****

# 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** | **10/19/25** | **Cielo Neal** |  |

## 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

Cielo Neal

## Algorithm Cipher

I would recommend AES-256 as the algorithm cipher for Artemis Financial’s usage. It is a block cipher that encrypts data in 128 blocks and generates 256-bit keys. It is highly trusted as an industry standard, even being used by the US government, being incredibly resistant to direct attacks.

In cryptography, random numbers are often used in key generation. In AES, a random number is used as an initial value for the matrix math used by the algorithm, which ensures that the same plaintext will generate different results if ran multiple times. Symmetric keys are used to both encrypt and decrypt data and are most frequently used for file encryption and secure data storage. Meanwhile asymmetric key encryption is when a public key, which can be freely shared, encrypts the data and a private key decrypts the data. This is more often used for digital signatures, SSL and TLS.

Encryption is an ever-evolving field that, as one half of the perpetual security arms race. DES was considered the standard encryption algorithm to use but was in dire need of replacement due to increased computing power allowing brute force to become possible. AES was developed for this reason, as part of competition held by NIST, and has been the standard since. The computing power does not currently exist to easily brute force AES-256 keys, though that may change in the future as computers continue to become more powerful.

## Certificate Generation

Insert a screenshot below of the CER file.

A computer screen shot of a black screen

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 screenshot of a computer screen

AI-generated content may be incorrect.A screenshot of a computer error

AI-generated content may be incorrect.

## Functional Testing

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

A black screen with white text

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

## Summary

In my refactoring of the code, I started by looking at the overall architecture of the application. This was done most easily by examining the pom.xml file and manually checking the dependencies. The first thing I changed was updating the version of spring the application was using to one that is still supported by the developer. I also updated the maven dependency checker to the newest version.

I added a RestController to the SSLServerApplication java file, which creates a SHA-256 hash and checksum verification. This addresses the concern of secure API interaction and cryptography. I also edited the application properties to use a self-signed certificate for testing purposes. In the future, a proper certificate should be issued by a certificate authority. This certificate will enable the use of HTTPS rather than HTTP. A converter configuration class was also created to automatically convert HTTP requests to HTTPS. This will secure the client/server connection and help ensure no data is leaked via URLs.

## Industry Standard Best Practices

In my refactoring, I used industry best standards in my use of comments throughout the code, so that anyone working on the same project can understand what each element does. I also made sure all resources used are up to date, which dramatically reduced the number of vulnerabilities present in the application. As I added elements to the code, I ensured that security did not degrade through frequent testing, both making sure the code ran and that no new vulnerabilities were added in my refactoring.

Applying industry best standards ensures the longevity and security of the program, as using industry standard naming conventions and comments mean that the code can be worked on by more programmers than just yourself. Additionally, using best practices such as graceful error handling and input validation are the first step in securing any program, as they are simple to implement and relevant for all applications.