

# 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** | **21 Mar 2021** | **Scott Baker** |  |

## 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

Scott Baker

## 1. Interpreting Client Needs

*Determine your client’s needs and potential threats and attacks associated with their application and software security requirements. Consider the following 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 about secure communications to consider?*
* *What external threats might be present now and in the immediate future?*
* *What are the “modernization” requirements that must be considered, such as the role of open-source libraries and evolving web application technologies?*

Being a financial institution, the value of secure communications to the company is extremely high. Users/customers are going to be communicating PII (Personal Identifying Information) through this API and because of this, the communications need to be as secure as possible.

From the brief scenario description, there does not seem to be any international transactions that will be taking place. However, there could be the option for these transactions to occur in the future; ultimately, we should take that into account to possibly help future proof the application.

Governmental restrictions around these secure communications are generally based around ensuring that security and safeguards are in place when it comes to healthcare, financial, social security numbers, and other specific types of data. (NCSL.org)

The two biggest threats that are present and will be in the future are hackers with their myriad of tools they can use to break into a system, and employee oversight. There are the ever-prevalent SQL injections that will be used to attempt security breaches, among other types of hacks. There is also the potential for employees to submit/transmit sensitive information through unsecure means.

As a means of “modernization”, Artemis Financial is wanting to update the software with the most current and effective software security, due to the open-source libraries that are in use; also, due to evolving web application technologies, new exploits can be created that need to be mitigated through these new security measures.

## 2. Areas of Security

*Referring to the Vulnerability Assessment Process Flow Diagram, identify which areas of security are applicable to Artemis Financial’s software application. Justify your reasoning for why each area is relevant to the software application.*

Because of the need for security with these financial communications, validating user input is an absolute necessity. Sensitive information is going to be inputted and transmitted to the server by users of the system. This can leave the door open for hackers to use denial-of-service attacks.

Because the system uses APIs, securing these API interactions is a critical aspect of securing the overall software. There are multiple ways for hackers to exploit weaknesses in a software's API like spoofing, man-in-the-middle attacks, and reverse engineering. (nordicapis.com)

The system also needs to utilize cryptography to help protect the communications that are going to and from the server/system. The amount of PII that will be flowing in and out of the system is going to be massive, so there is a major security concern/vulnerability if this information is not encrypted.

In relation to client/server vulnerabilities, having a secure connection between the two helps to prevent both client-side and server-side attacks. This is another obvious necessity for the system, especially when dealing with sensitive information.

The secure handling of errors in the system is an obvious need. If the different errors that the system can run into was public knowledge, it would be like giving hackers partially filled out schematics of how the system is built/operates. This would then give hackers plenty of information about how to exploit the system, and potentially steal sensitive data from the servers/users.

Code quality should always be a concern, because if the developers are not following secure coding practices/patterns, this is sloppy and leaves the system more vulnerable to exploitation that it already can be.

Encapsulating the data using secure data structures provides an extra layer of protection by not allowing a single class to “know” everything about the complete structure. This follows the military concept of compartmentalization of data. If someone were to be captured and “interrogated” for information, then the “interrogators” could not get all the information that is necessary to completely exploit an aspect of the military.

## 3. Manual Review

*Continue working through the Vulnerability Assessment Process Flow Diagram. Identify all vulnerabilities in the code base by manually inspecting the code.*

After looking through the code base, there were multiple issues that were discovered. One issue that was found was there seems to be no account security. There is no usage of passwords in the program, except for in the DocData.java class, and a password is only used for accessing a database named test. Another flaw is there is no reverification of the user and their credentials while using the different function of the software. We also find that there are duplicate public CRUD methods found in CRUD.java with both methods writing information to the same variables. There is also no input verification happening in the program.

## 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 dependency check report. Include the following:*

1. *The names or vulnerability codes of the known vulnerabilities*
2. *A brief description and recommended solutions provided by the dependency check report*
3. *Attribution (if any) that documents how this vulnerability has been identified or documented previously*

**bcprov-jdk15on-1.46.jar Dependency**

[**CVE-2013-1624**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2013-1624)

The TLS implementation in the Bouncy Castle Java library does not consider timing side-channel attacks on certain operations. This leads to distinguishing and plaintext-recovery attacks through analyzing the timing data for crafted packets. Recommended fix is to update to a newer version.

References:

* MISC - <http://www.isg.rhul.ac.uk/tls/TLStiming.pdf>
* MLIST - [[oss-security] 20130205 Re: CVE request: TLS CBC padding timing flaw in various SSL / TLS implementations](http://openwall.com/lists/oss-security/2013/02/05/24)
* OSSINDEX - [[CVE-2013-1624] Cryptographic Issues](https://ossindex.sonatype.org/vuln/7fa5f67c-cb63-43af-bdac-d129db38fb7c?component-type=maven&component-name=org.bouncycastle.bcprov-jdk15on&utm_source=dependency-check&utm_medium=integration&utm_content=6.1.2)

**CVE-2015-6644**

Bouncy Castle in android before 5.1.1 and 6.0 before January of 2016 allowed attackers to get sensitive information through crafted applications. Recommended fix is to update to a newer version.

