****

# Practices for Secure Software Report

Table of Contents

[Document Revision History 3](#_Toc102040754)

[Client 3](#_Toc102040755)

[Instructions 3](#_Toc102040756)

[Developer 4](#_Toc102040757)

[1. Algorithm Cipher 4](#_Toc102040758)

[2. Certificate Generation 4](#_Toc102040759)

[3. Deploy Cipher 4](#_Toc102040760)

[4. Secure Communications 4](#_Toc102040761)

[5. Secondary Testing 4](#_Toc102040762)

[6. Functional Testing 4](#_Toc102040763)

[7. Summary 4](#_Toc102040764)

[8. Industry Standard Best Practices 4](#_Toc102040765)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **06/22/2025** | **Eva Macaluso** |  |

## Client



## Instructions

Submit this completed practices for secure software report. Replace the bracketed text with the relevant information. You must document your process for writing secure communications and refactoring code that complies with software security testing protocols.

* Respond to the steps outlined below and include your findings.
* Respond using your own words. You may also choose to include images or supporting materials. If you include them, make certain to insert them in all the relevant locations in the document.
* Refer to the Project Two Guidelines and Rubric for more detailed instructions about each section of the template.

## Developer

Eva Macaluso

## Algorithm Cipher

I recommend using the SHA-256 (Secure Hash Algorithm 256-bit) cryptographic function. It is secure and designed to avoid collisions. It is used by government agencies, financial institutions, and major tech companies due to its reliability and strength. SHA-256 works by creating a unique code called a checksum for each file or message. This helps make sure nothing has been changed or manipulated.

Avoiding collisions is important because two different files can create the same hash value. SHA-256 is a great choice because it would be hard for anyone to create two different files with the same code. This will stop hackers from tricking people with fake files. If a hacker were to make a fake file, it could lead to data breaches and stolen information. SHA-256 is stronger than older ones such as MD5, which can be easily broken into.

## Certificate Generation

Insert a screenshot below of the CER file.

A screenshot of a computer program

AI-generated content may be incorrect.

A computer screen with white text

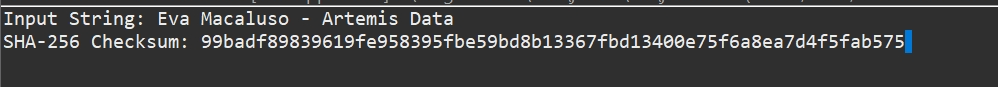
AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

## Deploy Cipher

Insert a screenshot below of the checksum verification.



## Secure Communications

Insert a screenshot below of the web browser that shows a secure webpage.

A screen shot of a computer

AI-generated content may be incorrect.

## Secondary Testing

Insert screenshots below of the refactored code executed without errors and the dependency-check report.

A black background with white lines

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Functional Testing

Insert a screenshot below of the refactored code executed without errors.

A screenshot of a computer

AI-generated content may be incorrect.

## Summary

For this project, I refactored a Spring Boot application to make it more secure. I followed the steps from the Vulnerability Assessment Process Flow, mainly focusing on cryptography by using SHA-256 for checksums and setting up HTTPS to secure client-server communication. I also ran a static test using OWASP Dependency Check to make sure I didn’t add any new issues. Overall, I went through the whole process from reviewing the architecture to testing the final version to make sure the app followed secure coding practices.

## Industry Standard Best Practices

I used HTTPS and SHA-256 to keep the app secure and avoid weak methods like MD5. I also ran the OWASP Dependency Check to catch any issues in the libraries. Using these best practices helps protect data and builds trust with users. It also helps the company avoid security problems and stay reliable.