SECURITY ASSESSMENT REPORT

(Task 1 : **Web Application Security Assessment Report)**

Intern Name: Tanish Jaiswal
Internship Program: Cyber Security Internship – Future Interns
Task: – Web Application Security Testing
Target Application: OWASP Juice Shop
Target URL: https://demo.owasp-juice.shop
Assessment Type: Vulnerability Assessment
Tool Used: OWASP ZAP
Date: 18-12-2025

1. Introduction

This project focuses on performing a vulnerability assessment of a deliberately vulnerable web application, OWASP Juice Shop. The objective of this assessment is to identify common web application security vulnerabilities using ethical hacking tools and to map the findings to OWASP Top 10 security risks.

The assessment simulates a real-world client engagement where web applications used by startups, SaaS platforms, and e-commerce companies must be tested and secured against potential cyber threats.

2. Scope of Testing

In-Scope

Target website: https://demo.owasp-juice.shop

All publicly accessible pages and directories

Out-of-Scope

External domains (Google, GitHub, CDN links, social media)

Third-party services

3. Tools and Methodology

Tools Used

OWASP ZAP– Automated vulnerability scanning and analysis

Web Browser – Manual verification

WPS office – Documentation

Methodology

1. Automated scan using OWASP ZAP
2. Identification of Medium-risk vulnerabilities
3. Manual verification using browser
4. Mapping vulnerabilities to OWASP Top 10
5. Documentation with screenshots and remediation steps

4. Vulnerability Findings

Vulnerability 1: Directory Browsing Enabled

Risk Level: Medium

OWASP Top 10 Category:

A05 – Security Misconfiguration

Description:

The application allows directory browsing, which exposes internal directories and files to unauthorized users. This allows attackers to gain insight into the application's internal structure.

Affected URL:

https://demo.owasp-juice.shop/ftp/

Impact:

An attacker can view, download, or analyze internal files, which may contain sensitive information and assist in further attacks.

Mitigation:

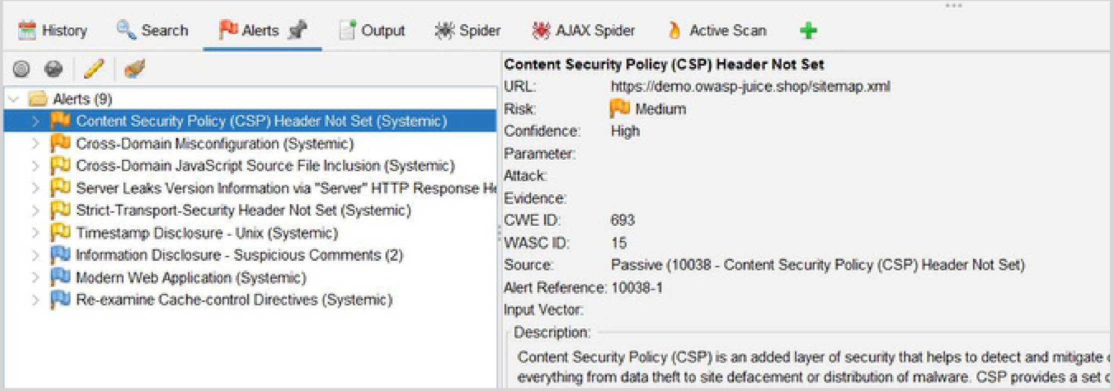
Disable directory listing on the server

Restrict access to sensitive directories

Apply proper access control rules

Screenshot:

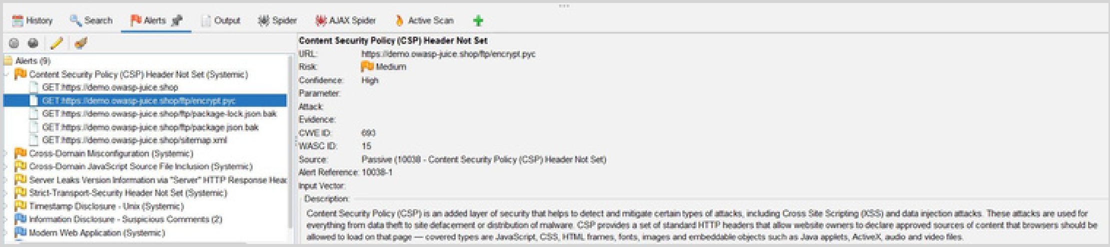
All alerts



Alerts (9)
- Content Security Policy (CSP) Header Not Set (Systemic)
- Cross-Domain Misconfiguration (Systemic)
- Cross-Domain JavaScript Source File Inclusion (Systemic)
- Server Leaks Version Information via "Server" HTTP Response He
- Strict-Transport-Security Header Not Set (Systemic)
- Timestamp Disclosure - Unix (Systemic)
- Information Disclosure - Suspicious Comments (2)
- Modern Web Application (Systemic)
- Re-examine Cache-control Directives (Systemic)

