# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **07/21/2024** | **Cody VanGosen** | **Added information associated with Interpreting Client Needs, Areas of Security, Manual Code Review, Static Testing, & a Mitigation Plan** |

## Client



## Instructions

Submit this completed vulnerability assessment report. Replace the bracketed text with the relevant information. In this report, identify your security vulnerability findings and recommend the next steps to remedy the issues you have found.

* Respond to the five steps outlined below and include your findings.
* Respond using your own words. You may also include images or supporting materials. If you include them, make certain to insert them in the relevant locations in the document.
* Refer to the Project One Guidelines and Rubric for more detailed instructions about each section of the template.

## Developer

Cody VanGosen

**1. Interpreting Client Needs**

Determine your client’s needs and potential threats and attacks associated with the company’s application and software security requirements. Consider the following questions regarding how companies protect against external threats based on the scenario information:

* What is the value of secure communications to the company?
* Are there any international transactions that the company produces?
* Are there governmental restrictions on secure communications to consider?
* What external threats might be present now and in the immediate future?
* What modernization requirements must be considered, such as the role of open-source libraries and evolving web application technologies?

Secure communications are paramount to data integrity and security. Being that Artemis Financial processes a wide array of sensitive financial information this is an especially important concern for the business. Data transmissions between clients and servers must be encrypted while also preventing unauthorized users from accessing resources while also helping to mitigate risks of data breaches.

The application must ensure it is maintaining secure communication channels to comply with local and international governmental policies. When conducting international transactions there are various data protection regulations that must be followed according to the governance standards of each country.

Government restrictions must be clearly understood, and application solutions must be clearly defined to best follow these requirements. Certain countries might facilitate the use of specific encryption standards or even restrict usage of specific transmission pathways.  
  
As with any modern software applications there are several external threats to data security and integrity. Such threats would include SQL injection attacks, XSS cross-site scripting attacks, or even man-in-the-middle attacks. These risks present potential flaws related to security, data integrity, application availability, and data security.  
  
Effective modernization techniques related to the use of open-source libraries should be of primary concern. These libraries should be kept up to date with the current patches to provide security against emerging vulnerabilities. Outdated libraries provide a large security risk to any vulnerabilities that might have been discovered. Emerging web and technology requirements should be considered for application implementation. Modern security practices such as including secure API interactions and following standard cryptography practices will help provide better overall security.

**2. Areas of Security**

Refer to the vulnerability assessment process flow diagram. Identify which areas of security apply to Artemis Financial’s software application. Justify your reasoning for why each area is relevant to the software application.

Input validation is a key consideration to provide an increased sense of security for the software solution. The input fields in the CRUDController and DocData must be validated in an attempt to mitigate risks associated with SQL injection.  
  
APIs present an inherent risk due to the external communication pathways they use. Careful review and analysis of APIs and their available source code should be used in conjunction with review of associated risks and vulnerabilities of said APIs to assess security risk imposed by their use. The RESTful API used in the GreetingController should be carefully secured to ensure prevention of data leaks while also preventing unauthorized user access.

Correct error handling is a requirement for improving overall application security. Currently, in DocData the SQL errors are only printed without being properly handled. Adding more robust error handling will help increase overall security due to prevention of unauthorized access to application resources.

Employing best practices such as encapsulation will increase overall data security and integrity. This is especially important when considering private or sensitive data fields. Due to the nature of security and privacy associated with financial operations this practice is necessary for secure data such as account numbers, routing numbers, and private customer personal information.

**3. Manual Review**

Continue working through the vulnerability assessment process flow diagram. Identify all vulnerabilities in the code base by manually inspecting the code.

The Customer class contains an issue associated with the private nature of account\_number field being private, while the method showInfo() returns this value. This poses a risk of exposure to sensitive information if this method were accessible through the API or user interface.

The DocData class possesses a potential vulnerability related to the read\_document method which includes a specific coded database connection. This could lead to security vulnerabilities if database credentials were revealed or improperly configured.

The Greeting class currently stores content and content2 without any checks or verification. This poses a risk to security if an unauthorized or malicious entity seeks to manipulate this content in an unsafe manner.

