****

# CS 305 Project Two

**Practices for Secure Software Report**

Jessica Ayer

[Jessica.ayer@snhu.edu](mailto:Jessica.ayer@snhu.edu)

CS-305-T5602 – Software Security 22EW5

Instructor: Dr. Vivian Lyon

06/19/2022

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** | **06/19/2022** | **Jessica Ayer** |  |

## 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

Jessica Ayer

## 1. Algorithm Cipher

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

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

The appropriate encryption algorithm cipher for Artemis Financials needs is AES-256-CBC encryption algorithm paired with the SHA-256 cryptic hash function. AES-256 was announced by the National Institution of Standards and Technology in 2001 and adopted by the U.S. federal government in 2002. It was the result of a competition to create a replacement for the previously used DES after vulnerabilities were found (*Advanced Encryption Standards, 2022)*. In the Federal Information Processing Standards Publication 197, Dworkin M. et at. describe AES-CBC is a symmetric block cipher algorithm meaning that utilizes the same key to encrypt and decrypt data unlike an asymmetric block cipher which require a separate key for each process (2021).

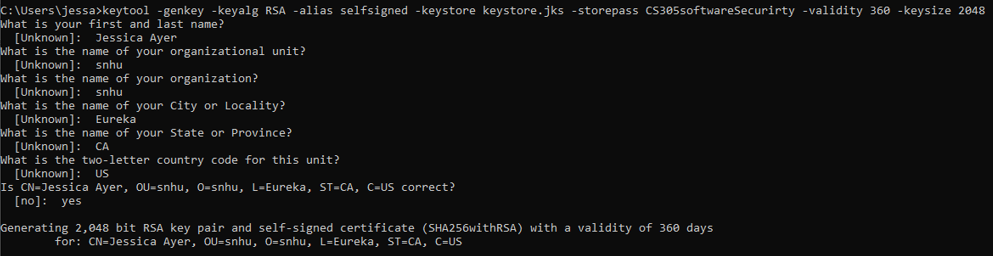
AES-CBC is capable of using cryptographic key bit lengths of 128, 192, or 256 bits to encrypt and decrypt 128-bit blocks of data and has been approved by the Secretary of Commerce as the standard for government organizations (Dworkin, 2001). I recommend going with a 256 key bit length as the longer bit length goes through 14 rounds of cipher block chaining (CBC) making it more secure and Artemis Financials’ database has enough computing power to be able to handle processing it (Hougen, 2021). The Federal Information Processing Standards breaks down CBC as the process of encrypting data by dividing it up into blocks, putting the first block through an initialized vector, and using the output as the ciphertext for the next block repeating or “chaining” until the last block is encrypted (*Des Modes Of Operation,* 1980).

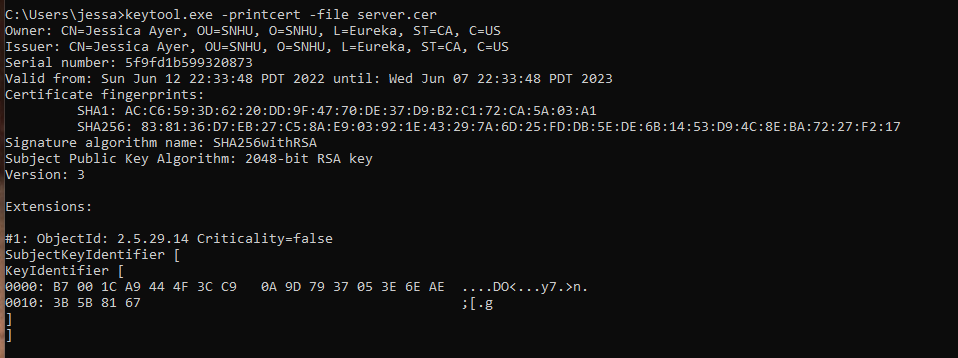
Pairing SHA-256 hash algorithm with AES-CBC cipher algorithm ensures data integrity by making it possible to identify if data has been tampered with during transfer after being signed and provides proof of the user’s identity. If data has been compromised than the hash value will change making the breach known (Crane, 2021). Like AES-256-CBC, SHA-256 algorithm utilizes a 256-bit key. The SHA-256 hash algorithm transforms the input data into what appears to be a random sequence of numbers producing a digest value with a 256 key bit length. If the receiver of the data were to run the data through the same hash function, they would receive a matching digest value if the data has not been tampered with. The SHA-256 hash algorithm prevents collision or, different plaintext inputs from outputting the same digest value (Simpilearn, 2022).

