****

# 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** | **April 13, 2022** | **Kathryn Snow** |  |

## 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

Kathryn Snow

## 1. Algorithm Cipher

Secure Hash Algorithm 256 (SHA 256) published in 2001 was a joint effort between the National Institute of Standards and Technology (NIST) and the National Security Agency (NSA). It was created to stop the blunt force attacks happening to a previous SHA. The significance of the 256 in the name stands for the final hash digest value. The features of the SHA 256 are the message length should be less than 264 bits in order to keep the digest as random as possible. The digest length will always be 265 bits in the SHA 256. The larger a digest is the more calculations which cost speed and space. The hash function of the digest is irreversible. The encryption algorithm cipher’s role is to make it impossible to find a way into the data. By creating a cipher with a secret key based on nonlinear functions the data is protected. We will wrap the data’s first cipher in a cipher to find the key. A non-technical example could be to imagine a box with a number and symbol lock inside a box with a key lock within a puzzle box where you must determine where to push and pull to find a lever to open it. This key wrap uses Advanced Encryption Standards (AES) as a primitive to encrypt plaintext key(s) with any associated integrity information and data. The ciphertext bit should be a highly non-linear function of each plaintext bit. When unwrapping each plaintext should be a highly non-linear function of each ciphertext bit. A simplified explanation is a key in the form of a non-linear function (like the formula for a parabola, exponents or square roots) is solved to solve another nonlinear function in order to gain access to ‘open’ the encrypted data.

There is a warning. If unwrapping produces the expected check value in A[0], then the chance that the key data is corrupt is 2^-64. If unwrapping produces an unexpected value, then the algorithm implementation must return an error, and it must not return any key data.

Ciphers examples were first noted in 600 B.C. with a leather strap and a wooden rod. If you received the leather strap and had the correct rod you would receive a private message. As methods evolved the private message evolved along with it. Currently the United States is on high alert for state-sponsored attacks. Cyberspace is defined as the fifth domain of warfare by NATO. In warfare cyberspace is used to spread fake news to manipulate humans to affect financial institutes in the way the attackers want. They attack directly with a variety of techniques to change or control the data. As a financial service, Artemis Financial needs to be aware of the crimes most likely to be used against them. Credential theft and identity theft are a concern for the security of the service. There has been a steady increase in both crimes to attain a customer’s funds. The Federal Trade Commission 2020 Report shares there have been 1,387,616 reports of identity theft and 498,278 imposter scams. Artemis Financial does not want to be responsible for allowing their clients data to be accessed by criminals. Employee error by opening phishing email was the most common cyberattack in 2016. The criminals are improving their techniques so that people and the employees of Artemis Financial need to be alert and focused when they read their email to avoid the error. We need to confirm our Cloud providers are secure along with any third-party vendors used. Data theft and data manipulation and ransomware are the areas we are going to address now. It is important to stop cybercrimes and it is positive to know the criminals are being prosecuted. In breaking news, the US Department of Justice just took down RaidForums. This is part of the report, “According to the affidavit filed by law enforcement, RaidForums operated from around 2016 through Feb. 22 of this year as a massive online marketplace for individuals to buy and sell hacked and stolen data, including sensitive personal and financial information from victims in the United States. Among those sales included leaked data from 178 million Facebook users. “The takedown of this online market for the resale of hacked or stolen data disrupts one of the major ways cybercriminals profits from the large-scale theft of sensitive personal and financial information,” said Assistant Attorney General Kenneth Polite Jr. of the Justice Department’s Criminal Division.” According to court documents, as a part of the investigation, law enforcement obtained a copy of the backend of RaidForums’ database, granting them a treasure trove of information on forum members including account registration information, user IP addresses and private messages with other members.”

