GLOBALRAIN

Practices for Secure Software Report

Table of Contents

Document Revision History

Version	Date	Author	Comments
1.0	6/22/24	Summer Bernotas	

Client

ARTEMIS
FINANCIAL

Developer

Summer Bernotas

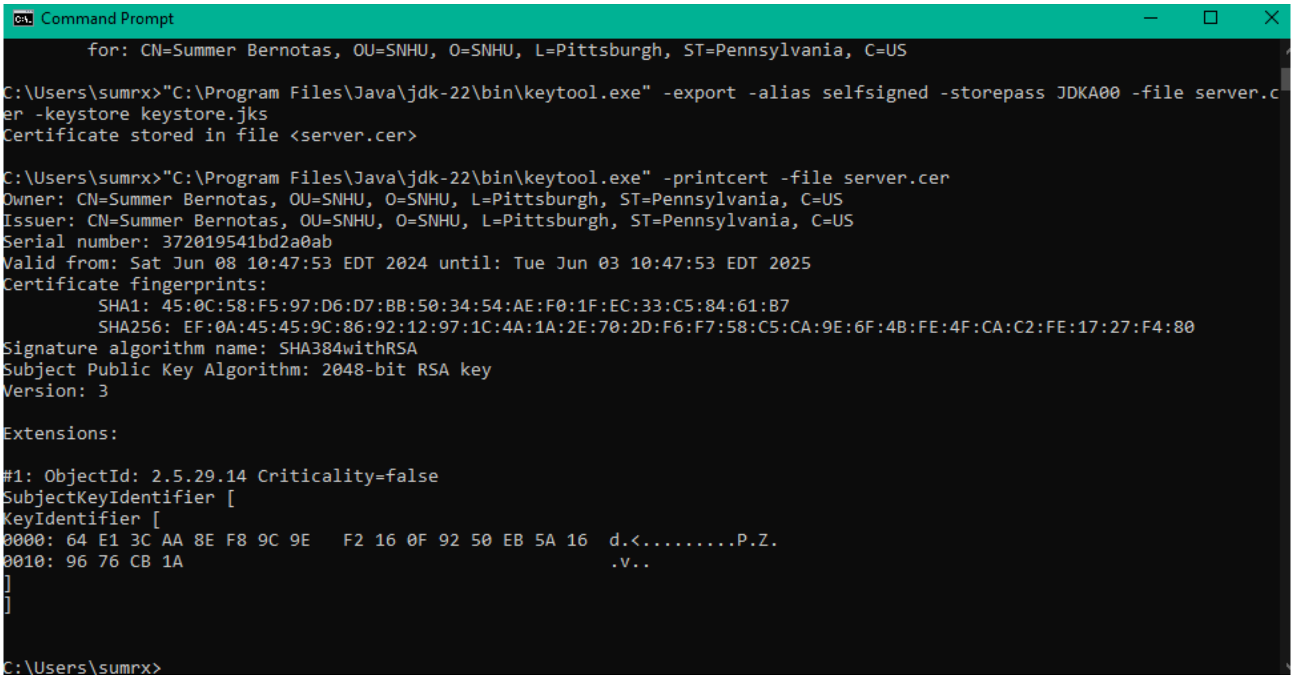
1. Algorithm Cipher

The encryption algorithm cipher is the best recommendation for Artemis Financial to ensure security. Encryption will ensure that files are secure by using mathematical models to change the data which can then only be accessed by those who have a key. It is also recommended that asymmetric communications are used over symmetric. This means that you will use different keys for the encryption (public) and decryption (private) of data. This is different than symmetric keys, as they use the same keys between encryption and the decryption. This goes hand in hand with the use of random numbers. The keys are generated at random, making it harder to crack, this is also why keys should not be easy to remember encryptions. The longer the key, and the more randomized it is, ensuring a higher difficulty of successfully breaking through. The use of the random number generator also ensures a non-reversible checksum that helps verify the validity of the data file.

For the highest level of security, SHA-256 is the best algorithm option. This is a 256-bit encryption to ensure the best security possible as the longer the key, the harder it is to figure it out. We then also have utilization of hash functions. These are created by a mathematical process that takes plaintext data and converts it into a cipher of specific length. No two pieces of information will have the same hash and if that content changes, the hash will as well. This ensures your data is intact and unaltered.

