****

# 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 5](#_Toc102040761)

[5. Secondary Testing 5](#_Toc102040762)

[6. Functional Testing 5](#_Toc102040763)

[7. Summary 6](#_Toc102040764)

[8. Industry Standard Best Practices 6](#_Toc102040765)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **02/17/2025** | **Robert Fjellin** |  |

## 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

Robert Fjellin

## Algorithm Cipher

Artemis Financial is a financial consulting company that creates individualized financial plans for their clients. SHA-256 would be an excellent fit for Artemis Financial because it provides an industry standard approach to file verification and secure communications. It is also widely adopted in the finance world. SHA-256 creates a fixed 256-bit hash value regardless of the input size. The hash is deterministic meaning for every time “SNHU” is the input, “xyz” will be the output, but if changed the hash value will completely change. Symmetric encryption uses one key to lock/unlock data, whereas asymmetric uses 2 different keys, one to lock and another to unlock. SSL originated in 1995 by Netscape and was the most popular security protocol. SSL 2.0 and 3.0 came not long after to address vulnerabilities.

## 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 black and white text

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.

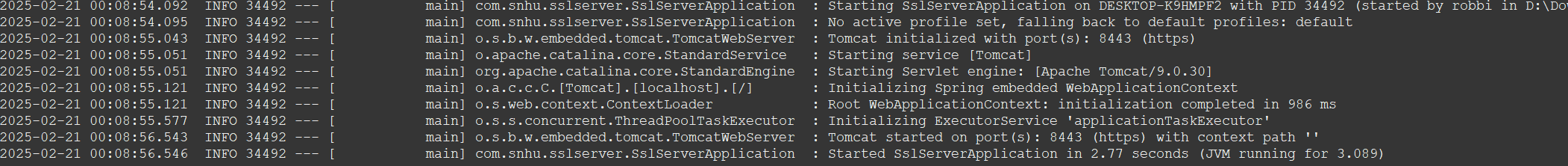
A close-up of a white background

AI-generated content may be incorrect.

## Functional Testing

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

A computer screen shot of a program

AI-generated content may be incorrect.

## Summary

The code for this software application underwent major changes. Multiple dependencies were vulnerable and needed addressing. All vulnerabilities were addressed with updates patching all of them. SSL was implemented through a keystore and it was configured on port 8443 with aa secure connection. Port 8443 was declared in the application.properties file along with a key store type, alias, and password. The code for the application was refactored and now uses SHA-256 hashing in the REST controller to ensure secure data transfers.

## Industry Standard Best Practices

Industry standard best practices were used throughout this project. One of the primary focuses was following the OWASP guidelines, with the dependency management. SSL/TLS was implemented using a keystore which adheres to the industry standards for encryption. SHA-256 (widely used cryptographic hash function) was used for secure data handling which follows the National Institute of Standards and Technology.. Maintaining updated dependencies and standard security protocols provides a great foundation for a great product/project.