****

# 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** | **8/17/2024** | **Serenity Rogers** |  |

## 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

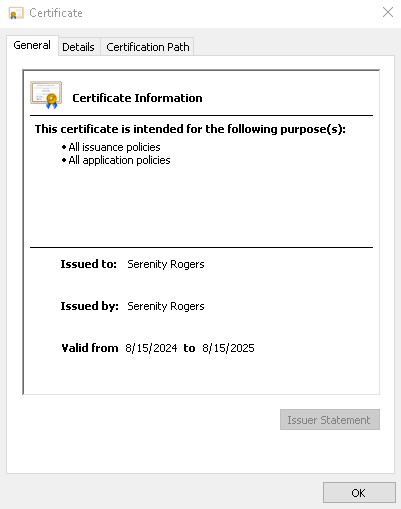
Serenity Rogers

## Algorithm Cipher

**SHA-256** hashing algorithm

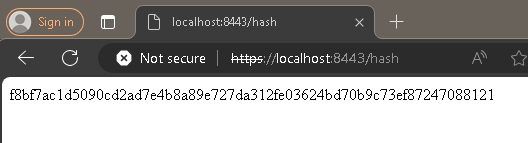
## Certificate Generation

Insert a screenshot below of the CER file.



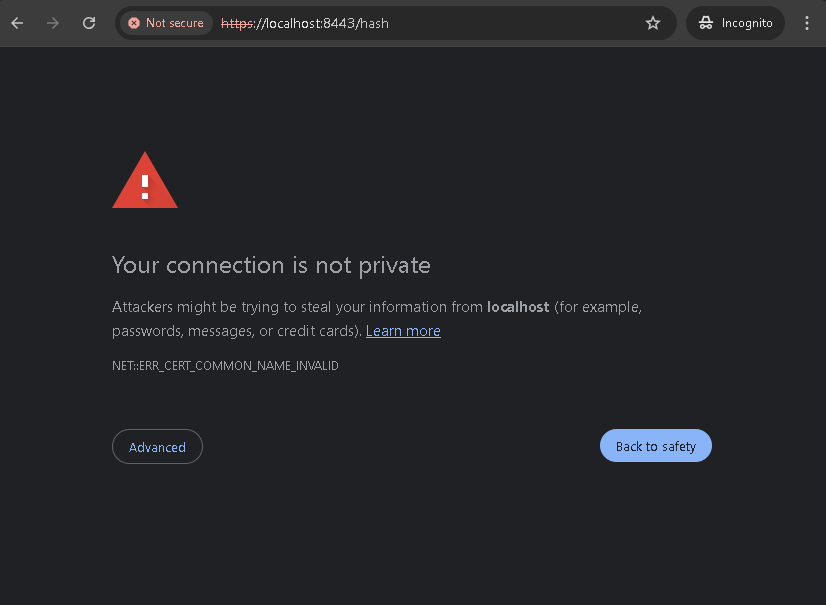
## Deploy Cipher

Insert a screenshot below of the checksum verification.



