****

# 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** | **2-20-2025** | **Jonathan abramavage** |  |

## 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

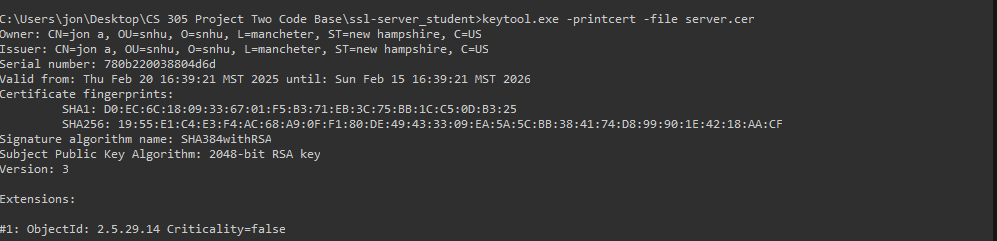
Jonathan abramavage

## Algorithm Cipher

With Artemis financial looking for good security on the web site and application we will use a SHA-256 algorithm. This will stop attacks by using a checksum that verifies the file to see if its valid using the hash function and with it being 256 bits there is a ton of different key combinations so your key will be different than other peoples. This algorithm will use a random number generator to make the check sum. We will use asymmetric to commutate. This will encrypt public keys then you will need a private key to decrypt. This is useful as any one can make the data encrypted and only senior managers will have the private keys.

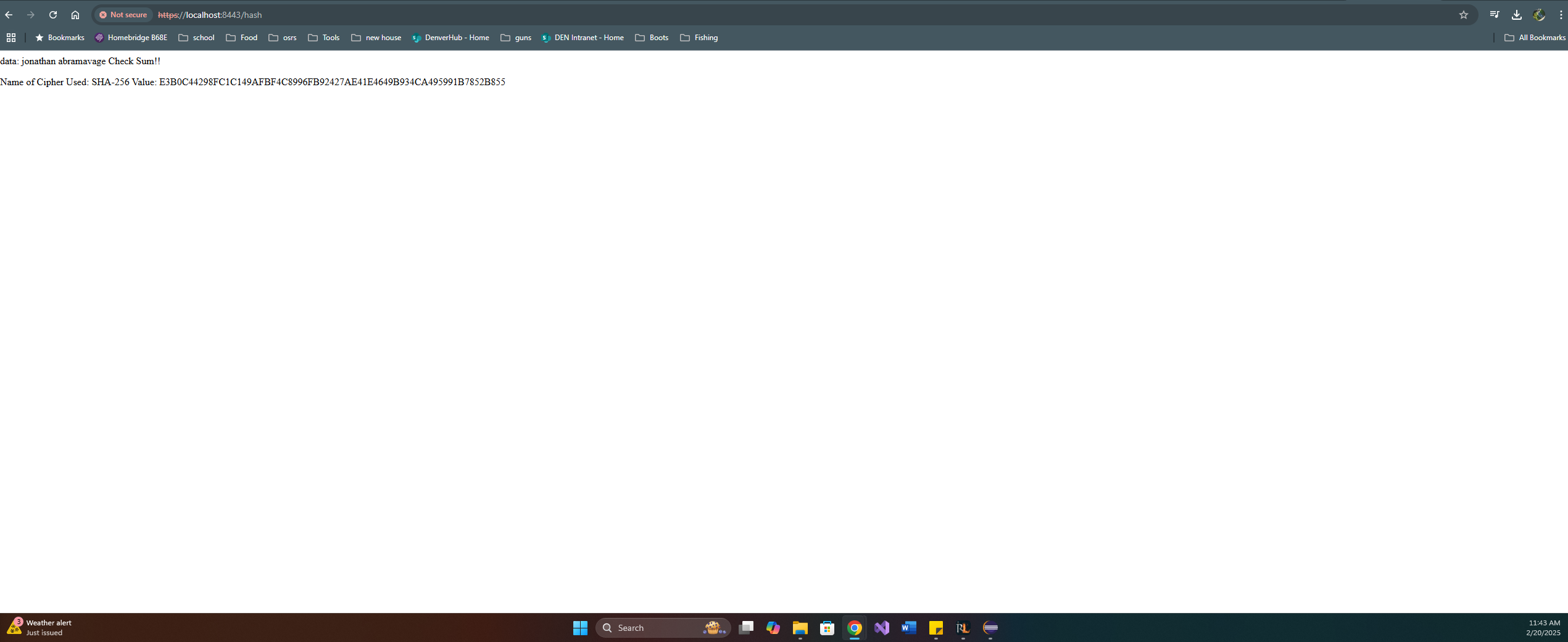
## Certificate Generation

Insert a screenshot below of the CER file.



