GLOBALRAIN

Practices for Secure Software Report

Table of Contents

Document Revision History

Version	Date	Author	Comments
1.0	10/17/2022	Jerry Barboza	

Client

ARTEMIS FINANCIAL

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.

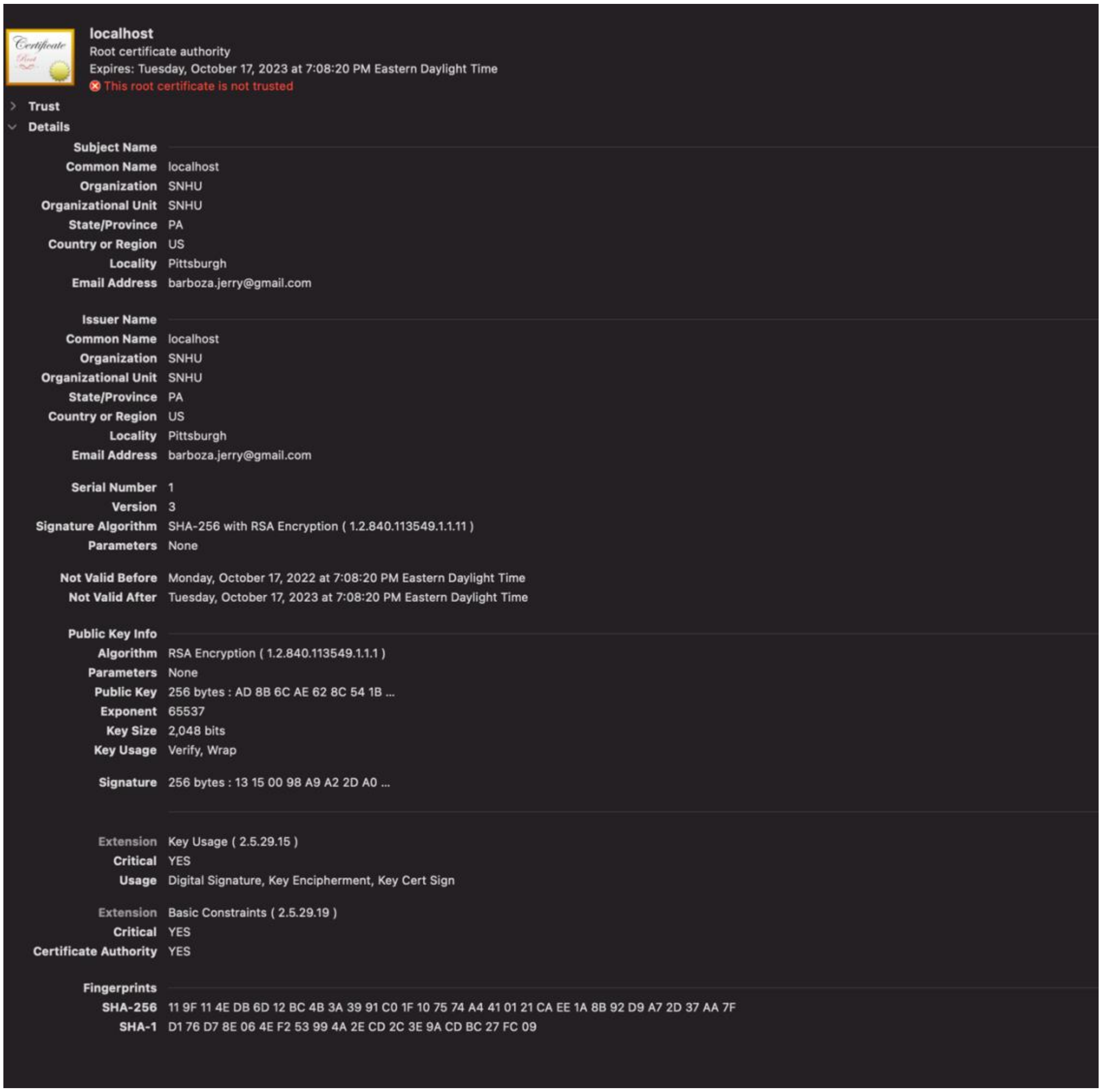
- Respond to the steps outlined below and include your findings.
- Respond using your own words. You may also choose to include images or supporting materials. If you include them, make certain to insert them in all the relevant locations in the document.
- Refer to the Project Two Guidelines and Rubric for more detailed instructions about each section of the template.