The CRUD class currently contains an issue related to the constructor copying the content field to content2 without any data validation. This can result in issues related to leakage or inconsistencies with the associated data and accompanying information.

The CRUDController class contains an issue with the /read endpoint invoking the DocData with it returning the string representation without any security checks or data validation occurring.

The myDateTime class poses a risk associated with the methods of retrieveDateTime and setMyDateTime are currently acting as placeholders that do not actually perform any validation or error handling.

The GreetingController currently contains a distinct lack of user authentication or authorization checks. This poses an extreme risk to allow unauthorized users to gain access to information or resources they do not have the respective authority to view.

Currently there are issues related to the CRUDController having a lack of validation or sanitization of its associated parameters. This poses a potential risk to injection-based attacks.

There is currently a significant risk of SQL injection attacks. This can be seen in the DocData class where the SQL query is constructed without the use of parameterized queries, which leads to the increase in risk associated to vulnerabilities of SQL injection attacks.

The DocData additionally poses some increased risks associated with error handling. The current implementation possesses a vulnerability related to the database connections being directly printed, which could in turn lead to issues of exposing stack traces or other sensitive information.

**4. Static Testing**

Run a dependency check on Artemis Financial’s software application to identify all security vulnerabilities in the code. Record the output from the dependency-check report. Include the following items:

* The names or vulnerability codes of the known vulnerabilities
* A brief description and recommended solutions provided by the dependency-check report
* Any attribution that documents how this vulnerability has been identified or documented previously
* [**CVE-2013-1624**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2013-1624)   
  The Bouncy Castle Java library does not consider timing side-channel attacks. This can result in remote attackers having the ability to perform plaintext-recovery attacks or distinguishing attacks with statistical analysis techniques. References: REDHAT - [RHSA-2014:0371](http://rhn.redhat.com/errata/RHSA-2014-0371.html) REDHAT - [RHSA-2014:0372](http://rhn.redhat.com/errata/RHSA-2014-0372.html)

[**CVE-2022-27772**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-27772)

Spring-boot versions prior to version v2.2.11 have a vulnerability related to temporary directory hijacking. This can lead to exposure of resources incorrectly. References:

* MISC - <https://github.com/JLLeitschuh/security-research/security/advisories/GHSA-cm59-pr5q-cw85>
* OSSINDEX - [[CVE-2022-27772] CWE-668: Exposure of Resource to Wrong Sphere](https://ossindex.sonatype.org/vulnerability/CVE-2022-27772?component-type=maven&component-name=org.springframework.boot%2Fspring-boot&utm_source=dependency-check&utm_medium=integration&utm_content=5.3.0)

[**CVE-2021-42550**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-42550)

Logback version 1.2.7 and earlier have a vulnerability associated with attacker’s possessing the necessary privileges to edit configuration files could create a malicious configuration to execute arbitrary code. References:

* CONFIRM - <http://logback.qos.ch/news.html>
* CONFIRM - <https://cert-portal.siemens.com/productcert/pdf/ssa-371761.pdf>
* CONFIRM - <https://security.netapp.com/advisory/ntap-20211229-0001/>
* FULLDISC - [20220721 Open-Xchange Security Advisory 2022-07-21](http://seclists.org/fulldisclosure/2022/Jul/11)
* MISC - <http://packetstormsecurity.com/files/167794/Open-Xchange-App-Suite-7.10.x-Cross-Site-Scripting-Command-Injection.html>
* MISC - <https://github.com/cn-panda/logbackRceDemo>
* MISC - <https://jira.qos.ch/browse/LOGBACK-1591>
* OSSINDEX - [[CVE-2021-42550] CWE-502: Deserialization of Untrusted Data](https://ossindex.sonatype.org/vulnerability/CVE-2021-42550?component-type=maven&component-name=ch.qos.logback%2Flogback-classic&utm_source=dependency-check&utm_medium=integration&utm_content=5.3.0)
* OSSINDEX - [[CVE-2021-42550] CWE-502: Deserialization of Untrusted Data](https://ossindex.sonatype.org/vulnerability/CVE-2021-42550?component-type=maven&component-name=ch.qos.logback%2Flogback-core&utm_source=dependency-check&utm_medium=integration&utm_content=5.3.0)

