****

# 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** | **08/08/2023** | **Chris Fant** |  |

## 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

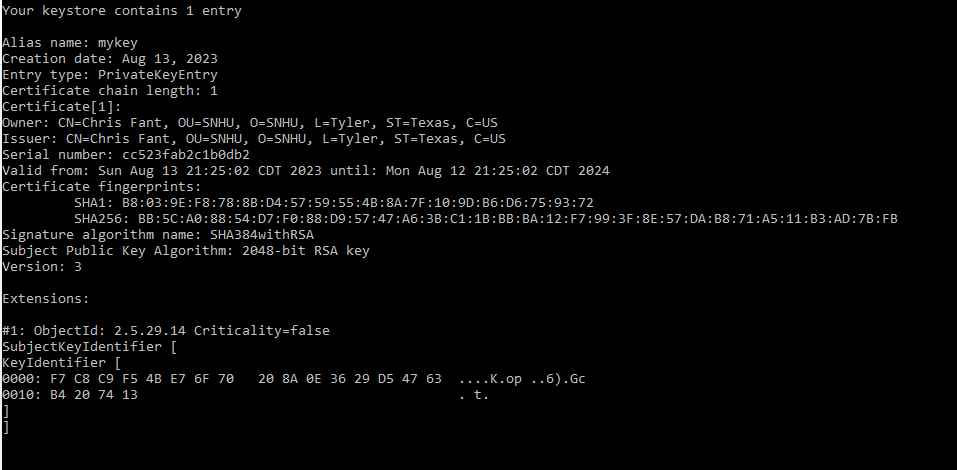
Chris Fant

## Algorithm Cipher

There are a few ciphers out there that get the job done. The ones that have come up the most were MD5, SHA-1, SHA-2/SHA-256, and the newcomer SHA-3. Out of these, I recommend using the SHA-256 variant. The SHA-256 cipher is the best option because it is the most secure. MD5 and SHA-1 are not secure; they have been susceptible to attacks and are no longer recommended as secure ciphers. As far as SHA-3, it is new and has not undergone enough testing to determine whether it is secure or not. On another level, 256 is slower than MD5 and SHA-1 but faster than SHA-3. Therefore, combing the secureness and speed aspects, SHA-256 is the best option.

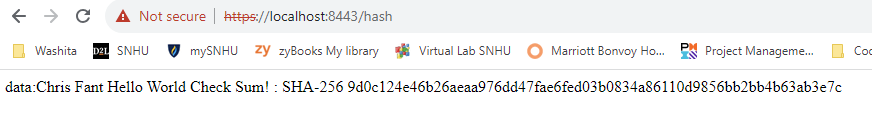
## Certificate Generation

Insert a screenshot below of the CER file.



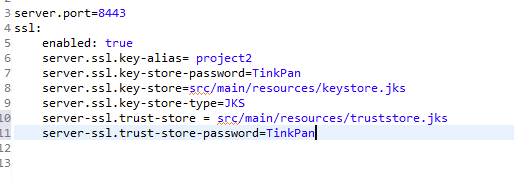
## Deploy Cipher

Insert a screenshot below of the checksum verification.



