

# **Securing the Digital Marketplace: Strengthening Security for Vulnerable Websites**

## Pentesting Project Report

Prepared by: Shaik Abdul Sameer  
Institution: IIITB Simplilearn  
Date: 11 December 2025

## **1. Introduction**

This project focuses on analyzing and exploiting security vulnerabilities in a deliberately insecure web application. The purpose of this assessment is to understand practical web security weaknesses, exploit them ethically, and document technical findings while proposing mitigation strategies. The test environment was hosted locally using OWASP Juice Shop, simulating real-world attack scenarios in a safe academic environment.

## **2. Scope of Testing**

The scope of the penetration testing included:

- User authentication workflow - API endpoints responsible for login, profile, and role authorization
- JWT-based session management
- Privilege escalation attempts through role manipulation
- IDOR and unauthorized endpoint access testing
- Endpoint fuzzing and security misconfiguration checks

## **3. Methodology**

The testing approach followed OWASP Testing Guide methodologies. The key phases included:

1. Information Gathering – Identifying exposed services and application behavior.
2. Authentication Testing – Intercepting login traffic, analyzing JWTs.
3. Authorization Testing – Modifying roles and attempting privilege escalation.
4. API Endpoint Enumeration – Testing unauthenticated and authenticated routes.
5. Vulnerability Validation – Confirming exploitability of identified issues.

## **4. Key Findings**

### **Finding 1: Weak JWT Implementation**

The JSON Web Token (JWT) used for user authentication revealed that user data, including roles, could be decoded and modified on the client side. No server-side validation ensured integrity after tampering.

### **Finding 2: Role Manipulation Vulnerability**

During Step 5.3, modifying the JWT role claim from 'customer' to 'admin' allowed access attempts to restricted endpoints. Although some endpoints returned 403, the vulnerability still indicates improper authorization controls.

### **Finding 3: Exposed API Endpoints**

By enumerating API routes, several publicly accessible endpoints were identified. Some responded with sensitive metadata, indicating insufficient access restrictions.

## **5. Technical Steps Performed**

### **5.1 Enumerating Localhost Service**

`curl -I http://127.0.0.1:3000/` was used to confirm service availability and capture response headers.

### **5.2 Intercepting Login Request**

Burp Suite Proxy intercepted the login request. The JWT token was extracted from server responses for analysis.

### **5.3 Privilege Escalation Attempt via JWT Manipulation**

The role field inside the JWT payload was modified and re-signed (unsigned for testing). Modified token was replayed to test authorization controls.

### **5.4 API Endpoint Testing**

`curl http://127.0.0.1:3000/rest/\*` endpoints were tested systematically to determine access restrictions.

## **6. Recommendations**

To mitigate the identified vulnerabilities, the following measures are strongly recommended:

- Implement server-side validation for all JWT claims.
- Use strong JWT signing algorithms and enforce signature verification.
- Enforce proper Role-Based Access Control (RBAC).
- Restrict access to sensitive endpoints with authentication and authorization checks.
- Enable rate limiting and monitoring for suspicious API activity.

## **7. Conclusion**

The security assessment successfully demonstrated multiple vulnerabilities within the test application, including weak JWT handling and authorization flaws. This project provided hands-on experience with real-world web application security issues and reinforced the importance of secure coding and robust access control mechanisms.