

# 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** | **06/22/25** | **Donovan Taylor** |  |

## 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

Donovan Taylor

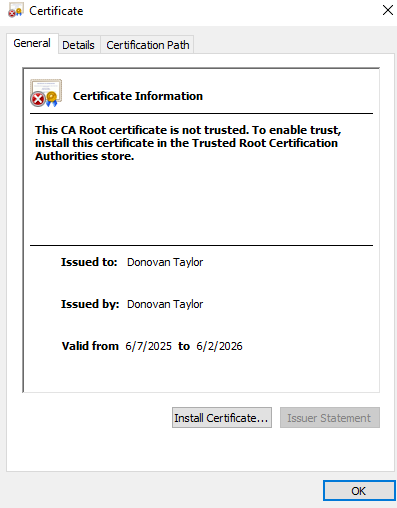
## Algorithm Cipher

SHA-256 was selected for checksum generation due to its strong security and wide acceptance. It produces a fixed 256-bit hash that helps verify data integrity without requiring encryption keys. Unlike older algorithms such as MD5 or SHA-1, SHA-256 is resistant to collision attacks and is recommended for modern applications.

SHA-256 is a one-way hash function, not an encryption method, and does not use symmetric or asymmetric keys. However, when keys are needed—for example, in HTTPS—random number generation and asymmetric key pairs (like RSA) are used for secure communication. SHA-256 remains a reliable standard for verifying data in financial and secure systems.

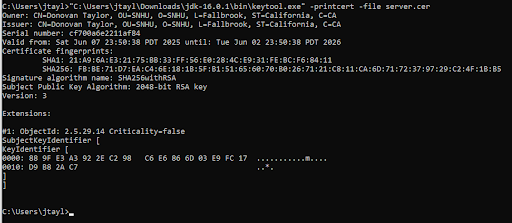
## Certificate Generation

Insert a screenshot below of the CER file.



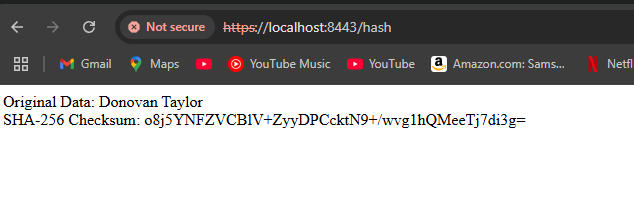
## Deploy Cipher

Insert a screenshot below of the checksum verification.



