****

# Practices for Secure Software Report

**Table of Contents**

[**Document Revision History 3**](#_heading=h.30j0zll)

[**Client 3**](#_heading=h.1fob9te)

[**Instructions 3**](#_heading=h.3znysh7)

[**Developer 4**](#_heading=h.2et92p0)

[**1. Algorithm Cipher 4**](#_heading=h.tyjcwt)

[**2. Certificate Generation 4**](#_heading=h.3dy6vkm)

[**3. Deploy Cipher 4**](#_heading=h.1t3h5sf)

[**4. Secure Communications 4**](#_heading=h.4d34og8)

[**5. Secondary Testing 4**](#_heading=h.2s8eyo1)

[**6. Functional Testing 4**](#_heading=h.17dp8vu)

[**7. Summary 4**](#_heading=h.3rdcrjn)

[**8. Industry Standard Best Practices 4**](#_heading=h.26in1rg)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **2/19/23** | **Thomas Conahan** |  |

## 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.

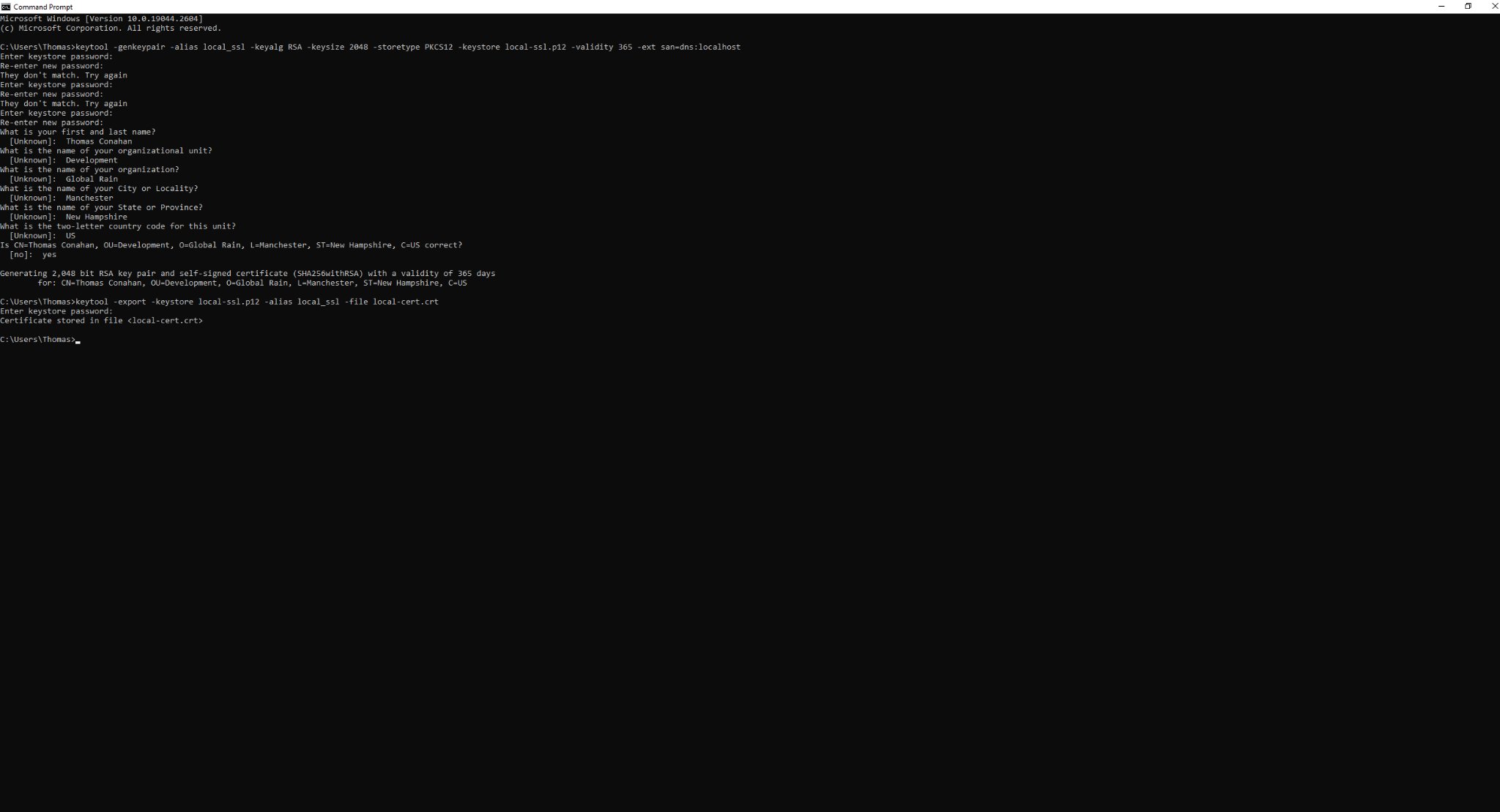
## Developer

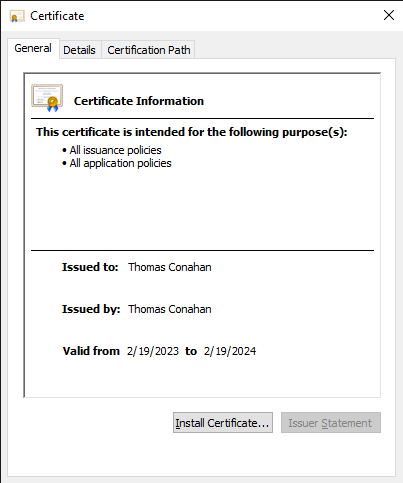
Thomas Conahan

## Algorithm Cipher

The cipher algorithm that would work the best for Artemis Financials needs would be the AES-256 cipher to encrypt files for transfer and the use of SHA-256 for hashing to generate checksum verification. This is the most useful cipher to them because their main goal is to secure data that is being transmitted between them and their customers. AES is a symmetric key algorithm. It uses 4 by 4 arrays to divide data, and data is systematically moved around and combined again based on a key (Jena, 2023). Another key is then generated and this process happens all over again, and will continue for 14 rounds. It splits data into blocks of 128 bits and uses keys of 256 bits (Baeldung, 2014). These splits are then joined back together after the encryption process is finished. Being a symmetric key cipher, AES uses the same key to encrypt and decrypt data(Baeldung, 2014). This is less secure than the use of asymmetric keys, which uses two different keys to encrypt and decrypt data, but is much faster. In addition, the use of the 14 rounds of encryption also makes up for the loss of security that the symmetric key causes (Kiteworks, 2023). Encryption has developed a lot in the past 20 - 30 years, but dates all the way back to the BC era, where Spartans would encrypt messages between each other to prevent enemies from reading them if they got intercepted (Thales, 2023). Encryption used to only be used for important messages and letters, but now, almost all data is now encrypted. In recent history, algorithms are constantly updated to protect against the new and improved methods of hackers. Encryptions get stronger with each update, and result in breachers becoming more creative in the way they try to get data.

