****

# 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** | **4/14/23** | **Vincent Messina** |  |

## 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.

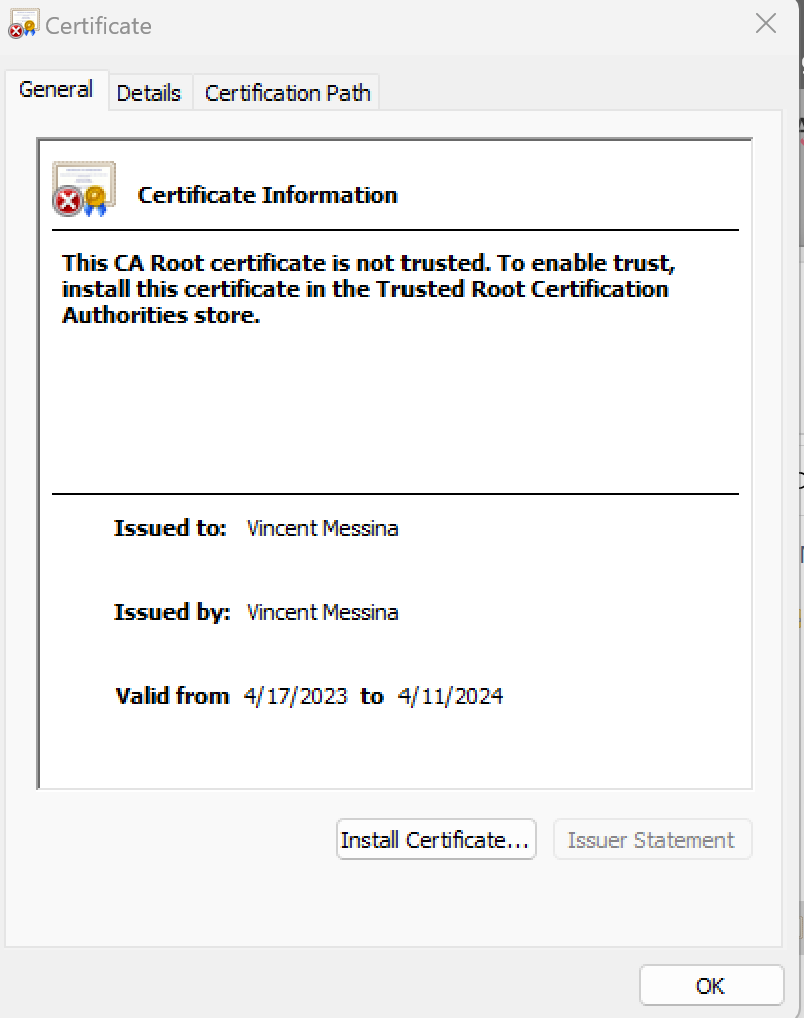
## Developer

Vincent Messina

## Algorithm Cipher

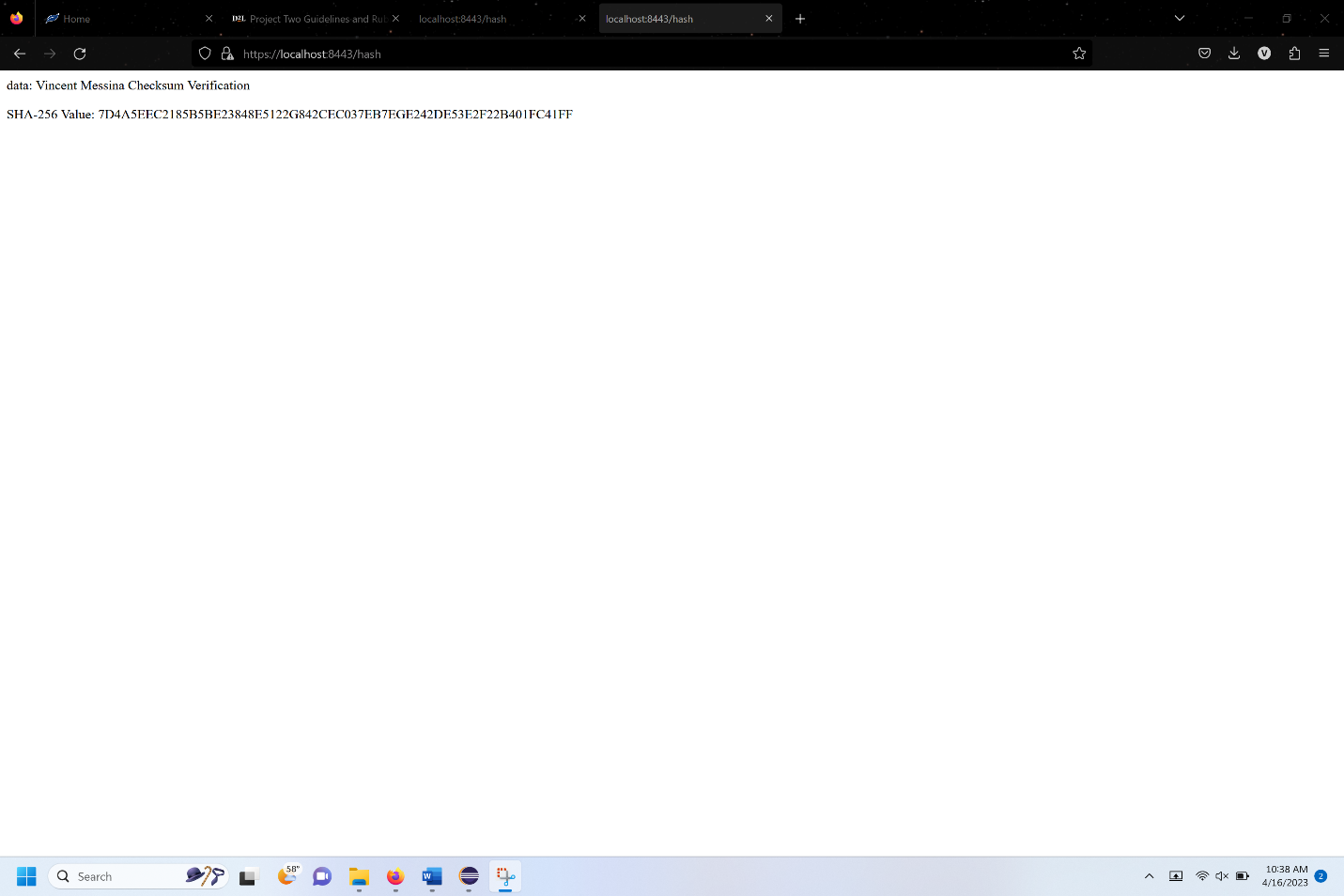
Artemis Financial has a public web interface and want to add a file verification step to their web application to ensure secure communications. I am recommending Artemis Financials use the SHA-256 encryption algorithm cipher. SHA-256 is NIST approved and uses a 256-bit key. SHA-256 uses symmetrical encryption keys. Symmetric keys both encrypt and decrypt information. SHA-256 makes use of random number generation to keep each encrypted file secure.

