****

# 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** | **October 15, 2022** | **Quincy Hodge** |  |

## 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

Quincy Hodge

## Algorithm Cipher

The algorithm that bests fit with the criteria that Artemis Financial would like accomplished is Secure Hashing Algorithm (SHA)-256. SHA is the industry standard for many global organizations including financial organizations such as Artemis Financial and the U.S. Government. The SHA-256 hashing generates a 512-bit can morph into a 256-bit string. Using this gives the chance of multiple hashes clashing a possibility of 1.47\*10^-29.

There are many ways to create a hash function, a user can use a random number generator, symmetric or non-symmetric keys along with many other ways. Random number generators are useful when in applications where the same sequence of numbers can be repeated easily like is simulations and modeling applications. It is important that random number generator is secure, so the hash key is suitable and not easily compromised. Symmetric keys are when the encryption and decryption key are the same key, which is secure but not as secure as non-symmetric, in the terms of an application a client and server must have the same key to be able to get access. In non-symmetric cryptography the encryption key and decryption key are separate keys instead of the same key, which adds to the security of the encryption.

## Certificate Generation

Insert a screenshot below of the CER file.

[Insert screenshots here.]

## Deploy Cipher

Insert a screenshot below of the checksum verification.

[Insert screenshots here.]

## Secure Communications

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

[Insert screenshots here.]

## Secondary Testing

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

[Insert screenshots here.]

## Functional Testing

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

[Insert screenshots here.]

## Summary

[Insert text.]

## Industry Standard Best Practices

[Insert text.]

Reference