## Certificate Generation

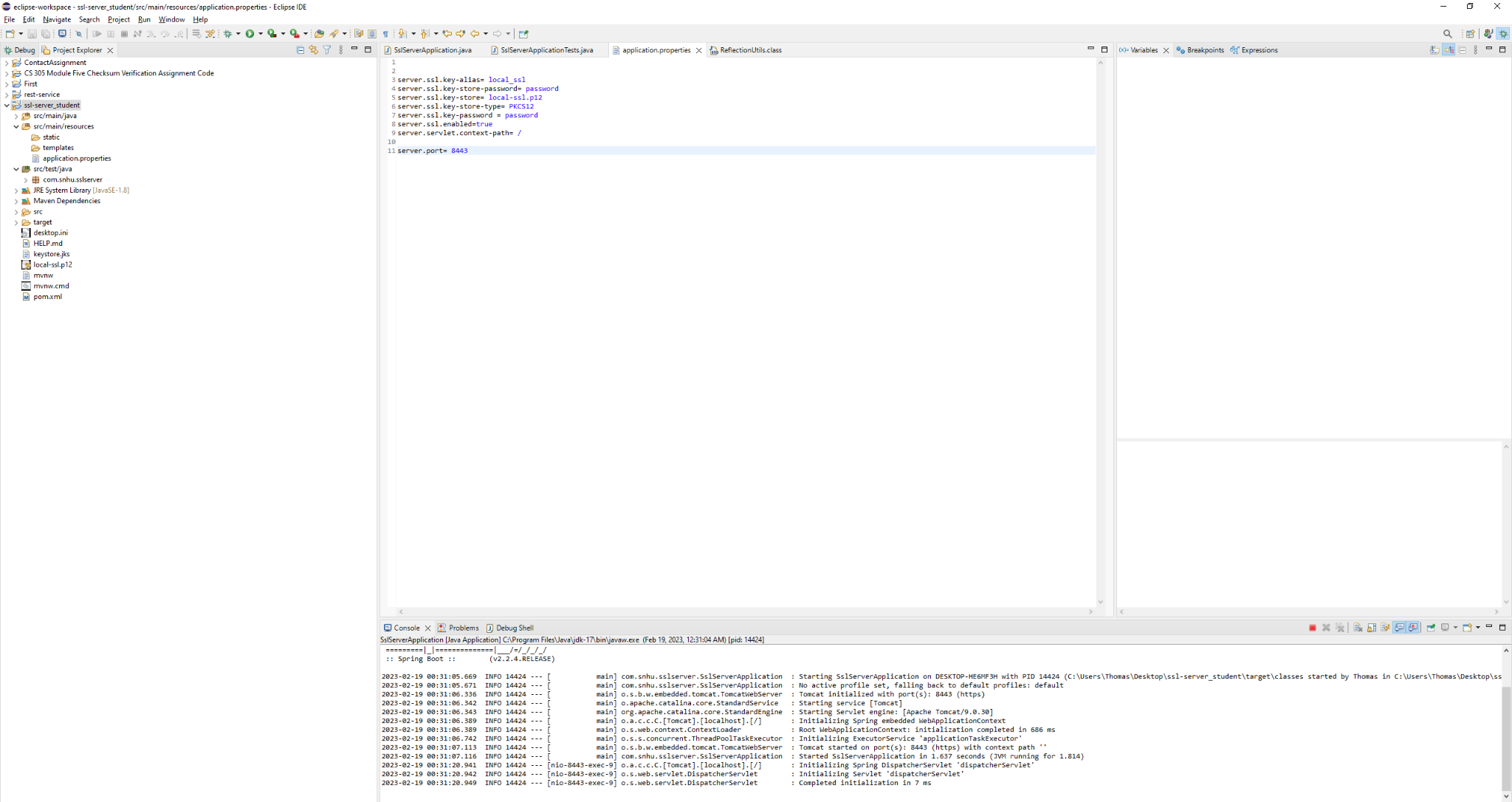




