GLOBALRAIN

Artemis Financial Practices for Secure Software Report

Table of Contents

Document Revision History

Version	Date	Author	Comments
1.0	12/05/2022	Eric Slutz	Completed algorithm cipher section.
	12/06/2022	Eric Slutz	Completed self signed certificate section.
			Started work on deploy cipher section.
	12/07/2022	Eric Slutz	Completed deploy cipher section.
			Completed secure connection section.
			Completed secondary testing section.
			Completed functional testing section.
	12/08/2022	Eric Slutz	Completed summary section.
			Completed industry best practices secion.

Client

ARTEMIS
FINANCIAL

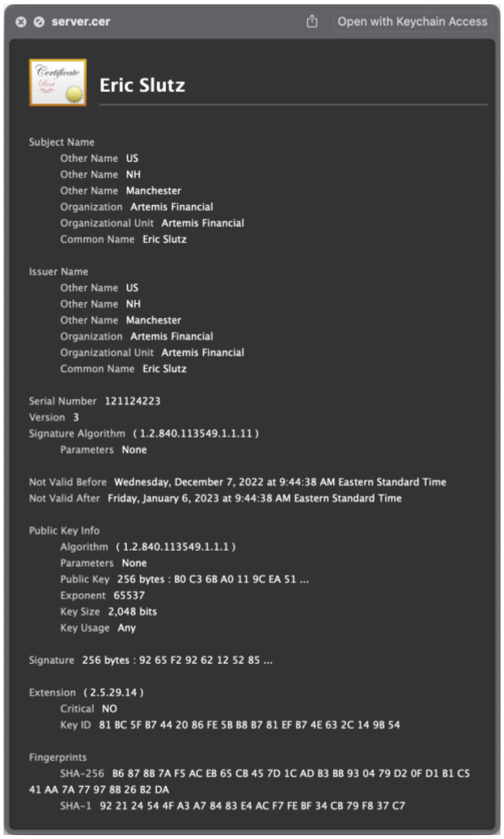
Developer

Eric Slutz

1. Algorithm Cipher

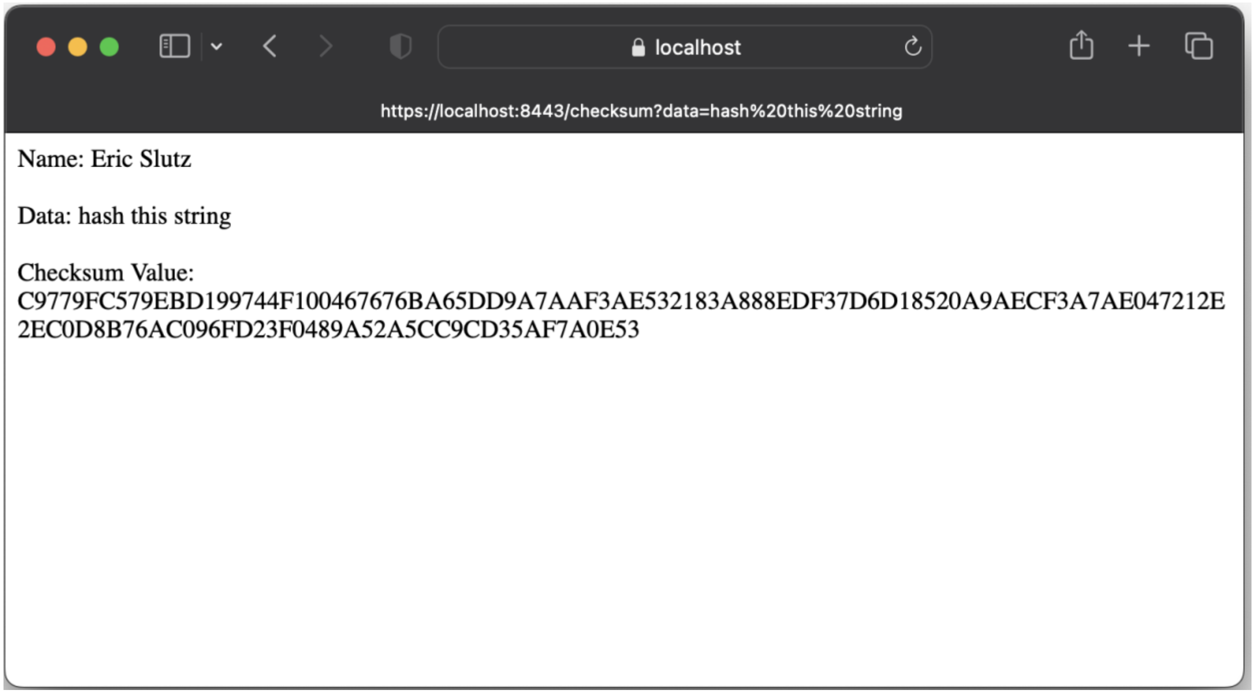
Encryption algorithm ciphers perform a series of steps to convert data to obscure the content of the data. The history of encryption can be traced back to ancient history. However, the widespread use of encryption didn't become commonplace until World War Two. The current state of encryption algorithms, such as AES, is that they are under no threat of being broken. The only way to get around it is through gaining access to the key. That should remain true until quantum computing becomes more common. At that point new encryption strategies will be needed to deal with that threat. To ensure secure communication of data in motion and from the available algorithm ciphers listed for MessageDigest in Java Security Standard Algorithm Names, SHA-512 is going to be the most secure and efficient option. In this case, a hash is created using SHA-512 to create the checksum. A hash collision is when a hash for a different value matches the hash of your value. Therefore, it is important to use a cipher algorithm that avoids