****

# 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** | **12/10/23** | **Riley Bailey** |  |

## 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

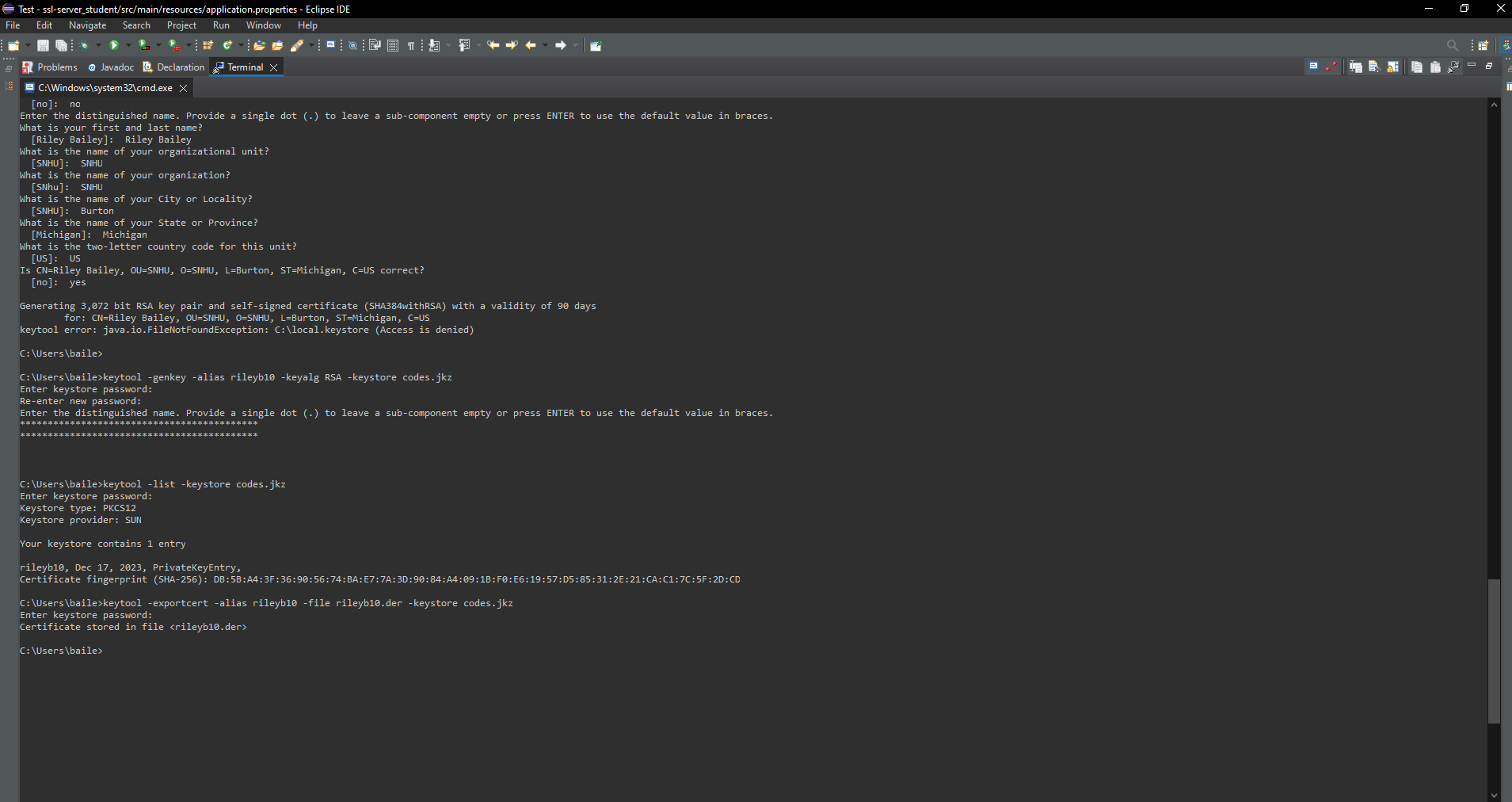
Riley Bailey

## Algorithm Cipher

For Artemis Financial, the algorithm cipher that I would recommend is AES, which stands for the Advanced Encryption Standard and will utilize SHA-256 for the checksums. Let me explain. AES is a high-security-level system that provides encryption for bit sizes ranging from 128, 192, to 256 bits, covering the various bit levels of the cipher. SHA-256 will handle the hash function aspect of this cipher, where it will be used to generate the checksums. The use of random numbers in this process is crucial because, to gain access to the certificate, one would have to correctly guess both numbers. For instance, if the bit size is 256, the odds of getting both right are 2^256, which is nearly impossible to predict due to the immense number of possibilities. To put that into perspective, there are 77 numbers after