Content Security Policy (CSP) Header Not Set
URL: https://demo.owasp-juice.shop/sitemap.xml
Risk: Medium
Confidence: High
Parameter:
Attack:
Evidence:
CWE ID: 693
WASC ID: 15
Source: Passive (10038 - Content Security Policy (CSP) Header Not Set)
Alert Reference: 10038-1
Input Vector:
Description:
Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate
everything from data theft to site defacement or distribution of malware. CSP provides a set

Screenshot of Vulnerability 1 Directory Browsing Enabled



Alerts (9)
- Content Security Policy (CSP) Header Not Set (Systemic)
  - GET:https://demo.owasp-juice.shop
  - GET:https://demo.owasp-juice.shop/ftp/encrypt.pyc
  - GET:https://demo.owasp-juice.shop/ftp/package-lock.json.bak
  - GET:https://demo.owasp-juice.shop/ftp/package.json.bak
  - GET:https://demo.owasp-juice.shop/sitemap.xml
- Cross-Domain Misconfiguration (Systemic)
- Cross-Domain JavaScript Source File Inclusion (Systemic)
- Server Leaks Version Information via "Server" HTTP Response Heac
- Strict-Transport-Security Header Not Set (Systemic)
- Timestamp Disclosure - Unix (Systemic)
- Information Disclosure - Suspicious Comments (2)
- Modern Web Application (Systemic)

Content Security Policy (CSP) Header Not Set
URL: https://demo.owasp-juice.shop/ftp/encrypt.pyc
Risk: Medium
Confidence: High
Parameter:
Attack:
Evidence:
CWE ID: 693
WASC ID: 15
Source: Passive (10038 - Content Security Policy (CSP) Header Not Set)
Alert Reference: 10038-1
Input Vector:
Description:
Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.

Vulnerability 2: Sensitive File Exposure

Risk Level: Medium

OWASP Top 10 Category: A05 – Security Misconfiguration

Description:

Sensitive files such as backup files (.bak), documentation files, and configuration-related files were publicly accessible through the FTP directory.

Examples of Exposed Files:

coupons_2013.md.bak

package.json.bak

Impact:

Exposure of sensitive files may lead to information disclosure, credential leakage, or further system compromise.

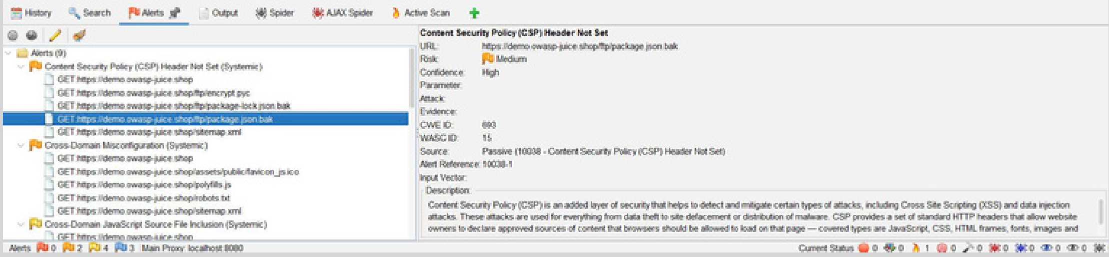
Mitigation:

