

# 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** | **10/22/25** | **Gabriel West** | **Creation of document** |

## 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

Gabriel West

## Algorithm Cipher

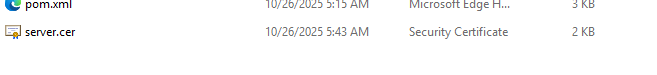
Given Artemis Financial's goal is to add a data verification via a checksum for data transferred through the web application, there are several factors we must consider when it comes to choosing an algorithm cipher. Since the data being transferred is considered sensitive, it is important to obfuscate the raw data using a cipher. Ciphers use either symmetric or asymmetric keys. Symmetric key ciphers use the same key for both encrypting and decrypting data, which proves to be an efficient way of sending frequent, large swaths of data. Asymmetric key ciphers use two sets of keys, private and public. Public keys, which as the name suggests, are public, and are used to encrypt data. Private keys, which are not meant to be shared, are used to decrypt the data encrypted with the public key.

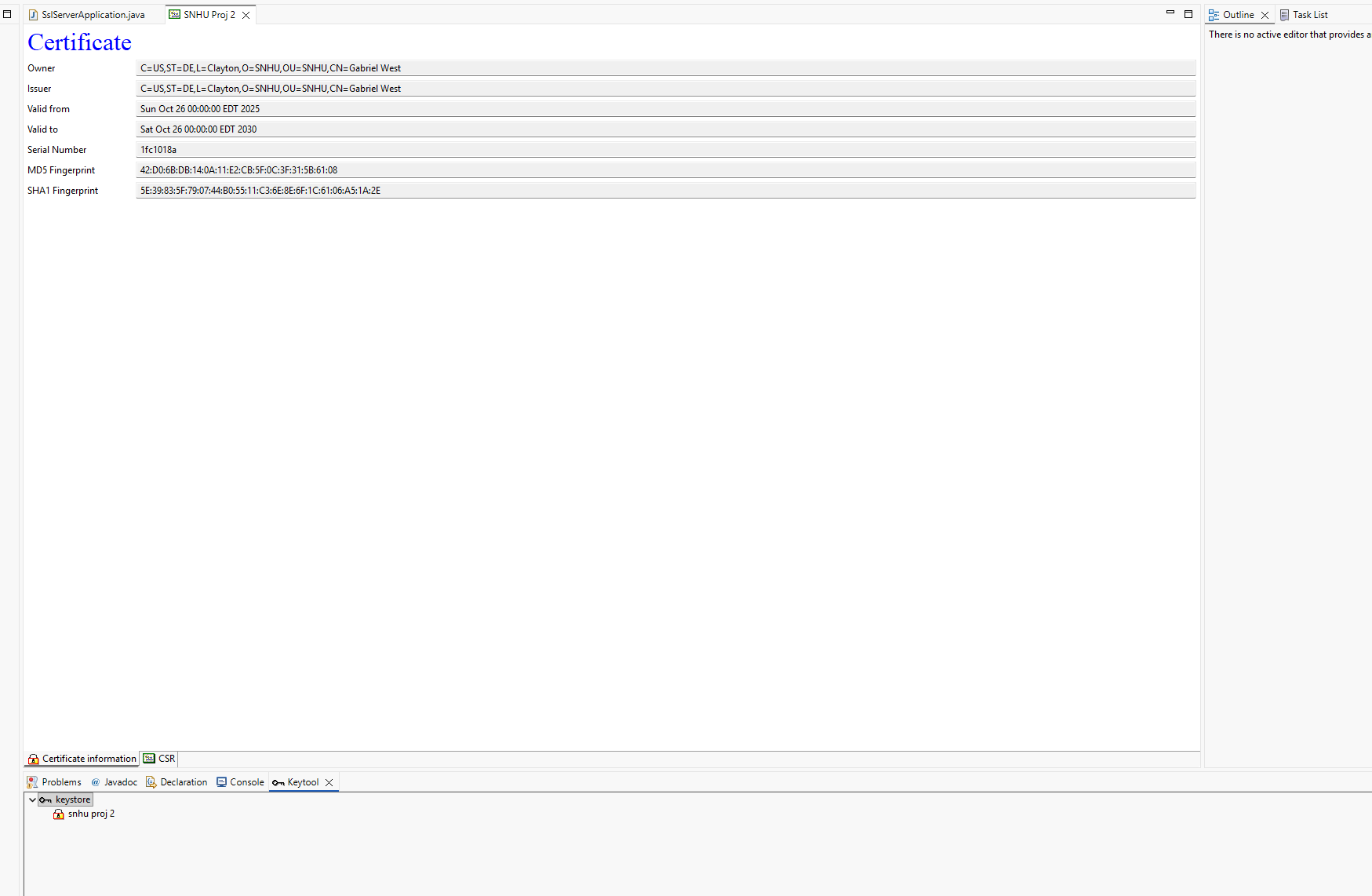
For our purposes, I would recommend a symmetric key cipher such as AES over an antiquated cipher such as 3DES. This is because ciphers can have different key sizes, such as 128-bit. This means that there are 2^128 different possibilities for the key that can be used. Starting from the one of the oldest standard ciphers, DES, it had a 56-bit key length, which could easily be decrypted with modern technology. We later upgraded to 3DES, which allowed for 168-bit key lengths. However, modern AES supports 256-bit key length while also being faster than 3DES. Since we are dealing with financial information, I recommend opting for a larger and more secure bit size, such as 256-bit. I would also run AES in GCM mode as it is modern and more secure than other modes, such as ECB Mode. As for the checksum, I would recommend SHA–512 because according to NIST, it is a collision resistant algorithm. SHA–512 is chosen over other options such as SHA–256 and SHA3-512 because it performs better on modern 64-bit processors. It does this while being secure enough for our intended use case.