collisions. Both SHA-256 and SHA-512 are collision resistant. The bit level of the cipher indicates the length of the key to encrypt or decrypt the data. In general, 256-bits or larger is considered very secure. The use of non-symmetric keys is also needed. This allows any data to be encrypted by the sender using the public key. The receiver can then verify that data with the checksum and decrypt that the data with their private key.

2. Certificate Generation



```
Eric Slutz

Subject Name
        Other Name  US
        Other Name  NH
        Other Name  Manchester
        Organization  Artemis Financial
        Organizational Unit  Artemis Financial
        Common Name  Eric Slutz

Issuer Name
        Other Name  US
        Other Name  NH
        Other Name  Manchester
        Organization  Artemis Financial
        Organizational Unit  Artemis Financial
        Common Name  Eric Slutz

Serial Number  121124223
Version  3
Signature Algorithm  ( 1.2.840.113549.1.1.11 )
        Parameters  None

Not Valid Before  Wednesday, December 7, 2022 at 9:44:38 AM Eastern Standard Time
Not Valid After  Friday, January 6, 2023 at 9:44:38 AM Eastern Standard Time

Public Key Info
        Algorithm  ( 1.2.840.113549.1.1.1 )
        Parameters  None
        Public Key  256 bytes : B0 C3 6B A0 11 9C EA 51 ...
        Exponent  65537
        Key Size  2,048 bits
        Key Usage  Any

Signature  256 bytes : 92 65 F2 92 62 12 52 85 ...

Extension  ( 2.5.29.14 )
        Critical  NO
        Key ID  81 BC 5F B7 44 20 86 FE 5B B8 B7 81 EF B7 4E 63 2C 14 9B 54

Fingerprints
        SHA-256  B6 87 8B 7A F5 AC EB 65 CB 45 7D 1C AD B3 BB 93 04 79 D2 0F D1 B1 C5
41 AA 7A 77 97 8B 26 B2 DA
        SHA-1  92 21 24 54 4F A3 A7 84 83 E4 AC F7 FE BF 34 CB 79 F8 37 C7
```

