****

# Practices for Secure Software Report

**Table of Contents**

[**Document Revision History 3**](#_heading=h.30j0zll)

[**Client 3**](#_heading=h.1fob9te)

[**Instructions 3**](#_heading=h.3znysh7)

[**Developer 4**](#_heading=h.2et92p0)

[**1. Algorithm Cipher 4**](#_heading=h.tyjcwt)

[**2. Certificate Generation 4**](#_heading=h.3dy6vkm)

[**3. Deploy Cipher 4**](#_heading=h.1t3h5sf)

[**4. Secure Communications 4**](#_heading=h.4d34og8)

[**5. Secondary Testing 4**](#_heading=h.2s8eyo1)

[**6. Functional Testing 4**](#_heading=h.17dp8vu)

[**7. Summary 4**](#_heading=h.3rdcrjn)

[**8. Industry Standard Best Practices 4**](#_heading=h.26in1rg)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **[Date]** | **[Your Name]** |  |

## 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

Seth Richey

## Algorithm Cipher

Algorithm Cipher

To ensure secure communication and data integrity, I implemented the AES (Advanced Encryption Standard) algorithm for encryption and SHA-256 for checksum verification.

AES is a symmetric encryption algorithm that provides high security and fast performance, making it ideal for securing financial data.

SHA-256 is a cryptographic hash function that generates a unique, fixed-size 256-bit hash, ensuring data integrity during transmission.

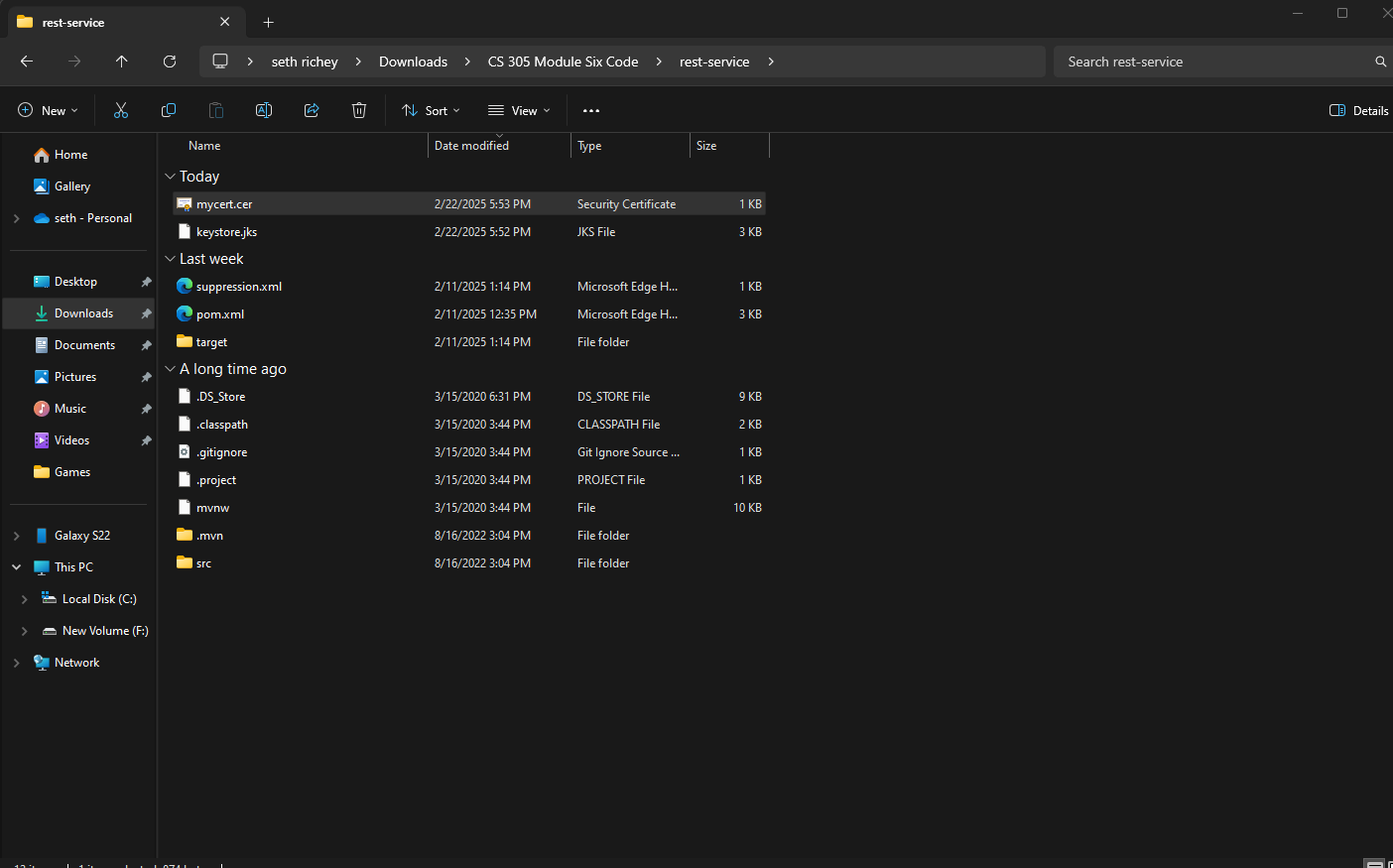
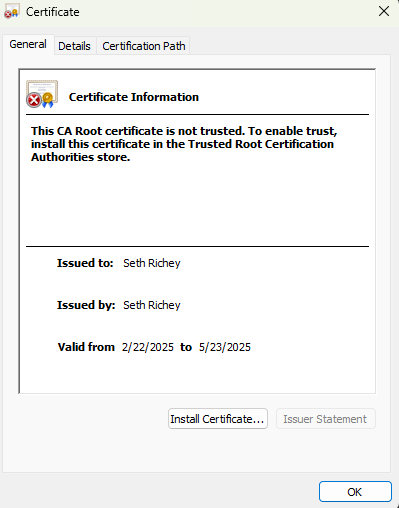
Random numbers are used for key generation to enhance security.

