**Project 2: Practices for Secure Software Report**

Shawn Way

Department of Computer Science, Southern New Hampshire University

CS 305: Software Security

Dr. Vivian Lyon

February 19, 2023

****

# 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** | **2/19/2023** | **Shawn Way** | **Initial Definition** |

## 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

Shawn Way

## Algorithm Cipher

The algorithm cipher that best matches the needs of the system is AES-256-CBC for file encryption and SHA-256 for hash values. The Advanced Encryption Standard is a strong standard, well-used standard, and the 256-bit value allows for the greatest security (Rimkiene, 2022). The cipher block chaining mode, or CBC, uses a chain to incrementally encrypt each ciphertext block as it progresses through the data. This mode is the industry-standard and is the most secure option for this application (Mustafeez, 2023). AES-256-CBC uses symmetric keys and uses 256-bit values in its random number generation. Symmetric key algorithms use a set of private keys that are used to encrypt and decrypt the data. Both of these keys must be kept secret for security to be maintained. Asymmetric key algorithms use a private and public key to encrypt and decrypt information. The public key is given to the client and the private key is held secret by the organization. The private key cannot be mathematically derived from the public key, and is therefore safe to use in order to secure data. In this case, a symmetric algorithm is more secure due to the nature of the archived data and its access (Daniel, 2022). Random number generation is technically impossible for a computer due to the inherently logical nature of its software and hardware. Therefore, we have to mimic randomness by using ingenious mathematic algorithms to produce apparently random numbers. SHA-256 is adequately complex and avoids collisions, therefore it is very secure (Callaghan, 2020).

## Certificate Generation

Insert a screenshot below of the CER file.

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

Graphical user interface, text, application, website

Description automatically generated

## Secure Communications

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

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Note: The second screenshot was added to show that the certificate was added to the trusted root certification authorities and still did not result in a secure HTTPS connection.

## 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 with medium confidence

Graphical user interface, text, application, email

Description automatically generated

## Functional Testing

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

Text

Description automatically generated

On line 37, there is a try/catch block that helps secure the code in the event of a NoSuchAlgorithmException, which could be caused by an injection to the algorithm used (Shea, 2020).

## Summary

The API was secured by using RESTful communications. Cryptography was implemented by using the SHA-256 algorithm to hash the data. Client/Server security was ensured by using secure keys. The process for adding layers of security to Artemis Financials system included generating and implementing a set of keys through that would ensure that the data transfer through HTTPS remained secure, using the SHA-256 algorithm to encrypt sensitive data and keep attackers from gaining any financial information (Callaghan, 2020). A RESTful communication protocol helped maintain security by simplifying communication and client access. These layers of security are important for Artemis Financial due to the fact that they must maintain sensitive, financial data safe from attacks. This security is what their customers will depend on, and losing it would mean losing their business. It is also important from a legal standpoint that security is maintained.

## Industry Standard Best Practices

It is best practice to handle errors correctly and securely to ensure maximum security and that attacks have a minimal affect on the system. Using try/catch blocks is one the best ways to handle errors, and, in this case, they were used to ensure that incorrect encryption algorithm would result in an a handled exception. Using industry best practices is a simple way to maintain security and prevent attacks through the correct and high-quality use of code to eliminate issues before they start (Shea, 2020).

**References**

Callaghan, P. (2020, August 19). *Why you should use SHA-256 in evidence authentication*. Pagefreezer. Retrieved February 18, 2023, from <https://blog.pagefreezer.com/sha-256-benefits-evidence-authentication#:~:text=The%20Benefits%20of%20SHA%2D256&text=Collisions%20are%20incredibly%20unlikely%3A%20There,the%20exact%20same%20hash%20value>

Daniel, B. (2022, March 24). *Symmetric vs. asymmetric encryption: What's the difference?* Trusted Computing Innovator. Retrieved February 18, 2023, from <https://www.trentonsystems.com/blog/symmetric-vs-asymmetric-encryption>

Mustafeez, A. Z. (2023). *What is CBC?* Educative. Retrieved February 18, 2023, from <https://www.educative.io/answers/what-is-cbc>

Rimkiene, R. (2022, August 29). *What is AES encryption and how does it work?* Cybernews. Retrieved February 18, 2023, from <https://cybernews.com/resources/what-is-aes-encryption/>

Shea, S. (2020, August 18). *Exception handling best practices call for secure code design: TechTarget*. Security. Retrieved February 18, 2023, from <https://www.techtarget.com/searchsecurity/feature/Exception-handling-best-practices-call-for-secure-code-design>