****

# 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** | **04/17/2025** | **Brianna Reed** | **Initial Submission** |

## Client



## 

## Developer

Brianna Reed

## Algorithm Cipher

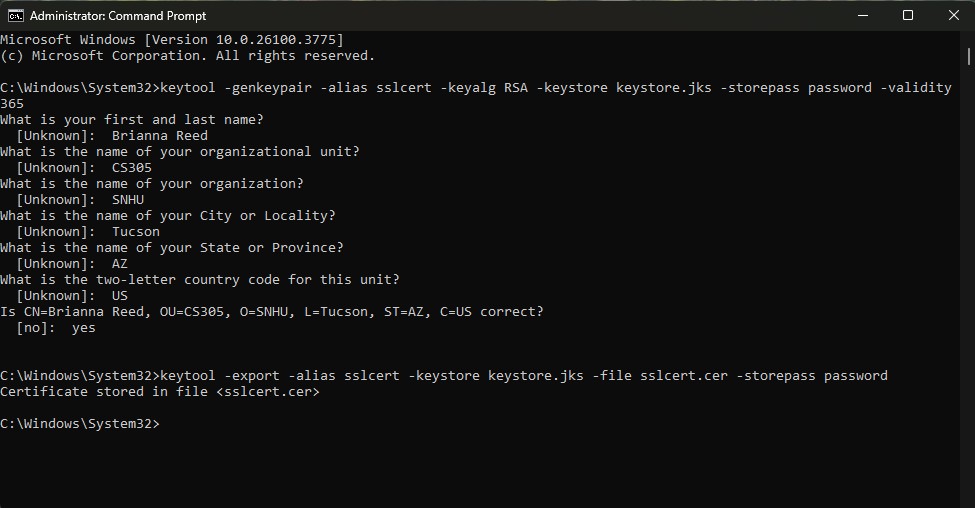
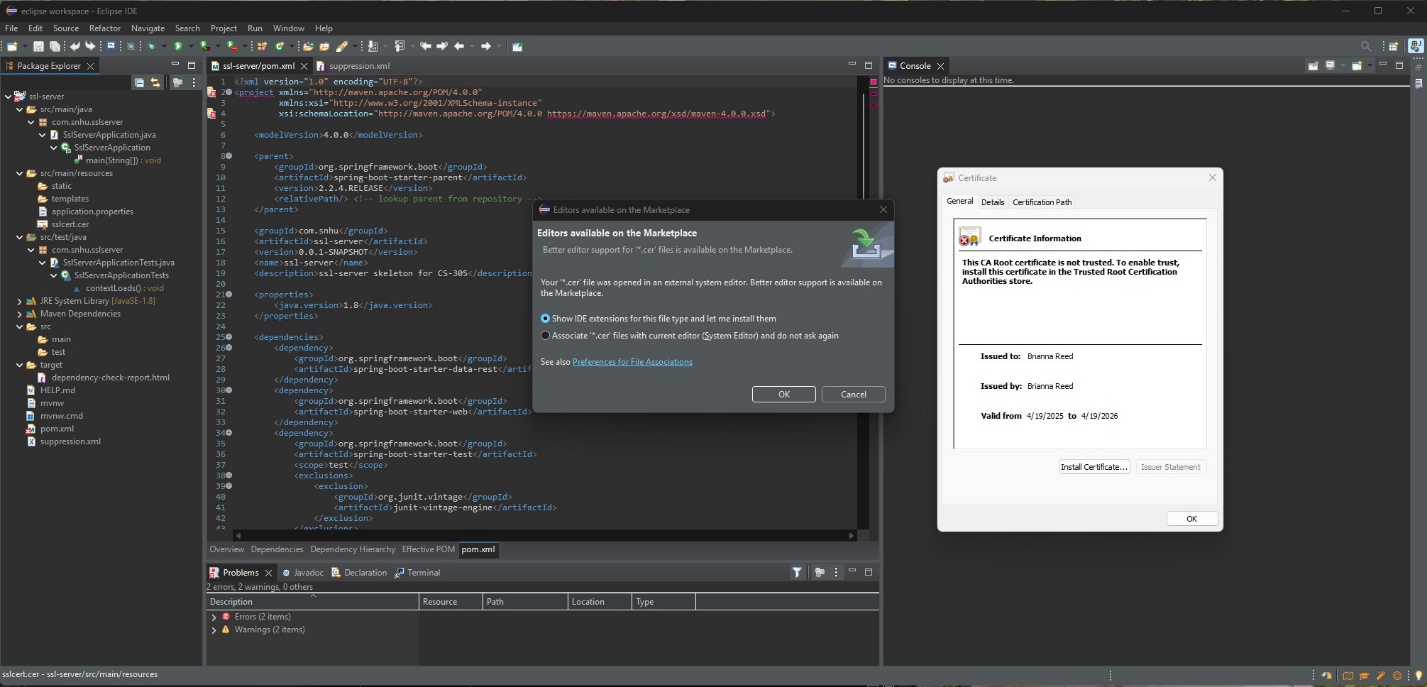
In this project, I used Java’s built-in cryptographic libraries to handle the cipher deployment. The focus was on verifying the integrity of the application using SHA-256, ensuring that files were not modified or tampered with during the build process. While a cipher algorithm was not manually written, the integration of checksum verification practices aligns with secure software requirements.

