****

# Practices for Secure Software Report

**Table of Contents**

[**Document Revision History 3**](#_heading=h.30j0zll)

[**Client 3**](#_heading=h.1fob9te)

[**Instructions 3**](#_heading=h.3znysh7)

[**Developer 4**](#_heading=h.2et92p0)

[**1. Algorithm Cipher 4**](#_heading=h.tyjcwt)

[**2. Certificate Generation 4**](#_heading=h.3dy6vkm)

[**3. Deploy Cipher 4**](#_heading=h.1t3h5sf)

[**4. Secure Communications 4**](#_heading=h.4d34og8)

[**5. Secondary Testing 4**](#_heading=h.2s8eyo1)

[**6. Functional Testing 4**](#_heading=h.17dp8vu)

[**7. Summary 4**](#_heading=h.3rdcrjn)

[**8. Industry Standard Best Practices 4**](#_heading=h.26in1rg)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **10/20/2023** | **Steven Rodas** | **Final Revision** |

## 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

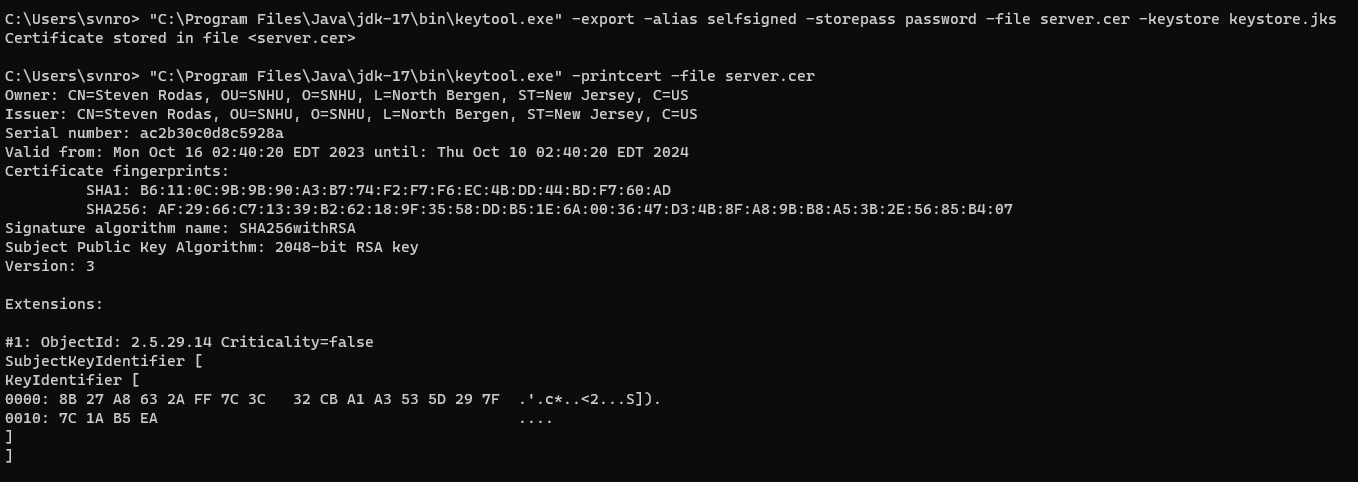
Steven Rodas

## Algorithm Cipher

After researching and sifting through different ciphers, I have decided to use the ‘SHA-256’ algorithm cipher. SHA-256 is one of the most reliable and trusted algorithms in the world for cryptographic hashing. Secure Hash Algorithm 256-bit (SHA-256) is a member of the SHA-2 family of algorithms that was initially developed by the NSA. SHA-256 is possibly the best option for averting collisions. One of the major strengths of SHA-256 is its ability to avoid collisions. Basically, this means it would be difficult to find two unique values that produce the same hash value. This is important as it applies to the safety and security of technologies like digital signatures and certificates of authority. SHA-256 works by taking an input, such as a password or username, and returning a fixed-size string of 256 bits. The output is known as the hash value and it usually contains a string of random numbers, digits, and characters. The idea is that it completely hashes the initial string of data to make it undecipherable and irreversible. SHA-256 as of today is one the most widely used and known algorithm ciphers in the world but as history has shown before, certain algorithms may become obsolete. Because SHA-1 became weaker as time went on and became more prone to attacks, SHA-2 was developed by the NSA to find a solution to this new problem. Due to the ever-changing landscape of technology and cryptography in our modern world, we may see a rise to potentially more secure and suitable algorithm ciphers that exceed the abilities of SHA-256.

