****

# 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]** | **[Your Name]** |  |

## 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

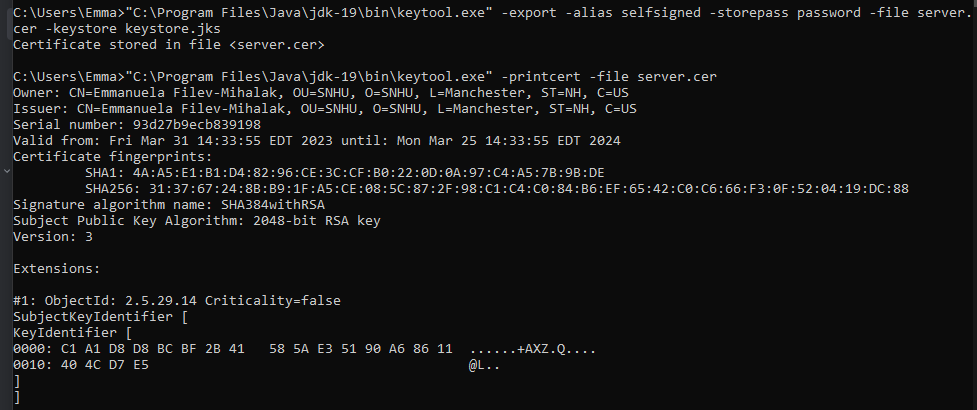
Emmanuela Filev-Mihalak

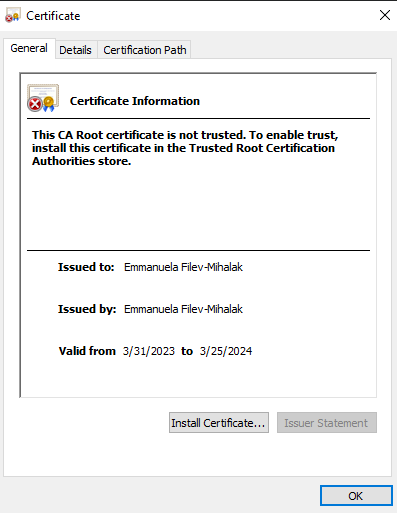
## Algorithm Cipher

I recommend using SHA-256 for this program. SHA-256 currently meets the standard for the safest recommended algorithm ciphers to use in programs. It uses 256-bit keys and is asymmetric, making it one of the most secure hash algorithms.

## Certificate Generation

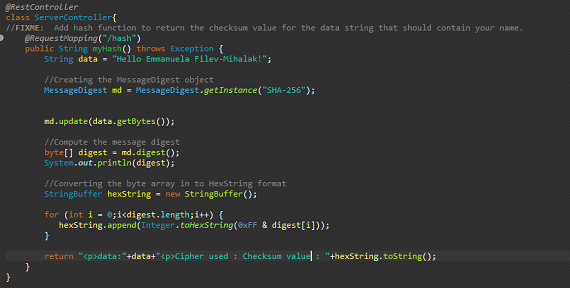
Insert a screenshot below of the CER file.





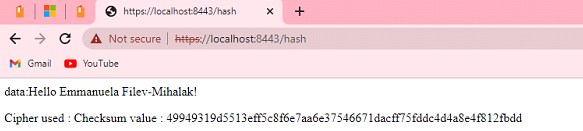
## Deploy Cipher

Insert a screenshot below of the checksum verification.