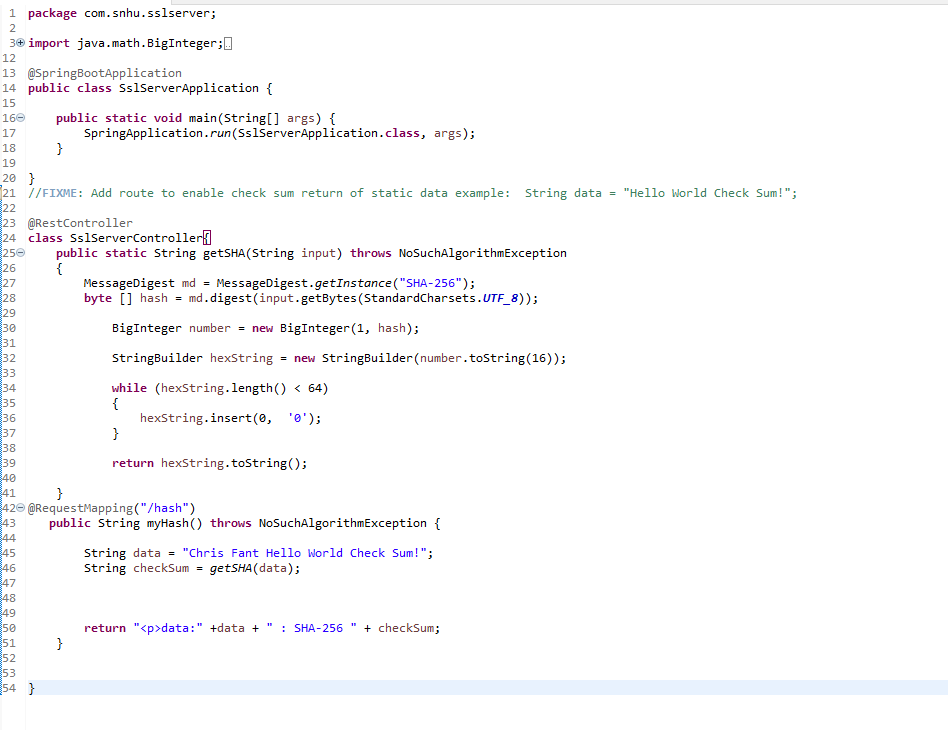
## Secure Communications

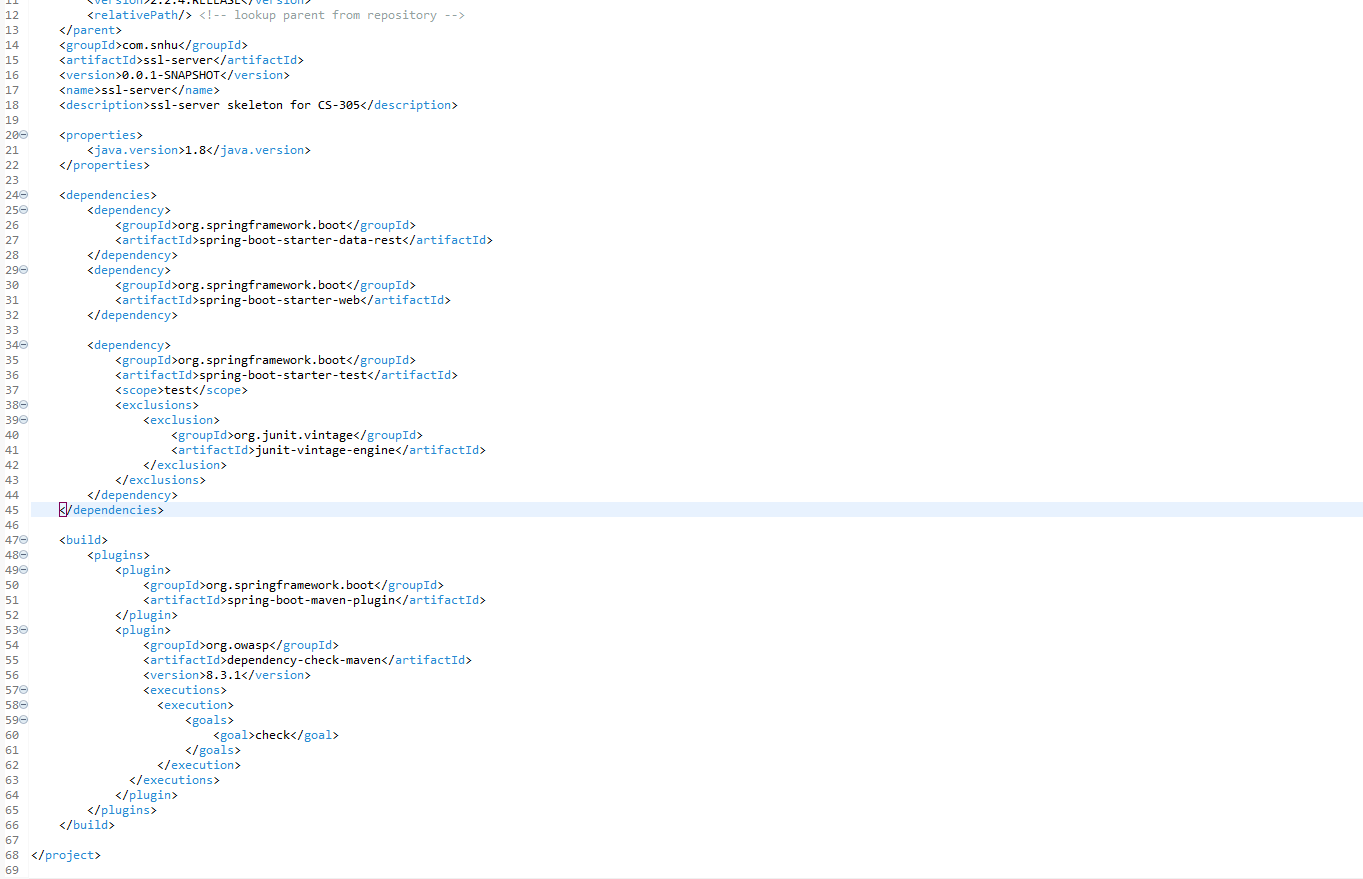
Insert a screenshot below of the web browser that shows a secure webpage.

