****

# 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** | **4/15/24** | **Nicholle Caudy** | **This version is the final step. Secure communications, design, and implementation are addressed in this document.** |

## 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

Nicholle Caudy

## Algorithm Cipher

After reviewing Artemis Financials’ needs and what they hope to accomplish, the algorithm cipher I believe would be best is the AES or Advanced Encryption Standard which was established by the NIST back in 2001 and is also used by the U.S. Government for encrypting classified information. AES is used to secure data at rest and in transit. Specifically, the AES-256 algorithm cipher is used for banking and is considered the most secure. This is because of the 256-bit symmetric-key encryption that protects sensitive information and makes it extremely difficult to hack.

The hash function converts the input data into a fixed-length string of bits compressing the value. The bit level is used to determine the length of the encryption, working together these functions keep secure information protected and make it harder to be infiltrated. Using the random number function will create an unpredictable sequence of random numbers making it more difficult for an intruder to guess or figure out the keys for the encryptions and decryptions. Symmetric key encryption uses the same key to encrypt and decrypt the data in transit or at rest. Asymmetric keys, typically used for signing, have two separate keys. One key is to be kept secret and only known by the sender and the other is the public key and can be known by anyone. Keysets are also useful because you can manage the use of several different keyset types (symmetric and asymmetric keys) in your application.

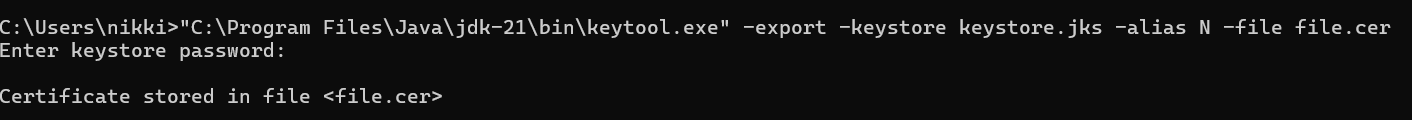
The beginning of modern cryptography started in the early 1970s with the Data Encryption Standard or DES until it was cracked in 1997. AES was approved in 2002 and became effective as a U.S. federal government standard. Encryption algorithms are still in use and even more so in today’s world with more and more information being transferred to the cloud and over the internet. The technology field is always looking for better, safer, faster, and more secure ways to protect people’s information whether in transit or being stored in a database.

## Certificate Generation

Insert a screenshot below of the CER file.

A screenshot of a computer program

Description automatically generated



A screenshot of a certificate

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A screenshot of a computer program

Description automatically generated

## Secure Communications

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

A screenshot of a computer program

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

I went ahead a suppressed false positives:

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 program

Description automatically generated

## Summary

The areas I address the most for this project using the Vulnerability Assessment Process Flow Diagram are Client/Server, Cryptography, and Code Quality. I implemented the Secure Hash Algorithm-256 or SHA-256 to protect sensitive information. This would include client names, financial plans, banking information, sensitive documents, and all other client information. I also integrated a security certificate using the Eclipse key tool to address the Client/Server aspect. This will ensure data is securely exchanged between client and server. While implementing this code it was checked manually and a dependency check was also used to ensure that the code quality was upheld. All these layers together make a more secure software application.

## Industry Standard Best Practices

Using the industry standard best practices is vital in today’s world, with the many instances of stolen information, virus installation wreaking havoc, and numerous other threats we need to do our best to prevent all these issues. We should be performing regular dependency checks to monitor vulnerabilities that may pop up with outdated versions in current use. I used the dependency check and then suppressed any false positives. A manual review of the code was completed to look for any issues. I also used an SSL certificate to ensure that the site is secure, allowing the client to have secure communication using encryption. Another way I used the industry standard best practices was by using the SHA-256 cipher algorithm to add another layer of security through hashing and encryption to help keep the information secure. This will help protect the company and its client’s private information. Ensuring the API is done properly so that all users cannot perform any functions outside of their authorization (principle of least privilege). With all the layers of security added to the Artemis Financial application it will keep the company and their client’s information secure.