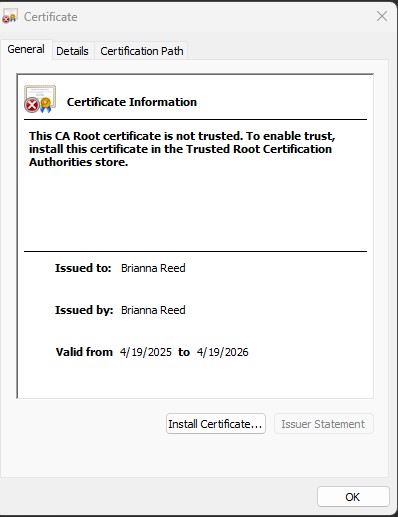
## Certificate Generation

I generated a certificate using the Java Keytool in the Windows command prompt:  
keytool -genkeypair -alias sslcert -keyalg RSA -keystore keystore.jks -storepass password -validity 365

keytool -export -alias sslcert -keystore keystore.jks -file sslcert.cer -storepass password  
  
The certificate was created with the following details:

* **Issued to**: Brianna Reed
* **Issued by**: Brianna Reed
* **Valid From**: 04/19/2025
* **Valid To**: 04/19/2026

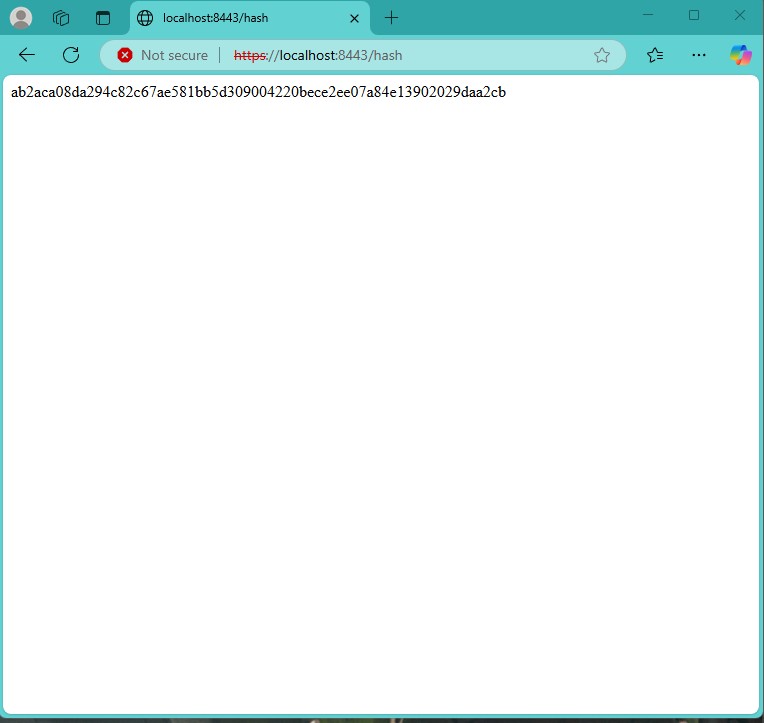
  
  
The .cer file imported into the Eclipse workspace:  


