****

# 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** | **6/21/24** | **Luke Kundinger** |  |

## 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

Luke Kundinger

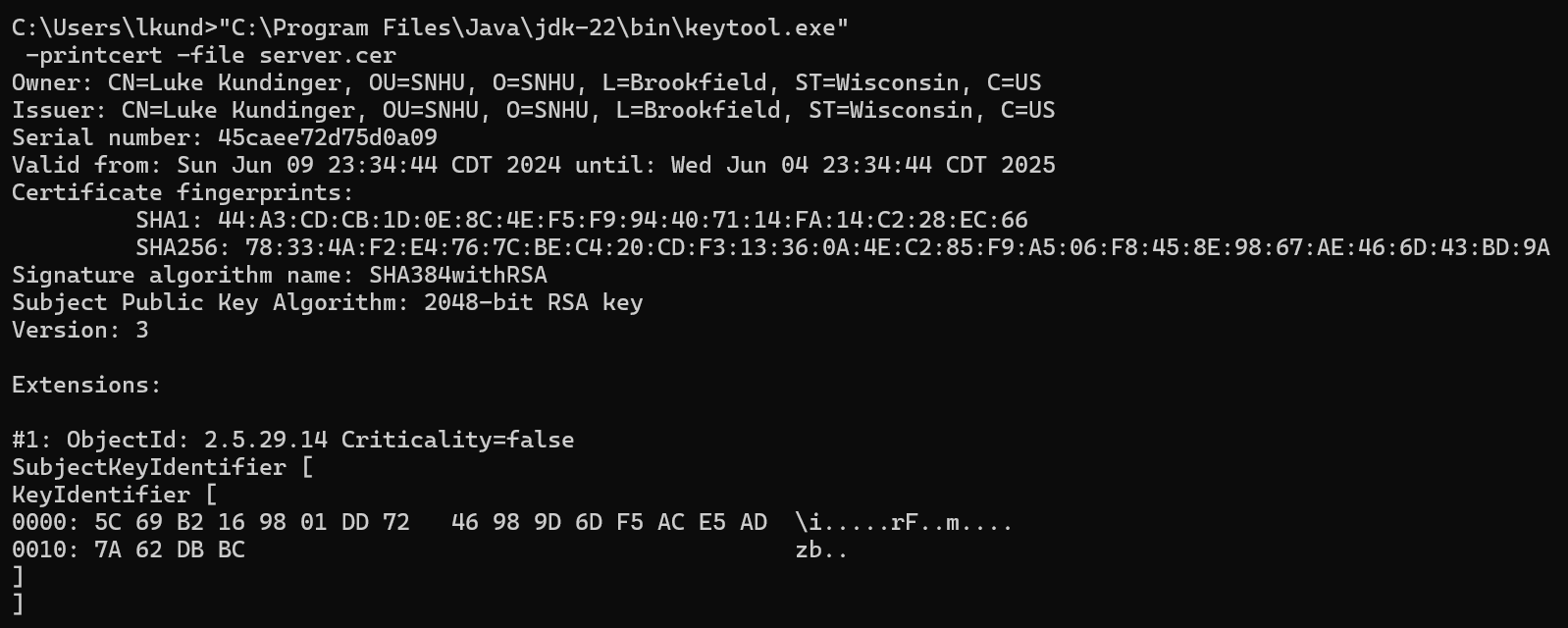
## Algorithm Cipher

The encryption algorithm cipher that I recommend deploying is SHA-256. It uses six hash functions and 256 bits. SHA-256 is a very good algorithm cipher for avoiding collisions. This is important because it is near impossible to get the same hash value for more than one input. Strong resistance to collisions is a key attribute of a secure algorithm cipher.

Random numbers play a role in generating secure keys for unpredictability. Symmetric keys use the same key for encryption and decryption, where non-symmetric keys use a public key and a private key. The process of encryption is much faster for a symmetric key compared to a non-symmetric key because of its complexity. The increased complexity of the non-symmetric key provides more security. Encryption algorithms were first used by Julius Caesar by shifting letters in the alphabet. Over the years encryption algorithms adopted the use of keys to make encryption more complex. In modern times the use of symmetric and non-symmetric keys is widespread, with AES and RSA being the most common. Encryption algorithms have become extraordinarily complex, and experts have estimated that RSA 2048 can be broken by 2035 with quantum computing.

