****

# Practices for Secure Software Report

**Table of Contents**

[**Document Revision History 3**](#_heading=h.30j0zll)

[**Client 3**](#_heading=h.1fob9te)

[**Instructions 3**](#_heading=h.3znysh7)

[**Developer 4**](#_heading=h.2et92p0)

[**1. Algorithm Cipher 4**](#_heading=h.tyjcwt)

[**2. Certificate Generation 4**](#_heading=h.3dy6vkm)

[**3. Deploy Cipher 4**](#_heading=h.1t3h5sf)

[**4. Secure Communications 4**](#_heading=h.4d34og8)

[**5. Secondary Testing 4**](#_heading=h.2s8eyo1)

[**6. Functional Testing 4**](#_heading=h.17dp8vu)

[**7. Summary 4**](#_heading=h.3rdcrjn)

[**8. Industry Standard Best Practices 4**](#_heading=h.26in1rg)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **12/6/2023** | **Christopher Fusaro** | **Software Security** |

## 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

Christopher Fusaro

## Algorithm Cipher

Recommend an appropriate encryption algorithm cipher to deploy, given the security vulnerabilities, and justify your reasoning. Review the scenario and the supporting materials to support your recommendation. In your practices for secure software report, be sure to address the following:

Provide a brief, high-level overview of the encryption algorithm cipher.

Discuss the hash functions and bit levels of the cipher.

Explain the use of random numbers, symmetric versus non-symmetric keys, and so on.

Describe the history and current state of encryption algorithms.

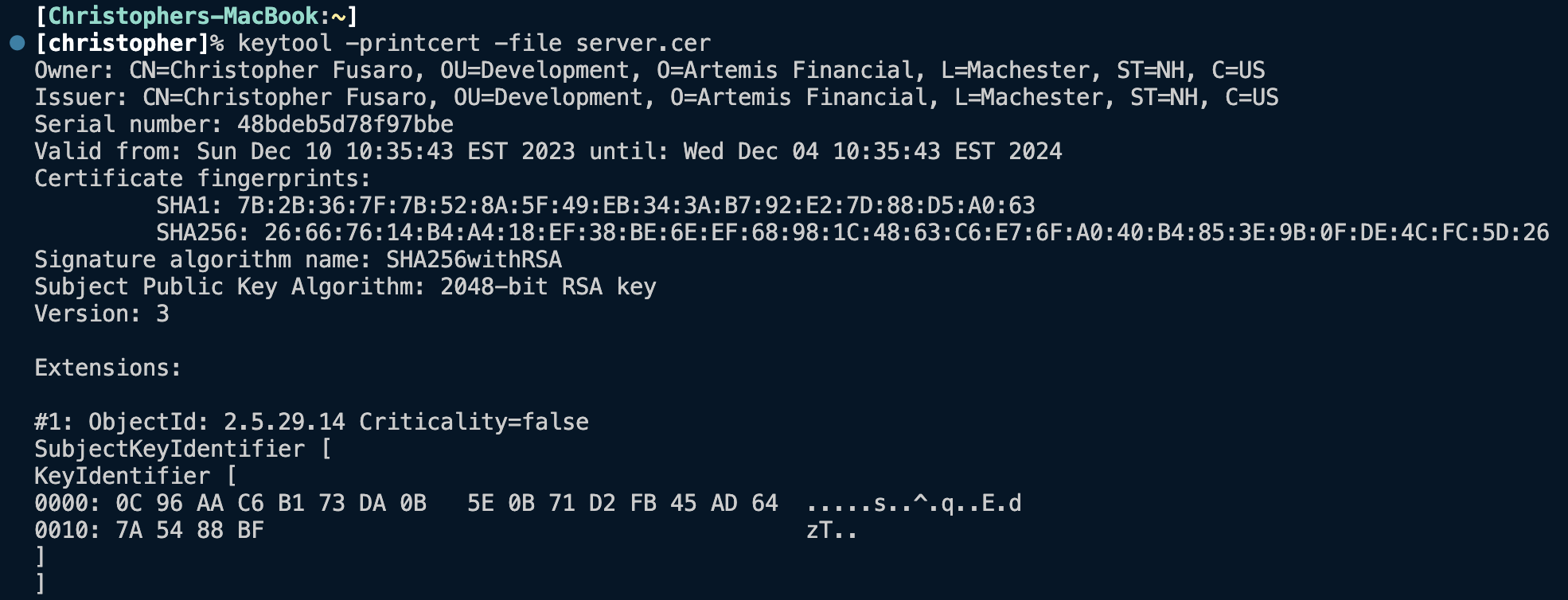
The encryption algorithm cipher I would recommend is the Advanced Encryption Standard (AES).

