****

# 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** | **6/20/2024** | **James Reid** |  |

## 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

James Reid

## Algorithm Cipher

I chose the SHA256 algorithm Cipher because of its high number of possible values while it also has a low collision factor. This is partially due to the 64 rounds that the data is changed. Even reputable businesses use this cipher to protect online data communication. NordVPN (2023)” Many consider SHA-256 to be one of the most secure hashing algorithms today. This is because it’s great at preventing values from being reversed back to the original content. Another problem that it solves well is avoiding hashing collisions.”

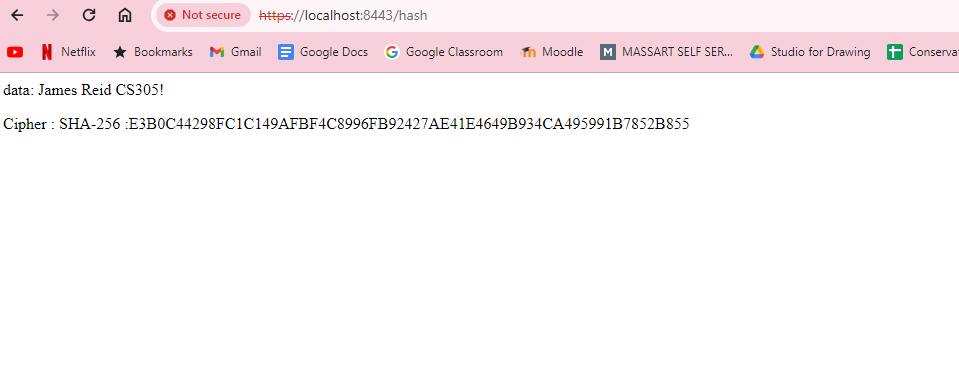
## Certificate Generation

A screenshot of a computer

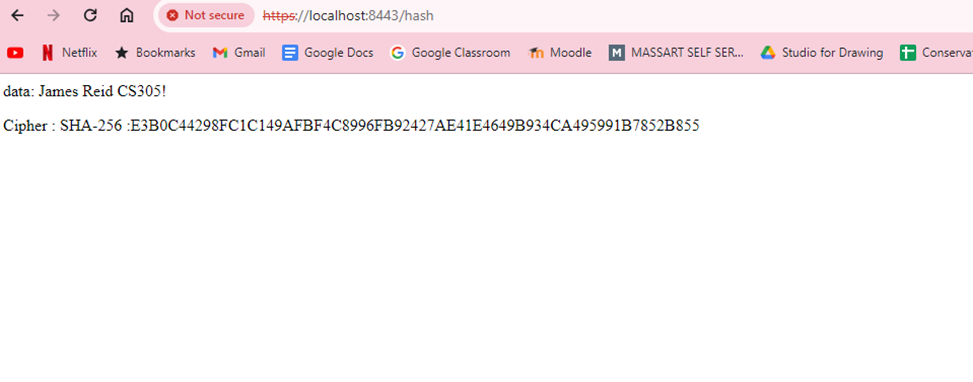
Description automatically generatedInsert a screenshot below of the CER file.

## Deploy Cipher

Insert a screenshot below of the checksum verification.

