WEB VULNERABILITY ASSESSMENT REPORT

Target :

OWASP Juice Shop

Intern :

ABAKTA Haana Camille

Organization :

Future Interns

Assignment :

Task 1 - Web Application Security Testing

Date :

January 5, 2026

SUMMARY

1. Introduction

The objective of this task was to perform a comprehensive vulnerability assessment of the **OWASP Juice Shop** web application. This report outlines the security flaws identified during the testing phase, evaluates their potential impact, and provides actionable remediation steps to secure the application.

2. Test Environment & Methodology

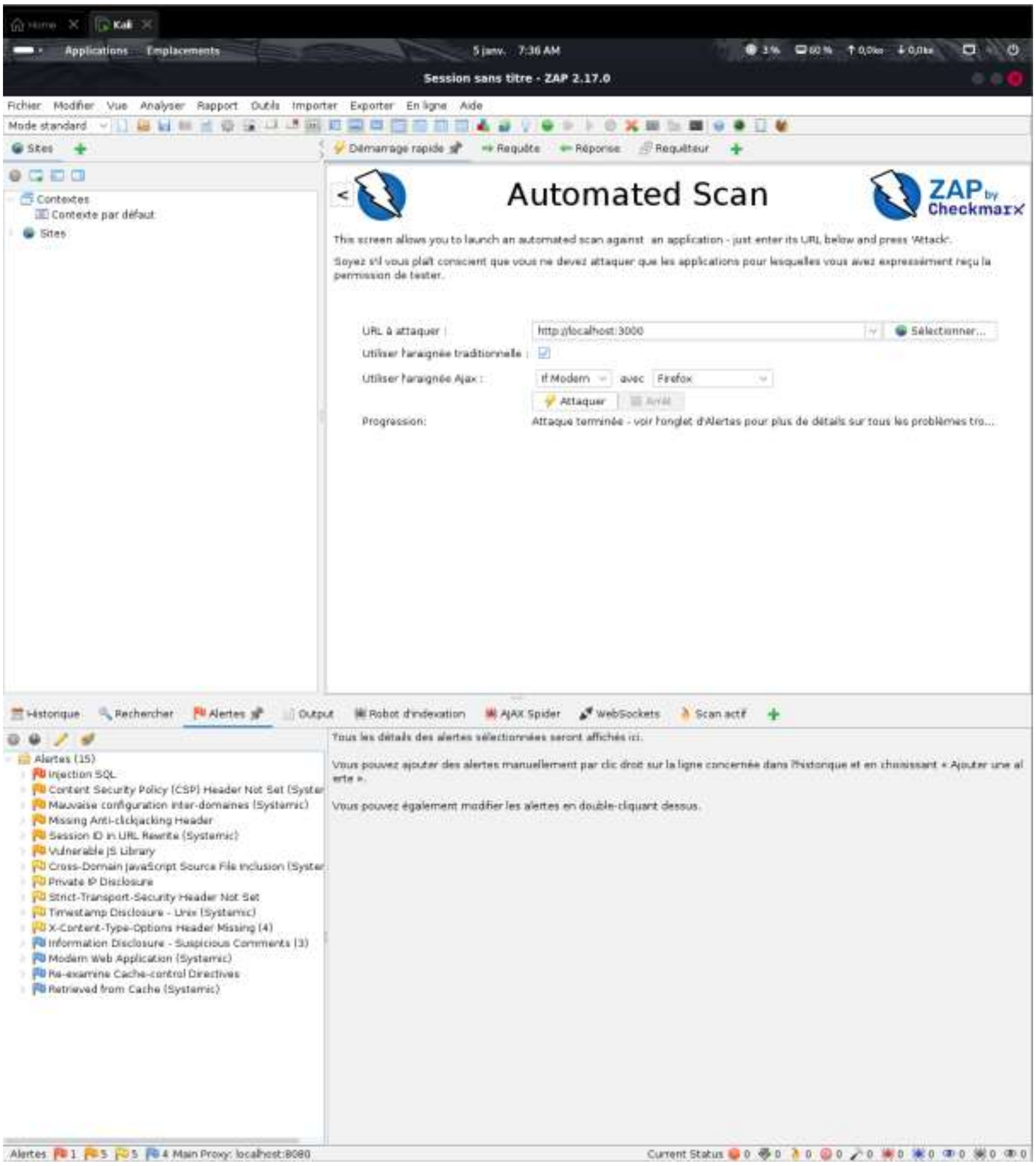
To ensure a professional and isolated testing environment, the following setup was utilized:

