GLOBALRAIN

Practices for Secure Software Report

Table of Contents

Document Revision History

Version	Date	Author	Comments
1.0	April 14, 2023	Cody Faircloth	

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.

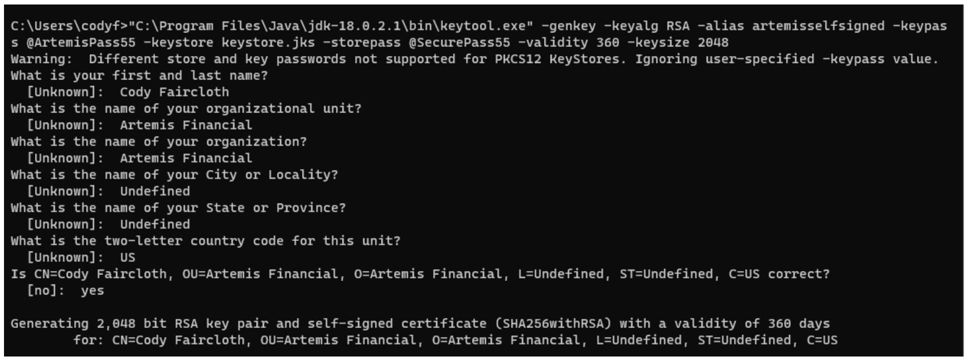
Developer
Cody Faircloth

1. Algorithm Cipher

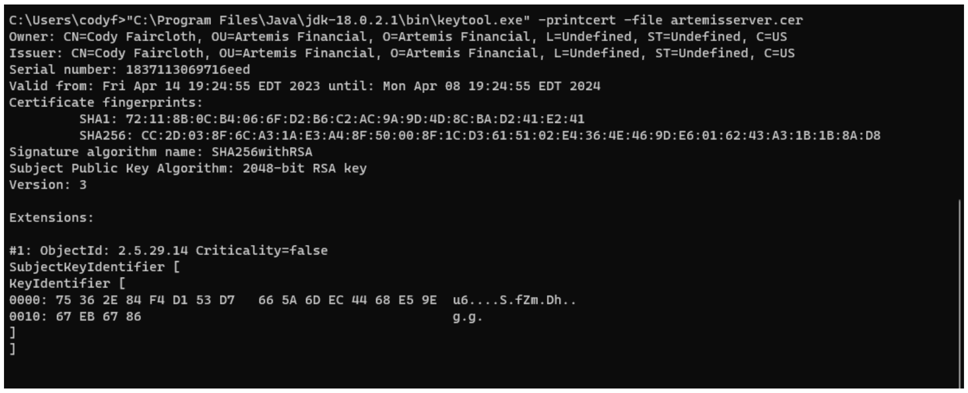
Artemis Financial is requesting an encryption algorithm for the encryption and archive of long-term files. Since these files will not be in transit SHA-256 would be the recommended encryption cipher. SHA-256 is a symmetric key cipher meaning that a single key is created which will be kept by Artemis Financial. The key will be created through the use of Java's random number generator using the system's entropy as the seed for the random number generator. The data will then be hashed based on the random number generator's output, encrypting the data. SHA-256 encrypts using a 256-bit level providing the highest level of security for brute force attacks. Since these long-term files will not be in transit asymmetric keys will not be implemented. The vulnerability of a symmetric key is that the key could be stolen but since the key will not be transferred outside of the local network of Artemis Financial, with the implementation of a secure network, the risk of key theft is minimal.

2. Certificate Generation

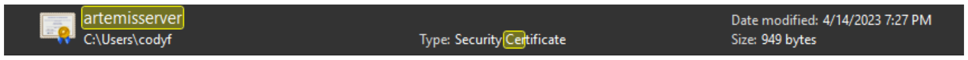
Insert a screenshot below of the CER file.



