**: Windows OS
- **Virtual Machines**:
    - Ubuntu Server (Target)
    - Kali Linux (Attacker)
- **Networking**:
    - Host-Only Adapter for inter-VM communication
- **Application Deployment**:
    - OWASP Juice Shop running inside Docker on Ubuntu

7. Lab Setup & Configuration

7.1 Virtual Machine Setup

- Installed Oracle VirtualBox
- Created Ubuntu Server VM with:
    - 2 GB RAM
    - 1 CPU

    - Host-Only + NAT network
- Created Kali Linux VM with Host-Only network

7.2 Docker & Juice Shop Deployment

Commands used on Ubuntu:

- sudo apt update
- sudo apt install docker.io -y
- sudo docker pull bkimminich/juice-shop
- sudo docker run -d --name juice -p 3000:3000 bkimminich/juice-shop

Juice Shop accessed via browser:

- http://<Ubuntu-IP>:3000

8. Vulnerability Analysis & Findings

8.1 Broken Authentication

- Admin login bypass identified
- Weak authentication logic exploited
- **Impact**: Unauthorized access to admin account

8.2 Broken Access Control

- Restricted pages accessed without proper authorization
- **Impact**: Privilege escalation

8.3 Injection Vulnerabilities

- Input fields susceptible to injection attacks
- **Impact**: Data manipulation risk

8.4 Security Misconfiguration

- Debug messages and unnecessary services exposed
- **Impact**: Information disclosure

9. OWASP Top 10 Mapping

Vulnerability	OWASP Category
Admin Login Bypass	A2 – Broken Authentication
Unauthorized Access	A5 – Broken Access Control
Injection Inputs	A3 – Injection
Misconfiguration	A6 – Security Misconfiguration

10. Mitigation & Security Recommendations

- Implement strong authentication mechanisms
- Use role-based access control (RBAC)
- Validate and sanitize all user inputs
- Disable unnecessary services and debug logs
- Apply security headers and HTTPS
- Conduct regular security audits

11. Results & Observations

- Successfully deployed OWASP Juice Shop in a controlled lab
- Identified multiple web application vulnerabilities
- Understood attacker techniques and defensive strategies
- Improved practical skills in ethical hacking and security analysis

12. Conclusion

This project successfully demonstrated real-world web application vulnerabilities using OWASP Juice Shop. It provided hands-on exposure to penetration testing concepts and emphasized the importance of secure coding practices. The experience strengthens foundational knowledge required for cybersecurity roles such as penetration tester and security analyst.

13. Ethical Disclaimer

This project was conducted **strictly for educational purposes** within a controlled environment.

No real systems, networks, or data were harmed.
All testing adhered to ethical cybersecurity practices.

14. References

- OWASP Foundation – https://owasp.org
- OWASP Juice Shop – https://owasp.org/www-project-juice-shop/
- Docker Documentation – https://docs.docker.com