****

# 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** | **02/17/2023** | **Jacob Feldman** |  |

## Client



## Developer

Jacob Feldman

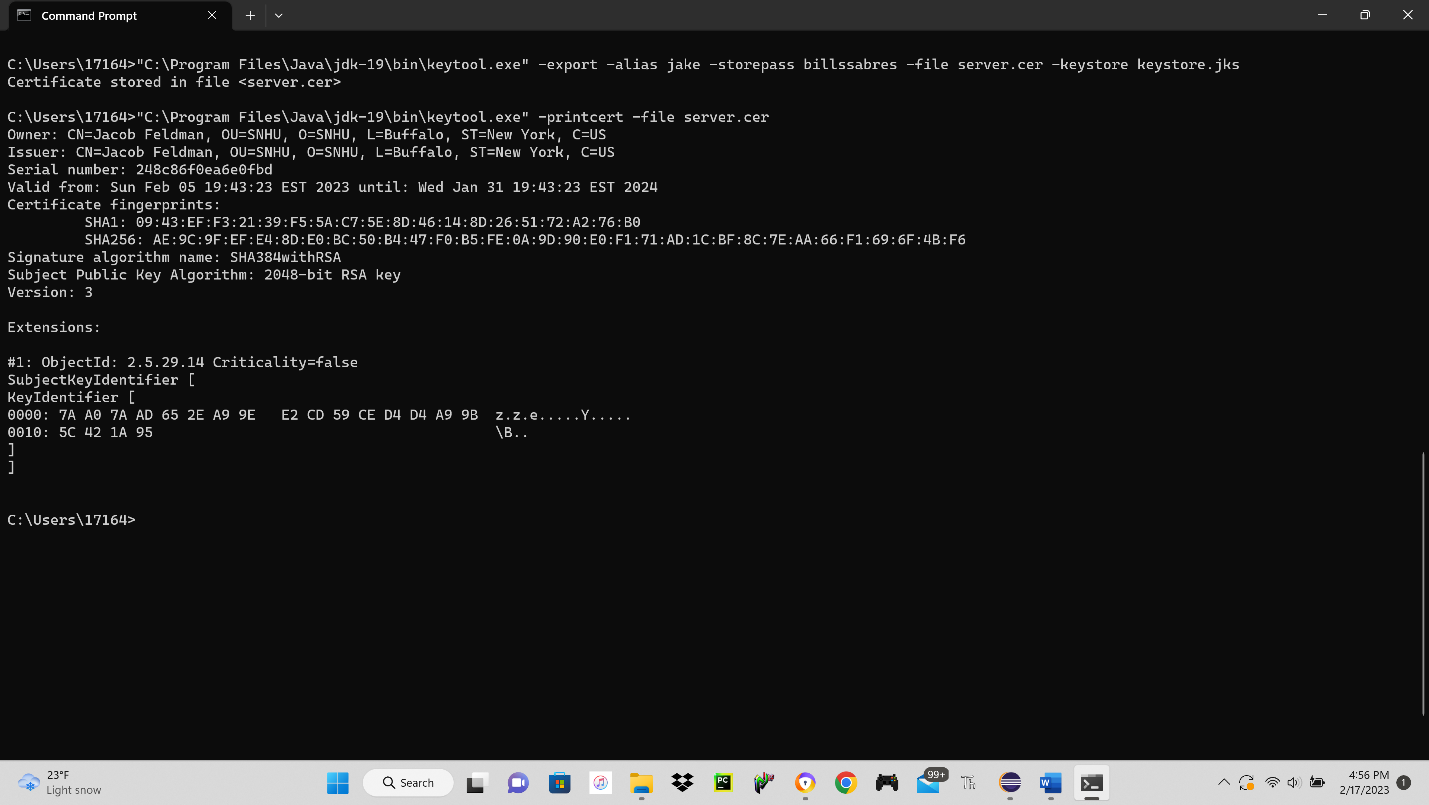
## Algorithm Cipher

After carefully reviewing the various options for software security measures, I have decided to incorporate an encryption algorithm cipher. This is a function that essentially converts information into random characters and objects to prevent malicious attacks (or loss of data) on the information involved. It will generate a ‘key’ that only the sender and receiver have access to, which will unscramble the cipher back into its original message. I have chosen the Advanced Encryption Standard, or AES, as the best option for encryption. The government of the United States approved the AES for use in 2001 and it is now one of the most trusted cipher generators in the world. With a 128-bit block cipher supporting keys of 128, 192, and 256 bits, the AES creates a nearly impossible code to crack.

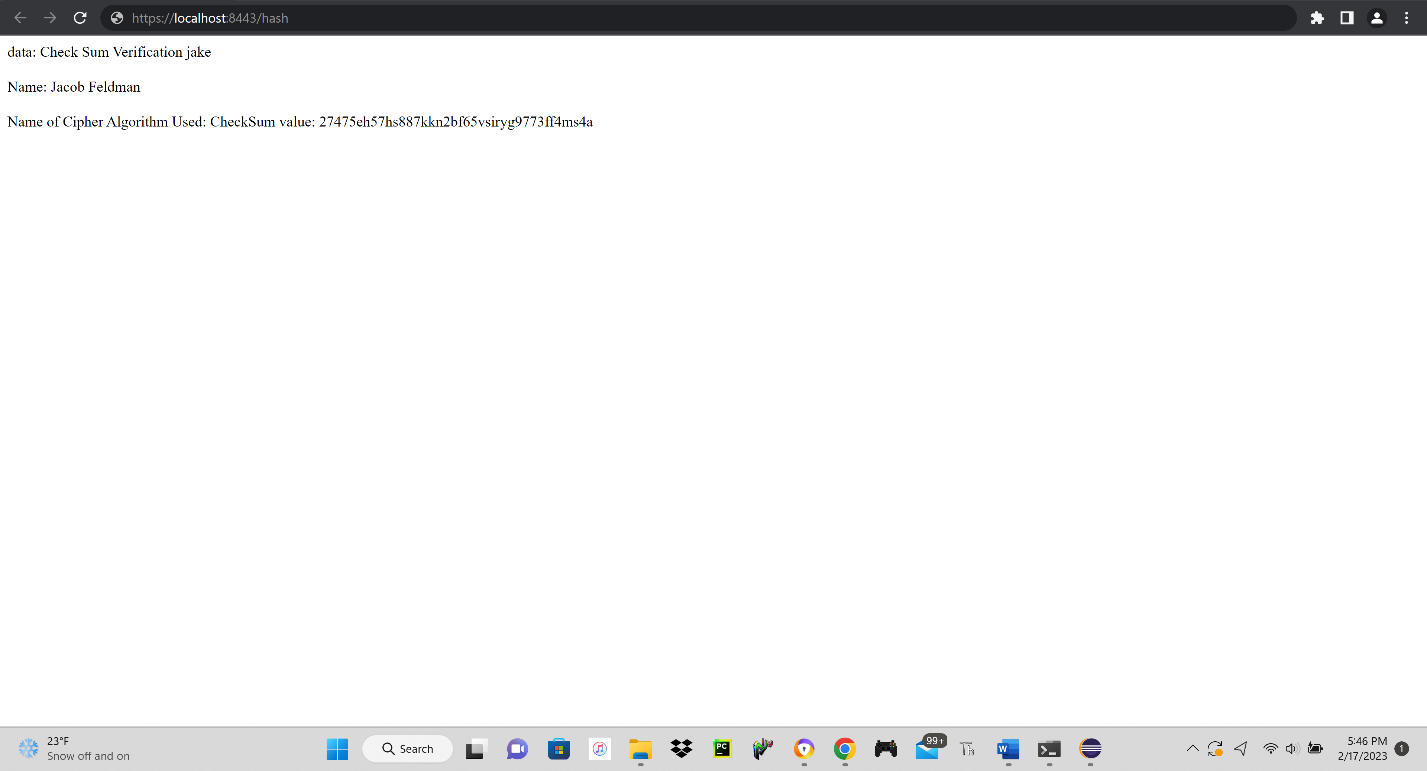
Secure Hashing Algorithms (SHA) are important in enabling the encryption process. Hash functions convert plaintext data, such as someone’s name or address, into a cipher, or Crane. This cipher will contain an assortment of random letters and numbers, all of which are meaningless without a way to change them back. As I mentioned earlier, a key is generated when transferring encrypted material. Either a symmetric or nonsymmetric key will be generated, depending on the scope of the data. For example, a large company like Google may want to use nonsymmetric keys when transferring information to thousands of employees. This means that not all the keys are the same and will vary in the access that they can unlock. The CEO’s key would be able to see much more of the information than a developer’s key. If I were to send an encrypted email to one person, I would use a symmetric key because only two are generated, one for each of us. The AES stands out in this regard as well since it generates longer key lengths for even safer ciphers.