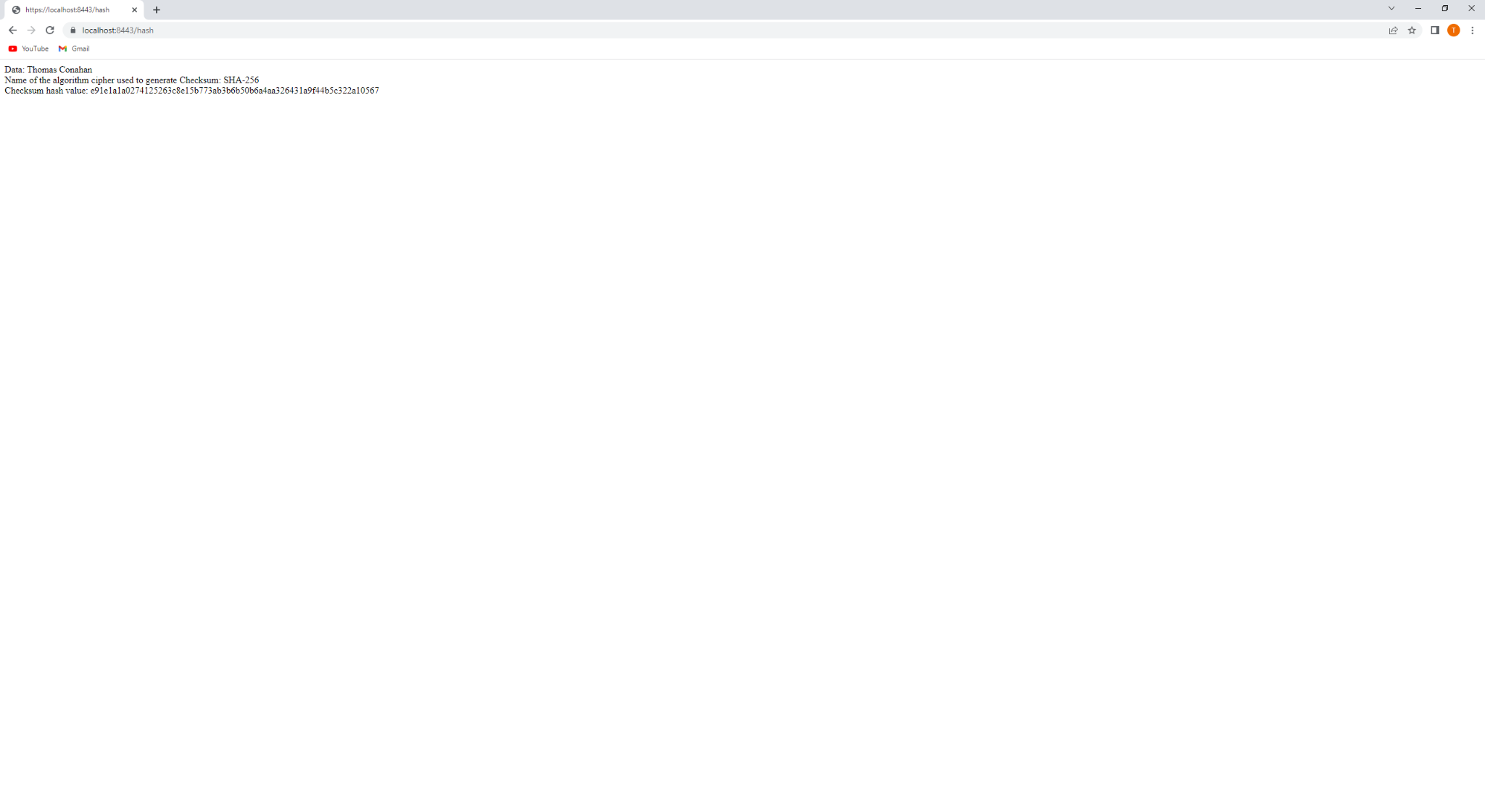
## Deploy