# CS 305 Module Three Project 1

Jessica Ayer

[Jessica.ayer@snhu.edu](mailto:Jessica.ayer@snhu.edu)

CS-305-T5602 – Software Security 22EW5

Instructor: Dr. Vivian Lyon

05/22/2022

****

# CS 305 Project One

**Artemis Financial Vulnerability Assessment Report**

Table of Contents

[Document Revision History 3](#_Toc32574607)

[Client 3](#_Toc32574608)

[Instructions 3](#_Toc32574609)

[Developer 4](#_Toc32574610)

[1. Interpreting Client Needs 4](#_Toc32574611)

[2. Areas of Security 4](#_Toc32574612)

[3. Manual Review 4](#_Toc32574613)

[4. Static Testing 4](#_Toc32574614)

[5. Mitigation Plan 4](#_Toc32574615)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **05/22/22** | **Jessica Ayer** |  |

## Client



## Instructions

Deliver this completed vulnerability assessment report, identifying your findings of security vulnerabilities and articulating recommendations for next steps to remedy the issues you have found.

Respond to the five steps outlined below and include your findings. Replace the bracketed text on all pages with your own words. If you choose to include images or supporting materials, be sure to insert them throughout.

## Developer

Jessica Ayer

## 1. Interpreting Client Needs

Security is a top priority for a financial institution. Artemis Financial not only depends on their customers feeling safe but also is required to comply with strict government regulations. As a financial planner consulting company, Artemis Financial has access to their client's personal information including all their identification numbers, insurance policies, investment accounts, and retirement funds. Communications within the financial institution must be secure whether it be gained in person, transmitted locally, or transmitted globally. As a Financial institution, there will always be the need to protect against external threats. “Financial organizations have become the prime target of cyber-attack” and have been documented experiencing as “many as 300 times more cyber-attacks than other companies” (Layton, 2021). Attacks on financial institutions are often Advanced Persistent Threats meaning they “infiltrate networks and conduct long-term operations” which go undetected by the systems security for prolonged periods “devastating the institutions customers and system” (Layton, 2021). Therefore, Artemis Financial needs to modernize their system and update their security.

## 2. Areas of Security

Looking at the Vulnerability Assessment Process Flow, there are several areas that are applicable to Artemis Financials software application. Following the flow of the diagram, the first area of vulnerability that is applicable is Input Validation. The system utilized both a CRUD cycle and a REST API, both of which will require input that needs to be securely validated. Next, we need to pay attention to APIs/Secure API Interactions. The API will need to utilize RESTful API to interact with a database and to securely display sensitive data to users. Due to the transferred data being sensitive, Cryptography will be important to address. At Global Rain we are working on the server side of the Client/Server system and therefore, it is an area of vulnerability that will play a role in this modernization. In addition to all of the previously mentioned vulnerabilities, we need to make sure that secure code error handling is implemented. “The most common problem is when detailed internal error messages are displayed to the hacker (Ferragamo et al). We want to limit hackers from knowing how the system operates so that they are not given easy access to information that might help them launch a cyber-attack. Last, Code Quality/Secure Coding Practices/Patterns are a must. The code should be developed with secure practices in mind to try and eliminate vulnerabilities before they even arise.

## 3. Manual Review

|  |
| --- |
| In the application properties we can see that the system is not secured with HTTPS. Port 8081 only supports HTTP. |
| The system does not have an authentication scheme. Acess controlls should be added to the calsses that contain/display sensitive data. In the customer class lones 7-9 would benefit with access controls when showing the user information. |
| The CRUDController (line 13) and GreetingCongroller (line 16) Request Parameter should include input validation and verification to prevent overflow and directory traversal attacks (*Secure coding Guidelines for Java SE*, 2022). |

## 4. Static Testing

Information for the following vulnerabilities was obtained through the National Vulnerability Database ([NVD - Home (nist.gov)](https://nvd.nist.gov/).