References:

* OSSINDEX - [[CVE-2015-6644] Information disclosure](https://ossindex.sonatype.org/vuln/3a59870b-28b3-4b6b-86b0-9629ebe9de40?component-type=maven&component-name=org.bouncycastle.bcprov-jdk15on&utm_source=dependency-check&utm_medium=integration&utm_content=6.1.2)

**CVE-2015-7940**

The library before 1.51 does not validate when a point is within the elliptic curve which allows hackers to obtain private keys through a series of crafted elliptic curve Diffie Hellman key exchanges. Recommended fix is to update to a newer version.

References:

* OSSINDEX - [[CVE-2015-7940] Information Exposure, Cryptographic Issues](https://ossindex.sonatype.org/vuln/58377ddb-36ee-4586-9d29-c11f5c1e78ba?component-type=maven&component-name=org.bouncycastle.bcprov-jdk15on&utm_source=dependency-check&utm_medium=integration&utm_content=6.1.2)

[**CVE-2016-1000338**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000338)

Bouncy Castle JCE Provider version 1.55 and earlier, the ASN.1 encoding of signature on verification is not fully validated through the DSA. Recommended fix is adding a length check for the signatures.

References:

* CONFIRM - <https://github.com/bcgit/bc-java/commit/b0c3ce99d43d73a096268831d0d120ffc89eac7f#diff-3679f5a9d2b939d0d3ee1601a7774fb0>
* MISC - <https://www.oracle.com/security-alerts/cpuoct2020.html>
* MLIST - [[debian-lts-announce] 20180707 [SECURITY] [DLA 1418-1] bouncycastle security update](https://lists.debian.org/debian-lts-announce/2018/07/msg00009.html)

[**CVE-2016-1000339**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000339)

In Bouncy Castle JCE Provider version 1.55 and earlier, the highly table-driven approach through the AESFastEngine allows for data leaks if the data channel on the CPU is monitored. Recommended fix is to update to newer version or obfuscate the table use in AESFastEngine.

References:

* CONFIRM - <https://github.com/bcgit/bc-java/commit/413b42f4d770456508585c830cfcde95f9b0e93b#diff-54656f860db94b867ba7542430cd2ef0>
* CONFIRM - <https://github.com/bcgit/bc-java/commit/8a73f08931450c17c749af067b6a8185abdfd2c0#diff-494fb066bed02aeb76b6c005632943f2>
* CONFIRM - <https://security.netapp.com/advisory/ntap-20181127-0004/>

[**CVE-2016-1000341**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000341)

DSA signature generation is vulnerable to timing attack on JCE Provider version 1.55 and earlier. This could allow the attacker to get information about the signature’s k value and the private value as well.

References:

* CONFIRM - <https://github.com/bcgit/bc-java/commit/acaac81f96fec91ab45bd0412beaf9c3acd8defa#diff-e75226a9ca49217a7276b29242ec59ce>
* CONFIRM - <https://security.netapp.com/advisory/ntap-20181127-0004/>
* MISC - <https://www.oracle.com/security-alerts/cpuoct2020.html>

[**CVE-2016-1000342**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000342)

Bouncy Castle JCE Provider version 1.55 and earlier, the ASN.1 encoding of signature on verification is not fully validated through the DSA. Recommended fix is adding a length check for the signatures and header validations.

References:

* CONFIRM - <https://github.com/bcgit/bc-java/commit/843c2e60f67d71faf81d236f448ebbe56c62c647#diff-25c3c78db788365f36839b3f2d3016b9>
* CONFIRM - <https://security.netapp.com/advisory/ntap-20181127-0004/>
* MISC - <https://www.oracle.com/security-alerts/cpuoct2020.html>

[**CVE-2016-1000343**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000343)

The DSA key pair generator generates a weak private key assuming a 1024 bit key size. Recommended fix is to explicitly pass parameters to the key pair generator.

References:

* CONFIRM - <https://github.com/bcgit/bc-java/commit/50a53068c094d6cff37659da33c9b4505becd389#diff-5578e61500abb2b87b300d3114bdfd7d>
* CONFIRM - <https://security.netapp.com/advisory/ntap-20181127-0004/>
* MISC - <https://www.oracle.com/security-alerts/cpuoct2020.html>

[**CVE-2016-1000344**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000344)

Bouncy Castle JCE Provider version 1.55 and earlier allowed the use of ECB mode. This mode is regarded as unsafe. Recommended fix is to update to a newer version.