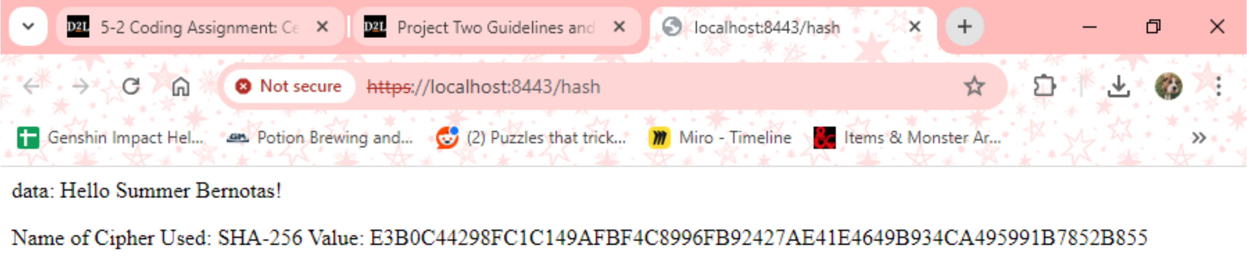
2. Certificate Generation

3.



```
        for: CN=Summer Bernotas, OU=SNHU, O=SNHU, L=Pittsburgh, ST=Pennsylvania, C=US

C:\Users\sumrx>"C:\Program Files\Java\jdk-22\bin\keytool.exe" -export -alias selfsigned -storepass JDKA00 -file server.c
er -keystore keystore.jks
Certificate stored in file <server.cer>

C:\Users\sumrx>"C:\Program Files\Java\jdk-22\bin\keytool.exe" -printcert -file server.cer
Owner: CN=Summer Bernotas, OU=SNHU, O=SNHU, L=Pittsburgh, ST=Pennsylvania, C=US
Issuer: CN=Summer Bernotas, OU=SNHU, O=SNHU, L=Pittsburgh, ST=Pennsylvania, C=US
Serial number: 372019541bd2a0ab
Valid from: Sat Jun 08 10:47:53 EDT 2024 until: Tue Jun 03 10:47:53 EDT 2025
Certificate fingerprints:
         SHA1: 45:0C:58:F5:97:D6:D7:BB:50:34:54:AE:F0:1F:EC:33:C5:84:61:B7
         SHA256: EF:0A:45:45:9C:86:92:12:97:1C:4A:1A:2E:70:2D:F6:F7:58:C5:CA:9E:6F:4B:FE:4F:CA:C2:FE:17:27:F4:80
Signature algorithm name: SHA384withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 64 E1 3C AA 8E F8 9C 9E   F2 16 0F 92 50 EB 5A 16  d.<.........P.Z.
0010: 96 76 CB 1A                                        .v..
]
]


C:\Users\sumrx>
```

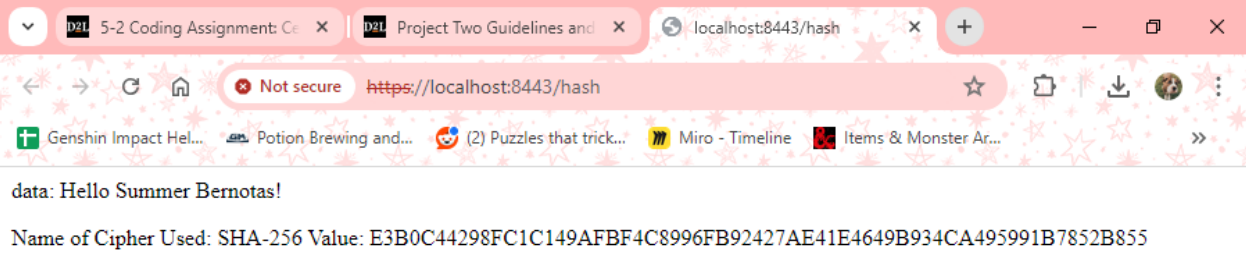
4. Deploy Cipher


5-2 Coding Assignment: Ce
Project Two Guidelines and
localhost:8443/hash
Not secure
~~https~~://localhost:8443/hash
Genshin Impact Hel...
Potion Brewing and...
(2) Puzzles that trick...
Miro - Timeline
Items & Monster Ar...

data: Hello Summer Bernotas!