Cipher

## 

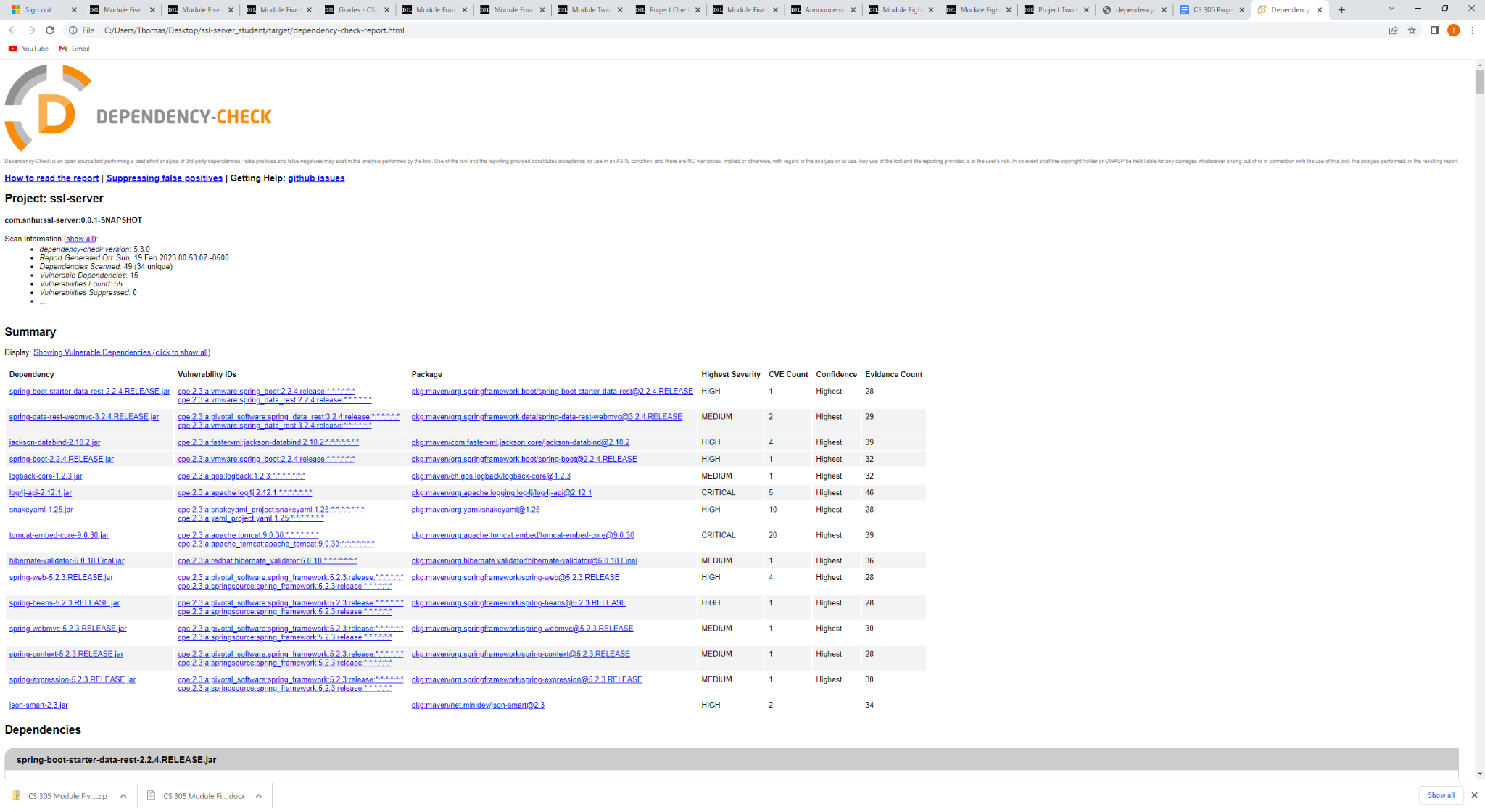