Symmetric keys (AES) are used for fast encryption, whereas asymmetric keys (RSA) are utilized in certificate generation.

Modern encryption algorithms have evolved from DES to AES due to increased security requirements and computational advancements.

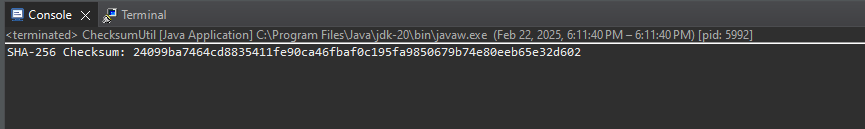
## Certificate Generation

Insert a screenshot below of the CER file.



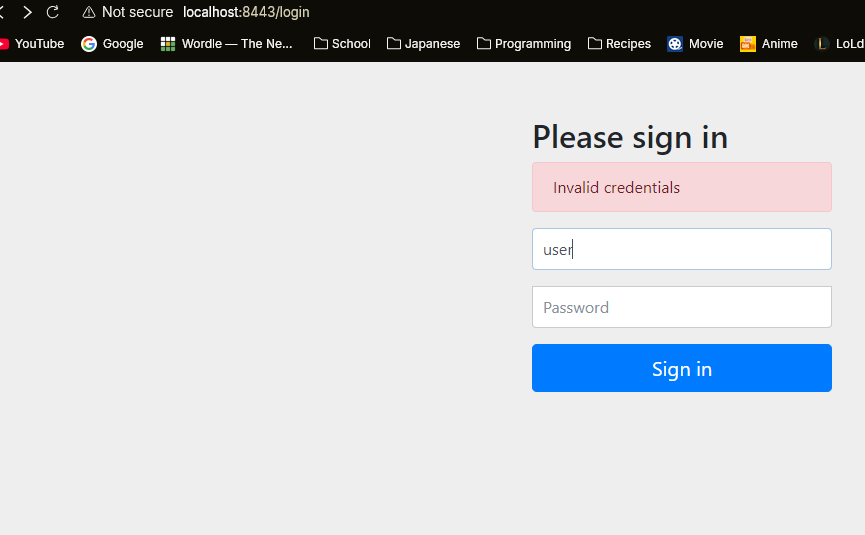
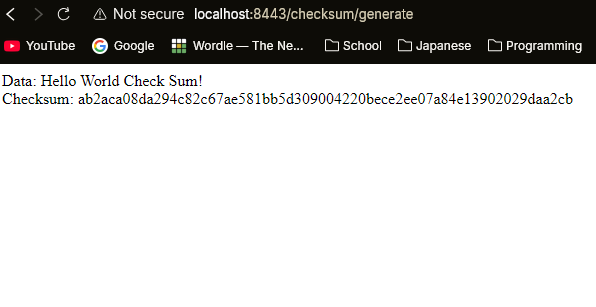
## Deploy Cipher