- **Operating System:** Kali Linux (running on VMware Workstation).
- **Primary Tool:** OWASP ZAP (Zed Attack Proxy) version 2.17.0.
- **Target Application:** OWASP Juice Shop, deployed via Docker container on http://localhost:3000.
- **Methodology:** An automated Dynamic Application Security Testing (DAST) approach was used to crawl the application and identify common vulnerabilities.

3. Executive Summary

The automated scan successfully identified **15 security alerts**, categorized by severity level:

- **High Risk:** 1 (SQL Injection)
- **Medium Risk:** 5
- **Low Risk:** 5
- **Informational:** 4


Session sans titre - ZAP 2.17.0

Automated Scan

This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.

Soyez s'il vous plaît conscient que vous ne devez attaquer que les applications pour lesquelles vous avez expressément reçu la permission de tester.

URL à attaquer : http://localhost:3000
Utiliser l'araignée traditionnelle : ✓
Utiliser l'araignée Ajax : If Modern avec Firefox
Progression: Attaque terminée - voir l'onglet d'Alertes pour plus de détails sur tous les problèmes tro...

Alertes (15)
- Injection SQL
- Content Security Policy (CSP) Header Not Set (Syster
- Mauvaise configuration inter-domaines (Systemic)
- Missing Anti-clickjacking Header
- Session ID in URL Rewrite (Systemic)
- Vulnerable JS Library
- Cross-Domain JavaScript Source File Inclusion (Syster
- Private IP Disclosure
- Strict-Transport-Security Header Not Set
- Timestamp Disclosure - Unix (Systemic)
- X-Content-Type-Options Header Missing (4)
- Information Disclosure - Suspicious Comments (3)
- Modern Web Application (Systemic)
- Re-examine Cache-control Directives
- Retrieved from Cache (Systemic)

Tous les détails des alertes sélectionnées seront affichés ici.

Vous pouvez ajouter des alertes manuellement par clic droit sur la ligne concernée dans l'historique et en choisissant « Ajouter une al erte ».

Vous pouvez également modifier les alertes en double-cliquant dessus.

Alertes 1 5 5 4 Main Proxy: localhost:8080

Figure 1 : Overview of vulnerabilities detected by OWASP ZAP

4. Detailed Vulnerability Analysis

4.1. SQL Injection (SQLi)

- **Severity:** High
- **Description:** The application fails to properly sanitize user input before including it in a database query.

- **Impact:** An attacker could bypass authentication, access sensitive user data (emails, passwords), or modify/delete database records.
- **Remediation:** Implement **Parameterized Queries** (Prepared Statements) and ensure strict input validation on the server side.

4.2. Vulnerable JS Library

- **Severity:** Medium
- **Description:** The application uses outdated versions of JavaScript libraries (e.g., jQuery) that contain known vulnerabilities (CVEs).
- **Impact:** This could lead to Cross-Site Scripting (XSS) or other client-side attacks, potentially compromising user sessions.
- **Remediation:** Update all dependencies to their latest stable versions using npm update and regularly audit packages with npm audit.

4.3. Missing Content Security Policy (CSP) Header

- **Severity:** Medium
- **Description:** The server does not provide a CSP header. CSP is a security layer that helps detect and mitigate certain types of attacks, including XSS.
- **Impact:** Without a CSP, the application is significantly more vulnerable to malicious script injections.
- **Remediation:** Configure the web server to return a Content-Security-Policy header that restricts where scripts can be loaded from.

