****

# 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** | **8/14/2024** | **Faris Malik** |  |

## 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

Faris Malik

## Algorithm Cipher

There are many encryption algorithms available but I proposed to use AES-256 algorithm for mitigating security issues. AES (Advanced Encryption Standard) is a symmetric encryption algorithm used to encrypt the sensitive data is recognized to be secured and efficient, and is the most common encryption algorithm right now.

Overview: AES operates on 128-bit blocks and uses key lengths of 128, 192, or 256 bits, with AES-256 being the most secure. It makes use of a symmetric key; that is the key used to encrypt the message is the same one used to decrypt the message, and therfore offers adequate security.

Hash Functions and Bit Levels: AES-256, together with SHA-256, guarantees the data and message’s integrity and validity since it prevents any modifications during its transmission. SHA-256 generates a 256 bit hash value of the original data, to thereby afford a high degree of collision resistance so that an attacker will find it almost impossible to try and change the data and have this go unnoticed.

Random Numbers and Key Management: AES-256 relies on secure random numbers for key generation. The randomness ensures that keys are unpredictable, which is important for resisting brute-force attacks. Symmetric key encryption like AES-256 requires careful management of key distribution, as the security of the system depends on keeping the key secret. Daily rotation of keys should be considered and there should be good measures put in with regard to key exchange.

History and Current State: AES is an algorithm created by the cryptographers Joan Daemen and Vincent Rijmen, and being the new standard encryption system in USA since the year 2001. It became the data encryption standard since it achieves both data protection and good processing rate. In contrast to the older algorithms such as DES, which have been cracked or proven to be weak, AES is still resistant to all the practically known attacks if implemented correctly. But today AES-256 is widely used by governments, financial institutions and organizations all over the world for encrypting classified and sensitive information.

## Certificate Generation

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedInsert a screenshot below of the CER file.

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A screenshot of a computer

Description automatically generated

## Secure Communications

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

A screenshot of a computer

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 program

Description automatically generated

A screenshot of a computer

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

## Summary

In the process of refactoring the code, the changes were carried out very carefully focusing on the issue of security of the software application and passing security testing. Refactoring was carried out for critical areas that showed up. Some of the areas of security were initially developed including replacing old libraries which were vulnerable to security threats, incorporating encryption standards for the security of the important information and input validation to avoid injection attacks. All the changes made where reviewed in order to conform the security requirements and standards outlined so that the application becomes less vulnerable to potential threats.

Aside from fixing critical threats, the process of refactoring included multiple layers of security. These were some of the measures such as use of encryption to protect data at rest and in transit, improving the authentication and authorization mechanism, and applying security patches on third-party libraries. By stacking these security measures one after another the layers of defense for the application were added which enhanced the potential to protect from the possible exploits by providing systematic security from all directions.

## Industry Standard Best Practices

The adoptions of the standard best practices guidelines provided by the industry regarding the security of code was instrumental in the effort to minimize or erase all identified security threats to the software. Some of the measures that would include input validation, adequate error checking and beginning of the security coding libraries were strictly implemented in the best practices. To eliminate the presence of various types of vulnerabilities, updates and patches were provided to third-party dependencies frequently. When incorporating these best practices, the codebase of the application was thus strengthened against such attacks, thus minimizing vulnerability.  
  
The application of the mentioned industry standard best practices is critical for the operation of the firm’s proper health. Furthermore, it guards the application against probable security invasion while at the same time creating trust in the minds of stakeholders and users. Secure coding standards help to improve the quality of the software so that people’s data remains protected, and the integrity of the service is preserved. Also, commitment to best practices builds the corporate image and, at the same time, complies with the legal obligations that may contribute to the longevity of the business and also reduces possible financial and reputation risks.