■ Web Application Security Assessment Report

Target Application: **OWASP Juice Shop**

Prepared by**: Aditya Mehta**

CIN ID: **FIT/JAN26/CS5511**

Date: **05 Feb 2026**

Introduction :

- This task focuses on identifying common security misconfigurations and vulnerabilities in a deliberately vulnerable application.
- For this assessment, OWASP Juice Shop was selected as the target application and OWASP ZAP was used to perform automated security scanning.

Objectives of this task:

- Understand web application vulnerabilities
- Perform automated vulnerability scanning
- Analyze security misconfigurations
- Document findings in a professional manner

Tools & Tech Used :

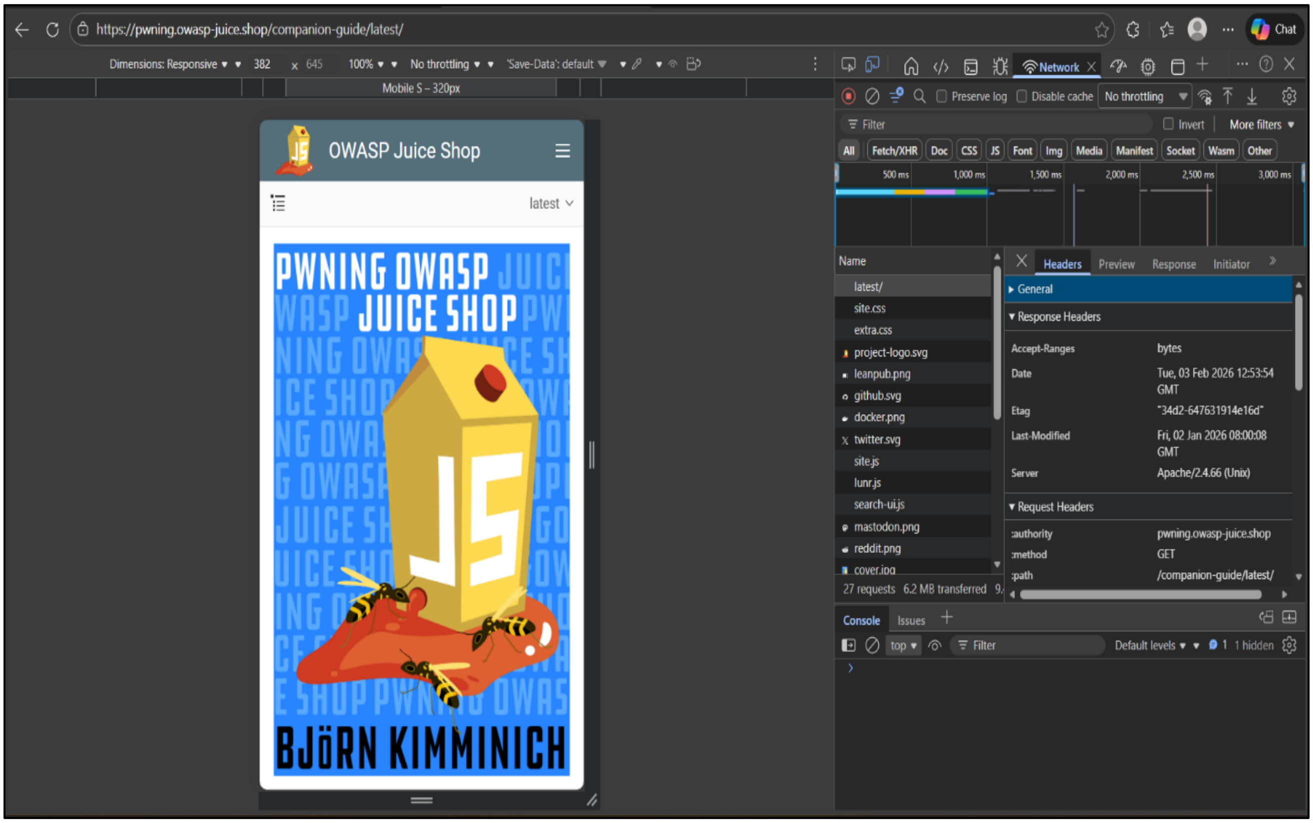
Tools	Purpose
OWASP ZAP	Automated vulnerability scanning
OWASP Juice Shop	Intentionally vulnerable web app
Browser DevTools	Header & request inspection
Windows Proxy Settings	Traffic interception

Target Application: OWASP Juice Shop

- OWASP Juice Shop is an intentionally vulnerable web application created for security testing and learning purposes.
- It contains common vulnerabilities listed in the OWASP Top 10.

