****

# 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** | **4/20/2024** | **Michael Evans** | **Its been a pleasure taking this course (:** |

## 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

Michael Evans

## Algorithm Cipher

**Recommendation for Encrypting Archive Files for Artemis Financial**

After looking into Artemis’ security needs and the current best practices in encryption, I recommend the use of the Advanced Encryption Standard (AES) algorithm cipher for encrypting their archive files.

**Reasoning:**

1. **Security Best Practices:** AES is recognized as one of the most secure encryption algorithms available today. It provides a high level of protection against various types of security attacks due to its design and key lengths.
2. **Risk Assessment:** By employing AES, Artemis can reduce risks associated with unauthorized access to sensitive financial data stored in their archive files. AES has undergone extensive cryptographic analysis and is promoted by government agencies for protecting classified information, indicating its reliability in protecting data.
3. **Compliance with Regulations:** AES aligns with current government regulations. Its use ensures compliance with data protection regulations and industry standards, offering some peace of mind regarding legal and regulatory requirements.
4. **Usage and Implementation:** AES can seamlessly be added into Artemis’ existing code. It supports both symmetric key encryption, suitable for encrypting large archive files efficiently, and non-symmetric key encryption for secure data transmission and key exchange.
5. **Justification:** AES uses hash functions and operates at various bit levels, including 128-bit, 192-bit, and 256-bit key lengths, allowing for users to have flexibility when balancing security and performance requirements. It uses random numbers very well to enhance cryptographic strength, ensuring a strong defense against cryptographic attacks.

**Brief Overview of AES:**

* **Purpose:** AES is a symmetric encryption algorithm designed to protect sensitive data by transforming plaintext into ciphertext using a cryptographic key.
* **Hash Functions and Bit Levels:** AES employs substitution-permutation network (SPN) structure and operates on fixed-size blocks of data. It utilizes hash functions to ensure integrity and authenticity of encrypted data.
* **Random Numbers:** AES incorporates random numbers during key generation and encryption processes, enhancing its resistance against cryptographic attacks.
* **Symmetric vs. Non-symmetric Keys:** AES supports both symmetric and non-symmetric key encryption, enabling secure data transmission and key exchange mechanisms.
* **History and Current State:** AES was established as the standard encryption algorithm by NIST in 2001, replacing the aging Data Encryption Standard (DES). It has since become the de facto encryption standard, widely adopted across industries and applications.

## Certificate Generation

Insert a screenshot below of the CER file.

A computer screen shot of a black screen

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A screen shot of a computer

Description automatically generated

## Secure Communications

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

A screen shot 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

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

Description automatically generated

## Summary

## In refactoring the provided code, security measures were implemented to enhance its robustness. Addition of a route ("/checksum") in the ChecksumController class allows retrieval of a checksum value for static data, enabling integrity verification. The data used for checksum generation now includes both first and last names, improving input complexity and thus security. The calculateChecksum method has been refactored to use the SHA-256 hashing algorithm, enhancing cryptographic strength. Proper error handling for NoSuchAlgorithmException ensures graceful degradation in case of algorithm unavailability. These refinements address security concerns outlined in the Vulnerability Assessment Process Flow Diagram, focusing on data integrity and cryptographic strength, aligning with best practices for secure software development.

## Industry Standard Best Practices

Industry-standard best practices, such as adopting the SHA-256 hashing algorithm for checksum generation, were applied to maintain the software application's security. Proper error handling for potential algorithm unavailability was implemented, enhancing the application's resilience to attacks targeting algorithm vulnerabilities. By adhering to these practices, the company benefits from fortified defenses against cyber threats, reducing the risk of data breaches and unauthorized access. This fosters trust among users and stakeholders, safeguarding the company's reputation and promoting a secure operating environment. Overall, applying industry-standard best practices for secure coding bolsters the company's wellbeing by mitigating risks, ensuring compliance, and fostering a culture of security-conscious development.