****

# 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** | **2/18/2025** | **Jose Lopez** |  |

## 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

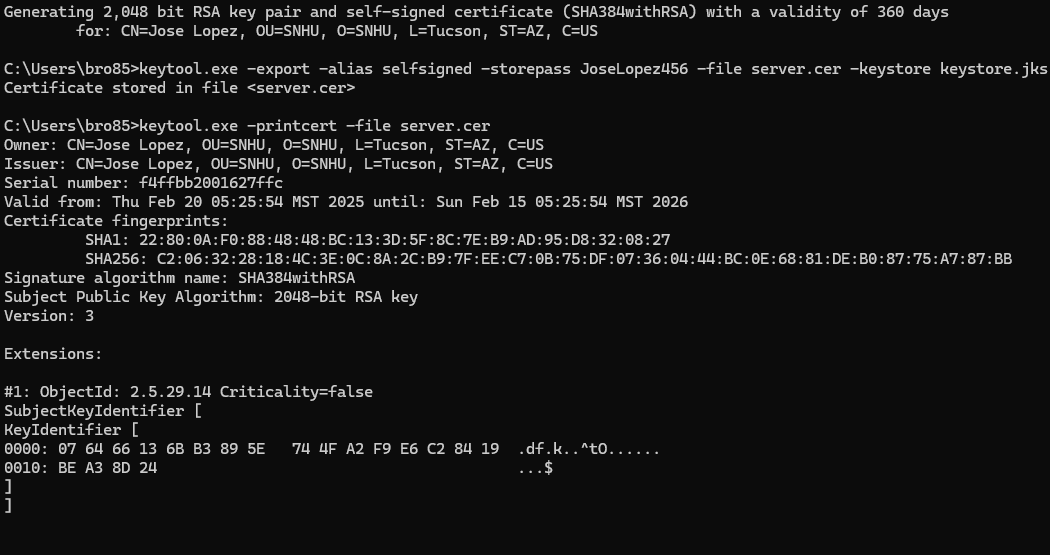
Jose Lopez

## Algorithm Cipher

I recommend using the SHA-256 algorithm cipher with 256-bit keys for Artemis Financial since they want to include a file verification step. This cipher is made to ensure that two different inputs never produce the same hash value or anything similar. This is a strong choice and is used across industries to encrypt different types of data. This will give Artemis Financial a safe solution for protecting their data in a web-based application. “SHA 256 is a part of the SHA 2 family of algorithms, where SHA stands for Secure Hash Algorithm. Published in 2001, it was a joint effort between the NSA and NIST to introduce a successor to the SHA 1 family, which was slowly losing strength against brute force attacks” (Jena, 2023). SHA-256 takes your input like a string or file and adds extra bits to make it fit a 32-byte output regardless of the input size. Then it breaks that into chunks and processes each chunk to mix everything up. The end result is a unique never duplicated fixed-size string of characters that represents the input. The 256-bit output provides a really high level of security which is important for protecting sensitive data. With SHA-256 the hash function and bit levels are from Java’s random number generator and are irreversible, meaning that plain text cannot be reverted or restored to its original value. This type of encryption is crucial for securing all types of data needed for financial companies. “Because cryptography has such a long and rich history, there is the opportunity to pick up on aspects of the history syllabus, such as the Elizabethans or the Second World War” (Singh, 2003). Encryption methods can be traced back to significant parts of our history as governments tried to protect their sensitive data. In the mid-1970s, the Data Encryption Standard (DES) was developed and was widely used to secure communications. Now today we use modern computer encryption methods to protect data on the web even with newer SHA versions SHA 2 still is the standard and trusted in many applications.

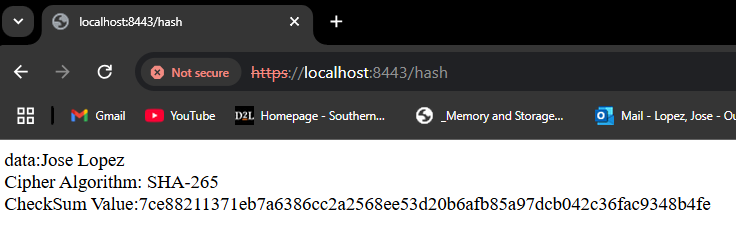
## Certificate Generation

Insert a screenshot below of the CER file.



