SECURING A VULNURABLE WEB APPLICATION

1. Acknowledgements

I would like to express my sincere gratitude to **Yenepoya University** and the faculty members for providing the opportunity to work on this industry-oriented cybersecurity project. I also acknowledge the **OWASP Foundation** for providing OWASP Juice Shop as an open-source vulnerable web application for learning and research purposes

2. Objective and Scope

Objective :

- To understand **real-world web application vulnerabilities**
- To perform **hands-on security testing**
- To map vulnerabilities with **OWASP Top 10**
- To document findings in a professional security report

Scope :

- Testing conducted only on **OWASP Juice Shop (local environment)**
- Focus on:
    - SQL Injection
    - Broken Authentication
    - Sensitive Data Exposure
    - Security Misconfiguration

3. Problem Statement

Modern web applications are frequently exposed to security threats due to poor input validation, insecure authentication, and misconfigured systems. Organizations need trained professionals who understand vulnerabilities practically, not just theoretically.

This project addresses the need for practical vulnerability analysis using a controlled, intentionally vulnerable application**.**

4. Existing Approaches

Traditional approaches include:

- Static learning (books, theory)
- Automated scanners without understanding results

Limitations:

- Lack of real attack simulation
- Poor understanding of exploitation impact

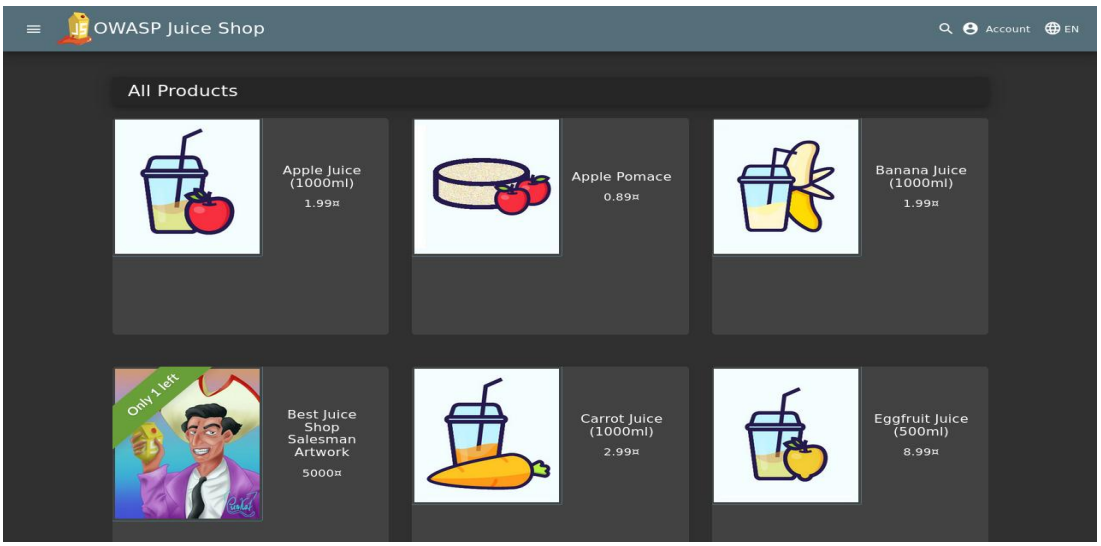
5. Approach / Methodology – Tools & Technologies Used

Tools Used

- **OWASP Juice Shop**
- **Kali Linux**
- **Docker**
- **Firefox Developer Tools**
- **VirtualBox**

Methodology

1. Environment setup
2. Application deployment
3. Vulnerability identification
4. Exploitation
5. Documentation
6. Mitigation analysis



6. Workflow

1. Install Kali Linux in VirtualBox
2. Start Docker service
3. Run OWASP Juice Shop container
4. Access application via browser
5. Perform vulnerability testing
6. Capture screenshots
7. Prepare project report.