5. OWASP Top 10 Checklist Mapping

Vulnerability Found	OWASP Top 10 Category	Description
SQL Injection	A03:2021 – Injection	Occurs when untrusted data is sent to an interpreter as part of a command or query.
Vulnerable JS Library	A06:2021 – Vulnerable and Outdated Components	Using components that are unsupported or out of date, exposing the app to known exploits.
Missing CSP / Security Headers	A05:2021 – Security Misconfiguration	Failure to implement secure configurations, such as missing security headers or default settings.
Information Disclosure	A01:2021 – Broken Access Control	(If you found sensitive data) Improper restriction of what users can see or access.
Cross-Domain JS Inclusion	A08:2021 – Software and Data Integrity Failures	Risks associated with using plugins/libraries from untrusted sources without verification.

6. Conclusion

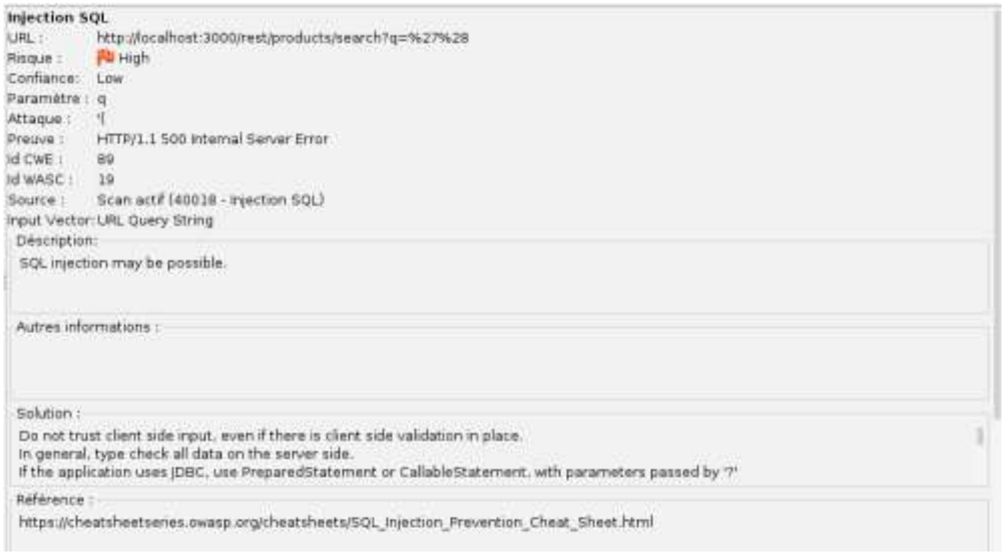
The assessment revealed critical vulnerabilities, most notably a **SQL Injection**, which poses a significant threat to data integrity and confidentiality. It is highly recommended to prioritize the remediation of High and Medium risk findings to improve the overall security posture of the application.

7. Technical Evidence (Screenshots)

This section provides visual evidence of the vulnerabilities identified during the automated scan and manual exploration of the OWASP Juice Shop application.

7.1. High Severity: SQL Injection Proof

This screenshot demonstrates the detection of a SQL Injection vulnerability. OWASP ZAP successfully injected a malicious payload to confirm that the application does not properly sanitize database queries.