Developer
Jerry Barboza

1. Algorithm Cipher

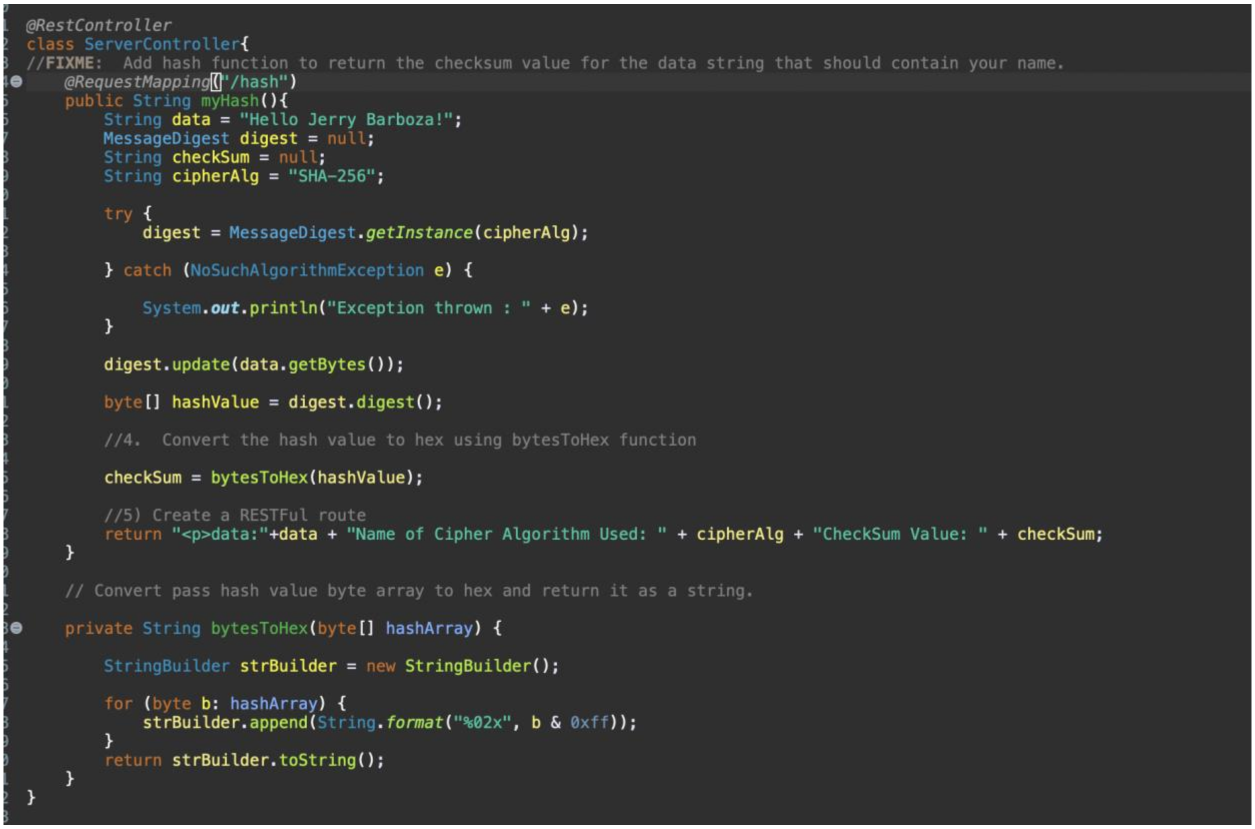
The recommended encryption algorithm cipher that avoids collisions is the SHA-256. If an attacker is using brute-force, it would take 2^ (256) attempts to generate the initial data. SHA-256 is the most secure cipher algorithm therefore it is the recommended encryption algorithm. A collision occurs when two distinct texts produce the same hash and since the SHA-256 has 2^ (256) attempts, it will also take 2^ (256) before a collision occurs. There is no computational power available to crack the SHA-256, hence we don't need to worry about collisions occurring with this cipher algorithm.

