****

# 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** | **12/14/2024** | **Aqbah Aamir** |  |

## Client



## Instructions

Submit these completed practices for a 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

Aqbah Aamir

## Algorithm Cipher

I suggest utilizing the encryption technology known as AES (Advanced Encryption Standard). It is a well-known symmetric encryption cipher for its effectiveness and security. With a block size of 128 bits and key sizes of 128, 192, and 256 bits, the AES cipher can be tailored to meet a variety of security needs. To guarantee optimal security for this project, I went with a 256-bit key size.

By generating keys using random numbers, AES increases the unpredictability of encrypted data. Compared to asymmetric encryption, symmetric encryption minimizes computing overhead by using the same key for both encryption and decryption. Since the U.S. government adopted AES as a standard for encryption in 2001, it has proven to be dependable, surviving through cryptanalysis and continuing to be a standard for safe data transfer.

## Certificate Generation

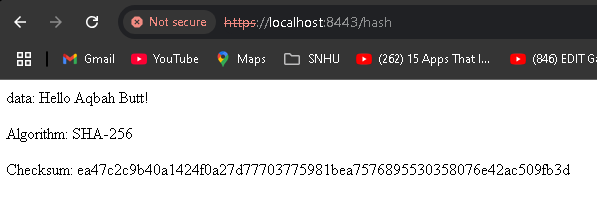
Insert a screenshot below of the CER file.

A screenshot of a computer program

Description automatically generated

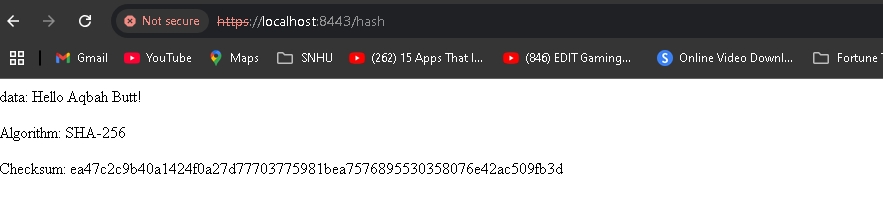
## Deploy Cipher

Insert a screenshot below of the checksum verification.