Remove backup files from production environments

Restrict access to sensitive files

Follow secure deployment practices

Screenshot *Exposed files list*

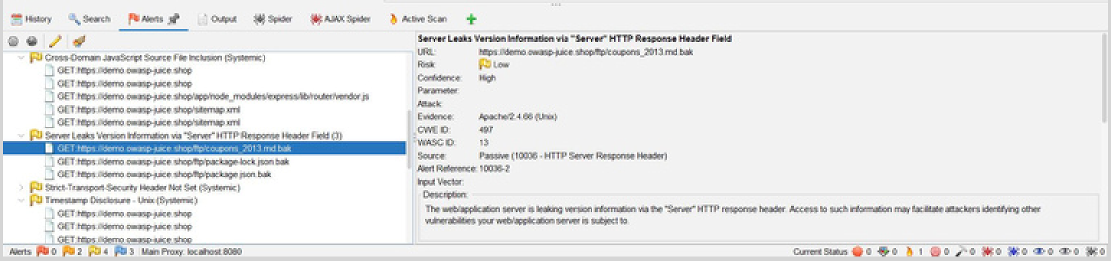


Alerts (9)
Content Security Policy (CSP) Header Not Set (Systemic)
GET https://demo.owasp-juice.shop
GET https://demo.owasp-juice.shop/ftp/encrypt.pyc
GET https://demo.owasp-juice.shop/ftp/package-lock.json.bak
GET https://demo.owasp-juice.shop/ftp/package.json.bak
GET https://demo.owasp-juice.shop/sitemap.xml
Cross-Domain Misconfiguration (Systemic)
GET https://demo.owasp-juice.shop
GET https://demo.owasp-juice.shop/assets/public/favicon_js.ico
GET https://demo.owasp-juice.shop/polyfills.js
GET https://demo.owasp-juice.shop/robots.txt
GET https://demo.owasp-juice.shop/sitemap.xml
Cross-Domain JavaScript Source File Inclusion (Systemic)
GET https://demo.owasp-juice.shop

Alerts 0 2 4 3 Main Proxy: localhost:8080

Content Security Policy (CSP) Header Not Set
URL: https://demo.owasp-juice.shop/ftp/package.json.bak
Risk: Medium
Confidence: High
Parameter:
Attack:
Evidence:
CWE ID: 693
WASC ID: 15
Source: Passive (10038 - Content Security Policy (CSP) Header Not Set)
Alert Reference: 10038-1
Input Vector:
Description:
Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and

Current Status 0 0 1 0 0 0 0 0 0



Cross-Domain JavaScript Source File Inclusion (Systemic)
GET https://demo.owasp-juice.shop
GET https://demo.owasp-juice.shop
GET https://demo.owasp-juice.shop/app/node_modules/express/lib/router/vendor.js
GET https://demo.owasp-juice.shop/sitemap.xml
GET https://demo.owasp-juice.shop/sitemap.xml
Server Leaks Version Information via "Server" HTTP Response Header Field (3)
GET https://demo.owasp-juice.shop/ftp/coupons_2013.md.bak
GET https://demo.owasp-juice.shop/ftp/package-lock.json.bak
GET https://demo.owasp-juice.shop/ftp/package.json.bak
Strict-Transport-Security Header Not Set (Systemic)
Timestamp Disclosure - Unix (Systemic)
GET https://demo.owasp-juice.shop
GET https://demo.owasp-juice.shop
GET https://demo.owasp-juice.shop

Alerts 0 2 4 3 Main Proxy: localhost:8080

Server Leaks Version Information via "Server" HTTP Response Header Field
URL: https://demo.owasp-juice.shop/ftp/coupons_2013.md.bak
Risk: Low
Confidence: High
Parameter:
Attack:
Evidence: Apache/2.4.66 (Unix)
CWE ID: 497
WASC ID: 13
Source: Passive (10036 - HTTP Server Response Header)
Alert Reference: 10036-2
Input Vector:
Description:
The web/application server is leaking version information via the "Server" HTTP response header. Access to such information may facilitate attackers identifying other vulnerabilities your web/application server is subject to.