3. Deploy Cipher

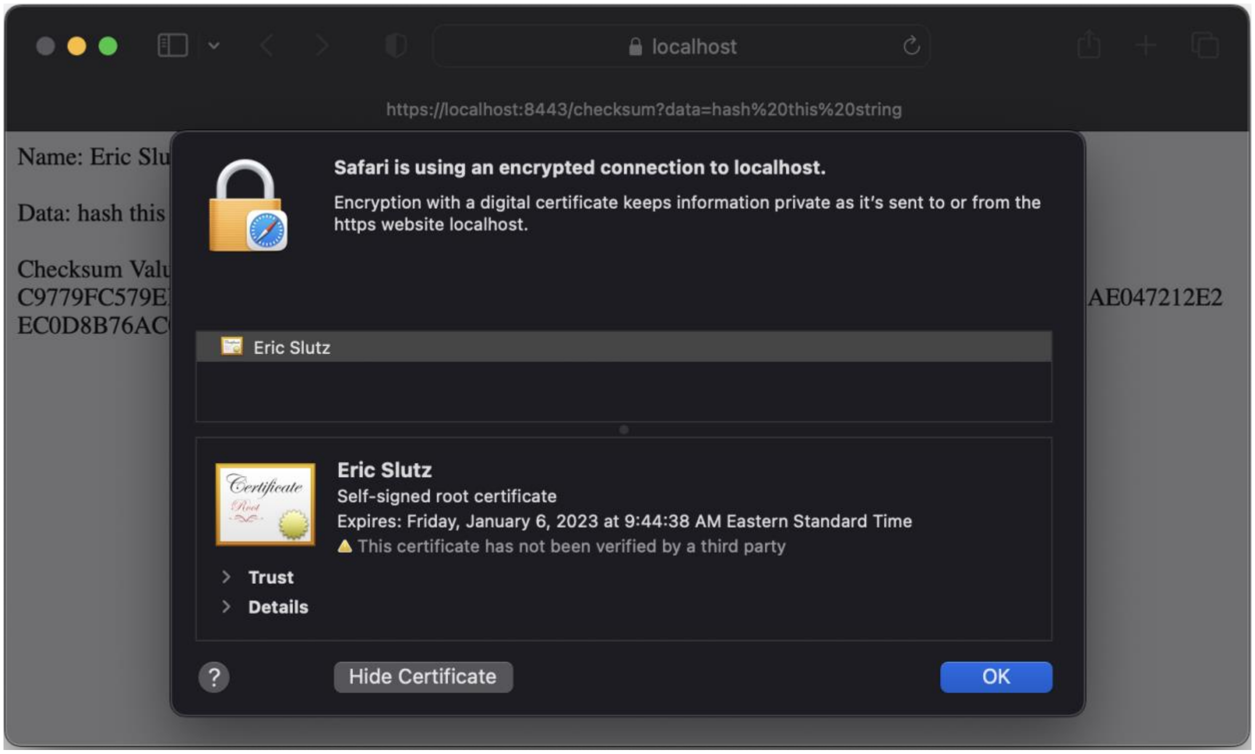


Name: Eric Slutz

Data: hash this string

Checksum Value:
C9779FC579EBD199744F100467676BA65DD9A7AAF3AE532183A888EDF37D6D18520A9AECF3A7AE047212E
2EC0D8B76AC096FD23F0489A52A5CC9CD35AF7A0E53

4. Secure Communications



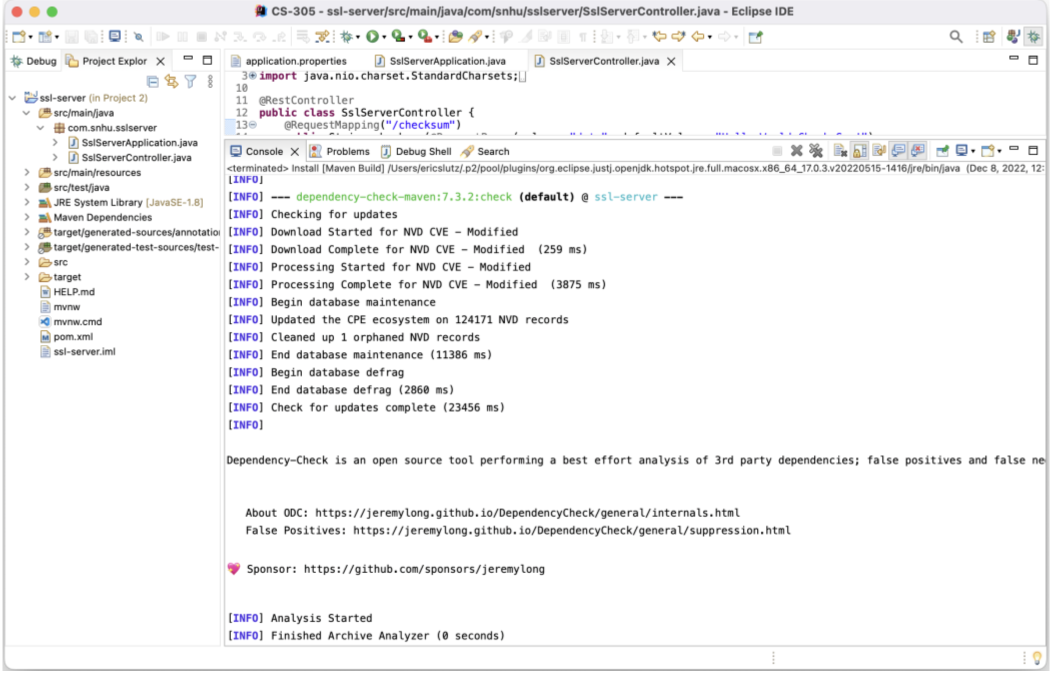
Safari is using an encrypted connection to localhost.

Encryption with a digital certificate keeps information private as it's sent to or from the https website localhost.