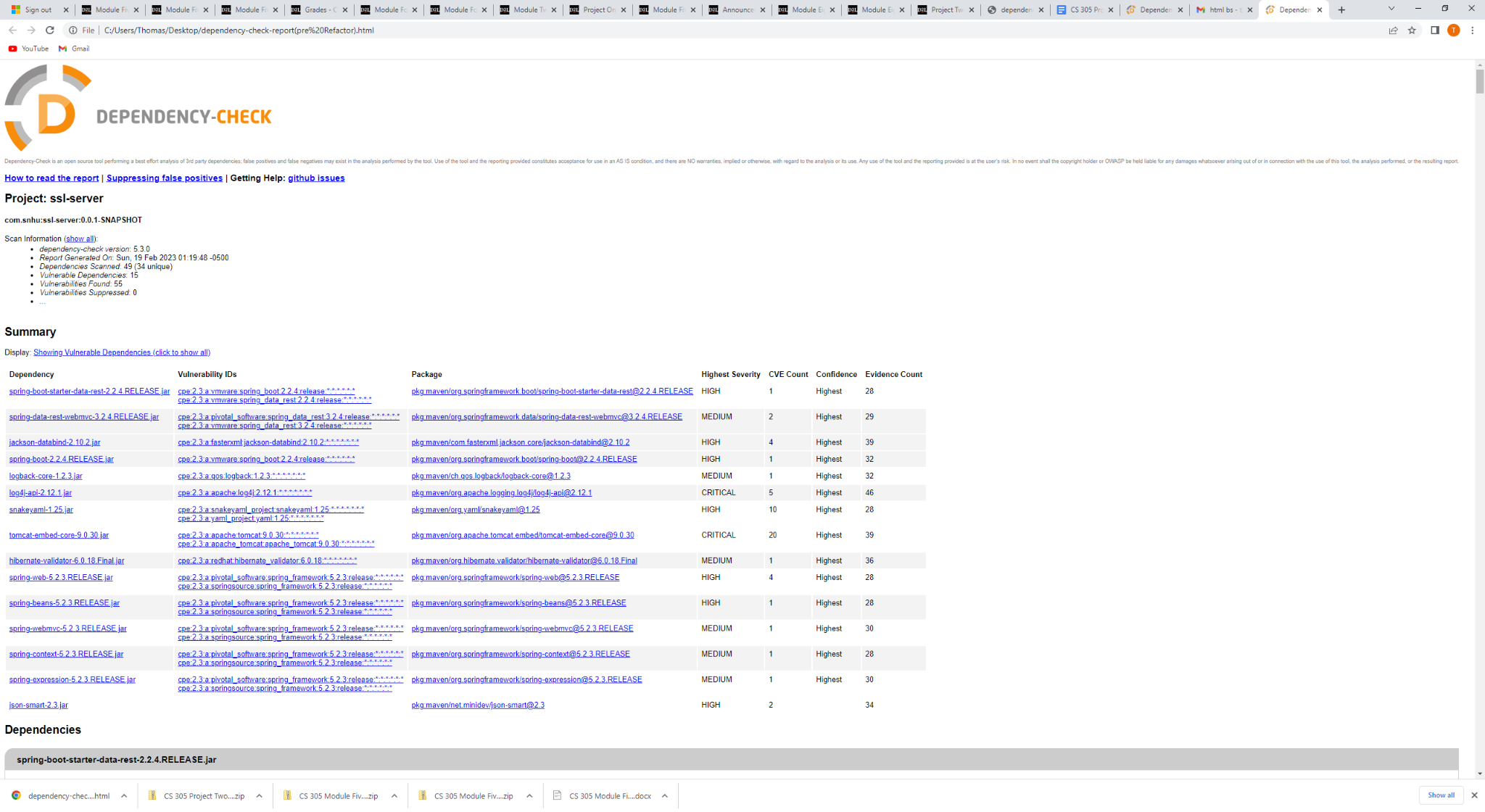
## Secure Communications



