## CYBER SECURITY TASK 01

# WEB APPLICATION SECURITY ASSESSMENT REPORT

## Target Application: OWASP Juice Shop

https://juice-shop.herokuapp.com

Assessment Type: **Passive Security Assessment**

Prepared by: **Razi Ammari**

Environment: **Kali Linux**

January 2026

# Confidentiality Statement

This document contains confidential security assessment information intended solely for evaluation and educational purposes. Redistribution or reuse of this report without explicit authorization is prohibited.

# Disclaimer

This security assessment represents a point-in-time analysis based on passive techniques only.

No exploitation, credential attacks, or service disruption techniques were used.

The findings reflect the application’s security posture at the time of testing and do not guarantee the absence of vulnerabilities outside the defined scope.

# Table of Contents

# 1. Executive Summary

This report presents the results of a passive web application security assessment conducted against the OWASP Juice Shop application. The objective was to identify exposed services, configuration weaknesses, and missing security controls without actively exploiting vulnerabilities.

The assessment identified several medium and low-risk security misconfigurations, primarily related to HTTP security headers, transport security enforcement, and information disclosure. While no critical vulnerabilities were exploited, these findings indicate areas that could be leveraged by an attacker in a real-world scenario if left unmitigated.

Overall, the application demonstrates functional security controls but lacks defense-in-depth hardening mechanisms expected in production-grade environments.

# 2. Scope of Assessment

In Scope

- Target domain: https://juice-shop.herokuapp.com
- Passive network reconnaissance
- HTTP response header analysis
- Client-side traffic inspection
- Passive vulnerability detection via OWASP ZAP

Out of Scope

- Active exploitation
- Authentication bypass
- Credential attacks
- Denial-of-Service testing
- Data manipulation or injection attacks

# 3. Methodology

The assessment followed a passive security testing methodology aligned with :

- OWASP Testing Guide (Passive Controls)
- Industry-accepted reconnaissance practices

**Assessment Phases**

1. **Reconnaissance**
   - Port and service discovery using Nmap
2. **Application Inspection**
   - HTTP header inspection via curl
   - Browser-side validation using Developer Tools
3. **Traffic Analysis**
   - Passive monitoring with OWASP ZAP
4. **Reporting**
   - Documentation of identified weaknesses and remediation guidance

No intrusive or disruptive actions were performed.

# 4.Tools Used

| Tool | Purpose |
|---|---|
| **Nmap** | Network port and service discovery |
| **curl** | HTTP security header inspection |
| **Browser DevTools** | Client-side validation of headers |
| **OWASP ZAP** | Passive vulnerability detection |

# 5.Technical Findings

## 5.1 Open Ports and Services

**Description:**

Passive Nmap scanning revealed that the application exposes only essential web services.

**Findings:**

- TCP Port 80 (HTTP)
- TCP Port 443 (HTTPS)

Security Impact:

✓ Minimal attack surface at network level

✓ No unnecessary services exposed

```
┌──(razi㉿kali)-[~/FUTURE_CS_01]
└─$ nmap -sT -Pn -F juice-shop.herokuapp.com
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-17 17:11 CET
Nmap scan report for juice-shop.herokuapp.com (54.220.192.176)
Host is up (0.068s latency).
Other addresses for juice-shop.herokuapp.com (not scanned): 54.73.53.134 46.137.15.86
rDNS record for 54.220.192.176: ec2-54-220-192-176.eu-west-1.compute.amazonaws.com
Not shown: 98 filtered tcp ports (no-response)
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 4.80 seconds
```

Nmap passive scan results

## 5.2 HTTP Security Header Analysis

**Description:**

HTTP response headers were analyzed to evaluate security hardening.

