****

# 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** | **02/15/2023** | **Marisa Kuyava** |  |

## Client



## Algorithm Cipher

The Advanced Encryption Standard (AES) is the most appropriate file encryption algorithm cipher for Artemis Financial’s needs. The AES is a widely adopted symmetric encryption algorithm that was developed to take place of the Data Encryption Standard (DES) due to brute force attack vulnerabilities in the DES. AES uses a single key for both encryption and decryption and utilizes 128-bit, 192-bit, and 256-bit key lengths. The U.S. Government uses AES 192-bit and 256-bit key lengths to protect Top Secret Information. Along with being relatively easily to implement AES has quick encryption and encryption times and required less memory than DES. To ensure that AES is secure it must be implemented properly, and encryption keys must be well protected.

Hash Functions take data of any side and convert it to a compressed fixed length value, which is the hash value. The bit levels, which for AES are 128-bit, 192-bit, and 256-bit, are directly related to the encryption strength, the higher the bit keys, the stronger the encryption. There are two types of encryption, symmetric and Asymmetric. AES uses symmetric encryption, which means that the same key, which is secret, is used to both encrypt and decrypt the data. Asymmetric encryption uses one key to encrypt the data, the public key, and another private key to decrypt the data. When encrypting data utilization of random numbers is important as it decreases the chances that logic can be applied to solve the encryption.

The Advanced Encryption Standard (AES) was developed to create more secure encryption then the DES algorithm that was first published in 1975 by the Federal Register. Because AES has 128-bit, 192-bit and 256-bit, encryption it is much more secure than DES which was only 56-bit encryption. Currently AES provides excellent secure data encryption, however as technology continues to evolve it will be important to continue to evolve security for data privacy as well.

## Certificate Generation

Screenshot of the CER file.

Graphical user interface, text, application, email

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

Graphical user interface, text, application, email

Description automatically generated

## Secure Communications

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

Graphical user interface, text, application

Description automatically generated

## Secondary Testing

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

Updated POM.xml file.

Text

Description automatically generated

Refactored code executed without errors.

Text

Description automatically generated

Text

Description automatically generated

Dependency -Check

Graphical user interface, text

Description automatically generated

## Functional Testing

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

Text

Description automatically generated

## Summary

The areas of security that were address by refactoring the code are APIs, Cryptography, Code Error, Code Quality, and Input validation.

* APIs
  + RESTful API was implemented to protect from system attacks.
* Cryptography
  + Refactoring was done to include a hash function to encrypt data.
* Client/Server
  + Certificate was added so that data transfer is more secure.
* Code Error
  + Secure error handling is implemented.
* Code Quality
  + Secure coding practices and patters are used.

The primary security that I added to the software was the self-signed certificates, this allowed for HTTPS to be utilized. Additionally, a hash function was added to encrypt all data handled.

## Industry Standard Best Practices

To maintain the current security of the application it is important to continue running dependency checks on the application to help mitigate any future potential vulnerabilities. Updates should be made to the out of date versions within the program to continue to ensure data security.