GLOBALRAIN

CS 305 Project Two
Practices for Secure Software Report

Table of Contents

Document Revision History

Version	Date	Author	Comments
1.0	4/12/21	Joshua Massey	

Client

Developer
Joshua Massey

1. Algorithm Cipher

In the use case presented by the Artemis Financial, the best solution would involve the use of the SHA-256 cryptographic algorithm. Since the client's web application will be public facing, neither the client making the request not the service will know much about each past what has been provided by the issuing certificate authority. The choice to use 256-bit encryption comes down to the fact that the more bits present in an algorithm, the harder it becomes to crack via methods such as brute force. At this length, it would be nearly impossible to implement a brute force attack against the server. Additionally, there would be no way for the data to be efficiently decrypted as a change to a single byte will result in a drastically different hash value. These characteristics make the proposed solution a prime candidate for a financial application such as this.

2. Certificate Generation

Server Certificate and Keystore Generation

3. Deploy Cipher

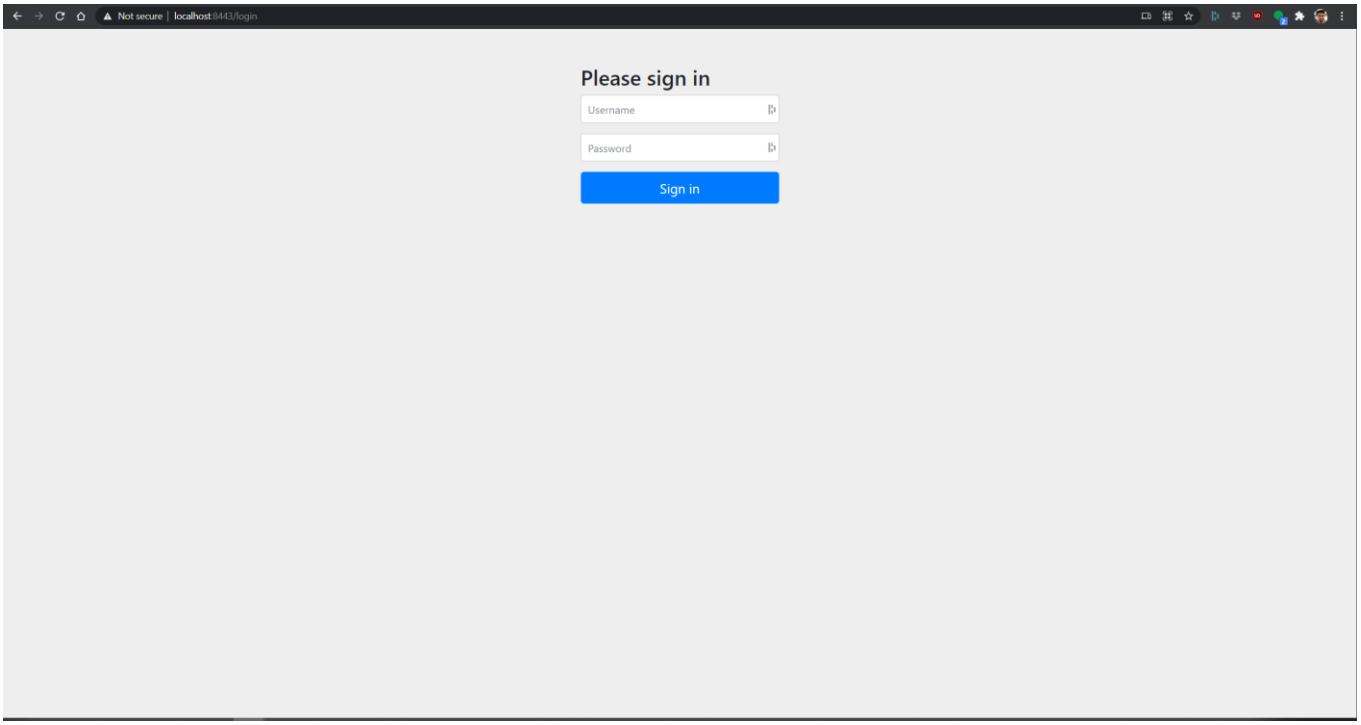
```
@Test
void runTest() {
  System.out.println(controller.helloHash());
}
```

Test Case #1: Calling "/hash" endpoint

```
original data: Joshua Massey
hash value: 6294a71a564575c3376ae48017fbc9c36d4cad7c7b191b7db1c68953d31a6348
checksum: 6294a71a564575c3376ae48017fbc9c36d4cad7c7b191b7db1c68953d31a6348
```

