****

# Practices for Secure Software Report

Table of Contents

[Practices for Secure Software Report 1](#_Toc201359807)

[Document Revision History 3](#_Toc201359808)

[Client 3](#_Toc201359809)

[Developer 4](#_Toc201359810)

[1. Algorithm Cipher 4](#_Toc201359811)

[2. Certificate Generation 4](#_Toc201359812)

[3. Deploy Cipher 5](#_Toc201359813)

[4. Secure Communications 6](#_Toc201359814)

[5. Secondary Testing 7](#_Toc201359815)

[6. Functional Testing 9](#_Toc201359816)

[7. Summary 10](#_Toc201359817)

[8. Industry Standard Best Practices 10](#_Toc201359818)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **[Date]** | **Brighton Ulery** |  |

## Client



## Developer

Brighton Ulery

## Algorithm Cipher

As Artemis Financial continues to modernize its operations, it must also modernize its encryption standards. To ensure that file transfers remain unaltered, a verification step should be in place. These files can be verified using a hash algorithm. The encryption algorithm cipher recommended is SHA-256.

SHA-256, a member of the SHA-2 family, accepts any size input and generates a fixed 256-bit length checksum. While there are other digest, or hash value, sizes available, such as 224, 384, and 512 bits, SHA-256 is widely supported, generates sufficiently secure hash values, and performs at a reasonable speed. The SHA-256 hash function is a one-way function, meaning that the information cannot be decrypted from the checksum. It also means that a system with access to the original data can also process it through the SHA-256 algorithm and achieve the same checksum. This ensures secure transmission while also providing file validation.

## Certificate Generation

A screenshot of a computer program

AI-generated content may be incorrect.

## Deploy Cipher

A screenshot of a computer

AI-generated content may be incorrect.

## Secure Communications

A screenshot of a computer

AI-generated content may be incorrect.

## Secondary Testing

A screen shot of a computer program

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

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

AI-generated content may be incorrect.

## Functional Testing

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer screen

AI-generated content may be incorrect.

## Summary

Currently, there is no input received from a user for processing by the system. A hardcoded string is used for testing purposes. In the future, hardcoded data should be avoided and any input received by the system should be properly verified, authenticated, and sanitized. The system was subject to an automated dependency check via OWASP to ensure that there were no vulnerabilities in the system’s current API usage. A secure SHA-256 hash algorithm was implemented for the system for the generation of checksums, as well as modifications to ensure the server ran on HTTPS protocol through the use of keystores. While the certificate used in this environment is self-signed, it is essential that Artemis Financial use a third-party certificate issuer in production. The use of exceptions prevents the stack trace from being leaked, ensuring the user only receives a simple error message regarding the issue. The code was subject to a manual review to ensure that there were no inconsistencies in code quality or remaining code errors. Encapsulation is used in the form of classes, as well as public and private access modifiers for functions. This ensures that users cannot access data or functions that they are not allowed to access.

## Industry Standard Best Practices

Going forward, it is in Artemis Financial’s best interest that they maintain their cryptographic algorithms as well as review documentation on them routinely to ensure that they remain secure. They should also ensure that they are using encapsulation correctly to prevent users from accessing private data and functionality. Static and functional testing should also be routinely performed to ensure errors in their code have not cropped up nor have any APIs in use been subject to vulnerabilities. Exceptions should also be handled cleanly, so that information does not get leaked through unexpected crashes.