****

# CS 305 Project Two

**Practices for Secure Software Report**

Table of Contents

[Document Revision History 3](#_Toc33111302)

[Client 3](#_Toc33111303)

[Instructions 3](#_Toc33111304)

[Developer 4](#_Toc33111305)

[1. Algorithm Cipher 4](#_Toc33111306)

[2. Certificate Generation 4](#_Toc33111307)

[3. Deploy Cipher 4](#_Toc33111308)

[4. Secure Communications 4](#_Toc33111309)

[5. Secondary Testing 4](#_Toc33111310)

[6. Functional Testing 5](#_Toc33111311)

[7. Summary 5](#_Toc33111312)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **12/12/2021** | **John Sporn** |  |

## Client



## Instructions

Deliver this completed Practices for Secure Software Report documenting your process for writing secure communications and refactoring code that complies with software security testing protocols.

Respond to the steps outlined below and replace the bracketed text with your findings in your own words. If you choose to include images or supporting materials, be sure to insert them throughout.

## Developer

John Sporn

## 1. Algorithm Cipher

Determine an appropriate encryption algorithm cipher to deploy given the security vulnerabilities, justifying your reasoning. Be sure to address the following:

To encrypt Artemis Financials archive files, we will use the number one standard algorithm cipher in the United States. The Advanced Encryption Standard (AES) is the most trusted algorithm cipher by the U.S. Government as well as many other organizations. AES is the most secure and one of the fastest algorithm ciphers. The government uses it to protect top secret information. Although, no encryption system is 100% secure, AES has yet to be cracked. It is practically impenetrable when using brute force methods which is when hackers use trial and error to crack passwords and encryption keys. AES is essential for U.S. government security, cybersecurity, and electronic data protection.

The AES cipher algorithm works by taking plain text and converting it into cipher text. The cipher text is made up of random characters. You need a special key to decrypt it. There is one secret key that is used to cipher and decipher the information which is called symmetric key encryption. Using this algorithm, we can design the software to secure data files on the computer system from unauthorized users. The data file will be encrypted with a specified password. As I stated above that same specified password which will be our secret key will also be used to decrypt the ciphered data file and then we will have access to the actual data file.

There are some advantages and disadvantages to consider. A disadvantage is that it can be difficult to implement with software because it is very complex so performance could be an issue. Another disadvantage is that every block is encrypted in the same way because it is a symmetric key encryption which means it is a one-way system. An advantage is that it is implemented in both hardware and software which makes it more secure. Uses higher length key sizes for encryption which provides more protection against hackers. I believe that the advantages heavily outweigh the risks especially since this cipher algorithm is the most trusted and used around the entire world.

The hash function that AES uses is called AES-hash. It is a secure hash function that takes an arbitrary bit string as input and returns a fixed length string as output. It checks to see if the data has been tampered with. This function has great performance in 256-bit key, 256-bit block mode. The file is padded to make its length be an even multiple of the block size and include a length encoding.

The AES cipher algorithm is the best option for Artemis Financial to use for encrypting their archive files. It is the fastest and most trusted cipher algorithm for encrypting files. As well as the most used by the U.S. government and most of the world. Although it has some disadvantages, they aren’t much of a threat and are heavily outweighed by the advantages of the algorithm.

## 2. Certificate Generation

Generate appropriate self-signed certificates using the Java Keytool, which is used through the command line.

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

## 3. Deploy Cipher

Refactor the code and use security libraries to deploy and implement the encryption algorithm cipher to the software application. Verify this additional functionality with a checksum.

Graphical user interface, text, application, email

Description automatically generated

## 4. Secure Communications

Refactor the code to convert HTTP to the HTTPS protocol. Compile and run the refactored code to verify secure communication by typing **https://localhost:8443/hash** in a new browser window to demonstrate that the secure communication works successfully.

Graphical user interface, text, application, Word

Description automatically generated

## 5. Secondary Testing

Complete a secondary static testing of the refactored code using the dependency check tool to ensure code complies with software security enhancements. You only need to focus on the code you have added as part of the refactoring. Complete the dependency check and review the output to ensure you did not introduce additional security vulnerabilities.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, Word

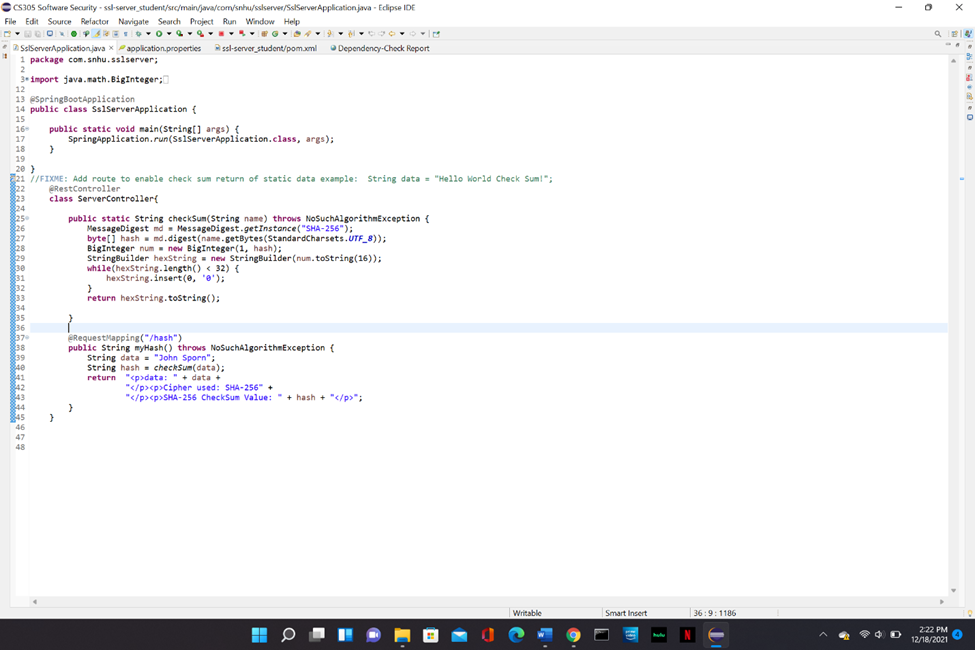
Description automatically generated

Graphical user interface, text

Description automatically generated

## 6. Functional Testing

Identify syntactical, logical, and security vulnerabilities for the software application by manually reviewing code.



Graphical user interface, text

Description automatically generated

## 7. Summary

Discuss how the code has been refactored and how it complies with security testing protocols. Be sure to address the following:

Many areas of security were addressed while refactoring the code for this system. APIs, cryptography, and code quality are the main areas that have been addressed. To ensure that the API endpoint was secure, I generated a self-signed SSL certificate which only sends data over a secure channel. It protects all credentials and data in transit using encryption. The data is encrypted using the SHA-256 hashing algorithm. This protects the data from being compromised. While implements this hashing algorithm I made sure the quality of my code was good. It is easy to understand and formatted professionally. If another developer needs to inspect or update the code that I have developed they will be able to easily read and maintain it. It is a good practice to make sure code is readable.

To maintain this application, it needs to be checked at least once a month to keep things up to date. Each time the system is reinspected a dependency check should be run to keep everything is up to date and no new issues have arisen. Keeping the application secure is vital to the company to ensure there is no issues for the users. Their data needs to be kept safe and know that is safe for them to keep using our system. This will keep the company happy and most importantly the users happy as well.