GLOBALRAIN

Practices for Secure Software Report

Table of Contents

Document Revision History

Version	Date	Author	Comments
1.0	02/15/2023	Marisa Kuyava	

Client

1. Algorithm Cipher

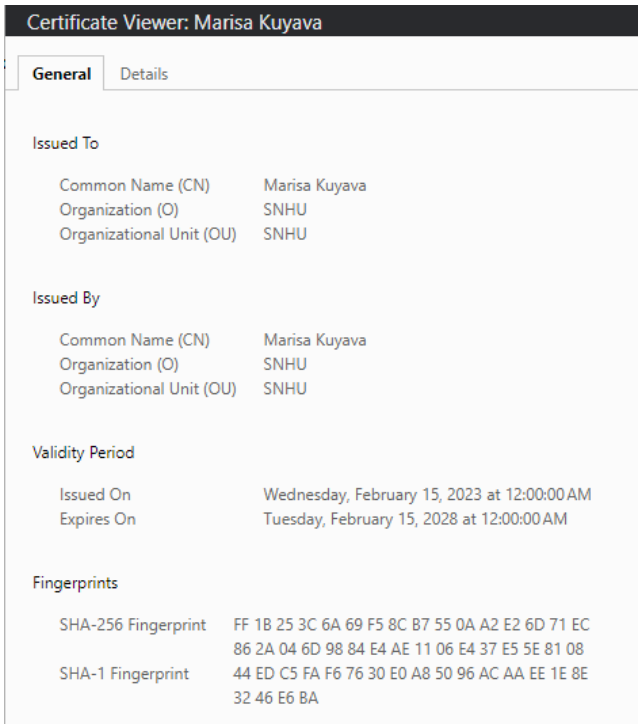
The Advanced Encryption Standard (AES) is the most appropriate file encryption algorithm cipher for Artemis Financial's needs. The AES is a widely adopted symmetric encryption algorithm that was developed to take place of the Data Encryption Standard (DES) due to brute force attack vulnerabilities in the DES. AES uses a single key for both encryption and decryption and utilizes 128-bit, 192-bit, and 256-bit key lengths. The U.S. Government uses AES 192-bit and 256-bit key lengths to protect Top Secret Information. Along with being relatively easily to implement AES has quick encryption and encryption times and required less memory than DES. To ensure that AES is secure it must be implemented properly, and encryption keys must be well protected.

Hash Functions take data of any side and convert it to a compressed fixed length value, which is the hash value. The bit levels, which for AES are 128-bit, 192-bit, and 256-bit, are directly related to the encryption strength, the higher the bit keys, the stronger the encryption. There are two types of encryption, symmetric and Asymmetric. AES uses symmetric encryption, which means that the same key, which is secret, is used to both encrypt and decrypt the data. Asymmetric encryption uses one key to encrypt the data, the public key, and another private key to decrypt the data. When encrypting data utilization of random numbers is important as it decreases the chances that logic can be applied to solve the encryption.

The Advanced Encryption Standard (AES) was developed to create more secure encryption then the DES algorithm that was first published in 1975 by the Federal Register. Because AES has 128-bit, 192-bit and 256-bit, encryption it is much more secure than DES which was only 56-bit encryption. Currently AES provides excellent secure data encryption, however as technology continues to evolve it will be important to continue to evolve security for data privacy as well.

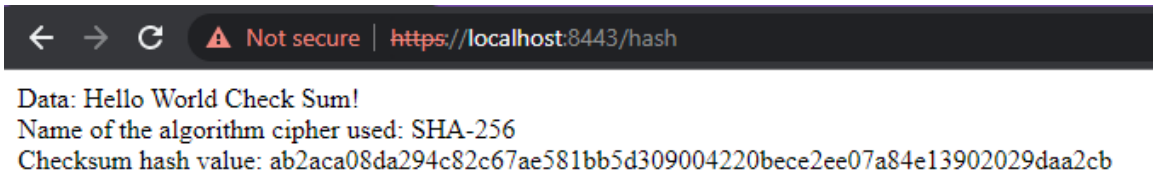
2. Certificate Generation

Screenshot of the CER file.