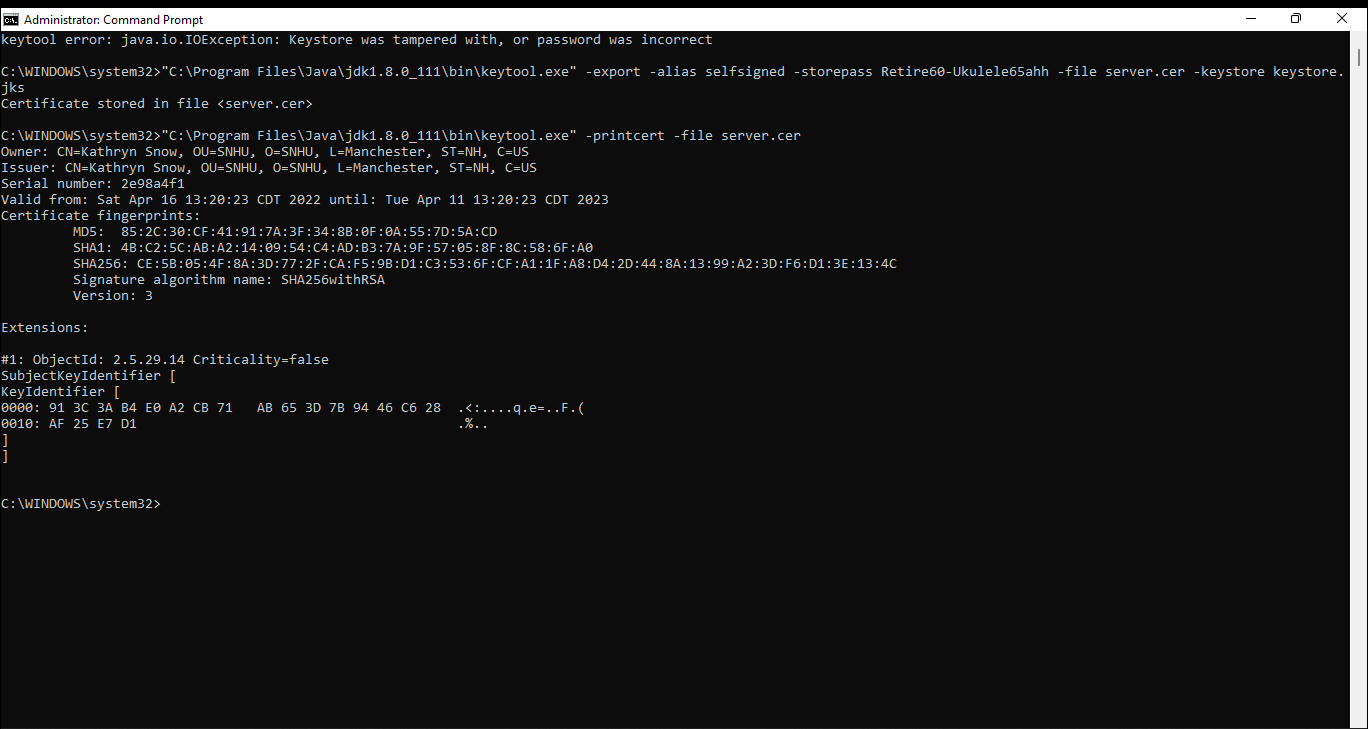
## 2. Certificate Generation

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.

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.

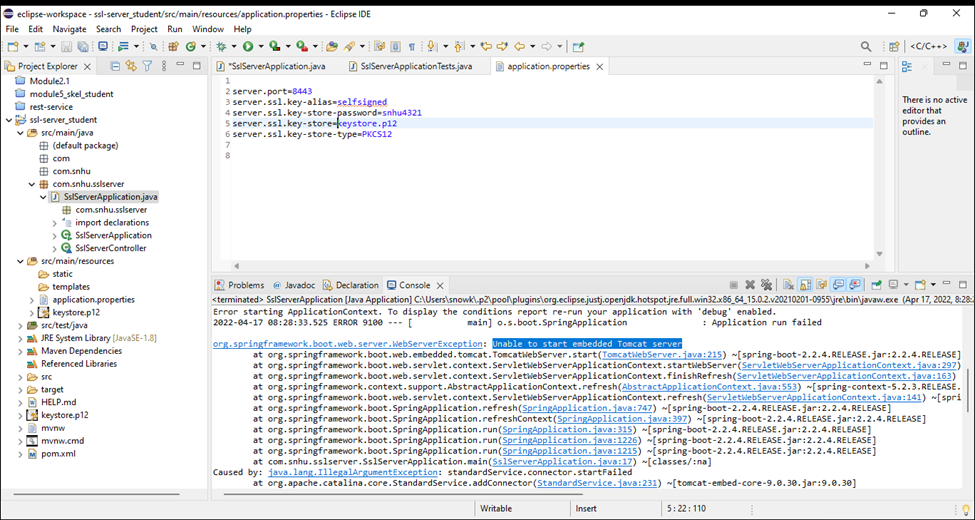
* Insert a screenshot below of the checksum verification. The screenshot must show your name and a unique data string that has been created.