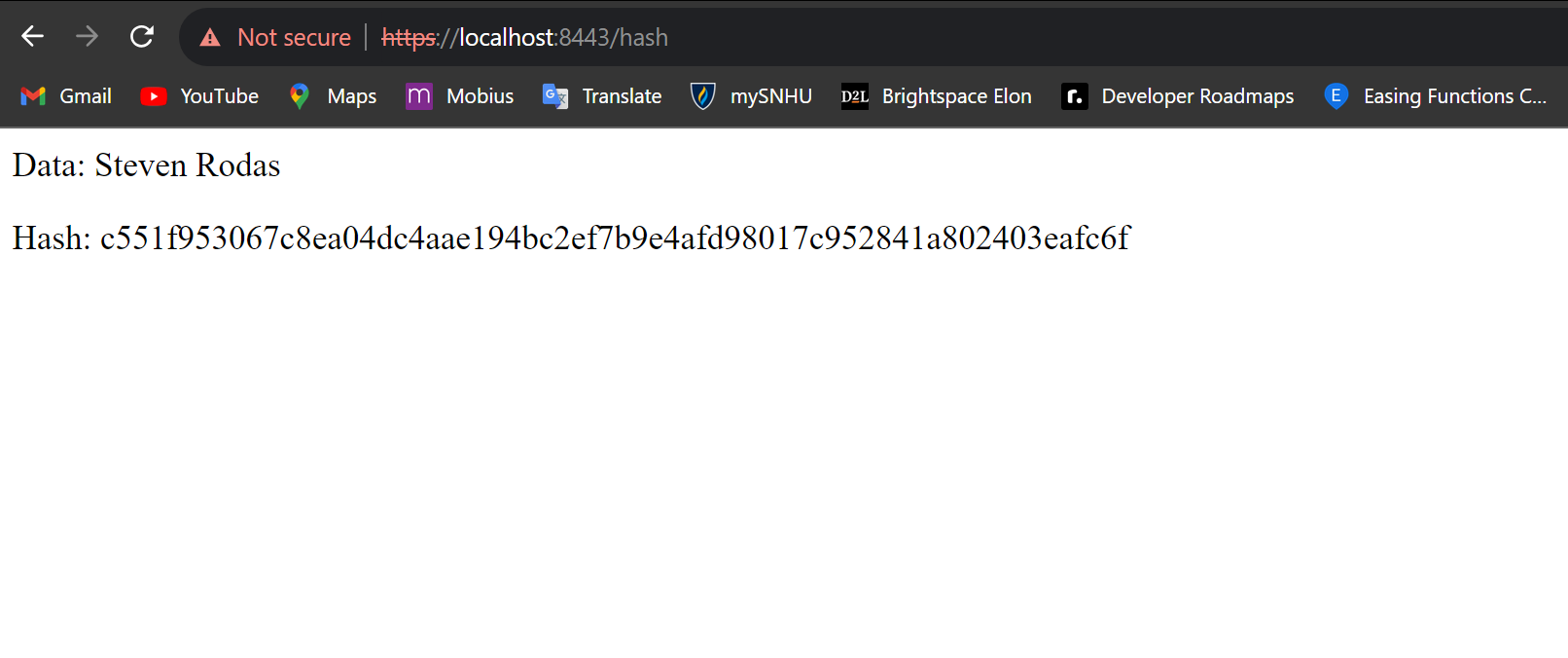
## Certificate Generation

Insert a screenshot below of the CER file.



