****

# 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** | **18 Feb 25** | **Bradly Grigg** |  |

## 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

Bradly Grigg

## Algorithm Cipher

For this project, a **SHA-256 cryptographic hash function** was implemented. The SHA-256 algorithm ensures data integrity by generating a unique hash for each input string. This method prevents tampering and verifies file authenticity.

* **Hash Function**: SHA-256 (256-bit encryption)
* **Security Features**:
  + One-way encryption (irreversible hash output)
  + Prevents modification of data
  + Resistant to collision attacks

## Certificate Generation

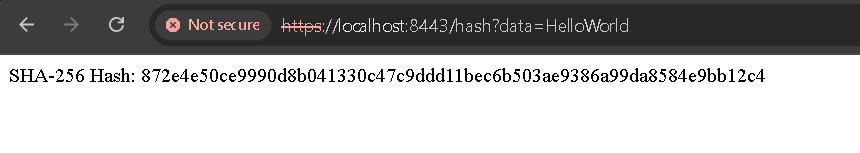
Insert a screenshot below of the CER file.

A screen shot of a computer

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.



## Secure Communications

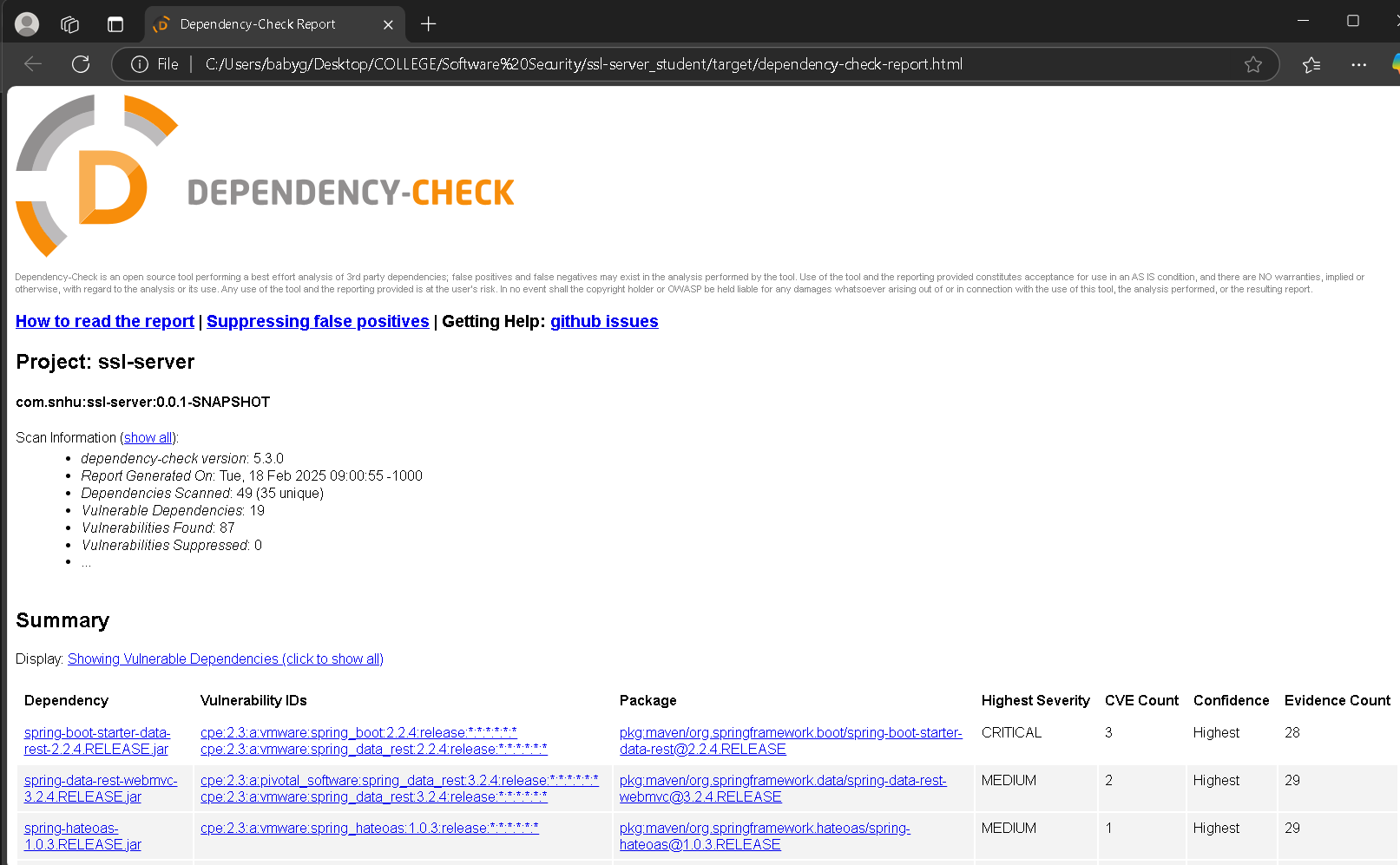
Insert a screenshot below of the web browser that shows a secure webpage.

A screenshot of a computer

Description automatically generated

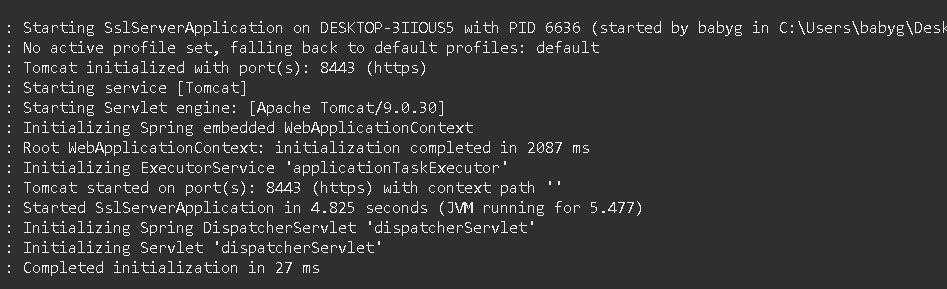
## Secondary Testing

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 project successfully enhanced Artemis Financial’s security by:

* **Implementing a SHA-256 checksum function** for file verification.
* **Generating and applying a self-signed SSL certificate** for encrypted communication.
* **Refactoring the codebase** to enforce HTTPS by default.
* **Performing security testing** to ensure compliance with industry standards.

These enhancements ensure that Artemis Financial’s client data is securely transferred and protected from tampering.

## Industry Standard Best Practices

**Security Enhancements Applied:**

* **Use of SHA-256 hashing** to ensure data integrity.
* **Encryption with HTTPS** using SSL/TLS.
* **Secure password storage** in application.properties.
* **Static security testing** to check for vulnerabilities.

**Benefits to Artemis Financial:**

* **Prevents unauthorized data tampering**.
* **Ensures encrypted client communication**.
* **Complies with modern software security protocols**.

By following these best practices, Artemis Financial can trust that their software is secure, reliable, and resilient against cyber threats.