****

# 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** | **08-18-2024** | **Rodrey McCoin** |  |

## 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

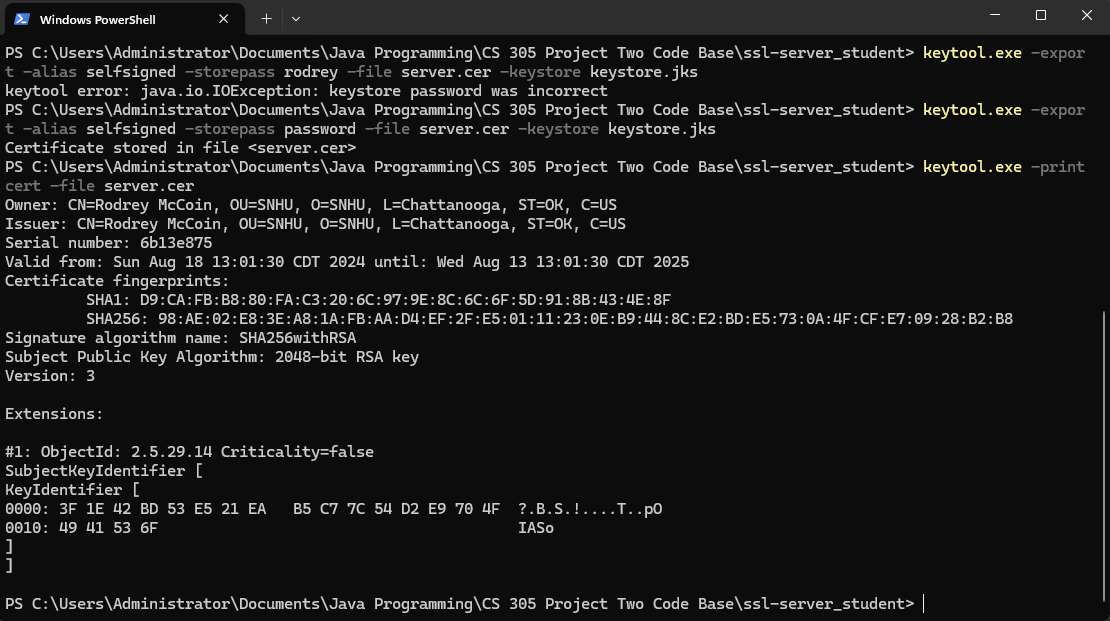
Rodrey McCoin

## Algorithm Cipher

Artemis Financials’ objective is to modernize their custom software so that they can provide their clients with the most current and effective software security for their data and financial information. The encryption algorithm cipher that I recommend based on the security vulnerabilities provided is Advanced Encryption Standard (AES). This cipher has the ability to secure data long term, provides immunity towards certain kinds of attacks, and provides confidentiality and authenticity. AES is well known and will greatly increase compatibility and minimize the risk in implementation. Because AES is a symmetric key cipher, both the sending and receiving end must have access to the same key, to encrypt and decrypt the data. The risk of this is that the key could be compromised or improperly secured. The recommendation I am proposing is AES with 256 bit-key in GCM mode. This will have a Hash function which employs cryptographic function of using hash function this will be integrated with SHA246 for checking integrity of data. It will confirm that 256-bit key size is secure against one kind of attack known as brute force. This will also utilize random numbers which are known as initialization vectors (IVs) and keys. AES will employ the same key for encryption and decryption processes. This will make it easier to manage keys for file storage. AES replaced Data Encryption Standard (DES) and has been widely used since its introduction. AES can be seen used in government and military applications as well as within businesses in the highest regulated industries.

## 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 certificate

Description automatically generated

A computer screen with text on it

Description automatically generated

A screen shot of a computer

Description automatically generated

## Secondary Testing

A screen shot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

## Functional Testing

A screenshot of a computer program

Description automatically generated

## Summary

In the application I have refactored my code to work with the SHA-256 hashing cipher. Using the SHA-256 provides more security and decreases the chances of collisions. In the pom.xml I have added the latest version of the dependency check and refactored the file, so that all vulnerabilities found during the dependency check have been fixed. I have also enabled the usage of HTTPS which increases the security of the transfer of data.

## Industry Standard Best Practices

## The best practice to maintain the software application’s existing security would be to conduct monthly dependency checks and to make sure that the plugins in the pom.xml file stay up to date with the latest version. Monthly dependency checks would help to make sure that the software is staying up to date on potential vulnerabilities, so that they can be prevented before happening. Another way to prevent attacks is to prevent them within the group. To do this we must provide users with access only to what they need, rather than access to everything.