

# 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** | **4/19/2024** | **Jeremiah Benjamin** |  |

## 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

Jeremiah Benjamin

## Algorithm Cipher

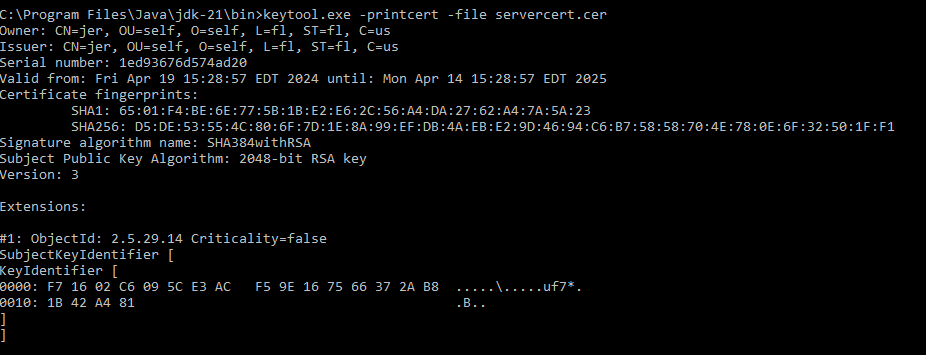
RSA is a public-key cryptosystem widely used for secure data transmission, especially suitable for environments like Artemis Financial, where confidentiality and authentication is essential. This encryption algorithm uses a pair of keys: a public key for encryption and a private key for decryption, enabling secure communication over open networks without the need for exchanging keys privately.

RSA does not inherently incorporate a hash function but often pairs with cryptographic hash functions to ensure message integrity through digital signatures. Common RSA key lengths are 2048 or 3072 bits, balancing robust security against brute force attacks with reasonable processing requirements.

The security of RSA is heavily reliant on the generation of large random prime numbers during the key creation process. While RSA remains popular due to its proven reliability and extensive implementation, it is susceptible to the future development of quantum computing, which could potentially break RSA encryption by efficiently factoring large numbers. Despite this, RSA's long-standing track record and theoretical backing make it a dependable choice for securing sensitive financial data, maintaining integrity and confidentiality as paramount concerns.

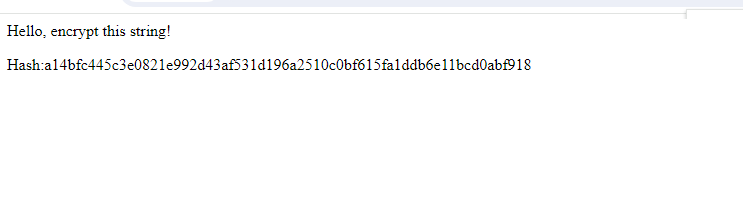
## Certificate Generation

Insert a screenshot below of the CER file.



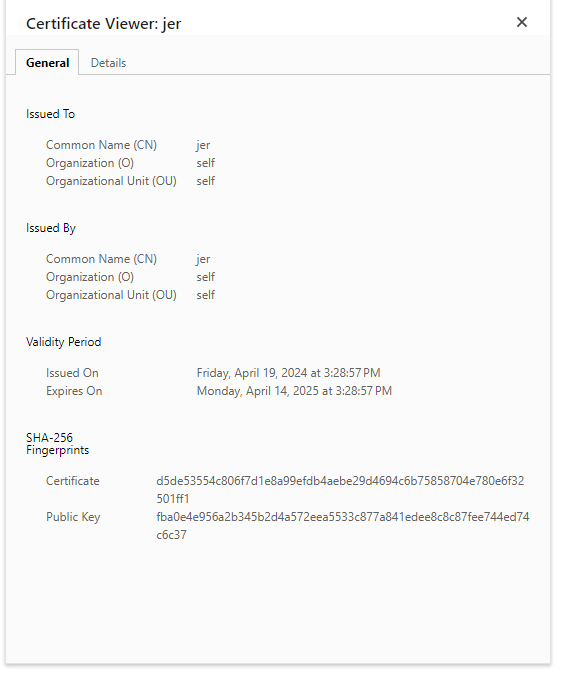
## Deploy Cipher

Insert a screenshot below of the checksum verification.



