****

# 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 24, 2024 | Karina Washington | Recommendations for AES encryption cipher, generated self-signed certificates using Java Keytool, implemented AES cipher and checksum verification, configured HTTPS protocol for secure communications, conducted static and functional testing, and summarized adherence to industry-standard best practices for secure coding. |

## Client



## Developer

Karina Washington

## Algorithm Cipher

For secure communication and data integrity, the Advanced Encryption Standard (AES) cipher was selected. AES is widely recognized for its efficiency and strong encryption capabilities, making it suitable for protecting sensitive financial information. It utilizes a symmetric key approach with key sizes of 128, 192, or 256 bits, providing robust security against unauthorized access.

## Certificate Generation

Self-signed certificates were generated using Java Keytool within Eclipse IDE. The process involved creating a private key and obtaining a corresponding certificate signing request (CSR), which was then used to generate a self-signed certificate. The certificate was exported as a CER file to facilitate secure HTTPS communication.

**A screenshot of a computer

Description automatically generated**

## Deploy Cipher

To ensure data integrity during transmission, the AES cipher was implemented within the application code. A cryptographic hash algorithm was integrated to compute checksums for verifying data integrity. The screenshot below demonstrates successful checksum verification, ensuring data was not altered during transfer.

A screenshot of a computer

Description automatically generated

## Secure Communications

The application’s properties file was updated to enable HTTPS protocol instead of HTTP. This configuration change was verified by accessing the application via "<https://localhost:8443/hash>" in a web browser. The screenshot below confirms successful access to a secure webpage, indicating HTTPS protocol implementation.

A screenshot of a computer

Description automatically generated

## Secondary Testing

Static analysis was conducted using the dependency-check tool to ensure the refactored code adhered to security enhancements. The screenshots below depict the refactored code executed without errors and the dependency-check report confirming no new security vulnerabilities were introduced.

A screen shot of a computer

Description automatically generated

## Functional Testing

## Manual functional testing was performed to identify and address syntactical, logical, and security vulnerabilities in the refactored code. The screenshot below illustrates the refactored code executed without errors, confirming the successful implementation of security enhancements.

## A screen shot of a computer Description automatically generated

## Summary

The code was refactored to integrate AES encryption for secure communication and data integrity. Through iterative testing and validation, vulnerabilities were identified and mitigated, ensuring compliance with Artemis Financial’s security requirements. The addition of HTTPS protocol, checksum verification, and adherence to industry-standard best practices further enhanced the application’s security posture.

## Industry Standard Best Practices

Industry-standard best practices were applied throughout the project to mitigate security risks and maintain application security. Techniques such as input validation, secure configuration management, and regular security testing were implemented to safeguard sensitive financial data. These practices not only protect Artemis Financial’s operations but also contribute to the overall resilience and trustworthiness of their software solutions.