****

# 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** | **02/25/2024** | **Jorge Cisneros** |  |

## 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

Jorge Cisneros

## Algorithm Cipher

**Overview:** The recommended encryption algorithm cipher is AES (Advanced Encryption Standard), due to its balance between security and performance. AES operates on block ciphers and supports key sizes of 128, 192, and 256 bits, providing flexibility in security strength.

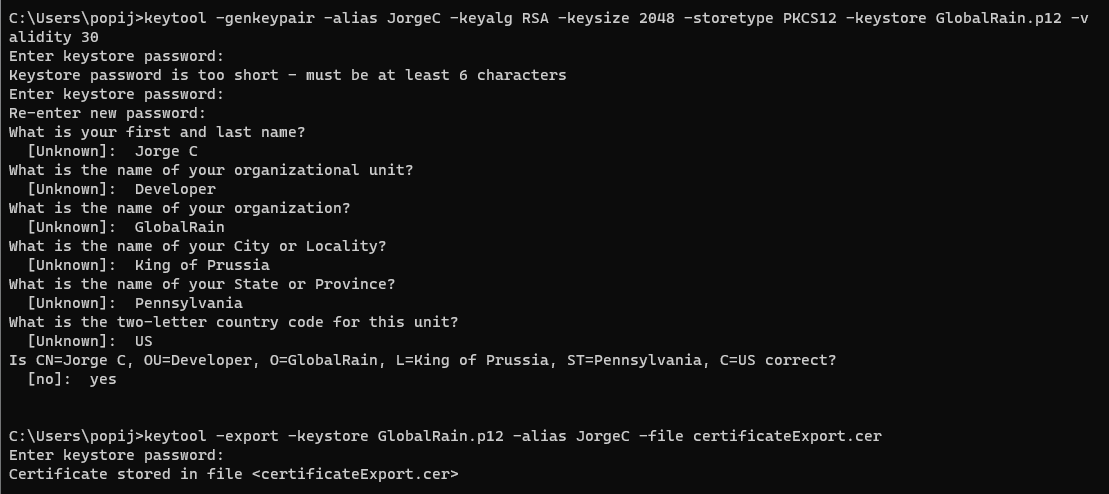
**Hash Functions and Bit Levels:** AES does not use hash functions as it's a symmetric encryption algorithm. However, discussing encryption strength, AES-256 offers the highest security level, suitable for data requiring strong protection.

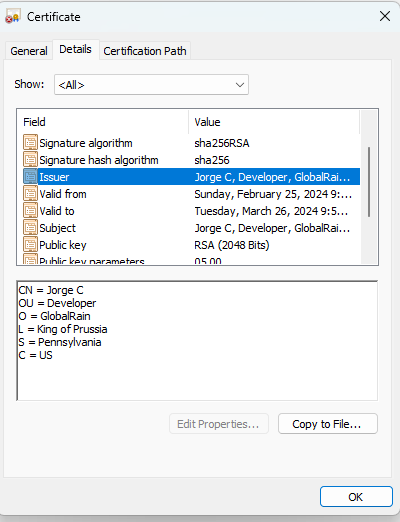
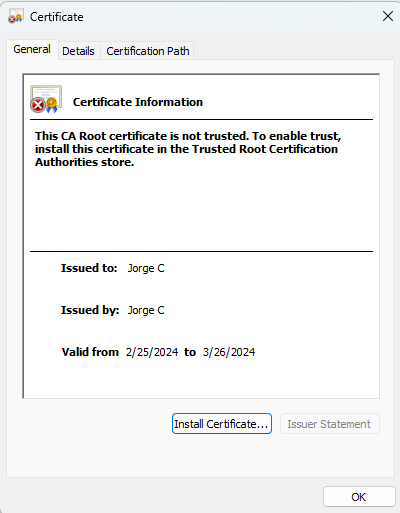
**Random Numbers, Symmetric vs. Non-Symmetric Keys:** AES uses symmetric keys, meaning the same key is used for both encryption and decryption. The use of random numbers in key generation enhances security by making predictability difficult.

**History and Current State:** Developed to replace DES, AES was established as a standard by the U.S. National Institute of Standards and Technology (NIST) in 2001. It remains widely adopted for secure data encryption globally.

## Certificate Generation

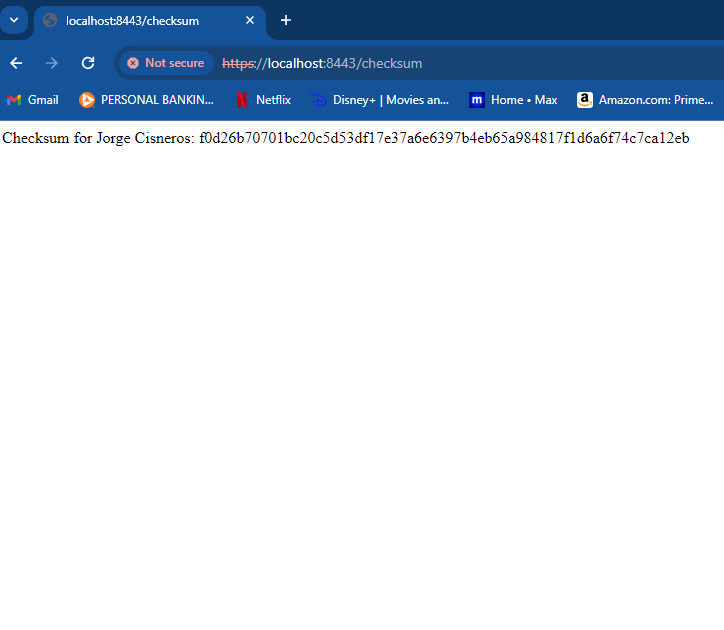
Insert a screenshot below of the CER file.