Name of Cipher Used: SHA-256 Value: E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855

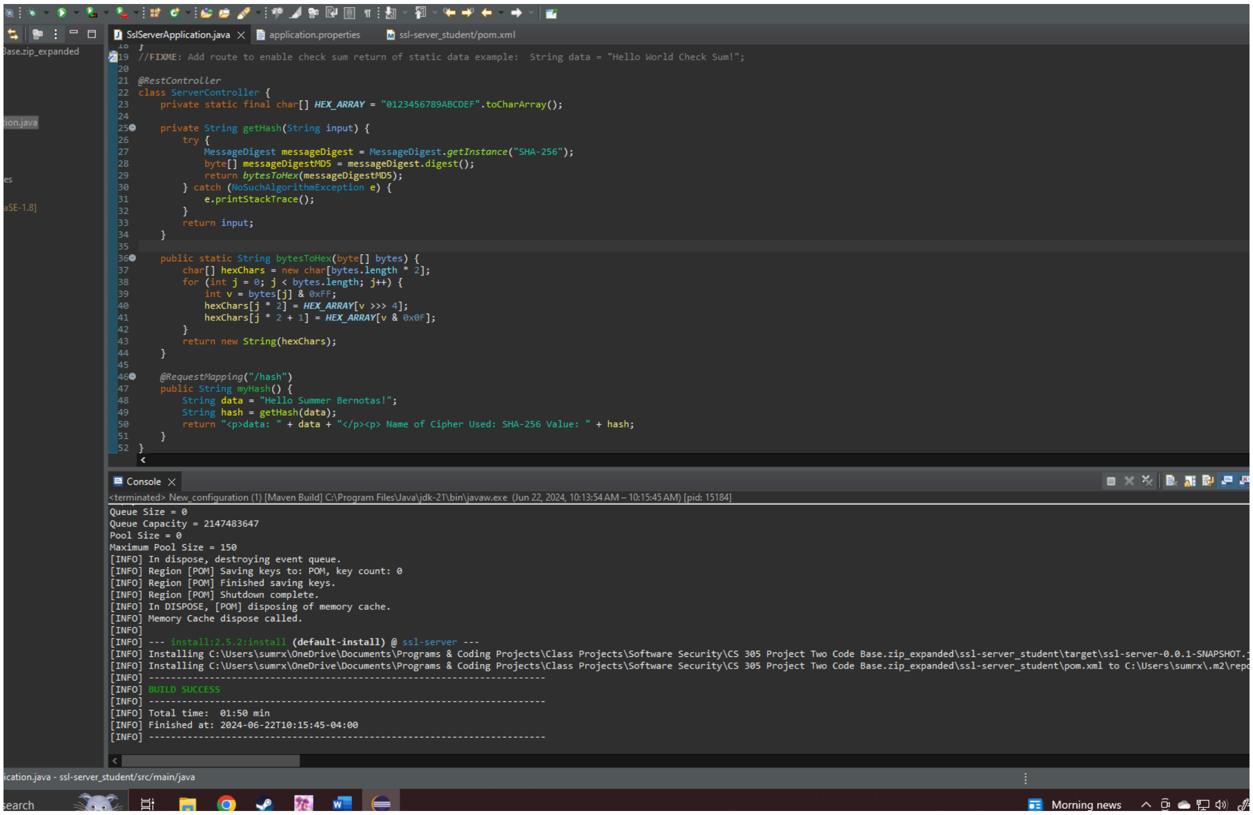
5. Secure Communications


5-2 Coding Assignment: Ce
Project Two Guidelines and
localhost:8443/hash
Not secure
~~https~~://localhost:8443/hash
Genshin Impact Hel...
Potion Brewing and...
(2) Puzzles that trick...
Miro - Timeline
Items & Monster Ar...

data: Hello Summer Bernotas!

