****

# 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 5](#_Toc102040763)

[7. Summary 5](#_Toc102040764)

[8. Industry Standard Best Practices 5](#_Toc102040765)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **06/23/2024** | **Ariel Foor** |  |

## 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

Ariel Foor

## Algorithm Cipher

For this project I recommend that we use the PBE with SHA256 and AES. AES with its 256-bit encryption is the strongest on the market and is typically used for financials and the military. SHA also boasts of having one of the lowest collision rates of encryption methods. With the two it can enable the user to have a secure connection since AES will encrypt everything and sha will hash things, lastly PBE will have the user enter a password, so it stays secure.

## Certificate Generation

Insert a screenshot below of the CER file.

A screenshot of a certificate

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A black and white text

Description automatically generated

## Secure Communications

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

I cant seem to get it to say that my connection is secure, and I cant add the certificate I created to avoid the issue.

## 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

A screenshot of a computer

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

The security concerns were cryptography, client/server, code error, and code quality. These were addressed with the refactored code as the hash function was implemented which covers cryptography and code quality. Code error was address by having a try and catch function if the code did not function as intended. Lastly we also implemented a certificate so that the connection would be secure covering the client server. We added in the certificate to the application properties to enhance security.

## Industry Standard Best Practices

We added to the security and did not remove anything and ensured that the cryptography we used did not have any collisions to the current security. Ensuring security is important to all companies especially those with sensitive records. When a company can be trusted by being secure, it allows the company to gain more business and ensure they are trusted.