



**GLOBAL RAIN**

**Practices for Secure Software Report**

## Table of Contents

DOCUMENT REVISION HISTORY .....	3
CLIENT.....	3
INSTRUCTIONS.....	ERROR! BOOKMARK NOT DEFINED.
DEVELOPER .....	4
1. ALGORITHM CIPHER .....	4
2. CERTIFICATE GENERATION .....	4
3. DEPLOY CIPHER.....	4
4. SECURE COMMUNICATIONS .....	5
5. SECONDARY TESTING.....	5
6. FUNCTIONAL TESTING .....	5
7. SUMMARY .....	6
8. INDUSTRY STANDARD BEST PRACTICES .....	7

#### Document Revision History

Version	Date	Author	Comments
1.0	10/14/2025	Gianna Screen	

#### Client



## Developer

Gianna Screen

### 1. Algorithm Cipher

After reviewing the specific needs of Artemis Financial, I recommend deploying the Advanced Encryption Standard (AES) algorithm cipher. AES was developed in 2001 in response to attackers exploiting the limitations of the Data Encryption Standard (DES) and its 64-bit key size. Due to this restricted length, DES became increasingly vulnerable to brute-force attacks, allowing hackers to eventually guess the key after numerous attempts (Haldar, 2024). AES resolves this vulnerability by supporting a fixed block size of 128 bits and key sizes of 128, 192, or 256 bits, making encryption both more secure and efficient.

AES works by converting readable plaintext data into ciphertext, which appears as a randomized sequence of numbers and characters. It is a symmetric encryption algorithm, meaning the same key is used to both encrypt and decrypt data (Kumar, 2020). Since Artemis Financial manages not only sensitive information but also large volumes of it, this is an ideal choice because it allows for faster encryption and decryption. These attributes are especially important because, even if attackers were to exploit a system with weak input validation or other structural flaws, the data they obtained would remain unreadable and ultimately useless without the correct decryption key. This further highlights the importance of implementing cryptographic protections such as AES-128, particularly when handling the sensitive data Artemis Financial oversees. In addition to its larger block size and symmetric key implementation, I recommend AES over the other cipher algorithms listed by Oracle because it has consistently proven secure over time and has been widely adopted by top organizations, including the United States government (Haldar, 2024).

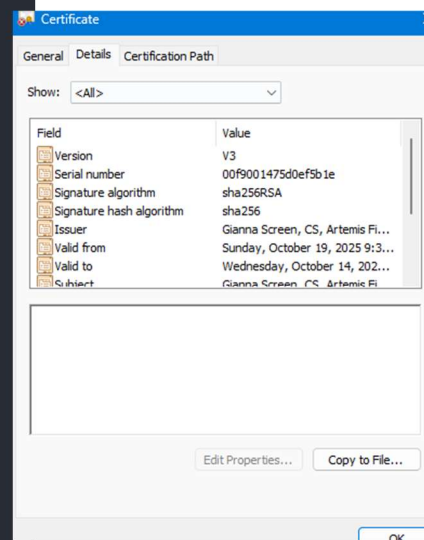
### 2. Certificate Generation

```
PS C:\Users\giann> keytool.exe -export -alias selfsigned -storepass GNSPSW@ -file server.cer -keystore keystore.jks
Certificate stored in file <server.cer>
PS C:\Users\giann> keytool.exe -printcert -file server.cer
Owner: CN=Gianna Screen, OU=CS, O=Artemis Financial, L=Providence, ST=RI, C=US
Issuer: CN=Gianna Screen, OU=CS, O=Artemis Financial, L=Providence, ST=RI, C=US
Serial number: f9001475d0ef5b1e
Valid from: Sun Oct 19 21:32:57 EDT 2025 until: Wed Oct 14 21:32:57 EDT 2026
Certificate fingerprints:
    SHA1: 9C:5C:2A:E1:12:23:1F:F7:16:CE:02:BE:11:EA:1F:26:D9:A9:52:B0
    SHA256: D5:4C:55:96:26:3E:E2:F8:35:BB:0B:3C:30:1C:16:1E:3F:EB:FA:10:A6:B4:C7:30:70:35:D9:B6:38:BE:57:DB
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

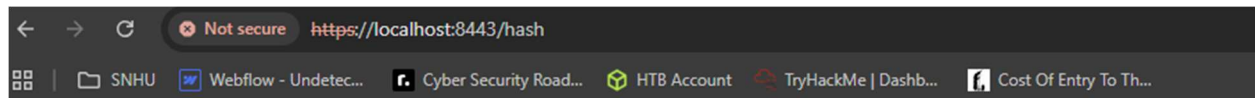
Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 40 BD A8 55 1D 70 2F CD 7E 47 33 1F 7A 72 5F FA @..U.p/..G3.zr_.
0010: 39 CD CB 13 9...
]
]

PS C:\Users\giann>
```