AES is a symmetric encryption algorithm widely used for securing sensitive data. It operates on fixed-size blocks of data and supports key sizes of 128, 192, and 256 bits. The algorithm consists of a series of transformations. In the case of AES, hash functions are typically separate components. For data integrity verification, a secure hash function like SHA-256 can be used. AES supports key sizes of 128, 192, and 256 bits, allowing for varying levels of security. The bit size corresponds to the length of the key used for encryption, with longer keys providing stronger security.

Random number generators are used to provide unique identifiers for transactions, enhancing security by adding unpredictability. This is especially useful for tracking events such as data transfers or communications. The scenario mentions the use of symmetric keys for shared communication between the server and the client. Non-symmetric keys, involving public and private keys, are part of the system for secure key exchange and potentially digital signatures.

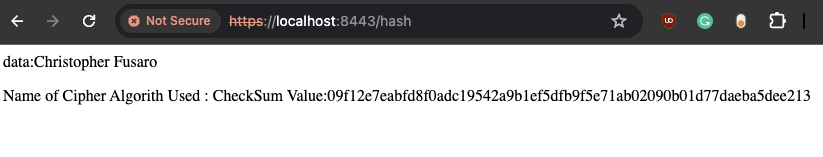
The U.S. National Institute of Standards and Technology (NIST) established AES as the standard encryption algorithm in 2001, replacing the aging Data Encryption Standard (DES). The selection process involved global scrutiny and rigorous testing, ensuring its robustness against various attacks. AES remains one of the most widely used and trusted encryption standards. Its widespread adoption, support for various key sizes, and resistance to known cryptographic attacks contribute to its continued relevance.

