Web Application Penetration Testing Report Template

Name: [Alexander Focsha]
Date: [06/27/2025]
Role: Junior Penetration Tester

Executive Summary

Scope of the Test The penetration test focused on assessing the OWASP Juice Shop, a deliberately insecure e-commerce application. The primary objectives were to identify vulnerabilities that could compromise customer data or disrupt business operations, utilizing both manual and automated testing methods.

Summary of Vulnerabilities Identified During the assessment, several critical vulnerabilities were identified, including:

- SQL Injection in the product search functionality
- Cross-Site Scripting (XSS) in the product review form
- Cross-Site Request Forgery (CSRF) on sensitive actions
- Authentication issues through fuzz payloads

Potential Business or User Impact Exploitation of these vulnerabilities could lead to unauthorized data access, user impersonation, data manipulation, and potential financial losses. The integrity and confidentiality of customer information are at significant risk.

High-Level Overview of Remediation Recommendations To mitigate these vulnerabilities, it is recommended to implement input validation, use prepared statements for database queries, enforce CSRF tokens, and enhance authentication mechanisms. Regular security assessments should also be scheduled to maintain application security.

Testing Methodology

Reconnaissance The application was mapped through manual exploration and automated tools, identifying critical pages, input fields, and authentication flows. This involved exploring various user roles and functionalities within the Juice Shop.

Tools Used

- **Burp Suite**: For intercepting and modifying HTTP requests.
- **OWASP ZAP**: For automated vulnerability scanning.

- **Browser Developer Tools**: For manual inspection of form inputs, headers, and responses.
- **Manual Exploration**: To identify unique vulnerabilities not covered by automated tools.

Selection Criteria Vulnerabilities were prioritized based on their potential impact on the application and ease of exploitation. Common OWASP vulnerabilities such as SQL Injection, XSS, and CSRF were specifically targeted due to their prevalence in web applications.

Testing Process The testing process involved iterative testing, where findings from initial scans were explored further, and new vulnerabilities were pursued based on initial results. Adjustments to the strategy were made based on the effectiveness of the tools and manual methods.

Ethical Considerations Responsible conduct was maintained throughout the engagement by ensuring that all testing activities were authorized, avoiding any actions that could disrupt business operations, and safeguarding customer data.

Findings & Exploitation Details

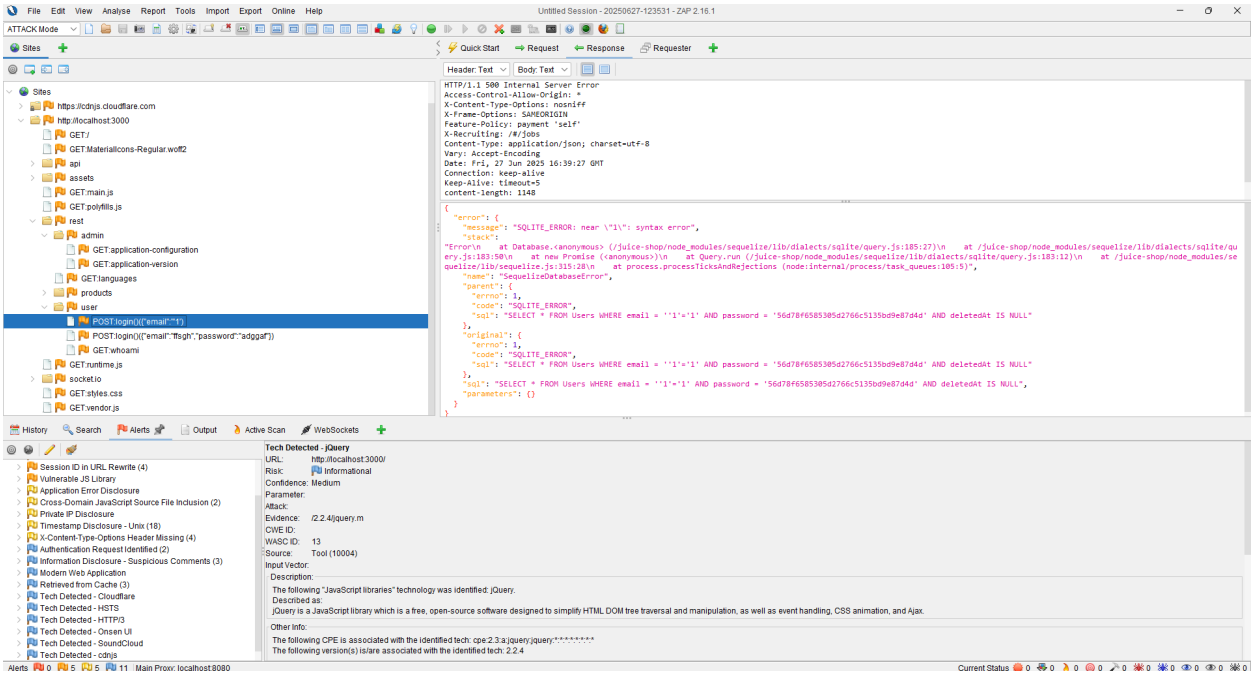
Field	Details
Vulnerability Title	SQL Injection in Product Search Functionality
Description	The application fails to properly sanitize user inputs in the product search query.
Discovery Method	Identified through manual testing and automated scans via OWASP ZAP.
Exploitation	By injecting a SQL payload, I was able to retrieve sensitive data from the database.