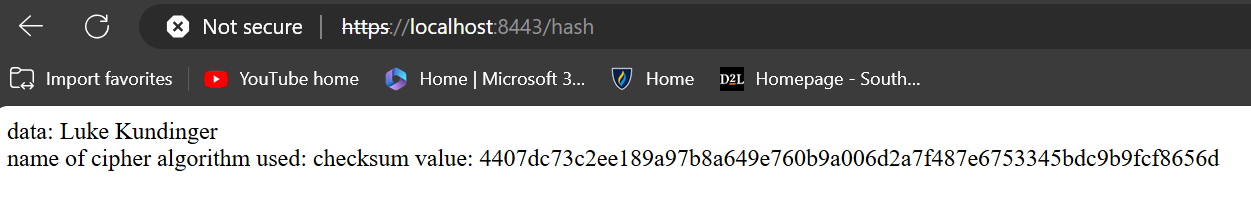
## Certificate Generation

Insert a screenshot below of the CER file.



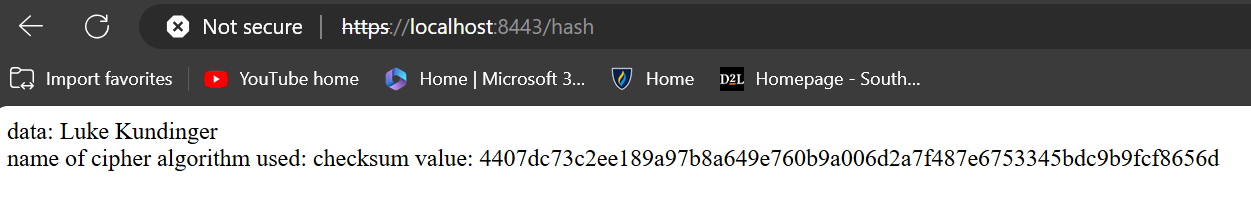
## Deploy Cipher

Insert a screenshot below of the checksum verification.