* **Bcprov-jdk150n-1.46.jar**: The Bouncy Castle Crypto package is a Java implementation of cryptographic algorithms. See dependency in Pom file bellow.
* Text

  Description automatically generated
  + **CVE-2018-1000613**: Legion of the Bouncy Castle Java Cryptography APIs makes use of Externally Controlled Input to Select Classes or Code vulnerability in XMSS/XMSS^MT private key deserialization. This may result in the execution of unexpected code. The attack is exploitable with a handcrafted private key that includes references to unexpected classes picked up from the class path for the executing application. **Solution:** Update to version 1.6 or later (MITRE, 07/09/2018).
  + **CVE-2015-6644**: Bouncy Castle in Android before 5.1.1 LMY49F and 6.0 before 2016-01-01 allows attackers to obtain sensitive information with a crafted application. **Solution:** Update to the latest version of Android where possible (Android, 01/06/2016).
  + **CVE-2015-1000342**, **CVE-2016-10000338**, **CVE-2016-1000341**, **CVE-2016-1000343**, **CVE-2016-1000344**, **CVE-2016-1000345 CVE-2016-1000346**, **CVE-2016-1000352:**  In Bouncy Castle JCE Provider version 1.55 and earlier the DSA does not fully validate ASN.1 encoding of signature on verification. This makes it possible to inject extra elements in the sequence that makes up the signature and still have it validate. This can allow 'invisible' data into a signed structure. Version 1.55 also generates a private value assuming a 1024-bit key size. The DSA signature generation is vulnerable to timing attack. Where timings can be closely observed for the generation of signatures, the lack of blinding in 1.55, or earlier, may allow an attacker to gain information about the signature's key value the private value. (MITRE). **Solution:** Red HatFuse 7.1 security update (Red Hat).
  + **CVE-2020-15522:** Bouncy Castle BC Java before 1.66, has a timing issue within the EC math library that can expose information about the private key when an attacker is able to observe timing information for the generation of multiple deterministic ECDSA signatures (MITRE, 05/20/2021). **Solution:** Update BC-FJA to 1.0.1.2 or later (Legion of the Bouncy Castle Inc, 2021).
  + **CVE-2016-1000339:** In the Bouncy Castle JCE Provider version 1.55 and earlier the primary engine class used for AES was AESFastEngine. Due to the highly table-driven approach used in the algorithm, the data channel on the CPU can be monitored. The lookup table accesses are sufficient to leak information on the AES key being used. There was also a leak in AESEngine. **Solution:** Upgrade to version 1.56 (MITRE, 06/04/2018).
  + **CVE-2020-26939:** In Legion of the Bouncy Castle BC before version 1.61 and BC-FJA before version 1.0.1.2, attackers can obtain sensitive information about a private exponent because of Observable Differences in Behavior to Error Inputs. Sending invalid ciphertext that decrypts to a short payload in the OAEP Decoder could result in the throwing of an early exception, potentially leaking some information about the private exponent of the RSA private key performing the encryption (MITRE, 11/02/2020). **Solution:** Upgrade to version 1.61 (WhiteSource, 2020).
  + **CVE-2015-7940:** The Bouncy Castle Java library before 1.51 does not validate a point is withing the elliptic curve, which makes it easier for remote attackers to obtain private keys via a series of crafted elliptic curve Diffie Hellman (ECDH) key exchanges (MITRE, 11/09/2015). **Solution:** Update to version 1.53 (openSUSE, 2015).
  + **CVE-2018-5382**: The default BKS keystore use an HMAC that is only 16 bits long, which can allow an attacker to compromise the integrity of a BKS keystore. **Solution:** Do not rely on BKS version 1 keystore files are not cryptographically sound. A more robust keystore format should be used instead (CERT, 2018).
  + **CVE-2013-1624:** The TLS implementation in the Bouncy Castle Java library before 1.48 does not properly consider timing side-channel attacks on a noncompliant MAC check operation during the processing of malformed CBC padding, which allows remote attackers to conduct distinguishing attacks and plaintext-recovery attacks via statistical analysis of timing data for crafted packets (MITRE, 02/08/2013). **Solution:**  Install update version (Red Hat, 2014).
* **Hibernate-validator-6.0.18.Final.jar:** Hibernate's Bean Validation (JSR-380) reference implementation.
  + **CVE-2020-10693:** A bug in the message interpolation processor enables invalid EL expressions to be evaluated as if they were valid. This flaw allows attackers to bypass input sanitation controls that developers may have put in place when handling user-controlled data in error messages **Solution:** Update, fixed in version hibernate-validator 7.0.0.Alpha2, 6.1.5.Final, and 6.0.2.0Final (Red Hat, Inc., 2020)
* **Jackson-databind-2.10.2.jar:** General data-binding functionality for Jackson works on core streaming API.
  + **CVE-2020-25649:** The FasterXML Jackson Databind did not have entity expansion secured properly. This allows vulnerability to XML external entity attacks **Solution:** Update, fixed in version Jackson-databind-2.11.0 (Red Hat, Inc., 2020).
  + **CVE-2020-36518:** jackson-databind before 2.13.0 allows a Java StackOverflow exception and denial of service through a large depth of nested objects (MITRE, 05/06/2022). **Solution:** Critical Patch update (Oracle, 2022).
* **Log4j-api-2.12.1.jar:** The Apache Log4j API
  + **CVE-2020-9488:** Improper validation of certificate with host mismatch in Apache Log4j SMTP appender. This could allow an SMTPS connection to be intercepted by a man-in-the-middle attack which could in turn leak any log messages sent through the appender. **Solution:** Upgrade to Versions 2.13.2 (Apache Software Foundation, 2020).
* **Logback-core-1.2.3.jar:** logback-core module
  + **CVE-2021-42550:** In logback version 1.2.7 and prior versions, an attacker with the required privileges to edit configurations files could create a malicious configuration allowing to execute arbitrary code loaded from LDAP servers (Switzerland Government Common Vulnerability Program, 12/16/2021). **Solution:** Fixed and auto-remediated with auto upgrade setting enabled to version 3.9.14 builde4 (NetApp, 2022).
* **Snakeyaml-1.25.jar:** YAML 1.1 parser and emitter for java
  + **CVE-2017-18640:** The Alias feature in SnakeYAML 1.18 allows entity expansion during a load operation (MITRE, 12/11/2019). **Solution:** Update to version 1.26 (Fedora, 2020).
* **Spring-aop-5.2.3.RELEASE.jar**