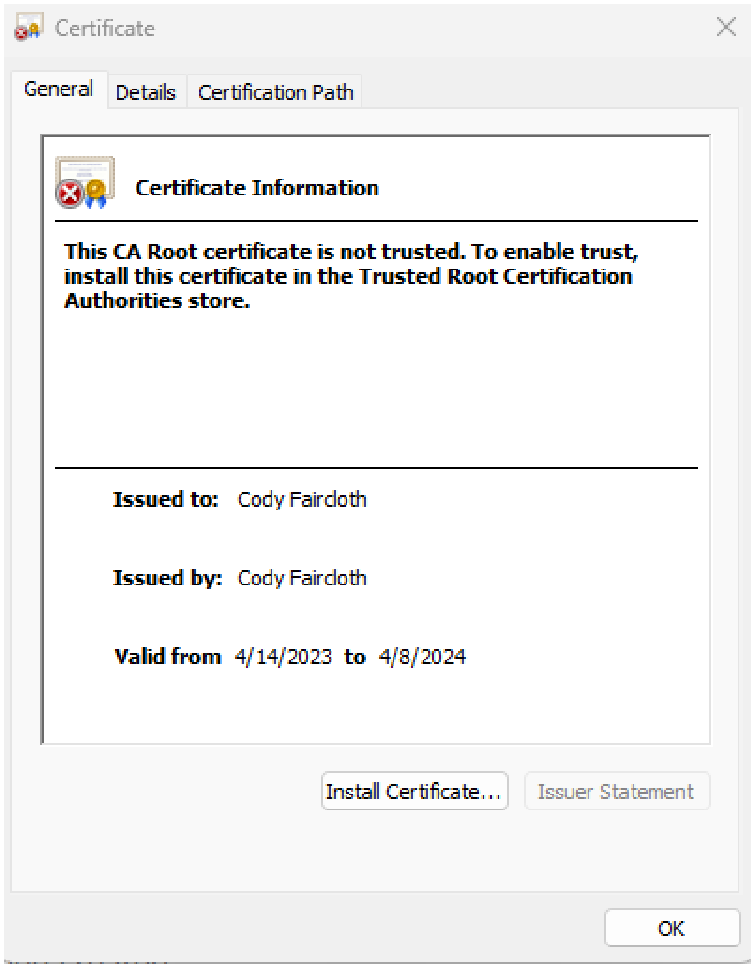
```
C:\Users\codyf>"C:\Program Files\Java\jdk-18.0.2.1\bin\keytool.exe" -genkey -keyalg RSA -alias artemisselfsigned -keypas
s @ArtemisPass55 -keystore keystore.jks -storepass @SecurePass55 -validity 360 -keysize 2048
Warning:  Different store and key passwords not supported for PKCS12 KeyStores. Ignoring user-specified -keypass value.
What is your first and last name?
  [Unknown]:  Cody Faircloth
What is the name of your organizational unit?
  [Unknown]:  Artemis Financial
What is the name of your organization?
  [Unknown]:  Artemis Financial
What is the name of your City or Locality?
  [Unknown]:  Undefined
What is the name of your State or Province?
  [Unknown]:  Undefined
What is the two-letter country code for this unit?
  [Unknown]:  US
Is CN=Cody Faircloth, OU=Artemis Financial, O=Artemis Financial, L=Undefined, ST=Undefined, C=US correct?
  [no]:  yes

Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 360 days
        for: CN=Cody Faircloth, OU=Artemis Financial, O=Artemis Financial, L=Undefined, ST=Undefined, C=US
```



```
C:\Users\codyf>"C:\Program Files\Java\jdk-18.0.2.1\bin\keytool.exe" -printcert -file artemisserver.cer
Owner: CN=Cody Faircloth, OU=Artemis Financial, O=Artemis Financial, L=Undefined, ST=Undefined, C=US
Issuer: CN=Cody Faircloth, OU=Artemis Financial, O=Artemis Financial, L=Undefined, ST=Undefined, C=US
Serial number: 1837113069716eed
Valid from: Fri Apr 14 19:24:55 EDT 2023 until: Mon Apr 08 19:24:55 EDT 2024
Certificate fingerprints:
         SHA1: 72:11:8B:0C:B4:06:6F:D2:B6:C2:AC:9A:9D:4D:8C:BA:D2:41:E2:41
         SHA256: CC:2D:03:8F:6C:A3:1A:E3:A4:8F:50:00:8F:1C:D3:61:51:02:E4:36:4E:46:9D:E6:01:62:43:A3:1B:1B:8A:D8
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 75 36 2E 84 F4 D1 53 D7   66 5A 6D EC 44 68 E5 9E  u6....S.fZm.Dh..
0010: 67 EB 67 86                                        g.g.
]
]
```



```
artemisserver                                                   Date modified: 4/14/2023 7:27 PM
C:\Users\codyf               Type: Security Certificate         Size: 949 bytes
```



Certificate Information

This CA Root certificate is not trusted. To enable trust, install this certificate in the Trusted Root Certification Authorities store.

