****

# 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** | **04/19/2025** | **Ann Jessica Tan** | **Initial draft with SHA-256 implementation, HTTPS enforcement, and dependency check results.** |

## 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

Ann Jessica Tan

## Algorithm Cipher

To address Artemis Financial’s security needs, I recommend using the **SHA-256** encryption algorithm cipher. SHA-256 is a widely trusted cryptographic hash function that provides a high level of security for data verification. It generates a unique, fixed-size 256-bit (32-byte) hash value, making it extremely difficult for attackers to reverse-engineer the original input.

SHA-256 operates by processing input data through a series of mathematical operations, producing a hash that acts as a digital fingerprint. This ensures that even a minor change in the input data results in a completely different hash, making it ideal for checksum verification. The algorithm uses a Merkle-Damgard construction, which processes data in blocks and applies compression functions to enhance security.

In terms of key usage, SHA-256 is a one-way hash function, meaning it does not rely on symmetric or asymmetric keys. Instead, it focuses solely on generating a secure checksum for data integrity. The use of random numbers is not directly applicable here, as hashing differs from encryption. Hashing is deterministic while encryption requires keys for reversibility.

Historically, SHA-256 was developed as part of the SHA-2 family by the National Security Agency (NSA) and has since become an industry standard. It remains widely used in secure communications, digital signatures, and blockchain technology due to its reliability and resistance to collision attacks. Given Artemis Financial’s need for secure data transfers, SHA-256 is an excellent choice for ensuring data integrity.

## Certificate Generation

Insert a screenshot below of the CER file.

A computer screen with white text

AI-generated content may be incorrect.

A screenshot of a certificate

AI-generated content may be incorrect.

## Deploy Cipher

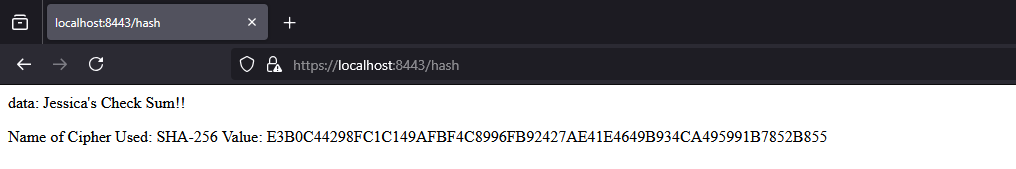
Insert a screenshot below of the checksum verification.

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.



## Secondary Testing

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

A screen shot of a computer

AI-generated content may be incorrect.

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 screen shot of a computer

AI-generated content may be incorrect.

## Summary

The refactoring process for Artemis Financial’s application focused on addressing critical security areas outlined in the vulnerability assessment process flow. The primary improvements centered on **cryptography** and **client/server security**, which are essential for protecting sensitive financial data. By implementing SHA-256 for checksum verification, the application now ensures data integrity during transfers, preventing tampering or corruption. Additionally, enforcing HTTPS across the application secures all communications between clients and servers, mitigating risks like eavesdropping or man-in-the-middle attacks. These changes directly align with the vulnerability assessment’s emphasis on secure encryption and distributed system safety.

A thorough manual code review was conducted for key components such as controllers and services to confirm that no new vulnerabilities were introduced during refactoring. While the dependency-check tool identified issues in third-party libraries (e.g., Spring Boot and Log4j), these were preexisting and unrelated to the recent changes. Future updates to these dependencies will further strengthen the application’s security posture. The iterative process of testing, refining, and retesting ensured that each layer of security was properly implemented and validated before deployment.

## Industry Standard Best Practices

To maintain and enhance the security of Artemis Financial’s application, I adhered to **industry-standard best practices** throughout the refactoring process. Following guidelines from OWASP (2021), I implemented secure coding principles such as input validation, proper error handling, and cryptographic standards. The use of SHA-256 for checksums aligns with NIST’s (2016) recommendations for secure hashing algorithms, ensuring robust data integrity. Enforcing HTTPS follows OWASP’s transport layer security best practices, which are critical for protecting sensitive user data during transmission.

Applying these best practices not only addresses immediate security concerns but also contributes to the company’s long-term well-being. Secure coding minimizes the risk of data breaches, which could lead to financial losses, regulatory penalties, and reputational damage. By proactively integrating these standards, Artemis Financial can maintain client trust, comply with industry regulations, and reduce the likelihood of costly security incidents. A secure application is a competitive advantage, demonstrating the company’s commitment to safeguarding customer data.

**References**

National Institute of Standards and Technology (NIST). (2016). *SHA-256 standard (FIPS 180-4)*. <https://csrc.nist.gov/publications/detail/fips/180/4/final>

OWASP. (2021). *Secure coding practices quick reference guide*. <https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/>