****

# 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** | **10/17/2024** | **Damian Duross** |  |

## 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

Damian Duross

## Algorithm Cipher

The goal for Artemis Financial is to include a means of file verification on their web application. The best hash function for this case will be SHA-256. This function is widely used for its efficiency, collision resistance, and compatibility with various platforms/frameworks. It is often used with the AES cipher algorithm, if a file needs to be securely encrypted for additional security. The web app will generate a SHA-256 hash of the file when uploaded, acting as a digital fingerprint. When downloading, the app re-calculates and compares the hash with the previously stored value. Matching values indicate the file has not been corrupted or tampered with.

Since it is a company specializing in finance, AES-256 would be recommended for the encryption and decryption of data. AES is used internationally for data privacy and is compliant with many security standards, including FIPS and the GDPR. The 256 is a reference to the bit-level, and while AES supports 128/192/256 bit levels, 256 is the longer output and the most secure. AES is a symmetric encryption algorithm, which requires only one key for both encryption and decryption. This makes it faster to deal with large datasets, but relies on the security of that key. In an asymmetric system like RSA or ECC, files are encrypted with a public key and must be decrypted with a private key. This makes key distribution much safer because the private key never needs to be distributed. You receive someone’s public key, encrypt a file for transfer, and only the holder of the private key will be able to decrypt it. Oftentimes, a hybrid system can be implemented to use asymmetric encryption to exchange symmetric keys, allowing you to securely send bulk files while being able to make use of the less computationally exhaustive symmetric decryption.

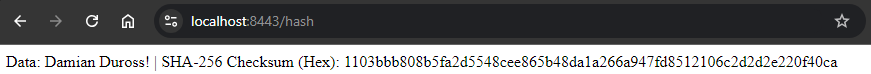
AES has remained the most used encryption standard worldwide since it replaced DES in 2001. Many encryption systems utilize hybrid encryption to benefit from the speed of AES and the secure exchange of RSA/ECC. However, with the advent of quantum computing, it is probable that the emergence of new quantum-resistant algorithms will replace these ciphers in time.

## Certificate Generation

A computer screen shot of a black screen

Description automatically generated

## Deploy Cipher



## Secure Communications

## Initially, I had a hard time getting Chrome to view my self-signed certificate as secure, since I am not a trusted CA. I was able to recreate my certificate, however, and incorporate the SAN field listing localhost and IP 127.0.0.1, which worked out in the end. What an adventure that was!

A screenshot of a computer

Description automatically generated

## Secondary Testing

Initial dependency-check (pre-refactoring):

A screenshot of a computer error

Description automatically generated

Secondary testing after code refactoring:

A screenshot of a computer error

Description automatically generated

It can be seen that no new vulnerabilities are introduced with the inclusion of my code.

## Functional Testing

Screenshot of refactored code running without error (I created a new class to address the FIXME):

A screenshot of a computer program

Description automatically generated

## Summary

My refactoring of the code included the introduction of a Spring Boot REST controller that generates a SHA-256 checksum for a given string (my first and last name). The checksum is then returned as a hexadecimal, which is a common way to display a hash. Referring to the vulnerability assessment process flow chart, our use of SHA-256 and our dependency-check reports showcasing a lack of newly introduced vulnerabilities suggest our use of cryptography was well-implemented. I also incorporated secure error handling in the form of NoSuchAlgorithmException catches to throw RuntimeException, which will allow the app to throw more easily readable diagnostic error messages. Using the keytool to generate a self-signed certificate and incorporating it into our application.properties adds yet another layer of security by establishing verifiable trust between the client / server.

## Industry Standard Best Practices

I utilized SHA-256 for our hashing algorithm, which is widely used internationally to protect data. SHA-256 is resilient against brute-force attacks and is collision-resistant. My proper implementation of error handling allows exceptions to be thrown securely without compromising the system. Using secure transmissions with https provides protection against the interception and tampering of data.

The industry standard best security practices are important to a company’s well-being for many reasons. Firstly, there are many regulations in place for businesses that require data protection, and the standard best practices are a good step in ensuring all of those requirements are met. There is also the obvious mitigation of potential attacks, which could result in breaches that could then lead to significant financial loss. Such attacks could also be damaging to a company’s reputation, but good coding practices goes a long way to establish trust and contribute to the longevity of business relationships.

In essence, not only do you *have* to implement industry standard best practices – you *should*.