7. Assumptions

- The application is intentionally vulnerable
- Testing is performed only on localhost
- User has basic Linux knowledge
- No automated exploitation tools used

8. Implementation-Vulnerability Analysis

Vulnerability 1: SQL Injection

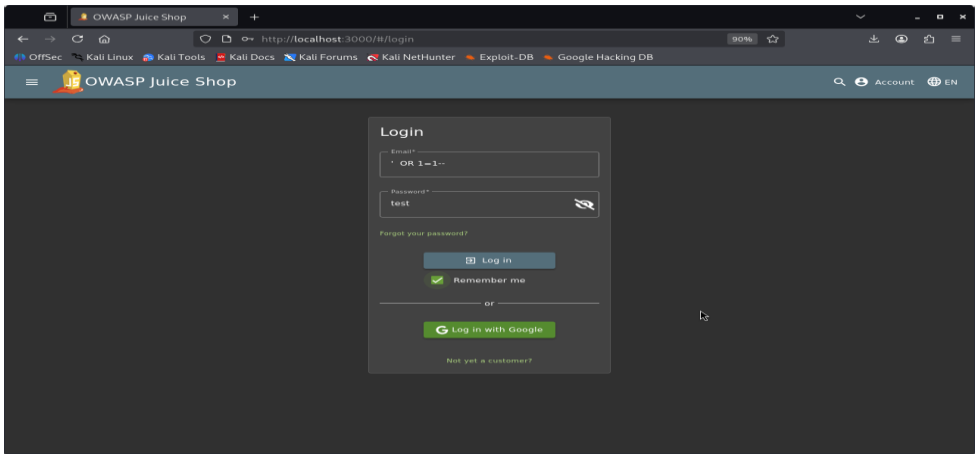
OWASP Category: A03 – Injection

Description: The login page does not properly validate user input, allowing SQL injection
Steps to Reproduce:

- Navigate to Login page

Enter email as: ‘ OR 1=1--

- Enter any password
- Click Login



Impact:
Unauthorized access without valid credentials

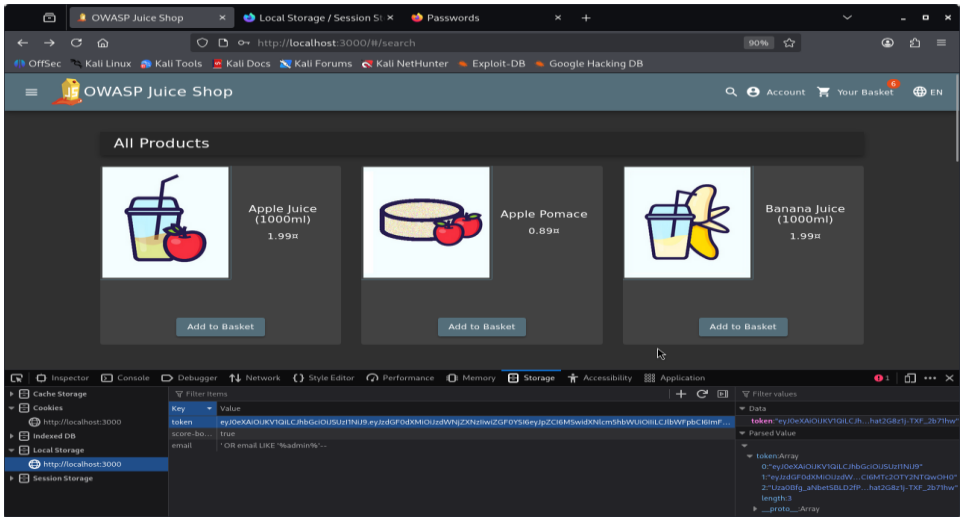
Severity: High

Vulnerability 2: Broken Authentication

OWASP Category: A07 – Identification & Authentication Failures

Description:
Session tokens are stored insecurely in browser storage.

Impact:
Attackers can hijack sessions.

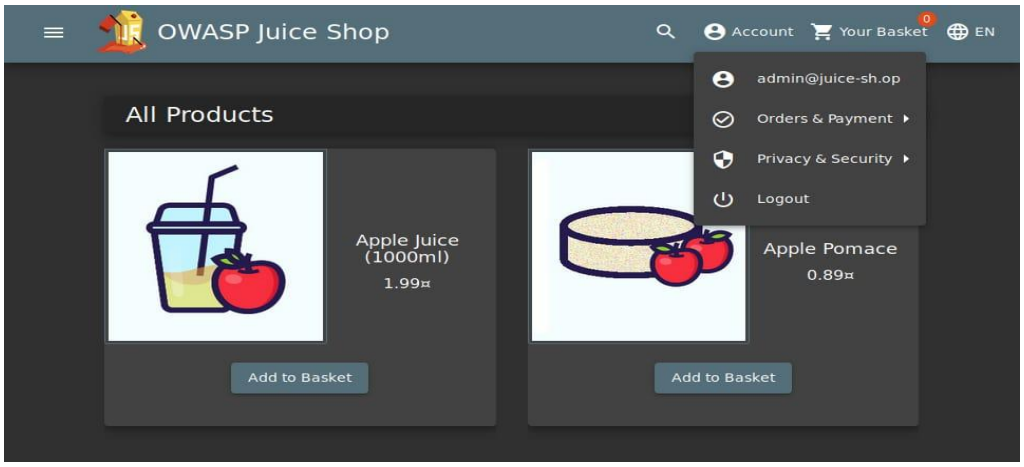


Vulnerability 3: Sensitive Data Exposure

OWASP Category: A02 – Cryptographic Failures

Description: Sensitive data is exposed through client-side storage.

