****

# 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** | **04/16/2023** | **Beth Campbell** | **Updated Developer, Algorithm Cipher, Certificate Generation, Deploy Cipher, Secure Communications, Secondary Testing, Functional Testing and Summary** |

## 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

Beth Campbell

## Algorithm Cipher

Artemis Financial has requested a recommendation for an encryption algorithm that is suitable for securing and storing long-term files, considering the potential threat of unauthorized access by hackers. As the files will not be moved to another location, asymmetric keys are unnecessary. The SHA-256 algorithm, which employs 256-bit keys for encryption, would be the most efficient and effective choice for this scenario. Exclusive access to the encryption keys will be granted only to Artemis Financial, ensuring complete control over the encryption process. In addition, the utilization of Java's random number generator in conjunction with SHA-256 enhances the security of the encrypted files by enabling the generation of a secure and irreversible checksum for verification and authentication purposes. The hash function, when employed alongside the SHA-256 cipher, produces a checksum signature that ensures reliable file verification.

## Certificate Generation

Insert a screenshot below of the CER file.

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

Text

Description automatically generated

## Secure Communications

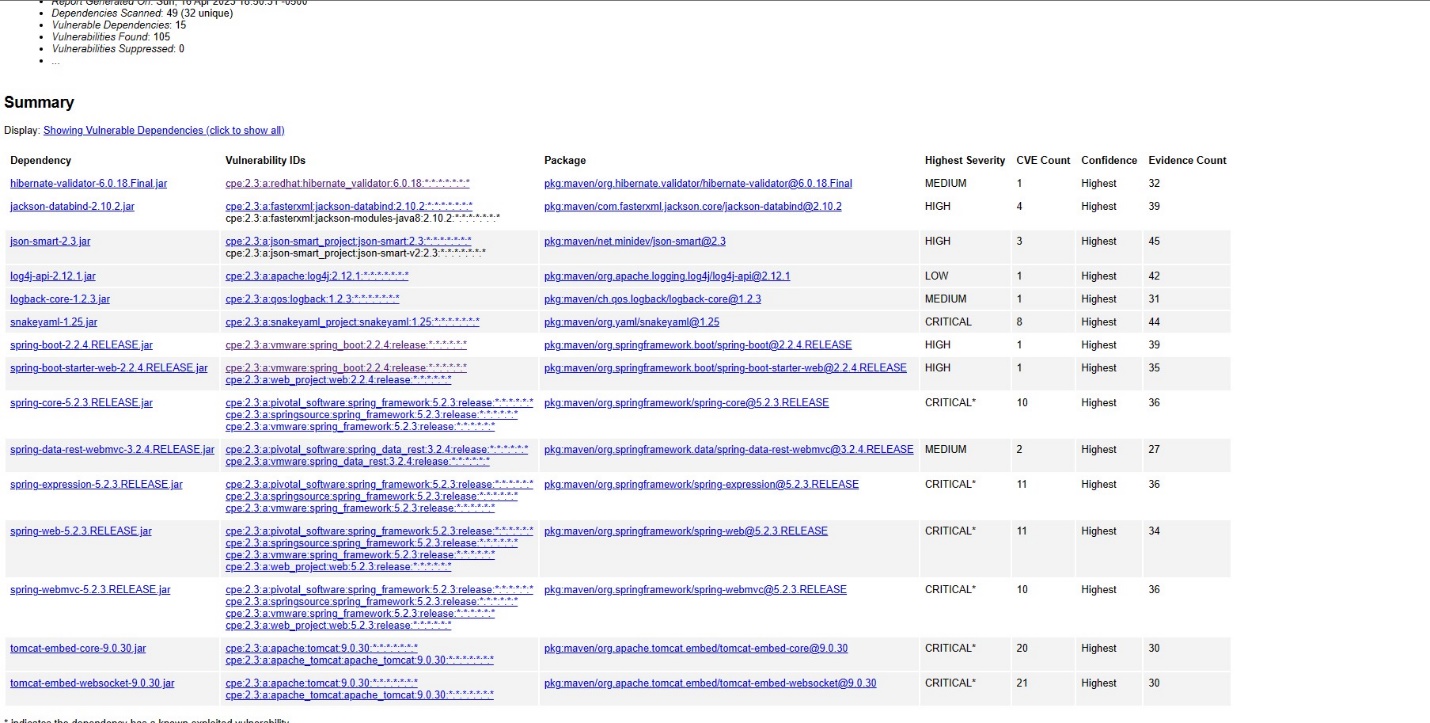
Insert a screenshot below of the web browser that shows a secure webpage.

[Insert screenshots here.]

## Secondary Testing

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

Text

Description automatically generated

## Functional Testing

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

Text

Description automatically generated

## Summary

By refactoring the code, I addressed input validation, secure API interacctions, cryptography, and encapsulation. The code can be run through HTTPS. RestController and ServerController validate the data and encapsulate it through SHA-256.

## Industry Standard Best Practices

In my code, I have incorporated a RestController with robust security measures to effectively handle RESTful operations. The ServerController class has been purposefully designed to rectify the vulnerabilities highlighted in the vulnerability assessment diagram. To bolster security, I have opted for the SHA-256 hashing algorithm, renowned for its formidable security features and negligible collision risks. In addition, weekly security updates will be performed in order to ensure the application remains secure.

Applying industry standard best practices for secure coding brings significant value to a company's overall wellbeing. It enhances security by identifying and mitigating vulnerabilities, ensures compliance with regulations and standards, fosters customer trust, saves costs, improves operational efficiency, protects intellectual property, and minimizes business disruptions. It is a proactive approach that safeguards a company's assets, reputation, and long-term success in today's threat landscape.