References:

* CONFIRM - <https://github.com/bcgit/bc-java/commit/9385b0ebd277724b167fe1d1456e3c112112be1f>
* CONFIRM - <https://security.netapp.com/advisory/ntap-20181127-0004/>
* MISC - <https://www.oracle.com/security-alerts/cpuoct2020.html>

[**CVE-2016-1000345**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000345)

On BC 1.55 and earlier, when the timings can be easily observed and with enough observation, when the decryption fails due to padding can be identified.

References:

* CONFIRM - <https://github.com/bcgit/bc-java/commit/21dcb3d9744c83dcf2ff8fcee06dbca7bfa4ef35#diff-4439ce586bf9a13bfec05c0d113b8098>
* CONFIRM - <https://security.netapp.com/advisory/ntap-20181127-0004/>
* MISC - <https://www.oracle.com/security-alerts/cpuoct2020.html>

[**CVE-2016-1000346**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000346)

On version 1.55 and earlier, other party DH public key is not fully validated. Invalid keys can then be used to reveal details about the other party’s private key when static Diffie-Hellman is in use. Recommended fix is to add TLS validation check for DH keys.

References:

* CONFIRM - <https://github.com/bcgit/bc-java/commit/1127131c89021612c6eefa26dbe5714c194e7495#diff-d525a20b8acaed791ae2f0f770eb5937>
* CONFIRM - <https://security.netapp.com/advisory/ntap-20181127-0004/>
* MISC - <https://www.oracle.com/security-alerts/cpuoct2020.html>

[**CVE-2016-1000352**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000352)

Bouncy Castle JCE Provider version 1.55 and earlier allowed the use of ECB mode. This mode is regarded as unsafe. Recommended fix is to update to a newer version.

References:

* CONFIRM - <https://github.com/bcgit/bc-java/commit/9385b0ebd277724b167fe1d1456e3c112112be1f>
* CONFIRM - <https://security.netapp.com/advisory/ntap-20181127-0004/>
* MISC - <https://www.oracle.com/security-alerts/cpuoct2020.html>

[**CVE-2017-13098**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2017-13098)

BouncyCastle TLS prior to version 1.0.3, when configured to use the JCE (Java Cryptography Extension) for cryptographic functions, provides a weak Bleichenbacher oracle when any TLS cipher suite using RSA key exchange is negotiated. An attacker can recover the private key from a vulnerable application. This vulnerability is referred to as "ROBOT."CWE-203 Information Exposure Through Discrepancy. Recommended fix is to confirm the size of decrypted PMS before using.

References:

* BID - [102195](http://www.securityfocus.com/bid/102195)
* CERT-VN - [VU#144389](http://www.kb.cert.org/vuls/id/144389)
* CONFIRM - <https://github.com/bcgit/bc-java/commit/a00b684465b38d722ca9a3543b8af8568e6bad5c>

[**CVE-2018-1000613**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2018-1000613) suppress

BC Java Cryptography APIs 1.58 up to 1.60 contain a CWE-470: an Externally-Controlled Input to Select Classes or Code vulnerability in private key deserialization can result in the execution of unexpected code. The attack is exploitable through a handcrafted private key that includes references to unexpected classes. Recommended fix is to add additional checking XMSS BDS tree parsing.

References:

* CONFIRM - <https://github.com/bcgit/bc-java/commit/4092ede58da51af9a21e4825fbad0d9a3ef5a223#diff-2c06e2edef41db889ee14899e12bd574>
* CONFIRM - <https://github.com/bcgit/bc-java/commit/cd98322b171b15b3f88c5ec871175147893c31e6#diff-148a6c098af0199192d6aede960f45dc>
* CONFIRM - <https://security.netapp.com/advisory/ntap-20190204-0003/>

[**CVE-2018-5382**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2018-5382)

BC BKS version 1 keystore files use an HMAC only 16 bits long, which allows the attacker to compromise the MKS-V1 keystore integrity. BC release 1.47 introduces BKS version 2 which uses a 160-bit MAC. Recommend updating to newer versions.

References:

* BID - [103453](http://www.securityfocus.com/bid/103453)
* CERT-VN - [VU#306792](https://www.kb.cert.org/vuls/id/306792)
* MISC - <https://www.bouncycastle.org/releasenotes.html>

**CVE-2020-26939**

Sending invalid ciphertext that decrypts to a short payload in the OAEP Decoder can lead to throwing an early exception and potentially leaking information about the private exponent. Recommended fix is to perform an initial raw RSA decryption to do a length check on the payload. Then either decrypt a known payload and then return failure where the original payload turns out too small, or pass the payload through for decryption where the payload is the correct size.