To summarize, I recommend using AES-256-GCM + SHA-512.

## Certificate Generation

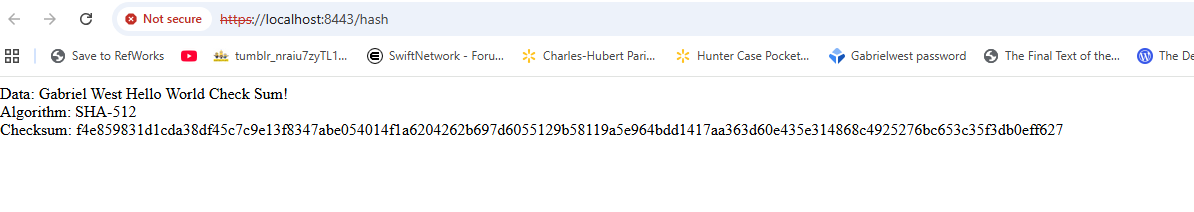
Insert a screenshot below of the CER file.



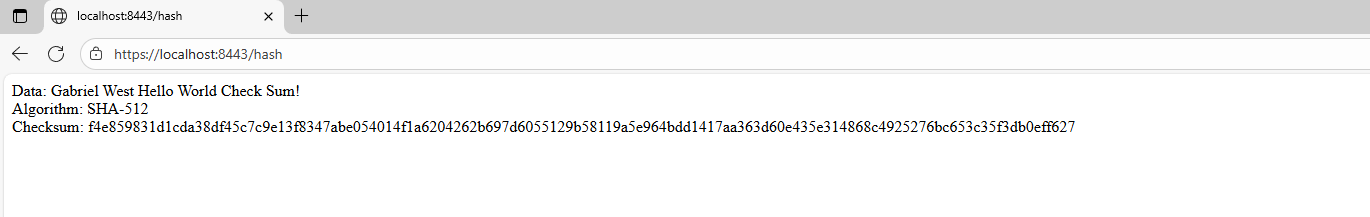


## Deploy Cipher

Insert a screenshot below of the checksum verification.

