****

# Practices for Secure Software Report

**Table of Contents**

Practices for Secure Software Report 1

Document Revision History 3

Client 3

Instructions 3

Developer 4

1. Algorithm Cipher 4
2. Certificate Generation 4
3. Deploy Cipher 4
4. Secure Communications 5
5. Secondary Testing 5
6. Functional Testing 6
7. Summary 6
8. Industry Standard Best Practices 7

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **December 9 2022** | **Roderick Fisher** |  |

## Client



## Instructions

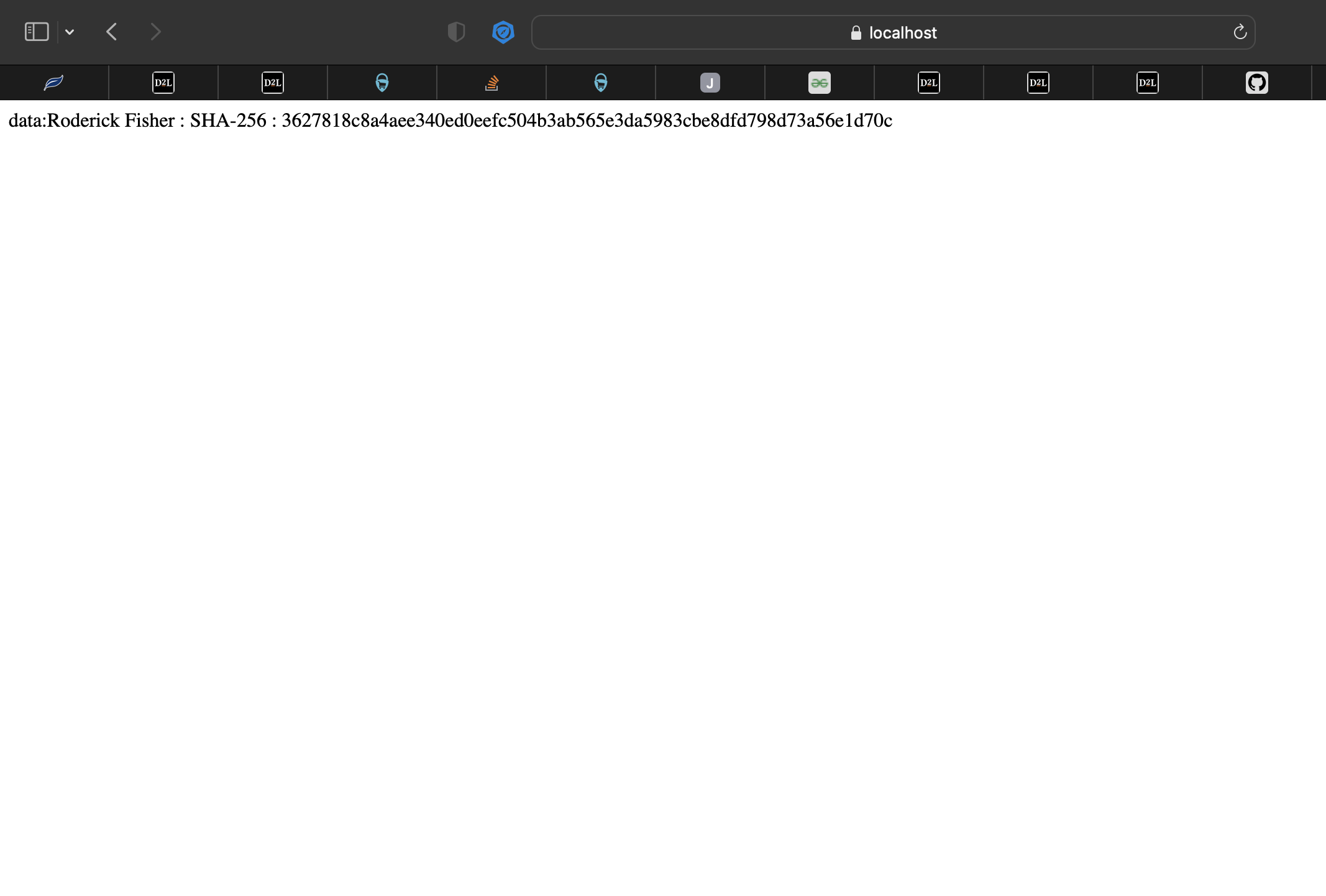
Submit this completed practices for secure software report. Replace the bracketed text with the relevant information. You must document your process for writing secure communications and refactoring code that complies with software security testing protocols.

