****

# 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** | **[Date]** | **Brian Chmura** |  |

## 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.

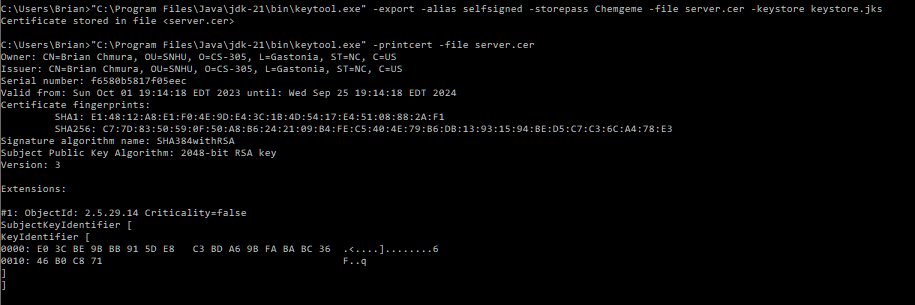
## Developer

Brian Chmura

## Algorithm Cipher

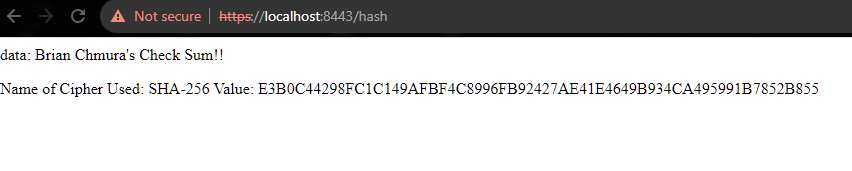
Artemis Financial is looking for an encryption algorithm that will help them encrypt long term data files. It is important for the data to be encrypted so that users that are not supposed to access the files cannot get in and to prevent attacks from people purposely trying to get into the files. I suggest using a SHA-256 (Secure Hash Algorithm 256-bit) as a cryptographic hash function for security purposes with an AES algorithm. This will allow for high security with differently generated keys through Java’s random number generator.

## Certificate Generation

Insert a screenshot below of the CER file.

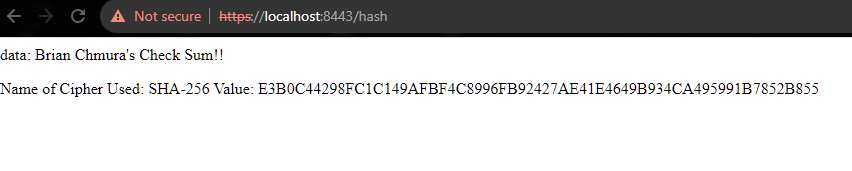
## Deploy Cipher

Insert a screenshot below of the checksum verification.