References:

* OSSINDEX - [[CVE-2020-26939] In Legion of the Bouncy Castle BC before 1.61 and BC-FJA before 1.0.1.2, attacke...](https://ossindex.sonatype.org/vuln/9e56f765-fe13-4d65-925a-241a8047f1b6?component-type=maven&component-name=org.bouncycastle.bcprov-jdk15on&utm_source=dependency-check&utm_medium=integration&utm_content=6.1.2)

**hibernate-validator-6.0.18.Final.jar**

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

A bug in the message interpolation processor allows invalid EL expressions to be evaluated if they were valid. This lets hackers bypass input sanitation. Recommended fix is to pass user input as an expression variable by unwrapping the context to HibernateConstraintValidatorContext.

References:

* CONFIRM - <https://bugzilla.redhat.com/show_bug.cgi?id=CVE-2020-10693>
* MISC - <https://www.ibm.com/support/pages/node/6348216>

**jackson-databind-2.10.2.jar**

[**CVE-2020-25649**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-25649)

Entity expansion was not secured properly in FasterXML Jackson Databind. This allows vulnerability to XXE attacks. Recommended fix is to upgrade to a newer version.

References:

* CONFIRM - <https://security.netapp.com/advisory/ntap-20210108-0007/>
* FEDORA - [FEDORA-2021-1d8254899c](https://lists.fedoraproject.org/archives/list/package-announce@lists.fedoraproject.org/message/6X2UT4X6M7DLQYBOOHMXBWGYJ65RL2CT/)
* MISC - <https://bugzilla.redhat.com/show_bug.cgi?id=1887664>

**log4j-api-2.12.1.jar**

[**CVE-2020-9488**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=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 leak any log messages sent through that appender.CWE-295 Improper Certificate Validation. Recommended fix is to upgrade to 2.13.2

References:

* CONFIRM - <https://issues.apache.org/jira/browse/LOG4J2-2819>
* CONFIRM - <https://security.netapp.com/advisory/ntap-20200504-0003/>
* MISC - <https://lists.apache.org/thread.html/rbc7642b9800249553f13457e46b813bea1aec99d2bc9106510e00ff3@%3Ctorque-dev.db.apache.org%3E>

**snakeyaml-1.25.jar**

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

The Alias feature in SnakeYAML 1.18 allows entity expansion during a load operation, a related issue to CVE-2003-1564. Recommended fix is to use a configuration switch to disable alias expansion.

References:

* FEDORA - [FEDORA-2020-23012fafbc](https://lists.fedoraproject.org/archives/list/package-announce@lists.fedoraproject.org/message/PTVJC54XGX26UJVVYCXZ7D25X3R5T2G6/)
* FEDORA - [FEDORA-2020-599514b47e](https://lists.fedoraproject.org/archives/list/package-announce@lists.fedoraproject.org/message/CKN7VGIKTYBCAKYBRG55QHXAY5UDZ7HA/)
* MISC - <https://bitbucket.org/asomov/snakeyaml/issues/377/allow-configuration-for-preventing-billion>

**spring-core-5.2.3.RELEASE.jar**

[**CVE-2020-5421**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-5421)

Protections from RFD attacks may be bypassed depending on the browser used by using a jsessionid path parameter. Recommended fix is to update to newer versions.

References:

* CONFIRM - <https://tanzu.vmware.com/security/cve-2020-5421>
* MISC - <https://www.oracle.com/security-alerts/cpujan2021.html>
* MLIST - [[ambari-commits] 20201019 [ambari] branch branch-2.7 updated: AMBARI-25571. Vulnerable Spring components in Ambari - CVE-2020-5398, CVE-2020-5421 (dlysnichenko) (#3246)](https://lists.apache.org/thread.html/r1c679c43fa4f7846d748a937955c7921436d1b315445978254442163@%3Ccommits.ambari.apache.org%3E)

**tomcat-embed-core-9.0.30.jar**

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

There is a regression that allows invalid Transfer-Encoding headers to be incorrectly processed. This leads to possible HTTP Request Smuggling if Tomcat was located behind a reverse proxy that did not handle the invalid T-E header in a particular manner. The reverse proxy is deemed unlikely. Recommend using the update that fixes three vulnerabilities.

References:

* CONFIRM - <https://security.netapp.com/advisory/ntap-20200327-0005/>
* DEBIAN - [DSA-4673](https://www.debian.org/security/2020/dsa-4673)
* DEBIAN - [DSA-4680](https://www.debian.org/security/2020/dsa-4680)

[**CVE-2020-11996**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-11996)

Specially crafted sequences of HTTP/2 requests can trigger a high CPU usage for several seconds; and if enough are sent, the server could become unresponsive. Recommend upgrading Tomcat to 9.0.36

References:

* CONFIRM - <https://lists.apache.org/thread.html/r5541ef6b6b68b49f76fc4c45695940116da2bcbe0312ef204a00a2e0%40%3Cannounce.tomcat.apache.org%3E>
* CONFIRM - <https://security.netapp.com/advisory/ntap-20200709-0002/>
* DEBIAN - [DSA-4727](https://www.debian.org/security/2020/dsa-4727)

[**CVE-2020-13934**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13934)

An h2c connection to certain versions of Apache Tomcat did not release the HTTP/1.1 processor after the upgrade to HTTP/2. If enough requests are made, then a DOS could occur. Recommend updating to newer version.