Eric Slutz

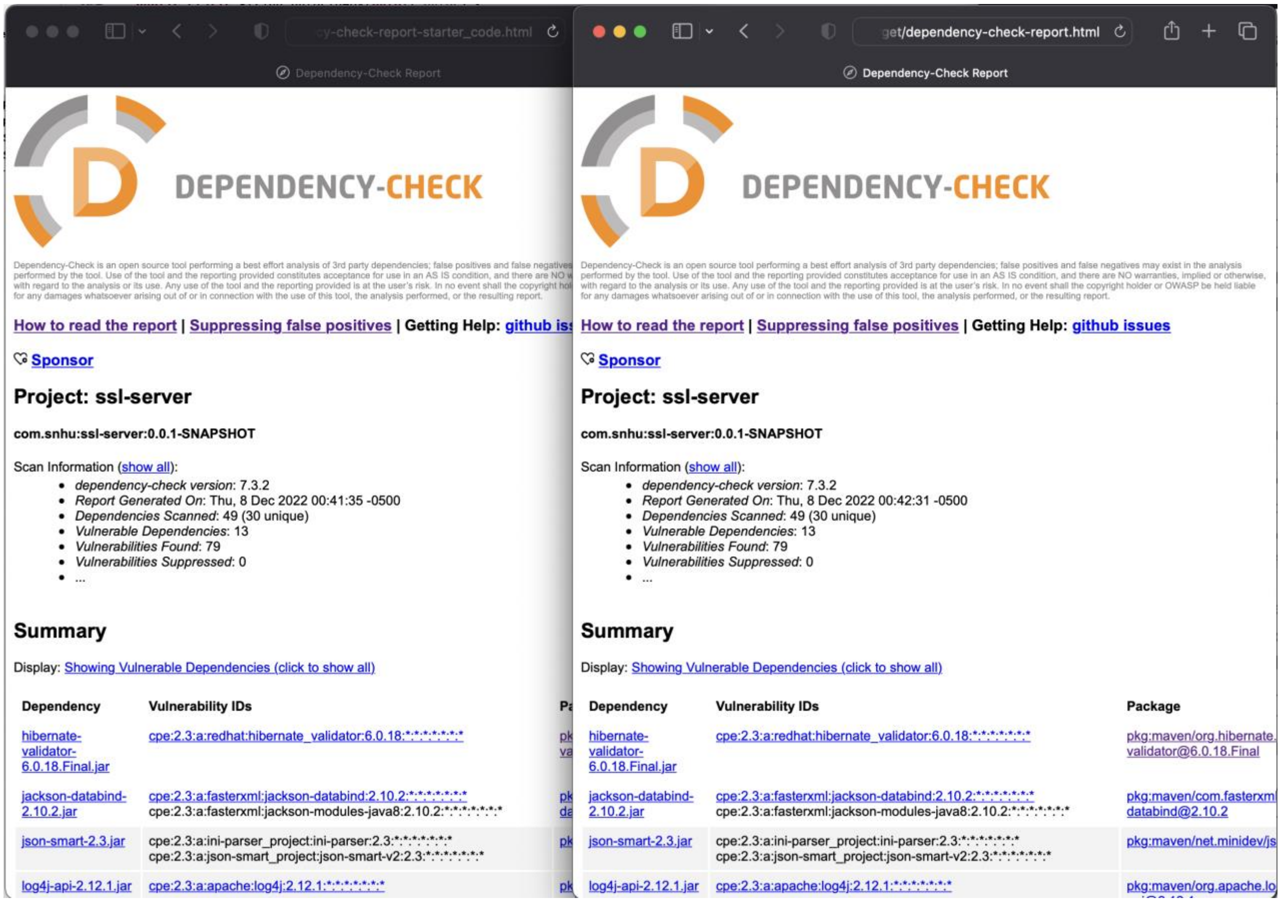
Eric Slutz
Self-signed root certificate
Expires: Friday, January 6, 2023 at 9:44:38 AM Eastern Standard Time
⚠ This certificate has not been verified by a third party