### 3. Deploy Cipher

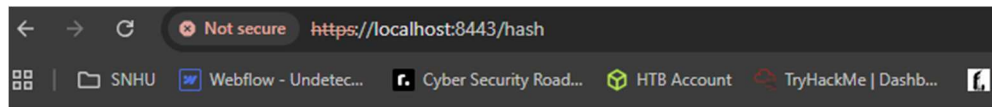


data: Creator: Gianna Screen || Hello World Check Sum!

Checksum Value: f76e926af49ee552a73ae34ce1b2fc48089e3b40a731e748452fccc181626b74

## 4. Secure Communications

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



data: Creator: Gianna Screen || Hello World Check Sum!

Checksum Value: f76e926af49ee552a73ae34ce1b2fc48089e3b40a731e748452fccc181626b74

Certificate Viewer: Gianna Screen

General

Details

Issued To

Common Name (CN)  
Organization (O)  
Organizational Unit (OU)

Gianna Screen  
Artemis Financial  
CS

Issued By

Common Name (CN)  
Organization (O)  
Organizational Unit (OU)

Gianna Screen  
Artemis Financial  
CS

Validity Period

Issued On  
Expires On

Sunday, October 19, 2025 at 12:21:09 AM  
Wednesday, October 14, 2026 at 12:21:09 AM

SHA-256 Fingerprints

Certificate  
Public Key

7358d2a9c2b1afc1541845181204036ac6a7cee714c62c9b0ebc4dee0ac  
e8695  
8bf472c13e64d9b788af8e103bf212a53c08250079e08692b4b8b2a10  
6d6a

## 5. Secondary Testing

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

```

1  https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
2  https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
3  https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
4  https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
5  https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
6  https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
7  https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
8  https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
9  https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
10 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
11 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
12 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
13 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
14 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
15 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
16 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
17 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
18 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
19 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
20 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
21 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
22 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
23 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
24 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
25 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
26 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
27 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
28 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
29 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
30 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
31 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
32 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
33 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
34 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
35 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
36 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
37 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
38 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
39 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
40 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
41 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
42 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
43 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
44 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
45 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
46 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
47 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
48 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
49 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
50 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
51 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
52 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
53 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
54 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
55 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
56 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
57 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
58 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
59 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
60 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
61 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
62 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
63 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
64 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
65 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
66 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
67 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
68 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
69 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
70 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
71 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
72 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
73 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
74 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
75 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
76 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
77 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
78 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
79 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
80 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
81 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
82 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
83 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
84 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
85 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
86 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
87 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
88 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
89 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
90 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
91 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
92 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
93 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
94 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
95 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
96 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
97 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
98 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
99 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz
100 https://www.apache.org/dist/maven-4.0.0-rc1/maven-4.0.0-rc1-binaries.tgz

```



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 as

[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: 12.1.8
- Report Generated On: Sun, 19 Oct 2025 02:00:47 -0400
- Dependencies Scanned: 49 (30 unique)
- Vulnerable Dependencies: 15
- Vulnerabilities Found: 158
- Vulnerabilities Suppressed: 0
- ...

Summary

## 6. Functional Testing

```

1 package com.snhu.sslserver;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5 import org.apache.tomcat.util.buf.HexUtils;
6 import java.security.MessageDigest;
7 import java.security.NoSuchAlgorithmException;
8
9 import org.springframework.web.bind.annotation.RequestMapping;
10 import org.springframework.web.bind.annotation.RestController;
11
12 @SpringBootApplication
13 public class SslServerApplication {
14
15     public static void main(String[] args) {
16         SpringApplication.run(SslServerApplication.class, args);
17     }
18
19 }
20 //FIXME: Add route to enable check sum return of static data example: String data = "Hello World Check Sum!";
21
22 @RestController
23 class ServerController{
24     @RequestMapping("/hash")
25     public String myHash()throws NoSuchAlgorithmException {
26         String data = "Creator: Gianna Screen || Hello World Check Sum!";
27         // Create MessageDigest object
28         MessageDigest message_digest = MessageDigest.getInstance("SHA-256");
29         byte[] byte_array = message_digest.digest(data.getBytes());
30         String checksum = HexUtils.toHexString(byte_array);
31         return "<ps> data: "+data+"</ps>"+ "<ps>Checksum Value: "+checksum+"</ps>";
32     }
33 }
34

```

```

2025-10-19 02:16:51.579 INFO 29356 --- [main] com.snhu.sslserver.SslServerApplication : Starting SslServerApplication on Gianna with PID 29356 (C:\Users\giann\Down
2025-10-19 02:16:51.581 INFO 29356 --- [main] com.snhu.sslserver.SslServerApplication : No active profile set, falling back to default profiles: default
2025-10-19 02:16:52.539 INFO 29356 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8443 (https)
2025-10-19 02:16:52.545 INFO 29356 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-10-19 02:16:52.546 INFO 29356 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.30]
2025-10-19 02:16:52.619 INFO 29356 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2025-10-19 02:16:52.619 INFO 29356 --- [main] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization completed in 1003 ms
2025-10-19 02:16:53.120 INFO 29356 --- [main] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
2025-10-19 02:16:53.494 INFO 29356 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8443 (https) with context path ''
2025-10-19 02:16:53.701 INFO 29356 --- [main] com.snhu.sslserver.SslServerApplication : Started SslServerApplication in 2.392 seconds (JVM running for 2.644)

```

## 7. Summary

The provided code base was refactored to comply with security testing protocols primarily through the implementation of cryptography using the SHA-256 hashing algorithm. Upon reviewing Artemis Financial's existing program, the vulnerability assessment process flow diagram was used to identify security gaps, with the most significant issue being the absence of cryptographic hashing for data protection. In the refactored version, sensitive data processed by the program is now cryptographically hashed, converting original plaintext into an unreadable and randomized sequence of characters.

In addition to implementing hashing, secure communication vulnerabilities were mitigated through the implementation of HTTPS protocol in place of HTTP, which ensures that communication between the browser and server is encrypted using self-signed certificates. Additionally, security vulnerabilities were assessed through secondary and functional testing. Secondary testing utilized the OWASP Dependency Check, which identifies known vulnerabilities within a program's dependencies, and functional testing involved manual code review to ensure there were no syntax errors and the program functioned as intended. For this code base, in addition to the refactored security features, I updated the Java and OWASP Dependency Check plugin versions to the most recent releases in the pom.xml file, which was essential for successfully building and running the program.

When thinking about the process for how each layer of security was added to the program, each step depended on how the previous one performed. For example, the first step was generating a digital certificate, stored as a CER file, to enable HTTPS encryption. This was followed by implementing the hash algorithm so that sensitive data displayed on the website appeared as a hashed sequence. With that complete and running, the application was configured to use the HTTPS protocol, which is dependent on the certificate already being generated and implemented appropriately within the codebase. Once those aspects were complete, the final steps were to verify that security requirements and protocols were met through testing, which was accomplished via manual review and OWASP dependency checks.

## **8. Industry Standard Best Practices**

In order to mitigate known security vulnerabilities, following industry-standard best practices is most important. For this project, this was accomplished by following the outline provided in the Vulnerability Assessment Process Flow Diagram and applying those checkpoints to the project workflow. Examples of implementing industry standard best practices include ensuring clean, maintainable code through manual code review and adding comments where necessary for clarity. Additionally, implementing the hashing algorithm, switching from HTTP to HTTPS protocol for secure communications, updating dependencies to current versions, and testing for known vulnerabilities using the OWASP Dependency Check tool all classify as following industry standard best practices. These practices are important because these guidelines and recommendations were established based on previous experience with cyberattacks and the severe consequences that occur when they are not followed.

Following industry standards also ensures system compatibility and reliability, as most dependencies and APIs are designed with the expectation that developers will adhere to these security practices. When best practices are ignored, it not only creates security vulnerabilities but can also lead to integration failures and broken functionality.

## Citations

Haldar, M. (2024, March 4). What is AES? How does it work? Encryption Consulting.  
<https://www.encryptionconsulting.com/education-center/what-is-aes/>