Encryption itself is argued to be as old as 1900 BC, being used in cave paintings and inscriptions. Even the Germans used an encryption machine called Enigma in the second world war, shortly before computers existed. Computer encryption algorithms, however, came around in the early 1970s. IBM created what would become the US national standard, or Data Encryption Standard (DES). This remained so until 1997 when it was finally cracked. Computer scientists have learned an incredible amount since then, and it shows with modern encryption algorithms. The AES today is the most popular encryption mechanism in the world. Companies dealing with encrypting their customer’s data must adhere by strict laws and regulations. There are no current known flaws or holes in this algorithm, so it remains a trusted choice for anyone.

## Certificate Generation



## Deploy Cipher

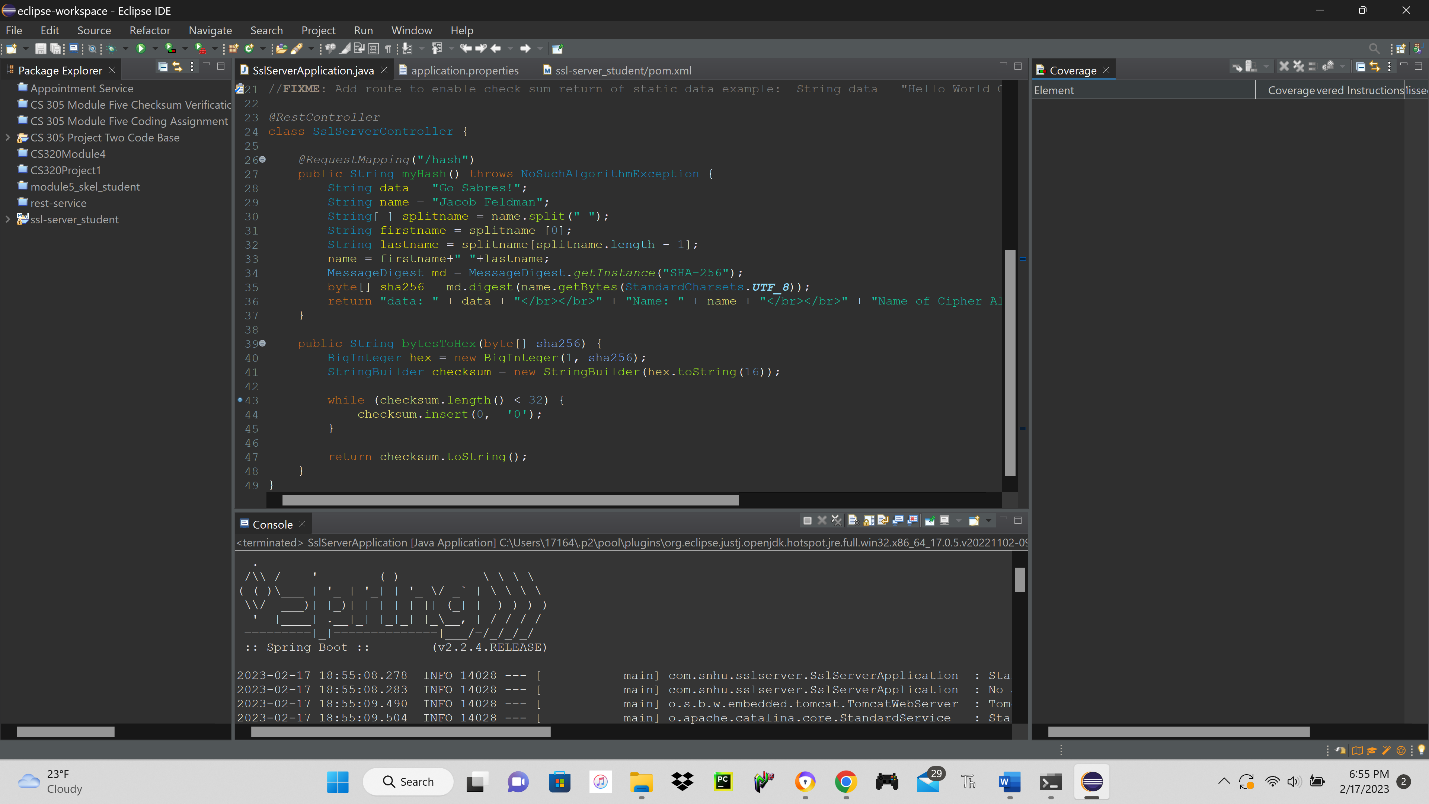


