****

# 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** | **February 25, 2024** | **July Wellman** |  |

## 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

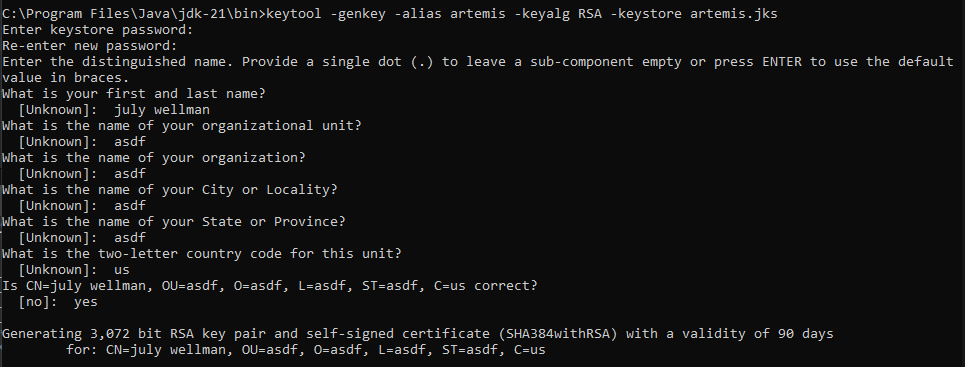
July Wellman

## Algorithm Cipher

A cipher changes inputted plain text into a series of symbols, text, and numbers. This is the process of encryption that makes sensitive information unreadable to the reader unless they have a key to decrypt the output value. This will disallow hackers from being able to decrypt sensitive information even if they can gain access. A message digest is also a term for hash functions, and hash functions are a length of bits representing the original input text. The number of bits in the cipher representing the text or key is shown by its bit levels. Random number usage is used to enforce hash protections. The bigger the encryption, the easier it is to be cracked by a hacker to decrypt it. A symmetric key utilizes only 1 key for both encryption and decryption of the identical information. In an Asymmetric key, two keys are used. The public key is generally made accessible to other individuals while the private key is kept safe and utilized only for decrypting information. Encryption algorithms have existed since Julius Caesar's time when he communicated secretly with his generals via a coded language. Germany also used a complex algorithm cipher during World War II. The use of encryption algorithms has developed as technology has advanced in recent years. Encryption algorithms will be in more demand as we continue to further research into the advancement of IoT.

## Certificate Generation

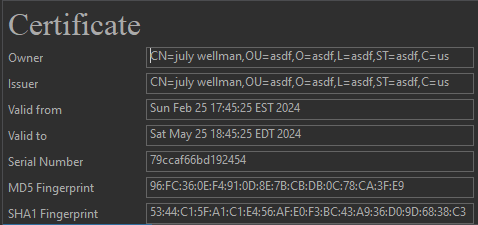
## // Generating the JDK



//Generating the certificate

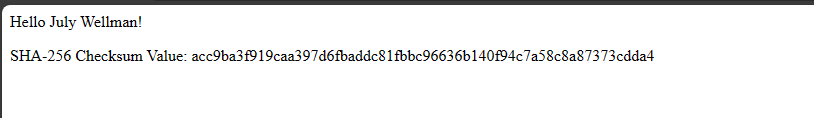


//Eclipse Certificate



## Deploy Cipher

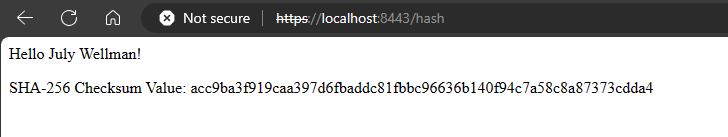
Insert a screenshot below of the checksum verification.