—



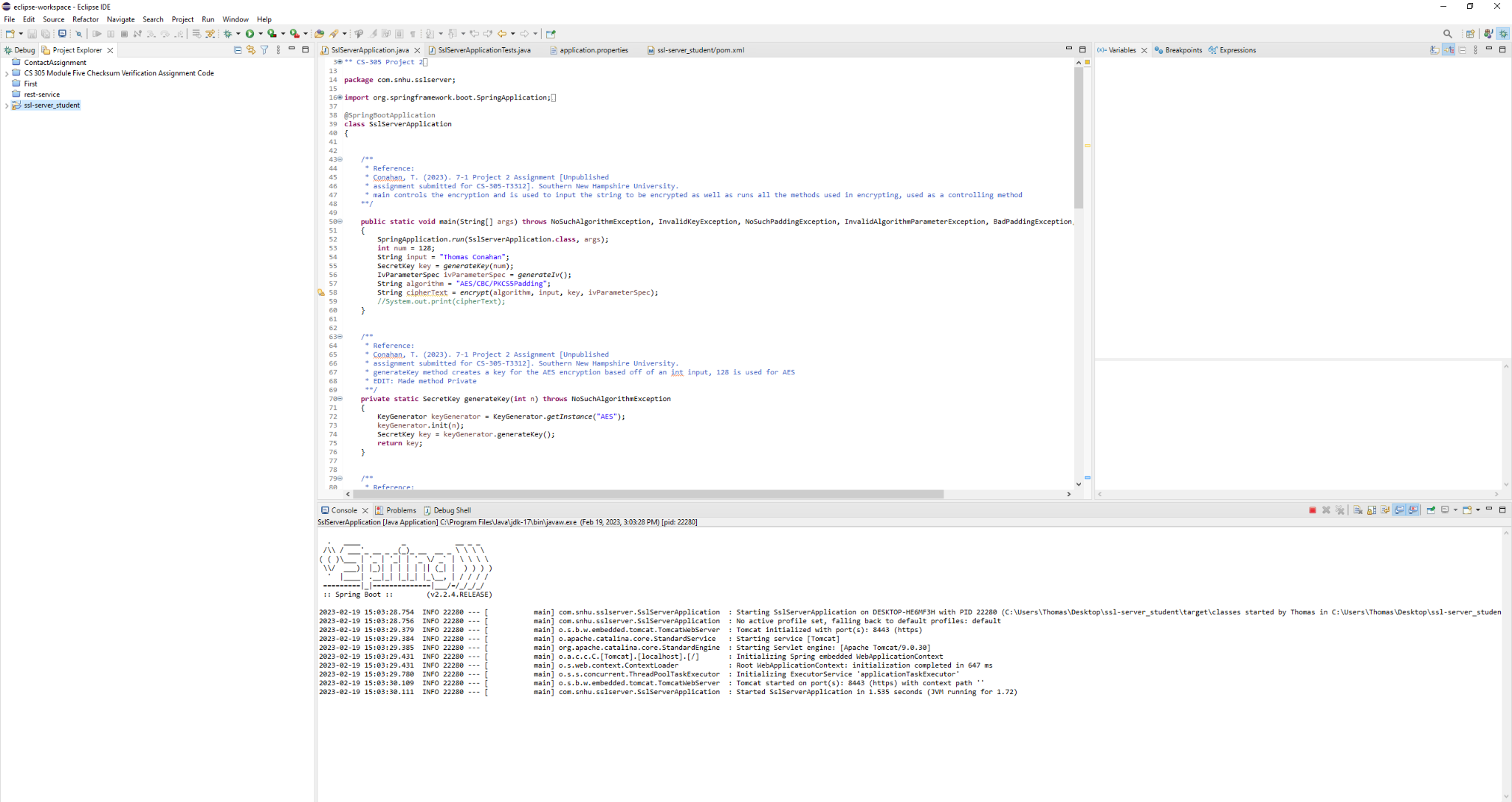
## Secondary Testing



## 

## 6. Functional Testing

After review of the code, I changed all the methods of the class to be private except for the main class. These methods should not, and now can not be called from outside of the original class they were created in. I also removed some unused imports that were either redundant, or became pointless after changing the code. Removed redundant variable declarations to clean code.