## Secure Communications

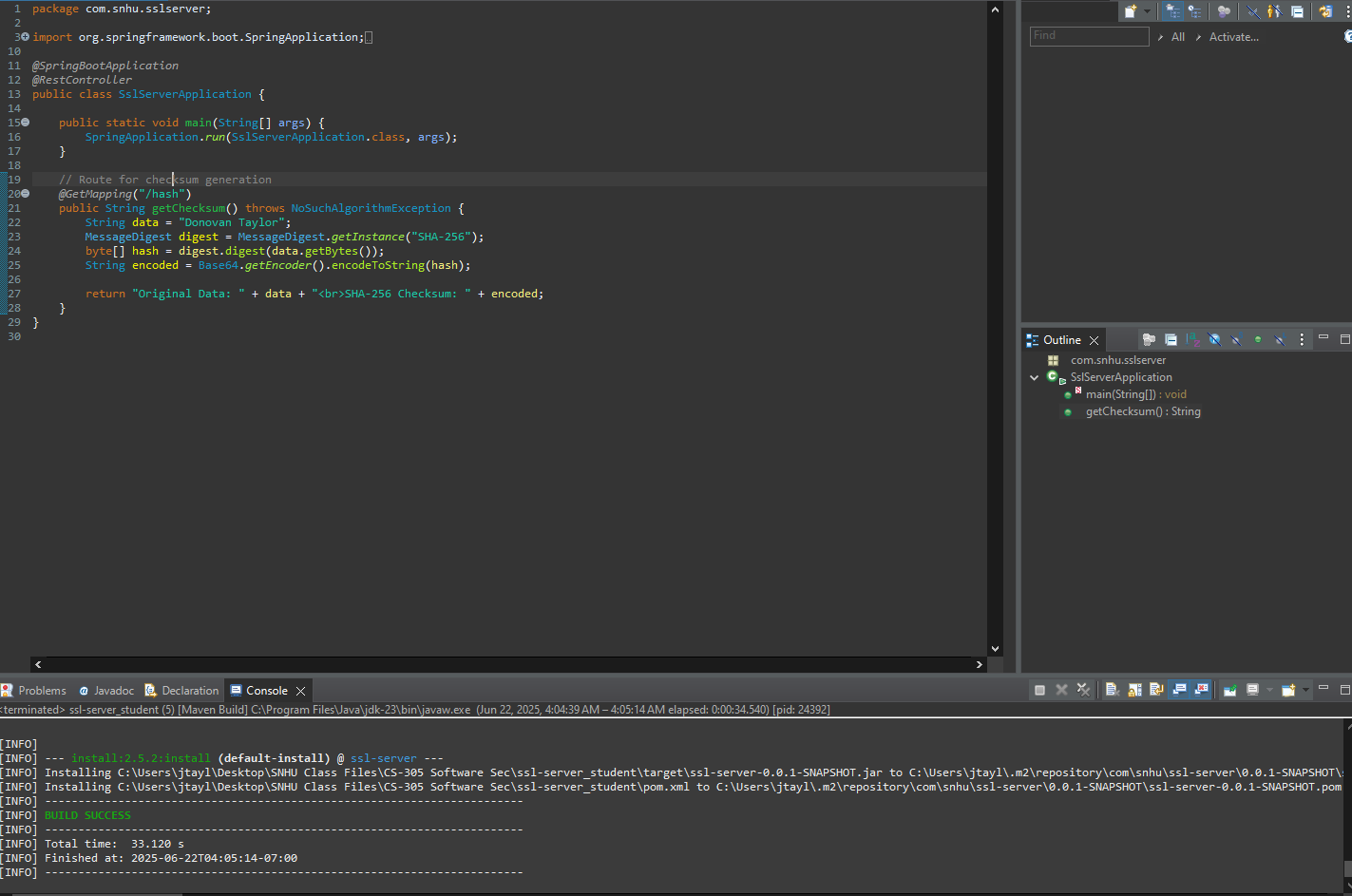
Insert a screenshot below of the web browser that shows a secure webpage.



