****

# 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** | **6/23/2024** | **Brady Goodwin** |  |

## 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

Brady Goodwin

## Algorithm Cipher

Deploying SHA-384 uses a strong hash function that produces a 384-bit hash value, providing high security and collision resistance. SHA-384 is a part of the SHA2 family, with SHA-224, SHA-256, and SHA-512, all named for their bit lengths. SHA-384 is keyless and generates fixed-size hash value regardless of the size of the input data. Random numbers are used cryptographic protocols to enhance security by creating unique hashes, and provide a higher encryption integrity, unlike encryption algorithms that use keys (either symmetric or asymmetric),. Encryption began around the time of Ceasar, with the Ceasar Cypher, which was not very secure. SHA2 was Developed by the NSA in 2001 to improve upon SHA-1, and since its development, SHA-2 has demonstrated its robustness. SHA2 is widely used for securing sensitive data, digital signatures, and certificates, providing an excellent balance between security and performance.

## Certificate Generation

A computer screen with white text

Description automatically generated

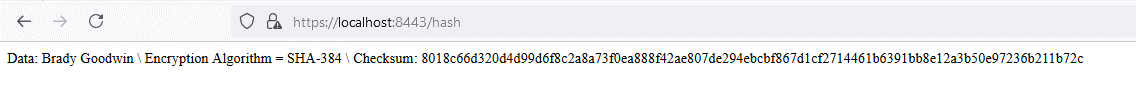
## Deploy Cipher

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

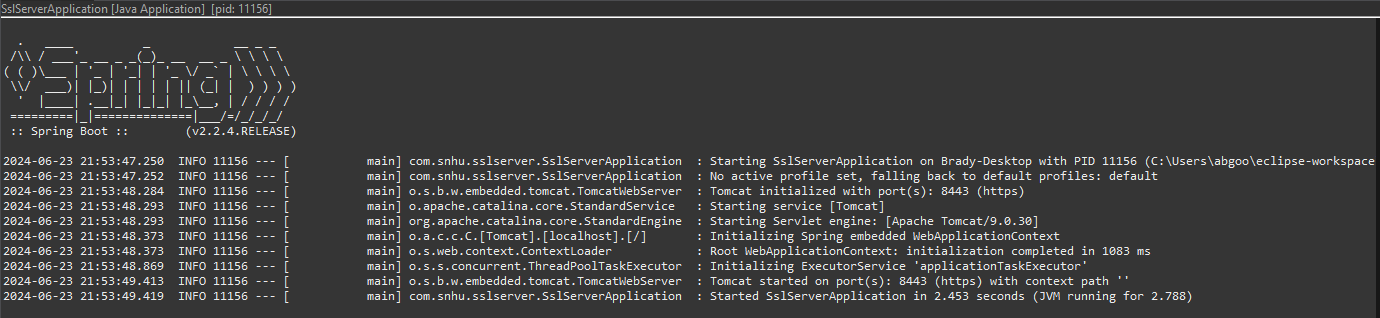
A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Functional Testing



A screenshot of a computer

Description automatically generated

## Summary

The code was refactored by adding a secure RestController to the SSLServerApplication.java file to handle the RESTful endpoint. The server controller class addresses the Vulnerability Assessment Diagram's secure coding concerns and uses SHA-384 as the hashing cipher. The Maven Dependency check version was updated from 5.3.0 to 9.2.0 for the latest software version.

## Industry Standard Best Practices

To maintain industry standard best practices;

I used input validation to prevent common attacks to the system, used a strong algorithm with 384 bits, and a self-signed certificate was used to secure a HTTPS connection