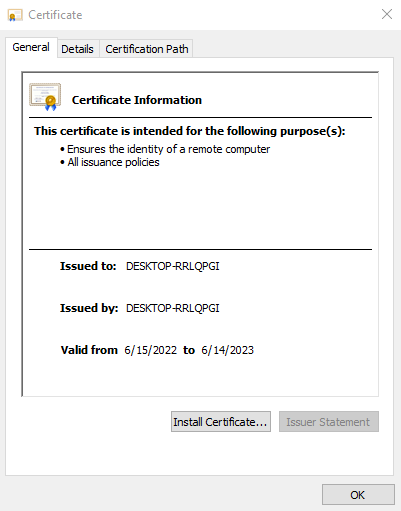
## 2. Certificate Generation

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

* To demonstrate that the keys were effectively generated, export your certificates (CER file) and submit a screenshot of the CER file below.



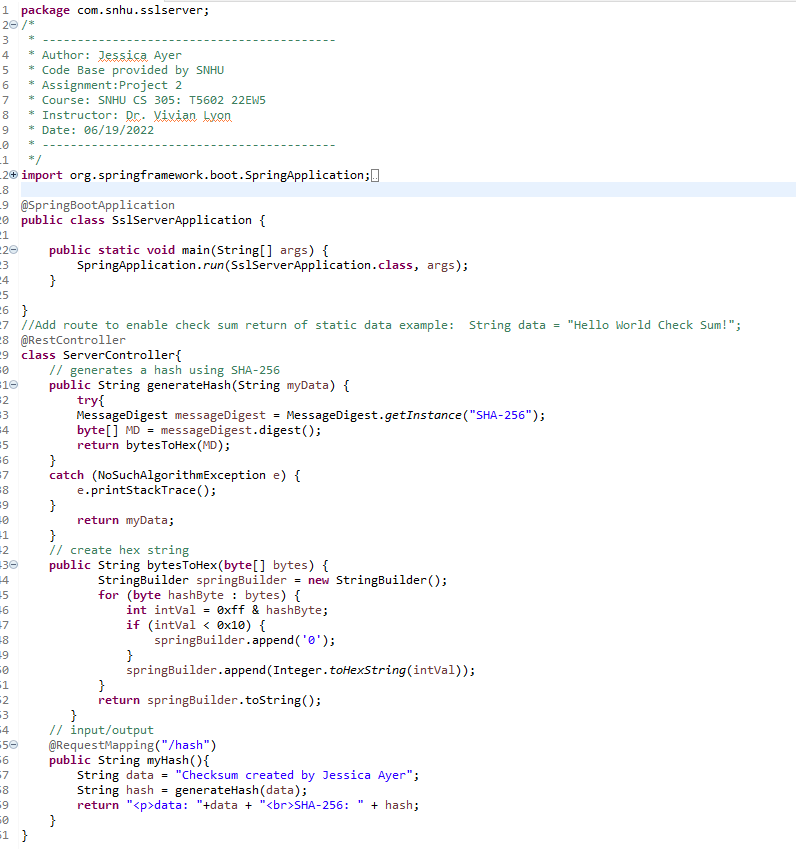




## 3. Deploy Cipher

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.