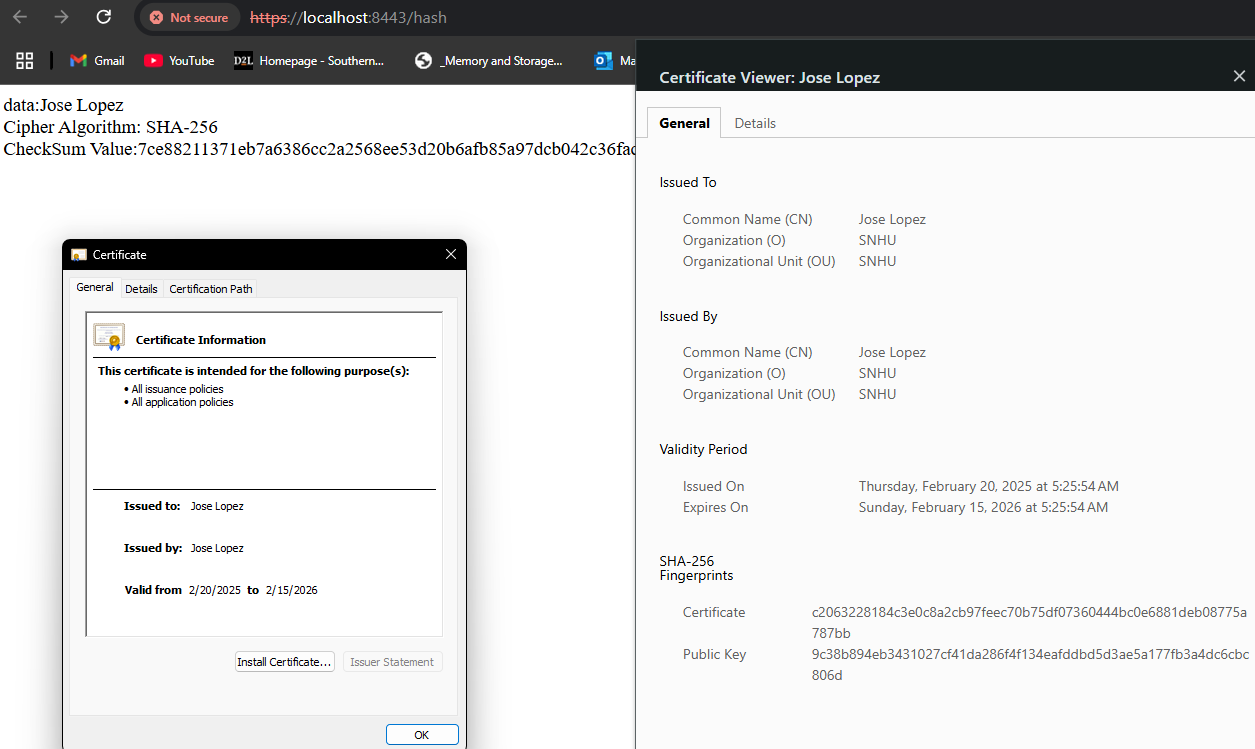
## Deploy Cipher

Insert a screenshot below of the checksum verification.



