****

# 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** | **10-14-2023** | **Nicholas Martin** |  |

## Client



## Instructions

Submit these 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

Nicholas Martin

## Algorithm Cipher

For the algorithm Cipher I recommend the AES 256 encryption. It is the gold standard for all decryption and certified by the US Government. AES 256 uses 256 bits to encrypt data, meaning the key generator is longer and more complicated therefore harder to break. AES uses a symmetric block system meaning the keys are generated using a block like table to both encrypt and decrypt the message. Despite this encryption method being around for over a decade it still hasn’t been fully broken to retrieve encrypted messages.

## Certificate Generation

Insert a screenshot below of the CER file.

A screenshot of a computer program

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

I was able to encrypt a message, but the decryption did not work as well and unsure why.

A screenshot of a computer

Description automatically generated

A screen shot of a computer program

Description automatically generated

## Secure Communications

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

I tried to use the spring framework but that did not work so much. As such I have no screenshot of it.

## Secondary Testing

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

A screen shot of a computer program

Description automatically generated

## Functional Testing

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

A screenshot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

## Summary

Though I was unable to decrypt the message the encryption process worked well. I was not able to figure out how to up it in the spring part of the code, but the summoning of AES 256 worked. More research would need to be done in order to figure out exactly what to do from here.

## Industry Standard Best Practices

I successfully deployed encryption and was able to encrypt a message. The internet part of the coding was not successful but a fully encrypted AES 256 message was generated for this project.