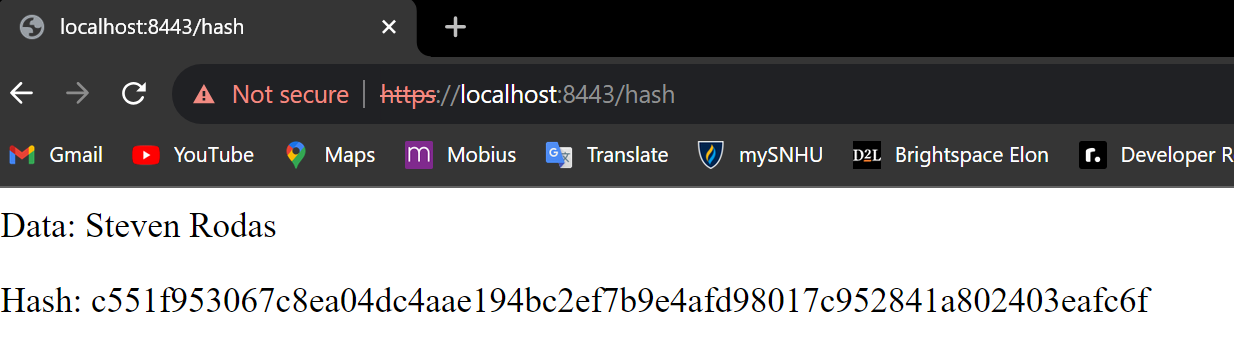
## Deploy Cipher

Insert a screenshot below of the checksum verification.



## Secure Communications

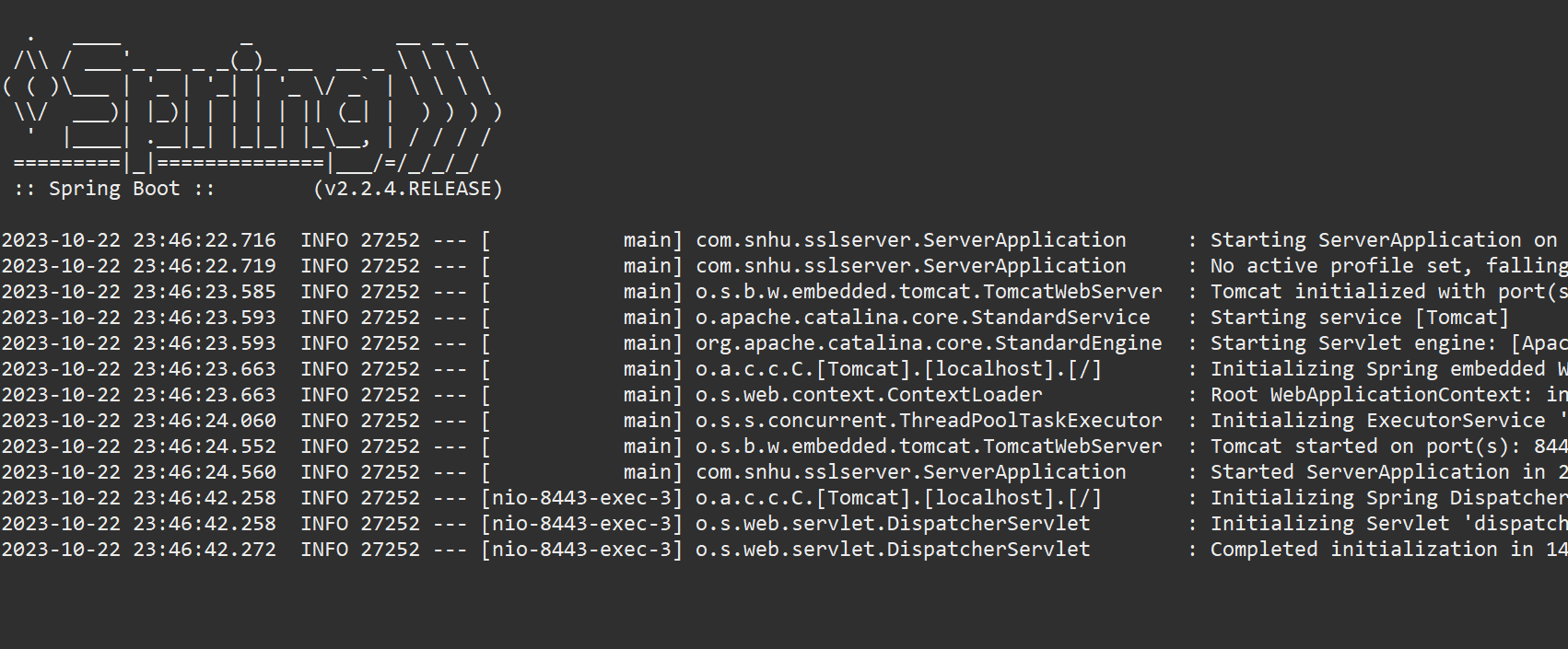
Insert a screenshot below of the web browser that shows a secure webpage.