## 

## 

## 

## 7. Summary

I refactored the code to add encryption through the use of the AES-256 cipher algorithm which covers the cryptography section to add security to the data that is transmitted between the user and the company. In addition, I added a checksum verification through the use of a hash function and the SHA-26 algorithm to verify users are who they say they are, in turn, securing the connection between the user and the server. Errors are handled through the use of throws in the method declaration that allows for the code to run smoothly and handle the errors that code may run into. Overall, the code follows the industry standard for security and declaration. Return types match and there is no redundant and/or useless code that is not used. Multiple layers of security were provided within the refactored code. Using the encryption algorithm as well as a checksum verification provides the company with a secure and easy to use application. In addition, small things like private methods and variables contribute to overall security as well as code cleanliness. Using things like the most recent version and updated algorithms, checks, and practices contribute to code being secure. Outdated algorithms can be easy to break as they have been around long enough for people to test on how to crack them, which is why it is one of the most important things to check when using an algorithm. Based on the company's needs, I knew that I needed to add an encryption algorithm for the data they would be sending, which is why I used the AES-256 cipher. In addition, the company wanted secure interaction between them and the user, so adding a checksum verification allows for that to be possible. Users will be verified that they are who they say they are. Overall, the company has secure interactions for their website. Protecting their customers is important because they are transmitting and storing important data and data breaches can be detrimental to the company.

## 

## 8. Industry Standard Best Practices

Many of the known security vulnerabilities include insufficient transport layer protection or SQL injections (O’Leary, n.d.). By using the checksum, we protected the transport layer of the application, when there is communication between the server and the user. SQL injections are protected against by encrypting data when in transport. This protects the data, even if accessed, from being read because of the layers of encryption. We protected against this by using the AES-256 encryption algorithm. Handling errors also contributes to security by preventing a crash of an application and limiting weaknesses in the code that can be broken through. We handled this by using throws as well as editing code to prevent no starts and/or crashes. Using the industry standard to protect the company is important not only to the company, but to you as a developer. It is important as a developer to keep up with the industry standards to always have the best protection against attacks when developing. Software and software security is constantly changing and updating, and keeping up to date with the latest algorithms, practices, syntactical components, etc is crucial. In regards to the company, providing them with the most up to date protection will provide them with a feeling of security knowing that their and their customer’s data is secure. Data breaches can be the downfall of companies, so using industry standards to protect their data is necessary.

**References:**

Everything you need to know about AES-256 encryption. Kiteworks. (2023, February 13). Retrieved February 19, 2023, from <https://www.kiteworks.com/risk-compliance-glossary/aes-256-encryption/#:~:text=AES%2D256%20encryption%20is%20based,for%20both%20encryption%20and%20decryption>

Jena, B. K. (2023, February 9). What is AES encryption and how does it work? - simplilearn. Simplilearn.com. Retrieved February 19, 2023, from <https://www.simplilearn.com/tutorials/cryptography-tutorial/aes-encryption>

Thales. (2023, February 1). A brief history of encryption (and cryptography). Thales Group. Retrieved February 19, 2023, from <https://www.thalesgroup.com/en/markets/digital-identity-and-security/magazine/brief-history-encryption>

Baeldung. (2021, November 14). Java AES encryption and decryption. Baeldung. Retrieved February 19, 2023, from <https://www.baeldung.com/java-aes-encryption-decryption>

O'Leary, R. (n.d.). Java application vulnerabilities - dzone refcardz. dzone.com. Retrieved February 19, 2023, from <https://dzone.com/refcardz/java-application-vulnerabilities#:~:text=There%20are%20three%20classes%20of,(SQLi)%20and%20Unpatched%20Libraries.&text=This%20is%20the%20most%20critical,at%2094%20percent%20of%20vulnerabilities>.