**Spring-core-5.2.3.Release.jar:** Spring AOP

* + **CVE-2016-1000027:** Pivotal Spring Framework through 5.3.16 allows for potential remote code execution issue if used for Java deserialization of untrusted data. This issue may or not occur, and authentication may be required (MITRE, 01/02/2020). **Solution:** No fix available
  + **CVE-2022-22965:** A Spring MVC or Spring WebFlux application running on JDK 9+ may be vulnerable to remote code execution (RCE) via data binding. The specific exploit requires the application to run on Tomcat as a WAR deployment. If the application is deployed as a Spring Boot executable jar it is not vulnerable to the exploit. However, the nature of the vulnerability is more general, and there may be other ways to exploit it. **Solution:** Upgrade to version 5.3.x (VMware, 2022).
  + **CVE-2021-22118:** In Spring Framework, versions 5.2.x prior to 5.2.15 and versions 5.3.x prior to 5.3.7, a WebFlux application is vulnerable to a privilege escalation: by (re)creating the temporary storage directory, a locally authenticated malicious user can read or modify files that have been uploaded to the WebFlux application or overwrite arbitrary files with multipart request data. **Solution:** Upgrade to version 5.3.x (VMware, 2021).
  + **CVE-2020-5421:** In Spring Framework versions 5.2.8 and older the protections against RFD attacks may be bypassed depending on the browser used using a jsessionid path parameter. **Solution:** Upgrade to version 5.2.9 (VMware, 2020).
  + **CVE-2022-22950:** In Spring Framework versions 5.3.16 and, it is possible for a user to provide a specially crafted SpEL expression that may cause a denial-of-service condition. **Solution:** Upgrade to version 5.3.17+ (VMware, 2022).
  + **CVE-2022-22968:** In Spring Framework version 5.3.18 and older the patterns for disallowedFields on a DataBinder are case sensitive meaning a field is not effectively protected unless listed with both upper and lower case for the first character of the field, including upper and lower case for the first character of all nested fields within the property path. **Solution:** Upgrade to newer version (VMware 2022).
  + **CVE-2021-22060**, **CVE-2021-22096:** In Spring Framework versions 5.3.13 and older it is possible for a user to provide malicious input causing the insertion of additional log entries. **Solution:** Upgrade to version 5.3.13 (VMware, 2022).
* **Spring-boot-2.2.4.RELEASE.jar:** Spring Boot See dependency in Pom file bellow.
* Text