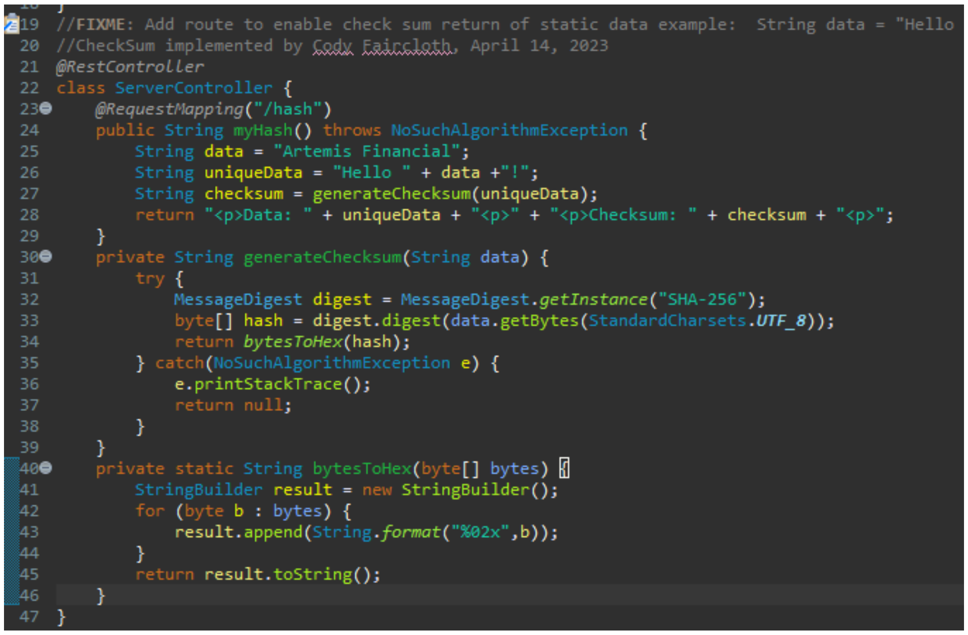
Issued to: Cody Faircloth

Issued by: Cody Faircloth

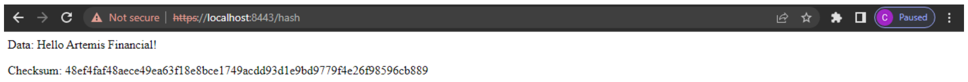
Valid from 4/14/2023 to 4/8/2024

3. Deploy Cipher

Insert a screenshot below of the checksum verification.



```java
}
//FIXME: Add route to enable check sum return of static data example:  String data = "Hello
//CheckSum implemented by Cody Faircloth, April 14, 2023
@RestController
class ServerController {
    @RequestMapping("/hash")
    public String myHash() throws NoSuchAlgorithmException {
        String data = "Artemis Financial";
        String uniqueData = "Hello " + data +"!";
        String checksum = generateChecksum(uniqueData);
        return "<p>Data: " + uniqueData + "<p>" + "<p>Checksum: " + checksum + "<p>";
    }
    private String generateChecksum(String data) {
        try {
            MessageDigest digest = MessageDigest.getInstance("SHA-256");
            byte[] hash = digest.digest(data.getBytes(StandardCharsets.UTF_8));
            return bytesToHex(hash);
        } catch(NoSuchAlgorithmException e) {
            e.printStackTrace();
            return null;
        }
    }
    private static String bytesToHex(byte[] bytes) {
        StringBuilder result = new StringBuilder();
        for (byte b : bytes) {
            result.append(String.format("%02x",b));
        }
        return result.toString();
    }
}
```



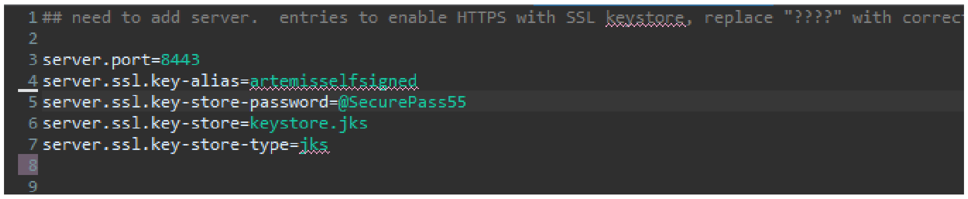
Not secure | ~~https~~://localhost:8443/hash

Data: Hello Artemis Financial!