A screenshot of a computer

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.

I saved the code. I did nothing to it and ran it again and again and it didn’t work. I saved it in notepad++ and tried other ideas. I reloaded it and placed the code in by copy and paste and by typing it in. I still received the error code “Unable to start embedded Tomcat server”. As I searched for answers in Java Oracle files, I found a link that took me to a personal site. As always, I learn more by trying to figure out why things are not working.

Graphical user interface, text, application

Description automatically generated

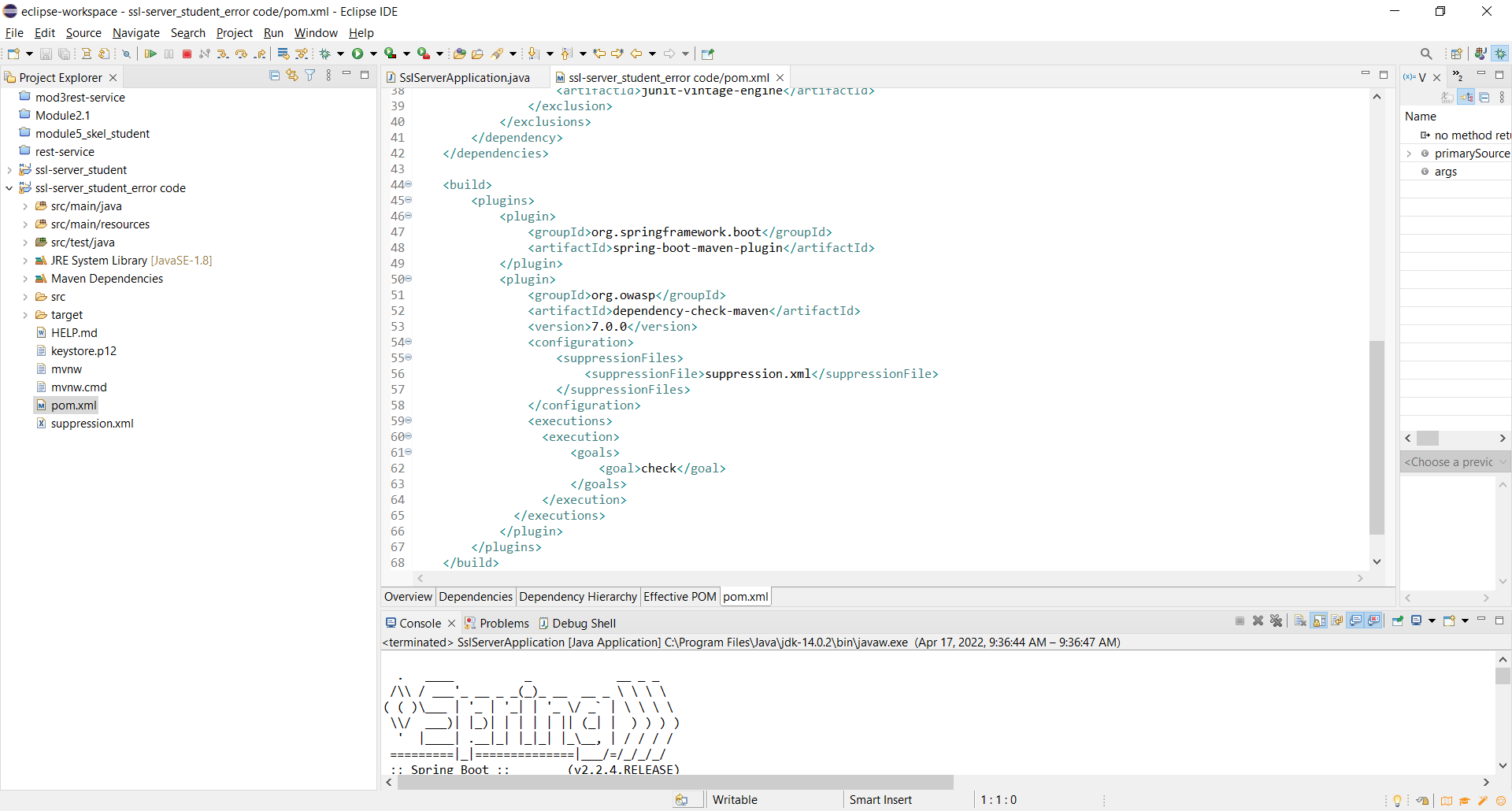
Graphical user interface, text, application

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.