Current Status 0 0 1 0 0 0 0 0 0 0

Vulnerability 3: Missing Security Headers

Risk Level: Medium

OWASP Top 10 Category: A05 – Security Misconfiguration

Description:
The application does not implement important HTTP security headers, which are used to protect against common web attacks.
Missing Headers Include:

X-Frame-Options

X-Content-Type-Options

Content-Security-Policy

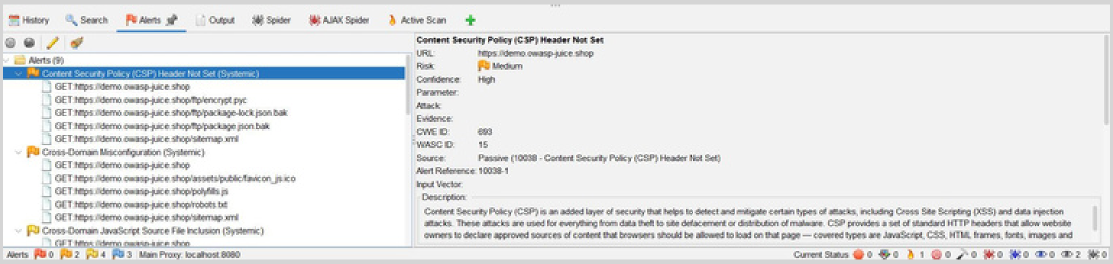
Impact:
The absence of these headers increases the risk of clickjacking, MIME-type attacks, and cross-site scripting.
Mitigation:

Configure security headers at the server level

Implement Content Security Policy (CSP)

Enable X-Frame-Options and related headers

***Screenshot**: ZAP alert showing missing headers*



```
Alerts (9)
  Content Security Policy (CSP) Header Not Set (Systemic)
    GET:https://demo.owasp-juice.shop
    GET:https://demo.owasp-juice.shop/ftp/encrypt.pyc
    GET:https://demo.owasp-juice.shop/ftp/package-lock.json.bak
    GET:https://demo.owasp-juice.shop/ftp/package.json.bak
    GET:https://demo.owasp-juice.shop/sitemap.xml
  Cross-Domain Misconfiguration (Systemic)
    GET:https://demo.owasp-juice.shop
    GET:https://demo.owasp-juice.shop/assets/public/favicon_js.ico
    GET:https://demo.owasp-juice.shop/polyfills.js
    GET:https://demo.owasp-juice.shop/robots.txt
    GET:https://demo.owasp-juice.shop/sitemap.xml
  Cross-Domain JavaScript Source File Inclusion (Systemic)
    GET:https://demo.owasp-juice.shop
```

Content Security Policy (CSP) Header Not Set

URL: https://demo.owasp-juice.shop
Risk: Medium
Confidence: High
Parameter:
Attack:
Evidence:
CWE ID: 693
WASC ID: 15
Source: Passive (10038 - Content Security Policy (CSP) Header Not Set)
Alert Reference: 10038-1
Input Vector:

Description:

Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and

Alerts 0 2 4 3 Main Proxy: localhost:8080

Current Status 0 0 1 0 0 0 0 0 2 0

Vulnerability 4: Information Disclosure

Risk Level: Medium

OWASP Top 10 Category: A01 – Broken Access Control

Description:
Files such as robots.txt and sitemap.xml disclose application structure and accessible paths, which can assist attackers during reconnaissance.

Impact:

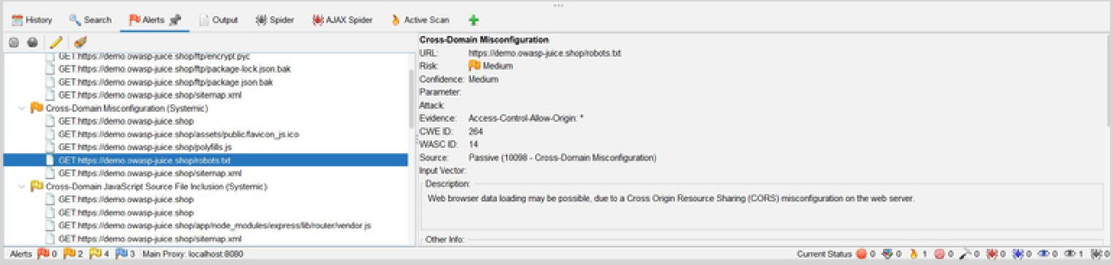
Reveals internal application paths that may be targeted in further attacks.

