****

# 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** | **04/15/2023** | **Madelena Ngo** |  |

## 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

Madelena Ngo

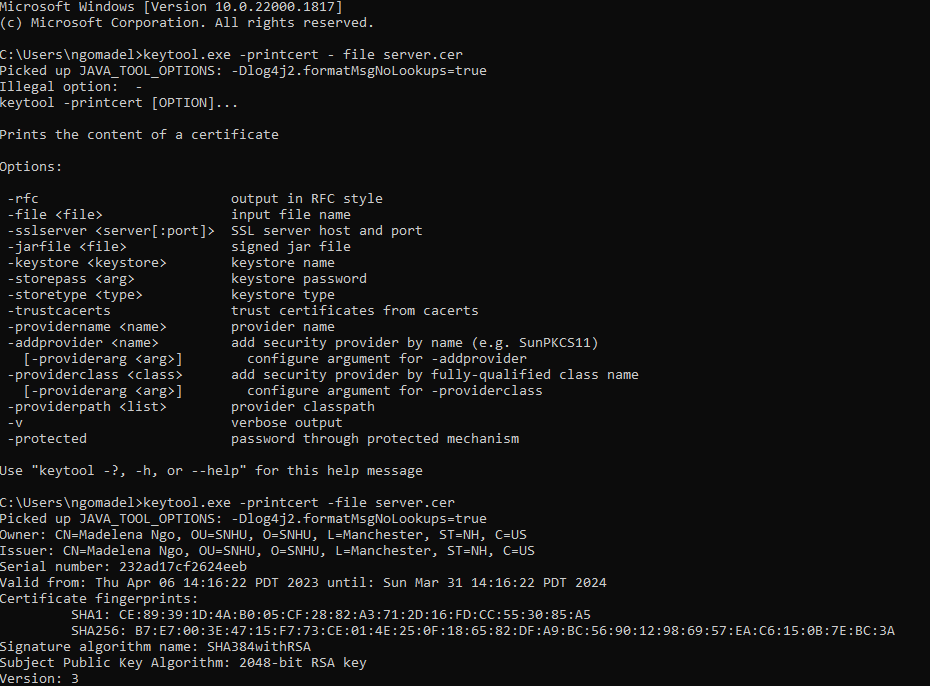
## Algorithm Cipher

Artemis Financial would like for us at Global Rain to assist in modernizing their applications by using the most current and effective security software applications. Since Artemis has a public web interface, they are seeking our services to secure their most valuable assets, client data and financial information. Artemis Financial would like our assistance in ensuring secure communication by affind a file verification step to their current web applications. We must take this into consideration by adding secure communication mechanisms to meet their software security requirements with adherence to security policies and quality controls.

SHA-256 is one of the many secure ciphers and has the lowest risk to no risk for collisions. The cipher provides secure hash functions when a secure hash function is required to protect sensitive and unclassified information. The algorithms are secure because it is infeasible. Any changes to a message will result in a different output, rendering it secure and are approved by the US government. To avoid collisions means SHA-256 will output a different set of characters with a high probability of computation and differing sets of outputs. Because of the high probability of different outputs, no two will be the same, ensuring its security and no collision possibility.

## Certificate Generation

Insert a screenshot below of the CER file.



## Deploy Cipher

Insert a screenshot below of the checksum verification.