5. Secondary Testing


CS-305 - ssl-server/src/main/java/com/snhu/sslserver/SslServerController.java - Eclipse IDE
Debug
Project Explor
ssl-server (in Project 2)
src/main/java
com.snhu.sslserver
SslServerApplication.java
SslServerController.java
src/main/resources
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
target/generated-sources/annotatio
target/generated-test-sources/test-
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml
ssl-server.iml
application.properties
SslServerApplication.java
SslServerController.java
Console
Problems
Debug Shell
Search

```java
import java.nio.charset.StandardCharsets;

@RestController
public class SslServerController {
    @RequestMapping("/checksum")
```

```
<terminated> Install [Maven Build] /Users/ericslutz/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_17.0.3.v20220515-1416/jre/bin/java  (Dec 8, 2022, 12:
[INFO]
[INFO] --- dependency-check-maven:7.3.2:check (default) @ ssl-server ---
[INFO] Checking for updates
[INFO] Download Started for NVD CVE - Modified
[INFO] Download Complete for NVD CVE - Modified  (259 ms)
[INFO] Processing Started for NVD CVE - Modified
[INFO] Processing Complete for NVD CVE - Modified  (3875 ms)
[INFO] Begin database maintenance
[INFO] Updated the CPE ecosystem on 124171 NVD records
[INFO] Cleaned up 1 orphaned NVD records
[INFO] End database maintenance (11386 ms)
[INFO] Begin database defrag
[INFO] End database defrag (2860 ms)
[INFO] Check for updates complete (23456 ms)
[INFO]

Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false ne


   About ODC: https://jeremylong.github.io/DependencyCheck/general/internals.html
   False Positives: https://jeremylong.github.io/DependencyCheck/general/suppression.html

💖 Sponsor: https://github.com/sponsors/jeremylong


[INFO] Analysis Started
[INFO] Finished Archive Analyzer (0 seconds)
```



Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false negatives 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 w 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 hol 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 iss

♡ Sponsor

Project: ssl-server

com.snhu:ssl-server:0.0.1-SNAPSHOT

Scan Information (show all):

- *dependency-check version*: 7.3.2
- *Report Generated On*: Thu, 8 Dec 2022 00:41:35 -0500
- *Dependencies Scanned*: 49 (30 unique)
- *Vulnerable Dependencies*: 13
- *Vulnerabilities Found*: 79
- *Vulnerabilities Suppressed*: 0
- ...

Summary

Display: Showing Vulnerable Dependencies (click to show all)

Dependency	Vulnerability IDs	Pa
hibernate-validator-6.0.18.Final.jar	cpe:2.3:a:redhat:hibernate_validator:6.0.18:*:*:*:*:*:*:*	pk va
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:*:*:*:*:*:*:*	pk da
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:*:*:*:*:*:*:*	pk
log4j-api-2.12.1.jar	cpe:2.3:a:apache:log4j:2.12.1:*:*:*:*:*:*:*	pk

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

♡ Sponsor

Project: ssl-server

com.snhu:ssl-server:0.0.1-SNAPSHOT

Scan Information (show all):

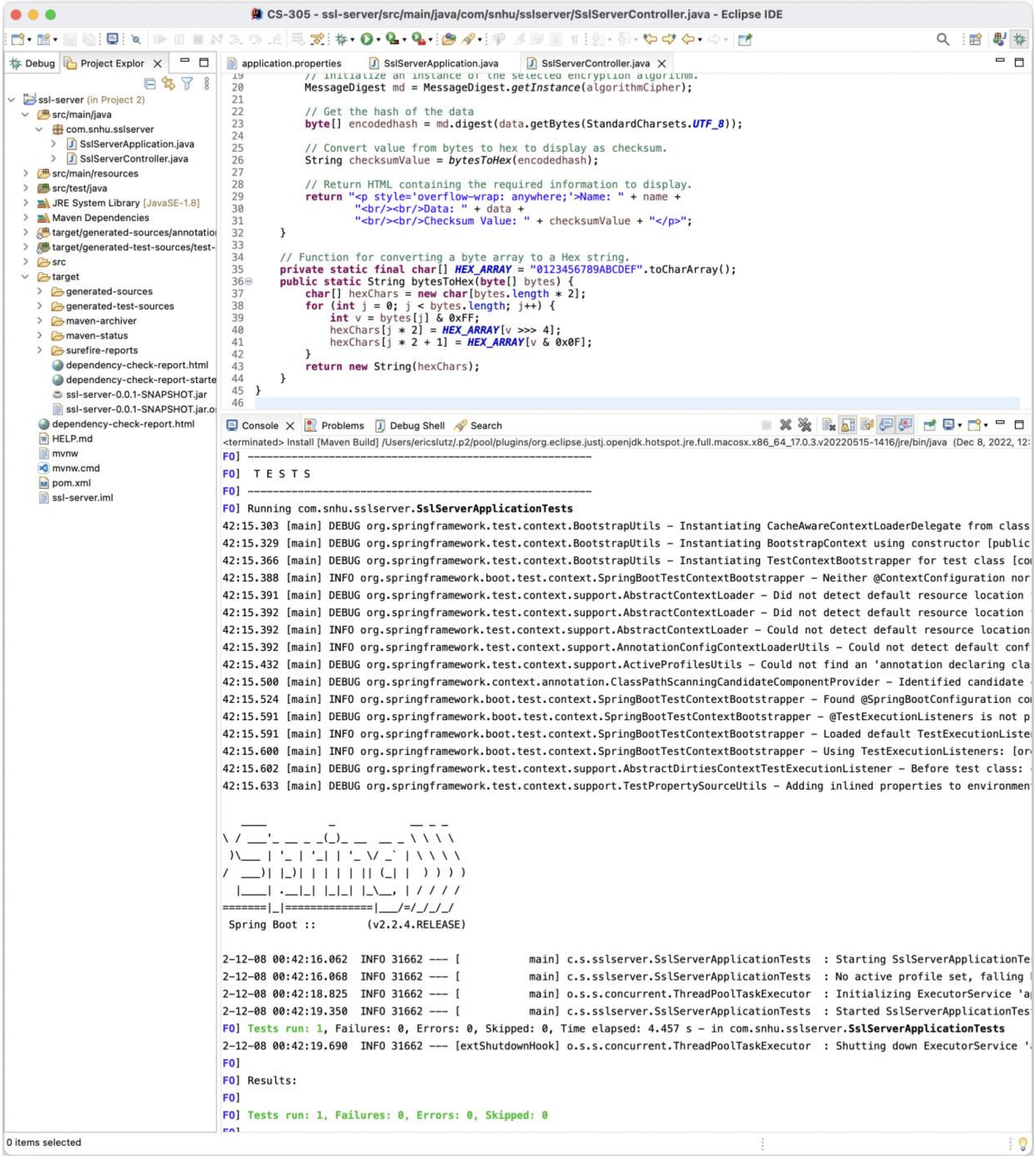
- *dependency-check version*: 7.3.2
- *Report Generated On*: Thu, 8 Dec 2022 00:42:31 -0500
- *Dependencies Scanned*: 49 (30 unique)
- *Vulnerable Dependencies*: 13
- *Vulnerabilities Found*: 79
- *Vulnerabilities Suppressed*: 0
- ...