[**CVE-2020-9488**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-9488)

Improper validation of certificates with a host mismatch could result in SMTPS connections to be intercepted by man-in-the-middle attacks leading to exposure of log messages. References:

* - [[db-torque-dev] 20200715 Build failed in Jenkins: Torque4-trunk #685](https://lists.apache.org/thread.html/r393943de452406f0f6f4b3def9f8d3c071f96323c1f6ed1a098f7fe4%40%3Ctorque-dev.db.apache.org%3E)
* - [[db-torque-dev] 20210127 Re: Items for our (delayed) quarterly report to the board?](https://lists.apache.org/thread.html/rc2dbc4633a6eea1fcbce6831876cfa17b73759a98c65326d1896cb1a%40%3Ctorque-dev.db.apache.org%3E)
* - [[db-torque-dev] 20210128 Antwort: Re: Items for our (delayed) quarterly report to the board?](https://lists.apache.org/thread.html/rd5d58088812cf8e677d99b07f73c654014c524c94e7fedbdee047604%40%3Ctorque-dev.db.apache.org%3E)
* - [[flink-issues] 20210510 [GitHub] [flink] zentol opened a new pull request #15879: [FLINK-22407][build] Bump log4j to 2.24.1](https://lists.apache.org/thread.html/r45916179811a32cbaa500f972de9098e6ee80ee81c7f134fce83e03a%40%3Cissues.flink.apache.org%3E)
* - [[hive-dev] 20201207 [jira] [Created] (HIVE-24500) Hive - upgrade log4j 2.12.1 to 2.13.2+ due to CVE-2020-9488](https://lists.apache.org/thread.html/r4db540cafc5d7232c62e076051ef661d37d345015b2e59b3f81a932f%40%3Cdev.hive.apache.org%3E)
* - [[hive-dev] 20210216 [jira] [Created] (HIVE-24787) Hive - upgrade log4j 2.12.1 to 2.13.2+ due to CVE-2020-9488](https://lists.apache.org/thread.html/r4d5dc9f3520071338d9ebc26f9f158a43ae28a91923d176b550a807b%40%3Cdev.hive.apache.org%3E)
* - [[hive-issues] 20201207 [jira] [Assigned] (HIVE-24500) Hive - upgrade log4j 2.12.1 to 2.13.2+ due to CVE-2020-9488](https://lists.apache.org/thread.html/ra632b329b2ae2324fabbad5da204c4ec2e171ff60348ec4ba698fd40%40%3Cissues.hive.apache.org%3E)
* - [[hive-issues] 20201207 [jira] [Updated] (HIVE-24500) Hive - upgrade log4j 2.12.1 to 2.13.2+ due to CVE-2020-9488](https://lists.apache.org/thread.html/r4ed1f49616a8603832d378cb9d13e7a8b9b27972bb46d946ccd8491f%40%3Cissues.hive.apache.org%3E)
* - [[hive-issues] 20201207 [jira] [Work started] (HIVE-24500) Hive - upgrade log4j 2.12.1 to 2.13.2+ due to CVE-2020-9488](https://lists.apache.org/thread.html/r1fc73f0e16ec2fa249d3ad39a5194afb9cc5afb4c023dc0bab5a5881%40%3Cissues.hive.apache.org%3E)
* - [[hive-issues] 20201208 [jira] [Updated] (HIVE-24500) Hive - upgrade log4j 2.12.1 to 2.13.2+ due to CVE-2020-9488](https://lists.apache.org/thread.html/r65578f3761a89bc164e8964acd5d913b9f8fd997967b195a89a97ca3%40%3Cissues.hive.apache.org%3E)
* - [[hive-issues] 20201208 [jira] [Work logged] (HIVE-24500) Hive - upgrade log4j 2.12.1 to 2.13.2+ due to CVE-2020-9488](https://lists.apache.org/thread.html/r9776e71e3c67c5d13a91c1eba0dc025b48b802eb7561cc6956d6961c%40%3Cissues.hive.apache.org%3E)
* - [[hive-issues] 20210125 [jira] [Work logged] (HIVE-24500) Hive - upgrade log4j 2.12.1 to 2.13.2+ due to CVE-2020-9488](https://lists.apache.org/thread.html/r3d1d00441c55144a4013adda74b051ae7864128ebcfb6ee9721a2eb3%40%3Cissues.hive.apache.org%3E)
* - [[hive-issues] 20210209 [jira] [Resolved] (HIVE-24500) Hive - upgrade log4j 2.12.1 to 2.13.2+ due to CVE-2020-9488](https://lists.apache.org/thread.html/r33864a0fc171c1c4bf680645ebb6d4f8057899ab294a43e1e4fe9d04%40%3Cissues.hive.apache.org%3E)
