****

# 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** | **04/10/2023** | **Emily Wood** | **Final** |

## 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

Emily Wood

## Algorithm Cipher

Based on their need to encrypt static archive files, the AES using the Rjindael cipher would be my recommendation. AES is a block cipher using symmetric encryption. Symmetric encryption uses a single key for both encryption and decryption. It was originally created as a replacement for DES within the United States government. AES has become the most popular algorithm because of its successful use within the US government. So far, only theoretical attacks have been proposed by researchers. One of which is quantum computing which has never been realized. While AES-256 would be considered the most secure, it requires a lot of computing power and cost. AES-128 is considered sufficient for commercial use.

I chose to use the Secure Hash Algorithm (SHA-256) as it was listed under the MessageDigest section of the Java Security Standard Algorithm Names provided by Oracle.

The organization requests a cryptographic hash function that is not susceptible to collisions. SHA-256 is an industry standard and widely used. There are 2256 possible hash values which makes it almost impossible to have a collision of the exact same hash value (Callaghan, 2020).

## Certificate Generation

Insert a screenshot below of the CER file.

Graphical user interface, text, application, email

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

Graphical user interface, text, application, email

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

## Secondary Testing

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

Text

Description automatically generated

Graphical user interface, text

Description automatically generated



## Functional Testing

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

Graphical user interface, text, application

Description automatically generated

## Summary

Input Validation – not yet addressed. So far everything is hard coded, and this will need to be changed.

APIs – Updated so a secure HTTPS could be used. Ensure all dependencies are up to date to ensure old vulnerabilities aren’t available to security breaches.

Cryptography – Implemented a SHA-256 algorithm cipher.

Client/Server – Client and server are currently the same when displaying the checksum. Will need to be addressed in the future.

Code Error – errors were present prior to implementing the refactored code. No errors exist.

Code Quality – Code quality looks good, and comments were used for clarity.

## Industry Standard Best Practices

Ensure all dependencies are in their latest version to remove previous vulnerabilities. Ensure code has comments so that future developers can understand and keep the software up to date. Ensure all licenses and certificates are obtained. Use the best available cryptography algorithms within the scope of work.

**References:**

Callaghan, P. (2020, August 19). *Why you should use SHA-256 in evidence authentication*. Pagefreezer Blog. Retrieved April 10, 2023, from https://blog.pagefreezer.com/sha-256-benefits-evidence-authentication#:~:text=Predictably%2C%20these%20are%20also%20the,SHA%2D1%20have%20known%20vulnerabilities.