****

# 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** | **04/12/2023** | **Alex Wells** |  |

## 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

Alex Wells

## Algorithm Cipher

I have chosen to use SHA-256 as my algorithm cipher. It’s very important to avoid collisions in your encryption algorithm, which SHA-256 does. A collision is when two sperate messages encrypt to the same result. This is commonly avoided by padding the data randomly to prevent it from being at all possible. The purpose of SHA-256’s hash function is to take “plain text” data and “hash” it by putting the data through a mathematical function to be unreadable without the “key” to decrypt the text back into plain text. The purpose of the bit level of the cipher is that the more bits the cipher has the more “entropy” or randomness is in the encryption and as such it makes it exponentially more difficult to crack, especially if a malicious actor is trying to break into the system by brute force. For ciphers you want to use randomly generated numbers to generate a key to make the system more secure, if you were to generate keys sequentially for instance, once a bad actor figures that out they have access to every key you make in the future. There are two ways to handle keys, symmetric keys asymmetric. Symmetric key-based encryption has 1 key that is sent securely with the encrypted data to decrypt it. Asymmetric keys have 2 keys, one public key that encrypts the text, and a private key that the user uses to decrypt the data. Encryption is practically as old as writing, dating back to ancient Greece or even earlier. Modern encryption is a “mature” field and has settled into symmetric and asymmetric algorithms although there’s new algorithms and new ways to break them being created all the time.

## Certificate Generation

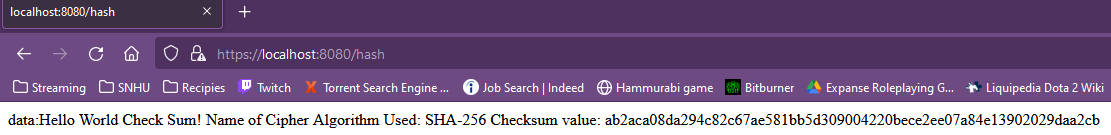
Insert a screenshot below of the CER file.

Text

Description automatically generated

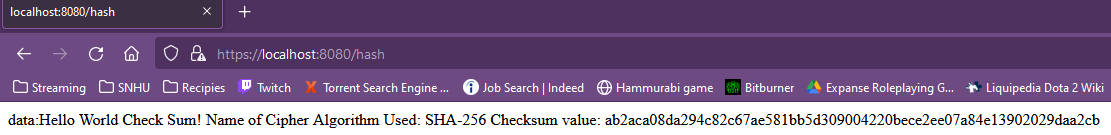
## Deploy Cipher

Insert a screenshot below of the checksum verification.



## Secure Communications

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



## Secondary Testing

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

Text

Description automatically generated

Graphical user interface, text, application, email, website

Description automatically generated

## Functional Testing

Insert a screenshot below of the refactored code executed without errors.

Graphical user interface, text

Description automatically generatedSummary

The code has been refactored to use the HTTPS protocol and the SHA-256 algorithmic cipher. This provided secure encrypted communication between the client and the server to protect user data.

## Industry Standard Best Practices

By using secure encryption algorithms and creating a security certificate we have complied to some industry standard best practices, this allow us to have secure communication between the client and server as well as ensure that users are able to trust the content and that the application is secure. It’s important that the company follow those best practices as they have been honed for years on the cat and mouse game that is creating a secure program. By following these they will have a leg up from a security perspective over those companies that choose to ignore them or skimp on them to save time or money, which leaves them more vulnerable to malicious actors wreaking havoc on their system or even worse, their users data.