

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

Executive Summary	1
Scope of Assessment	2
Methodology	3
Tools Used	4
Technical Findings	5
Risk Summary	6
Recommendations	7
Conclusion	8

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.

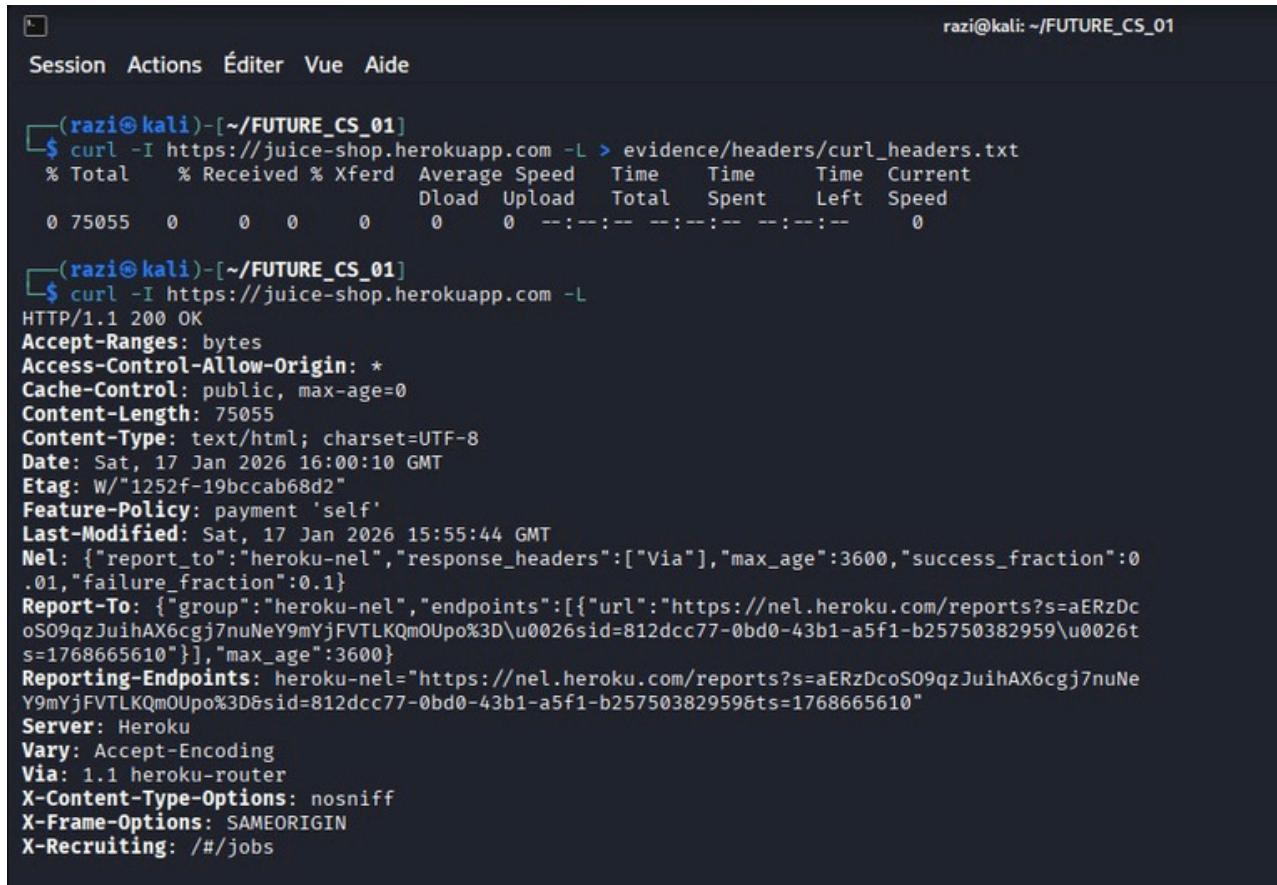
Identified Issues:

- 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



The screenshot shows a terminal window with the following content:

```
razi@kali: ~/FUTURE_CS_01
Session Actions Éditer Vue Aide

(razi@kali)-[~/FUTURE_CS_01]
$ curl -I https://juice-shop.herokuapp.com -L > evidence/headers(curl_headers.txt
HTTP/1.1 200 OK
Accept-Ranges: bytes
Access-Control-Allow-Origin: *
Cache-Control: public, max-age=0
Content-Length: 75055
Content-Type: text/html; charset=UTF-8
Date: Sat, 17 Jan 2026 16:00:10 GMT
Etag: W/"1252f-19bccab68d2"
Feature-Policy: payment 'self'
Last-Modified: Sat, 17 Jan 2026 15:55:44 GMT
Nel: {"report_to": "heroku-nel", "response_headers": ["Via"], "max_age": 3600, "success_fraction": 0.01, "failure_fraction": 0.1}
Report-To: {"group": "heroku-nel", "endpoints": [{"url": "https://nel.herokuapp.com/reports?s=aERzDcoS09qzJuhiAX6cgj7nuNeY9mYjFVTLKQm0Upo%3D\u0026sid=812dcc77-0bd0-43b1-a5f1-b25750382959\u0026ts=1768665610"}], "max_age": 3600}
Reporting-Endpoints: heroku-nel="https://nel.herokuapp.com/reports?s=aERzDcoS09qzJuhiAX6cgj7nuNeY9mYjFVTLKQm0Upo%3D&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&ts=1768665610"
Server: Heroku
Vary: Accept-Encoding
Via: 1.1 heroku-router
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-Recruiting: /#/jobs
```