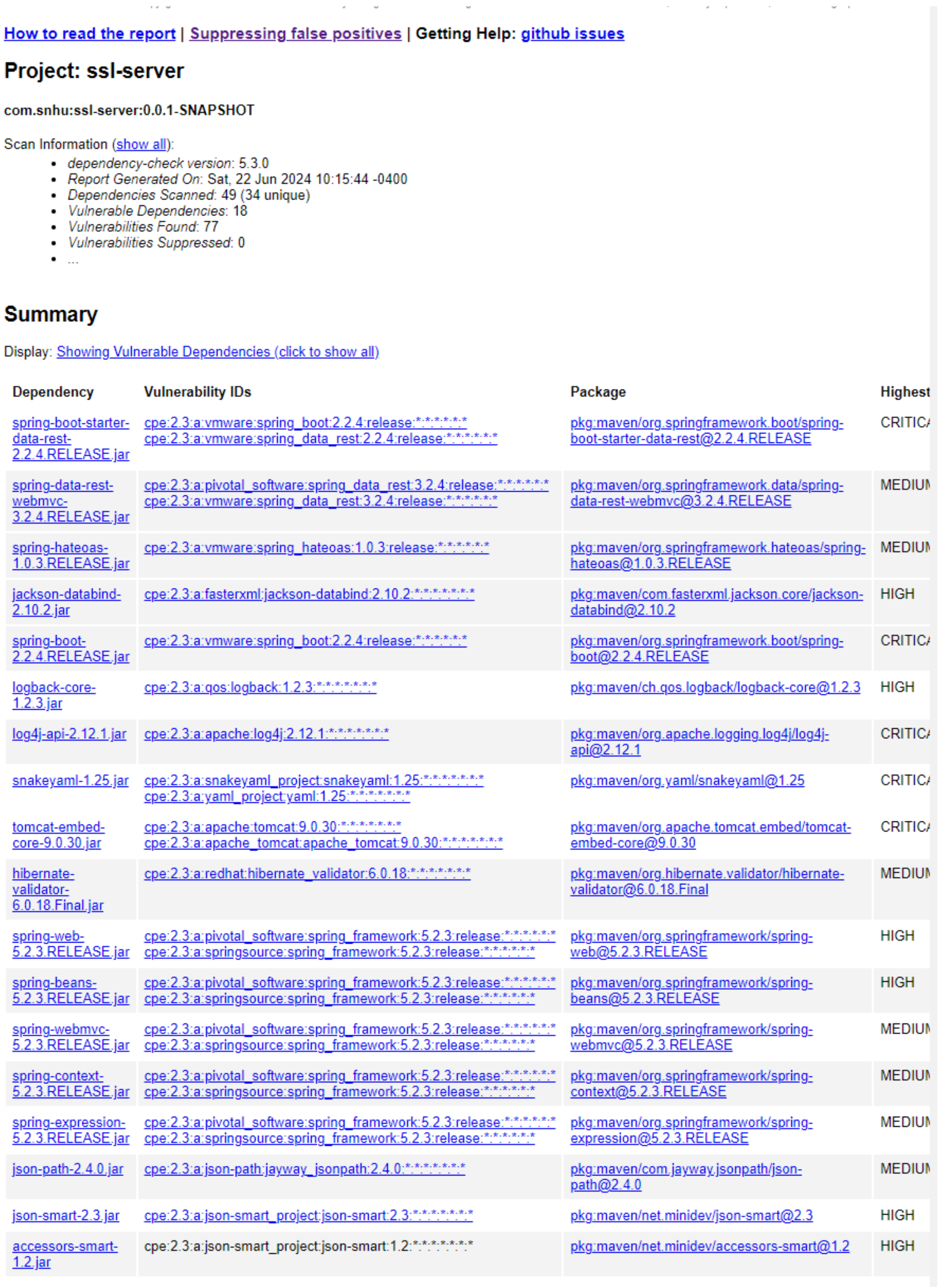
Name of Cipher Used: SHA-256 Value: E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855