the initial 1 in this immense number. Next, let us dive into the differences between symmetric and non-symmetric keys. Symmetric keys use only a single shared key, which is great for its simplicity. However, the drawback is that it is only one key, so securely distributing it to the necessary people could be challenging. On the other hand, the asymmetric key has two separate keys: the first key is the public key, which provides information to the plaintext, while the other key provides information to the ciphertext (Group). Before 2001, the widely used encryption algorithm was DES, providing security up to 56 bits. The transition to AES in 2001 was driven by the need for stronger security measures, as AES offered key lengths of up to 256 bits, surpassing the limitations of DES. This shift to AES illustrates why it became the preferred choice for many.

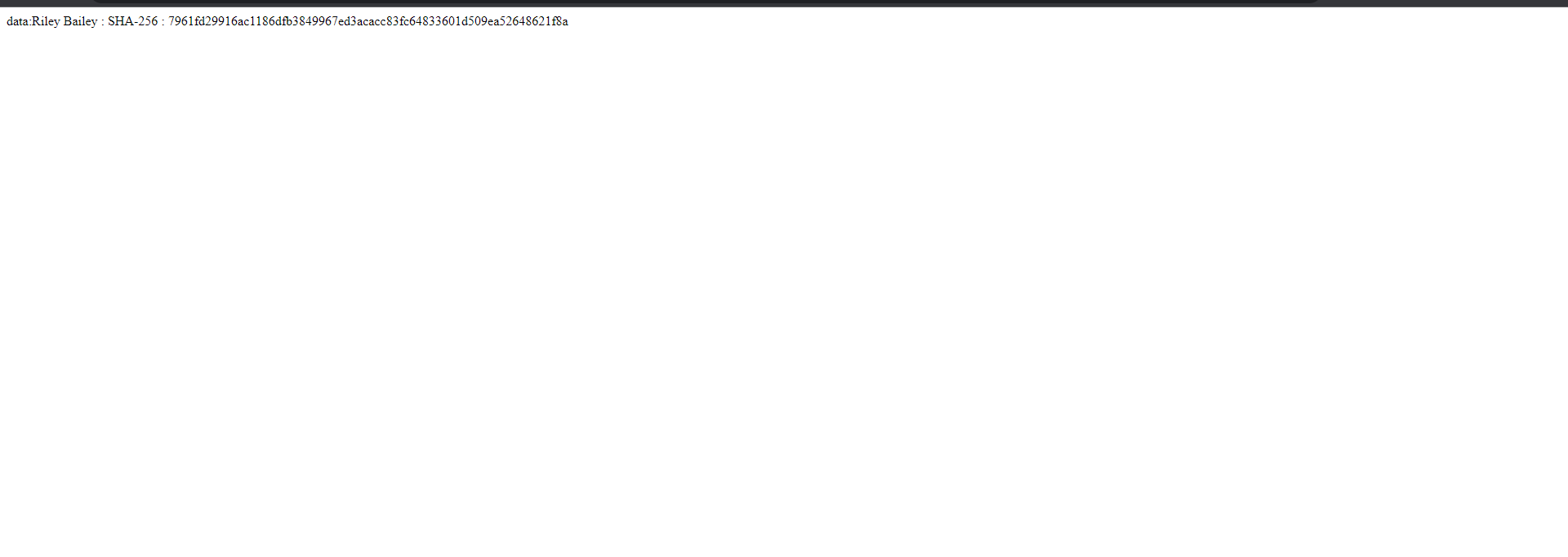
## Certificate Generation

Insert a screenshot below of the CER file.

[]

## Deploy Cipher

Insert a screenshot below of the checksum verification.

