****

# 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** | **2-19-2022** | **Matt Sandoval** |  |

## 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

Matt Sandoval

## Algorithm Cipher

Global Rain recommends SHA-512 encryption for Artemis Financial. SHA-512 is a secure, complex cipher that integrates well with the java programming language. Global Rain has chosen the SHA family to align with the National Institute of Standards and Technology’s adoption of the SHA family as the federal standard (NIST, 2015).

SHA is an asymmetric encryption method, meaning it can only be used to encrypt data, not decrypt it. This type of encryption is a valuable tool for verifying data integrity after transmission. With SHA, a file of any size can be transformed into a hash, typically 512 bits or less. It is recommended that Artemis Financial use SHA to create a hash when transmitting data. Upon receiving the information, customers can confirm the validity of the data by checking if the hash they produced matches the hash generated by Artemis. If the data were intercepted and modified during transmission, the receipted hash would be different.

SHA is a hash function that generates a fixed-length output regardless of the input data size. SHA hashes are typically 224, 256, 384, and 512 bits in length. For Artemis Financial, using the 512-bit length hash is recommended, as it provides higher security against collisions. A collision occurs when two different data inputs generate the same hash output. This can pose a security threat as a bad actor could potentially alter the data and create a hash that matches the original, leading to undetected data interception. A bit can be equal to 1 or 0, meaning SHA-224 has 2^224 possible outputs versus SA-512 has 2^512. A longer hash length has more possible results, reducing the likelihood of collisions.

## Certificate Generation

Insert a screenshot below of the CER file.

Text

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

Graphical user interface, text, application, email

Description automatically generated

## Secure Communications

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

Graphical user interface, text, application

Description automatically generated

## Secondary Testing

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

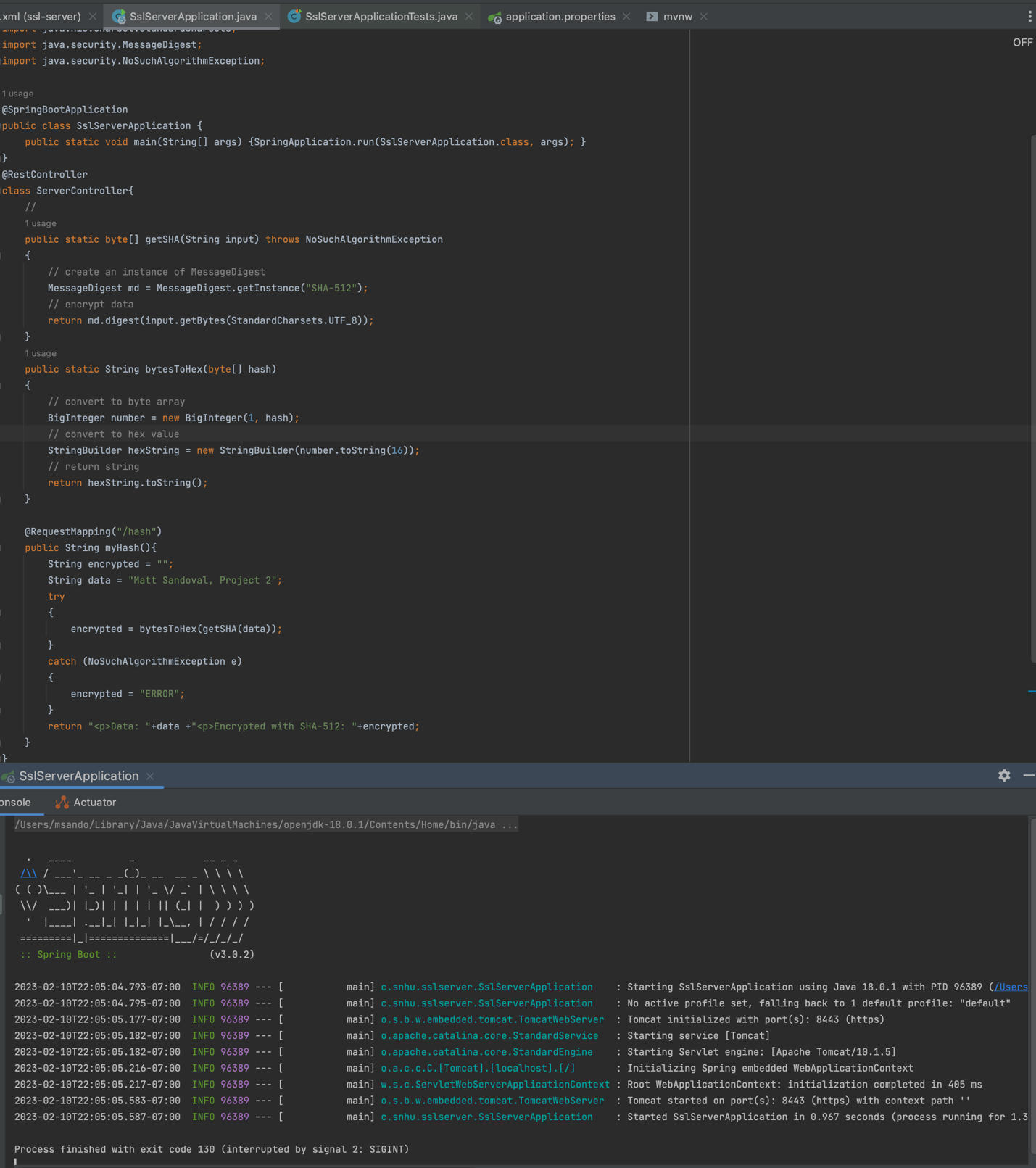
Description automatically generated

Graphical user interface, text, application, Teams

Description automatically generated

## Functional Testing

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



1. **Summary**

The first step in the Vulnerability Assessment Process Flow diagram is validating inputs. Currently, user inputs are not processed by the SslServerApplication; thus, there is no risk. The second step is securing API interactions. The Spring Framework was used to host the HTTPS site that displayed the hash function. During testing, it was found that the Spring Framework was outdated and had numerous vulnerabilities. Spring Framework was updated to the current release, 3.0.2. The Maven Dependency Vulnerability test was rerun, and it was discovered that only one vulnerability existed, snakeyaml. The purpose of this report is not to add any new vulnerabilities. Correcting snakeyaml vulnerabilities is out of the scope of this report. Artemis Financial has been notified of the security concern.

## Industry Standard Best Practices

The Vulnerability Assessment Process Flow Diagram contains industry standard best practices and was followed during the refactoring of ssl-sever\_student

1. Architecture Review
   1. The code structure was review and deemed to use best practices.
2. Input Validation
   1. The code does not allow user input, no security concern.
3. API Security
   1. Spring Framework was update to current stable version.
   2. Snakeyaml needs to be update.
4. Cryptography
   1. SHA-512 was chosen after thorough review.
5. Client / Server
   1. Ssl-server\_student is locally hosted. A self-signed SSL certificate was implemented to confirm identify.
6. Code Errors
   1. The code was checked for errors and deemed to follow best practices
7. Code Quality
   1. The code was check for unused dependency, classes, functions, and variable. The code was deemed to have only necessary code.
8. Encapsulation
   1. Encapsulation was effectively used in this project
9. Code review
   1. The code was reviewed and deemed to follow best industry practices

Work Cited

National Institute of Standard and Technology (August 2015) Secure Hash Standard (SHS). Federal Information Processing Standards Publication. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf