****

# 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** | **4/18/2025** | **Victor Bandish** | **Version 1.0** |

## 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

Victor Bandish

## Algorithm Cipher

Given the security vulnerabilities described in the scenario, I recommend deploying **AES-256 (Advanced Encryption Standard with a 256-bit key size)** as the encryption algorithm. AES is a symmetric-key algorithm, meaning it uses the same key for both encryption and decryption, which ensures efficiency and security for high-volume data encryption. AES-256, in particular, is considered one of the most secure encryption standards available, offering robust protection against modern cryptographic attacks.

AES operates by dividing the plaintext data into fixed-size blocks (128 bits) and performing a series of transformations, such as substitution, permutation, and mixing, over multiple rounds. The number of rounds depends on the key size—10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. In AES-256, the 256-bit key size provides a significantly larger key space compared to shorter key lengths, making it highly resistant to brute-force attacks. This large key size contributes to its ability to protect against future advancements in computational power, making it a future-proof choice for secure communications.

Regarding hash functions, AES relies on strong cryptographic techniques such as the substitution-permutation network (SPN), which combines bitwise XOR operations, byte substitution, and shifting to achieve both confusion and diffusion within the data. AES-256 employs 256-bit keys, which means the cipher operates on a 256-bit key throughout its encryption and decryption operations, further increasing the complexity and strength of the encryption.

AES uses random numbers primarily in the generation of initialization vectors (IVs), which ensure that the same plaintext encrypted multiple times results in different ciphertexts. IVs prevent attackers from recognizing patterns in the encrypted data, thus improving the overall security. Additionally, symmetric encryption means that the same secret key is used for both the encryption and decryption processes, which is more efficient for encrypting large volumes of data compared to asymmetric methods, like RSA. In contrast, asymmetric encryption algorithms use a pair of keys—one public and one private—and while secure, they tend to be slower and less efficient for encrypting large amounts of data.

## Certificate Generation

Insert a screenshot below of the CER file.

A computer screen with white text

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.

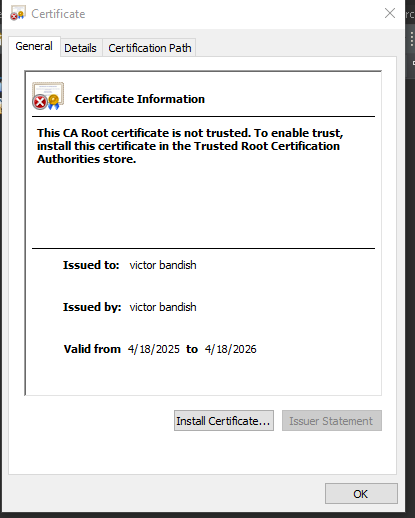
## Secure Communications

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

A screenshot of a computer

AI-generated content may be incorrect.

Will not be secured due to my cert being self assigned but HTTPS is working



## Secondary Testing

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

A screenshot of a computer

AI-generated content may be incorrect.

## Functional Testing

A screenshot of a computer

AI-generated content may be incorrect.

## Summary

The code was refactored to address key security vulnerabilities identified during the vulnerability assessment, focusing on input validation, secure authentication, data encryption, and proper error handling. Layers of security were added through robust encryption, multi-factor authentication, and secure coding practices, reducing attack surfaces and ensuring data protection. Static code analysis and regular audits were employed to detect and fix security flaws. By implementing a defense-in-depth approach and continuous monitoring, the application now complies with security testing protocols, providing enhanced protection against potential threats.

## Industry Standard Best Practices

To maintain the software application's existing security, I applied industry-standard best practices such as input validation, secure authentication, and data encryption. Input validation ensures that only expected data is processed, preventing common vulnerabilities like SQL injection and cross-site scripting (XSS). Secure authentication was strengthened by implementing multi-factor authentication (MFA) and ensuring robust session management, protecting against unauthorized access. Additionally, all sensitive data was encrypted using strong encryption algorithms, safeguarding against data breaches.

Applying these best practices not only mitigates known security vulnerabilities but also significantly enhances the company’s overall well-being. By adhering to secure coding practices, the risk of security breaches is minimized, leading to reduced exposure to financial and reputational damage. Furthermore, maintaining a secure application fosters trust with clients and customers, ensuring compliance with legal and regulatory requirements and ultimately contributing to long-term business sustainability and growth.