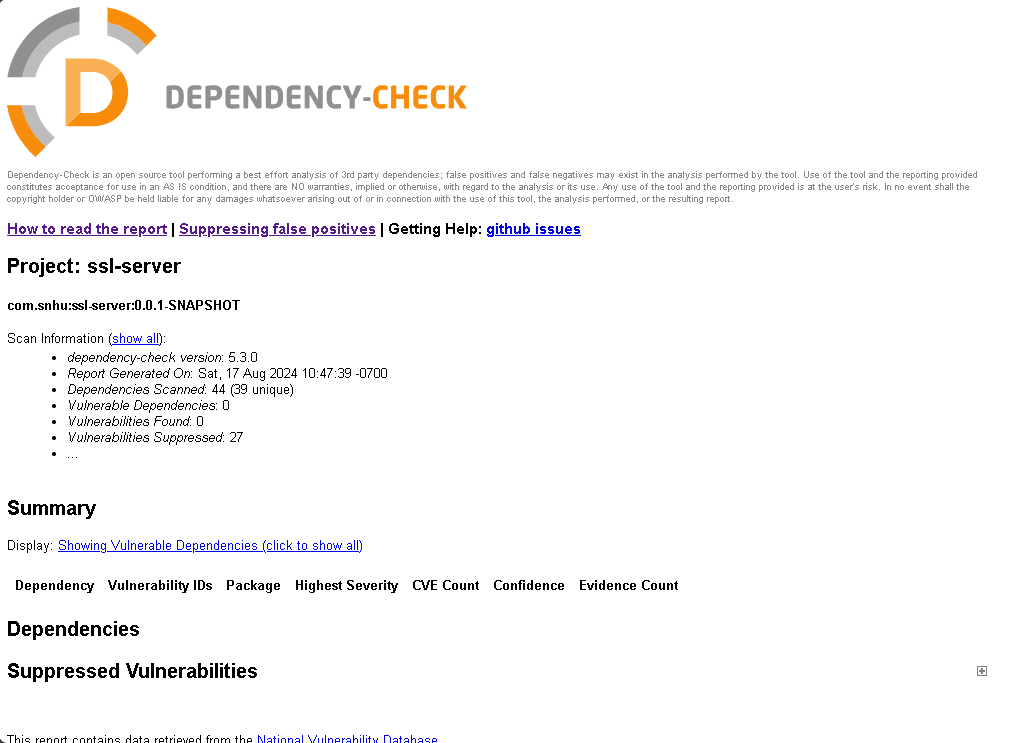
## Secure Communications

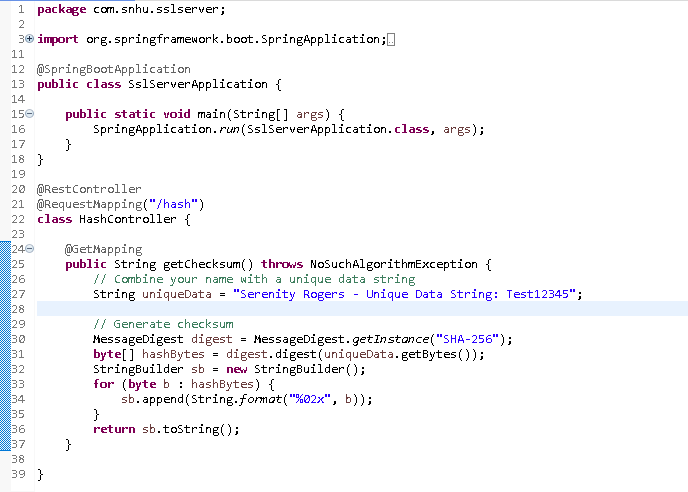
Insert a screenshot below of the web browser that shows a secure webpage.



## Secondary Testing

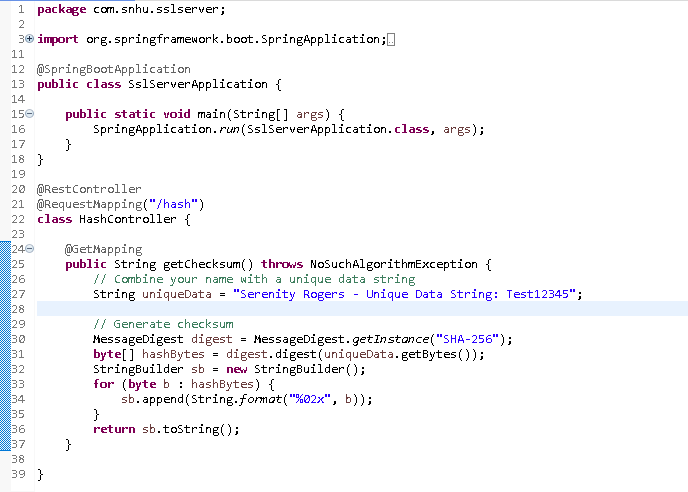
Insert screenshots below of the refactored code executed without errors and the dependency-check report.





## Functional Testing

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



A screen shot of a computer

Description automatically generated

## Summary

Refactoring Code and Compliance with Security Testing Protocols

The primary goal for this project was to secure a Java Spring Boot application by addressing identified vulnerabilities and ensuring compliance with security testing protocols. Initially, the OWASP dependency check plugin identified vulnerabilities in the Spring Boot and Tomcat libraries. By analyzing the reports, I pinpointed specific CVEs associated with outdated dependencies. The refactoring involved updating the Spring Boot version from ‘2.2.4.RELEASE’ to ‘2.7.18’ to mitigate identified vulnerabilities. This upgrade removed several critical security flaws, including CVEs related to temporary directory hijacking and potential security bypass issues. I applied a suppression configuration in the ‘suppression.xml’ file, for the Tomcat vulnerability that could not immediately be resolved, ensuring that only relevant issues are flagged during subsequent scans.

The layered security process included upgrading dependencies to their latest stable versions, suppressing vulnerabilities that could not be immediately resolved, and continuous integration of security testing tools such as the OWASP Dependency Check.

## Industry Standard Best Practices

Using the latest stable versions of Spring Boot and Tomcat dependencies followed industry best practices of using the most recent stable versions of libraries to benefit from the latest security patches. The integration of the OWASP Dependency Check plugin aligns with best practices of automating security scans within the development pipeline, ensuring that vulnerabilities are identified early in the development process. By carefully managing dependencies and ensuring that unnecessary libraries or plugins are not included, the application minimizes the potential attack surface, adhering to the principle of least privilege.

By applying industry best practices, the company mitigates the risk of security breaches that could result in data loss, financial penalties and damage to the company’s reputation. Ensuring that the software adheres to security standards helps the company maintain compliance with industry regulations and builds trust with clients and stakeholders, who expect secure handling of their data.