****

# 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** | **12/11/2022** | **Trenton Walty** |  |

## 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

Trenton Walty

## Algorithm Cipher

Artemis Financial’s goal is to provide secure access and communications to their web application. They want to do this by adding a file verification step. With this being the goal, I believe recommending SHA-256 as the encryption algorithm cipher. SHA-256 protects all information in transit and makes it so no one can see what’s in transit. There are 1.1579209e+77(2256) possible hash values which make it near impossible to brute force into and on the off chance it does, it would take years to crack the cipher. Due to the cipher’s nature and configuration collisions are nearly impossible as well as this would require the hash values to be identical.

## Certificate Generation

Text

Description automatically generated

## Deploy Cipher

Graphical user interface, text, application, email

Description automatically generated

## Secure Communications

Graphical user interface, text, application

Description automatically generated

## Secondary Testing

Text

Description automatically generated

## Functional Testing

Text

Description automatically generated

## Summary

Referring to the Vulnerability Assessment Process Flow Diagram, some areas of security that need to be addressed with the code are Input Validation, Cryptography, Client/Server, and Encapsulation. With the use of SHA-256, Encapsulation has been addressed as the cipher can access the data stored and ensures it doesn’t break the data structures. This is applied by using the @RestController in the code and it knows where that data needs to done with the @RequestMapping in the code. By doing this an extra layer of security is added to the software application.

## Industry Standard Best Practices

It would be a good idea to continue to do dependency changes on a regular basisso that when new vulnerabilities arise so they can be addressed and protected from. Using SHA-256 will be a great fit for Artemis Financial and will result in them protecting their data and help guarantee their clients trust and business.