****

# 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 5](#_Toc102040759)

[3. Deploy Cipher 5](#_Toc102040760)

[4. Secure Communications 5](#_Toc102040761)

[5. Secondary Testing 6](#_Toc102040762)

[6. Functional Testing 7](#_Toc102040763)

[7. Summary 8](#_Toc102040764)

[8. Industry Standard Best Practices 8](#_Toc102040765)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **2/25/2024** | **Joseph Langley** | **Final** |

## 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

Joseph Langley

## Algorithm Cipher

Given the need to modernize and enhance Artemis Financial’s web application, I recommend employing the Advanced Encryption Standard (AES) as the encryption algorithm cipher. When dealing with financial and personal information, security is paramount, and AES is the top industry-standard algorithm that’s been tested and proven. The highest encryption key for AES is 256-bit algorithm, which corresponds to the length of the encryption key. The larger the key, the greater resistance against related key or brute force attacks. AES is also symmetric algorithm which means that the same key is used to both encrypt and decrypt data transfers. AES operates through multiple rounds based on the key size making it better suited for transferring and securing larger volumes of data.

Encryption is the use of keys to transform plaintext data into ciphertext. In symmetric encryption, the same key is used to encrypt and decrypt the data, whereas in asymmetric encryption, different keys, often public keys, are used for the two sides. In encryption, random numbers are used to generate the keys. Hashing, on the other hand, takes data and produces a fixed string of characters (the hash). This is irreversible and often used for data verification or password storage. Blocks of data are encrypted block by block in through more rounds given the larger bit size. While this might slow the process slightly, with financial and personal information at stake, the security is worth the effort.

Encryption has evolved over centuries, from basic alphabet exchanges to more robust operations used in today’s software security environment. The Data Encryption Standard (DES) was widely used until AES replaced it as a more secure alternative.

## Certificate Generation

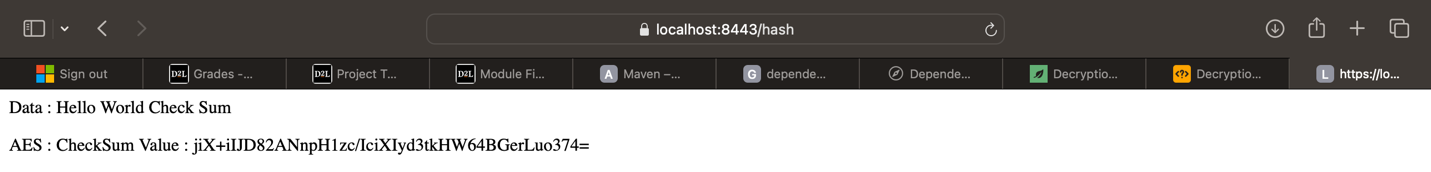
Screenshot of CER Generation

A screenshot of a computer

Description automatically generated

## Deploy Cipher

Screenshot of Check Sum verification.



## Secure Communications

Screenshot of the web browser that shows a secure webpage. (Attached: applications.properties)

A grey line on a black surface

Description automatically generated

A screen shot of a computer program

Description automatically generated

## Secondary Testing

Screenshots of the refactored code executed and the dependency-check report. (Attached: pom.xml & suppression.xml)

