****

# Practices for Secure Software Report

**Table of Contents**

[**Document Revision History 3**](#_30j0zll)

[**Client 3**](#_1fob9te)

[**Instructions 3**](#_3znysh7)

[**Developer 4**](#_2et92p0)

[**1. Algorithm Cipher 4**](#_tyjcwt)

[**2. Certificate Generation 4**](#_3dy6vkm)

[**3. Deploy Cipher 4**](#_1t3h5sf)

[**4. Secure Communications 4**](#_4d34og8)

[**5. Secondary Testing 4**](#_2s8eyo1)

[**6. Functional Testing 4**](#_17dp8vu)

[**7. Summary 4**](#_3rdcrjn)

[**8. Industry Standard Best Practices 4**](#_26in1rg)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **04/15/2023** | **Jackson Foster** |  |

## 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

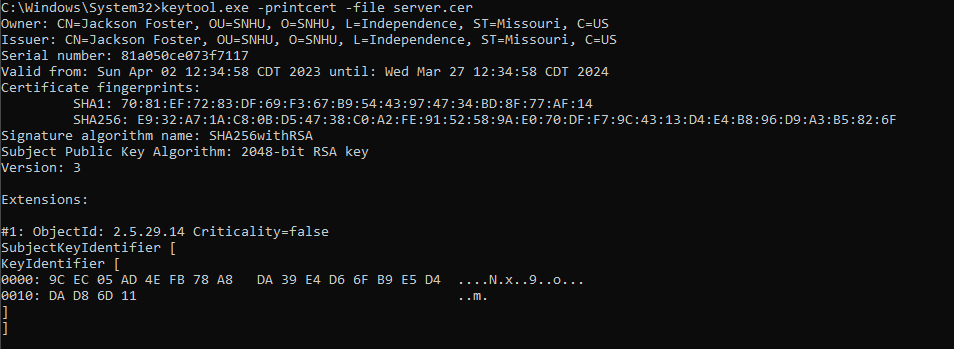
Jackson Foster

## Algorithm Cipher

I used the SHA-256 algorithm which is a cryptographic hash function that is widely used to secure digital data. It takes an input message of any length and produces a fixed-sized output of 256 bits, which is unique to the input message. The algorithm works by applying a series of logical operations to the input message, including bitwise operations, modular arithmetic, and Boolean logic. The resulting hash value is a digital fingerprint of the input message, which can be used to verify its integrity and authenticity. The SHA-256 algorithm is considered to be a secure and robust cryptographic hash function, and it is widely used in various applications, including digital signatures, password storage, and blockchain technology.

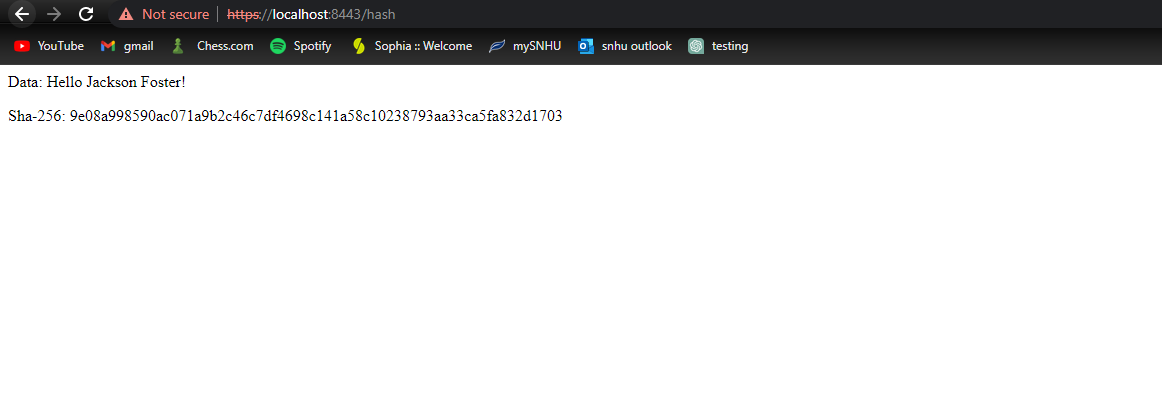
## Certificate Generation

Insert a screenshot below of the CER file.



