



GLOBAL RAIN

Practices for Secure Software Report

Table of Contents

| | |
|--|---|
| DOCUMENT REVISION HISTORY | 3 |
| CLIENT..... | 3 |
| INSTRUCTIONS..... | 3 |
| 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..... | 5 |
| 8. INDUSTRY STANDARD BEST PRACTICES..... | 5 |

Document Revision History

| Version | Date | Author | Comments |
|---------|-----------|-------------|----------|
| 1.0 | 2/21/2026 | AARON VOGEL | |

Client



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

AARON VOGEL

1. Algorithm Cipher

Artemis Financial needs a dependable way to protect financial and client data. AES-256 is a good choice because it provides strong security without slowing down systems. The 256-bit key size boosts protection and lowers the chance of brute-force attacks.

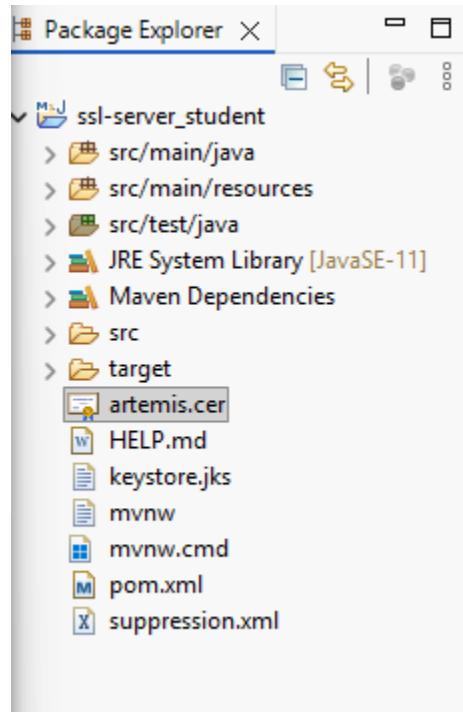
AES uses the same secret key to encrypt and decrypt data, so managing that key carefully is crucial to keep information secure.

Since AES uses the same key for encryption and decryption, keeping that key safe is very important. Encryption also depends on good random numbers, like initialization vectors (IVs), to make sure the encrypted data stays unpredictable.

To verify data integrity, the application uses the SHA-256 hashing algorithm. SHA-256 creates a unique 256-bit hash that represents the input data. Unlike encryption, hashing is one-way and helps detect any changes instead of hiding the data.

Older algorithms like DES and 3DES are now outdated because their keys are weaker and computers are more powerful. AES is the current industry standard thanks to its strong security, good performance, and wide use.

2. Certificate Generation



3. Deploy Cipher



Data: Hello AARON VOGEL - CS305 Project Two - 2026-02-21
Algorithm: SHA-256
Checksum: ef15d8c44f0c5902fe5e3d96ddd87c7c53dd6e81e69a4965c248818044103f8b

4. Secure Communications



Data: Hello AARON VOGEL - CS305 Project Two - 2026-02-21
Algorithm: SHA-256
Checksum: ef15d8c44f0c5902fe5e3d96ddd87c7c53dd6e81e69a4965c248818044103f8b

5. Secondary Testing

```
Problems Javadoc Declaration Console
SslServerApplication [Java Application] C:\Users\Aaron\p2pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe (Feb 21, 2026, 12:26:52 PM elapsed: 0:08:33) [pid: 11056]

:: Spring Boot :: (v2.2.4.RELEASE)

2026-02-21 12:26:56.732 INFO 11056 --- [main] com.snu.sslserver.SslServerApplication : Starting SslServerApplication on DESKTOP-3V0111C with PID 11056 (C:\Users\Aaron\OneDrive\Documents\CS)
2026-02-21 12:26:56.735 INFO 11056 --- [main] com.snu.sslserver.SslServerApplication : No active profile set, falling back to default profiles: default
2026-02-21 12:26:58.437 INFO 11056 --- [main] o.s.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8443 (https)
2026-02-21 12:26:58.451 INFO 11056 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2026-02-21 12:26:58.452 INFO 11056 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.30]
2026-02-21 12:26:58.572 INFO 11056 --- [main] o.a.c.c.C.[Tomcat].[localhost].[] : Initializing Spring embedded WebApplicationContext
2026-02-21 12:26:59.418 INFO 11056 --- [main] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization completed in 1787 ms
2026-02-21 12:26:59.418 INFO 11056 --- [main] o.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
2026-02-21 12:27:00.145 INFO 11056 --- [main] o.s.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8443 (https) with context path ''
2026-02-21 12:27:00.145 INFO 11056 --- [main] com.snu.sslserver.SslServerApplication : Started SslServerApplication in 3.787 seconds (JVM running for 6.751)
2026-02-21 12:27:05.198 INFO 11056 --- [nio-8443-exec-5] o.a.c.c.C.[Tomcat].[localhost].[] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2026-02-21 12:27:05.198 INFO 11056 --- [nio-8443-exec-5] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2026-02-21 12:27:05.224 INFO 11056 --- [nio-8443-exec-5] o.s.web.servlet.DispatcherServlet : Completed initialization in 26 ms
```