2. Certificate Generation



```
localhost
Root certificate authority
Expires: Tuesday, October 17, 2023 at 7:08:20 PM Eastern Daylight Time
This root certificate is not trusted

> Trust
v Details
           Subject Name
            Common Name  localhost
           Organization  SNHU
    Organizational Unit  SNHU
         State/Province  PA
      Country or Region  US
               Locality  Pittsburgh
          Email Address  barboza.jerry@gmail.com

            Issuer Name
            Common Name  localhost
           Organization  SNHU
    Organizational Unit  SNHU
         State/Province  PA
      Country or Region  US
               Locality  Pittsburgh
          Email Address  barboza.jerry@gmail.com

          Serial Number  1
                Version  3
    Signature Algorithm  SHA-256 with RSA Encryption ( 1.2.840.113549.1.1.11 )
             Parameters  None

       Not Valid Before  Monday, October 17, 2022 at 7:08:20 PM Eastern Daylight Time
        Not Valid After  Tuesday, October 17, 2023 at 7:08:20 PM Eastern Daylight Time

        Public Key Info
              Algorithm  RSA Encryption ( 1.2.840.113549.1.1.1 )
             Parameters  None
             Public Key  256 bytes : AD 8B 6C AE 62 8C 54 1B ...
               Exponent  65537
               Key Size  2,048 bits
              Key Usage  Verify, Wrap

              Signature  256 bytes : 13 15 00 98 A9 A2 2D A0 ...

              Extension  Key Usage ( 2.5.29.15 )
               Critical  YES
                  Usage  Digital Signature, Key Encipherment, Key Cert Sign

              Extension  Basic Constraints ( 2.5.29.19 )
               Critical  YES
  Certificate Authority  YES

           Fingerprints
                SHA-256  11 9F 11 4E DB 6D 12 BC 4B 3A 39 91 C0 1F 10 75 74 A4 41 01 21 CA EE 1A 8B 92 D9 A7 2D 37 AA 7F
                  SHA-1  D1 76 D7 8E 06 4E F2 53 99 4A 2E CD 2C 3E 9A CD BC 27 FC 09
```

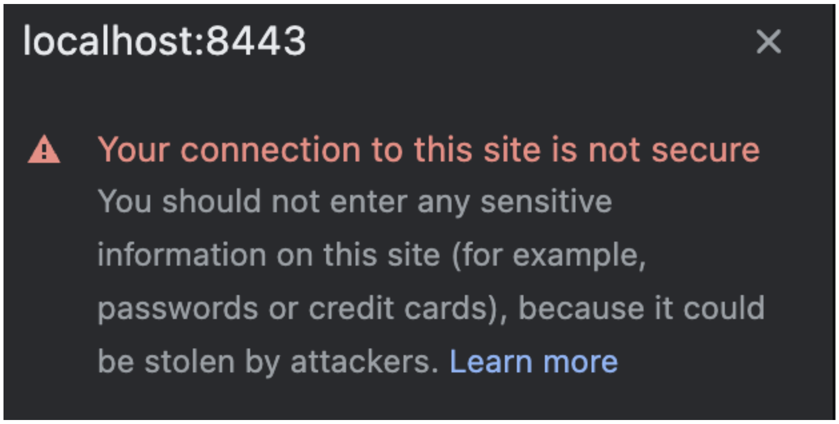
3. Deploy Cipher



```java
@RestController
class ServerController{
//FIXME:  Add hash function to return the checksum value for the data string that should contain your name.
    @RequestMapping("/hash")
    public String myHash(){
        String data = "Hello Jerry Barboza!";
        MessageDigest digest = null;
        String checkSum = null;
        String cipherAlg = "SHA-256";

        try {
            digest = MessageDigest.getInstance(cipherAlg);

        } catch (NoSuchAlgorithmException e) {

            System.out.println("Exception thrown : " + e);
        }

        digest.update(data.getBytes());

        byte[] hashValue = digest.digest();

        //4.  Convert the hash value to hex using bytesToHex function

        checkSum = bytesToHex(hashValue);

        //5) Create a RESTFul route
        return "<p>data:"+data + "Name of Cipher Algorithm Used: " + cipherAlg + "CheckSum Value: " + checkSum;
    }

    // Convert pass hash value byte array to hex and return it as a string.

    private String bytesToHex(byte[] hashArray) {

        StringBuilder strBuilder = new StringBuilder();

        for (byte b: hashArray) {
            strBuilder.append(String.format("%02x", b & 0xff));
        }
        return strBuilder.toString();
    }
}
```

4. Secure Communications

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



localhost:8443

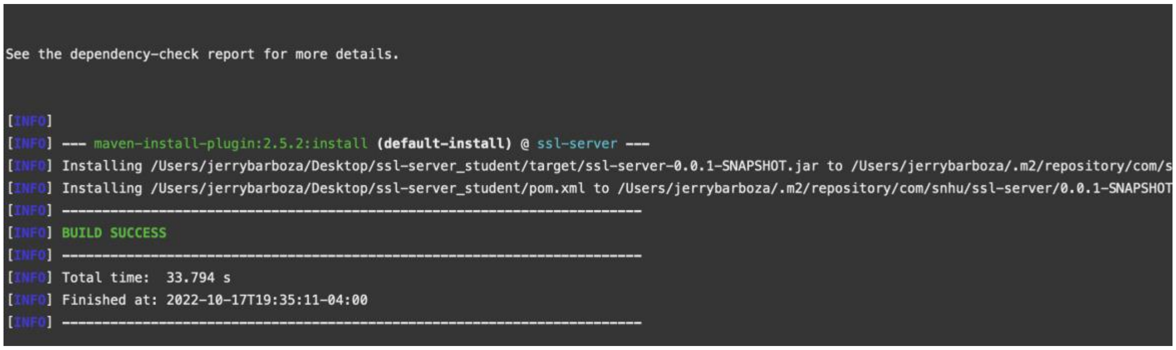
Your connection to this site is not secure