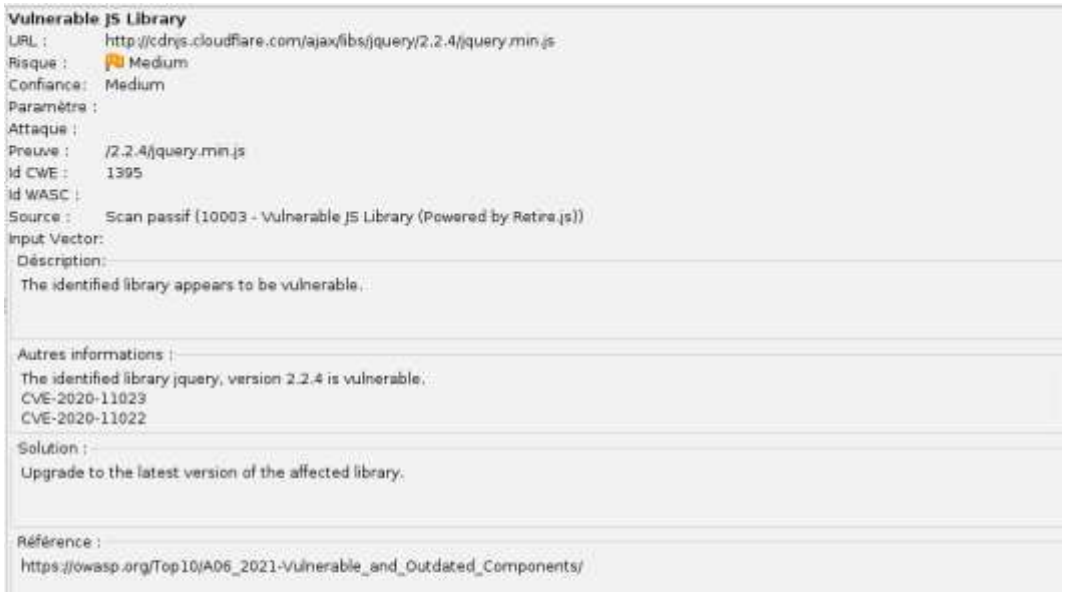


```
Injection SQL
URL :         http://localhost:3000/rest/products/search?q=%27%28
Risque :      High
Confiance:    Low
Paramètre :   q
Attaque :     '(
Preuve :      HTTP/1.1 500 Internal Server Error
Id CWE :      89
Id WASC :     19
Source :      Scan actif (40018 - Injection SQL)
Input Vector: URL Query String
Description:
SQL injection may be possible.
Autres informations :
Solution :
Do not trust client side input, even if there is client side validation in place.
In general, type check all data on the server side.
If the application uses JDBC, use PreparedStatement or CallableStatement, with parameters passed by '?'
Référence :
https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html
```

Description: OWASP ZAP Alert panel showing the "Evidence" and the specific URL parameter affected by the SQL Injection.

7.2. Medium Severity: Vulnerable JavaScript Libraries

The following evidence shows that the application relies on outdated third-party libraries. These libraries have documented vulnerabilities (CVEs) that could be exploited to compromise the client-side security.

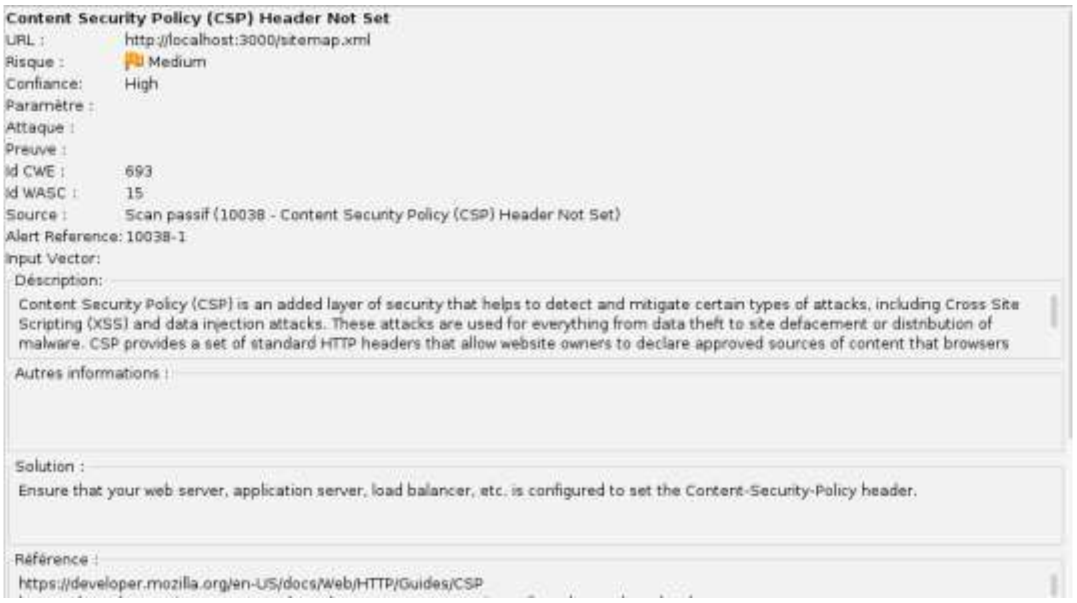


```
Vulnerable JS Library
URL :         http://cdnjs.cloudflare.com/ajax/libs/jquery/2.2.4/jquery.min.js
Risque :      Medium
Confiance:    Medium
Paramètre :
Attaque :
Preuve :      /2.2.4/jquery.min.js
Id CWE :      1395
Id WASC :
Source :      Scan passif (10003 - Vulnerable JS Library (Powered by Retire.js))
Input Vector:
Description:
The identified library appears to be vulnerable.
Autres informations :
The identified library jquery, version 2.2.4 is vulnerable.
CVE-2020-11023
CVE-2020-11022
Solution :
Upgrade to the latest version of the affected library.
Référence :
https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/
```

