****

# 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** | **July 28, 2024** | **Ryan Shaffer** |  |

## 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

Ryan Shaffer

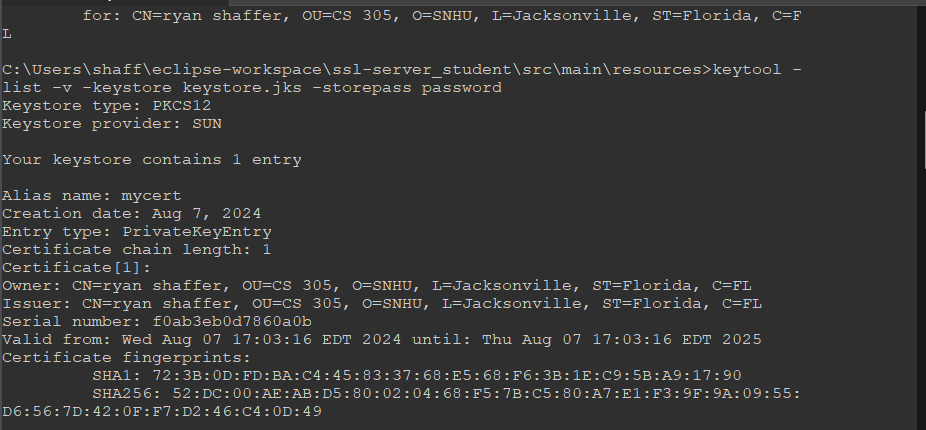
## Algorithm Cipher

Recommended Algorithm: AES (Advanced Encryption Standard)

* Overview: AES is a symmetric encryption algorithm widely used across the globe. It provides high security and performance and is considered the standard for encrypting sensitive data.
* Hash Functions and Bit Levels: AES supports various key lengths, including 128, 192, and 256 bits. The longer the key, the stronger the encryption.
* Symmetric vs. Asymmetric: AES is a symmetric key algorithm, which means the same key is used for encryption and decryption. This makes it fast and suitable for encrypting large amounts of data.
* Random Numbers: AES uses random initialization vectors (IVs) for encrypting data blocks, ensuring that the same plaintext will encrypt to different ciphertexts.
* History and Current State: AES was established by the U.S. National Institute of Standards and Technology (NIST) in 2001. It is widely adopted and is the encryption standard for many government and private-sector applications.

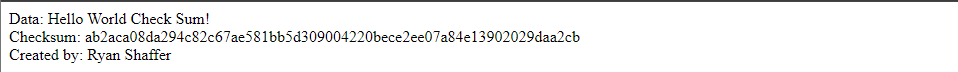
## Certificate Generation

Insert a screenshot below of the CER file.



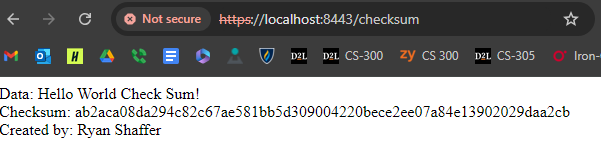
## Deploy Cipher

Insert a screenshot below of the checksum verification.



