****

# 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** | **[Date]** | **Alexander Diederich** |  |

## 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

Alexander Diederich

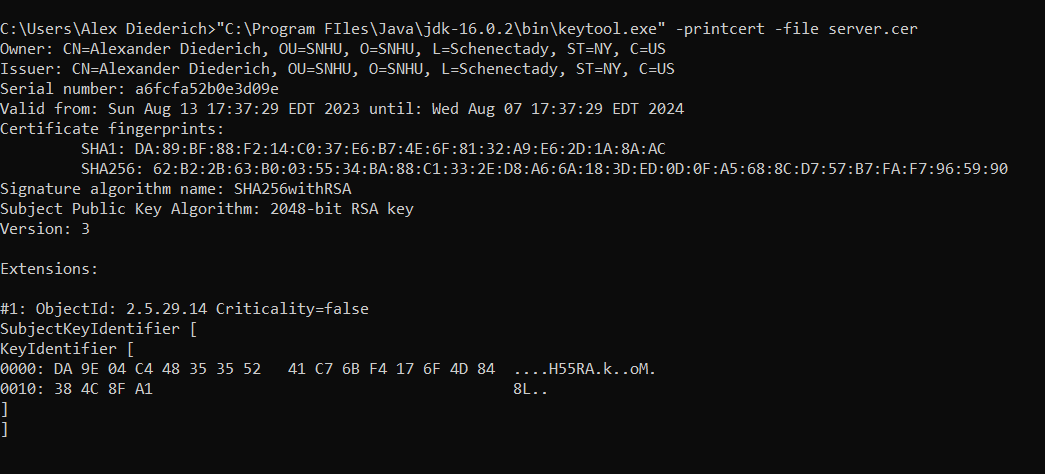
## Algorithm Cipher

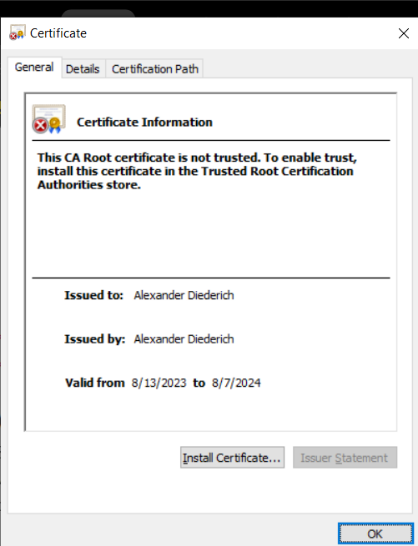
The primary purpose of Artemis Financial is to offer clients around the world financial programs. I personally recommend using SHA-256 for the encryption algorithm cipher due to this goal of Artemis Financial. This particular cipher protects all information from being accessed by an outside user. It is one of the best ciphers because it is practically hard to penetrate or break. It would end up taking years to try and brute force open this cipher. When dealing with the communication between financial institutions, SHA-256 is typically the suggested cipher of choice, due to the fact that the hash’s that come from the cipher are randomized every time. When creating a hash function, the value of the input is compressed before it is even used. Hash value is the name for compressed data. The bit levels determine the length that the encryption will be.

The quantity of encryption combination possibilities determines a 256-bit encryption, making it more difficult for hackers to gain access to any information due to the fact that the encryption uses random numbers. Symmetric keys are regarded for being one of the most basic forms of encryption, and an advantage of them is that it takes less time to execute, and it only requires one key to decrypt it. In light of these facts, AES-256 typically uses symmetric keys and will encrypt plain text using a key. Asymmetric keys on the other hand require two keys, making them seem more secure than symmetric, and this is why we typically use asymmetric for internet communication. Thanks to encryption it is now possible to safeguard data from illegal access, which is good seeing as humans have always prioritized security above everything else.

## Certificate Generation

Insert a screenshot below of the CER file.





## Deploy Cipher

Insert a screenshot below of the checksum verification.