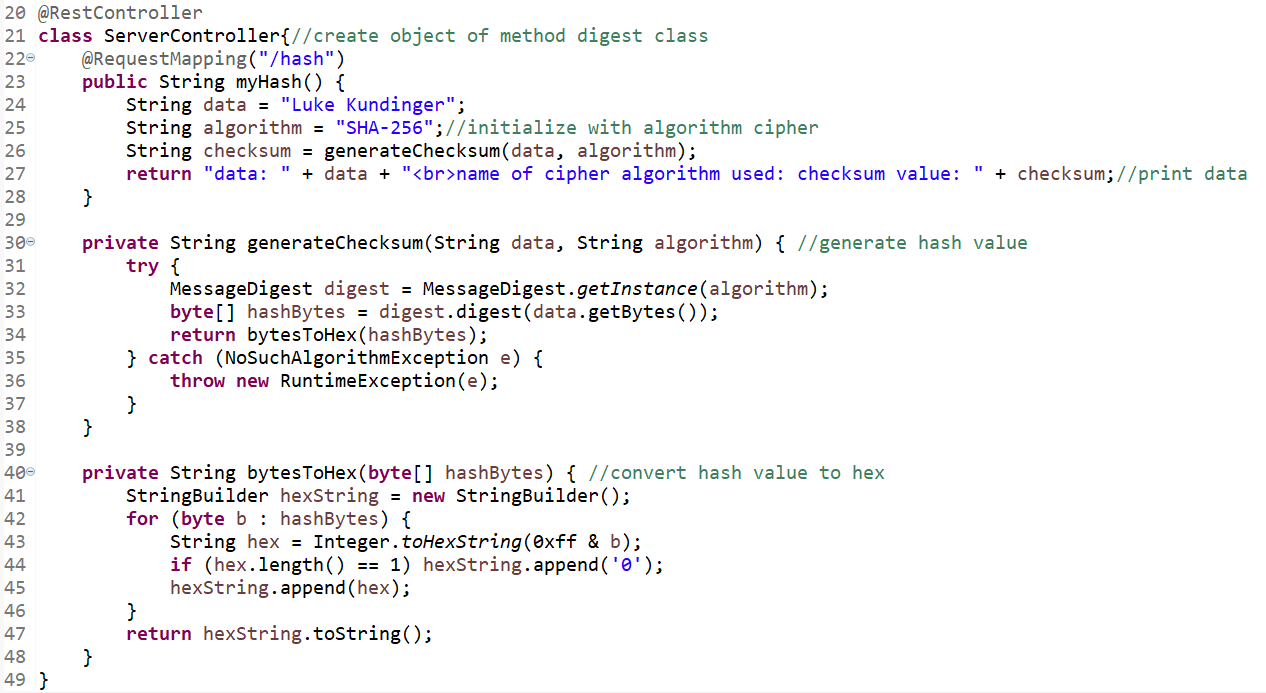
## Secure Communications

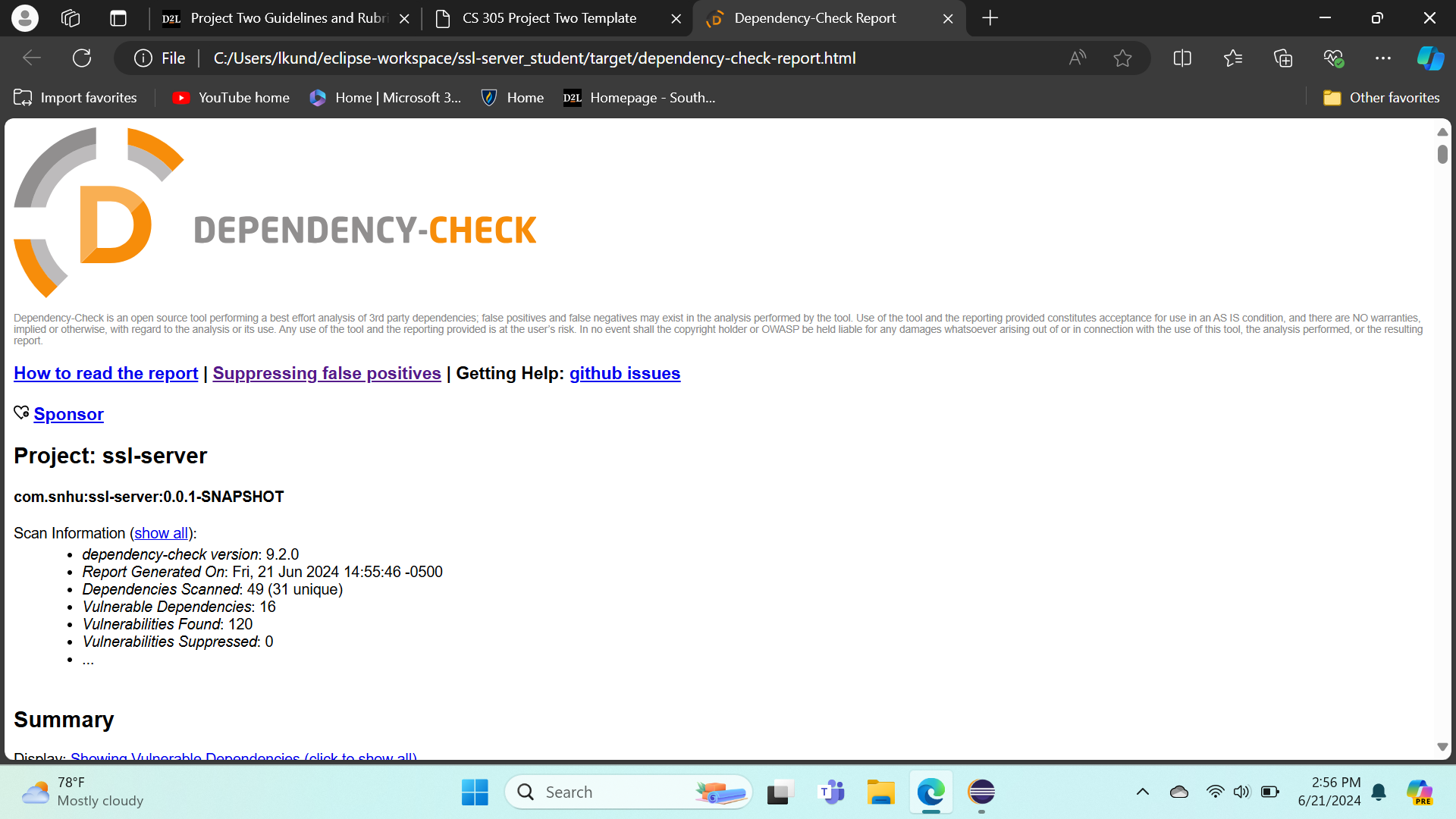
Insert a screenshot below of the web browser that shows a secure webpage.