## Certificate Generation



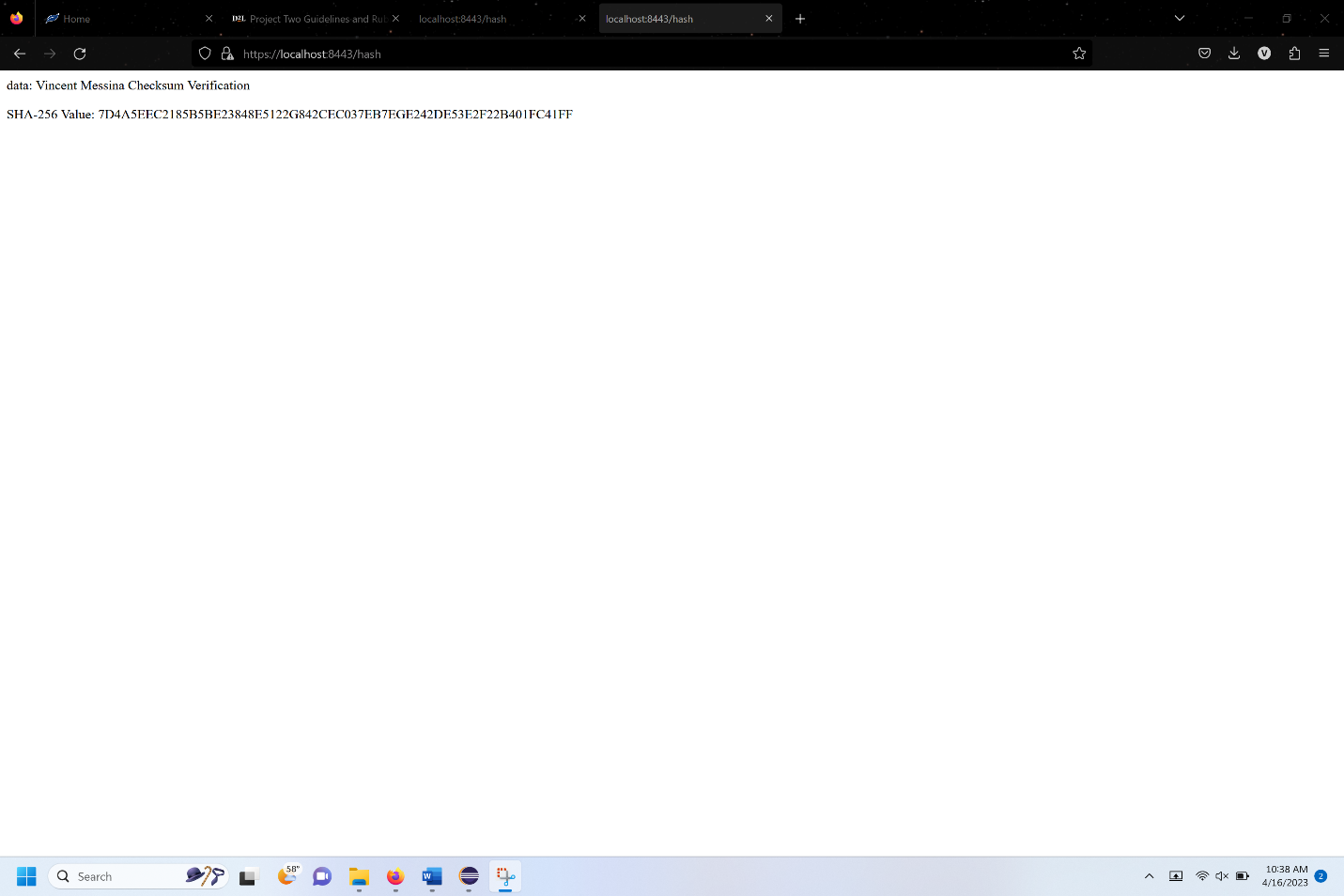
## Deploy Cipher

Insert a screenshot below of the checksum verification.