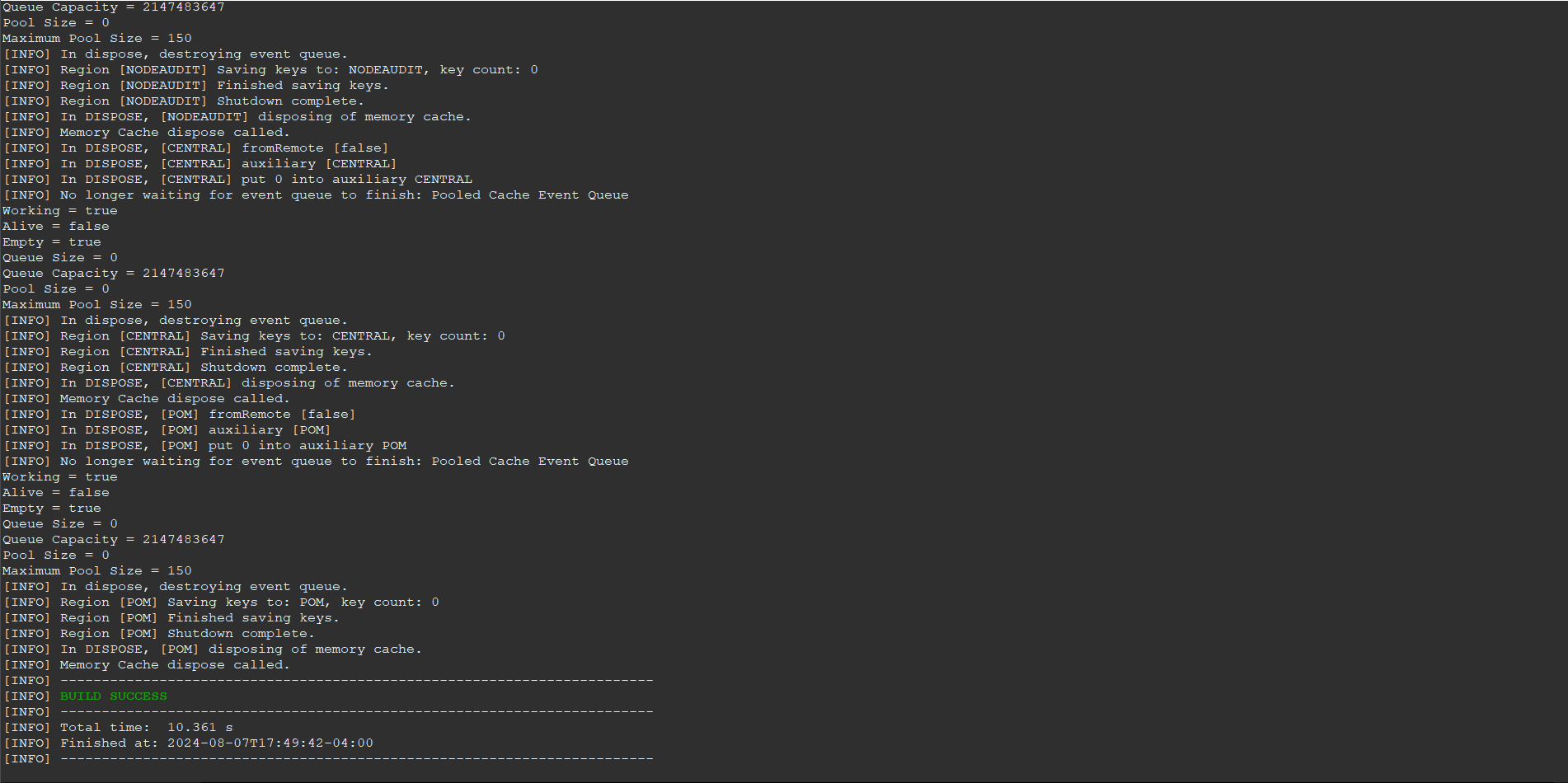
## Secure Communications

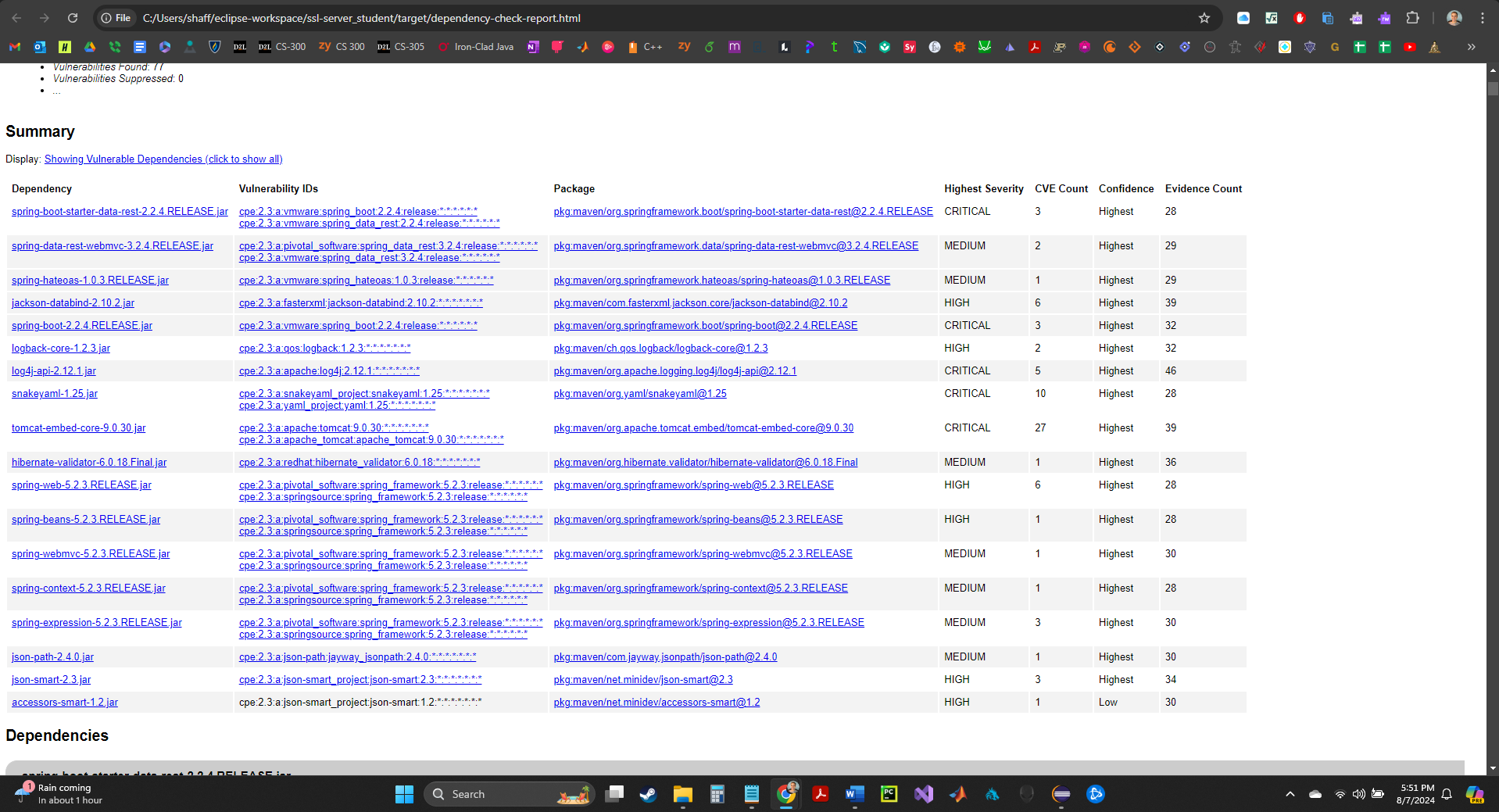
Insert a screenshot below of the web browser that shows a secure webpage.



