****

# 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** | **04/12/2023** | **Daniel Tarulis** |  |
| **1.1** | **4/23/2023** | **Daniel Tarulis** |  |

## Client



## Instructions

Submit these 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

Daniel Tarulis

## Algorithm Cipher

Determine an appropriate encryption algorithm cipher to deploy given the security vulnerabilities, justifying your reasoning. Be sure to address the following:

* Provide a brief, high-level overview of the encryption algorithm cipher.
* Discuss the hash functions and bit levels of the cipher.
* Explain the use of random numbers, symmetric vs non-symmetric keys, and so on.
* Describe the history and current state of encryption algorithms.

Artemis Financial wants to modernize their operations by using the most current and effective security software in order to protect their long-term client data and financial information. Specifically, Artemis Financial would like to increase the security of their web application by verifying data transfers in the form of a checksum. The most likely attack method for this sensitive data will be attackers accessing client files in one way or another. Utilizing a proper cipher, the data will be unreadable and useless even if it were to be obtained without the key. I recommend using the SHA-256 cipher algorithm. This algorithm uses 256-bit keys to encrypt sensitive files, which is the highest level of bitwise encryption. More bits mean a longer key, making it extremely difficult to brute force and less likely to have collisions. The SHA-256 cipher algorithm also makes use of Java’s random numbers, making each key as secure as possible.

## Certificate Generation

Text

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

Unable to find correct pathing for application.properties file.

## Secure Communications

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

Unable to find correct pathing for application.properties file.

## Secondary Testing

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

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

## Functional Testing

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

Text

Description automatically generated

## Summary

1. Refer to the Vulnerability Assessment Process Flow Diagram and highlight the areas of security that you addressed by refactoring the code.
2. Discuss your process for adding layers of security to the software application and the value that security adds to the company’s overall wellbeing.
3. Point out best practices for maintaining the current security of the software application to your customer.

Refactoring the provided code included adding a RestController and a ServerController class. The RestController acts as a secure controller for the hash RESTful endpoint and the ServerController manages the cipher I have chosen. The chosen cipher is the SHA-256, which uses minimal code and is recommended for government use. I have also updated the Maven Dependency Check version to 8.2.1 for the most up-to-date and accurate vulnerability checks.

## Industry Standard Best Practices

Monthly maintenance should be performed to identify any new vulnerabilities via the Maven Dependency Check. Updating the plugins in the pom.xml file will also be required to ensure that the plugins used are up-to-date.