Web Application Security Testing – Future Interns Task 1

Name: Aman Mali
Date: 12 November 2025
Tool Used: OWASP ZAP 2.16.1
Target URL: https://juice-shop.herokuapp.com

---

1. Overview

The purpose of this cybersecurity task was to perform a vulnerability assessment of the OWASP Juice Shop web application using OWASP ZAP.
The goal was to identify security flaws following OWASP Top 10 standards and document the results in a professional report.

---

2. Tools & Methodology

Tools Used:

- OWASP ZAP (Automated Scan – Quick Start)
- Google Chrome (for viewing pages)

Testing Steps:

1. The Juice Shop web application was selected as the test target.
2. The Automated Scan option in OWASP ZAP was used with the target URL.
3. Alerts were recorded and analyzed after the scan completed.
4. Screenshots were captured for documentation.

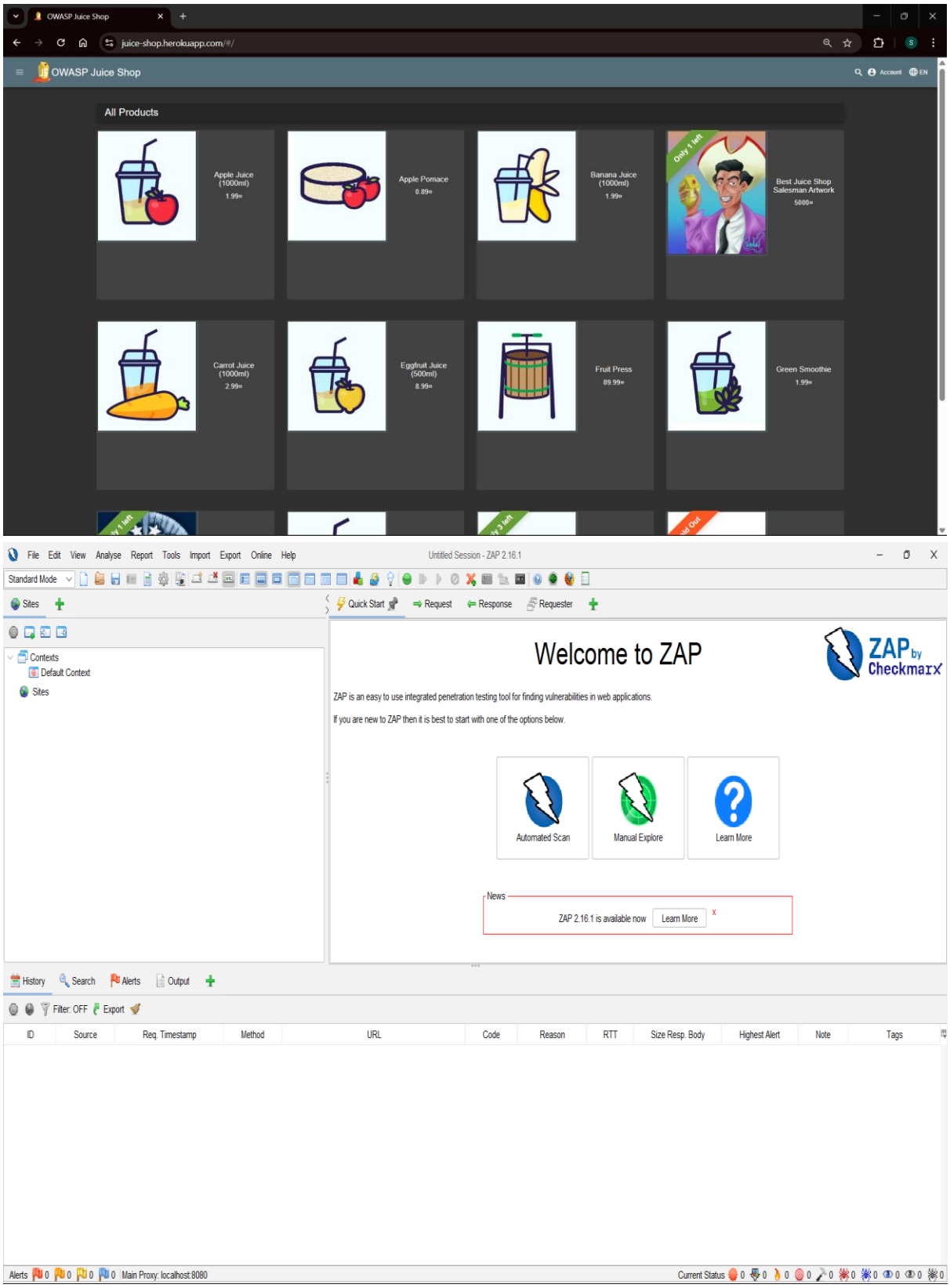
Scan Duration: ~2 minutes

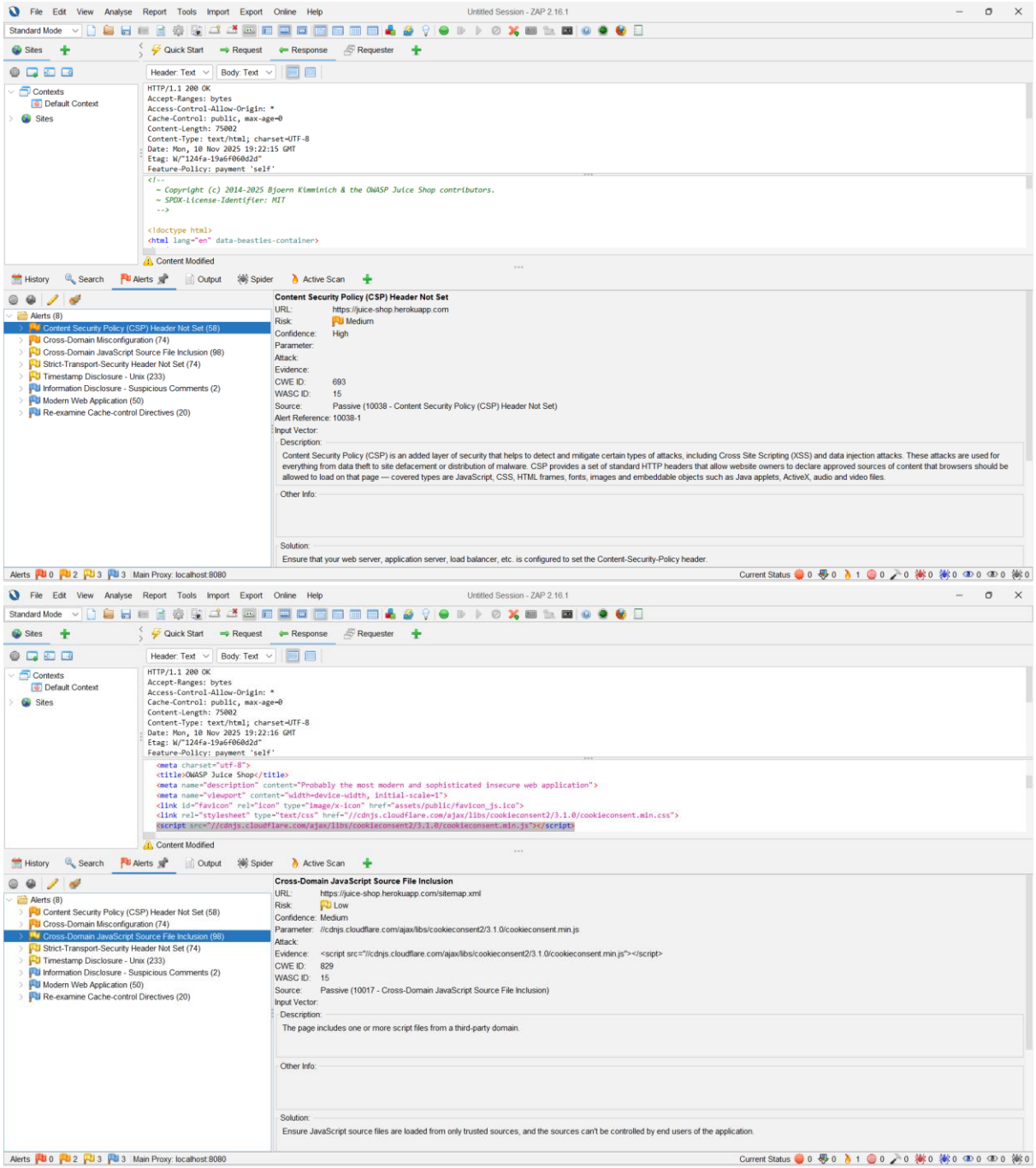
---

3. Findings Summary

No	Vulnerability	OWASP Category	Risk	Description / Evidence	Recommended Fix
1	Content Security Policy Header Not Set	A06: Security Misconfiguration	High	Missing CSP header may allow cross-site scripting	Add "Content-Security-Policy" header
2	Strict-Transport-Security Header Not Set	A06: Security Misconfiguration	High	Site doesn’t enforce HTTPS	Implement HSTS header
