****

# 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** | **October 16, 2025** | **Adryan Knight** |  |

## 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

Adryan Knight

## Algorithm Cipher

For Artemis Financial’s refactored software, the Advanced Encryption Standard (AES) with a 256-bit key length was implemented as the encryption algorithm. AES-256 is a symmetric block cipher recognized globally as one of the most secure and efficient encryption methods. It operates on fixed-size blocks of data (128 bits per block) and uses substitution–permutation networks to transform plaintext into ciphertext, ensuring both confusion and diffusion.

The AES algorithm relies on symmetric key encryption, where the same key is used for both encryption and decryption. This method provides faster performance and stronger protection for sensitive financial data in transit and at rest. Random numbers and initialization vectors (IVs) are incorporated to prevent pattern recognition and ensure each encryption instance produces unique ciphertext, even with identical input.

Modern cryptography standards, such as FIPS 197, recommend AES-256 due to its balance of performance, scalability, and resistance to brute-force attacks. Implementing AES ensures Artemis Financial meets current security best practices and mitigates potential vulnerabilities associated with older or weaker encryption methods.

## 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.

The AES-256 cryptographic hash algorithm was deployed and tested through the application’s refactored code. This implementation ensures message integrity and authenticity by generating a checksum (hash value) for transmitted files. The checksum verification confirmed that any alteration in the transmitted data would be detected immediately.

Checksum generation was implemented using Java’s built-in MessageDigest class, leveraging the SHA-256 hashing algorithm. Each checksum was compared before and after transmission to verify data integrity.

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.

## Functional Testing

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

A screenshot of a computer

AI-generated content may be incorrect.

## Summary

The refactored code successfully mitigated previously identified vulnerabilities and complied with security testing protocols. By applying the **Vulnerability Assessment Process Flow**, the following security areas were addressed:

* **Cryptography:** AES-256 encryption replaced outdated algorithms to strengthen data confidentiality.
* **Client/Server Security:** Implemented HTTPS and certificate validation to secure communications.
* **Code Quality:** Improved error handling and input validation to prevent injection or buffer overflow attacks.
* **Encapsulation:** Enhanced class structures to prevent unauthorized data exposure.

The refactoring process focused on systematically applying multiple layers of security — encryption, certificate validation, input sanitization, and secure transport. These updates ensure Artemis Financial’s application can safely handle client financial data while complying with software security standards.

## Industry Standard Best Practices

The refactoring and encryption upgrades incorporated several **industry standard best practices** derived from **OWASP** and **NIST** guidelines, ensuring Artemis Financial’s application aligns with current cybersecurity frameworks.

* **Applied Practices:**
  + HTTPS enforced for all data transmissions.
  + AES-256 encryption for data confidentiality.
  + SHA-256 for checksum and integrity verification.
  + Secure key management within Java KeyStore.
  + Static dependency scanning to identify known vulnerabilities.
* **Value to the Organization:**  
  Applying industry best practices enhances Artemis Financial’s credibility and trustworthiness. Strong encryption and secure communication protocols protect clients’ personal and financial data from unauthorized access, mitigating reputational and financial risk. By following these standards, Artemis Financial ensures compliance with data protection regulations and fosters long-term client confidence in its secure software solutions.