References:

* CONFIRM - <https://security.netapp.com/advisory/ntap-20200724-0003/>
* DEBIAN - [DSA-4727](https://www.debian.org/security/2020/dsa-4727)
* MISC - <https://lists.apache.org/thread.html/r61f411cf82488d6ec213063fc15feeeb88e31b0ca9c29652ee4f962e%40%3Cannounce.tomcat.apache.org%3E>

[**CVE-2020-13935**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13935)

Payload length in WebSocket frame was not correctly validated in select Apache Tomcat versions. The invalid length can trigger an infinite loop. Multiple of these requests can lead to a DOS. Recommend upgrading to newer version of Apache Tomcat.

References:

* CONFIRM - <https://kc.mcafee.com/corporate/index?page=content&id=SB10332>
* CONFIRM - <https://security.netapp.com/advisory/ntap-20200724-0003/>
* DEBIAN - [DSA-4727](https://www.debian.org/security/2020/dsa-4727)

[**CVE-2020-13943**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13943)

If HTTP/2 client connecting to select Apache Tomcat versions exceeds the maximum number of concurrent streams, it is possible that a subsequent request made on that connection could contain HTTP headers. Recommend upgrading to Later version of Apache Tomcat.

References:

* CONFIRM - <https://security.netapp.com/advisory/ntap-20201016-0007/>
* DEBIAN - [DSA-4835](https://www.debian.org/security/2021/dsa-4835)
* MISC - <https://lists.apache.org/thread.html/r4a390027eb27e4550142fac6c8317cc684b157ae314d31514747f307%40%3Cannounce.tomcat.apache.org%3E>

[**CVE-2020-17527**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-17527)

Select Apache Tomcat versions could re-use an HTTP request header value from the previous stream received for the request associated with the subsequent stream. This makes it possible for information to be leaked between requests. Recommend upgrading to latest version of Apache Tomcat.

References:

* CONFIRM - <https://security.netapp.com/advisory/ntap-20201210-0003/>
* DEBIAN - [DSA-4835](https://www.debian.org/security/2021/dsa-4835)
* GENTOO - [GLSA-202012-23](https://security.gentoo.org/glsa/202012-23)

[**CVE-2020-1935**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1935)

In select versions of Apache Tomcat, the HTTP header parsing code used an end-of-line parsing approach that allowed some invalid HTTP headers to be parsed as valid. This led to potential HTTP Request Smuggling if Tomcat was located behind a reverse proxy that incorrectly handled the invalid Transfer-Encoding header. Recommend upgrading to latest version of Apache Tomcat.

References:

* CONFIRM - <https://security.netapp.com/advisory/ntap-20200327-0005/>
* DEBIAN - [DSA-4673](https://www.debian.org/security/2020/dsa-4673)
* DEBIAN - [DSA-4680](https://www.debian.org/security/2020/dsa-4680)

[**CVE-2020-1938**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1938)

Because of the hierarchy of trusted connections, the Apache JServ Protocol could allow any files on the web application to be saved and processed as JSP files. This could allow a hacker the ability to remotely execute code. It is noted that mitigation is only required if an AJP port is accessible to untrusted users. It is recommended to upgrade to a later version of Apache Tomcat.

References:

* CONFIRM - <http://support.blackberry.com/kb/articleDetail?articleNumber=000062739>
* CONFIRM - <https://security.netapp.com/advisory/ntap-20200226-0002/>
* DEBIAN - [DSA-4673](https://www.debian.org/security/2020/dsa-4673)

[**CVE-2020-9484**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-9484)

Select versions of Apache Tomcat can allow a hacker remote code execution through deserialization of the file under their control. The conditions are: the hacker can control the contents and name of a file on the server, the server is using the PersistenceManager with a FileStore, the PersistenceManager is configured with sessionAttributeValueClassNameFilter=”null”, and the hacker knows the relative file path from the storage location used by FileStore. It is recommended to upgrade to later version of Apache Tomcat.

References:

* CONFIRM - <https://kc.mcafee.com/corporate/index?page=content&id=SB10332>
* CONFIRM - <https://security.netapp.com/advisory/ntap-20200528-0005/>
* DEBIAN - [DSA-4727](https://www.debian.org/security/2020/dsa-4727)

[**CVE-2021-24122**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-24122)

When serving resources from a network location using the NTFS file system, select Apache Tomcat versions were susceptible to JSP source code disclosure. Recommend upgrading to later versions of Apache Tomcat.