  Description automatically generated
  + **CVE-2022-27772:** spring-boot versions prior to version v2.2.11.RELEASE was vulnerable to temporary directory hijacking. (MITRE, 03/30/2022). **Solution:** Upgrade to version 2.2.11.RELEASE (WhiteSource).
* **Tomcat-embed-core-9.0.30.jar**

**Tomcat-embed-wbsocket-9.0.30.jar:** Core Tomcat implementation

* + **CVE-2020-1938:** Tomcat treats AJP connections as having higher trust than a similar HTTP connection. If such connections are available to an attacker, they can be exploited. In Apache Tomcat shipped with an AJP Connector enabled by default that listened on all configured IP addresses. This vulnerability report identified a mechanism that allowed returning arbitrary files from anywhere in the web application and processing any file in the web application as a JSP. Further, if the web application allowed file upload and stored those files within the web application then this, along with the ability to process a file as a JSP, made remote code execution possible. **Solution:** Update to version 9.0.31 (Apache Software Foundation, 02/24/2020).
  + **CVE-2020-11996:** A specially crafted sequence of HTTP/2 requests sent to Apache Tomcat could trigger high CPU usage for several seconds. If a sufficient number of such requests were made on concurrent HTTP/2 connections, the server could become unresponsive (Apache Software Foundation, 06/26/2020). **Solution:** Upgrade to Apache Tomcat 10.0.0-M6 or later (Thomas, 2020).
  + **CVE-2020-13934:** An h2c direct connection to Apache Tomcat did not release the HTTP/1.1 processor after the upgrade to HTTP/2. If a sufficient number of such requests were made, an OutOfMemoryException could occur leading to a denial of service (Apache Software Foundation, 07/14/2020). **Solution:** Upgrade to Apache Tomcat 10.0.0-M7 or later (Thomas, 2020).
  + **CVE-2020-13935:** The payload length in a WebSocket frame was not correctly validated in Apache Tomcat 9.0.0.M1 to 9.0.36. Invalid payload lengths could trigger an infinite loop. Multiple requests with invalid payload lengths could lead to a denial of service (Apache Software Foundation, 07/14/2020). **Solution:** Upgrade to Apache Tomcat 10.0.0-M7 or later (Thomas, 2020).
  + **CVE-2020-17527:** Apache Tomcat could re-use an HTTP request header value from the previous stream received on an HTTP/2 connection for the request associated with the subsequent stream. While this would most likely lead to an error and the closure of the HTTP/2 connection, it is possible that information could leak between requests (Apache Software Foundation, 12/03/2020). **Solution:** Upgrade to Apache Tomcat 10.0.0-M10 or later (Thomas, 2020).
  + **CVE-2021-25122:** When responding to new h2c connection requests, Apache Tomcat could duplicate request headers and a limited amount of request body from one request to another meaning user A and user B could both see the results of user A's request. (Apache Software Foundation, 03/01/2021). **Solution:** Upgrade to Apache Tomcat 10.0.2 or later (Thomas, 2021).
  + **CVE-2021-41079:** Apache Tomcat does not properly validate incoming TLS packets. When Tomcat was configured to use NIO+OpenSSL or NIO2+OpenSSL for TLS, a specially crafted packet could be used to trigger an infinite loop resulting in a denial of service (Apache Software Foundation, 09/16/2021). **Solution:** Upgrade to Apache Tomcat 10.0.4 or later (Thomas, 2021).