## Secure Communications

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

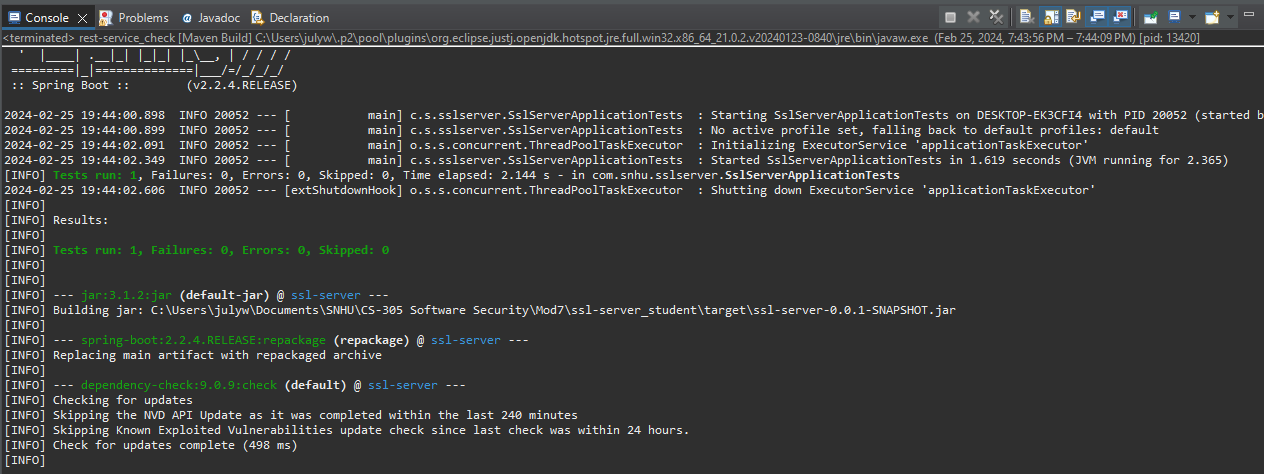
Showing not secure due to me using a self-signed certificate.