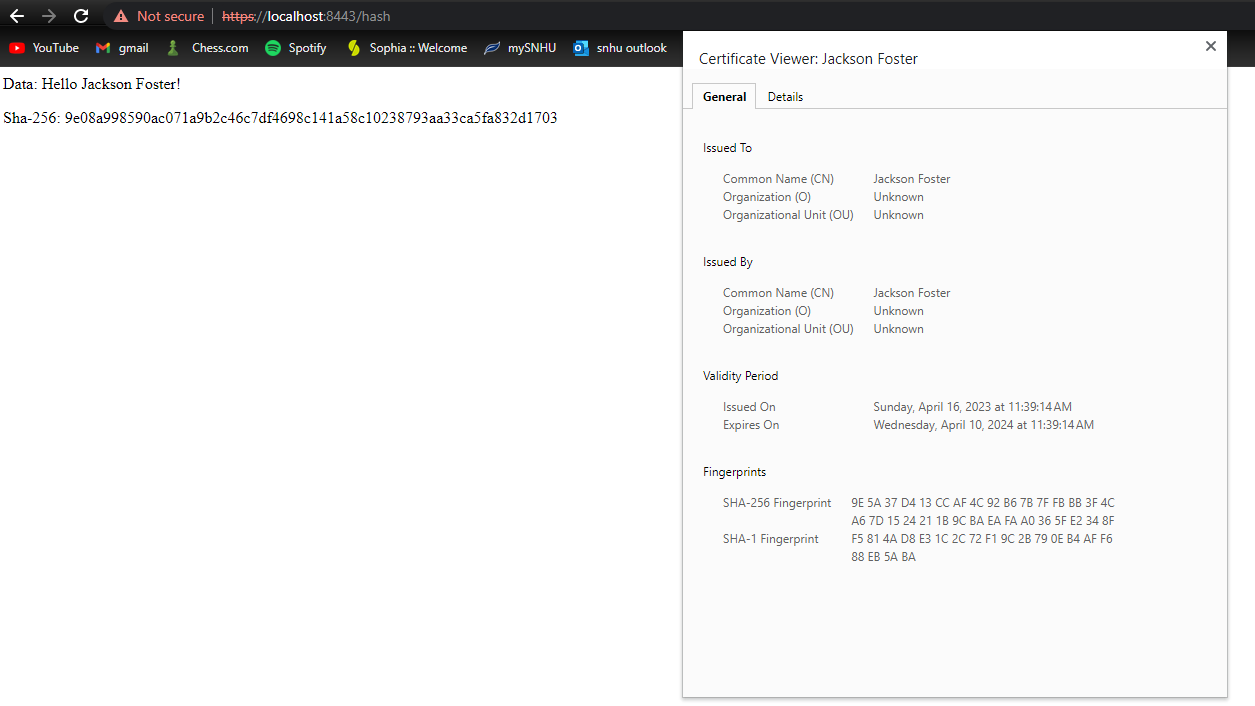
## Deploy Cipher

Insert a screenshot below of the checksum verification.



