****

# 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** | **6/24/2024** | **Andrew Park** |  |

## 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

Andrew Park

## Algorithm Cipher

An algorithm cipher is a method used in encryption/decryption of plain text into cipher text. Commonly used in security applications regarding data in transit (being sent across routers on the internet) and data at rest (stored in servers that are targeted by threat actors). Specifically ciphers are used in encryption and decryption using either symmetric (meaning both processes use the same keys) or asymmetric (meaning there are two different keys both used for encrypting and decrypting). Cryptography began before World War II and was used by German forces through the Enigma cipher which used a symmetric key consisting of configuration settings for the hardware used to scramble each plain text character into a cipher text character. Currently, algorithm ciphers are used to keep communications between host clients and servers/services safe. In this project, the SHA-256 cryptographic hash function was used to create a hash value for digital verification of the plaintext. This method ensures that that data has not been tampered with.

## Certificate Generation

Insert a screenshot below of the CER file.

A computer screen shot of a computer program

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A screenshot of a computer

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 generated

## Secondary Testing

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

A screen shot of a computer program

Description automatically generated

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 screen

Description automatically generated

## Summary

The areas of security addressed in the refactored code are cryptography, secure error handling, and input validation. The code uses secure algorithms to hash out the data provided. At this stage, the input is validated only because the data is trusted for testing. Furthermore, secure error handling has been created to ensure that no out of bounds input is attempted. On top of focusing on these areas of security, static testing has been done on the dependencies to ensure that the correct versions are being used.

## Industry Standard Best Practices

This project follows secure cryptographic algorithms, error handling, input validation and dependency management. These best practices ensure improper data is not ingested into the application that could cause harm. Furthermore, data in transit is signed and encrypted to ensure that it is not tampered or viewed by unauthorized persons. Using these standards and creating secure communications is important for all companies who utilize the internet and reach customers remotely. Handling their payment data and personal information is critical for daily business transactions.