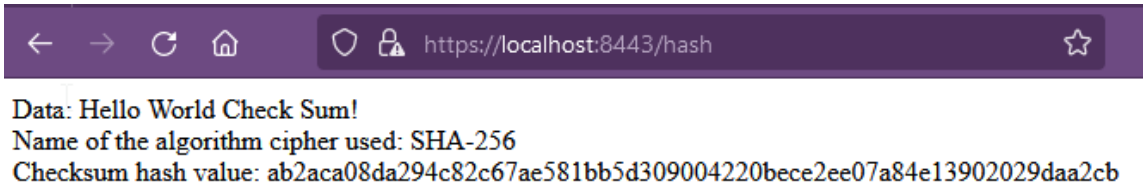
3. Deploy Cipher

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

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

Updated POM.xml file.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.2.4.RELEASE</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.snhu</groupId>
    <artifactId>ssl-server</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>ssl-server</name>
    <description>ssl-server skeleton for CS-305</description>

    <properties>
        <java.version>1.8</java.version>
    </properties>

    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-rest</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
            <exclusions>
                <exclusion>
                    <groupId>org.junit.vintage</groupId>
                    <artifactId>junit-vintage-engine</artifactId>
                </exclusion>
            </exclusions>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
            <plugin>
                <groupId>org.owasp</groupId>
                <artifactId>dependency-check-maven</artifactId>
                <version>8.1.0</version>
                <executions>
                  <execution>
                      <goals>
                          <goal>check</goal>
                      </goals>
                  </execution>
              </executions>
            </plugin>
        </plugins>
    </build>

</project>
```

Refactored code executed without errors.

*SslServerApplication.java application.properties

```java
/*
 * Marisa Kuvava
 * CS 305
 * Project Two
 * */

package com.snhu.sslserver;

import org.springframework.boot.SpringApplication;

@SpringBootApplication
public class SslServerApplication {

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

@RestController
class ServerController {
    @RequestMapping("/hash")
    public String myHash() {

        MessageDigest messageDigest = null; // declare MessageDigest object
        String data = "Hello World Check Sum!"; // data string to be hashed
        String checkSum = null; // Checksum value

        try {
            messageDigest = MessageDigest.getInstance("SHA-256"); // initialize object using SHA-256
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        messageDigest.update(data.getBytes()); // pass data to messageDigest
        byte[] digest = messageDigest.digest(); // compute messageDigest
        checkSum = this.bytesToHex(digest); // create hash value

        return "<p>Data: " + data + "<br>Name of the algorithm cipher used: SHA-256" + "<br>Checksum hash value: "
                + checkSum + "</p>"; // return formatted string
    }

    // Converts byte array to hexadecimal string
    public String bytesToHex(byte[] bytes) {
        StringBuilder springBuilder = new StringBuilder(); // initialize
        for (byte hashByte : bytes) { // loop through byte array
            int intVal = 0xff & hashByte;
            if (intVal < 0x10) {
                springBuilder.append('0'); // append elements
            }
            springBuilder.append(Integer.toHexString(intVal));
        }
        return springBuilder.toString(); // return hexadecimal string
    }

}
```

Problems Javadoc Declaration Console Keytool

```
SslServerApplication [Java Application] D:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (Feb 15, 2023, 3:23:14 PM) [pid: 22172]
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.2.4.RELEASE)

