****

# Practices for Secure Software Report

Table of Contents

[Document Revision History 3](#_Toc102040754)

[Client 3](#_Toc102040755)

[Instructions 3](#_Toc102040756)

[Developer 3](#_Toc102040757)

[1. Algorithm Cipher 3](#_Toc102040758)

[2. Certificate Generation 4](#_Toc102040759)

[3. Deploy Cipher 4](#_Toc102040760)

[4. Secure Communications 5](#_Toc102040761)

[5. Secondary Testing 5](#_Toc102040762)

[6. Functional Testing 6](#_Toc102040763)

[7. Summary 6](#_Toc102040764)

[8. Industry Standard Best Practices 6](#_Toc102040765)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **4/21/24** | **Tyler Mort** | **Added cipher algorithm and deployment, certificate generation, secure communications, testing, summary, and best practices.** |

## Client



## Developer

Tyler Mort

## Algorithm Cipher

The cipher I have used to encrypt data for the web server is SHA-256. This cipher is commonly used for secure information such as passwords or digital signatures. This algorithm uses up to 256 bits of information and performs several operations on the input such as bitwise logical functions, addition, rotation, and more. This results in a dramatically different output than the original input message. For example, the string “Tyler Mort and some unique data string to decrypt” becomes e364771c0d1196122c68e1ed98b5ca94e3ecd15b22edf3aea53737b43185e654. This new string shares no resemblance to the original and very importantly is a completely different length. This is because there is a fixed length to the output which helps prevent hackers from reverse engineering the size of the input. SHA-256 is also not prone to collisions in which two different inputs would have the same output. This would make it hard for the program to determine the original input and could allow a malicious person to use an incorrect password and still gain access to the system if they share the same hash. Since SHA-256 does not collide, it is safe in this regard. Symmetrical keys use the same key for both encryption and decryption while non-symmetrical keys use a private key to encrypt and a separate public key that gets shared to those needing to decrypt. Symmetric keys are faster and easier to use generally than a non-symmetrical key. However, if the key is discovered or compromised, the entire system is vulnerable. In a non-symmetric system, you do not need to share the private key and it is therefore less susceptible to attack. Each system has its uses, however, an asymmetric system generally works better at scale when connecting with unknown 3rd parties. The current state of encryption algorithms is rapidly changing from its past. While originally the data buffers on algorithms were quite small such as 8-bit or 16-bit encryption, they now commonly start with as much as 256-bits. We are constantly needing to improve the buffer size as well as the algorithms used to generate keys and encrypt messages. As the technology to encrypt gets better, the technology used by hackers gets better as well. The growth of artificial intelligence, machine learning, and general compute power has led to much stronger decryption processes and may eventually lead to the reverse engineering of several popular keys. As such, the current battleground of cyber security involves constantly updating ciphers, encryption standards, and a large rotation of keys to keep data fresh and secure.

## A screenshot of a computer Description automatically generatedCertificate Generation

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A screenshot of a computer

Description automatically generated

## A screenshot of a computer Description automatically generatedSecure Communications

## Secondary Testing

A computer screen with many lines and text

Description automatically generated

A screenshot of a computer

Description automatically generated

## A screenshot of a computer Description automatically generatedFunctional Testing

## Summary

In order to comply with security testing protocols, I made sure that I used the latest form of the OWASP dependency check report, version 9.1.0. Using the latest version of this maven-based dependency tester helps to ensure that we are catching the latest vulnerabilities possible. The cyber space is constantly evolving, and with that comes new vulnerabilities. As dependencies get updated, so do their vulnerabilities. OWASP also updates their vulnerability lists to properly reflect this. This helps give the developer a better idea of how their program may be open to attacks. This allows them to adjust the dependencies or better protect inputs in order to prevent malicious attacks and bolster their API interactions. It also helps to bolster input validation, code quality, and code error protection. While using OWASP to refine API interactions and dependencies, there is also information provided on how SQL injection may occur for inputs or how attackers can use specific errors to gain malicious access. Developers can then refine their input sanitation methods or insert error catches to prevent these attacks.

## Industry Standard Best Practices

Industry standard best practices are crucial for maintaining a safe and secure application environment. To comply with these best practices, I made sure to use the most recent versions of dependencies such as OWASP. Keeping these dependencies up to date helps to prevent older loopholes from being utilized and ensures hackers must continue to invent new methods to attack a system, rather than having easy access with their old methods. Another way to comply with best practices is through the use of a cipher with checksum validation. By using the SHA-256 cryptographic cipher, I can safely encrypt data such that it is not readable by malicious third parties. This helps to keep confidential company information safe as well as user data. In addition, a valid certificate is provided for the website. This helps to verify ownership of the website and ensure it has not been manipulated or spoofed by hackers. Combining all three of these best practices helps to ensure that the end user can be confident in their usage of the site.