6. Secondary Testing



```java
}
//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 = "Hello Summer Bernotas!";
        String hash = getHash(data);
        return "<p>data: " + data + "</p><p> Name of Cipher Used: SHA-256 Value: " + hash;
    }
}
```

```
<terminated> New_configuration (1) [Maven Build] C:\Program Files\Java\jdk-21\bin\javaw.exe  (Jun 22, 2024, 10:13:54 AM – 10:15:45 AM) [pid: 15184]
Queue Size = 0
Queue Capacity = 2147483647
Pool Size = 0
Maximum Pool Size = 150
[INFO] In dispose, destroying event queue.
[INFO] Region [POM] Saving keys to: POM, key count: 0
[INFO] Region [POM] Finished saving keys.
[INFO] Region [POM] Shutdown complete.
[INFO] In DISPOSE, [POM] disposing of memory cache.
[INFO] Memory Cache dispose called.
[INFO]
[INFO] --- install:2.5.2:install (default-install) @ ssl-server ---
[INFO] Installing C:\Users\sumrx\OneDrive\Documents\Programs & Coding Projects\Class Projects\Software Security\CS 305 Project Two Code Base.zip_expanded\ssl-server_student\target\ssl-server-0.0.1-SNAPSHOT.j
[INFO] Installing C:\Users\sumrx\OneDrive\Documents\Programs & Coding Projects\Class Projects\Software Security\CS 305 Project Two Code Base.zip_expanded\ssl-server_student\pom.xml to C:\Users\sumrx\.m2\repo
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  01:50 min
[INFO] Finished at: 2024-06-22T10:15:45-04:00
[INFO] ------------------------------------------------------------------------
```



How to read the report | **Suppressing false positives** | **Getting Help: github issues**

Project: ssl-server

com.snhu:ssl-server:0.0.1-SNAPSHOT

Scan Information (show all):

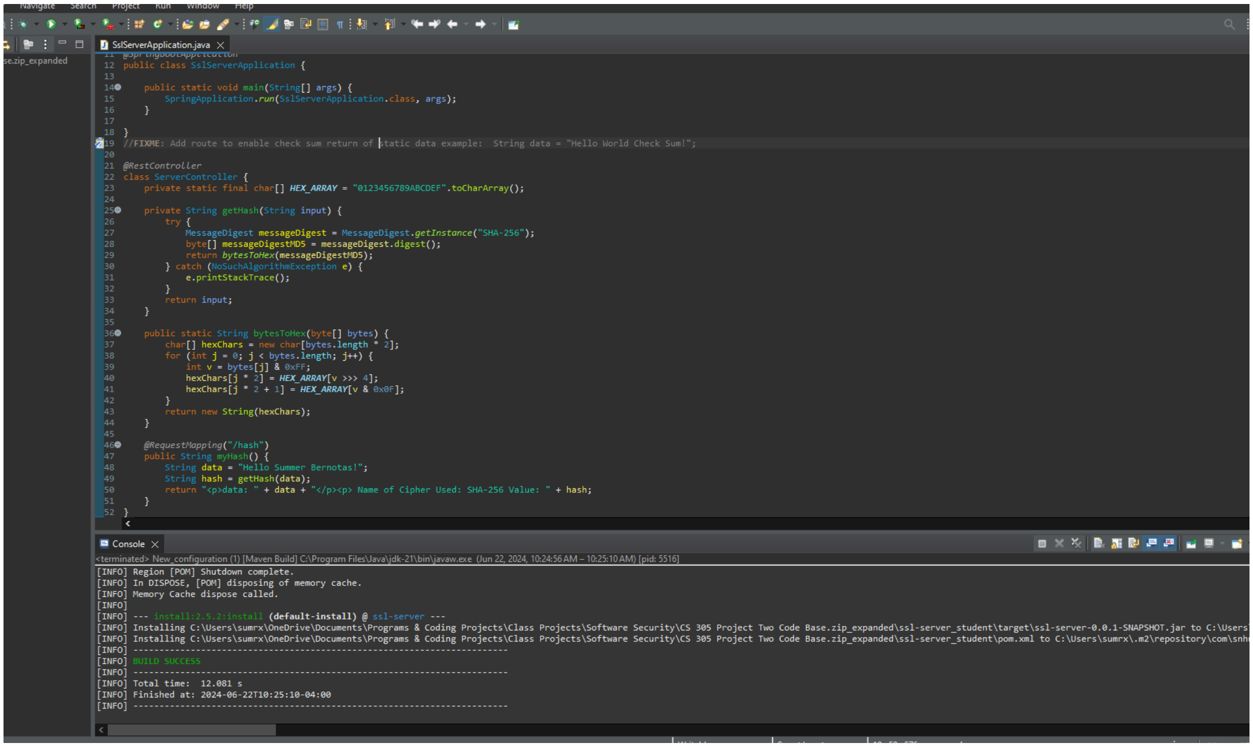
- *dependency-check version*: 5.3.0
- *Report Generated On*: Sat, 22 Jun 2024 10:15:44 -0400
- *Dependencies Scanned*: 49 (34 unique)
- *Vulnerable Dependencies*: 18
- *Vulnerabilities Found*: 77
- *Vulnerabilities Suppressed*: 0
- ...

