****

# 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** | **04/17/2023** | **Dre’ Scheetz** | **Refactored and tested code** |

## 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

Dre’ Scheetz

## Algorithm Cipher

SHA-256 was the cipher chosen based on Artemis Financial’s previous desires for minimal collision with the SHA-256 providing minimal likelihood of that event occurring. It is also a cipher prevalent in real world application being used by both Bitcoin and NIST’s recommendation.

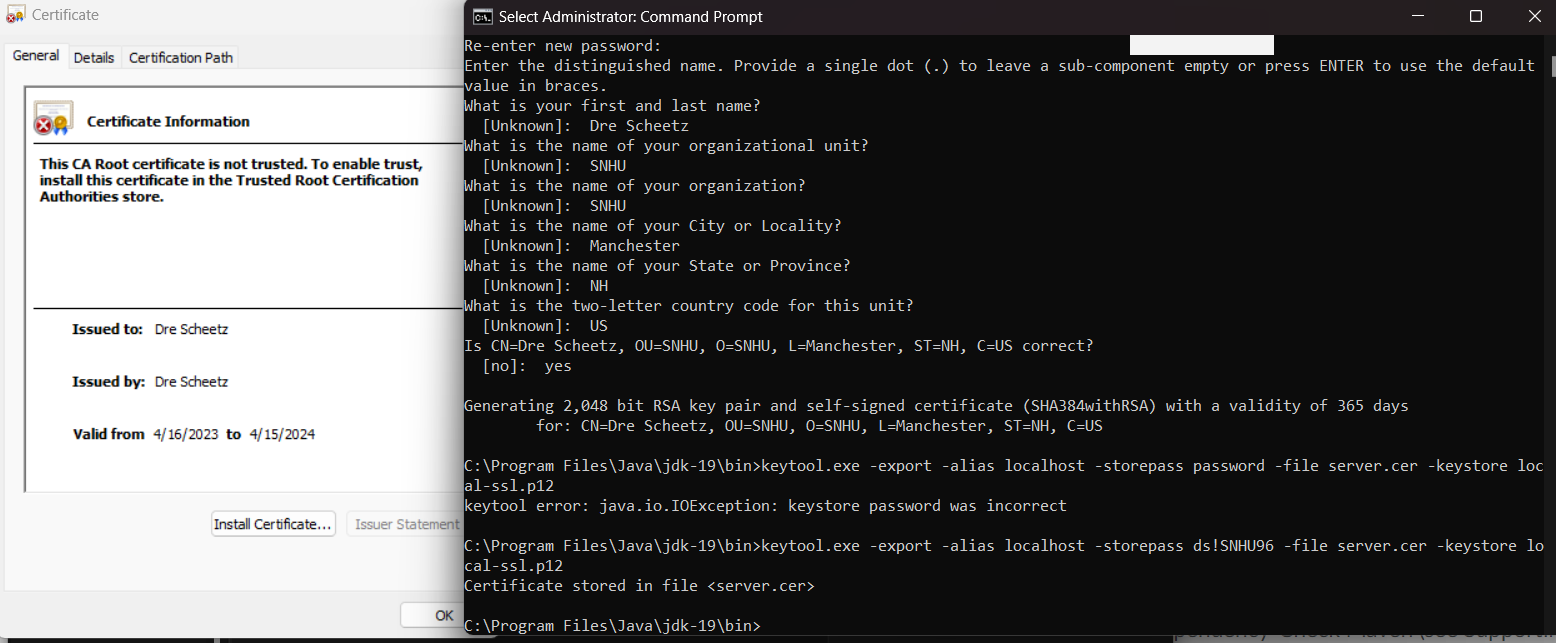
The SHA-256 hash function has a 256-bit output regardless of the input making the output secure. SHA-256 is part of SHA 2, an advancement from the previous iterations of the Secure Hash Algorithm. While different versions of SHA exist with SHA 3 and SHA 2, the SHA 256 benefits from being secure with current technology while also being performative by comparison to SHA 512.

The primary purpose of a hash function is to perform operations on the input data, generally text of any length to a fixed number of bits. The process is not capable of decryption; however, a comparison can be made of the hash outputs to easily identify if the files are equal. With that in mind the hash can be used to verify the integrity of the data and is also useful for authentication. All hash functions output a fixed number of bits in the example of SHA 256 this is 256 bits, but other hash functions produce a variety, commonly 128 or 256.