References:

* CONFIRM - <https://security.netapp.com/advisory/ntap-20210212-0008/>
* MISC - <https://lists.apache.org/thread.html/r1595889b083e05986f42b944dc43060d6b083022260b6ea64d2cec52%40%3Cannounce.tomcat.apache.org%3E>
* MLIST - [[announce] 20210114 [SECURITY] CVE-2021-24122 Apache Tomcat Information Disclosure](https://lists.apache.org/thread.html/r1595889b083e05986f42b944dc43060d6b083022260b6ea64d2cec52@%3Cannounce.apache.org%3E)

[**CVE-2021-25122**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-25122)

When responding to new h2c connection request, select versions of Apache Tomcat could duplicate request headers and a limited amount of request body from one request could be seen by the other request. Recommend upgrading to later versions of Apache Tomcat.

References:

* CONFIRM - [N/A](https://lists.apache.org/thread.html/r7b95bc248603360501f18c8eb03bb6001ec0ee3296205b34b07105b7%40%3Cannounce.tomcat.apache.org%3E)
* MLIST - [[announce] 20210301 [SECURITY] CVE-2021-25122 Apache Tomcat h2c request mix-up](https://lists.apache.org/thread.html/r7b95bc248603360501f18c8eb03bb6001ec0ee3296205b34b07105b7@%3Cannounce.apache.org%3E)
* MLIST - [[debian-lts-announce] 20210316 [SECURITY] [DLA 2596-1] tomcat8 security update](https://lists.debian.org/debian-lts-announce/2021/03/msg00018.html)

[**CVE-2021-25329**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-25329)

When using select versions of Apache Tomcat with a configuration edge case that was highly unlikely to be used, the Tomcat instance was still vulnerable to CVE-9496. Recommend upgrading to later versions of Apache Tomcat.

References:

* CONFIRM - [N/A](https://lists.apache.org/thread.html/rfe62fbf9d4c314f166fe8c668e50e5d9dd882a99447f26f0367474bf%40%3Cannounce.tomcat.apache.org%3E)
* MLIST - [[announce] 20210301 [SECURITY] CVE-2021-25329 Apache Tomcat Incomplete fix for CVE-2020-9484 (RCE via session persistence)](https://lists.apache.org/thread.html/rfe62fbf9d4c314f166fe8c668e50e5d9dd882a99447f26f0367474bf@%3Cannounce.apache.org%3E)
* MLIST - [[debian-lts-announce] 20210316 [SECURITY] [DLA 2596-1] tomcat8 security update](https://lists.debian.org/debian-lts-announce/2021/03/msg00018.html)

**tomcat-embed-websocket-9.0.30.jar**

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

The refactoring present in select Apache Tomcat versions introduced a regression that could result in invalid Transfer-Encoding headers that were incorrectly processed. This could lead to HTTP Request Smuggling if Tomcat is located behind a reverse proxy that incorrectly handled the invalid T-E header in a particular manner. Recommend upgrading to later version of Apache Tomcat.

References:

* CONFIRM - <https://security.netapp.com/advisory/ntap-20200327-0005/>
* DEBIAN - [DSA-4673](https://www.debian.org/security/2020/dsa-4673)
* DEBIAN - [DSA-4680](https://www.debian.org/security/2020/dsa-4680)

[**CVE-2020-11996**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-11996)

Specially crafted sequences of HTTP/2 requests can trigger a high CPU usage for several seconds; and if enough are sent, the server could become unresponsive. Recommend upgrading Tomcat to 9.0.36

References:

* CONFIRM - <https://lists.apache.org/thread.html/r5541ef6b6b68b49f76fc4c45695940116da2bcbe0312ef204a00a2e0%40%3Cannounce.tomcat.apache.org%3E>
* CONFIRM - <https://security.netapp.com/advisory/ntap-20200709-0002/>
* DEBIAN - [DSA-4727](https://www.debian.org/security/2020/dsa-4727)

[**CVE-2020-13934**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13934)

An h2c direct connection to select Apache Tomcat versions did not release the HTTP/1.1 processor after the upgrade to HTTP/2. If enough requests were made, an OutOfMemoryException could occur leading to a DOS. Recommend upgrading to later versions of Apache Tomcat.

References:

* CONFIRM - <https://security.netapp.com/advisory/ntap-20200724-0003/>
* DEBIAN - [DSA-4727](https://www.debian.org/security/2020/dsa-4727)
* MISC - <https://lists.apache.org/thread.html/r61f411cf82488d6ec213063fc15feeeb88e31b0ca9c29652ee4f962e%40%3Cannounce.tomcat.apache.org%3E>

[**CVE-2020-13935**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13935)

Payload length in a WebSocket frame was not correctly validated in select Apache Tomcat versions, which could trigger an infinite loop. Multiple of these requests could lead to a DOS. Recommend upgrading to later versions of Apache Tomcat.

