****

# 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** | **4/19/2025** | **Adam Morales** |  |

## 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

Adam Morales

## Algorithm Cipher

I had initially though that AES or Advanced Encryption Standard would be the best encryption algorithm cipher since it is a very secure and has 256 bit encryption. However, SHA-256 would be a better option considering we are working with financial information. SHA also uses 256 bit encryption however, it doesn’t use symmetric key encryption. Instead of having a symmetric key, it is a one way function that generates a hash without a clear way of reversing it.

This cipher will be used to encrypt data in order to make it extremely difficult for hackers to access sensitive information such as SSNs addresses and other financial information. Some reasons that you might not want to use the most secure cipher is if it runs very poorly and you have a lot of users on your system, but to be honest, any potential slow downs are worth the safety of your users.

The purpose of cipher hash functions is to make readable data, that may be sensitive information, obfuscated so that it’s unreadable to someone who doesn’t have the key to decipher it. Bit levels are essentially just the amount of characters in said key, so the higher the bit level, the more secure your key is, or in other words, the harder it will be for someone to crack your code. Random numbers are a great way to create hard to decipher encryption since there is complete randomness that nobody knows unless you have the key. If the password to your computer was your birthdate, it would be much easier to guess, than if your password was a string of random numbers and letters that even yourself didn’t pick out.

In a symmetric system, you have one key which is the same for the encryption and the decryption. So you’d only need one key in order to view the information inside. With an asymmetric system, you need two keys in order to gain access, one for encryption and a separate one to decrypt it which ends up making it far more secure.

## Certificate Generation

Insert a screenshot below of the CER file.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a certificate

AI-generated content may be incorrect.

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A screenshot of a computer

AI-generated content may be incorrect.

## Secure Communications

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

A screenshot of a certificate

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

Still not secure despite registering my certificate

## Secondary Testing

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

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a web page

AI-generated content may be incorrect.

## Functional Testing

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

A screen shot of a computer

AI-generated content may be incorrect.

## Summary

The code has been refactored to use HTTPS over HTTP which complies with security testing protocols since websites need to be HTTPS in order to be trusted and safe to send sensitive information through. I have also changed the code to use SHA-256 encryption to encrypt incoming data into a hash that cannot be read by a human eye and makes it so incredibly difficult for a nefarious user to obtain and read. This makes it the perfect encryption method for a financial company such as Artemis Financial that deals with plenty of sensitive information

## Industry Standard Best Practices

Some industry standards are things such as input validation, which in my opinion is one of the most important, running on HTTPS as opposed to HTTP, and making sure you have good error handling and outputs. It’s also very important to just write code in a standard and cohesive way so that the team will be able to read and edit your code if needed.