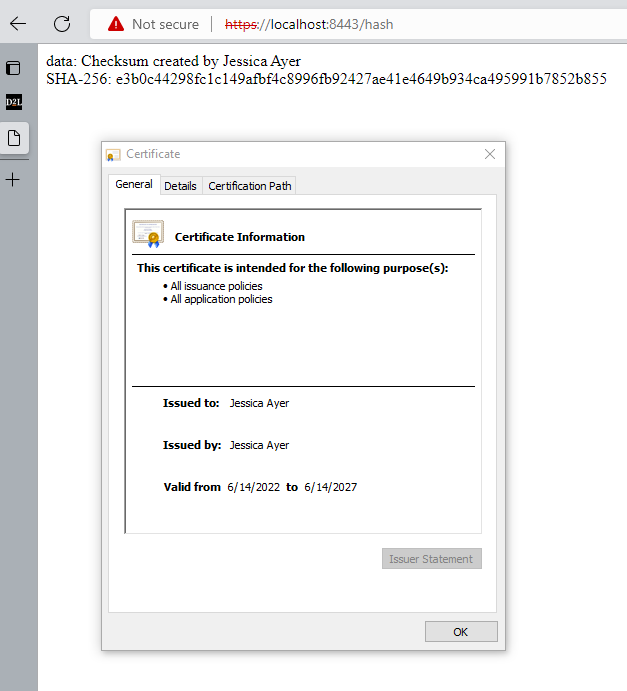
* Insert a screenshot below of the checksum verification. The screenshot must show your name and a unique data string that has been created.



## 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.

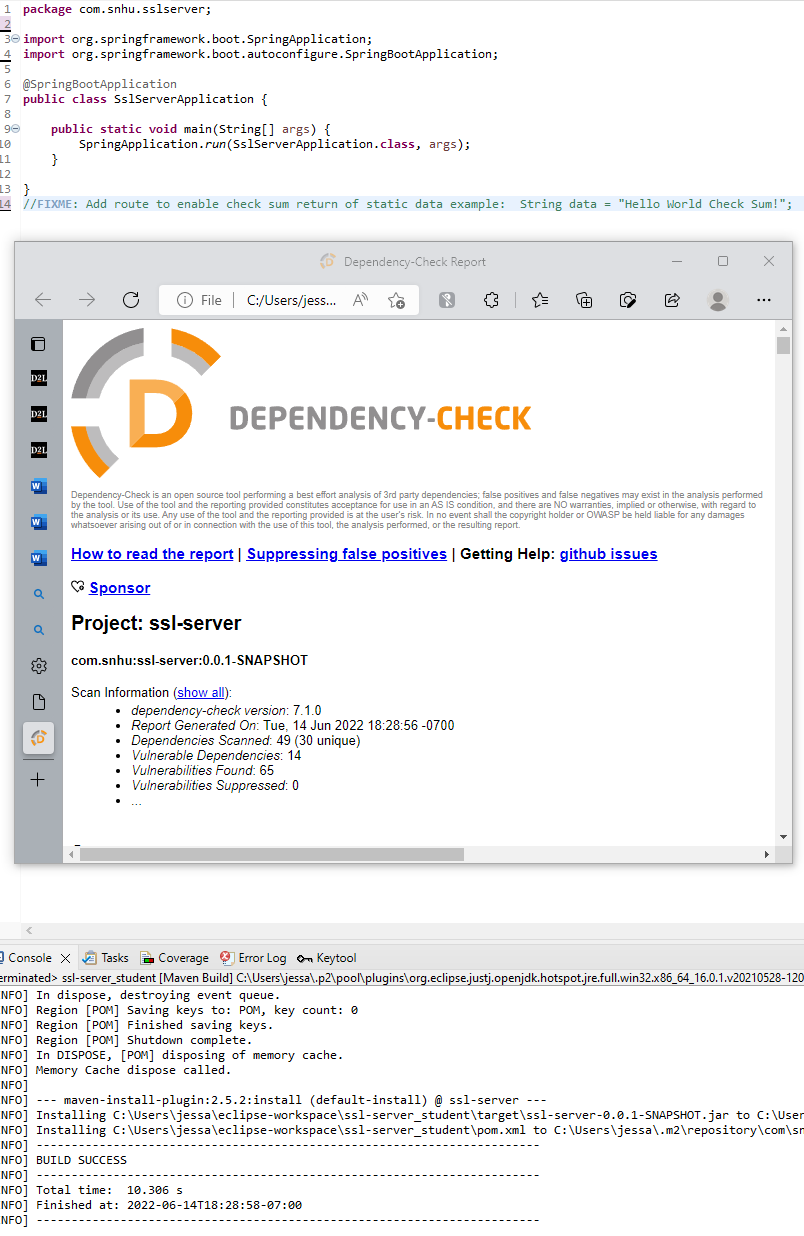
* Insert a screenshot below of the web browser that shows a secure webpage.