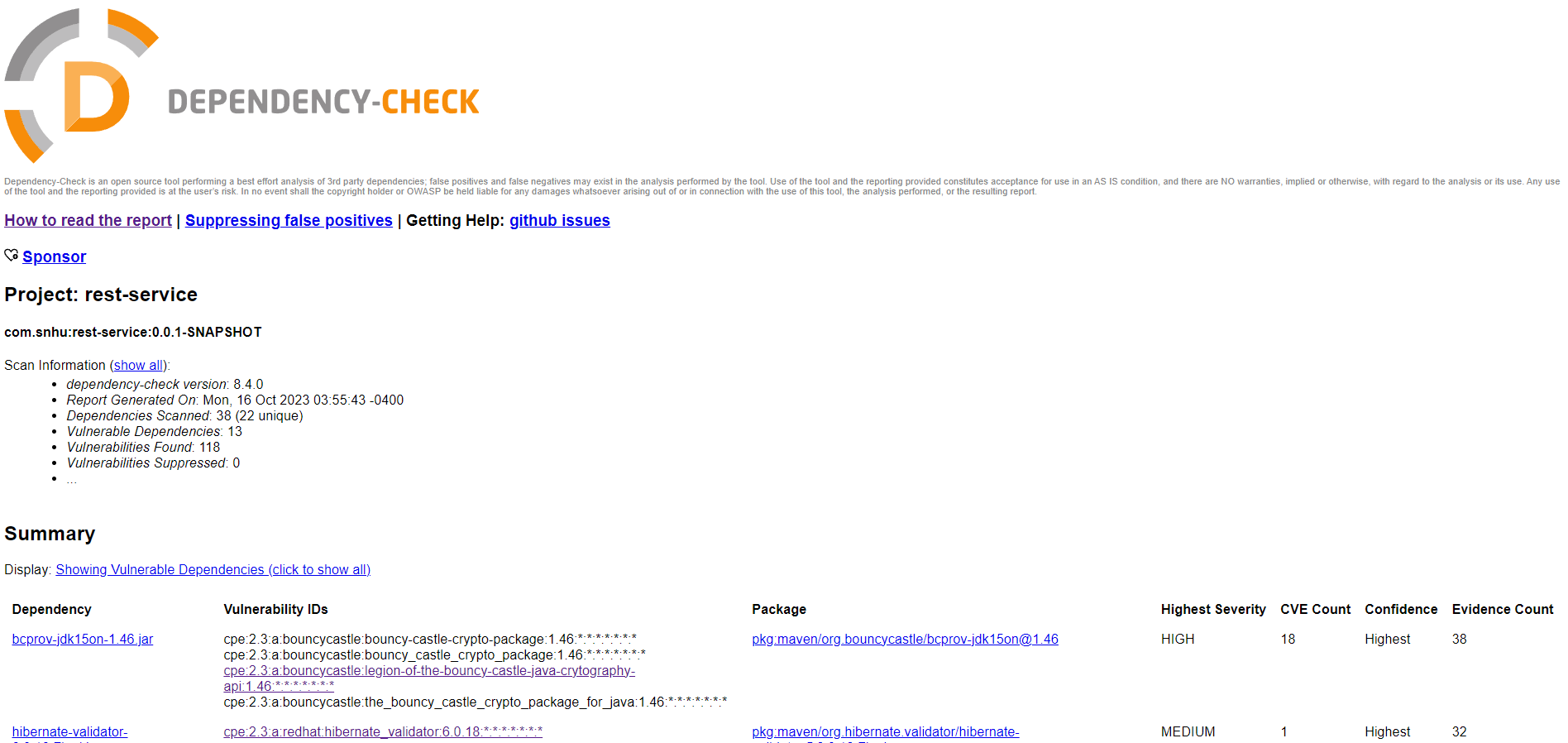


I had a problem with the https being red and marked out but after further research, this is apparently expected behavior from Chrome to my knowledge.