* - [[hive-issues] 20210216 [jira] [Assigned] (HIVE-24787) Hive - upgrade log4j 2.12.1 to 2.13.2+ due to CVE-2020-9488](https://lists.apache.org/thread.html/r5a68258e5ab12532dc179edae3d6e87037fa3b50ab9d63a90c432507%40%3Cissues.hive.apache.org%3E)
* - [[hive-issues] 20210216 [jira] [Resolved] (HIVE-24787) Hive - upgrade log4j 2.12.1 to 2.13.2+ due to CVE-2020-9488](https://lists.apache.org/thread.html/r22a56beb76dd8cf18e24fda9072f1e05990f49d6439662d3782a392f%40%3Cissues.hive.apache.org%3E)
* - [[hive-issues] 20210218 [jira] [Updated] (HIVE-24787) Hive - upgrade log4j 2.12.1 to 2.13.2+ due to CVE-2020-9488](https://lists.apache.org/thread.html/ra051e07a0eea4943fa104247e69596f094951f51512d42c924e86c75%40%3Cissues.hive.apache.org%3E)
* - [[kafka-dev] 20200514 [jira] [Created] (KAFKA-9996) upgrade zookeeper to 3.5.8 to address security vulnerabilities](https://lists.apache.org/thread.html/r7e5c10534ed06bf805473ac85e8412fe3908a8fa4cabf5027bf11220%40%3Cdev.kafka.apache.org%3E)
* - [[kafka-dev] 20200514 [jira] [Created] (KAFKA-9997) upgrade log4j lib to address CVE-2020-9488](https://lists.apache.org/thread.html/rf1c2a81a08034c688b8f15cf58a4cfab322d00002ca46d20133bee20%40%3Cdev.kafka.apache.org%3E)
* - [[kafka-jira] 20200514 [jira] [Created] (KAFKA-9996) upgrade zookeeper to 3.5.8 to address security vulnerabilities](https://lists.apache.org/thread.html/r8e96c340004b7898cad3204ea51280ef6e4b553a684e1452bf1b18b1%40%3Cjira.kafka.apache.org%3E)
* - [[kafka-jira] 20200514 [jira] [Created] (KAFKA-9997) upgrade log4j lib to address CVE-2020-9488](https://lists.apache.org/thread.html/r0a2699f724156a558afd1abb6c044fb9132caa66dce861b82699722a%40%3Cjira.kafka.apache.org%3E)
* - [[kafka-jira] 20200515 [jira] [Commented] (KAFKA-9997) upgrade log4j lib to address CVE-2020-9488](https://lists.apache.org/thread.html/r48bcd06049c1779ef709564544c3d8a32ae6ee5c3b7281a606ac4463%40%3Cjira.kafka.apache.org%3E)
* - [[kafka-users] 20210617 vulnerabilities](https://lists.apache.org/thread.html/r2721aba31a8562639c4b937150897e24f78f747cdbda8641c0f659fe%40%3Cusers.kafka.apache.org%3E)
* - [[mina-dev] 20210225 [jira] [Created] (FTPSERVER-500) Security vulnerability in common/lib/log4j-1.2.17.jar](https://lists.apache.org/thread.html/rf9fa47ab66495c78bb4120b0754dd9531ca2ff0430f6685ac9b07772%40%3Cdev.mina.apache.org%3E)
* - [[pulsar-commits] 20201215 [GitHub] [pulsar] yanshuchong opened a new issue #8967: CVSS issue list](https://lists.apache.org/thread.html/rd0e44e8ef71eeaaa3cf3d1b8b41eb25894372e2995ec908ce7624d26%40%3Ccommits.pulsar.apache.org%3E)
* - [[zookeeper-commits] 20200504 [zookeeper] branch branch-3.5 updated: ZOOKEEPER-3817: suppress log4j SmtpAppender related CVE-2020-9488](https://lists.apache.org/thread.html/rec34b1cccf907898e7cb36051ffac3ccf1ea89d0b261a2a3b3fb267f%40%3Ccommits.zookeeper.apache.org%3E)
* - [[zookeeper-commits] 20200504 [zookeeper] branch branch-3.6 updated: ZOOKEEPER-3817: suppress log4j SmtpAppender related CVE-2020-9488](https://lists.apache.org/thread.html/r48efc7cb5aeb4e1f67aaa06fb4b5479a5635d12f07d0b93fc2d08809%40%3Ccommits.zookeeper.apache.org%3E)
* - [[zookeeper-commits] 20200504 [zookeeper] branch master updated: ZOOKEEPER-3817: suppress log4j SmtpAppender related CVE-2020-9488](https://lists.apache.org/thread.html/rbc45eb0f53fd6242af3e666c2189464f848a851d408289840cecc6e3%40%3Ccommits.zookeeper.apache.org%3E)
