****

# 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** | **[Date]** | **[Your Name]** |  |

## 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

Brian Blackman

## Algorithm Cipher

Artemis Financial wants to add a file verification step to its web application to ensure secure communications. When the web application is used to transfer data, the company will need a data verification step in the form of a checksum. A strong hash function I recommend is SHA-256. SHA-256 is a cryptographic hash that creates a unique 256-bit string from any input of data.( Ballejos, 2024) Jeff Lowery mentioned in his 2020 article “MD5 vs SHA-1 vs SHA-2 - Which is the Most Secure Encryption Hash and How to Check Them”, That the ideal hash function returns no collisions. A collision being two different inputs of data creating the same hash value. Having two different inputs that create the same hash value can lead to secure data getting into the wrong hands. This is like the open sesame analogy in Iron Clad Java, where you create a message to open a door and someone records your message and they use it to open your door. While MD5 may be faster, SHA-256 is more secure and was created by National Institute of Standards and Technology and they recommend use SHA-256 over MD5 or SHA-1.(Lowery, 2020) It is a part of the SHA-2 family and as of today it is the most widely used and considered the most robust. SHA-256 does not utilize random numbers or symmetric and non-symmetric keys, but the key generation is random. SHA-256 can be paired with tools that of random numbers or symmetric and non-symmetric keys.

## Certificate Generation

Insert a screenshot below of the CER file.

A computer screen with white text

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A screenshot of a computer

Description automatically generated

## Secure Communications

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

A screenshot of a computer

Description automatically generated

## Secondary Testing

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

A computer screen shot of a program

Description automatically generated

A screenshot of a computer

Description automatically generated

## Functional Testing

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

A screen shot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A black screen with a black border

Description automatically generated with medium confidence

## Summary

When the dependency check we had hundreds of vulnerabilities must of the needed to be updated so I did this and that took care of a lot. I had to suppress 5 of them due to the fact I needed to update my JRE library which would allow me to run better versions of software which would be an easy fix. I have also updated the dependencies check to 11.1.1.

## Industry Standard Best Practices

I believe using the most up-to-date version of software will allow less vulnerabilities. Also use SHA-256 the most robust hash algorithm ensures data is safe and secure.

References:

Lowery, J. M. (2020, March 26). MD5 vs sha-1 vs SHA-2 - which is the most secure encryption hash and how to check them. freeCodeCamp.org. https://www.freecodecamp.org/news/md5-vs-sha-1-vs-sha-2-which-is-the-most-secure-encryption-hash-and-how-to-check-them/

Ballejos, L. (2024, July 1). What is SHA-256?: Definition & overview. NinjaOne. https://www.ninjaone.com/it-hub/endpoint-security/what-is-sha-256/#:~:text=SHA%2D256%20plays%20a%20critical,updates%20to%20mitigate%20potential%20vulnerabilities.

Jim Manico, & August Detlefsen. (2015). Iron-Clad Java : Building Secure Web Applications. McGraw Hill.