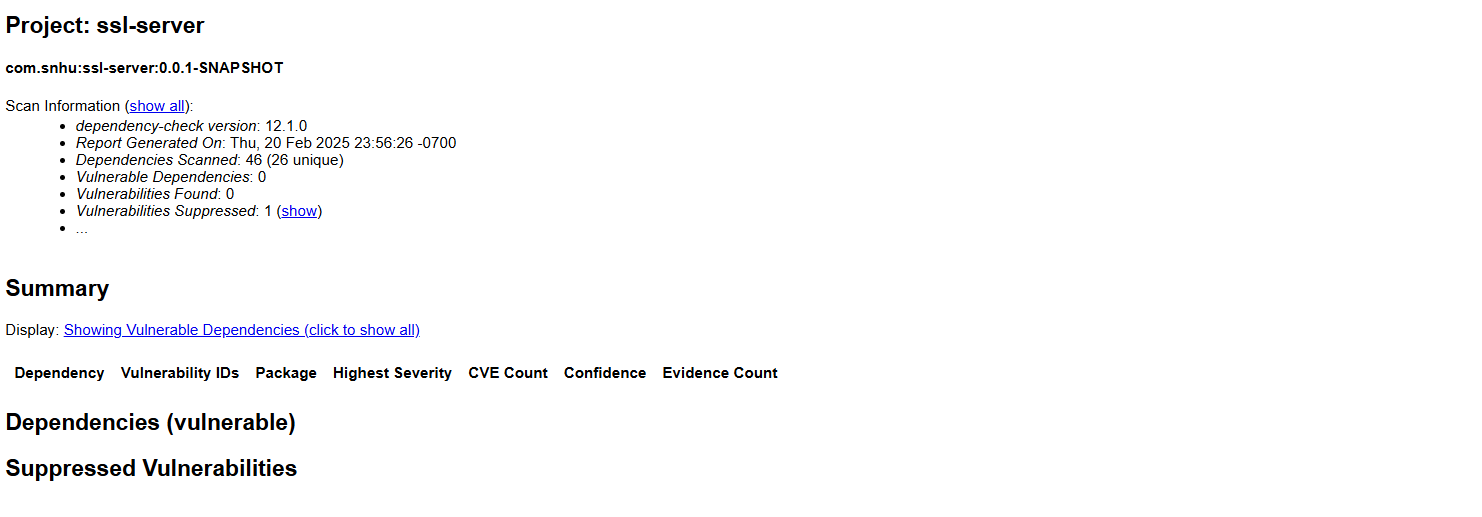
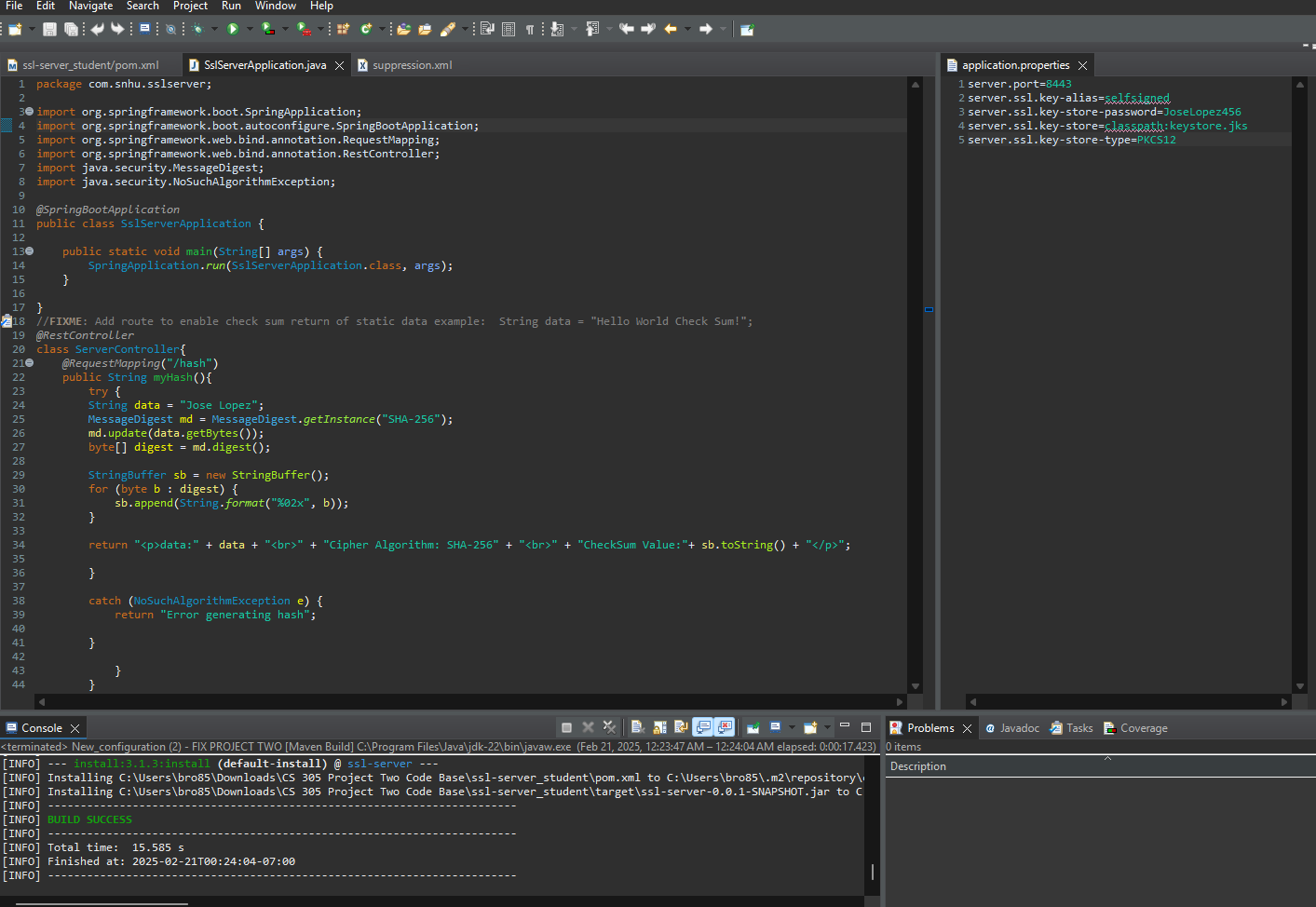
## Secure Communications