  + **CVE-2021-42340:** The fix for bug 63362 introduced a memory leak. The object introduced to collect metrics for HTTP upgrade connections was not released for WebSocket connections once the connection was closed. This could lead to a denial of service via an OutOfMemoryError (Apache Software Foundation, 10/14/2021). **Solution:** Upgrade to Apache Tomcat 10.0.0-M5 or later (Thomas, 2021).
  + **CVE-2020-9484**, **CVE-2021-25329:** An attacker is able to control the contents and name of a file on the server, the server is configured to use the PersistenceManager with a FileStore, the PersistenceManager is configured with sessionAttributeValueClassNameFilter="null" or a sufficiently lax filter to allow the attacker provided object to be deserialized, and the attacker knows the relative file path from the storage location used by FileStore to the file the attacker has control over; then, using a specifically crafted request, the attacker will be able to trigger remote code execution via deserialization of the file under their control. Note that all of conditions must be true for the attack to succeed (Apache Software Foundation, 05/20/2020). **Solution:** Upgrade to Apache Tomcat 10.0.0-M5 or later (Thomas, 2020).
  + **CVE-2021-30640:** A vulnerability in the JNDI Realm of Apache Tomcat allows an attacker to authenticate using variations of a valid username and/or to bypass some of the protection provided by the LockOut Realm. (Apache Software Foundation, 07/12/2021). **Solution:** Upgrade to Apache Tomcat 10.0.6 or later (Thomas, 2021).
  + **CVE-2021-2422:** When serving resources from a network location using the NTFS file system, Apache Tomcat was susceptible to JSP source code disclosure in some configurations. The root cause was the unexpected behavior of the JRE API File.getCanonicalPath() which in turn was caused by the inconsistent behavior of the Windows API in some circumstances (Apache Software Foundation, 01/14/2021). **Solution:** Upgrade to Apache Tomcat 10.0.0-M10 or later (Thomas, 2021).
  + **CVE-2021-33037:** Apache Tomcat did not correctly parse the HTTP transfer-encoding request header in some circumstances leading to the possibility to request smuggling when used with a reverse proxy. Specifically, Tomcat incorrectly ignored the transfer encoding header if the client declared it would only accept an HTTP/1.0 response, Tomcat honored the identify encoding, and Tomcat did not ensure that, if present, the chunked encoding was the final encoding (Apache Software Foundation, 07/12/2021). **Solution:** Upgrade to Apache Tomcat 10.0.7 or later (Thomas, 2021).
  + **CVE-2019-17569**, **CVE-2020-1935:** The refactoring present in Apache introduced a regression that resulted in invalid Transfer-Encoding headers were incorrectly processed leading to a possibility of HTTP Request Smuggling if Tomcat was located behind a reverse proxy that incorrectly handled the invalid Transfer-Encoding header in a particular manner. Such a reverse proxy is considered unlikely (Apache Software Foundation, 02/24/2020). **Solution:** Upgrade to Apache Tomcat 9.0.31 or later (Thomas, 2020).
  + **CVE-2020-13943:** If an HTTP/2 client connecting to Apache Tomcat exceeded the agreed maximum number of concurrent streams for a, it was possible that a subsequent request made on that connection could contain HTTP headers - including HTTP/2 pseudo headers - from a previous request rather than the intended headers. This could lead to users seeing responses for unexpected resources (Apache Software Foundation, 10/12/2020). **Solution:** Upgrade to Apache Tomcat 10.0.0-M8 or later (Thomas, 2020).

