****

# 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** | **4/15/2023** | **Javon Jackson** |  |

## 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

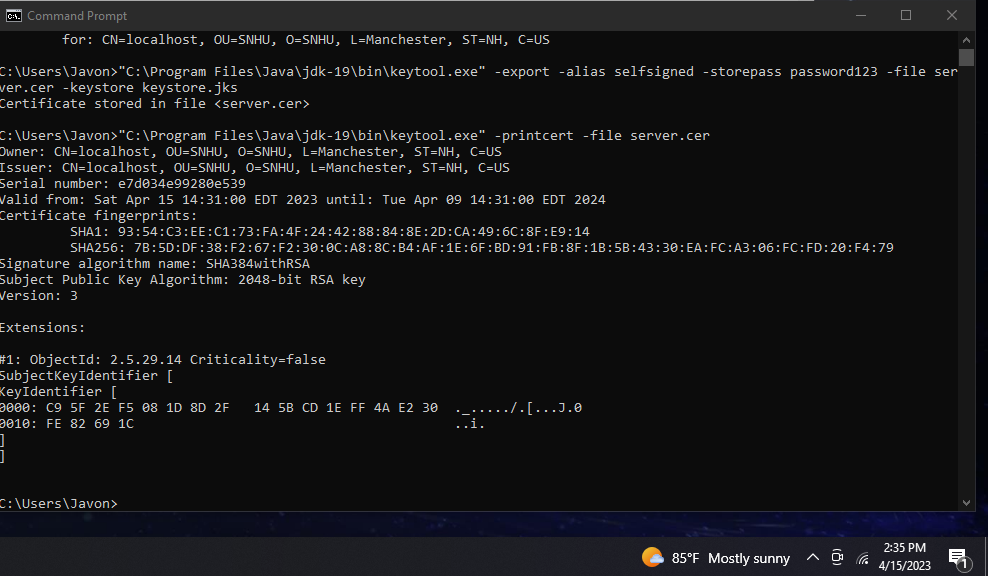
Javon Jackson

## Algorithm Cipher

Encryption began as a way to protect knowledge and today exists as a way to protect sensitive data from falling into the wrong hand. Making use of a trusted cipher algorithm is the best way to protect data used by the Artemis Financial web application. Encryption often makes use of randomness to generate unique cryptographic keys. Keys can be symmetric or symmetric depending on security needs. Using symmetric keys means the same key is used for encryption and decryption. Using asymmetric keys means a public key is provided to anyone who wants to send data to you, but the data is decrypted by a private key. The algorithm cipher I recommend for generating a Checksum is the SHA-256 algorithm. This encryption algorithm is used to transform data into a fixed size output using a hash function. The SHA-256 algorithm is highly collision resistant, with the rate of collision being about 1 / 2^256. Avoiding hash collisions is important so that two data items are not assigned the same hash value. Giving each piece of data a unique hash value makes it easy to verify if the contents have been altered by comparing the hash value before and after a data transfer. The algorithm is trusted by the United States Government and is the trusted standard for many technology companies.

## Certificate Generation

Insert a screenshot below of the CER file.



## Deploy Cipher

Insert a screenshot below of the checksum verification.

Graphical user interface, text, application

Description automatically generated

## Secure Communications

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

A picture containing text, monitor, screenshot, screen

Description automatically generated

## Secondary Testing

Insert screenshots below of the refactored code executed without errors and the dependency-check report.