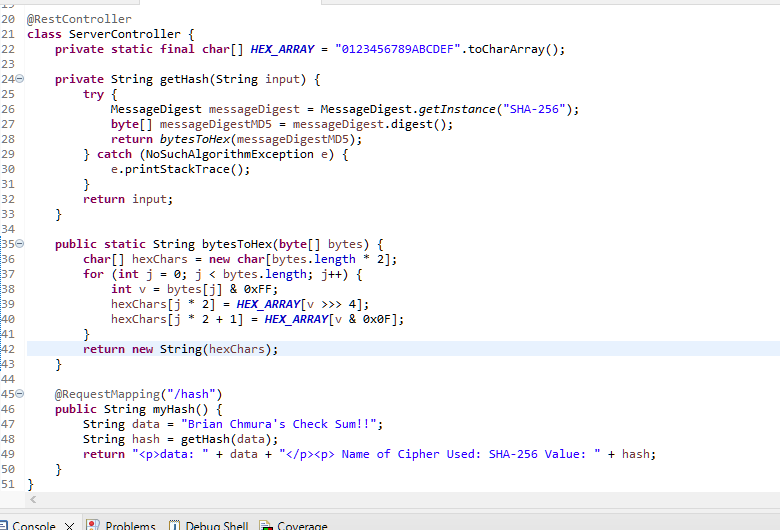
## Secure Communications

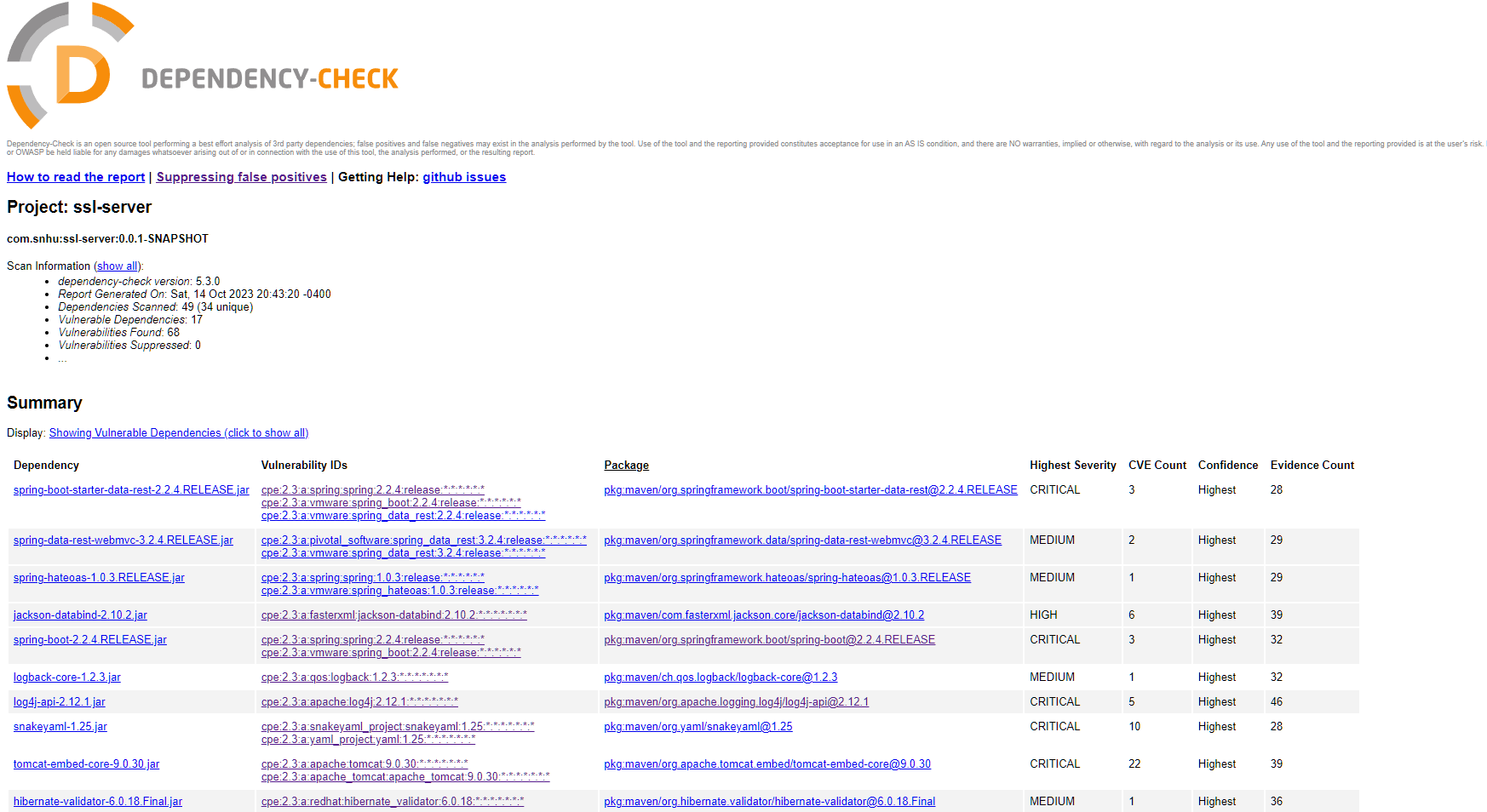
Insert a screenshot below of the web browser that shows a secure webpage.



## Secondary Testing

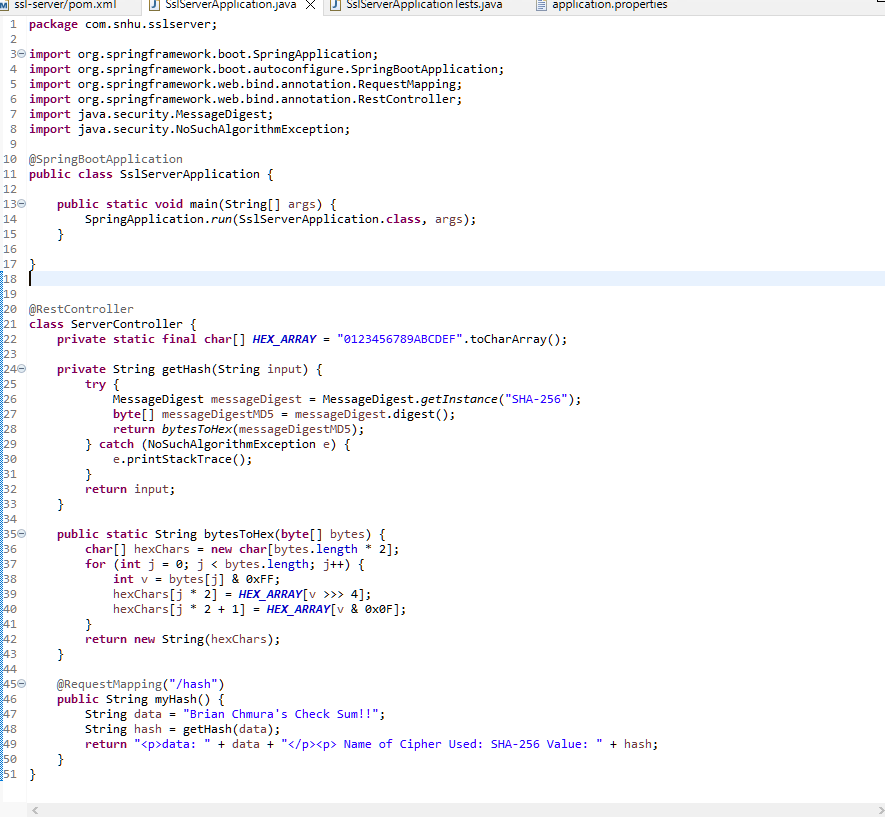
Insert screenshots below of the refactored code executed without errors and the dependency-check report.





## Functional Testing

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



## Summary

In Summary, I refactored the code to secure the RestController and used a SHA-256 hash cipher to protect the data from attacks. In addition, I updated the version of the dependency check to accurately report issues because most issues that arise can be fixed with a simple upgrade or update to the current version. The ServerController class secures the concern with encryption and meets the Vulnerability Assessment Diagram concerns.

## Industry Standard Best Practices

I applied industry-standard best practices for secure coding, including those recommended by organizations like OWASP (Open Web Application Security Project). OWASP recommends regular updates and patching to mitigate known vulnerabilities, training on security programs, allowing access to only what is necessary, and adding security into the development lifecycle from the beginning.