****

# 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** | **June 2023** | **Victor Jarvis** |  |

## 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

Victor Jarvis

## Algorithm Cipher

SHA-256 was chosen as the applications encryption as its 256 bit encryption provides and strong and secure method in order to protect the data of Artemis Financial and its customers. Its resistance to brute force and other common attacks make it a great defense for this particular application in more ways than one, as multiple different kinds of data can be encrypted in order to provide multilayer protection.

## Certificate Generation

A screenshot of a computer program

Description automatically generated with medium confidence

## Deploy Cipher

Insert a screenshot below of the checksum verification.

This is where things went a little wrong. I have no idea why I’ve had so much trouble re-doing the checksum verification. I’ve tried it across multiple IDEs fixing various issues as they come up and with a fully refactored codebase that worked just fine when we originally did the exercise. All of my various server attributes are correct but I am consistently hit with refused connections that will not allow me to go any further despite the code running without error. I’ve tried with both BigInteger and bytesToHex and I’m honestly just incredibly lost on this. The dependency checks worked just fine but unlike the first time we did this I cannot get the checksum to come up, secured or unsecured, Live Server or the localhost 8443/hash path.

## 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.

A screenshot of a computer

Description automatically generated with medium confidence

## Functional Testing

Insert a screenshot below of the refactored code executed without errors. A picture containing text, electronics, software, screenshot

Description automatically generatedA screen shot of a computer program

Description automatically generated with low confidence

## Summary & Industry Standard Best Practices

In the application, a hashing function was added in order to take a string and encrypt it using SHA-256 bit encryption, while generating a checksum in order to display that the encryption has been successfully accomplished. A positive is that this could be utilized for other forms of data within the application. Consistent maintenance and attention to dependencies would further guarantee the security of the application. If anything were to happen that would render the current encryption process vulnerable, or similar happening to a dependency, then that would need to be addressed and refactored into the code. Despite following the processes and ensuring certificate generation and refactoring as much as possible I was unable to complete the task.