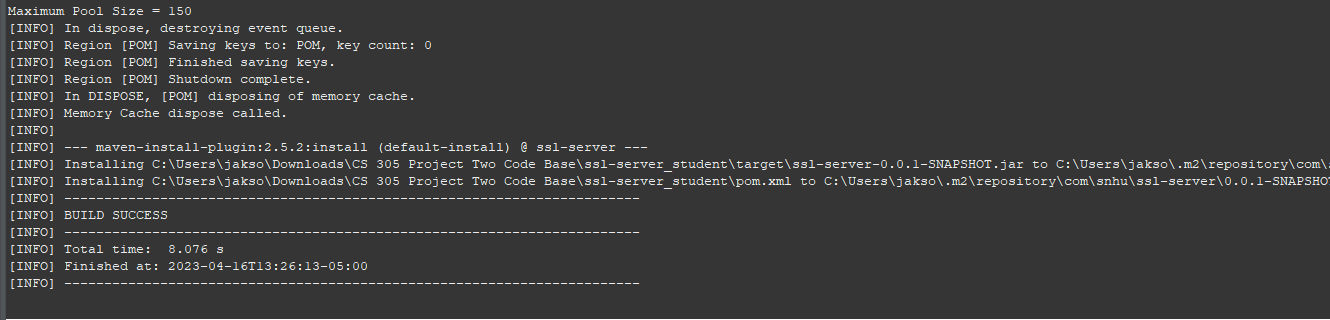
## Secure Communications

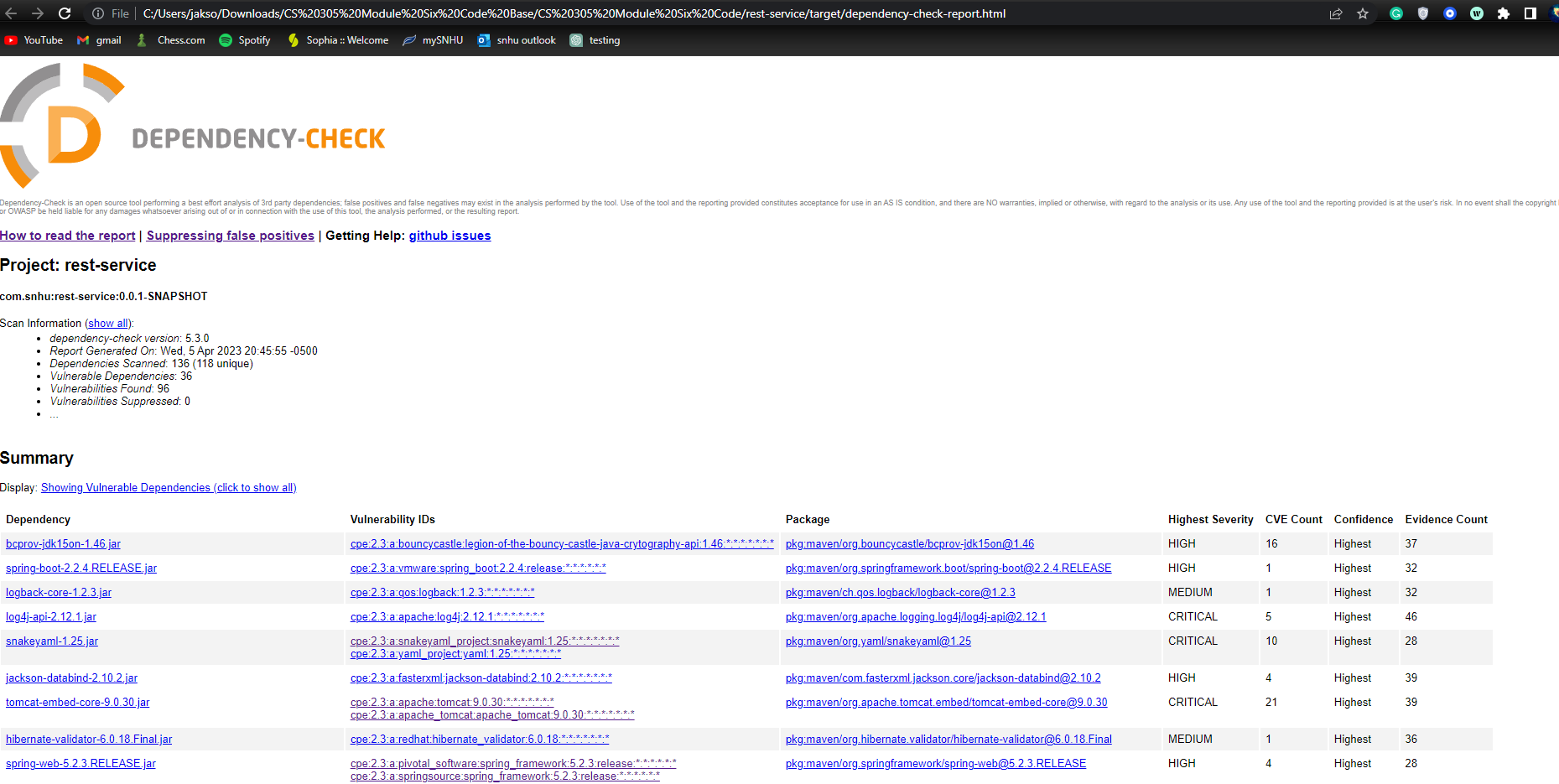
Insert a screenshot below of the web browser that shows a secure webpage.



