****

# 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** | **12/12/22** | **Gregory Greene** |  |

## 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

Gregory Greene

## Algorithm Cipher

As a financial institution, I would recommend the SHA-256 algorithm cipher due to it’s high level of encryption. SHA-256 makes use of Java’s RNG (Random Number Generator) which helps create non-reversible verification. There are a large number of key combinations with SHA-256 which makes it harder to decrypt. I feel SHA-256 would be the most beneficial algorithm cipher to implement as Artemis Financial will most commonly be dealing with outside attacks looking to gain access to private information.

## Certificate Generation

Insert a screenshot below of the CER file.

Text

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

Graphical user interface, text, application, chat or text message

Description automatically generated

## Secure Communications

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

Graphical user interface, text, application, chat or text message

Description automatically generated

## Secondary Testing

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

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

## Functional Testing

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

Graphical user interface, text, application

Description automatically generated

## Summary

When refactoring the code for this project I chose to add a RestController for this programs RESTful stop. I also have chosen to use the SHA-256 cipher due to its low collision chance and enhanced security. Dependency checks being run periodically will help to improve security and keep necessary plug-ins up to date.

## Industry Standard Best Practices

Following industry standard best practices allows for any program to be easily managed by multiple developers or a team working on a project together. Maintaining best practices also allows for enhanced security, and clear and concise code that is easily managed.