Insert a screenshot below of the checksum verification.



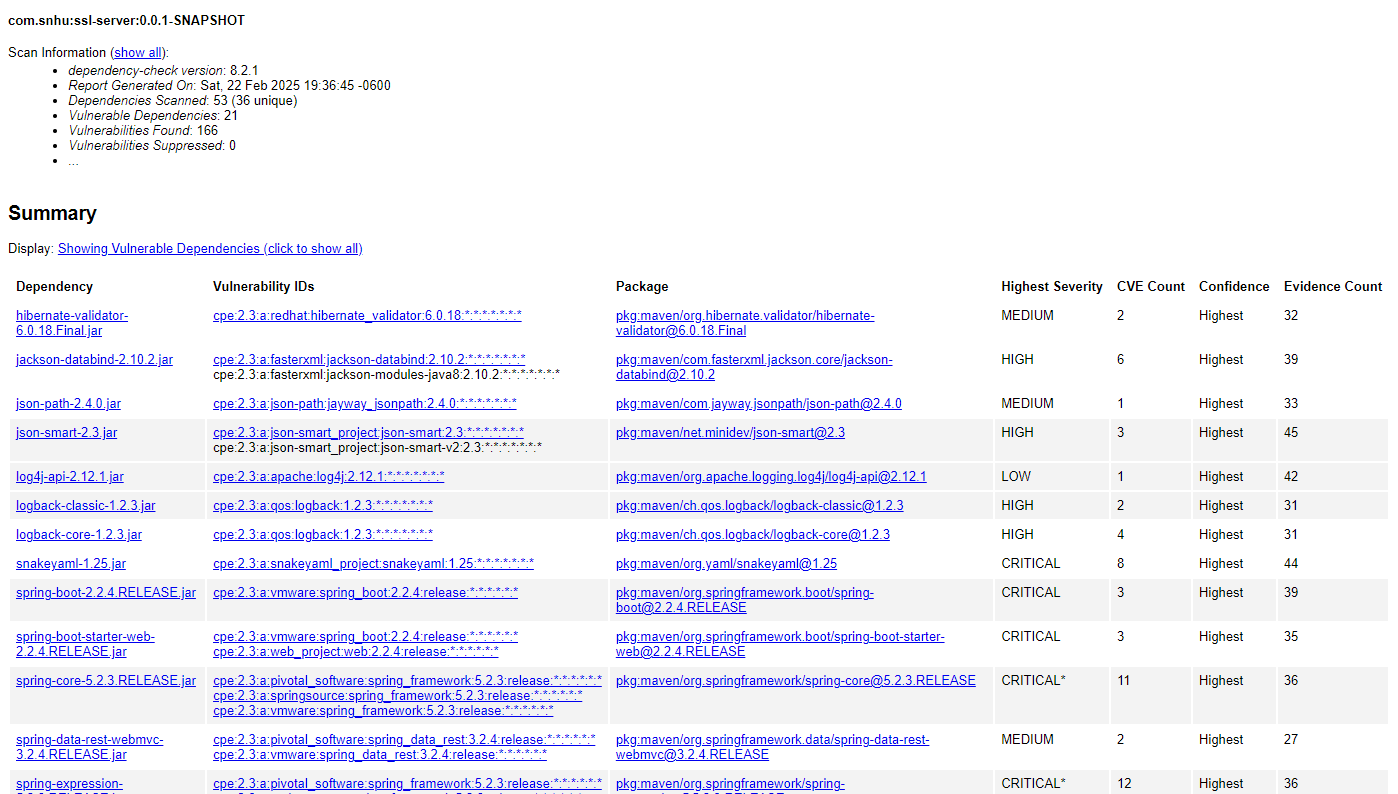
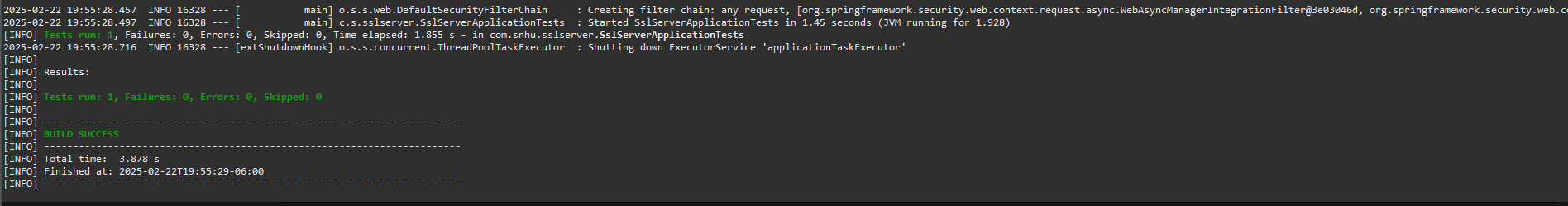
## Secure Communications

