****

# 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** | **8/14/25** | **Faith Agepogu** |  |

## 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

Faith Agepogu

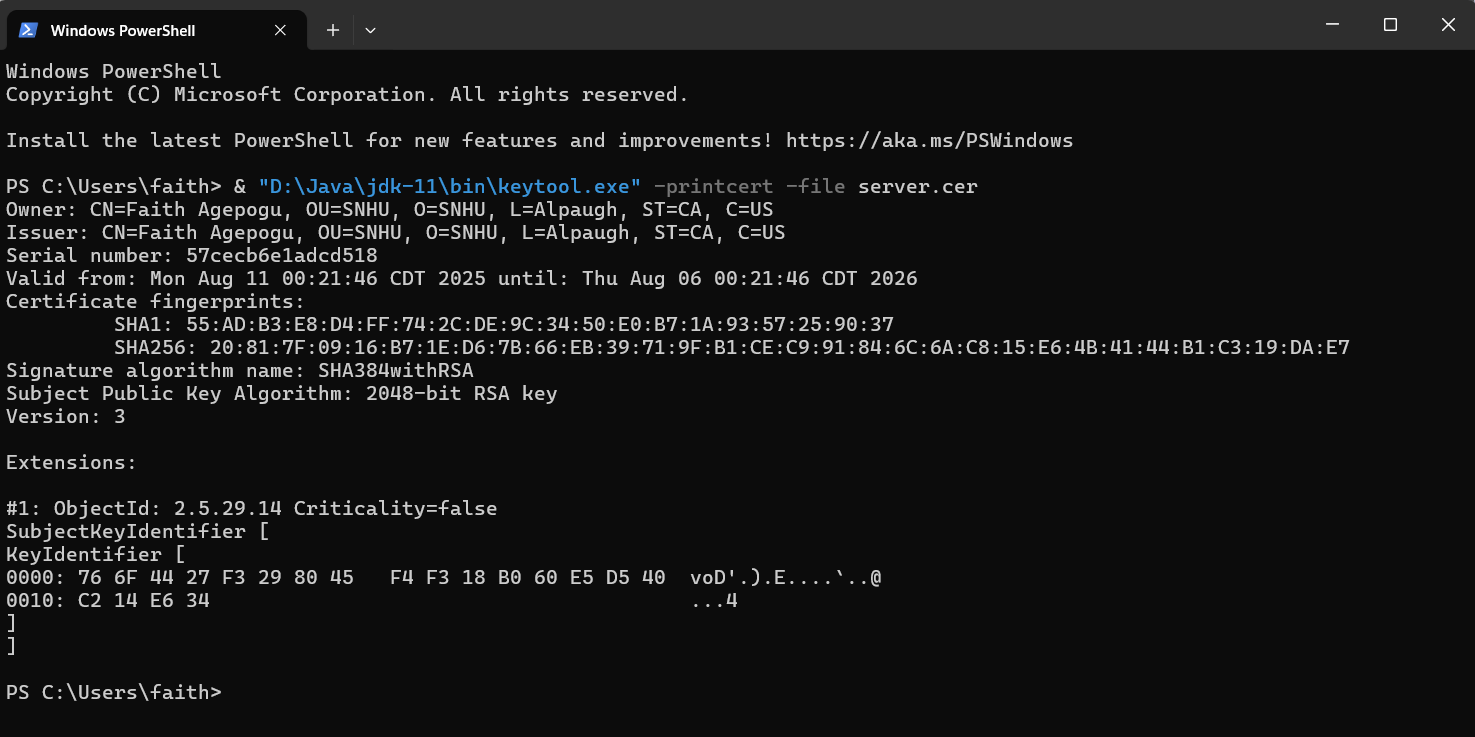
## Algorithm Cipher

The purpose of the project is to modify the existing code base to improve its security. The company Artemis Financial wants to offer financial programs to their clients, and their customer base is international. Because of this I would use a SHA-256 algorithm since it is widely trusted and secure, and it is even used in government systems. This cipher is difficult to break because the generated keys are always different, guaranteeing no pattern can be observed for an attacker to use to break into the system. It works by creating a hash, and it first compresses the input before randomizing it. The contents compressed will determine the length of the hash output.

Symmetric keys take less time to set up and are easier to work with since only one key is being referred to, so that is the plus side to using symmetric keys. For asymmetric keys, two keys are used instead of one, which means it can be more secure of a method vs the symmetric one that has only one key. The additional key provides an extra layer of security and makes it more difficult for an attacker to break the system, since they would have to work with two keys instead. It could take years for an attacker to guess the key values because of this. The earliest observation of encryption dates back to 600 BC, and the improvements made since then make it possible for systems to be fully secure, given the proper safety measurements are implemented.

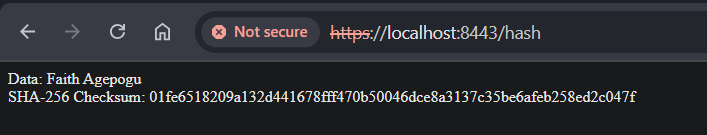
## Certificate Generation

Insert a screenshot below of the CER file.



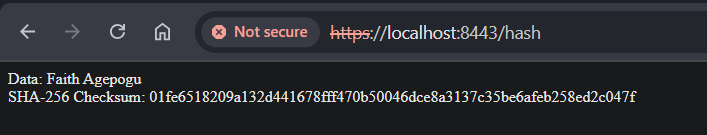
## Deploy Cipher

Insert a screenshot below of the checksum verification.



## Secure Communications

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



## Secondary Testing

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

[Insert screenshots here.]

## Functional Testing

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

[Insert screenshots here.]

## Summary

[Insert text.]

## Industry Standard Best Practices

[Insert text.]