****

# 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** | **6/23/2024** | **Daniel Kleiner** |  |

## 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

Dan Kleiner

## Algorithm Cipher

For the algorithm cipher I would recommend that we use the Advanced Encryption Standard (AES). This Algorithm was developed to replace the Data Encryption Standard (DES) and has been adopted by the U.S. National Institute of Standards and Technology (NIST) and the Federal Information Processing Standard (FIPS) has established that it is what should be used in banking. Our algorithm selection must also comply with other regulatory requirements set forth by the Gramm-Leach-Bliley Act (GLBA) and the Federal Financial institutions Examination Council (FFIEC). These tell us what data needs to be encrypted such as Personal Identifiable information (PII) which includes names, addresses, and social security numbers and will at Non-public Personal Information (NPI), which includes income, credit scores, collections history and family member PII.

This has been chosen by all the regulatory authorities because DES had become insecure, and hardware was developed that could crack the algorithm. The NIST held a competition to find a new encryption. The winner was a variant of the Rijndael algorithm developed by Joan Daemen and Vincent Rijmen. This algorithm uses 128-bit encryption with different key lengths, either 128, 192 and 256 bits. Encryption works by using multiple rounds of encryption. Firs the plain text is expanded so that it can be separated into 128-bit blocks and then using various matrix operations the data is encrypted, this is done for 10 rounds if using the 128-bit key but will go up to 12 and 14 for the 192 and 256 bits respectively.

Since our data will be more than 128 bits in length, we will need to use a block cipher mode to make them work together. There are a few options, but I would recommend using Galois/Counter Mode (GCM). This mode will preserve performance and add an additional later of security. The GMC has parallelized encryption and decryption, so it keeps the process fast by encrypting all blocks at the same time, it also gives us the ability to randomly access data in each block. With GCM a authentication code is produced so that the integrity of the data can be verified.

## Certificate Generation

Insert a screenshot below of the CER file.

A computer screen with white text

Description automatically generated

## Deploy Cipher

A screenshot of a computer

Description automatically generated

## Secure Communications

I was unable to get the secure connection as I am using a self-signed certificate.

.

A screenshot of a computer screen

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

## Functional Testing

A screen shot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

## Summary

When refactoring my code there were several security vulnerabilities, most of there were due to using out of date dependencies. Not keeping dependencies up to date is a security risk because this means that there are known routes of attack for those who wish to do harm. After checking the dependencies, I would go back and manually check the code to see if I found any errors that may have been created because of the updates.

The other areas where we added security was adding layers of encryption. Here we created certificates and checksums to make sure that our communications were able to be kept secure. This was done by adding the SSL certificate and checksum.

## Industry Standard Best Practices

Security is not a one-time thing when it comes to software. As fast as we can make things secure there are those who are trying to find a way to thwart our efforts to keep things secure. We need to constantly watch for new attacks that we can learn from and make sure that we keep our vulnerabilities to a minimum.

Another practice we can do to keep ourselves secure is to make sure we are changing our certificates and cipher keys at least annually, doing this will aid in keeping our data secure. Changing the certificates and cipher keys regularly will make sure that even if one gets leaked that hopefully it is not the current one thus adding another layer of security.

Following these basic principles will aid in keeping Artemis Financial secure. In today’s world where there is a new data breach all the time, keeping our customers data secure means that we can keep their trust and in turn their business.