****

# 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** | **8/17/2024** | **Tyler Eaker** |  |

## 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

Tyler Eaker

## Algorithm Cipher

The primary goal of Artemis Financial is to provide financial programs to clients around the world. To achieve this, I recommend using the SHA-256 encryption algorithm. This cipher is highly secure and can protect information from unauthorized access. It is widely regarded as one of the most robust encryption methods, making it nearly impossible to break through brute force, a process that would take years.

SHA-256 is commonly recommended for communications with financial institutions. Its hash function is characterized by randomness, where the input value is compressed to generate a hash value, which represents the encrypted data. The encryption strength is determined by the bit levels, with 256-bit encryption offering a vast number of possible combinations, making it extremely difficult for hackers to gain unauthorized access. Randomness in the encryption process enhances unpredictability, further securing the data.

Symmetric keys are among the simplest encryption methods and are advantageous due to their faster execution time (Yedakula, K., 2019), as they require only one key. AES-256, a popular encryption standard, uses symmetric keys to encrypt plaintext. However, asymmetric keys, which require two keys, are considered more secure and are often used for internet communication.

Encryption has a long history, dating back to around 600 BC (A brief history of encryption. Thales Group, 2016). Over time, the evolution of encryption has made it possible to protect data from unauthorized access, reflecting humanity’s enduring commitment to security.

## Certificate Generation

Insert a screenshot below of the CER file.

A black screen with white text

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

I tried endlessly to get my checksum verification to work but no matter what I did I could not get the server to work.

## Secure Communications

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

Could not get the server to work so I can not successfully show the secure webpage.

## Secondary Testing

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

A screenshot of a computer

Description automatically generated

## Functional Testing

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

A screen shot of a computer program

Description automatically generated

## Summary

The key security enhancement in our application was the implementation of self-signed certificates, which enabled HTTPS usage. We also refactored the pom.xml file to address all vulnerabilities identified during the dependency check. The first step in ensuring HTTPS functionality once the application was live was properly creating the certificates. This security measure not only secures our website but also assures users that they are interacting with us, not a fraudulent site, thereby strengthening our business's credibility.

## Industry Standard Best Practices

I ensured our hashing function was working correctly and verified it with a checksum. This security measure provides peace of mind by ensuring that our customers' data is properly encrypted and difficult to retrieve, further enhancing our business's security.

Finally, I made sure all vulnerabilities were patched. This step guarantees that our business is well prepared, with all internal systems up-to-date and functioning as intended. Keeping our software and systems current is the best practice for maintaining the security of our application, protecting outdated systems from potential attacks. Additionally, enforcing the principle of least privilege is crucial. By limiting user access to only what is necessary, we protect the organization from potential internal threats, even if this approach is not fully implemented in our current system.