

# 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** | **10/13/2023** | **Christian Mendoza** | **Revising Artemis financial** |

## 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

Christian Mendoza

## Algorithm Cipher

I would personally advise for the encryption algorithm cipher to use the AES cipher because it mainly supports a variety of key size and it's also the best standard in place now a days. This 128-bits and 256-bit would likely be the best choice for this project with 256-bit that would be extremely hard for hackers to attack. The symmetric keys would allow the application to encrypt data when needed and would deliver keys to the client's which would be the recipients of communications. The main use of symmetric and non-symmetric keys that would varied on what the application would mainly need to satisfy the clients needs. Symmetric keys would basically shared between the server and the client where non-symmetric keys that would include public and private keys, with the public and private keys being known only to the specific client . These keys would mainly be used to encrypt data and decrypt only when the correct key is used . A disadvantage would be if the key is lost, the data would be lost as well, since it would be difficult to decrypt the data without the key. Random generators would only be used to provide a specific identifier to transactions that might help with identifying certain events like data transfers of communications taking place. This AES would encrypt using up to 256-bit that is immune to being hacked as the number of different values that can be created is larger. In this project we would use 128-bit encryption that would provide secure communications between the client and server.

## Certificate Generation

## Deploy Cipher

## Secure Communications

## Secondary Testing

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

## Functional Testing

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

## Summary

In conclusion by refactoring the code, we discussed and also addressed the areas of security, API's, cryptography, client/server, quality, and code. We therefore created self-signed certificate and also generated keys that would be used for this project that allowed us to connect with 128-bit, AES encryption. This encryption would make it that the specific recipients would be able to read the data which will provide security to the applications communications. Secure communications would be very important in maintaining this application as un-secure communications that would cause data to be seen by hackers and they can also obtain confidential information. The results of security breach would cause, but would not be limited to loss of a trust between the applications owner and the client. Financial loss that would be due to fines, that one incurred by the government requiring secure communications, and financial loss due to resource, spent repairing all security issues. When we maintain security in the application would protect the companies product and assets and would keep their clients and business. Best practice to maintain current security is to frequently checking the code for vulnerabilities, particularly after implementing newer functions and changes current updated and before making changes to the code. When new errors or vulnerabilities are discover during development of new updates of functions, the development team would remove all vulnerabilities. For any specific vulnerabilities that can be solved, they would decide what will work best and determine if those error vulnerabilities impact any elements that this project encounters once we learned about all vunerabilities exist for any dependence, used but don't impact any element used by the code, they can suppress those on the report.

## Industry Standard Best Practices

[Insert text.]