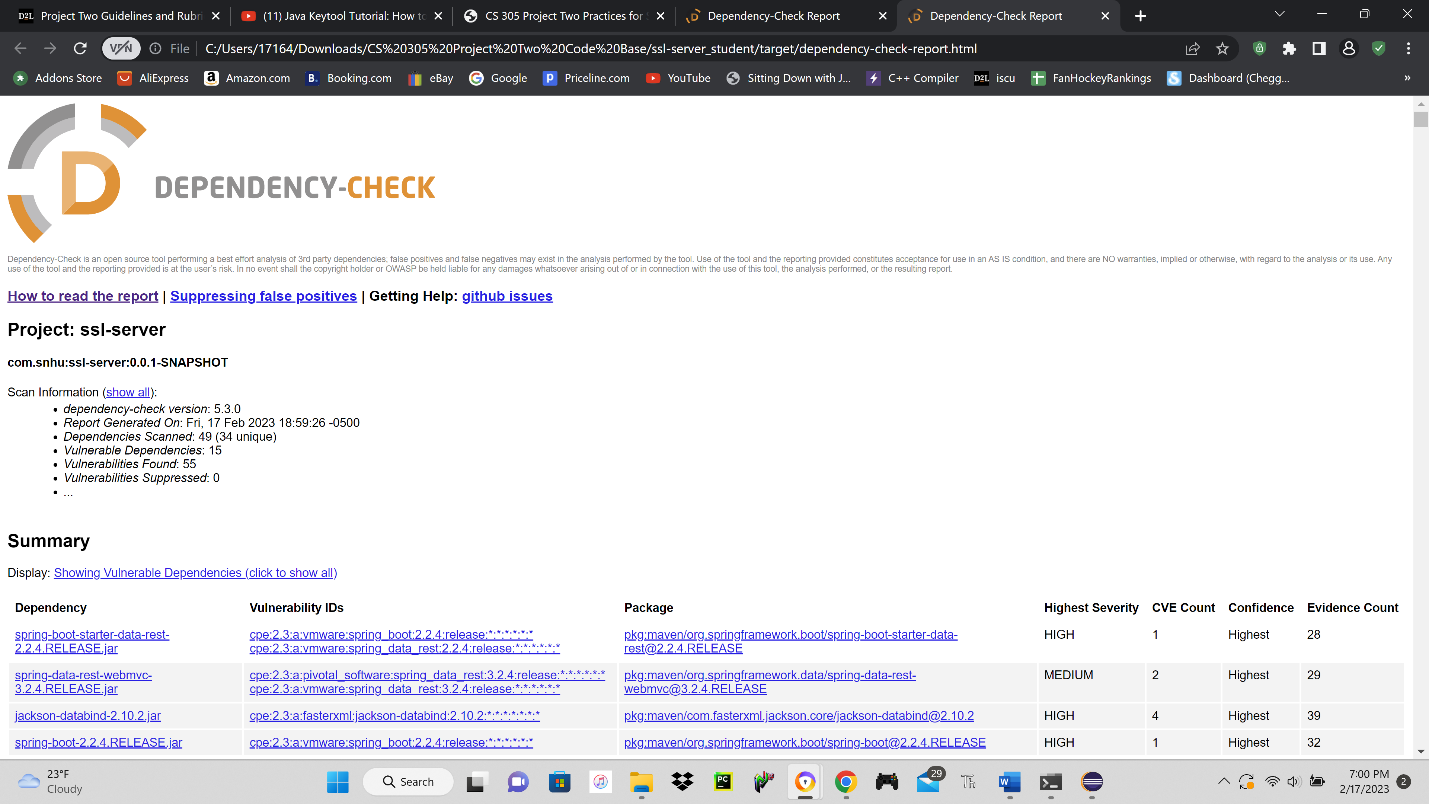
## Secure Communications

Graphical user interface, text, application

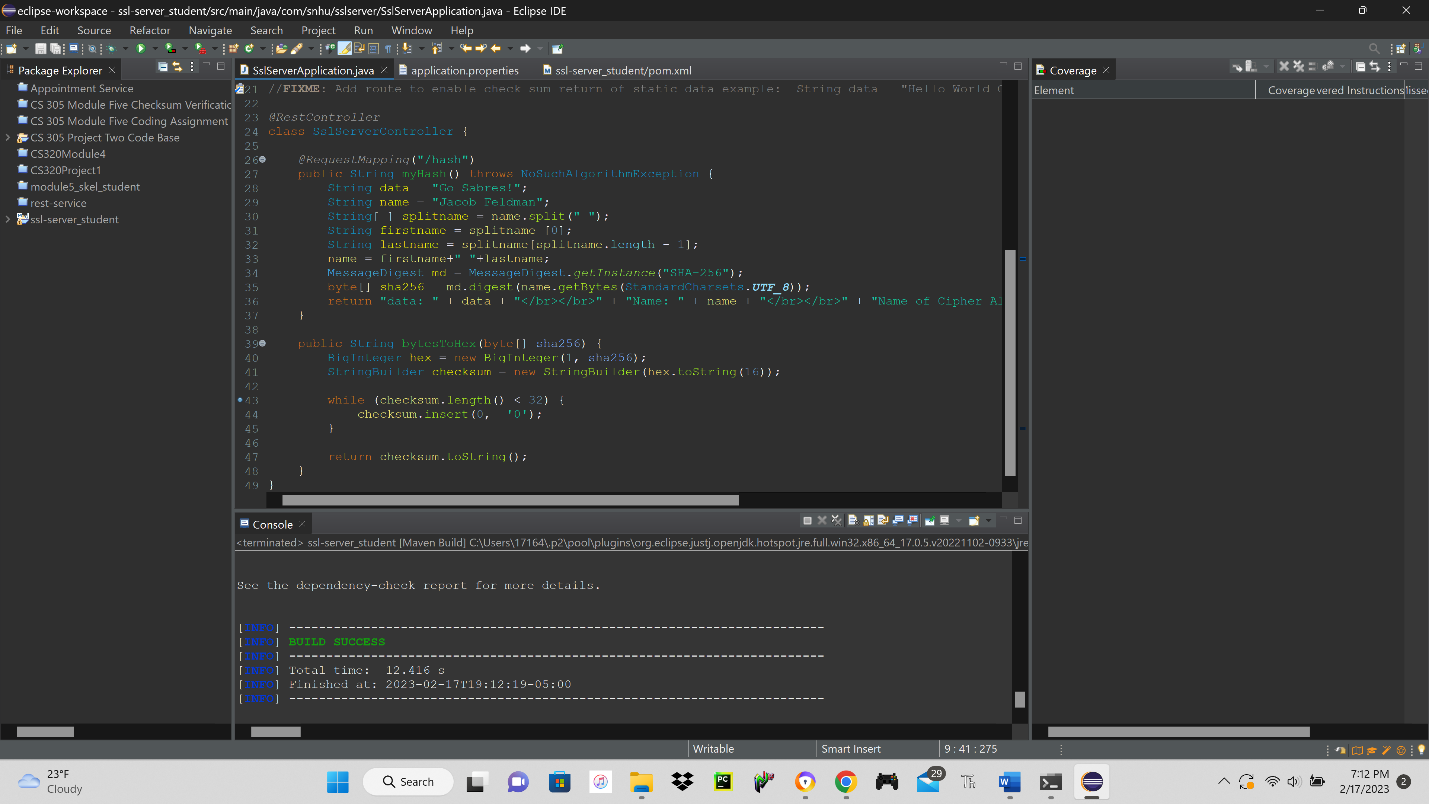
Description automatically generated

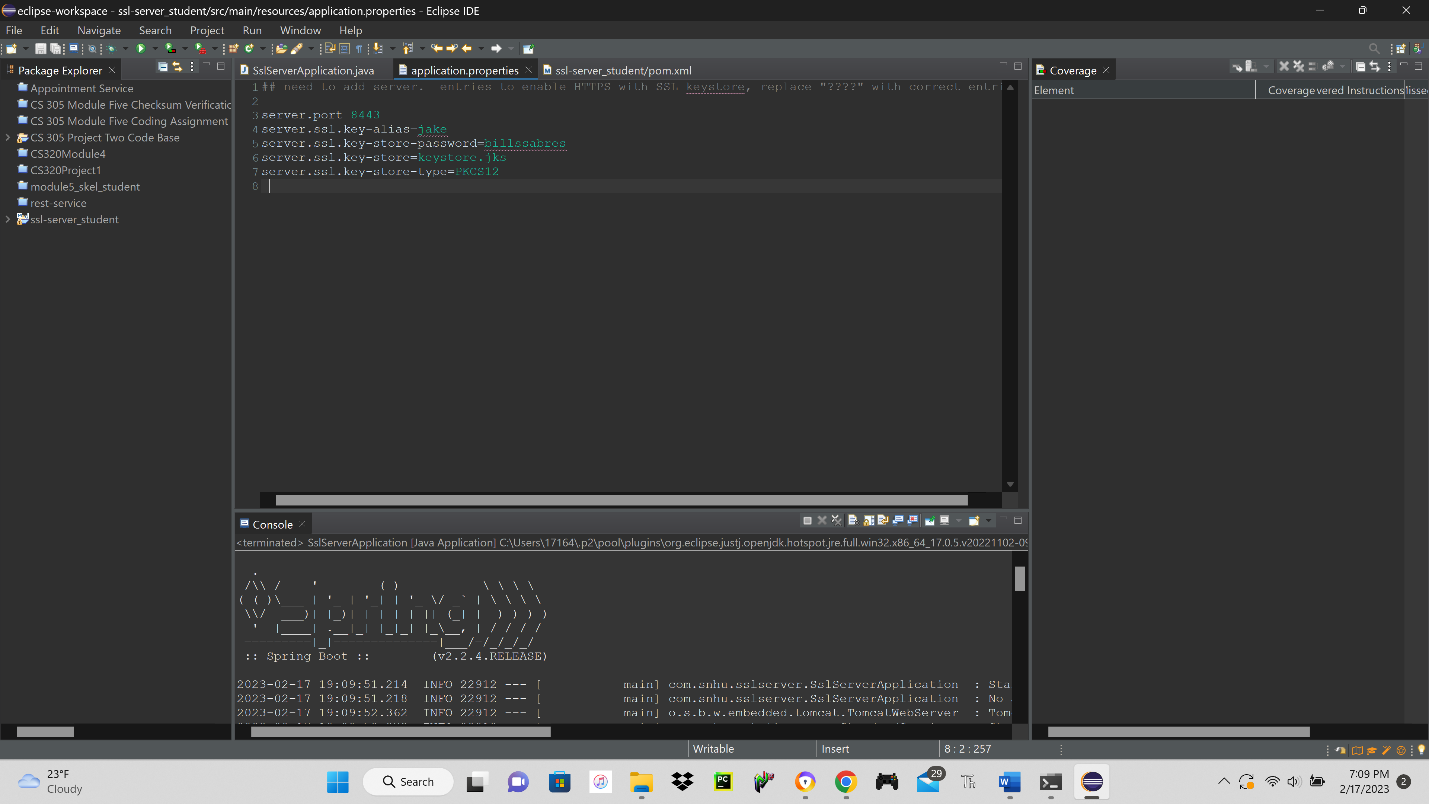
## Secondary Testing





## Functional Testing





## Summary

To refactor the code properly, you must comply with all of the security testing protocols. I have carefully reviewed the Vulnerability Assessment Process Flow Diagram, and I have addressed several concerns regarding software security. Without a proper encryption tool, both users and developers are at risk of losing sensitive information. The implementation of an encryption algorithm (AES) ensured that any transfer of data would be secure for all parties involved.