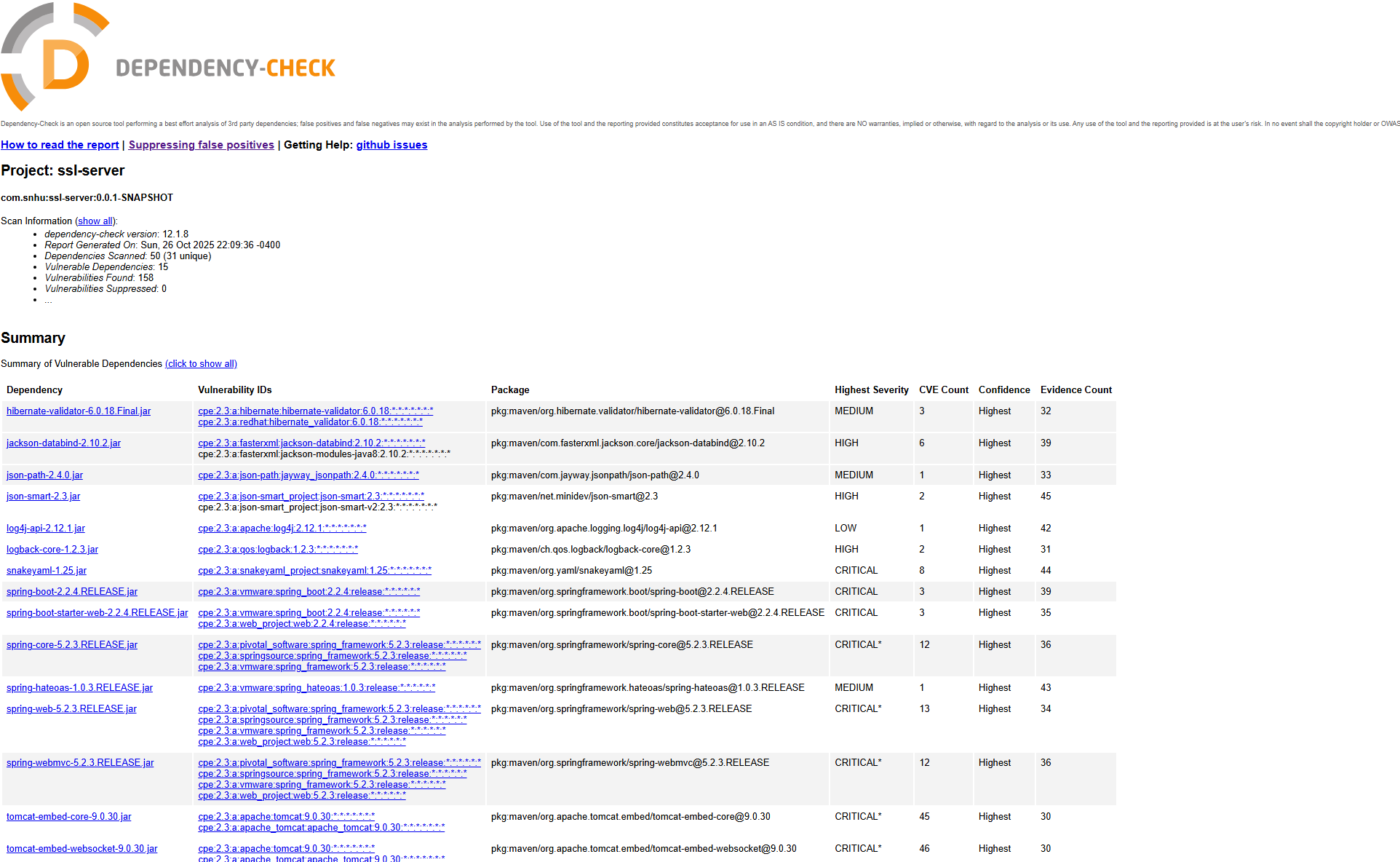