You should not enter any sensitive information on this site (for example, passwords or credit cards), because it could be stolen by attackers. Learn more

(Still working on fixing this in the code)

5. Secondary Testing

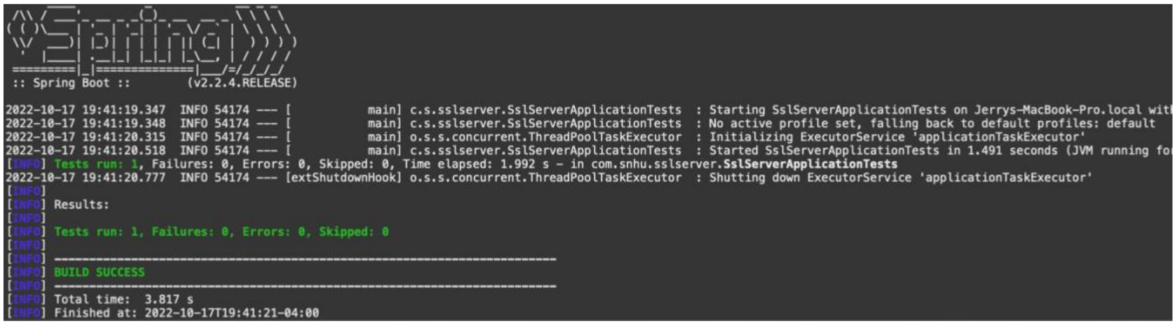
Insert screenshots below of the refactored code executed without errors and the dependency-check report.



```
See the dependency-check report for more details.


[INFO]
[INFO] --- maven-install-plugin:2.5.2:install (default-install) @ ssl-server ---
[INFO] Installing /Users/jerrybarboza/Desktop/ssl-server_student/target/ssl-server-0.0.1-SNAPSHOT.jar to /Users/jerrybarboza/.m2/repository/com/s
[INFO] Installing /Users/jerrybarboza/Desktop/ssl-server_student/pom.xml to /Users/jerrybarboza/.m2/repository/com/snhu/ssl-server/0.0.1-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  33.794 s
[INFO] Finished at: 2022-10-17T19:35:11-04:00
[INFO] ------------------------------------------------------------------------
```

6. Functional Testing

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



```
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.2.4.RELEASE)

2022-10-17 19:41:19.347  INFO 54174 --- [           main] c.s.sslserver.SslServerApplicationTests  : Starting SslServerApplicationTests on Jerrys-MacBook-Pro.local wit
2022-10-17 19:41:19.348  INFO 54174 --- [           main] c.s.sslserver.SslServerApplicationTests  : No active profile set, falling back to default profiles: default
2022-10-17 19:41:20.315  INFO 54174 --- [           main] o.s.s.concurrent.ThreadPoolTaskExecutor  : Initializing ExecutorService 'applicationTaskExecutor'
2022-10-17 19:41:20.518  INFO 54174 --- [           main] c.s.sslserver.SslServerApplicationTests  : Started SslServerApplicationTests in 1.491 seconds (JVM running fo
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.992 s - in com.snhu.sslserver.SslServerApplicationTests
2022-10-17 19:41:20.777  INFO 54174 --- [extShutdownHook] o.s.s.concurrent.ThreadPoolTaskExecutor  : Shutting down ExecutorService 'applicationTaskExecutor'
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  3.817 s
[INFO] Finished at: 2022-10-17T19:41:21-04:00
```

7. Summary

Additional layers of security were used on this software application. We started by adding a Certificate of Authentication (CA) and then we deployed a cipher by implementing the cryptographic hash algorithm by refactoring the code. This cipher algorithm that was used is the SHA-256 as mentioned in Step 1 and Step 3. We then secure communications by refactoring the code to convert HTTP to the HTTPS protocol. Then for the second testing, I ran a secondary static testing of the refactored code by using the OWASP Dependency-check Maven to ensure the code compiles with software security enhancements.

8. Industry Standard Best Practices

Certificate authorities (CA) are very important for security since it is a trusted organization that verifies websites. There are a lot of malicious fake websites created by hackers and these websites try to copy other popular websites to steal information. Hackers can create fake websites that look identical to other ones however the real one will have its own certificate authorization making it trustworthy and safe to use. By looking at the certificate we can see if that website is really who they claimed to be. Applying industry standard best practices to secure coding to the company's overall wellbeing is very important and a priority to maintain the data of the users safe and make it a safe place for users to use.