2023-02-15 15:23:14.801  INFO 22172 --- [           main] com.snhu.sslserver.SslServerApplication  : Starting SslServerApplication on Eris with PID 22172 (started by mkuya in D:\SNHU\CS 305\Week7\ssl-server_student)
2023-02-15 15:23:14.803  INFO 22172 --- [           main] com.snhu.sslserver.SslServerApplication  : No active profile set, falling back to default profiles: default
2023-02-15 15:23:15.727  INFO 22172 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8443 (https)
2023-02-15 15:23:15.735  INFO 22172 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2023-02-15 15:23:15.735  INFO 22172 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache Tomcat/9.0.30]
2023-02-15 15:23:15.811  INFO 22172 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2023-02-15 15:23:15.811  INFO 22172 --- [           main] o.s.web.context.ContextLoader            : Root WebApplicationContext: initialization completed in 974 ms
2023-02-15 15:23:16.246  INFO 22172 --- [           main] o.s.s.concurrent.ThreadPoolTaskExecutor  : Initializing ExecutorService 'applicationTaskExecutor'
2023-02-15 15:23:16.586  INFO 22172 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8443 (https) with context path ''
2023-02-15 15:23:16.589  INFO 22172 --- [           main] com.snhu.sslserver.SslServerApplication  : Started SslServerApplication in 2.027 seconds (JVM running for 2.332)
2023-02-15 15:23:29.015  INFO 22172 --- [nio-8443-exec-2] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring DispatcherServlet 'dispatcherServlet'
2023-02-15 15:23:29.016  INFO 22172 --- [nio-8443-exec-2] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
2023-02-15 15:23:29.030  INFO 22172 --- [nio-8443-exec-2] o.s.web.servlet.DispatcherServlet        : Completed initialization in 14 ms
```

application.properties

```properties
## need to add server.  entries to enable HTTPS with SSL keystore, replace "????" with correct entries

server.port=8443
server.ssl.key-alias=selfsigned
server.ssl.key-store-password=password
server.ssl.key-store=keystore.jks
server.ssl.key-store-type=jks
```

Dependency -Check

DEPENDENCY-CHECK

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

Sponsor

Project: ssl-server

com.snhu:ssl-server:0.0.1-SNAPSHOT

Scan Information (show less):

- *dependency-check version*: 8.1.0
- *Report Generated On*: Wed, 15 Feb 2023 15:27:30 -0800
- *Dependencies Scanned*: 49 (31 unique)
- *Vulnerable Dependencies*: 14
- *Vulnerabilities Found*: 88
- *Vulnerabilities Suppressed*: 0
- *NVD CVE Checked*: 2023-02-15T15:27:13
- *NVD CVE Modified*: 2023-02-15T15:00:03
- *VersionCheckOn*: 2023-02-07T10:35:36
- *kev.checked*: 1676488106

Summary

Display: Showing Vulnerable Dependencies (click to show all)

Dependency	Vulnerability IDs	Package	Highest Severity	CVE Count	Confidence	Evidence Count
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	34
jackson-databind-2.10.2.jar	cpe:2.3:a:fasterxml:jackson-databind:2.10.2:*:*:*:*:*:*:* cpe:2.3:a:fasterxml:jackson-modules-java8:2.10.2:*:*:*:*:*:*:*	pkg:maven/com.fasterxml.jackson.core/jackson-databind@2.10.2	HIGH	4	Highest	41
json-smart-2.3.jar	cpe:2.3:a:ini-parser_project:ini-parser:2.3:*:*:*:*:*:*:* cpe:2.3:a:json-smart_project:json-smart-v2:2.3:*:*:*:*:*:*:*	pkg:maven/net.minidev/json-smart@2.3	HIGH	2	Low	47
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	LOW	1	Highest	44
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	33
snakeyaml-1.25.jar	cpe:2.3:a:snakeyaml_project:snakeyaml:1.25:*:*:*:*:*:*:*	pkg:maven/org.yaml/snakeyaml@1.25	HIGH	8	Highest	46
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	HIGH	1	Highest	39
spring-boot-starter-web-2.2.4.RELEASE.jar	cpe:2.3:a:vmware:spring_boot:2.2.4:release:*:*:*:*:*:* cpe:2.3:a:web_project:web:2.2.4:release:*:*:*:*:*:*	pkg:maven/org.springframework.boot/spring-boot-starter-web@2.2.4.RELEASE	HIGH	1	Highest	35
spring-core-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:*:*:*:*:*:* cpe:2.3:a:vmware:spring_framework:5.2.3:release:*:*:*:*:*:*	pkg:maven/org.springframework/spring-core@5.2.3.RELEASE	CRITICAL*	9	Highest	36
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	27
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:*:*:*:*:*:* cpe:2.3:a:vmware:spring_framework:5.2.3:release:*:*:*:*:*:* cpe:2.3:a:web_project:web:5.2.3:release:*:*:*:*:*:*	pkg:maven/org.springframework/spring-web@5.2.3.RELEASE	CRITICAL*	10	Highest	34
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:*:*:*:*:*:* cpe:2.3:a:vmware:spring_framework:5.2.3:release:*:*:*:*:*:* cpe:2.3:a:web_project:web:5.2.3:release:*:*:*:*:*:*	pkg:maven/org.springframework/spring-webmvc@5.2.3.RELEASE	CRITICAL*	9	Highest	36
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*	19	Highest	33
tomcat-embed-websocket-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-websocket@9.0.30	CRITICAL*	20	Highest	32

6. Functional Testing

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

SslServerApplication.java