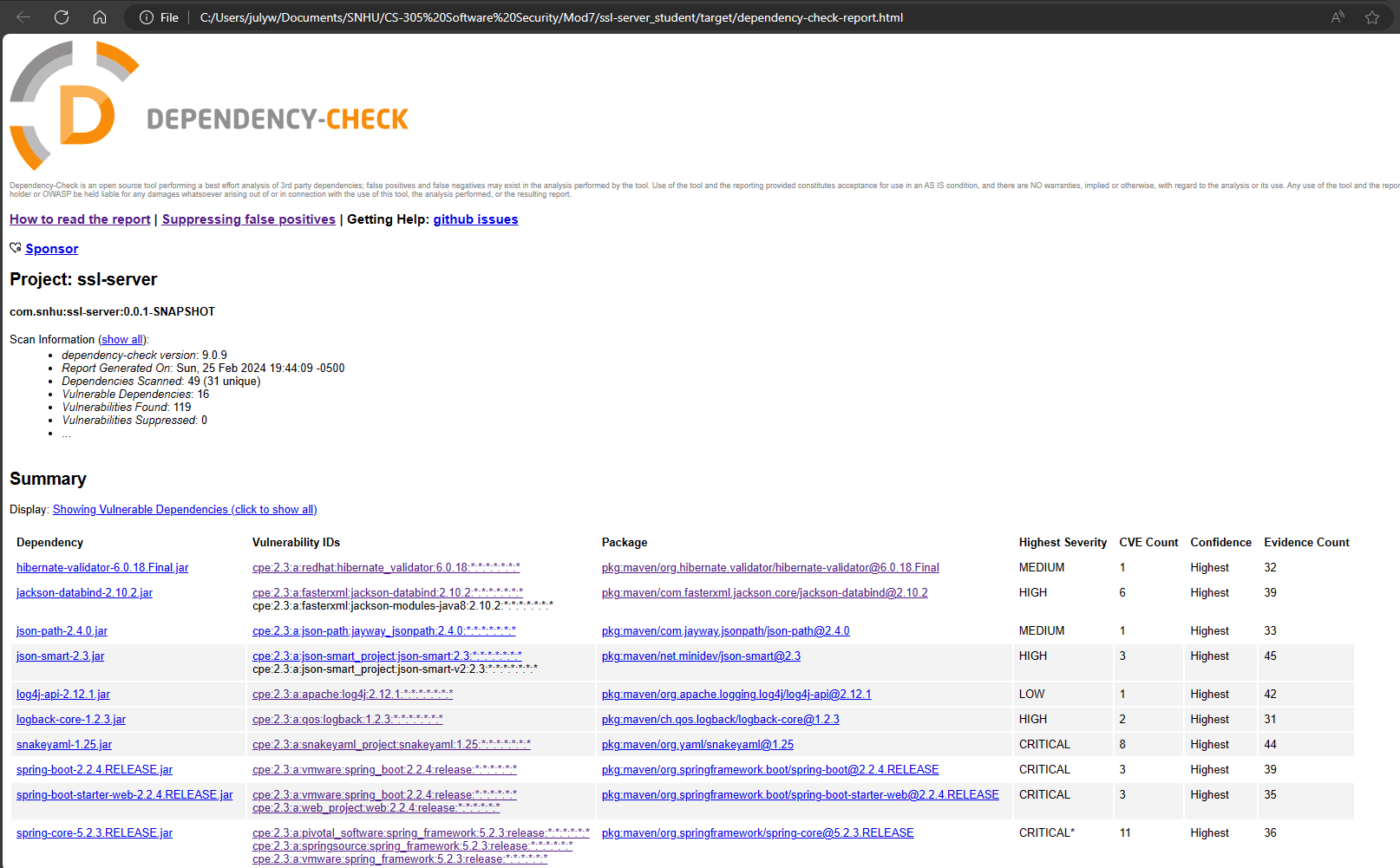
## Secondary Testing

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

// Refactored code executed without errors



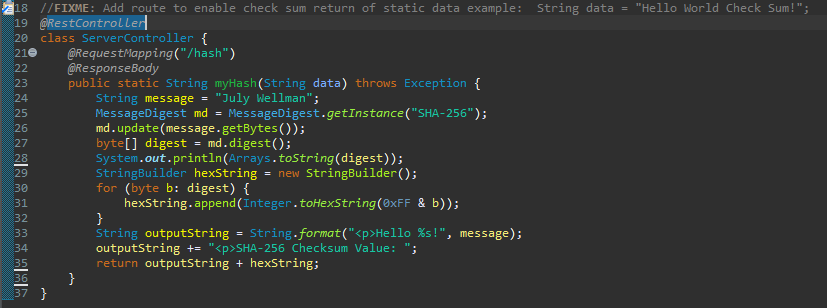
// Dependency Report Check



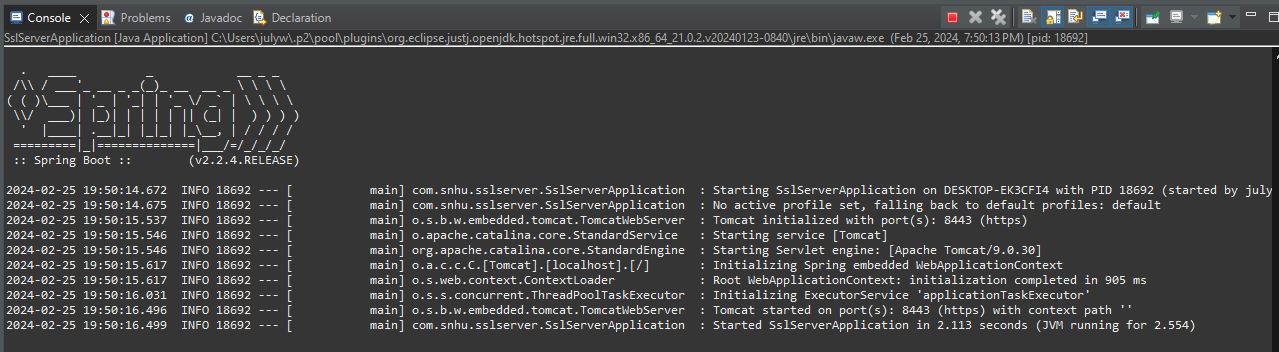
## Functional Testing

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

// Refactored code



// Refactored code without errors



## Summary

The main section of refactored code that I focused on was creating the rest controller. I used a SHA-256 algorithm to output a Hello world text welcome greeting and added another line to display the checksum value. Based on the vulnerability assessment process flow, I addressed the cryptography section by generating a check sum. The only other adjustments that made was updating the version of the dependency report to reflect the most current version of 9.0.9. Other than that, I put most of my time into the certificate generation. I did not realize that you needed to generated the jdk file first in the Java bin folder, and then export a certificate from the jdk file. Once I resolved this, everything else went much smoother.

## Industry Standard Best Practices

I used industry best practices by making sure that versions that needed to be up to date were up to date. Generating the dependency report showed many exposures to the program that could benefit greatly in security by doing simple version updates. I also tested the code often in between refactoring and uploading certificates. This allowed me to minimize the window of the error that I would need to backtrack if I made a mistake.