## 5. Mitigation Plan

From the initial manual review we can see several areas of vulnerability that need to be mitigated. Frist input validation needs to be added by defining wrappers around Native methods to safely perform necessary Input validation prior to the invocation of the native method and protect from buffer overflow attacks (Secure Coding Guidelines for Java SE, 2022). Security needs to be upgraded by switching from HTTP protocol to HTTPS by obtain SSL certificate and editing the .htaccess file (Ayodeji, 2019). Resource limit checks need to be implemented to prevent integer overflow that could lead to Denial of Service (Secure Coding Guidelines for Java SE, 2022). Log error handling needs to be implemented to help detect malicious behavior and prevent stack traces from displaying in a user’s browser (Manico & Detlefsen, 2014). Make special care not to log highly sensitive information (Secure Coding Guidelines for Java SE, 2022). Access controllers need to be added to check permissions and prevent callback methods from bridging security (Secure Coding Guidelines for Java SE, 2022).

After running a Dependency Check for files using spring-data-rest-webmvc we can see that several dependencies need to be upgraded to the newest versions and security patches downloaded. See the about Static testing results to determine appropriate updates.

**References:**

Android Open Source Project (January 4, 2016). *Nexus Security Bulletin-January 2016.* Source. <https://source.android.com/security/bulletin/2016-01-01>

Apache Software Foundation (04/13/2020). *Add support for specifying an SSL configuration for SmtpAppender.* <https://issues.apache.org/jira/browse/LOG4J2-2819>

Ayodeji, Bolaji (June 13, 2019). *How to redirect HTTP to HTTPS Using .htaccess*. FreeCodeCamp. https://www.freecodecamp.org/news/how-to-redirect-http-to-https-using-htaccess/

Baeldung (December 15, 2021). *Validating RequestParams and PathVariables in Spring.* Baeldung. <https://www.baeldung.com/spring-validate-requestparam-pathvariable>

CERT/CC. *Software Engineering Institute, Cert coordination Center: Vulnerability Nores Database*. <https://www.kb.cert.org/vuls/>

Fedora (04/25/2020).  *[Security] Fedora 31 Update: snakeaml-1.26-1.fc31.* <https://lists.fedoraproject.org/archives/list/package-announce@lists.fedoraproject.org/message/PTVJC54XGX26UJVVYCXZ7D25X3R5T2G6/>

Ferragamo, J., Wichers, Bird, J. Kingthorin. *Improper Error Handling.*  OWASP https://owasp.org/www-community/Improper\_Error\_Handling

javaTpoint. *Implementing Authentication with Spring Security*. https://www.javatpoint.com/restful-web-services-basic-authentication-with-spring-security

Legion of the Bouncy Castle Inc. (May 18, 2021). *CVE 2020 15522.*  GitHub Inc. https://github.com/bcgit/bc-csharp/wiki/CVE-2020-15522

Layton, Roslyn (March 17, 2021). *Hackers Are Targeting U.S. Banks, And Hardware May Give Them An Open Door.* <https://www.forbes.com/sites/roslynlayton/2021/03/17/hackers-are-targeting-us-banks-and-hardware-may-give-them-an-open-door/?sh=343c381b14dc>

Manico Jim, Detlefsen, August (September 2014). *Iron-Clad Java.* McGraw Hill Computing. Retrieved from https://learning.oreilly.com/library/view/iron-clad-java/9780071835886/

MITRE *Solving Problems for A safer World.* <https://www.mitre.org/>

National Institute of Standards and Technology (September 20, 2021). National Vulnerability database. <https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aredhat&cpe_product=cpe%3A%2F%3Aredhat%3Ahibernate_validator&cpe_version=cpe%3A%2F%3Aredhat%3Ahibernate_validator%3A6.0.18>

National Vulnerability Database. National Institute of Standards and Technology U.S. Department of Commerce. https://nvd.nist.gov/

NetApp Product Security(02/16/2022). *CVE-*2021-42550 Logback Vulnerability in NetApp Products. <https://security.netapp.com/advisory/ntap-20211229-0001/>

OpenSuse security (11/4/2015). *[Security-announce] openSUSE-SU-2015:1911-1 important: Security update for bouncycastle*. <https://lists.opensuse.org/archives/list/security-announce@lists.opensuse.org/message/ZDOWMUYB7VUPT46DHIZUFPHDFEH2YNBG/>

Oracle. *Critical Patch Updates, Security Alerts and Bulletins.* <https://www.oracle.com/security-alerts/>

Oracle(January 2022). *Secure Coding Guidelines for Java SE* <https://www.oracle.com/java/technologies/javase/seccodeguide.html>

Red Hat Inc. *RHSA-2018:2669-Security Advisory.* <https://access.redhat.com/errata/RHSA-2018:2669>

Thomas, Mark. *announce@tomcat.apache.org.* Apache Pony Mail. <https://lists.apache.org/list.html?announce@tomcat.apache.org>

VMware Tanzu. *Notable Vulnerabilities in Dependencies (USNs) and legacy VMware Tanzu Advisories.*  VMware, Inc. <https://tanzu.vmware.com/security>

*WhiteSource Vulnerability Database: The largest open source vulnerability DB*. WhieSource. https://www.whitesourcesoftware.com/vulnerability-database/