Web Application Security Assessment Report

Internship: Future Interns – Cyber Security Track (Task 1)

Tested Application: OWASP Juice Shop

Testing Type: Vulnerability Assessment & Penetration Testing (VAPT)

1. Introduction

This report documents the results of a web application security assessment conducted as part of the Future Interns Cyber Security Internship. The objective of this task was to identify common web vulnerabilities, analyze their impact, and recommend appropriate mitigation strategies using industry-standard tools and methodologies aligned with the OWASP Top 10.

2. Scope of Assessment

The assessment was limited to a controlled testing environment using the intentionally vulnerable OWASP Juice Shop application hosted locally. No external or production systems were tested.

Target URL: http://localhost:3000
Environment: Localhost (Docker-based deployment)

3. Tools & Technologies Used

- OWASP Juice Shop (Vulnerable Web Application)
- OWASP ZAP (Automated & Passive Vulnerability Scanner)
- Docker (Used to pull and run the OWASP Juice Shop image)
- Web Browser (Google Chrome)
- GitHub (Documentation & Version Control)

Docker Note: The OWASP Juice Shop application was deployed using the official Docker image (*bkimminich/juice-shop*). Docker was used to ensure a clean, isolated, and reproducible testing environment, closely simulating real-world DevSecOps deployment practices.

4. Testing Methodology

The assessment followed a structured security testing methodology consisting of reconnaissance, passive analysis, automated scanning, and manual validation. The approach was aligned with OWASP Top 10 security testing principles.

• Passive Scanning using OWASP ZAP while browsing the application
• Spidering to discover hidden endpoints and application structure
• Active Scanning to identify injection flaws and security misconfigurations
• Manual testing of input fields such as search and login forms

5. Vulnerability Findings

5.1 Cross-Site Scripting (XSS)

Description: Cross-Site Scripting vulnerabilities occur when user-supplied input is not properly validated or sanitized, allowing attackers to inject malicious scripts into web pages viewed by other users.

Testing Method: Manual payload testing in search input fields and automated detection using OWASP ZAP.

Impact: Attackers may execute malicious scripts in a victim's browser, leading to session hijacking, data theft, or defacement.

OWASP Mapping: A03:2021 – Injection
Risk Level: High

5.2 SQL Injection (Potential)

Description: SQL Injection vulnerabilities arise when application queries are constructed using unsanitized user input, allowing attackers to manipulate backend database queries.

Testing Method: Automated scanning using OWASP ZAP. Direct manual exploitation attempts did not result in immediate compromise due to the use of parameterized queries.

Impact: Successful exploitation may allow attackers to bypass authentication or extract sensitive data.

OWASP Mapping: A03:2021 – Injection
Risk Level: Medium

5.3 Security Misconfiguration

Description: Security misconfigurations include missing security headers, verbose error messages, and unnecessary exposed endpoints.

Testing Method: Identified through passive and active scanning using OWASP ZAP.

Impact: Can assist attackers in reconnaissance and exploitation of other vulnerabilities.

OWASP Mapping: A05:2021 – Security Misconfiguration
Risk Level: Medium

6. Recommendations

- Implement strict server-side input validation and output encoding
- Use Content Security Policy (CSP) to mitigate XSS risks
- Ensure parameterized queries are enforced across all database interactions
- Apply security headers and follow secure configuration best practices

7. Conclusion

This assessment demonstrated common web application vulnerabilities and the importance of secure coding practices. The combination of automated scanning and manual testing provided valuable insight into application security posture. This task enhanced practical knowledge of ethical hacking, vulnerability assessment, and professional security reporting.

Disclaimer: This assessment was performed in a controlled lab environment for educational purposes only. No real-world systems were harmed or tested without authorization.