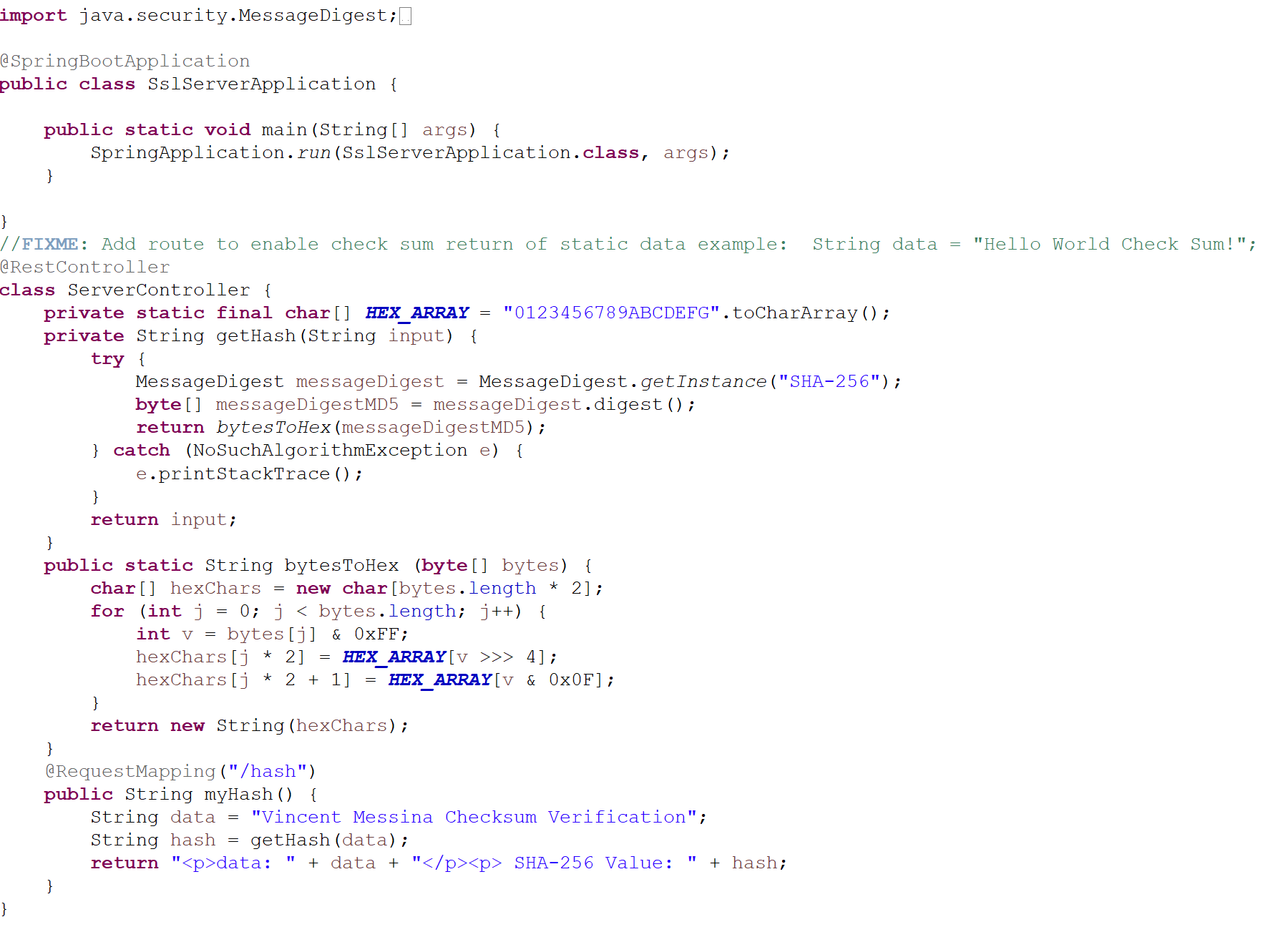
## Secure Communications

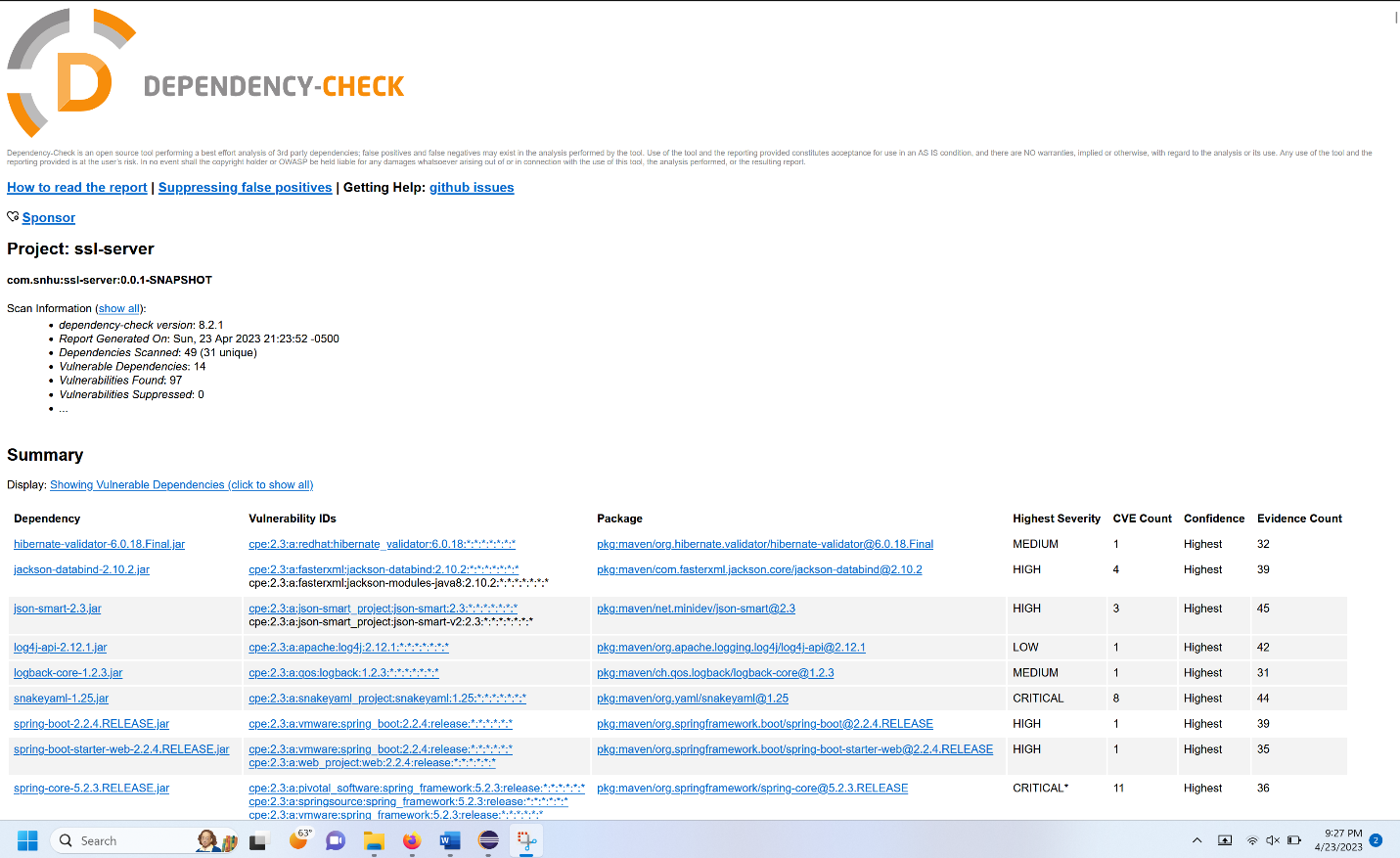
Insert a screenshot below of the web browser that shows a secure webpage.