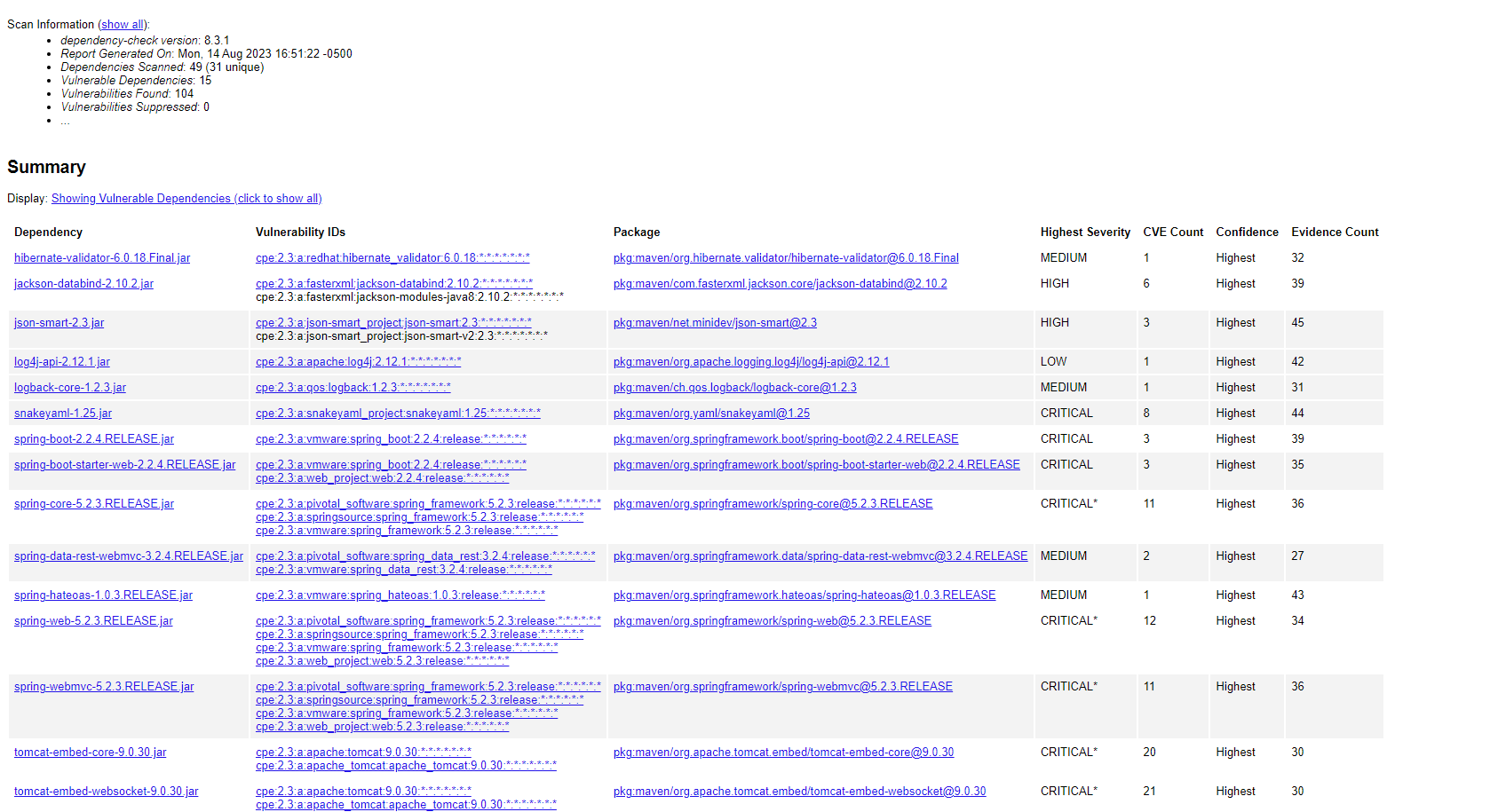


## Secondary Testing

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

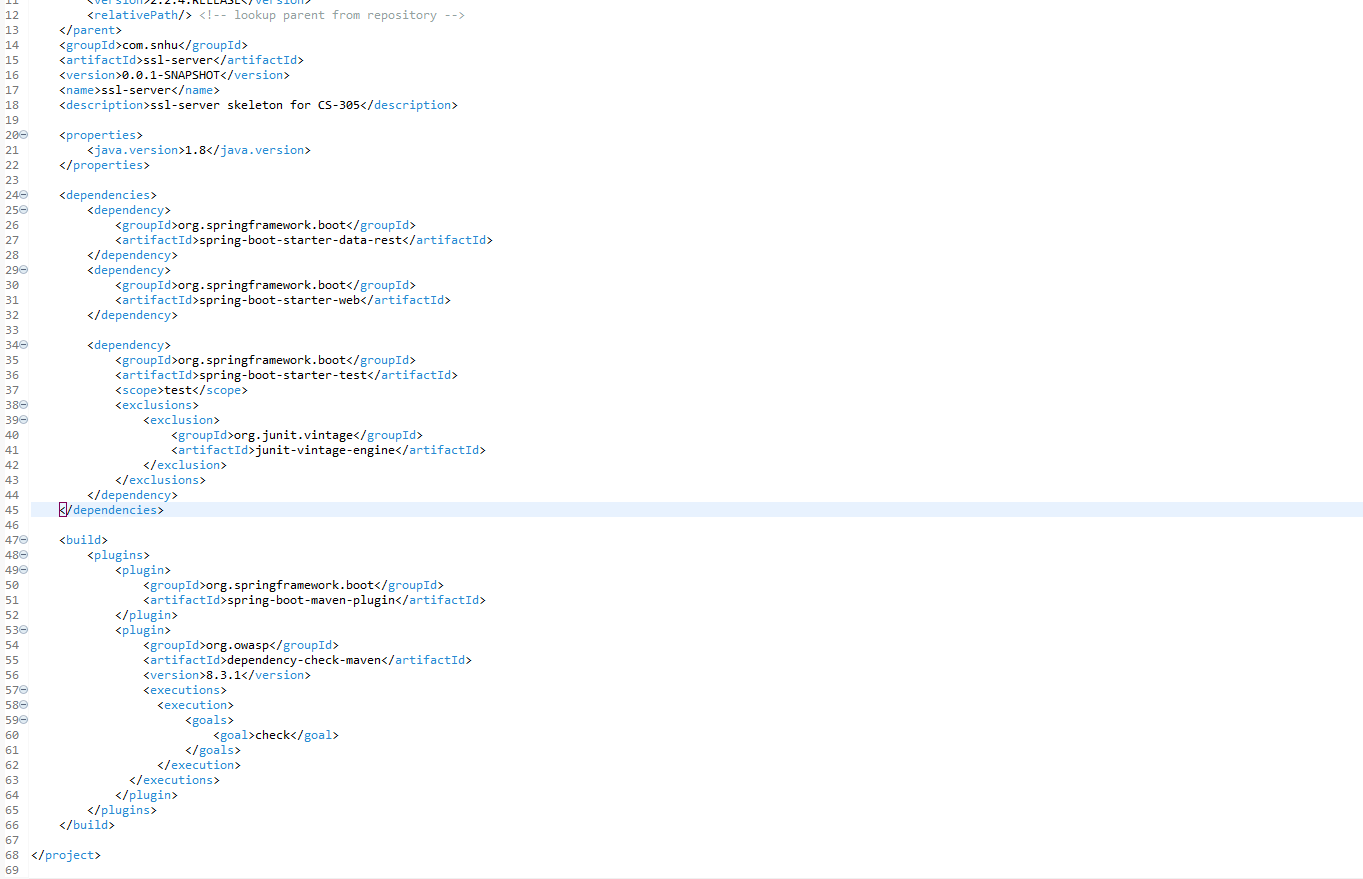


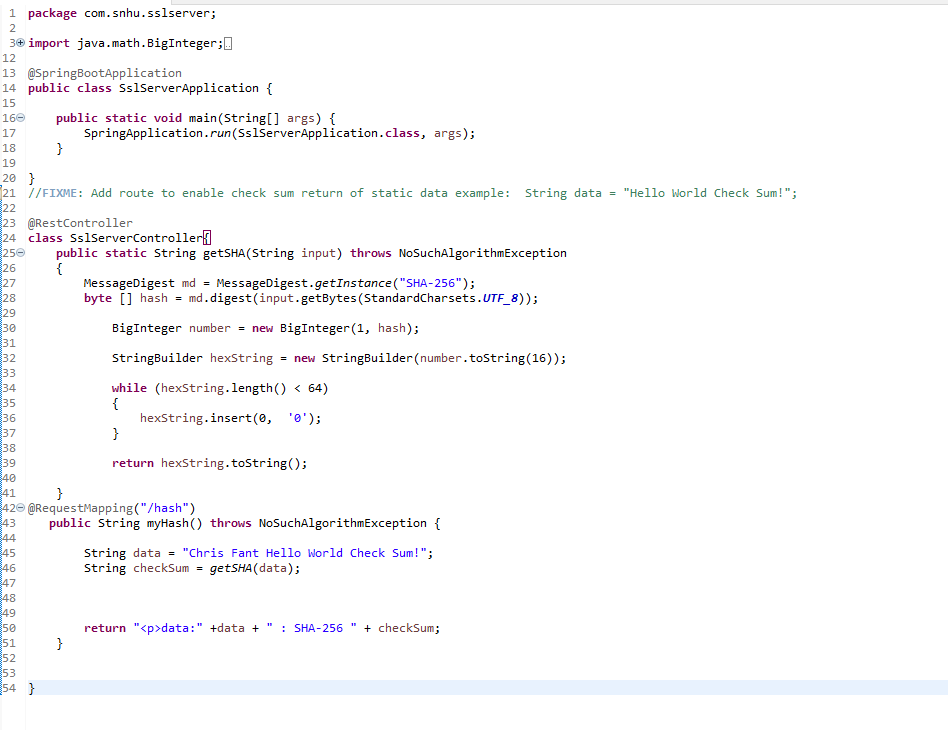


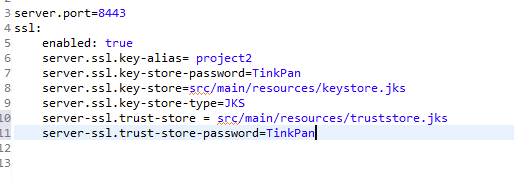


## Functional Testing

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







## Summary

The code had a good foundation but there were a few spots that needed to be shored up. The first thing to look at is the input being received from the user. This needed to be validated to confirm all entries are accepted. This also provides a layer of security to prevent nay attacks to overload the system. Another way is to add a sign-in function to the local host to prevent access from unwanted users. The creates a level of security to prevent anybody from accessing any information be portrayed. Also, adding code to confirm that the site/server has a secure connection. This prevents any possible data loss, or the possibility or information being leaked.

## Industry Standard Best Practices

There are a few things used for the best practices but a couple of them are foundational to this. Simplicity and modularity are two of the most important things to this code. Simplicity helps create a level of understanding and support that allows for the code to be fixed easily. This is good for security because if an issue arises, it will be easy to read the code and understand how to fix it. Modularity is important because it allows for each class or function to be plugged in, in multiple places. For instance, you can use a security function in multiple places to create multiple layers of security. These two thing create the foundation of this code and provides security to the highest level.

Sources

Lowery, J. (2020, March 26). *MD5 vs SHA-1 vs SHA-2 - Which is the Most Secure Encryption Hash and How to Check Them*. FreeCodeCamp.org. https://www.freecodecamp.org/news/md5-vs-sha-1-vs-sha-2-which-is-the-most-secure-encryption-hash-and-how-to-check-them/