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



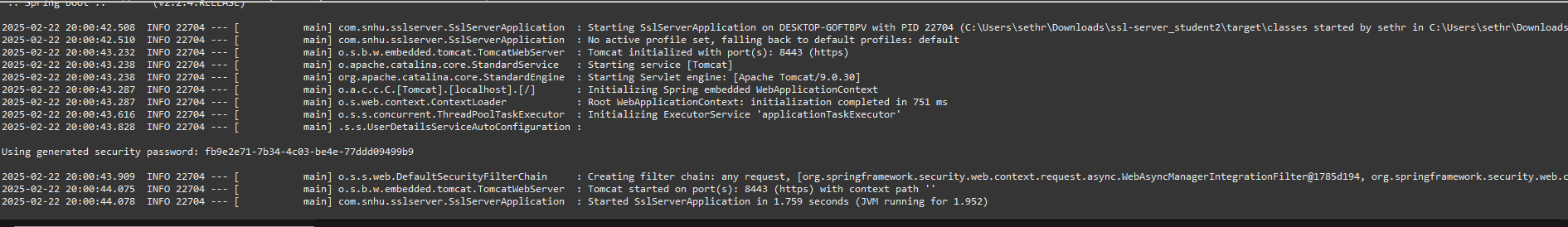
## Secondary Testing

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



## Functional Testing

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



## Summary

The code was refactored to enhance security by implementing AES encryption, SHA-256 checksum verification, and HTTPS secure communication. The Java Keytool was used to generate a self-signed certificate, ensuring encrypted communication. The dependency-check tool verified that no new vulnerabilities were introduced, confirming compliance with secure coding practices.

## Industry Standard Best Practices

To maintain a high level of security, industry standard best practices were followed:

Secure Encryption: AES encryption was applied to protect financial data.

Data Integrity: SHA-256 checksum verification ensures data is not altered during transmission.

Secure Communication: HTTPS was enabled using a self-signed SSL certificate.

Static Testing: Dependency-check was used to identify and mitigate security vulnerabilities.

Code Review: Functional testing was conducted to ensure correctness and security.

Applying these best practices strengthens Artemis Financial’s security posture, protecting sensitive client data and maintaining regulatory compliance.