## Deploy Cipher

Insert a screenshot below of the checksum verification.

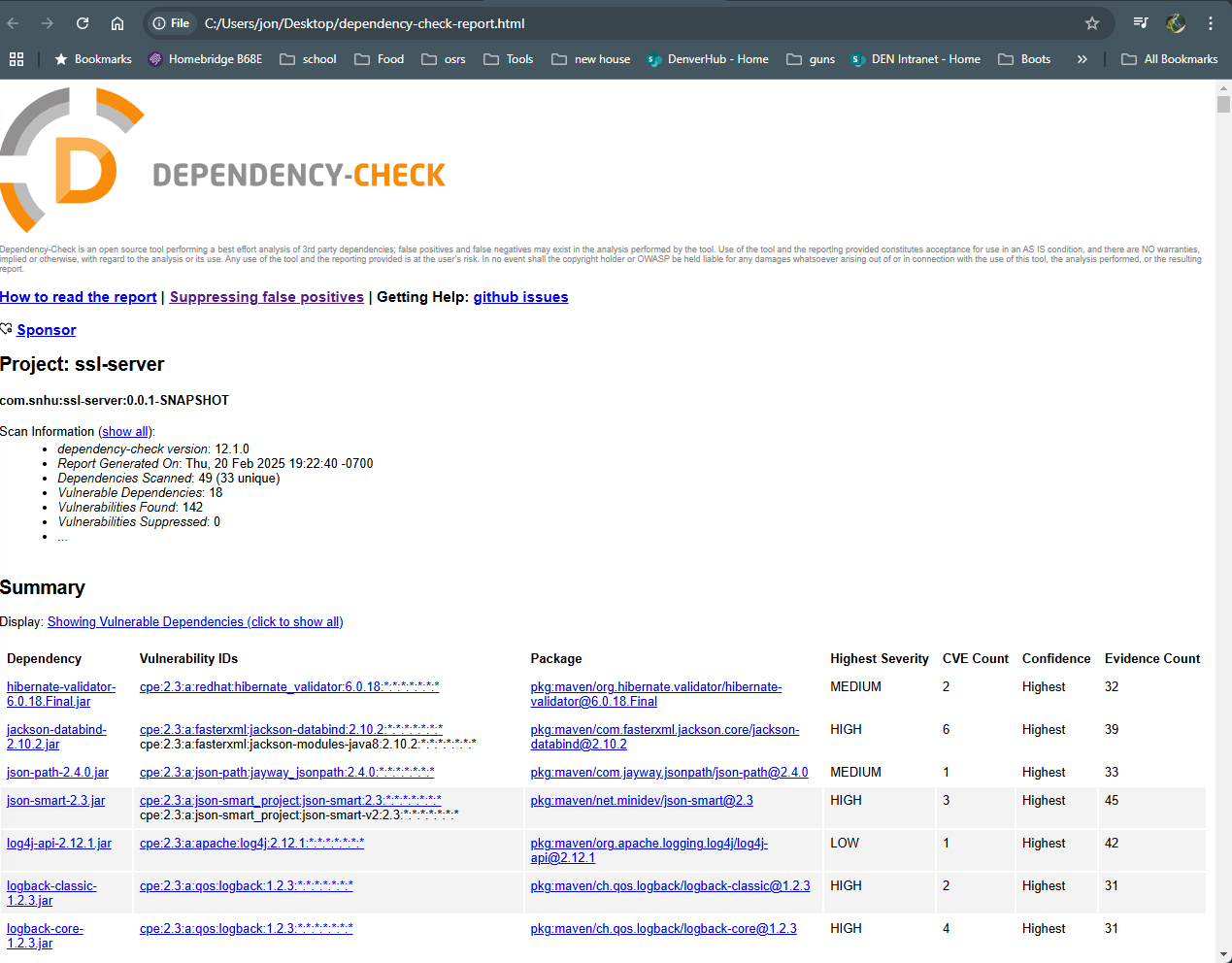