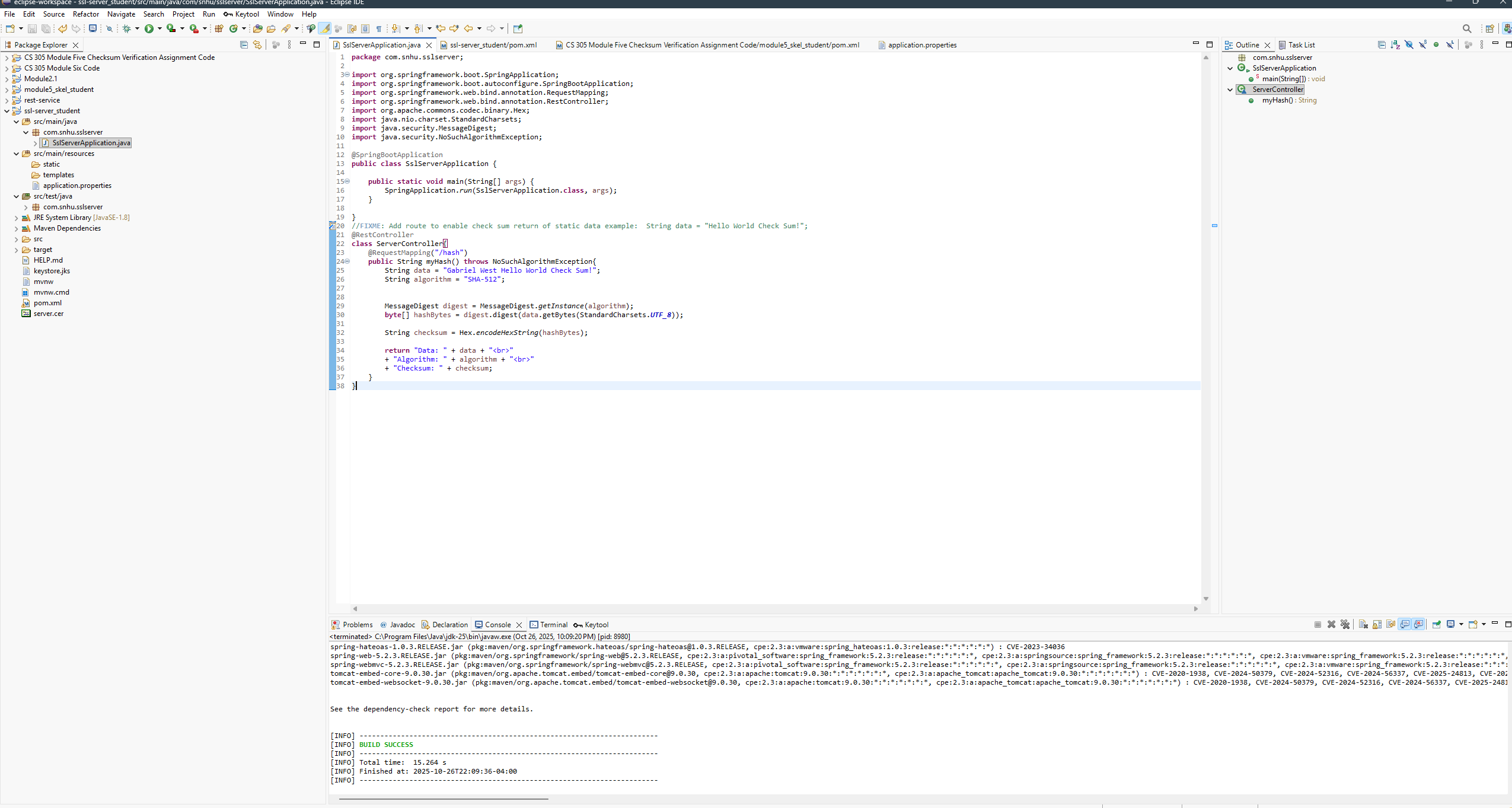


