****

# 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** | **Dec 10, 2022** | **Mason Utt** |  |

## 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

Mason Utt

## Algorithm Cipher

I am choosing SHA-256 as my algorithm cipher. I am choosing it because it avoids collisions by being large, there are different possibilities which means it is practically impossible for two different strings to have the same value. To avoid collisions means that two different strings will not have the same hash. SHA-256 changes completely even if just one character value is changed. This increases security. SHA-256 does not use a key and will provide the same hash with the same data. It uses pseudorandom numbers to make the hash by changing by each subsequent character value that is passed. The history of encryption starts a long time ago. The first encryption in modern history would be the Enigma cipher in the 1940’s. The DES cipher was created and was very powerful but is commonly broken now. Currently encryption algorithms are very strong and almost impossible to crack without knowing the original message or key.

## Certificate Generation

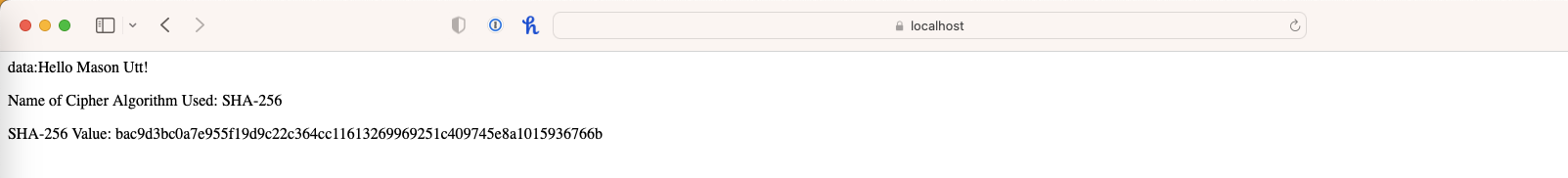
Insert a screenshot below of the CER file.

Graphical user interface, text, application, email

Description automatically generated

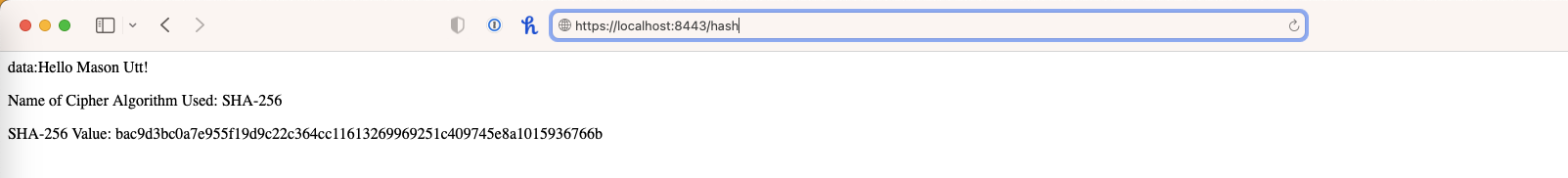
## Deploy Cipher

Insert a screenshot below of the checksum verification.



## Secure Communications

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



Graphical user interface, text, application, email, Teams

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

Graphical user interface, application

Description automatically generated

## Functional Testing

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

A computer screen capture

Description automatically generated with medium confidence

## Summary

There was no input in my code so there was no need for input validation, this in part is an added layer of security, removing an option entirely. I chose to implement a REST API to secure the interactions. For cryptograph I chose to implement a SHA-256 encryption. I added a REST controller to handle the client/server end.

## Industry Standard Best Practices

I used industry best practices to make sure there were the fewest weaknesses in my code as possible. The advantage of adhering to these are a reduced rate of failure and an increase in security of software (goldskysecurity.com p1). I recommend using all of the standard best practices.

DOC:

*The advantages of secure coding best practices*. GoldSky Security | Cyber Security Solutions. (2022, June 8). Retrieved December 11, 2022, from https://www.goldskysecurity.com/security-by-design-the-advantages-of-secure-coding-best-practices/