****

# 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** | **1-6-23** | **Tiffany McDonnell** | **Initial Release** |

## 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

Tiffany McDonnell

## Algorithm Cipher

Artemis financial is a company who helps people in the world of finance. Everything must be secure to the highest extent because it can affect people’s lives. Therefore, Artemis Financial will use SHA 256 cipher. The cipher will be used to help encrypt data passing from server to user and vice versa.

Hashing the data is one of the keys to encryption and helping everything stay secure. When a cipher is 256 bits that means the data is separated into 256 coded pieces. These pieces are then scrambled up into unreadable code. Another factor that can help while hashing is the use of random numbers. Since random numbers have no rhyme or reason, cracking the algorithm is much more difficult. Ciphers have a long history. Though we see them used all over, in our everyday lives for security, it was back around 400 BC that we started to see where it was first implemented. Ancient Greeks started to use ciphers to communicate to those in military standings without others knowing the orders.

## Certificate Generation

A screenshot of a computer

Description automatically generated

## Deploy Cipher

package com.snhu.sslserver;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import java.security.MessageDigest;

import java.security.NoSuchAlgorithmException;

import java.nio.charset.StandardCharsets;

import java.math.BigInteger;

*@SpringBootApplication*

public class SslServerApplication {

public static void main(String[] args) {

SpringApplication.*run*(SslServerApplication.class, args);

}

}

*@RestController*

class ServerController{

//Used NoSuchAlgorithmExeption to run as Java application

public static String hashFunc(String newData) throws NoSuchAlgorithmException{

MessageDigest md = MessageDigest.*getInstance*("SHA-256");

byte[] biteDigest = md.digest(newData.getBytes(StandardCharsets.***UTF\_8***));

BigInteger stringNum = new BigInteger(1, biteDigest);

StringBuilder stringAsNum = new StringBuilder(stringNum.toString(16));

while(stringAsNum.length() < 32) {

stringAsNum.insert(0,'0');

}

return stringAsNum.toString();

}

*@RequestMapping*("/hash")

public String myHash() throws NoSuchAlgorithmException{

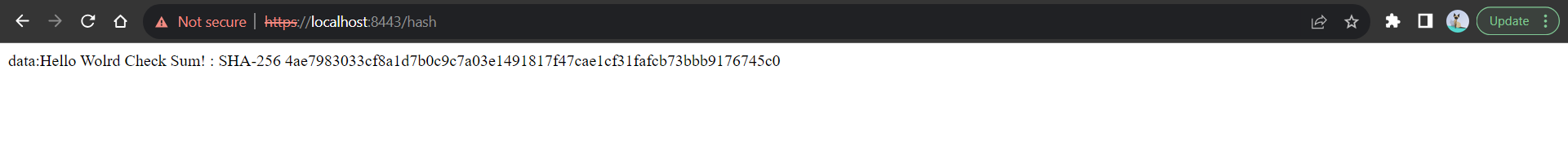
String data = "Hello Wolrd Check Sum!";

String newCheckSum = *hashFunc*(data);

return "<p>data:"+ data + " : SHA-256 " + newCheckSum;

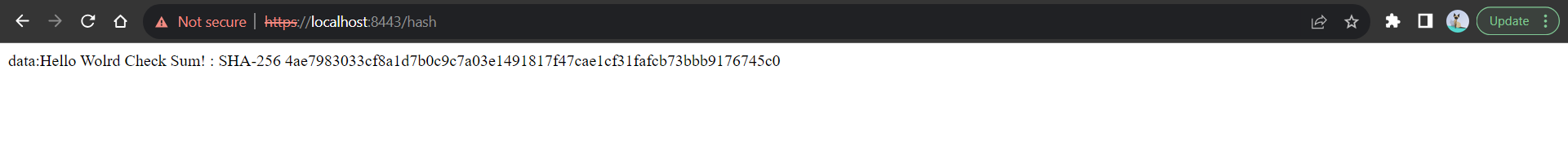
}

## }



## Secure Communications

## Localhost is considered an untrusted webpage, there for is unsecured. Though it is not considered a secure connection the connection was successful. There is an importance to having a secure connection. The connection helps to authenticate a website and helps to invoke trust from an individual user. If done properly the website would show that it is an https and not an http.



## Secondary Testing

## Testing was successful

A screenshot of a computer

Description automatically generated

Mavin Build was also successful.  
A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

Dependency check after refactored code shows that there are no additional vulnerabilities.

A screenshot of a computer

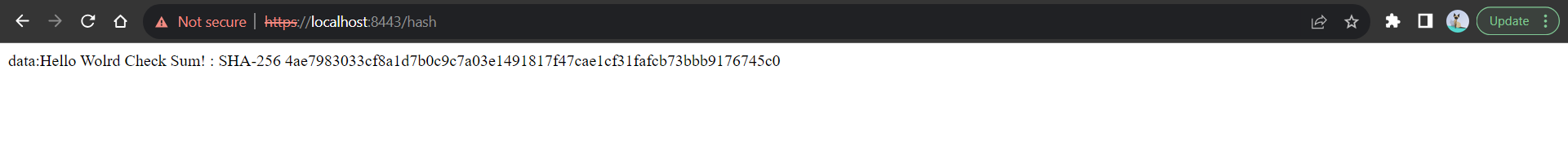
Description automatically generated

## Functional Testing

A screenshot of a computer

Description automatically generated

Everything ran successfully.



## Summary

Following the Vulnerability Assessment Process Flow Diagram, we can see that the code helps to cover a few of these aspects. The first being cryptography. Cryptography is being used when we hash the information. The next is client server. This assessment is when a user tries to access the server, which can be seen when we access local host. Code error and code quality are also used. We can see this based on the secure connections that are shown using a certificate.

There are many ways to add layers of security to the code we are developing. The first was using cryptography and hashing the data that goes over the server for a user to see. The data is considered not visible without the key to decrypt it. Another layer of security is adding a certificate. Whether it is done by a CA or if it is self-signed and then registered, this proves to the user that what they are viewing is safe and can be trusted. It allows proof of authenticity.

## Industry Standard Best Practices

Industry best practices cover a lot of different things. When it comes to security, one of the practices was to use cryptography to hide data from prying eyes. In this case we hashed the data. We also incorporated a certificate to authenticate that the data being received is safe and malicious. Within our code we created a self-signed certificate instead of purchasing one from a CA.

Using these practices not only helps the user but also helps the company’s wellbeing. When a company utilizes these practices, their clients begin to grow trust. This trust allows the company to grow further clients as well as keep their current. Basically, the more trust that is earned the better the business becomes.

**Citations**

Callaghan, P. (n.d.). *Why you should use SHA-256 in evidence authentication*. Pagefreezer Blog. https://blog.pagefreezer.com/sha-256-benefits-evidence-authentication

Denishev, V., About The AuthorVladislav Denishev is an independent expert, & Author, A. T. (2017, June 12). *The Complete Guide to switching from HTTP TO HTTPS*. Smashing Magazine. https://www.smashingmagazine.com/2017/06/guide-switching-http-https/

Encyclopædia Britannica, inc. (n.d.). *Cipher*. Encyclopædia Britannica. https://www.britannica.com/topic/cipher

Graham-Cumming, J. (2023, July 5). *Why secure systems require random numbers*. The Cloudflare Blog. https://blog.cloudflare.com/why-randomness-matters/

*Java cryptography - message digest*. Online Courses and eBooks Library. (n.d.). https://www.tutorialspoint.com/java\_cryptography/java\_cryptography\_message\_digest.htm