Text

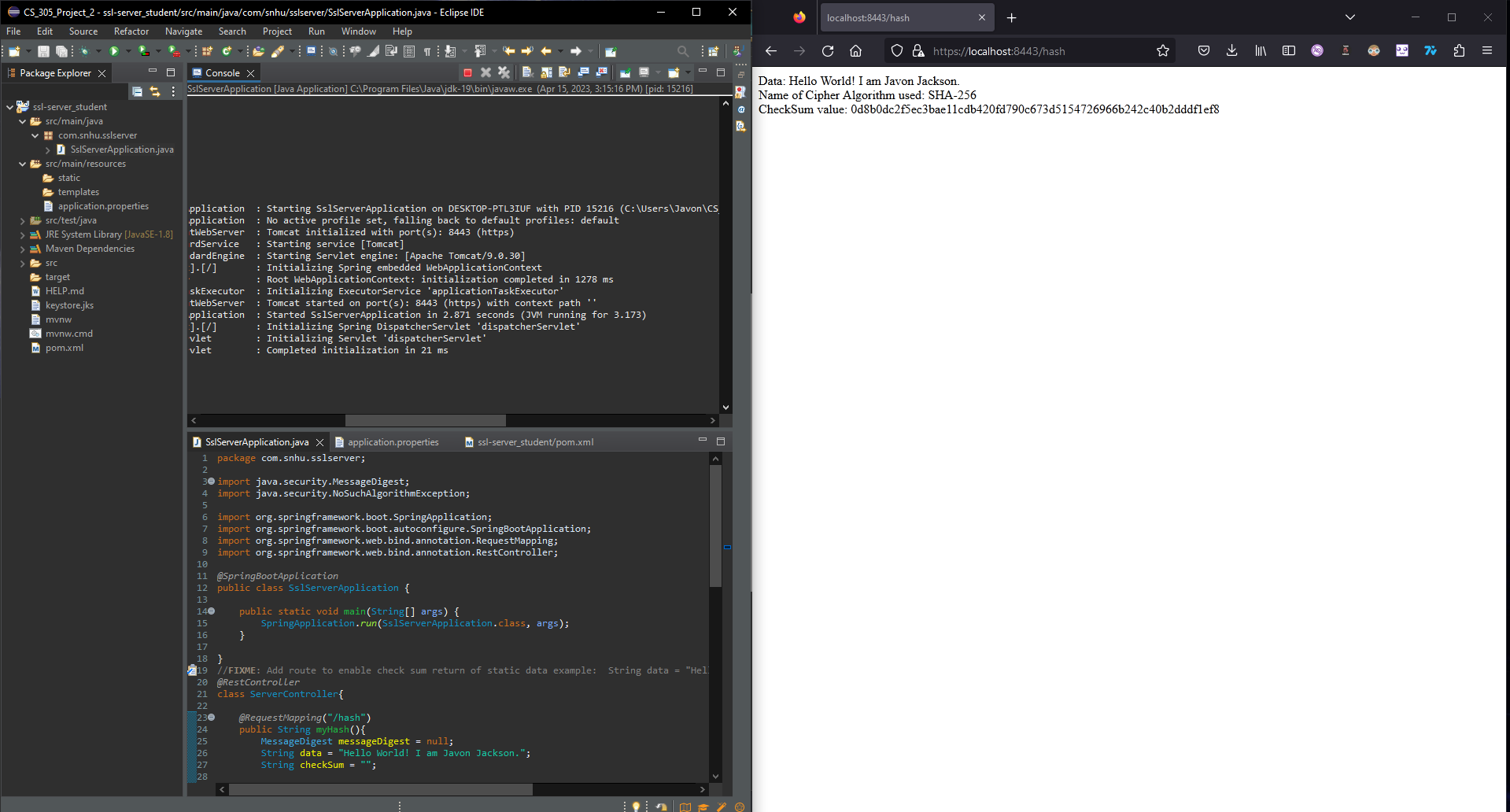
Description automatically generated

Graphical user interface, application

Description automatically generated

## Functional Testing

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



## Summary

The three areas of the vulnerability assessment process flow diagram that I have addressed are secure API interactions, use of encryption and secure coding practices. Certificate generation has been implemented as a security layer to verify that the certificate bearer is who or what they claim to be when they want to interact with the Artemis Financial web app. Encryption of data has been implemented using a secure hash algorithm to verify that no changes are made to data during transit. The code has been debugged to verify that no errors occur with spring boot or with the coded operations themselves. Finally, a dependency check was performed after adding these features to ensure that no new vulnerabilities were introduced.

## Industry Standard Best Practices

Industry standard best practices are important for maintaining high code quality and making the debugging process easier. For example, a try/catch block has been implemented to handle errors related to instantiating the message digest with the selected cipher algorithm. The cipher algorithm used is an industry standard for generating a checksum and is even trusted by the United States government. The code has also been tested to verify that the added features do not introduce new errors or vulnerabilities to the program. By following industry standard practices, I believe the changes I have implemented to the Artemis Financial web application will improve its security overall.