****

# 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** | **04/20/2025** | **Carlos Trujillo** |  |

## 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

Carlos Trujillo

## Algorithm Cipher

For Artemis Financial’s secure file verification, I implemented a cryptographic hash function using the SHA-256 algorithm. SHA-256 is part of the SHA-2 family developed by the NSA and standardized by NIST. It produces a fixed-length 256-bit hash, which makes it collision-resistant and secure against pre-image and second pre-image attacks.

SHA-256 is a one-way function, meaning it cannot be reversed, making it ideal for verifying data integrity without exposing original content. Since it doesn’t require a key, it’s not a symmetric or asymmetric cipher — it’s a pure hash function. Random numbers are not used with SHA-256 hashing. This algorithm is widely used across the industry for validating the integrity of data in applications like SSL certificates, blockchain, and digital signatures.

## Certificate Generation

Insert a screenshot below of the CER file.

A screenshot of a computer program

AI-generated content may be incorrect.

## Deploy Cipher

A screenshot of a computer

AI-generated content may be incorrect.

## Secure Communications

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

A screenshot 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 screenshot of a computer

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

In this project, I refactored the Artemis Financial web app to support secure communication and data verification using SSL and SHA-256. I created a custom checksum endpoint, configured HTTPS using a self-signed certificate, and ensured the application compiled and ran without vulnerabilities introduced by the new code. I verified that data was hashed securely and transmitted over a secure channel, and I confirmed functionality through manual testing and static analysis. The refactored components passed all tests, supporting Artemis’s mission of secure, modern operations.

## Industry Standard Best Practices

I applied industry-standard secure coding practices throughout this project. I used a widely accepted hash algorithm (SHA-256), followed Java conventions for certificate handling using keytool, and configured Spring Boot for HTTPS with a self-signed certificate stored in the classpath. To validate my changes, I used OWASP Dependency Check to confirm that my updates did not introduce new vulnerabilities.

Using best practices ensures the long-term safety of both user data and application integrity. These practices are essential for companies like Artemis Financial that handle sensitive financial information. It also ensures our applications can scale securely as they integrate with real-world users and services.