Target URL:

https://pwning.owasp-juice.shop



Name
latest/
site.css
extra.css
project-logo.svg
leanpub.png
github.svg
docker.png
twitter.svg
site.js
lunr.js
search-ui.js
mastodon.png
reddit.png
cover.jpg

27 requests 6.2 MB transferred

▸ General

▾ Response Headers

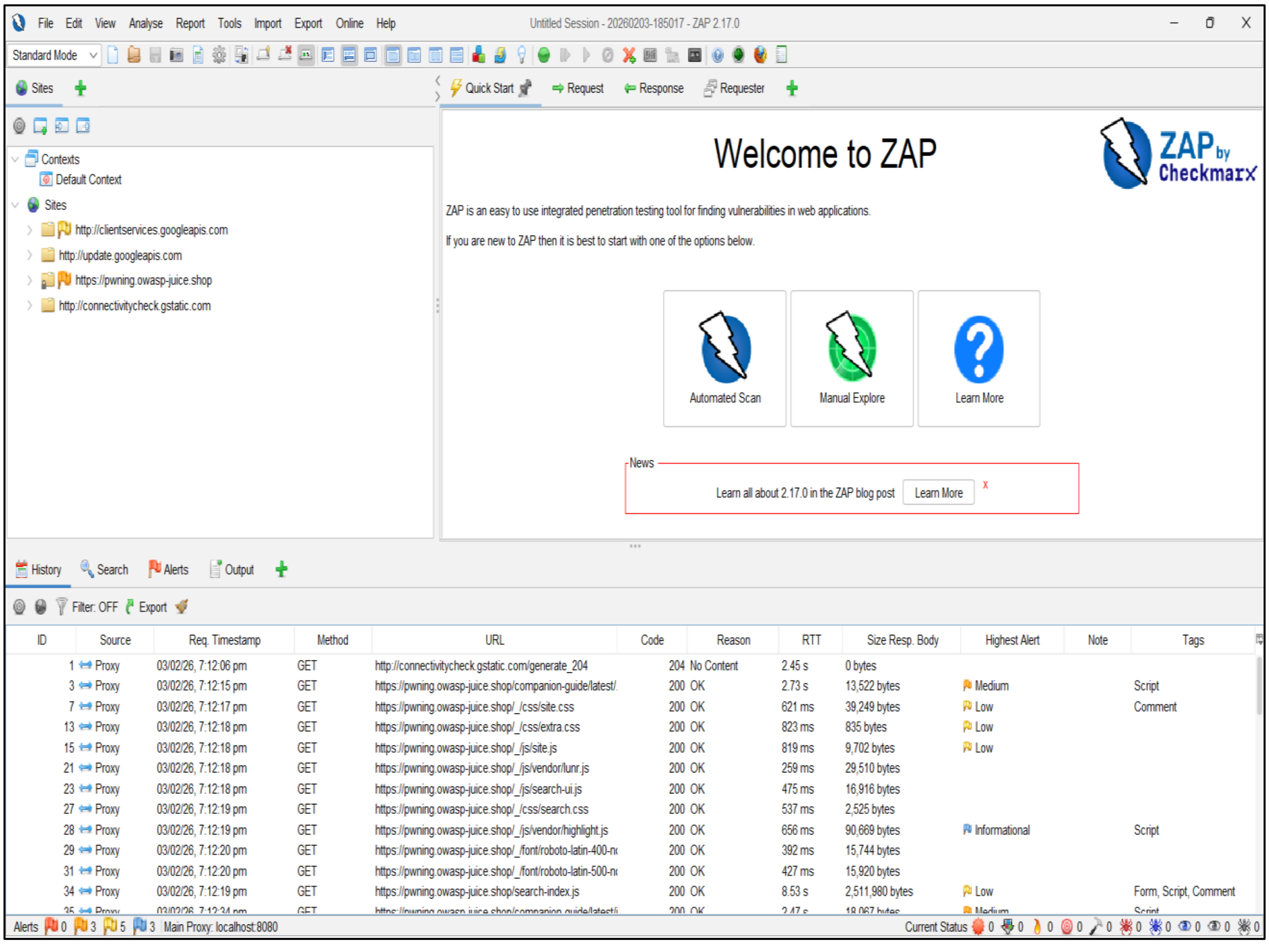
Accept-Ranges	bytes
Date	Tue, 03 Feb 2026 12:53:54 GMT
Etag	"34d2-647631914e16d"
Last-Modified	Fri, 02 Jan 2026 08:00:08 GMT
Server	Apache/2.4.66 (Unix)

▾ Request Headers

:authority	pwning.owasp-juice.shop
:method	GET
:path	/companion-guide/latest/

Assessment Methodology :

- The following steps were performed during the assessment:
- Configured system proxy for OWASP ZAP
- Loaded OWASP Juice Shop through browser
- Intercepted requests using ZAP proxy
- Performed automated scanning
- Reviewed alerts and security findings
- Documented vulnerabilities



Untitled Session - 20260203-185017 - ZAP 2.17.0

Contexts
Default Context
Sites
http://clientservices.googleapis.com
http://update.googleapis.com
https://pwning.owasp-juice.shop
http://connectivitycheck.gstatic.com

Welcome to ZAP

ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

If you are new to ZAP then it is best to start with one of the options below.