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



* I left medium level warnings in to demonstrate removing high and critical warnings.

A screenshot of a computer

Description automatically generated

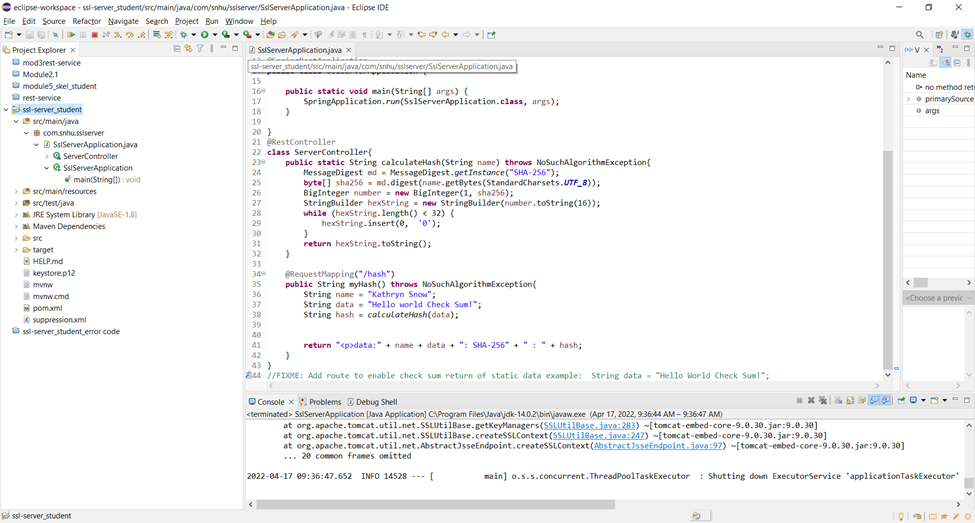
## 6. Functional Testing

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.

Graphical user interface, text, application

Description automatically generated

[Insert screenshot(s) here.] 

## 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.

APIs, Cryptography, Client/Server, Code Error, and Code Quality was addressed in the code. APIs were addressed with the implementation of HTTPS and the browser interface. Cryptography was addressed with the usage of the SHA 256 algorithm and the checksum verification. Client/Server was demonstrated in the keystore usage. Code Error was addressed with the NoSuchAlgorithm exception. And Code Quality was addressed as the code was reviewed to ensure that it worked.

* Discuss your process for adding layers of security to the software application and the value that security adds to the company’s overall wellbeing.

Adding Layers of security acknowledges that no single point in a computer system can be completely secure. Layering security seeks to block a variety of attacks from many different sources. In our project we created the self-signed certificate and refactored the pom.xml creating the suppression file to hold the vulnerabilities. The checksum, which worked then didn’t, was another layer of security. With all of these working correctly we would know our company was secure.

* Point out best practices for maintaining the current security of the software application to your customer.

It is best practice to maintain our application’s security by using up-to-date files. We then need to be aware of their weaknesses which we find in a dependency check report. We can evaluate other means of security based on this report. Regardless, we will use the check sum to ensure the application is correctly reaching the browser. And we will create a certificate that states we are a trusted source, and we are who we say we are when the user reaches us. It is expected that our code will follow the highest standards and most up-to-date information on cyber-crime and recommended protection. Protecting Artemis Financial is an ongoing responsibility as the climate of cyber-crime changes moment by moment.

S. (2022b, February 17). *A Definitive Guide to Learn The SHA-256 (Secure Hash Algorithms)*. Simplilearn.Com. Retrieved April 15, 2022, from <https://www.simplilearn.com/tutorials/cyber-security-tutorial/sha-256-algorithm#what_is_the_sha256_algorithm>

Riley, T. (2022, April 12). *Justice Department seizes major cybercrime spot RaidForums*. CyberScoop. Retrieved April 14, 2022, from <https://www.cyberscoop.com/raidforums-seized-doj/>

I. (2021b, April 24). *Importance of Layered Security in Cyber Defense*. Identity Management Institute®. Retrieved April 16, 2022, from <https://identitymanagementinstitute.org/importance-of-layered-security-in-cyber-defense/>