****

# 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** | **June 25, 2024** | **Akila Jones** |  |

## 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

Akila Jones

## Algorithm Cipher

I would recommend AES (Advanced Encryption Standard) cipher.

**Overview:** The Advanced Encryption Standard (AES) is a symmetric encryption algorithm established by the National Institute of Standards and Technology (NIST) in 2001. It is widely used for securing data due to its strong encryption capabilities and efficiency.

**Hash Function and Bit Levels:** AES can function at several bit levels, commonly 128, 192, or 256 bits. A key size of 256 bits provides a high degree of security.

**Random Numbers and Key Management:** Random numbers ensure the uniqueness and unpredictability of AES keys and IVs, which are fundamental to the security of the encryption process. Proper key management practices, including secure storage, distribution, rotation, and backup, are essential to maintaining the integrity and confidentiality of encrypted data.

**History and current state:** AES, established as the successor to DES, has become the global standard for symmetric key encryption due to its strong security, efficiency, and adaptability. Its continued widespread adoption and resilience against cryptographic attacks underscore its critical role in modern data protection.

## Certificate Generation

Insert a screenshot below of the CER file.



## Deploy Cipher

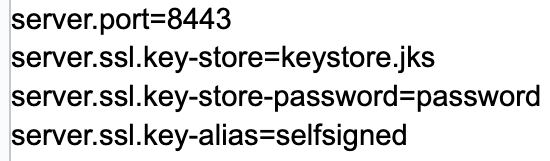
Insert a screenshot below of the checksum verification.

To implement AES encryption, the software code must be refactored to incorporate cryptographic libraries (such as the Java Cryptography Extension, JCE) for managing encryption and decryption operations. A checksum can be utilized to verify that the data remains unchanged after encryption and decryption, thereby ensuring data integrity.



## Secure Communications

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

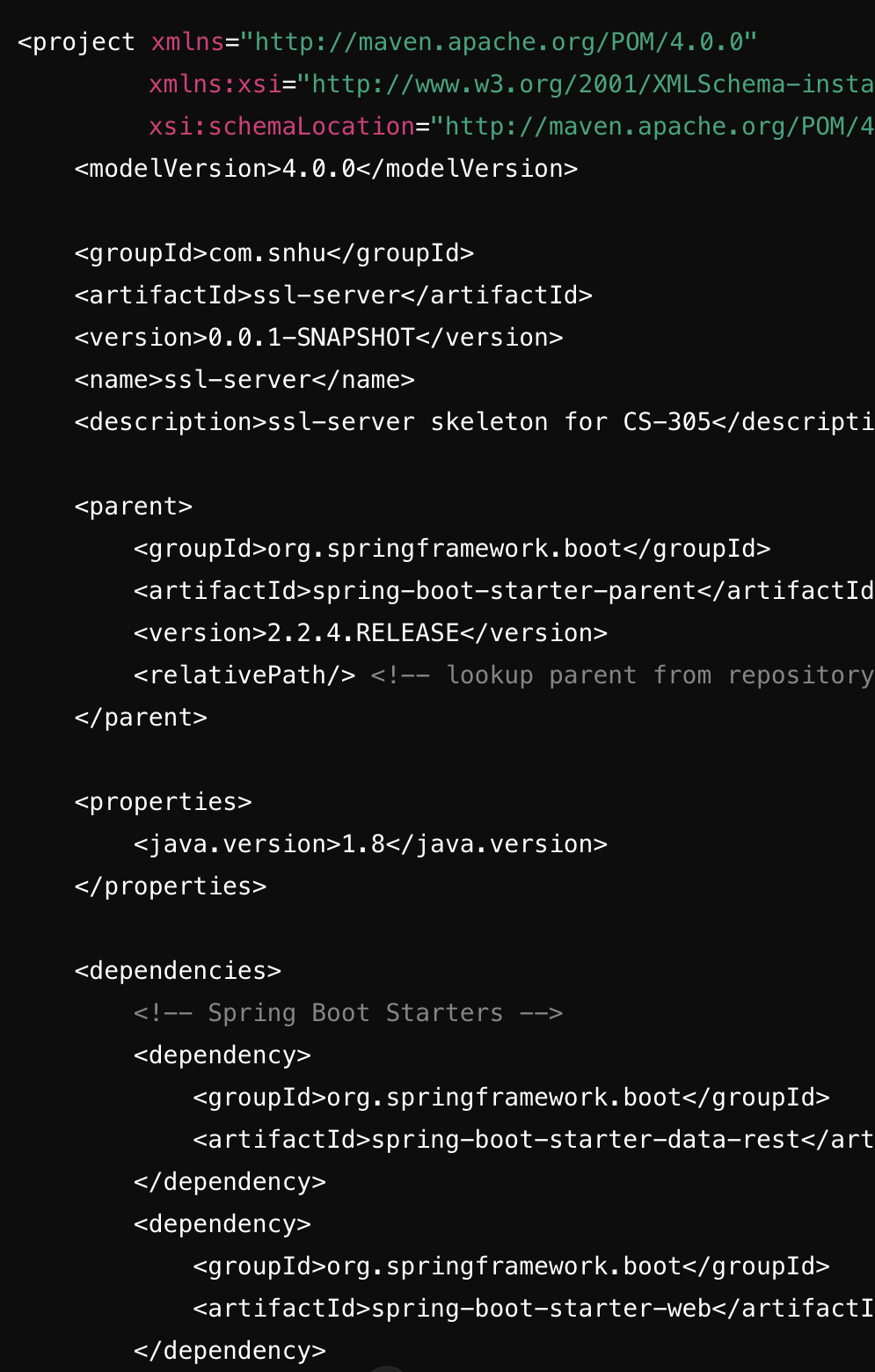


## Secondary Testing

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 refactored pom.xml file is cleaner, more maintainable, and better aligned with best practices. By incorporating the OWASP Dependency Check plugin and managing dependencies through the Spring Boot parent POM, the project adheres to security testing protocols, ensuring that vulnerabilities are identified and addressed early in the development lifecycle. This approach not only improves the security posture of the application but also facilitates ongoing maintenance and updates.

## Industry Standard Best Practices

By applying these industry-standard best practices, the refactored pom.xml file ensures a secure and maintainable project configuration. The use of trusted libraries, automated security checks, consistent environment configuration, clear separation of test and production dependencies, regular updates, and minimizing the attack surface collectively enhance the security posture of the application. These measures help mitigate known security vulnerabilities and align with best practices for secure coding.