## Secure Communications



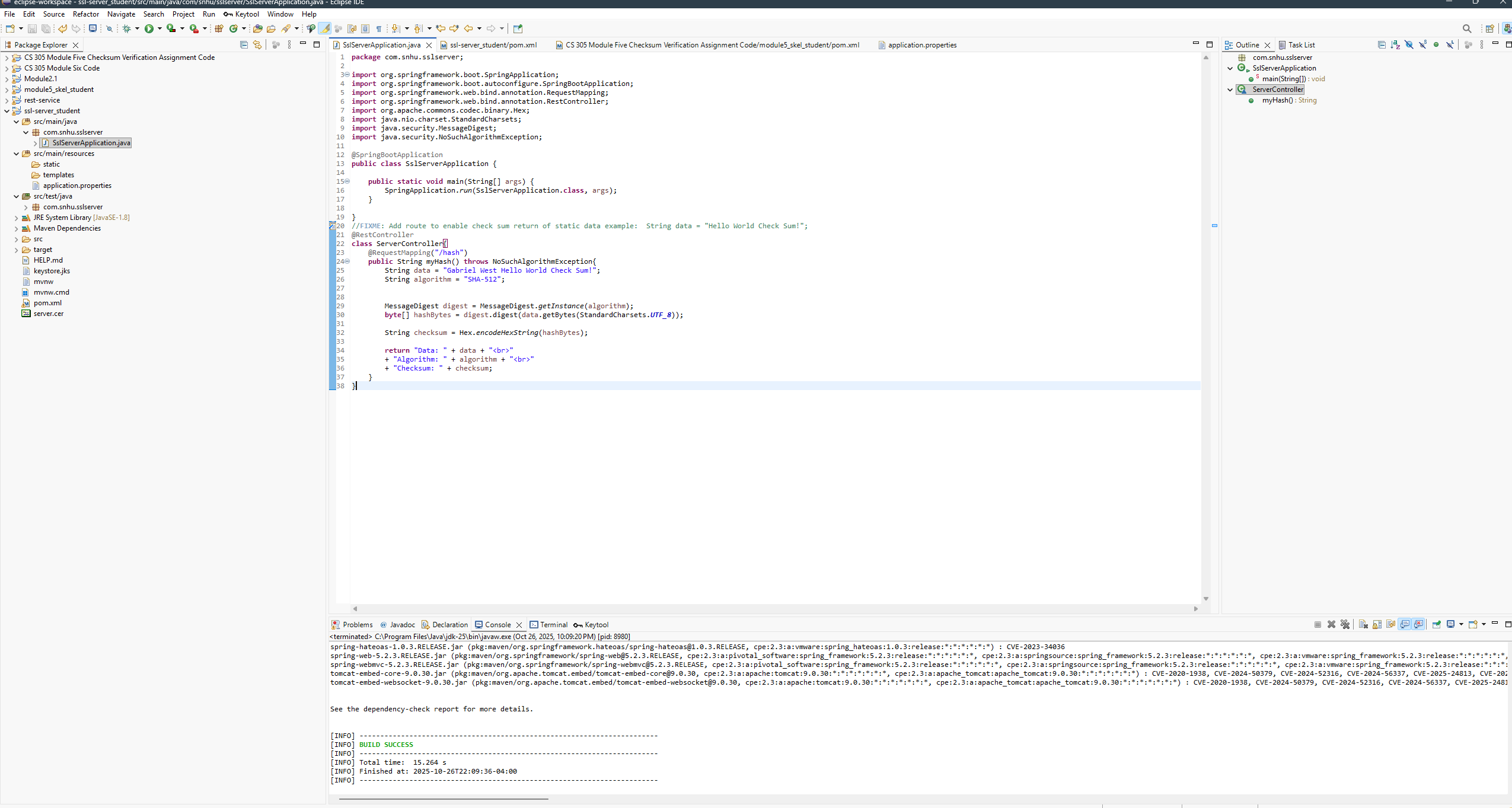
## Secondary Testing

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

By refactoring the Artemis Financial software, the code saw great improvements to its security in several key areas. First, there were massive improvements regarding cryptography by my implementation of a checksum value. This is used to verify the integrity of data sent. The recommendation to use AES-256 in GCM will also further strengthen the security of the codebase by not transferring data in raw text. Also through the implementation of a security certificate, the client/server communication was secured by refactoring the website from HTTP to HTTPS. Finally, through the implementation of the Maven dependency check plugin, I scanned for vulnerabilities in the code that I had added, which demonstrates code review.

## Industry Standard Best Practices

Industry Standard Best Practices were followed throughout the refactoring. The use of a collision resistant hash function aligned with NIST secure principles. The use of the industry standard HTTPS in place of HTTP in conjunction with a security certificate. Also, the proper storage of keys within Java keystore follows industry standard best practices.