****

# 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** | **10-16-2022** | **William Leaver** |  |

## 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

William Leaver

## Algorithm Cipher

The Cipher that I am going to use is SHA-256. I will use SHA-256 because it is not decryptable and is collision-resistant. This makes for an extremely secure checksum that would be near impossible to replicate. There is a ^256 change for a collision in SHA-256, and to this day a known collision has never been produced. The SHA-256 cipher is neither asymmetric nor symmetric as it falls into the has function category. Both symmetric and asymmetric algorithms can be decipher using the appropriate key. While using SHA-256 is a one-way encryption. This makes it perfect for the checksum in Artemis Financials application.

## Certificate Generation

Text

Description automatically generated

## Deploy Cipher

Graphical user interface, text, application, email

Description automatically generated

## Secure Communications

Graphical user interface, text, application

Description automatically generated

## Secondary Testing

Text

Description automatically generated

## Functional Testing

Graphical user interface, text

Description automatically generated

## Summary

I ensured this applications security by using signed certificates and HTTPS. I also used a checksum that undecipherable. This allows for a one-way checksum that someone theoretically will not be able to reverse engineer. The cipher SHA-256 also has no reported collisions which will ensure it always works as expected.

## Industry Standard Best Practices

It is always important to use industry standards when securing software. If you do not, you may be leaving a hole that a malicious user could exploit to cause harm to your business or other users. The easiest way to ensure that you are keeping up with industry standards is to do research prior to each project. Using a Vulnerability Assessment Process Flow Diagram, you can determine what security protocols will work best for that project. It will also give you an idea of where you will need to implement security and where you do not.