Learn all about 2.17.0 in the ZAP blog post

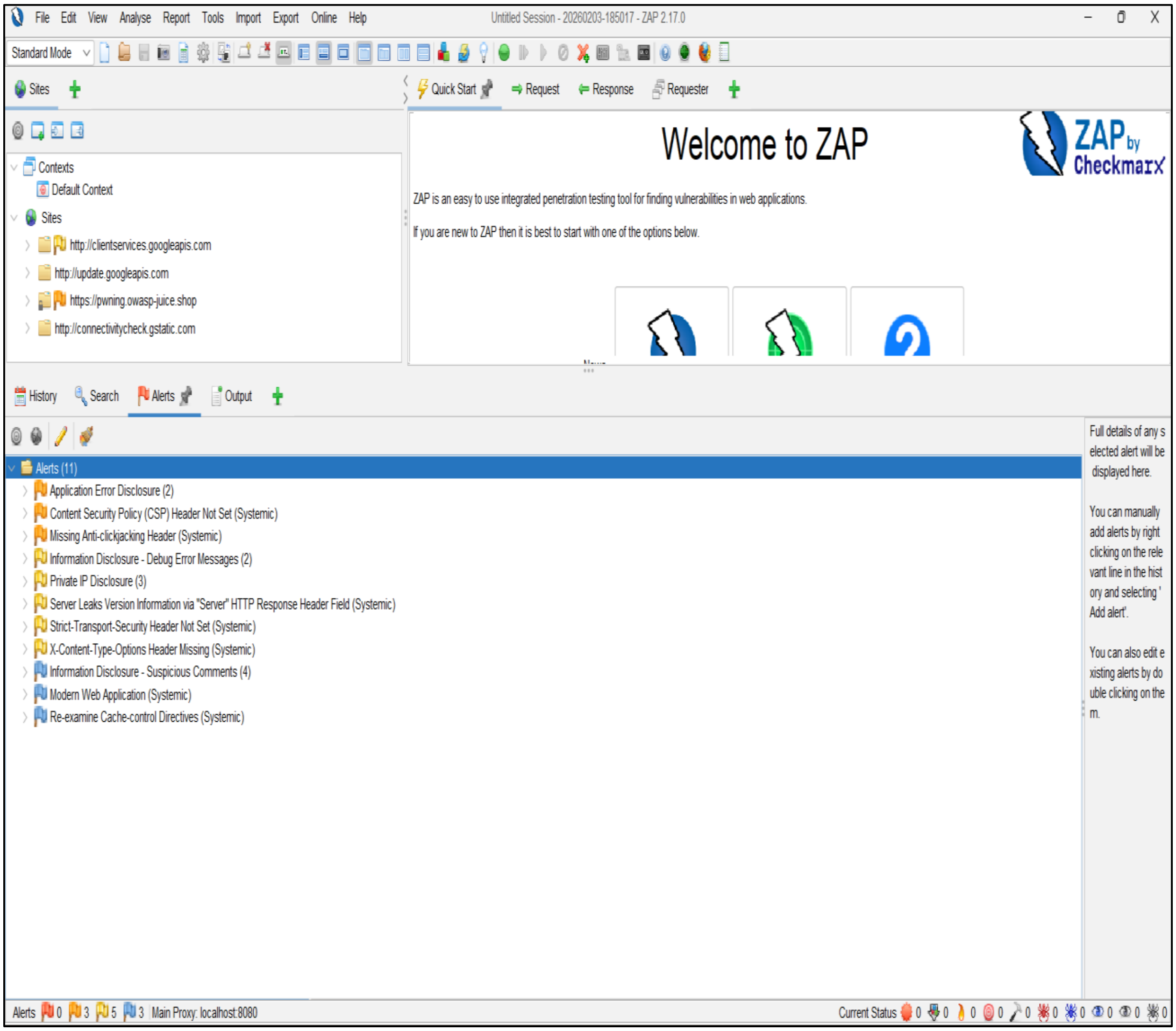
ID	Source	Req. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Body	Highest Alert	Note	Tags
1	Proxy	03/02/26, 7:12:06 pm	GET	http://connectivitycheck.gstatic.com/generate_204	204	No Content	2.45 s	0 bytes			
3	Proxy	03/02/26, 7:12:15 pm	GET	https://pwning.owasp-juice.shop/companion-guide/latest/.	200	OK	2.73 s	13,522 bytes	Medium		Script
7	Proxy	03/02/26, 7:12:17 pm	GET	https://pwning.owasp-juice.shop/_/css/site.css	200	OK	621 ms	39,249 bytes	Low		Comment
13	Proxy	03/02/26, 7:12:18 pm	GET	https://pwning.owasp-juice.shop/_/css/extra.css	200	OK	823 ms	835 bytes	Low		
15	Proxy	03/02/26, 7:12:18 pm	GET	https://pwning.owasp-juice.shop/_/js/site.js	200	OK	819 ms	9,702 bytes	Low		
21	Proxy	03/02/26, 7:12:18 pm	GET	https://pwning.owasp-juice.shop/_/js/vendor/lunr.js	200	OK	259 ms	29,510 bytes			
23	Proxy	03/02/26, 7:12:18 pm	GET	https://pwning.owasp-juice.shop/_/js/search-ui.js	200	OK	475 ms	16,916 bytes			
27	Proxy	03/02/26, 7:12:19 pm	GET	https://pwning.owasp-juice.shop/_/css/search.css	200	OK	537 ms	2,525 bytes			
28	Proxy	03/02/26, 7:12:19 pm	GET	https://pwning.owasp-juice.shop/_/js/vendor/highlight.js	200	OK	656 ms	90,669 bytes	Informational		Script
29	Proxy	03/02/26, 7:12:20 pm	GET	https://pwning.owasp-juice.shop/_/font/roboto-latin-400-n(	200	OK	392 ms	15,744 bytes			
31	Proxy	03/02/26, 7:12:20 pm	GET	https://pwning.owasp-juice.shop/_/font/roboto-latin-500-n(	200	OK	427 ms	15,920 bytes			
34	Proxy	03/02/26, 7:12:19 pm	GET	https://pwning.owasp-juice.shop/search-index.js	200	OK	8.53 s	2,511,980 bytes	Low		Form, Script, Comment
35	Proxy	03/02/26, 7:12:34 pm	GET	https://pwning.owasp-juice.shop/companion-guide/latest/i	200	OK	2.47 s	18,067 bytes	Medium		Script