* 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

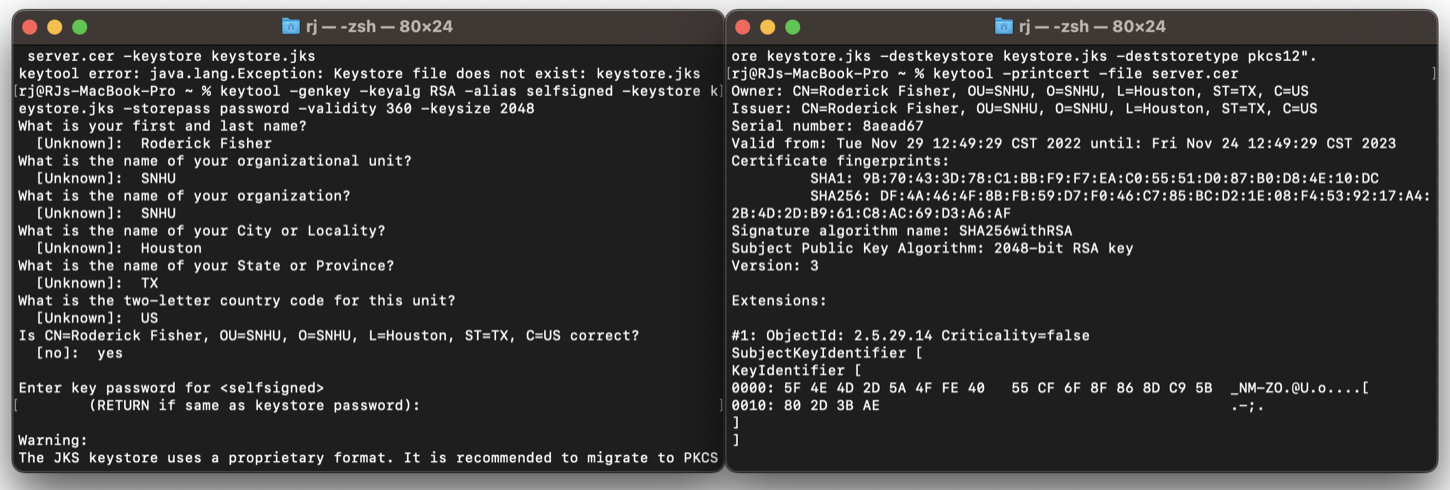
Roderick Fisher

## Algorithm Cipher

With Artemis being a company that is holding important financial data it is of upmost importance that that data be protected. To protect this data I would recommend the AES algorithm. AES is already implemented in software and hardware throughout the world to encrypt sensitive data. AES includes three block ciphers: AES-128, which uses a 128-bit key length to encrypt and decrypt a block of messages, AES-192, which uses a 192-bit key length to encrypt and decrypt a block of messages, and AES-256, which uses a 256-bit key length to encrypt and decrypt a block of messages.Each cipher encrypts and decrypts data in blocks of 128 bits using cryptographic keys of 128, 192 and 256 bits, respectively. It is important to keep in mind that AES uses symmetric or secret keys, which means that the ciphers use the same key for encryption and decryption. Specifically I would use AES-128 over 256 and 192 for better performance. 128 bit keys is more than enough security for most data types. AES 256 while more secure against Brute force attacks uses a lot of compute power and may not give the best performance.

