****

# 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** | **8/16/2024** | **Andre Freitas** |  |

## 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

Andre Freitas

## Algorithm Cipher

The algorithm cipher we should use for Artemis Financial is SHA-256. SHA-256 is considered the best available option for encryption due to the enormous amount of encryption numbers it can generate. It’s asymmetric, meaning that the key to encrypt is public but to decrypt is private and also means that having the encryption key does not allow someone to decrypt an encryption as well. SHA-256 also have 256 bit keys to encrypt which is what gives it it’s near uncrackable encryption method. There’s simply too many possible options on what the encryption could possibly be. SHA-256 creates a cipher using the hash function to generate the checksum of the provided message. The random numbers generated by the encryption method are what keeps the information safe.

## Certificate Generation

Insert a screenshot below of the CER file.

A computer screen with white text

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A computer screen shot of a computer

Description automatically generated

## Secure Communications

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

A screenshot of a computer

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

Description automatically generated

## Functional Testing

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

A screenshot of a computer

Description automatically generated

## Summary

For security we have added a certificate for the HTTPS usage. With this certificate we’ve shown that the site can be trusted and it is a valid website. We’ve also implemented SHA-256 which is the best method of encryption available at the moment. What we did was use the checksum to show that the user data is encrypted and randomized which will keep their info safe. Clearing vulnerabilities was also done and the system needs to be kept up to date to ensure that is stays safe.

## Industry Standard Best Practices

To keep up with the best practices of the industry, we need to keep up to date with any patches or new versions of security. We need to make sure we are on top of any better encryption methods than SHA-256 to make customer information stays secure. Vulnerabilities need to be patched up as they appear as well.