****

# 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** | **June 23 2024** | **Julien Dorka** |  |

## 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

[Insert your name here.]

## Algorithm Cipher

## Artemis Financial is seeking enhanced security measures for their web application to ensure secure communications. Given that the primary threat to financial institutions is typically unauthorized access by bad actors seeking financial gain, encryption is the most effective recommendation. Encryption renders files unusable to any potential attacker without the decryption key.

## For securing communications, I recommend implementing asymmetric encryption. This method uses a public key for encryption and a private key for decryption, ensuring that only the intended recipient can access the information.

## To achieve the highest level of security, especially for external communications, I suggest using the SHA-256 cipher algorithm with 256-bit keys for encryption. SHA-256 provides a robust level of encryption with a vast number of possible key combinations, given its 256-bit key length.

## Additionally, the SHA-256 algorithm leverages Java's random number generator, enhancing security by creating a non-reversible checksum that verifies the file's integrity. The hash function will utilize the SHA-256 cipher to generate a checksum for the provided message, ensuring that the encryption is both secure and tamper-evident.

## Certificate Generation

Insert a screenshot below of the CER file.

A computer screen with white text

Description automatically generated

## 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 screenshot of a computer

Description automatically generated

## Secondary Testing

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

A screenshot of a computer

Description automatically generated

## Functional Testing

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

A computer screen shot of a program

Description automatically generated

## Summary

In refactoring my code, I have added a secured `RestController` to serve as the secure controller for my program's RESTful API. The `ServerController` class is designed to address the vulnerabilities identified in the assessment diagram. I opted to use the SHA-256 hashing algorithm because it is highly secure and has a minimal risk of collisions.

To maintain the current security of the application, I recommend conducting dependency checks once or twice a month. This practice will help ensure that the application remains up-to-date with the latest security patches and protects the company's sensitive data.

Additionally, keeping the plugins in the `pom.xml` file up-to-date will ensure that the application uses the latest versions of these plugins, thereby maintaining the highest level of security.

## Industry Standard Best Practices

Working on this project for Artemis has shown me that implementing industry standard best practices for secure coding is essential for maintaining software security and preventing known vulnerabilities. These practices i used include input validation, error handling, strong authentication, access control, cryptographic measures, logging, quality assurance checks, and proper code management. By adhering to these standards, it enhances overall well-being of the company’s business by reducing risks, costs, and reputational damage associated with security breaches.