* - [[zookeeper-dev] 20200504 [jira] [Created] (ZOOKEEPER-3817) owasp failing due to CVE-2020-9488](https://lists.apache.org/thread.html/r2f209d271349bafd91537a558a279c08ebcff8fa3e547357d58833e6%40%3Cdev.zookeeper.apache.org%3E)
* - [[zookeeper-dev] 20200504 log4j SmtpAppender related CVE](https://lists.apache.org/thread.html/r0df3d7a5acb98c57e64ab9266aa21eeee1d9b399addb96f9cf1cbe05%40%3Cdev.zookeeper.apache.org%3E)
* - [[zookeeper-issues] 20200504 [jira] [Assigned] (ZOOKEEPER-3817) owasp failing due to CVE-2020-9488](https://lists.apache.org/thread.html/rd8e87c4d69df335d0ba7d815b63be8bd8a6352f429765c52eb07ddac%40%3Cissues.zookeeper.apache.org%3E)
* - [[zookeeper-issues] 20200504 [jira] [Commented] (ZOOKEEPER-3817) owasp failing due to CVE-2020-9488](https://lists.apache.org/thread.html/r4285398e5585a0456d3d9db021a4fce6e6fcf3ec027dfa13a450ec98%40%3Cissues.zookeeper.apache.org%3E)
* - [[zookeeper-issues] 20200504 [jira] [Created] (ZOOKEEPER-3817) owasp failing due to CVE-2020-9488](https://lists.apache.org/thread.html/r8c001b9a95c0bbec06f4457721edd94935a55932e64b82cc5582b846%40%3Cissues.zookeeper.apache.org%3E)
* - [[zookeeper-issues] 20200504 [jira] [Resolved] (ZOOKEEPER-3817) owasp failing due to CVE-2020-9488](https://lists.apache.org/thread.html/rd55f65c6822ff235eda435d31488cfbb9aa7055cdf47481ebee777cc%40%3Cissues.zookeeper.apache.org%3E)
* - [[zookeeper-issues] 20200504 [jira] [Updated] (ZOOKEEPER-3817) owasp failing due to CVE-2020-9488](https://lists.apache.org/thread.html/r9a79175c393d14d760a0ae3731b4a873230a16ef321aa9ca48a810cd%40%3Cissues.zookeeper.apache.org%3E)
* - [[zookeeper-notifications] 20200504 Build failed in Jenkins: zookeeper-master-maven-owasp #489](https://lists.apache.org/thread.html/r7641ee788e1eb1be4bb206a7d15f8a64ec6ef23e5ec6132d5a567695%40%3Cnotifications.zookeeper.apache.org%3E)
* - [[zookeeper-notifications] 20200504 [GitHub] [zookeeper] symat commented on pull request #1346: ZOOKEEPER-3817: suppress log4j SmtpAppender related CVE-2020-9488](https://lists.apache.org/thread.html/rc6b81c013618d1de1b5d6b8c1088aaf87b4bacc10c2371f15a566701%40%3Cnotifications.zookeeper.apache.org%3E)
* - [[zookeeper-notifications] 20200504 [GitHub] [zookeeper] symat opened a new pull request #1346: ZOOKEEPER-3817: suppress log4j SmtpAppender related CVE-2020-9488](https://lists.apache.org/thread.html/r7e739f2961753af95e2a3a637828fb88bfca68e5d6b0221d483a9ee5%40%3Cnotifications.zookeeper.apache.org%3E)
* - <https://lists.apache.org/thread.html/rbc7642b9800249553f13457e46b813bea1aec99d2bc9106510e00ff3%40%3Ctorque-dev.db.apache.org%3E>
* - <https://lists.apache.org/thread.html/re024d86dffa72ad800f2848d0c77ed93f0b78ee808350b477a6ed987%40%3Cgitbox.hive.apache.org%3E>
* CONFIRM - <https://issues.apache.org/jira/browse/LOG4J2-2819>
* CONFIRM - <https://security.netapp.com/advisory/ntap-20200504-0003/>
* DEBIAN - [DSA-5020](https://www.debian.org/security/2021/dsa-5020)
* MISC - <https://www.oracle.com/security-alerts/cpuApr2021.html>
* MISC - <https://www.oracle.com/security-alerts/cpuapr2022.html>
* MISC - <https://www.oracle.com/security-alerts/cpujan2021.html>
* MISC - <https://www.oracle.com/security-alerts/cpujul2020.html>
* MISC - <https://www.oracle.com/security-alerts/cpuoct2020.html>
* MISC - <https://www.oracle.com/security-alerts/cpuoct2021.html>
* MLIST - [[debian-lts-announce] 20211226 [SECURITY] [DLA 2852-1] apache-log4j2 security update](https://lists.debian.org/debian-lts-announce/2021/12/msg00017.html)

