****

# 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** | **10/19/2024** | **Ethan Mayberry** |  |

## 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

Ethan Mayberry

## Algorithm Cipher

For Artemis Financial, my recommendation would be to use a public encryption key, and a private decryption key, or Asymmetric Communication. SHA-256 is the cipher I would recommend for them, as it provides grand levels of encryption with its 256 bit keys. The possibilities for a key encryption of that size would be nearly possible to crack, leaving Artemis with safety and security in keeping their data safe for both themselves and their customers. SHA-256 also uses a number randomizer to create a checksum that verifies all files validity.

## Certificate Generation

Insert a screenshot below of the CER file.

A computer screen shot of a black screen

Description automatically generated

## 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.

* I was able to get my certificate through the command prompt. Webpage would not connect for me.

A screenshot of a certificate

Description automatically generated

## Secondary Testing

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

A screenshot of a computer

Description automatically generated

A screenshot of a computer code

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

A computer screen shot of a computer

Description automatically generated

## Summary

In looking back over the code and refactoring the necessary portions, I had determined the biggest areas of focus from our vulnerability assessment flow diagram to be as follows:

* API’s
* Code Quality
* Cryptography

In adding a RestController, it now functions as the secure controller for our RESTful API. The added serverController class help to match issues found in the flow diagram as well. I chose to go with the SHA-256 cipher, as it seemed to be the best choice for security, while maintaining the lowest probability of any collisions within the project. I had also created a self-signed certificate, and while it could not be brought up within the code, I was able to generate it and attach it to the project.

## Industry Standard Best Practices

Industry standard best practices are key for maintaining a code that is easy to have collaboration on. Making your code easy to read helps both yourself and the others who may be working with you in the future. Maintaining a neat order and structure ensures everything is easy to locate when needed, without having to sort through jumbled messes. Meaningful names are also a large part in working larger projects, as understanding each parts function becomes easier. Reading a code where your limbs are named after care bears is much harder than reading a code where each limb is named after its unique function.