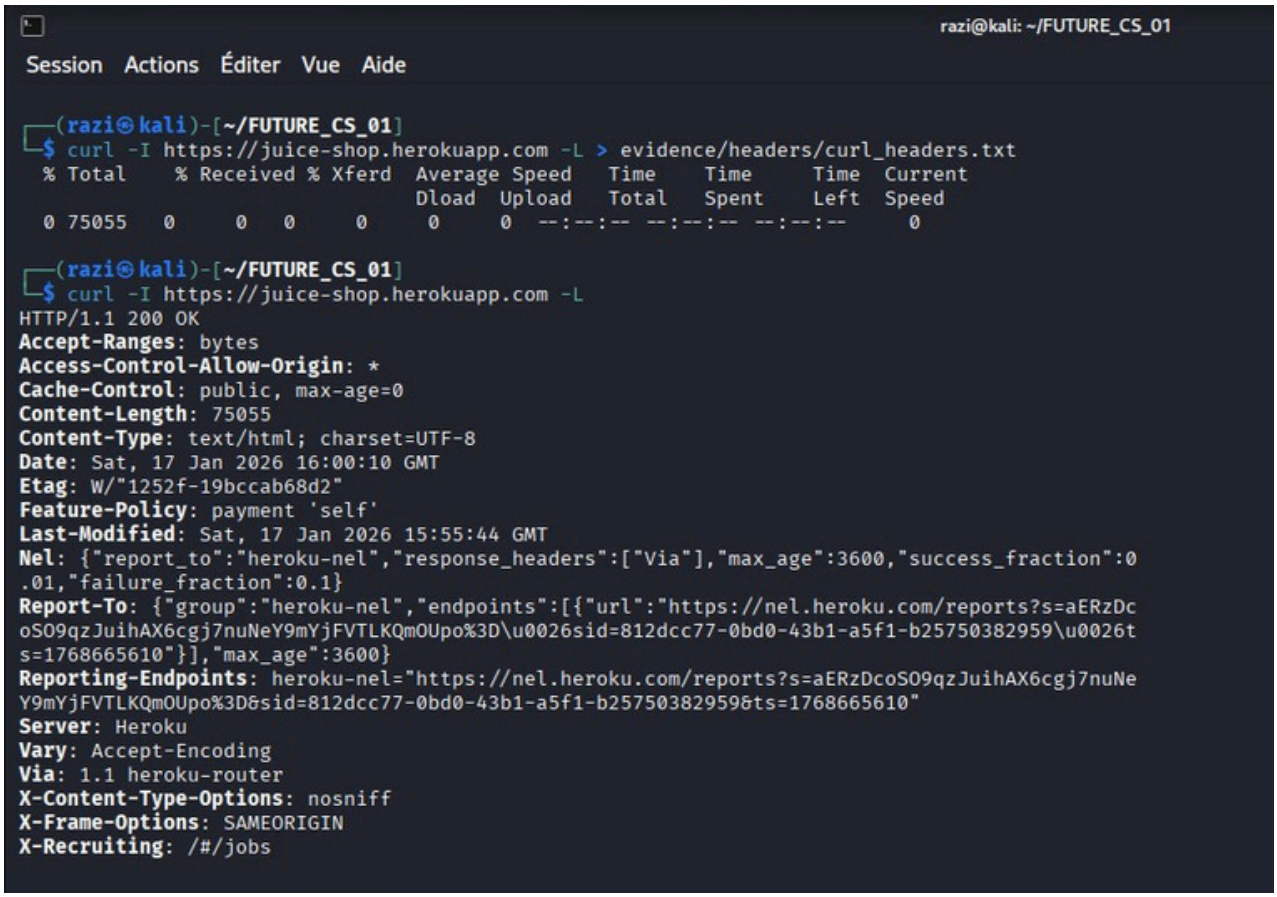
**Identified Issues:**

- Missing Content-Security-Policy (CSP)
- Missing Strict-Transport-Security (HSTS)
- Permissive Access-Control-Allow-Origin: *
- Server information disclosure via headers

**Security Impact:**

These misconfigurations increase exposure to:

- Cross-Site Scripting (XSS)
- Clickjacking
- HTTPS downgrade attacks
- Information leakage




curl HTTP header inspection

# 5.3 Browser-Side Header Validation

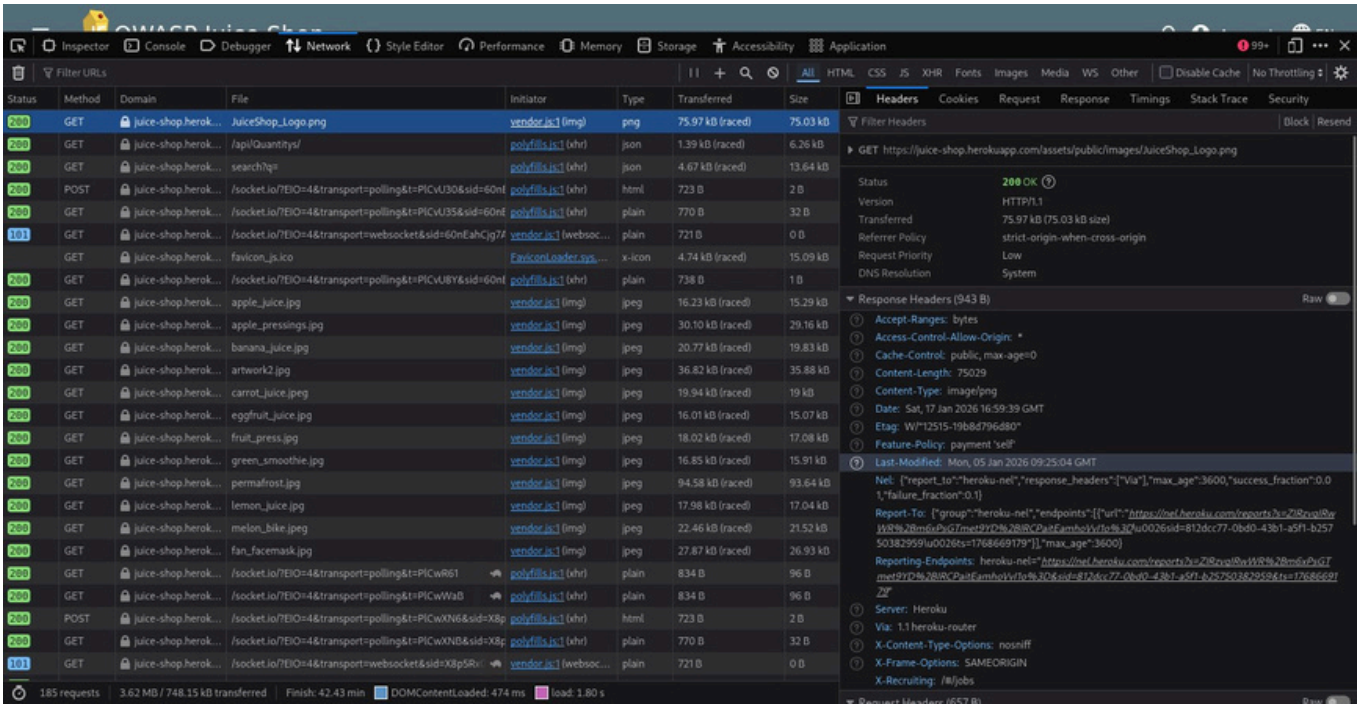
**Description:**

Browser Developer Tools were used to confirm server-side findings from the client perspective.

**Observation:**

Headers observed in the browser match curl results, confirming:

- Missing CSP
- Missing HSTS
- Insecure CORS policy



Browser Network tab – response headers

## 5.4 Passive Vulnerability Detection (OWASP ZAP)

**Description:**

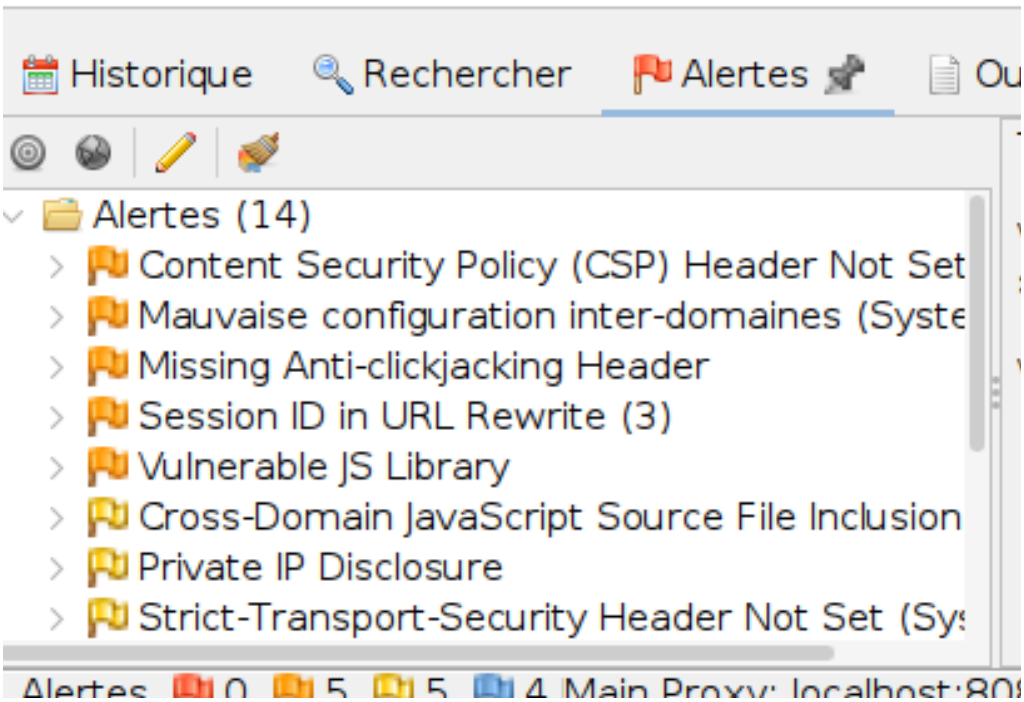
OWASP ZAP was used in passive mode while navigating the application.

**Detected Alerts:**

- Content Security Policy Header Not Set
- Missing Anti-Clickjacking Header
- Missing HSTS Header
- Information Disclosure via Headers
- Vulnerable JavaScript Library (Informational)

**Risk Levels:**

- Medium Risk: Multiple
- Low Risk: Multiple
- High/Critical: None




curl HTTP header inspection

## 5.5 Request & Response Traffic Inspection

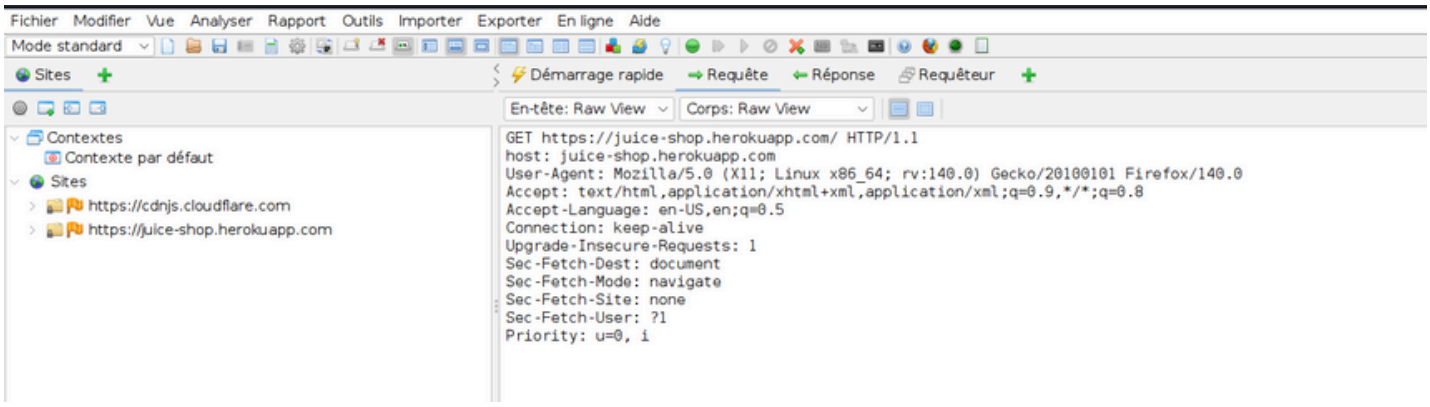
**Description:**

Captured HTTP requests and responses were reviewed to validate:

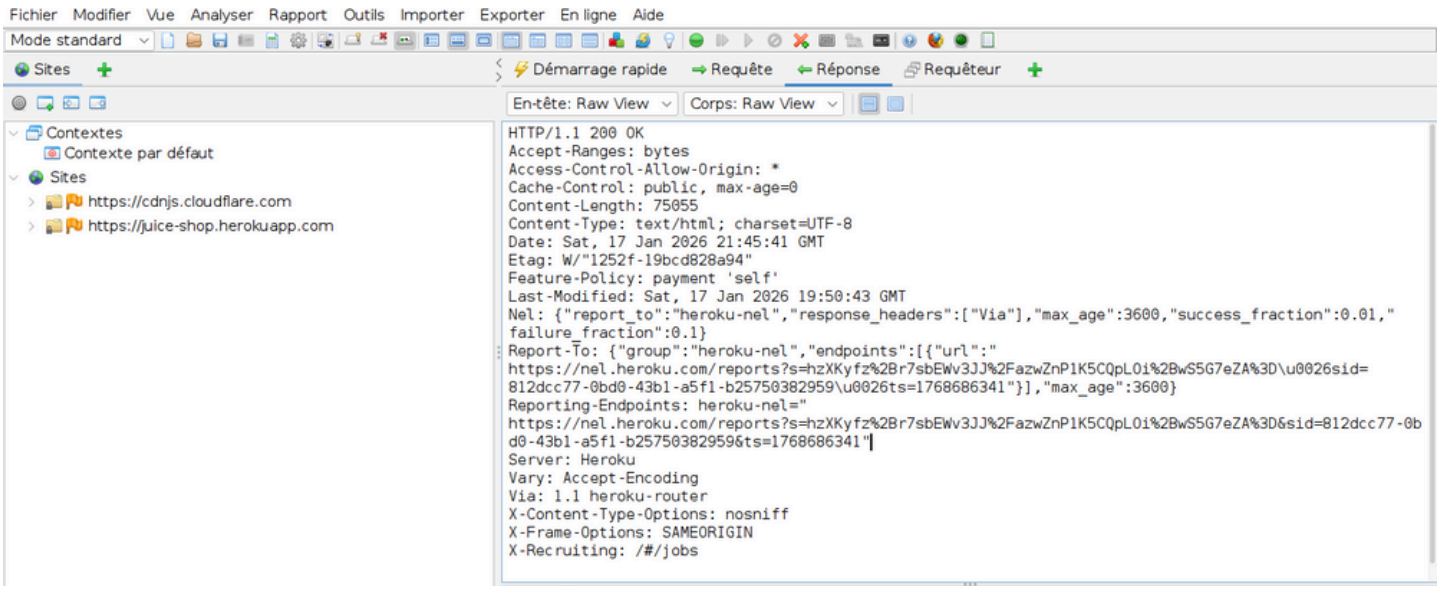
- Proper status codes
- Header consistency
- Absence of sensitive data in responses

**Observation:**

Responses return valid status codes (200 OK) and do not expose sensitive payload data.




ZAP Request raw view




ZAP Response raw view

# 6.Risk Summary

| Severity | Count |
|---|---|
| **High / Critical** | **0** |
| **Medium** | **4** |
| **Low** | **2** |
| **Informational** | **Several** |

# 7.Recommendations

**Security Hardening**

- Implement Content Security Policy (CSP)
- Enforce Strict-Transport-Security (HSTS)
- Restrict CORS to trusted origins only
- Remove unnecessary server header disclosures

**Operational Security**

- Regular passive security reviews
- Automated header compliance checks
- Periodic penetration testing for active threats

# 8.Conclusion

The OWASP Juice Shop application demonstrates a stable and functional architecture with limited network exposure. However, the absence of key HTTP security headers and relaxed configuration policies present avoidable risks.

By implementing the recommended security controls, the application's resilience against common web-based attacks would be significantly improved. The findings indicate a medium overall risk posture, suitable for educational use but requiring additional hardening for production environments.