****

# 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** | **8.24.2025** | **Nate Ryals** |  |

## Client



## Developer

Nate Ryals

## Algorithm Cipher

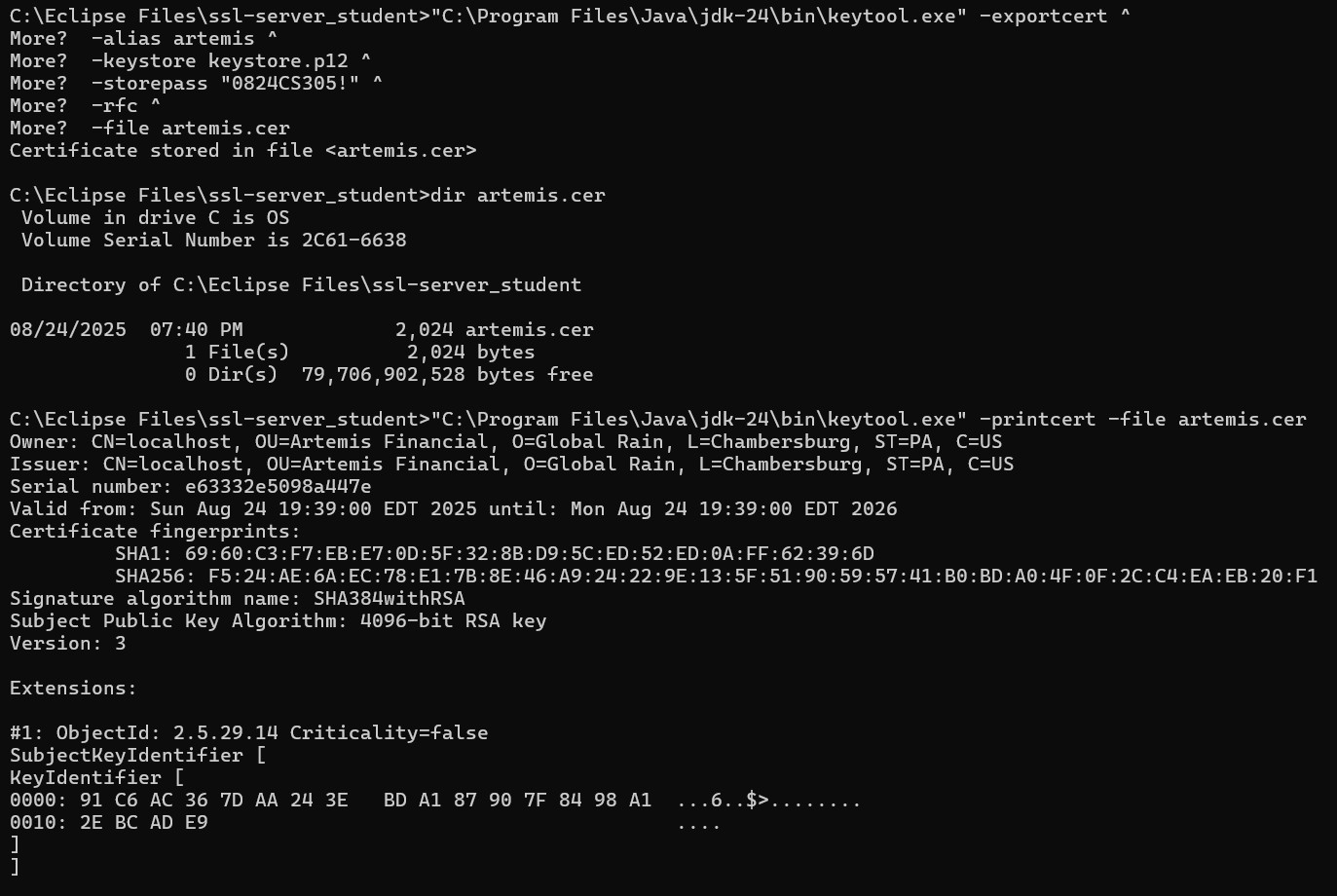
I recommend the Advanced Encryption Standard (AES) to secure the data, with the SHA-256 hash algorithm for data verification. AES is a symmetric block cipher that uses the same key for both encryption and decryption. It is the U.S. government’s standard and is a favorite among financial institutions, military organizations, and corporations. AES can utilize 128, 192, or 256-bit keys, but the 256-bit key size is the strongest in comparison and incredibly useful against brute-force attacks.

For integrity verification, the system will use SHA-256, which is a cryptographic hash function that creates a unique 256-bit hash from input data. This ensures that even small adjustments to the input creates a different hash value. SHA-256 is collision resistant and is considered a very secure method for protecting financial data.