3	Cross-Domain Misconfiguration	A01: Broken Access Control	Medium	Allows access from other domains	Restrict CORS to trusted origins
4	Cross-Domain JavaScript Source File Inclusion	A06: Security Misconfiguration	Medium	JS loaded from external CDN without integrity check	Use Subresource Integrity or trusted sources
5	Timestamp Disclosure (Unix)	A05: Security Misconfiguration	Low	Reveals server timestamps	Remove or mask timestamp info
6	Information Disclosure – Suspicious Comments	A02: Cryptographic Failures	Low	Developer comments visible in code	Remove debug comments before deployment
7	Modern Web Application	—	Info	General information about web framework	No action needed
8	Cache-Control Directives – Re-Examine	A06: Security Misconfiguration	Low	Missing secure cache headers	Add "Cache-Control: no-store, no-cache"

4. Evidence Screenshots





5. Conclusion

The OWASP ZAP scan identified multiple configuration-related vulnerabilities within the Juice Shop web application.
Most are low-to-medium risk, but implementing the recommended fixes (especially around HTTP security headers and CORS policies) will significantly enhance the site's security posture and compliance with OWASP Top 10 best practices.

6. Appendix

Supporting Files:

Web Application Security Scan Report Summary

Target Site: https://juice-shop.herokuapp.com

Scanning Tool Utilized: OWASP ZAP

Date of Scan: 2025-11-12

Tool Status: OWASP ZAP was successfully initialized and operating.

Scan Execution: The target, https://juice-shop.herokuapp.com, was subjected to a security scan.

Scan Results: The automated scan concluded successfully.

Alerts Identified: 8 security alerts were discovered during the scan.

---

7. Detailed Vulnerability Analysis

7.1. Content Security Policy Header Not Set (High Risk)

This vulnerability is a critical security hardening deficiency. The lack of a Content Security Policy (CSP) header means the browser has no restrictions on where content (scripts, styles, images) can be loaded from, or how it can be executed.

Impact: Highly susceptible to Cross-Site Scripting (XSS) attacks, which can lead to session hijacking, data theft, and unauthorized actions.

Recommendation: Implement a robust Content-Security-Policy header on all server responses. The policy should be carefully defined to only allow scripts and resources from trusted domains (self or known CDNs).

7.2. Strict-Transport-Security Header Not Set (High Risk)

The absence of the HTTP Strict Transport Security (HSTS) header means that users who initially navigate to the site via unencrypted HTTP (even unintentionally or via a bookmark) are vulnerable to Man-in-the-Middle (MITM) attacks before being redirected to HTTPS.

Impact: Users are exposed to SSL stripping and session hijacking if they access the site over an insecure network.