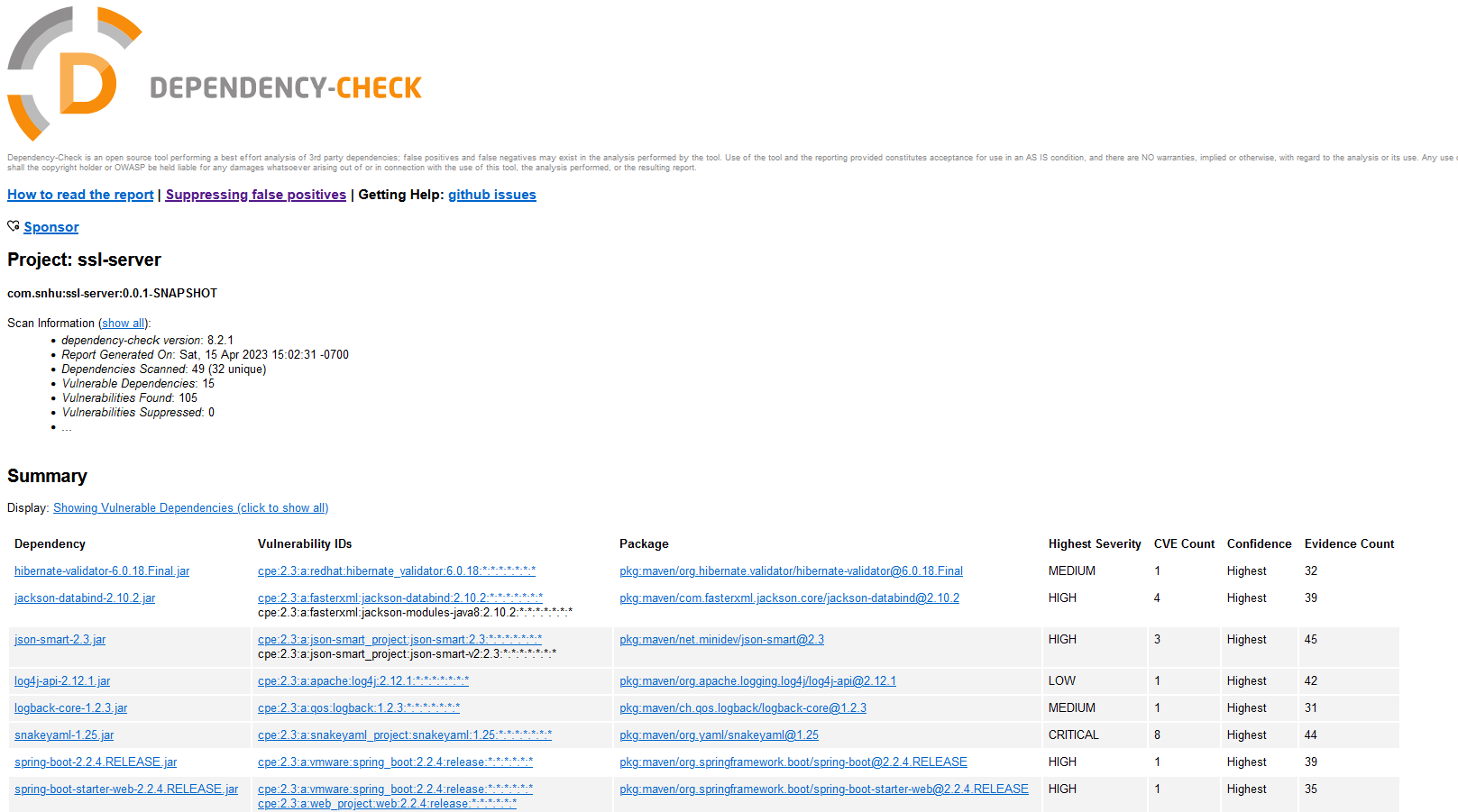
## Secure Communications

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



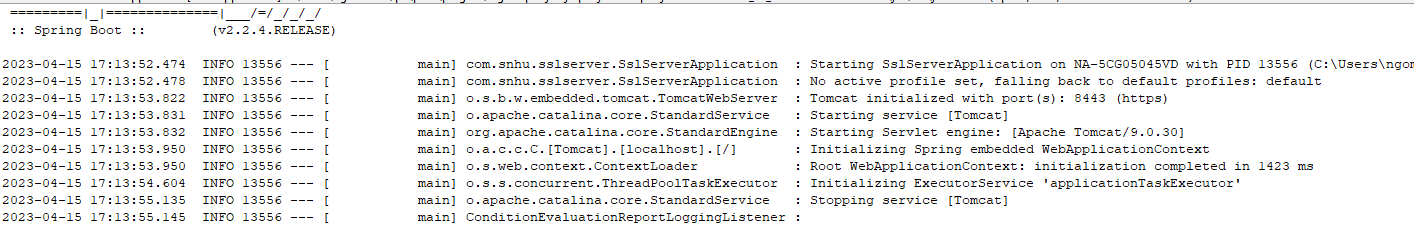
## Secondary Testing

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

I chose SHA-256 algorithm in this work as it is the most secure and reliable algorithm cipher. Refactoring all the code allows us to look for and view such vulnerabilities within the web application. Fixing these issues and refactoring the code allows for a new level of security within Artemis Financial’s web applications.

## Industry Standard Best Practices

Industry standard best practices help us in retaining and maintaining the software application’s current security by allowing for updates to occur. When updates become available, it is best to update when possible to avoid potential breaches in the older version.