Udin – DevSecOps Team

Sobelow Static Analysis:
Potion Shop

Prepared by: John Ghaly

Submitted to: Mr. Peter Shoukry

Date: 17/8/2024

Introduction

This report presents the results of a static code analysis conducted using Sobelow on the Phoenix application ‘potion_shop’. This application is intentionally designed to be vulnerable to serve as a learning tool. It has been forked to our organization's GitHub repositories and was used to explore and address common security vulnerabilities in Phoenix applications while utilizing the Sobelow tool. The analysis aimed to identify potential vulnerabilities and security misconfigurations within the application.

Commands Used

The following commands were used to perform the analysis, after installing the sobelow tool:

1. **mix sobelow**

   Provides detailed findings, including the location and type of the vulnerability, along with the corresponding confidence level indicating the likelihood that it is indeed a vulnerability.

2. **mix sobelow -v verbose**

   It prints out the code, highlighting the part that contains the vulnerability for easy identification.

Findings

1. **Missing CSRF Protections**

- File: d:/lib/carafe_web/router.ex
- Confidence level: High
- Pipeline: browser_auth
- Line: 16
- Description: The browser_auth pipeline is missing CSRF protection. This can make your application vulnerable to CSRF attacks where malicious requests could be sent on behalf of authenticated users.

2. **CSRF via Action Reuse**

- File: d:/lib/carafe_web/router.ex
- Confidence level: High
- Action: edit_bio
- Line: 98
- Description: The route /users/settings/edit_bio is defined for both GET and POST requests, which can be exploited if CSRF protection is not in place.

3. **Missing Content-Security-Policy (CSP)**

- File: d:/lib/carafe_web/router.ex

- Confidence level: High
- Pipeline: browser_auth
- Line: 21
- Description: The browser_auth pipeline lacks a Content-Security-Policy header, which is crucial for mitigating XSS attacks and controlling which resources can be loaded by the browser.

- File: d:/lib/carafe_web/router.ex
- Confidence level: High
- Pipeline: browser
- Line: 12
- Description: Similar to browser_auth, the browser pipeline also does not set a Content-Security-Policy header.

4. **HTTPS Not Enabled**

- Confidence level: High
- Description: The production configuration does not enforce HTTPS, which is essential for securing data in transit.

5. **SQL Injection**

- File: d:/lib/carafe/potions.ex
- Confidence level: Low
- Line: 20
- Function: search_potions/1

- Variable: q
- Description: The SQL query uses string interpolation, which can lead to SQL injection if user input is not properly handled.

6. **XSS (Cross-Site Scripting)**

- File: d:/lib/carafe_web/templates/potion/show.html.heex
- Confidence level: Low
- Line: 19
- Variable: review
- Description: The use of the raw function to render HTML content from review.body could expose the application to XSS attacks if the content is not sanitized.

Conclusion

This Sobelow analysis has highlighted several critical and minor security vulnerabilities within the potion_shop application. Addressing these issues will not only improve the security posture of the application but also provide valuable learning experiences in the field of application security. Implementing countermeasures to these vulnerabilities will help protect against common threats and ensure that best practices are followed, this is crucial for building robust and secure web applications.