[**CVE-2017-18640**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2017-18640)

The Alias feature included in SnakeYAML before v1.26 allow entity expansion during load operations. In addition, can have issues associated with XML Entity expansion. References:

* - [FEDORA-2020-23012fafbc](https://lists.fedoraproject.org/archives/list/package-announce%40lists.fedoraproject.org/message/PTVJC54XGX26UJVVYCXZ7D25X3R5T2G6/)
* - [FEDORA-2020-599514b47e](https://lists.fedoraproject.org/archives/list/package-announce%40lists.fedoraproject.org/message/CKN7VGIKTYBCAKYBRG55QHXAY5UDZ7HA/)
* - [[atlas-commits] 20200915 [atlas] branch master updated: ATLAS-3940 : Upgrade snakeyaml to a version without CVE-2017-18640 (#110)](https://lists.apache.org/thread.html/rce5c93bba6e815fb62ad38e28ca1943b3019af1eddeb06507ad4e11a%40%3Ccommits.atlas.apache.org%3E)
* - [[atlas-commits] 20200916 [atlas] 02/02: ATLAS-3940 : Upgrade snakeyaml to a version without CVE-2017-18640 (#110)](https://lists.apache.org/thread.html/r1058e7646988394de6a3fd0857ea9b1ee0de14d7bb28fee5ff782457%40%3Ccommits.atlas.apache.org%3E)

[**CVE-2020-25649**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-25649)   
There is a flaw in FasterXML Jackson Databind, where it does not have entity expansion properly secured. This allow for XXE attacks resulting in risks to data integrity. References:

* - [FEDORA-2021-1d8254899c](https://lists.fedoraproject.org/archives/list/package-announce%40lists.fedoraproject.org/message/6X2UT4X6M7DLQYBOOHMXBWGYJ65RL2CT/)
* - [[druid-commits] 20201208 [GitHub] [druid] jihoonson opened a new pull request #10655: Bump up jackson-databind to 2.10.5.1](https://lists.apache.org/thread.html/r2b6ddb3a4f4cd11d8f6305011e1b7438ba813511f2e3ab3180c7ffda%40%3Ccommits.druid.apache.org%3E)
* - [[flink-issues] 20210121 [GitHub] [flink-shaded] HuangXingBo opened a new pull request #93: [FLINK-21020][jackson] Bump version to 2.12.1](https://lists.apache.org/thread.html/ra95faf968f3463acb3f31a6fbec31453fc5045325f99f396961886d3%40%3Cissues.flink.apache.org%3E)
* - [[flink-issues] 20210122 [GitHub] [flink-shaded] HuangXingBo opened a new pull request #93: [FLINK-21020][jackson] Bump version to 2.12.1](https://lists.apache.org/thread.html/r45e7350dfc92bb192f3f88e9971c11ab2be0953cc375be3dda5170bd%40%3Cissues.flink.apache.org%3E)

[**CVE-2019-17569**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2019-17569)

Apache Tomcat versions included in this implementation have issues with regression. This results in risks associated with HTTP Request smuggling if Tomcat was behind a reverse proxy. References:

* - [[tomee-commits] 20200320 [jira] [Created] (TOMEE-2790) TomEE plus(7.0.7) is affected by CVE-2020-1935 & CVE-2019-17569 vulnerabilities](https://lists.apache.org/thread.html/rc31cbabb46cdc58bbdd8519a8f64b6236b2635a3922bbeba0f0e3743%40%3Ccommits.tomee.apache.org%3E)
* - [[tomee-commits] 20200323 [jira] [Commented] (TOMEE-2790) TomEE plus(7.0.7) is affected by CVE-2020-1935 & CVE-2019-17569 vulnerabilities](https://lists.apache.org/thread.html/r7bc994c965a34876bd94d5ff15b4e1e30b6220a15eb9b47c81915b78%40%3Ccommits.tomee.apache.org%3E)
* CONFIRM - <https://security.netapp.com/advisory/ntap-20200327-0005/>
* DEBIAN - [DSA-4673](https://www.debian.org/security/2020/dsa-4673)

[**CVE-2020-10693**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-10693)

There is a flaw in hibernate validator version 6.1.2 where the message interpolation processor allows for invalid EL expressions to be processed as if they were valid. This allows for bypass of input sanitation. References:

* - [[portals-pluto-dev] 20210714 [jira] [Closed] (PLUTO-791) Upgrade to hibernate-validator-6.0.20.Final due to CVE-2020-10693 and CVE-2019-10219](https://lists.apache.org/thread.html/rd418deda6f0ebe658c2015f43a14d03acb8b8c2c093c5bf6b880cd7c%40%3Cpluto-dev.portals.apache.org%3E)
* - [[portals-pluto-dev] 20210714 [jira] [Created] (PLUTO-791) Upgrade to hibernate-validator-6.0.20.Final due to CVE-2020-10693 and CVE-2019-10219](https://lists.apache.org/thread.html/rf9c17c3efc4a376a96e9e2777eee6acf0bec28e2200e4b35da62de4a%40%3Cpluto-dev.portals.apache.org%3E)
* - [[portals-pluto-scm] 20210714 [portals-pluto] branch master updated: PLUTO-791 Upgrade to hibernate-validator-6.0.20.Final due to CVE-2020-10693 and CVE-2019-10219](https://lists.apache.org/thread.html/rb8dca19a4e52b60dab0ab21e2ff9968d78f4b84e4033824db1dd24b4%40%3Cpluto-scm.portals.apache.org%3E)
* CONFIRM - <https://bugzilla.redhat.com/show_bug.cgi?id=CVE-2020-10693>
* MISC - <https://www.oracle.com/security-alerts/cpuapr2022.html>
* OSSINDEX - [[CVE-2020-10693] CWE-20: Improper Input Validation](https://ossindex.sonatype.org/vulnerability/CVE-2020-10693?component-type=maven&component-name=org.hibernate.validator%2Fhibernate-validator&utm_source=dependency-check&utm_medium=integration&utm_content=5.3.0)