## Certificate Generation

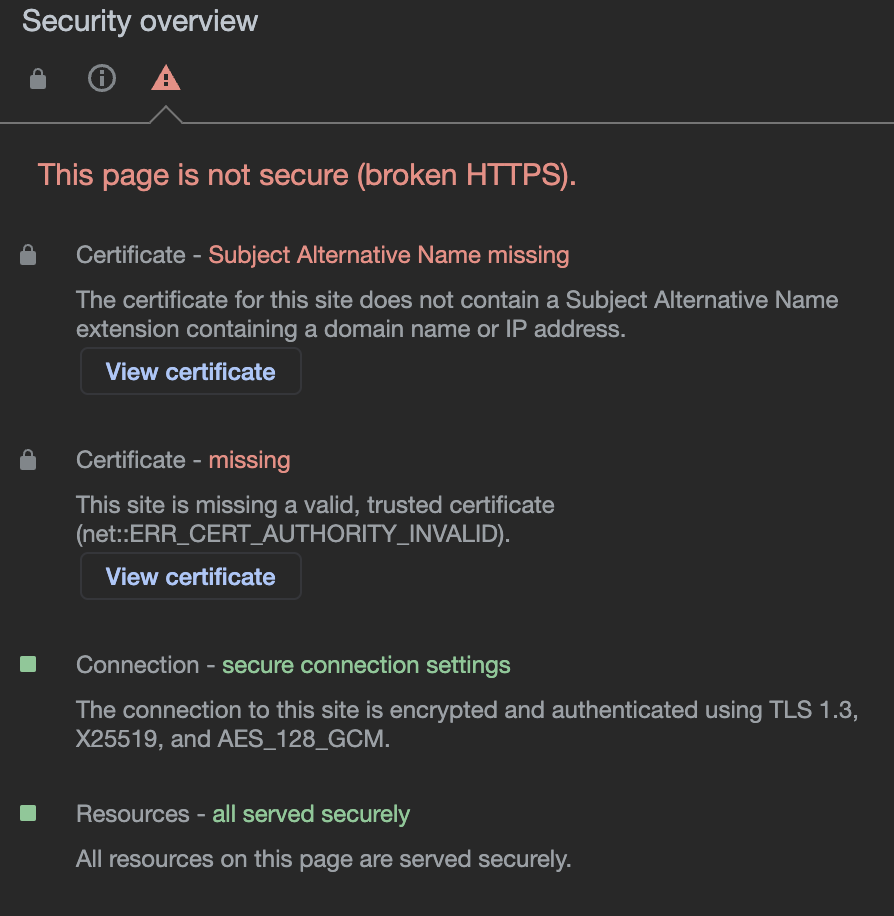


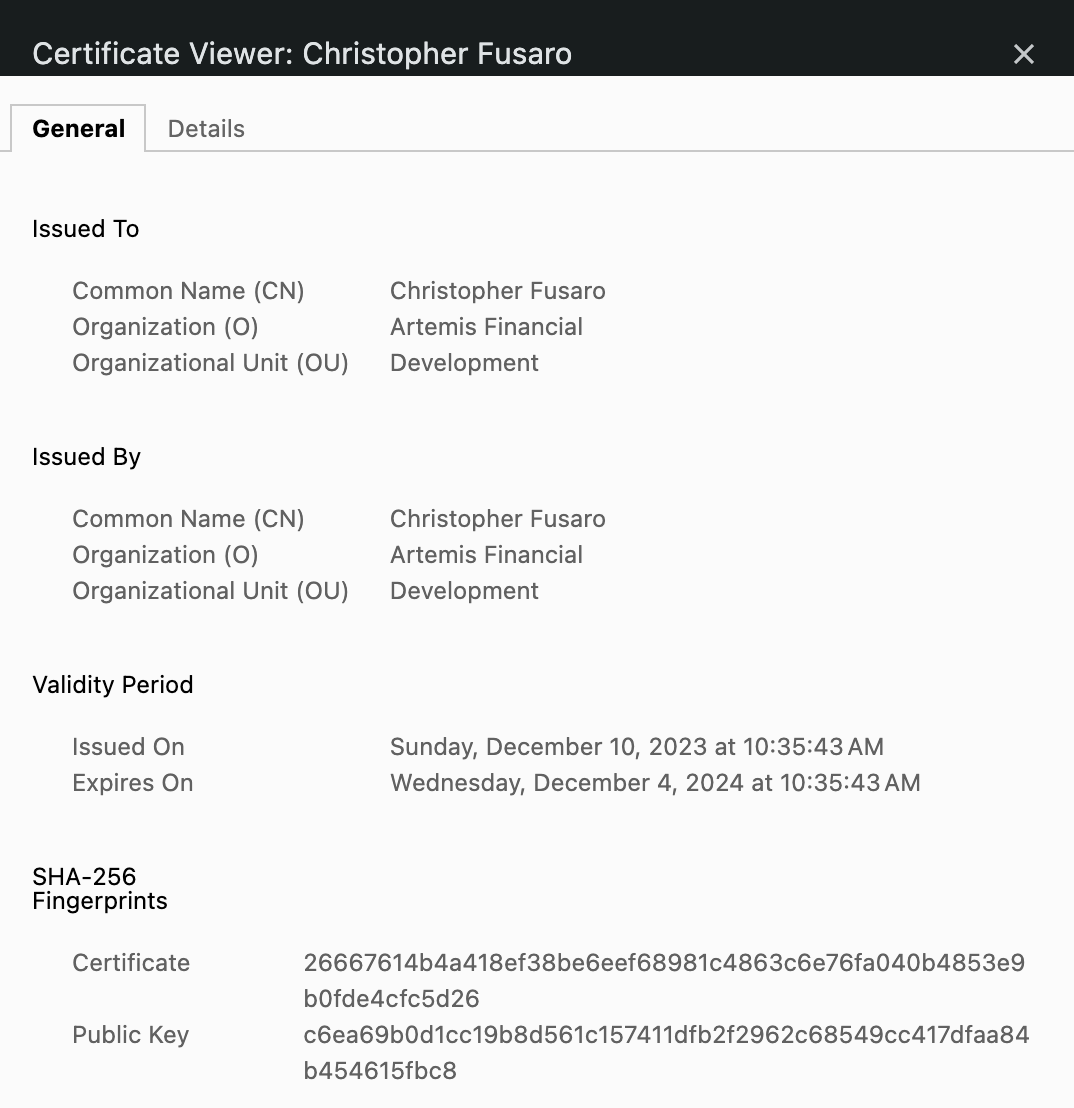
## Deploy Cipher

Insert a screenshot below of the checksum verification.