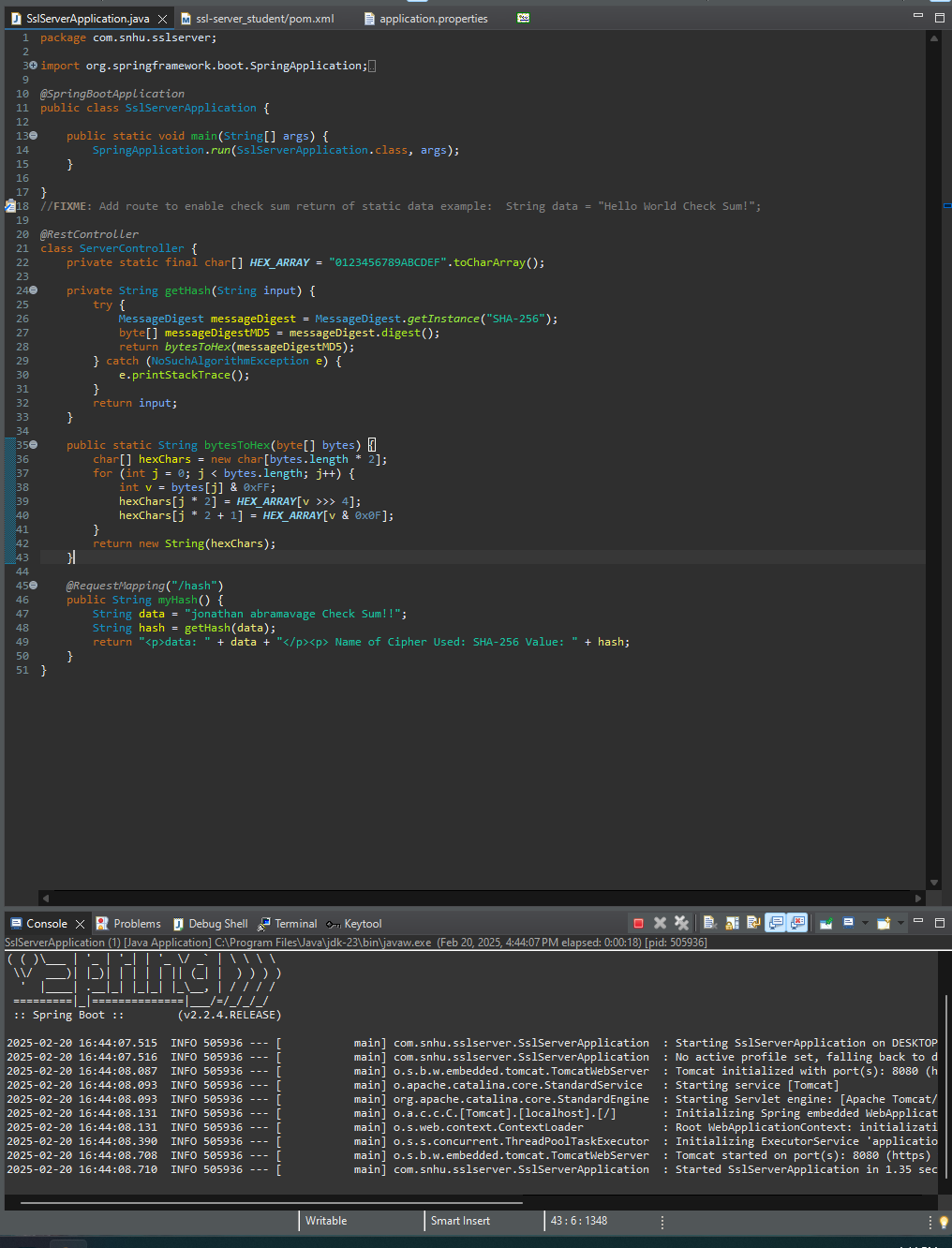
## Secure Communications

