****

# 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** | **4-27-25** | **Jeff Leazenby** | **New Revision** |

## 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

Jeff Leazenby – Global Rain

## Algorithm Cipher

Artemis Financial is looking to take its operations and modernize them. Part of this effort is incorporating modern and effective software security. Our company, Global Rain, will be focused on implementing a solution that protects any client and financial data through a file verification system in its web application. To accomplish this, we will be recommending the use of SHA-256 as the encryption standard. This will protect the information, and files, from being stolen or used by any outside person or group. SHA-256 is an extremely secure algorithm which would require vast amounts of time and resources to penetrate.

With regards to the hash function built into SHA-256 one of the key factors is that every input results in a unique output. What that means is that there are no two pieces of data, file names, etc. that can result in the same output when going through this hashing process. When the hash function is used, the input is compressed and then converted through a hash table to a unique output. The output is unique to any individual input. In the process of incorporating SHA-256 we will be using symmetric keys. The impact that this will have on operations is that the transfer of data and files will be faster. The difference between a symmetric and asymmetric key system is whether the system is using one or 2 keys. It can be thought of as encrypted the same way in both directions and encrypting different ways in each direction. When a file is called in an asymmetric system, a key is used to encrypt the data which is different from the key that was used in the storage of the same file. The use of random numbers is an important aspect of any encryption scheme. Pre-defining encryption has less strength than incorporating a random aspect. Within SHA-256 an authentication code can incorporate both the hash function and a secret key which comes from randomly generated numbers. This incorporation further strengthens the fidelity of the data being transferred between two devices.

## Certificate Generation

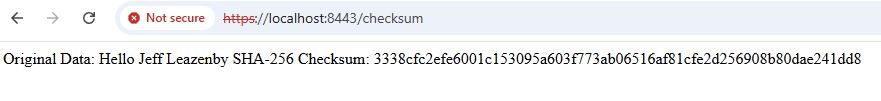
A computer screen with white text

AI-generated content may be incorrect.A screenshot of a computer

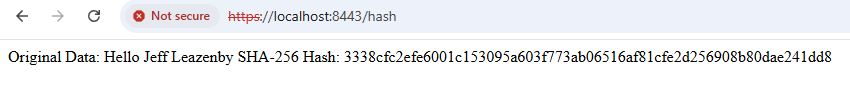
AI-generated content may be incorrect.A screenshot of a computer program

AI-generated content may be incorrect.

## Deploy Cipher



## Secure Communications



## Secondary Testing

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

## Functional Testing

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

## Summary

To improve safety, I added a Controller and went through the pom file, updating and editing/removing dependencies as necessary. This corrected most of the vulnerabilities. I then updated to the newest version of SPRING which further removed issues but was left with 5. I investigated each of these five and found that no action was necessary. They are still present but would only be an issue under certain circumstances which are not implemented in the code.

## Industry Standard Best Practices

To further improve security there are a variety of practices that should also be followed:

Least privilege: Users should only have access to the rights that are required in their roles, which prevents people who gain access to these users from having full access to all possible functions and commands.

Logging: Logging should be used to track what users are attempting to do and what functions are being used. This should also include repeated attempts to gain unauthorized access from external parties.

Encryption: Data should be secured using industry-standard algorithms and secure communication protocols.

Regular updates: Even after releasing a product, the software should be constantly monitored for issues. When vulnerabilities arise or are discovered, the software should be updated to combat these vulnerabilities. This includes updating dependencies.

Authentication: Secure passwords and password requirements should be implemented for all users. Strong password policies make it difficult for malicious parties from guessing or determining passwords.

Input Validation: Input that is sent to a system should be screened and sanitized to prevent the injection of harmful code or commands.

Implementing these helps secure data and files, which was the stated goal of Artemis Financial.