## Secondary Testing

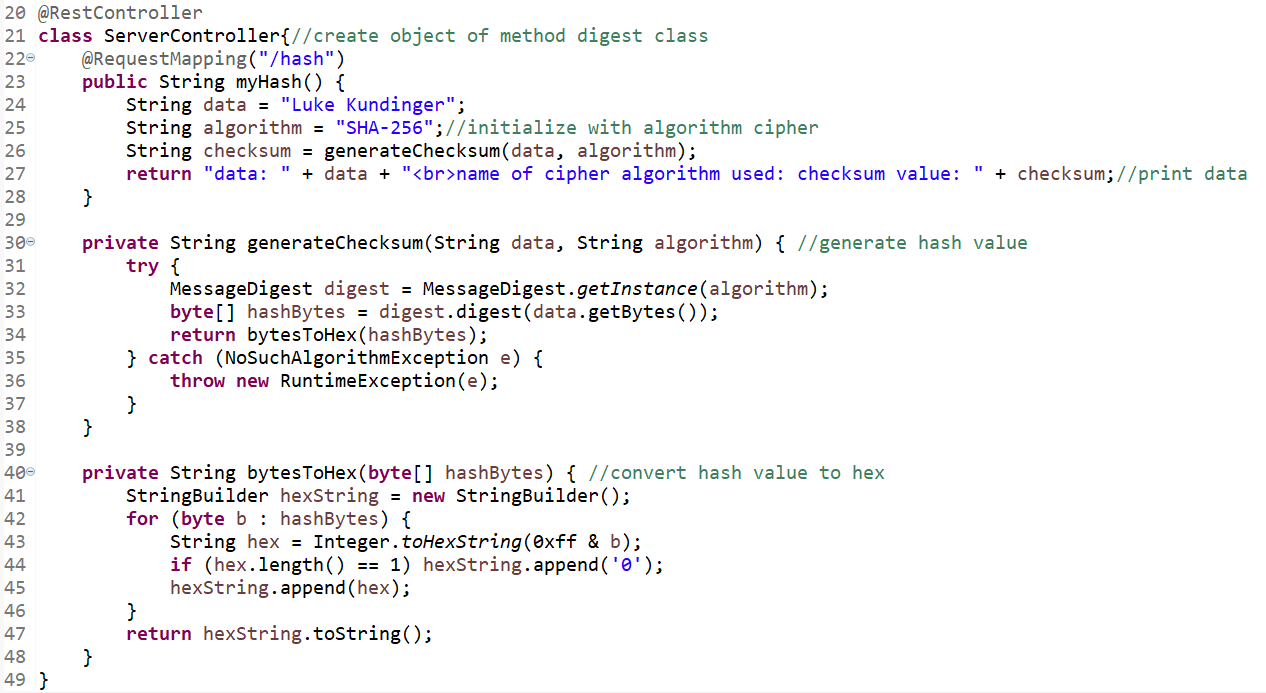
Insert screenshots below of the refactored code executed without errors and the dependency-check report.





## Functional Testing

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



## Summary

By refactoring the code, I addressed security with cryptography, client/server, and code quality. By using SHA-256, I used encryptions to keep the data secure. The client/server pattern was used by generating a certificate and connecting the application to the web. Code quality was addressed mostly by adding the encryption and client/server functions, but also for changing the dependency check version. The process of adding security to the code was implementing cryptography and use of client/server communication. A dependency check was also done to review the security and find vulnerabilities.

## Industry Standard Best Practices

Industry standard best practices include manually inspecting the code and running regular dependency checks. When changes are made to the code it is important to make sure the program still functions as it should. Dependency checks should be conducted regularly to ensure the program isn’t vulnerable to new dependencies.