A screenshot of a computer

Description automatically generated

## Secure Communications

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

A screenshot of a computer

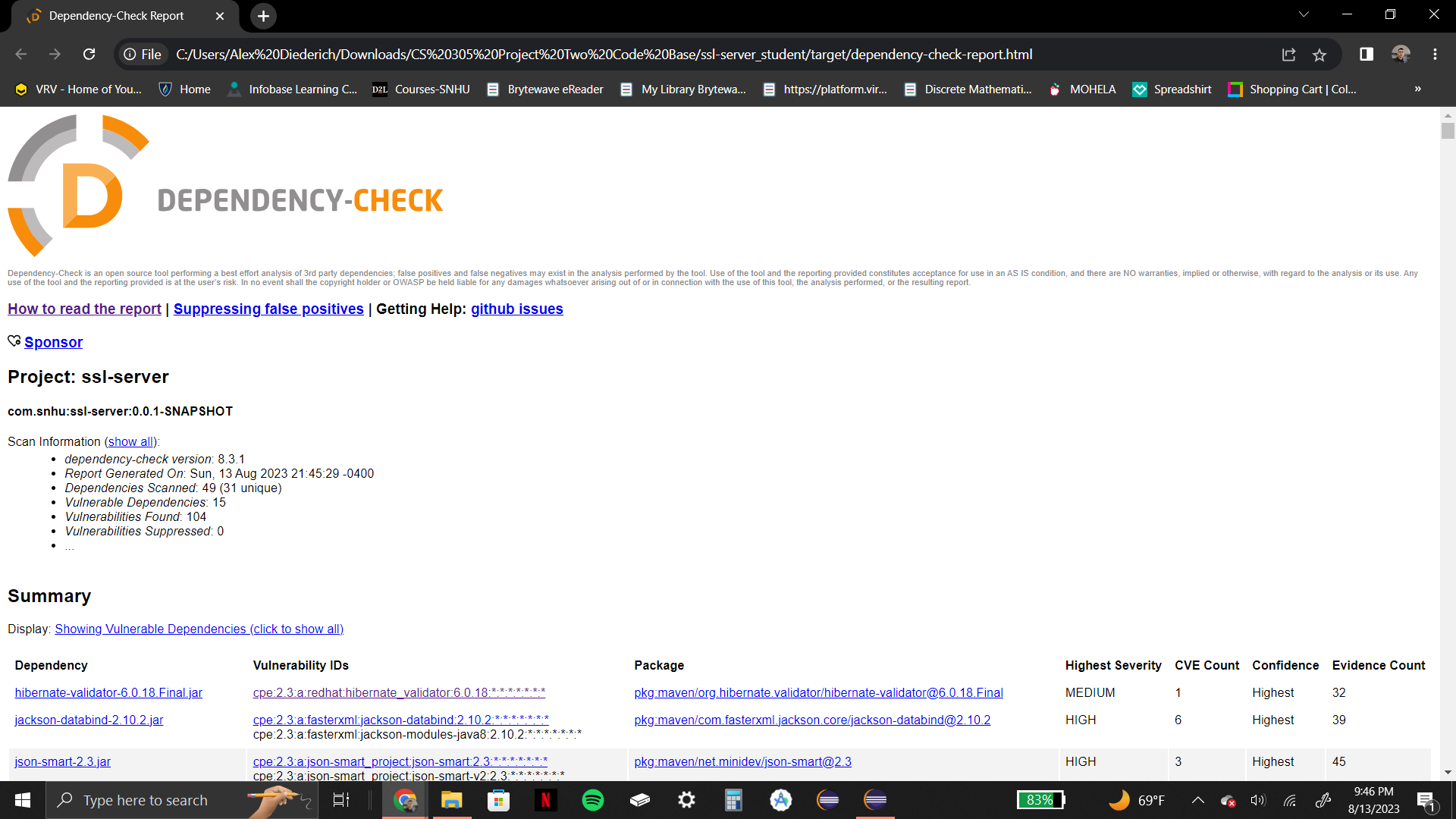
Description automatically generated

