****

# 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** | **28/2/2025** | **Mike Bitts** |  |

## 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

Mike Bitts

## Algorithm Cipher

For this program, I suggest using the Secure Hash Algorithm 256-bit (SHA-256) as the algorithm cypher. SHA-256 is a cryptographic hash function that generates a fixed 256-bit, or 32-byte, hash value from user inputted data, encrypting the users information. This algorithm is very widely used and reliable, very useful for data integrity and security. The wide=spread use and overall reliability of the function makes it the best option for this program, as it is also quite memory efficient and easy to use.

## Certificate Generation

Insert a screenshot below of the CER file.

A screenshot of a computer program

AI-generated content may be incorrect.

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A screenshot of a computer

AI-generated content may be incorrect.

Checksum (SHA-256): 0fb388b5aae916f2aa2e008bee40fa659643c8f4351e6f6cda244a60ca437424

## Secure Communications

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

A screenshot of a computer

AI-generated content may be incorrect.

Regardless of what I did, I was unable to get the program to launch with HTTPS.

## Secondary Testing

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

A screenshot of a computer program

AI-generated content may be incorrect.

## Functional Testing

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

## Summary

The changes that I made to the now refactored code were: adding the SHA-256 hash function, attempted an HTTPS configuration in application.properties (which failed), and added documentation for clarity. The code base was not only incomplete, but it was also insecure in the way that it was given to me. By adding the hash function, I was able to ensure the codes security and that it will maintain integrity as it ages and gets used more and more. In addition to the hash function,. I also updated the application.properties file in order to successfully work with and create/delete keystore files/values. In summation, I completed the program and added the hash function and integrated keystore functionality.

## Industry Standard Best Practices

The industry standard best practices that I used for this code were: secure coding, documentation, and dependency management. I used secure coding by including the aforementioned hash algorithm and attempted HTTPS setup, ensuring double security of the users information and web-communications. I included adequate, clear, and easy to read documentation to help the next developer to know exactly what is happening and what each bit of code will do and how it should behave. I utilized the built-in maven dependency checks to ensure that the dependencies and code worked as intended.