References:

* CONFIRM - <https://kc.mcafee.com/corporate/index?page=content&id=SB10332>
* CONFIRM - <https://security.netapp.com/advisory/ntap-20200724-0003/>
* DEBIAN - [DSA-4727](https://www.debian.org/security/2020/dsa-4727)

[**CVE-2020-13943**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13943)

If an HTTP/2 client connecting to select Apache Tomcat versions exceeded the maximum number of concurrent streams for a connection, subsequent request made on that connection could contain HTTP headers. Recommend upgrading to later versions of Apache Tomcat.

References:

* CONFIRM - <https://security.netapp.com/advisory/ntap-20201016-0007/>
* DEBIAN - [DSA-4835](https://www.debian.org/security/2021/dsa-4835)
* MISC - <https://lists.apache.org/thread.html/r4a390027eb27e4550142fac6c8317cc684b157ae314d31514747f307%40%3Cannounce.tomcat.apache.org%3E>

[**CVE-2020-17527**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-17527)

Select Apache Tomcat versions could re-use an HTTP request header value from previous streams received on an HTTP/2 connection for a request associated with the subsequent stream. This could lead to information being leaked between requests. Recommended fix is to upgrade to later versions of Apache Tomcat.

References:

* CONFIRM - <https://security.netapp.com/advisory/ntap-20201210-0003/>
* DEBIAN - [DSA-4835](https://www.debian.org/security/2021/dsa-4835)
* GENTOO - [GLSA-202012-23](https://security.gentoo.org/glsa/202012-23)

[**CVE-2020-1935**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1935)

The HTTP header parsing code, in select version of Apache Tomcat, used an end-of-line parsing that allowed invalid HTTP headers to be parsed as valid. This could lead to HTTP Request Smuggling if Tomcat is located behind a reverse proxy that incorrectly handles the invalid T-E header. Recommend upgrading to later versions of Apache Tomcat.

References:

* CONFIRM - <https://security.netapp.com/advisory/ntap-20200327-0005/>
* DEBIAN - [DSA-4673](https://www.debian.org/security/2020/dsa-4673)
* DEBIAN - [DSA-4680](https://www.debian.org/security/2020/dsa-4680)

[**CVE-2020-1938**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1938)

Because of the hierarchy of trusted connections, the Apache JServ Protocol could allow any files on the web application to be saved and processed as JSP files. This could allow a hacker the ability to remotely execute code. It is noted that mitigation is only required if an AJP port is accessible to untrusted users. It is recommended to upgrade to a later version of Apache Tomcat.

References:

* CONFIRM - <http://support.blackberry.com/kb/articleDetail?articleNumber=000062739>
* CONFIRM - <https://security.netapp.com/advisory/ntap-20200226-0002/>
* DEBIAN - [DSA-4673](https://www.debian.org/security/2020/dsa-4673)

[**CVE-2020-8022**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-8022)

An incorrect default permissions vulnerability in select versions of Tomcat Enterprise allows local attackers to escalate from group tomcat to root. Recommended fix is to remove the write permission.

References:

* CONFIRM - <https://bugzilla.suse.com/show_bug.cgi?id=1172405>
* MLIST - [[axis-java-dev] 20210228 axis2 1.7.9 is exposed to CVE-2020-8022 via tomcat dependency](https://lists.apache.org/thread.html/r5be80ba868a11a1f64e4922399f171b8619bca4bc2039f79cf913928@%3Cjava-dev.axis.apache.org%3E)
* MLIST - [[axis-java-dev] 20210307 Re: axis2 1.7.9 is exposed to CVE-2020-8022 via tomcat dependency](https://lists.apache.org/thread.html/r393d4f431683e99c839b4aed68f720b8583bca6c35cd84adccaa02be@%3Cjava-dev.axis.apache.org%3E)

[**CVE-2020-9484**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-9484)

Select versions of Apache Tomcat can allow a hacker remote code execution through deserialization of the file under their control. The conditions are: the hacker can control the contents and name of a file on the server, the server is using the PersistenceManager with a FileStore, the PersistenceManager is configured with sessionAttributeValueClassNameFilter=”null”, and the hacker knows the relative file path from the storage location used by FileStore. It is recommended to upgrade to later version of Apache Tomcat.