Summary

Display: Showing Vulnerable Dependencies (click to show all)

Dependency	Vulnerability IDs	Package	Highest
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/org.springframework.boot/spring-boot-starter-data-rest@2.2.4.RELEASE	CRITIC
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	MEDIU
spring-hateoas-1.0.3.RELEASE.jar	cpe:2.3:a:vmware:spring_hateoas:1.0.3:release:*:*:*:*:*:*	pkg:maven/org.springframework.hateoas/spring-hateoas@1.0.3.RELEASE	MEDIU
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
spring-boot-2.2.4.RELEASE.jar	cpe:2.3:a:vmware:spring_boot:2.2.4:release:*:*:*:*:*:*	pkg:maven/org.springframework.boot/spring-boot@2.2.4.RELEASE	CRITIC
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	HIGH
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	CRITIC
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	CRITIC
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	CRITIC
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	MEDIU
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/org.springframework/spring-web@5.2.3.RELEASE	HIGH
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/org.springframework/spring-beans@5.2.3.RELEASE	HIGH
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/org.springframework/spring-webmvc@5.2.3.RELEASE	MEDIU
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/org.springframework/spring-context@5.2.3.RELEASE	MEDIU
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/org.springframework/spring-expression@5.2.3.RELEASE	MEDIU
json-path-2.4.0.jar	cpe:2.3:a:json-path:jayway_jsonpath:2.4.0:*:*:*:*:*:*:*	pkg:maven/com.jayway.jsonpath/json-path@2.4.0	MEDIU
json-smart-2.3.jar	cpe:2.3:a:json-smart_project:json-smart:2.3:*:*:*:*:*:*:*	pkg:maven/net.minidev/json-smart@2.3	HIGH
accessors-smart-1.2.jar	cpe:2.3:a:json-smart_project:json-smart:1.2:*:*:*:*:*:*:*	pkg:maven/net.minidev/accessors-smart@1.2	HIGH

7. Functional Testing



SslServerApplication.java

```java
@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 = "Hello Summer Bernotas!";
        String hash = getHash(data);
        return "<p>data: " + data + "</p><p> Name of Cipher Used: SHA-256 Value: " + hash;
    }
}
```

Console

```
<terminated> New_configuration (1) [Maven Build] C:\Program Files\Java\jdk-21\bin\javaw.exe  (Jun 22, 2024, 10:24:56 AM – 10:25:10 AM) [pid: 5516]
[INFO] Region [POM] Shutdown complete.
[INFO] In DISPOSE, [POM] disposing of memory cache.
[INFO] Memory Cache dispose called.
[INFO]
[INFO] --- install:2.5.2:install (default-install) @ ssl-server ---
[INFO] Installing C:\Users\sumrx\OneDrive\Documents\Programs & Coding Projects\Class Projects\Software Security\CS 305 Project Two Code Base.zip_expanded\ssl-server_student\target\ssl-server-0.0.1-SNAPSHOT.jar to C:\Users
[INFO] Installing C:\Users\sumrx\OneDrive\Documents\Programs & Coding Projects\Class Projects\Software Security\CS 305 Project Two Code Base.zip_expanded\ssl-server_student\pom.xml to C:\Users\sumrx\.m2\repository\com\snh
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  12.081 s
[INFO] Finished at: 2024-06-22T10:25:10-04:00
[INFO] ------------------------------------------------------------------------
```

8. Summary

By using the self signed certificate I was able to enable the use of HTTPS allowing for better key security. This gives users the verification that they are running a system from us directly and not a third-party. I then implemented a hash function utilizing the SHA-256 algorithm cipher and verified it using a checksum verification. Lastly was to check that vulnerabilities were patched. This ensures that the application was functioning properly and as intended.

9. Industry Standard Best Practices

I used industry best practices to ensure that I maintained the software applications current security. I did this by validating and user input, implementing strong password management policies and hashing algorithms for authentication, ensuring that sensitive data was encrypted at rest using secure protocols (HTTPS), and by handling error to prevent sensitive information from getting leaked