[]

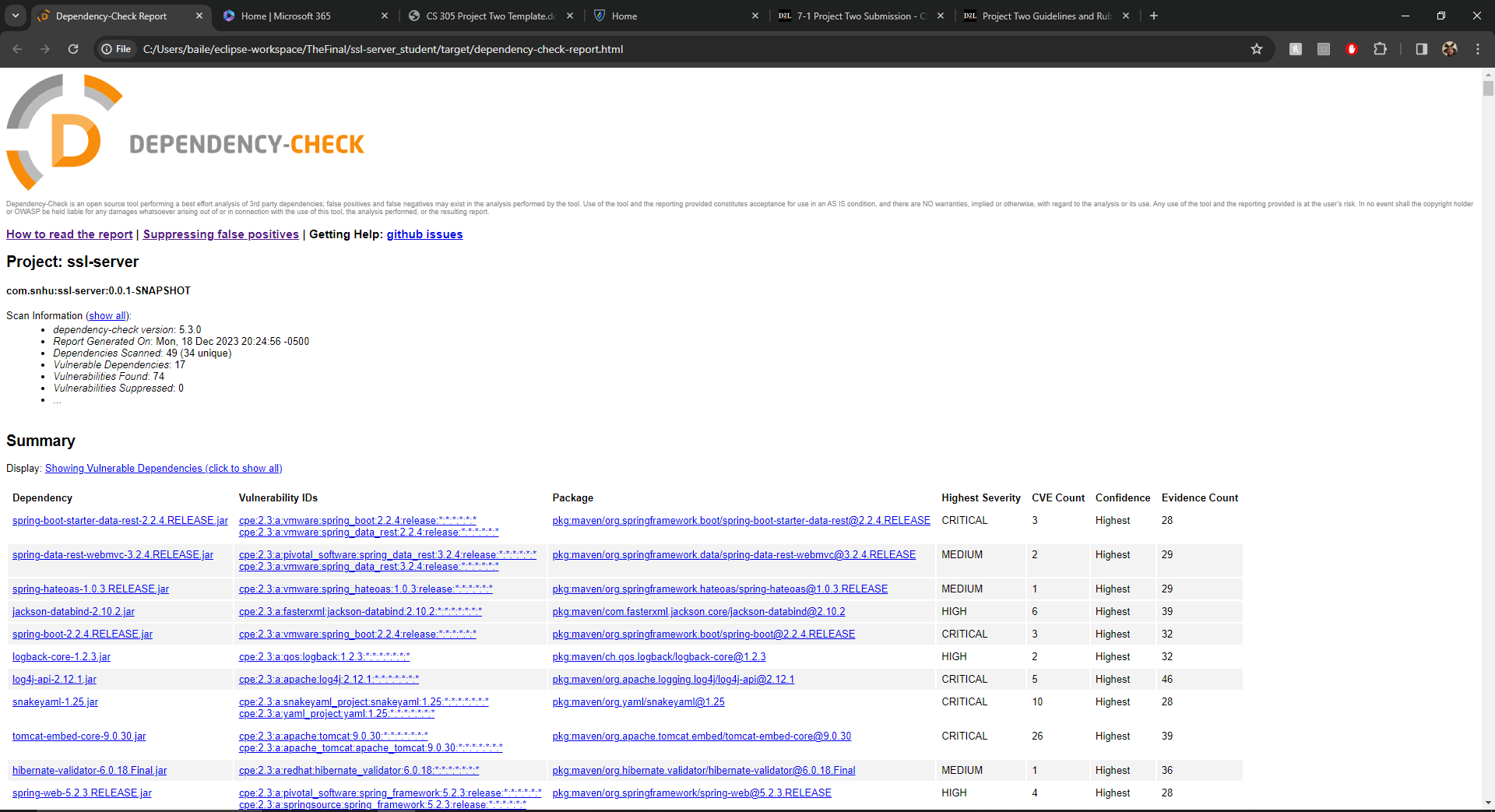
## Secure Communications

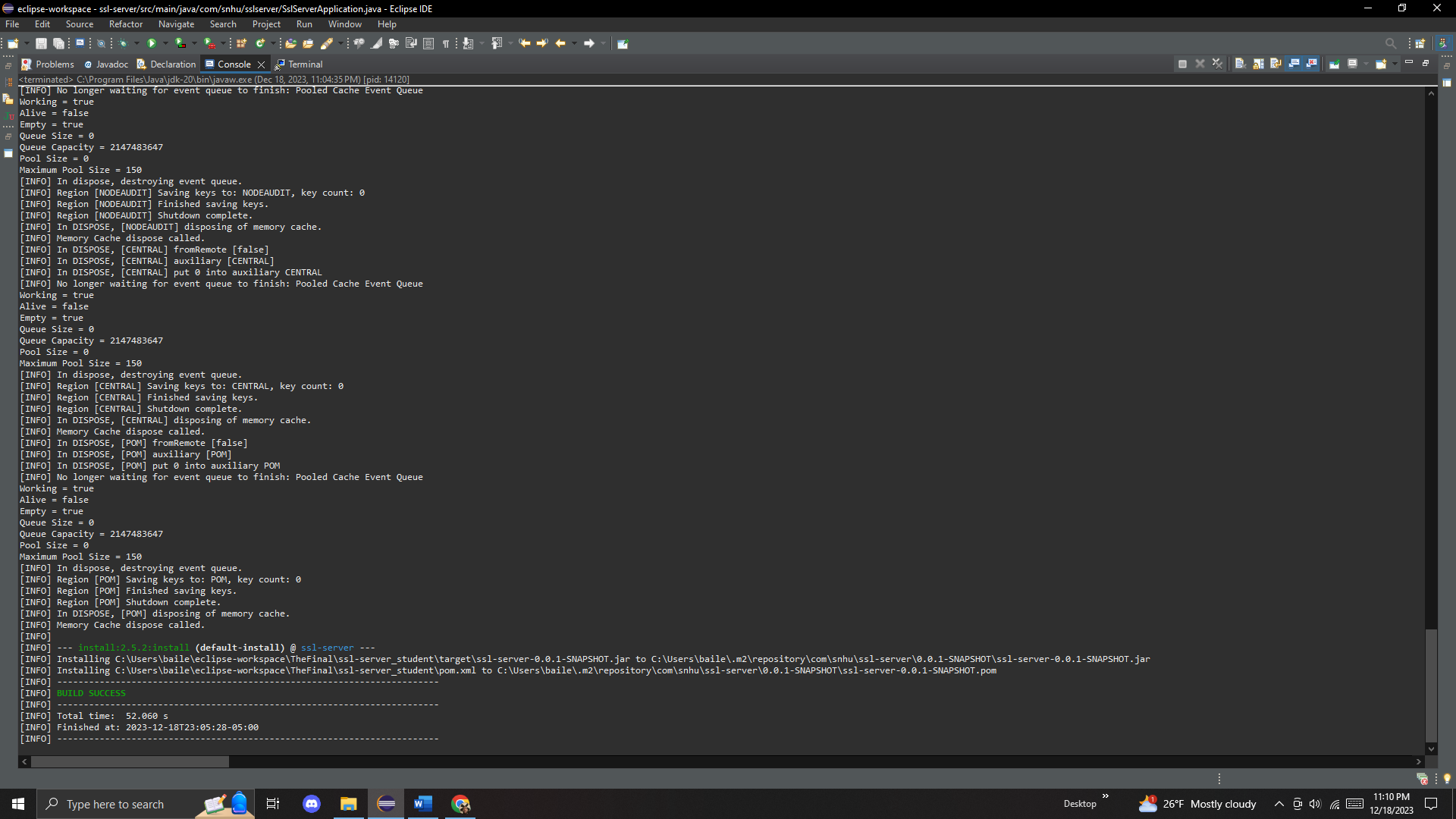
Insert a screenshot below of the web browser that shows a secure webpage.

[Insert screenshots here.]