## Secure Communications

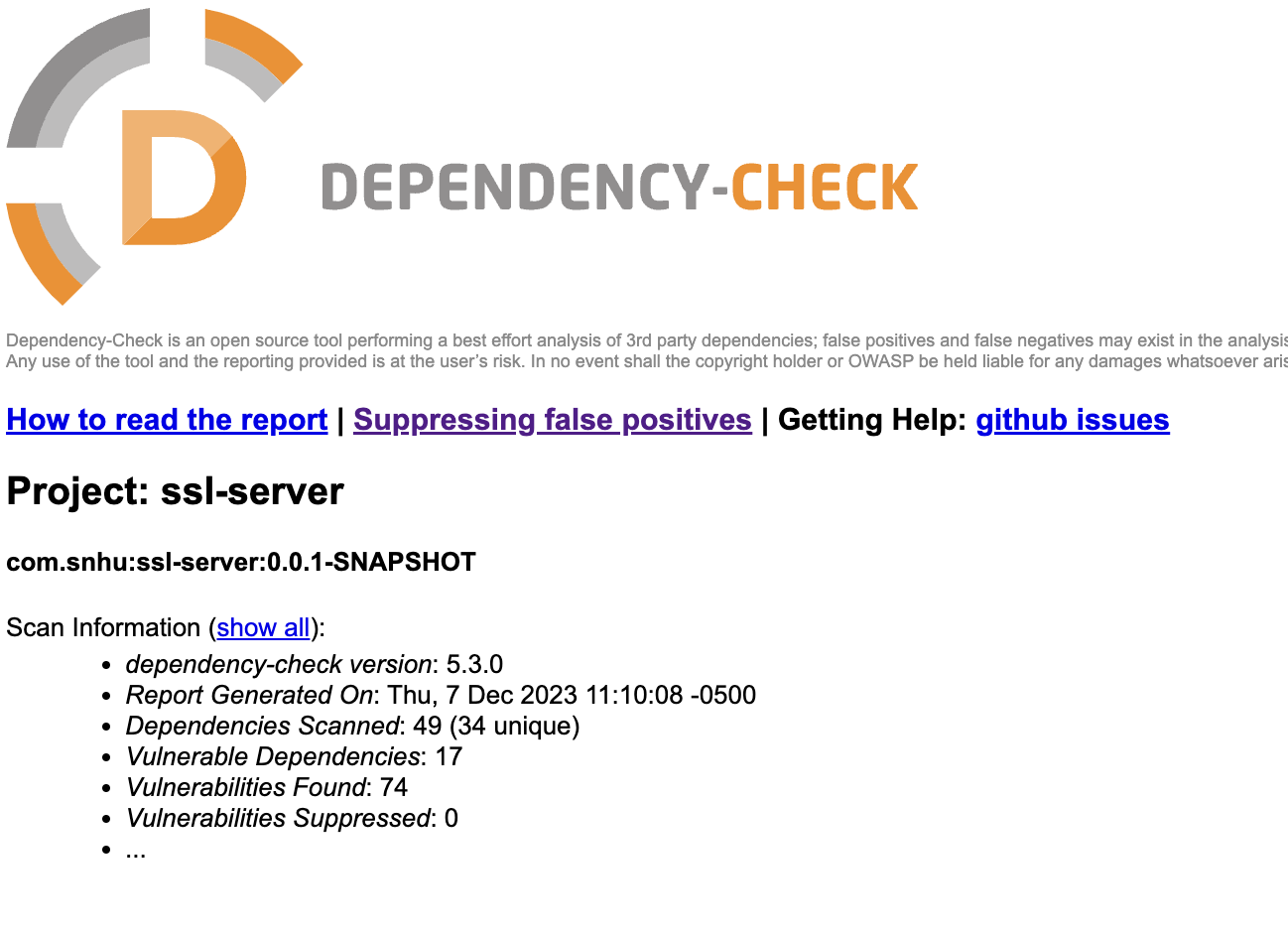