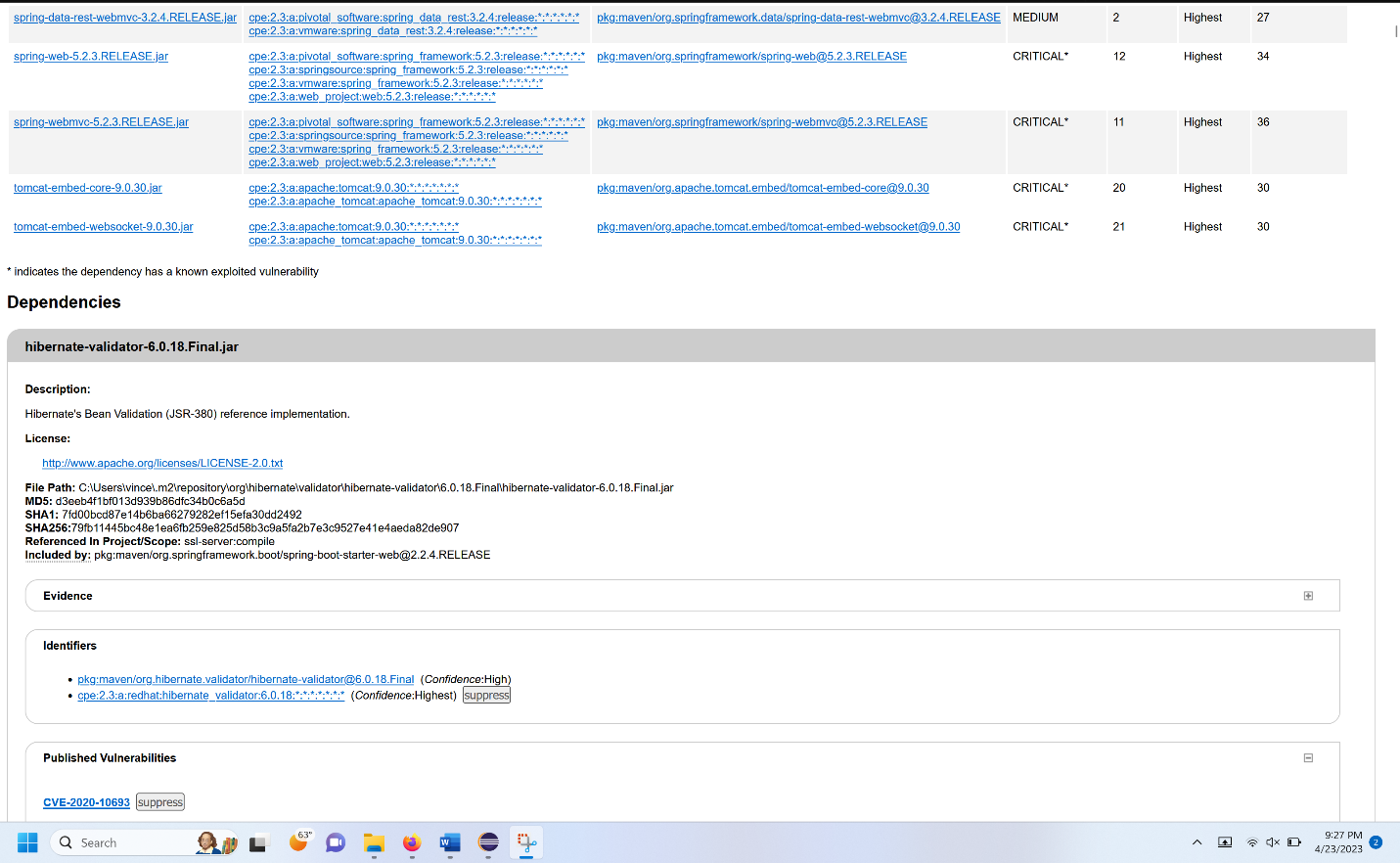


## Secondary Testing

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

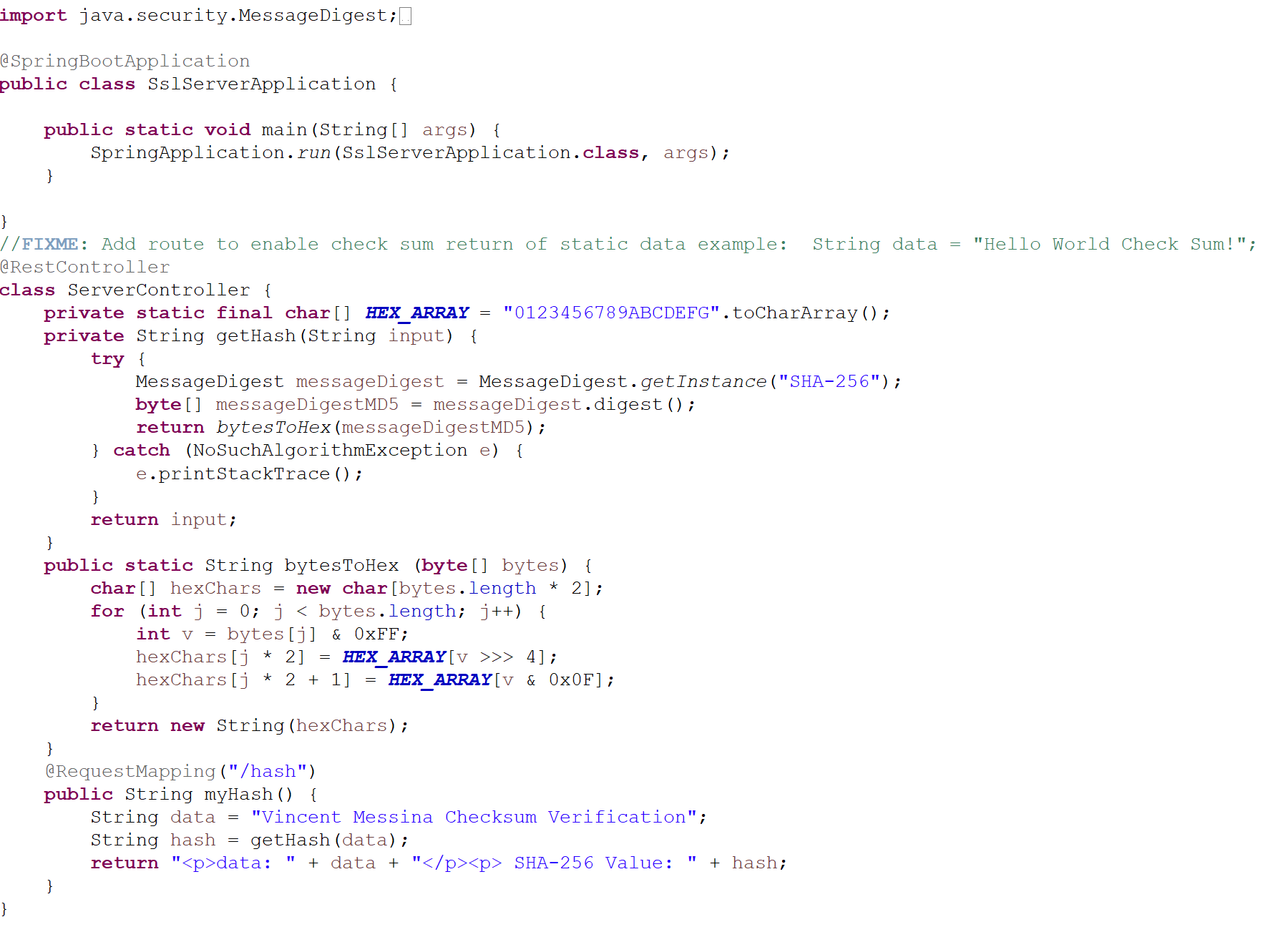






## Functional Testing

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



## Summary

I added a secure RestController to the application to serve as the secure controller for the hash RESTful endpoint. The ServerController class addresses the secure coding concern in the vulnerability assessment diagram. I used SHA-256 as the algorithm cipher. I also updated to the latest version on the dependency check.

## Industry Standard Best Practices

I wrote clean code and used the most recent version of Maven and the dependency check to maintain the software application’s current security. This is important to have little to no data leaks so the company can relax about software security.