****

# 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** | **06/18/2025** | **Christopher Diaz** | **Putting security first** |

## 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 Diaz

## Algorithm Cipher

In order for Artemis Financial to strengthen its software security, I would recommend implementing the integration of Advanced Encryption Standard alongside the Galois/Counter Mode of operation. Using this will allow the company to stay secure because they can encrypt data and generate authentication tags to make sure that data cannot be accessed or tampered with by outside parties easily. AES is well known and heavily used for its symmetric cipher while GCM adds a layer of security to AES by allowing authenticated encryption. This will suit the company’s needs because they will be handling private and sensitive personal information involving clients, and their own data. Additionally, with AES, the company will have the option of using different key sizes depending on how high they want their security to be. The higher the bits for key size, the harder it would be to decipher. GCM will generate a 128-bit block size specifically meant to hold the authentication tags, adding another layer of security. To absolutely maximize the security that is provided, GCM will require a unique and random 96-bit initialization vector for each encryption operation that takes place. This randomness will make it difficult for an attacker to intercept because the initialization vector will never be the same. When making the comparison between symmetric and asymmetric cryptography, symmetric encryption like AES will be notably faster because of factors like how it uses the same secret key for both encryption and decryption. It is suitable in this case for this company because this company will require encrypting financial data and client information as that data and information are provided. For asymmetric encryption, it tends to be slower, and it requires higher computational power because of factors like how it uses larger bit sizes like a 2048-bit sized encryption, but it does have its uses. For instance, it is used for digital signatures and secure key exchanges which can be valuable when dealing with the clients because of how it establishes trust. When looking back into the past, we originally used DES for encryption, but that was soon replaced by AES in 2001. Since then, AES has become the new standard as the most popular and widely used. AES is utilized in HTTPS, disk encryption, cloud storage, and a number of other areas, and when paired with GCM, it provides a superior sense of security.

## Certificate Generation

Insert a screenshot below of the CER file.

A screenshot of a certificate

AI-generated content may be incorrect.

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A black text on a white background

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 computer

AI-generated content may be incorrect.

## Secondary Testing

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

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer error

AI-generated content may be incorrect.

## Functional Testing

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

A screenshot of a computer program

AI-generated content may be incorrect.

## Summary

In order to satisfy the security needs of this company, I had to implement the generation of a SHA-256 checksum. Using SHA-256 essentially makes it very difficult for any attacker to take advantage of revealing data in transit because it is used as a way of verifying the integrity of the data that is in transit, which will ultimately allow the detection of tampering if tampering does occur. Additionally, I used HTTPS for the webpage because HTTPS provides encryption, authentication, and integrity in comparison to HTTP. This will make sure that any data in transit is protected and remains safe from attackers that may try to tamper with it. Aside from this, I also utilized a dependency checker so that I could look over any potential gaps in our security. Even beyond the dependency checker, I still have to go and manually look over code to make sure that I do not have any errors or flaws that an attacker can take advantage of and utilize.

## Industry Standard Best Practices

I followed the best practices for the industry by keeping my code secure through the use of HTTPS and encryption, and I made sure my code was secure by utilizing a dependency check alongside a manual review of my code. Throughout the process of creating my code for the ServerController class, I used SHA-256 because I know it is widely popular and heavily used in the tech industry, and I know that its reputation is very strong and trusted among the tech companies. Using HTTPS will simply help keep data protected as it is in transit on the web. As for the dependency check, I looked over the dependencies, making sure to deal with any notable dependencies that may have popped up due to the code I implemented. Aside from the dependency check, manually looking over the code is essential too since it allows me to find errors, flaws, and bugs in the code that I would have missed otherwise. Fixing these errors, flaws, and bugs will help prevent an attacker from taking advantage of such a vulnerability because it will be more difficult for the attacker to manipulate the code.