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

The screenshot shows the OWASCD UI Network tab. It displays a list of network requests and their details, including initiator, type, transferred size, and status. On the right side, there is a detailed view of the response headers for a specific request, including fields like Status, Version, Transferred, and various header names like Accept-Ranges, Access-Control-Allow-Origin, Cache-Control, Content-Length, Content-Type, Date, Etag, Feature-Policy, Last-Modified, Report-To, Server, Via, X-Content-Type-Options, X-Frame-Options, and X-Recruiting.

Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	GET	Juice-shop.herokuapp.com	JuiceShop_Logo.png	vendor.js:1 (img)	png	75.97 kB (raced)	75.03 kB
200	GET	Juice-shop.herokuapp.com	/api/Quantities/	polyfills.js:1 (xhr)	json	1.39 kB (raced)	6.26 kB
200	GET	Juice-shop.herokuapp.com	search?qa	polyfills.js:1 (xhr)	json	4.67 kB (raced)	13.64 kB
200	POST	Juice-shop.herokuapp.com	/socket.io/4&transport=polling&t=PiCVU3O&sid=60nf	polyfills.js:1 (xhr)	html	723 B	2 B
200	GET	Juice-shop.herokuapp.com	/socket.io/4&transport=polling&t=PiCVU3S&sid=60nf	polyfills.js:1 (xhr)	plain	770 B	32 B
101	GET	Juice-shop.herokuapp.com	/socket.io/4&transport=websocket&sid=60nEahCjg77	vendor.js:1 (websock...	plain	721 B	0 B
200	GET	Juice-shop.herokuapp.com	favicon_js.ico	FaviconLoader.js:1	x-icon	4.74 kB (raced)	15.09 kB
200	GET	Juice-shop.herokuapp.com	/socket.io/7/EIO=4&transport=polling&t=PiCVUY&sid=60nf	polyfills.js:1 (xhr)	plain	738 B	1 B
200	GET	Juice-shop.herokuapp.com	apple_juice.jpg	vendor.js:1 (img)	jpeg	16.23 kB (raced)	15.29 kB
200	GET	Juice-shop.herokuapp.com	apple_pressings.jpg	vendor.js:1 (img)	jpeg	30.10 kB (raced)	29.16 kB
200	GET	Juice-shop.herokuapp.com	banana_juice.jpg	vendor.js:1 (img)	jpeg	20.77 kB (raced)	19.83 kB
200	GET	Juice-shop.herokuapp.com	artwork2.jpg	vendor.js:1 (img)	jpeg	36.82 kB (raced)	35.88 kB
200	GET	Juice-shop.herokuapp.com	carrot_juice.jpeg	vendor.js:1 (img)	jpeg	19.94 kB (raced)	19 kB
200	GET	Juice-shop.herokuapp.com	eggfruit_juice.jpg	vendor.js:1 (img)	jpeg	16.01 kB (raced)	15.07 kB
200	GET	Juice-shop.herokuapp.com	fruit_press.jpg	vendor.js:1 (img)	jpeg	18.02 kB (raced)	17.08 kB
200	GET	Juice-shop.herokuapp.com	green_smoothie.jpg	vendor.js:1 (img)	jpeg	16.85 kB (raced)	15.91 kB
200	GET	Juice-shop.herokuapp.com	permafrost.jpg	vendor.js:1 (img)	jpeg	94.58 kB (raced)	93.64 kB
200	GET	Juice-shop.herokuapp.com	lemon_juice.jpg	vendor.js:1 (img)	jpeg	17.98 kB (raced)	17.04 kB
200	GET	Juice-shop.herokuapp.com	melon_bike.jpeg	vendor.js:1 (img)	jpeg	22.46 kB (raced)	21.52 kB
200	GET	Juice-shop.herokuapp.com	fan_facemask.jpg	vendor.js:1 (img)	jpeg	27.87 kB (raced)	26.93 kB
200	GET	Juice-shop.herokuapp.com	/socket.io/4&transport=polling&t=PiCwR6I	polyfills.js:1 (xhr)	plain	834 B	96 B
200	GET	Juice-shop.herokuapp.com	/socket.io/7/EIO=4&transport=polling&t=PiCwXN6&sid=X8p	polyfills.js:1 (xhr)	plain	834 B	96 B
200	POST	Juice-shop.herokuapp.com	/socket.io/7/EIO=4&transport=polling&t=PiCwR6I	polyfills.js:1 (xhr)	html	723 B	2 B
200	GET	Juice-shop.herokuapp.com	/socket.io/7/EIO=4&transport=websocket&sid=60p5Rv	polyfills.js:1 (xhr)	plain	770 B	32 B
101	GET	Juice-shop.herokuapp.com	/socket.io/7/EIO=4&transport=websocket&sid=60p5Rv	vendor.js:1 (websock...	plain	721 B	0 B