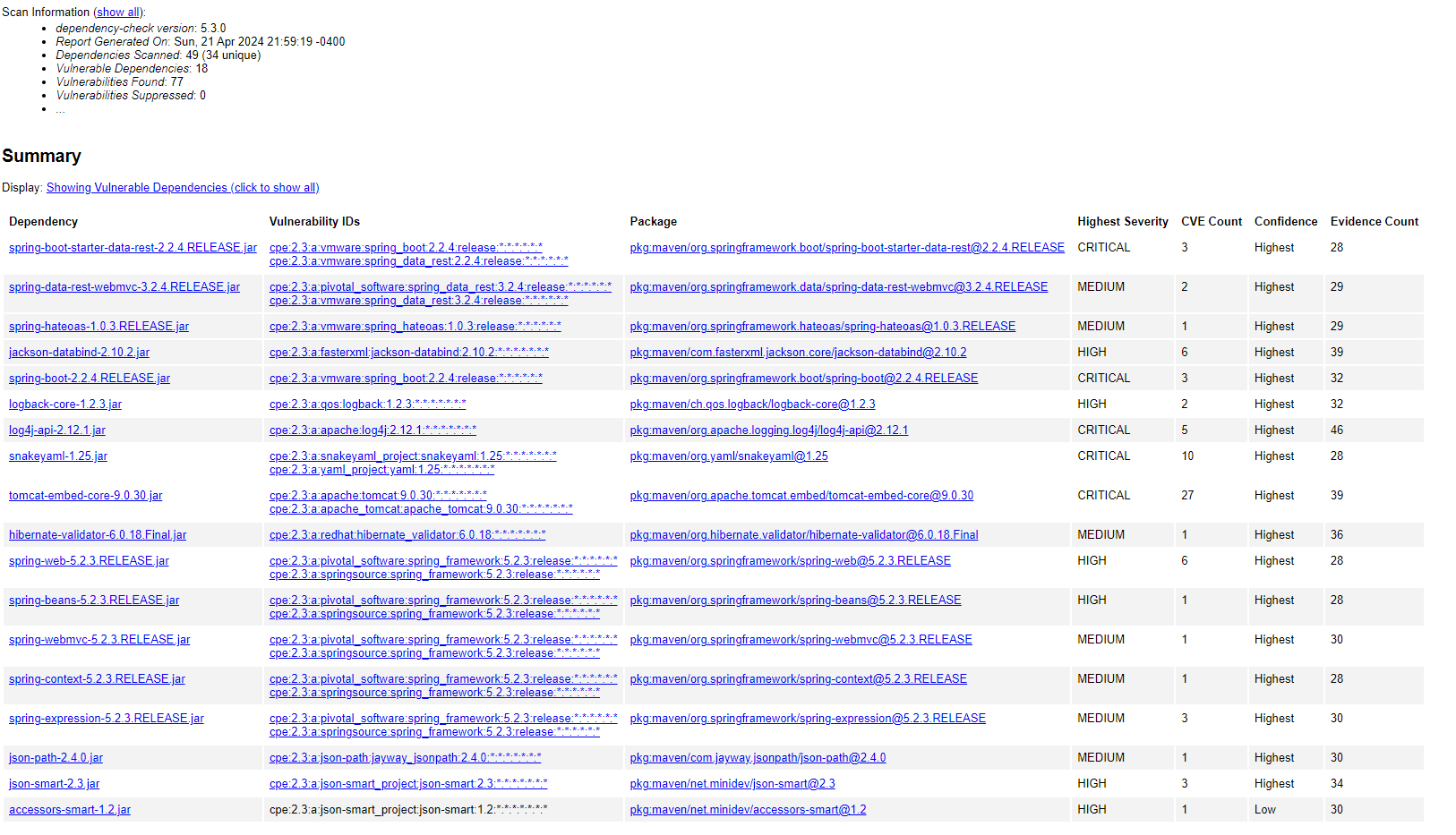
## Secure Communications

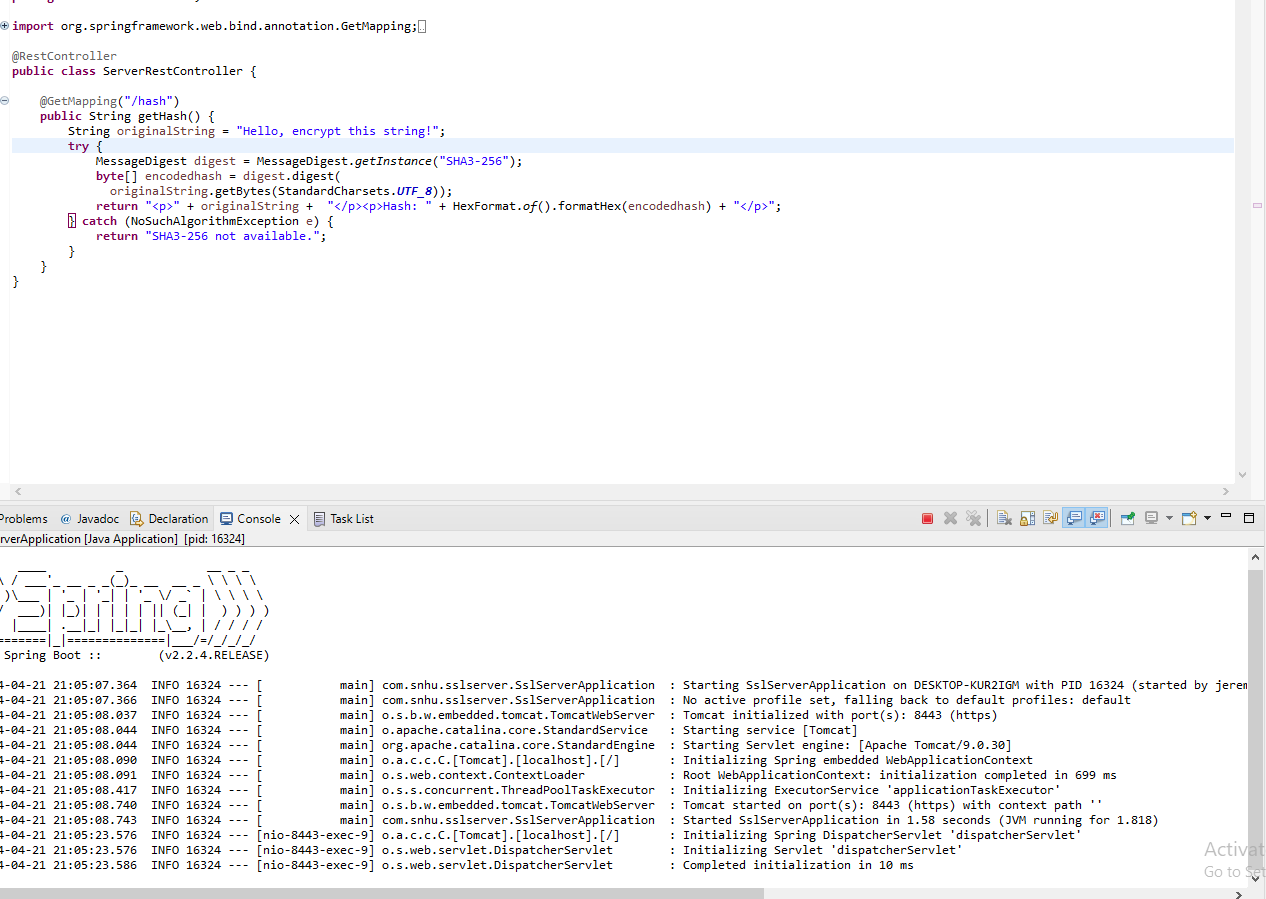
Insert a screenshot below of the web browser that shows a secure web page.