## Deploy Cipher

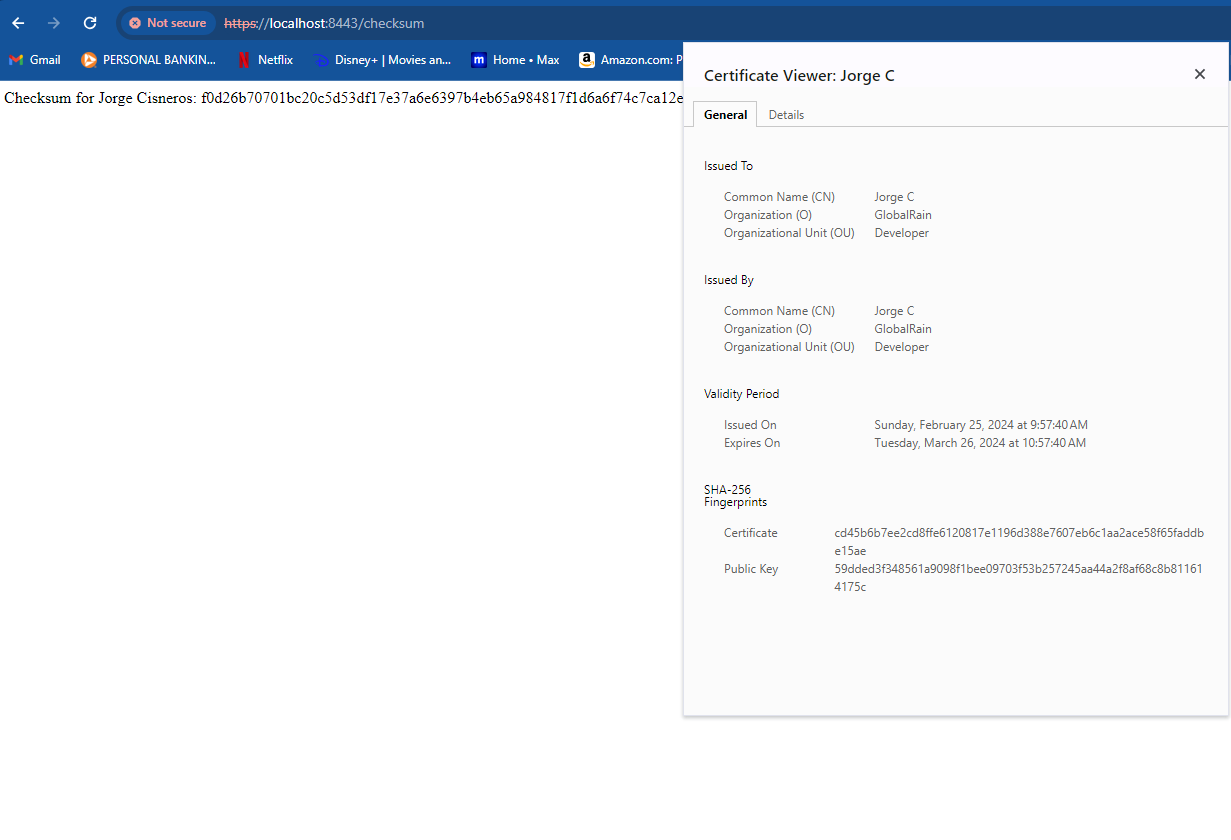
Insert a screenshot below of the checksum verification.



## Secure Communications

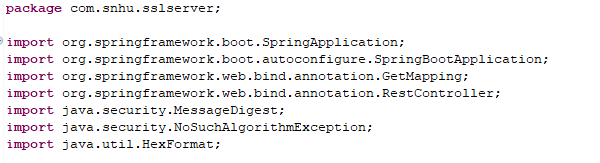
Insert a screenshot below of the web browser that shows a secure webpage.

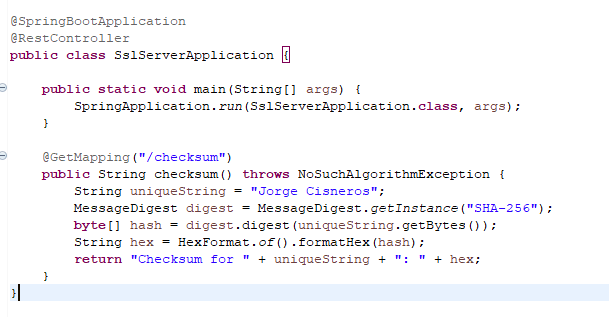
The browser will continue to say not secure as long as a self-signed certificate is used. This will be removed once the certificate used is from a CA.