## Certificate Generation

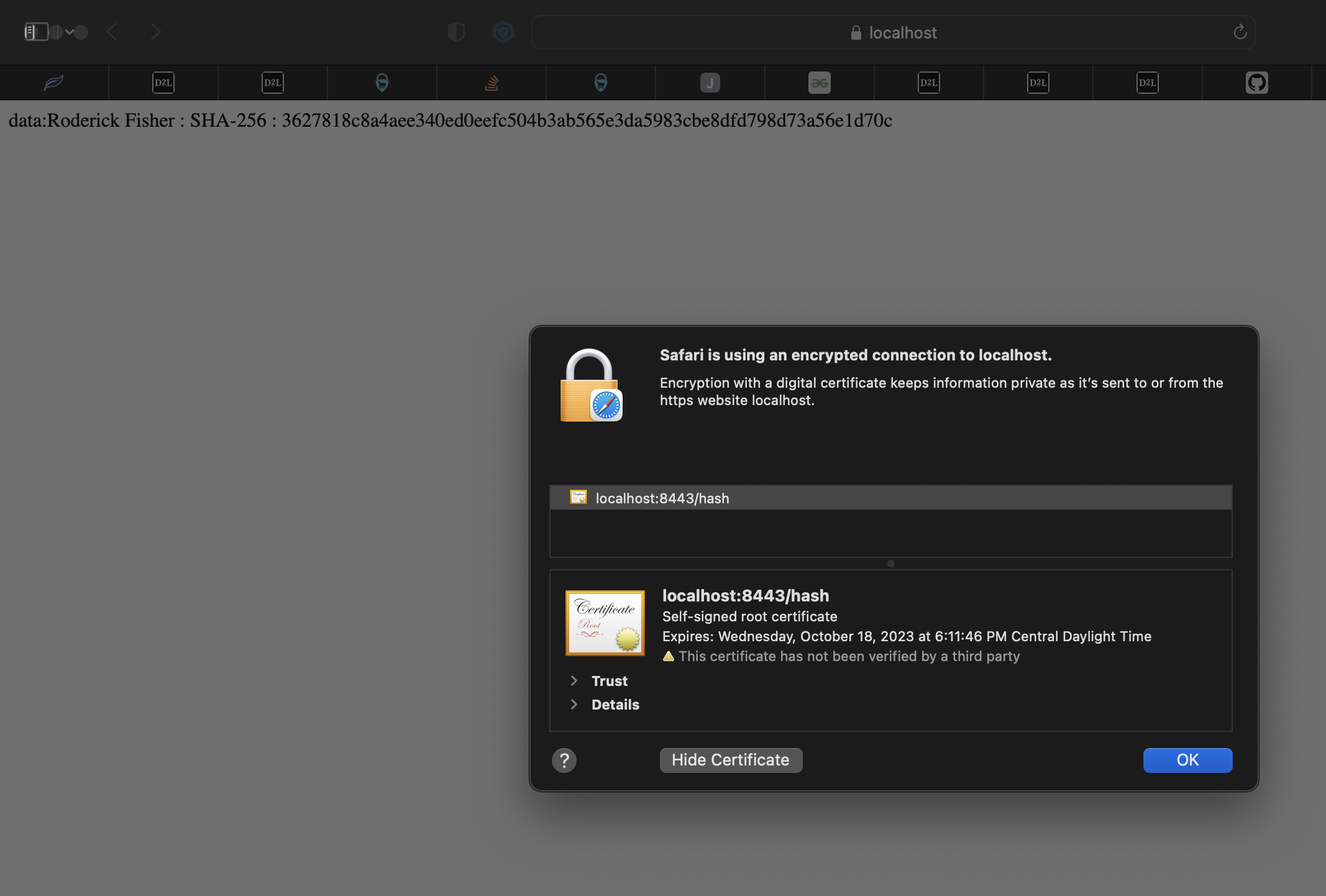
Insert a screenshot below of the CER file.