Console Output from Test Case #1

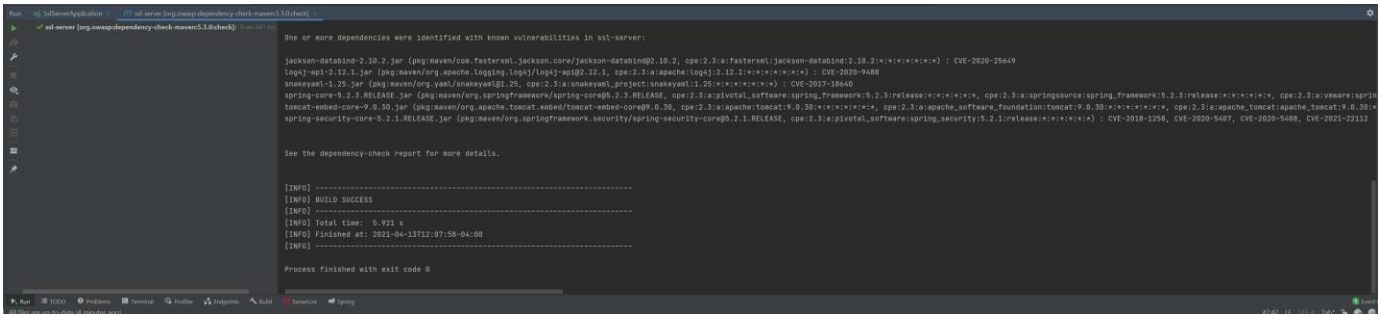
4. Secure Communications



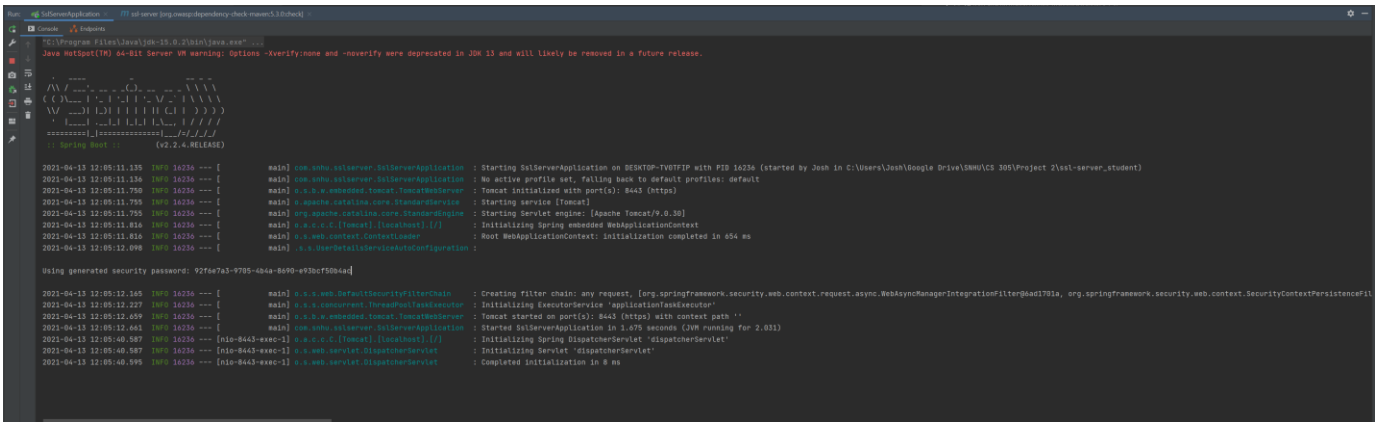
Using HTTPS to Reach Out to Endpoint

Using HTTP to Reach Out to Endpoint

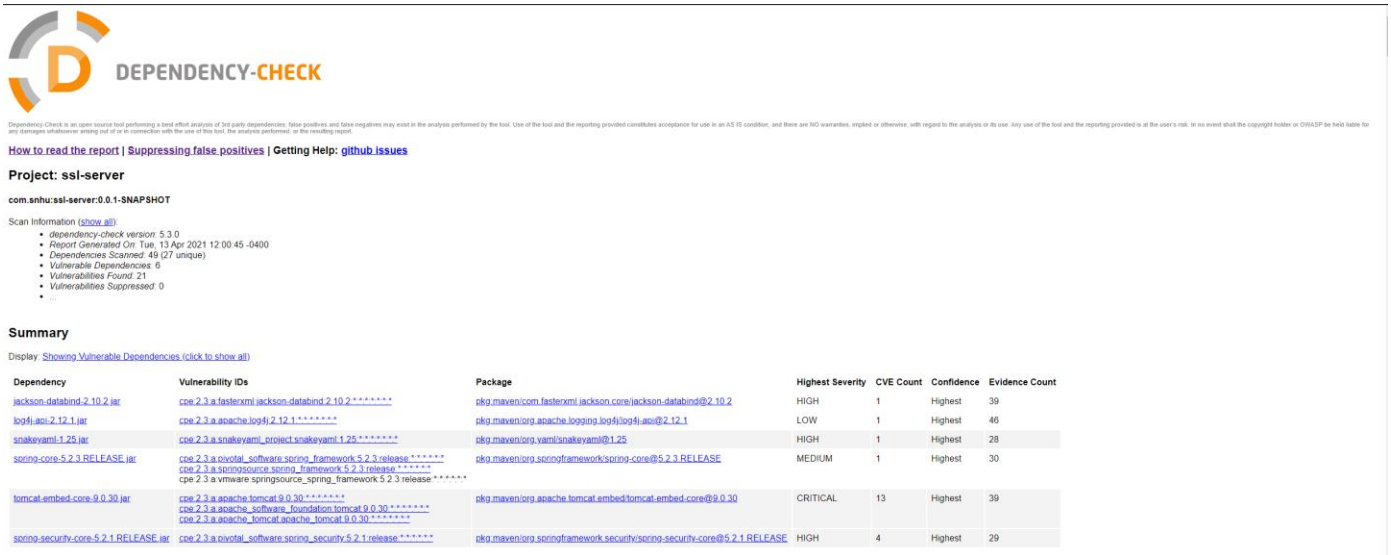
5. Secondary Testing



Dependency Check Runs Successfully

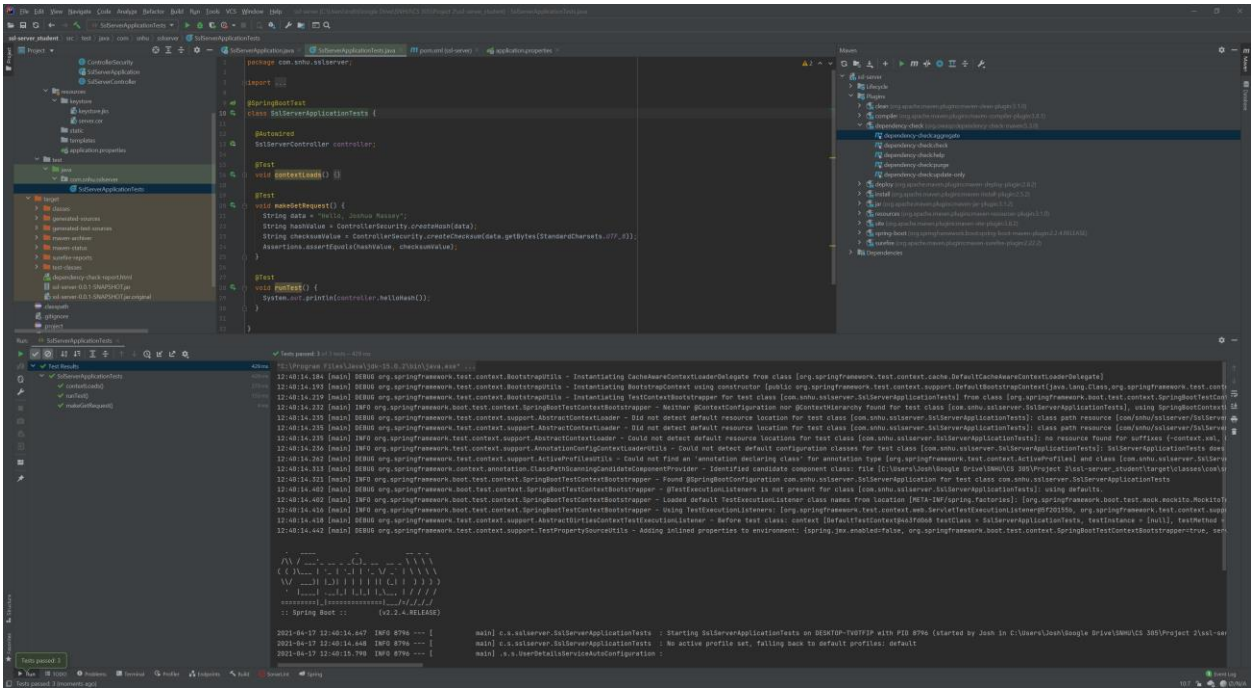


Spring application Starts without Errors



New Dependencies Did Not Introduce New Vulnerabilities

6. Functional Testing



Functional Test Showing Successful Calling of Endpoint and Checksum Validation

7. Summary

With the implementation of HTTPS over TLS, we have added an extra layer of security to the service Artemis wants to deploy. In addition to registering the web-facing with a certificate authority to allow for secure consumer interactions, the login page enabled by this implementation will further protect the client's data from unauthenticated users. By implementing a hashing algorithm, any sensitive data being sent over the network is rendered effectively useless to any interception by a malicious actor. The introduction of the checksum algorithm for verification ensures that, in the case the service is being sent altered data, we can catch it early and avoid exposure to any new attacks. Additional measures have been taken to mitigate the possibility that the service's dependencies cannot be manipulated in order to perform a breach. These measures culminate in a final product for Artemis that assures the principles of cryptography, secure API interactions, error handling, data access, and plug-ins are closely followed.

Submitted with this report is a ZIP file containing the codebase with all the aforementioned implemented security measures as well as the generated dependency report.