Impact: User data leakage



9. **Solution Design**

1. Use prepared statements
2. Secure authentication mechanisms
3. Avoid storing sensitive data on client side
4. Implement HTTPS & secure cookies.

10. **Challenges & Opportunities**

Challenges

- Understanding real exploitation logic
- Initial setup issues
- Debugging Docker & browser tools

Opportunities

- Strong cybersecurity foundation
- Industry-relevant skills
- Confidence in vulnerability testing

11. **Reflections on the Project**

This project helped me understand how small coding mistakes can lead to serious security breaches. Practical exposure improved my confidence and analytical skills.

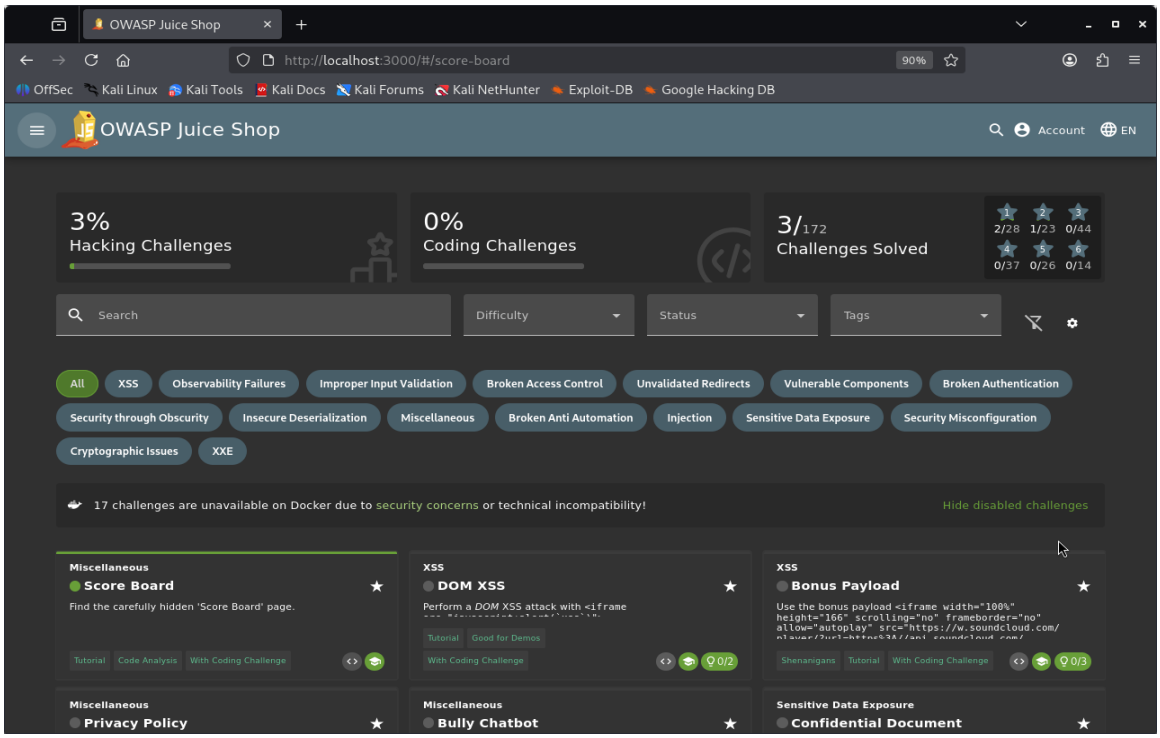
.

12. **Recommendations**

- Regular security testing
- Developer security training
- Secure coding practices
- Follow OWASP Top 10 guidelines

13. **Outcome / Result**

- Successfully deployed OWASP Juice Shop
- Identified and exploited multiple vulnerabilities
- Understood real-world security flaws
- Created professional documentation



14. **Enhancement Scope**

- Automate testing with tools like Burp Suite
- Cover advanced OWASP challenges
- Perform secure code review

15. **Link to Code & Screenshots**

GitHub Repository:

16. **References**

- OWASP Top 10
- OWASP Juice Shop Documentation
- Kali Linux Official Docs