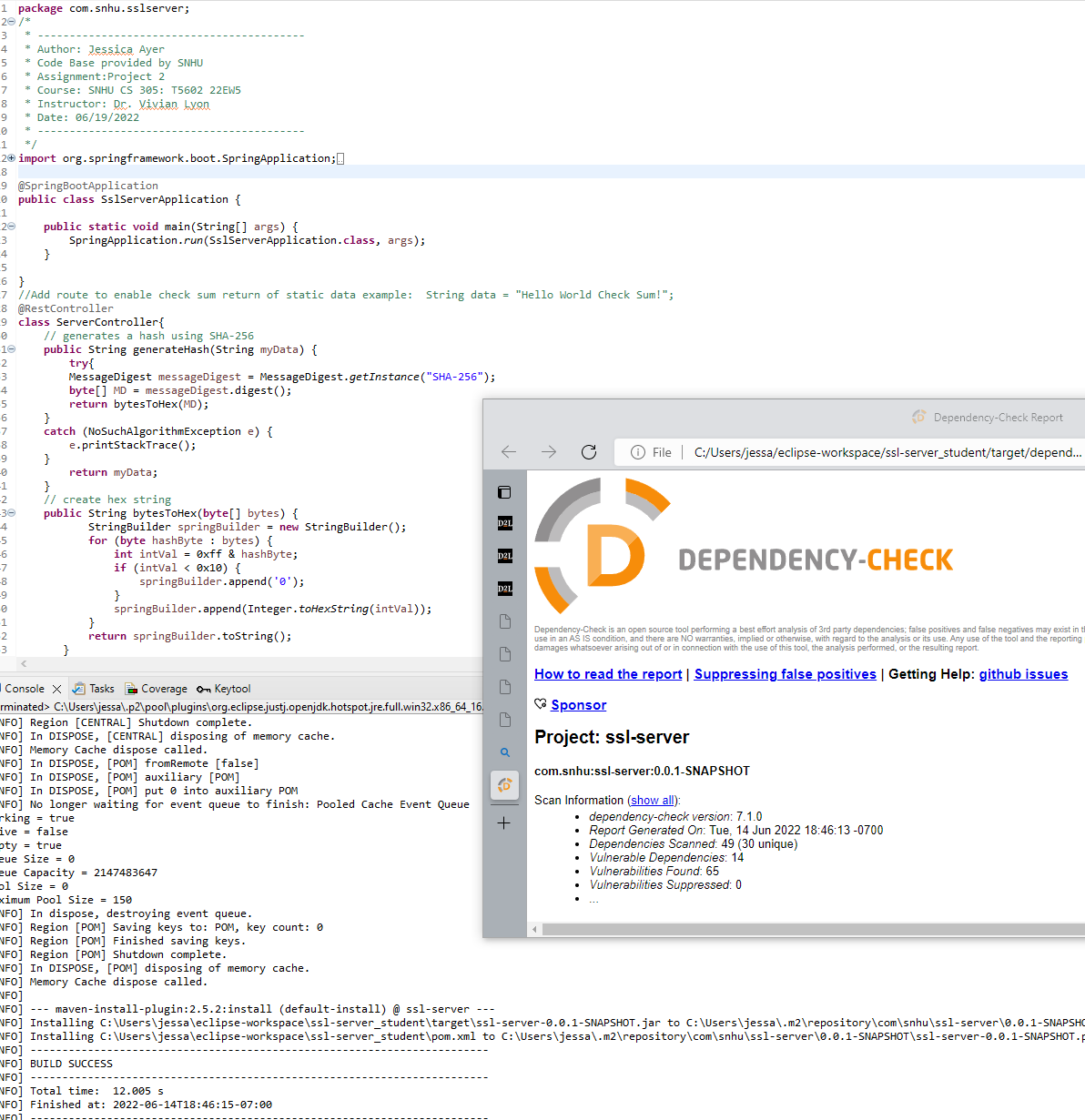


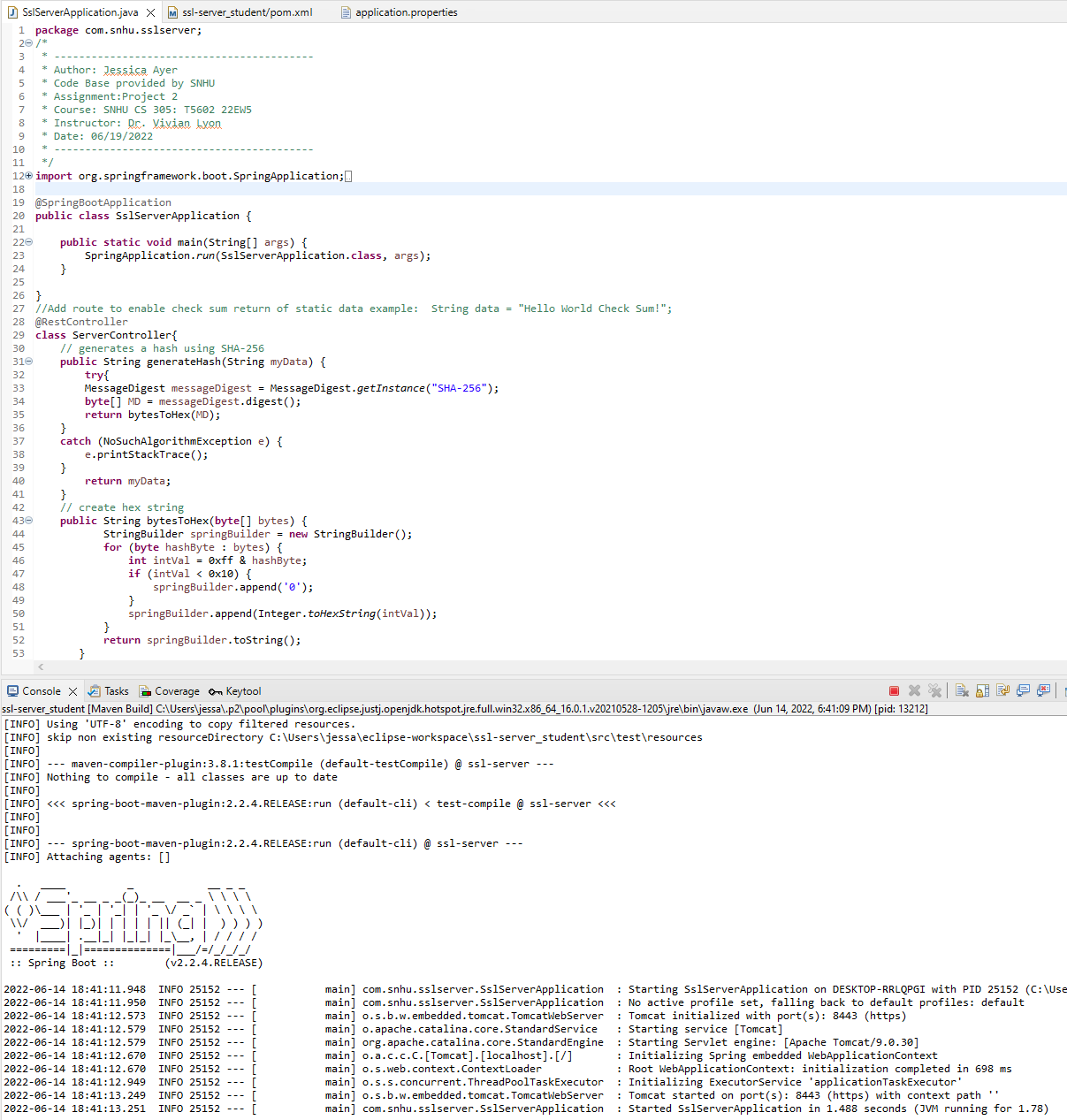
## 5. Secondary Testing

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.

