****

# 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** | **17JUN2023** | **Lukas Pentowski** |  |

## 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

Lukas Pentowski

## Algorithm Cipher

The client, Artemis Financial, wants to modernize their operations and ensure their software uses secure communications. In order to meet the software security requirements, I recommend the use of the SHA-256 cryptographic hash function in order to implement the required checksum verification. SHA-256 is a member of the Secure Hash Algorithms 2 function set that was created in 2001. It has a hash value of eight 32-bit words. SHA-256 is also a US Government standard with no expected depreciation anytime soon.

I recommend using SHA-256 for several reasons. The first reason is due to the low risk of a collision attack. A collision attack consists of finding two messages that share the same hash value. Once a hash value is found the encryption algorithm can be reverse engineered to allow for decryption of intercepted messages. Another reason that I recommend using SHA-256 is that it uses symmetric encryption. In using symmetric encryption both the encryption and decryption keys are the same key. The use of symmetric encryption reduces the resources required by the network to encrypt and decrypt the message. Though asymmetric encryption uses two different keys, a public key to encrypt and a private key to decrypt, and provides better access control through individual key information, it would not be required for the client’s file system.

## Certificate Generation

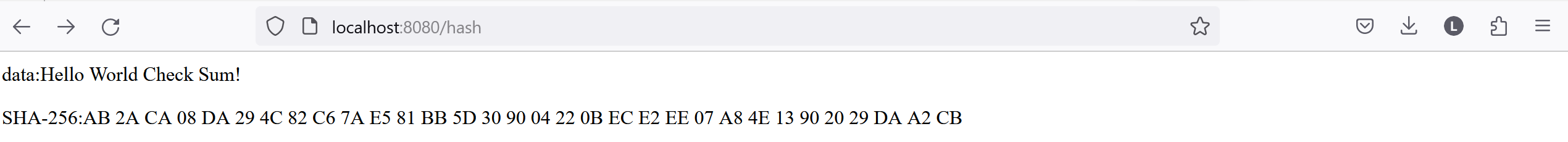
Insert a screenshot below of the CER file.

A screenshot of a certificate

Description automatically generated

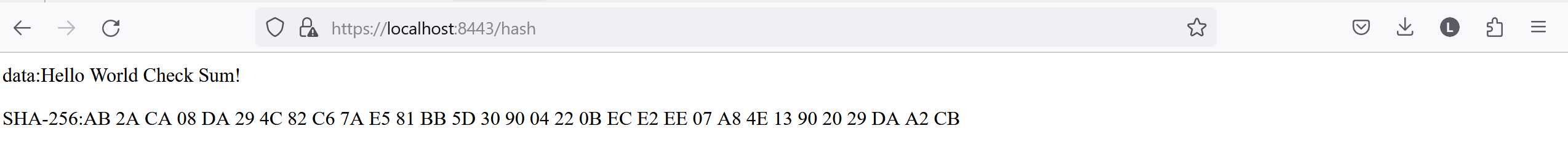
## Deploy Cipher

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 screenshot of a computer program

Description automatically generated with medium confidence

A picture containing text, screenshot, font

Description automatically generated

A screen shot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A picture containing text, font, line, screenshot

Description automatically generated

## Functional Testing

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

A screenshot of a computer program

Description automatically generated with medium confidence

## Summary

In refactoring the provided code I used the @RestController to implement Spring’s RESTful web services and HTTP requests. My ServerController class establishes “/hash” though the @RequestMapping function. In ServerController I also implemented the SHA-256 encryption algorithm. In order to have a secure HTTPS connection I configured the application.properties to use the self-signed generated key found in the keystore.jks file. Lastly, in the provided pom.xml file I updated the Spring Boot version to 3.0.7, Maven to 3.2.1, and ensured Java 1.8 was used. Updating these items ensures reduced vulnerabilities, as well as ensuring Maven is up to date for dependency checks.

## Industry Standard Best Practices

The industry standard best practices of verifying requirements, implementing encryption for sensitive data, following secure coding practices, and completing manual and automatic reviews were implemented. The use of manual code reviews informed of known outdated frameworks that required updating. Automatic reviews through the OWASP Maven Dependency check informed of any vulnerabilities that should be addressed. Verifying software requirements ensured that client security concerns were addressed and that the proper course of action was followed. It is also recommended that Artemis Financial establishes a company security policy that addresses acceptable use, change management, security training, password and security key management, and incident response.

Following the recommendation of a security policy and continuing to follow the industry standard best practices will benefit Artemis Financial by providing risk control and ensuring they are within financial sector regulatory requirements with respect to network and application information security.