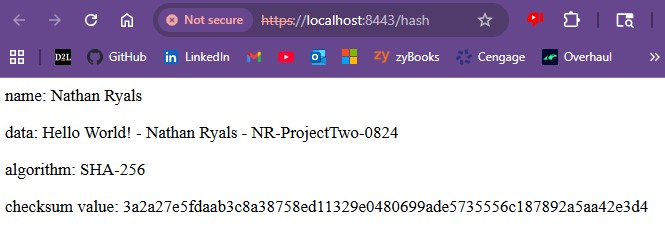
AES is a symmetric cipher and uses random numbers for initialization vectors and key generation. AES uses a single key for sharing and storing. It is faster and more efficient for encrypting large amounts of data. On the other hand, asymmetric encryption (such as RSA) is much more secure for key exchanges and digital signatures, but it is slower and impractical when working with large data systems. A secure channel is unnecessary because each user has a unique key pair. The private key used for decryption isn’t shared with anyone.

AES was adopted in 2001 and replaced older standards like DES, which became insecure due to its shorter key lengths. AES has become the industry standard and is approved under FIPS 140-2, HIPAA, GLBA, and GDPR compliance requirements. AES-256 when combined with SHA-256 (for checksums) is the best practice for financial institutions.

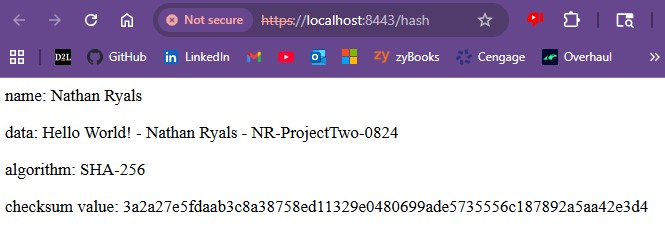
## Certificate Generation



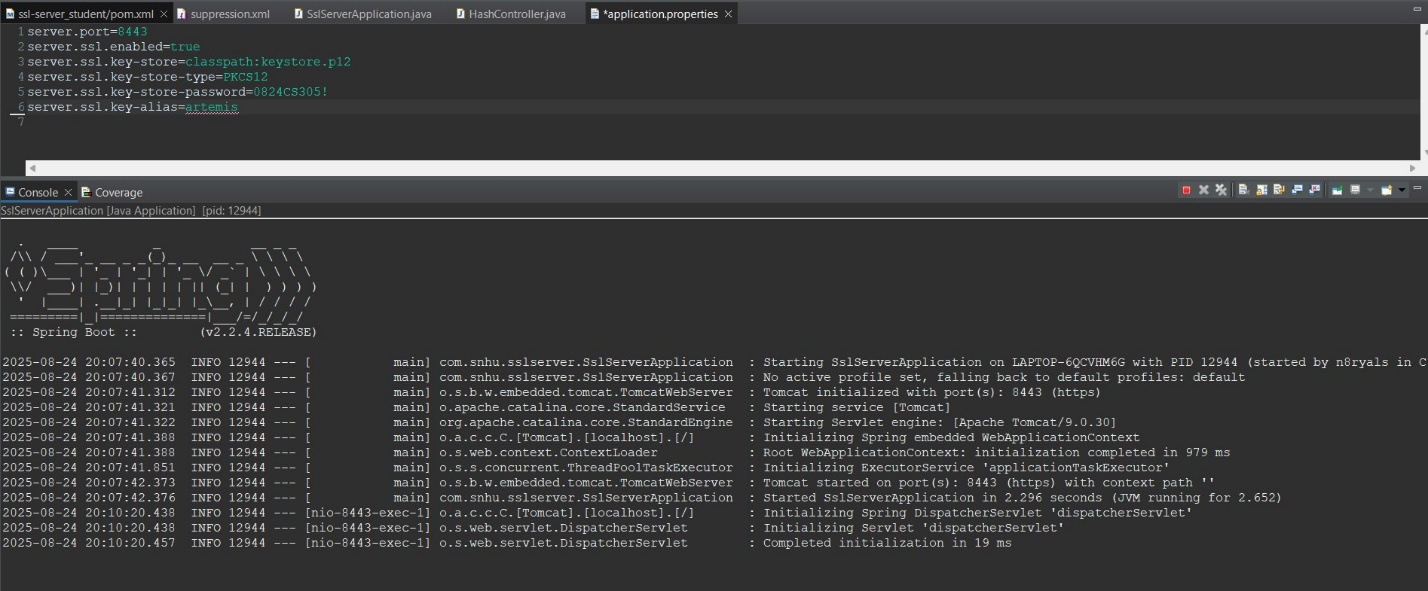
## Deploy Cipher

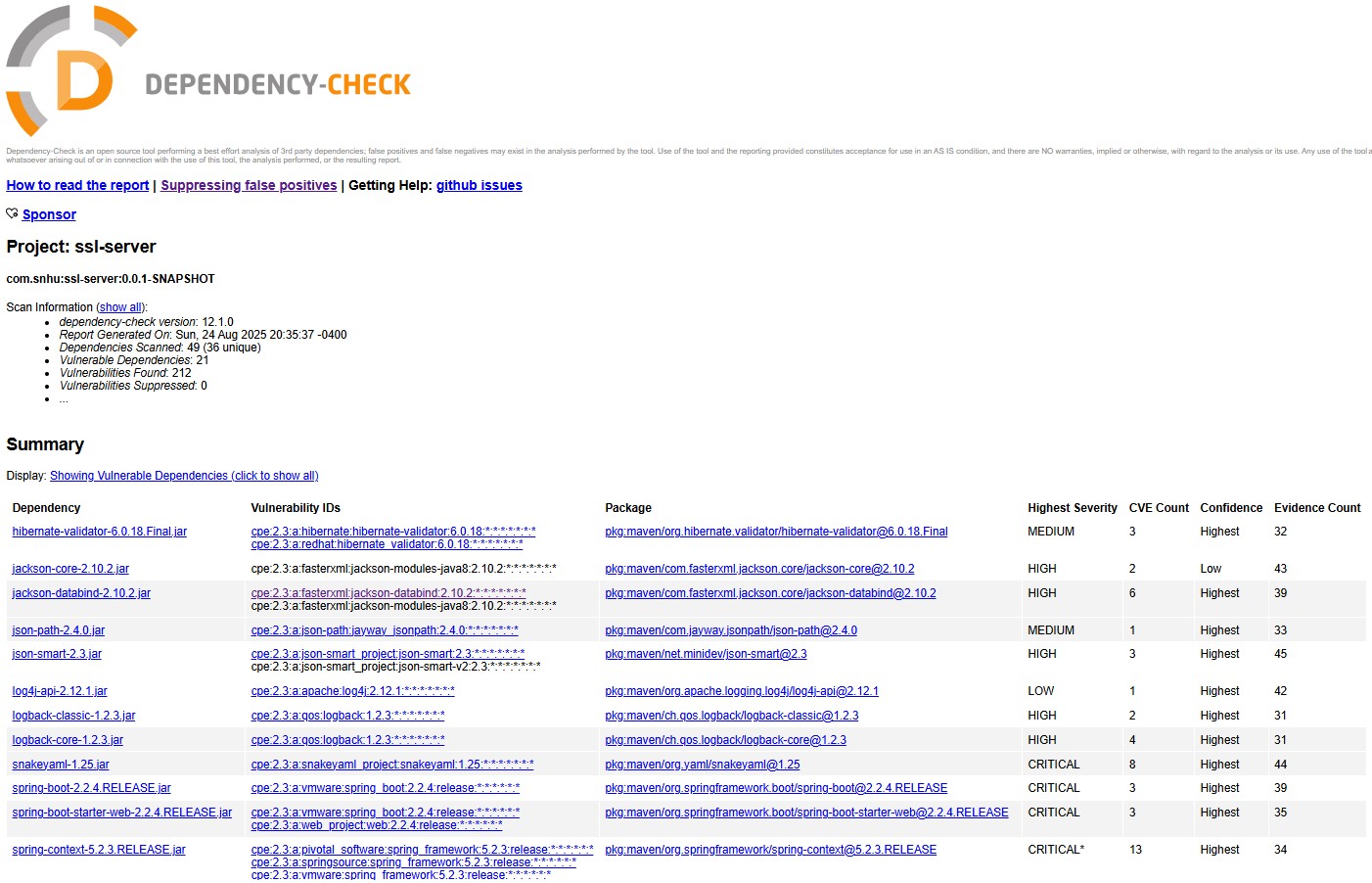


## Secure Communications



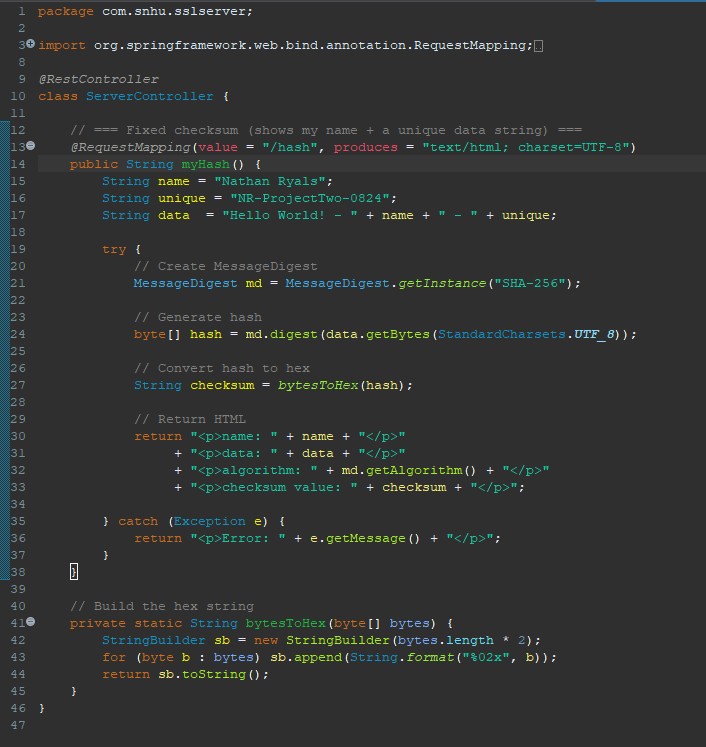
## Secondary Testing

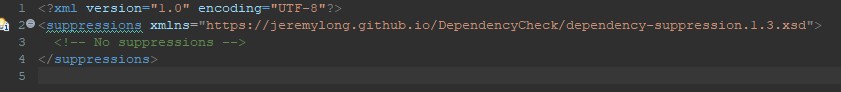


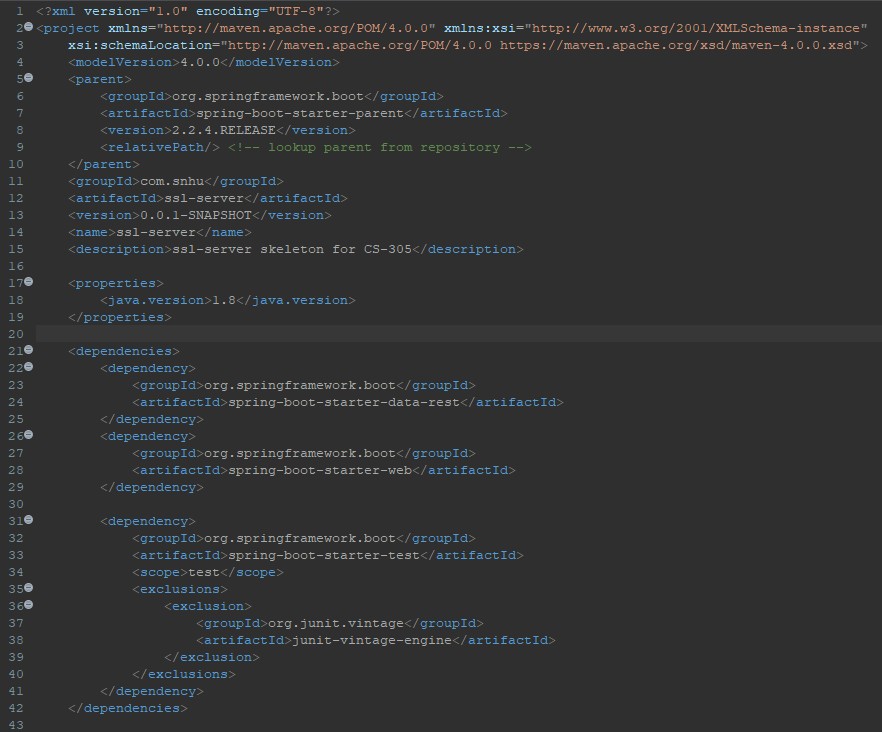


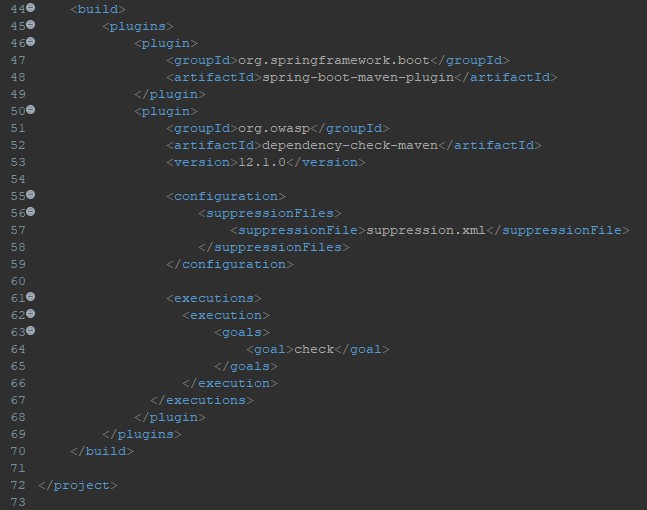


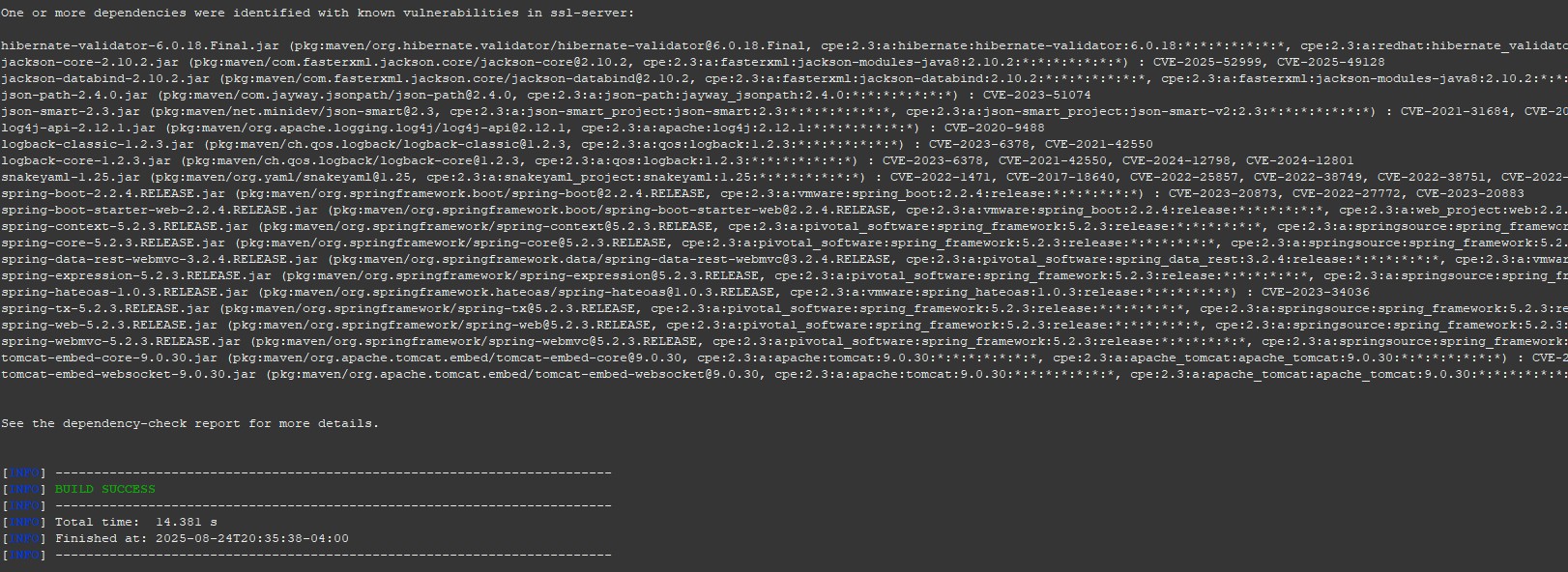
## Functional Testing











## Summary

I added a SHA-256 checksum to check that files were not changed during transfer. I also changed HTTP to HTTPS so the connection between the user and the server is encrypted now. Running Dependency-Check also gave me a way to make sure I did not create new vulnerabilities. I reviewed the Vulnerability Assessment Process Flow diagram and the areas I focused on were:

* **Cryptography**: using SHA-256 for hashing.
* **Client/Server**: switching to HTTPS with a keystore.
* **Error Handling**: adding try/catch blocks so errors don’t break the program or leak information.
* **Input Validation**: escaping user input so it can’t be taken advantage of to run malicious code.

## Industry Standard Best Practices

I followed common best practices for secure coding. I implemented SHA-256 because it is an industry approved algorithm. I set up HTTPS with a keystore so data is encrypted over network. I made sure errors do not show sensitive details. I escaped user input before displaying it in the browser.

I ran the OWASP Dependency-Check tool, to show any vulnerabilities, which existed due to some of the older libraries. Using best practices is important because it protects customer data, helps the company meet laws/regulations, builds trust with users, and makes the program more adaptable during maintenance.