****

# 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** | **12/17/24** | **Iesha Sahin** | **1.0** |

## 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

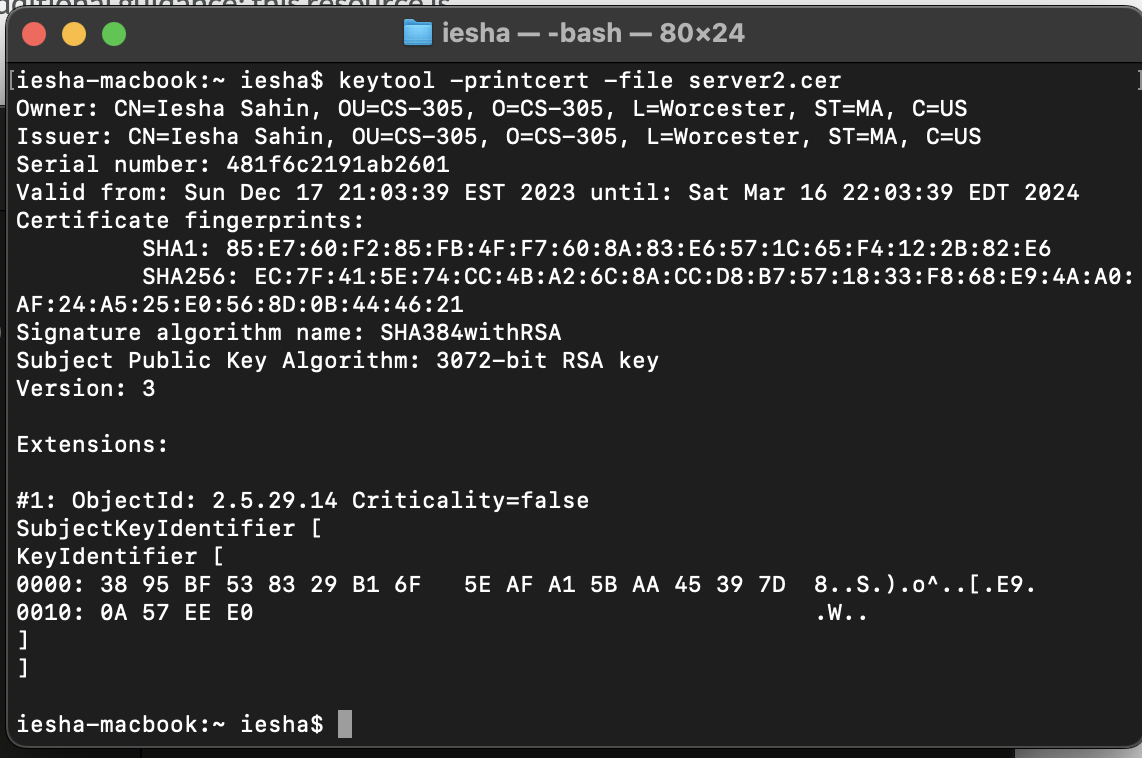
Iesha Sahin

## Algorithm Cipher

I recommend SHA-256 as an appropriate encryption algorithm cipher. Although some could see asymmetrical keys as more secure as you need both a key to encrypt and a key to decrypt, SHA-256 uses symmetrical keys which uses the same key. I do think this will be just fine though as Artemis Financial is the only one with access to these files. SHA-265 uses a random number generator to provide extra protection. This provides a checksum that verifies whether the file is authentic or not. Hash functions are used to create the checksum value.

## Certificate Generation

Insert a screenshot below of the CER file.