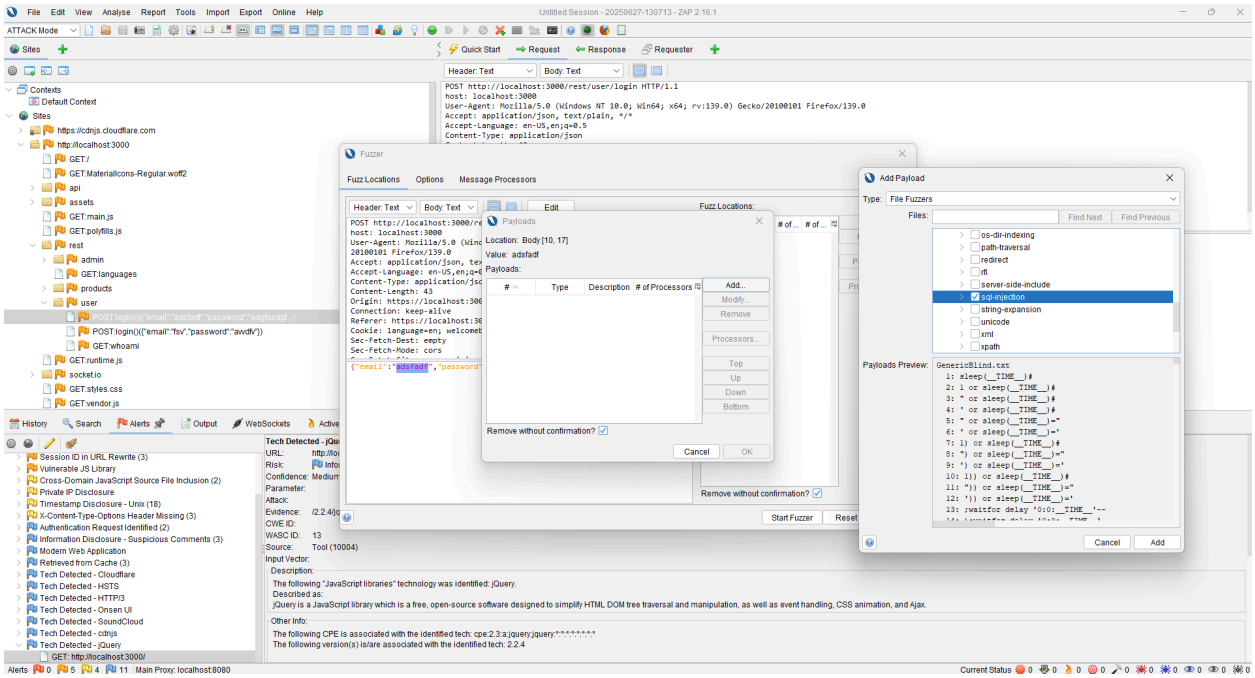
Steps*
*

Justification Exploiting this vulnerability could allow unauthorized access to all product data.

