****

# 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** | **12/10/2023** | **Ben Miller** |  |

## 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

Ben Miller

## Algorithm Cipher

PBKDF2WithHmacSHA1

## Certificate Generation

Insert a screenshot below of the CER file.



A screenshot of a computer

Description automatically generated

A screen shot of a computer code

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A close up of letters

Description automatically generated

## Secondary Testing

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

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Summary

The base code barely works after spending hours researching what the problem might be, and through trial and error finally getting it. It barely works because the application file in \target\classes must be edited for the code to run, but it gets overwritten after the code runs. This causes a lot of problems trying to implement more functionality and test the project. I implemented the requirements as much as I could.

## Industry Standard Best Practices

For a cypher algorithm, it is best practice to use one of the SHA-2 algorithms, such as SHA-256 or SHA-512. This provides sufficient protection with a hash code that is nearly impossible to crack. When it comes to packages and dependencies, using up-to-date code is a must, since the older code has many issues, as shown on the dependency check report. Also, the older code does not support the latest Keystore types, making the process even less secure.