Description: Detailed view of the "Vulnerable JS Library" alert, displaying the list of identified CVEs (Common Vulnerabilities and Exposures) associated with the libraries used.

7.3. Security Misconfiguration: Missing Security Headers

The HTTP response headers analyzed by ZAP reveal a lack of modern security protections. The absence of Content-Security-Policy and X-Frame-Options makes the application susceptible to XSS and Clickjacking attacks.



Content Security Policy (CSP) Header Not Set

URL :	http://localhost:3000/sitemap.xml
Risque :	Medium
Confiance:	High
Paramètre :	
Attaque :	
Preuve :	
Id CWE :	693
Id WASC :	15
Source :	Scan passif (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

Autres informations :

Solution :

Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

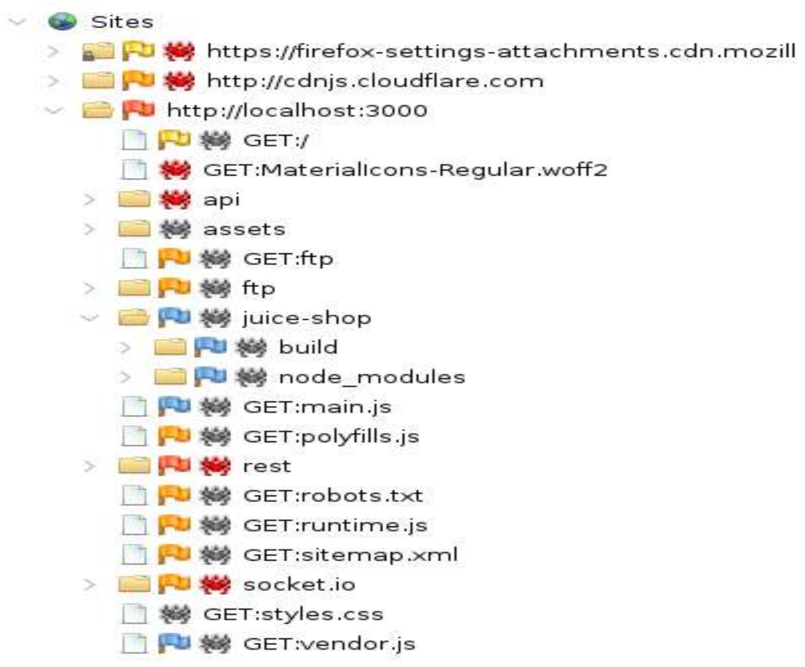
Référence :

https://developer.mozilla.org/en-US/docs/Web/HTTP/Guides/CSP

Description: Comparison of the Header tab in ZAP, showing the missing security flags in the HTTP response from the server.

7.4. Site Mapping and Crawling (Spider Results)

To ensure full coverage of the assessment, the ZAP Spider was used to map the entire directory structure of the application. This screenshot shows the "Site Tree" generated during the process.



```
Sites
  https://firefox-settings-attachments.cdn.mozill
  http://cdnjs.cloudflare.com
  http://localhost:3000
    GET:/
    GET:MaterialIcons-Regular.woff2
    api
    assets
    GET:ftp
    ftp
    juice-shop
      build
      node_modules
    GET:main.js
    GET:polyfills.js
    rest
    GET:robots.txt
    GET:runtime.js
    GET:sitemap.xml
    socket.io
    GET:styles.css
    GET:vendor.js
```

Description: The left-hand "Sites" panel in OWASP ZAP showing the fully expanded directory structure of http://localhost:3000.