GLOBALRAIN

Practices for Secure Software Report

Table of Contents

Document Revision History

Version	Date	Author	Comments
1.0	12/4/2023	Jeba Singh Emmanuel	

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
Jeba Singh Emmanuel

1. Algorithm Cipher

Of the available message authentication classes listed I recommend the HMACSHA256. It has a large enough output size to prevent collisions and provides randomness. It has not been compromised yet and provides fixed size out put.

Provide a brief, high-level overview of the encryption algorithm cipher

HMACSHA256 is a keyed hash algorithm that is uses the SHA-256 hash algo and is used as a Hash-based Message Authentication code.

Discuss the hash functions and bit levels of the cipher.

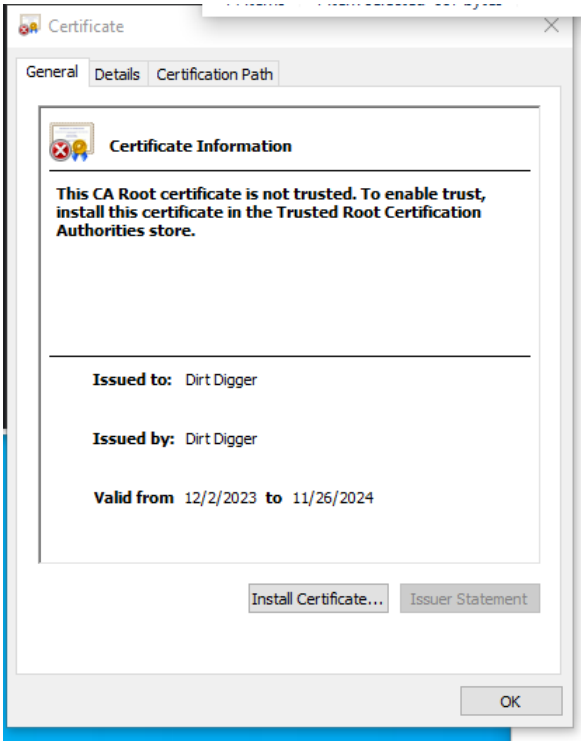
mixes a secret key with message data, hashes the result with a hash function, mixes that hash value with the secret key again, then applies the hash function a second time. I returns an output that is 256 bits in length.

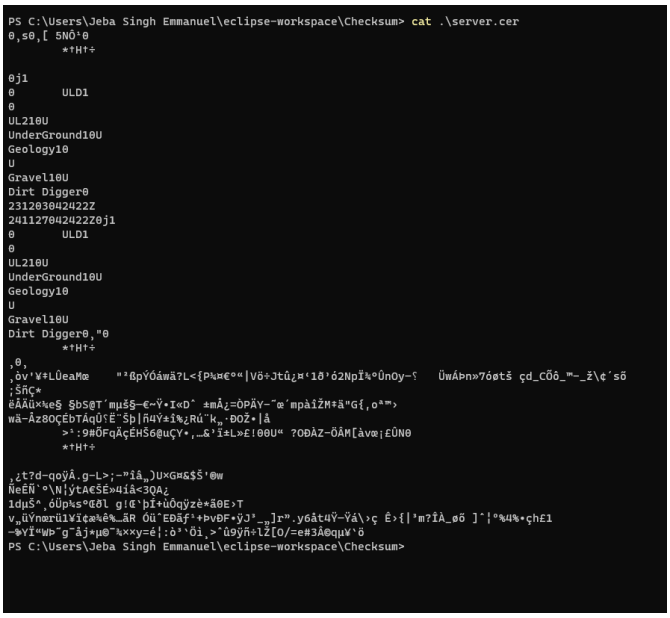
Explain the use of random numbers, symmetric versus non-symmetric keys, and so on.

The algorithm accepts keys of any size and returns a same sized output. It uses symmetric keys since the purpose is signing and signing is one way. In order to have truly random output the keys need to be fairly random. Therefore a cryptographically strong pseudo-random generator seeded with a random seed is needed for key generation. The actual hash generation needs to avoid any randomness (given a secret key and plain text) and need to be reproducible.

2. Certificate Generation

Insert a screenshot below of the CER file.

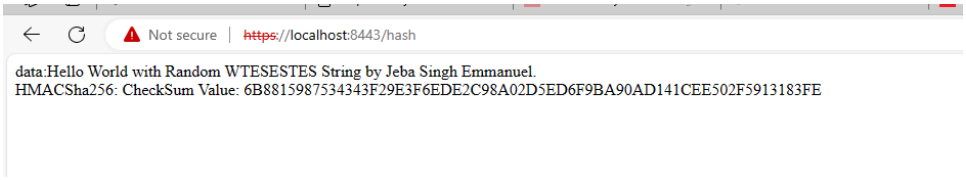




[Insert screenshots here.]

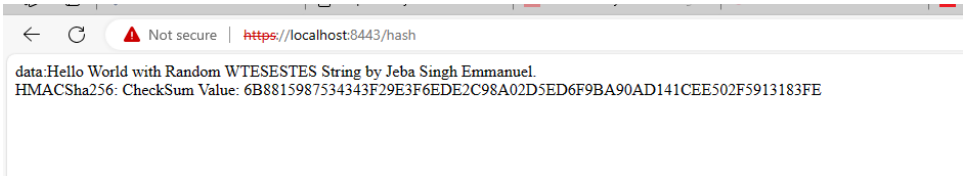
3. Deploy Cipher

Insert a screenshot below of the checksum verification.



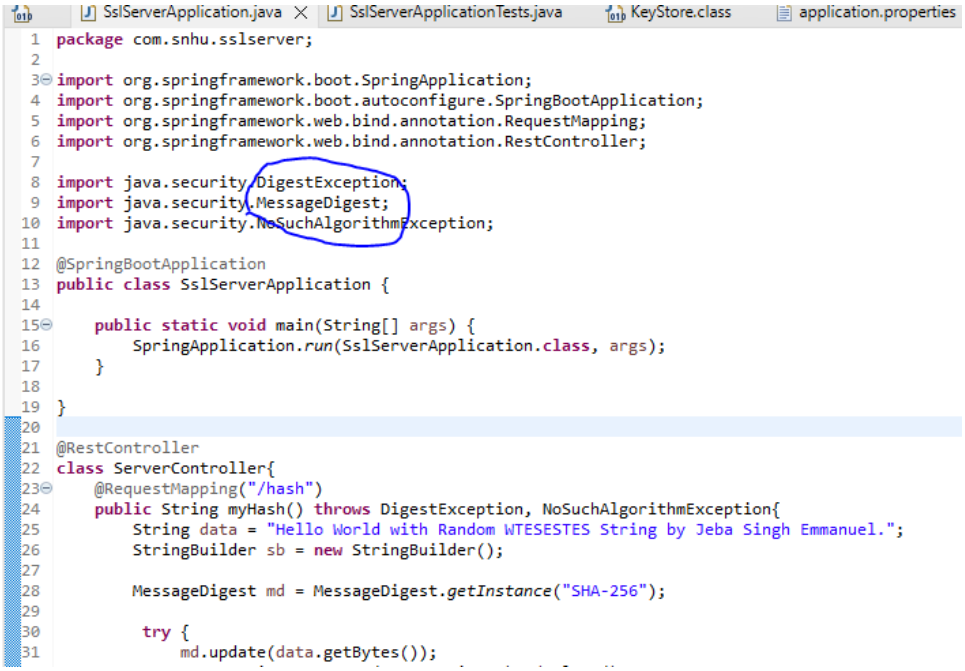
4. Secure Communications

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



5. Secondary Testing

We did not introduce any dependencies. Maven OWASP Dependency checks do not show any vulnerabilities for the new imports we introduced while refactoring code (MessageDigest)



Summary

Display: Showing Vulnerable Dependencies (click to show all)

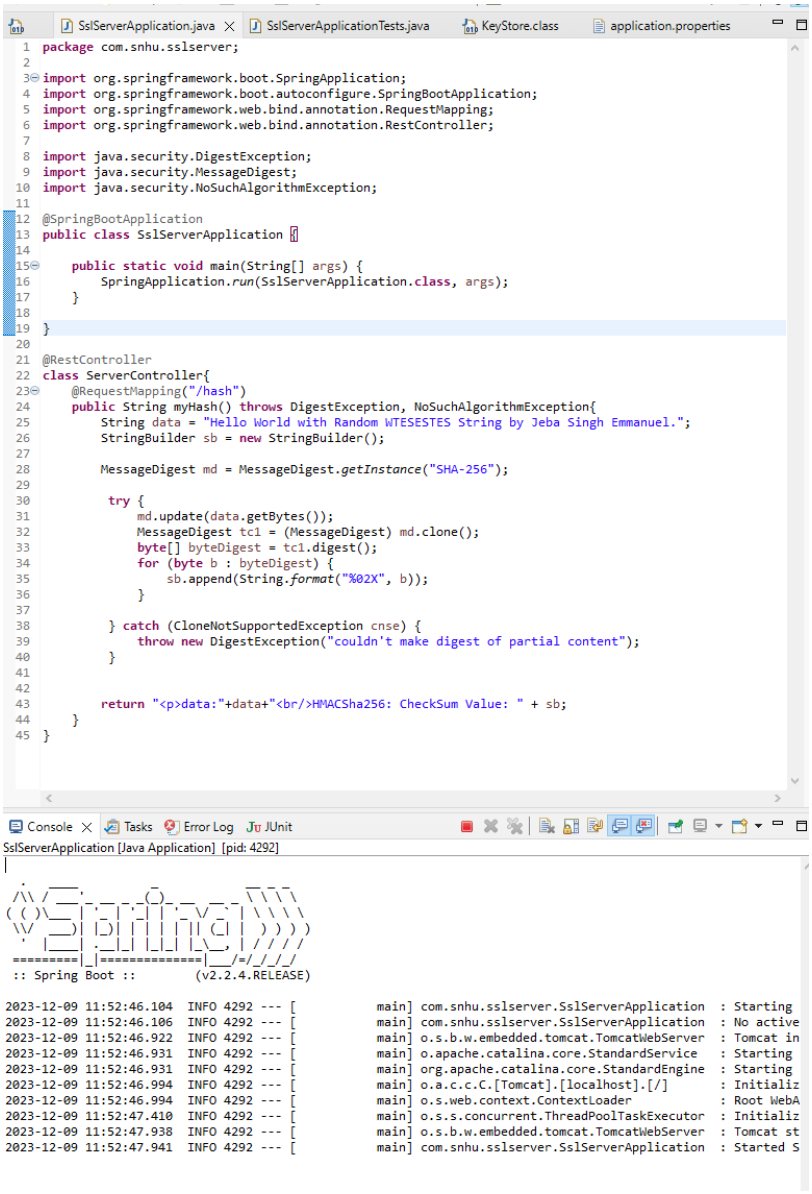
Dependency	Vulnerability IDs	Package
spring-boot-starter-data-rest-2.2.4.RELEASE.jar	cpe:2.3:a:vmware:spring_boot:2.2.4:release:*:*:*:*:*:*:*:* cpe:2.3:a:vmware:spring_data_rest:2.2.4:release:*:*:*:*:*:*:*:*	pkg:maven/or
spring-data-rest-webmvc-3.2.4.RELEASE.jar	cpe:2.3:a:pivotal_software:spring_data_rest:3.2.4:release:*:*:*:*:*:*:*:* cpe:2.3:a:vmware:spring_data_rest:3.2.4:release:*:*:*:*:*:*:*:*	pkg:maven/or
spring-hateoas-1.0.3.RELEASE.jar	cpe:2.3:a:vmware:spring_hateoas:1.0.3:release:*:*:*:*:*:*:*:*	pkg:maven/or
jackson-databind-2.10.2.jar	cpe:2.3:a:fasterxml:jackson-databind:2.10.2:*:*:*:*:*:*:*:*	pkg:maven/cc
spring-boot-2.2.4.RELEASE.jar	cpe:2.3:a:vmware:spring_boot:2.2.4:release:*:*:*:*:*:*:*:*	pkg:maven/or
logback-core-1.2.3.jar	cpe:2.3:a:qos:logback:1.2.3:*:*:*:*:*:*:*:*	pkg:maven/ch
log4j-api-2.12.1.jar	cpe:2.3:a:apache:log4j:2.12.1:*:*:*:*:*:*:*:*	pkg:maven/or
snakeyaml-1.25.jar	cpe:2.3:a:snakeyaml_project:snakeyaml:1.25:*:*:*:*:*:*:*:* cpe:2.3:a:yaml_project:yaml:1.25:*:*:*:*:*:*:*:*	pkg:maven/or
tomcat-embed-core-9.0.30.jar	cpe:2.3:a:apache:tomcat:9.0.30:*:*:*:*:*:*:*:* cpe:2.3:a:apache_tomcat:apache_tomcat:9.0.30:*:*:*:*:*:*:*:*	pkg:maven/or
hibernate-validator-6.0.18.Final.jar	cpe:2.3:a:redhat:hibernate_validator:6.0.18:*:*:*:*:*:*:*:*	pkg:maven/or
spring-web-5.2.3.RELEASE.jar	cpe:2.3:a:pivotal_software:spring_framework:5.2.3:release:*:*:*:*:*:*:*:* cpe:2.3:a:springsource:spring_framework:5.2.3:release:*:*:*:*:*:*:*:*	pkg:maven/or
spring-beans-5.2.3.RELEASE.jar	cpe:2.3:a:pivotal_software:spring_framework:5.2.3:release:*:*:*:*:*:*:*:* cpe:2.3:a:springsource:spring_framework:5.2.3:release:*:*:*:*:*:*:*:*	pkg:maven/or
spring-webmvc-5.2.3.RELEASE.jar	cpe:2.3:a:pivotal_software:spring_framework:5.2.3:release:*:*:*:*:*:*:*:* cpe:2.3:a:springsource:spring_framework:5.2.3:release:*:*:*:*:*:*:*:*	pkg:maven/or
spring-context-5.2.3.RELEASE.jar	cpe:2.3:a:pivotal_software:spring_framework:5.2.3:release:*:*:*:*:*:*:*:* cpe:2.3:a:springsource:spring_framework:5.2.3:release:*:*:*:*:*:*:*:*	pkg:maven/or
spring-expression-5.2.3.RELEASE.jar	cpe:2.3:a:pivotal_software:spring_framework:5.2.3:release:*:*:*:*:*:*:*:* cpe:2.3:a:springsource:spring_framework:5.2.3:release:*:*:*:*:*:*:*:*	pkg:maven/or
json-path-2.4.0.jar	cpe:2.3:a:json-java_project:json-java:2.4.0:*:*:*:*:*:*:*:*	pkg:maven/cc
json-smart-2.3.jar	cpe:2.3:a:json-smart_project:json-smart:2.3:*:*:*:*:*:*:*:*	pkg:maven/ne

Dependencies

spring-boot-starter-data-rest-2.2.4.RELEASE.jar

6. Functional Testing

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



SslServerApplication.java | SslServerApplicationTests.java | KeyStore.class | application.properties

```
1  package com.snhu.sslserver;
2
3  import org.springframework.boot.SpringApplication;
4  import org.springframework.boot.autoconfigure.SpringBootApplication;
5  import org.springframework.web.bind.annotation.RequestMapping;
6  import org.springframework.web.bind.annotation.RestController;
7
8  import java.security.DigestException;
9  import java.security.MessageDigest;
10 import java.security.NoSuchAlgorithmException;
11
12 @SpringBootApplication
13 public class SslServerApplication {
14
15     public static void main(String[] args) {
16         SpringApplication.run(SslServerApplication.class, args);
17     }
18
19 }
20
21 @RestController
22 class ServerController{
23     @RequestMapping("/hash")
24     public String myHash() throws DigestException, NoSuchAlgorithmException{
25         String data = "Hello World with Random WTESESTES String by Jeba Singh Emmanuel.";
26         StringBuilder sb = new StringBuilder();
27
28         MessageDigest md = MessageDigest.getInstance("SHA-256");
29
30          try {
31              md.update(data.getBytes());
32              MessageDigest tc1 = (MessageDigest) md.clone();
33              byte[] byteDigest = tc1.digest();
34              for (byte b : byteDigest) {
35                  sb.append(String.format("%02X", b));
36              }
37
38          } catch (CloneNotSupportedException cnse) {
39              throw new DigestException("couldn't make digest of partial content");
40          }
41
42
43         return "<p>data:"+data+"<br/>HMACSha256: CheckSum Value: " + sb;
44     }
45 }
```

Console | Tasks | Error Log | JUnit

SslServerApplication [Java Application] [pid: 4292]

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

2023-12-09 11:52:46.104  INFO 4292 --- [           main] com.snhu.sslserver.SslServerApplication  : Starting
2023-12-09 11:52:46.106  INFO 4292 --- [           main] com.snhu.sslserver.SslServerApplication  : No active
2023-12-09 11:52:46.922  INFO 4292 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat in
2023-12-09 11:52:46.931  INFO 4292 --- [           main] o.apache.catalina.core.StandardService   : Starting
2023-12-09 11:52:46.931  INFO 4292 --- [           main] org.apache.catalina.core.StandardEngine  : Starting
2023-12-09 11:52:46.994  INFO 4292 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializ
2023-12-09 11:52:46.994  INFO 4292 --- [           main] o.s.web.context.ContextLoader            : Root WebA
2023-12-09 11:52:47.410  INFO 4292 --- [           main] o.s.s.concurrent.ThreadPoolTaskExecutor  : Initializ
2023-12-09 11:52:47.938  INFO 4292 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat st
2023-12-09 11:52:47.941  INFO 4292 --- [           main] com.snhu.sslserver.SslServerApplication  : Started S
```

7. Summary

The code has been refactored to return a validation hash when data is provided.

I addressed the following areas by refactoring code

- Cryptography
- Client/Server
- Code Error
- Code Quality

I added layers of security by

1. Manually verifying no syntactical or logical bugs exist then ensuring no security bugs exist. (Code error and Code quality)
2. Using a strong algorithm for encrypting data (Cryptography)
3. Modifying the server to secure data in transit by using https/TLS(Client/Server)
4. Checking for dependency errors (Code error)

8. Industry Standard Best Practices

I used industry standard best practices by regularly running static tests to identify vulnerabilities and by using algorithms for encryption created and reviewed by cryptographic firms and experts.