* Include the following below:
  + A screenshot of the refactored code executed without errors
  + A screenshot of the dependency check report



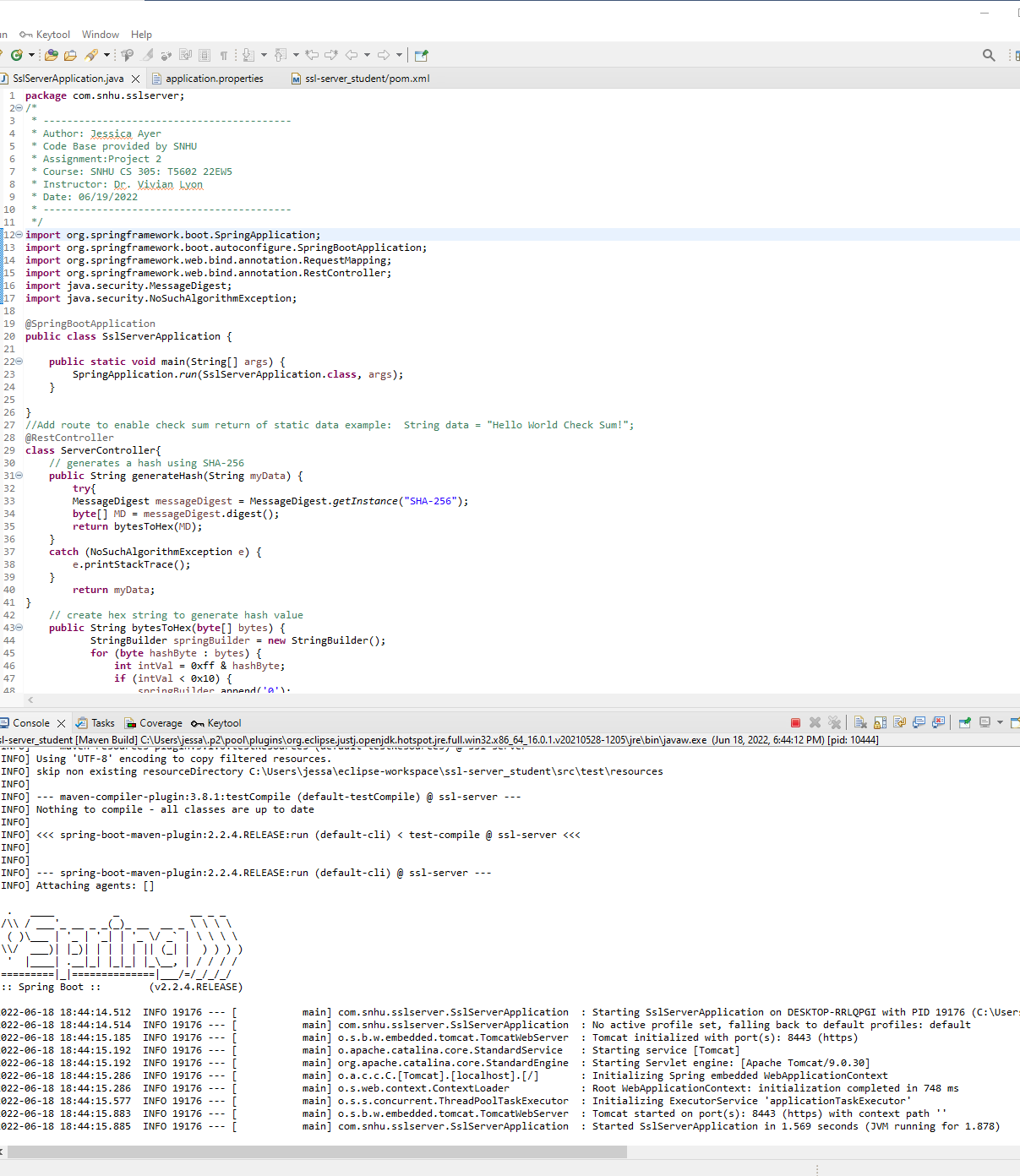




## 6. Functional Testing

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

* Complete this functional testing and include a screenshot below of the refactored code executed without errors.



