****

# 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** | **02/20/2025** | **Trevor Hegge** | **Version 1.0** |

## 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

Trevor Hegge

## Algorithm Cipher

For Artemis Financial's secure communications needs, I recommend implementing the SHA-256 (Secure Hash Algorithm 256-bit) encryption algorithm cipher. SHA-256 represents an optimal choice due to its robust security features and widespread industry adoption. This algorithm excels in secure file verification and digital signatures by generating a unique 256-bit (32-byte) hash value from input data of any length. The SHA-256 implementation employs a sophisticated Merkle Damgaard construction featuring 64 rounds of cryptographic operations, which provides exceptional security through its one-way hash function properties. Key technical aspects include the algorithm's use of eight 32-bit hash values initialized with specific prime number fractions, bitwise operations (AND, XOR, OR), and rotational shifts that work together to create an avalanche effect where small input changes result in significantly different hash values. This modern cryptographic solution aligns perfectly with Artemis Financial's requirements for secure data verification while maintaining high performance and reliability.

## Certificate Generation

Insert a screenshot below of the CER file.

A screenshot of a computer

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 screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

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 program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

## Summary

Through the implementation of this security enhancement project for Artemis Financial, I have successfully addressed multiple critical security areas outlined in the vulnerability assessment process flow diagram. The refactoring process focused on several key security layers: Cryptography through the implementation of SHA-256 hashing, Client/Server security via HTTPS protocol implementation, and Code Quality through the addition of input validation and secure coding practices. The application now features a robust checksum verification system that ensures data integrity during transfers, complemented by secure HTTPS communications that protect data in transit. The refactoring process included implementing defensive programming practices, proper error handling, and secure configuration management. Each security enhancement was thoroughly tested to ensure functionality while maintaining the application's core business requirements.

## Industry Standard Best Practices

In implementing security measures for Artemis Financial's application, I adhered to several industry standard best practices that significantly enhance the application's security posture. The implementation follows OWASP (Open Web Application Security Project) guidelines for secure application development, including proper certificate management, secure communication protocols, and robust cryptographic practices. The use of SHA-256 for checksums aligns with NIST (National Institute of Standards and Technology) recommendations for cryptographic hash functions. Additionally, the implementation incorporates proper error handling, input validation, and secure configuration management following CWE (Common Weakness Enumeration) guidelines.

These security implementations provide substantial value to Artemis Financial by protecting sensitive financial data, maintaining client trust, and ensuring compliance with industry regulations. The secure communication channels and data verification mechanisms help prevent unauthorized access and data tampering, directly supporting the company's mission to provide secure financial services to its clients. This comprehensive security approach not only protects against current threats but also provides a foundation for future security enhancements as new challenges emerge in the financial technology landscape.

**References:**  
  
1. OWASP (Open Web Application Security Project) - <https://owasp.org/>

2. NIST (National Institute of Standards and Technology) Cryptographic Standards –

<https://csrc.nist.gov/projects/cryptographic-standards-and-guidelines>

3. CWE (Common Weakness Enumeration) - <https://cwe.mitre.org/>