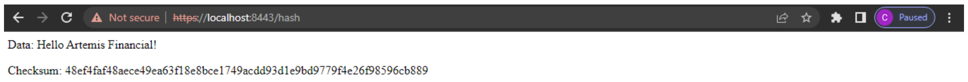
Checksum: 48ef4faf48aece49ea63f18e8bce1749acdd93d1e9bd9779f4e26f98596cb889

4. Secure Communications

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



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

server.port=8443
server.ssl.key-alias=artemisselfsigned
server.ssl.key-store-password=@SecurePass55
server.ssl.key-store=keystore.jks
server.ssl.key-store-type=jks
```



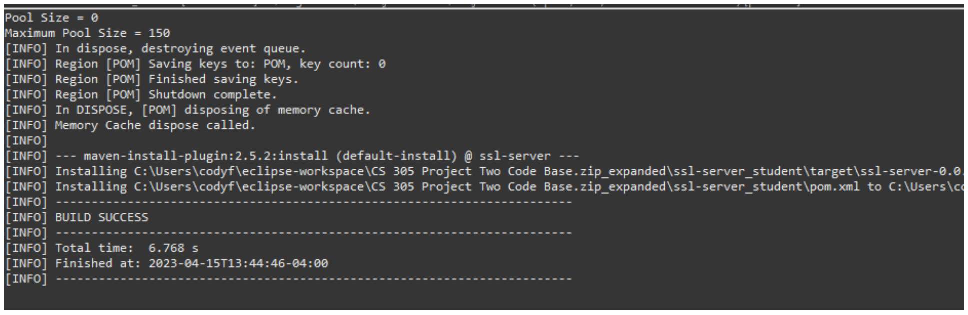
Not secure | ~~https~~://localhost:8443/hash

Data: Hello Artemis Financial!

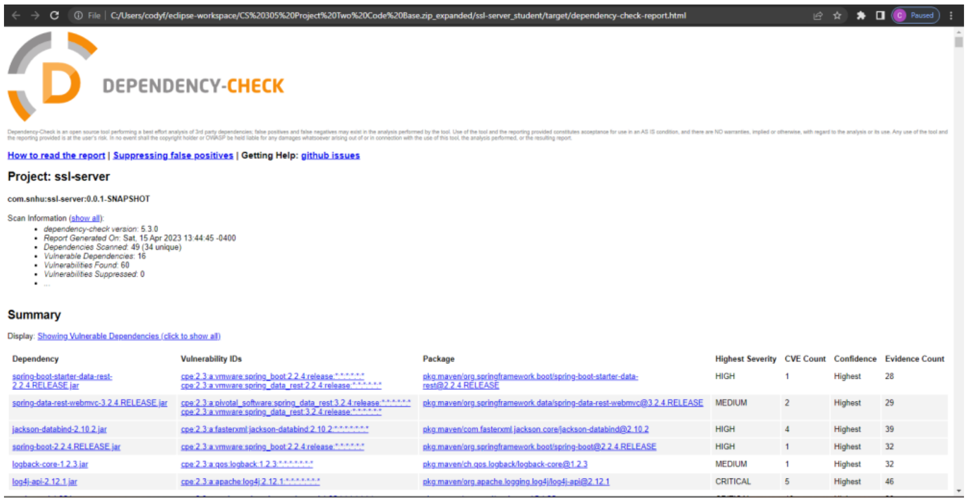
Checksum: 48ef4faf48aece49ea63f18e8bce1749acdd93d1e9bd9779f4e26f98596cb889

5. Secondary Testing

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