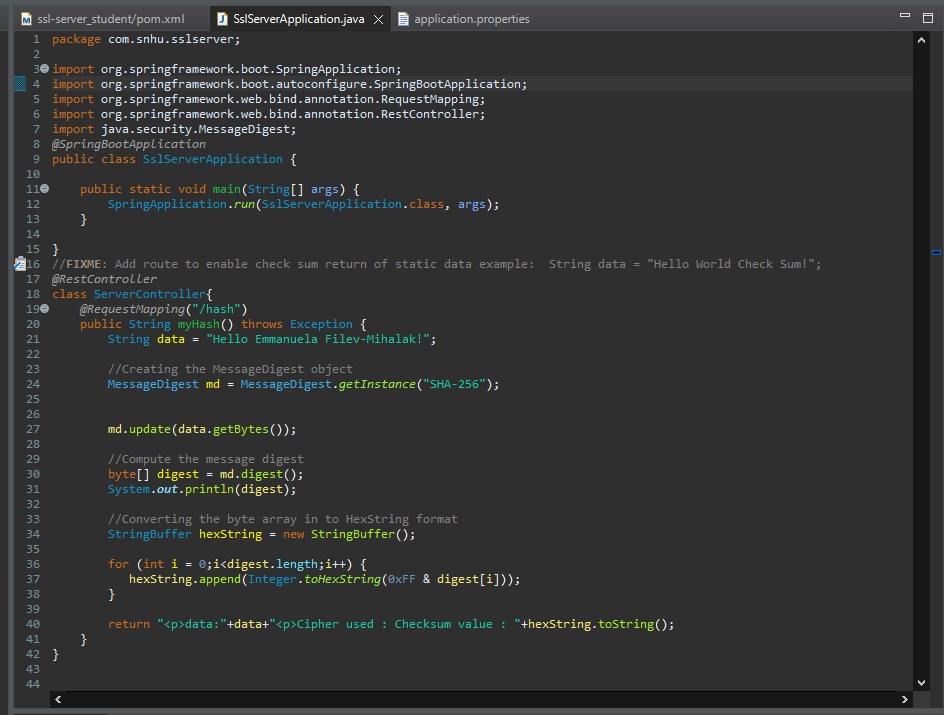
## Secure Communications

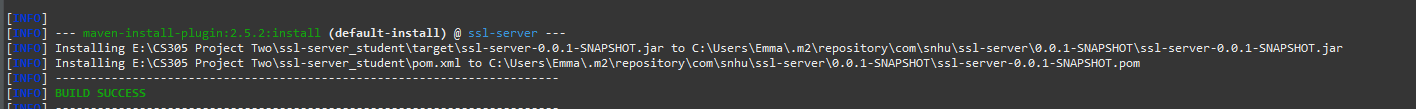
Insert a screenshot below of the web browser that shows a secure webpage.

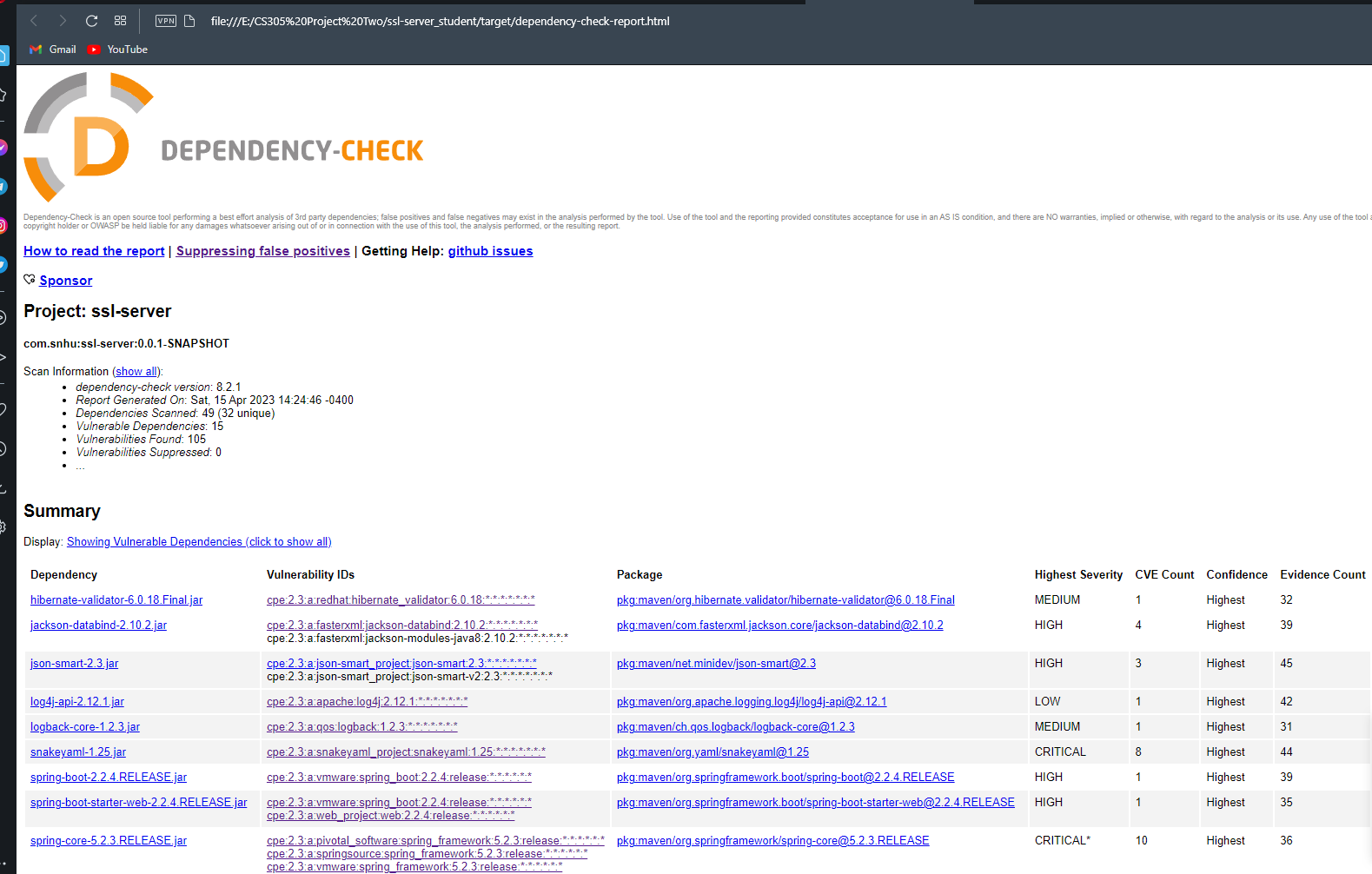


## Secondary Testing

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







## Functional Testing

After manually checking the code, no additional refactoring was required.

## Summary

To create a secure application, I first used a self-signed certificate that only I would use for testing. I then updated my application.properties file to use all of the properties that my certificate contains. I also made sure the application runs on 8443 for security. According to the vulnerability assessment process flow chart, I maintained secure API interactions, used cryptography effectively, and followed secure coding practices and patterns.

## Industry Standard Best Practices

I made sure I followed industry standard best practices by following the vulnerability assessment process flow chart, referring to Java security standards, and simply following the requirements set for the application.