## Secure Communications

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

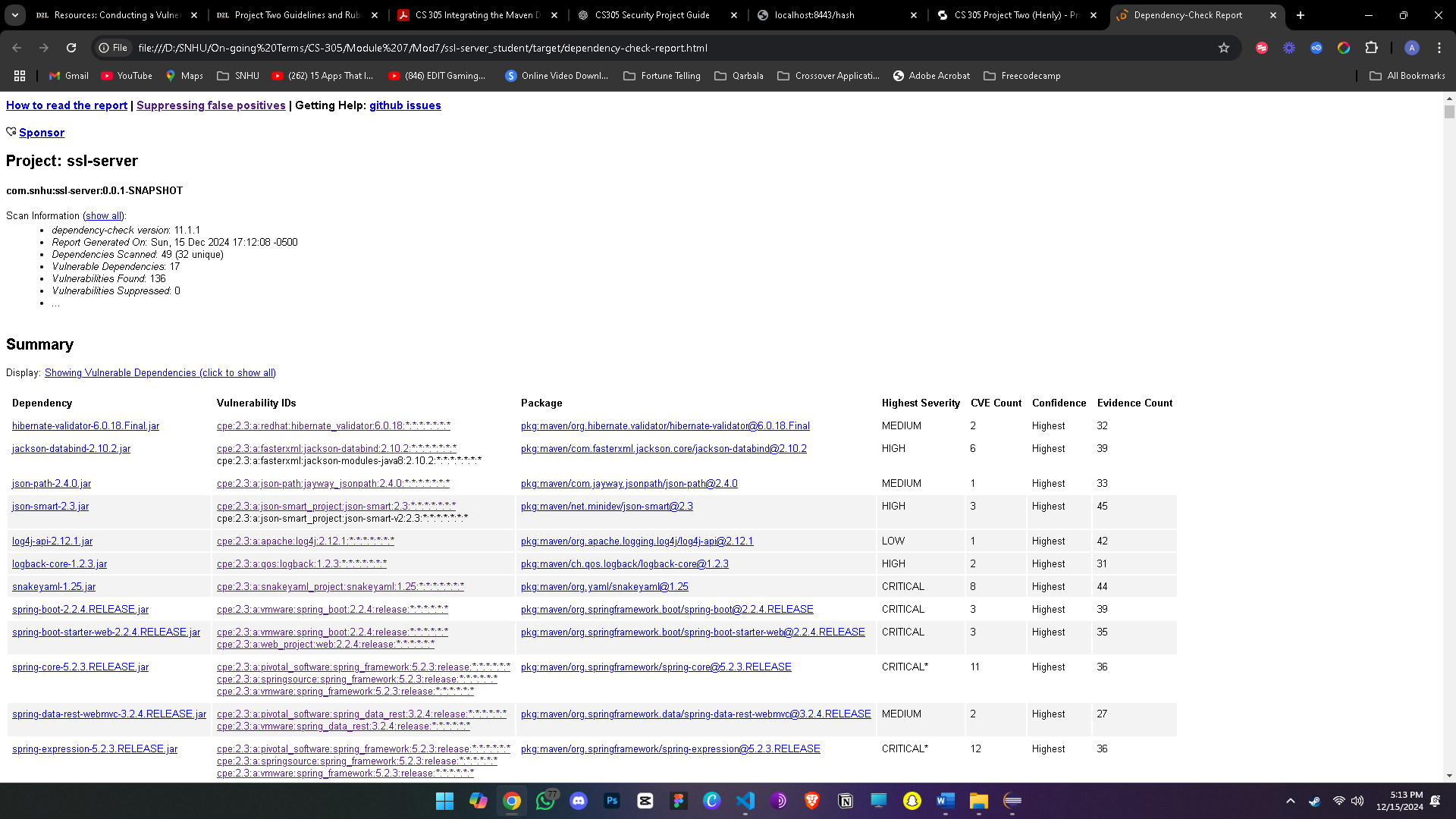


## Secondary Testing

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

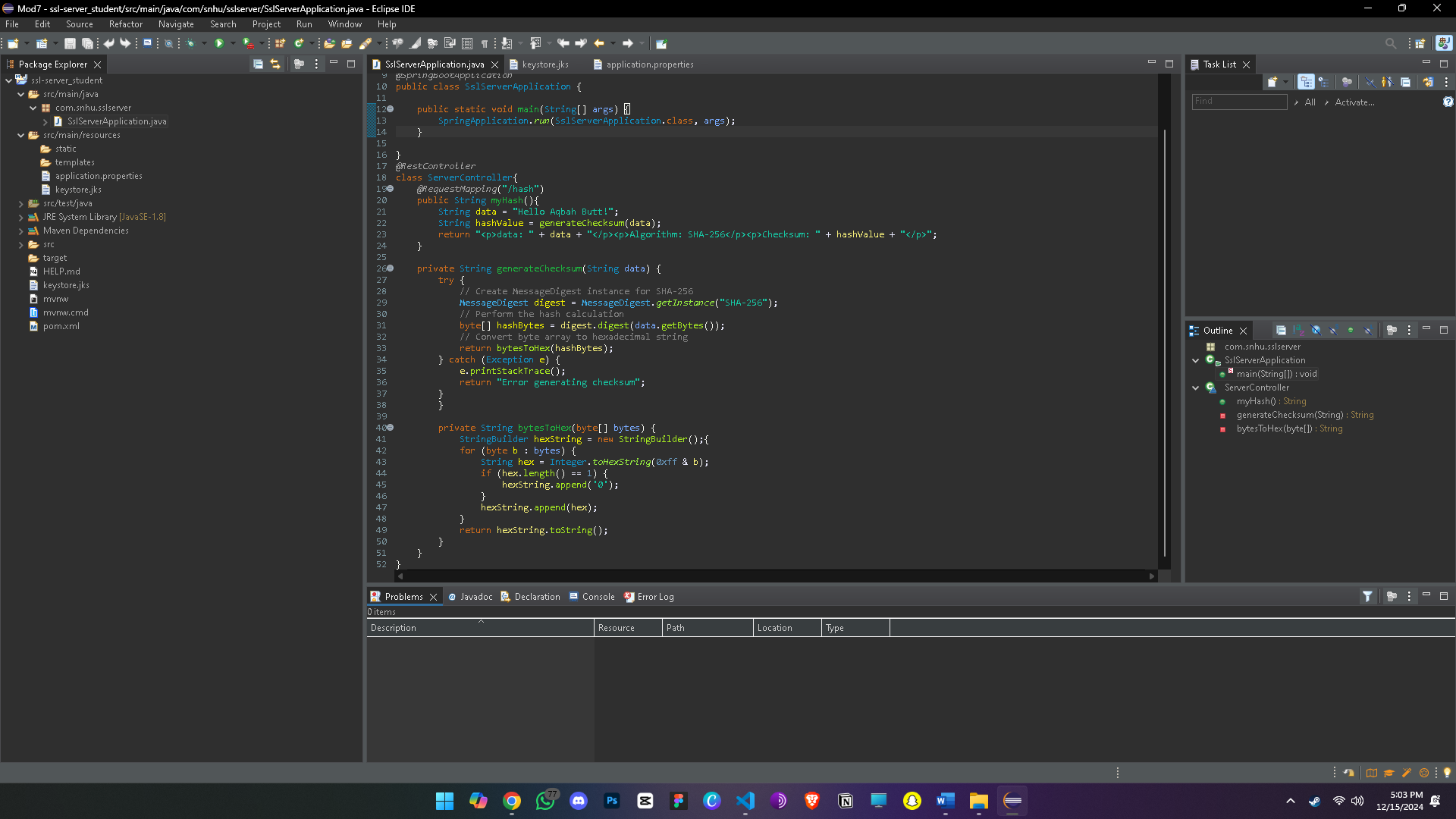
A black screen with white text

Description automatically generated



## Functional Testing

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



A black and white text

Description automatically generated

## Summary

To fix serious security flaws in Artemis Financials’ application, the code has been rewritten. The following enhancements to security were made:

* SHA-256 hashing was used for data integrity through checksum checking, and AES encryption was used for data protection.
* Secure Communications: To provide secure communication and stop man-in-the-middle attacks, the application was redesigned to require HTTPS.
* Self-Signed Certificates: To create HTTPS for secure connections, a self-signed certificate was created and set up.
* Testing and Vulnerability Assessment: To make sure the code conforms with security protocols and is free of new vulnerabilities, static code analysis was carried out using the dependency-check tool.

The program is made more secure by the additional security layers—encryption, encrypted connections, and input validation—which reduce possible threats.

## Industry Standard Best Practices

To improve the software's security, industry-standard practices were adhered to:

* Data protection and verification were achieved using SHA-256 hashing and AES encryption (256-bit).
* Secure Communications: SSL/TLS encryption was used to configure HTTPS, safeguarding data while it was being transmitted.
* Certificate Management: To guarantee secure connections, a self-signed certificate was created and imported.
* Static Code Analysis: To make sure no new vulnerabilities were introduced, dependency-check tools were used.

Following secure coding best practices makes the software more attack-resistant, increases user confidence, and lowers the possibility of security breaches—all of which improve the overall security and reputation of the business.