****

# Practices for Secure Software Report

Table of Contents

[Document Revision History 3](#_Toc102040754)

[Client 3](#_Toc102040755)

[Developer 4](#_Toc102040757)

[1. Algorithm Cipher 4](#_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** | **10/15/2023** | **Charles Beavers** |  |

## Client



## Developer

Charles Beavers

## Algorithm Cipher

The SHA-256 algorithm is the cipher I chose for this encryption purpose. It has minimal possibility of collision and is nearly impossible to break with brute force. The cipher produces a 256-bit hash value from eight 32-bit words.

## Certificate Generation

A black screen with a black background

Description automatically generated

## Deploy Cipher

A screenshot of a computer

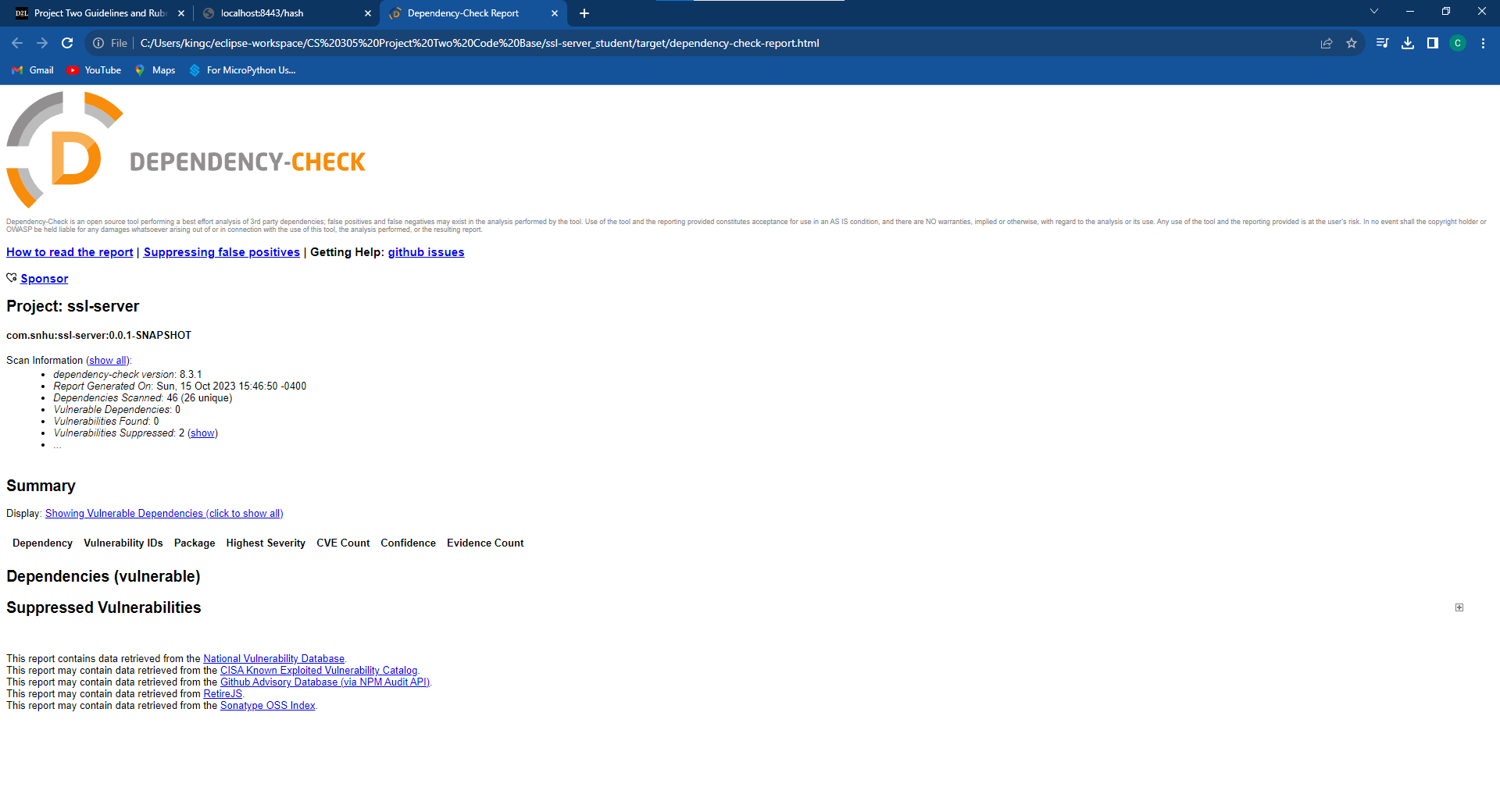
Description automatically generated

## Secure Communications

A screenshot of a computer

Description automatically generated

## Secondary Testing



A screen shot of a computer

Description automatically generated

## Functional Testing

A screen shot of a computer

Description automatically generated

## Summary

For security purposes, the SHA-256 hashing algorithm was implemented. This allowed for secure communication with the server to reduce the risk of data breaches. To further rid the software of vulnerabilities, I updated all dependencies and libraries to the current iteration. This can increase the chance that vendors or creators have solved issues related to the vulnerabilities in question.

API’s, cryptography, code error, and code quality are areas of the Vulnerability Assessment Process Flow Diagram I have addressed in refactoring the code. First, by updating the Spring Framework, the latest API security is being created and used for secure communication. Second, I used the SHA-256 algorithm to encrypt the data being communicated. Third, the refactored code uses secure error handling with an exception thrown if an unknown algorithm is detected. And lastly, I debugged the code to remove any unwanted errors for code quality.

## Industry Standard Best Practices

I have implemented industry standard best practices by using a secure hashing algorithm and

maintaining the latest version of the tools, dependencies, and/or libraries used. Along with this, common vulnerable patterns were avoided to ensure they cannot be utilized as attack points. Avoiding them involved running dependency checks until all vulnerabilities were assessed and gone.