Impact A malicious actor could manipulate product data, leading to data integrity issues.

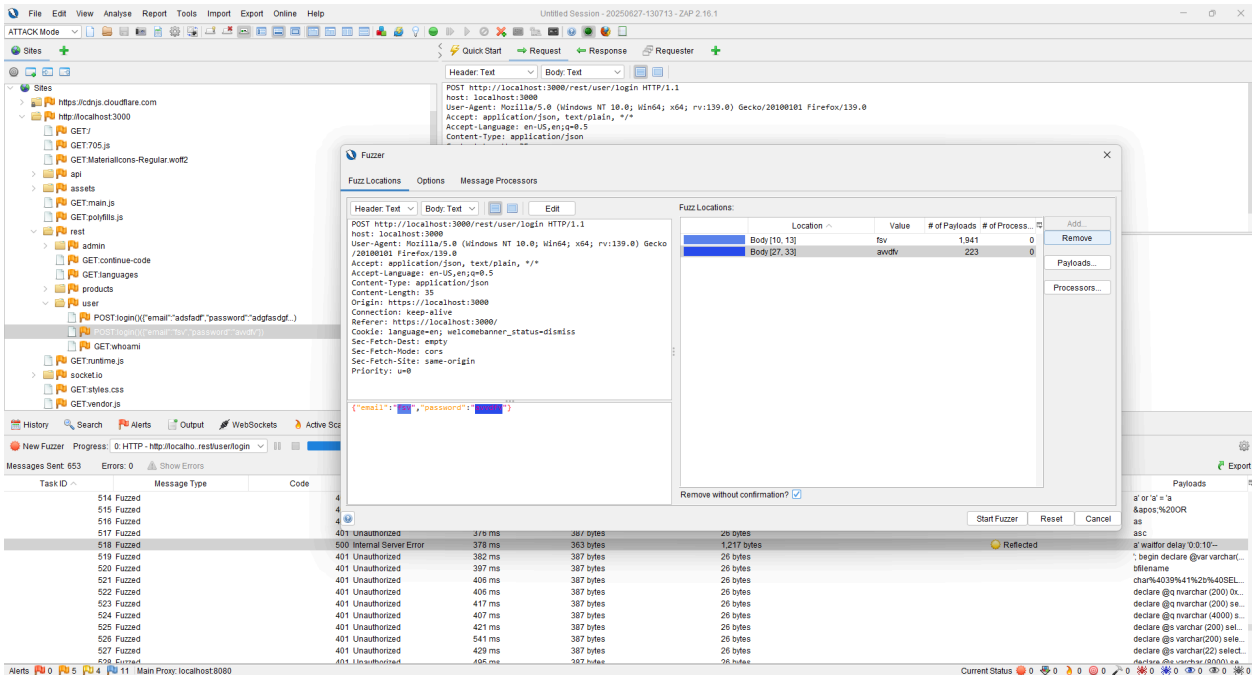
Evidence ![SQL Injection Evidence](link_to_screenshot)

