****

# 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** | **10/15/2025** | **Patrick Jovez** | **Template Created** |
| **1.1** | **10/15/2025** | **Patrick Jovez** | **Algorithm Cipher Referral** |
| **1.2** | **10/16/2025** | **Patrick Jovez** | **ChecksumUtil and HashController classes implemented**  **Self-Signed Certificate Generated** |
| **1.3** | **10/17/2025** | **Patrick Jovez** | **Deploy Cipher, Functional and Secondary Testing Completed** |
| **1.4** | **10/19/2025** | **Patrick Jovez** | **Completed Summary and Industry Standard and Best Practices** |

## 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

Patrick Jovez

## Algorithm Cipher

I recommend Artemis Financial implements AES-256-GCM. This cipher is widely adopted across many industries and is the standard of NIST. GCM will provide confidentiality and integrity through its authenticated encryption which is ideal for secure data transmission. Although AES-256 is not a hash function, it can be used along SHA-256 to produce a strong, secure framework. AES-256 would provide a high level of security for encryption while SHA-256 would handle hashing and checksums. AES-GCM utilizes securely generated initialization vectors for each of its encryptions which helps prevent attacks and uses the same key for encryption and decryption which allows for the transfer of large data easily. Asymmetric encryption is used for the protection of these keys while used in the establishment of a secure communication channel such as HTTPS. NIST had developed AES to replace the outdated DES algorithm. Because of AES’s speed, security, and strength, it has now become the global standard for data encryption.

## Certificate Generation

Insert a screenshot below of the CER file.

A screenshot of a computer program

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 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 screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Secondary testing confirmed there were no new vulnerabilities after refactoring code for SHA-256 Checksum.

## Functional Testing

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

A computer screen shot of a computer screen

AI-generated content may be incorrect.

## Summary

The web application implemented for Artemis Financial was refactored to enhance its security by integrating a checksum verification for all data transactions. We used SHA-256 as a hash algorithm and added it to our ChekcsumUtil class, tested it within our ChecksumTest class, and ran it through the HashController endpoint. We utilized HTTPS on the server to secure our communication line, while ensuring traffic encryption by generating a self-signed certificate. Before refactoring our code, we conducted a dependency check as a test base to see what vulnerabilities were produced. Upon code refactoring and implementation of additional code, we have found that there were no new vulnerabilities detected. These additions to the code add a greater level of security and integrity that would greatly benefit Artemis Financial and its clients by providing secure and authenticated data transfers. Functional testing has revealed that this application meets Artemis Financials’ security requirements and operates correctly.

## Industry Standard Best Practices

We applied Industry Standard Best Practices while implementing security standards for Artemis Financials web application. The use of SHA-256 hash ensured the integrity of data while self-signed HTTPS certificates will provide encryption for communication. This is imperative to the success of Artemis Financial as they deal with highly sensitive personal information. The implementation of code has been thoroughly tested, and results yield no new vulnerabilities after refactoring. Clients of Artemis Financial will be pleased to know that their information will be secure and confidential throughout the use of this application.