****

# 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** | **08/13/2023** | **Jose Soqui** |  |

## Client



## Instructions

Submit these 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

[Insert your name here.]

## Algorithm Cipher

Selected by NIST in 2001 to replace DES, AES has been widely adopted due to its robust security and efficiency. It's resistant to brute force attacks, performs well on modern systems, and is standardized for interoperability. Artemis Financial can choose a key length according to their risk assessment. For Artemis Financials' security requirements, I recommend using the Advanced Encryption Standard (AES) algorithm cipher to secure communications and data transfers. AES is an extensively deployed symmetric encryption algorithm which functions on fixed portions of data inputs of fixed size. AES uses substitution-permutation operations to scramble data, requiring the correct key for decryption.

## Certificate Generation

Insert a screenshot below of the CER file.

A screenshot of a computer program

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A screenshot of a computer

Description automatically generated

## Secure Communications

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

A screenshot of a certificate

Description automatically generated

I tried to trust the certificate, but chrome browser would not allow it and could not be trusted.

## Secondary Testing

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

A screen shot of a computer program

Description automatically generated

## Functional Testing

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

A screen shot of a computer program

Description automatically generated

## Summary

Changes in my code I have put in a protected controller to function as the protected controller for my systems hash RESTful stop. The ServerController classification acts to match the problems shown by the weakness assessment diagram. I furthermore opted to work with the SHA-256 encoding cryptogram as it is very secure and runs with an exceptionally tiny probability at collisions. To best continue the present security of the application I would propose one or two times each month reliance checks of the application to stay up to date on possible susceptibilities. This will assist to shield the company and their sensitive information. Ensuring the plugins within the pom.xml would also do well to keep the latest versions of the plugins running guaranteeing the greatest security.

## Industry Standard Best Practices

I implemented secure coding practices by developing a Spring Boot application that incorporates an endpoint hash to generate hash values using the SHA-256 algorithm. Utilizing Spring Boot's annotations and capabilities, I ensured input validation, secure configuration management, and effective error handling, while establishing a dedicated route for hash generation. These efforts align with industry-standard best practices, reinforcing the application's security and thwarting potential vulnerabilities like injection attacks and unauthorized entry. By embracing these practices, I contribute to our company's overall welfare, fortifying user data protection, sustaining application integrity, and nurturing trust among stakeholders.