FUTURE INTERN

Vulnerability Assessment Report Task 1

- **Target Application:** OWASP Juice Shop (Demo)
- **Tools Used:** Nmap, OWASP ZAP, Browser Dev Tools, Canva (report design)

Prepared By:
Yasaswini Vanukuri

Executive Summary

This report presents the results of a passive vulnerability assessment conducted on the OWASP Juice Shop demo web application. The objective of this assessment was to identify potential security weaknesses and misconfigurations using industry-standard tools.

The assessment identified a total of 27 security findings, including 7 medium-risk issues, 11 low-risk issues, and 9 informational alerts. No critical (high-risk) vulnerabilities were discovered during the scan.

Most of the findings are related to missing security headers, cookie security misconfigurations, and general security hardening practices. While these issues do not represent immediate critical threats, they increase the application's exposure to attacks such as Cross-Site Scripting (XSS), session hijacking, and clickjacking.
Overall Risk Rating: Medium

Scope of Assessment

Target Application: OWASP Juice Shop (Demo Version)
Target URL: https://demo.owasp-juice.shop/#/

Assessment Scope Includes:

- Network port scanning
- Passive web application vulnerability scanning
- Manual browsing of application features

Out of Scope::

- Active exploitation
- Denial of Service attacks
- Database intrusion attempts

This assessment was conducted strictly for educational purposes on a publicly available demo application.

Methodology

The vulnerability assessment was performed in the following phases:

Phase 1: Network Reconnaissance

Nmap was used to identify open ports and exposed services on the target server.

Phase 2: Passive Web Scanning

OWASP ZAP was configured as a local proxy and used to perform passive scanning of HTTP responses.

Phase 3: Manual Exploration

The application was manually explored by:

- Creating an account
- Logging into the system
- Browsing products
- Adding items to the cart

Phase 4: Risk Classification

Identified vulnerabilities were categorized based on severity levels (High, Medium, Low, Informational).

Nmap Scan Results

The Nmap scan identified the following open ports:

- **Port 21** – FTP
- **Port 80** – HTTP
- **Port 443** – HTTPS
- **Port 8080** – HTTP Proxy

Risk Analysis:

The presence of additional open services increases the attack surface of the application. Services such as FTP and HTTP proxy can potentially be misused if not properly secured.

Risk Level: **Medium**

Vulnerability Findings

Alert 1: Content Security Policy (CSP) Header Not Set

Risk Level: Medium

Description:

The application does not define a Content Security Policy header, which helps prevent Cross-Site Scripting (XSS) and code injection attacks.

Business Impact:

Attackers may inject malicious scripts, leading to session hijacking or credential theft.

Recommendation:

Implement a strict Content-Security-Policy HTTP header.

Alert 2: Cookie Without HttpOnly Flag

Risk Level: Medium

Description:

Session cookies are set without the HttpOnly attribute.

Business Impact:

Cookies may be accessed through client-side scripts if an XSS vulnerability exists.

Recommendation:

Enable HttpOnly and Secure flags for all session cookies.

Alert 3: Missing Anti-Clickjacking Header

Risk Level: Medium

Description:

The application does not use X-Frame-Options or equivalent protection.

Business Impact:

The application may be vulnerable to clickjacking attacks.

Recommendation:

Add X-Frame-Options header or use frame-ancestors directive in CSP.

Alert 4: Strict-Transport-Security Header Not Set

Risk Level: Low

Description:

The HSTS header is not enabled.

Business Impact:

Users may be vulnerable to HTTPS downgrade attacks.

Recommendation:

Enable HTTP Strict Transport Security (HSTS).

Risk Classification Summary

Severity Level	Number of Issues
High	0
Medium	7
Low	11
Informational	9

Total Issues Identified: 27

Overall Risk Rating: Medium

Recommendations

To improve the security posture of the application, the following actions are recommended:

- Implement missing security headers (CSP, HSTS, X-Frame-Options).
- Enable HttpOnly, Secure, and SameSite cookie attributes.
- Close or secure unnecessary open ports.
- Perform regular vulnerability scanning.
- Keep server software and dependencies updated.
- Conduct periodic security audits.

Conclusion

The OWASP Juice Shop demo application demonstrates a moderate security posture. Although no critical vulnerabilities were detected during passive scanning, several medium-risk issues related to security misconfiguration and session management were identified.

Addressing these findings will significantly improve the application’s resilience against common web-based attacks.

This assessment highlights the importance of implementing secure configuration practices and continuous monitoring to maintain application security.