

Web Penetration Testing

الطالبة : شهد احمد محمد شكري عودة الله

الطالبة : دجى مدار الله محمود الرواجفة

DOCUMENT VERSION CONTROL

Data Classification – Client Confidential

Client Name	
Project name	Web Penetration Testing
Authors	
Approved by	
Version	
Submission Date	

Table of contents

1. Executive summary
 - 1.1 Scope details
2. Detailed penetration testing report
 - 2.1 Introduction
 - 2.2 Restrictions
 - 2.3 Tools
 - 2.4 Discovered vulnerability
3. Conclusion

1: Executive Summary

This report documents the findings after testing web application. A series of tests were conducted against the targeted scope, using testing tools and where appropriate, manual testing techniques, to establish the presence of actual or potentially exploitable security vulnerabilities, which if exploited could result in direct or indirect damage to Customer Name. These key findings (classified as “**High**”, “**Medium**” or “**Low**”) have been highlighted in this report. Please refer to the Detailed Penetration Testing Results of the report for in depth details the key findings, their associated risks, and recommended actions.

The following represents the definition and the description of each severity rate.

Impact	Description
High	A vulnerability that can be exploited by the attacker and cause huge damage to application components and data.
Medium	A vulnerability that can be exploited by the attacker and cause moderate damage to application components and data.
Low	A vulnerability that can be exploited by the attacker to understand application underlying technologies and versions which can be utilized in further attacks.

1.1 Scope Details

The scope of evaluation and testing covered the following assets:

#	Host	Platform
1		OWASP Juice Shop

2: Detailed Penetration Testing Results

2.1 Introduction

Simulated e-commerce web application with login, file upload, shopping cart, and comment features

2.2 Restrictions

practical penetration test on a vulnerable web application, identify key security flaws, and document findings using standard tools and methodologies

2.3 Tools

1. Burp Suite - for intercepting and modifying HTTP requests

2. OWASP ZAP - for automated vulnerability scanning

3. curl - for manual request testing

4. Chrome Developer Tools - for inspecting DOM and client-side behavior

2.4 Discovered vulnerability

1. SQL Injection (**High**)

Location: User login endpoint

Discovery Method: Injected ' OR 1=1 --

Exploitation: Allowing login without valid credentials

Impact: Exposure of user and product data

Recommended Fix: Implement parameterized queries (Prepared Statements)

```
Pretty Raw JSON Actions ▾
1 POST /rest/user/login HTTP/1.1
2 Host: 10.10.23.164
3 Content-Length: 20
4 Accept: application/json, text/plain, */*
5 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4108.116 Safari/537.36
6 Content-Type: application/json
7 Origin: http://10.10.23.164
8 Referer: http://10.10.23.164/
9 Accept-Encoding: gzip, deflate
10 Accept-Language: en-US,en;q=0.9
11 Cookie: language=en; cookieconsent_status=dismiss; continueCode=[5v0o]0eg730xz06e8420E9HvyJr0XYAq9pv5xYWhk;2P11Ld[8K7vPbKE; iemLtz5@_xfmjicdfRpAMAF;
12 Connection: close
13
14 {
15     "email": " or 1=1--",
16     "password": "f"
17 }
```