## Deploy Cipher

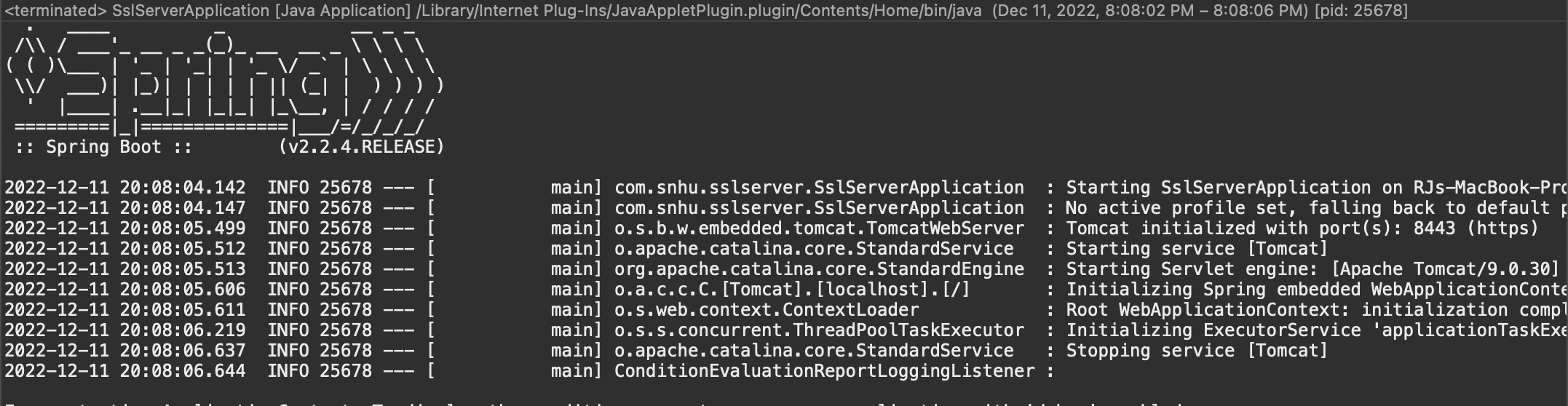
## Secure Communications

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



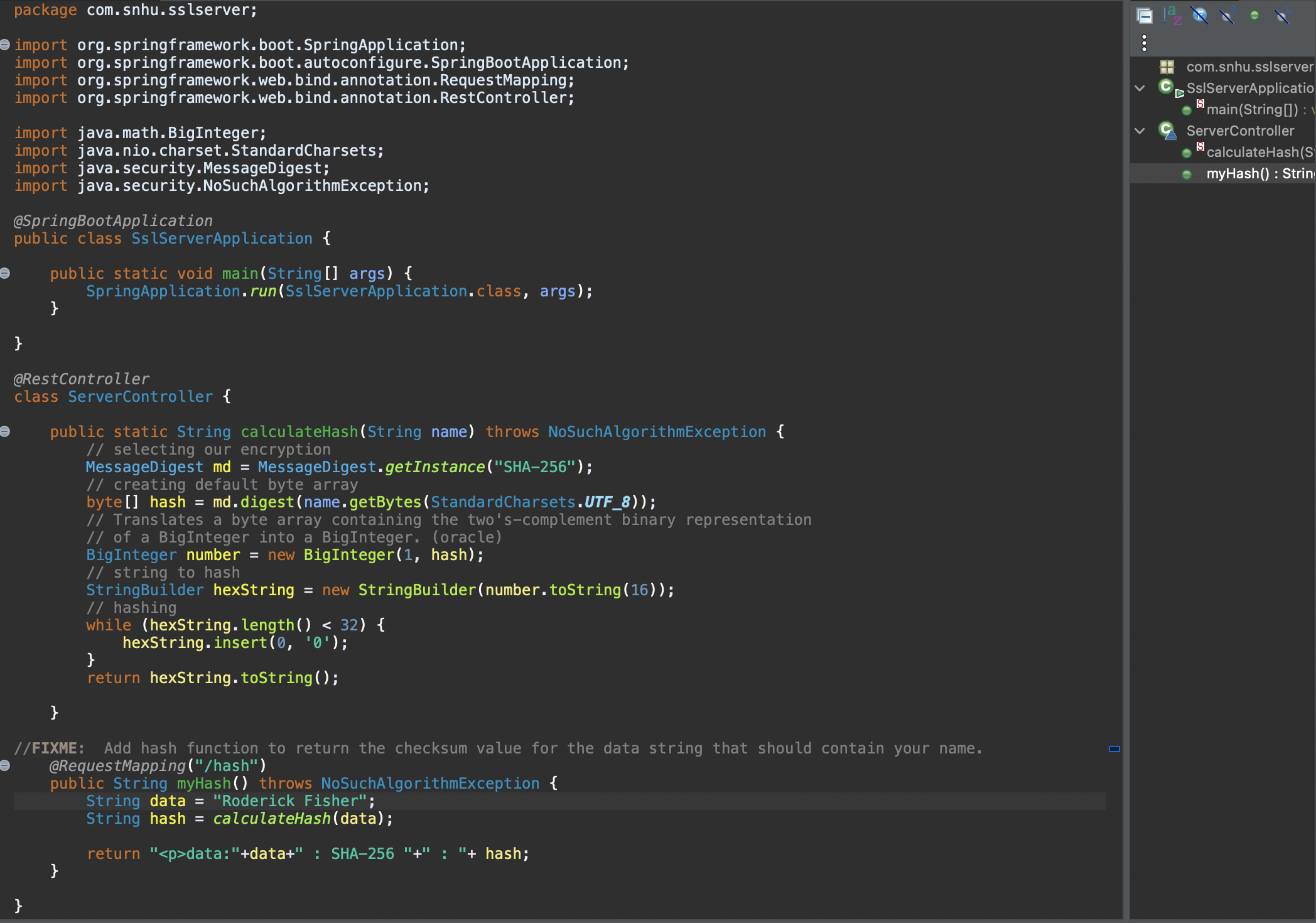
## SImageecondary Testing

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

I integrated OWASP static testing into this project. I have also integrated Apache Commons Codec for SHA 256. I added a secured REST controller to return a hashed value of a specified data. I also ensured that I did not have any errors in the code, and checked to make sure that I did not add any vulnerabilities in static testing.

## Industry Standard Best Practices

I utilized industry standards best practices by integrating static testing and testing my code often. I also made sure to comment out my code to explain to other developers what all blocks of code do. I was able to suppress threats to keep new threats visible. I incorporated SHA-256 encryptions into the code for the best security.