Intern Name: Engr. Nida Kanwal
Project: OWASP Juice Shop – Security Enhancement

Week 1: Security Assessment

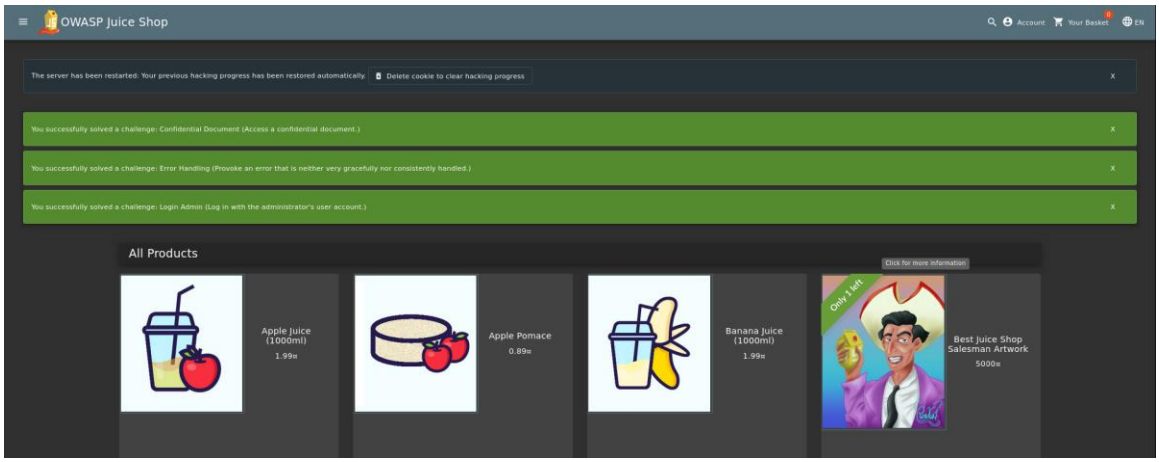
1. Application Setup

- Cloned and installed Juice Shop from GitHub

- Ran `npm install` and `npm start`

- Accessed app via `http://localhost:3000`

- Explored key pages: Signup, Login, Profile



2. Basic Vulnerability Assessment

Tools Used:

- OWASP ZAP

- Browser Developer Tools

- SQLite (for password storage verification)

Tests Performed:

- XSS Attack: Input `<script>alert('XSS');</script>` triggered alert popup ✅

- SQL Injection: Input `' OR '1'='1` successfully bypassed login ✅

- Weak Password Storage: Found MD5 hash for user `admin@juice-sh.op` = `0192023a7bbd73250516f069df18b500` → Decoded to 'admin'

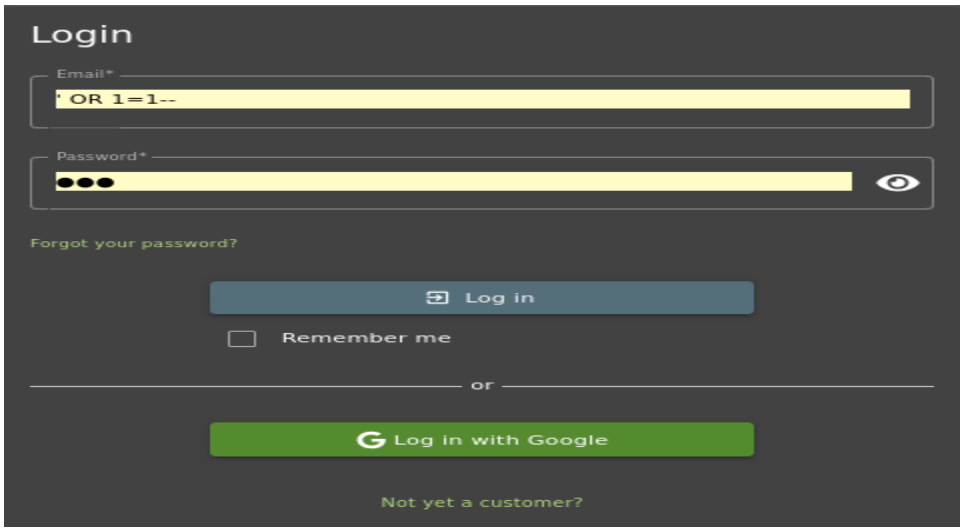
3. Findings & Screenshots

- XSS Alert Screenshot

- **Method:** Entered `<script>alert('XSS');</script>` in input fields using browser developer tools.
- **Result:** ✅ Alert box appeared. This confirms a stored or reflected XSS vulnerability.

- SQL Injection Success Screenshot

- **Method:** Entered `' OR '1'='1` in both username and password fields on the login page.
- **Result:** ✅ Successfully bypassed login. This confirms the site is vulnerable to SQL Injection.



- MD5 Password Hash Screenshot

Method: Used `sqlite3` to access `juiceshop.sqlite` database, then ran:

SELECT email, password FROM Users LIMIT 10;

Result: ✅ Found MD5 hashed passwords like `0192023a7bbd73250516f069df18b500`
→ Reversed to "admin" using an online MD5 decoder.
→ Confirms **weak password storage** (MD5 is insecure and outdated).

OWASP ZAP Scan

- **Method:** Ran a full ZAP scan on http://localhost:3000
- **Result:**
  - **High Alert:** SQL Injection
  - **Medium Alerts:**
    - Content Security Policy (CSP) Header Not Set
    - Session ID in URL
    - Cookie Without Secure Flag
    - XSS Protection Header Missing

- **Low Alerts:**
  - Cross-Domain JavaScript Source File Inclusion
  - Private IP Address Disclosure
- **Informational:** Server header disclosure, missing X-Frame options

Here's a clear table format for the **OWASP ZAP scan alerts** :

OWASP ZAP Scan Alerts

Alert Level	Issue Description	Risk Description
High	**SQL Injection**	Allows attacker to modify database queries and access data.
Medium	Content Security Policy (CSP) Header Not Set	Makes site vulnerable to XSS and data injection attacks.
Medium	Session ID in URL	Exposes session in URL, increasing risk of session hijacking.
Medium	Cookie Without Secure Flag	Cookies can be sent over unencrypted connections.
Medium	XSS Protection Header Missing	Browser won't block reflected XSS attacks without this header.
Low	Cross-Domain JavaScript Source File Inclusion	Loads JS from other domains, which can be risky.
Low	Private IP Address Disclosure	Reveals internal IPs that should be hidden.
Informational	Server Header Disclosure	Reveals server technology (e.g. Express, Node.js).
Informational	Missing X-Frame Options Header	Site can be embedded in iframes, leading to clickjacking risks.

4. Areas of Improvement

- Add Content Security Policy (CSP) header

- Use bcrypt or Argon2 instead of MD5 for password hashing

- Avoid exposing sensitive data via URLs

- Sanitize all user inputs to prevent XSS/SQLi

- Configure proper session security