2. Stored XSS (High)

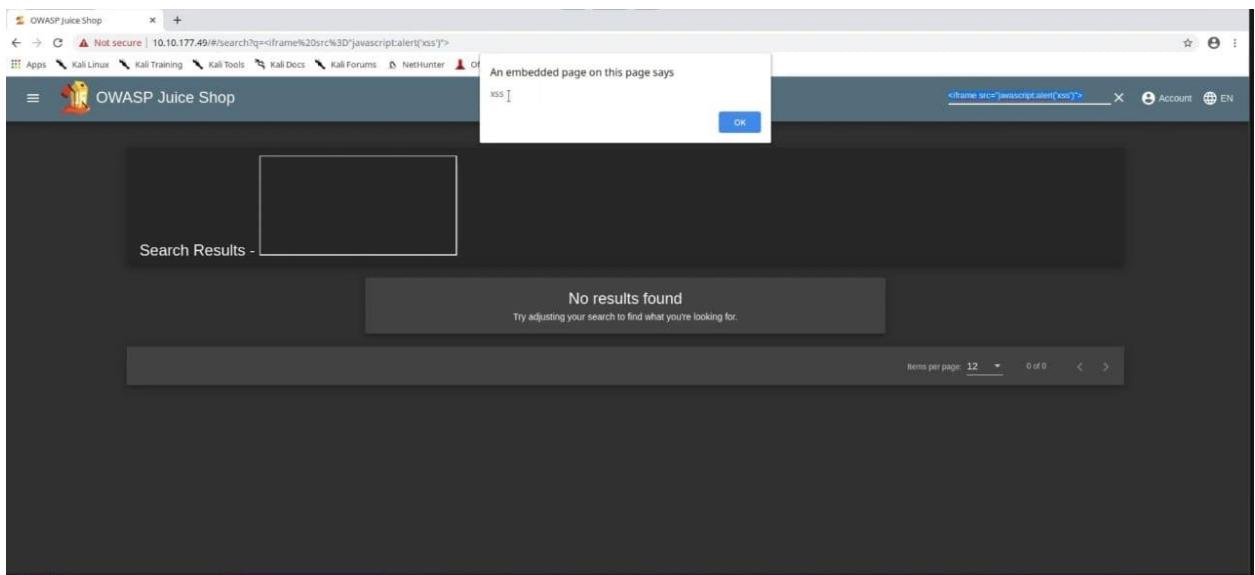
Location: Product comment section

Discovery Method: injected <iframe src="javascript:alert('xss')">

Exploitation: Execution of malicious JavaScript in users' browsers

Impact: Session hijacking and UI manipulation

Recommended Fix: Sanitize user input and encode output using functions



3. File Upload Vulnerability (High)

Location: File upload endpoint

Discovery Method: Uploaded a file containing executable code

Exploitation: Remote code execution on the server

Impact: Full system compromise

Recommended Fix: Validate file type, extension, and content before saving

4. Broken Access Control (High)

Location: Administration section

Discovery Method: Direct access to the hidden /administration endpoint without proper authorization checks

Exploitation: Unauthorized data access and modification

Impact: Violation of user privacy and data integrity

Recommended Fix: Enforce server-side authorization checks for all administrative endpoints

The screenshot shows the OWASP Juice Shop application interface. At the top, there's a navigation bar with links like 'Most Visited', 'Offensive Security', 'Kali Linux', 'Kali Docs', 'Kali Tools', 'Exploit-DB', 'Aircrack-ng', 'Kali Forums', 'NetHunter', 'Kali Training', and 'Getting Started'. Below the navigation is a header with the logo 'OWASP Juice Shop v7.5.1' and a search bar. A green banner at the top of the main content area says 'You successfully solved a challenge: Admin Section (Access the administration section of the store.)'. The main content is divided into two sections: 'Administration Registered Users' and 'Customer Feedback'. The 'Administration Registered Users' section lists user emails: admin@juice-shop, jim@juice-shop, bender@juice-shop, björn.lammelich@googlemail.com, cisco@juice-shop, support@juice-shop, morty@juice-shop, mc.saksearch@juice-shop, and test@test.com. The 'Customer Feedback' section displays three reviews with their respective ratings:

User	Comment	Rating
1	I love this shop! Best products in town! Highly recommended!	★★★★★
2	Great shop! Awesome service!	★★★★★
3	Incompetent customer support! Can't even upload photo of broken purchase! Support Team: Sorry, only order confirmation PDFs can be attached to complaints!	★★★★★
	This is the store for awesome stuff of all kinds!	★★★★★
	Never gonna buy anywhere else from now on! Thanks for the great service!	★★★★★
	Keep up the good work!	★★★★★
3	Nothing useful available here!	★★★★★

At the bottom, there's a 'Recycling Requests' section with one entry for a user named '2'.

3: Conclusion

A practical penetration test was conducted on the OWASP Juice Shop application, revealing multiple vulnerabilities across key functionalities. Each flaw was documented with exploitation techniques, and recommended mitigations.

These findings highlight the importance of secure coding practices and robust access control mechanisms in web applications.