A screen shot of a computer

Description automatically generated

A screen shot of a computer program

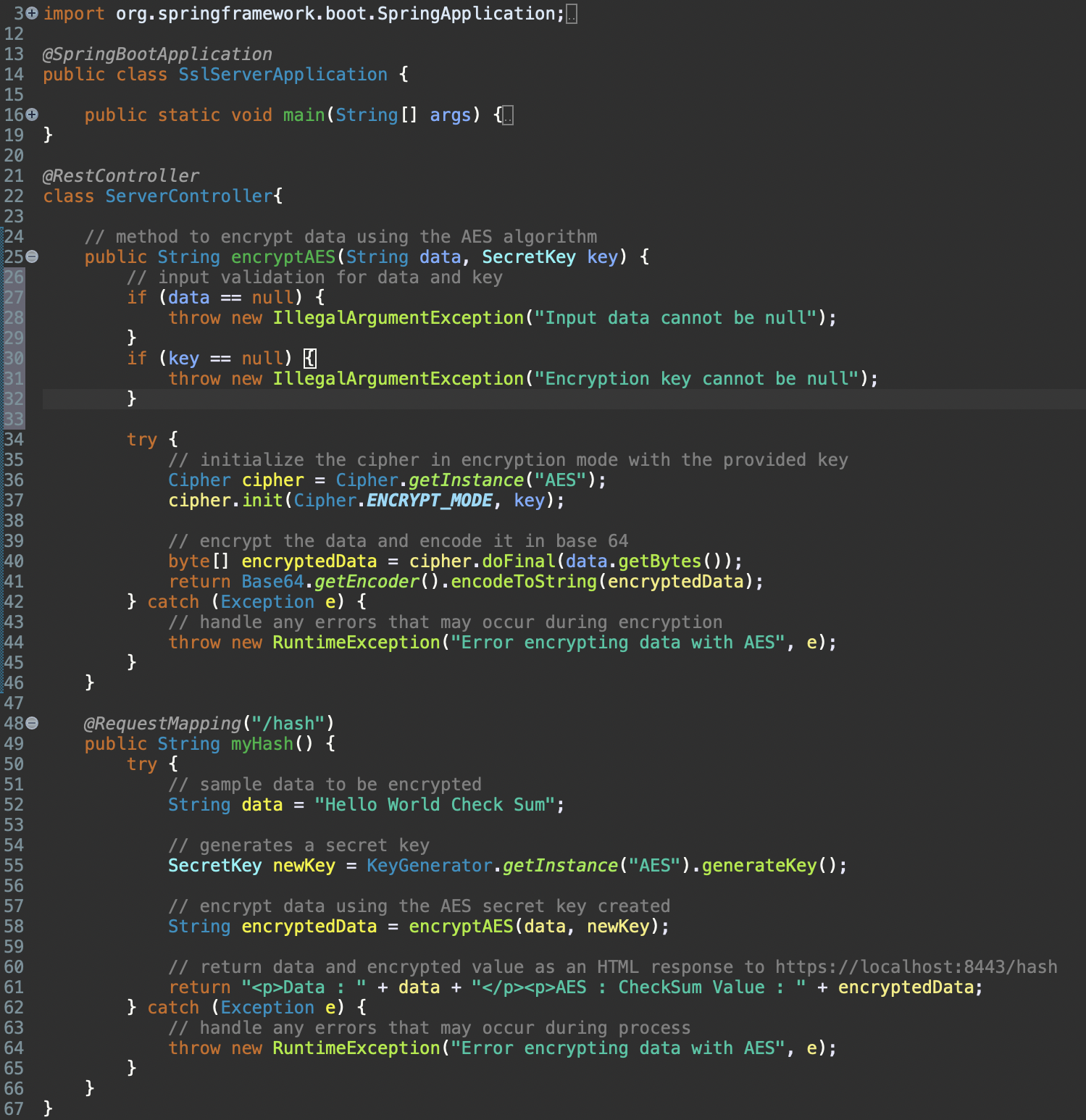
Description automatically generated

A screenshot of a computer

Description automatically generated

## Functional Testing

Screenshot of refactored code. (Attached: SslServerApplication.jar)



## Summary

Referring to the Vulnerability Assessment Process Flow Diagram (VAPFD), the refactoring process started with the creation of solid code following good input validation procedures, making input was there and couldn’t be null. The API interaction is secure using HTTPS and the self-signed certificate generation. The data is encrypted with the recommended AES encryption process by generating a secret key and using that to encrypt the data being sent. Since the Spring Boot application is configured to run over HTTPS, the communication between the client server is encrypted. The code is free of error and represents good code quality by providing comments highlighting code process, using error handling techniques, and by encapsulating information appropriately within the class. Static testing using a the OWASP dependency check tool through Maven exposes any potential vulnerabilities and encourages reviewing and updating code plug-ins to ensure top security. Combined, the input validation, secure encryption method, appropriate use of API and client/server interaction, and the static and manual review of the code showcases the layers of security involved in the application.

## Industry Standard Best Practices

The value of applying industry best practices extends far beyond the technical aspects of good development. Ensuring code is secure is paramount for the success and reputation of a company. Industry best practices include regular code review, both static, by regular dependency scanning and updating of appropriate plug-ins, and manual, ensuring the application is using the most updated encryption available. There is going to be more code added to the application that will require continuous input validation and error handling both within the new code and to check that the new and old code operate together securely. Applying these practices mitigate the risk of data breaches and unauthorized access. Regular maintenance also ensures compliance with industry regulations to safeguard the security of the data that Artemis Financial is entrusted with.

Works Cited

Rimkiené, R. (2022, August 29). *What is AES encryption and how does it work?* Cybernews. <https://cybernews.com/resources/what-is-aes-encryption/>

baeldung, W. by: (2024, January 8). *Java AES encryption and decryption*. Baeldung. <https://www.baeldung.com/java-aes-encryption-decryption>

Gupta, L. (2023, October 27). *Java AES encryption and decryption: AES-256 example*. HowToDoInJava. <https://howtodoinjava.com/java/java-security/aes-256-encryption-decryption/>