## Secondary Testing

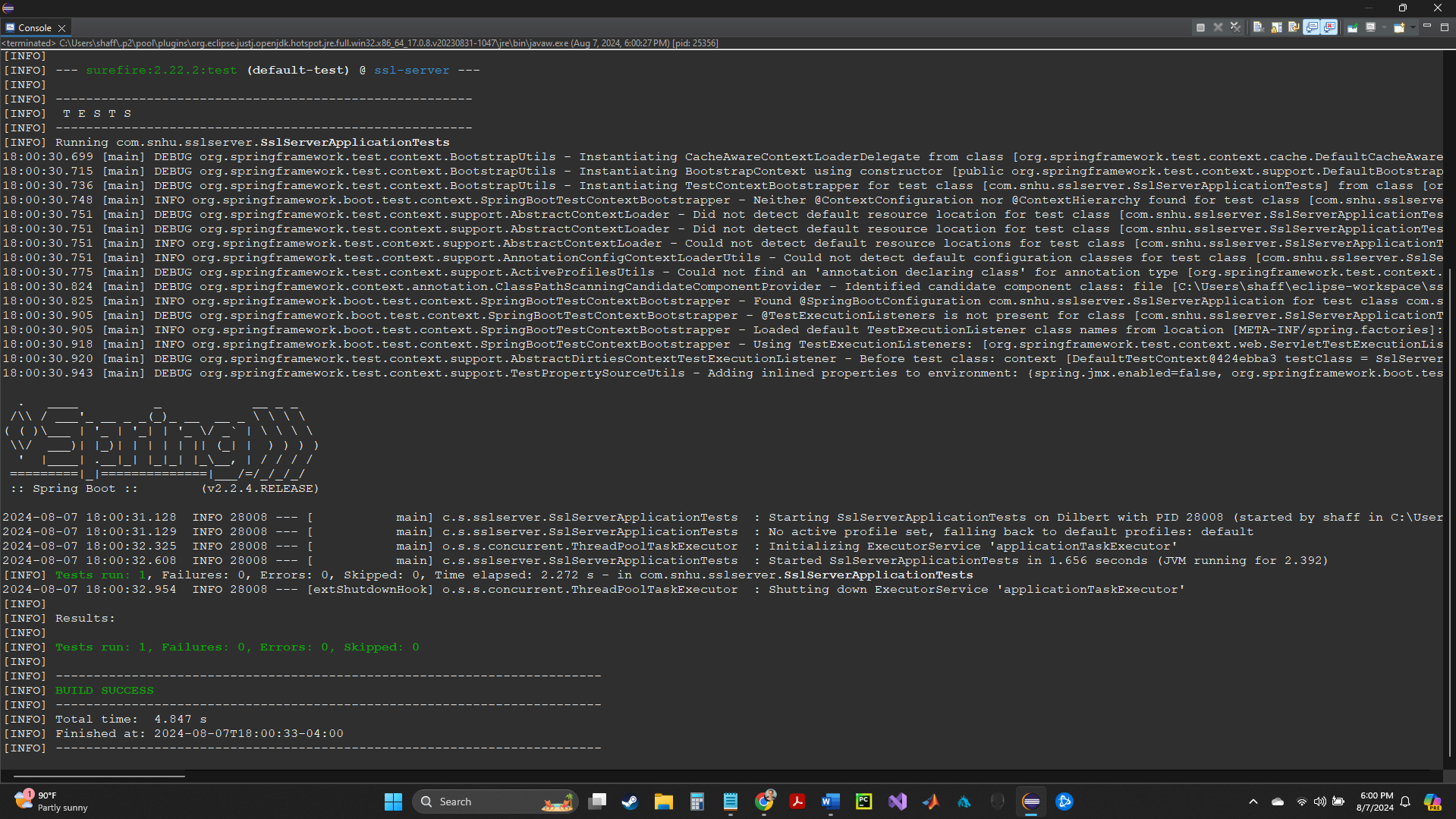
Insert screenshots below of the refactored code executed without errors and the dependency-check report.





## Functional Testing

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



## Summary

The code has been refactored to enhance security and ensure compliance with security testing protocols. Key areas of security addressed during the refactoring include the implementation of HTTPS for secure communication, the use of SHA-256 hashing for data integrity verification, and proper error handling to prevent information leakage. These enhancements align with the vulnerability assessment process flow diagram, focusing on identifying and mitigating potential security risks throughout the application's lifecycle.

To add layers of security to the software application, I implemented several measures. First, I configured the application to use HTTPS by setting up an SSL certificate, ensuring that all data transmitted between the client and server is encrypted. Next, I used the SHA-256 algorithm to generate checksums, providing a robust method for verifying data integrity. Additionally, I reviewed the code for potential security flaws, such as hardcoded sensitive information, and ensured that proper error handling mechanisms were in place to prevent the exposure of sensitive data.

## Industry Standard Best Practices

Throughout the development process, I applied industry-standard best practices for secure coding to mitigate known security vulnerabilities. This involved adhering to secure coding guidelines, such as input validation, error handling, and proper management of sensitive data, to maintain the application's existing security. Using established security libraries and frameworks, such as Spring Boot's built-in support for HTTPS, ensured that the application leveraged well-tested components with a track record of security.

Applying industry-standard best practices for secure coding is crucial to the company's overall well-being. It minimizes the risk of security breaches, protects sensitive customer data, and enhances the organization's reputation as a provider of secure and reliable software solutions. By adhering to these best practices, the company demonstrates a commitment to security, which is essential for maintaining customer trust and ensuring compliance with regulatory requirements. Additionally, a robust security posture reduces the potential costs associated with data breaches and security incidents, contributing to the company's long-term success and sustainability.