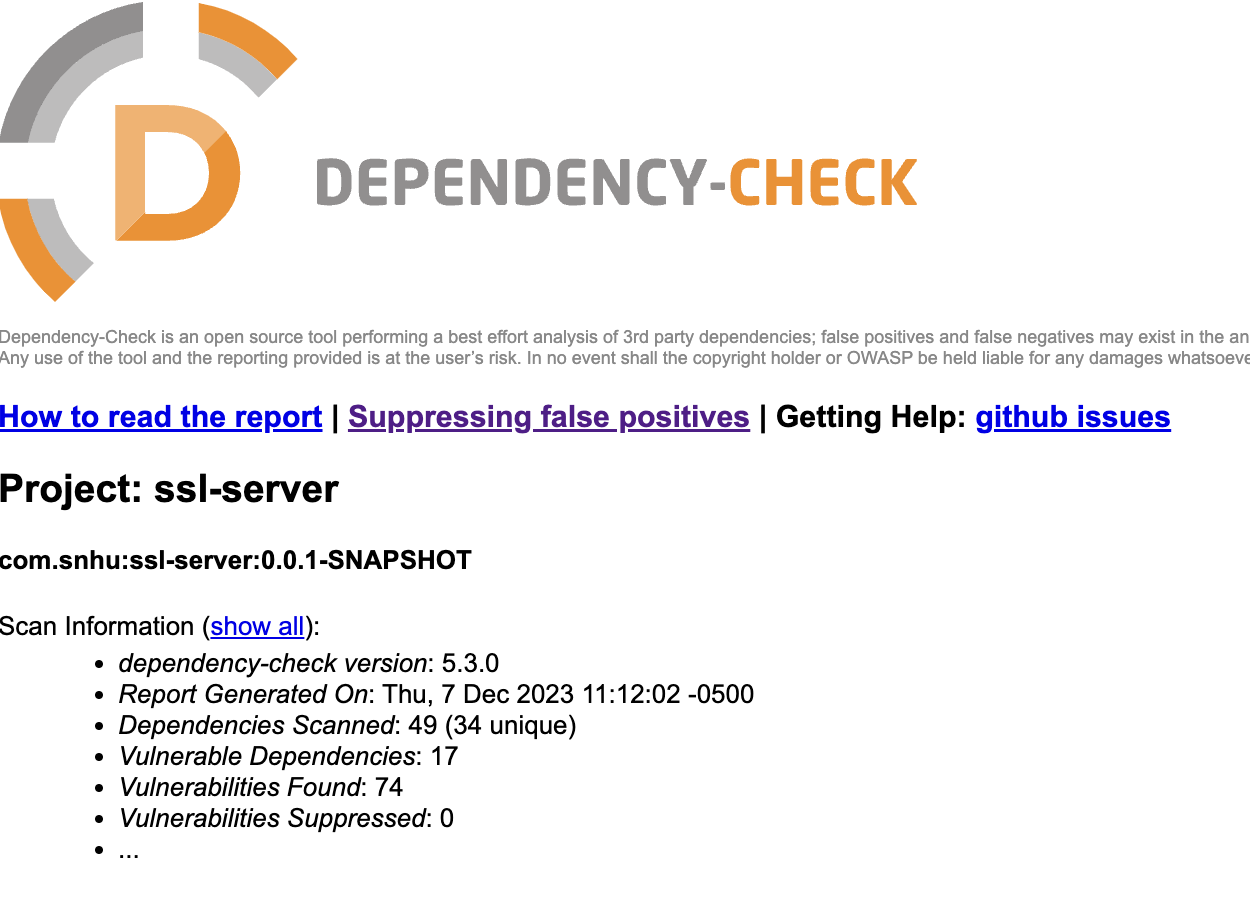
[Insert screenshots here.]