Mitigation:

Avoid exposing sensitive endpoints

Restrict access to internal paths where possible

***Screenshot:** robots.txt / sitemap.xml*



```
    GET:https://demo.owasp-juice.shop/ftp/encrypt.pyc
    GET:https://demo.owasp-juice.shop/ftp/package-lock.json.bak
    GET:https://demo.owasp-juice.shop/ftp/package.json.bak
    GET:https://demo.owasp-juice.shop/sitemap.xml
  Cross-Domain Misconfiguration (Systemic)
    GET:https://demo.owasp-juice.shop
    GET:https://demo.owasp-juice.shop/assets/public/favicon_js.ico
    GET:https://demo.owasp-juice.shop/polyfills.js
    GET:https://demo.owasp-juice.shop/robots.txt
    GET:https://demo.owasp-juice.shop/sitemap.xml
  Cross-Domain JavaScript Source File Inclusion (Systemic)
    GET:https://demo.owasp-juice.shop
    GET:https://demo.owasp-juice.shop
    GET:https://demo.owasp-juice.shop/app/node_modules/express/lib/router/vendor.js
    GET:https://demo.owasp-juice.shop/sitemap.xml
```

Cross-Domain Misconfiguration

URL: https://demo.owasp-juice.shop/robots.txt
Risk: Medium
Confidence: Medium
Parameter:
Attack:
Evidence: Access-Control-Allow-Origin: *
CWE ID: 264
WASC ID: 14
Source: Passive (10098 - Cross-Domain Misconfiguration)
Input Vector:

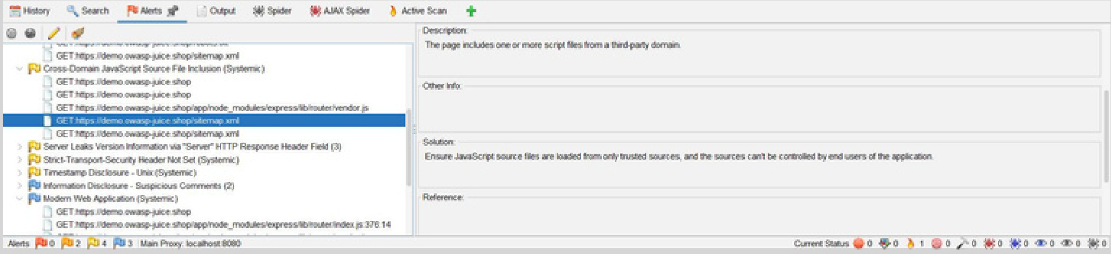
Description:

Web browser data loading may be possible, due to a Cross Origin Resource Sharing (CORS) misconfiguration on the web server.

Other Info:

Alerts 0 2 4 3 Main Proxy: localhost:8080

Current Status 0 0 1 0 0 0 0 0 1 0



GET https://demo.owasp-juice.shop/sitemap.xml
Cross-Domain JavaScript Source File Inclusion (Systemic)
GET https://demo.owasp-juice.shop
GET https://demo.owasp-juice.shop
GET https://demo.owasp-juice.shop/app/node_modules/express/lib/router/vendor.js
GET https://demo.owasp-juice.shop/sitemap.xml
GET https://demo.owasp-juice.shop/sitemap.xml
Server Leaks Version Information via "Server" HTTP Response Header Field (3)
Strict-Transport-Security Header Not Set (Systemic)
Timestamp Disclosure - Unix (Systemic)
Information Disclosure - Suspicious Comments (2)
Modern Web Application (Systemic)
GET https://demo.owasp-juice.shop
GET https://demo.owasp-juice.shop/app/node_modules/express/lib/router/index.js:376:14

Description:
The page includes one or more script files from a third-party domain.

Other Info:

Solution:
Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application.

Reference:

Alerts 0 2 4 3 Main Proxy: localhost:8080

Current Status 0 0 1 0 0 0 0 0 0 0

5. OWASP Top 10 Mapping Summary

Vulnerability	OWASP Top 10
DirectoryBrowsing	A05–SecurityMisconfiguration
SensitiveFileExposure	A05–SecurityMisconfiguration
Missing Security Headers	A05 – Security Misconfiguration
InformationDisclosure	A01–BrokenAccessControl

6.Conclusion

The vulnerability assessment identified multiple medium-risk security issues primarily related to security misconfiguration and information disclosure. Although the application is intentionally vulnerable for learning purposes, similar issues in real-world applications could lead to serious security breaches.

Implementing proper access controls, removing exposed files, and applying recommended security headers would significantly improve the application's security posture.

7. Disclaimer

This assessment was conducted solely for educational purposes on an intentionally vulnerable application. No unauthorized testing was performed on real-world production systems.

THANK YOU