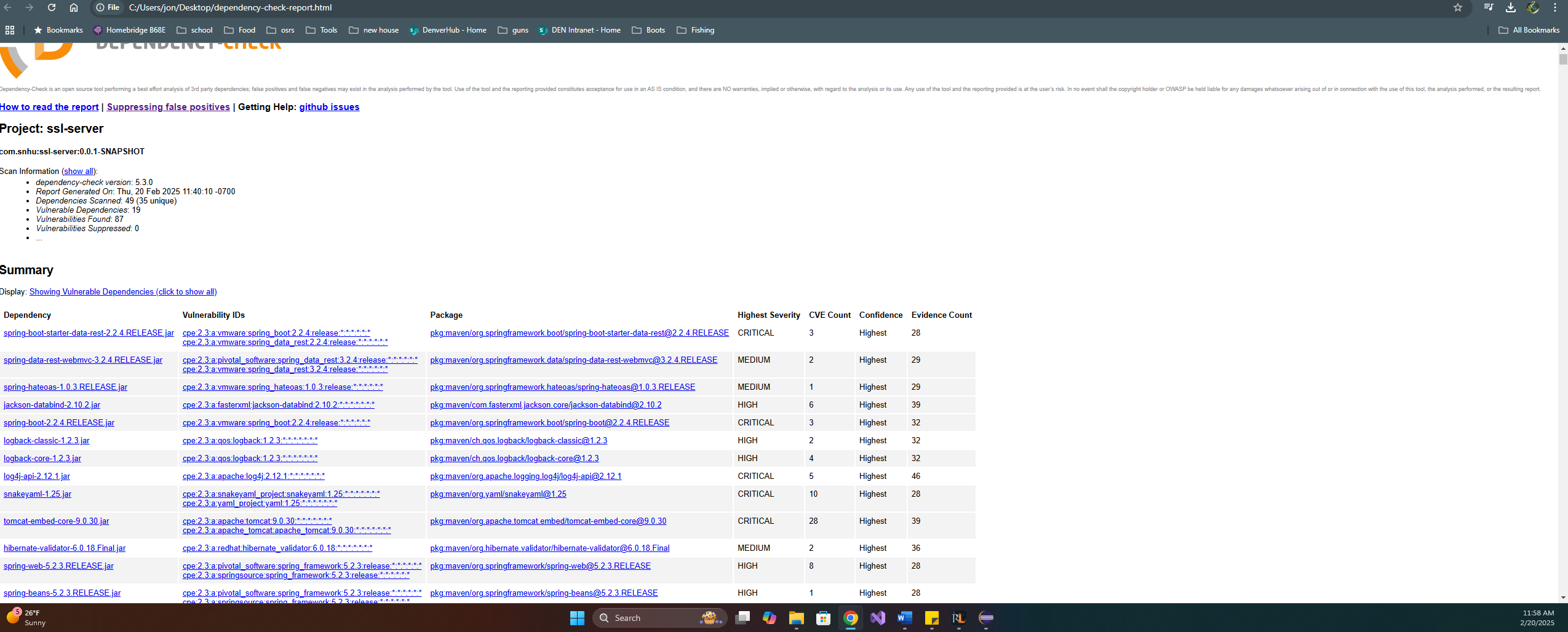
Insert a screenshot below of the web browser that shows a secure webpage.