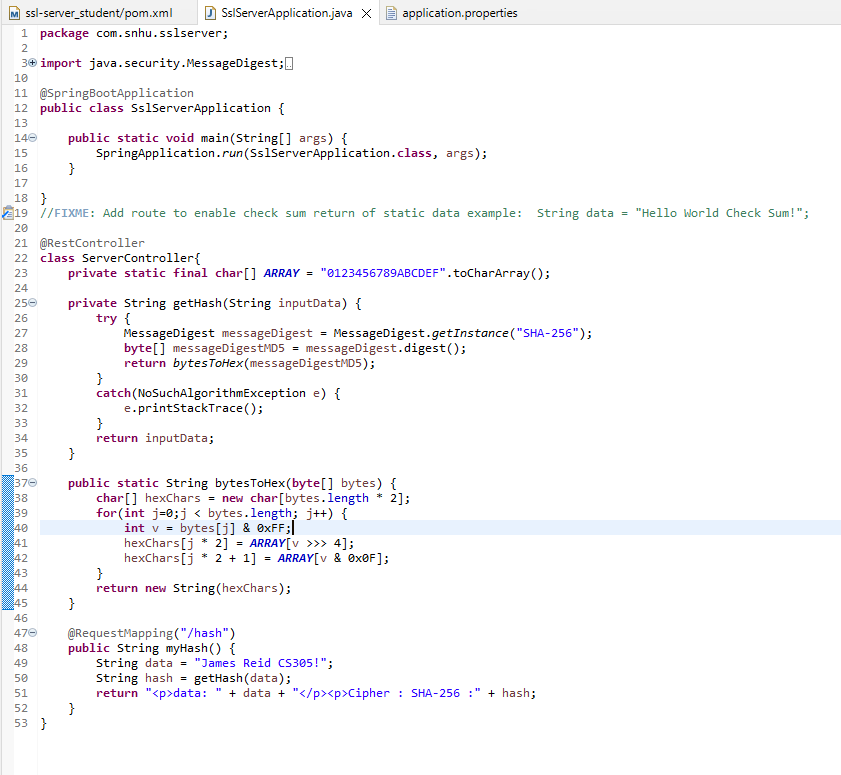
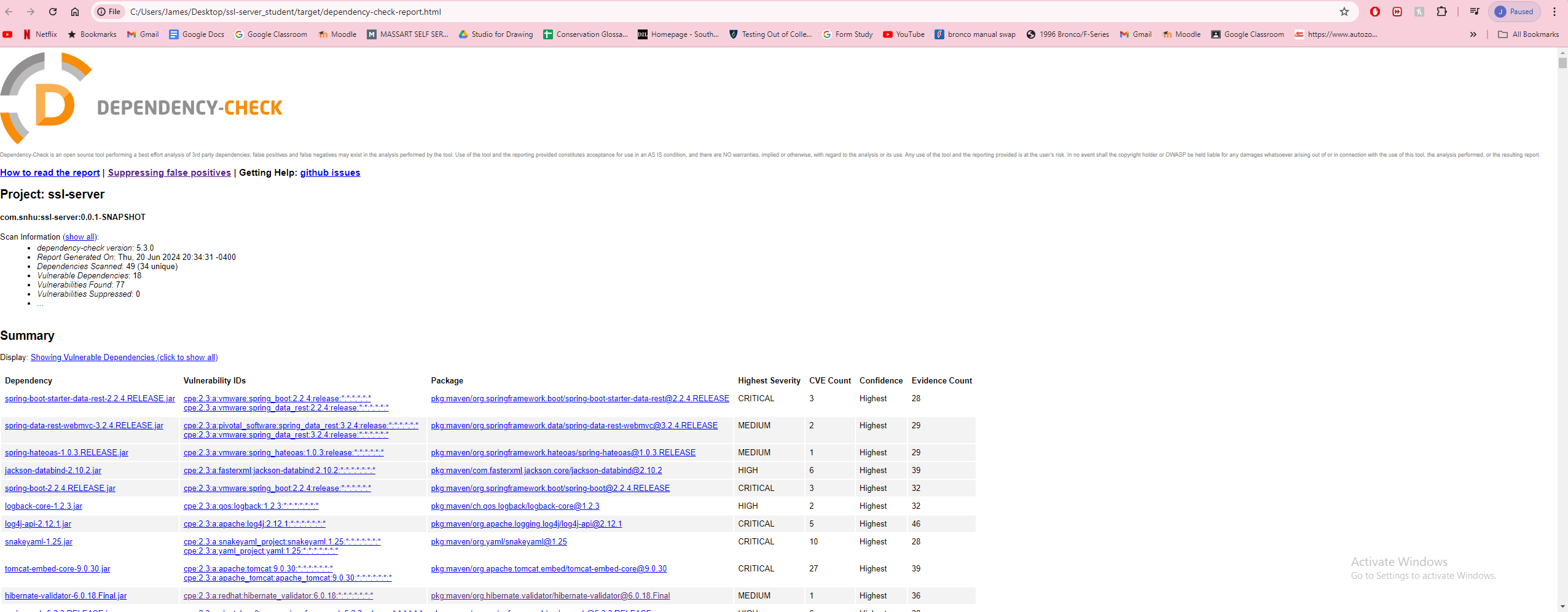


## Secure Communications

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

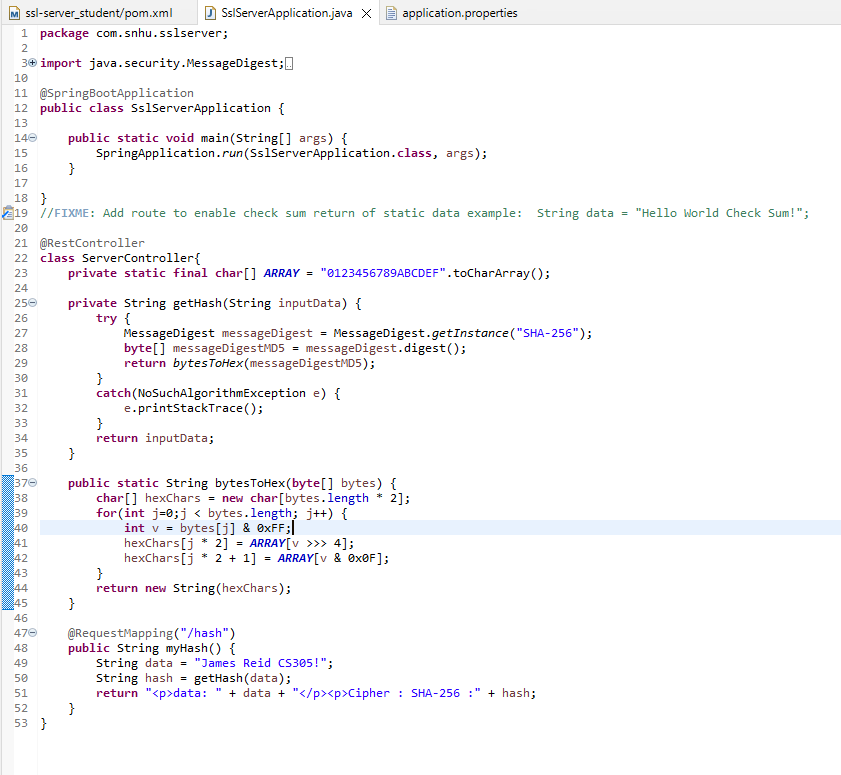
## Secondary Testing

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

I started my refactoring by updating my maven dependency check to the latest version. This reduces security concerns as it keeps everything up to date and accurate. I implemented a REST controller which allows the program a RESTful endpoint. I incorporated the cypher SHA-256 as it has a slim chance of collisions whilst also keeping the data secure.

## Industry Standard Best Practices

To keep best practices and security, the dependency should be checked in regular intervals at least once a month or more. All used plugins and dependencies should also be updated regularly when new versions are available.

References

*What is the SHA-256 algorithm?*. NordVPN. (2023, September 1). https://nordvpn.com/blog/sha-256/