6. Functional Testing

```
Problems Javadoc Declaration Console
SslServerApplication [Java Application] C:\Users\Aaron\p2pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe (Feb 21, 2026, 12:26:52 PM elapsed: 0:08:33) [pid: 11056]

:: Spring Boot :: (v2.2.4.RELEASE)

2026-02-21 12:26:56.732 INFO 11056 --- [main] com.snu.sslserver.SslServerApplication : Starting SslServerApplication on DESKTOP-3V0111C with PID 11056 (C:\Users\Aaron\OneDrive\Documents\CS)
2026-02-21 12:26:56.735 INFO 11056 --- [main] com.snu.sslserver.SslServerApplication : No active profile set, falling back to default profiles: default
2026-02-21 12:26:58.437 INFO 11056 --- [main] o.s.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8443 (https)
2026-02-21 12:26:58.451 INFO 11056 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2026-02-21 12:26:58.452 INFO 11056 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.30]
2026-02-21 12:26:58.572 INFO 11056 --- [main] o.a.c.c.C.[Tomcat].[localhost].[] : Initializing Spring embedded WebApplicationContext
2026-02-21 12:26:59.418 INFO 11056 --- [main] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization completed in 1787 ms
2026-02-21 12:26:59.418 INFO 11056 --- [main] o.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
2026-02-21 12:27:00.145 INFO 11056 --- [main] o.s.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8443 (https) with context path ''
2026-02-21 12:27:05.198 INFO 11056 --- [nio-8443-exec-5] o.a.c.c.C.[Tomcat].[localhost].[] : Started SslServerApplication in 3.787 seconds (JVM running for 6.751)
2026-02-21 12:27:05.198 INFO 11056 --- [nio-8443-exec-5] o.s.web.servlet.DispatcherServlet : Initializing Spring DispatcherServlet 'dispatcherServlet'
2026-02-21 12:27:05.224 INFO 11056 --- [nio-8443-exec-5] o.s.web.servlet.DispatcherServlet : Completed initialization in 26 ms
```

7. Summary

In this project, we updated the Artemis Financial app to boost security and enable safe communications. I set up HTTPS by configuring SSL and creating a self-signed certificate with Java Keytool. This means data sent between the client and server is now encrypted.

I added a checksum verification feature using the SHA-256 hashing algorithm. This helps the app check data integrity by creating a unique hash for each data string. If the data changes, the checksum will be different, so tampering can be detected.

I ran security tests with the OWASP Dependency-Check tool. The scan showed that our changes did not add any new vulnerabilities. I also completed functional tests to make sure the app works properly without errors.

Overall, these updates improved transport security, enhanced data integrity verification, and aligned the application with modern secure coding standards.

8. Industry Standard Best Practices

I followed industry best practices during the refactoring process. I chose modern cryptographic algorithms like AES-256 for encryption and SHA-256 for hashing. These are widely trusted and secure when used correctly.

I secured communication by switching the application from HTTP to HTTPS. SSL/TLS protects sensitive financial data from being intercepted during transmission. Correctly setting up certificates makes sure connections are both authenticated and encrypted.

I added static security testing with OWASP Dependency-Check to find any known vulnerable dependencies. This helps me manage risks proactively and keep the software secure over time.

Following these best practices protects client data, lowers organizational risk, and ensures it meets current security standards.