## Secondary Testing

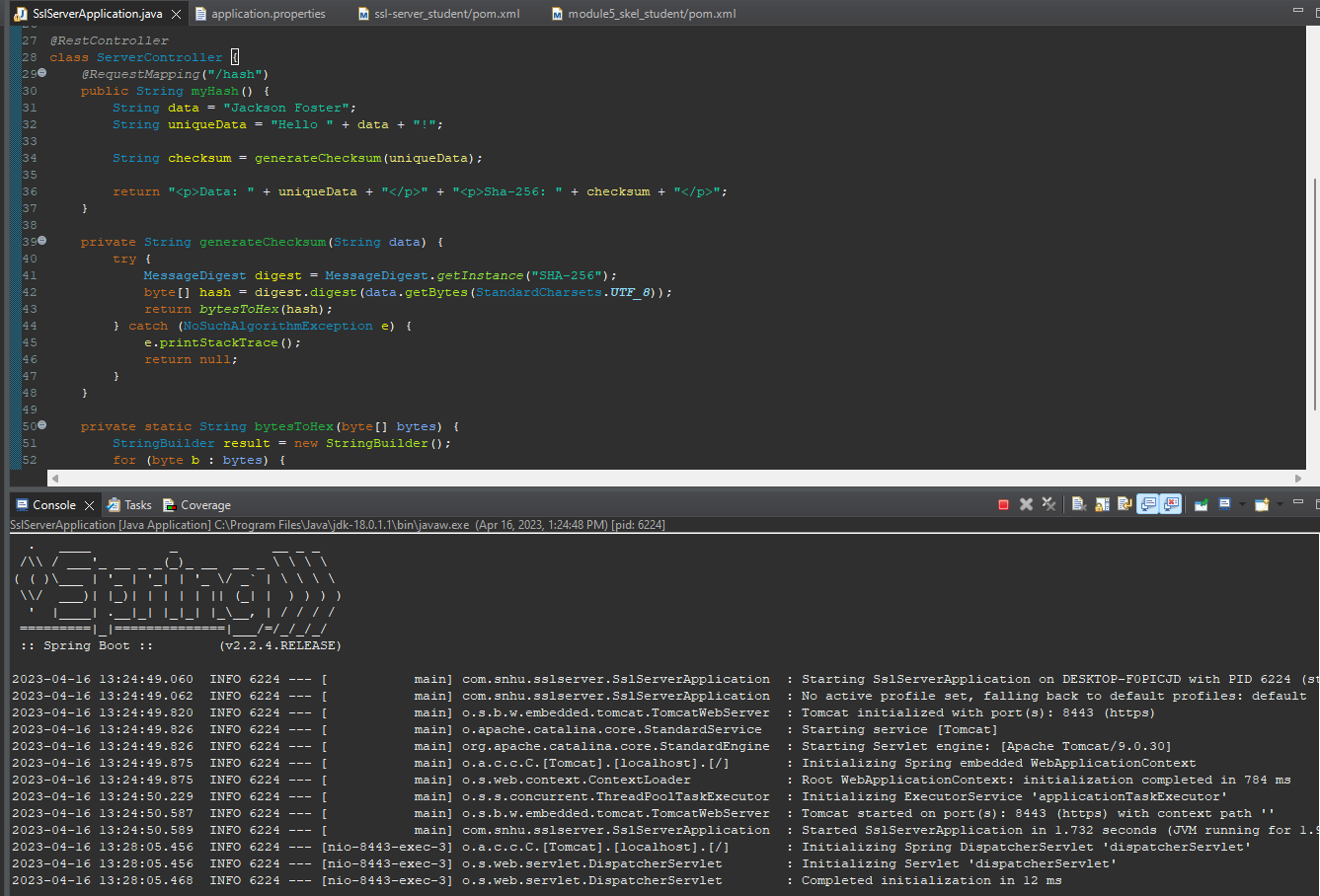
Insert screenshots below of the refactored code executed without errors and the dependency-check report.





## Functional Testing

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



## Summary

To address security concerns in a software application, it is essential to follow a vulnerability assessment process flow diagram that highlights areas of security that need to be addressed. Refactoring the code can help improve the overall security of the application such as in secure communications. Refactoring code can help ensure that secure communication channels, such as SSL/TLS, are used to protect data in transit. This can also help in other aspects from the Vuneranility Assessmnet Process Flow Diagram with Encryption and Decryption, by Refactoring code it can help ensure that data is encrypted and decrypted securely to protect sensitive information from unauthorized access. These steps are important for a secure website to be able to mitigate alot of possible vulnerabilities and secure user data.

## Industry Standard Best Practices

Applying industry standard best practices is critical to maintaining the software application's security. This involves using security frameworks and tools that adhere to industry standards, such as OWASP (Open Web Application Security Project) which was used in this project. To apply industry standard best practices for secure coding, it is essential to regularly update the software application with the latest security patches and fixes. It is also important to implement secure coding practices, such as input validation and output sanitization. For best practices it is a good idea to conduct regular security audits and assessments to identify any vulnerabilities and address them promptly. The value of applying industry standard best practices for secure coding is crucial to the company's overall well-being. It ensures that sensitive data is protected, reducing the risk of data breaches, which can result in significant financial losses and damage to the company's reputation. Additionally, adhering to industry standards ensures that the company is compliant with relevant regulations, reducing legal risks.