GLOBALRAIN

CS 305 Project Two
Practices for Secure Software Report

Table of Contents

Document Revision History

Version	Date	Author	Comments
1.0	10/14/2023	Caio Mauro	

Client

Instructions

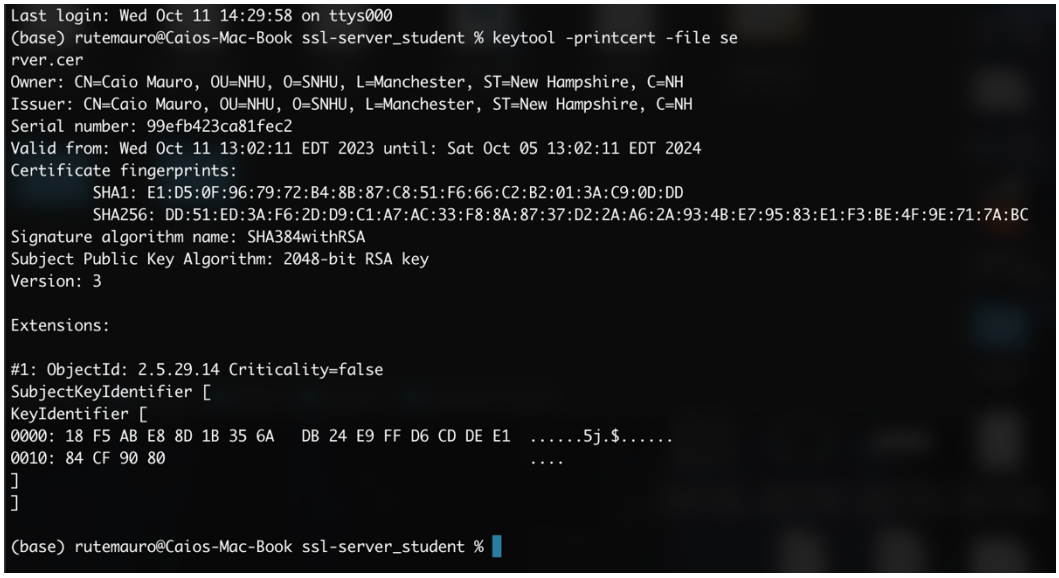
Deliver this completed Practices for Secure Software Report documenting your process for writing secure communications and refactoring code that complies with software security testing protocols. Respond to the steps outlined below and replace the bracketed text with your findings in your own words. If you choose to include images or supporting materials, be sure to insert them throughout.

Developer
Ethan Daugherty

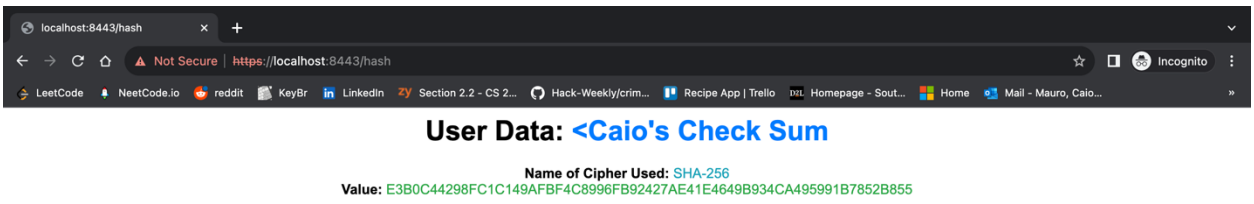
1. Algorithm Cipher

Encryption will protect the files from being read by those who do not have a key. We would recommend a form of asymmetric communication so that the key to encrypt is public and the key to decrypt is private. We recommend the SHA-256 algorithm with a 256-bit key. SHA-256 encryption provides high level encryption due to the possible combinations of the 256-bit key. The check sums which verify the integrity are also nonreversible due to Java's random number generation scheme.

