****

# 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** | **October 10, 2023** | **Christopher Sharp** | **First Release** |

## 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

Christopher Sharp

## Algorithm Cipher

Due to the nature of the data that Artemis Financial stores for its users, it must be protected with as strong as protection that can be afforded. This requirement that the data be protected is put in place by many governmental and regulatory agencies, as set forth in the Sarbanes-Oxley Act (Evans, 2023). This includes personal identification data and that of a financial bearing. To secure this data, we have selected the AES encryption with the CTR mode. With this encryption, the key lengths can be 128,192, or 256 bits in length making it harder to brute force the provided protection (Oracle, 2007). The hash functions of the algorithm transpose the plaintext format of the data into a hexadecimal form of the same output that is indistinguishable from the original and cannot be unencrypted without the key. For the bit size, it is referring to the size of the blocks of data that are encrypted at a time. When it comes to the use of symmetric or asymmetric keys, we are referring to the types of keys that are used to encrypt and decrypt data. For symmetric keys, it means the same key is used to both encrypt and decrypt the data. For asymmetric keys, a public-private pair is used. In that one person provides their public key for someone to encrypt the data then decrypts the data with a private key. Now finally touching on the current state of the encryption algorithms is very straightforward. With the current algorithms, the ability to brute force the encryptions is very intensive, both time and resource. This comes with a big question mark though, as technology and hardware continue to increase in performance, there may come a time when these algorithms are no longer secure and need to be updated and replaced by newer, more secure algorithms.

## Certificate Generation

Insert a screenshot below of the CER file.

A screenshot of a computer

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A screenshot 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.

**Before Refactored Code:**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**After Refactored Code:**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer error

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

Firstly, when we are looking over the requirements that must be met for Artemis Financial, we can see that there are specific requirements since they are a financial institution. This includes that all personal or financial data must be encrypted as stated by the Gramm-Leach-Bliley Act (Probasco). What we have not hit on is what the software will need in the future as it continues to expand. These include:

* Input Validation: This is to protect against vulnerabilities where the user has access by injecting unforeseen types of inputs and lengths
* Secure API: The continued strengthening of the API by limiting the number of calls and the type of data that is acceptable and also implementing access controls to the different functions of the API.
* Code Error: Maintaining the error handling to not allow overflow errors that could have unintended consequences and access violations.

## Industry Standard Best Practices

By using industry standards, we were able to update the code base to not allow new vulnerabilities to become active in the code base. This included using different testing techniques and running dependency checks to maintain the code. It is of paramount importance to follow the best practices that have been set forth by the industry standards to not only protect the company's well-being but that of its user base as well. By maintaining this throughout, we can help to eliminate the potential leak of data that could lead the company to fines from regulatory compliance, distrust from the general public, and eventually the collapse of the entire company.

**References**

Evans, L. (2023, August 7). *Financial Institutions Regulatory Checklist*. Arctic Wolf. https://arcticwolf.com/resources/blog/a-simplified-regulatory-checklist-for-financial-institutions/

Probasco, L. (n.d.). *Encryption requirements for Banks & Financial Services*. Encryption Requirements for Banks & Financial Services. https://info.townsendsecurity.com/encryption-requirements-for-banks-financial-services

Oracle. (2007). *Java Security Standard Algorithm Names*. Java security standard algorithm names. https://docs.oracle.com/javase/9/docs/specs/security/standard-names.html#cipher-algorithm-names