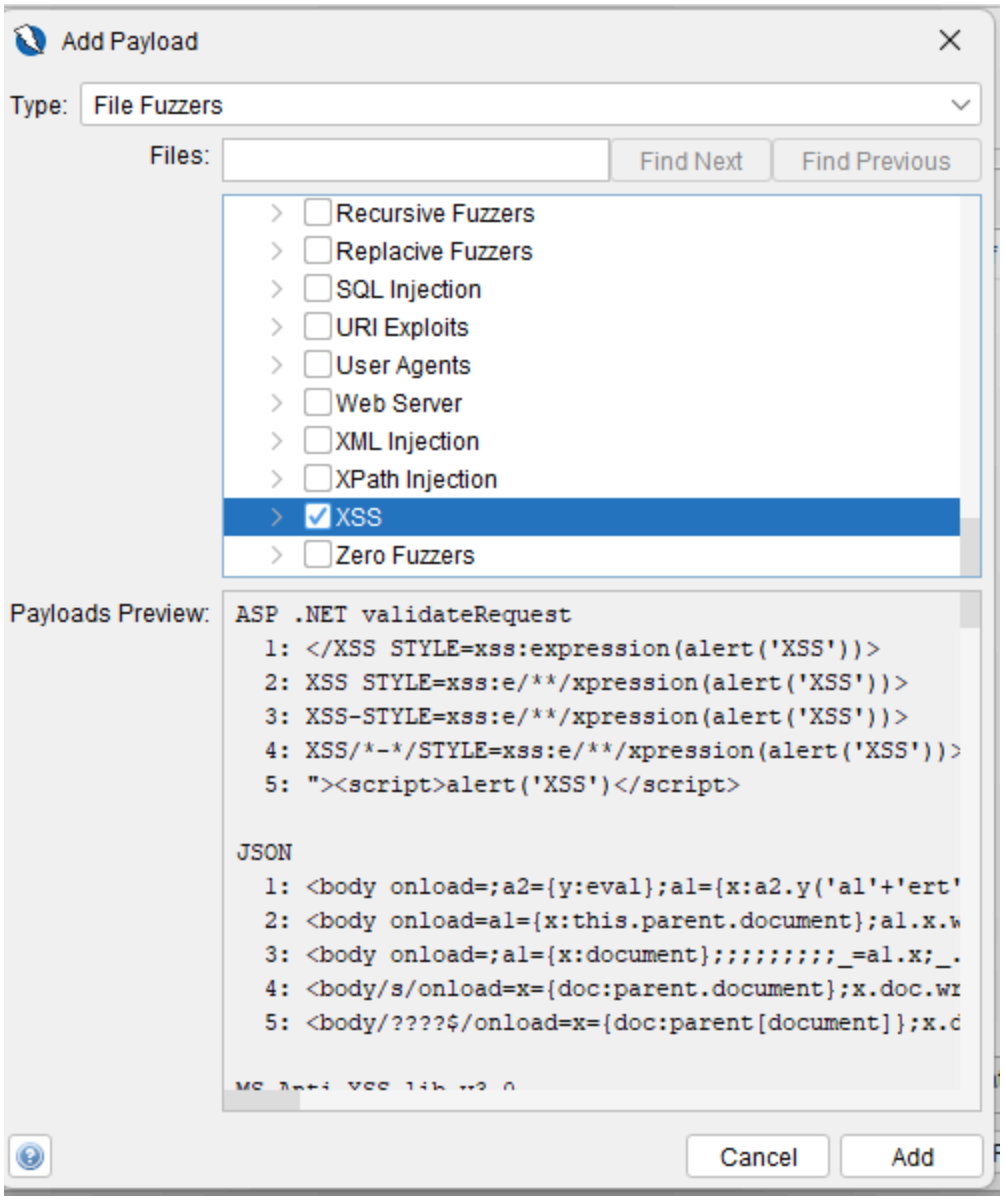




Field	Details
Vulnerability Title	Cross-Site Scripting (XSS) in Product Review Form
Description	The product review form lacks proper output encoding, allowing for script injection.
Discovery Method	Detected through manual exploration of the review submission functionality.
Exploitation Steps	Submitted a crafted review containing a script tag, which executed upon page render.