## Secondary Testing

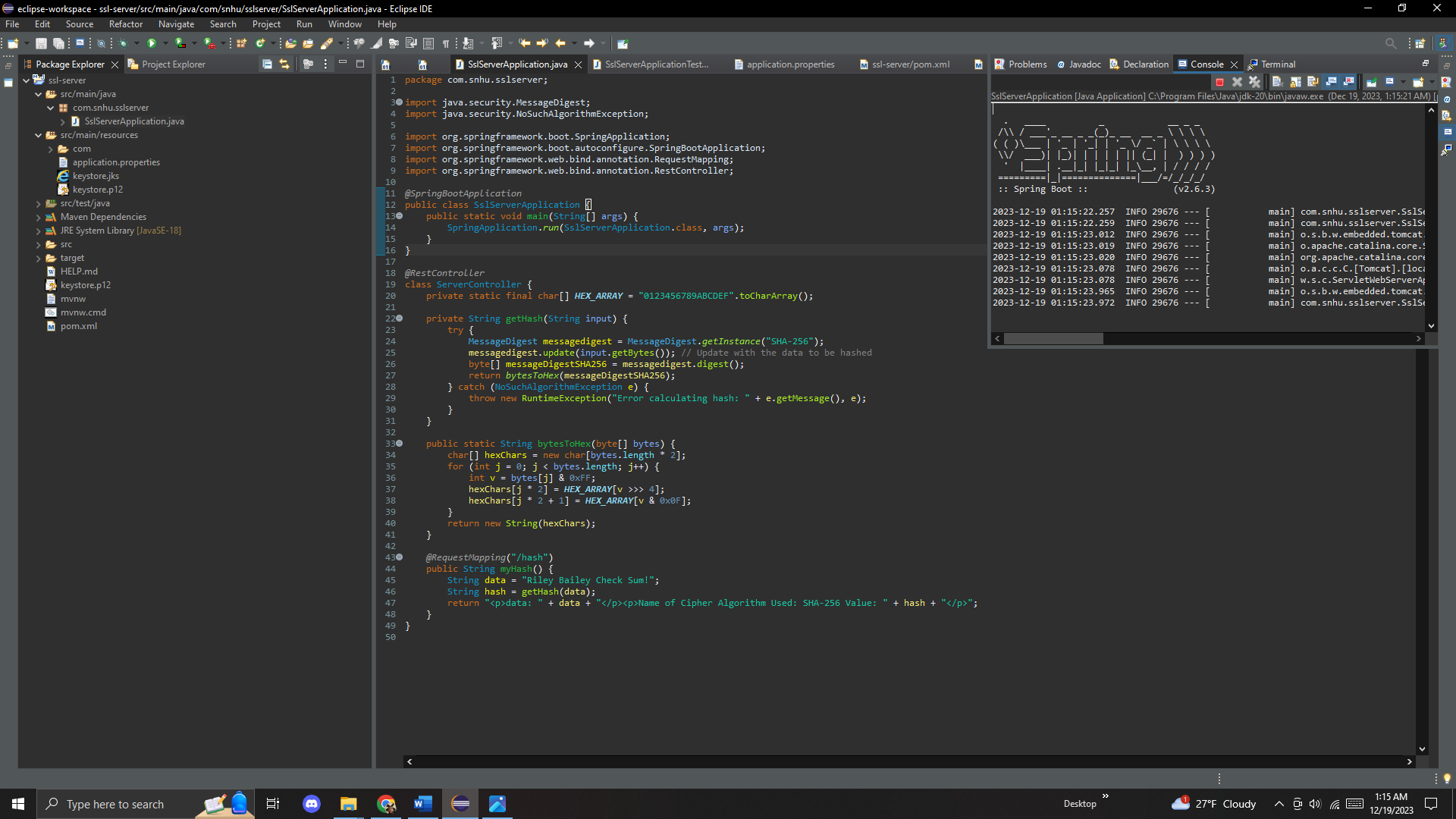
Insert screenshots below of the refactored code executed without errors and the dependency-check report.

[]



## Functional Testing

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

[]

## Summary

The way I handled it was by inputting APIs to enable secure communication, achieved by generating the certificate. I initially chose AES and SHA-256, but as I progressed, I leaned more towards SHA-256. I also delved into dependency checking, contributing to code quality. Not having a robust dependency checker could lead to issues that may go unnoticed, potentially becoming a significant problem, especially when dealing with attempts to hack into sensitive information. A vulnerability opens numerous possibilities in such scenarios. The refactored code encompasses all the mentioned measures, enhancing security against hackers and ensuring robust encryption.

## Industry Standard Best Practices

In developing the software, I attempted implementing industry standard best practices for secure coding, focusing on secure input validation, authentication, and encryption using some AES principles and mainly SHA-256. Regular code reviews and continuous improvement efforts were undertaken to promptly address and rectify security vulnerabilities, ensuring the software's ongoing resilience against evolving threats. The application of these practices mitigated immediate risks and contributed to the company.

References

Group, T. (n.d.). *What is an asymmetric key or asymmetric key cryptography?* <https://cpl.thalesgroup.com/faq/key-secrets-management/what-asymmetric-key-or-asymmetric-key-cryptography#:~:text=Asymmetric%20keys%20are%20the%20foundation,kept%20private%20(private%20key>).