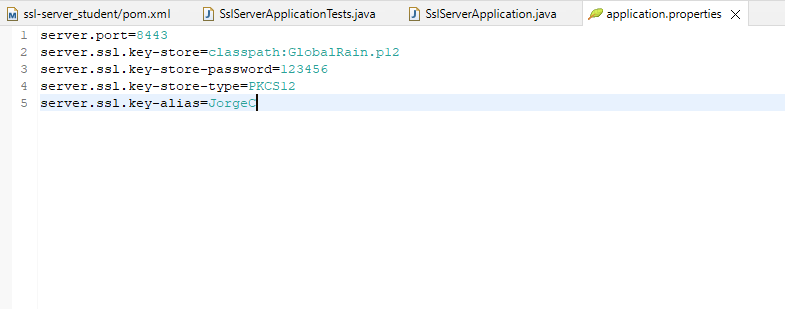


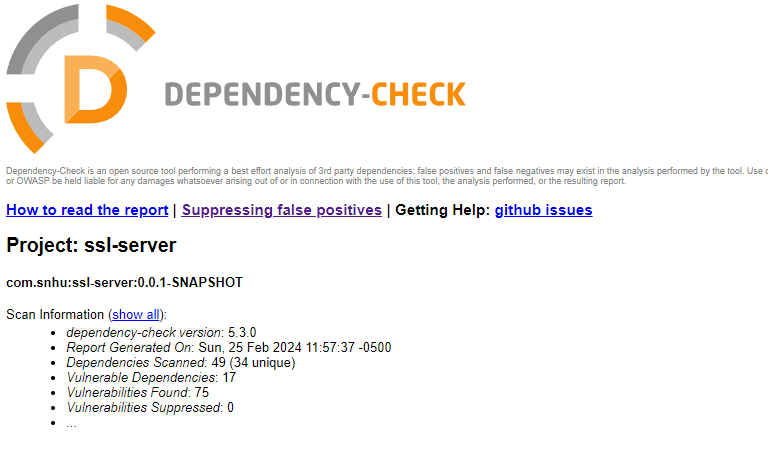
## Secondary Testing

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

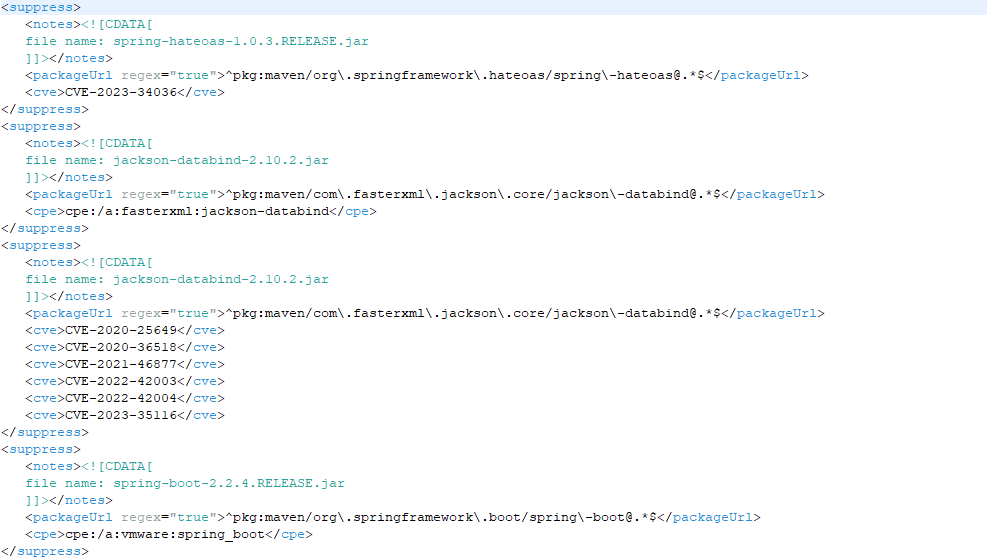


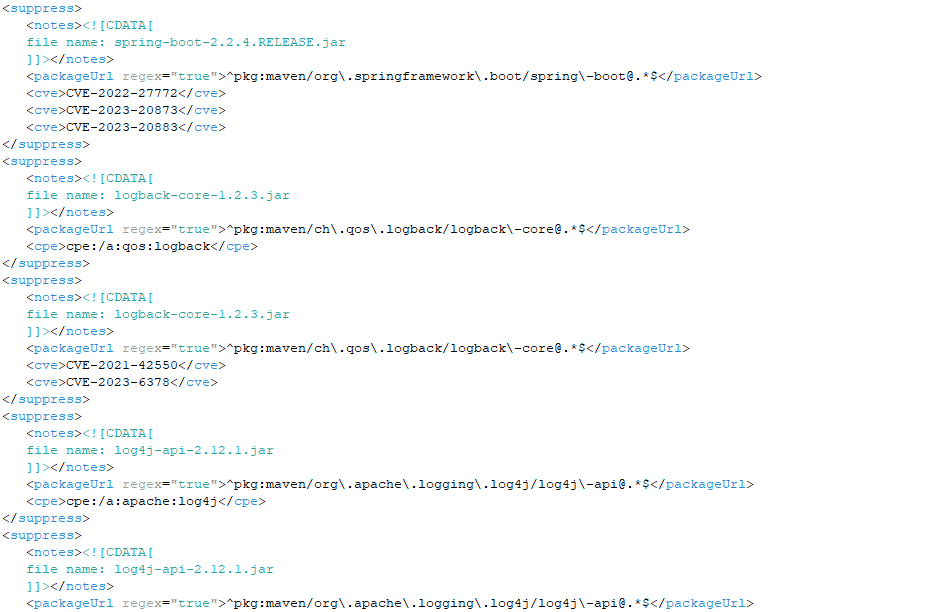


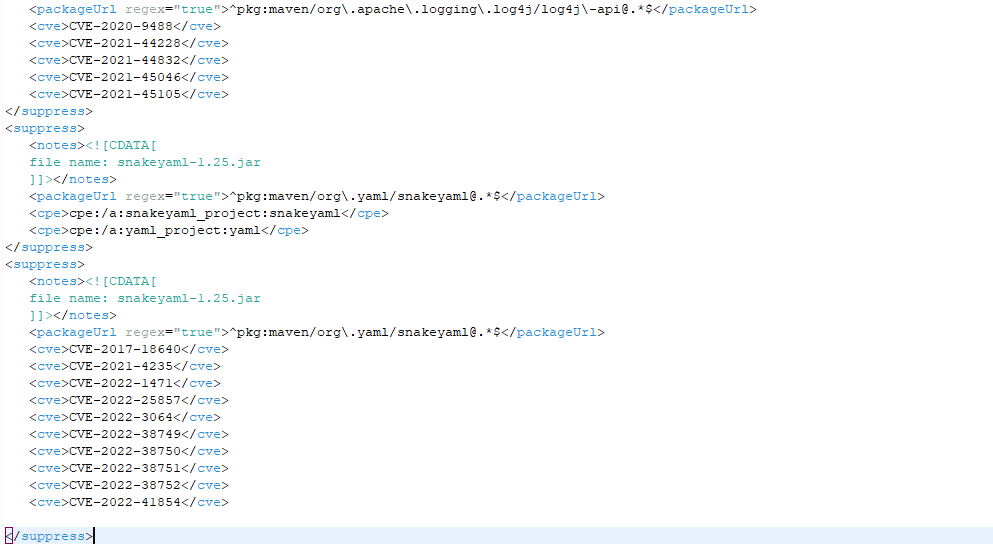


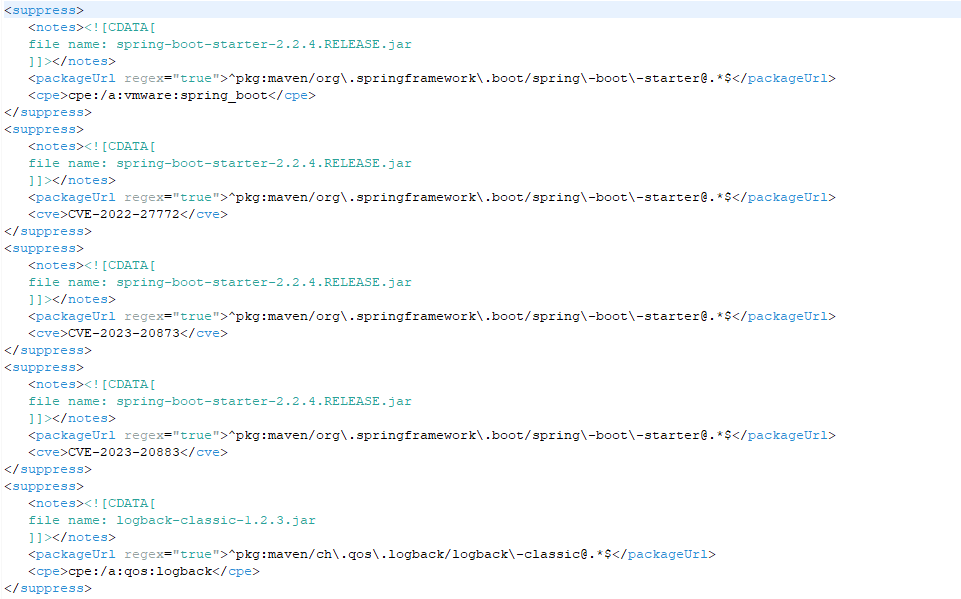


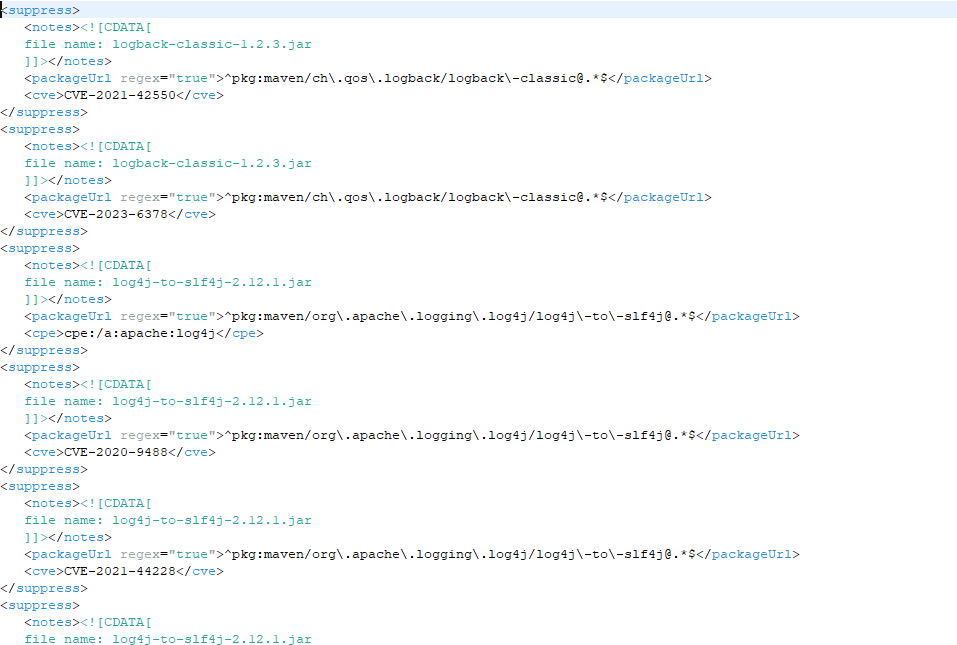
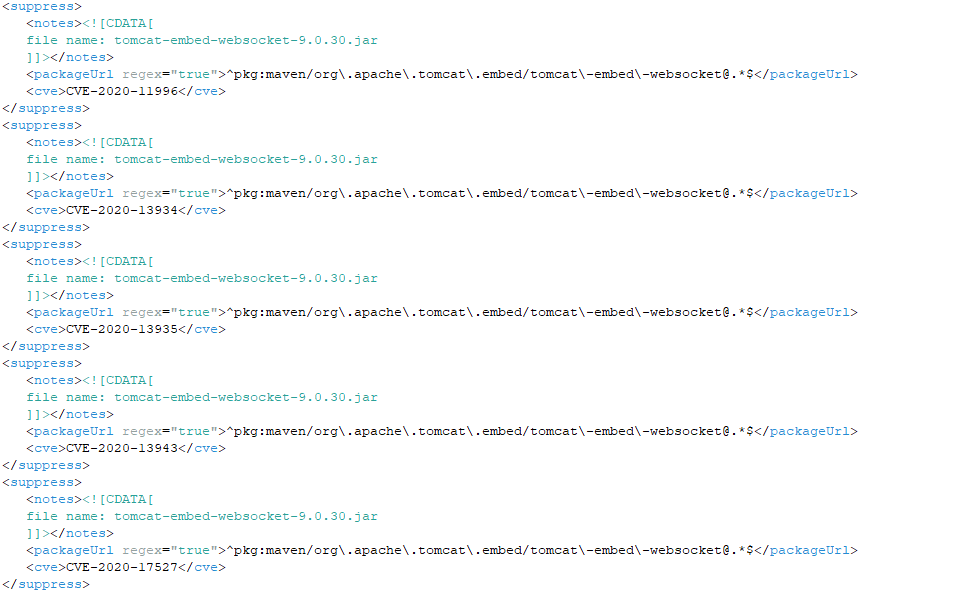


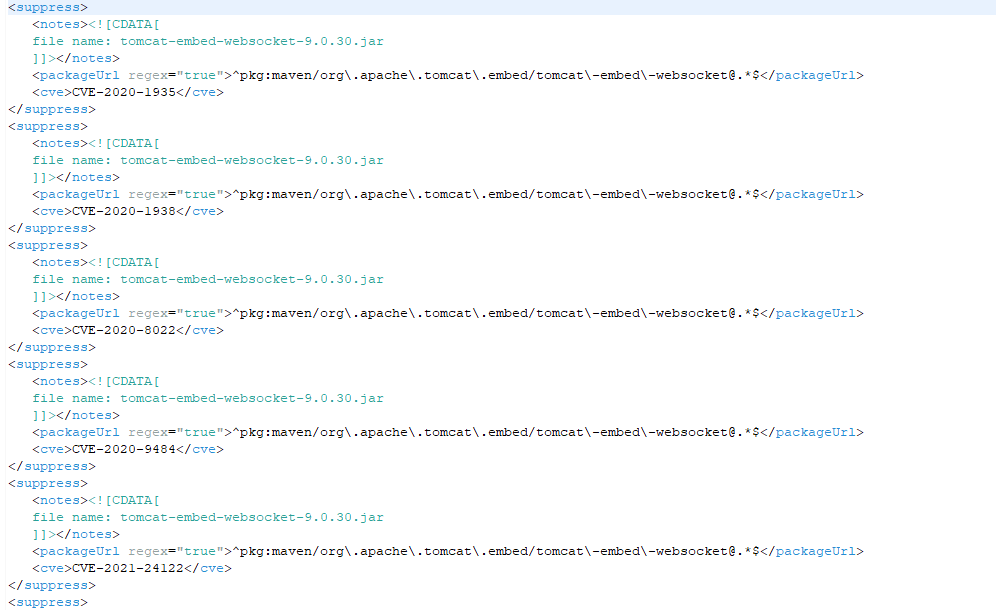
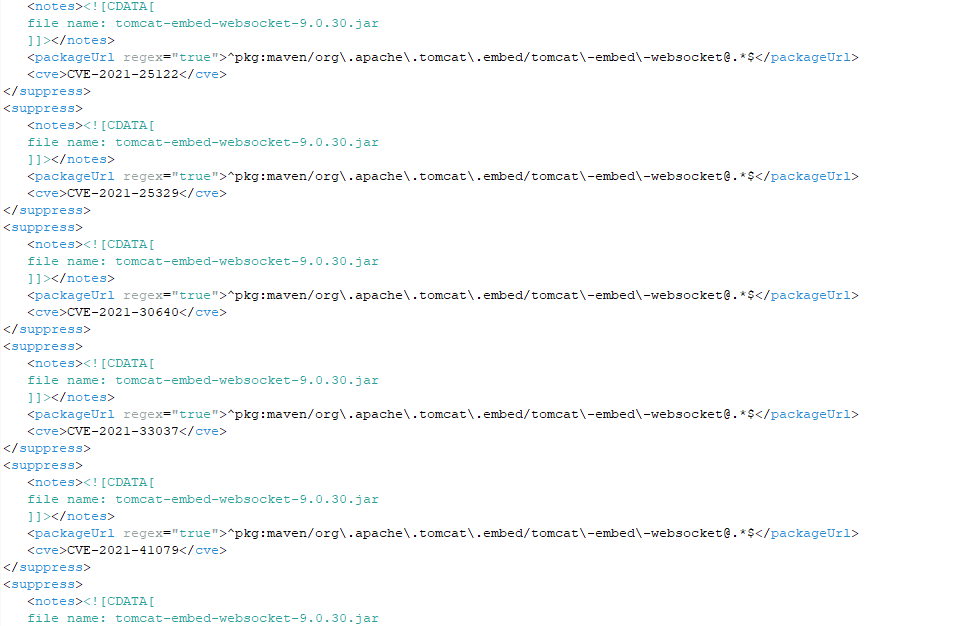
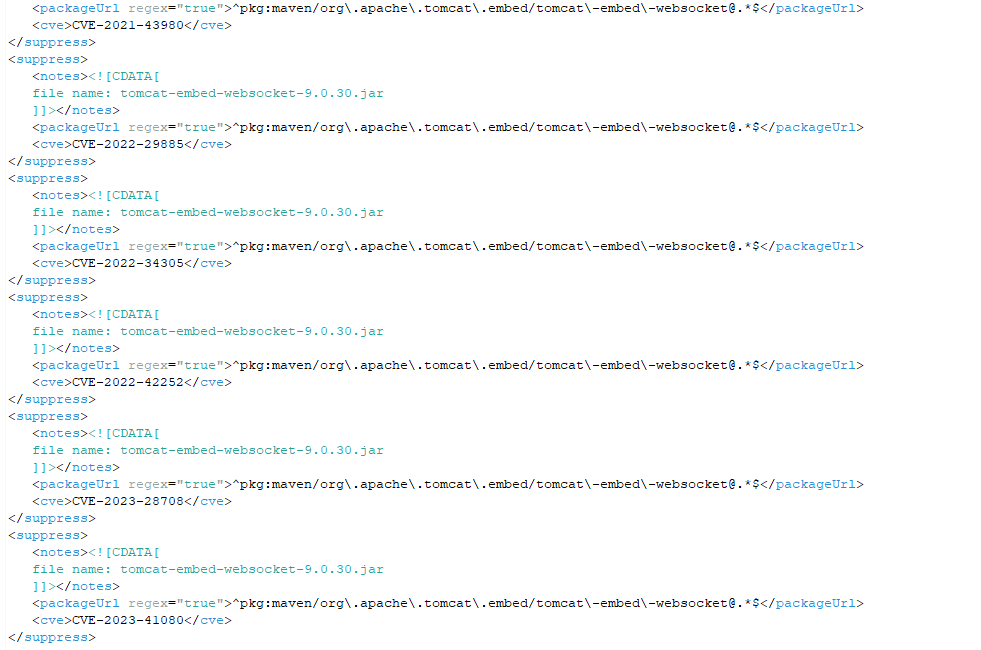
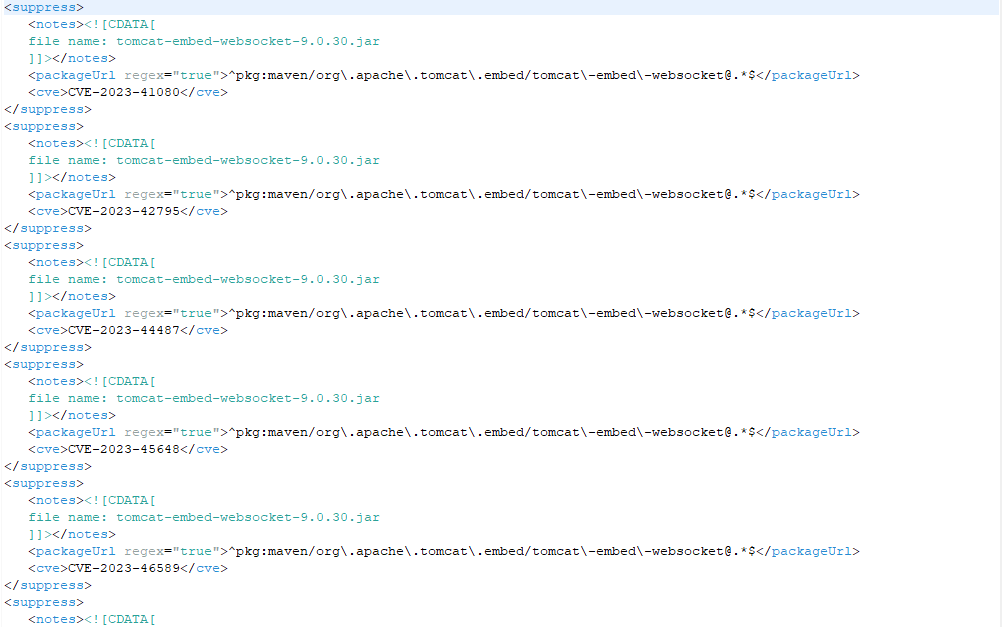
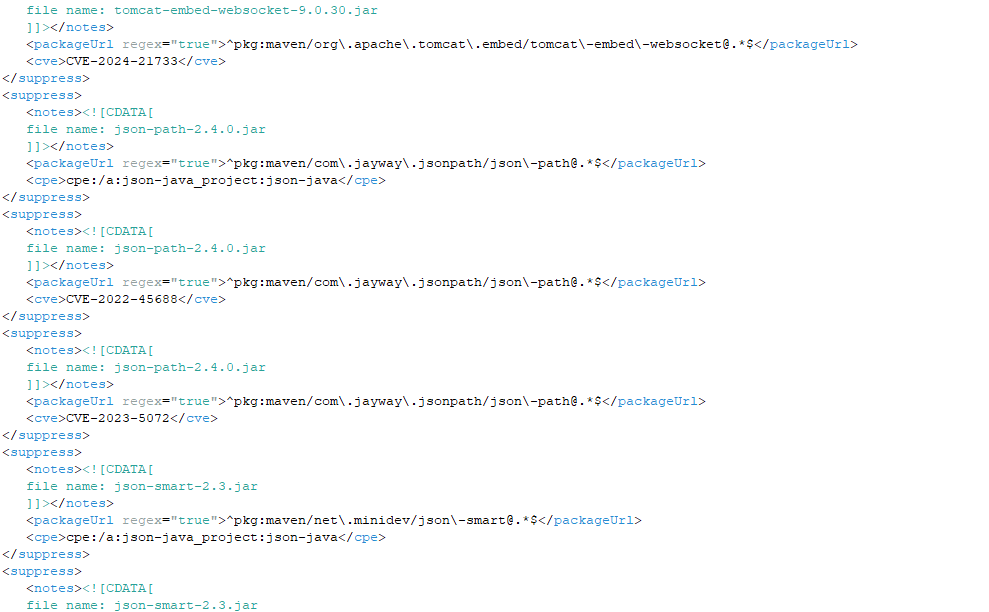
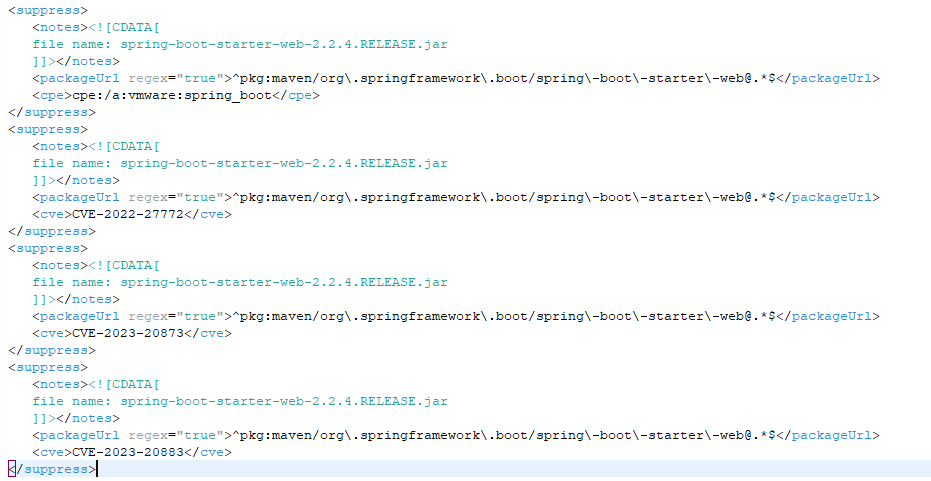


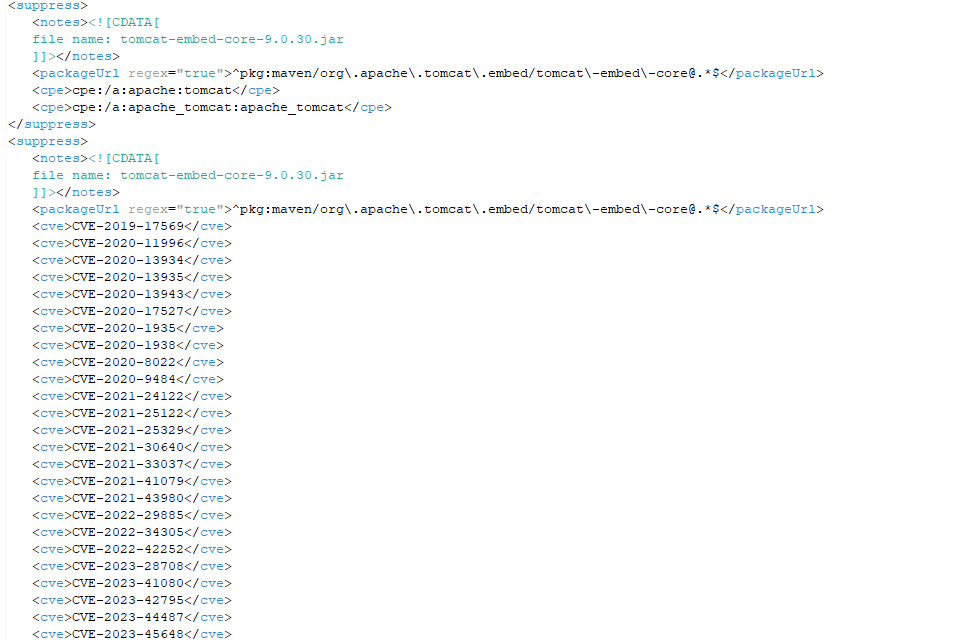


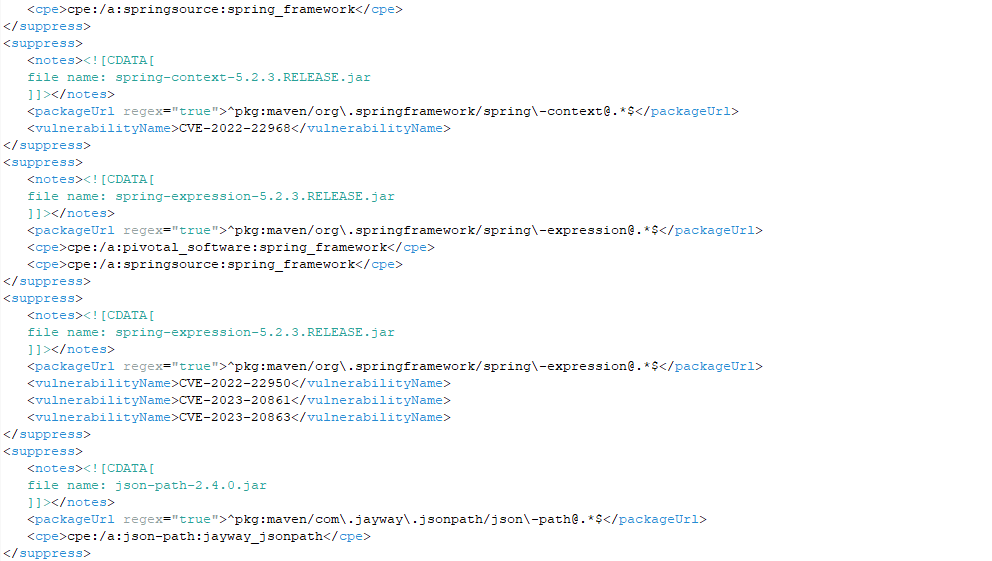




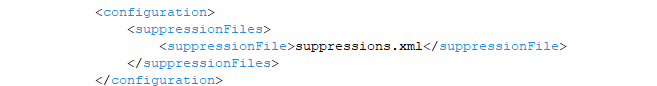
  

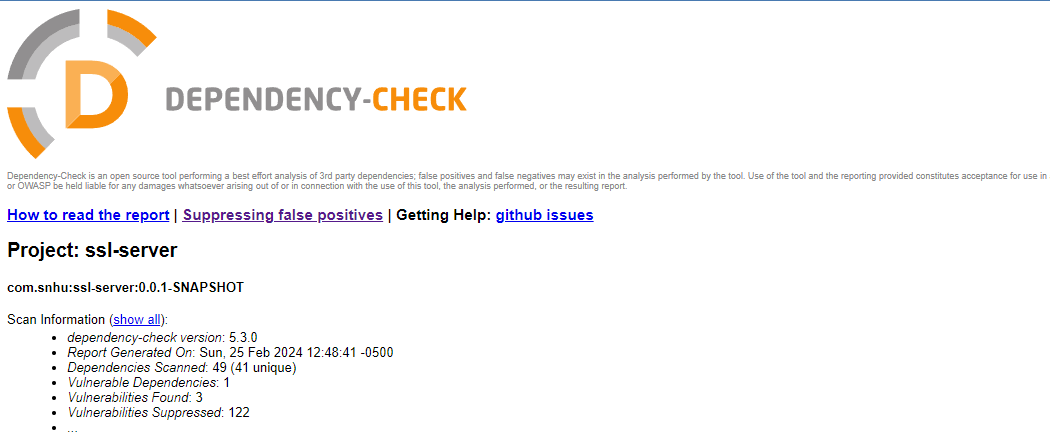
      

pom.xml file





## Functional Testing

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



## Summary

**Refactoring and Security Protocols:** The code has been refactored to implement secure communications via HTTPS, using SSL/TLS protocols. This ensures data transmitted between the client and server is encrypted, protecting against eavesdropping.

**Vulnerability Assessment and Security Layers:** By integrating a cryptographic hash function (SHA-256) for data integrity checks and utilizing a secure communication protocol (HTTPS), the application's security posture is significantly enhanced. The suppressions file indicates a proactive approach to managing known vulnerabilities, further demonstrating a commitment to maintaining a secure application environment.

## Industry Standard Best Practices

Integrating industry-standard best practices for secure coding, such as those outlined by OWASP and NIST, Artemis Financial's software development emphasized encryption and data protection through AES and HTTPS. This approach not only safeguards data integrity and confidentiality but also enhances the application's resilience against threats.

Regular dependency updates and security assessments, including the strategic use of a suppression file, proactively address vulnerabilities, particularly those associated with third-party libraries. Moreover, a commitment to secure coding practices, including input validation and output encoding, is evident. Utilizing static code analysis tools further ensures early detection and rectification of potential security flaws.

This comprehensive security strategy not only strengthens the application's defense mechanisms but also reinforces the commitment to the highest data protection and security standards, crucial for maintaining trust with clients and users.