Alerts 0 3 5 3 Main Proxy: localhost:8080

Current Status 0 0 0 0 0 0 0 0 0 0

OWASP ZAP Configuration :

- ZAP was configured as a local proxy (localhost:8080)
- Browser traffic was routed through ZAP
- Target site was explored manually
- Automated scan was initiated



Untitled Session - 20260203-185017 - ZAP 2.17.0

- Contexts
  - Default Context
- Sites
  - http://clientservices.googleapis.com
  - http://update.googleapis.com
  - https://pwning.owasp-juice.shop
  - http://connectivitycheck.gstatic.com

Welcome to ZAP

ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

If you are new to ZAP then it is best to start with one of the options below.

- Alerts (11)
  - Application Error Disclosure (2)
  - Content Security Policy (CSP) Header Not Set (Systemic)
  - Missing Anti-clickjacking Header (Systemic)
  - Information Disclosure - Debug Error Messages (2)
  - Private IP Disclosure (3)
  - Server Leaks Version Information via "Server" HTTP Response Header Field (Systemic)
  - Strict-Transport-Security Header Not Set (Systemic)
  - X-Content-Type-Options Header Missing (Systemic)
  - Information Disclosure - Suspicious Comments (4)
  - Modern Web Application (Systemic)
  - Re-examine Cache-control Directives (Systemic)

Full details of any s elected alert will be displayed here.

You can manually add alerts by right clicking on the rele vant line in the hist ory and selecting ' Add alert'.

You can also edit e xisting alerts by do uble clicking on the m.

Alerts 0 3 5 3 Main Proxy: localhost:8080

Current Status 0 0 0 0 0 0 0 0 0 0

Identified Vulnerabilities :

- Some of the key findings include:
- Missing Content Security Policy (CSP) Header
- Missing Anti-Clickjacking Header
- Application Error Disclosure
- Information Disclosure via Debug Messages
- Server Version Disclosure
- Missing Strict-Transport-Security Header
- X-Content-Type-Options Header Missing

HEADER ANALYSIS :

- Using browser developer tools, HTTP response headers were analyzed.

Findings :

- Server version exposed (Apache/2.4.66)
- Security headers missing
- Improper cache control directives

IMPACT ANALYSIS :

⚠ Security Impact

- If such vulnerabilities exist in a real-world application, attackers could:
- Perform clickjacking attacks
- Gather sensitive server information
- Exploit misconfigurations

Conclusion :

- This task helped in understanding how automated security tools like OWASP ZAP can identify vulnerabilities in web applications.
- OWASP Juice Shop served as an excellent platform to practice:
- Web security testing
- Vulnerability identification
- Security reporting
- Overall, the task strengthened foundational knowledge in web application security assessment.

GitHub Repository :

- All work, screenshots, and documentation are available at:

https://github.com/iamadityamehta/FUTURE_CS_01