References:

* CONFIRM - <https://kc.mcafee.com/corporate/index?page=content&id=SB10332>
* CONFIRM - <https://security.netapp.com/advisory/ntap-20200528-0005/>
* DEBIAN - [DSA-4727](https://www.debian.org/security/2020/dsa-4727)

[**CVE-2021-24122**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-24122) When serving resources from a network location using the NTFS file system, select versions of Apache Tomcat were susceptible to JSP source code disclosure. Recommend upgrading to later versions of Apache Tomcat.

References:

* CONFIRM - <https://security.netapp.com/advisory/ntap-20210212-0008/>
* MISC - <https://lists.apache.org/thread.html/r1595889b083e05986f42b944dc43060d6b083022260b6ea64d2cec52%40%3Cannounce.tomcat.apache.org%3E>
* MLIST - [[announce] 20210114 [SECURITY] CVE-2021-24122 Apache Tomcat Information Disclosure](https://lists.apache.org/thread.html/r1595889b083e05986f42b944dc43060d6b083022260b6ea64d2cec52@%3Cannounce.apache.org%3E)

[**CVE-2021-25122**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-25122)

When responding to new h2c connection request, select versions of Apache Tomcat could duplicate request headers and a limited amount of request body from one request could be seen by the other request. Recommend upgrading to later versions of Apache Tomcat.

References:

* CONFIRM - [N/A](https://lists.apache.org/thread.html/r7b95bc248603360501f18c8eb03bb6001ec0ee3296205b34b07105b7%40%3Cannounce.tomcat.apache.org%3E)
* MLIST - [[announce] 20210301 [SECURITY] CVE-2021-25122 Apache Tomcat h2c request mix-up](https://lists.apache.org/thread.html/r7b95bc248603360501f18c8eb03bb6001ec0ee3296205b34b07105b7@%3Cannounce.apache.org%3E)
* MLIST - [[debian-lts-announce] 20210316 [SECURITY] [DLA 2596-1] tomcat8 security update](https://lists.debian.org/debian-lts-announce/2021/03/msg00018.html)

[**CVE-2021-25329**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-25329) suppress

The fix for CVE-2020-9484 was incomplete. When using Apache Tomcat 10.0.0-M1 to 10.0.0, 9.0.0.M1 to 9.0.41, 8.5.0 to 8.5.61 or 7.0.0. to 7.0.107 with a configuration edge case that was highly unlikely to be used, the Tomcat instance was still vulnerable to CVE-2020-9494. Note that both the previously published prerequisites for CVE-2020-9484 and the previously published mitigations for CVE-2020-9484 also apply to this issue.NVD-CWE-noinfo

References:

* CONFIRM - [N/A](https://lists.apache.org/thread.html/rfe62fbf9d4c314f166fe8c668e50e5d9dd882a99447f26f0367474bf%40%3Cannounce.tomcat.apache.org%3E)
* MLIST - [[announce] 20210301 [SECURITY] CVE-2021-25329 Apache Tomcat Incomplete fix for CVE-2020-9484 (RCE via session persistence)](https://lists.apache.org/thread.html/rfe62fbf9d4c314f166fe8c668e50e5d9dd882a99447f26f0367474bf@%3Cannounce.apache.org%3E)
* MLIST - [[debian-lts-announce] 20210316 [SECURITY] [DLA 2596-1] tomcat8 security update](https://lists.debian.org/debian-lts-announce/2021/03/msg00018.html)

## 5. Mitigation Plan

*After interpreting your results from the manual review and static testing, identify the steps to remedy the identified security vulnerabilities for Artemis Financial’s software application.*

Before continuing the development process, we should update all of our dependency libraries, and then perform another Maven-dependency check to see where the issues are standing, and if there are any new issues that arise from these changes. There are instances where implementing patches is going to be needed, but if we update the libraries first, there could potentially be patches and updates in those newer versions.

Works cited:

Limited, C. S. (n.d.). Vulnerabilities and Threats. Retrieved March 21, 2021, from <https://www.cyberteamsecurity.com/vulnerabilities-threats.html#:~:text=Client-based> vulnerabilities place the,of attack, compromise and destruction.&text=Usually, when security attacks are,or server-side computing device.

Pam Greenberg, M. M. (n.d.). Data Security Laws: State Government. Retrieved March 21, 2021, from <https://www.ncsl.org/research/telecommunications-and-information-technology/data-security-laws-state-government.aspx>

Sandoval, K. (2018, November 20). 5 Ways To Hack An API (And How To Defend): Nordic APIs |. Retrieved March 21, 2021, from <https://nordicapis.com/5-ways-to-hack-an-api-and-how-to-defend/>

Secure Coding Guidelines for Java SE. (2020, September 28). Retrieved March 21, 2021, from <https://www.oracle.com/java/technologies/javase/seccodeguide.html>

What is SQL Injection? Tutorial & Examples: Web Security Academy. (n.d.). Retrieved March 21, 2021, from <https://portswigger.net/web-security/sql-injection>