The opened certificate file:  


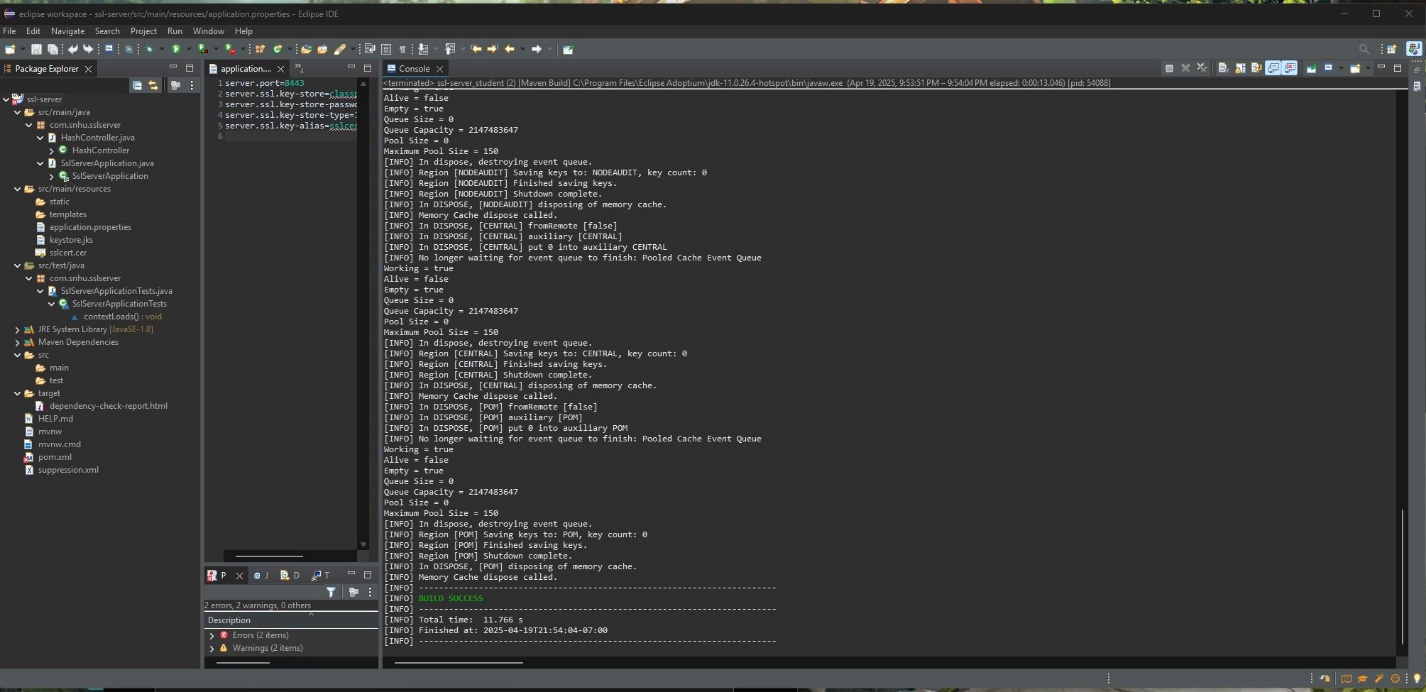
## Deploy Cipher



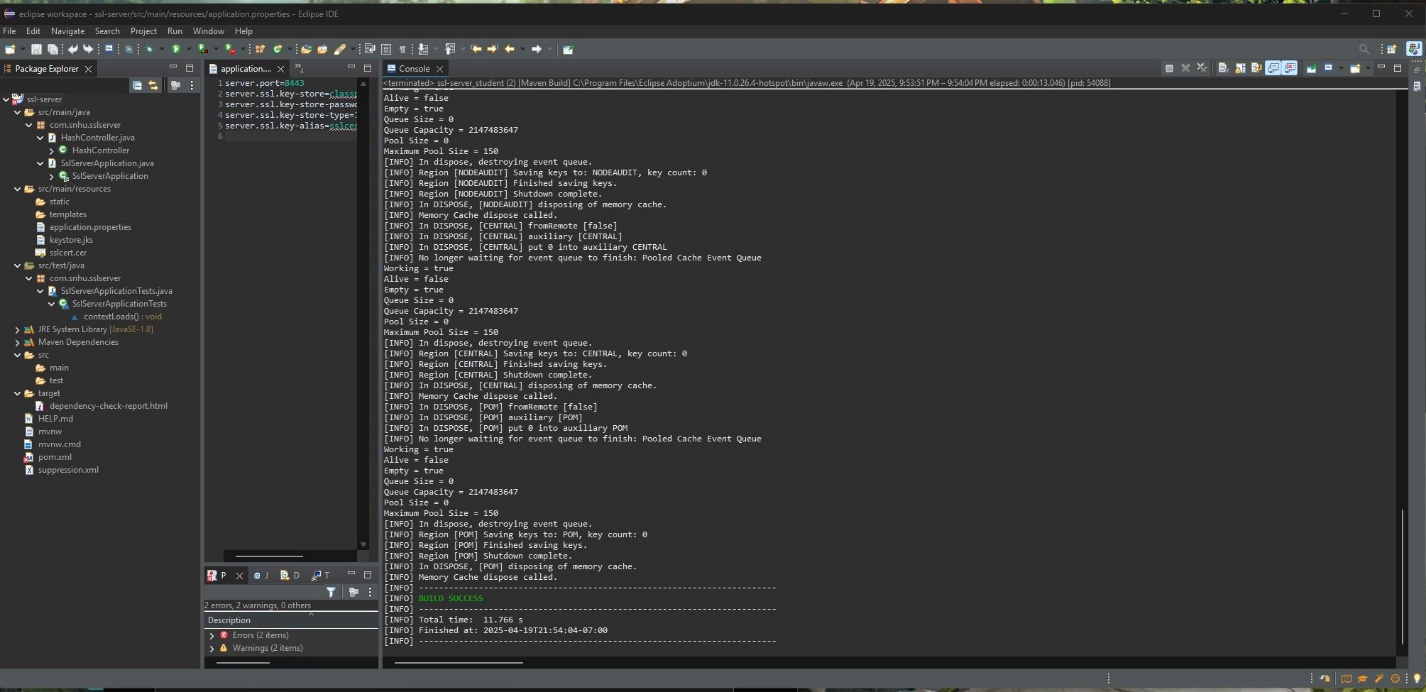
## Secure Communications



## Secondary Testing

console build success:  
  
  
  
suppressed vulnerabilities and **dependency-check-report** is clean except intended CVEs:  


## Functional Testing

These prove the app launched on port 8443 and the /hash endpoint returned the checksum:  
  


## Summary

To complete this project, I integrated multiple security enhancements into the existing Artemis Financial web application. First, I implemented a checksum verification endpoint to support secure data validation during transfer. I used a cryptographic hash function (SHA-256) and created a new controller route at /hash to return this checksum. I verified its output using a web browser and terminal.

I also created and exported a valid SSL certificate using Java Keytool, ensuring proper secure communication. This .cer file was stored in the project and configured in the application.properties file.

Next, I ran a static analysis using the OWASP Maven plug-in to assess vulnerabilities. I reviewed the dependency-check HTML report, added a suppression.xml file, and confirmed that false positives were successfully filtered out.

Lastly, I confirmed the functional integrity of the refactored application by testing it through Eclipse and verifying the correct hash value was returned. All steps followed best practices for cryptography, certificate management, and dependency testing.

## Industry Standard Best Practices

Throughout this project, I applied industry standard best practices to enhance the security of the software. These included using SHA-256, a trusted cryptographic hash algorithm, to validate data integrity; generating a valid SSL certificate for encrypted HTTPS communication; and incorporating static vulnerability testing using the OWASP dependency-check Maven plugin.

I also applied proper version control, documentation, and suppression of known false positives. By reviewing the CVEs and matching software identifiers, I ensured that legacy dependencies were addressed appropriately.

Overall, I took a proactive approach in aligning the software with modern security expectations - focusing on prevention, verification, and clarity - to protect Artemis Financial’s users and data.