I tried installing the cer file and could not get it to become secure

## Secondary Testing

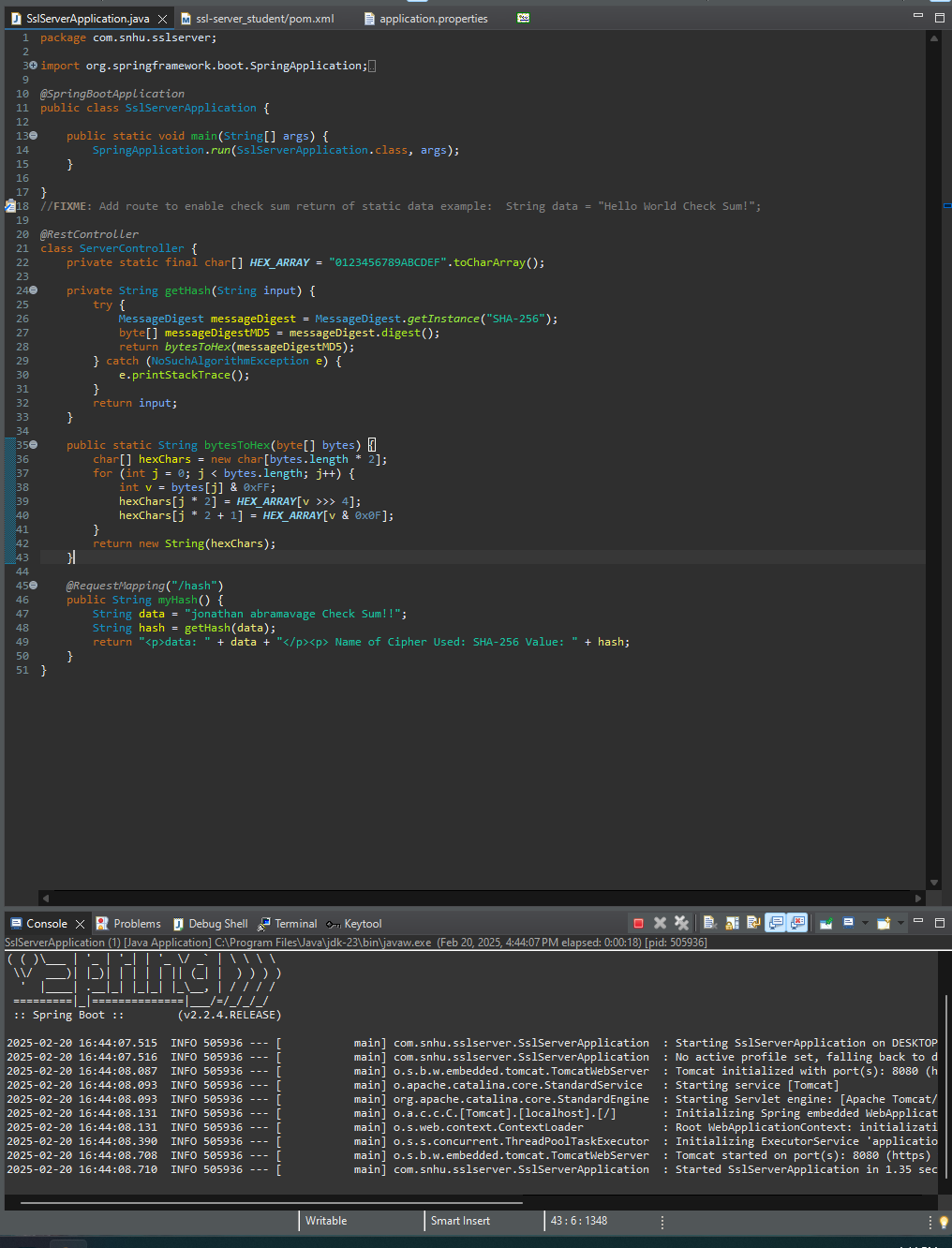
Insert screenshots below of the refactored code executed without errors and the dependency-check report. 





## Functional Testing

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



## Summary

we made an app that use self singed certificates this allows us to use a secure browsing protocol like https. We use the pom file to run a dependency report that shows all the vulnerabilities. We added a rest controller that serves as security for the hash rest endpoint. We used an updated version of 12.1.0 instead of the 5.3.0 we did this to find any new issues that be harmful. We chose a cipher that is the industry standard

## Industry Standard Best Practices

When we look at the standard to prevent cyber-attacks, we look at Symmetric encryption or asymmetric to decide if we want the same keys or different keys. We use the government standard aes 256 bit which uses block ciphers. We look at adding regular security patches to stop new threats. We limit the people that have access to the data as a need to know. We use in put validation to stop certain types of attacks. They also use an event logging to check what is going on and who did what this letts us know if someone is doing something malicious