## Secondary Testing

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



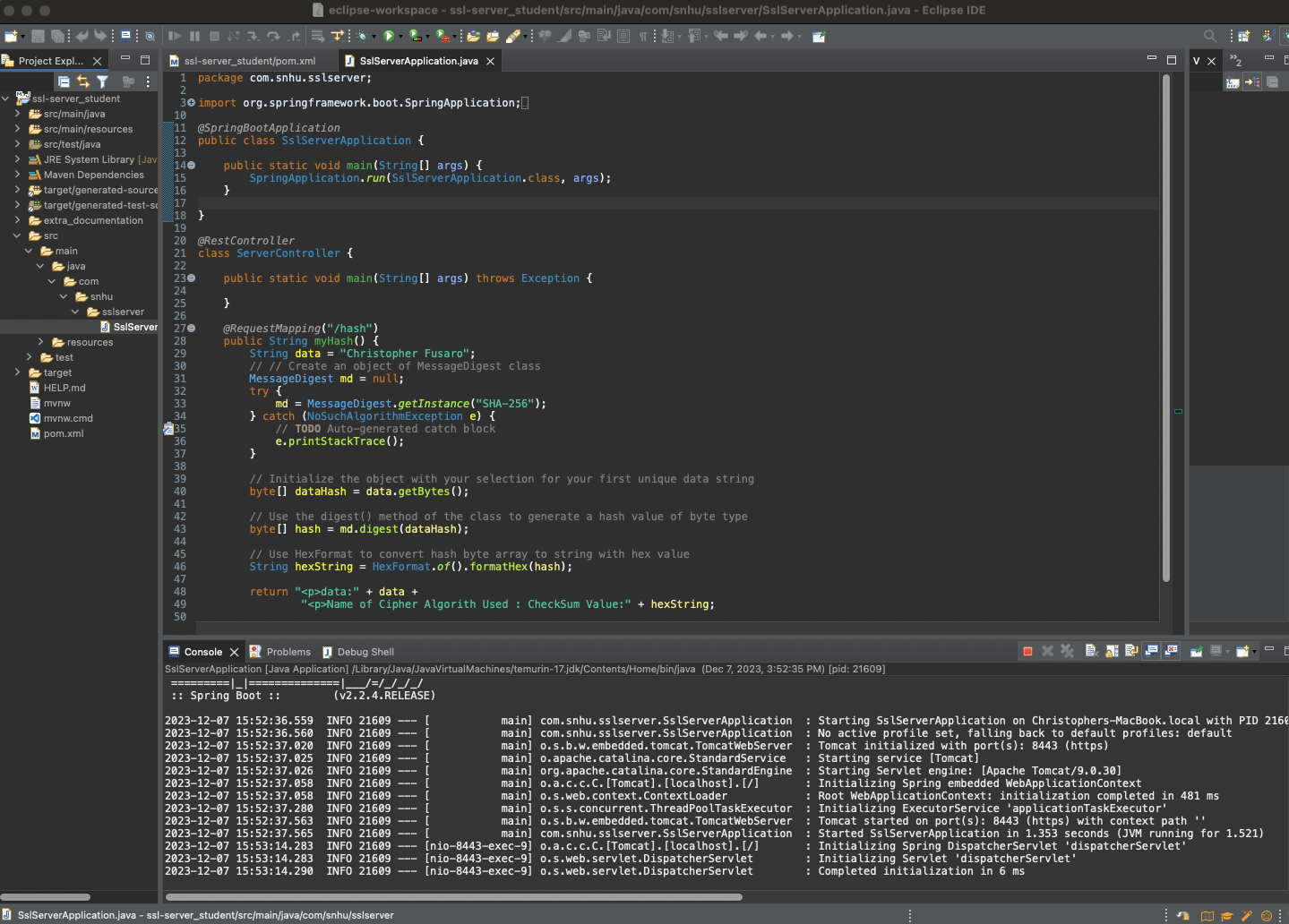
Dependency-check run on the initially supplied code base.



Dependency-check after code was implemented and new packages were called for hashing. No new vulnerabilities were found. So, there was no need for a suppression file on code that was added to the project.

## Functional Testing

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



## Summary

Summary: Discuss how the code has been refactored and how it complies with security testing protocols. In the summary of your practices for secure software report, be sure to address the following:

Refer to the Vulnerability Assessment Process Flow Diagram. Highlight the areas of security that you addressed by refactoring the code.

Discuss your process for adding layers of security to the software application.

With the use of the Vulnerability Assessment Process Flow Diagram, I have concluded that the following areas are potential security risks. APIs, cryptography, client/server, and code quality. The application itself is designed as a client-server architecture. It has been secured with self-generated SSL certificates that are AES 128-bit encryption. AES 128 bit that has been deployed properly has not been compromised. So I feel this will be strong enough for the present. An eye should be kept on the future as we stand on a precipice that threatens to undermine all encryption as we know it. Quantum computing is the future, and it maybe the undoing of all encryption in its current form.

As for code quality there were 74 vulnerabilities found by running a static testing suite. No new vulnerabilities were added with additional code. The quality of code addition to the Artemis financial application may be modest, but it was scrutinized, and error protection was added to ensure that the code is safe and effective. This multi-faceted approach to security adds layers of protection, so if one is breached, the other layers will be able to ensure that the application or data is not compromised.

## Industry Standard Best Practices

Some industry-standard best practices that may be employed in this application are to ensure all software packages are up to date and that any vulnerable packages that are used are not compromising to the organization. Secure communication is paramount in today's day and age, where you can be held responsible for compromising employee and customer data. A balance must be struck with the level of encryption. strong enough to protect user data but not so resource-intensive that it will slow down the application and cause unacceptable latency.