**CVE-2016-1000027** (OSSINDEX)

This spring framework suffers from potential remote code execution through Java deserialization of trusted data. This I dependent upon implementation and authentication settings.   
References:

* OSSINDEX - [[CVE-2016-1000027] CWE-502: Deserialization of Untrusted Data](https://ossindex.sonatype.org/vulnerability/CVE-2016-1000027?component-type=maven&component-name=org.springframework%2Fspring-web&utm_source=dependency-check&utm_medium=integration&utm_content=5.3.0)

**CVE-2022-22965** (OSSINDEX)

A Spring MVC running on JDK 9+ may be vulnerable to RCE through data binding. This requires the application to run on Tomcat in a WAR deployment.   
References:

* OSSINDEX - [[CVE-2022-22965] CWE-94: Improper Control of Generation of Code ('Code Injection')](https://ossindex.sonatype.org/vulnerability/CVE-2022-22965?component-type=maven&component-name=org.springframework%2Fspring-beans&utm_source=dependency-check&utm_medium=integration&utm_content=5.3.0)

**CVE-2021-22060** (OSSINDEX)   
The Spring Framework and older versions included issues with users providing malicious input to result in insertion of additional log entries. References:

* OSSINDEX - [[CVE-2021-22060] CWE-117: Improper Output Neutralization for Logs](https://ossindex.sonatype.org/vulnerability/CVE-2021-22060?component-type=maven&component-name=org.springframework%2Fspring-webmvc&utm_source=dependency-check&utm_medium=integration&utm_content=5.3.0)

**CVE-2022-22968** (OSSINDEX)

The Spring framework includes risks associated with fields being case sensitive which leads to results in fields not being adequately protected unless it accounts for upper- and lower-case characters of the first character of related fields. References:

* OSSINDEX - [[CVE-2022-22968] CWE-178: Improper Handling of Case Sensitivity](https://ossindex.sonatype.org/vulnerability/CVE-2022-22968?component-type=maven&component-name=org.springframework%2Fspring-context&utm_source=dependency-check&utm_medium=integration&utm_content=5.3.0)

**CVE-2022-22950 (OSSINDEX)**The Spring framework results in vulnerabilities associated with a user providing SpEL expressions that can pose a risk of denial of service. References:

* OSSINDEX - [[CVE-2022-22950] CWE-770: Allocation of Resources Without Limits or Throttling](https://ossindex.sonatype.org/vulnerability/CVE-2022-22950?component-type=maven&component-name=org.springframework%2Fspring-expression&utm_source=dependency-check&utm_medium=integration&utm_content=5.3.0)

**5. Mitigation Plan**

Interpret the results from the manual review and static testing report. Then identify the steps to mitigate the identified security vulnerabilities for Artemis Financial’s software application.

Secure input handling is a necessity to increase overall security. The implementation of input validation in CRUDController and DocData should be included. This would require the ability to sanitize user inputs as well as mitigate risks associated with SQL injection.

Database connections pose a significant risk if sensitive information and connection information are not adequately protected. It is important to store database credentials in a secure configuration file or a solution such as an environment variable rather than hardcoding these details.

Error handling is essential in maintaining data integrity, security, and overall program flow. For this reason, implementing proper error handling within DocData will help alleviate risks associated with exposing sensitive user information and transaction details.

API inclusion is essential in program function, but special review and security considerations must be performed. Authentication and authorization mechanisms for associated APIs should be included to alleviate risks, especially those included at sensitive endpoints.

The inclusion of encapsulation provides a means for more safely storing secure or sensitive data fields and their associated information. For this reason, all sensitive or confidential data fields should only be accessible through their associated getter methods.

Routine maintenance and scheduled operations of statis code analysis will help to ensure security during the maintenance and live deployment/update stages of production. Static code analysis tools must be kept updated and run regularly to ensure wider coverage against emerging vulnerabilities.

The usage of dependencies and various libraries is widespread throughout the implementation of this project. This further demonstrates the importance of following routine updates for dependencies and secure libraries to ensure coverage against identified vulnerabilities.