Recommendation: Add the Strict-Transport-Security header with a sufficient max-age directive (e.g., one year) and include the includeSubDomains directive if applicable.

7.3. Cross-Domain Misconfiguration (Medium Risk)

The application appears to allow cross-origin requests (CORS) from potentially unauthorized or wildcard domains. While necessary for some functionality, overly permissive CORS settings can allow an attacker's website to read sensitive data from the Juice Shop domain if the victim is authenticated.

Impact: Potential for sensitive data leakage (e.g., user tokens, private information) to external malicious websites.

Recommendation: Restrict the Access-Control-Allow-Origin header to only explicitly trusted domains required for legitimate cross-origin communication. Avoid using the wildcard (*) unless absolutely necessary and secure.

7.4. Cross-Domain JavaScript Source File Inclusion (Medium Risk)

This finding indicates that external JavaScript files are being loaded, likely from a Content Delivery Network (CDN), without the use of Subresource Integrity (SRI) checks. SRI is a security feature that allows browsers to verify that resources they fetch have not been manipulated.

Impact: If the external CDN is compromised, an attacker could inject malicious code into the JavaScript file, which would then be executed by the Juice Shop user's browser, leading to a supply-chain attack (e.g., XSS).

Recommendation: Implement Subresource Integrity (SRI) checks for all externally loaded scripts and stylesheets. This involves adding an integrity attribute with a cryptographic hash to the script tag.

7.5. Timestamp Disclosure (Unix) (Low Risk)

The server response or page content exposes Unix timestamps. While not a direct exploit vector, this information can be used by attackers to aid in fingerprinting the server, guessing internal processes, or correlating events, increasing the effectiveness of a targeted attack.

Impact: Minor information leakage that assists in reconnaissance and advanced targeted attacks.

Recommendation: Configure the web server or application framework to mask or entirely remove detailed server-side timestamps from publicly visible headers or error messages.

7.6. Information Disclosure – Suspicious Comments (Low Risk)

The scan detected comments within the HTML or JavaScript code that appear to be debug notes, to-do lists, or sensitive remarks intended only for developers.

Impact: This type of disclosure can reveal internal system architecture, unreleased features, or potential logic flaws, providing valuable reconnaissance data to an attacker.

Recommendation: Ensure all application code is stripped of developer comments, debug statements, and unnecessary internal documentation before being deployed to a production environment.

7.7. Cache-Control Directives – Re-Examine (Low Risk)

The application is missing or incorrectly configured secure caching directives, potentially leading to sensitive data being cached by the browser or intermediary proxies. Specifically, the recommended headers like Cache-Control: no-store, no-cache are not consistently applied where needed.

Impact: Sensitive user data (e.g., shopping cart contents, tokens after logout) could be retrieved from the browser's cache, especially on shared computers.

Recommendation: Implement strict Cache-Control: no-store, no-cache, must-revalidate headers on all pages containing sensitive or dynamic user data to prevent caching.

---

8. OWASP Top 10 Compliance Checklist

No	Vulnerability	OWASP Category	Compliance Status	Risk Level
1	Content Security Policy Header Not Set	A06: Security Misconfiguration	✕ Non-Compliant	High
2	Strict-Transport-Security Header Not Set	A06: Security Misconfiguration	✕ Non-Compliant	High
3	Cross-Domain Misconfiguration	A01: Broken Access Control	✕ Non-Compliant	Medium
4	Cross-Domain JavaScript Source File Inclusion	A06: Security Misconfiguration	✕ Non-Compliant	Medium
5	Timestamp Disclosure (Unix)	A05: Security Misconfiguration	✕ Non-Compliant	Low
6	Information Disclosure – Suspicious Comments	A02: Cryptographic Failures	✕ Non-Compliant	Low
7	Modern Web Application (Info)	—	☑ Informational Only	Info
8	Cache-Control Directives – Re-Examine	A06: Security Misconfiguration	✕ Non-Compliant	Low