## Secondary Testing

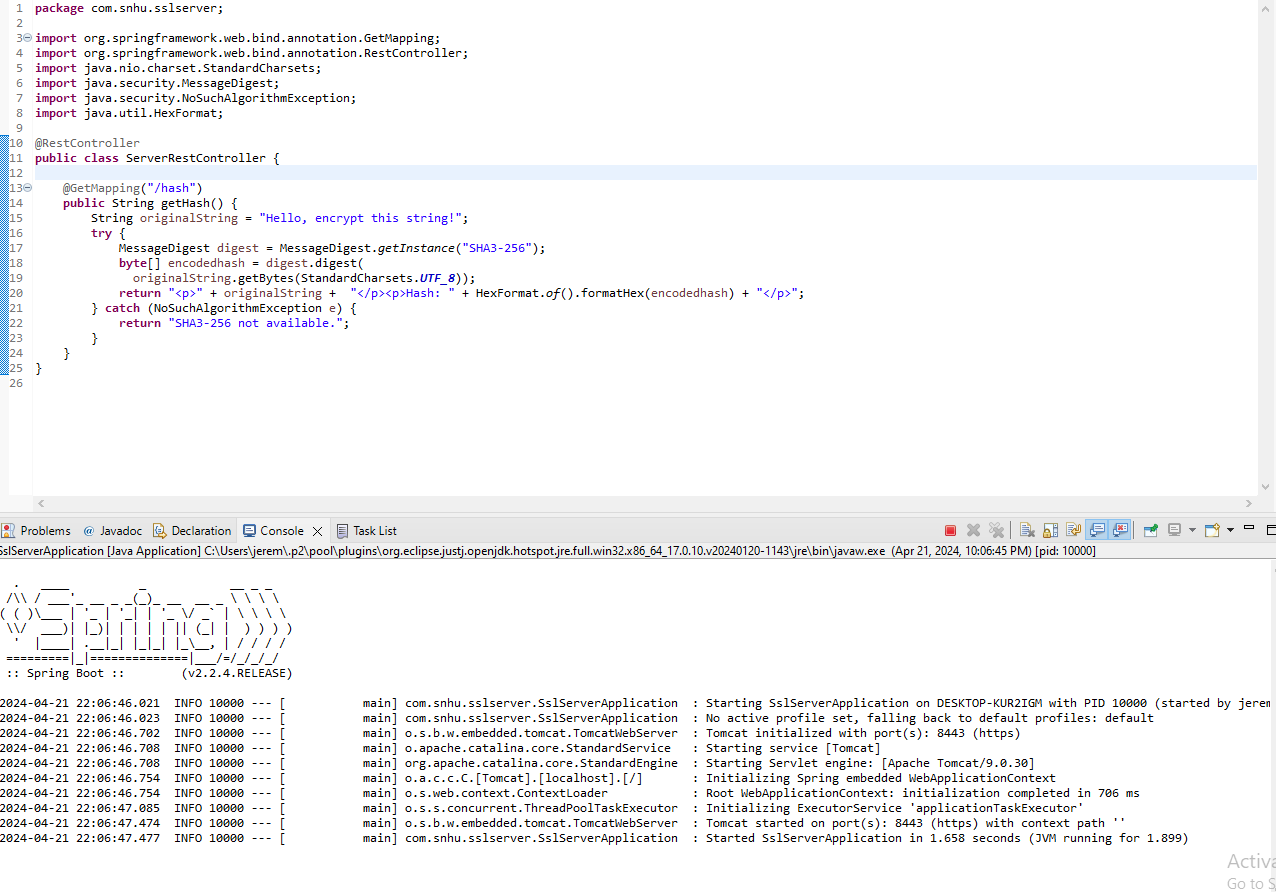
Insert screenshots below of the refactored code executed without errors and the dependency-check report.





## Functional Testing

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



## Summary

The code was refactored to include a keystore, enhancing the site's security before performing any hashing operations. By integrating this, all content transmitted to the web server is encrypted. Although the RSA algorithm is slower than AES, it significantly boosts security, approaching the highest possible level. Regarding the vulnerability process flowchart, critical areas include client/server interaction, cryptography, code quality, and the code review process. This involves using OWASP tools for dependency reporting and functional testing. The security enhancement began with the generation of a self-signed certificate and the implementation of a REST controller to establish a route to the /hash endpoint. Subsequently, a dependency report added a second layer of security by verifying the absence of new vulnerabilities.

## Industry Standard Best Practices

To uphold the security of the software application, industry-standard best practices were meticulously employed. By embedding a keystore, the system was fortified to encrypt all transmissions to and from the web server. This implementation is crucial for safeguarding against unauthorized data breaches and maintaining data integrity. Opting for the RSA algorithm, known for its strong security assurances, aligns with these best practices, where prioritizing security in the financial industry is a must.

The adoption of these secure coding practices brings significant benefits to the company's overall health. It enhances the organization's reputation by demonstrating a proactive approach to protecting customer information, which is paramount in the financial sector where trust is a key component of customer relationships. Moreover, these practices help in mitigating risks associated with security vulnerabilities, thereby avoiding potential financial losses and legal repercussions that could arise from data breaches. This strategic focus on security not only helps in maintaining regulatory compliance but also positions the company as a reliable entity in its industry, fostering growth and sustaining long-term success.