185 requests | 3.62 MB / 748.15 kB transferred | Finish: 42.43 min | DOMContentLoaded: 474 ms | load: 1.80 s

Browser Network tab – response headers

5.4 Passive Vulnerability Detection (OWASP ZAP)

Description:

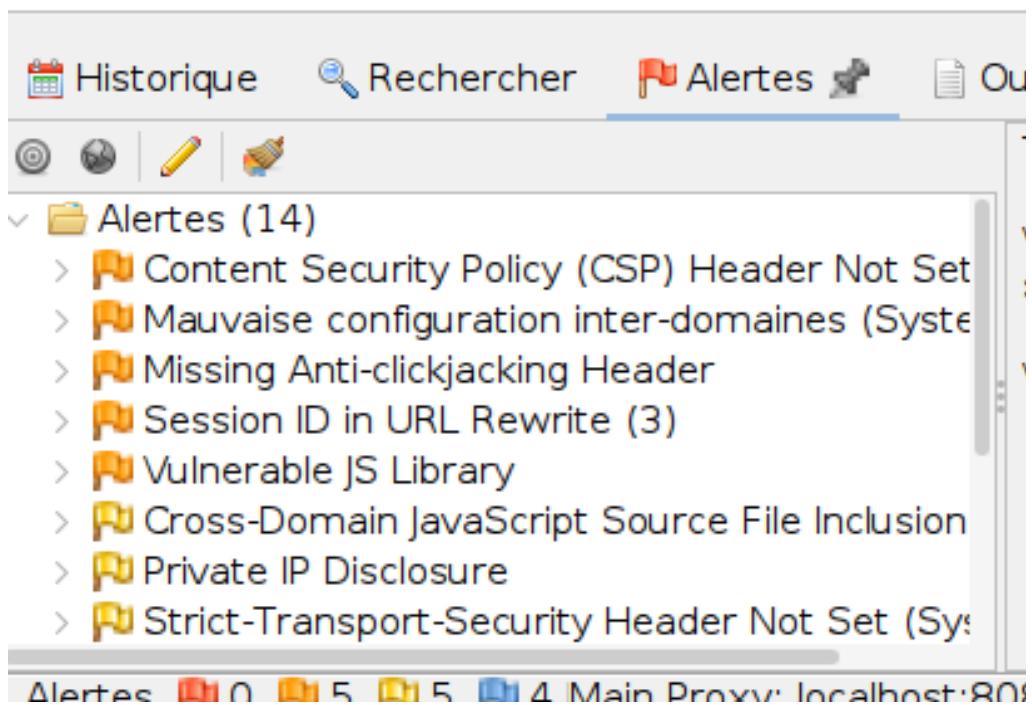
OWASP ZAP was used in passive mode while navigating the application.

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.

```

GET https://juice-shop.herokuapp.com/ HTTP/1.1
host: juice-shop.herokuapp.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Priority: u=0, i
  
```

ZAP Request raw view

```

HTTP/1.1 200 OK
Accept-Ranges: bytes
Access-Control-Allow-Origin: *
Cache-Control: public, max-age=0
Content-Length: 75055
Content-Type: text/html; charset=UTF-8
Date: Sat, 17 Jan 2026 21:45:41 GMT
Etag: W/"1252f-19bcd828a94"
Feature-Policy: payment 'self'
Last-Modified: Sat, 17 Jan 2026 19:50:43 GMT
Nel: {"report_to": "heroku-nel", "response_headers": [{"Via": "1.1 heroku-router", "Age": 3600}], "max_age": 3600, "success_fraction": 0.01, "failure_fraction": 0.1}
Report-To: {"group": "heroku-nel", "endpoints": [{"url": "https://nel.herokuapp.com/reports?s=hzXKyfz%2Br7sbEWv3JJ%2FazwZnPlK5CQpL0i%2BwS5G7eZA%3D\u0026sid=812dcc77-0bd0-43b1-a5f1-b25750382959\u0026ts=1768686341"}], "max_age": 3600}
Reporting-Endpoints: heroku-nel
Reporting-Endpoint-Order: https://nel.herokuapp.com/reports?s=hzXKyfz%2Br7sbEWv3JJ%2FazwZnPlK5CQpL0i%2BwS5G7eZA%3D&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&ts=1768686341
Server: Heroku
Vary: Accept-Encoding
Via: 1.1 heroku-router
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-Recruiting: /#/jobs

```

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.