****

# 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** | **08/06/2023** | **Justin Hancock** | **Version 1.0** |

## 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

Justin Hancock

## Algorithm Cipher

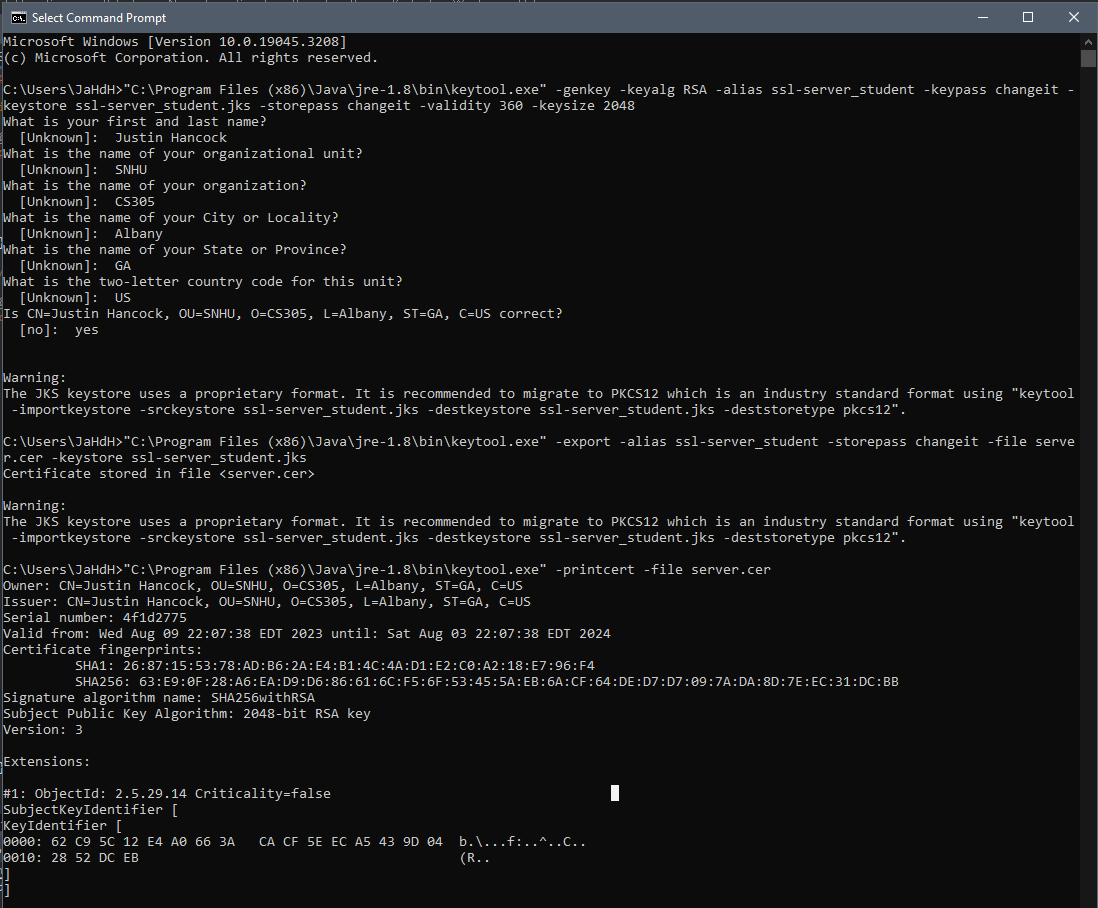
*Recommend an appropriate encryption algorithm cipher to deploy, given the security vulnerabilities, and justify your reasoning. Review the scenario and the supporting materials to support your recommendation. In your practices for secure software report, be sure to address the following:*

1. *Provide a brief, high-level overview of the encryption algorithm cipher.*
2. *Discuss the hash functions and bit levels of the cipher.*
3. *Explain the use of random numbers, symmetric versus non-symmetric keys, and so on.*
4. *Describe the history and current state of encryption algorithms.*

Artemis Financial is looking for recommendations for additional security for their web application to further secure communications. I believe that encryption would be the best recommendation for more developed security. It is safe to predict that the most likely tactic to attempt at gaining information would be accessing the financial information of the clients for this company, and encryption will help to overcome this vulnerability. To give an overview picture of how this works to prevent unwanted access, encryption uses public and private keys to gain access to information. In this case, we will use asymmetric communication which means that the public key is used for encryption, but the private key is used for decryption. So, someone will only be able to view the information if they have the proper key. These keys aren’t like the keys that you use for the lock on your door either; they are made up of a 256-bit code which is the best on the current market. SHA-256 is the standard encryption process for governments and financial agencies around the world. Breaking this down a little simpler, SHA-256 basically uses a password that is the length of 256 characters which means there are 8.57817775 x 10^506 possibilities of combinations for the password.

## Certificate Generation

Insert a screenshot below of the CER file.



## 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

A screenshot of a certificate

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 program

Description automatically generated

## Summary

Refactoring the code addresses input validation, APIs, code errors, code quality, and encapsulation. It has allowed me to add a secure controller for the program hash. This controller matches the problems that are viewed on the vulnerability assessment flow chart diagram. Maintaining the security of the program can be done by completing dependency checks a few times a month to stay up to date on potential vulnerabilities. With the different layers built into this program, Artemis Financial and their client’s information will be safe and secure.

## Industry Standard Best Practices

Using industry standard best practices has become a natural phenomenon when programming. It is becoming second nature to follow the way we have been taught to write our programs. By using in line comments, consistent naming conventions, clear spacing, etc., it allows for any other programmer to go into the details of the code and understand what is being written and for what purpose it serves. Industry standard best practices just help with communication and inclusivity to running great programs. Ultimately, our job is to ensure the client that their information along with their client’s information will be secure; that is why using these practices holds so much value.