Summary

Display: Showing Vulnerable Dependencies (click to show all)

Dependency	Vulnerability IDs	Package
hibernate-validator-6.0.18.Final.jar	cpe:2.3:a:redhat:hibernate_validator:6.0.18:*:*:*:*:*:*:*	pkg:maven/org.hibernate. validator@6.0.18.Final
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 databind@2.10.2
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/js
log4j-api-2.12.1.jar	cpe:2.3:a:apache:log4j:2.12.1:*:*:*:*:*:*:*	pkg:maven/org.apache.lo

6. Functional Testing


CS-305 - ssl-server/src/main/java/com/snhu/sslserver/SslServerController.java - Eclipse IDE
Debug
Project Explor
ssl-server (in Project 2)
src/main/java
com.snhu.sslserver
SslServerApplication.java
SslServerController.java
src/main/resources
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
target/generated-sources/annotatio
target/generated-test-sources/test-
src
target
generated-sources
generated-test-sources
maven-archiver
maven-status
surefire-reports
dependency-check-report.html
dependency-check-report-starte
ssl-server-0.0.1-SNAPSHOT.jar
ssl-server-0.0.1-SNAPSHOT.jar.o
dependency-check-report.html
HELP.md
mvnw
mvnw.cmd
pom.xml
ssl-server.iml
application.properties
SslServerApplication.java
SslServerController.java
Console
Problems
Debug Shell
Search
0 items selected

```java
        // initialize an instance of the selected encryption algorithm.
        MessageDigest md = MessageDigest.getInstance(algorithmCipher);

        // Get the hash of the data
        byte[] encodedhash = md.digest(data.getBytes(StandardCharsets.UTF_8));

        // Convert value from bytes to hex to display as checksum.
        String checksumValue = bytesToHex(encodedhash);

        // Return HTML containing the required information to display.
        return "<p style='overflow-wrap: anywhere;'>Name: " + name +
                "<br/><br/>Data: " + data +
                "<br/><br/>Checksum Value: " + checksumValue + "</p>";
    }

    // Function for converting a byte array to a Hex string.
    private static final char[] HEX_ARRAY = "0123456789ABCDEF".toCharArray();
    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);
    }
}
```

```
<terminated> Install [Maven Build] /Users/ericslutz/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_17.0.3.v20220515-1416/jre/bin/java (Dec 8, 2022, 12:
FO] -------------------------------------------------------
FO]  T E S T S
FO] -------------------------------------------------------
FO] Running com.snhu.sslserver.SslServerApplicationTests
42:15.303 [main] DEBUG org.springframework.test.context.BootstrapUtils - Instantiating CacheAwareContextLoaderDelegate from class
42:15.329 [main] DEBUG org.springframework.test.context.BootstrapUtils - Instantiating BootstrapContext using constructor [public
42:15.366 [main] DEBUG org.springframework.test.context.BootstrapUtils - Instantiating TestContextBootstrapper for test class [co
42:15.388 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper - Neither @ContextConfiguration nor
42:15.391 [main] DEBUG org.springframework.test.context.support.AbstractContextLoader - Did not detect default resource location
42:15.392 [main] DEBUG org.springframework.test.context.support.AbstractContextLoader - Did not detect default resource location
42:15.392 [main] INFO org.springframework.test.context.support.AbstractContextLoader - Could not detect default resource location
42:15.392 [main] INFO org.springframework.test.context.support.AnnotationConfigContextLoaderUtils - Could not detect default conf
42:15.432 [main] DEBUG org.springframework.test.context.support.ActiveProfilesUtils - Could not find an 'annotation declaring cla
42:15.500 [main] DEBUG org.springframework.context.annotation.ClassPathScanningCandidateComponentProvider - Identified candidate
42:15.524 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper - Found @SpringBootConfiguration co
42:15.591 [main] DEBUG org.springframework.boot.test.context.SpringBootTestContextBootstrapper - @TestExecutionListeners is not p
42:15.591 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper - Loaded default TestExecutionListe
42:15.600 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper - Using TestExecutionListeners: [or
42:15.602 [main] DEBUG org.springframework.test.context.support.AbstractDirtiesContextTestExecutionListener - Before test class:
42:15.633 [main] DEBUG org.springframework.test.context.support.TestPropertySourceUtils - Adding inlined properties to environmen


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

2-12-08 00:42:16.062  INFO 31662 --- [           main] c.s.sslserver.SslServerApplicationTests  : Starting SslServerApplicationTe
2-12-08 00:42:16.068  INFO 31662 --- [           main] c.s.sslserver.SslServerApplicationTests  : No active profile set, falling
2-12-08 00:42:18.825  INFO 31662 --- [           main] o.s.s.concurrent.ThreadPoolTaskExecutor  : Initializing ExecutorService 'a
2-12-08 00:42:19.350  INFO 31662 --- [           main] c.s.sslserver.SslServerApplicationTests  : Started SslServerApplicationTes
FO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 4.457 s - in com.snhu.sslserver.SslServerApplicationTests
2-12-08 00:42:19.690  INFO 31662 --- [extShutdownHook] o.s.s.concurrent.ThreadPoolTaskExecutor  : Shutting down ExecutorService '
FO]
FO] Results:
FO]
FO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
```

7. Summary

The code has been refactored keeping in mind many of the areas of security from the Vulnerability Assessment Process Flow Diagram. The program was reviewed for code errors and code quality to practice secure error handling and secure coding practices and patterns. Cryptography was also at use in the program. A string input was taken and then encrypted using industry standard levels of encryption and a checksum was created for validation. Additionally, since the string input is being taken from the user, the security area of input validation was taken into consideration. However, since any user input was desired to create a string and then just encrypt it, there was no validation that needed to be performed on that input. The last security area of concern was APIs. When creating the checksum API, work was done to ensure secure API interactions. The only input into the API is a string entered by the user. That string is then encrypted and a checksum of that value is returned.

8. Industry Standard Best Practices

To maintain the current level of security for this application, several industry standard best practices were applied. First a static analysis was performed on the system dependencies to search for any vulnerabilities. Then additional analysis and review were performed on the code looking for any security issues. This further review was done on the controllers, view, and APIs for the program. Applying industry standard best practices for secure coding helps to limit vulnerabilities and security threats to an application. By producing secure programs, this in turn will improve the company’s overall wellbeing and reputation.