```
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] --- maven-install-plugin:2.5.2:install (default-install) @ ssl-server ---
[INFO] Installing C:\Users\codyf\eclipse-workspace\CS 305 Project Two Code Base.zip_expanded\ssl-server_student\target\ssl-server-0.0
[INFO] Installing C:\Users\codyf\eclipse-workspace\CS 305 Project Two Code Base.zip_expanded\ssl-server_student\pom.xml to C:\Users\c
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  6.768 s
[INFO] Finished at: 2023-04-15T13:44:46-04:00
[INFO] ------------------------------------------------------------------------
```



C:/Users/codyf/eclipse-workspace/CS%20305%20Project%20Two%20Code%20Base.zip_expanded/ssl-server_student/target/dependency-check-report.html

DEPENDENCY-CHECK

Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false negatives may exist in the analysis performed by the tool. Use of the tool and the reporting provided constitutes acceptance for use in an AS IS condition, and there are NO warranties, implied or otherwise, with regard to the analysis or its use. Any use of the tool and the reporting provided is at the user's risk. In no event shall the copyright holder or OWASP be held liable for any damages whatsoever arising out of or in connection with the use of this tool, the analysis performed, or the resulting report.

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, 15 Apr 2023 13:44:45 -0400
- *Dependencies Scanned*: 49 (34 unique)
- *Vulnerable Dependencies*: 16
- *Vulnerabilities Found*: 60
- *Vulnerabilities Suppressed*: 0
- ...

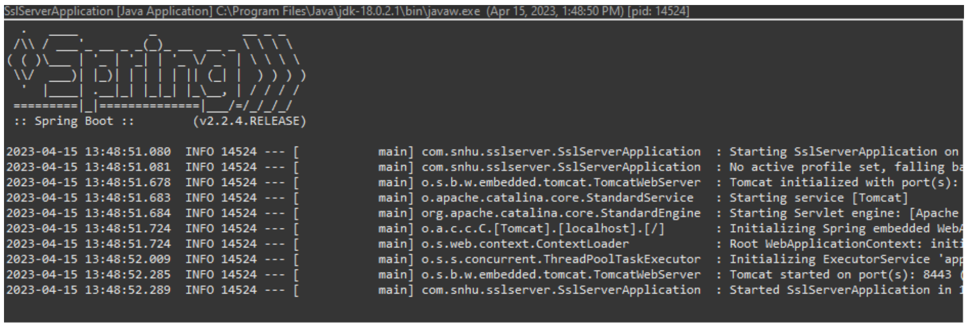
Summary

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: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	HIGH	1	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
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	4	Highest	39
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	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

6. Functional Testing

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



SslServerApplication [Java Application] C:\Program Files\Java\jdk-18.0.2.1\bin\javaw.exe (Apr 15, 2023, 1:48:50 PM) [pid: 14524]

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

2023-04-15 13:48:51.080  INFO 14524 --- [           main] com.snhu.sslserver.SslServerApplication  : Starting SslServerApplication on
2023-04-15 13:48:51.081  INFO 14524 --- [           main] com.snhu.sslserver.SslServerApplication  : No active profile set, falling ba
2023-04-15 13:48:51.678  INFO 14524 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s):
2023-04-15 13:48:51.683  INFO 14524 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2023-04-15 13:48:51.684  INFO 14524 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache
2023-04-15 13:48:51.724  INFO 14524 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebA
2023-04-15 13:48:51.724  INFO 14524 --- [           main] o.s.web.context.ContextLoader            : Root WebApplicationContext: initi
2023-04-15 13:48:52.009  INFO 14524 --- [           main] o.s.s.concurrent.ThreadPoolTaskExecutor  : Initializing ExecutorService 'app
2023-04-15 13:48:52.285  INFO 14524 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8443
2023-04-15 13:48:52.289  INFO 14524 --- [           main] com.snhu.sslserver.SslServerApplication  : Started SslServerApplication in 1
```

7. Summary

I implemented a checksum encryption function as shown above to encrypt data into the local server. This adds a layer of security through encryption as requested by Artemis Financial. Additionally, I implemented a self-signed certificate to convert the HTTP link to an HTTPS link adding an additional layer of security to the application. I checked vulnerabilities through dependencies using the OWASP dependency test and determined no additional vulnerabilities were created through my code modification. The areas of security addressed in this project are APIs, cryptography, client/server, code error, and code quality. APIs were secured through dependency testing. Cryptography was implemented through the inclusion of a checksum function. The client/server was secured through the implementation of HTTPS via a self-signed certificate. Code error was addressed by handling the NoSuchAlgorithmException in the checksum function. Code quality was addressed by using industry best practices when creating the program. My process for adding these layers of security was to implement the requirements in the order they were presented and then test to ensure each requirement was implemented properly. If a requirement was not implemented properly, I would research ways to solve the issue. For example, the server would not host on localhost:8443 at first due to a dependency not being located in the classpath. I added the keystore file to the classpath and the server hosted correctly.

8. Industry Standard Best Practices

I used industry standard best practices in several ways. First, I coded using proper syntax. Using proper syntax allows your code to be maintainable and readable to other programmers who may modify the code in the future. Additionally, I implemented industry standard practices when creating certificates. I created my certificate using the data given to me and created a certificate that provided an additional layer of security. Using industry standard best practices is valuable in software development because it keeps all developers on the same standard in regards to security.