****

# 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** | **[Date]** | **Connor** |  |

## 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

Connor Casey

## Algorithm Cipher

A good encryption algorithm for Artemis Financial to use would be AES, Advanced Encryption Standard. AES is a symmetric key cipher. This means that the same key is used for encrypting and decrypting data. AES also works with different key sizes, 256 bits being the strongest (AES-256). AES is widely used for financial and enterprise systems. It is considered to be highly secure. In general, the higher the bit level, the harder it is for attackers to break the encryption. This is why AES-256 is often preferred for protecting sensitive data such as financial information. Random numbers can also play a role in encryption by helping generate secure keys and initialization vectors. This makes it harder for attackers to predict patterns. When comparing asymmetric encryption versus symmetric encryption, symmetric encryption is usually faster, making it ideal for frequent and secure communication. Historically, AES replaced DES (Data Encryption Standard) after DES was proven to be too weak and easily attacked. AES remains the standard recommendation by organizations like NIST and is one of the strongest and most trusted encryption algorithms.

## Certificate Generation

Insert a screenshot below of the CER file.

A screenshot of a certificate

AI-generated content may be incorrect.

## Deploy Cipher

Insert a screenshot below of the checksum verification.

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 screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a document

AI-generated content may be incorrect.

## Functional Testing

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

A screenshot of a computer screen

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

## Summary

The code was refactored to enhance security by introducing HTTPS communication and a cryptographic hash function to verify data integrity. By adding SSL configuration and using a self-signed certificate, I was able to get the application to encrypt all traffic between the client and the server. This ensures that sensitive data is transmitted securely. The REST controller was updated to include the @GetMapping annotation which makes the checksum endpoint accessible. This addresses key areas highlighted in the vulnerability assessment process flow diagram, including secure coding practices, secure client-server interactions, and secure input and representations.

Adding layers of security followed a multi-step process. First, secure data structures and encapsulation were applied by keeping the checksum logic contained within a controller class. Then, the HTTPS configuration and keystore setup enforce secure transport. Static testing and dependency analysis were performed to ensure no new vulnerabilities were introduced during refactoring. The refactored application fully addresses the key steps of the vulnerability assessment process flow.

## Industry Standard Best Practices

Industry standard best practices for secure coding were applied to maintain and enhance the application’s security. HTTPS was enforced using a properly configured SSL keystore to ensure encrypted communication between clients and the server. This prevents data interception and man-in-the-middle attacks. The checksum endpoint was implemented using SHA-256, a widely used cryptographic hash function. Additionally, manual code review and dependency scanning with OWASP Dependency-Check tool ensured that no new vulnerabilities were introduced during the refactoring process.

Applying these industry standard best practices has significant value for the company’s overall well-being. By enforcing secure communication, validating data integrity, and following secure coding conventions, the software becomes more resilient to attacks. This not only protects Artemis Financials’ clients but also strengthens Global Rain’s reputation and trustworthiness in the financial application market. Addressing potential vulnerabilities proactively aligns the software development process with regulatory and industry security standards.