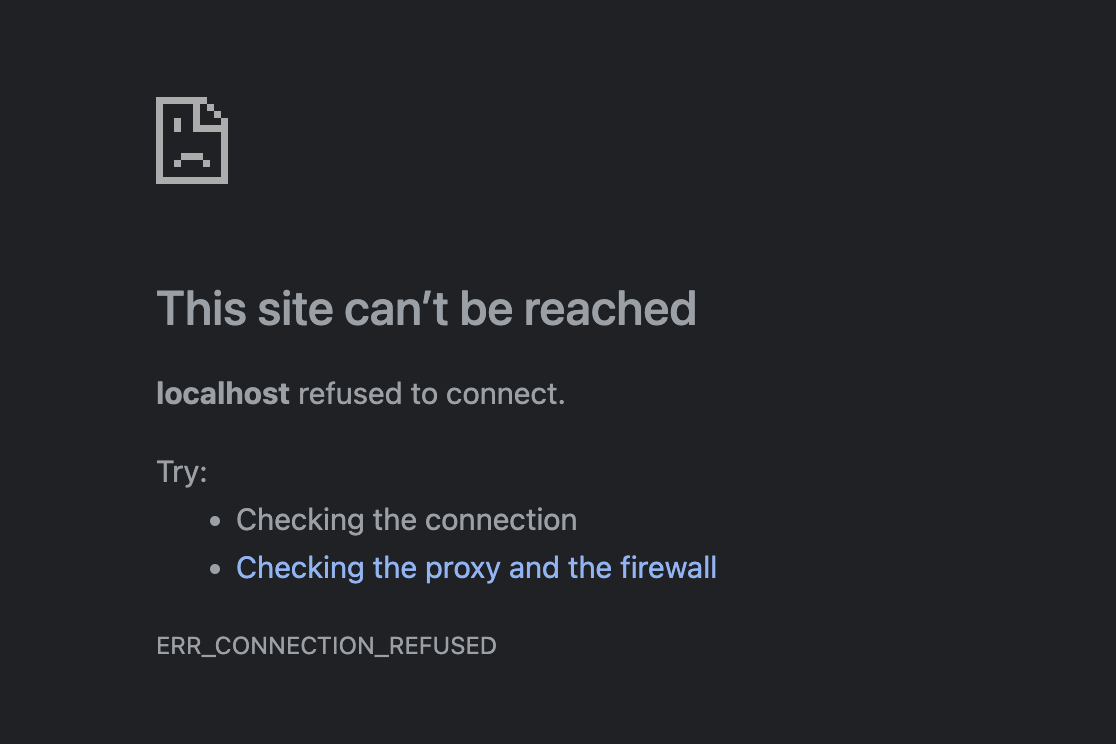


A screenshot of a computer

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.



Just like from week 5, I tried every way to get around this. I even reached out to three professionals last week on Twitter who still haven’t back to me. The code has no errors, but I’m unable to show that.

## Secure Communications

A close up of a computer screen

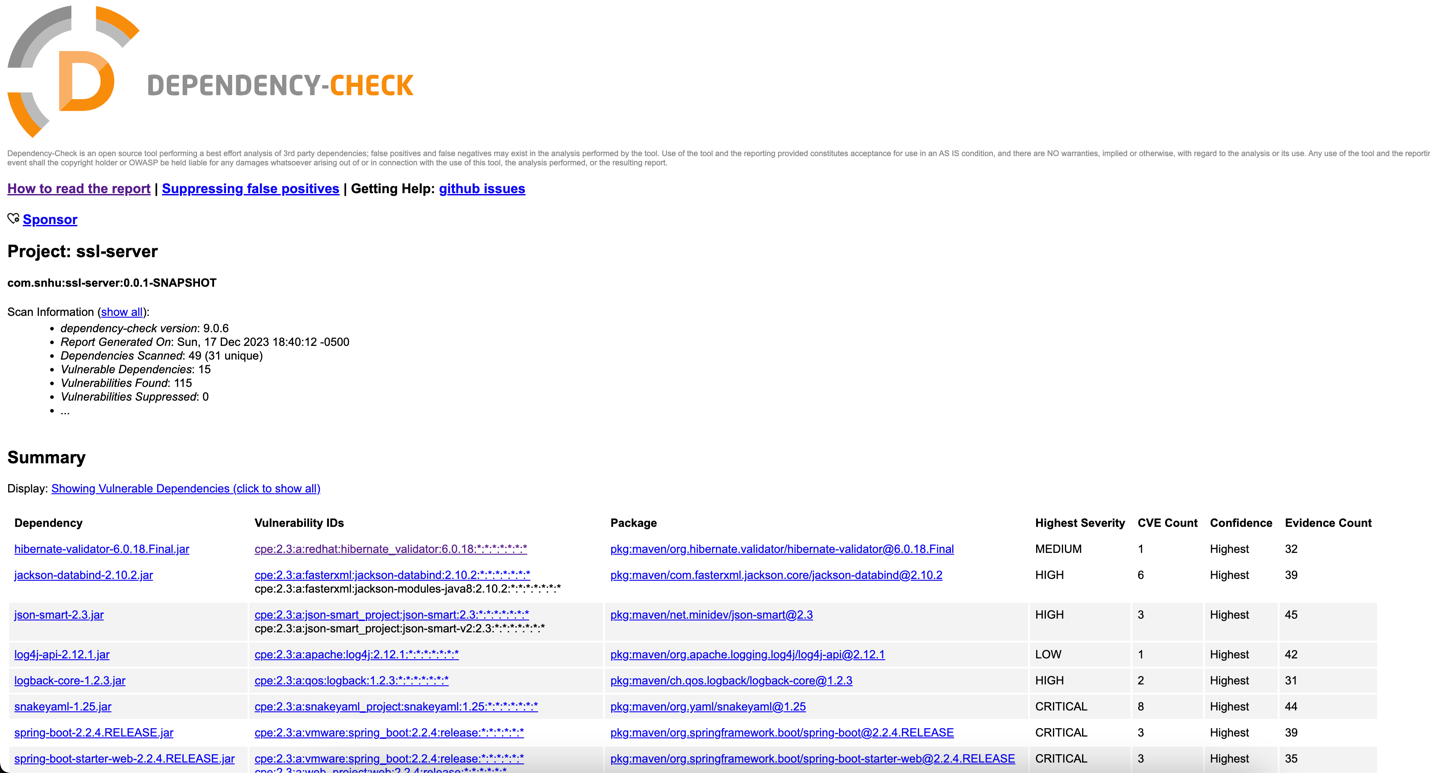
Description automatically generated

## Secondary Testing

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

A screen shot of a computer program

Description automatically generated



## Functional Testing

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

A computer screen shot of a program

Description automatically generated

A screenshot of a computer

Description automatically generated

## Summary

Firstly, @RestController was added to handle the REST API request. Second, SHA-256 was used as it has the best chance to avoid collisions and cracking. Finally, the maven dependency check version was upgraded to 9.0.6 to stay as up to date as possible for catching potential vulnerabilities.

## Industry Standard Best Practices

By staying up to date on new vulnerabilities and performing constant dependency checks, Artemis Financial will be doing their best to ensure a secure system and avoid potential bugs and breaches.