With symmetric encryption, the key used to encrypt and decrypt is identical. For asymmetric, a public key is used to encrypt while a private key is used to decrypt. Because an asymmetric encryption requires a private key to decrypt, the public key is typically available through web applications. The use of random numbers is used to generate these keys so the keys are less likely to be compromised without direct access.

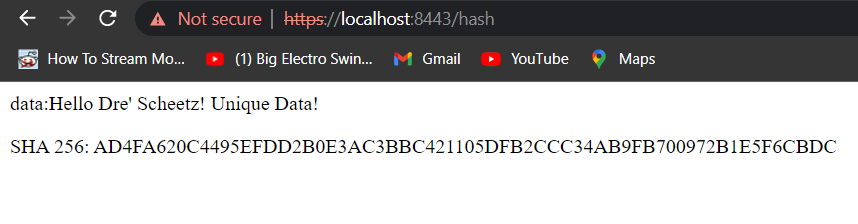
Encryption of data has taken many forms over the course of human history, a common example being the Greek Scytale, where letters were re-arranged using a cylinder and parchment. Much like all forms of security a standard system works until it is compromised. SHA 1 was a former iteration of SHA 2, which is prevalent. Another older hash function that has been compromised is the MD5 originating in 1992. Collisions became problematic as an ordinary home computer could find collisions in 2008 quickly. Collisions can lead to fraud and jeopardize integrity and authenticity.

## Certificate Generation



## Deploy Cipher

Insert a screenshot below of the checksum verification.



## Secure Communications

The web browser was secure aside from having the certificate being self-signed as seen below with the ERR\_CERT\_AUTHORITY\_INVALID. Many attempts were made to circumvent this minor issue; however, none remedied the issue.