|  |  |
| --- | --- |
|  | Pom.xml file line 8: The Spring boot starter plugin is out of date and needs to be updated. The current version is 2.6.6 |
|  | application.properties line 5: The application is currently using a self-signed certificate with the password displayed. A trusted third certificate needs to be purchased for secure https protocol and the password needs to be protected. |

## 7. Summary

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

* Refer to the Vulnerability Assessment Process Flow Diagram and highlight the areas of security that you addressed by refactoring the code.
* Discuss your process for adding layers of security to the software application and the value that security adds to the company’s overall wellbeing.
* Point out best practices for maintaining the current security of the software application to your customer.

Several areas of vulnerability have already been improved by refactoring the ssl-server class. The refactoring improved the secure API interactions by making improvements to the REST API and adding in a layer of cryptography to create a secure message digest. This was done by implementing the SHA-256 algorithm. This combined with adding and SSL certificate and HTTPS protocol improved the client/server secure distributed composing. Utilizing SHA-256 hash algorithm to create irreversible digests ensuring secure communication when data is to be transferred. By adding in a try/catch statement for generating the hash value the I’ve improved the systems error handling.

Upon Review, I’ve updated the OWSAP dependency check plug in to the latest version 7.1.0 to makes sure we are catching all currently knowns CVEs. In maintenance, the version should be periodically checked to makes sure it is still the latest and the dependency check report should be run at regular intervals. The Spring-boot-server still needs to be updated for improved security however that was out of the spectrum of this refactor. When user interface is added in it will be crucial to make sure secure input validation takes place as well as implementing AES for encryption/decryption of data. Regression testing should be performed after each update is made to make sure the integrity of the system is not compromised by said changes.

All of the above updates and suggestions were made with the intention of creating secure coding practices and making sure that Artemis Financials’ data was safe from exploitation. It is important for Artemis Financials’ customers to be able to entrust them with their personal information and know that the company is doing everything in their power to maintain their privacy. “Financial organizations have become the prime target of cyber-attack” experiencing as “many as 300 times more cyber-attacks than other companies” (Layton, 2021). We want Artemis Financials to feel confident that we are providing them with the top security available to protect their large database of sensitive data.

**References:**

*Advanced Encryption Standard* (2022, May 17). Wikipedia. https://en.wikipedia.org/wiki/Advanced\_Encryption\_Standard.

Crane, C. (2021, January 25). *What is Hash Function in Cryptography? A Beginner’s Guide*. https://www.thesslstore.com/blog/what-is-a-hash-function-in-cryptography-a-beginners-guide/.

*Des Modes Of Operation* (1980, December 2). Federal Information Processing Standard Publication 81, National Institute of Standards and Technology [online]. https://csrc.nist.gov/csrc/media/publications/fips/81/archive/1980-12-02/documents/fips81.pdf.

Dworkin, M., Barker, E., Nechvatal, J., Foti, J., Bassham, L., Roback, E., and Dray, J. (2001, November 26). *Advanced Encryption Standard (AES)*. Federal Information Processing Standard Publication 197, National Institute of Standards and Technology [online]. <https://doi.org/10.6028/NIST.FIPS.197>.

Hougen, A. (2021, May 26) *What Is AES Enctyption & How Does It Work in 2022? 256-bit vs 128-bit*. Cloudwards. <https://www.cloudwards.net/what-is-aes/>.

Layton, Roslyn (March 17, 2021). *Hackers Are Targeting U.S. Banks, And Hardware May Give Them An Open Door.* <https://www.forbes.com/sites/roslynlayton/2021/03/17/hackers-are-targeting-us-banks-and-hardware-may-give-them-an-open-door/?sh=343c381b14dc>

Simplilearn (2022, May 25). *A Definitive Guide to Learn The SHA-256 (Secure Hash Algorithms)*. https://www.simplilearn.com/tutorials/cyber-security-tutorial/sha-256-algorithm