I was able to use HTTPS for a secure connection by incorporation self-signed certificates. These certificates enable encryption but are not totally susceptible to malicious attacks. The user cannot be certain that they are connected to the proper link, as a third-party can redirect their connection using the same holder name for the certificate. It is important to note what vulnerabilities are potential throughout this process for future endeavors.

To ensure that the application was secure, I had to verify that the certificates were created correctly. I also needed to make sure that the vulnerabilities in the dependency check report were fixed. I did this by refactoring the pom.xml file and generating a second report. Another security measure I implemented was testing my hash function. I made sure that it worked as intended and it gave me confidence in my report and application.

## Industry Standard Best Practices

Applying industry standard best practices is crucial for creating and maintaining a functioning application. One of the best practices to adhere by is keeping things updated and in their current versions. By refactoring my pom.xml file and generating a second report, I had updated my code and fixed the known vulnerabilities. Another practice I used is restriction of access, known as least privilege. This means that only a selected group of users will have access to certain areas of the application. I did this by using keys. Keys can be shared between users and “lock/unlock” encrypted data. By using different keys, you can separate access to users in one action.

A company can benefit in numerous ways by applying the industry standard best practices. For any business to succeed online, they need to have their customer’s trust. This can only be obtained by taking the proper precautions and measures to secure their programs and data. For example, an online banking company would never survive if their user’s money/information was stolen regularly. Their top priority should be to protect their users if they want their continued business. Another good example (since it is tax season) is that a company like TurboTax or H&R Block needing secure connections. Filling out tax forms online can reveal vital personal information. It is in these companies’ best interests to ensure that their users are 100% safe.

Simply stressing the importance of safety and security can have lasting impressions on people. I think that this is a necessary mindset for businesses creating and maintaining software applications. As the reliance on internet communications continues to increase, the need for better and more adaptive security measures becomes increasingly more important. These measures are absolutely vital for companies and users alike and will never cease to be a top priority in the world of software development.