## Secondary Testing

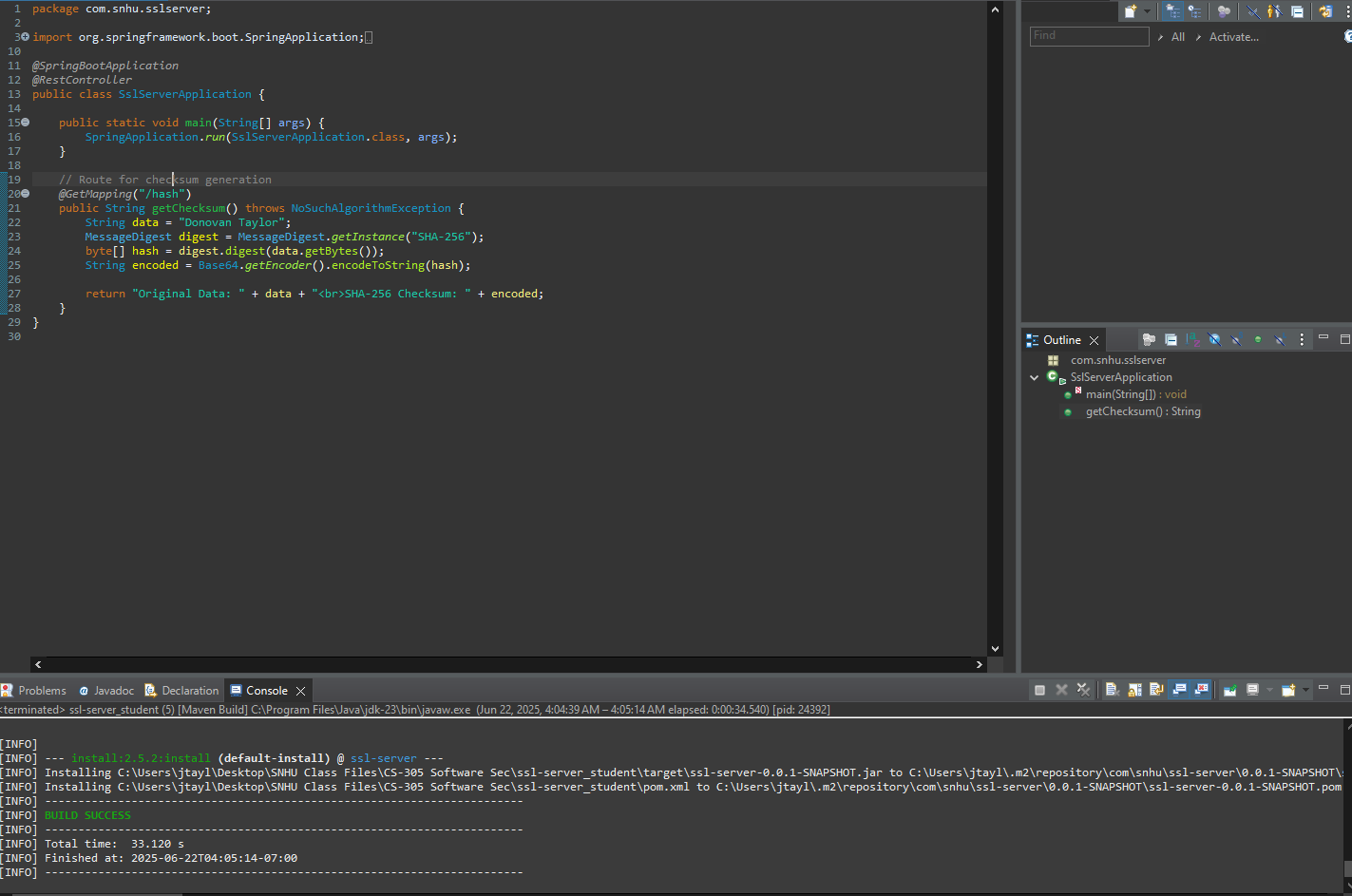
Insert screenshots below of the refactored code executed without errors and the dependency-check report.





## Functional Testing

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



## Summary

The code was refactored to implement SHA-256 hashing, enable HTTPS with a self-signed certificate, and support secure communication. These enhancements align with the security requirements outlined in the vulnerability assessment process. Areas addressed include data integrity (via checksum), confidentiality (via HTTPS), and authenticity (via certificate verification).

The development process followed a layered approach:

1. Checksum functionality ensures tamper detection.
2. HTTPS configuration ensures secure data transfer.
3. Static testing and code review verify that no new vulnerabilities were introduced.

These additions strengthen the security posture of Artemis Financial’s application.

## Industry Standard Best Practices

Industry standard secure coding practices were followed throughout this project. These include using SHA-256 for hashing, enabling TLS for encrypted communication, and avoiding deprecated or insecure cryptographic functions. Dependencies were scanned using OWASP tools to catch known vulnerabilities.

By implementing these standards, the application maintains data integrity, confidentiality, and reliability. Following secure coding practices also reduces the risk of data breaches and supports Artemis Financial’s goal of protecting sensitive client information. It also demonstrates due diligence in software development, which contributes to the organization’s overall trust and reputation in the financial sector.