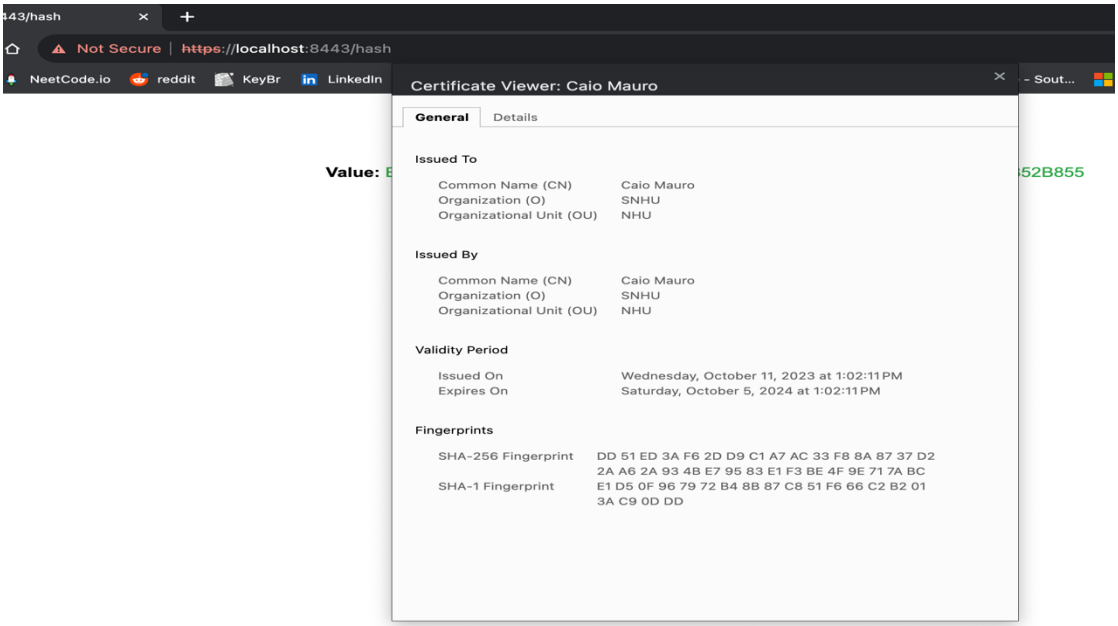
2. Certificate Generation



3. Deploy Cipher



4. Secure Communications (warning due to self-signed)



5. Secondary Testing

```java
@RestController
class ServerController {
    private static final char[] HEX_ARRAY = "0123456789ABCDEF".toCharArray();

    private String getHash(String input) {
        try {
            MessageDigest messageDigest = MessageDigest.getInstance("SHA-256");
            byte[] messageDigestMD5 = messageDigest.digest();
            return bytesToHex(messageDigestMD5);
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        return input;
    }

    public static String bytesToHex(byte[] bytes) {
        char[] hexChars = new char[bytes.length * 2];
        for (int j = 0; j < bytes.length; j++) {
            int v = bytes[j] & 0xFF;
            hexChars[j * 2] = HEX_ARRAY[v >>> 4];
            hexChars[j * 2 + 1] = HEX_ARRAY[v & 0x0F];
        }
        return new String(hexChars);
    }


@RequestMapping("/hash")
public String myHash() {
    String data = "<Caio's Check Sum";
    String escapedData = HtmlUtils.htmlEscape(data);
    String hash = getHash(data);
    return "<html><head></head><body>" +
    "<div style='font-family: Arial, sans-serif; text-align: center;'>" +
    "<h1>User Data: <span style='color: #007BFF;'>" + escapedData + "</span></h1>" +
    "<p>" +
    "<strong>Name of Cipher Used:</strong> <span style='color: #17A2B8;'>SHA-256</span><br>" +
    "<strong>Value:</strong> <span style='color: #28A745;'>" + hash + "</span>" +
    "</p>" +
    "</div>" +
    "</body></html>";
    }
}
```

Summary

Display: Showing Vulnerable Dependencies (click to show all)

