****

# 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** | **12-17-23** | **Josue Rosario** |  |

## 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

Josue Rosario

## Algorithm Cipher

Based on the scenario of Artemis Financial and upon looking through the list of possible algorithm ciphers that can be used, I would recommend Artemis Financial to implement AES, or the Advanced Encryption Standard for providing encryption security to the software and data. AES is relied upon as being one of the most secure forms of encryption and would take “billions of years to crack using a brute-force attack” with the computing power available today if the standard is implemented correctly (Oswald, 2022, para. 4). The AES algorithm will be used in order to protect the sensitive data stored within the software, as well as hiding the data in transit and ensuring that consumers can make secure financial transactions. AES offers three levels of encryption: AES-128, AES-192, and AES-256, which gives developers flexibility to balance between security and performance and while AES-256 provides more security than AES-128 and AES-192, a developer may choose to go with either of the last two if there is not enough computing power at their disposal or to make a small tradeoff between performance and security. This cipher’s hash functions take in data of different lengths and returns a fixed length string, or a hash, that makes it nearly impossible to gather information from the original input just based on the hash. The cipher’s bit levels specify the length of the encryption, so AES-256 has 256-bit encryption. Using random numbers within the process of encryption allows for an unpredictable sequence to be introduced, further hindering the efforts of attackers to interfere with the software. Symmetric keys, like AES-256, use the same key to both encrypt and decrypt data while asymmetric keys use two different keys for encrypting and decrypting the data. AES-256 encrypts data into smaller blocks and then goes into 14 rounds of encryption after. It does this by first substituting the plain text with encrypted text, shifting all of the rows but the first by one, then mixing the columns of data, and finished with more encryption with a small portion of the encryption key (Oswald 2022). AES was developed in the 1990’s and replaced the existing standard, which at the time was DES, or the Data Encryption Standard, which was becoming more and more vulnerable to brute-force attacks. Currently, encryption algorithms like AES are being used for encrypting data on hard drives, encrypting government data, encrypting electronic communications, in internet browsers, and for creating secure transactions.

## Certificate Generation

Insert a screenshot below of the CER file.

A screenshot 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.



## Secondary Testing

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

A screen shot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

## Functional Testing

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

A screen shot of a computer program

Description automatically generated

## Summary

I made changes to the code to provide a more secure application that meets company needs. I first added a RestController that functions as the hash RESTful stop for the software. Through the implementation of the ServerController and the additional code reviews that were conducted, along with the encryption that was used and the other secure coding practices that were implemented throughout the program, they all work in unison to fulfill the requirements presented by the Vulnerability Assessment Process. The use of the SHA-256 hashing cipher algorithm provides strong security for the application, use well as secure communications coming from the HTTPS protocol. A security certificate was also used along with the encryption key.

## Industry Standard Best Practices

## I used industry standard best practices throughout the entire process and this can be seen through the cipher algorithm that I decided on implementing, the updated dependency check that was used, and the security certificate that was implemented. The value of this for the company overall is that it provides general security for the company, which creates trust among the clients and allows the focus of the company to remain on what it’s actually meant on doing, which is financial advising, instead of customers having to worry about their money or information being secure.