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.

A screenshot of a computer program

AI-generated content may be incorrect.

## Summary

When refactoring the code I added a REST controller to the SslServerApplication.java file to handle the requests. Inside the controller there is a method that takes the input and generates a SHA-256 hash for the string. This SHA-256 hash function is crucial for ensuring we maintain secure data. When the input is taken it transforms it into a fixed length, irreversible, never duplicated code that can’t be cracked. If there are any errors during this process an error message is returned instead. The application.properties file has also been updated to secure communications over HTTPS. This will ensure that all data transmitted is encrypted to prevent attacks. Lastly, the pom.xml file was updated to the latest most secure dependency versions and improves security by suppressing one vulnerability. All of this together create a secure system that addresses areas of security such as input validation, cryptography, secure client/server composition, error handling, code quality, and secure APIs as shown in the vulnerability assessment.

## Industry Standard Best Practices

To make sure the software application was secure I followed several best practices to address common vulnerabilities. First I incorporated the OWASP Dependency-Check plugin into the Maven build to regularly scan for vulnerabilities. This was important in ensuring that any risks or vulnerabilities are known and can be suppressed to prevent any attacks. The SSL/TLS was also built to secure communication between the server and clients. This will protect data against any man-in-the-middle attacks and is an important best practice. The use of the SHA-256 algorithm for hashing data strings also strengthened the security. I also implemented proper error handling to prevent sensitive system information from being exposed and used input validation to ensure only safe data enters the system. These practices together increase the company's overall well-being by protecting its users from ever being attacked. These vulnerabilities keep data confidential and shows that a company is committed to their users privacy and security. This not only raises the company's reputation but also builds trust with its users. The long-term stability of a company will be greater than the short-term gain from taking shortcuts.

References

* Jena, Baivab Kumar. “What Is SHA-256 Algorithm: How It Works & Applications | Simplilearn.” Simplilearn.com, 29 Aug. 2023, [www.simplilearn.com/tutorials/cyber-security-tutorial/sha-256-algorithm](http://www.simplilearn.com/tutorials/cyber-security-tutorial/sha-256-algorithm).
* Singh, S. (2003). The History of Cryptography: How the History of Codebreaking Can Be Used in the Mathemathics Classroom with Resources on a New CD-ROM. Mathematics in School, 32(1), 2–6. <https://research.ebsco.com/c/ix3dnl/viewer/pdf/krhyuxrq4j>