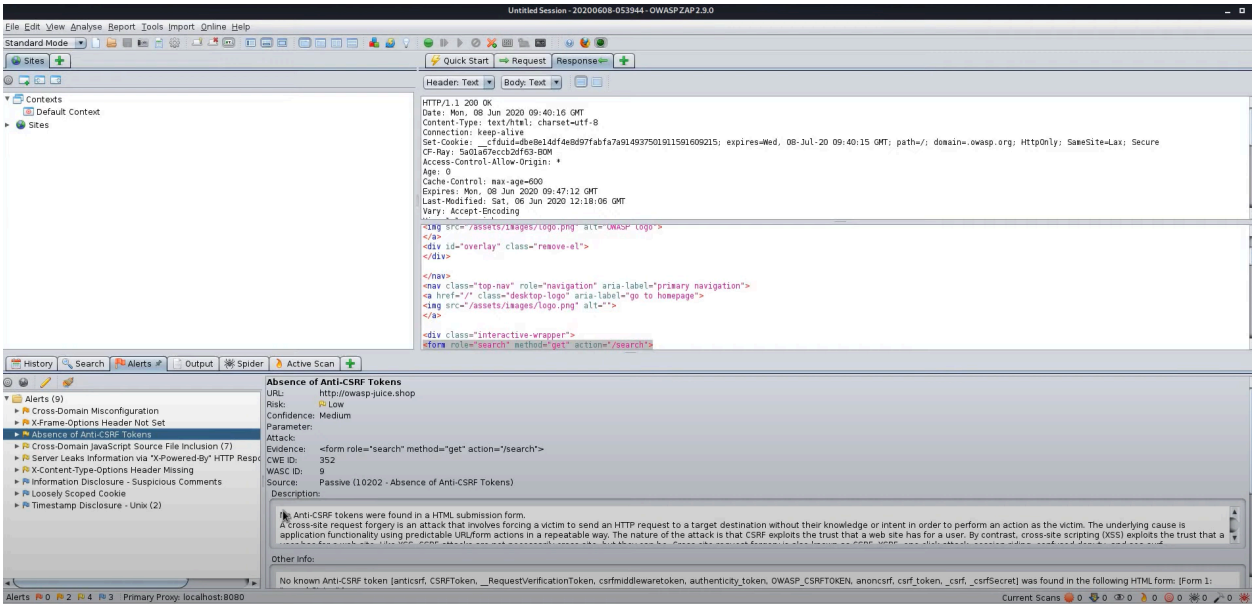
Justification	XSS can lead to session hijacking and spreading malware to other users.
Impact	A malicious actor could execute scripts in the context of a user's session.
Evidence	![XSS Evidence](link_to_screenshot)





Field	Details
Vulnerability Title	Cross-Site Request Forgery (CSRF) on Sensitive Actions

Description	The application does not implement CSRF tokens for state-changing requests.
Discovery Method	Identified through analyzing the request patterns for sensitive actions.
Exploitation Steps	Created a malicious website that, when visited, performed unauthorized actions on behalf of the user.
Justification	CSRF can exploit user sessions without their consent, leading to unauthorized transactions.
Impact	A malicious actor could manipulate user actions without their knowledge.
Evidence	![CSRF Evidence](link_to_screenshot)



Impact Analysis

Likelihood of Exploitation The likelihood of exploitation for SQL Injection and XSS vulnerabilities is high due to the commonality of these attack vectors. CSRF also poses a significant risk, especially if users are authenticated.

Potential Impact on Business or User Data Successful exploitation could lead to unauthorized access to sensitive customer data, financial losses, and damage to the organization's reputation. The integrity and confidentiality of user data are at serious risk.

Severity Classification

- SQL Injection: Critical
- Cross-Site Scripting (XSS): High
- Cross-Site Request Forgery (CSRF): Medium

These issues matter in real-world settings as they can lead to severe consequences, including data breaches and operational disruptions.

Remediation Recommendations

SQL Injection

- Implement prepared statements and parameterized queries to prevent injection attacks.
- Validate and sanitize all user inputs on both client and server sides.

Cross-Site Scripting (XSS)

- Apply proper output encoding for all user-generated content.
- Utilize Content Security Policy (CSP) headers to mitigate XSS risks.

Cross-Site Request Forgery (CSRF)

- Implement anti-CSRF tokens for all state-changing operations.
- Enforce same-origin policies to limit cross-origin requests.

References

- OWASP SQL Injection Cheat Sheet
- OWASP XSS Prevention Cheat Sheet
- OWASP CSRF Prevention Cheat Sheet

Conclusion & Ethical Reflection

Overall Process Reflection The structured approach to testing, utilizing a combination of tools and manual techniques, proved effective in identifying vulnerabilities.

Challenges Encountered One challenge was navigating the application's complex flows while ensuring thorough testing. This was addressed through careful planning and documentation of testing steps.

Future Improvements In future engagements, I would allocate more time for exploratory testing to uncover less obvious vulnerabilities.

Ethical Boundaries Ethical considerations were maintained by ensuring all activities were conducted with the client's consent and by respecting user data privacy throughout the engagement.