## Secondary Testing

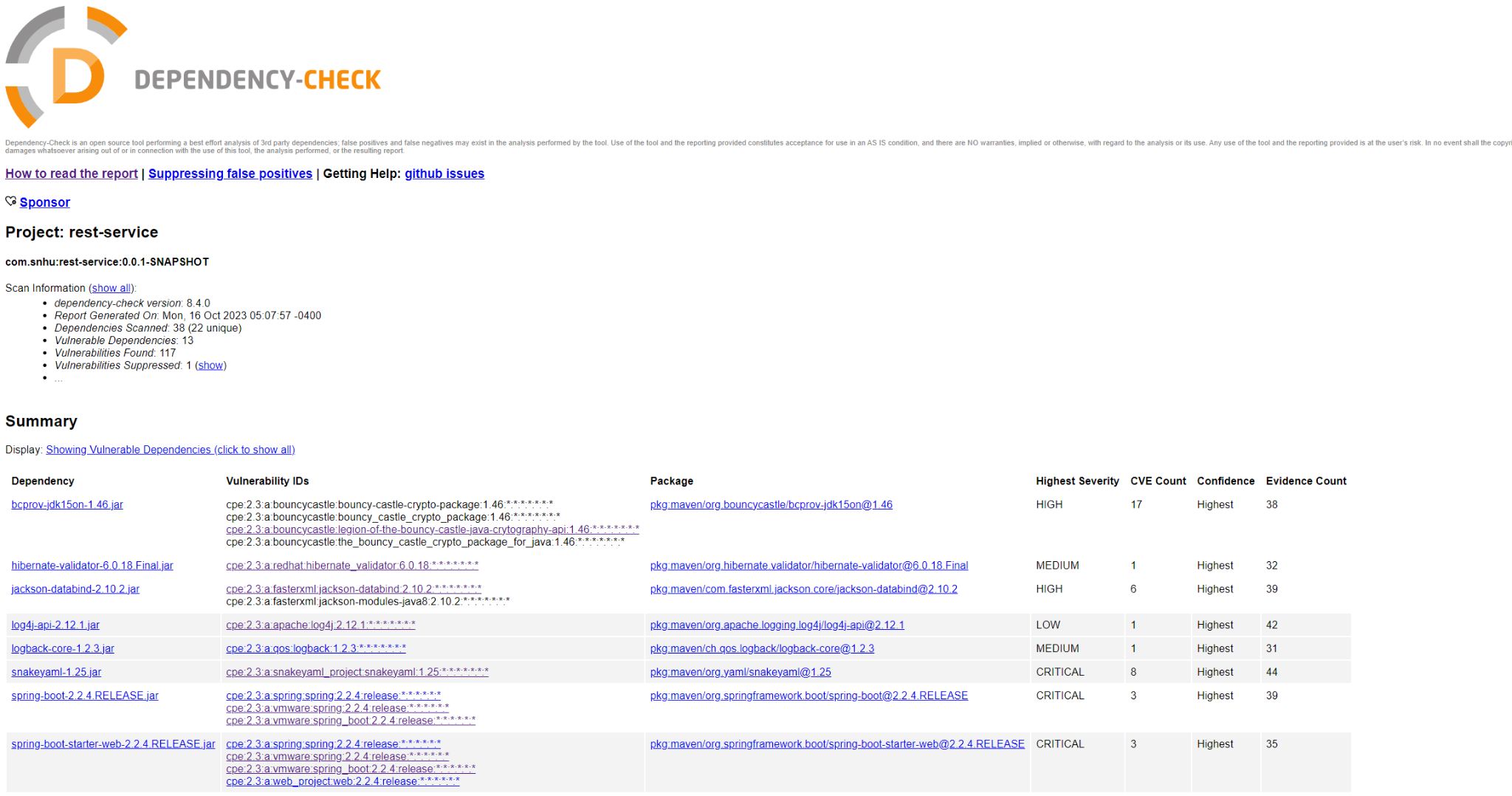
Insert screenshots below of the refactored code executed without errors and the dependency-check report.





## Functional Testing

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



## Summary

The refracted code was developed with the Vulnerability assessment Process Flow diagram in mind. The code has been refactored to secure input validation, secure communications with APIs, encryption use with SHA-256 hashing algorithm, and Client/Server communications to establish a secure connection with both the server and the end user’s browser. This is important since we want a secure and established pipeline for transferring sensitive private data. SHA-256 also provides an additional layer of security by securing the password and connection status behind an irreversible hashing algorithm.

## Industry Standard Best Practices

I ensured the use of Industry Standard Best Practices by reviewing and testing my code. These practices include encryption for securing sensitive data such as passwords/aliases, employing HTTPS to establish a secure connection between server and user, and writing secure code that is clean, readable, and manageable.