## Secondary Testing

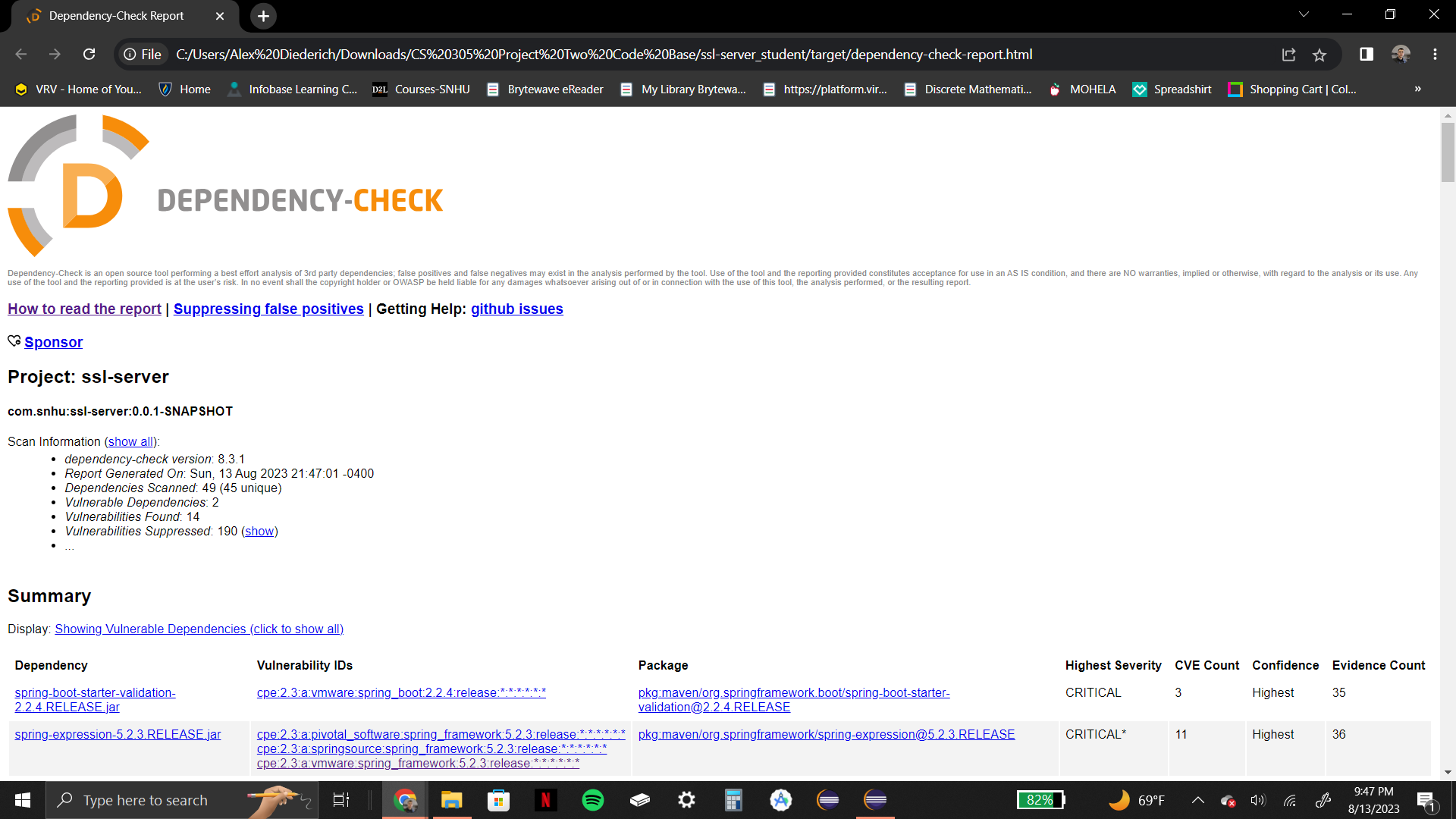
Insert screenshots below of the refactored code executed without errors and the dependency-check report.

I lowered the amount of vulnerabilities to a much lower amount.

Before suppression:

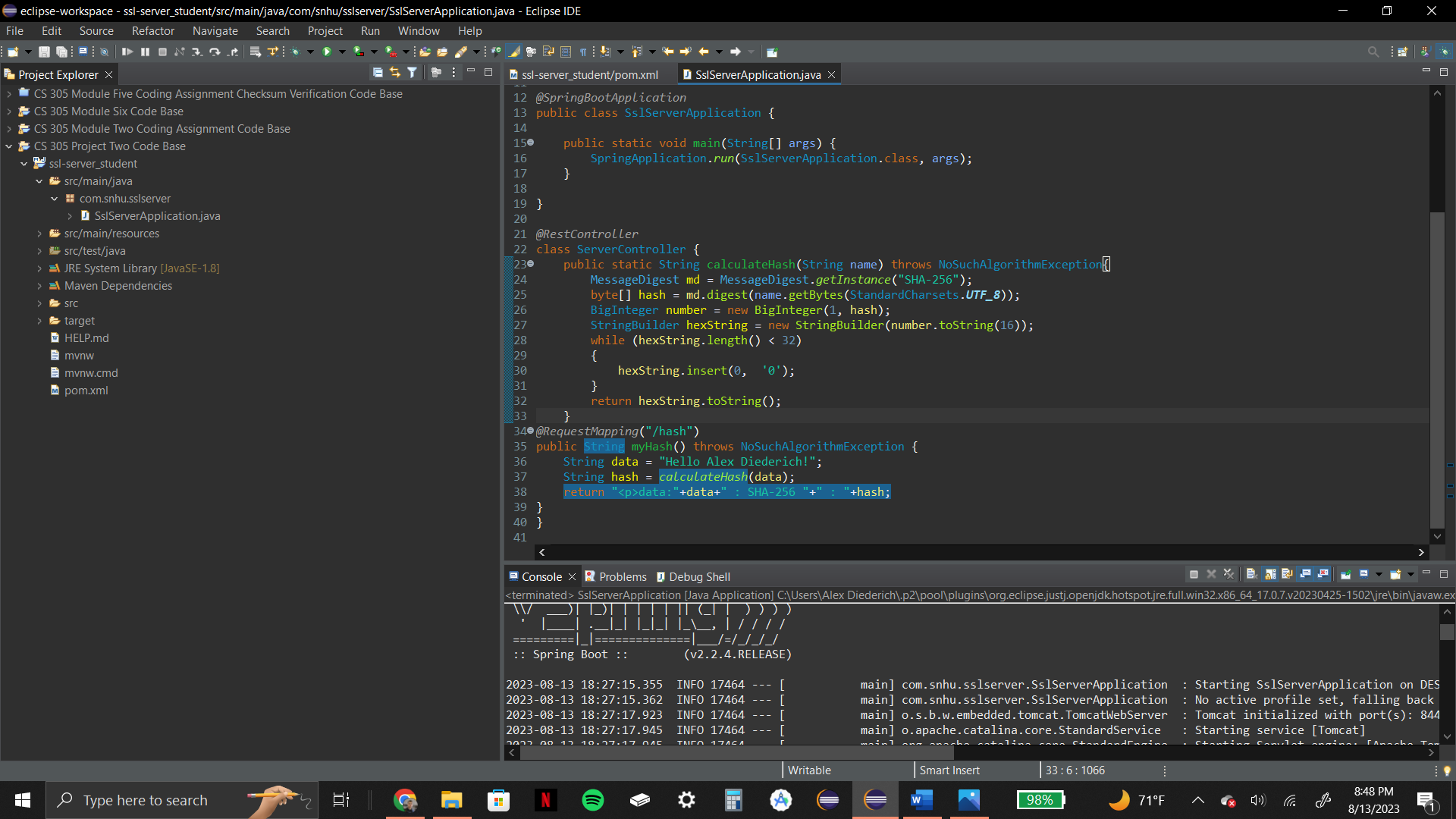


After suppression:



## Functional Testing

Insert a screenshot below of the refactored code executed without errors.



## Summary

Our application has self-signed certificates, enabling the usage of HTTPS, were the security key was enhanced. We also refactored the code for the pom.xml file to erase as many vulnerabilities as possible.

In order for us to be able to use HTTPS when our application gets up and running, the first step was to ensure that the certificated were generated correctly. By ensuring that our website is secure, users will be able to trust that they are using our service and not a fake, meaning that this security helps the business.

Next step was to make sure that our hashing function worked properly and verifying this with checksum. By making sure that the data that our customers put into our system gets scrambled correctly and is difficult to recover helps improve the security of the business and will help clients to trust us.

The final step was to patch up the vulnerabilities that were popping up. Being able to get rid of these vulnerabilities helps ensure that our business is prepared and that the application’s internal workings are functioning and as current as possible as intended.

Making sure that our system and software are up to date is a great practice for maintaining security for our app. Doing this, protects our outdated systems from being attacked, and is essential for us to be able to use least privilege. Artemis Financial is protected from attacks from within, because users will not be able to access data that is needed for them to perform their duties.

## Industry Standard Best Practices

I followed industry standards by trying to keep the code as short as possible by writing only as many lines as needed. To maintain security I got rid of as many vulnerabilities as I could without jeopardizing the application, and I did this by adding a new file to help get rid of the vulnerabilities. I ensured that there were no repeat functions in the code to avoid unnecessary repetition.

The reason that it is important to do these things is because it makes it easier for people to understand the code. The easier it is for us to understand our code the easier it is to catch any mistakes that we made. We should always try to follow the standards of our fields so that we can make successful applications that people will want to use.