****

# 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** | **6/19/24** | **Justin Brown** |  |

## 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

Justin Brown

## Algorithm Cipher

Recommendation: SHA-256 (Secure Hash Algorithm 256-bit) is recommended for securing Artemis Financial's web application due to its robust security features, widespread adoption, and efficiency in computational performance.

Provide a brief, high-level overview of the encryption algorithm cipher:

SHA-256 is a cryptographic hash function that produces a fixed-size, 256-bit hash value. The main benefits of this cipher include high collision resistance and its strong security.

Discuss the hash functions and bit levels of the cipher:

SHA-256 operates by transforming input data into a 256-bit hash output. This process ensures that even a minor change in the input data results in a significantly different hash value. The purpose of this is to prevent collisions, ensuring that each key generated is incredibly different from each other.

Explain the use of random numbers, symmetric versus non-symmetric keys, and so on:

Random numbers are used for generating unique values like cryptographic keys. They add unpredictability, which is useful to protect against brute force attacks, as an example. Symmetric keys are used for both encryption and decryption, making them faster and more efficient for bulk data encryption. Asymmetric keys use a pair of keys for encryption and decryption, providing secure key exchange and digital signatures. Hash functions like SHA-256 ensure data integrity by generating unique fixed-size hash values from input data.

Describe the history and current state of encryption algorithms:

Early encryption methods, such as classical ciphers, can be tracked back to ancient Egypt. During this time, simple substitution ciphers were used as opposed to the more complex methods we know of today. focused on substituting or transposing plaintext characters to create ciphertext. These methods provided basic confidentiality but were susceptible to frequency analysis and other attacks.

Modern encryption algorithms, starting with the Data Encryption Standard (DES) in the 1970s, introduced more robust techniques based on mathematical functions and key management principles. DES was succeeded by the Advanced Encryption Standard (AES) in the early 2000s, which became the de facto standard for symmetric encryption due to its security and efficiency.

## Certificate Generation

## A screenshot of a computer Description automatically generated

## A screenshot of a certificate Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A screenshot of a computer code

Description automatically generated

## Secure Communications

Insert a screenshot below of the web browser that shows a secure webpage.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

## Secondary Testing

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

A screenshot of a computer program

Description automatically generated

A white background with black and white clouds

Description automatically generated

## Functional Testing

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

A screenshot of a computer

Description automatically generated

## Summary

**Refer to the vulnerability assessment process flow diagram in the Supporting Materials section. Highlight the areas of security that you addressed by refactoring the code:**

Cryptography: SHA-256 encryption was implemented to securing sensitive data, ensuring data integrity and confidentiality both during rest and transit.

Client/Server Communication: HTTPS was used to encrypt communications between clients and servers, protecting data during transmission from interception.

## Industry Standard Best Practices

**Explain how you used industry standard best practices to maintain the software application’s existing security:**

As multiple dependencies were used in this project, I made sure to update each dependency to their latest version within the pom.xml file. SHA-256 hashing was also used to protect sensitive data. This ensured data integrity and confidentiality, protecting information from unauthorized access or tampering both during transmission and storage. Implementing HTTPS also further enhanced the application's security posture and maintained the security between the client and server.

**Explain the value of applying industry standard best practices for secure coding to the company’s overall well-being:**

Applying industry standard best practices for secure coding is essential for a company's overall well-being. It builds confidence among customers, partners, and stakeholders that their sensitive information is being properly handled and protected against cybersecurity threats and potential data leaks.

By maintaining quality security measures, the company can reduce financial and reputational risks associated with data breaches cyber threats. This builds trust and credibility, enhancing customer loyalty and retention.

**Sources used:**

*Manico, J., & Detlefsen, A. (n.d.). Iron-clad Java. O’Reilly Online Learning.* [*https://learning.oreilly.com/library/view/iron-clad-java/9780071835886/ch06.html*](https://learning.oreilly.com/library/view/iron-clad-java/9780071835886/ch06.html)

*Sidhpurwala, H. (2023, January 12). A brief history of cryptography. Red Hat - We make* *open source technologies for the enterprise.* [*https://www.redhat.com/en/blog/brief-history-cryptography*](https://www.redhat.com/en/blog/brief-history-cryptography)