Text

Description automatically generated

Text

Description automatically generated

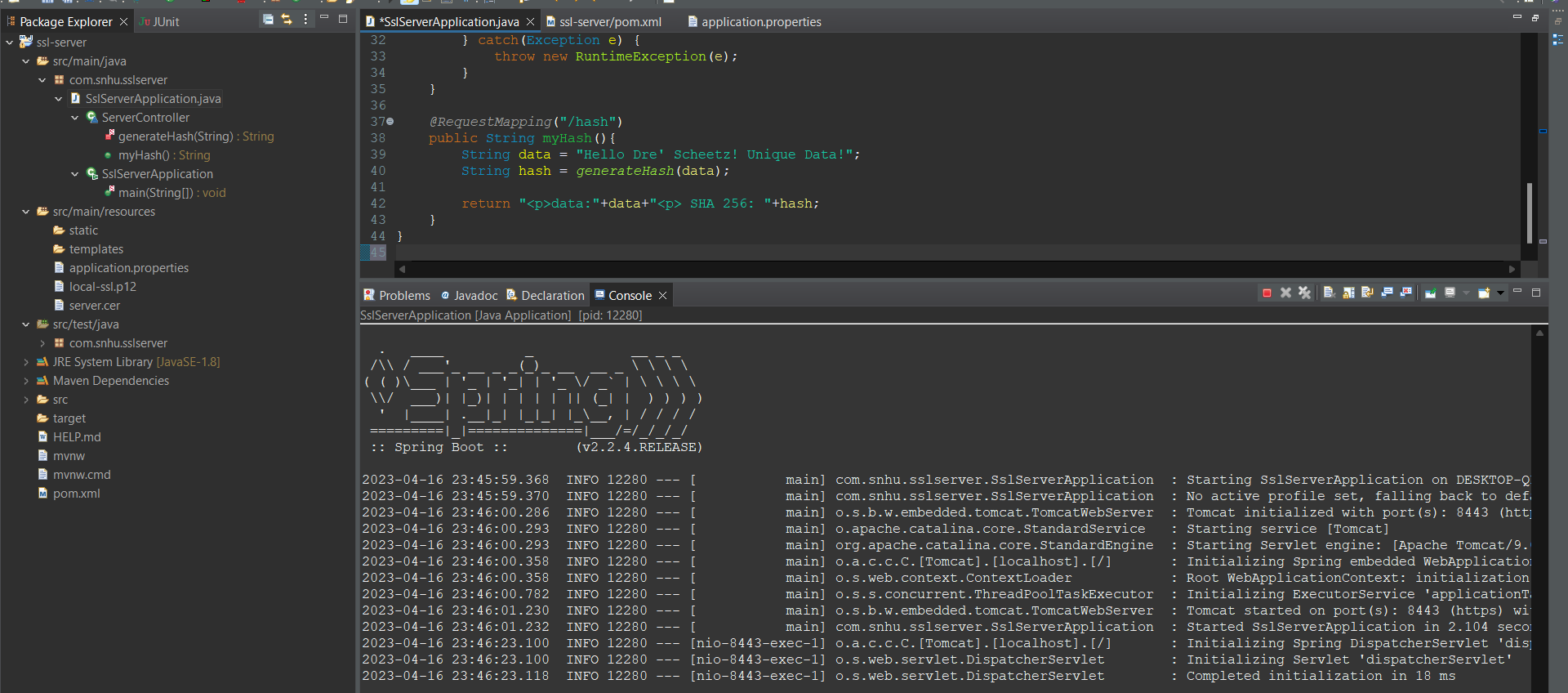
## Secondary Testing

No new security vulnerabilities were introduced with the Cipher or SSL Certificate. As demonstrated by the following links:

<file:///C:/Users/wilde/Desktop/Software%20Engineering/CS305/CS%20305%20Project%20Two%20Code%20Base/ssl-server_student/target/dependency-check-report.html>

<file:///C:/Users/wilde/Desktop/Software%20Engineering/CS305/DreScheetz%20Project%20Two/ssl-server_student/target/AfterReport.html>

Both links may be found in the Project 2 zip file.

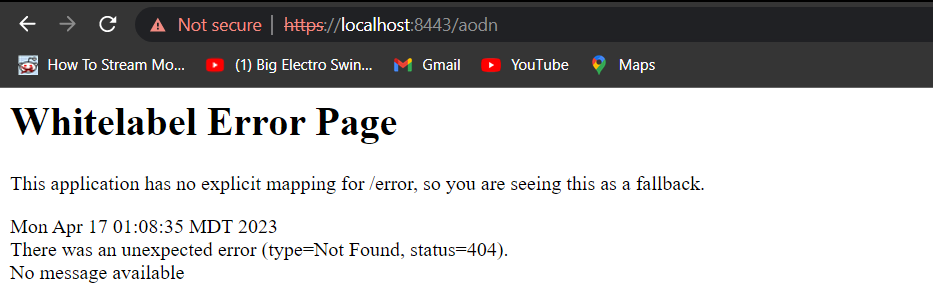




## Functional Testing

Given the instructions provided no security vulnerabilities are evident. The requirements for a static string displayed specifically for localhost:8443/hash have been met. In the future additional security for error pages may be necessary or may not depending if the “Remote Code Execution (RCE) in Java Springboot framework” per Spring Boot RCE exists with the Whitelabel Error Page after updating to the latest versions of the dependencies. In either case proper error handling should extend beyond the added RestController.

Text

Description automatically generated

## Summary

A few important security measures were taken regarding the Vulnerability Assessment Process Flow Diagram. Cryptography was added to the static data required using SHA 256. Error Handling was addressed within the generateHash method of the ServerController class. Code Quality is also present by leaving the hash function’s input as static to prevent insertion. The process used to add these layers of security were to meet the requirements of the project while not bloating the code base, which could lead to potential security risks. Real world examples and governmental agencies were also utilized to ensure the additional security was backed by well proven design patterns.

## Industry Standard Best Practices

Best standards were maintained in several ways, by controlling who has access to the application properties the certificate can be maintained by the company without risk of users unintentionally providing information to a fraudulent party. Using the Maven dependency report. Current security risks can be assessed and dealt with to mitigate potential attacks. To this end the dependency was updated to 8.2.1. Secure connections were also addressed with the IOT and potential remote code execution has been flagged.

Some of the benefits of being aligned with industry best practices is the protection of clients, company data, cost, and to maintain regulation guidelines. By using secure coding patterns and practices during development, cost to the company in both time and actual development costs can be mitigated as problems become more costly to fix the longer the issue is not resolved. Being inline with regulation guidelines can prevent costly legal problems and ensures liability needs are met. The protection of data allows users to trust the company as well as preventing hackers from abusing the system or stealing data for nefarious purposes.

**Reference**

Chitty, T. (2023, February 3). *The Mathematics of Bitcoin — SHA-256 - The Startup - Medium*. Medium. https://medium.com/swlh/the-mathematics-of-bitcoin-74ebf6cefbb0

Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, U.S. Department of Commerce. (n.d.). *NIST Policy on Hash Functions - Hash Functions | CSRC | CSRC*. https://csrc.nist.gov/Projects/Hash-Functions/NIST-Policy-on-Hash-Functions

*Spring Boot RCE*. (n.d.). https://f002.backblazeb2.com/file/sec-news-backup/files/writeup/deadpool.sh/\_2017\_RCE\_Springs\_/index.html