```java
/*
 * Naniza Kuvava
 * CS 305
 * Project Two
 * */

package com.snhu.sslserver;

import org.springframework.boot.SpringApplication;

@SpringBootApplication
public class SslServerApplication {

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

@RestController
class ServerController {
    @RequestMapping("/hash")
    public String myHash() {

        MessageDigest messageDigest = null; // declare MessageDigest object
        String data = "Hello World Check Sum!"; // data string to be hashed
        String checkSum = null; // Checksum value

        try {
            messageDigest = MessageDigest.getInstance("SHA-256"); // initialize object using SHA-256
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        messageDigest.update(data.getBytes()); // pass data to messageDigest
        byte[] digest = messageDigest.digest(); // compute messageDigest
        checkSum = this.bytesToHex(digest); // create hash value

        return "<p>Data: " + data + "<br>Name of the algorithm cipher used: SHA-256" + "<br>Checksum hash value: "
                + checkSum + "</p>"; // return formatted string
    }

    // Converts byte array to hexadecimal string
    public String bytesToHex(byte[] bytes) {
        StringBuilder springBuilder = new StringBuilder(); // initialize
        for (byte hashByte : bytes) { // loop through byte array
            int intVal = 0xff & hashByte;
            if (intVal < 0x10) {
                springBuilder.append('0'); // append elements
            }
            springBuilder.append(Integer.toHexString(intVal));
        }
        return springBuilder.toString(); // return hexadecimal string
    }

}
```

Problems | Javadoc | Declaration | Console | Keytool

SslServerApplication [Java Application] [pid: 7632]

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

2023-02-15 16:03:17.972  INFO 7632 --- [           main] com.snhu.sslserver.SslServerApplication  : Starting SslServerApplication on Eris with PI
2023-02-15 16:03:17.975  INFO 7632 --- [           main] com.snhu.sslserver.SslServerApplication  : No active profile set, falling back to defaul
2023-02-15 16:03:18.901  INFO 7632 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8443 (https)
2023-02-15 16:03:18.909  INFO 7632 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2023-02-15 16:03:18.909  INFO 7632 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache Tomcat/9.0.3
2023-02-15 16:03:18.975  INFO 7632 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationCo
2023-02-15 16:03:18.975  INFO 7632 --- [           main] o.s.web.context.ContextLoader            : Root WebApplicationContext: initialization co
2023-02-15 16:03:19.388  INFO 7632 --- [           main] o.s.s.concurrent.ThreadPoolTaskExecutor  : Initializing ExecutorService 'applicationTask
2023-02-15 16:03:19.732  INFO 7632 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8443 (https) with
2023-02-15 16:03:19.735  INFO 7632 --- [           main] com.snhu.sslserver.SslServerApplication  : Started SslServerApplication in 1.988 seconds
```

7. Summary

The areas of security that were address by refactoring the code are APIs, Cryptography, Code Error, Code Quality, and Input validation.

- APIs
  - RESTful API was implemented to protect from system attacks.
- Cryptography
  - Refactoring was done to include a hash function to encrypt data.
- Client/Server
  - Certificate was added so that data transfer is more secure.
- Code Error
  - Secure error handling is implemented.
- Code Quality
  - Secure coding practices and patters are used.

The primary security that I added to the software was the self-signed certificates, this allowed for HTTPS to be utilized. Additionally, a hash function was added to encrypt all data handled.

8. Industry Standard Best Practices

To maintain the current security of the application it is important to continue running dependency checks on the application to help mitigate any future potential vulnerabilities. Updates should be made to the out of date versions within the program to continue to ensure data security.