Dependency	Vulnerability IDs	Package	Highest Severity	CVE Count	Confidence	Evidence Count
spring-boot-starter-data-rest-2.2.4.RELEASE.jar	cpe:2.3:a:spring:spring:2.2.4:release:*:*:*:*:*:*:*:* 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/org.springframework.boot/spring-boot-starter-data-rest@2.2.4.RELEASE	CRITICAL	3	Highest	28
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/org.springframework.data/spring-data-rest-webmvc@3.2.4.RELEASE	MEDIUM	2	Highest	29
spring-hateoas-1.0.3.RELEASE.jar	cpe:2.3:a:spring:spring:1.0.3:release:*:*:*:*:*:*:*:* cpe:2.3:a:vmware:spring_hateoas:1.0.3:release:*:*:*:*:*:*:*:*	pkg:maven/org.springframework.hateoas/spring-hateoas@1.0.3.RELEASE	MEDIUM	1	Highest	29
jackson-databind-2.10.2.jar	cpe:2.3:a:fasterxml:jackson-databind:2.10.2:*:*:*:*:*:*:*:*	pkg:maven/com.fasterxml.jackson.core/jackson-databind@2.10.2	HIGH	6	Highest	39
spring-boot-2.2.4.RELEASE.jar	cpe:2.3:a:spring:spring:2.2.4:release:*:*:*:*:*:*:*:* cpe:2.3:a:vmware:spring_boot:2.2.4:release:*:*:*:*:*:*:*:*	pkg:maven/org.springframework.boot/spring-boot@2.2.4.RELEASE	CRITICAL	3	Highest	32
logback-core-1.2.3.jar	cpe:2.3:a:qos:logback:1.2.3:*:*:*:*:*:*:*:*	pkg:maven/ch.qos.logback/logback-core@1.2.3	MEDIUM	1	Highest	32
log4j-api-2.12.1.jar	cpe:2.3:a:apache:log4j:2.12.1:*:*:*:*:*:*:*:*	pkg:maven/org.apache.logging.log4j/log4j-api@2.12.1	CRITICAL	5	Highest	46
snakeyaml-1.25.jar	cpe:2.3:a:snakeyaml_project:snakeyaml:1.25:*:*:*:*:*:*:*:* cpe:2.3:a:yaml_project:yaml:1.25:*:*:*:*:*:*:*:*	pkg:maven/org.yaml/snakeyaml@1.25	CRITICAL	10	Highest	28
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/org.apache.tomcat.embed/tomcat-embed-core@9.0.30	CRITICAL	22	Highest	39
hibernate-validator-6.0.18.Final.jar	cpe:2.3:a:redhat:hibernate_validator:6.0.18:*:*:*:*:*:*:*:*	pkg:maven/org.hibernate.validator/hibernate-validator@6.0.18.Final	MEDIUM	1	Highest	36
spring-web-5.2.3.RELEASE.jar	cpe:2.3:a:pivotal_software:spring_framework:5.2.3:release:*:*:*:*:*:*:*:* cpe:2.3:a:spring:spring:5.2.3:release:*:*:*:*:*:*:*:* cpe:2.3:a:springsource:spring_framework:5.2.3:release:*:*:*:*:*:*:*:*	pkg:maven/org.springframework/spring-web@5.2.3.RELEASE	HIGH	4	Highest	28
spring-beans-5.2.3.RELEASE.jar	cpe:2.3:a:pivotal_software:spring_framework:5.2.3:release:*:*:*:*:*:*:*:* cpe:2.3:a:spring:spring:5.2.3:release:*:*:*:*:*:*:*:* cpe:2.3:a:springsource:spring_framework:5.2.3:release:*:*:*:*:*:*:*:*	pkg:maven/org.springframework/spring-beans@5.2.3.RELEASE	HIGH	1	Highest	28
spring-webmvc-5.2.3.RELEASE.jar	cpe:2.3:a:pivotal_software:spring_framework:5.2.3:release:*:*:*:*:*:*:*:* cpe:2.3:a:spring:spring:5.2.3:release:*:*:*:*:*:*:*:* cpe:2.3:a:springsource:spring_framework:5.2.3:release:*:*:*:*:*:*:*:*	pkg:maven/org.springframework/spring-webmvc@5.2.3.RELEASE	MEDIUM	1	Highest	30
spring-context-5.2.3.RELEASE.jar	cpe:2.3:a:pivotal_software:spring_framework:5.2.3:release:*:*:*:*:*:*:*:* cpe:2.3:a:spring:spring:5.2.3:release:*:*:*:*:*:*:*:* cpe:2.3:a:springsource:spring_framework:5.2.3:release:*:*:*:*:*:*:*:*	pkg:maven/org.springframework/spring-context@5.2.3.RELEASE	MEDIUM	1	Highest	28
spring-expression-5.2.3.RELEASE.jar	cpe:2.3:a:pivotal_software:spring_framework:5.2.3:release:*:*:*:*:*:*:*:* cpe:2.3:a:spring:spring:5.2.3:release:*:*:*:*:*:*:*:*	pkg:maven/org.springframework/spring-expression@5.2.3.RELEASE	MEDIUM	3	Highest	30

6. Functional Testing

```java
package com.snhu.sslserver;

import org.springframework.boot.SpringApplication;

@SpringBootApplication
public class SslServerApplication {

    public static void main(String[] args) {
        SpringApplication.run(SslServerApplication.class, args);
    }

}
//FIXME: Add route to enable check sum return of static data example:  String data = "Hello World Check Sum!";

@RestController
class ServerController {
    private static final char[] HEX_ARRAY = "0123456789ABCDEF".toCharArray();

    private String getHash(String input) {
        try {
            MessageDigest messageDigest = MessageDigest.getInstance("SHA-256");
            byte[] messageDigestMD5 = messageDigest.digest();
            return bytesToHex(messageDigestMD5);
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        return input;
    }

    public static String bytesToHex(byte[] bytes) {
        char[] hexChars = new char[bytes.length * 2];
        for (int j = 0; j < bytes.length; j++) {
            int v = bytes[j] & 0xFF;
            hexChars[j * 2] = HEX_ARRAY[v >>> 4];
            hexChars[j * 2 + 1] = HEX_ARRAY[v & 0x0F];
        }
        return new String(hexChars);
    }


@RequestMapping("/hash")
public String myHash() {
    String data = "<Caio's Check Sum";
    String escapedData = HtmlUtils.htmlEscape(data);
    String hash = getHash(data);
    return "<html><head></head><body>" +
    "<div style='font-family: Arial, sans-serif; text-align: center;'>" +
    "<h1>User Data: <span style='color: #007BFF;'>" + escapedData + "</span></h1>" +
    "<p>" +
    "<strong>Name of Cipher Used:</strong> <span style='color: #17A2B8;'>SHA-256</span><br>" +
    "<strong>Value:</strong> <span style='color: #28A745;'>" + hash + "</span>" +
    "</p>" +
    "</div>" +
    "</body></html>";
    }
}
```

7. Summary

We have added a secured Rest Controller to work as the controller for the /hash endpoint. We selected the SHA-256 hashing cipher as it's very secure and has minimal chance at collisions. We would also recommend keeping either set staff for security or assigning security roles to certain developers. They would be tasked with keeping up to date on current encryption best practices and to run dependency check on the active libraries being used. It is very important to always keep the software and the tools being used up to date so that the chance of security risks are minimal.